UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

OFFICIAL MASTER'S DEGREE IN THE ELECTRIC POWER INDUSTRY

Master’s Thesis

Taking advantage of Artificial Neural Networks for the Unit Commitment problem resolution

Author:	**Manuel Florez Montes (201503672@alu.comillas.edu)**
Supervisor:	**Pedro de Otaola Arca (pdeotaola@iit.comillas.edu)**
Co-Supervisor:	**David Domínguez Barbero (ddominguez@comillas.edu)**

Madrid, July 2021

Master's Thesis Presentation Authorization

THE STUDENT:

Manuel Florez Montes

THE SUPERVISOR

Pedro de Otaola Arca

Signed by DE OTAOLA ARCA PEDRO - 39486410X on the date 14/07/2021 with a certificate issued by AC FNMT Usuarios

Signed: …………………… Date: 14/07/2021

THE CO-SUPERVISOR

David Domínguez Barbero

Firmado digitalmente por DOMINGUEZ BARBERO DAVID - 03942565C el día 14/07/2021 con un certificado emitido por AC FNMT Usuarios

Signed: …………………… Date: 14/07/2021

Authorization of the Master's Thesis Coordinator

Dr. Luis Olmos Camacho

OLMOS CAMACHO LUIS - 33531178F

Firmado digitalmente por OLMOS CAMACHO LUIS - 33531178F
Fecha: 2021.07.15 19:27:05 +02'00'

Signed: …………………… Date: ……/ ……/ ……

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

OFFICIAL MASTER'S DEGREE IN THE ELECTRIC POWER INDUSTRY

Master's Thesis

Taking advantage of Artificial Neural Networks for the Unit Commitment problem resolution

Author:	**Manuel Florez Montes (201503672@alu.comillas.edu)**
Supervisor:	**Pedro de Otaola Arca (pdeotaola@iit.comillas.edu)**
Co-Supervisor:	**David Domínguez Barbero (ddominguez@comillas.edu)**

Madrid, July 2021

AUTHORIZATION FOR DIGITALIZATION, STORAGE AND DISSEMINATION IN THE NETWORK OF END-OF-DEGREE PROJECTS, MASTER PROJECTS, DISSERTATIONS OR BACHILLERATO REPORTS

1. Declaration of authorship and accreditation thereof.

The author Mr. /Ms. Manuel Florez Montes **HEREBY DECLARES** that he/she owns the intellectual property rights regarding the piece of work: Taking advantage of Artificial Neural Networks for the Unit Commitment problem resolution, that this is an original piece of work, and that he/she holds the status of author, in the sense granted by the Intellectual Property Law.

2. Subject matter and purpose of this assignment.

With the aim of disseminating the aforementioned piece of work as widely as possible using the University's Institutional Repository the author hereby **GRANTS** Comillas Pontifical University, on a royalty-free and non-exclusive basis, for the maximum legal term and with universal scope, the digitization, archiving, reproduction, distribution and public communication rights, including the right to make it electronically available, as described in the Intellectual Property Law. Transformation rights are assigned solely for the purposes described in a) of the following section.

3. Transfer and access terms

Without prejudice to the ownership of the work, which remains with its author, the transfer of rights covered by this license enables:

a) Transform it in order to adapt it to any technology suitable for sharing it online, as well as including metadata to register the piece of work and include "watermarks" or any other security or protection system.

b) Reproduce it in any digital medium in order to be included on an electronic database, including the right to reproduce and store the work on servers for the purposes of guaranteeing its security, maintaining it and preserving its format.

c) Communicate it, by default, by means of an institutional open archive, which has open and cost-free online access.

d) Any other way of access (restricted, embargoed, closed) shall be explicitly requested and requires that good cause be demonstrated.

e) Assign these pieces of work a Creative Commons license by default.

f) Assign these pieces of work a HANDLE (*persistent* URL). by default.

4. Copyright.

The author, as the owner of a piece of work, has the right to:

a) Have his/her name clearly identified by the University as the author

b) Communicate and publish the work in the version assigned and in other subsequent versions using any medium.

c) Request that the work be withdrawn from the repository for just cause.

d) Receive reliable communication of any claims third parties may make in relation to the work and, in particular, any claims relating to its intellectual property rights.

5. Duties of the author.

The author agrees to:

a) Guarantee that the commitment undertaken by means of this official document does not infringe any third party rights, regardless of whether they relate to industrial or intellectual property or any other type.

b) Guarantee that the content of the work does not infringe any third party honor, privacy or image rights.

c) Take responsibility for all claims and liability, including compensation for any damages, which may be brought against the University by third parties who believe that their rights and interests have been infringed by the assignment.

d) Take responsibility in the event that the institutions are found guilty of a rights infringement regarding the work subject to assignment.

6. Institutional Repository purposes and functioning.

The work shall be made available to the users so that they may use it in a fair and respectful way with regards to the copyright, according to the allowances given in the relevant legislation, and for study or research purposes, or any other legal use. With this aim in mind, the University undertakes the following duties and reserves the following powers:

a) The University shall inform the archive users of the permitted uses; however, it shall not guarantee or take any responsibility for any other subsequent ways the work may be used by users, which are non-compliant with the legislation in force. Any subsequent use, beyond private copying, shall require the source to be cited and authorship to be recognized, as well as the guarantee not to use it to gain commercial profit or carry out any derivative works.

b) The University shall not review the content of the works, which shall at all times fall under the exclusive responsibility of the author and it shall not be obligated to take part in lawsuits on behalf of the author in the event of any infringement of intellectual property rights deriving from storing and archiving the works. The author hereby waives any claim against the University due to any way the users may use the works that is not in keeping with the legislation in force.

c) The University shall adopt the necessary measures to safeguard the work in the future.

d) The University reserves the right to withdraw the work, after notifying the author, in sufficiently justified cases, or in the event of third party claims.

Madrid, on 14 of July, 2021

HEREBY ACCEPTS

Signed by Manuel Florez Montes mfm

Reasons for requesting the restricted, closed or embargoed access to the work in the Institution's Repository

1. TAKING ADVANTAGE OF ARTIFICIAL NEURAL NETWORKS FOR THE UNIT COMMITMENT PROBLEM RESOLUTION

Author: Florez Montes, Manuel.
Supervisor: de Otaola Arca, Pedro.
Co-Supervisor: Domínguez Barbero, David
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This thesis implements and solves a unit commitment model that can be run for many consecutive weeks by changing input data. Once the solution set is created, Machine Learning is applied, specifically an Artificial Neural Network (ANN), which learns from that dataset to improve the optimization process for future input data.

Keywords: Unit Commitment (UC), Artificial Neural Network (ANN), Machine Learning (ML), Multi-Layer Perceptron (MLP), Optimization, Optimizer, Execution Time, Constraints, Variables, Parameters, Optimal Solution.

1. Introduction

The unit commitment (UC) problem is the process of optimizing the most cost-effective combination of generation units and their output power within a power system to meet forecasted demand and other requirements.

To solve this problem, multiple optimization algorithms have been applied over the past years. However, these approaches suffer from different problems related to the algorithms' characteristics.

Therefore, machine learning (ML) algorithms could be valuable at helping to understand the relationship between the UC problem information and the operation schedule, which could help to address the UC problems in a more precise and accurate approach.

2. Scope of the project

This thesis tries to create an ANN that can boost the performance of the traditional resolution method in the UC problem. This procedure is aimed at helping to analyze certain aspects without needing to run the UC model or even to give extra information to guide the UC model optimization process.

Therefore, this ANN could be used to provide approximations to the unit commitment problem that could help other systems within the electric power industry, or even could be used to improve the optimization process of the UC problem.

3. Methodology

The methodology of this thesis can be stated into a cyclical procedure divided into three iterative steps.

First, this Master Thesis focusses in developing a UC model and providing a solution for several weeks of input data.

The next step of the project is creating an ANN in order to learn all the equations of the model and obtaining behavior patterns from the solution.

Lastly, the ANN is used to enhance the UC optimization algorithm by providing a starting point for the optimization process.

This methodology is resumed in Illustration 1.

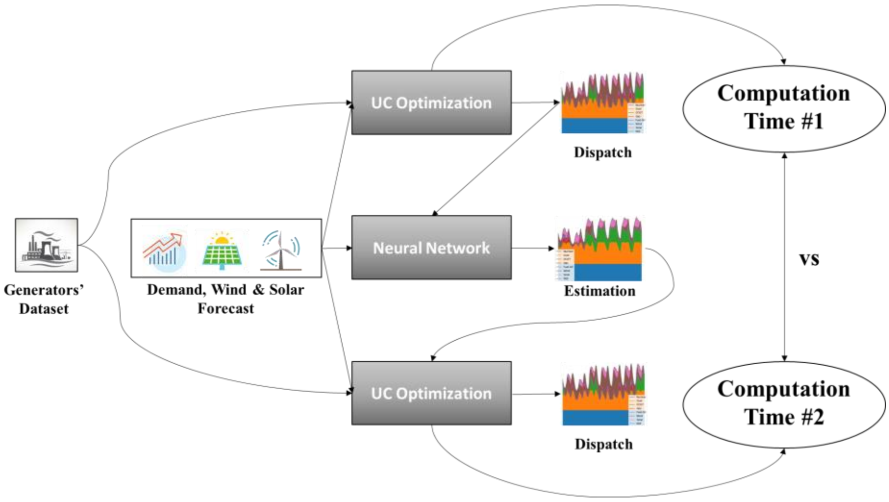

Illustration 1. Methodology

4. Results

As stated in Table A, it can be appreciated that when applying the exposed methodology in a large-scale UC problem, the execution time required to obtain an optimal solution is reduced in 41.1% compared to the traditional resolution procedure.

Table A. Results for Large Scale Case Study

Case		Case A	Case B
Optcr Requested		1%	1%
Max Time Allowed (mins)		20	20
Optcr Achieved		0,917%	0,854%
Time (mins)	Feasibilization	2,758	-
	Optimization	8,895	19,781
	Total	11,653	19,781
Warm-start Objective Function		2,481E+09	-
Estimated Objective Function		1,873E+08	1,876E+08
Achieved Objective Function		1,890E+08	1,892E+08

5. Conclusions

Regarding UC, the model solves the problem presented successfully. Results obtained are rational and the model aims to minimize operating costs by always trying to reduce deviations with respect to demand.

Regarding NN, the model manages to learn patterns of different generators. This is that those generators that are a base technology are identified as such, and those that present residual behavior are also recognized successfully.

Lastly, the large-scale case study has verified that there is a considerable improvement in the execution time of the UC problem resolution when applying the proposed methodology.

In general terms, this thesis fulfills the objectives that were proposed.

Table of Contents

List of Figures

List of Tables

Chapter 1. INTRODUCTION

1.1 BACKGROUND

The formulation of generation units' allocation planning for power systems suffers from several complex problems such as the number, type or size of generation facilities and demand fluctuations.

These complexities imply that operators have to face several decision-making processes, such as scheduling generation units in a system at any given time, since it is not optimal to run all the units required to satisfy peak load during base load periods.

The unit commitment (UC) problem is the process of optimizing the most cost-effective combination of generation units and their output power within a power system to meet forecasted demand and other requirements or constraints such as spinning reserve constraints, generation constraints or transmission constraints.

Thus, in analytical terms, this problem is defined as the determination of the start-up and shut-down schedules and output power of generation units so that the total production cost is minimized whereas satisfying various system, transmission, and generation constraints.

This optimization problem is fundamental within the electric industry, since low-cost solutions of these problems will directly be related into low production costs for generation utilities, and any improvements in the UC schedule may result in significant reduction in the total production costs. However, this is a very complex problem since it is a non-linear, mixed-integer optimization problem, and it becomes more complex as the size of the problem increases.

To solve this problem, multiple optimization algorithms have been applied over the past years. However, these approaches suffer from different problems related to the algorithms' characteristics.

Indeed, traditional decision-making processes for power systems such as UC problem primarily rely on a physical model and numerical calculations. However, there is plenty of historical data for the power system that could provide useful information and knowledge of the power grid.

Therefore, machine learning (ML) algorithms could be valuable at helping to understand the relationship between the UC problem information and the operation schedule, which could help to address the UC problems in a more precise and accurate approach.

During the last two decades, Artificial Intelligence methods, and hybrid models have also been proposed as approaches to solve the UC problem. ML techniques have been successfully used for more than a decade ago to solve UC problems and researchers have specially investigated the effectiveness of Artificial Neural Networks (ANN) in solving this problem.

ML techniques can be revealed as a valuable alternative or complement to the model-based existing methods. As a result, it is expected that the effective combination of ML approaches and physical model approaches could lead in more efficient solutions for the UC problem that could improve system reliability and economic operation.

1.2 MOTIVATION

This thesis tries to create an ANN that can boost the performance of the traditional resolution method in the UC problem. This procedure is aimed at helping to analyze certain aspects without needing to run the UC model or even to give extra information to guide the UC model optimization process.

In other words, the benefit of the technique that has been used is that for each upcoming week, with the new current forecasts of demand, available solar power, and available wind power, it is possible to avoid running a full UC model, which would imply long waiting times to have answers about the possible schedule of the system.

Instead, this ANN could be used to provide initial approximations, which could quickly provide answers to other types of systems. For example, the department that manages gas purchases for CCGTs operation requires approximations of how much gas will be used, but they do not need a perfectly detailed schedule.

In addition, these ANN approximations could also be used as input to a full UC model, to be used as an initial solution in the optimization process, which would mean having the detailed solution in less time than if the UC problem was run directly.

1.3 OBJECTIVES

Therefore, some concrete milestones can be determined in order to assess the thesis's achievement as satisfactory. These objectives are as follows:

- Develop a UC optimization model.
- Automatize the execution and storage of solution of the UC model for several sets of input data.
- Develop a Neural Network to learn from all the consecutive executions of the optimization problem. A Multi-Layer Perceptron (MLP) architecture will be developed.
- Use the Neural Network to estimate different output variables of the UC problem.

1.4 DOCUMENT STRUCTURE

All the research and development of this thesis is reflected in this single document, which is divided into eight (8) main chapters and one (1) annex with complementary information.

Chapter 1. Introduction, consists of an initial section to orient the reader to the motivations and reasons for undertaking this project as well as the challenges faced and the objectives to be achieved.

Chapter 2. State of the Art, presents a literature review of different areas that are relevant to this research.

Chapter 3. Problem Setting, presents the problem to be solved, its possible difficulties, and the methodology carried out to solve the problem.

Chapter 4. Unit Commitment Model, details and explains the UC optimization model used in this thesis.

Chapter 5. Neural Network, details and explains the neural network model used in this thesis.

Chapter 7. Results, provides a critical analysis of the results obtained.

Chapter 8. Conclusions & Future Developments, summarizes the difficulties, main discussions and conclusions that can be drawn from this thesis. In addition, it also points the direction for possible future research.

Chapter 9. References, presents the references and bibliography used to develop this thesis.

ANNEX I. Python Scripts, details the programming scripts and algorithms used to develop this thesis.

Chapter 2. STATE OF THE ART

2.1 UNIT COMMITMENT

UC problem tries to determine the dispatch of generation units within an electric power system, subject to technical and system constraints. The decision-making process selects the units that will be on or off, the output energy generated by each unit and their reserve margins according to a certain objective function.

2.1.1 CLASSIFICATION

There is not a unique UC problem for all the possible scenarios in the power industry, so the UC can be classified within three different categories. With respect to security, UC can be divided into traditional UC, security-constrained UC (SCUC) or price-based UC (PBUC). Next, regarding market operation, UC can be divided into a vertically integrated environment or a deregulated environment. Finally, regarding the approach taken for future events, UC can be divided into deterministic and stochastic.

The traditional UC problem has been formulated as a non-combinatorial, non-convex, large-scale, non-linear, and constrained optimization problem [9]. Non-linear problems cannot be solved easily because they include combinations of binary and non-linear variables. Therefore, the UC problem is represented as a Mixed Integer Linear Programming (MILP) problem.

Within this framework, the objective function is stated as the minimization of the operational costs of the generation units. Thence, the problem is modeled using binary variables for the commitment decisions of the units (ON/OFF) and continuous variables for representing its output power.

SCUC is fundamental for the reliable and economic functioning of power systems, especially in large-scale systems. It refers to the economic dispatch of generation units for meeting the demand while timing and operational limits in contingency power system are satisfied [9]. Mathematically, it is proposed as a large-scale Mixed Integer Programming (MIP) with many variables, binary, continuous and discrete and several restrictions [10].

In PBUC, hourly demand balancing restriction is removed from the set of constrains, security is considered an ancillary service and the objective is a profit-maximization function. Therefore, the signal that determines the commitment decisions of the generation units is the market price for different products such as energy, ancillary services, and fuel. In this problem, suppliers are responsible for the way they sell energy in order to satisfy demand and reserve markets [11].

Moreover, since 1980s, power systems have turned from a vertical integrated structure to a deregulated environment. In an integrated environment, customers of generation companies (GENCOs) are already defined. However, in a deregulated environment, GENCOs construct their offers based on their available information [9]. Each offer consists of a cost function and several parameters that define the operational constraints of the generation units, which allows to determine the marginal cost for each period [12]. In addition, in the deregulated environment, the objective also changes from cost minimization to profit maximization.

Next, the stochastic approach is used to give solution to those optimization problems under considerable uncertainty, represented by a set of different scenarios. This technique is based on scenario tree in which uncertainty of each node is expected to be known by probability distributions [9]. Therefore, several scenarios are needed to achieve an acceptable solution, but as the number of scenarios increases, the computational effort increases exponentially, so it limits its application to large-scale systems. For this reason, scenario reduction techniques are commonly used in order to reduce the scale of the problem by withdrawing low-probability scenarios or very similar ones [13].

Finally, deterministic approach can be considered as a special case of its respective stochastic formulation, in which it is considered only a single scenario with certain forecasting values and system parameters.

2.1.2 TRADITIONAL RESOLUTION METHODS

Several methods and techniques have been proposed over the past years to get an optimal generation schedule in any scenario. Some of these are Priority Ordering Methods, Dynamic Programming (DP), MIP and MILP, Lagrangian Relaxation (LR), and Branch and Bound (B&B), among other approaches.

First, Priority listing methods re-arrange generation units in a start-up heurist list by a combination of operation and transition costs. This order is then used to commit the units such that system demand is met [14].

Then, DP technique searches the units status for an optimal solution. Each time period of the horizon is known as the stage of the DP problem, which typically represents one hour of operation. The combinations of units within a time period are known as the states of the DP problem. This technique finds the optimal commitment by starting at the initial stage and accumulating total costs, and then returning back from the combination of least accumulated cost at the last stage to the initial stage [14].

Furthermore, MIP and MILP approaches solve the UC problem by systematically reducing the solution search space by withdrawing infeasible subsets. The general concept is based on the resolution of a linear programming problem and checking for an integer solution. So if the solution is not integer, linear problems and subproblems are continuously solved.

Next, LR technique decompose the UC problem into a major problem and several subproblems that are iteratively solved until a quasi-optimal solution is obtained. Each subproblem determines the commitment of a single unit and then is linked to the master problem by LaGrange multipliers in order to build a Dual problem [14].

The solution is then obtained by an iterative process in which LaGrange multipliers are computed on the master problem, then passed to the subproblems, and then, the solution of the subproblems is returned to the master problem obtaining updated multipliers. This process is repeated until the solution converges.

Finally, B&B determines a lower limit to the optimal solution and finds a near optimal feasible schedule. That lower bound can be determined from a dual problem that uses Langrangian relaxation [14]. Also, that dual problem is used to get information for producing priority lists, which are useful for finding feasible solutions and helping in determining an upper bound. Therefore it is built a branch and bound tree for searching the best solution, but only few nodes are examined if an upper bound is found.

However, these approaches suffer from different problems related to the algorithms themselves such as high sensitivity to the choice of architecture, manual parameter turning, different cost functions [1], or exponential computational time growth as the size of the problem increases.

2.1.3 CHALLENGES & INNOVATION

Recently, the increase of participation of renewable energy sources (RES) and more price-responsive demand, have transformed the UC into an even more difficult problem, due to the uncertainty and high variability of RES. It has become necessary to have an effective methodology that produces robust UC decisions and ensures system reliability against the increasing uncertainty in real time [14].

In addition, with the advent of restructuring in power industries, power systems have undergone greater changes in structure and operation. In fact, utilities under deregulated environments have to design new strategies and tools for their efficient operation and management in this stochastic scenario, which is inevitable due to the higher generation of RES.

Therefore, models must be able to face the current challenges of the power system and provide efficient solutions. For instance, Machine Learning (ML) algorithms have found to be successful in solving decision making problems under uncertainty. In particular, Artificial Neural Networks (ANN) can be very useful in solving or helping to improve the resolution of UC problem.

Estimates of ANN are based on load curves and their corresponding UC schedules. The pattern of the load curve is compared to the information in the database to select the most economical UC schedule. In fact, if an ANN estimation schedule is not feasible for the entire UC period, it can be used as an initial starting point for a near-optimal solution [14].

Moreover, it has been proposed another innovative approach to the resolution to the UC problem in [15]. In this model, the continuous variables (output power) are used as the primary decision variables, and several functions are defined to characterize the commitment, start-up, and shut-down logic constraints.

One of the main advantages of this approach is that it can be solved by continuous optimization tools such as non-linear solvers, and no integer programming tools are needed.

Nevertheless, the focus of that model is on a new format of the UC model, rather than investigating the global optimization algorithms. Currently, the main attention is being paid to solve large scale UC problems with this new method, and further progress will be reported in the near future.

2.2 ARTIFICIAL NEURAL NETWORKS

Artificial Intelligence (AI) models are characterized by their flexibility and their capability of integrating several methodologies and algorithms that try to imitate the behaviors of biological systems somehow [16]. Within this field, Artificial Neural Networks (ANN) try to emulate human brain functions.

An ANN is an algorithm based on the imitation of the computation of a human brain, which essentially transmits information through neurons. More specifically, an ANN is a large number of highly connected but simple elements (neurons) that can receive, process, and transmit information, each element mimicking a human neuron [2].

Therefore, a neural network is created by interconnecting some consecutive layers with certain neurons allocated in a particular manner. Then, each neuron in each layer receives some input information, processes it, and transmits it to another layer, which will receive that information as an input.

The way that each neuron processes the information depends on the underlying network parameters, which are continuously being adjusted to appropriately respond to the problem to be solved, until the problem reaches a certain accuracy in solving the problem or some termination requirements are met.

ANN models are data-driven, self-adaptable, and non-linear methods that do not require specific assumptions about the underlying model. Therefore, there are many potential benefits presented by ANNs: i) input-output mapping, this is that network learns based on examples, ii) adaptability, this is that network can adapt its features even in real time, iii) response capacity, this is pattern classification and decision making reliability, iv) integrated large scale, which means that is potentially fast for certain tasks and capturing certain patterns, and v) uniformity, since the methodology is standardized independently of its application [16].

One of the well-known and most common architecture is the Multi-Layer Perceptron (MLP), which is characterized for being a universal approximator.

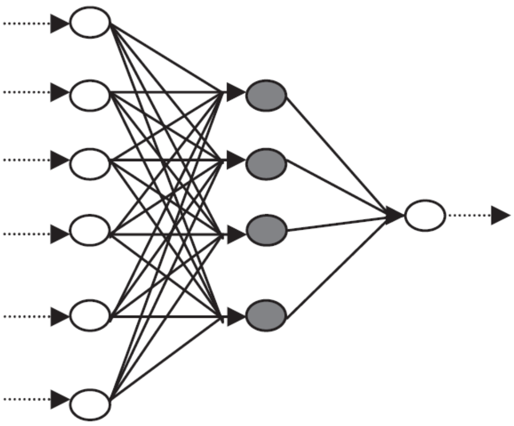

Figure 2-1: MLP architecture [16]

MLP has an input layer (with any processing), an output layer, and one or more hidden layers. All layers are formed by a certain number of neurons and are fully connected. This is that each neuron of each layer is connected to all the neurons of the following layer.

In the case of the hidden layer, its inputs come from the neurons of the previous layer and its outputs are sent to the following layer. Moreover, input and output layers represent the information flow that is implemented in the learning algorithm [17].

The learning algorithm for MLP is further explained in [16] and [17]. Finally, once the training process is terminated, network weights are established, and the network can be used to make estimations for new entries.

2.3 UC MODELS WITH ANN

As in many other engineering disciplines, artificial neural networks (ANNs) are becoming useful in several aspects and problems for the electric power industry. For instance, the application of ANNs in UC problems has become an active research field in recent years. There are several different approaches to this technique.

One of these approaches was presented by Sasaki et al. who proposed the application of Hopfield NN to the UC problem considering minimum up and down periods, reserves, and demand constraints [19]. The main problem of such approach is that for solving a medium size scenario with 30 generators, the size of the NN becomes considerably large. Therefore, the average computational time was approximately 2.5 times slower than the Langrangian method.

Furthermore, Gee et al. considered another methodology to improve the NN performance [20]. Authors achieved a more precise, and fast, so therefore more efficient, approach than the conventional Hopfield NN approach.

This new technique has been used by Yalcinoz and Short [21] to solve large-scale economic dispatch problems and has achieved efficient and precise solutions to different systems with different sizes varying from 3 to 240 generation units.

In [18] it is proposed to use the improved methodology of Hopfield NN to solve the UC problem. For developing the model, it has been considered transmission capacity, transmission losses, start-up, and shut-down costs, minimum up and down periods and demand constraints.

This methodology has been tested in a 3-unit system and in a 10-unit system, and results showed that the improved Hopfield NN methodology can be applied to UC problems and could achieve a faster solution than conventional resolution methods.

Lastly, another approach is proposed in [22]. This presents a hybrid algorithm between an ANN and a simulated annealing approach to solve the UC problem. The ANN is used to determine the discrete variables such as commitment, start-up, or shut-down variables whereas the annealing approach is used to obtain continuous variables such as output power or production costs.

This method uses a MLP trained by backpropagation as the ANN. In order to train the network, a set of load profiles are settled as inputs and the corresponding unit-commitment (satisfying the UC constraints) schedules as outputs.

The proposed approach has been tested in a sample 10-unit power system, and results showed that this methodology could solve the UC problem with an optimal generation schedule and in a reduced computational time.

Chapter 3. PROBLEM SETTING

This Master Thesis can be stated as a cyclical process that can be separated into three (3) blocks in terms in methodology and problem setting.

First, this Master Thesis will be focused in developing a Unit Commitment model that can be ran for several consecutive weeks changing input data such as demand, wind power available or solar power available.

Once this model is developed, the next step of the project is to apply Machine Learning algorithms in order to learn all the equations of the model so that this algorithm enables to estimate a preliminary solution for the upcoming weeks.

Lastly, once trained, the NN is used to enhance the optimization, and it will be assessed the feedback of the NN results on the improvement of the UC optimization algorithm.

As explained in section 1.2 - Motivation, the aim of this thesis is to reduce the overall optimization time for a large amount of input data. But contrary to what is proposed in that section, as it was not available a large enough amount of input data to be split up, all available data was used both to solve the optimization model and to train the network.

So, the way to evaluate the improvement of the optimization is to compare the differences in the time (or alternatively, the number of iterations) taken to solve the optimization model directly from the input data and from the estimated neural network solution.

Therefore, the methodology and problem setting can be broadly summarized in the diagram shown in Figure 3-1 below.

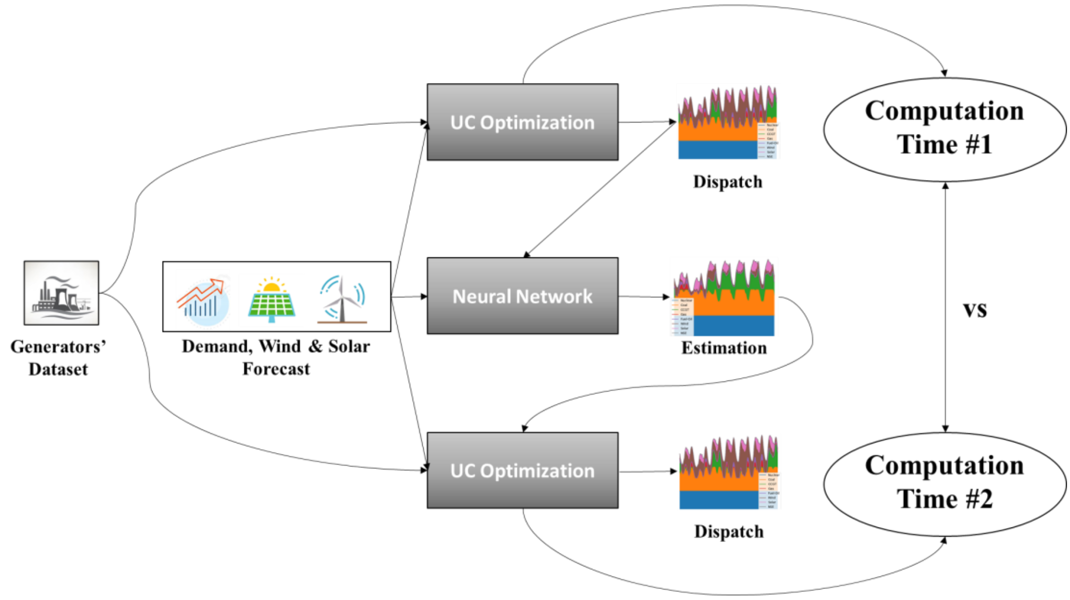

Figure 3-1: Thesis Problem Setting

3.1 GENERAL SETTINGS

All the research and main tasks of this master thesis have been performed through an online repository (digital library accessible via the internet). This is aimed at having a periodical control of the programming code and scripts on which the history of changes and updates can easily be monitored, and also to provide open access to the real time code to the master thesis supervisors.

All the models and algorithms in this master thesis are developed using Python programming language. Unit commitment modelling is developed with Pyomo library and the neural network is implemented with Tensorflow and Keras libraries.

All programming has been done through the Visual Studio Code IDE interface, which enables an online connection with a GitHub repository in order to monitor and have control to the improvements of the models through all the duration of the project.

Regarding the unit commitment model, it was not developed from scratch, but has been based on an existing optimization model [8]. This reference base model consisted of an experimental and very simple optimization algorithm that tried to maximize the profit of fictitious generating sets, assuming a fictitious market price and fictitious generation costs.

This model then served as a starting point for implementing the modifications and improvements needed to develop the UC model.

Therefore, according to the above-mentioned features, next they are presented some of the main tasks carried out throughout the completion of this master's thesis.

Repository completion.

- Create an online and interconnected GitHub repository that enables to work on the project and state and review changes.
- Connect that online repository to the local working path where the project will be developed.
- Ensure proper operation of the system.
- Learn and familiarize with the system in order to acquire agility with it.

UC model in Python.

- Learn and familiarize with the base model.
- Find appropriate information on costs, limits and characteristics of different thermal units.
- Modify generator input data.
- Modify input temporal and generation sets.
- Model Constraints and Objective Function. A complete re-adaptation of the model was carried out. All those equations that were unnecessary were modified, and new constraints were added.
- Reorganize the model and add new items useful for the case study: new output variables, demand-generation plots, etc.
- Ensure the functioning of the model and check the correct and quick resolution of the optimization.

Automatic run algorithm.

- Get input data from OMIE. (Demand and available wind and solar power) for several weeks.
- Adequate input data to the model.
- Develop a script that reads the input data for all weeks and gives a solution to all of them automatically.

Machine learning algorithm.

- Define the input features.
- Study the convenience and usefulness of the predicted features.
- Build the dataset: training and test sets.
- Develop a Neural Network model.
- Train the model and observe the learning behavior.
- Explore different hyper-parameters.
- Study Neural Network – UC model interactions.

Finally, it should be mentioned that it has been used 40 consecutive weeks of real data. In addition demand, solar and wind energy forecasted data used in this thesis correspond to the whole Iberian Peninsula, so it consists of really large demand values.

Therefore, in order to satisfy this demand, a large number of generators are required. Initially, an attempt was made to develop the model with these conditions, but the idea was quickly changed, as the computational times were too long to be able to develop the whole investigation.

It was therefore decided to reduce the number of generators to make the model more accessible to work with. To achieve this, two different alternatives were carried out, but the initial alternative was finally withdrawn due to disappointing results.

3.2 PRELIMINARY APPROACH

The preliminary approach consisted of grouping the generator sets by technology (Nuclear, Coal, CCGT…) so that instead of having a large number of real generation units, their costs and technical specifications and characteristics would be added up in order to have only a few sets corresponding to the different technologies.

Hence, it would not be necessary to reduce demand values, as actually the installed capacity (in terms of maximum power) would be the same as in the previous case, but with a lower number of generation units, which reduced the computation time considerably.

However, once tested, this alternative was found to have a fundamental drawback. This was that by grouping units by technology and creating fictitious and very large generation units, some of the intrinsic and fundamental conditions of the UC model became meaningless.

In terms of optimization, this meant that the model lost complexity significantly and several constraints became non-binding. Consequently, obtained results were easily predictable, so when estimations were made with the neural network and checked its validity, the errors obtained were not large enough to consider this situation as realistic and valid.

Therefore, it was decided to implement another and more realistic approach.

3.3 FINAL APPROACH

This approach consisted of an intermediate point between the initial approach and the approach explained in the previous section. This final approach consists of reducing demand values and maintaining a considerable number of generating units but with realistic characteristics.

This reduction in demand can be justified by arguing that the system to be treated is an island or archipelago rather than the previous case. Moreover, this avoids the loss of complexity of the model and all constraints remain valid.

This approach has finally been used for the development of the master thesis and its development and results obtained will be discussed in the upcoming sections of the document.

Chapter 4. UNIT COMMITMENT MODEL

This section shows the different features and settings of the UC model used as well as the justification for their use. In the same way, the obtained results and solutions are also analyzed in order to justify the correct performance of the optimization model.

4.1 GENERAL FEATURES

In general terms, this model is a weekly unit commitment (UC) model with periods of 1 hour (168 periods of 1 hour per week).

Moreover, this model is thermal and RES based, but there is no hydro power nor water storage system included in the model. In addition, the model works through cost and power terms and not fuel terms. This is that start-up, operation, and shutdown are not measured in terms of fuel consumption and then transferred into costs, but directly measured in terms of cost.

The objective of this problem is the minimization of total operating costs while meeting demand. Therefore, the objective function encompasses all costs, including start-up costs, shut-down costs, and variable costs, and also penalizes excess of energy and non-served energy. Renewables, on the other hand, are assumed to be dispatched without operating costs.

In addition, the main constraints that have been applied in the model are: i) supply-demand balance, ii) maximum and minimum output power of generation units, iii) logic coherence startup-commitment-shutdown, iv) start-up and shut-down at minimum stable load, v) spinning reserves (upwards and downwards), vi) ramp rates (upwards and downwards), and vii) minimum up and down time.

The entire formulation used in the problem is shown in section 4.3. Variables, Parameters & Equations.

4.2 INPUT DATA

Within the input data, two different types of datasets are distinguished: i) wind and solar forecasts and demand, and ii) number of generators employed as well as generators' data.

However, both datasets are related since the demand value determines to some extent the number of generators to be used in the model. This number of generators must be large enough to meet the demand, but not excessively large since then the number of variables in the model (and thus the computation time) grows exponentially.

The demand data used, as well as wind and solar forecasts, were obtained directly from data provided by the Spanish market operator (OMIE).

The data collected corresponds to 40 weeks of 2018 (January-October). Each week is divided into 7 days and each day into 24 hours.

To facilitate the resolution of the model, this organization of the input data to the model has been re-structured by grouping each day of the week consecutively so that the input to the model results in 40 weeks with 168 hours.

Figure 4-1 below shows an example of demand, and Figure 4-2 and Figure 4-3 shows an example of wind and solar forecasts for one week.

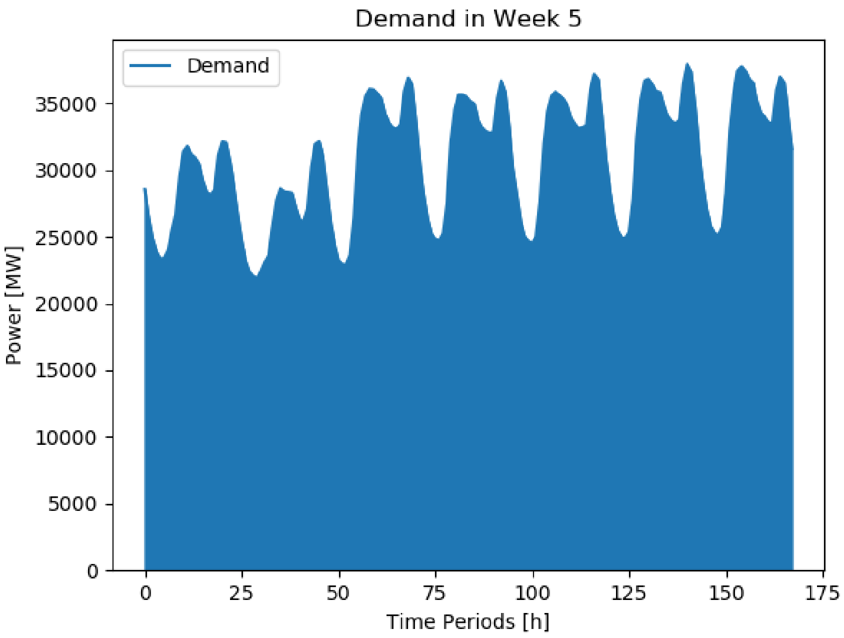

Figure 4-1: Demand in Week 5

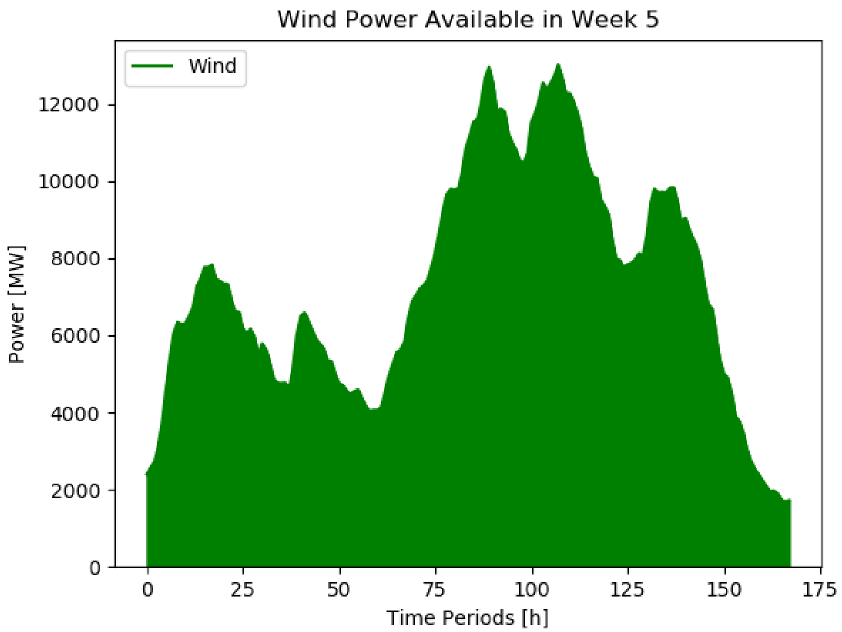

Figure 4-2: Wind Power Available in Week 5

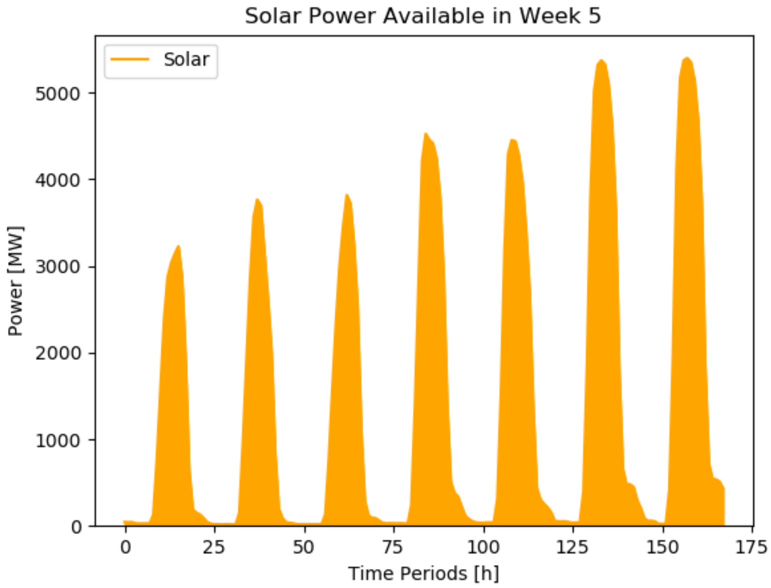

Figure 4-3: Solar Power Available in Week 5

On the other hand, these demand values were really large, so a very high number of generators (>50) was needed to satisfy the demand.

So, in order not to excessively increase the computation time while maintaining the complexity of the model, it has been decided to reduce the value of the demand data (and the wind and solar predictions) to 20% of the original value, in order to establish a more manageable number of generators.

Thus, 16 generator units of different types have been established for this model. Generators' data as well as the distribution of generators has been determined based on data used in other optimization models [23], [24]. Generators' data is presented in Table 1 and

Table 2.

Table 1. Generators' Costs

Units	Type	€/h	€/MWh	€/arr	€
gts		p_GTScaco	p_GTScvar	p_GTScarr	p_GTScpar
g01	Nuclear	10400,00	10,00	70000	7000,00
g02	Nuclear	10400,00	10,00	70000	7000,00
g03	Coal	990,54	18,71	12000	1200,00
g04	Coal	959,67	18,13	12000	1200,00
g05	Coal	930,51	17,58	12000	1200,00
g06	Coal	1204,20	17,06	15000	1500,00
g07	Coal	1186,8	16,81	15000	1500
g08	Coal	1169,88	16,57	15000	1500
g09	CCGT	1151,5	29,43	6000	600
g10	CCGT	1105,49	28,25	6000	600
g11	CCGT	1083,78	27,7	6000	600
g12	CCGT	1063,04	27,17	6000	600
g13	CCGT	1042,96	26,65	6000	600
g14	CCGT	1023,68	26,16	6000	600
g15	Gas	1266,71	34,55	2800	280
g16	Gas	1169,86	32,9	2800	280

Table 2. Generators' Technical Data

Units	Type	MW	MW	MW/h	MW/h	h	h
gts		p_GTSpmn	p_GTSpmx	p_GTSrs	p_GTSrb	p_GTStd	p_GTStu
g01	Nuclear	1000	1000	1000	1000	0	0
g02	Nuclear	1000	1000	1000	1000	0	0
g03	Coal	150	300	30	30	6	6
g04	Coal	150	300	30	30	6	6
g05	Coal	150	300	30	30	6	6
g06	Coal	200	400	40	40	6	6
g07	Coal	200	400	40	40	6	6
g08	Coal	200	400	40	40	6	6
g09	CCGT	180	450	60	60	3	3
g10	CCGT	180	450	60	60	3	3
g11	CCGT	180	450	60	60	3	3
g12	CCGT	180	450	60	60	3	3
g13	CCGT	180	450	60	60	3	3
g14	CCGT	180	450	60	60	3	3
g15	Gas	132	330	60	60	3	3
g16	Gas	128	320	60	60	3	3

4.3 VARIABLES, PARAMETERS & EQUATIONS

The objective function used in this optimization model is shown below in (4.1).

$$\min \sum_t \left[P_{nse} \cdot p_{nse_t} + P_{nse} \cdot p_{ex_t} + \sum_g ct_{g,t} \right] \tag{4.1}$$

where:

t is the set of time periods.

g is the set of generation units.

P_{nse} is the cost of non-served energy and excess of energy. Its value was set large enough (1000 €/MWh) so that the model tries to minimize energy excesses and defects.

p_{nse_t} is the non-served energy in period t.

p_{ex_t} is the excess of energy in period t.

$ct_{g,t}$ is the thermal cost of unit g in period t. This variable is calculated as showed in (4.2).

$$ct_{g,t} = c_{arr_g} \cdot y_{g,t} + c_{par_g} \cdot z_{g,t} + c_{aco_g} \cdot v_{g,t} + c_{var_g} \cdot t_{g,t} \tag{4.2}$$

where:

c_{arr_g} is the start-up cost of unit g.

c_{par_g} is the shut-down cost of unit g.

c_{aco_g} is the no-load cost of unit g.

c_{var_g} is the slope of the linear cost function of unit g.

$y_{g,t}$ is the start-up decision of unit g at the beginning of period t.

$z_{g,t}$ is the shut-down decision of unit g at the beginning of period t.

$v_{g,t}$ is the commitment of unit g state in period t.

$t_{g,t}$ is the power generated by unit g in period t.

The constraints used in this optimization model are shown below (i-vii).

i) The supply-demand balance constraint is showed in (4.3).

$$\sum_{g} t_{g,t} + p_{w_t} + p_{s_t} + p_{nse_t} - p_{ex_t} = d_t \tag{4.3}$$

where:

d_t is the demand in period t.

p_{w_t} is the wind power generation in period t.

p_{s_t} is the solar power generation in period t.

ii) The maximum and minimum output power constraints are showed in (4.4)-(4.8).

$$0 \leq p_{w_t} \leq P_t^w \tag{4.4}$$

$$0 \leq p_{s_t} \leq P_t^s \tag{4.5}$$

$$0 \leq p_{nse_t} \tag{4.6}$$

$$0 \leq p_{ex_t} \tag{4.7}$$

$$p_{g,t} \leq v_{g,t} \cdot \left(\overline{P_g} - \underline{P_g}\right) \tag{4.8}$$

where:

P_t^w is the maximum wind power available in period t.

P_t^s is the maximum solar power available in period t.

$\overline{P_g}$ is the maximum power of unit g.

$\underline{P_g}$ is the minimum power of unit g.

$p_{g,t}$ is the power produced above the minimum stable load by unit g in period t. This variable is related to $t_{g,t}$ according to (4.9).

$$t_{g,t} = \underline{P_g} \cdot v_{g,t} + p_{g,t} \tag{4.9}$$

iii) The logic coherence startup-commitment-shutdown constraints is showed in (4.10).

$$y_{g,t} - v_{g,t} - z_{g,t} + v_{g,t-1} + EI_g = 0 \tag{4.10}$$

It should be noticed that for the period t=1, $v_{g,t-1}$ should be substituted by EI_g, which is the initial commitment state of unit g.

iv) The shut-down and start-up of the units at minimum stable load constraints are showed in (4.11) and (4.12).

$$p_{g,t} \leq \left(\overline{P_g} - \underline{P_g}\right) \cdot \left(v_{g,t} - z_{g,t+1}\right) \tag{4.11}$$

$$p_{g,t} \leq \left(\overline{P_g} - \underline{P_g}\right) \cdot \left(v_{g,t} - y_{g,t}\right) \tag{4.12}$$

v) The spinning reserves constraints are showed in (4.13) and (4.15).

$$\sum_g \left[v_{g,t} \cdot \overline{P_g} - t_{g,t}\right] \geq res_{sub_t} \tag{4.13}$$

where:

res_{sub_t} is the upwards spinning reserve in period t. It is calculated as showed in (4.14).

$$res_{sub_t} = Res_{sub} \cdot d_t \tag{4.14}$$

where:

Res_{sub} is level of backup generation to ensure to cover demand in case there is any outage in any unit. It is set as 2% of the demand value.

$$\sum_{g} [p_{g,t}] \geq res_{baj_t} \tag{4.15}$$

where:

res_{baj_t} is the downwards spinning reserve in period t. It is calculated as showed in (4.16).

$$res_{baj_t} = Res_{baj} \cdot d_t \tag{4.16}$$

where:

Res_{baj} is ability of the system to reduce generation quickly in the case of a drop in demand. It is set as 1% of the demand value.

vi) The ramp rates constraints are showed in (4.17) and (4.18).

$$p_{g,t} - p_{g,t-1} \leq RS_g \tag{4.17}$$

$$p_{g,t-1} - p_{g,t} \leq RB_g \tag{4.18}$$

where:

RS_g is the upwards ramp rate of unit g.

RB_g is the downwards ramp rate of unit g.

It should be noticed that for the period t=1, $p_{g,t-1}$ should be substituted by PI_g, which is the initial output power above minimum stable load of unit g.

vii) The minimum up and down time constraints are showed in (4.19) and (4.20).

$$\sum_{i=t-TU_g+1}^{t} y_{g,i} \leq v_{g,t} \tag{4.19}$$

$$\sum_{i=t-TD_g+1}^{t} z_{g,i} \leq 1 - v_{g,t} \tag{4.20}$$

where:

TU_g is the minimum time that unit g should be committed once it is started up.

TD_g is the minimum time that unit g should be off once it is shut down.

4.4 SEVERAL WEEKS MODEL

As explained in Section 4.2. Input Data, there are 40 consecutive weeks as input datasets. Therefore, this model is designed so that several weeks can be ran consecutively as desired instead of going week by week individually. In addition, there is also the possibility to start in a week different from the first one and to end in a week different from the last one, in chronological terms.

This means that the 40 weeks of the input dataset can be ran automatically (starting with the 1st and ending with the last in chronological order), or also a smaller number of input weeks can be ran, for example 20 (starting with any desired week and ending 20 weeks later). In other words, there is complete flexibility in deciding the time horizon of the optimization.

In order to do this, it has been taken into consideration that the initial state (EI_g and PI_g) of the coming week (week n+1) should matched to the commitment state of the final period ($v_{g,168}$ and $p_{g,168}$) of the previous week (week n).

4.5 OPTIMIZATION SETTINGS

The solver used for the optimization is Gurobi, which is also fully compatible with python. The solving method used is dual simplex.

This model is a small-scale problem, so it is quickly solved. In addition, a maximum execution time of 300s for each week has been determined, although this is not really significant since at the time of running the optimization, all the optimal solutions were reached before that time.

Finally, one last aspect related to nuclear units that affects optimization should be mentioned. In terms of logic, it is inconsistent that nuclear units are being turned on and off indiscriminately, both because of their slow start-up and cooldown times and because of their start-up and shut-down costs.

In practice, these units are always on and act as the base technology. Therefore, to simulate this, it has been decided to set the commitment variables of the nuclear units always on. In addition, this facilitates optimization by "eliminating" these variables from the iteration process.

Chapter 5. NEURAL NETWORK

This section explains the main characteristics of the ANN that has been designed to estimate and generalize the UC model.

The ANN will use as input the hourly demand data and the wind and solar forecasts and will make predictions of the output power of the thermal units. This is shown in the scheme of Figure 5-1

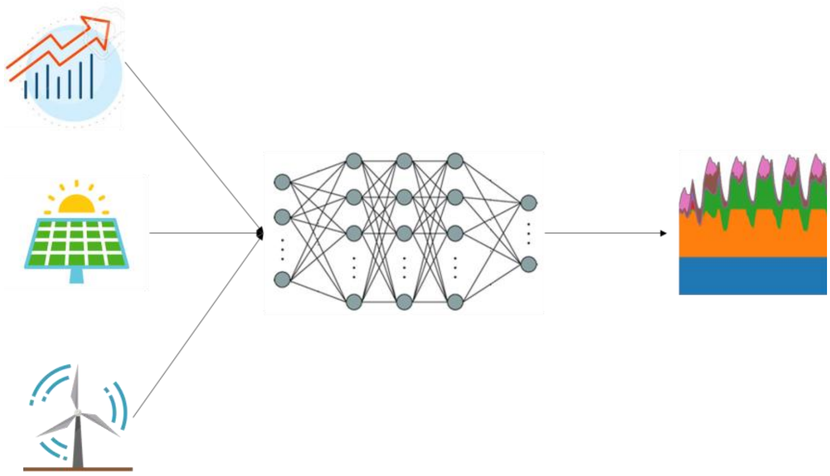

Figure 5-1: Neural Network Data

In other words, the model being created is actually an ANN that performs a linear regression, since it takes as input a set of linear values and generalizes them to another set of linear values.

5.1 DATA PROCESSING

Before working with the model, it is necessary to perform a pre-processing stage for the input data. The neural network model works with data tensors, due to the library used in the development of the model (further explained in the next section), so it is necessary to reformat the available data to these tensors.

Therefore, the matrices shown in Table 3 and Table 4 have been formed, which later are transformed into tensors.

Table 3. Matrix X Structure

Weeks	Demand (168 periods)	Wind Forecast (168 periods)	Solar Forecast (168 periods)
01	Demand_t_{01},…,Demand_t_{168}	Wind_t_{01},…,Wind_t_{168}	Solar_t_{01},…,Solar_t_{168}
02	Demand_t_{01},…,Demand_t_{168}	Wind_t_{01},…,Wind_t_{168}	Solar_t_{01},…,Solar_t_{168}
03	Demand_t_{01},…,Demand_t_{168}	Wind_t_{01},…,Wind_t_{168}	Solar_t_{01},…,Solar_t_{168}
04	Demand_t_{01},…,Demand_t_{168}	Wind_t_{01},…,Wind_t_{168}	Solar_t_{01},…,Solar_t_{168}
…	…	…	…
40	Demand_t_{01},…,Demand_t_{168}	Wind_t_{01},…,Wind_t_{168}	Solar_t_{01},…,Solar_t_{168}

Table 4. Matrix Y Structure

Weeks	P_g1 (168 periods)	…	P_g16 (168 periods)
01	P_g1_t_{01},…,P_g1_t_{168}	…	P_g16_t_{01},…,P_g16_t_{168}
02	P_g1_t_{01},…,P_g1_t_{168}	…	P_g16_t_{01},…,P_g16_t_{168}
03	P_g1_t_{01},…,P_g1_t_{168}	…	P_g16_t_{01},…,P_g16_t_{168}
04	P_g1_t_{01},…,P_g1_t_{168}	…	P_g16_t_{01},…,P_g16_t_{168}
…	…	…	…
40	P_g1_t_{01},…,P_g1_t_{168}	…	P_g16_t_{01},…,P_g16_t_{168}

In addition, the objective sought when designing a neural network is that the model should be able to generalize, which means being able to solve the problem that was presented during training, but with new data, which the network has not seen before.

Then, if all the available data are used to train the model, the model's ability to make predictions for new data would be unknown, i.e., the model's ability to generalize would be unknown. So, to solve this, three different datasets are used: training, validation, and test.

The training set is used to train the model, the validation set is used to evaluate the model's ability to generalize while training is in progress, and the test set is used to evaluate the model's ability to generalize with data that it has never seen before and has not been used for training.

Therefore, only one week has been set as test dataset to evaluate the neural network, and the 39 weeks remaining have been used for training the model. Within these 39 weeks of training, a validation split of 0.2 (20% validation, 80% train) has been established, since it is a very widespread practice.

The last aspect with respect to data processing is normalization. This technique consists of leveling the data so that they follow a normal distribution and are centered, to prevent their own scale from affecting the model's predictions.

In fact, as this model is a generalization of a UC problem, which contains several sets of large power values, the numerical values that will be used for training the model will be certainly large, so the implicit training and estimation errors will be excessive as well.

Therefore, it has been performed an automatic pre-scaling and normalization of the data to unitary values (between 0 and 1). In this way, large error values in the estimations and training will be avoided, and the values obtained can be directly perceived as percentages, which is more illustrative in analytical terms.

This normalization technique allows to decrease the error and reduce the time it takes to train the model. This can be seen in Figure 5-2, which shows a comparison between the model being trained with normalization and the model being trained without normalization of the data.

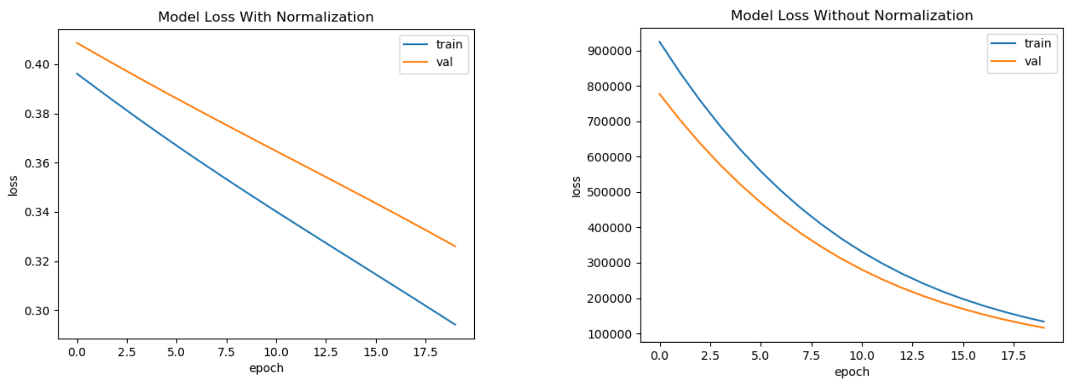

Figure 5-2: Normalization vs No Normalization

Lastly, nuclear units have been removed for the modelling of the neural network. Nuclear power is always a constant, so if introduced in the model, it may introduce distortions in so worse results would be obtained. In addition this would increase the number of variables, which leads to an increase in the output layers and a longer execution time.

5.2 CONFIGURATION

Once data has been processed, the following phase is modelling the neural network. This is an iterative process, not sequential as shown through this section, in which the optimal architecture and parameters must be found simultaneously to provide the optimal results overall.

First, it is necessary to select the programming environment in which the model will be developed. In this sense, there are several libraries for developing Machine Learning models, although the three most notorious options are TensorFlow, Keras and Pytorch.

In addition, there is also the possibility of developing the entire model mathematically, but this is withdrawn in advance for any type of project, given that this would be a very complex approach and would involve a large amount of time.

The existence of libraries makes it possible to avoid this mathematical development and simplifies the process to basically calling some default functions that already include all the equations and mathematical algorithms internally.

On the one hand, TensorFlow has the main strength of presenting a great balance between flexibility and scalability, which allows to create from quick and easy pre-defined models to create undefined ones. This library is based on two aspects: tensors (multidimensional matrices), and graphs that represent the flow, but the latter are used to design the algorithms and do not take part in the design level that will be carried out in this project.

Then, Keras is a library that can be used in combination with others, including TensorFlow. Its main strength is its manageability, which allows to develop models simply. This is because this library offers many possibilities, from easily connectable layers to pre-trained models, which makes it very popular and attractive to users.

Finally, Pytorch implies that all steps of the model must be programmed in detail. This means that the layers (parameters, sizes...) of the model must be manually decided, and data flow and loops for training must be manually programmed. In other words, this brings a lot of flexibility and customizability to the model, but it implies a large amount of time spent in detailed programming.

Therefore, based on a cost-benefit analysis, the most attractive choice for developing the model is a combination of the TensorFlow and Keras libraries. Although PyTorch is more flexible than both libraries, the complexity in programming it implies represents a very large cost. Contrary, the benefit obtained from the increase in flexibility is relatively limited since the flexibility offered by the combination of TensorFlow and Keras is more than sufficient for the UC problem to be generalized.

The next step in the model development is the choice of the "neural network type", i.e. to select the architecture of the model: the configuration of the layers and the ways in which they interact.

Based on simplicity and versatility, a three-layer multilayer perceptron (MLP) has been chosen. Those three layers are known as input layer, hidden layer, and output layer. This configuration offers many possibilities and stands out for its flexibility and understandability, which makes this architecture ideal for the proposed scenario.

In each layer, 4096, 4096 and 2352 neurons have been determined, respectively. For the input and hidden layers, the number of neurons has been adjusted continuously according to the results obtained in each iteration of the designing process, but always setting this number as a power of 2, which is a generalized practice.

On the other hand, for the output layer, the number of neurons has been set to 2352, which is equal to the number of output variables of the neural network model (168 time periods x 14 generator units). This is due to the model's training algorithms configuration.

Then, the next step in the neural network model design process are the training algorithms. In general terms, neural network training is based on two mutually dependent algorithms: forward-propagation and back-propagation.

The forward-propagation algorithm consists of processing the data from the input layer to the output layer, so that, in each layer, by interacting with the neurons (weights), data is transformed to generate an output.

Next, the back-propagation algorithm consists of comparing the outputs obtained in the forward-propagation process with the real outputs of the dataset, so that the weights of the neurons are adjusted to make the output of the algorithm and the dataset similar. In other words, the network "learns" and adapts to the data.

This two-step procedure is an iterative process whose number of repetitions, known as number of epochs, is also one of the parameters of the training process, which must be carefully determined as explained below.

In this sense, as the model is developed under the TensorFlow and Keras environment, there are several predefined functions and blocks that simplify this training process. One of these possibilities is the use of an optimizer that executes the forward-propagation and back-propagation algorithms of the network automatically.

There are several possible optimizers (Momentum, SDG, RMSprop, Ftrl...), but it has been used the Adam optimizer since it is a combination of the other optimizers and also has relatively low memory requirements.

Nevertheless, the libraries allow to change some default parameters of this optimizer, to offer a greater customization. This facilitates the iterative process of optimizing the model design. In particular, the most interesting parameters are learning rate, number of epochs, and activation function.

First, the learning rate establishes the speed at which the network weights are changed in the back-propagation algorithm, i.e., the degree to which the weights are modified as a function of the forward-propagation result. It therefore determines the speed of network learning. This parameter always has a positive value but is generally small; lower than 1.

Neither an excessively large nor an excessively small learning rate should be set, since both cases are detrimental to training. An excessively small learning rate implies greatly lengthening the training process, while, if this value is excessively large, the model may not converge, never finding an optimal solution. This can be seen in Figure 5-3 - Figure 5-9.

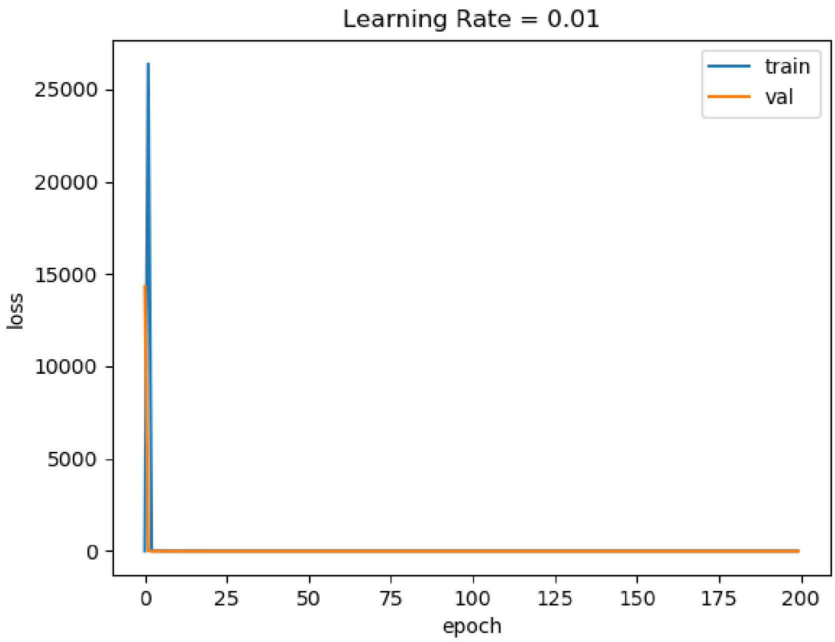

Figure 5-3: Learning Rate = 0.01

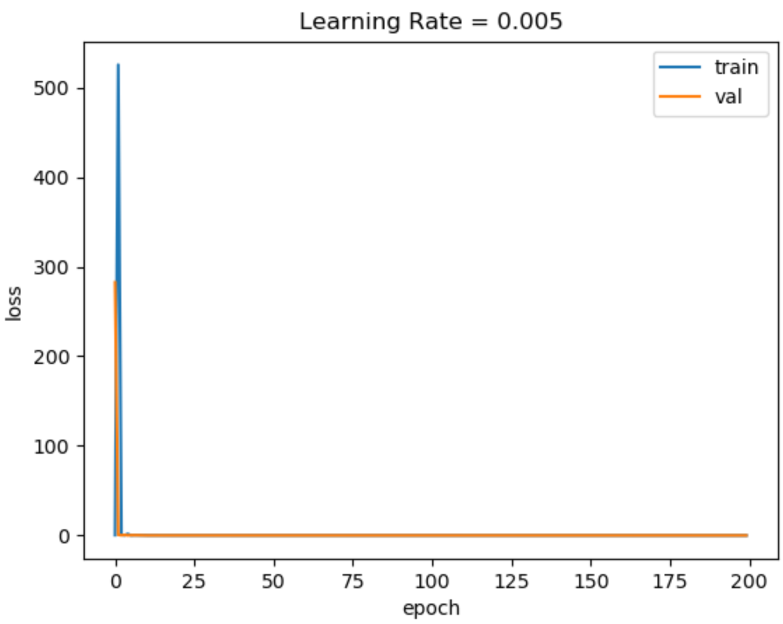

Figure 5-4: Learning Rate = 0.005

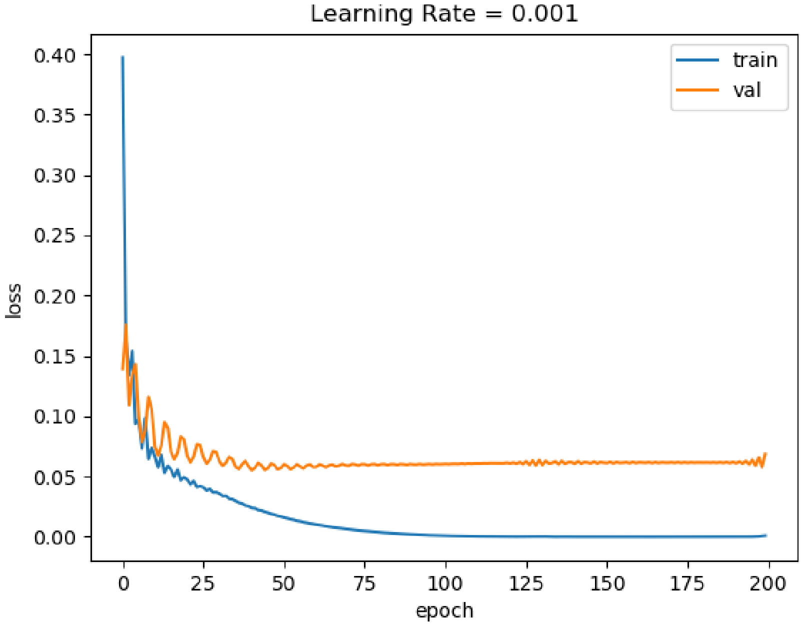

Figure 5-5: Learning Rate = 0.001

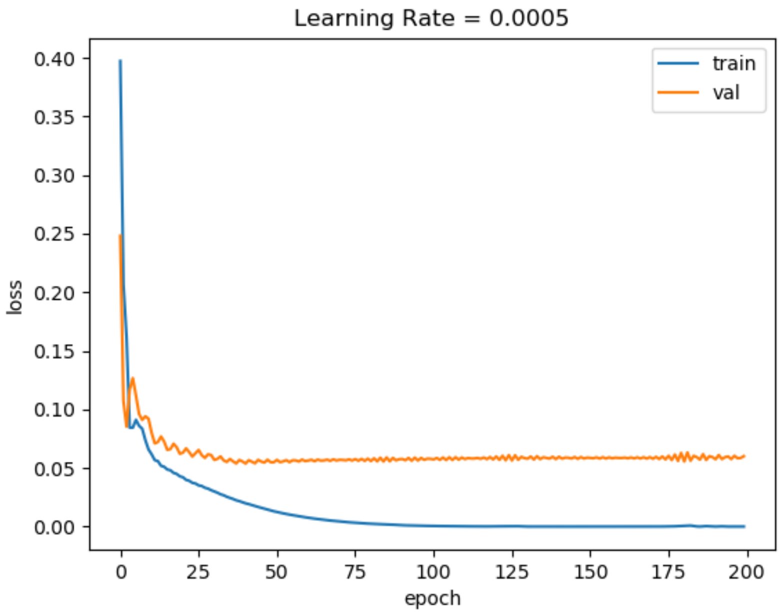

Figure 5-6: Learning Rate = 0.0005

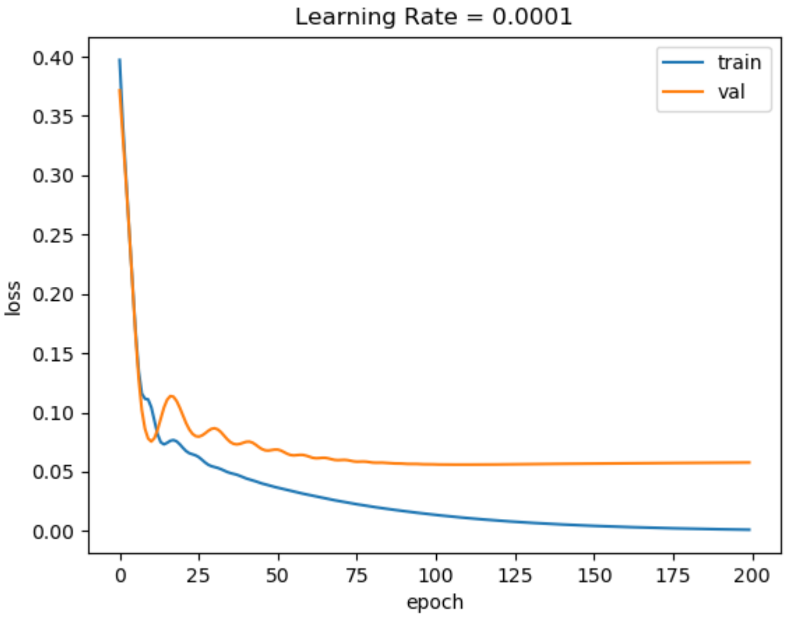

Figure 5-7: Learning Rate = 0.0001

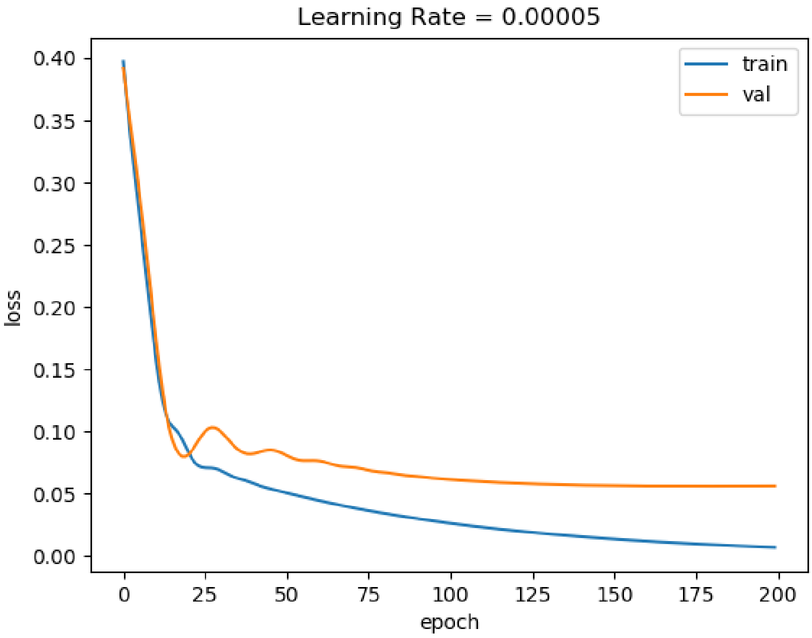

Figure 5-8: Learning Rate = 0.00005

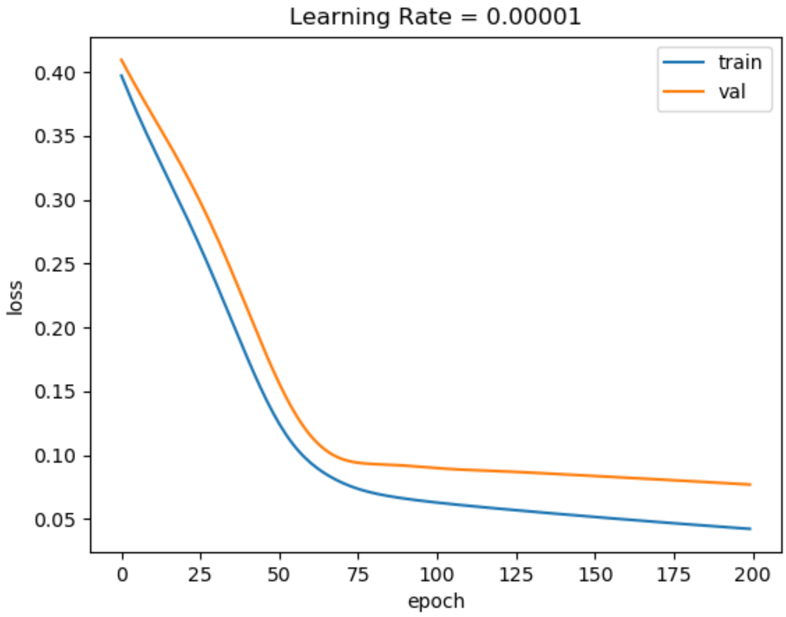

Figure 5-9: Learning Rate = 0.00001

The technique used to determine this parameter consists of starting with a clearly high learning rate value and gradually reducing it until it is close to the optimal. This is, initially starting with a "rough adjustment" of this parameter, and when better results are obtained, a "fine adjustment" is made.

On the other hand, an epoch is the process of running the entire dataset forward (forward propagation) and backwards (back-propagation) through the model. The number of epochs is a fundamental parameter for defining the length of model training. The training of the model does not end until the epochs that have been set are completed, so selecting this number correctly allows the training process to stop at the optimal time.

Finally, the activation function defines the transformation that occurs at the output of each neuron in the network, after the transformation by the network weights. There are different types of activation functions (sigmoid, tanh...), and each one is appropriate for certain situations.

The proposed problem only presents positive values, so it has been determined to use the ReLu function, which does not modify those positive values, and ignores negative values. In this way, possible inconsistencies in the results of the problem are avoided.

Once the main parameters have been explained, the technique used to optimize their value must be explained. Although there are possibilities of programming (Grid Search) or using algorithms (Bayesian Search) to find these optimal values, these techniques are really complex to implement and require a lot of computing time.

Therefore, it has been implemented a simpler technique, known as Manual Search, which consists of manually modifying the parameter values, according to the results obtained in each iteration.

Then, the last step of this iterative process consists of evaluating the training process and its parameters, and subsequently making the changes considered appropriate to achieve better results.

The main evaluation method consists in the study of the evaluation graphs, which are fundamental to understand how the training has evolved over the course of the different epochs. These graphs thus allow the identification of harmful trends in the model (overfitting, underfitting...) and helps to identify the modifications needed to improve the predictions.

Generally, an analysis is carried out on the loss plot, which presents two curves, one representing the results on the training dataset and the other with the validation. This is shown in Figure 5-10 below.

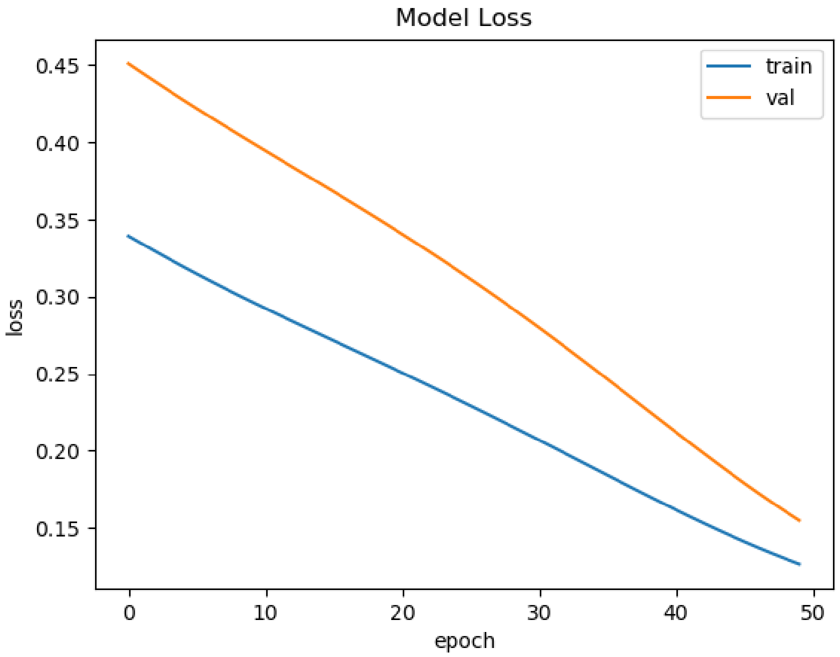

Figure 5-10: Loss Plots

This graph shows the variation of the loss function over the course of the training epochs. This function quantifies the difference (the error) between the model estimates and the actual data values, i.e., it measures the similarity between predictions and actual values.

Naturally, this graph shows a decreasing trend, i.e., as training proceeds, the model learns more about the data, and the estimates are closer to the actual values.

In addition, having two curves (training and validation) in this graph is fundamental to evaluate how the model evolves, since by comparing both curves it is possible to detect whether it is learning to identify patterns and also is possible to evaluate the model's capacity to generalize adequately.

5.3 PARAMETERIZATION & EVALUATION

In conclusion, as explained in previous sections, this neural network design process is an iterative process in which the main training parameters are adjusted until an optimal solution is found..

After applying a Manual Search technique, the optimal parameters values were found. A sum-up table with the values of the main parameters used in the model is shown below in Table 5.

Table 5. Neural Network Parameters

Train Set	39 Weeks		
Test Set	1 Week		
Validation Split	0.2		
Library	TensorFlow + Keras		
Architecture	MLP		
Layers	Input	Hidden	Output
Neurons	4096	4096	2352
Activation Function	ReLu	ReLu	Linear
Learning Rate	1,00E-05		
Technique	Manual Search		

The last step in the process is model evaluation and choosing the number of epochs of the training process. Therefore, it is necessary to check the loss plots and select the number of epochs in which there is an elbow in validation and train loss plots.

For this reason, the number of epochs has been set to 65. This can be seen in Figure 5-11.

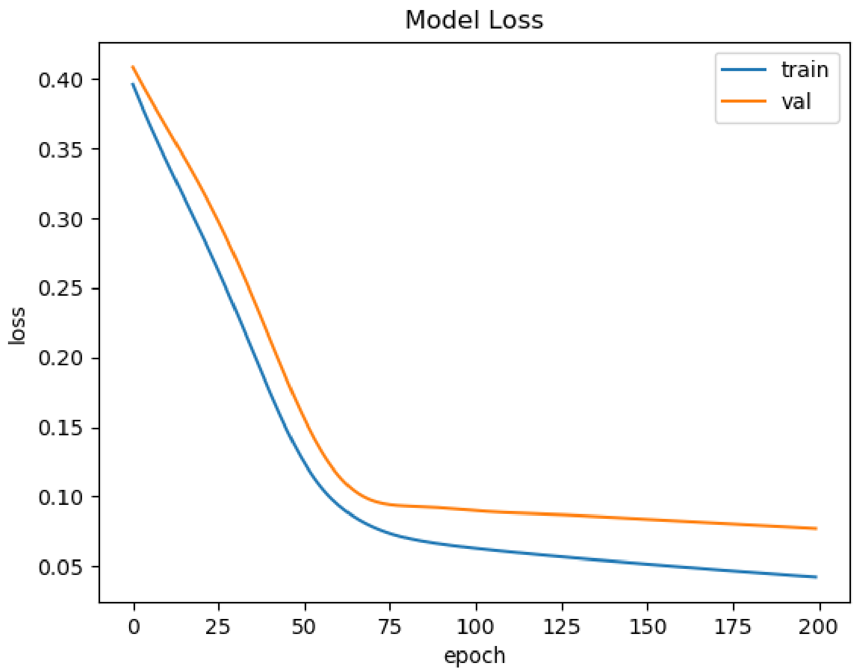

Figure 5-11: Model Loss Plot

Chapter 6. NN-UC INTERACTION

The next step in this thesis methodology is achieving interaction between the ANN and the UC optimization model. That is, getting the ANN estimates to serve as a starting point for the UC model.

A priori, this may seem trivial since at first sight this would only imply imposing that the optimization model starts the iterative process from the powers estimated by the ANN.

However, there is a problem intrinsic to the nature of the ANN that delays this process of interaction with the UC model. This is because ANN predictions are based on learning and generalization of trends in the data, not on the equations by which these data are governed.

Thus, ANN results are really approximations, so there are some deviations in the predictions from the actual values. Then, when interacting with an optimization model, these deviations can be critical, since a constraint can be satisfied or not fulfilled depending on these small variations.

More specifically, within the UC model there are a number of constraints that any feasible solution must satisfy (power limits of the groups, ramps, etc...). However, ANN predictions may not meet some of these constraints, making its solution infeasible, even if it is a good approximation.

A priori, this would not be a drawback in practical terms, but the main problem is that these ANN approximations cannot be used as a starting point to begin the iterative optimization process of the UC model.

That is, the ANN approximations, although valid in terms of logic and usability, might not be valid to feed the UC optimization model in the iterative solving process.

In programming jargon, when it is imposed to start the optimization procedure from a solution instead of starting from scratch, this is called warm-start. Then, if the warm-start is an infeasible solution, the optimizer automatically rejects it and chooses to start the process from scratch.

To solve this, and to prevent possible errors, it has been created a feasibility optimization model that is in charge of making the ANN solution feasible. This is that converts an infeasible solution into a feasible one, minimizing the necessary deviations with respect to the ANN solution.

In other words, this feasibility model is not a complementary UC model, but simply a model that tries to adapt the ANN solution to an admissible solution as a starting point for the UC model.

Then, in order to convert the ANN solution into a feasible starting point for the UC model, this feasibility model must consider the constraints of the UC model. However, the main difference with respect to the UC model is the objective function.

While in the UC, the objective is about minimizing operating costs and avoiding non-compliance with demand (both in excesses and defects), in this model the objective is to minimize the difference with respect to the ANN solution while respecting the operating constraints of the generating sets (power limits of the sets, ramps, etc...).

That is to say, in mathematical terms, this feasibility model maintains the constraints expressed in equations (4.2)-(4.20), but instead replaces the objective function expressed in equation (4.1) by the function of equation (6.1), and also adds a new constraint (6.2).

$$\sum_{t}\sum_{g}[\alpha_{g,t} + \beta_{g,t}] \tag{6.1}$$

$$t_{g,t} = NN_{g,t} + \alpha_{g,t} - \beta_{g,t} \tag{6.2}$$

where:

$NN_{g,t}$ is the ANN approximation for unit g in period t.

$\alpha_{g,t}$ is excess of energy with respect to ANN approximation for unit g in period t.

$\beta_{g,t}$ is deficiency of energy with respect to ANN approximation for unit g in period t.

Naturally, this intermediate feasibility process involves a longer overall computation time until a solution of the complete UC problem is obtained. However, this feasibilization process is unavoidable, and although it entails additional time, it does not imply that the methodology stated in this thesis is no longer competitive with respect to the classical solution of the UC model.

Chapter 7. RESULTS

7.1 UNIT COMMITMENT MODEL

As explained earlier in this thesis, the UC model was run for the 40 weeks for which demand data were available.

Some graphs (Figure 7-1 - Figure 7-5) with the results obtained are shown below. In order to simplify the analysis of the solution, the dispatch of the units is grouped according to the type of technology of each generation unit.

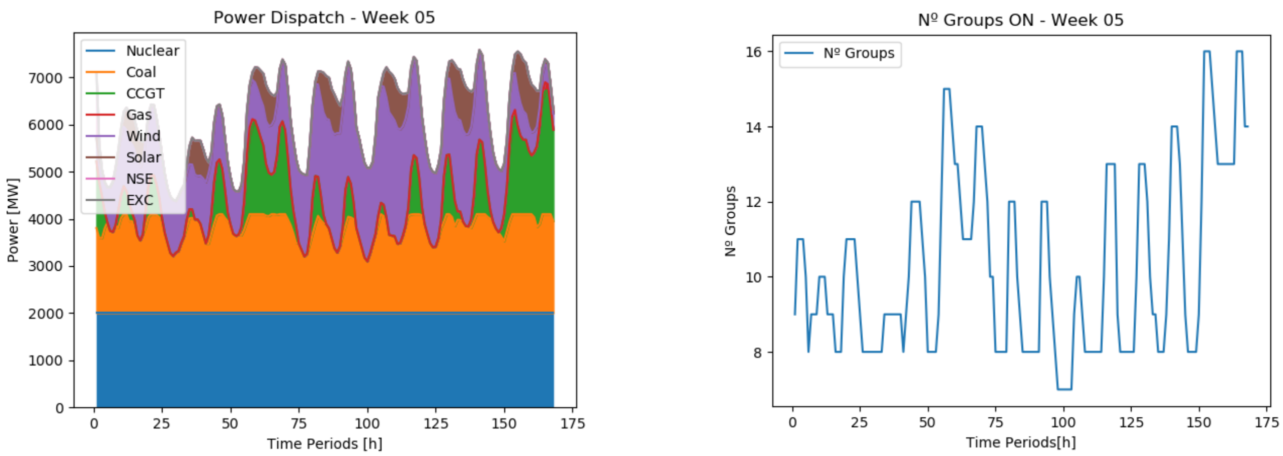

Figure 7-1: UC Results Week 5

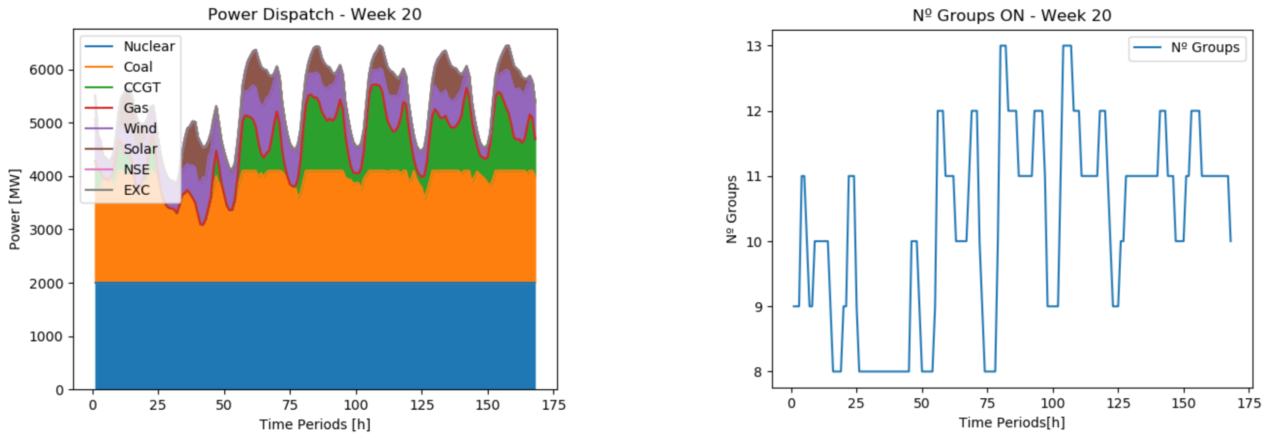

Figure 7-2: UC Results Week 20

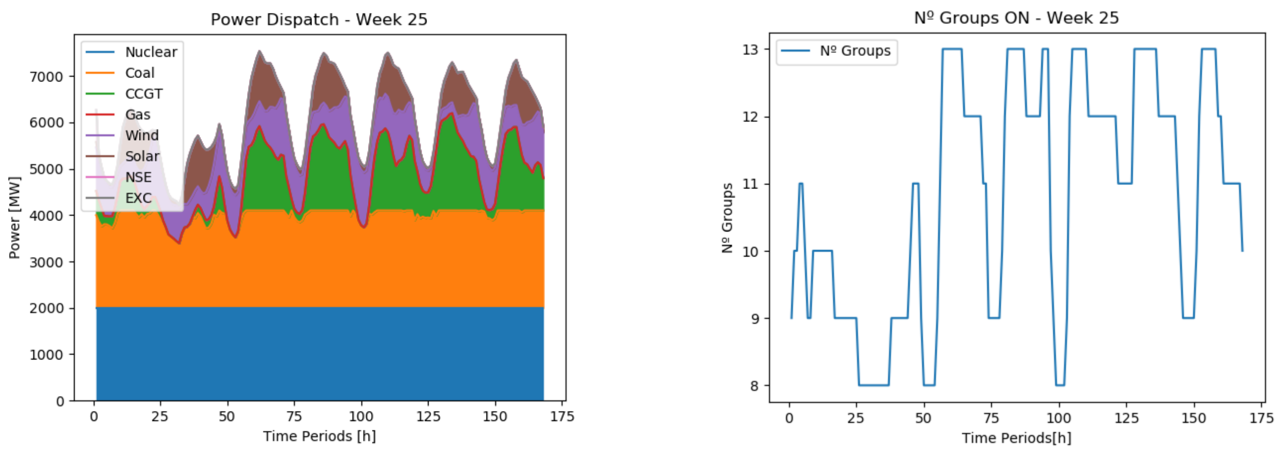

Figure 7-3: UC Results Week 25

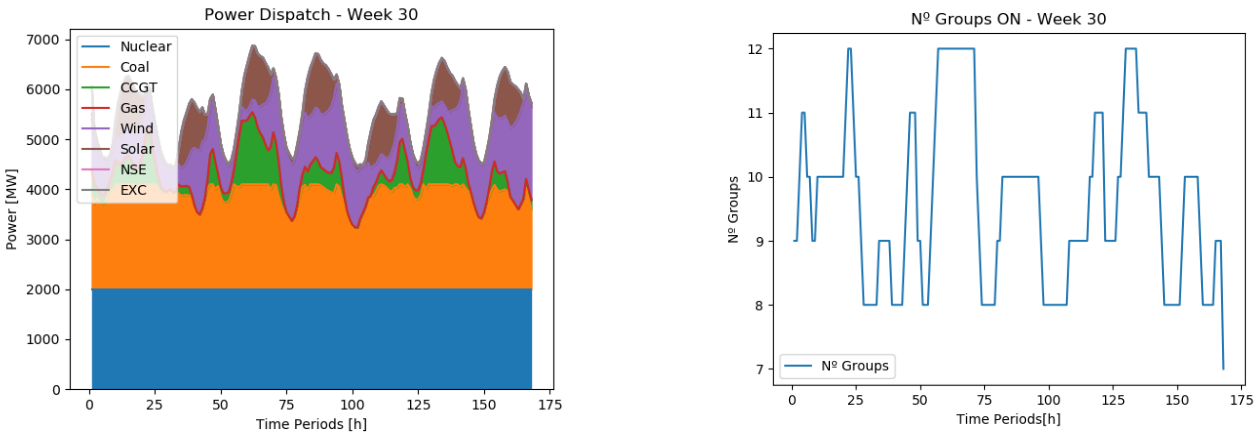

Figure 7-4: UC Results Week 30

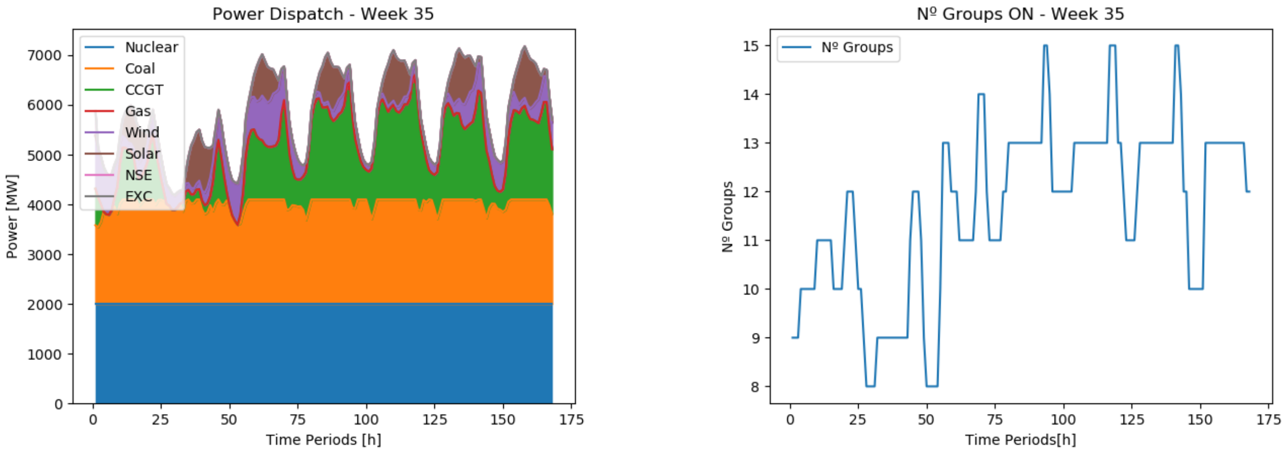

Figure 7-5: UC Results Week 35

Firstly, it can be seen that, as mentioned in 4.5. Optimization Settings, the nuclear units are kept turned on during all the hourly periods and their power has remained constant at their maximum. In other words, the nuclear units have been established as the base technology.

However, it can be seen how other technologies such as Coal units are turned ON during all the periods, but its output power is variable, depending on the demand required by the system.

On the other hand, other technologies such as CCGTs and more notably Gas units, have a residual character, and are only used as peak technology to cover specific demand defects.

These differences in the behavior of the technologies are due to their different costs. That is to say, those with higher fixed costs (start-up costs, shut-down cost, etc.) have a base behavior and those with higher variable costs have a residual behavior.

In addition, it can be appreciated how the model has tried to maximize the use of solar and wind technologies. This is logical since the use of these technologies is not penalized as a cost in the objective function.

Furthermore, it can be inferred from the solutions obtained that there is a similar behavior of all the units within most weeks. However, there are also some other outlier weeks that differ from average, for instance, Week 5 has a lot of wind power and a very few CCGT participation. This is a significant aspect because it directly affects the estimates and learning of the neural network.

7.2 NEURAL NETWORK MODEL

To evaluate the results of the neural network model, some graphs are presented (Figure 7-6 - Figure 7-10) showing a comparison between the network prediction and the results of the optimization model. In these graphs the continuous line represents the results of the optimization model while the dashed lines represent network predictions.

It should be noted that the nuclear technology does not appear in these graphs since it was not introduced in the learning process to avoid possible distortions.

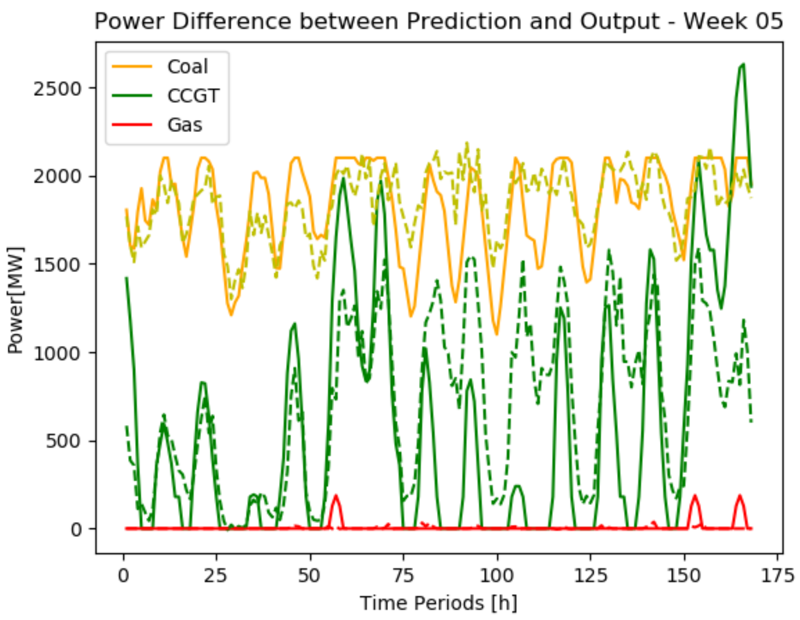

Figure 7-6: ANN Predictions Week 5

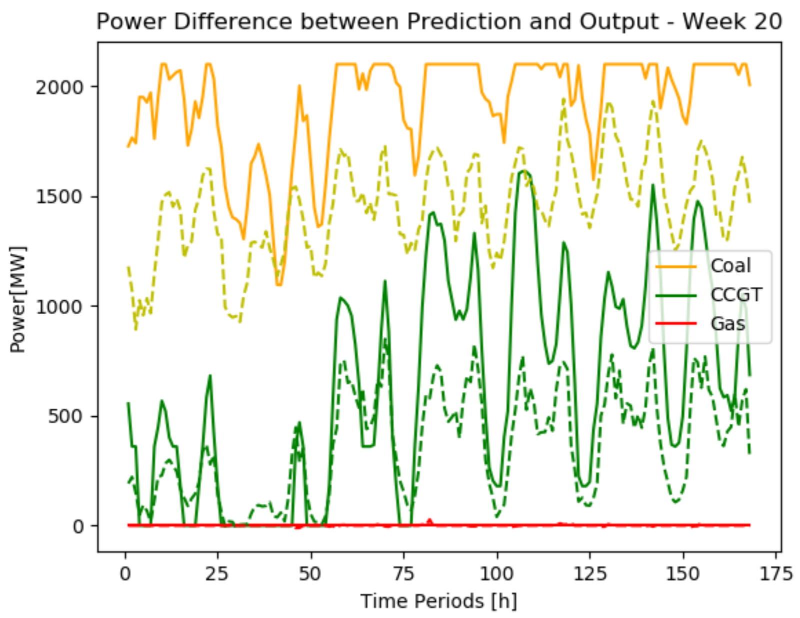

Figure 7-7: ANN Predictions Week 20

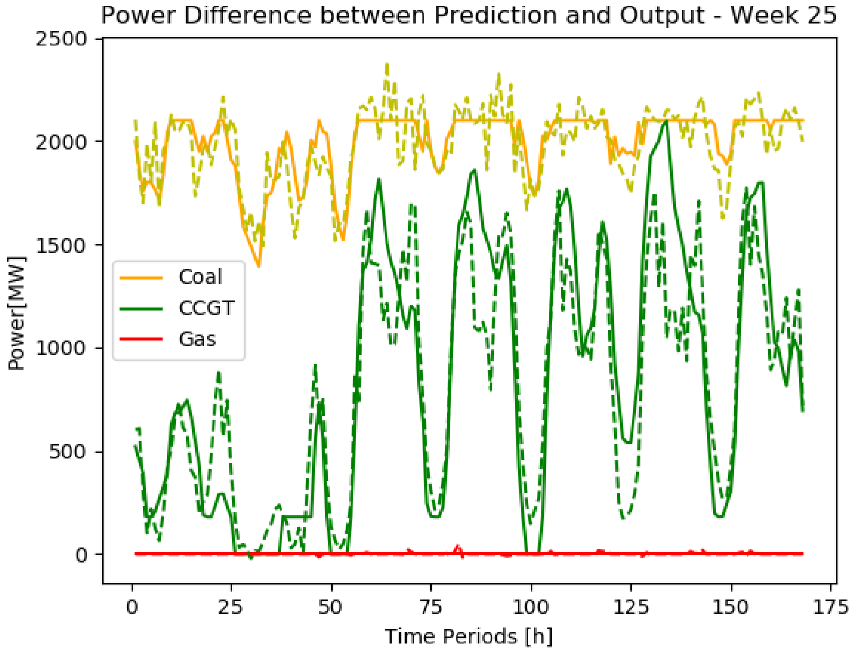

Figure 7-8: ANN Predictions Week 25

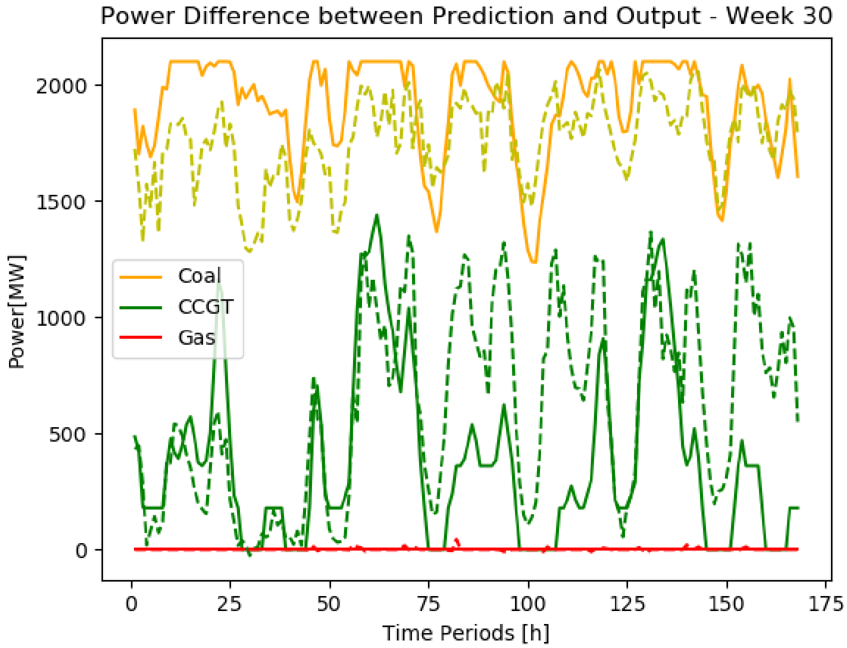

Figure 7-9: ANN Predictions Week 30

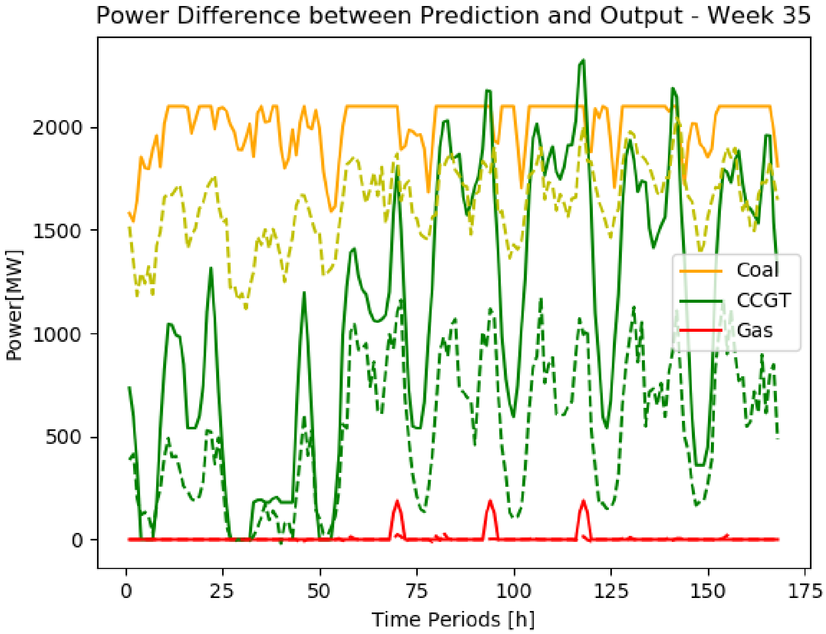

Figure 7-10: ANN Predictions Week 35

In general, the neural network model learns satisfactorily the patterns presented by the UC model. That is, it recognizes those technologies that act with a higher base technology tendency and those with more residual behaviors.

This can be clearly observed if behavior of CCGTs or Coal Units and Gas units is compared, while first ones are always present, but its power varies, the latter only appears residually with small peaks.

7.3 UNIT COMMITMENT – NEURAL NETWORK INTERACTION

When evaluating the interaction, first it is shown a comparison between the results provided by the feasibility model with respect to the estimates of the neural network model. This is presented in Figure 7-11 - Figure 7-15. In these graphs the continuous lines represent the feasibilization model results while the dashed lines represent the network predictions.

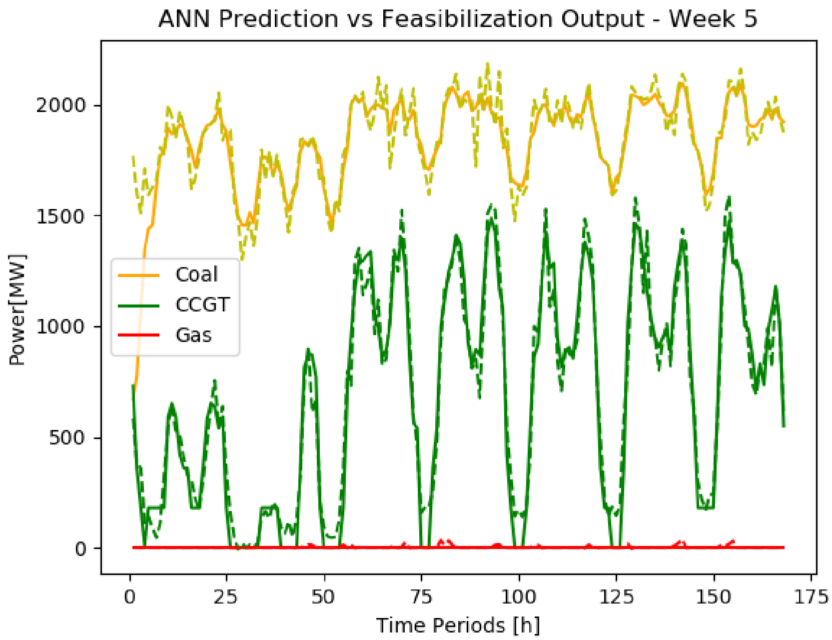

Figure 7-11: ANN vs Feasibilization – Week 5

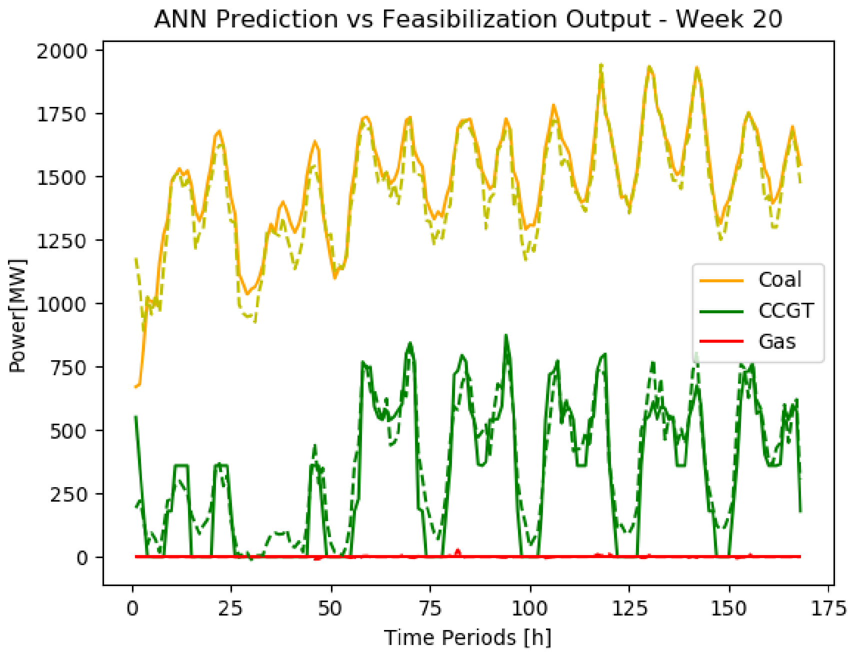

Figure 7-12: ANN vs Feasibilization – Week 20

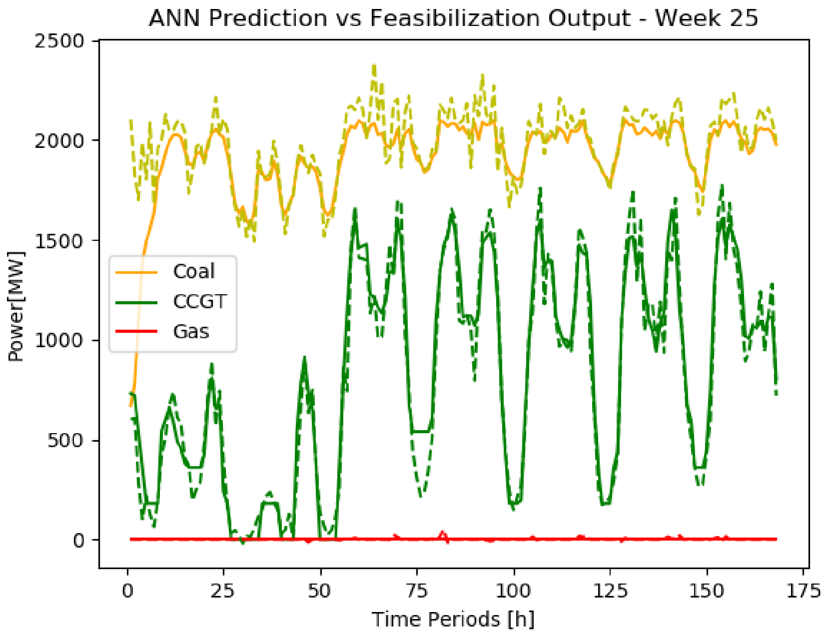

Figure 7-13: ANN vs Feasibilization – Week 25

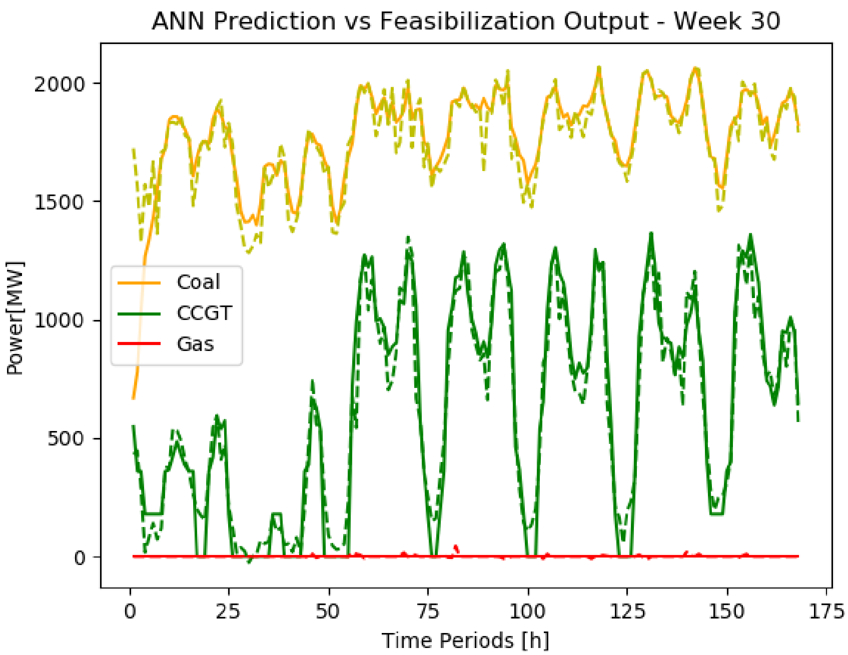

Figure 7-14: ANN vs Feasibilization – Week 30

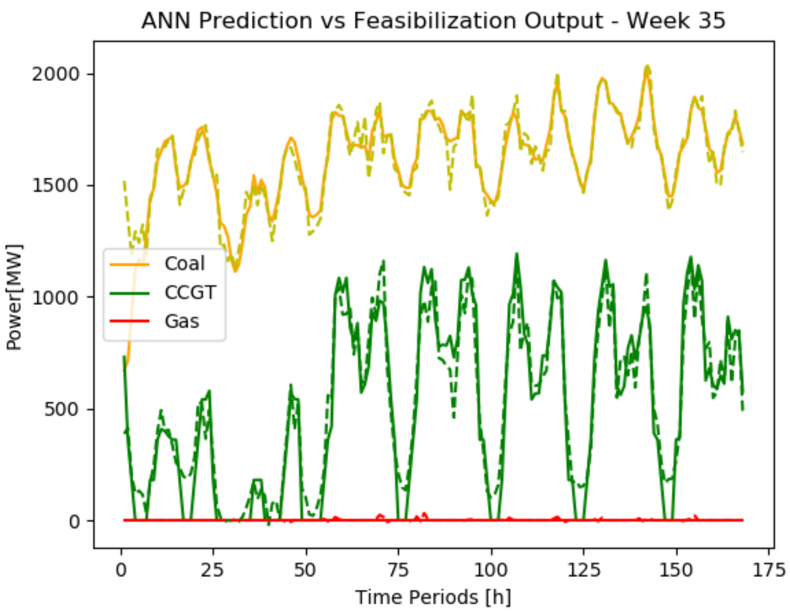

Figure 7-15: ANN vs Feasibilization – Week 35

It can be seen that, as expected, the results obtained once the feasibility model is executed are slightly modified with respect to the neural network estimations.

Finally, the time it takes to run the optimization model starting from scratch, with no warm-start (Case B), compared to the time it takes to run the optimization model starting from the solution of the feasibilized neural network (Case A) has been calculated. Results are shown in Table 6.

Table 6. Interaction UC – NN. Results

Computation Times (secs)				
Case	**Case A**			**Case B**
Proccess	Feasibilization	Optimization	Total	Total
Week 05	13,54	69,17	82,71	53,41
Week 20	11,93	25,77	37,7	27,43
Week 25	12,37	13,26	25,63	11,53
Week 30	13,58	37,50	51,08	38,80
Week 35	15,49	18,08	33,57	12,99

The optimization times for both cases, shown in Table 6, are very similar in all the weeks that been executed. However, if the execution time of the Feasibilization model is added to the execution time of the optimization model, the overall execution times on the proposed methodology are larger than the conventional optimization procedure.

However, this is because the model used, although valid, is not sufficiently complex and extensive to show a significant improvement in execution times.

If the model used was more extensive and required a longer execution time on its own, it would be appreciated an improvement in the overall execution time when applying this methodology.

This means that the methodology explained in this thesis is focused for large-scale systems with large execution times, while for smaller scale problems, where execution times are much smaller, it loses effectiveness. This can be appreciated in the following section.

7.4 LARGE-SCALE CASE STUDY

In order to analyze the impact in a large-scale scenario, all the process has been repeated changing the input data. The demand was multiplied by a factor of 10, and the number of generation units is 10 times bigger. For the generation units, to avoid problem degeneration, their variable cost has been modified a little with a gaussian noise to avoid having identical units.

The execution of the 40 weeks to feed the results to the ANN took a 14h, and the ANN training took half an hour.

Then, the main results obtained are summarized in Table 7, in which the main differences between the two optimization procedures can be observed. Case A uses the proposed methodology in this thesis, in which ANN estimations are used as starting point for the optimization procedure and Case B is the direct optimization procedure.

Table 7. Large Scale Case Study - Results

Case		Case A	Case B
Optcr Requested		1%	1%
Max Time Allowed (mins)		20	20
Optcr Achieved		0,917%	0,854%
Time (mins)	Feasibilization	2,758	-
	Optimization	8,895	19,781
	Total	11,653	19,781
Warm-start Objective Function		2,481E+09	-
Best Bound Objective Function		1,873E+08	1,876E+08
Achieved Objective Function		1,890E+08	1,892E+08

It can be appreciated that the computation time in Case A is much lower than in Case B. In fact, the execution time required to obtain an optimal solution using ANN estimations as starting point (Case A) is reduced in 41.1% compared to the traditional resolution procedure (Case B).

In addition, both the objective function and optcr in both cases are also presented in Table 7. This is because it is possible that a lower objective function does not necessarily means a better result, since it could have a worse optcr.

Optcr is the parameter that determines the difference between the obtained solution and the best solution that the optimizer estimates that could be achieved. When this value is reduced, it could be due to two different aspects: i) the optimizer improves the obtained solution or ii) the optimizer makes a more realistic estimation.

However, in both cases, objective function and optcr are similar, so Case A and Case B are considered to have equivalent solutions, but being execution time for Case A considerably lower than Case B.

In general, in MILP type problems, an initial point does not notably improve the execution time. In fact, in many cases, what is gained will be similar if not less than what is invested in generating that initial point.

However, there are also cases where the input data makes especially difficult for the solver to find the first solution. In those cases, the execution time becomes much longer than usual, and having an initial point saves a lot of time. Therefore, an initial point makes the execution time of a problem more consistent, avoiding too long cases.

Chapter 8. CONCLUSIONS & FUTURE DEVELOPMENTS

This section summarizes the main conclusions and insights that have been drawn throughout the development of this thesis. In the same way, different possibilities are offered to further develop the methodology presented in this thesis.

With respect to the UC model, it can be stated that the model effectively solves the problem presented. Thus, the results obtained are consistent with what could be rationally intuited ex-ante. That is, the model aims to minimize operating costs and always tries to reduce deviations with respect to demand.

With respect to the NN model, the model manages to learn the patterns of the different generators in an adequate manner. That is, those generators that have behaviors as base technology are identified as such, and those that present more residual trends are also correctly recognized.

With respect to interaction, as it could be appreciated with the large-scale case study, the execution time registered with the proposed methodology is significantly lower than with the traditional optimization procedure from scratch. This means that there has been a considerable improvement in the optimization process.

Accordingly, the following is a specific evaluation of the different objectives presented in section 1.3. Objectives.

The first objective of the project was to develop a UC optimization model. In this sense, it can be considered that the stated objective has been fulfilled, since the developed optimization model is able to solve the presented UC problem in a satisfactory way.

Then, the next objective was aimed at automatizing the execution and storage of solution of the UC model for several sets of input data.

Again, it can be considered that the stated objective has been fulfilled, since the model presents total flexibility in the execution as explained in more depth in section 4.4. Several Weeks Model.

Thirdly, there was the objective of developing a Neural Network to learn from all the consecutive executions of the optimization problem.

As in the previous cases, this objective has been satisfied. After exploring and analyzing different possibilities during the development of the network model, it has been possible to build a model that allows estimating and generalizing patterns of the UC model.

Finally, the objective of using the Neural Network to estimate output variables of the UC problem has also been met.

Although it is true that finally it has been chosen to study only the prediction of one output variable (output power of thermal units), it is also true that it has been possible to establish an interaction and coordination of both models. Therefore, this objective is also considered as satisfied.

In other words, in general terms, it can be confirmed that this thesis fulfills the objectives proposed at the beginning of its development.

Some possibilities for further research related with this thesis are stated below.

To begin with, the first option for further research on the project could be the development of a more complex UC model. To this end, there are several possibilities that would make it possible to generate this new approach to the model.

For example, a possibility would be to introduce hydro technology into the system. This would provide much greater workability and would introduce new constraints into the model, which would also adapt the model developed in this thesis to an even more realistic model.

In addition, this development of a more extensive and complex model would imply that the improvement when implementing the methodology of this thesis in this model would be significant and the execution times would be greatly reduced.

In fact, this would certainly be in line with the motivation of this project, by allowing to run realistic models quickly or to obtain useful information in a simple way avoiding having to resort to a long execution of the optimization models.

On the other hand, another promising possibility would be the use of advanced optimization techniques, such as Bayesian optimization, to improve the neural network architecture and hyperparameters. That is, the implementation of techniques using optimization algorithms that allow finding the optimal parameters of the network.

The possibility of using other libraries to develop the neural network model could also be explored. And independently of the library, it would be possible to create functions more adapted to the problem to be generalized.

Alternatively, another possibility would be to use a neural network with a different approach from what is explained through this thesis. This could be the estimation of the commitment, start-up and shut-down decisions of the thermal units, rather than their power output. That is, in this case, instead of a regression model, it would be a classification model, working with binary variables. Due to that the binary variables are the most complex part of the model, having a good estimation of them could be useful.

However, the development of this type of model is likely to require a larger amount of data than has been considered in this thesis. That is, a much larger historical data set would be necessary for the model to be able to generalize the patterns of these new variables.

Finally, it must be concluded that this project, although it still offers possibilities for future developments, can be really promising and useful when it comes to obtain information about the dispatch of the units or to improve the classical process of optimization of the UC model.

Although perhaps at a premature stage, the feasibility of the methodology employed has been demonstrated, and future opportunities of research may prove that the development and results of this thesis may prove to be of great use in many areas other than electric power systems.

Chapter 9. REFERENCES

[1] Gal, D.; Mannor, S.. "Reinforcement learning for the unit commitment problem". 2015 IEEE Eindhoven PowerTech Eindhoven, Netherlands. Jul. 2015.

[2] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015).

[3] Zendehdel, N.; Karimpour, A.; Oloomi, M.. "Optimal unit commitment using equivalent linear minimum up and down time constraints". 2008 IEEE 2nd International Power and Energy Conference, Johor Bahru, Malaysia, Dec. 2008.

[4] Pan, X.; Zhao, T.; Chen, M; Zhao, T.. "DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow". IEEE Transactions on Power Systems (Early Access), University of Hong Kong, Sep. 2020.

[5] Yalcinoz, T.; Short, M.J.; Cory, B.J.. "Application of neural networks to unit commitment". IEEE Africon'99, South Africa, Oct. 1999.

[6] Zareian, M.; Hosseini, M. M.; Rashidinejad, M.; Fadaeinedjad, R.. "Solution to the unit commitment problem using an artificial neural network". Turkish Journal of Electrical Engineering and Computer Sciences, Oct. 2011.

[7] Gao, W.; Tang, N.; Mu, X.. "An Algorithm for Unit Commitment Based on Hopfield Neural Network". 2008 Fourth International Conference on Natural Computation, Jinan, China, Oct. 2008.

[8] De Otaola Arca, P.; Domínguez Barbero, D. "Modelo de Unit Commitment para maximización de beneficio partiendo de predicción de precios". https://gitlab001.iit.comillas.edu/pdeotaola/Ejemplo_Optimizacion_Python-Pyomo

[9] Abdou, I.; Tkiouat, M. "Unit Commitment Problem in Electrical Power System: A Literature Review". International Journal of Electrical and Computer Engineering (IJECE), Jun 2018.

[10] H. Dai, et al. "A literature review of stochastic programming and unit commitment", Journal of Power and Energy Engineering, vol. 3, no. 4, pp. 206, 2015.

[11] H. Y. Yamin. "Review on methods of generation scheduling in electric power systems"- Electric Power Systems Research, vol. 69, no. 2, pp. 227-248, 2004.

[12] B. Saravanan, et al. "A solution to the unit commitment problem--a review", Frontiers in Energy, vol. 7, no. 2, pp. 223, 2013.

[13] L. Wu and M. Shahidehpour. "Security-Constrained Unit Commitment with Uncertainties", Power Grid Operation in a Market Environment: Economic Efficiency and Risk Mitigation, pp. 115-168, 2016.

[14] Sheble, G and Fahd, G. "UNIT COMMITMENT LITERATURE SYNOPSIS", IEEE Transactions on Power Systems, Vol. 9, No.1, Feb 1994.

[15] Lui, X. "A new method of solving the unit commitment problem", IEE Power and Energy Society General Meeting (PES), 2013.

[16] Gómez Ramos, E and Venegas-Martínez, F. "A Review of Artificial Neural Networks: How Well Do They Perform in Forecasting Time Series?", Escuela Superior de Economía, Instrituto Politécnico Nacional, Mexico DF, Jun. 2013.

[17] Ramchoun, H; Amine Janati Idrissi, M; Ghanou, Y; Ettaouil, M. "Multilayer Perceptron: Architecture, Optimization and Training", International Journal of Interactive Multimedia and Artificial Intelligence, Vol.4, Nº1, Jan. 2016.

[18] T. Yalcinoz, M.J. Short and B.J. Cory, 'Application of neural networks to unit commitment', IEEE Africon'99, pp. 649-654, 28 Sept. – 1 Oct. 1999, South Africa.

[19] Sasaki H., Watanabe M., et al.,"A solution method of unit by artificial neural networks", IEEE Trans. on Power Systems, Vol. 7, No. 3, pp. 974-981, Aug. 1992.

[20] Gee A.H., Aiyer S.V.B. and Prager R.W., "An analytical framework for optimizing neural networks", Neural Networks, Vol. 6, No. 1, pp. 79-97, 1993.

[21] Yalcinoz T. and Short M.J., 'Large-scale economic dispatch using an improved Hopfield neural network', IEE Proc. Gener. Transm. Distrib., Vol. 144, No. 2, pp. 181-185, March 1997.

[22] R. Nayak, J.D Sharma, "A Hybrid Neural Network and Simulated Annealing Approach to the Unit Commitment Problem", Computers & Electrical Engineering, Vol. 26, Iss.6, pp. 461-477, ISSN 0045-7906, August 2000.

[23] Javier García González, "Short Term Models: Weekly Planning", Institute for Research in Technology (IIT), ICAI School of Engineering, Comillas Pontificial University, Madrid, Spain.

[24] Javier García González, "Weekly Unit Commitment Model (UC)", Institute for Research in Technology (IIT), ICAI School of Engineering, Comillas Pontificial University, Madrid, Spain

ANNEX I – CODING

In this Annex, it is stated a review of the main coding scripts that were used for the development of this master thesis.

There are three main coding scripts and also some files that contain different functions that were used in the main scripts. The three main scripts are: i) UC_model.py ii) NeuralNetwork.py and iii) Interaction.py

i) UC_model.py is the main script for executing the UC model for several consecutive weeks. This script calls the file ModeloOptimizacion.py, where the UC model is defined, and utils.py, where some useful functions (data reading, csv creation…) can be found.

ii) NeuralNetwork.py is the main script for training the NN and making estimations. This scripts also calls the file utils.py, where some other useful functions (data processing,…) can be found.

iii) Interaction is the main script for the feasibilization model and for executing the UC model using ANN estimations as starting point. This script calls the file ModeloOptimizacion.py, where the UC model and feasilibization model are defined, and utils.py, where some useful functions can be found.

Finally, all the coding scripts, functions and data used in this model can be found in https://github.com/pdeotaola/TFM_ManuelFlorezMontes, which is the online repository that has been used for development of this thesis.