



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

PARAMETRIZACIÓN Y OPTIMIZACIÓN DE UNA RED NEURONAL RECURRENTE APLICADA A LA GENERACIÓN DE TEXTO

Autor: García Muñoz, Venancio

Director: López López, Álvaro

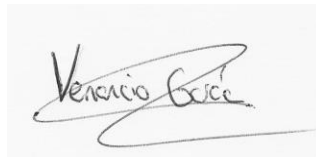
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Parametrización y optimización de una red neuronal aplicada a la generación de texto
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2019/20 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.

Fdo.: Venancio García Muñoz

Fecha:07/07/2020



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Álvaro López López

Fecha: ..07/..07/2020

Agradecimientos

A mi abuelo Juan Manuel

A mi tutor Álvaro López

PARAMETRIZACIÓN Y OPTIMIZACIÓN DE UNA RED NEURONAL RECURRENTE APLICADA A LA GENERACIÓN DE TEXTO

Autor: García Muñoz, Venancio.

Director: López López, Álvaro.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Las redes neuronales artificiales son uno de los métodos de procesamiento y generación de información con una mayor capacidad de desarrollo en el futuro. A lo largo de este proyecto se desarrolla una búsqueda de los mejores valores de los hiperparámetros que componen una red neuronal recurrente, en este caso destinada a la generación de poesía. Los hiperparámetros son las características de una red neuronal artificial establecidas manualmente de forma previa al procesamiento de los datos, que determinan la forma en que la red va a gestionar la información de entrada y que por tanto van a afectar de forma activa al comportamiento de la red. Los valores buscados serán aquellos que optimicen el modelo creado, de forma que permita generar versos con la mayor calidad posible, sin comprometer en exceso la complejidad de la red. Para obtenerlos, será necesario establecer un sistema de evaluación que posibilite calificar los versos. A través de dicho sistema, se evaluará la influencia de los hiperparámetros en los versos generados por la red, atendiendo a la calificación que se ha obtenido para los diferentes valores de los hiperparámetros. Finalmente, se establecerá cuál es la combinación óptima de los hiperparámetros, así como su nivel de influencia final sobre el comportamiento de la red.

Palabras clave: Inteligencia Artificial, Redes Neuronales, Parametrización, Análisis de datos

1. Introducción

La Inteligencia Artificial (IA) es uno de los sectores con un futuro más prometedor dentro de la industria, y debido al gran desarrollo de las capacidades computacionales de los equipos informáticos, es cada vez más común observar la aplicación de la IA en muy diferentes y variados ámbitos.

Dentro de la Inteligencia Artificial se localizan una gran variedad de tecnologías diferentes, que están siendo desarrolladas en mayor medida con el paso del tiempo. Dos de las más importantes son probablemente el aprendizaje automático y el aprendizaje profundo, más conocidos por sus nombres en inglés: Machine Learning y Deep Learning, respectivamente. De acuerdo al científico americano Tom Mitchell [1], la principal cualidad del Machine Learning radica en la mejora de sus prestaciones a través de la experiencia, se trata de un aprendizaje gradual y supervisado. Por su parte, como refleja David Foster en su libro acerca de las aplicaciones del aprendizaje profundo [2], las técnicas de Deep Learning resultan idóneas para inferir dependencias y relaciones entre los datos a estudiar, y hacer uso de esas interconexiones de forma no supervisada.

La mayoría de las técnicas de aprendizaje profundo se corresponden con la creación de redes neuronales artificiales, que consisten en una serie de capas formadas por unidades de procesamiento de la información con varias entradas, una serie de pesos asociados a cada una de las entradas y una única salida que depende tanto de las entradas como los pesos. Estas unidades o celdas son las llamadas neuronas artificiales. Las neuronas se agrupan en capas, y es la unión de varias capas lo que genera una red neuronal artificial. El objetivo del aprendizaje profundo es obtener los pesos ideales para el mejor funcionamiento posible de la red. Esto se lleva a cabo a partir del entrenamiento con el conjunto de datos de entrada, y depende de las diferentes características con las que se ha construido la red neuronal, entre las que se sitúan los hiperparámetros, aspectos de la red establecidos manualmente, y de forma previa al proceso de entrenamiento.

Dentro de las redes neuronales se ubican las redes tipo Long Short-Term Memory (LSTM) que se caracterizan por ser recurrentes: son capaces de retener datos durante un periodo de tiempo, de forma que las salidas anteriores de la red tienen incidencia directa sobre las generadas posteriormente. Este tipo de redes son especialmente interesantes para el manejo de grandes cantidades de datos, como se muestra en el estudio realizado por el científico coreano Donghyun Lee [3]. Precisamente dentro de este tipo de redes se realiza este proyecto, la optimización de una red neuronal recurrente.

2. Definición del proyecto

En este proyecto se busca optimizar el rendimiento de una red LSTM destinada a la generación de poesía, emulando al poeta andaluz Juan Ramón Jiménez. En este contexto, ya se han desarrollado redes neuronales capaces de generar texto, posibles ejemplos son algoritmos que pueden emular a la escritora británica J. K. Rowling con nuevos capítulos de Harry Potter [4], o de imitar los discursos del presidente americano John F. Kennedy [5]. Todas estas redes han requerido de un análisis de los valores de los hiperparámetros óptimos que aseguran el mejor comportamiento del sistema, y es precisamente este el objetivo del proyecto: optimizar la red a partir de la búsqueda de los mejores valores de los hiperparámetros.

Precisamente en el objetivo del proyecto radica la principal dificultad de este: cómo valorar la calidad de los versos. Por ello, uno de los pilares en los que se sustenta la realización de este proyecto es en la creación de un sistema de valoración de versos que permita obtener un resultado numérico con el que establecer una calificación de los versos. Con este sistema es posible determinar cómo la variación de las características a optimizar influye en las salidas de la red neuronal. Una vez delimitadas los patrones de calificación de los versos, se realizará el estudio de los hiperparámetros. Un hiperparámetro es una característica de la red neuronal cuyo valor se determina antes del proceso de entrenamiento, y que por tanto no depende de los datos de entrada del sistema, sino que es establecido por la persona al cargo del desarrollo de la red. Como no se pueden conocer a priori, los valores óptimos de los hiperparámetros serán obtenidos por un estudio prueba y error, a través del análisis de los versos de salida para un determinado valor del hiperparámetro, y haciendo lo propio con un amplio rango de valores para cada uno de los mismos.

3. Descripción del sistema desarrollado

Dado que la mejor forma de evaluar las características de una entidad es a través de valores numéricos, se ha establecido un sistema de evaluación de los versos con el que se persigue cuantificar la calidad de estos. A diferencia de un proceso de clasificación, puramente objetivo, donde el resultado de la red será comparado con el real, y se determinará si es correcto o equívoco; en este caso es necesario señalar que la calificación de versos es un proceso subjetivo, dado que es una persona la que establece su resultado en función de su opinión acerca del mismo. Aun así, con el ánimo de objetivar lo máximo posible un proceso puramente subjetivo, se han establecido una serie de patrones de clasificación de los versos, los cuales se muestran en la Tabla 1 y en la Tabla 2, a través de los que se le concederá a cada verso un resultado numérico, una calificación entre 0 y 10.

Nota Gramatical	Tipo de verso
10	Gramaticalmente correcto
9	Verso con un solo error gramático
8	Entre dos y tres errores, pero mantiene la coherencia
7	Verso coherente, con 4 o 5 errores
6	Verso con varios errores, pero capaz de mantener un sentido claro
5	Se puede intuir el sentido del verso, pese a los errores de gravedad
4	Verso sin coherencia, pero con varias combinaciones correctas de palabras
3	Verso incoherente con varias palabras bien ubicadas
2	Verso sin sentido con alguna combinación de palabras correcta
1	Carente de sentido y palabras aleatorias
0	Verso vacío

Tabla 1: Patrones de valoración gramatical

Nota Artística	Tipo de verso
10	Transmite un sentimiento y emplea algún recurso literario
9	Transmite un sentimiento de forma clara
8	Algún recurso literario bien ejecutado
7	Verso con sentido, pero sin capacidad de transmitir un sentimiento
6	Verso con cierta emoción, aunque pueda resultar incoherente
5	Se puede intuir una emoción tras el verso
4	Se puede intuir alguna figura literaria
3	Alguna palabra que transmite una emoción
2	Alguna combinación de palabras correcta
1	Sin ningún tipo de coherencia
0	Verso vacío

Tabla 2: Patrones de valoración artística

Como se puede comprobar en las tablas anteriores, se ha asignado dos tipos de valoración a los versos del sistema. Esto es debido a que la poesía es un género literario que persigue la transmisión de emociones y sentimientos. Por tanto, a la hora de evaluar los versos se establecerá su calificación gramatical de acuerdo a la corrección del verso escrito (coherencia, género y número correctos, buen uso de las palabras en el contexto...), y su nota artística en función de su capacidad de transmitir una emoción.

A través de este sistema, se realizará una evaluación de la influencia de los hiperparámetros en el comportamiento de la red neuronal recurrente. Existen una gran cantidad de hiperparámetros que rigen el comportamiento de una red, pero se han seleccionado los tres que se consideran más importantes en el caso de una red generativa de texto: la temperatura, el tamaño del embedding y el número de neuronas de la red.

La temperatura es el hiperparámetro que representa el grado de aleatoriedad con la que el modelo ejecuta una predicción, a través de un ajuste en la escala de la función softmax con la que se obtienen las probabilidades de cada palabra en función de la posición en el verso y las palabras anteriores. El word embedding es un espacio vectorial que emplea una red neuronal para ubicar las palabras del diccionario, y de esta forma poder establecer relaciones entre las mismas a través de sus coordenadas; el tamaño del embedding se refiere a la dimensión de dicho espacio vectorial. Por su parte, el número de neuronas es la cantidad de neuronas presentes en cada una de las capas de la red neuronal, y está directamente relacionado con la velocidad y la profundidad de aprendizaje del sistema. A su vez, y de forma previa al análisis de los hiperparámetros anteriormente descritos, se ha calculado el número de capas de la red que optimiza los resultados de los versos.

Además, para seleccionar qué valor de los hiperparámetros es el idóneo, se ha realizado una analogía con el proceso de entrenamiento de una red de clasificación a través de la curva de error. Este sistema diferencia el error en el conjunto de datos entrenamiento, que disminuye a medida que aumenta el nivel de entrenamiento de la red; del error en el conjunto de datos de test, que sin embargo puede llegar a aumentar si se sobreentrena la red para el conjunto de datos de entrenamiento. Mediante la representación gráfica de ambos errores se elegirá el valor con mejor compromiso en los resultados de entrenamiento y test. En el caso de este proyecto, el error de entrenamiento es calculado automáticamente por el sistema al crear cada modelo, en base a la comparación de la predicción de cada palabra, a partir de las anteriormente presentes en el verso, con la palabra que realmente aparece a continuación. Sin embargo, no existe un error de test como tal, puesto que los versos generados no se comparan con nada para establecer su corrección. Por tanto, se ha determinado el error de test a través de la siguiente expresión:
$$Error\ de\ test = \frac{Nota\ Máxima - Nota\ Media}{Nota\ Máxima}$$
, siendo 10 la nota máxima, y la nota media el promedio entre nota gramatical y artística del conjunto de versos evaluados. De esta forma se podrá representar gráficamente la evolución tanto del error de entrenamiento como el de test, y elegir el valor que consiga el mejor compromiso entre calidad y complejidad.

4. Resultados

Para cada uno de los aspectos de la red neuronal anteriormente citados, se ha buscado qué valor optimiza la calidad de los versos obtenidos. Por ello, se ha variado su valor en un amplio rango y se han calificado diez versos para cada uno de los valores elegidos del hiperparámetro, siendo la media aritmética de esas diez calificaciones el valor que establece la calidad del modelo generado.

Se ha elegido evaluar diez versos de cada modelo para alcanzar una cantidad de versos suficiente como para reducir la influencia de valores extremos, pero que a su vez no derive en una cantidad de versos inmanejable.

- **Número de capas de la red**

En función del número de capas de la red se han evaluado versos generados por modelos formados desde una única capa hasta cuatro de ellas, los resultados se muestran en la Ilustración 1.

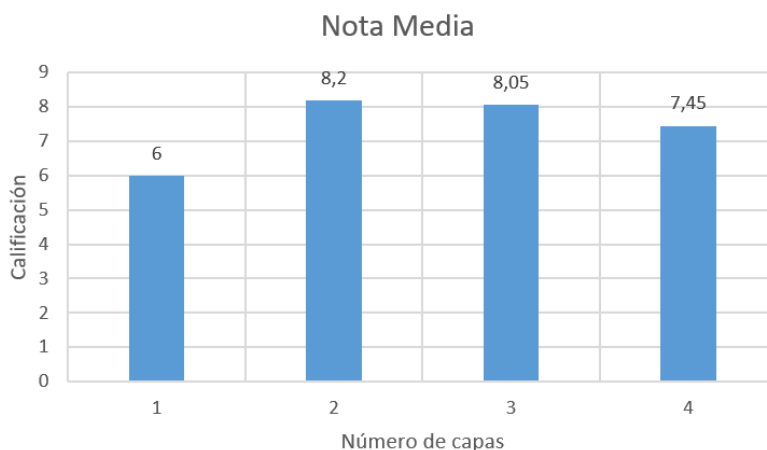


Ilustración 1: Resultados en función del número de capas

Como se puede comprobar, los resultados son mejores si se eligen dos o tres capas, siendo un resultado muy similar el obtenido para estos dos valores. Por tanto, como se trata de un modelo más sencillo con un resultado muy similar, se elige una red de dos capas de neuronas.

- **Temperatura**

Se ha analizado la influencia de la temperatura del sistema de dos formas distintas.

Primeramente, se ha mantenido la temperatura constante a lo largo del proceso de generación del verso. Los resultados se muestran en la Ilustración 2, alcanzando un máximo para una temperatura de valor 1,3.

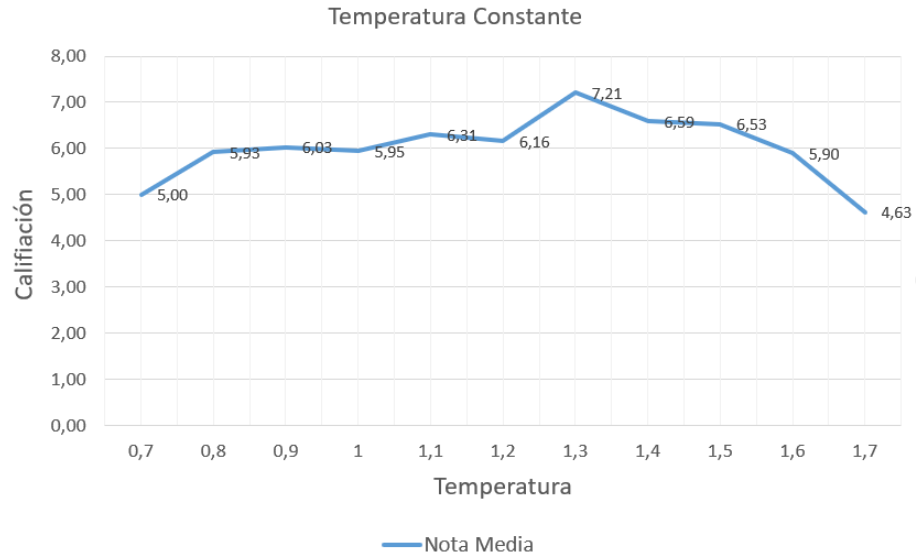


Ilustración 2: Resultados en función de la temperatura constante

Como la temperatura elevada genera problemas en la parte final de los versos, se ha utilizado un valor decreciente de la temperatura a lo largo de la generación de los versos, a través de la siguiente expresión:

$$\text{Temperatura (n)} = \text{Temperatura Inicial} \cdot \text{Decay}^n$$

Donde la temperatura para la palabra n del verso depende del valor inicial de la temperatura y un valor entre 0 y 1, al que se ha llamado decay, que se encarga de disminuirla a medida que se avanza en el verso.

Este tratamiento de la temperatura se basa en la técnica de Simulated Annealing, que se emplea actualmente en gestión de información [6], y consiste en aumentar inicialmente una característica para posteriormente ir reduciéndola gradualmente.

Los resultados obtenidos son óptimos para un valor de temperatura inicial igual a 2,3 y un decay de 0,87, como se muestra en la Ilustración 3.

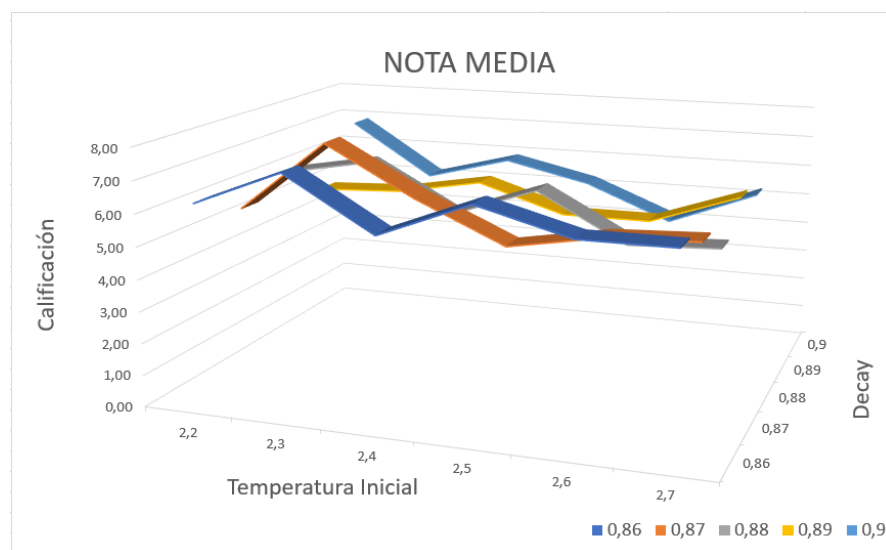


Ilustración 3: Resultados en función de la temperatura decreciente

- **Tamaño del embedding**

En función del tamaño del embedding se han evaluado versos generados por modelos creados con un tamaño de embedding desde 25 hasta 600. El error, tanto el calculado por el modelo o error de entrenamiento, como el obtenido a partir de la valoración de los versos, al que se ha llamado error de test, se ha representado en función del tamaño del embedding en la Ilustración 4, con el objetivo de buscar gráficamente el resultado que garantiza unos mejores resultados, tanto en calidad como en complejidad del modelo.

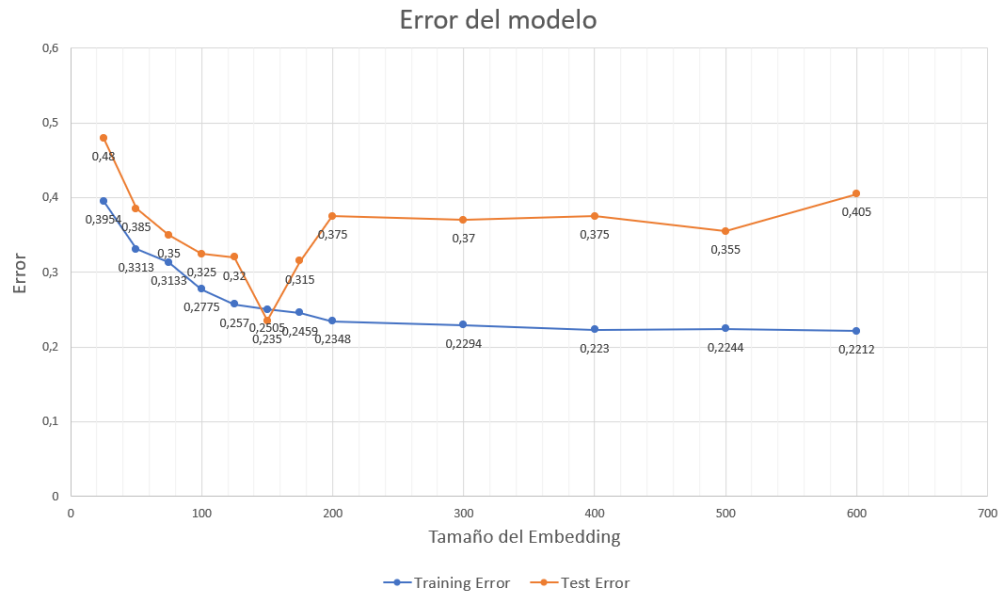


Ilustración 4: Error en función del tamaño del embedding

Cabe recalcar que se trata de una analogía, además de que los datos del error de test provienen de una clasificación subjetiva, por lo que puede presentar incongruencias, como por ejemplo que aparezca un valor del tamaño del embedding donde el error del test es menor que el de entrenamiento, algo que en un problema de clasificación sería imposible. Pese a ello, los resultados obtenidos al visualizar la gráfica son evidentes: el valor óptimo para el tamaño del embedding es 150, puesto que para valores inferiores la red no está lo suficientemente entrenada, mientras que para valores superiores el sistema tiene problemas de sobre entrenamiento.

En este caso el valor del mínimo error es evidente, además de ser un valor de embedding lo suficientemente comedido como para no comprometer en exceso la complejidad del sistema, por lo que se elige 150 como tamaño óptimo de este hiperparámetro.

- **Número de neuronas**

Se ha analizado, de la misma forma en la que se ha hecho el estudio del hiperparámetro anterior, cómo afecta el número de neuronas al comportamiento de la red neuronal recurrente. En la Ilustración 5 se muestra la evolución del error en función del número de neuronas.

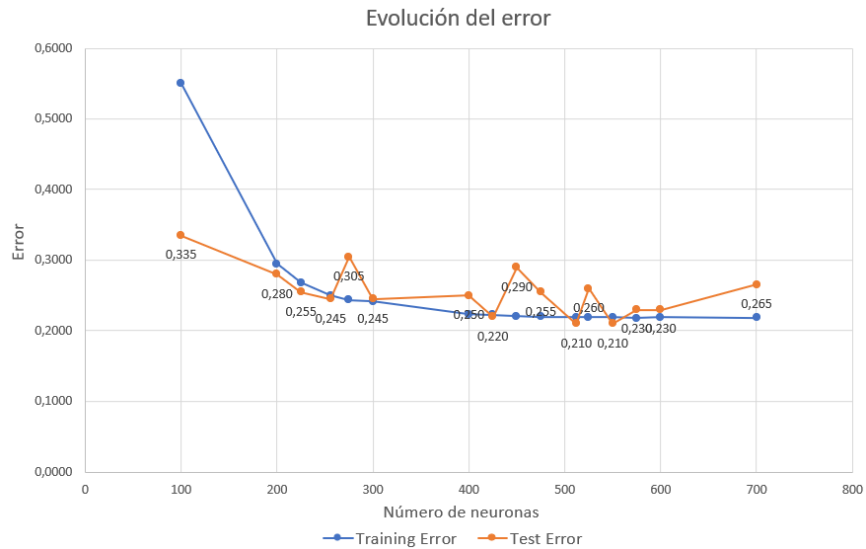


Ilustración 5: Error en función del tamaño del número de neuronas

En este caso y a diferencia del anterior, no existe un valor evidente en el que los resultados sean claramente mejores que en el resto. Se puede apreciar una zona, para valores del número de neuronas inferior a 200, donde al aumentar el valor del hiperparámetro el resultado mejora: el entrenamiento del modelo no es suficiente. También hay otra zona, para valores de más de 550 neuronas, donde aumentar el valor supondrá una disminución de los resultados obtenidos: el sistema está sobre entrenado. En el rango existente entre 256 y 512 neuronas los resultados fluctúan, manteniéndose alrededor de un valor de error de 0,25. Por tanto, el valor óptimo elegido para el número de neuronas es 256, ya que, aunque los resultados sean ligeramente peores que para 512, son lo suficientemente parejos y aseguran un mejor compromiso entre la calidad de los versos obtenidos y la complejidad del modelo.

5. Conclusiones

El principal objetivo de este proyecto era obtener los valores que permitieran alcanzar los mejores resultados para una red neuronal recurrente destinada a la generación de texto. Una vez se han obtenido estos mismos, se han utilizado todos los resultados óptimos de cada uno de los hiperparámetros para la creación de un mismo modelo.

La influencia de los valores de los hiperparámetros en la calidad de los versos resultantes es evidente. Para comprobarlo basta con observar la diferencia entre los primeros diez versos analizados antes de llevar a cabo la optimización de la red, que se muestran en la Ilustración 6, con los diez resultantes después del análisis para los valores óptimos de los hiperparámetros, que se muestran en la Ilustración 7.

```
amigos es agrupadas describidme . una ilusión . decirlo ?  
contacto fundó en nombre casi une ,  
ascienden lo ved esa mismo rosas monedas ,  
arpegio de amor locas acaba en llamamiento ;  
no me granito tápame allí columnillas impasible juzgó súplica tirabuzones durmieron arribamos cánticos .  
aleves como espuma de a vida tirabuzones .  
alza aplicasteis sino uniendo céspedes busca golpeado ,  
trémulamente sentado y humedecido complace .  
engañas , en cambio , mayor modulaba flora que tú tabla  
musical inerme a condesa que par o nuestra requerido sumos ios vagaba predice ondeante iris
```

Ilustración 6: Primeros versos generados por la red

```
rojiza miradme el pecho .  
ninfas o miradas ,  
estos pasos en la arena .  
dejadme fuisteis , emergido por la luz como una cera ;  
alguien , hace ya mucho , hace muchísimo , puso con su mano más :  
el rosa está de materia  
alzó falda revuelto remotas últimas , más misma que un sol .  
mi enamorada súbita .  
indemne en peregrina su voz ,  
descifrará acaso bloque sobre finura de dicha .  
aquí olor silencioso aquí color de unos ojos ;
```

Ilustración 7: Versos generados por el modelo optimizado

Se puede comprobar perfectamente como los versos del modelo optimizado tienen unas características gramaticales y artísticas realmente superiores a las de los versos originales, que eran prácticamente ilegibles. El análisis de los hiperparámetros ha mejorado notablemente la calidad de los versos de la red.

No obstante, entre los versos del modelo optimizado también existen errores gramaticales, así como versos sin excesiva capacidad de transmitir emociones. Por lo que la red neuronal recurrente todavía presenta capacidad de mejora. Para ello, se ha analizado a través de qué posibles caminos se podría continuar con el optimizado de la red.

Una de las posibles vías de mejora se encuentra en variar los datos con los que se ha llevado a cabo el entrenamiento del sistema, anteriormente versos sueltos en forma de secuencias. Se ha optado por la implementación de estrofas en lugar de versos como nuevas secuencias de entrada de la red. El objetivo de esta decisión es comprobar que la elección de los hiperparámetros es generalizable a una red neuronal de generación de texto pese a que varíe el formato de los datos de entrada, y conseguir una mayor conexión entre los versos generados por el modelo que, como se ha podido comprobar en la Ilustración 7, no guardaban ningún tipo de relación entre sí. Un ejemplo de la nueva generación en forma de estrofas se muestra en la Ilustración 8.

```
pues le no veía . ¡ oh , yo ,  
no , no es eso . no es un hombre .  
no es un hombre . mas no es un hombre . es la noche .  
y lo mas lo esperando . y es lo que es lo mismo .  
es una sombra que cae y está el mar .  
no es el hombre .  
no es un hombre . ¿ es un niño ? ¿ quién es un hombre ?  
no es un lecho de niño que pasa .  
la noche es un día . ¿ quién qué amo ?  
¿ quién soy tú ?
```

Ilustración 8: Estrofa generada por el sistema

Los resultados resultan satisfactorios, porque los versos están ligados entre sí y la gran mayoría de ellos presenta una gramática impoluta. No obstante, los errores presentes en la generación de versos sueltos siguen presentes al cambiar al uso de estrofas, por lo que se debe seguir afinando la gramática de la red.

Otra posible vía de desarrollo, que en este caso se centraría en lograr la perfección gramatical, sería la implementación de una red generativa adversaria, más conocida como GAN, muy utilizadas actualmente en la generación y tratamiento de imágenes, como muestran los estudios realizados por J. Gauthier [7] y B.Dolhansky [8]. Estas redes están formadas por la contraposición de un generador y un clasificador, la función del primero será crear versos que emulen ser reales, mientras que la función del segundo será determinar qué versos son artificiales y cuáles no lo son.

De seguirse esta vía de mejora, la red desarrollada a lo largo de este proyecto debe ejercer de generador, y se tiene que desarrollar un clasificador. El cual recibirá como entradas versos, y tendrá que decidir si se trata de un verso generado por el modelo o si por el contrario se trata de un verso real. En este tipo de redes neuronales, la práctica hace mejorar tanto a generador como a clasificador de forma simultánea, puesto que el generador tiene que aumentar su capacidad de emular los versos originales para lograr su objetivo de engañar al clasificador, mientras que este último debe desarrollarse de forma que el generador no sea capaz de hacerse pasar por los versos reales.

6. Referencias

- [1] T. M. Mitchell, "Machine Learning", libro de la editorial McGraw-Hill Science/Engineering/Math, 1 de marzo de 1997, pp: 1-5
- [2] D. Foster, "Generative Deep Learning. Teaching Machines to Paint, Write, Compose and Play", libro publicado por O`Reilly Media Inc, 26 de junio de 2019, pp: 37-42
- [3] D. Lee et al., "Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus," artículo publicado por China Communications, vol. 14, no. 9, pp. 23-31, Sept. 2017, doi: 10.1109/CC.2017.8068761, septiembre 2007, pp: 23-31.
- [4] S. Ferro, "An ai program wrote harry potter fan fiction and the results are hilarious", artículo para la revista digital Mental Floss, 13 diciembre 2017, último acceso: 24/06/2020 <https://www.mentalfloss.com/article/520897/ai-program-wrote-harry-potter-fan-fiction-and-results-are-hilarious>
- [5] A. Rogan, "Kennedy's "lost" Dallas speech in his own voice", artículo para el diario The Times, 15 de marzo de 2018.
- [6] C. Wang y Z. Li, "Inversion of Wave Impedance Using Improved Simulated Annealing Genetic Algorithm," artículo publicado en la Tercera Conferencia Internacional sobre Computación y Gestión de Información, IEEE, 2010, pp: 121-124.
- [7] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation", artículo académico, Stanford University: Symbolic Systems Program, Natural Language Processing Group
- [8] Br. Dolhansky y C. Canton Ferrer, "Eye In-Painting with Exemplar Generative Adversarial Networks", 2018 IEEE/CVF artículo original de la Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 7902-7911, doi: 10.1109/CVPR.2018.00824.

PARAMETRIZATION AND OPTIMIZATION OF A RECURRENT NEURAL NETWORK APPLIED TO TEXT GENERATION

Author: García Muñoz, Venancio.

Supervisor: López López, Álvaro.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

Artificial neural networks are one of the methods of processing and generating information with the greatest capacity for development in the future. Throughout this project, a search for the best values of the hyperparameters that make up a recurrent neural network is being developed, in this case for poetry generation. Hyperparameters are the characteristics of an artificial neural network established manually prior to data processing, which determine the way in which the network will manage the input information and will therefore actively affect the behavior of the network. The values sought will be those that optimize the model created, so that it allows verses to be generated with the highest possible quality, without excessively compromising the complexity of the network. To obtain them, it will be necessary to establish an evaluation system that makes it possible to grade the verses. Through this system, the influence of the hyperparameters on the verses will be evaluated, considering the qualification that has been obtained for the different values of the hyperparameters. Finally, from the different evaluations obtained, the results will be analyzed and the optimal combination of the hyperparameters will be established, as well as their final level of influence on the behavior of the network.

Key Words: Artificial Intelligence, Neural Networks, Parametrization, Data Analysis

1. Introduction

Artificial Intelligence (AI) is one of the sectors with the most promising future within the industry, and due to the great development of the computational capacities of computer equipment, it is increasingly common to observe the application of AI in very different fields.

Within Artificial Intelligence there are a great variety of different technologies, which are being developed to a greater extent over time. Two of the most important are probably Machine Learning and Deep Learning. According to the American scientist Tom Mitchell [1], the main quality of Machine Learning lies in the improvement of its performance through experience, it is a gradual and supervised learning. On the other hand, as reflected by David Foster in his book about the applications of Deep Learning [2], Deep Learning techniques are ideal for inferring dependencies and relationships between the data, and making use of these interconnections in an unsupervised way.

Most of the Deep Learning techniques correspond to the creation of artificial neural networks, which consist of a series of layers formed by information processing units. Those units have several inputs, a series of weights associated to each of the inputs and a single output that depends on both the inputs and the weights. These units or cells are called artificial neurons. Neurons are grouped in layers, and it is the union of several layers that generates an artificial neural network. The objective of the Deep Learning is

to obtain the ideal weights for the best possible functioning of the network, and this is carried out by the training of the network with the input data set, and depends on the different characteristics with which the neuronal network has been built, among which are the hyperparameters.

Hyperparameters are the aspects of the network established manually before the training process. A very important type of neural network is the Long Short-Term Memory (LSTM), that are characterized by being recurrent: they are capable of retaining data over a period of time, so that the previous outputs of the network have a direct impact on those generated later. This type of network is especially interesting for handling large amounts of data, as shown in the study by Korean scientist Donghyun Lee [3]. It is precisely within this type of network that this project, the optimization of a recurrent neural network, is carried out.

2. Project Definition

This project seeks to optimize the performance of an LSTM network for the generation of poetry, emulating the Spanish poet Juan Ramon Jimenez. In this context, neuronal networks capable of generating text have already been developed, possible examples being algorithms able to emulate the British writer J. K. Rowling with new chapters of Harry Potter [4], or imitating the speeches of the American President John F. Kennedy[5]. All these networks have required an analysis of the values of the optimum hyperparameters that ensure the best behavior of the system, and this is precisely the aim of the project: to optimize the network by searching the best values of the hyperparameters. Precisely in the objective of the project lies the main difficulty of this one: how to value the quality of the verses. For this reason, one of the pillars on which this project is based is the creation of a system for evaluating verses that allows a numerical result to be obtained with which to establish a rating of the verses. With this system it is possible to determine how the variation of the characteristics to be optimized, influences the outputs of the neural network.

Once the verse rating patterns have been defined, the study of the hyperparameters will be carried out. A hyperparameter is a characteristic of the neural network whose value is determined before the training process, and therefore does not depend on the input data of the system, but it is established by the person in charge of the development of the network. As they cannot be known before the training, the optimal values of the hyperparameters will be obtained by a trial and error study, through the analysis of the output verses for a given hyperparameter value, and doing the same with a wide range of values for each of them.

3. Developed System

Given that the best way to evaluate the characteristics of an entity is through numerical values, a system of evaluation of the verses has been established with the aim of quantifying the quality of the verses. Unlike a classification process, which is purely objective, and where the results of the network will be compared with the real ones, and then it will be determined whether it is right or wrong; in this case it is necessary to point out that the classification of verses is a subjective process, since it is a person who establishes a result based on his or her opinion about it. Even so, with the aim of objectifying as much as possible a purely subjective process, a series of patterns of classification of the verses have been established, which are shown in Table 1 and Table 2, through which each verse will be given a numerical result, a score between 0 and 10.

Grammar Score	Verse
10	Grammatically correct
9	Verse with a single grammatical error
8	Only two or three errors, but maintaining consistency
7	Coherent verse with four or five errors
6	Verse with several errors, but able to maintain a clear sense
5	The sense of the verse can be intuited, despite the errors of gravity
4	Verse without coherence, but with several correct word combinations
3	Inconsistent verse with several well-placed words
2	Nonsense verse with some correct word combination
1	Lack of meaning and random words
0	Empty verse

Table 1: Grammar assessment patterns

Artistic Score	Verse
10	It conveys a feeling and employs some literary resource
9	It conveys a feeling in a clear way
8	Some well-executed literary resource
7	Verse with meaning, but without the ability to convey a feeling
6	Verse with some emotion, although it may be inconsistent
5	You can sense an emotion behind the verse
4	You can sense some literary figure
3	Some word that conveys an emotion
2	Some combination of words right
1	Verso Without any consistency
0	Empty verse

Table 2: Artistic assessment patterns

As it can be seen from the tables above, two types of valuation have been assigned to the verses in the system. This is because poetry is a literary genre that seeks to convey emotions and feelings. Therefore, when evaluating the verses, their grammatical qualification will be established according to the correctness of the written verse (coherence, correct genre and number, good use of the words in context...), and their artistic note according to their capacity to transmit an emotion.

Through this system, an evaluation of the influence of the hyperparameters on the behavior of the recurrent neural network will be made. There are a large number of hyperparameters that govern the behavior of a network, but the three that are considered most important in the case of a text-generative network have been selected: the temperature, the size of the word embedding and the number of neurons in the network.

The temperature is the hyperparameter that represents the degree of randomness with which the model executes a prediction, through an adjustment in the scale of the softmax function. With the softmax function the probabilities of each word are obtained according to the position in the verse and the previous words. The word embedding is a vector space that uses a neural network to locate the words in the dictionary, and thus be able to establish relationships between them through their coordinates. And the size of the embedding refers to the dimension of this vector space. For its part, the number of neurons is the number of neurons present in each of the layers of the neural network and it is directly related to the speed and the depth of learning of the system. Prior to the analysis of the previously described hyperparameters, the number of network layers has been calculated to optimize the results of the verses.

In addition, to select which value of the hyperparameters is ideal, an analogy has been made with the training process of a classification network through the error curve. This system differentiates the error in the training data set, which decreases as the training level of the network increases, from the error in the test data set, which can however increase if the network is over trained for the input data set. The graphical representation of both errors will help to choose the value with the best compromise between training and test results. In this project, the training error is automatically calculated by the system when creating each model, based on the comparison of the prediction of each word, from those previously present in the verse, with the word that actually appears below. However, there is no test error as such since the verses generated are not compared with anything to establish their correction. Therefore, the test error has been determined through the following expression: $Test\ Error = \frac{Max\ Score - Mean\ Score}{Max\ Score}$, where 10 is the maximum score and the mean score is the arithmetic mean between the grammatical and artistic note of the set of verses evaluated. In this way it will be possible to represent graphically the evolution of both the training and test errors, and to choose the value that achieves the best compromise between quality and complexity.

4. Results

For each of the aspects of the neural network mentioned above, we have sought to find out what value optimizes the quality of the verses obtained. To this end, their value has been varied over a wide range and ten verses have been rated for each of the chosen values of the hyperparameter, the arithmetic mean of these ten ratings is the value that establishes the quality of the model generated.

Ten verses of each model have been evaluated to reach enough verses to reduce the influence of extreme values, but which in turn do not result in an unmanageable number of verses.

- **Number of layers**

Depending on the number of layers in the network, verses generated by models formed from a single layer to four layers have been evaluated, the results are shown in Figure 1.

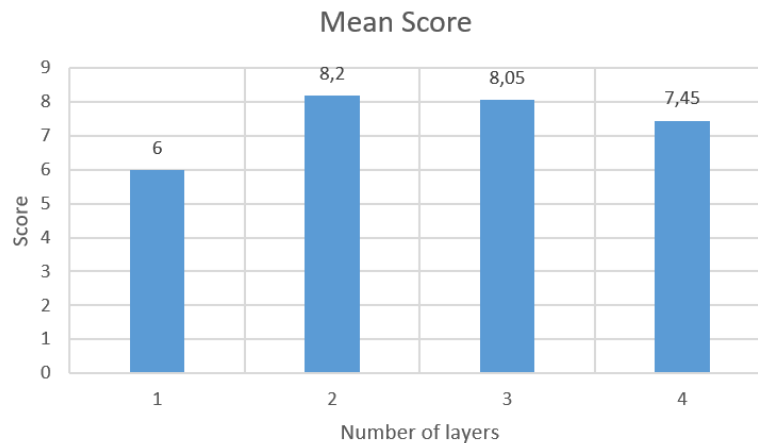


Figure 1: Results according to the number of layers

As you can see, the results are better if two or three layers are chosen, with a very similar result being obtained for these two values. Therefore, as this is a simpler model with a very similar result, a network of two layers of neurons is chosen.

- **Temperature**

The influence of the system temperature has been analyzed in two different ways. Firstly, the temperature has been kept constant throughout the process of generating the complete verse. The results are shown in Figure 2, reaching a maximum for a temperature value of 1.3.

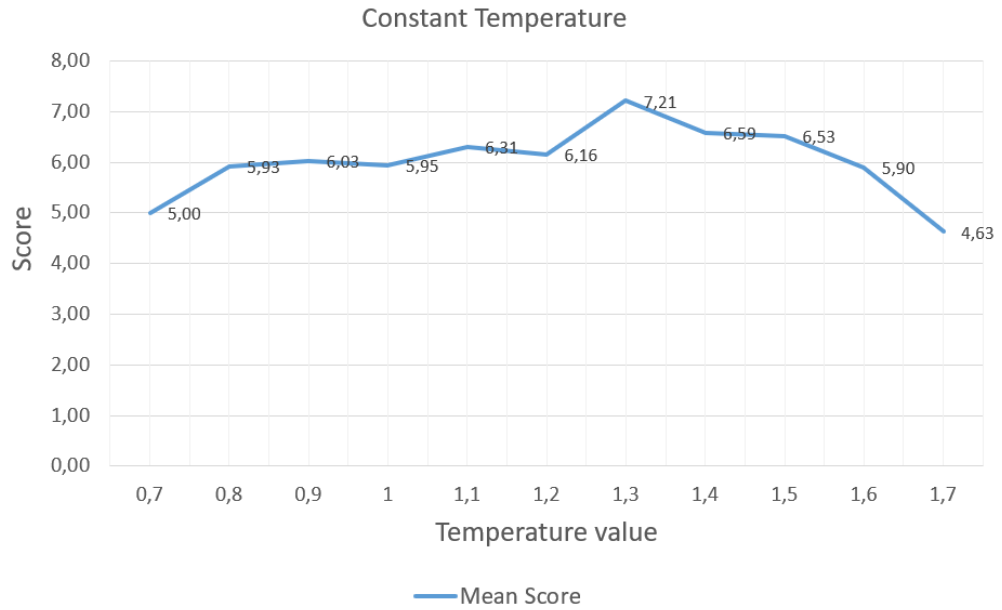


Figure 2: Results depending on constant temperature

As the high temperature generates problems in the final part of the verses, a decreasing value of the temperature has been used throughout the generation of them, through the following expression:

$$\text{Temperature (n)} = \text{Initial Temperature} \cdot \text{Decay}^n$$

Where the temperature for the n word in the verse depends on the initial value of the temperature and a value between 0 and 1, which has been called the decay, and it is responsible for decreasing the temperature as the verse proceeds.

This treatment of the temperature is based on the Simulated Annealing technique, which is currently used in data analysis [6], and consists of initially increasing a characteristic and then gradually reducing it.

The results obtained are optimal for an initial temperature value of 2.3 and a decay of 0.87, as shown in Figure 3.

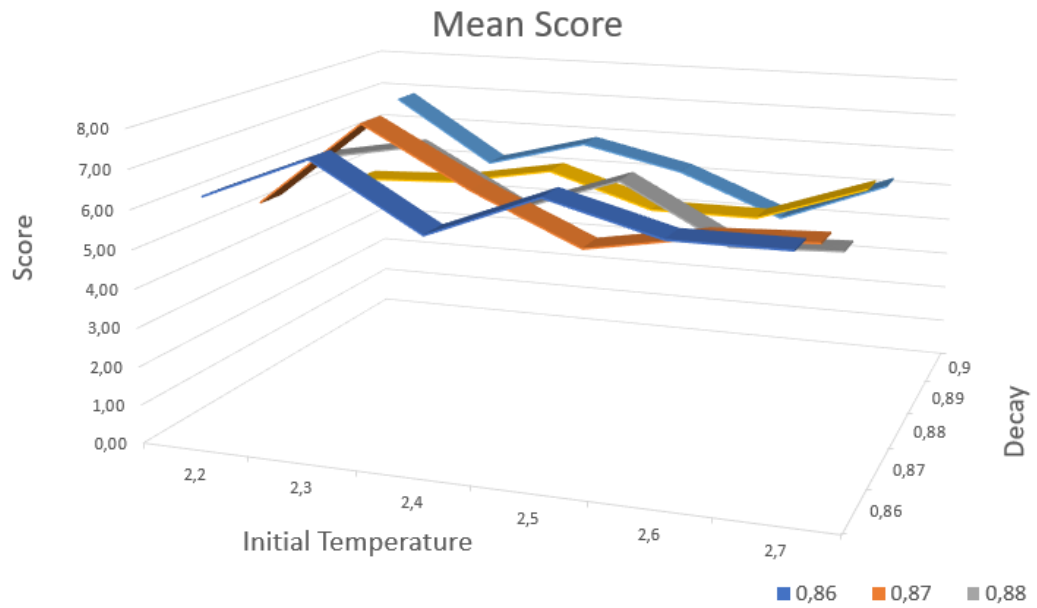


Figure 3: Results depending on decreasing temperature

- **Embedding Size**

Depending on the embedding size, verses generated by models with an embedding size from 25 to 600 have been evaluated. The error, both that calculated by the model (training error), and the one obtained from the evaluation of the verses (test error), have been represented as a function of the size of the embedding in Figure 4, with the aim of graphically seeking the result that guarantees better results, both in quality and complexity of the model.



Figure 4: Error depending on the embedding size

It is important to emphasize that this is an analogy, besides the fact that the test error data come from a subjective classification, so it can present inconsistencies, such as the appearance of an embedding size value where the test error is lower than the training error, something that would be impossible in a classification problem. Despite this, the results obtained when viewing the graph are evident: the optimum value for the size of the embedding is 150, since for lower values the network is not sufficiently trained, while for higher values the system has problems of over-training. In this case the value of the minimum error is evident, as well as being an embedding value that is sufficiently measured so it does not compromise the complexity of the system too much, which is why 150 is chosen as the optimal size for this hyperparameter.

- **Number of Neurons**

In the same way as the study of the previous hyperparameter, how the number of neurons affects the behavior of the recurrent neural network has been analyzed. The Figure 5 shows the evolution of the error depending on the number of neurons.

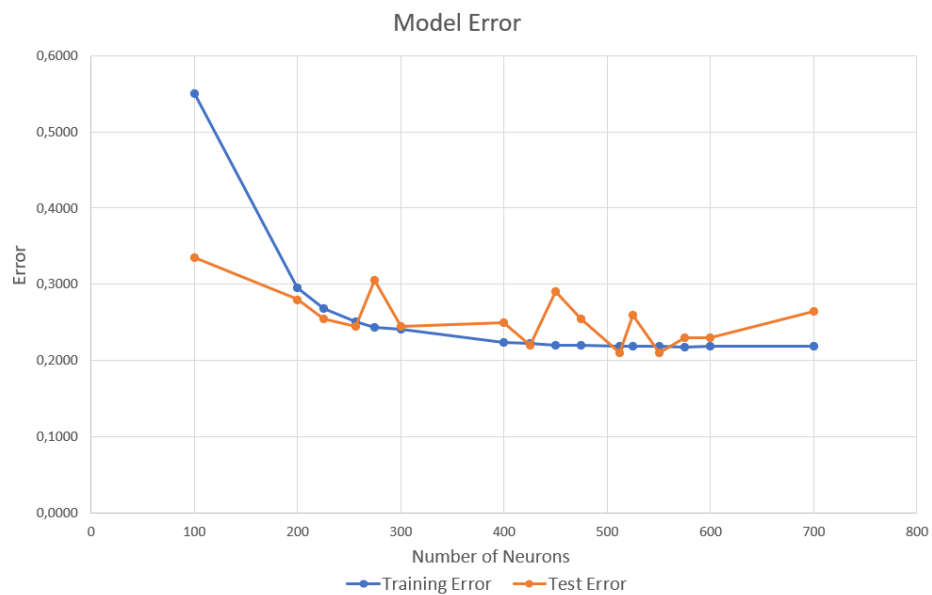


Figure 5: Error depending on the number of neurons

In this case and unlike the previous one, there is no evident value in which the results are clearly better than in the rest. It is possible to appreciate a zone, for values of the number of neurons lower than 200, where increasing the hyperparameters the result improves: the training of the model is not sufficient. There is also another zone, for values higher than 550 neurons, where increasing the value involves a reduction in the results obtained: the system is over-trained. In the range between 256 and 512 neurons the results fluctuate, remaining around an error value of 0.25. Therefore, the optimal value chosen for the number of neurons is 256, since, although the results are slightly worse than for 512, they are sufficiently similar and ensure a better compromise between the quality of the verses obtained and the complexity of the model.

5. Conclusions

The main objective of this project was to obtain the values that would allow the best results to be achieved for a recurrent neural network for text generation. Once these were obtained, the optimal results of each of the hyperparameters were used to create a model.

The influence of the values of the hyperparameters on the quality of the resulting verses is evident. Analyzing the final verses, it is perfectly possible to see how the verses of the optimized model have improved grammatical and artistic characteristics than the original ones, which were practically unreadable. The analysis of the hyperparameters has notably improved the quality of the verses of the network. However, among the verses of the optimized model there are also grammatical errors, as well as verses without excessive ability to convey emotions. Therefore, the recurrent neural network still has room for improvement. For this reason, we have analyzed which possible paths could continue with the optimization of the network.

One of the possible ways of improvement is to vary the data with which the system training has been carried out, previously verses in the form of sequences. The implementation of stanzas has been chosen instead of single verses as new network input sequences. The aim of this decision is to check that the choice of hyperparameters can be generalized to a text generation neural network if the input data vary, and to achieve a greater connection between the verses generated by the model that previously were not related to each other.

The results of the model created using stanzas are satisfactory, because the verses are linked to each other and most of them have an impeccable grammar. However, the errors that were formerly present are still there when changing to the use of stanzas, so the grammar of the network must be further refined.

Another possible route of development, which in this case would focus on achieving grammatical perfection, would be the implementation of a generative adversarial network (GAN) which is widely used in the generation and processing of images, as shown by the studies carried out by J. Gauthier[7] and B.Dolhansky[8]. These networks are formed by the contraposition of a generator and a classifier, the function of the first one will be to create verses that emulate the real ones, while the function of the second one will be to determine which verses are artificial and which are not.

If this path of improvement is followed, the network developed throughout this project must act as a generator, and a verse classifier must be developed, which will receive as inputs verses and will have to decide whether it is a verse generated by the model or if on the contrary it is a real verse. In this type of neural network, practice makes both generator and classifier improve simultaneously, since the generator has to increase its capacity to emulate the original verses to achieve its objective of deceiving the classifier, while the classifier has to be developed in such a way that the generator is not able to impersonate the original verses.

References

- [1] T. M. Mitchell, "Machine Learning", McGraw-Hill Science/Engineering/Math, March 1997, pp: 1-5
- [2] D. Foster, "Generative Deep Learning. Teaching Machines to Paint, Write, Compose and Play", O'Reilly Media Inc, 26/06/2019, pp: 37-42
- [3] D. Lee et al., "Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus," China Communications, vol. 14, no. 9, pp. 23-31, Sept. 2017, doi: 10.1109/CC.2017.8068761, septiembre 2007, pp: 23-31.
- [4] S. Ferro, "An ai program wrote harry potter fan fiction and the results are hilarious", magazine paper, Mental Floss, December 2017, last access: 24/06/2020
<https://www.mentalfloss.com/article/520897/ai-program-wrote-harry-potter-fan-fiction-and-results-are-hilarious>
- [5] A. Rogan, "Kennedy's "lost" Dallas speech in his own voice", paper in The Times, March 2018.
- [6] C. Wang and Z. Li, "Inversion of Wave Impedance Using Improved Simulated Annealing Genetic Algorithm," Third International Conference on Information and Computing, IEEE, 2010, pp: 121-124.
- [7] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation", academic paper, Stanford University: Symbolic Systems Program, Natural Language Processing Group
- [8] Br. Dolhansky y C. Canton Ferrer, "Eye In-Painting with Exemplar Generative Adversarial Networks", 2018 IEEE/CVF paper from Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 7902-7911, doi: 10.1109/CVPR.2018.00824.

Índice de la memoria

Capítulo 1. Introducción	5
Capítulo 2. Descripción de las Tecnologías.....	9
Capítulo 3. Estado de la Cuestión	15
Capítulo 4. Definición del Trabajo	19
4.1 Justificación.....	19
4.2 Objetivos	20
4.3 Metodología	21
4.4 Planificación y Estimación Económica.....	22
Capítulo 5. Sistema Desarrollado	25
5.1 Valoración cuantitativa de los versos	26
5.1.1 Valoración gramatical.....	27
5.1.2 Valoración Artística	28
5.2 Número de Capas de la Red	30
5.3 Parámetros e Hiperparámetros	33
5.4 Temperatura	36
5.4.1 Temperatura Constante	38
5.4.2 Temperatura Decreciente.....	41
5.5 Tamaño del Embedding.....	51
5.5.1 Obtención del Valor Óptimo del Tamaño del Embedding.....	54
5.5.2 Error en Función del Embedding.....	57
5.6 Número de Neuronas.....	60
5.6.1 Obtención del Valor Óptimo del Número de Neuronas.....	61
5.6.2 Error en Función del Número de Neuronas.....	65
5.7 Generación de estrofas	68
Capítulo 6. Análisis de Resultados.....	69
6.1 Resultado del número de capas de la red.....	69
6.2 Resultado de la temperatura	70

6.3	Resultado del tamaño del Embedding	72
6.4	Resultado del número de neuronas	73
Capítulo 7. Conclusiones y Trabajos Futuros		75
Referencias: Bibliografía		79
ANEXO I: Objetivos de Desarrollo Sostenible		83
ANEXO II: Código Fuente		86

Índice de figuras

Figura 1: Tendencias de la inteligencia artificial[4]	7
Figura 2: Diferencias entre IA, Machine Learning y Deep Learning [8]	10
Figura 3: Funcionamiento de una neurona artificial.....	11
Figura 4: Estructura de una red neuronal [11].....	12
Figura 5: Estructura de una red neuronal recurrente [12].....	12
Figura 6: Diferencias entre distintos tipos de red neuronal [14]	13
Figura 7: Cronograma del proyecto Robot Poeta	22
Figura 8: Jerarquía de una red neuronal [27].....	30
Figura 9: Tipos de capas en una red neuronal [27].....	31
Figura 10: Ubicación de los hiperparámetros en el funcionamiento de la red	35
Figura 11: Representación gráfica del análisis de temperatura constante	39
Figura 12: Fases del análisis de la temperatura variable	43
Figura 13: Representación gráfica de la influencia del decay	45
Figura 14: Representación gráfica de la influencia de la temperatura inicial	47
Figura 15: Influencia conjunta del decay y la temperatura inicial en la nota gramatical	49
Figura 16: Influencia conjunta del decay y la temperatura inicial en la nota artística	50
Figura 17: Influencia conjunta del decay y la temperatura inicial en la nota media	50
Figura 18: Ejemplo de word embedding	51
Figura 19: Word Embedding de tres dimensiones	52
Figura 20: Primer análisis del Embedding	55
Figura 21: Análisis gráfico detallado del tamaño del embedding	56
Figura 22: Error en el conjunto de entrenamiento y de test.....	58
Figura 23: Primer análisis del número de neuronas	62
Figura 24: Análisis detallado del número de neuronas.....	63
Figura 25: Evolución del error con el número de neuronas	66
Figura 26: Primeros diez versos analizados	76
Figura 27: Versos del modelo optimizado.....	77
Figura 28: Estrofa obtenida con el algoritmo	77

Índice de tablas

Tabla 1: Patrones de valoración gramatical.....	27
Tabla 2: Patrones de valoración artística.....	28
Tabla 3: Resultados en función del número de capas.....	31
Tabla 4: Diferencias entre parámetros e hiperparámetros.....	34
Tabla 5: Resultados temperatura constante.....	38
Tabla 6: Valor de la temperatura en función del Decay.....	42
Tabla 7: Influencia del decay en los versos.....	44
Tabla 8: Influencia de la temperatura inicial en los versos.....	46
Tabla 9: Calificación gramatical con decay y temperatura inicial variables.....	48
Tabla 10: Calificación artística con decay y temperatura inicial variables.....	48
Tabla 11: Nota media con decay y temperatura inicial variables.....	49
Tabla 12: Resultado del primer análisis del embedding.....	54
Tabla 13: Análisis detallado del tamaño del embedding.....	56
Tabla 14: Error en función del embedding.....	58
Tabla 15: Primer análisis del número de neuronas.....	61
Tabla 16: Análisis detallado del número de neuronas.....	63
Tabla 17: Error de entrenamiento y de test en función del número de neuronas.....	65

Capítulo 1. INTRODUCCIÓN

El desarrollo tecnológico es probablemente el factor que ha tenido mayor relevancia en la evolución que ha experimentado el mundo durante los últimos tiempos. Siendo por tanto la tecnología uno de los pilares más importantes en los que se apoyan las sociedades actuales, y nada parece indicar que vaya a disminuir su importancia sino más bien todo lo contrario.

Para entender su relevancia, se debe comenzar por comprender qué es la tecnología. En su artículo para la revista online Economipedia, Paula Roldán define la tecnología como “*el conjunto de conocimientos y técnicas que se aplican de manera ordenada para alcanzar un determinado objetivo o resolver un problema*”, a lo que añade que “*La tecnología es una respuesta al deseo del hombre de transformar el medio y mejorar su calidad de vida*”[1]. Atendiendo a esta definición, la tecnología es fruto del deseo humano de hacer progresar a la sociedad.

Por tanto, desde el día en que el primer ser humano descubrió la utilidad que podría tener una piedra como elemento destinado a la caza, el hombre ha utilizado la tecnología, por lo que esta ha evolucionado relacionándose a diferentes sectores a lo largo de la historia humana. Si nos centramos en el presente, probablemente sea en el sector informático en el que se encuentre la mayor capacidad de desarrollo tecnológico de cara al futuro. El avance de las capacidades de procesamiento de los equipos de trabajo ha habilitado una nueva forma de gestionar la información, y la estandarización de la tecnología puntera ha hecho de la programación y alta computación algo accesible para prácticamente toda la población, no solo para las grandes compañías que puedan realizar enormes desembolsos en equipos de alta gama. El desarrollo de este tipo de tecnología sigue siendo uno de los principales focos de inversión, como se puede comprobar con los primeros ordenadores cuánticos [2], miles de veces más potentes a los actuales.

Este gran avance en la capacidad de computación se ve reflejado en el auge de los servicios de código abierto que se encuentran por la red, donde cada vez más programadores comparten y ponen al servicio del mundo sus avances y descubrimientos.

Es este acercamiento de la tecnología al gran público el que ha permitido un crecimiento exponencial en el avance, ya que el desarrollo en paralelo siempre será más rápido que el desarrollo individual por la mayor cantidad de personas y mentes disponibles. Aun así el software de código abierto ha encontrado oposición en importantes sectores, pero han sido unas críticas que actualmente parecen equívocas, como demuestra la rectificación de una compañía del calibre de Microsoft al referirse a sus críticas en el pasado [3].

De entre todas las capacidades que ofrece el desarrollo computacional, una de las más interesantes y que hoy en día ocupa un lugar privilegiado en el interés científico es la inteligencia artificial. Un sector que va aumentando su red de aplicaciones y que por ello es cada vez más habitual encontrar su representación en diversos sectores. Eduardo Fernández, en un artículo para el diario El Mundo, establece los sectores con una mayor influencia de la inteligencia artificial [4], donde se puede contrastar que la mitad de su mercado está ocupado por el área de la salud (21%) las finanzas (18%) y la comunicación o entretenimiento (12%).

Sin embargo, áreas como la educación se quedan con un porcentaje muy inferior. Esto puede referirse a la complejidad existente en aplicar esta tecnología al sector, o bien a que el rédito económico es realmente inferior al que pudiera dar una inversión en la aplicación de la inteligencia artificial a sectores como puedan ser las finanzas.

Con todo ello, se puede hablar de las principales tendencias de la inteligencia artificial, aquellos sectores en los que ha adquirido un gran peso en el mercado y donde sus aplicaciones han sido adoptadas en gran medida a la hora de ejecutar la actividad en cuestión. Estas tendencias precisamente se muestran en la Figura 1.

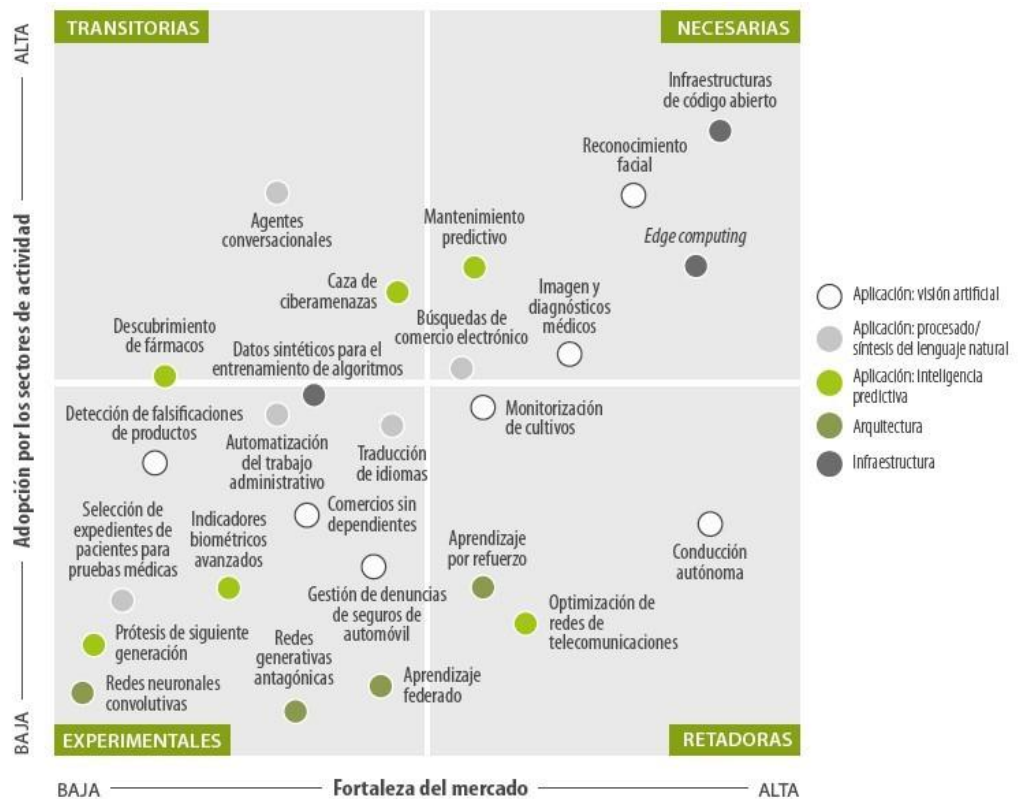


Figura 1: Tendencias de la inteligencia artificial[4]

Dentro de todos los ámbitos de la inteligencia artificial, este proyecto se sitúa dentro de las redes neuronales, un sector que como se puede apreciar en la gráfica anterior no se sitúa entre aquellos en los que la inteligencia artificial goce de una mayor fortaleza de mercado. Todo ello supone que se trata de un ámbito con unas grandes posibilidades de desarrollo y evolución. Esto, unido a la gran diversidad de áreas de aplicación de las redes neuronales, hacen de este campo uno de aquellos en los que más camino hay por recorrer para implementarlas en la sociedad del futuro.

Y es precisamente dentro de este ámbito donde se ubica el proyecto: una red neuronal capaz de desarrollar por sí misma una labor que hasta ahora ha sido única y exclusivamente capaz de hacer por los hombres, como lo es escribir. Y más concretamente, escribir poesía. Esto supone un verdadero reto para la sociedad de hoy en día: establecer hasta qué punto un algoritmo es capaz de imitar ciertos comportamientos humanos. Este debate está en continuo desarrollo, más aun con el avance exponencial de la tecnología, y da lugar a continuos

análisis de la situación como pudieran ser los realizados por David Gámez [5] o Cristóbal Fuentes Barassi [6] acerca de la capacidad de una máquina de experimentar emociones o tener conciencia. Este proyecto es una forma de aplicar los avances en la tecnología, en un nuevo intento de imitar un comportamiento humano como es la creación artística. También por esta razón se ha elegido la poesía como género literario a desarrollar, puesto que una de las características del género es la capacidad de transmitir sentimientos o emociones.

De esta forma, la motivación que lleva a la realización de este proyecto es conseguir que una red neuronal de generación de texto consiga producir una poesía en castellano, capaz de transmitir emociones y sentimientos. Para ello se analizará qué aspectos de una red neuronal van a ser los encargados de establecer una mejor imitación del comportamiento humano, y estudiar su influencia en los modelos. Por todo ello este proyecto va a consistir en un análisis, parametrización y optimización de una red neuronal capaz de escribir poesía. Con el objetivo de aplicar los conocimientos adquiridos sobre la programación y la estadística en analizar cuáles son los parámetros óptimos para el mejor funcionamiento posible de la red.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

Una de las características fundamentales de la tecnología es la constante evolución, transformación y desarrollo que sufre, en todos sus ámbitos. Sectores que recientemente han sido punteros, y donde se encontraban las mayores inversiones, presupuestos e intereses, pueden transformarse rápidamente en obsoletos si aparece una nueva tecnología que los sustituye, y en el momento en que esta nueva tecnología los sobrepase será en ella donde se invierta y apueste de cara al futuro. En este contexto, uno de los sectores tecnológicos que parece hoy en día tener un futuro más brillante es la inteligencia artificial.

Evidentemente, la inteligencia artificial es un campo realmente abierto, en el que se incluyen muchos y muy diversos ámbitos de avance tecnológicos, entre ellos, uno de los más importantes es el conocido como aprendizaje automático, aprendizaje automatizado o aprendizaje de máquinas, aunque sin duda es su nombre en inglés el que resulta más utilizado para nombrarlo: Machine Learning. Esta rama de la inteligencia artificial tiene como principal objetivo el aprendizaje de las máquinas, un aprendizaje que llega derivado de la experiencia. Para ello, hay que utilizar la definición que dio Tom Mitchell al Machine Learning: *“Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y medida de rendimiento P , si su rendimiento en tareas en T , medido por P , mejora con la experiencia”*[7]. De acuerdo con esta definición, la particularidad que define el machine Learning como una rama de la inteligencia artificial es precisamente el hecho de que el aprendizaje deriva de la experiencia, cuanto mayor experiencia adquiera una máquina, mejor será su rendimiento. Sin embargo, ese aprendizaje deberá ser guiado por una persona, es un aprendizaje supervisado: las máquinas necesitan un entrenador humano.

Como alternativa, dentro del propio Machine Learning se ha desarrollado la tecnología conocida como Deep Learning o aprendizaje profundo, que hereda del Machine Learning el aprendizaje a través de la experiencia pero se diferencian en un aspecto fundamental, la inexistencia de un entrenador humano que supervise el aprendizaje de la máquina: en este caso es el propio sistema quien aprende por sí solo.

Además, al igual que ocurre en Machine Learning, en Deep Learning las bases de datos deben ser lo más grandes posible, para que la máquina mejore sus resultados a través de la experiencia, por lo que el aumento de la masa de datos de entrenamiento supondrá un mayor aprendizaje.

En la Figura 2 se representa cómo se engloban estas tres importantes ramas del desarrollo tecnológico.

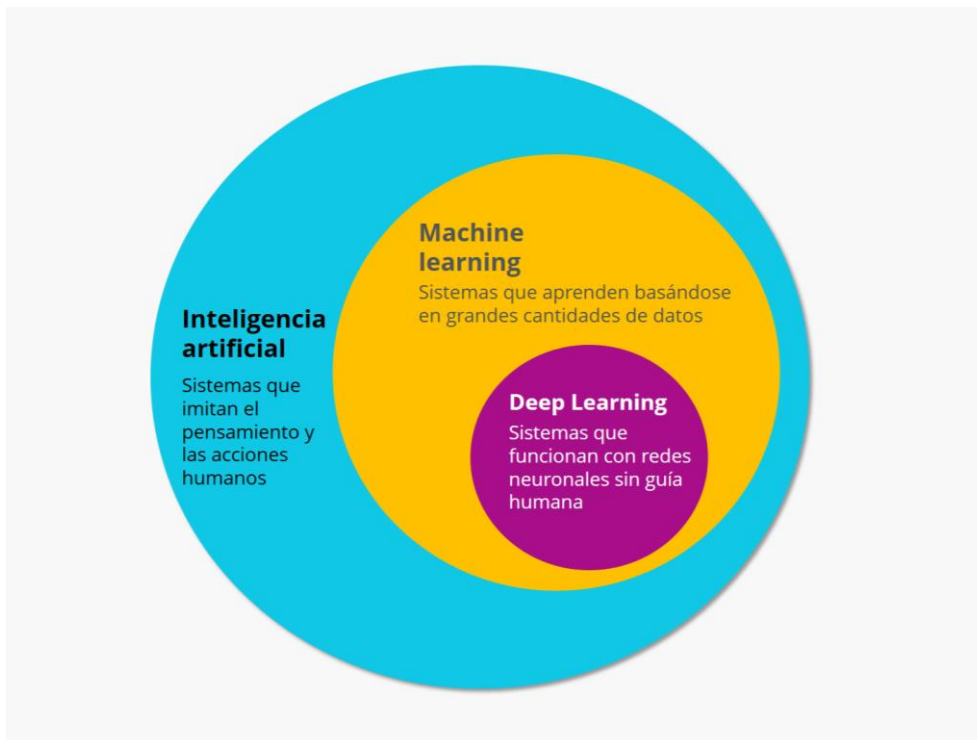


Figura 2: Diferencias entre IA, Machine Learning y Deep Learning [8]

Este proyecto se va a desarrollar a través de técnicas de Deep Learning, que se basan precisamente en el manejo de redes neuronales artificiales.

El nombre de red neuronal proviene de su inspiración en el funcionamiento del cerebro, y está formado por una cantidad variable, que depende del tipo de red que se desarrolle, de nodos que reciben y transmiten la información. Estos nodos se denominan neuronas artificiales, y su característica principal radica en el hecho de que pueden recibir una cantidad alterable de datos de entrada, pero tan solo tienen una única salida, que a su vez puede ser conectada a varias neuronas diferentes.

Dentro de cada neurona, se establece una función que asocia unos pesos determinados a cada una de esas entradas y que permitirá establecer el valor de la salida en función de los valores de entrada. En la Figura 3 se muestra el funcionamiento de cada uno de los nodos de la red neuronal.

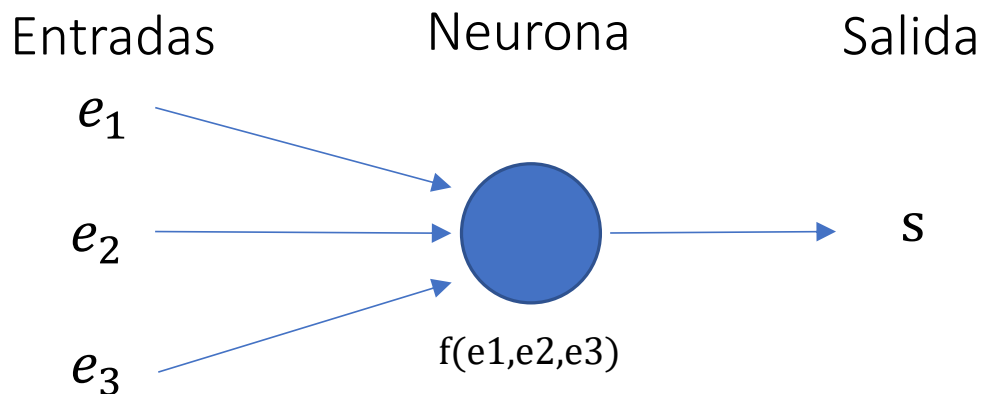


Figura 3: Funcionamiento de una neurona artificial

Lo realmente importante es la red de neuronas, que incluye una gran cantidad de nodos interconectados entre sí a través de los que se transmite la información. Estas neuronas estarán agrupadas en niveles o capas, la complejidad de la red neuronal crece exponencialmente con el aumento del número de capas.

Aunque no se va a estudiar a fondo a lo largo de la elaboración de este proyecto, es necesario recalcar la gran importancia que tiene el tipo de arquitectura elegida para el uso de las redes neuronales artificiales, como desarrolla Laurene Fausett en su libro *Fundamentals of Neural Networks* [9], un estudio de los tipos de redes, su arquitectura y su aplicación en la industria. En la Figura 4 se puede apreciar la estructura general de una red neuronal en la que cada uno de los nodos recibe unos valores de entrada y a través de los pesos específicos que se asignan a esas entradas, elabora un valor de salida que se transmitirá a través de la red, hasta obtener un valor final de salida, que dependerá del tipo de datos que esté manejando la red. Este es el concepto de perceptrón, que es explicado por Simon Haykin en su libro *Neural Networks and Learning Machines* [10].

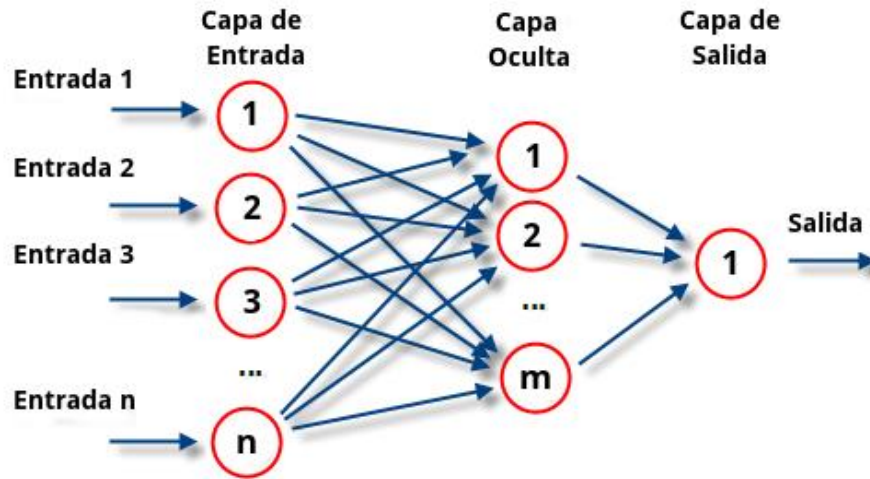


Figura 4: Estructura de una red neuronal [11]

Dentro de las redes neuronales, este proyecto se va a enfocar en el desarrollo de un tipo particular de red neuronal: una red neuronal recurrente. Este tipo de redes es especialmente adecuado para manejar datos secuenciales, debido a que utilizan la salida anterior como un valor de entrada de la nueva iteración. Mientras que en una red neuronal clásica la información de salida de cada iteración es olvidada una vez es emitida, en una red neuronal recurrente esta información persiste durante un tiempo, por lo que influye directamente en el valor de salida que se obtiene de la red durante las posteriores iteraciones. En la Figura 5 se puede observar a la perfección el citado comportamiento: las salidas de la red neuronal denominadas o_t se añaden a los valores de x_t como entradas del siguiente bloque: $t+1$.

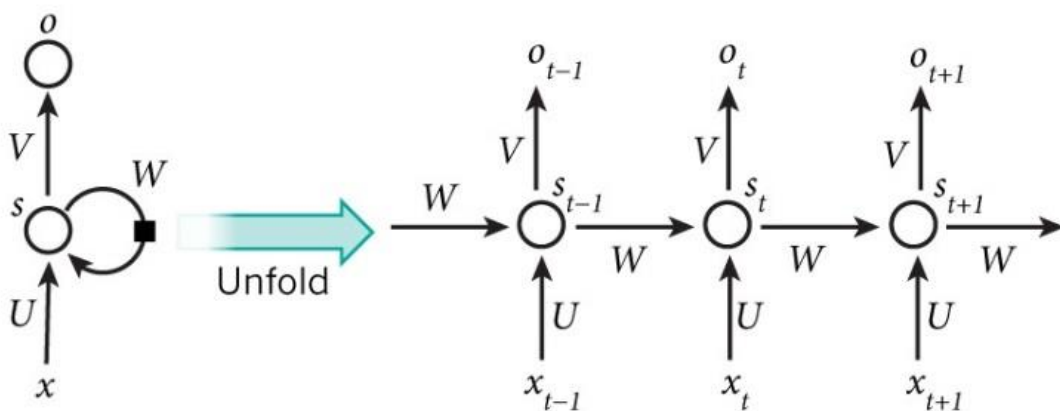


Figura 5: Estructura de una red neuronal recurrente [12]

Dentro de las redes neuronales recurrentes, se sitúan las redes Long Short-Term Memory, o redes LSTM. La principal diferencia respecto a las redes neuronales básicas radica en el hecho de que las redes LSTM tienen tanto memoria a corto plazo, características que comparten con las básicas, como memoria a largo plazo: son capaces de conservar durante un tiempo prolongado la información que consideren relevante. Son especialmente utilizadas para el manejo de grandes cantidades de datos de entrada, como refleja el estudio realizado por el profesor de matemáticas coreano Donghyun Lee [13].

Esta gran característica la consiguen gracias a la aparición de un tipo de celda adicional: la celda de estado. Esta celda de estado contiene la información más relevante que la red ha encontrado a lo largo del proceso de entrenamiento de la red, y a esta celda se puede acceder en pleno proceso de entrenamiento tanto para eliminar como para añadir o actualizar la información que contiene.

La celda de estado resulta de especial utilidad en el procesamiento de texto, ya que permitirá a la red establecer conexiones y relaciones de género y número en las frases, permitiendo al algoritmo generar texto con una gramática correcta. Por ello, se ha elegido este tipo de red para desarrollar la red de este proyecto de generación de poesía, dado que es el tipo de red óptimo para el procesamiento de texto.

En la Figura 6 se muestran las principales diferencias entre los tres tipos de red neuronal explicados previamente, en el caso de la red LSTM el rectángulo superior se corresponde con la celda de estado en la que, a diferencia de las redes básicas o recurrentes, se almacena la información más relevante de la red.

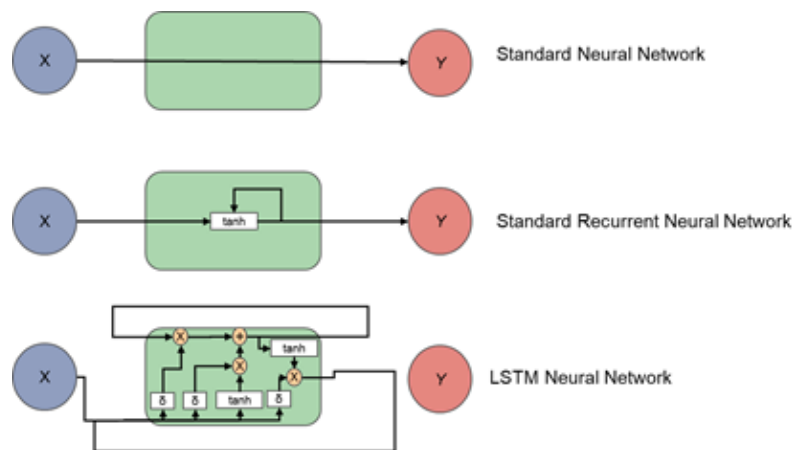


Figura 6: Diferencias entre distintos tipos de red neuronal [14]

Capítulo 3. ESTADO DE LA CUESTIÓN

Como se ha indicado previamente, uno de los sectores con mayor crecimiento en la actualidad es sin duda la inteligencia artificial. Es cada vez más común ver la aplicación de técnicas de inteligencia artificial en muchos y muy diversos ámbitos, la sociedad hace uso de ella sin prácticamente darse cuenta en el día a día.

Sin duda uno de los sectores con mayor aplicación de la inteligencia artificial es el sector empresarial, donde cada vez más empresas se suman a aplicar estas técnicas con el objetivo de automatizar en la mayor medida posible las labores. Aun así, todavía mantiene una gran capacidad de mejora en el futuro, ya que muchas de las posibles implementaciones que permite la inteligencia artificial están aún por explorar. Esto, que en otros sectores, como podrían ser aquellos más técnicos, imposibilitaría su aplicación hasta no tener una cierta base de seguridad a la hora de aplicar una técnica, no supone un problema en el ámbito empresarial, pues como justifica Wendy B. Rauch-Hindin en su libro *Análisis de la Inteligencia Artificial en la Actividad Empresarial la Ciencia y la Industria* [15], si cualquier técnica de automatización es rentable puede ser aplicada inmediatamente y no debe esperar hasta que se profundice en su conocimiento, ya que será productiva desde el primer momento.

No obstante, la aplicación de técnicas de inteligencia artificial en la ciencia está cada vez más a la orden del día, puesto que ya se conoce cada vez más en detalle el funcionamiento y las enormes posibilidades que nos ofrece. El mejor ejemplo de ello es la aplicación de la tecnología para diagnosticar enfermedades raras [16], donde la inteligencia artificial permite detectar los síntomas a través del tratamiento masivo de datos. En este mismo sector también se emplean técnicas de inteligencia artificial para detectar la gravedad de las lesiones cerebrales como se cita en la noticia de *El Confidencial* [17], donde se refleja que las enfermedades cerebrales utilizan una cantidad inmensa de datos que genera una información difícilmente manejable para personas pero que a través de los algoritmos puede ser aprovechada al máximo.

Si centramos el foco en el Machine Learning, la rama de la inteligencia artificial a la que se vincula este proyecto, se aprecia que probablemente el sector donde tiene un mayor peso sea la seguridad, tanto como protección frente a virus informáticos como en la regulación de las personas en focos de acumulación como podrían ser aeropuertos o estaciones. Pero cada vez se pueden encontrar aplicaciones en una mayor diversidad de sectores, como se refleja en la lista de las mejores Startups que aplican Machine Learning [18] donde las compañías aplican esta tecnología en sectores que van desde la ciencia de la salud, hasta la asistencia en la compra de productos.

Aumentando de nuevo el foco en el proyecto dentro del Machine Learning, es en el Deep Learning donde se ubica la red neuronal de generación de poesía. Como ya se ha indicado en el Capítulo 2. se trata de una forma de aprendizaje de los algoritmos no supervisado, por lo que resulta idóneo para trabajar con grandes masas de datos que resultan inmanejables para las personas pero que sin embargo, para una máquina que utilice el Deep Learning tal cantidad de volumen de información resulta beneficioso. También se están aplicando estas técnicas en el sector de la medicina, el ejemplo más reciente es la búsqueda de patrones que indiquen la presencia del Covid-19 a través de la creación de una gran base de datos de pacientes, tanto infectados como no infectados. Una posible vía de análisis es a través de la grabación de la tos de las personas [19], que permitiría establecer si una persona está o no infectada por el virus a partir de un simple muestreo de su tos.

Pero sin duda alguna, una de las áreas de negocio en las que se está utilizando en mayor medida el Deep Learning es precisamente en la detección y clasificación de imágenes. Se trata de redes neuronales recurrentes capaces de reconocer objetos en imágenes y clasificarlas en función de su contenido. Un ejemplo de ello es el estudio sobre la capacidad de un algoritmo de detectar emociones en imágenes de personas reales realizado por Adrián Sáez de la Pascua [20], en el que concluye que a medida que aumenta la base de datos a partir de la cual realizar el aprendizaje de la red neuronal, el Deep Learning mejora considerablemente las prestaciones del algoritmo respecto a los resultados obtenidos empleando técnicas propias del Machine Learning.

Sin embargo, a diferencia de los ejemplos anteriores en los que el objetivo principal era la correcta detección y clasificación de los datos, en este proyecto se ha implementado la tecnología Deep Learning a la generación artificial, y en concreto, de una obra literaria. En este ámbito, el estado del arte se encuentra en la creación de redes neuronales que son capaces de imitar a un artista, ya sea escritor, músico, pintor... y crear una obra original, puesto que no ha sido creado antes, es algo único, pero que provoque en el receptor la sensación de haber sido creada por dicho artista, y no por un algoritmo. Precisamente acerca de este tema discurre la investigación de David Foster [21], quien estudia la capacidad de las redes neuronales como generadores de obras artísticas.

Bajo esta perspectiva, hoy en día ya se pueden encontrar ejemplos de la aplicación de estas tecnologías en la generación de texto. Una referencia clara es la inteligencia artificial capaz de escribir una historia alternativa del personaje de Harry Potter [22], el personaje creado por la famosa escritora británica J. K. Rowling. Este algoritmo ha sido entrenado precisamente con las 7 novelas que componen la saga, y ha sido capaz de escribir una obra que recuerda a la escritora británica en la forma de narrar y describir, pero cuyo principal problema está en la inconsistencia e incoherencia de la historia. Cabe señalar que la gran mayoría del desarrollo de la generación artificial de texto con técnicas de Deep Learning, como el propio ejemplo anteriormente citado, se ha llevado a cabo en inglés, por lo que en este proyecto habrá que adaptar la red neuronal a las particularidades de un idioma tan diferente al inglés como es el castellano, donde es menos común encontrar ejemplos de creación de una obra literaria. En este contexto, hay que señalar el estudio realizado por Claudio Verdú [23], sobre un generador de texto en castellano, en el que se analiza la corrección gramatical de los textos generados.

Además, en este proyecto se ha elegido una rama de literatura especialmente interesante: la poesía, que se diferencia de la prosa en esa capacidad de transmitir sentimientos, una capacidad muy vinculada al hombre, al que ahora va a tener que sustituir un algoritmo, con la dificultad añadida que eso supone.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

El uso creciente de las redes neuronales se demuestra en el incremento en el número de proyectos que trabajan con esta tecnología. En este sentido, en todos ellos han tenido que establecer unos valores para sus hiperparámetros, que serán los encargados de definir el comportamiento de la red. Aquí radica la principal razón por la que se ha desarrollado este proyecto: se busca establecer una forma de obtener resultados óptimos de una red neuronal a través del estudio de los hiperparámetros.

Más concretamente, al tratarse de una red de generación de texto, los resultados y las conclusiones serán aplicables a cualquier variación en el artista o género literario de los versos a generar, bastará con cambiar la base de datos de entrada a la del estilo o personaje literario por aquel a quien se desee emular.

Además, como se ha ido razonando en los apartados anteriores, las redes neuronales son un método de gestión de la información y de los datos en auge. Por tanto, el análisis de este proyecto puede ser aplicado de forma análoga en cualquier otra labor que conlleve una interacción con una red neuronal, evidentemente estará más relacionada con otra red neuronal de generación de texto, pero también podrá aplicarse en otros sectores, adaptando los hiperparámetros al tipo de red neuronal en cuestión.

Finalmente, este proyecto plantea un método de valoración y clasificación que permite establecer una graduación en los resultados de forma que se pueda cuantificar y aplicar un análisis de los datos ante resultados no numéricos, en este caso versos. Esta clasificación y la forma empleada al analizar los versos puede ser empleada de forma equivalente con otro tipo de redes de generación, como podría ser el caso de imágenes o piezas musicales.

4.2 OBJETIVOS

El proyecto persigue varios objetivos, relacionados de forma estrecha con la motivación que lleva a ponerlo en práctica.

El primero de ellos es evidente: conseguir desarrollar una red neuronal capaz de imitar la capacidad humana de la escritura y que efectivamente tenga la capacidad de escribir una poesía que logre engañar a un lector externo, no siendo este capaz de distinguir el origen del verso resultante. En este caso, se ha elegido la figura del poeta Juan Ramón Jiménez, por lo que los versos creados por el algoritmo deben emular el estilo literario del escritor andaluz.

De forma global a lo largo de todo el proyecto, se persigue el acercamiento y la familiarización a las técnicas de aplicación y desarrollo de Deep Learning. Dado que es una tecnología puntera, el acercamiento a esta permitirá a su vez entender el comportamiento de otro tipo de redes neuronales, cada vez más presentes en la industria.

El objetivo central del proyecto es estudiar y analizar el peso de los distintos parámetros e hiperparámetros que definen la estructura de una red neuronal recurrente. Dichos valores serán los encargados de diferenciar los modelos que se van a entrenar, y cada uno de ellos tiene un objetivo específico dentro del modelo. A su vez, será de gran interés analizar la relación que guardan entre sí, y prestar especial atención al análisis del efecto que puede tener en la calidad los versos resultantes la posible correlación que exista entre los parámetros del modelo. Para ello se utilizarán herramientas estadísticas, y se desarrollará un modelo de evaluación de versos que permita clasificar y diferenciar la influencia de los parámetros.

Este sistema de clasificación también deriva de uno de los objetivos existentes a lo largo del desarrollo del proyecto: asignar un valor numérico que represente la calidad de las salidas de la red neuronal y que por tanto permita valorar de forma cuantitativa el efecto de las variables de la red sobre el resultado de los versos generados.

4.3 METODOLOGÍA

A lo largo de la realización de este proyecto, se va a seguir la siguiente metodología.

En primer lugar, será necesario establecer una forma de valorar la influencia de las modificaciones realizadas en los hiperparámetros sobre los resultados de las salidas generadas por la red. Como las salidas son versos, se desarrollará un método que permita cuantificar la calidad de los versos. Pese a ser subjetivo, se buscará establecer una serie de reglas de valoración a las que se ceñirá la clasificación de los versos.

Una vez se haya desarrollado dicho sistema de valoración, se procederá a analizar los distintos aspectos de la red neuronal que se pretenden optimizar. Dicho análisis se realizará hiperparámetro por hiperparámetro de forma separada e independiente, es decir, cuando se estudie la influencia de cada uno se mantendrán el resto constantes.

Como no se conoce en qué valores se encuentran los óptimos de forma previa a la realización del análisis, se estudiarán los hiperparámetros partiendo de rangos muy amplios, para realizar un primer acercamiento. A partir de los resultados de este primer estudio, se aumentará el foco en aquella zona en la que se hayan obtenido mejores resultados, reduciendo el rango de valores de estudio y ampliando la resolución al analizar valores más cercanos entre sí. Con este segundo estudio se determinarán qué valores arrojan los mejores resultados, así como cuáles aseguran un mejor compromiso entre calidad de los versos y complejidad de la red.

Finalmente se comprobará si la optimización de los parámetros ha resultado efectiva comprobando las diferencias existentes entre los primeros versos generados por la red antes del análisis, con aquellos generados por la red para la combinación idónea de valores que se ha obtenido en el estudio. Y atendiendo a los resultados obtenidos, se plantearán métodos de perfeccionamiento y afinado de la red, así como futuras implementaciones y posibles trabajos derivados de este análisis.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

El desarrollo de este proyecto se adhiere a la labor realizada por Carlos Geuens Álvarez, quien ha desarrollado el estudio sobre la creación de la base de datos, el diccionario y funcionamiento de la red LSTM que será la encargada de elaborar la poesía. Este proyecto se centraba en perfeccionar, a través de la optimización de los hiperparámetros, el comportamiento de la red originalmente creada por Carlos.

La unión de ambos proyectos da como resultado el robot poeta, que sigue el cronograma que se ha mostrado en la Figura 7, separando las labores realizadas por cada uno de los responsables a lo largo de estos meses de trabajo.

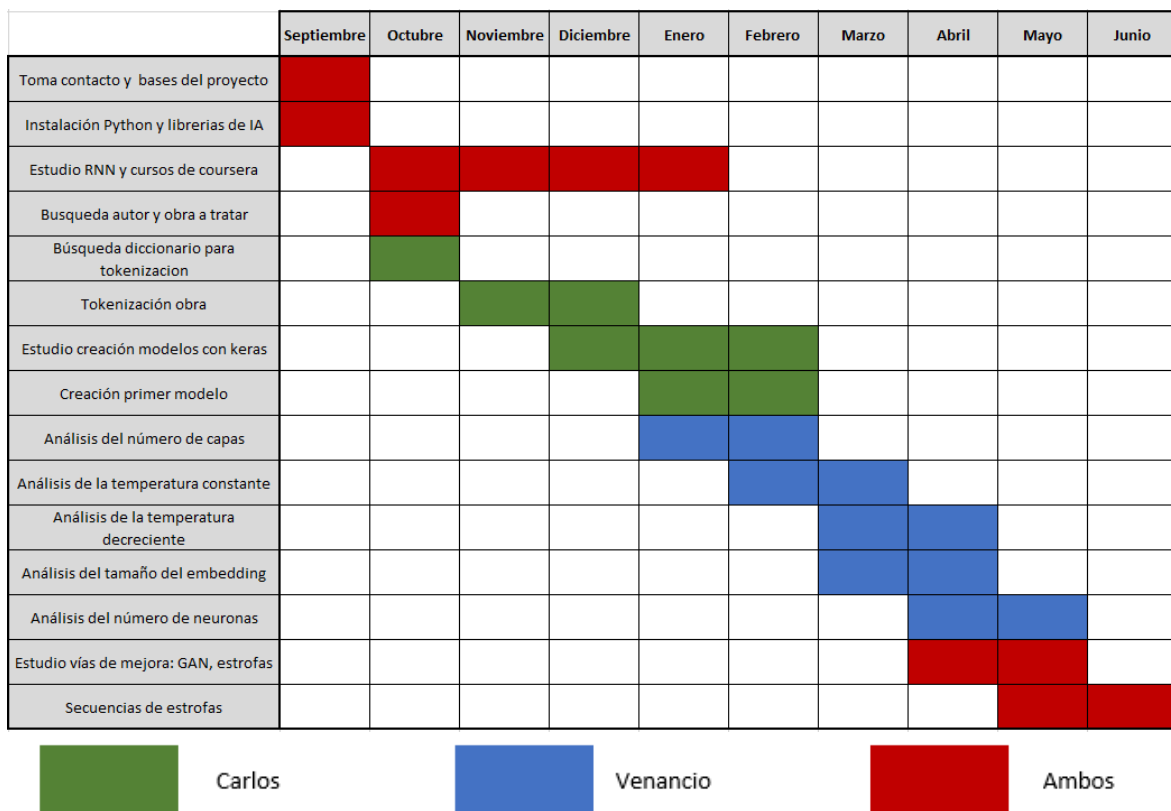


Figura 7: Cronograma del proyecto Robot Poeta

En cuanto a la estimación económica, en el desarrollo de este proyecto se va a emplear el lenguaje de programación Python, más concretamente se va a hacer uso de las librerías que Python ofrece para trabajar con redes neuronales: Tensorflow y Keras.

Tensorflow [24] es una biblioteca de código abierto de Python desarrollada por Google, con el objetivo de suministrar a los usuarios las herramientas con las que crear y entrenar redes neuronales. Keras [25] también es una biblioteca de código abierto que en este caso es capaz de ejecutarse sobre Tensorflow, su función es trabajar con redes de Deep Learning de una forma sencilla e intuitiva para el usuario.

Pese a tratarse de bibliotecas de código abierto, establecen una serie de requerimientos de hardware para un correcto procesamiento del sistema, dado que se requiere una capacidad computacional alta y por ello un ordenador capaz de soportarlo. El primer requerimiento es acerca de la CPU, debiendo ser un procesador Intel Core i5 de sexta generación o modelos superiores (siendo también válidos los equivalentes de la compañía AMD), por su parte se requiere un mínimo de 8 GB de memoria RAM, siendo altamente aconsejable disponer de una cantidad mayor. En cuanto a la GPU, se necesita del uso de una tarjeta de NVIDIA, partiendo de la GeForce GTX 960 o superiores. En caso de no disponer de ella no se podrá hacer uso de la GPU, lo que ralentizará enormemente el proceso de entrenamiento de la red. Finalmente, se recomienda trabajar en un sistema operativo Microsoft Windows (a partir de su versión 10) o Linux, siendo Ubuntu la distribución idónea para ello.

Capítulo 5. SISTEMA DESARROLLADO

El modelo desarrollado se trata de una categorización y análisis cuantitativo de los hiperparámetros de una red neuronal, a través de un sistema de valoración de los argumentos de salida de la red que, en este caso, al estar aplicada a la generación de texto y en esta ocasión en particular a la poesía, serán versos.

Como punto de partida, se procederá a explicar el sistema creado para analizar de forma cuantitativa la influencia de cada uno de los hiperparámetros en el funcionamiento de la red neuronal. Esto se va a realizar a través de un sistema de valoración de versos que, pese a tener una base subjetiva, va a pretender categorizar los valores de la forma más objetiva posible, distinguiendo a su vez entre los resultados gramáticos y artísticos, dado que se trata de un estilo literario en el que transmitir sentimientos y emociones juega un papel esencial.

Para una comprensión completa de lo realizado en el estudio, será fundamental comprender en profundidad el funcionamiento de los hiperparámetros. Por ello, se va a definir qué es un parámetro de una red neuronal, y qué es lo que lo diferencia de un hiperparámetro. Esta definición ayudará a comprender la importancia que tienen en el modelo, así como la dependencia que tiene la red de los valores establecidos para ellos, siendo los encargados de determinar su comportamiento. También será de vital importancia diferenciar en qué punto del proceso de la creación de la red neuronal aparece cada uno de los hiperparámetros estudiados.

De esta forma, a lo largo del análisis se desarrollará en detalle cada uno de los tres hiperparámetros que se han elegido para definir el comportamiento de la red neuronal: la temperatura de excitación del modelo, el tamaño del embedding y el número de neuronas de la red. Para ello, se concretará su funcionamiento, delimitando sus características y origen, así como se profundizará en el rango de los valores de cada hiperparámetro entre los que suelen encontrarse los límites de funcionamiento.

5.1 VALORACIÓN CUANTITATIVA DE LOS VERSOS

Para poder establecer de forma correcta la influencia que tienen sobre el comportamiento de la red neuronal los cambios en los valores de los hiperparámetros, es necesario analizar los elementos de salida de la red: los versos.

La principal dificultad radica en que, a diferencia de otro tipo de aplicaciones donde la categorización de los elementos de salida de la red se establece de forma objetiva, en este caso la clasificación de versos resulta ser un proceso puramente subjetivo. En el caso de un algoritmo clasificador de los correos electrónicos como spam, la calidad de la red clasificadora puede determinarse comprobando las salidas y determinando si existe o no error. Pero en este proyecto se necesita dotar a los argumentos de salida de la red de una valoración para poder determinar en qué medida realmente podrían llegar a ser versos, o simplemente secuencias de palabras sin sentido gramatical o artístico alguno.

Para valorar los versos, se ha establecido un sistema de evaluación de 0 a 10 puntos a través del cual se han juzgado todos y cada uno de ellos, para cada combinación de valores de cada uno de los hiperparámetros analizados.

Como se trata de un proceso realmente subjetivo, se han establecido una serie de criterios de evaluación para realizar la valoración, buscando hacer algo más objetiva la evaluación, de forma que se sigan siempre unos parámetros de calidad. Esta evaluación califica los versos atendiendo a dos características: su valor gramático y su valor artístico.

Para evitar resultados extremos, para cada valor de los hiperparámetros se han tomado 10 versos de salida del sistema, y ha sido la media aritmética de esos 10 valores precisamente el número que establece la calidad (ya sea gramatical o artística) del modelo para unos determinados valores de los hiperparámetros. Se ha elegido evaluar 10 versos de forma que sea un valor lo suficientemente grande como para reducir la influencia de casos extremos que puedan producirse, pero a su vez sea un valor lo suficientemente comedido como para evitar una cantidad de datos inabarcable.

5.1.1 VALORACIÓN GRAMATICAL

Con esta clasificación, se pretende establecer el grado de corrección gramatical que tienen los versos del modelo. Para clasificarlos, se han calificado de 0 a 10 atendiendo a una serie de patrones, los cuales están expuestos en la Tabla 1.

Nota	Tipo de verso
10	Gramaticalmente correcto
9	Verso con un solo error gramático
8	Entre dos y tres errores, pero mantiene la coherencia
7	Verso coherente, con 4 o 5 errores
6	Verso con varios errores, pero capaz de mantener un sentido claro
5	Se puede intuir el sentido del verso, pese a los errores de gravedad
4	Verso sin coherencia, pero con varias combinaciones correctas de palabras
3	Verso incoherente con varias palabras bien ubicadas
2	Sin sentido con alguna combinación de palabras correcta
1	Carente de sentido y palabras aleatorias
0	Verso vacío

Tabla 1: Patrones de valoración gramatical

De esta forma, a cada verso se le va a asignar el valor de la primera condición de la tabla que cumpla. Esta clasificación pretende establecer una graduación gramatical a los versos, se evalúa que las palabras guarden una correcta de tipo de palabra (sustantivo, adjetivo, verbo...), género y número. Por lo tanto, esta clasificación también podría ser utilizada de forma análoga en la evaluación de otro género literario, ya sea prosa o teatro.

A modo de ejemplo, se muestran algunos ejemplos de versos de salida del modelo, con la calificación gramatical obtenida: “*Caracol radiante que destellas besando*” es un verso gramaticalmente correcto, todas las relaciones de tipo de palabras, género y número son correctas, por lo que tiene un 10 de nota gramatical. Sin embargo, “*callados pluma tranquilos quemados que cada para olvido*” se trata de un verso sin sentido gramatical, con alguna relación de número correcta entre las palabras (callados, tranquilos, quemados) por lo que se le asignará una calificación de 4.

5.1.2 VALORACIÓN ARTÍSTICA

Dado que se está analizando una red neuronal destinada a la generación de poesía, también es necesario establecer una forma de clasificar las salidas en su capacidad de ser versos de un poema. Atendiendo a la definición de la Real Academia Española, la poesía se trata de “manifestación de la belleza o del sentimiento estético por medio de la palabra, en verso o en prosa” [26]. Por tanto, se va a realizar una clasificación que establezca un número entre 0 y 10 que refleje la capacidad de cada verso de cumplir dicha definición. A esta calificación se le ha denominado valoración artística.

Como ocurría con la valoración gramatical, se han establecido una serie de patrones de comportamiento de los versos a través de los cuales los vamos a clasificar, los cuales se muestran en la Tabla 2.

Nota	Tipo de verso
10	Transmite un sentimiento y emplea algún recurso literario
9	Transmite un sentimiento de forma clara
8	Algún recurso literario bien ejecutado
7	Verso con sentido, pero sin capacidad de transmitir un sentimiento
6	Verso con cierta emoción, aunque pueda resultar incoherente
5	Se puede intuir una emoción tras el verso
4	Se puede intuir alguna figura literaria
3	Alguna palabra que transmite una emoción
2	Alguna combinación de palabras correcta
1	Sin ningún tipo de coherencia
0	Verso vacío

Tabla 2: Patrones de valoración artística

En esta ocasión, a diferencia de la evaluación gramatical, esta clasificación no sería aplicable a otro género literario, dado que es una característica especial de la poesía.

En esta clasificación un verso que recibiría una calificación 10 es por ejemplo “*cargados de infinita pesadumbre brillar*” puesto que expresa emoción y emplea un recurso literario como es la hipérbole, pero que sin embargo no recibiría un 10 en nota gramatical, ya que la

conjugación verbal es errónea. El ejemplo opuesto sucede en el siguiente verso: “*nosotros con los codos*” puesto que es un verso gramaticalmente correcto que tendrá una calificación de 10, mientras que no es capaz de transmitir sentimiento alguno, por lo que se calificará con un 7 de nota artística.

De esta forma, cada modelo generado para una determinada combinación de hiperparámetros será evaluado a través de la media aritmética de las notas gramaticales y artísticas de 10 versos de salida. Y finalmente se realizará la media aritmética de ambas calificaciones para obtener una nota de calidad del modelo, entre 0 y 10, que servirá para determinar para qué valores de los hiperparámetros se obtienen resultados óptimos.

5.2 NÚMERO DE CAPAS DE LA RED

De forma previa a realizar el análisis de los hiperparámetros, se ha estudiado cuál es el número de capas que optimiza los resultados de la red.

Una red neuronal puede estar formada por un número variable de capas, siendo cada capa un nivel de agrupación de neuronas. Es decir, el elemento básico o esencial es la neurona, a su vez un número determinado de neuronas se agrupan en capas y finalmente la unión de un número determinado de capas da lugar a la red neuronal. Esta jerarquía se muestra en la Figura 8.

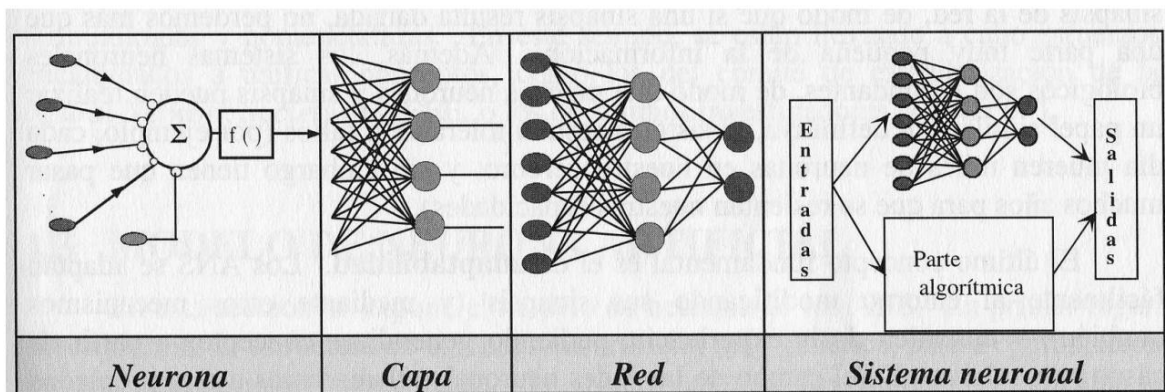


Figura 8: Jerarquía de una red neuronal [27]

Existen tres tipos de capas de una red neuronal: capas de entrada, capas de salida y capas ocultas. La capa de entrada está formada por las neuronas que reciben los datos de entrada de la red, la capa de salida por su parte es la encargada de elaborar el elemento de salida de la red neuronal (en el caso de este proyecto, un verso) y finalmente las capas ocultas son aquellas que no tienen relación directa con los datos de entrada o con el elemento de salida, sino que están a su vez conectadas entre capas. Este último tipo de capa tiene relación directa con el aumento del número de grados de libertad del sistema, lo que puede suponer que la red neuronal cree conexiones añadidas a los datos de entrada, que aumentan la complejidad de la red y permiten establecer relaciones que no aparecen en caso de incluir exclusivamente redes sin capas ocultas. La Figura 9 refleja los tipos de capas de una red neuronal.

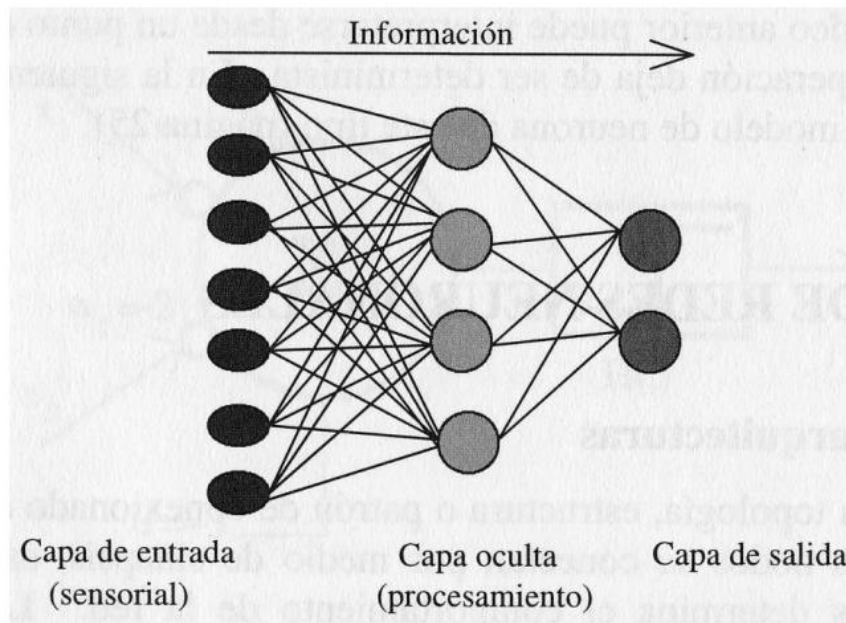


Figura 9: Tipos de capas en una red neuronal [27]

Para establecer el número óptimo de capas que se va a utilizar en el generador de poesía, se realiza un análisis de los versos de salida obtenidos por un mismo modelo, únicamente variando el número de capas del modelo, manteniendo los hiperparámetros constantes. Como se ha indicado en el 5.1, se analizarán 10 versos de cada modelo, variando el número de capas desde 1 hasta 4. Los resultados obtenidos se muestran en la Tabla 3.

<i>Nº Capas</i>	<i>Nota Gramatical</i>	<i>Nota Artística</i>	<i>Nota Media</i>
1	6	6	6
2	8,1	8,3	8,2
3	8,2	7,9	8,05
4	7,5	7,4	7,45

Tabla 3: Resultados en función del número de capas

Como se puede comprobar en los resultados anteriores, cuando tan solo se utiliza una capa los resultados son considerablemente peores que el resto, debido que esa capa de neuronas tendrá que hacer la función tanto de capa de entrada como capa de salida, y por tanto estará directamente relacionada tanto con los datos de entrada como con los de salida. Se trata de un caso evidente en que la red neuronal no tiene profundidad suficiente.

Al aumentar el número de capas de la red los resultados mejoran, pero como se puede apreciar perfectamente en la Tabla 3 los mejores resultados en nota media son para dos capas, aunque el resultado es muy similar al obtenido para tres capas.

Pero resulta interesante observar cómo los resultados empeoran cuando se aumenta el número de capas hasta 4, es decir, con un total de dos capas ocultas. Esto supone que la red neuronal dispone de dos grados de libertad y por tanto las neuronas pueden establecer conexiones entre palabras que no aparecen para un número de capas inferior. Sin embargo, los resultados obtenidos son peores. Es un ejemplo evidente de un problema de sobreentrenamiento de la red: la profundidad de la red es excesiva. Por ello, no será bueno para el funcionamiento de la red establecer el número de capas por encima de 4.

Por tanto, se tendrá que determinar el número óptimo de capas de la red eligiendo entre dos o tres capas. La calidad de los resultados es muy similar en ambos casos, y dado que se trata de una forma de evaluación subjetiva, el resultado de la calificación no será un factor diferencial entre las dos opciones. Sin embargo, hay que tener en cuenta la optimización de los resultados, una de las bases del funcionamiento del mundo industrial: si dos procesos tienen resultados similares, se elegirá siempre el más sencillo. En este contexto, el modelo creado con dos capas es considerablemente más simple que el de tres, puesto que existe un menor número de conexiones entre las neuronas.

Teniendo todo esto en cuenta, se ha decidido optar por una red neuronal recurrente formada por dos capas: una capa de entrada y otra de salida. De ahora en adelante, todos los modelos creados para el análisis de los hiperparámetros serán redes neuronales recurrentes formadas por dos capas, estando ambas capas formadas por el mismo número de neuronas. Este valor del número de neuronas se analizará en el apartado 5.6.

5.3 PARÁMETROS E HIPERPARÁMETROS

En una red neuronal, los parámetros e hiperparámetros son los valores encargados de regir el funcionamiento de la red. Estos valores serán ajustados a lo largo del desarrollo de una red neuronal recurrente, y determinarán los comportamientos de la red en todos sus niveles.

Es importante recalcar la diferencia existente entre los parámetros e hiperparámetros de una red neuronal, para evitar confundirlos. Por un lado, los parámetros son variables que se deducen del entrenamiento de la red a través de los datos de entrada, los valores que toman los parámetros de la red no son asignados por una persona, sino que son adoptados por el modelo: dependen del conjunto de datos de entrada con el que se va a entrenar la red. La estimación de los parámetros es la parte fundamental del funcionamiento de una red neuronal recurrente: si la estimación es buena, la red funcionará correctamente.

Por su parte, los hiperparámetros son valores que no se obtienen de los datos de entrada, sino que son introducidos por un programador de forma previa al entrenamiento de la red, es decir: son independientes de los valores de entrada. Sus valores determinarán las distintas configuraciones con las que la red hará uso de los datos de entrenamiento para estimar los parámetros. Mientras que los valores óptimos de los parámetros (para cada configuración de hiperparámetros existirán unos valores óptimos de los parámetros) son deducidos por la propia red a través del entrenamiento, no se pueden conocer a priori los mejores valores de los hiperparámetros, y ese es precisamente el objetivo de este proyecto: encontrar a través de los datos y la estadística los valores óptimos de los hiperparámetros de la red neuronal.

Además, dado que en Deep Learning el entrenamiento de la red se realiza de manera no supervisada, es decir, la red no necesita de una persona que controle el aprendizaje, sino que es ella misma la que aprende a través de la experiencia, la única forma que tiene una persona de influir en el comportamiento de una red neuronal recurrente es a través del control de los hiperparámetros, de ahí su vital importancia. En la Tabla 4 se resumen las principales diferencias entre parámetros e hiperparámetros.

<i>PARÁMETROS</i>	<i>HIPERPARÁMETROS</i>
Dependientes de los datos de entrada	Independientes de los datos de entrada
Fijados por la propia red	Establecidos por un programador
Los valores óptimos se obtienen en el proceso de entrenamiento	Los valores óptimos deben ser buscados por el programador
Determinan las predicciones del modelo	Determinan la forma de tratar los datos de entrada

Tabla 4: Diferencias entre parámetros e hiperparámetros

Existe una gran variedad de hiperparámetros, cuyo número puede aumentar a medida que aumenta la complejidad de la red, sin ser un valor fijo. Cada uno de ellos puede influir en la red neuronal de una forma muy distinta, y también dependiendo de la finalidad y futuro funcionamiento de la propia red, estos tendrán un mayor o menor peso.

Para poder abordar el análisis se ha realizado una selección de tres hiperparámetros sobre los que se va a profundizar, puesto que son tres de los más influyentes en el funcionamiento de la red, así como por ser tres de los más utilizados a la hora de emplear redes neuronales recurrentes: la temperatura de excitación del modelo, el tamaño del embedding y el número de neuronas de la red.

Pese a ser todos hiperparámetros de una red neuronal recurrente, entre los tres existen varias diferencias, tanto en sus características como en sus valores. Una de las diferencias fundamentales es sin duda el momento en el que aparecen en el desarrollo de la red neuronal. Mientras que el tamaño del Embedding y el número de neuronas son valores que se establecen para el entrenamiento de los modelos, la temperatura es un hiperparámetro que aparece a la hora de excitar el modelo. La Figura 10 refleja esta diferenciación de los hiperparámetros.

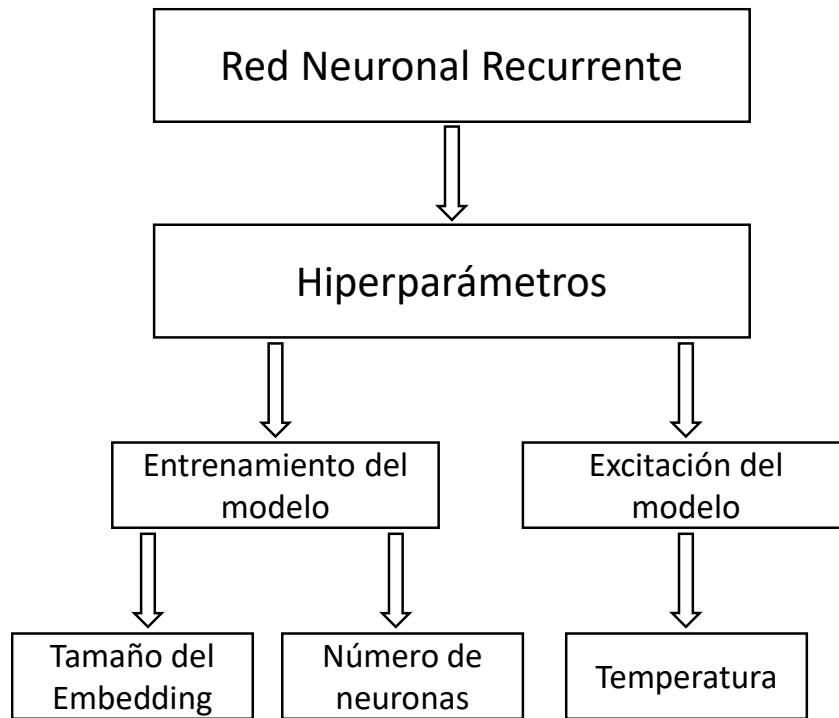


Figura 10: Ubicación de los hiperparámetros en el funcionamiento de la red

Esta diferenciación será clave a la hora de evaluar la influencia que tienen los hiperparámetros en la red neuronal, ya que para analizar el efecto que suponen las diversas modificaciones en los valores de los hiperparámetros se tendrá que hacer una variación de cada uno de ellos manteniendo el resto constantes.

Como el momento de aparición de la temperatura es diferente respecto al de los otros dos, en un primer acercamiento se va a analizar la influencia de la temperatura de forma independiente al número de neuronas y tamaño del embedding.

5.4 TEMPERATURA

La temperatura de excitación del modelo es un hiperparámetro que se utiliza para controlar el grado de aleatoriedad que la red neuronal utiliza para llevar a cabo una predicción. Este grado de aleatoriedad es aplicado al ejecutar la función softmax que arrojará como resultado la salida de la red.

La función softmax[28], también conocida como función exponencial normalizada, es una función que se emplea para transformar un vector de n dimensiones con valores variables, en un vector de dimensión idéntica, pero cuyos valores están todos situados entre 0 y 1, como se muestra en la ecuación (1). Estos resultados, para cada uno de los valores de salida, resultan del cociente de la función exponencial de probabilidad para esa salida en concreto, entre el sumatorio de las exponenciales de todas las funciones de salida para el resto de posibles valores de salida, como se ve en la ecuación (2), de esta forma se asegura que todos los resultados estén dentro del intervalo $[0, 1]$.

$$(1) \quad \sigma: \mathbb{R}^n \rightarrow [0, 1]^n$$

$$(2) \quad \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^n e^{z_k}}$$

El estudio realizado por Christopher M. Bishop [29] sobre las aplicaciones de la función softmax en computación determina que es una de las funciones óptimas para el establecimiento de enormes cantidades de probabilidades, precisamente el caso de este proyecto.

En redes neuronales, la salida de la función es un vector de probabilidades de los n posibles valores de la salida. Esas posibilidades están condicionadas al tipo de entrenamiento que se ha empleado en la red, en el caso de este proyecto, al ser una red natural recurrente, las salidas anteriores influyen directamente en la probabilidad de salida de las siguientes, por lo que entrarán en la función softmax de las salidas posteriores.

No obstante, es posible alterar en cierto modo ese vector de probabilidades, y se va a realizar precisamente dividiendo los exponentes de la función por un valor numérico, de forma que se altere el resultado de la función exponencial tanto en el numerador como en el sumatorio

del denominador, como se puede apreciar en la ecuación (3). Ese valor numérico es precisamente la temperatura.

$$(3) \quad \sigma(z)_j = \frac{e^{\frac{z_j}{T}}}{\sum_{k=1}^n e^{\frac{z_k}{T}}}$$

Cuando el valor de la temperatura es igual a 1, la función softmax tenemos de nuevo el caso anterior: la función softmax se ejecuta sin ningún tipo de modificación. Sin embargo, si este valor se modifica existen dos escenarios diferentes.

En caso de que la temperatura sea menor que 1, las probabilidades más altas de la función softmax original ganan aún más importancia, por lo que la red neuronal de generación de texto tiende a volverse más segura y conservadora, de forma que los versos generados ganan en coherencia, pero pierden en variedad y riqueza del lenguaje.

Por otro lado, si la temperatura es mayor que 1, el vector de probabilidades se hace más relativo: tienen menor importancia las salidas originales de la función softmax puesto que el cociente dentro de la exponencial las equipara. De este modo, la red neuronal generará versos con un aumento de la diversidad y variación existente entre los versos, pero lo que también trae consigo una disminución de la seguridad a la hora de la creación de los versos, lo que puede significar un aumento de la inconsistencia e incoherencia en las salidas del sistema.

5.4.1 TEMPERATURA CONSTANTE

En esta parte del estudio de la temperatura, se va a analizar cómo afecta este hiperparámetro si se mantiene constante durante toda la creación de cada secuencia de salida, es decir, de cada verso. Esto supone que en todas y cada una de las palabras que componen un verso, la función softmax será ejecutada con un mismo valor de temperatura.

Para realizar el análisis, se va a realizar un recorrido por valores de temperatura menores, mayores e iguales a 1, para analizar si en una red neuronal recurrente se alcanzan resultados óptimos aumentando, disminuyendo o conservando la aleatoriedad del comportamiento del sistema.

Por tanto, se han analizado 10 versos para cada uno de los valores de temperatura del intervalo contenido entre 0.7 y 1.7, con variaciones de 0.1 entre los versos. Con todo ello, se obtienen los resultados que se muestran en la Tabla 5.

<i>Temperatura</i>	<i>Nota Gramatical</i>	<i>Nota Artística</i>	<i>Nota Media</i>
0,7	5,00	5,00	5,00
0,8	6,05	5,80	5,93
0,9	5,95	6,11	6,03
1	6,00	5,89	5,95
1,1	6,40	6,21	6,31
1,2	6,00	6,32	6,16
1,3	7,26	7,16	7,21
1,4	6,37	6,80	6,59
1,5	6,85	6,20	6,53
1,6	5,80	6,00	5,90
1,7	4,25	5,00	4,63

Tabla 5: Resultados temperatura constante

Para un mejor análisis de los resultados, también se representan las calificaciones de forma gráfica en la Figura 11.

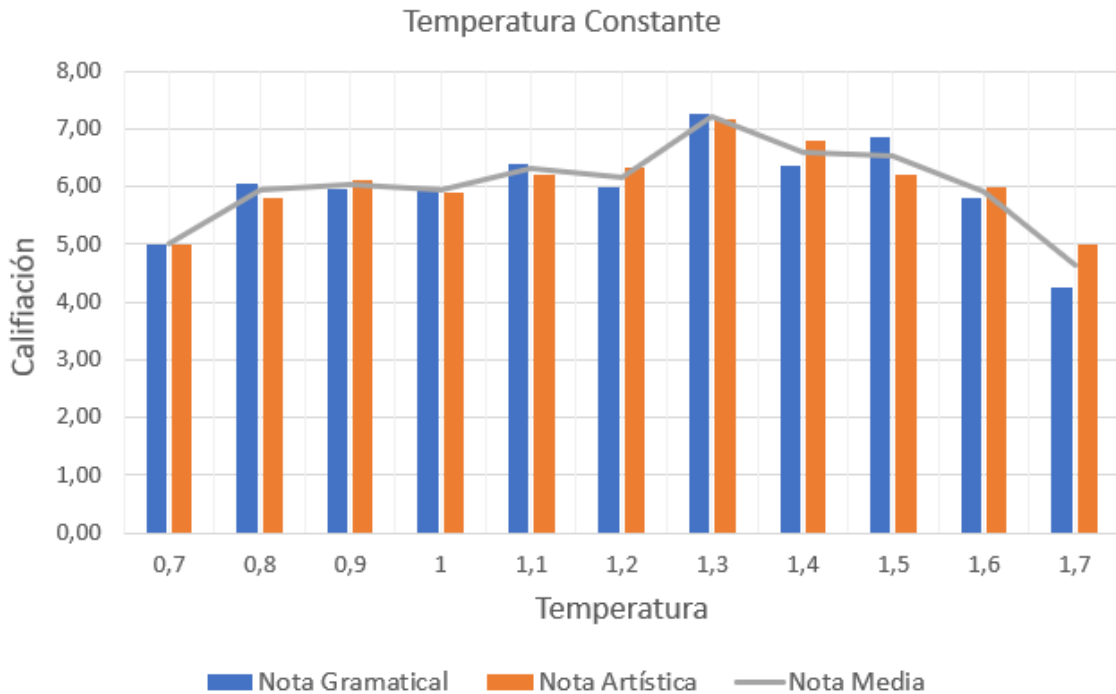


Figura 11: Representación gráfica del análisis de temperatura constante

Como primera conclusión se puede establecer que la calidad de los versos, tanto a nivel artístico como a nivel gramatical, aumenta con la temperatura hasta que esta alcanza un valor de 1.3, momento a partir del cual la calidad se va reduciendo al aumentar el valor del hiperparámetro. Por tanto, será precisamente 1.3 el valor de la temperatura con el que se alcanzan unos versos óptimos.

También se debe analizar qué sucede en ambos extremos del análisis. Por un lado, cuando el valor de la temperatura toma valores excesivamente bajos, la función softmax se ejecuta de una forma más conservadora, lo que supone que se elegirán las palabras con una probabilidad condicionada mayor. Cuando esto se lleva al extremo, el resultado se concreta en que la red prácticamente siempre el mismo verso, o en su defecto versos realmente similares entre sí. En este caso, el verso generado por la red es: “*Tus toda mismo río, es que no existe*” que es la secuencia de palabras que tiene una mayor probabilidad atendiendo a la obra que se ha utilizado como base de datos de entrada. Si la red neuronal recurrente se ciñese simplemente a analizar la obra y crear el verso más probable, sería precisamente ese verso.

Por otro lado, a medida que crece el valor de la temperatura, la variedad de palabras presentes en los versos aumenta. Esto es debido a que la función softmax no se limita únicamente a elegir la palabra con una mayor probabilidad para cada palabra de un verso, sino que, siguiendo un comportamiento dependiente de todas esas probabilidades condicionadas, establecerá diversas palabras en cada ocasión. De nuevo, si se lleva este comportamiento al extremo, la variedad en los versos deriva en una función softmax excesivamente relajada, en la que las probabilidades de los versos pierden importancia pudiendo llegar a perder toda la información extraída del entrenamiento de la red. Además, otro problema derivado de esta situación es que la red generará versos hasta encontrar el vector de fin de verso: *eof*. La probabilidad en la función softmax de encontrar el *eof* aumenta al avanzar en verso. Sin embargo, si la temperatura es excesiva, este aumento de la probabilidad no será lo suficientemente grande, lo que da lugar a versos excesivamente largos que pueden colapsar al sistema.

Finalmente, del análisis de la temperatura constante se extrae una conclusión clara. Un valor reducido de temperatura da como resultado versos coherentes, de longitud comedida y con un buen nivel de relación entre las palabras, pero que sin embargo la variedad y riqueza de palabras es insuficiente. En el lado opuesto un valor elevado de la temperatura da como resultado versos variados y con una mayor amplitud en el abanico de palabras empleadas, pero que pueden llegar a ser incoherentes y excesivamente largos. Por tanto, el valor óptimo de la temperatura constante será aquel que permita alcanzar un compromiso entre coherencia y variedad, de forma que se encuentre un equilibrio en los versos analizados. En este análisis se ha alcanzado un mejor compromiso para un valor de la temperatura de 1,3.

5.4.2 TEMPERATURA DECRECIENTE

Del análisis del valor constante de temperatura se extrae en claro que, al aumentar dicho valor, los versos de salida ganan en variedad y riqueza de vocabulario, pero sin embargo se encuentra con un inconveniente claro: la pérdida de consistencia a lo largo de los versos, que puede derivar incluso en versos extremadamente largos e inconexos.

El principal problema para temperaturas elevadas se encuentra en la parte final de los versos, ya que para finalizar la creación de uno debe encontrar la palabra *eof* (*end of file*) y no será hasta ese momento donde se finalice el verso y comience a crear el siguiente. La situación ideal estaría en encontrar un sistema que permita utilizar temperaturas elevadas al comienzo de los versos, para tener una variabilidad considerable de palabras; pero que la temperatura fuera decreciendo a medida que se va construyendo cada verso, de forma que se asegure un cierto grado de coherencia y conexión entre las palabras.

Por ello, se ha tomado como referencia el algoritmo de recocido simulado, utilizado con el cocido y recocido de acero que consiste en inicialmente aumentar enormemente la temperatura y posteriormente reducirla de forma gradual, ya empleado en procesos de gestión de información como por ejemplo con datos genéticos [30]. De esta forma, se ha implementado la aplicación de una temperatura variable para cada palabra a lo largo de un verso, que dependa de la posición que ocupe cada palabra en el verso: a medida que se avanza a lo largo del verso, la temperatura va decreciendo de forma que la probabilidad de encontrar el final de frase *eof* también aumenta.

De esta forma, la temperatura utilizada toma un valor asignado por la ecuación (4), dependiente de la posición de la palabra en el verso: n .

$$(4) \quad Temp_n = T \cdot dec^n$$

Donde además T es el valor inicial de la temperatura, el que se utiliza para obtener la primera palabra del verso, y dec es un valor por el que se va a multiplicar la temperatura inicial de forma recursiva, al que de ahora en adelante se va a denominar decay. Si el valor del decay fuera mayor que 1, el resultado de esa multiplicación recursiva sería un valor cada vez más

grande de temperatura. Por el contrario, si ese valor fuera menor que 1, el decay traería consigo un decremento gradual del valor de la temperatura: En caso de que fuera exactamente 1, la temperatura se mantendría constante. Estas tres situaciones, se muestran en la Tabla 6.

Decay	Temp	Valor de tendencia
> 1	Creciente	∞
= 1	Constante	$Temp_0 = T$
< 1	Decreciente	0

Tabla 6: Valor de la temperatura en función del Decay

Como se busca una temperatura decreciente al avanzar a lo largo del verso, resulta evidente que los valores de decay elegidos pertenecerán a un rango de números que se extiende entre el 0 y el 1. Dentro de ese rango, a medida que ese valor se acerca al 0 el decrecimiento de la temperatura a lo largo del verso se da de una forma más pronunciada, mientras que en los valores más cercanos al 1 este decrecimiento se realizará de una forma más suave. Hay que tener en cuenta el comportamiento exponencial de la expresión, por lo que cambios muy pequeños en el valor asignado al decay pueden conllevar cambios considerablemente grandes en la temperatura.

Para analizar la influencia de la temperatura en esta nueva configuración, se ha procedido a realizar una clasificación en tres fases de los versos, ya que en este caso el hiperparámetro depende de dos valores que serán analizados por separado, y posteriormente se analizará la relación existente entre ambos, de forma que los versos obtenidos en cada modelo sean los mejores posibles.

Para comenzar, se realizará un análisis del valor del decay, que como se ha razonado anteriormente tomará valores entre 0 y 1. Para este análisis se ha tomado un valor de temperatura inicial constante, que será el mismo en todos los versos donde tan solo se modificará el valor del decay. De este análisis se obtendrá el valor óptimo del decay para dicho valor de temperatura inicial.

La segunda fase será analizar la influencia de la temperatura inicial $Temp_n$, manteniendo en todos los versos el mejor valor del decay obtenido en la fase 1. De este modo se obtendrá el valor óptimo de la temperatura inicial para un valor fijo del decay.

Finalmente, se va a analizar la relación entre ambos parámetros, ya que es de su combinación de donde sale finalmente el valor de la temperatura para que el modelo elija cada palabra a lo largo de un verso, utilizando como referencia los valores obtenidos en las fases anteriores. El orden en el análisis se muestra en la Figura 12.

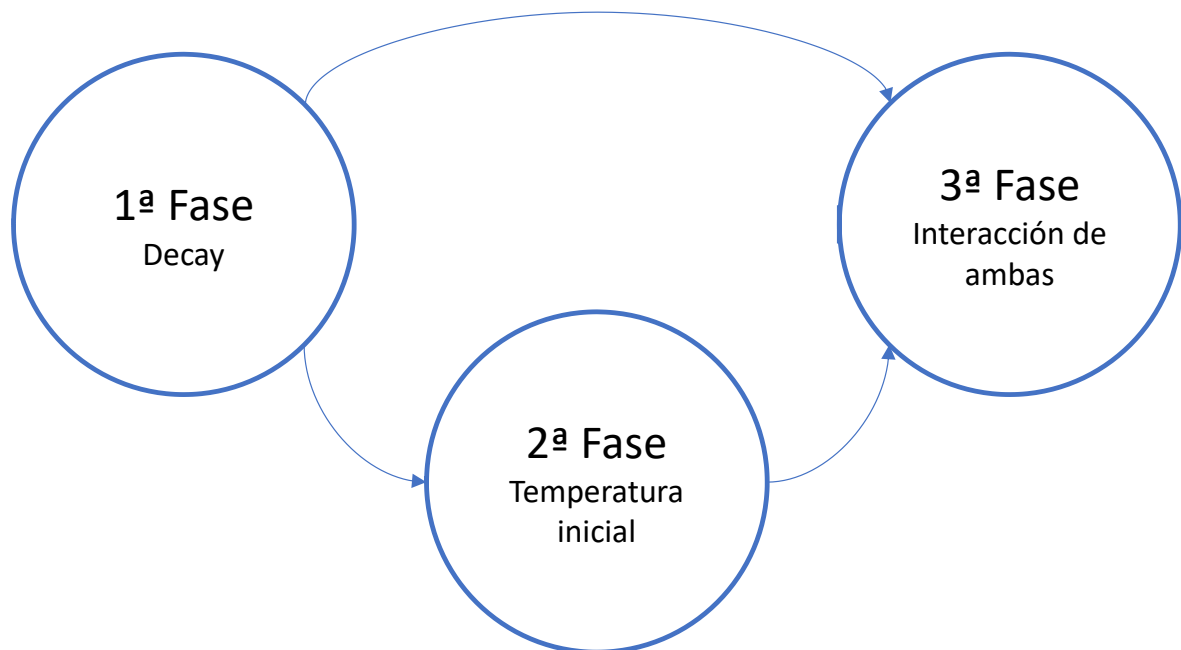


Figura 12: Fases del análisis de la temperatura variable

5.4.2.1 Análisis del valor del decay

Se trata de la primera fase del análisis de la temperatura variable en una red neuronal recurrente. Precisamente el hecho de ser el comienzo del análisis supone que no existe ningún argumento de entrada, por lo que se elegirá un determinado valor de la temperatura inicial sin información sobre si se trata o no de un valor óptimo, y que se mantendrá constante durante todo el estudio. El valor elegido es el mejor valor del análisis de la temperatura constante: $T=1.3$.

Por tanto, se realizará una variación del valor del decay a lo largo de un rango establecido que permita un decrecimiento progresivo de la temperatura (entre 0 y 1). Previamente, se han establecido de forma cualitativa unos límites que permitan un análisis razonable de los versos. Estos límites se han situado en 0.85 y 0.95, dado que valores externos a ese rango dan versos cualitativamente peores que los obtenidos dentro del intervalo.

Dentro de ese rango, se desarrollará una valoración gramatical y artística de 10 versos de salida del modelo, para cada uno de los siguientes valores del decay. Una vez calificados, se realiza la media aritmética de los 10 valores asignados, obteniéndose los resultados que se muestran en la Tabla 7.

<i>Decay</i>	<i>Nota Gramatical</i>	<i>Nota Artística</i>	<i>Media</i>
0,85	6,75	6,50	6,63
0,86	6,75	7,00	6,88
0,87	7,50	8,00	7,75
0,88	6,75	7,50	7,13
0,89	6,25	6,00	6,13
0,90	7,25	7,25	7,25
0,91	5,50	6,00	5,75
0,92	4,00	4,25	4,13
0,93	4,75	5,25	5,00
0,94	4,75	5,00	4,88
0,95	4,00	5,00	4,50

Tabla 7: Influencia del decay en los versos

También se representa, para estudiar el caso de una forma más visual e intuitiva, una representación gráfica del análisis realizado. La Figura 13 refleja esta clasificación.

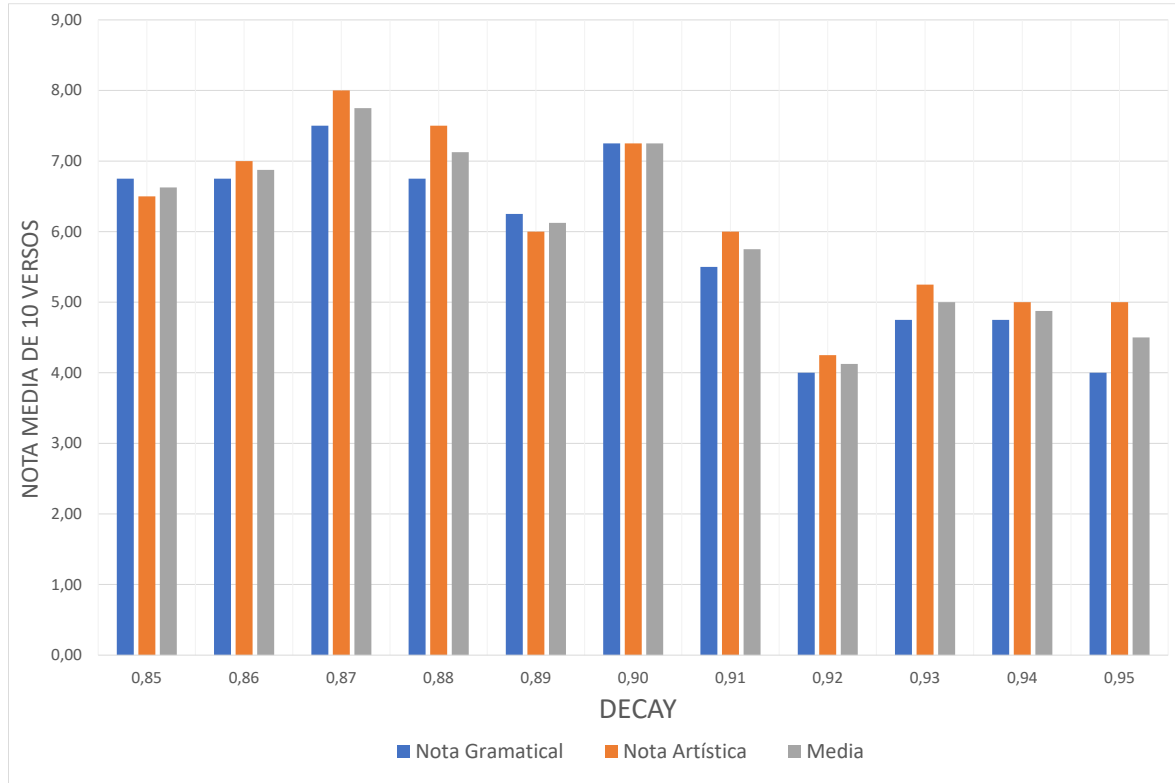


Figura 13: Representación gráfica de la influencia del decay

Como se puede comprobar en las representaciones anteriores, la calidad de los versos, tanto artística como gramatical, alcanza unos valores máximos para un valor del decay de 0.87. Será precisamente este valor el que se utilizará y se mantendrá constante en la fase 2 para establecer la temperatura inicial óptima. De la misma forma, la fase 3 tomará los valores alrededor de los cuales se ha obtenido un resultado mejor, que se corresponden con el intervalo [0.86 – 0.9], estos valores se agregarán al rango que se obtenga de la fase 2 para realizar un análisis de la relación entre ambos.

5.4.2.2 Análisis del valor de temperatura inicial

La temperatura inicial se ha analizado en un rango más amplio, dado que se ha comenzado por los valores entorno a los cuales se alcanzó un mejor resultado en el caso de temperatura constante. Los resultados obtenidos se muestran en la Tabla 8.

<i>Temperatura</i>	<i>Nota Gramatical</i>	<i>Nota Artística</i>	<i>Media</i>
1,00	4,80	4,70	4,75
1,10	5,80	5,80	5,80
1,20	5,90	6,00	5,95
1,30	6,00	5,90	5,95
1,40	6,70	6,80	6,75
1,50	6,60	7,10	6,85
1,60	6,00	6,10	6,05
1,70	6,00	5,50	5,75
1,80	6,30	6,30	6,30
1,90	6,90	6,60	6,75
2,00	6,80	6,80	6,80
2,10	5,75	5,75	5,75
2,20	6,75	7,00	6,88
2,30	7,50	7,25	7,38
2,40	7,00	7,25	7,13
2,50	6,25	6,75	6,50
2,60	7,25	7,25	7,25
2,70	6,25	6,50	6,38
2,80	6,25	6,00	6,13
2,90	5,75	6,25	6,00
3,00	5,50	6,00	5,75
3,10	5,75	5,75	5,75
3,20	5,25	4,75	5,00
3,30	6,00	4,75	5,38
3,40	4,50	4,75	4,63
3,50	5,00	5,25	5,13
3,60	4,00	4,75	4,38
3,70	4,50	4,25	4,38

Tabla 8: Influencia de la temperatura inicial en los versos

De la misma forma que se ha realizado en el análisis del decay, la Figura 14 representa gráficamente la calificación de los versos, en este caso para cada valor de la temperatura inicial.

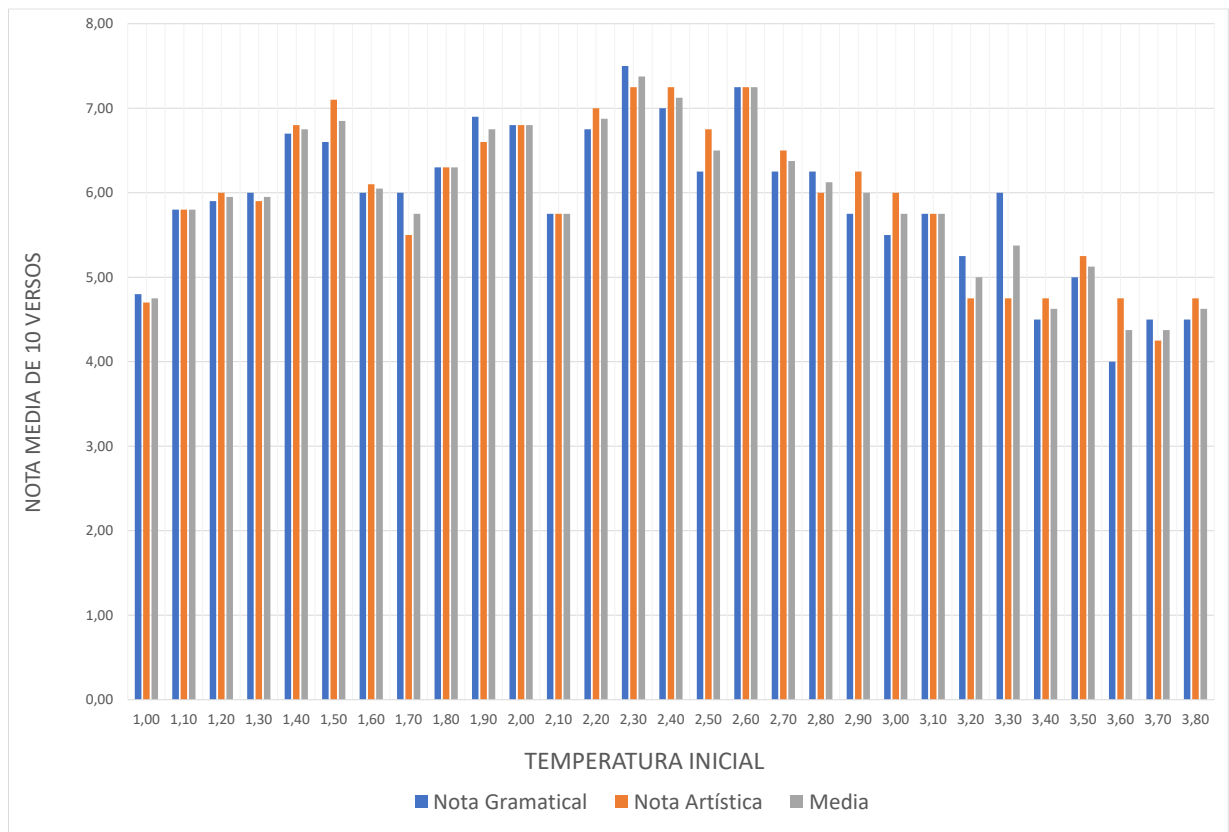


Figura 14: Representación gráfica de la influencia de la temperatura inicial

En este caso, y a diferencia del análisis de la fase 1, el parámetro del decay es el que se mantiene constante en todos los versos, tomándose el valor que dio un mejor resultado en el anterior análisis, es decir: $\text{decay} = 0.87$.

Como se puede comprobar de forma visual en la Figura 14, los valores de la temperatura inicial que dan lugar a unos versos de mayor calidad son los que se incluyen en el intervalo comprendido entre 2.2 y 2.7. Serán estos los valores de la temperatura inicial que se utilicen en la fase 3, junto con el rango de valores para el decay definido en la fase 1, para obtener la combinación óptima de ambos.

5.4.2.3 Análisis del decay junto con la temperatura inicial

En esta fase 3 del análisis de la temperatura variable, se va a estudiar la influencia de la variación simultánea de ambos valores en el modelo, por ello en esta ocasión no se va a mantener constante ninguno de los dos valores, sino que variarán ambos a lo largo del rango de valores extraídos de los análisis anteriores: fase 1 y fase 2.

De esta forma, se realiza una valoración de 10 versos para cada combinación de valores y se realiza la media aritmética. El resultado de la calidad gramatical de los versos se muestra en la Tabla 9, el de la calidad artística por su parte se refleja en la Tabla 10, mientras que la media de ambas calificaciones se muestra en la Tabla 11.

NOTA GRAMÁTICA		Decay				
		0,86	0,87	0,88	0,89	0,9
Temperatura Inicial	2,2	6,50	5,75	6,00	5,00	6,75
	2,3	7,25	7,50	6,25	5,00	4,75
	2,4	5,50	6,25	4,75	5,25	5,75
	2,5	7,00	4,75	6,25	4,50	5,75
	2,6	5,75	5,25	4,75	4,50	4,25
	2,7	6,00	6,00	5,00	5,75	5,50

Tabla 9: Calificación gramatical con decay y temperatura inicial variables

NOTA ARTÍSTICA		Decay				
		0,86	0,87	0,88	0,89	0,9
Temperatura Inicial	2,2	6,00	5,25	6,25	5,00	7,00
	2,3	7,50	8,00	7,00	5,25	5,50
	2,4	6,00	6,25	5,50	6,00	6,00
	2,5	6,75	5,25	6,00	5,00	4,75
	2,6	6,50	5,75	4,50	5,00	4,00
	2,7	6,25	5,25	4,50	5,75	5,00

Tabla 10: Calificación artística con decay y temperatura inicial variables

NOTA MEDIA		Decay				
		0,86	0,87	0,88	0,89	0,9
Temperatura Inicial	2,2	6,25	5,50	6,13	5,00	6,88
	2,3	7,38	7,75	6,63	5,13	5,13
	2,4	5,75	6,25	5,13	5,63	5,88
	2,5	6,88	5,00	6,13	4,75	5,25
	2,6	6,13	5,50	4,63	4,75	4,13
	2,7	6,13	5,63	4,75	5,75	5,25

Tabla 11: Nota media con decay y temperatura inicial variables

Como en el estudio de ambos valores por separado, resulta interesante representar gráficamente el contenido de las tablas anteriores. Para ello, se ha realizado una representación en tres dimensiones. Se muestra la Figura 15 para la calificación gramatical, la Figura 16 para la artística, y la Figura 17 para la media entre las dos anteriores. Donde los ejes horizontales se corresponden con los valores de temperatura inicial y decay, mientras que el vertical lo hace con la nota obtenida.

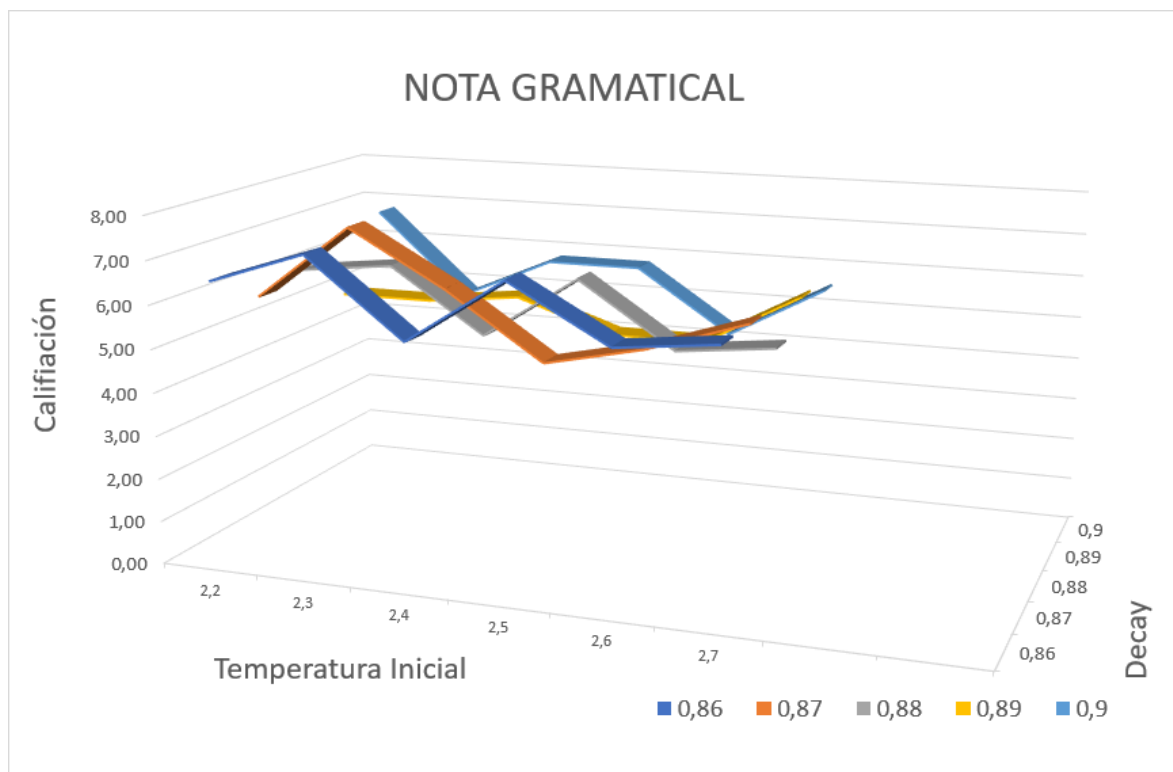


Figura 15: Influencia conjunta del decay y la temperatura inicial en la nota gramatical

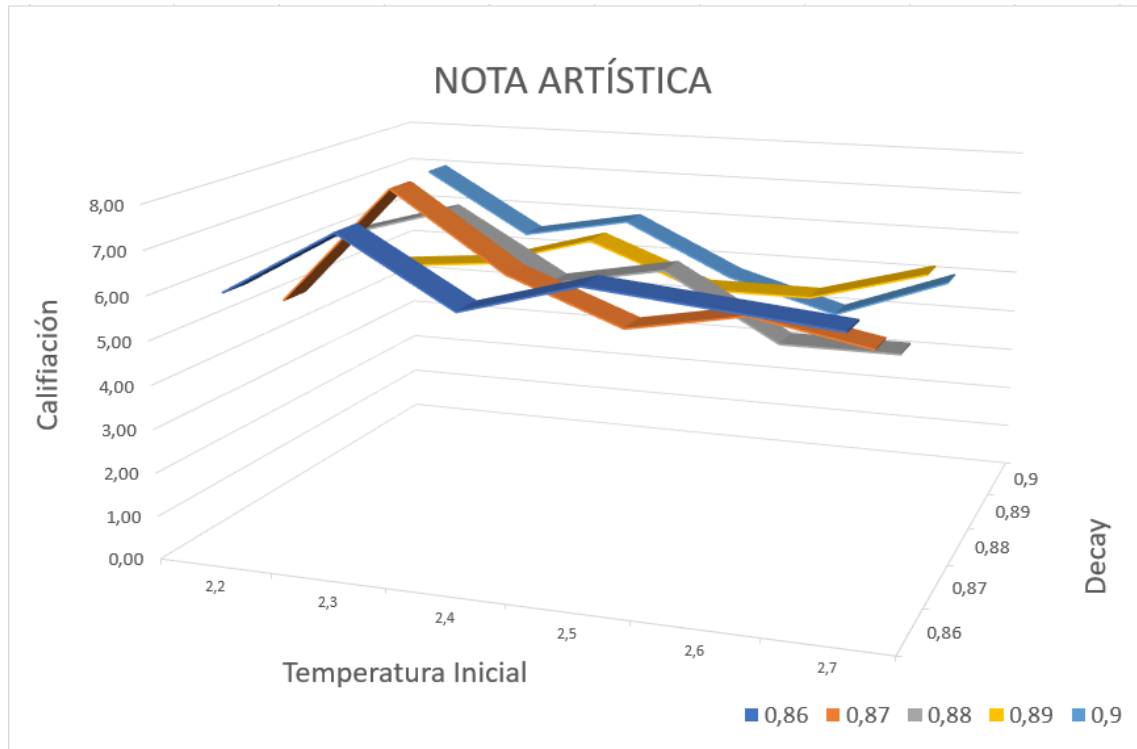


Figura 16: Influencia conjunta del decay y la temperatura inicial en la nota artística

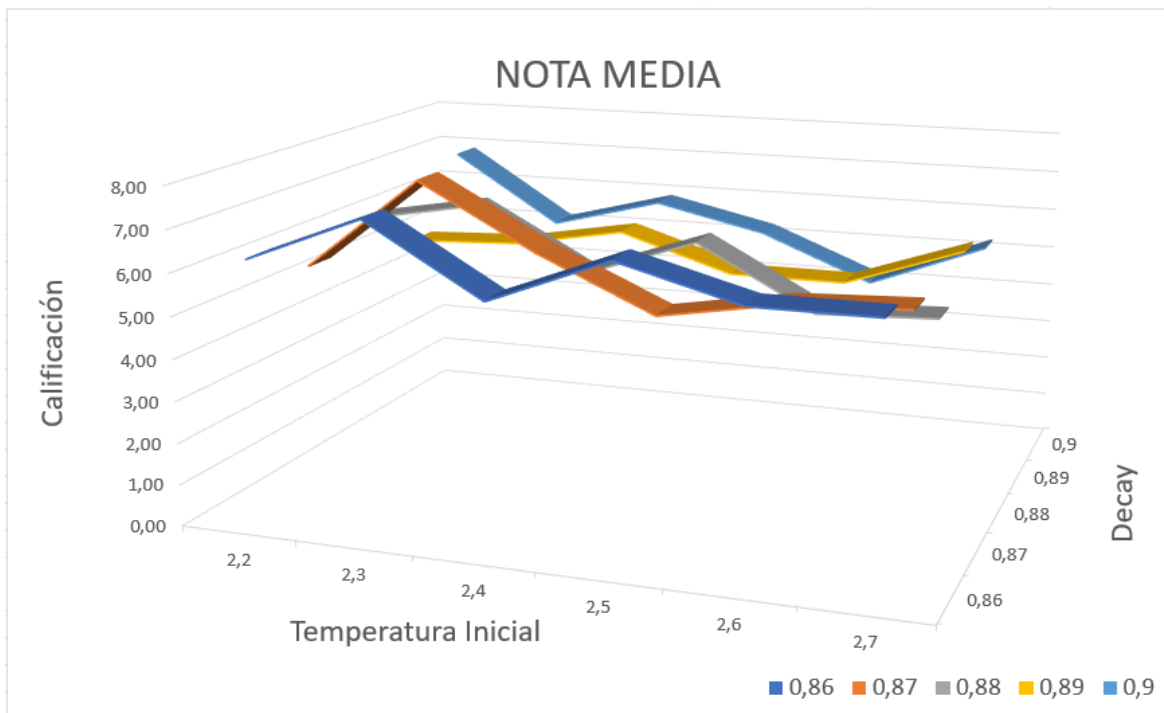


Figura 17: Influencia conjunta del decay y la temperatura inicial en la nota media

5.5 TAMAÑO DEL EMBEDDING

El word embedding es una vía de organización de las palabras que componen un lenguaje. Consiste en agrupar las palabras en un espacio vectorial de forma que la distancia entre ellas dependa de su grado de relación. Es importante recalcar que esta relación se refiere única y exclusivamente a su nivel semántico.

Esta herramienta es fundamental en redes neuronales de procesamiento de texto porque permite a la red neuronal establecer un tejido de conexiones y relaciones entre las palabras. Para ello, asigna un vector a cada palabra del diccionario que permita ubicarla en el espacio vectorial del word embedding, y cuya posición será cercana a la de otras palabras con las que guarde una cierta relación semántica.

La forma más clara de comprender este concepto es a través de un ejemplo: en un word embedding la palabra *mesa* estará más cerca de la palabra *silla* que de la palabra *neumático*, porque su significado está más relacionado al de la palabra *silla* que al de *neumático*, estando esta última más cercana en el embedding a palabras como *coche*, *rueda*... Este ejemplo se representa en la Figura 18.

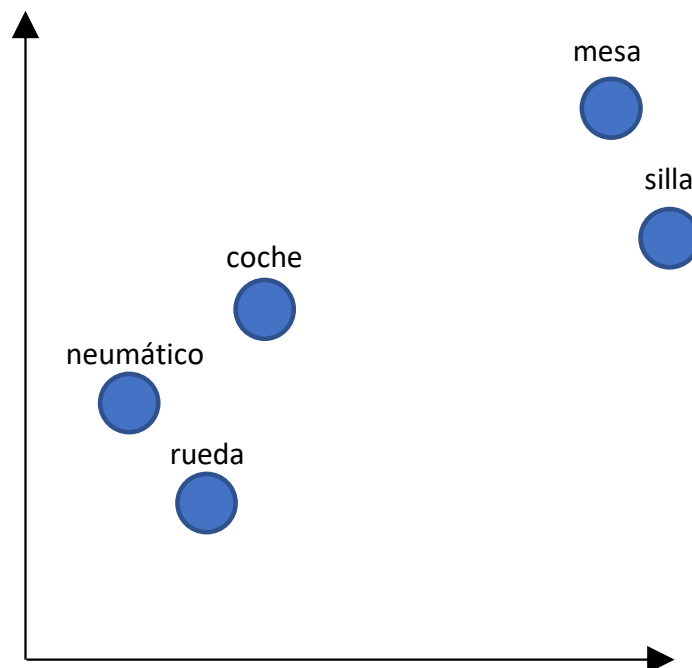


Figura 18: Ejemplo de word embedding

En caso de tener un embedding de tres dimensiones, las palabras anteriores se distribuirían por las tres dimensiones del espacio de forma que las distancias absolutas que las separan sigan dependiendo de las relaciones semánticas entre ellas. La Figura 19 es un ejemplo de un embedding tridimensional.

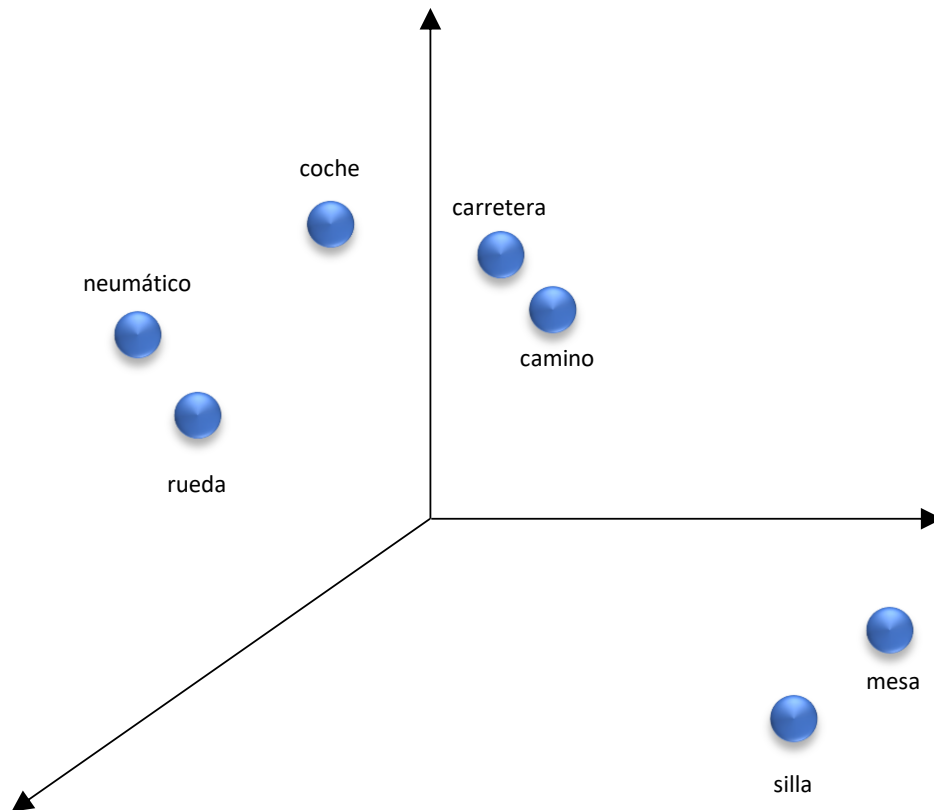


Figura 19: Word Embedding de tres dimensiones

A partir de las cuatro dimensiones resulta complejo hacer una representación gráfica del funcionamiento, pero es fácil intuir por analogía, que las palabras se agruparán en función de sus relaciones semánticas en un espacio vectorial determinado por el tamaño del embedding. Por tanto, cuando en el análisis de una red neuronal recurrente se habla del tamaño del word embedding, se está haciendo referencia al número de dimensiones sobre los que se va a construir ese espacio vectorial. Es decir, un word embedding de tamaño 20 supone que las palabras se van a ubicar en un espacio de 20 dimensiones, por lo que a cada palabra le corresponderán unas coordenadas determinadas en un vector de 20 posiciones.

Como se puede comprobar en las figuras anteriores, a mayor tamaño del embedding mayor será la precisión con el que el sistema puede categorizar a cada palabra y de esta forma conocer las relaciones que existen entre ellas. Por lo tanto, si el valor del embedding se queda corto, la red neuronal no será capaz de categorizar con suficiente precisión las palabras que componen el lenguaje. No obstante, también hay que tener en cuenta que a medida que aumenta el tamaño del embedding la red se vuelve más compleja. Por lo que un valor excesivo podría traer consigo una computación extremadamente lenta y pesada, así como un sobrepaso en el entrenamiento de la red, dando lugar a modelos sobreentrenados.

Por tanto, con el análisis de este hiperparámetro se busca obtener un valor lo suficientemente grande que permita a la red neuronal elaborar una correcta clasificación de las palabras y que el modelo elabore conexiones coherentes entre ellas, pero a su vez se trate de un valor suficientemente bajo como para no alcanzar el extremo opuesto: una extrema complejidad que podría llevar a la red al sobre entrenamiento.

Para alcanzar dicho valor óptimo se comenzarán analizando versos de valores muy diversos del tamaño del embedding, y a medida que se avance en el análisis, se irá concretando en los alrededores de los valores que arrojen mejores resultados en el análisis inmediatamente anterior.

5.5.1 OBTENCIÓN DEL VALOR ÓPTIMO DEL TAMAÑO DEL EMBEDDING

Como ocurría con la temperatura, se busca establecer qué tamaño del embedding traerá unos resultados óptimos para el entrenamiento de la red neuronal. Se realizará por tanto un estudio de cómo influyen los cambios en el tamaño del embedding si se mantienen el resto de valores constantes a la hora de crear el modelo, y esa influencia se estudiará de acuerdo a los resultados que arrojen todas las distintas variaciones del modelo tras someter a los versos de salida de los mismos a los patrones de valoración, tanto artística como gramatical, desarrollados en el apartado 5.1.

Como primer acercamiento, y dado que no se tiene ningún conocimiento previo para establecer un valor de referencia en torno al cual empezar el análisis, se comenzará estudiando los modelos que genera la red neuronal recurrente a partir de un tamaño del embedding igual a 100. A partir de ahí, se irá aumentando 100 unidades el valor del hiperparámetro hasta alcanzar los 600. Y en caso de que los mejores resultados se dieran en alguno de los extremos del intervalo anterior, se ampliará el rango de valores en la segunda fase del análisis.

Los resultados se muestran en la Tabla 12, y se representan gráficamente en la Figura 20.

EMBEDDING SIZE	NOTA GRAMATICAL	NOTA ARTISTICA	MEDIA
100	7,9	8,1	8
200	6,9	7	6,95
300	6,3	6,3	6,3
400	6,2	6,3	6,25
500	6,3	6,6	6,45
600	5,7	6,2	5,95

Tabla 12: Resultado del primer análisis del embedding

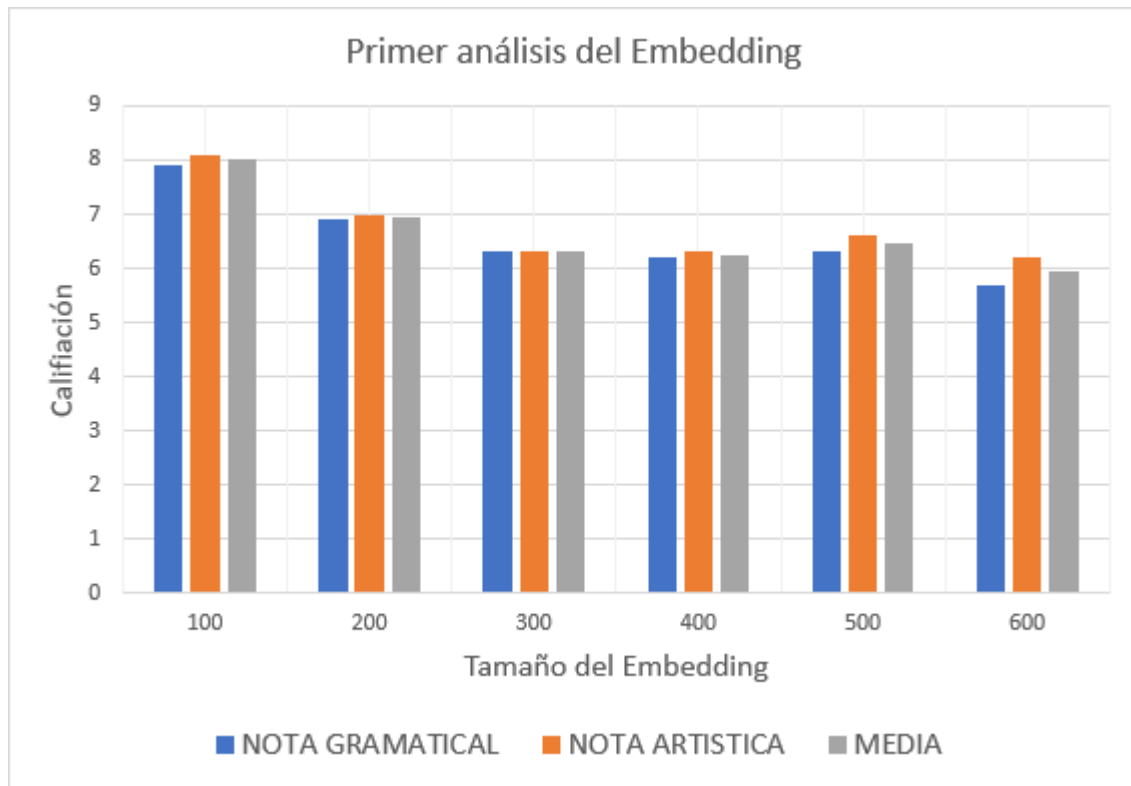


Figura 20: Primer análisis del Embedding

Como se puede comprobar en la Figura 20, el mejor resultado se obtiene para el primer valor de estudio: un tamaño del embedding igual a 100. Con los valores superiores, el resultado cae ligeramente a partir de 200 hasta que la caída se hace notable a partir de un valor de 600. Por tanto, se concentrará el foco en torno al valor del óptimo, y se reducirá la diferencia del tamaño del embedding entre los diferentes modelos a analizar.

De esta forma, el segundo análisis se centrará en el valor de 100 para el tamaño del embedding, y las variaciones de este entre las muestras serán de 25 unidades, por lo que el abanico de valores analizados recorrerá tamaños del embedding desde 25 hasta 175. Esto permitirá establecer cuál es el valor idóneo para este hiperparámetro.

De nuevo, se analizarán 10 versos para cada valor del tamaño del embedding, y se establecerán las notas gramaticales, artísticas y medias atendiendo a los patrones desarrollados en el apartado 5.1.

Los resultados obtenidos para este análisis detallado se muestran en la Tabla 13.

EMBEDDING SIZE	NOTA GRAMATICAL	NOTA ARTISTICA	MEDIA
25	5,1	5,3	5,2
50	6,1	6,2	6,15
75	6,6	6,4	6,5
100	6,8	6,7	6,75
125	6,7	6,9	6,8
150	7,9	7,4	7,65
175	6,8	6,9	6,85
200	6,1	6,4	6,25

Tabla 13: Análisis detallado del tamaño del embedding

Para una mejor comprensión de los resultados, estos mismos se representan gráficamente en la Figura 21.

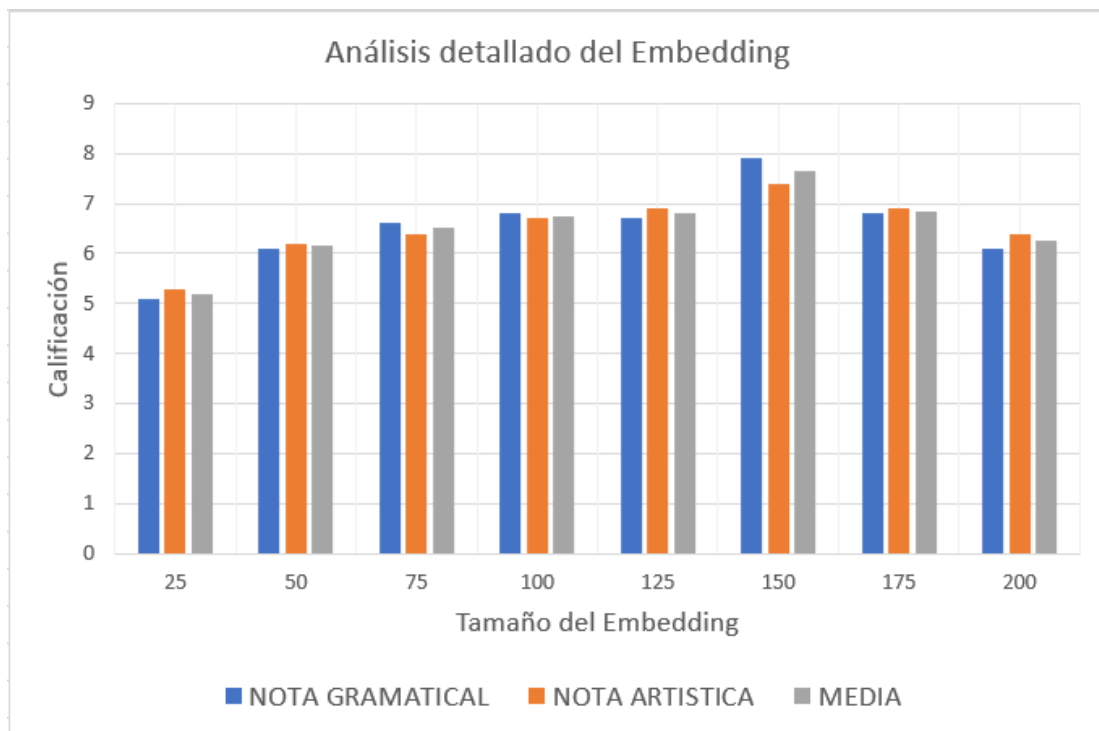


Figura 21: Análisis gráfico detallado del tamaño del embedding

5.5.2 ERROR EN FUNCIÓN DEL EMBEDDING

Una de las técnicas más empleadas en clasificación es la comparación entre los errores en el conjunto de entrenamiento y el conjunto de test.

En el caso de esta red neuronal aplicada a la generación de texto, el error en el conjunto de entrenamiento se calcula al elaborar la red de forma directa. Sin embargo, es complicado establecer el error en el conjunto de test, ya que no se trata de una red de clasificación sino de una red de generación. Por tanto, se necesita una forma de establecer un error en la generación de texto (test error) y compararlo con el error en la creación del modelo a partir del conjunto de entrenamiento (training error).

Para ello, se establece el error en el conjunto de test como la diferencia de la calificación máxima (en nuestro caso, 10) menos la calificación obtenida (nota media) por el modelo en la valoración de los versos, y para tener una magnitud similar al del error de entrenamiento, se dividirá el resultado de esta diferencia entre la nota máxima, como se muestra en la ecuación (5).

$$(5) \quad \text{Error de test} = \frac{\text{Nota Máxima} - \text{Nota Media}}{\text{Nota máxima}}$$

Con esta operación se pretende imitar el error en el conjunto de test de la mejor forma posible. En la Tabla 14 se muestran ambos errores en función de los distintos valores del tamaño del embedding.

Embedding	Training Error	Nota media	10 - Nota media	Test Error
25	0,3954	5,2	4,8	0,48
50	0,3313	6,15	3,85	0,385
75	0,3133	6,5	3,5	0,35
100	0,2775	6,75	3,25	0,325
125	0,257	6,8	3,2	0,32
150	0,2505	7,65	2,35	0,235
175	0,2459	6,85	3,15	0,315
200	0,2348	6,25	3,75	0,375
300	0,2294	6,3	3,7	0,37
400	0,223	6,25	3,75	0,375
500	0,2244	6,45	3,55	0,355
600	0,2212	5,95	4,05	0,405

Tabla 14: Error en función del embedding

Una vez se ha establecido una forma de calcular el error de test, se representa gráficamente junto al error de entrenamiento en la Figura 22.

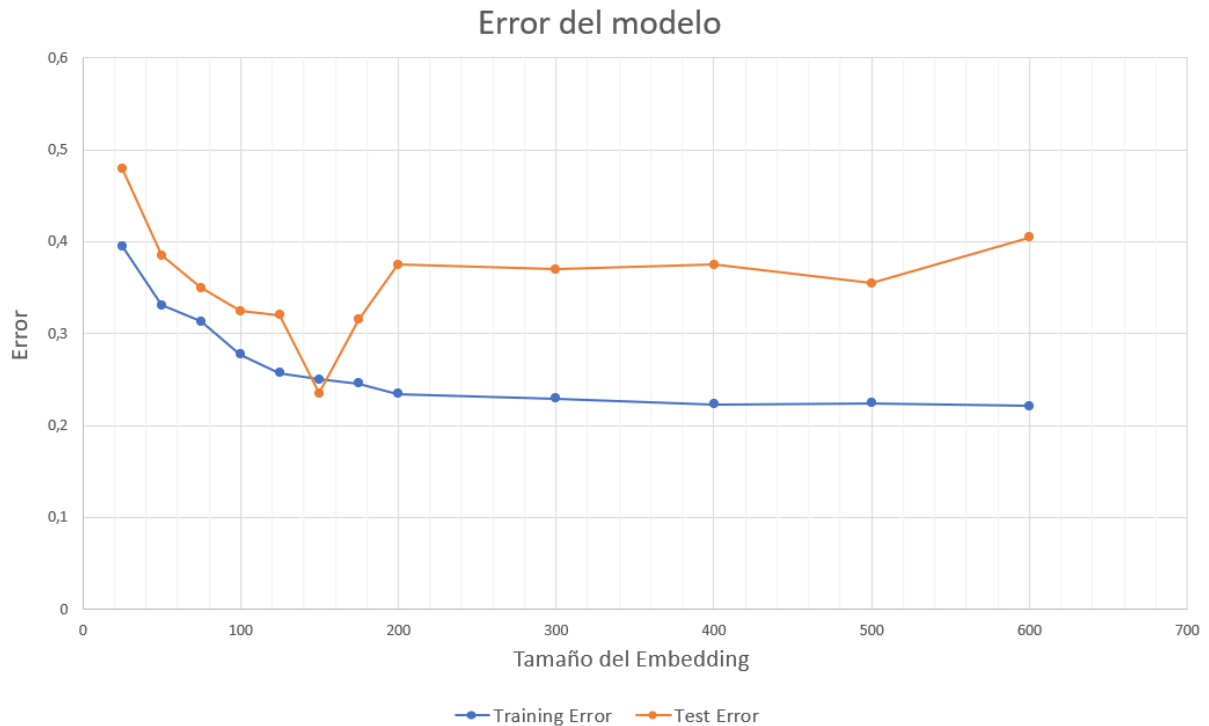


Figura 22: Error en el conjunto de entrenamiento y de test

Como se puede comprobar en la gráfica, conforme aumenta el tamaño del embedding, disminuye el error en entrenamiento. Esto resulta evidente: la propia red neuronal crea un embedding atendiendo a las relaciones entre las palabras de los versos que constituyen el conjunto de entrenamiento. A mayor tamaño, más específicas serán esas relaciones y por tanto el modelo se amolda en mayor medida a los datos de entrada. Sin embargo, si observamos la curva de test, el error disminuye gradualmente conforme aumenta el tamaño del embedding hasta alcanzar un pico, a partir del cual vuelve a aumentar.

Realizando una analogía con lo que sucedería en un problema de clasificación, ese pico se trata precisamente del punto óptimo, en el que se encuentra el mejor compromiso entre el error en el conjunto de entrenamiento y el error del conjunto de test. Para valores superiores del tamaño del embedding el modelo estará sobre entrenado: no merece la pena aumentar

más el valor del tamaño del embedding porque los resultados no mejorarán. Mientras que para valores inferiores se encontrará infra entrenado: habrá que aumentar el tamaño del embedding ya que los resultados en el conjunto de test mejorarán.

También cabe señalar el hecho de que, al tratarse de una analogía, los resultados no son tan fiables como pudieran serlo en un problema de clasificación. Sobre todo, en lo que concierne a la curva del error en el conjunto de test. Un ejemplo evidente de ello es el hecho de que en la Figura 22 se puede observar cómo en el punto óptimo el valor del error es inferior en el conjunto de test respecto al obtenido en el conjunto de entrenamiento, algo que resulta imposible en un caso real de clasificación, siendo siempre el error de test superior al de entrenamiento.

5.6 NÚMERO DE NEURONAS

Como se ha desarrollado en el Capítulo 2, las redes neuronales se dividen en un conjunto de capas, que a su vez son una agrupación de un determinado número de neuronas. Las neuronas son unidades de procesamiento de la información, cuya estructura ya se ha explicado en la Figura 3. El número de neuronas que forman cada una de las capas de una red neuronal es un hiperparámetro, y como tal es un valor variable que se establece previamente al entrenamiento del modelo.

Un aspecto que tener muy en cuenta acerca del número de neuronas, es que se trata de un hiperparámetro ligado de forma muy estrecha a la velocidad de entrenamiento de la red neuronal, de forma que si el número de neuronas es excesivamente elevado puede derivar en un tiempo de entrenamiento de muchas horas o incluso días. Del mismo modo, un aumento en el número de neuronas conlleva un aumento de la complejidad del modelo, dando lugar a archivos más pesados y difíciles de manejar.

En caso de tratarse de una red neuronal con varias capas de neuronas, el número de neuronas de cada capa puede ser diferente. En el caso de la red neuronal utilizada en este proyecto, se ha elaborado utilizando dos capas de neuronas, como se ha determinado con el razonamiento desarrollado en el Capítulo 5.2. No obstante, con el objetivo de simplificar el análisis y obtener un resultado que sirva de referencia para la aplicación de redes LSTM al procesamiento de texto, se ha realizado el análisis con el mismo número de neuronas para ambas capas, de modo que se pueda aplicar al análisis de otras redes con un número distinto de capas.

5.6.1 OBTENCIÓN DEL VALOR ÓPTIMO DEL NÚMERO DE NEURONAS

De nuevo, el objetivo es establecer el valor o rango de valores de este hiperparámetro para obtener unos resultados óptimos. Y como ocurría con el tamaño del embedding, se tratará de un proceso dividido en dos fases. Como no existe un valor establecido como idóneo, sino que se trata de un ensayo prueba-error, se comenzará ampliando un amplio abanico de valores, que va desde las 100 hasta las 700 neuronas, con saltos de 100 neuronas entre los diferentes modelos. Estos valores de los límites se han elegido porque para valores inferiores a 100 neuronas la red no tiene una potencia suficiente como para elaborar versos suficientemente correctos, mientras que para valores superiores a 700 neuronas el proceso de entrenamiento de la red resulta excesivamente pesado.

Además, en este caso se han empleado dos valores que no son múltiplos de 100: 256 y 512 neuronas. Esto se debe a que ambos comparten una característica fundamental, son potencias de base 2, con lo que ello supone en la computación al estar basada fundamentalmente en código binario. Como 512 es realmente cercano a 500 simplemente se ha analizado la red para 512 neuronas, mientras que como 256 se encuentra a mitad de camino entre 200 y 300 se ha añadido al estudio como nuevo valor. En la Tabla 15 se representan los resultados obtenidos.

NEURONAS	NOTA GRAMATICAL	NOTA ARTISTICA	MEDIA
100	6,6	6,7	6,65
200	7,3	7,1	7,2
256	7,7	7,4	7,55
300	7,9	7,2	7,55
400	7,9	7,1	7,5
512	8,3	7,5	7,9
600	7,7	7,7	7,7
700	7,2	7,5	7,35

Tabla 15: Primer análisis del número de neuronas

Como se ha ido realizando a lo largo del proyecto, para una mejor comprensión de los resultados obtenidos, también se representan gráficamente en la Figura 23.

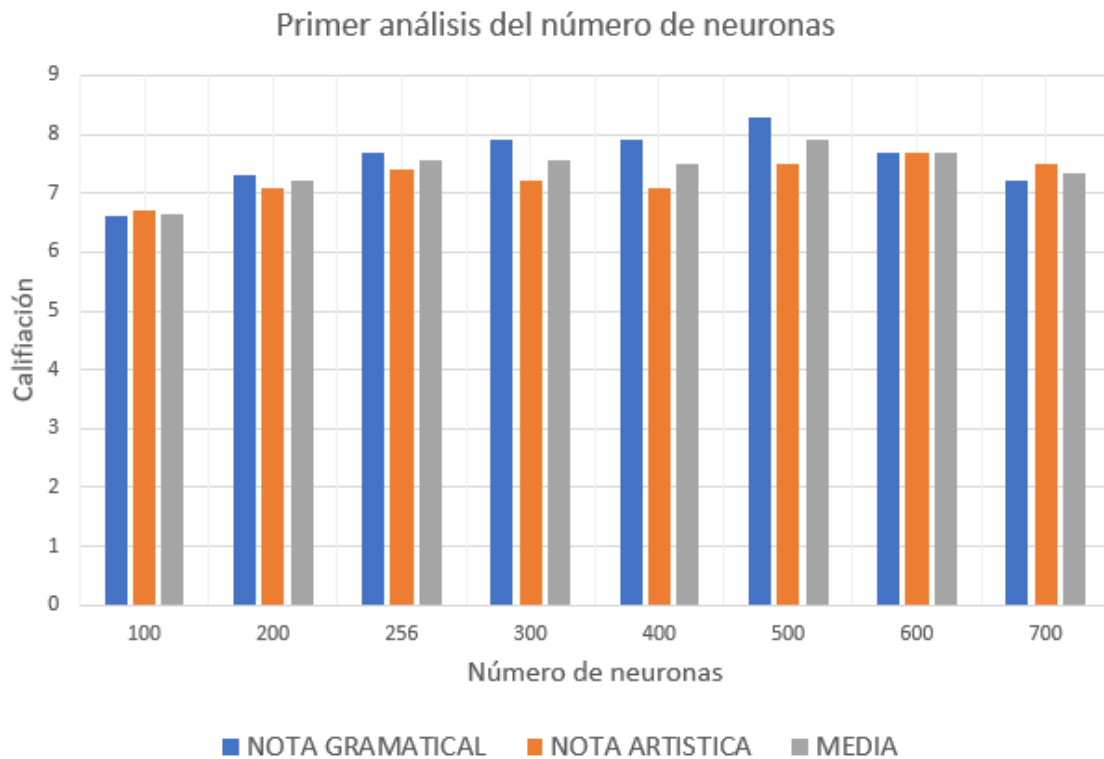


Figura 23: Primer análisis del número de neuronas

De la gráfica anterior se puede sacar una conclusión clara: para valores de neuronas bajos, la calidad de los versos va aumentando a medida que crece el número de neuronas de la red, hasta que la red alcanza valores cercanos a 512, a partir del cual la calidad de los versos comienza a decaer. Pero también es evidente que, pese a que aumenta considerablemente la complejidad del modelo, con todos los inconvenientes que ello conlleva como un mayor tiempo de creación del modelo y archivos más pesados y difíciles de manejar, la calidad no aumenta notoriamente si se comparan los resultados entre 256 y 512 neuronas.

Por tanto, en el análisis detallado se va a poner el foco en dos zonas: la primera es aquella donde se ha encontrado un mejor resultado: alrededor de las 512 neuronas; mientras que la segunda zona a tener en cuenta es aquella donde se han obtenido buenos resultados con un número más contenido de neuronas: alrededor de las 256.

En la Tabla 16 se muestran los resultados obtenidos, con una mayor resolución alrededor de los puntos establecidos anteriormente. En la Figura 24 por su parte, se muestran estos resultados de forma gráfica.

NEURONAS	NOTA GRAMATICAL	NOTA ARTISTICA	MEDIA
100	6,6	6,7	6,65
200	7,3	7,1	7,2
225	7,5	7,4	7,45
256	7,7	7,4	7,55
275	6,6	7,3	6,95
300	7,9	7,2	7,55
400	7,9	7,1	7,5
425	7,7	7,9	7,8
450	7,1	7,1	7,1
475	7,4	7,5	7,45
512	8,3	7,5	7,9
525	7,3	7,5	7,4
550	7,9	7,9	7,9
575	7,6	7,8	7,7
600	7,7	7,7	7,7
700	7,2	7,5	7,35

Tabla 16: Análisis detallado del número de neuronas

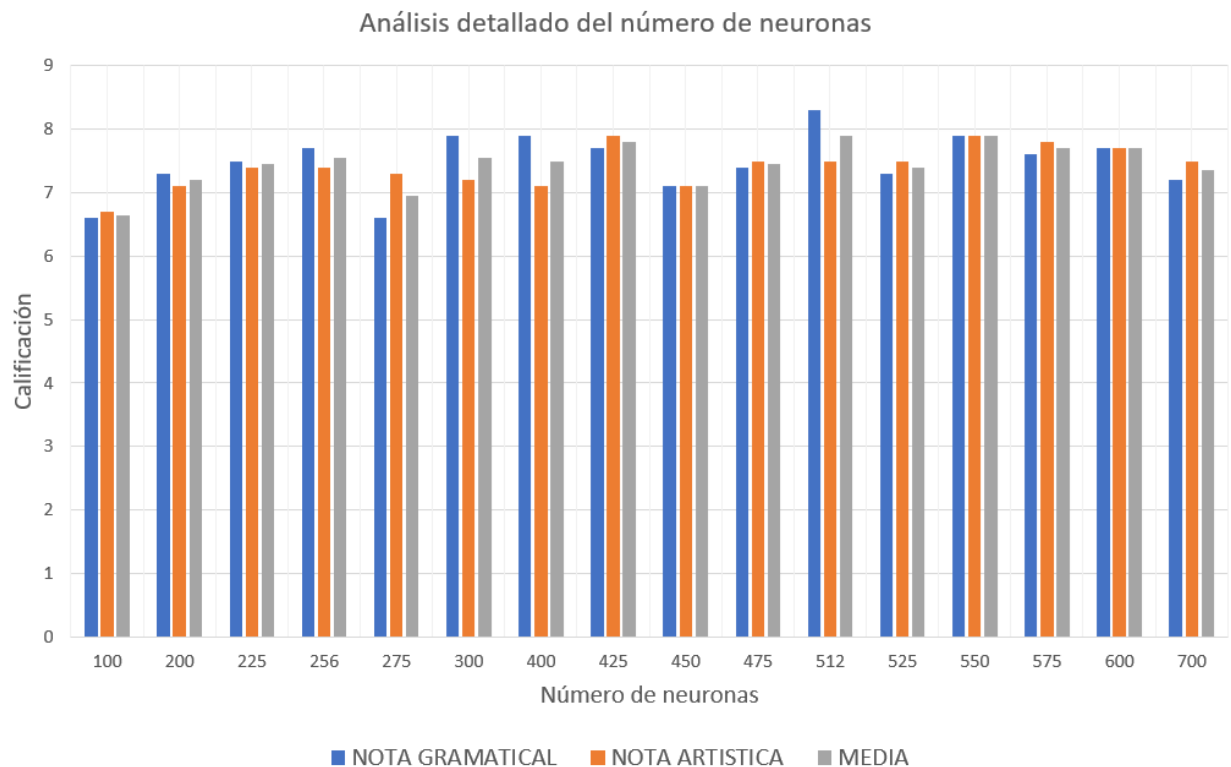


Figura 24: Análisis detallado del número de neuronas

Con estos resultados se pueden resaltar varias cosas. La primera de ellas es que la influencia del número de neuronas en el comportamiento de la red neuronal no encuentra unos valores máximos en un intervalo claro, sino que hay un abanico más amplio de valores con unos resultados parejos en cuanto a calificación que si lo comparamos con la influencia del tamaño del embedding en el apartado 5.5.1, donde resultaba más evidente qué valores eran los que tenían un mejor resultado en el modelo.

Por otro lado, también debe ser resaltado cómo en los valores del número de neuronas que son potencias de 2, los resultados alcanzan un máximo local: son mejores que los obtenidos en su entorno. Además, el máximo absoluto se alcanza para 512 neuronas, de nuevo una potencia del número 2.

Esta situación, se traduce en una complicada decisión para la elección del valor óptimo del número de neuronas, en la que pueden aparecer dos escenarios distintos. El primero es elegir el valor que ha tenido una mejor calificación media, que en este caso sería 512 neuronas, este escenario sería el elegido en caso de que se buscara la máxima calidad posible en los resultados.

El otro escenario posible es que, dado que no existe una gran diferencia de calidad entre el máximo y otros resultados para un menor tamaño de neuronas, se elija el valor que alcance un compromiso entre la complejidad del modelo y la calidad de los resultados obtenidos, se busca el resultado óptimo. En este caso cabe señalar que a medida que aumenta el número de neuronas, de la misma forma lo hace la complejidad del modelo. Para encontrar dicho compromiso se realizará el análisis del error en función del número de neuronas.

5.6.2 ERROR EN FUNCIÓN DEL NÚMERO DE NEURONAS

Como se realizó con el tamaño del embedding en el apartado 5.5.2, con este análisis el objetivo es establecer el mejor compromiso entre la complejidad del modelo elaborado y la calidad de los versos obtenidos, a través del estudio de los errores cometidos por el modelo, tanto en el proceso de entrenamiento, a lo que se ha denominado Training Error, así como en la creación de los versos, a lo que se ha denominado test-error. Esto se debe a que de nuevo se va a realizar una analogía entre el funcionamiento de la red neuronal con el modo de funcionamiento de una red de clasificación.

Resulta necesario recalcar que la nota obtenida por los versos se trata de un método de evaluación que, pese a estar guiado por una serie de patrones establecidos de antemano, es puramente subjetiva, a diferencia de una red de clasificación donde la objetividad de los resultados es incuestionable. Por tanto, la curva no será exacta, pero nos permitirá aproximar alrededor de qué valores se encuentran los resultados óptimos.

Los valores obtenidos se muestran tanto en la Tabla 17 como en la Figura 25.

Nº Neuronas	Training Error	Nota media	10 – Nota media	Test Error
100	0,5503	6,65	3,35	0,335
200	0,2950	7,2	2,8	0,280
225	0,2685	7,45	2,55	0,255
256	0,2505	7,55	2,45	0,245
275	0,2437	6,95	3,05	0,305
300	0,2413	7,55	2,45	0,245
400	0,2237	7,5	2,5	0,250
425	0,2229	7,8	2,2	0,220
450	0,2205	7,1	2,9	0,290
475	0,2202	7,45	2,55	0,255
512	0,2191	7,9	2,1	0,210
525	0,2190	7,4	2,6	0,260
550	0,2190	7,9	2,1	0,210
575	0,2181	7,7	2,3	0,230
600	0,2193	7,7	2,3	0,230
700	0,2188	7,35	2,65	0,265

Tabla 17: Error de entrenamiento y de test en función del número de neuronas

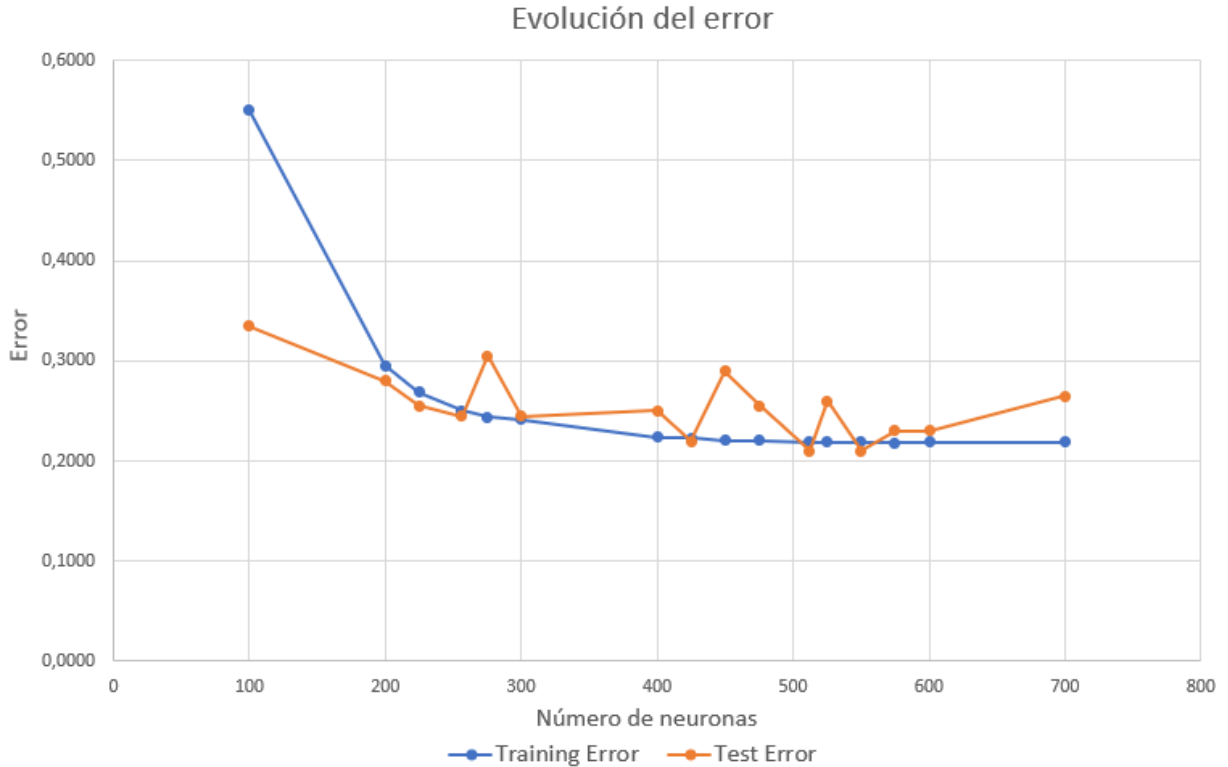


Figura 25: Evolución del error con el número de neuronas

En la gráfica anterior se comprobó como el error en el entrenamiento decae enormemente en los primeros valores, hasta alcanzar un valor alrededor de 0.22 donde continúa decayendo, pero de una forma mucho menos pronunciada. Por su parte en el error con el conjunto de test, para lo cual se ha determinado una calificación de error a partir de la valoración de los versos, se pueden distinguir tres zonas claras.

La primera zona es la parte izquierda del gráfico, para valores inferiores a 200 neuronas, donde se puede comprobar como el error de test disminuye a medida que aumenta el número de neuronas. La zona de la derecha, a partir de 550 neuronas, donde se comprueba que los resultados empeoran si se sigue aumentando el número de neuronas. Y finalmente la zona central, donde el comportamiento del error no es nada constante, y nos encontramos valores tanto muy altos como realmente bajos. Esto puede ser debido a la naturaleza subjetiva de la valoración. No obstante, sí que resulta claro que, en todo ese rango de valores, se alcanza el mínimo error en el conjunto de test es para 512 neuronas, pero es un resultado realmente

similar al obtenido para valores inferiores de neuronas, que por tanto tienen una menor complejidad.

Por lo que el resultado que arroja un mayor compromiso entre calidad del modelo es aquel donde se encuentra el codo en la Figura 25, es decir, 256 neuronas.

5.7 GENERACIÓN DE ESTROFAS

Una vez se han determinado los valores los valores de los hiperparámetros que optimizan la red LSTM, se procede a comprobar si los resultados obtenidos son generalizables a otro tipo de datos de entrada, o si por el contrario es una combinación de hiperparámetros únicamente válida para una red de generación de versos.

Por ello, como última fase en la optimización de la red neuronal se ha decidido alterar la base de datos de entrada de la red neuronal. Mientras que anteriormente era una serie de secuencias de versos, que comenzaban con un token de comienzo de verso que se ha llamado “inicio” y terminaban con token fin de verso, denominado “eof”. En esta actualización de la red los datos de entrada serán estrofas de versos.

Las secuencias de entrada ahora comenzarán en un archivo de inicio de estrofa y no terminarán con un token fin de verso, sino con un token fin de estrofa, al que se ha llamado “eop”. Esto supone que dentro de una estrofa pueden o no agruparse varios versos distintos, y el generador no terminará su labor al encontrar un archivo “eof” sino que lo hará al encontrar un archivo “eop”. Encontrar un archivo “eof” lo único que conllevará es que se comenzará la creación de un nuevo verso, pero de la misma estrofa.

De esta forma, el único cambio existente en la red respecto a lo realizado anteriormente se encuentra en las secuencias de entrada, ahora mayores al ser secuencias en vez de versos. El objetivo de este último análisis es doble: por un lado, se busca que los versos de salida estén más relacionados entre sí: que dejen de ser versos sueltos sino que tenga una interconexión. Por otro lado también se pretende analizar si los hiperparámetros, conservando los valores analizados como óptimos cuando las secuencias eran versos sueltos, mantienen sus buenas características al cambiar el formato de los datos de entrada, o si por el contrario simplemente son útiles para la red de generación de versos desarrollada previamente. Si el análisis resulta satisfactorio, los valores de los hiperparámetros obtenidos podrían ser generalizables a cualquier red de generación de texto.

Capítulo 6. ANÁLISIS DE RESULTADOS

En este capítulo se van a analizar los resultados obtenidos, para determinar el grado de influencia que tienen en la red neuronal cada uno de los principales aspectos de estudio a lo largo de la realización del proyecto. El orden seguido es el idéntico al orden de estudio de estos.

6.1 RESULTADO DEL NÚMERO DE CAPAS DE LA RED

El primer aspecto analizado de una red neuronal LSTM ha sido el número de capas que la constituyen. Para ello se han analizado los versos de salida de modelos generados manteniendo todos los hiperparámetros constantes y únicamente variando el valor del número de capas.

Los valores entre los que se ha estudiado el tamaño de la red variaban desde la utilización una única capa hasta la inclusión de cuatro capas, debido a que para mayores valores el entrenamiento de la red resulta excesivamente lento y costoso, dado que la complejidad de la red aumenta enormemente; además de porque a la vista de los resultados se ha comprobado que la calidad de los versos comenzaba ya a decrecer a partir de la inclusión de la cuarta capa. Esto se debe a que con una sola capa el entrenamiento al que se ha visto situado el modelo no es suficiente, mientras que, a partir de la cuarta capa, el modelo ha sufrido un proceso de sobre entrenamiento que ha traído consigo un empeoramiento de la calidad de los versos resultantes.

Los resultados obtenidos, que se muestran en la Tabla 3, muestran claramente como los resultados mejoran con la inclusión de la segunda capa, y empeoran cuando se hace lo propio con la cuarta. No obstante, no hay una razón de peso para decantarse por la utilización de dos o tres capas, al menos analizando simplemente el valor de las calificaciones.

Por ello, era necesario establecer un factor de decisión que permita elegir cuál es el tamaño óptimo, y este ha sido la complejidad del modelo. Como se puede comprobar en el peso de los archivos generados, el modelo de dos capas es un 30% menos pesado que el de tres capas, lo que se debe a la existencia de una capa menos, y por tanto un menor número tanto de neuronas como de conexiones entre ellas: lo que aumenta tanto la complejidad del modelo como el tiempo empleado en el proceso de creación de este.

6.2 RESULTADO DE LA TEMPERATURA

El primer hiperparámetro analizado individualmente ha sido la temperatura de excitación del modelo.

En un primer momento, se ha mantenido el valor de la temperatura constante durante todo el proceso de generación de los versos. En este primer análisis, gracias al sistema de valoración de versos elaborado en el capítulo 5.1, se ha determinado que el valor óptimo si la temperatura es constante resulta ser 1,3. Al tratarse de un valor mayor que 1, como se ha explicado en el capítulo 5.4, la función softmax encargada de establecer el vector de probabilidades se vuelve menos restrictiva que en caso de utilizar un valor inferior, lo que supone un aumento de la diversidad en los versos, pero también puede tener como principal desventaja una pérdida de la cohesión en los versos resultantes, puesto que hace que la generación de versos relativice las probabilidades obtenidas en el proceso de entrenamiento, ganando flexibilidad pero a costa de perder seguridad.

Este resultado es significativo, y puede indicar que en las tareas de generación de texto es necesario elevar ligeramente la temperatura por encima de la unidad de forma que la red gane un poco de libertad y pierda ese comportamiento más mecánico y rígido que otorgan las bajas temperaturas, con el objetivo de crear versos más naturales y espontáneos. No obstante, los resultados con temperaturas superiores a 1,3 empeoraban también sus resultados, dando lugar a versos extremadamente largos e inconsistentes.

Como el aumento de la temperatura era favorable para el comienzo de los versos, puesto que aumentaba la variedad de palabras con la que se empezaban las secuencias, pero se convertía en negativo a medida que se avanzaba en el verso, puesto que traía consigo inconsistencia y versos extremadamente largos al disminuir la probabilidad del elemento fin de verso, se ha realizado un segundo análisis de la temperatura. Esta vez, el valor de la temperatura no ha sido fijo a lo largo de la generación de todo el verso, sino que iba disminuyendo gradualmente desde un valor inicial de temperatura, a través del producto de ese valor inicial y una función exponencial con una base decimal menor que uno, a la que se ha denominado decay, como se muestra en la ecuación (4).

De nuevo empleando la valoración de los versos de salida, se ha establecido cuál es la combinación de valores de temperatura inicial y decay con la que se obtienen unos mejores resultados de calidad en los versos. Finalmente, esa combinación ha resultado ser 2.3 para el valor de la temperatura inicial y 0.87 para el valor del decay. Lo que da lugar al siguiente vector de temperaturas, en función de la posición de cada palabra en el verso la temperatura con la que se obtiene la misma es: $[2,3 - 2,00 - 1,74 - 1,51 - 1,31 - 1,14 - 0,99 - 0,87 - 0,75 - 0,65 - 0,57 - 0,49 - 0,43 - 0,4 - 0,4 - \dots - 0,4]$. Para evitar colapsar la generación, se introduce un valor mínimo de la temperatura lo suficientemente bajo. A partir del momento en el que la temperatura alcanza ese valor, esta se mantiene constante e igual a este, para el que se ha elegido 0,4.

Los resultados obtenidos con este enfoque de la temperatura decreciente han sido considerablemente mejores a los obtenidos bajo una temperatura constante, por lo que este segundo análisis ha sido beneficioso. Y esto se debe a que una temperatura elevada es idónea para el comienzo de los versos, puesto que incluye una gran variedad y riqueza de palabras, mientras que a medida que se avanza en un verso se requieren las mejores prestaciones de la temperatura baja: consistencia y longitud comedida.

6.3 RESULTADO DEL TAMAÑO DEL EMBEDDING

Se llama word embedding al espacio dimensional en el que el modelo establece las relaciones entre las palabras de un lenguaje a través de la distancia existente entre ellas. El tamaño del embedding es por ello la dimensión de dicho espacio dimensional.

El análisis de este hiperparámetro ha sido realizado en dos fases. En una primera fase el rango de los valores entre los que se ha estudiado era más amplios, partiendo de un tamaño de 100 dimensiones hasta alcanzar las 600, con saltos de 100 entre los modelos. El mejor resultado se ha obtenido para un tamaño de 100, y a partir de ese punto los resultados decaían considerablemente.

A partir del resultado del primer análisis, se ha puesto el foco en los valores del entorno de 100, en esta ocasión se ha recorrido un rango que se iniciaba en 25 hasta alcanzar un tamaño de 200 con saltos de 25 entre los modelos. Como resultado de esta segunda fase, se ha determinado que el valor que maximiza los resultados es un tamaño del embedding de 150, como se puede apreciar en la Figura 21.

De este resultado, se puede determinar que existen dos zonas diferentes en función del tamaño del embedding. La primera zona es aquella donde el tamaño del embedding es menor que 150 y por ello insuficiente para realizar una correcta clasificación de las palabras, y por ello los resultados mejoran si se aumenta el tamaño del embedding. En esta zona el modelo está infra entrenado. Por otro lado, cuando el tamaño del embedding es superior a 150 los resultados empeoran si se continúa aumentando el valor del hiperparámetro. Se trata de una zona de sobre entrenamiento, puesto que el tamaño del embedding es tan grande que genera incluso conexiones extras entre las palabras que traen consigo un peor desempeño de la red neuronal: el modelo está excesivamente ajustado a los datos de entrada.

Por tanto, el valor óptimo del tamaño del embedding que se ha obtenido es 150. Para valores superiores tendremos problemas de sobre entrenamiento, mientras que para valores inferiores sucederá justamente lo contrario: el entrenamiento del modelo es insuficiente.

6.4 RESULTADO DEL NÚMERO DE NEURONAS

El resultado del análisis del número de neuronas es el menos evidente de todos los hiperparámetros analizados. Este hiperparámetro se refiere a la cantidad de neuronas que forman cada una de las capas del modelo, que como se ha determinado en el 5.2, se trata de dos capas.

Como se ha hecho con los anteriores, primeramente se ha analizado un amplio rango de valores del hiperparámetro, que va desde 100 hasta 700 neuronas, con variaciones de 100 neuronas entre los modelos de estudio. En este caso, los resultados obtenidos han reflejado dos zonas evidentes, una zona en la que el número de neuronas es inferior a 200 donde la calidad de los versos mejora si se aumenta el valor del hiperparámetro. En esa zona la red no está suficientemente entrenada, ya que merece la pena aumentar el valor del hiperparámetro. Por otro lado, se puede apreciar una segunda zona, para valores mayores de 600, en la que claramente los resultados empeoran si se aumenta el número de neuronas. En esa zona aumentar el valor del hiperparámetro supondrá que el modelo se ciña de una forma más ajustada todavía al conjunto de datos de entrada, por lo que se producirá un problema de sobre entrenamiento.

Sin embargo, existe una zona entre 200 y 600 neuronas donde los resultados fluctúan y no es fácil detectar un comportamiento claro en función del número de neuronas. Por ello se ha realizado el segundo análisis poniendo el foco entre estos dos valores, y se reduce la variación del número de neuronas entre los modelos a 25. Además se han agregado dos valores al estudio, 256 y 512, dado que son potencias directas de 2 y por ello tienen una mayor trascendencia en código binario.

Con este análisis se obtienen los resultados de la Figura 24 y de la Figura 25. En ellos se observa que el mejor resultado se obtiene para 512 neuronas, no obstante se trata de un valor relativamente cercano al obtenido para un valor de neuronas considerablemente menor: 256. Por tanto el valor del número de neuronas con mejores resultados puede tomar dos alternativas.

En caso de que se esté trabajando con una unidad de procesamiento y computación que pueda gestionar de forma holgada los datos y la creación de la red neuronal, se determinará como número de neuronas 512, ya que los resultados obtenidos son los mejores.

No obstante, al tratarse de un proceso de computación complejo, y de cara a agilizar el proceso de creamiento de los modelos, se elegirá el valor de 256 neuronas en caso de que no se disponga de un gran margen de procesamiento del modelo. Además al tratarse del punto donde se sitúa el codo en la gráfica del error de la Figura 25, 256 neuronas es el valor del número de neuronas que asegura un mayor compromiso entre calidad de los versos generados con la complejidad del modelo creado.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

La primera conclusión que se obtiene del desarrollo de este proyecto es la necesidad de cuantificar los resultados, en caso de que estos no sean valores objetivos, para poder establecer la influencia de la variación de cualquier aspecto de una red neuronal recurrente. El sistema de patrones propuesto puede ser utilizado de forma análoga en cualquier otra red generativa, ya sea de texto, pintura, música... Simplemente se requiere de establecer una serie de requerimientos que permitan clasificar las salidas de la red y de este modo compararlas en función de los valores de los aspectos a optimizar.

De cada hiperparámetro analizado se han obtenido una serie de conclusiones, muy útiles a la hora de comprender a fondo el funcionamiento del hiperparámetro y que pueden ser utilizados incluso en otro tipo de redes neuronales.

En cuanto al número de capas, el valor óptimo será aquel que alcance unos resultados correctos sin excederse en el detalle. En este caso, al tratarse de una red de procesamiento de texto se demuestra cómo será suficiente incluir dos o tres capas, no más, ya que los resultados empeoran. Por tanto, el procesamiento de texto requiere de menos capas que por ejemplo la catalogación y clasificación de imágenes, donde la profundidad de la red es un factor fundamental.

Por su parte, el análisis de la temperatura ha demostrado la importancia de establecer una temperatura decreciente, para evitar tanto una gran monotonía al comienzo de los versos generada por una temperatura baja, como una excesiva incoherencia de las palabras generada por una temperatura excesivamente alta en las últimas palabras del verso. Este uso de la temperatura puede utilizarse también en la generación de otros tipos de obra artística, como en pintura o composición musical, para que la variedad de la obra generada sea amplia, pero a su vez el desarrollo artístico tenga cohesión y coherencia.

El tamaño del word embedding es probablemente el hiperparámetro estudiado más relacionado con el tipo de uso que se le da a la red LSTM: la generación de texto. Como tal, también es uno de los más influyentes, lo que supone que la calidad de ellos versos está estrechamente relacionada con lo cercano que esté el hiperparámetro a su valor óptimo. También es el hiperparámetro en el que se han distinguido con mayor facilidad las zonas de falta y exceso de entrenamiento, siendo incuestionable el valor del embedding para un mejor resultado de la red.

Finalmente, el resultado del análisis del número de neuronas ha planteado un debate evidente si los resultados son parejos: ¿qué tiene más valor, obtener el mejor resultado o ser menos complejo? Evidentemente la respuesta a esta pregunta dependerá enormemente de la finalidad y objetivo del proyecto en concreto, dado que hay sectores en los que un cambio ínfimo puede suponer el éxito o el fracaso. Pero en caso de una aplicación de generación de texto, se ha optado por emplear el hiperparámetro que da como resultado un sistema más sencillo, pues se considera que el aumento de la complejidad no está justificado por tan pequeña mejora en la calidad de los versos.

Para concluir si los resultados obtenidos verdaderamente son significativos en el comportamiento de la red, se realizará la comparación de los primeros diez versos obtenidos previamente a realizar la deducción de los valores óptimos de los hiperparámetros, que se muestran en la Figura 26, con los obtenidos utilizando el modelo creado para la mejor combinación de valores de los objetivos de estudio de la red, que a su vez se muestran en la Figura 27.

```
amigos es agrupadas describidme . una ilusión . decirlo ?  
contacto fundó en nombre casi une ,  
ascienden lo ved esa mismo rosas monedas ,  
arpegio de amor locas acaba en llamamiento ;  
no me granito tápame allí columnillas impasible juzgó súplica tirabuzones durmieron arribamos cánticos .  
aleves como espuma de a vida tirabuzones .  
alza aplicasteis sino uniendo céspedes busca golpeado ,  
trémulamente sentado y humedecido complace .  
engañas , en cambio , mayor modulaba flora que tú tabla  
musical inerme a condesa que par o nuestra requerido sumos ios vagaba predice ondeante iris
```

Figura 26: Primeros diez versos analizados


```
ninfas o miradas ,  
estos pasos en la arena .  
dejadme fuisteis , emergido por la luz como una cera ;  
alguien , hace ya mucho , hace muchísimo , puso con su mano más :  
el rosa está de materia  
alzó falda revuelto remotas últimas , más misma que un sol .  
mi enamorada súbita .  
índemne en peregrina su voz ,  
descifrará acaso bloque sobre finura de dicha .  
aquí olor silencioso aquí color de unos ojos ;
```

Figura 27: Versos del modelo optimizado

Como se comprueba en las imágenes anteriores, el aumento de calidad de los versos es notable, pues en un primer momento las secuencias de palabras eran prácticamente ilegibles y en la Figura 27 la mitad de los versos son gramaticalmente perfectos, y los errores presentes en ellos no opacan el significado del verso, siendo muchos menos groseros que los obtenidos en la Figura 26, por lo que el análisis de los hiperparámetros ha resultado útil: se han mejorado las prestaciones de la red neuronal.

No obstante, los versos obtenidos para los hiperparámetros óptimos, pese a ser mucho mejores que los primeros generados, siguen sin ser perfectos. Sus principales fallos están en la inconexión de los versos de salida, ya que no tienen por qué estar relacionados entre sí, así como su imperfección gramatical, que puede seguir siendo mejorada.

Por todo ello el proyecto continúa abierto a un desarrollo futuro. Este desarrollo puede darse por varias vías, la primera de ellas es cambiar el argumento de entrada de versos a estrofas, es decir, que las nuevas secuencias sean la unión de varios versos, como se ha desarrollado en 5.7. En la Figura 28 se muestra un ejemplo de una estrofa obtenida con este algoritmo.

```
pues le no veía . ¡ oh , yo ,  
no , no es eso . no es un hombre .  
no es un hombre . mas no es un hombre . es la noche .  
y lo mas lo esperando . y es lo que es lo mismo .  
es una sombra que cae y está el mar .  
no es el hombre .  
no es un hombre . ¿ es un niño ? ¿ quién es un hombre ?  
no es un lecho de niño que pasa .  
la noche es un día . ¿ quién qué amo ?  
¿ quién soy tú ?
```

Figura 28: Estrofa obtenida con el algoritmo

Uno de los objetivos de este análisis era comprobar que los hiperparámetros obtenidos no fueran completamente dependientes del formato de los datos obtenidos a la entrada, sino que mantuvieran un buen comportamiento al cambiar el tipo de secuencias con las que se entrena el modelo. Como se puede apreciar a la perfección, los resultados tienen unas condiciones gramaticales buenas, sin ser perfectas, lo que significa que algo muy similar a lo que ocurría con el uso de versos sueltos. Por lo tanto, los valores de los hiperparámetros también son correctos para el uso de estrofas.

Otra posible vía de mejora de los modelos generados es la creación de una red generativa antagónica, más conocida por las siglas de su nombre en inglés, generative adversarial network: GAN. Este tipo de red está formada por dos bloques principales. Por un lado se tiene el bloque generador, que se encargará de generar versos artificiales, por lo que podría utilizarse la red desarrollada a lo largo de este proyecto como generador. Por otro lado se encuentra un bloque clasificador, que recibirá como datos de entrada versos artificiales generados por el modelo y versos reales escritos por el poeta, teniendo que clasificar cada uno de los versos como artificiales o reales en función de su origen. La gran ventaja que presentan las redes generativas antagónicas se sitúa en la mejora simultánea de ambas partes: el clasificador debe mejorar sus prestaciones para que el generador no le engañe con los versos artificiales, mientras que a medida que el clasificador mejora el generador debe hacer lo mismo para hacer pasar sus versos por los del poeta.

Este tipo de redes es utilizado actualmente en ámbitos como la generación automática de caras, como se ve en el estudio realizado por Jon Gauthier para la universidad de Stanford [31], donde se determina que este tipo de redes están entre las más adecuadas dentro de este ámbito. Pero también se pueden emplear en la modificación de imágenes, como se muestra en el estudio llevado a cabo por Brian Dolhansky y Cristian Cantón Ferrer [32], que emplea una GAN para editar los ojos de imágenes de personas.

Por tanto, un posible desarrollo futuro del proyecto podría ser la implementación de una red generativa antagónica, que fuerce al generador a mejorar en la creación de versos.

REFERENCIAS: BIBLIOGRAFÍA

- [1] P. N. Roldán, Artículo “Tecnología”, revista online Economipedia, 21 de agosto de 2017, último acceso: 7/06/2020 <https://economipedia.com/definiciones/tecnologia.html>
- [2] P. Pardo, “El Ordenador del futuro llega a España”, noticia para el periódico El Mundo, 14 junio 2019, último acceso: 9/06/2020 <https://www.elmundo.es/ciencia-y-salud/ciencia/2019/06/14/5d02a1f0fdddf542d8b463a.html>
- [3] G. Vega, “Microsoft admite que se equivocó con su guerra contra el software libre”, artículo para el diario El País, 19 mayo 2020, último acceso: 10/06/2020, <https://elpais.com/tecnologia/2020-05-19/microsoft-admite-que-se-equivoco-con-su-guerra-contra-el-software-libre.html>
- [4] E. Fernández, “Inteligencia Artificial, la Tecnología que se extiende con el coronavirus”, artículo para el diario El Mundo, 21 mayo 2020, último acceso: 22/06/2020 <https://www.elmundo.es/economia/2020/05/21/5ec66629fdddf9c128b466f.html>
- [5] D. Gámez, “Progress in Machine Consciousness.” Estudio de investigación para el departamento de Computer Science, University of Essex, 2007, pp: 880-905, 2007
- [6] C. Fuentes Barassi, “Conciencia e Inteligencia Artificial Consideraciones Críticas sobre la Plausibilidad de que una Máquina Programada Posea Conciencia Fenoménica”, Tesis realizada en la Universidad de Chile, Facultad de Filosofía y Humanidades, pp 43-50, 2011.
- [7] T. M. Mitchell, “Machine Learning”. Libro publicado por la editorial McGraw-Hill Science, marzo de 1997.
- [8] “Deep learning vs. machine learning: ¿qué diferencia hay?”, artículo de la revista digital Ionos, mayo de 2020, último acceso: 20/06/2020, <https://www.ionos.es/digitalguide/online-marketing/marketing-para-motores-de-busqueda/deep-learning-vs-machine-learning/>
- [9] L. Fausett, “Fundamentals of Neural Networks”, libro publicado por la editorial Prentice-Hall, New Jersey, 1994, pp: 39-96
- [10] S. Haykin, “Neural Networks and Learning Machines”, libro publicado por la editorial Pearson Education Ltd, New Jersey, 2008, pp: 46-61
- [11] “Qué son las redes neuronales y sus funciones”, artículo para la revista digital Atria Innovation, octubre 2019, último acceso: 18/06/2020,

- <https://www.atriainnovation.com/QUE-SON-LAS-REDES-NEURONALES-Y-SUS-FUNCIONES/>
- [12] P. Cornieles, “Entendiendo las redes neuronales: De la neurona a RNN, CNN y Deep Learning”, artículo para la revista online IALatam, febrero 2019, último acceso: 25/06/2020
<https://ia-latam.com/2019/02/06/entendiendo-las-redes-neuronales-de-la-neurona-a-rnn-cnn-y-deep-learning/>
- [13] D. Lee et al., "Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus," artículo publicado por China Communications, vol. 14, no. 9, pp. 23-31, Sept. 2017, doi: 10.1109/CC.2017.8068761, septiembre 2007, pp: 23-31.
- [14] J. I. Garzón, “Cómo usar redes neuronales LSTM en la predicción de averías en las máquinas”, artículo para el blog online GFT, 6 de noviembre 2018, último acceso: 9/06/2020
<https://blog.gft.com/es/2018/11/06/como-usar-redes-neuronales-lstm-en-la-prediccion-de-averias-en-las-maquinas/>
- [15] W. B. Rauch-Hindin, “Aplicaciones de la inteligencia artificial en la actividad empresarial, la ciencia y la industria”, Ediciones Díaz de Santos, diciembre 1985. pp 652
- [16] T. Montero, “Genome4care, la medicina del futuro ya está presente” noticia para el diario La Voz de Galicia, 14 junio 2020, último acceso: 25/06/2020
https://www.lavozdeg Galicia.es/noticia/mercados/2020/06/14/genome4care-medicina-futuro-presente/0003_202006SM14P10991.htm
- [17] “Una inteligencia artificial para detectar la gravedad de las lesiones cerebrales”, noticia para El Confidencial, 18 de mayo de 2020, último acceso: 24/06/2020
https://www.elconfidencial.com/tecnologia/ciencia/2020-05-18/inteligencia-artificial-detectar-lesiones-cerebrales_2599228/
- [18] L. Columbus, “Top 25 Machine Learning Startups to Watch in 2020”, artículo para la revista Forbes, 26 abril 2020, último acceso: 22/06/2020
<https://www.forbes.com/sites/louis Columbus/2020/04/26/top-25-machine-learning-startups-to-watch-in-2020/#3cdfad141f52>
- [19] E. Blanco, “¿Es posible diagnosticar el covid-19 con deep Learning?: “Tosa, por favor””, artículo para el blog online de Telefónica Think Big, 20 mayo 2020, último acceso: 25/06/2020
<https://empresas.blogthinkbig.com/es-posible-diagnosticar-el-covid-19-con-deep-learning-tosa-por-favor/>

- [20] A. Sáez de la Pascua, “Deep learning para el reconocimiento facial de emociones básicas”, Trabajo de Fin de Grado para la Universidad Politécnica de Cataluña, Escuela de Telecomunicaciones y Aeroespacial, enero de 2019, pp: 33-41
- [21] D. Foster, “Generative Deep Learning. Teaching Machines to Paint, Write, Compose and Play” Libro publicado por O`Reilly Media Inc, 26 de junio de 2019, PP 37-42
- [22] S. Ferro, “An ai program wrote harry potter fan fiction and the results are hilarious”, artículo para la revista digital Mental Floss, 13 diciembre 2017, último acceso: 24/06/2020 <https://www.mentalfloss.com/article/520897/ai-program-wrote-harry-potter-fan-fiction-and-results-are-hilarious>
- [23] C. Verdú Ruíz “Generador Heurístico de textos inteligente” Proyecto de Fin de Grado en la Universidad Politécnica de Madrid, ETSI Sistemas Informáticos, pp 41-45
- [24] Tensorflow, web oficial, último acceso: 26/06/2020 <https://www.tensorflow.org/>
- [25] Keras, web oficial, último acceso: 26/06/2020 <https://keras.io/>
- [26] Diccionario de la Lengua Española, actualización del año 2019, Real Academia Española, último acceso: 12/06/2020, <https://dle.rae.es/poes%C3%ADa>
- [27] P. Larrañaga, I. Inza y A. Moujahid, “Redes neuronales” fragmento académico de la Universidad del País Vasco, departamento de ciencias de la computación, último acceso_ 30/06/2020 <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t8neuronales.pdf>
- [28] Wikipedia, Función Softmax, noviembre de 2019, último acceso: 20/06/2020 https://es.wikipedia.org/wiki/Funci%C3%B3n_SoftMax
- [29] C. M. Bishop, “Pattern Recognition and Machine Learning”, libro publicado por la Editorial Springer Science Media, Estados Unidos, 2006, pp: 223-231
- [30] C. Wang y Z. Li, "Inversion of Wave Impedance Using Improved Simulated Annealing Genetic Algorithm," artículo publicado en la Tercera Conferencia Internacional sobre Computación y Gestión de Información, IEEE, 2010, pp: 121-124.
- [31] J. Gauthier, “Conditional generative adversarial nets for convolutional face generation”, artículo académico, Stanford University: Symbolic Systems Program, Natural Language Processing Group
- [32] Br. Dolhansky y C. Canton Ferrer, “Eye In-Painting with Exemplar Generative Adversarial Networks”, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 7902-7911, doi: 10.1109/CVPR.2018.00824.

- [33] R. Alonso, “¿Cuánto cuesta la electricidad que consume tu PC?”, artículo para el blog HardZone, 31 de marzo de 2015, último acceso: 24/06/2020,
<https://hardzone.es/2015/03/31/cuanto-cuesta-la-electricidad-que-consume-tu-pc/>
- [34] “El uso de la nube para guardar datos dispara el consumo de energía de servidores”, artículo para el diario La Vanguardia, 11 de marzo de 2015, último acceso: 04/07/2020,
<https://www.lavanguardia.com/vida/20150311/54428065694/el-uso-de-la-nube-para-guardar-datos-dispara-consumo-energia-de-servidores.html>

ANEXO I: OBJETIVOS DE DESARROLLO SOSTENIBLE

De entre los 17 objetivos de desarrollo sostenible establecidos por la ONU, este proyecto se centra fundamentalmente en dos de ellos: la educación de calidad, que es el objetivo número 4; así como en el objetivo número 12: la producción y consumo responsables.

Con respecto al primero de ellos, se puede enfocar el proyecto de dos formas distintas. La primera, se corresponde con la posibilidad que ofrece este proyecto de generar poesía de forma automática. Este robot poeta podría emplearse para emular a poetas que han escrito en lenguas en las que el desarrollo poético existente es muy pobre, dando la oportunidad de educar a las nuevas generaciones en la poesía en su propia lengua materna a través de los versos generados por el algoritmo, y de esta forma fomentar aquellos idiomas en los que no se ha alcanzado un gran desarrollo artístico.

Por otro lado, existe el debate acerca de hasta qué punto es ético desarrollar una máquina capaz de sustituir a las personas en una labor hasta el momento tan arraigada al ser humano como es la creación de una obra artística. El avance tecnológico de las últimas décadas ha provocado que las máquinas hayan superado a la capacidad humana en una enorme cantidad de procesos, fundamentalmente en casos de computación y cálculo numérico, donde los algoritmos existentes son capaces de resolver cálculos y problemas que resultarían imposibles para una persona. Pero hay una serie de ámbitos en los que sin duda no se ha conseguido que las máquinas alcancen el nivel de los seres humanos. Por ello, existe el temor de que la tecnología alcance un nivel capaz de eclipsar al ser humano también en estos aspectos, llegando a un punto en que una máquina pudiera sustituir completamente a una persona.

En esta situación, que el desarrollo de la computación permitiese a las máquinas ser artistas, podría suponer el fin de profesiones humanas tales como pintor, escritor, compositor... No obstante, el desarrollo de este proyecto también debe traer consigo una evolución en la educación artística del futuro, donde el ser humano deberá desarrollar una capacidad de innovar que le permita diferenciarse de las máquinas.

Por tanto, el desarrollo de este proyecto, de acuerdo al objetivo de una educación de calidad establecido, trae consigo el beneficio de la posible creación de una obra poética en lenguas con escasez de este tipo de obras, pero también conlleva un cambio necesario en la educación que permita el desarrollo artístico humano en el momento en que las máquinas también sean capaces de generar una obra artística.

En cuanto al segundo objetivo, en este análisis de los hiperparámetros se han establecido los valores con un mejor compromiso entre calidad y complejidad del modelo. La complejidad del modelo se traduce en un mayor consumo en la generación de la red neuronal recurrente, debido a que requiere un mayor tiempo de creación de modelos, que es un espacio temporal en el que un ordenador tiene que ejecutar una serie de procesos realmente complejos para el entrenamiento de la red.

Con la configuración óptima de hiperparámetros, el tiempo de creación del modelo ha sido de: 46 minutos y 13 segundos. Por su parte, con los valores más grandes de los hiperparámetros estudiados este mismo tiempo ha resultado ser de: 83 minutos y 32 segundos, un valor casi el doble de grande, para unos resultados peores de los versos. Durante esos casi 40 minutos extras, asumiendo un gasto medio de 220 W por hora[33], tendría un consumo extra de 146 W que si se toma como referencia un coste de 0.145 € por kWh supondría un gasto de unos 2 céntimos de euro extras. Este número puede parecer insignificante, pero es el gasto de un único ordenador, generando un solo modelo. En caso de contar con varios ordenadores realizando varios modelos cada uno, este valor aumentará, por lo que el uso de los valores óptimos a la larga conllevará un ahorro tanto en energía como en coste.

Además del tiempo de creación, una mayor complejidad del modelo también trae consigo un aumento del peso de los archivos en los que se almacenan los modelos. Dado que el almacenamiento de información es uno de los aspectos tecnológicos que generan un mayor consumo hoy en día, este aspecto también es muy importante.

El almacenamiento en la nube de información es algo que tiene una importancia trascendental en el uso y transferencia de archivos tal y cómo los conocemos actualmente, y es un sistema que requiere del uso de servidores cada vez más grandes que por tanto

consumen una mayor cantidad de energía. Actualmente, se estima que el gasto eléctrico de estos servidores equivale al gasto eléctrico de toda Italia [34].

En este caso, si se mantienen el resto de hiperparámetros constantes, un archivo generado con un tamaño de embedding igual a 150, valor elegido como óptimo, tiene un peso de 42,5 MB, mientras que uno creado con un embedding de 600 dimensiones pesa 85,02 MB, el doble. De la misma forma, el peso del archivo de un modelo generado con 512 neuronas en cada capa es de 84,31 MB, mientras que el del modelo generado con los mismos hiperparámetros pero con un total de 256 neuronas por capa tan solo ocupa 43.6 MB de memoria, de nuevo un ahorro de prácticamente el 50% pese a que los resultados sean realmente similares. De hecho, el archivo resultante de combinar las dos mayores combinaciones de los hiperparámetros estudiada ocupa 130,4 MB, cuando el creado por la combinación óptima de los mismos tan solo 44,7 MB, lo que es casi un 65% menos de espacio.

Con este análisis por tanto, queda demostrado que una correcta elección de los hiperparámetros, además de asegurar unos mejores resultados en la red, también trae consigo la gran ventaja de ser mucho más eficiente que una red con los hiperparámetros sin optimizar, colaborando con el objetivo de la ONU al asegurar un menor consumo que deriva en un desarrollo más sostenible.

ANEXO II: CÓDIGO FUENTE

Código del programa de creación del modelo

```
import numpy as np
import keras
from keras.utils import np_utils
from keras_self_attention import SeqSelfAttention

token_list = open('SecuenciasObraCompleta.txt', 'r', encoding="windows-1252").read().replace('"', ' ').replace(',', ' ').split()
diccionario = open('Diccionario_Estrofas.txt', 'r', encoding="windows-1252").read().split('\n')

for i in range(len(token_list)):
    token_list[i] = int(token_list[i])

print("token")
print(token_list)
print(token_list[0])

def generate_sequences(token_list, step, longitud, total_words):
    X = np.zeros((0, longitud), dtype=int)+27
    y = np.zeros((0, longitud), dtype=int)+27
    control_bucle = 0
    aux_fila = np.zeros((1, longitud), dtype=int)+27
    ctrl_fila = 0
    while(control_bucle<len(token_list)):
        if(token_list[control_bucle]!=27):
            aux_fila[0, ctrl_fila]=token_list[control_bucle]
            ctrl_fila += 1
        else:
            ctrl_fila = 0
            X = np.append(X, aux_fila, axis=0)
            y = np.append(y, [np.append(aux_fila[0, 1:], 27)], axis=0)
            #padded_sequences = pad_sequences(sequences, maxlen=max_length,
padding='post')
            aux_fila[0, :]=27
            control_bucle+=1
    y_out = np.zeros((len(X), longitud, total_words), dtype=int)
    for i in range(len(X)):
        for j in range(longitud):
            y_out[i][j][y[i][j]]=1
        #y = np_utils.to_categorical(y, len(diccionario)) # Converts a class vector
(integers) to binary class matrix.

    num_seq = len(X)
```

```
    return X, y_out, num_seq

step = 1;

X, y, num_seq = generate_sequences(token_list, step, 480, len(diccionario))

## ARCHITECTURE

import tensorflow as tf
from tensorflow.keras.layers import Dense, LSTM, Input, Embedding, Dropout,
Bidirectional
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import RMSprop

n_units = 128
n_units2 = 128
n_units3 = 32
embedding_size = 150
text_in = Input(shape=(None,))

total_words = len(diccionario)
x = Embedding(total_words, embedding_size)(text_in)

x = LSTM(n_units, dropout=0.2, recurrent_dropout=0.2, return_sequences=True)(x)
x = LSTM(n_units, dropout=0.2, recurrent_dropout=0.2, return_sequences=True)(x)

text_out = Dense(total_words, activation='softmax')(x)
print("Total words", total_words)

model = Model(text_in, text_out)
opti = RMSprop(lr=0.001)
model.compile(loss='categorical_crossentropy', optimizer=opti)
epochs = 200
batch_size = 8
model.summary()
model.fit(X, y, epochs=epochs, batch_size=batch_size, shuffle=True)

model.summary()
model.save('Estrofas_ObraCompleta.h5');
```

Código del programa generador de texto

```
import numpy as np
from keras.utils import np_utils
from keras.preprocessing.sequence import pad_sequences
import tensorflow as tf
import time

model= tf.keras.models.load_model('Estrofas_ObraCompleta.h5')
#token list
token_list = open('SecuenciasObraCompleta.txt','r',encoding="windows-1255").read().replace('","').replace(',',' ').split()
diccionario = open('Diccionario_Estrofas.txt','r',encoding="windows-1252").read().split('\n')

for palabra in range(len(diccionario)):
    if(diccionario[palabra] == 'eop'):
        eop=palabra
    if(diccionario[palabra] == 'sop'):
        sop=palabra

for i in range(len(token_list)):
    token_list[i]= int(token_list[i])

#funciones
def sample_with_temp(preds, temperature):
    # helper function to sample an index from a probability array
    preds = preds[-1,:]
    preds = np.asarray(preds).astype('float64')
    preds = np.log(preds) / temperature
    exp_preds = np.exp(preds)
    preds = exp_preds / np.sum(exp_preds)
    probs = np.random.multinomial(1, preds, 1)
    return np.argmax(probs)
#def sample_wo_temp(preds)
#    return np.random.

def generate_text(seed_text, next_words, model, max_sequence_len, temp, decay):

    output_word=[seed_text]
    output_text=[seed_text]

    while output_word != eop & len(output_text)<max_sequence_len:
        #for _ in range(10):
            token_list=[output_text]

            probs = model.predict(tf.convert_to_tensor(token_list), verbose=0)[0]
            #aux = model.predict(tf.convert_to_tensor(token_list), verbose=0)
            y_class = sample_with_temp(probs, temperature = temp)
            temp = np.maximum(0.2,temp * decay)

            output_word = y_class
```

```
        if output_word == eop: #"eop"
            break

        output_text.append(output_word)
        probs2 = probs

    return output_text

num_estrofas = 5

#generacion texto
generated_token=[]
temperatura = 2.3
decay = 0.87
for i in range(num_estrofas):
    generated= generate_text(sop,20,model,350,temperatura,decay)
    #generated = generate_text(1, 20, model, 3, temperatura, decay)
    generated_token.append(generated)

for j in range(num_estrofas):
    verso = " "
    for i in generated_token[j]:
        if(diccionario[i]!= 'sop'):

            if (diccionario[i] == 'eof'):
                diccionario[i]="\n"
            if (diccionario[i] == 'inicio'):
                diccionario[i]=" "
            #verso.append(diccionario[i])
            verso += diccionario[i] + " "
        else :
            verso += "\n"

print(verso)
```