



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Máster en Ingeniería Industrial

MASTER EN INGENIERÍA INDUSTRIAL

**ESCUELA DE INGENIERIAS INDUSTRIALES
UNIVERSIDAD DE VALLADOLID**

TRABAJO FIN DE MASTER

**DESARROLLO CON MATLAB DE UNA RED
NEURONAL PARA ESTIMAR LA DEMANDA DE
ENERGÍA ELÉCTRICA**

Autor: Jesús San Miguel Salas

Tutor: Daniel Moríñigo Sotelo

Cotutor: Fernando Espí García



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Máster en Ingeniería Industrial

MASTER EN INGENIERÍA INDUSTRIAL

**ESCUELA DE INGENIERIAS INDUSTRIALES
UNIVERSIDAD DE VALLADOLID**

TRABAJO FIN DE MASTER

**DESARROLLO CON MATLAB DE UNA RED
NEURONAL PARA ESTIMAR LA DEMANDA DE
ENERGÍA ELÉCTRICA**

Autor: Jesús San Miguel Salas

Tutor: Daniel Moríñigo Sotelo

Cotutor: Fernando Espí García

Resumen.

En el presente proyecto se hace una predicción del consumo eléctrico de un hospital a través de una red neuronal perceptrón multicapa que deberá ser entrenada mediante una serie de datos como la temperatura o la potencia consumida. Estos datos deberán ser tratados antes de que puedan ser usados por la red neuronal. Se estudiarán diferentes formas de suavizado para las curvas de potencia y se experimentará con las variables de entrada a la red neuronal esperando encontrar una solución óptima que permita predecir con la precisión requerida.

Abstract.

The current project carries on an hospital consumption prediction using a multilayer perceptron neuronal network. The Neuronal Network has to be trained using data such as temperature or power, this last kind of data has to be treated before the neuronal network uses it. This project shows different forms of treating the power data before it is used for the predictions. It will also be experimented with the number of inputs looking for the best way to achieve the required accuracy.

1. MOTIVACIÓN Y OBJETIVOS DEL TRABAJO FIN DE MASTER	1
2. INTRODUCCIÓN	3
2.1. IMPORTANCIA DE LA PREDICCIÓN DE LA DEMANDA Y CONSUMO ELÉCTRICO	3
2.2. PREDICIONES A LARGO Y A CORTO PLAZO	4
2.2.1. PREDICIONES A LARGO Y MEDIO PLAZO	4
2.2.2. PREDICIONES A CORTO PLAZO	4
2.2.3. OTRA CLASIFICACIÓN	4
2.3. MÉTODOS PARA LA PREDICCIÓN	5
2.3.1. MÉTODOS BASADOS EN INTELIGENCIA ARTIFICIAL	5
2.3.2. MÉTODOS PARAMÉTRICOS	5
3. REDES NEURONALES	9
3.1. INTRODUCCIÓN A LAS REDES NEURONALES	9
3.2. REDES NEURONALES ARTIFICIALES	9
3.2.1. REDES NEURONALES SUPERVISADAS Y NO SUPERVISADAS	10
3.2.2. FUNCIONES BASE Y DE ACTIVACIÓN	11
3.2.3. TAMAÑO DE LAS REDES NEURONALES	12
3.3. PERCEPTRÓN MULTICAPA	12
3.3.1. ARQUITECTURA	13
3.3.2. ALGORITMO DE APRENDIZAJE BACKPROPAGATION	13
3.3.3. APLICACIONES DE LA RED NEURONAL PERCEPTRÓN MULTICAPA	16
4. ORGANIZACIÓN DE LOS DATOS.	19
4.1. CODIFICACIÓN DE LOS DATOS DE POTENCIA.	21
4.1.1. SUAVIZADO DE LA CURVA DE POTENCIA CONSUMIDA MEDIANTE SPLINES CÚBICOS.	22
4.1.2. SUAVIZADO DE LA CURVA DE POTENCIA CONSUMIDA MEDIANTE EL FILTRO DE SAVITZY Y GOLAY.	26
4.1.3. ESCALADO DE LOS VALORES DE POTENCIA	30
4.2. CODIFICACIÓN DE LOS DATOS DE TEMPERATURA.	31
4.3. CODIFICACIÓN DE LOS DIAS LABORABLES Y DIAS NO LABORABLES.	31
4.4. CODIFICACIÓN DE LA HORA.	32
4.5. ORGANIZACIÓN DE LOS DATOS PARA EL ENTRENAMIENTO DE LA RED NEURONAL.	32
4.5.1. MÉTODO MÁXIMO Y MÍNIMO DE TEMPERATURA.	32
4.5.2. TARGETS.	33
5. DISEÑO DE LA RED NEURONAL.	35
5.1. NÚMERO DE CAPAS	35
5.2. NÚMERO DE NEURONAS	35
5.3. ALGORITMO DE ENTRENAMIENTO	36
5.3.1. SELECCIÓN DE DATOS EN LOS DISTINTOS CONJUNTOS DE ENTRENAMIENTO	37
5.3.2. VENTAJAS E INCONVENIENTES DEL USO DE REDES NEURONALES ARTIFICIALES.	38

6. CASO PRÁCTICO	41
6.1. CARACTERÍSTICAS DE LA RED NEURONAL DE MATLAB	41
6.3. ELECCIÓN DE LOS DATOS DE ENTRENAMIENTO.	44
6.4. SIMULACIONES.	49
6.4.1. DIFERENCIAS ENTRE EL SUAVIZADO DE SPLINES Y EL SUAVIZADO MEDIANTE EL FILTRO DE GOLAY-SAVITZKY.	49
6.5. SIMULACIONES CON SUAVIZADO DE SPLINES Y CON SUAVIZADO MEDIANTE EL FILTRADO DE SAVITZKY GOLAY.	57
6.6. RESULTADOS GOLAY -SPLINES	59
6.6.1. OTOÑO.	60
6.6.2. INVIERNO	63
6.6.3. PRIMAVERA	66
6.6.4. VERANO	69
6.7. CONCLUSIONES	72
6.8. IMPORTANCIA DE LAS VARIABLES DE POTENCIA	72
6.8.1. OTOÑO	75
6.8.2. INVIERNO	77
6.8.3. PRIMAVERA	79
6.8.4. VERANO	81
6.8.5. OTRAS SIMULACIONES	83
6.8.6. CONCLUSIONES	87
7. RESUMEN RESULTADOS CONCLUSIONES Y PROPUESTAS DE CONTINUACIÓN	89
7.1. RESUMEN RESULTADOS	89
7.2. CONCLUSIONES	92
7.3. LINEAS FUTURAS	92
8. COSTES DERIVADOS DEL PROYECTO	95
8.1. COSTES DIRECTOS	95
8.1.1. CALCULO DE COSTE TEMPORAL	96
8.1.2. COSTE DE MANO DE OBRA DIRECTA	97
8.1.3. COSTES MATERIAL	99
8.1.4. GASTOS DIRECTOS.	101
8.2. COSTES INDIRECTOS.	101
8.3. COSTES TOTALES	101
9. BIBLIOGRAFÍA	102

1. MOTIVACIÓN Y OBJETIVOS DEL TRABAJO FIN DE MASTER

La fuerte dependencia energética, la presión social y la toma de conciencia por parte de los gobiernos en la lucha contra el cambio climático, han dado lugar a que se adopten marcos regulatorios que favorecen el desarrollo tanto de recursos energéticos renovables, más limpios y sostenibles como de técnicas que ayuden a un ahorro energético.

La estimación del consumo eléctrico va a cobrar mucha importancia en los próximos años, el aumento de la energía consumida y la disminución de los combustibles fósiles junto con el desarrollo de las energías renovables y la economía sostenible hacen que un buen rendimiento sea aún más importante de lo que es ahora

Una buena predicción de demanda eléctrica ayudará a que todos los medios productores de energía eléctrica generen exactamente la energía necesaria para confort y necesidad de los usuarios de la electricidad tanto a nivel pequeño (como puede ser una casa), como a gran nivel (una industria).

Con una buena predicción se logra un gran equilibrio entre la energía generada y la energía consumida. Si no se lograra este equilibrio se tendrían problemas o bien por la menor generación de energía, lo cual conlleva cortes de suministro o bien por el desperdicio de energía eléctrica, ya que esta no se puede almacenar en grandes cantidades ni durante un periodo de tiempo elevado. Estas premisas hablan de lo importante que es desde el punto de vista de un país una buena predicción.

La predicción que se realiza en este proyecto se lleva a cabo desde el punto de vista de un hospital, que está interesado en conocer su futuro consumo eléctrico debido a:

- Dimensionado de instalaciones, como puede ser todos los motores productores de energía eléctrica de emergencia.
- Planificado de las distintas tareas a realizar en el hospital.
- Cálculo del coste de la energía consumida.

En este proyecto, se busca realizar una predicción lo más exacta posible con el material disponible, así como realizar un estudio de la dependencia de la red neuronal respecto las variables de potencia consumida en el pasado, es decir, si esta potencia consumida en el pasado puede ayudar a la red neuronal a predecir mejor y por lo tanto si hay que tenerla en cuenta o no en futuros proyectos.

Así mismo se intentará justificar los resultados de predicción ofrecidos por la red neuronal y el motivo por el que algunas predicciones serán mejores que otras, en aquellas predicciones que se consideren erróneas, se intentará aportar alguna solución que pueda ser aplicada en trabajos venideros.

El fin último de este trabajo consiste en seguir aportando información relevante que puede ayudar a los siguientes proyectos a realizar mejores predicciones como la información que aportaron los proyectos anteriores realizados en el departamento.

2. INTRODUCCIÓN

2.1. IMPORTANCIA DE LA PREDICCIÓN DE LA DEMANDA Y CONSUMO ELÉCTRICO

La eficiencia energética de los edificios representa uno de los aspectos clave en las políticas internacionales de hoy en día. La reducción de emisiones y el cumplimiento del protocolo de Kioto se están convirtiendo en un tema de trabajo fundamental para los ingenieros y técnicos que trabajan en el sector de gestión energética. Una gestión de la energía óptima necesita un mejor conocimiento de las incertidumbres introducidas por la demanda y por la generación de energía por ello el desarrollo de métodos de predicción de consumo y demanda de energía está tomando mucha importancia debido a la gran utilidad que pueden tener para desarrollar una gestión energética mejorada.

Los grandes edificios de nuestras ciudades representan uno de los principales consumidores de energía a gran escala. Se estima que la energía demandada por estos edificios en todo el mundo representa un 32% de la energía total, en Europa estos son responsables del 36% de las emisiones de CO₂(International Energy Agency, 2014).La necesidad de lograr en Europa el objetivo de mejora de la eficiencia energética en un 20% así como la reducción de los gases de efecto invernadero de un 20% en referencia a los niveles de 1990 hace necesario marcarse unas metas para la reducción de consumo energético e impacto ambiental de estos grandes edificios.

Un hospital puede ser definido como una organización compleja bajo un punto de vista, funcional y tecnológico. Un hospital moderno, puede ser comparado con una planta industrial por la multidisciplinariedad de sus funciones y tareas. La energía eléctrica sirve principalmente para dar servicio a la instalaciones del hospital así que tiene que ser calculada y gestionada teniendo en cuenta tanto criterios económicos como funcionales. En los últimos años los problemas relacionados con el consumo de energía han llegado a tomar una considerable importancia. La adquisición y el uso adecuado de la energía eléctrica son fundamentales en cualquier estructura compleja que quiera optimizar la gestión de la energía. Por eso los métodos de predicción de consumo para instalaciones como las de un hospital han tomado tanta importancia.

La predicción de la demanda y consumo energético es una función genérica de los sistemas de gestión energética de edificios, que representa un problema común y un factor clave en los sistemas de distribución eléctrica. Muchos estudios han sido desarrollados para la predicción a corto plazo ayudando de esta manera a mejorar la gestión energética de los edificios.

2.2. PREDICIONES A LARGO Y A CORTO PLAZO

2.2.1. PREDICIONES A LARGO Y MEDIO PLAZO

Una predicción precisa de la demanda eléctrica a medio-largo plazo juega un papel fundamental en la planificación de sistemas de potencia eléctrica. Estas **predicciones a largo plazo** se encargan de la predicción de la demanda con un tiempo suficiente de antelación como para poder efectuar una planificación energética a medio-largo plazo, afectando de manera activa a la planificación para el desarrollo de nuevas instalaciones de generación, la compra de unidades de generación y el desarrollo de la red de transporte y distribución.

La exactitud de la predicción a medio-largo plazo tiene un impacto significativo en el desarrollo de la planificación de los sistemas de generación y distribución. Una sobreestimación de la demanda conlleva en un incremento sustancial de la inversión para la construcción de instalaciones con excesiva potencia, en cambio, una subestimación implica el descontento de los consumidores (por cortes de suministro).

El rango de tiempo para las **predicciones a medio y largo plazo** va desde unas pocas semanas hasta unos cuantos años, por desgracia, es difícil tener una buena exactitud en las predicciones con periodos de tanto tiempo.

2.2.2. PREDICIONES A CORTO PLAZO

La mayoría de los métodos de predicción, están dedicados a la **predicción a corto plazo**, desde unos pocos minutos hasta 24 horas. Las predicciones a muy corto plazo, de minuto en minuto, se utilizan para una asignación satisfactoria de cargas que beneficie a las unidades de generación. La predicción a corto plazo que oscila entre una hora y un día es valiosa para la gestión de las unidades de consumo energético, esta previsión permitirá planificar y controlar la puesta en marcha y el apagado de las unidades de generación, a su vez permite conocer si estas unidades de generación son capaces de asumir las variaciones de demanda provenientes de la unidad consumidora.

2.2.3. OTRA CLASIFICACIÓN

A parte del tipo de predicciones relacionadas con el periodo de tiempo predicho también podemos distinguir las predicciones según se quiera una predicción exacta de la potencia consumida en cada instante de tiempo definido **predicción horaria**, o solo interesen los puntos críticos de consumo de potencia, es decir, unos puntos clave que den toda la información necesaria, este tipo de predicción es conocida como **predicción por picos**.

2.3. MÉTODOS PARA LA PREDICCIÓN

Los métodos de predicción se dividen en dos grandes grupos, métodos basados en inteligencia artificial y métodos paramétricos.

2.3.1. MÉTODOS BASADOS EN INTELIGENCIA ARTIFICIAL

- Métodos basados en redes neuronales.

Los métodos basados en redes neuronales no requieren un modelo físico-matemático para representar el consumo de energía eléctrica, pero sí unos parámetros fundamentales, que son: el tipo de arquitectura, las entradas, las funciones de transferencia, el algoritmo de entrenamiento y el número de neuronas.

Los métodos basados en redes neuronales usan diferentes algoritmos de aprendizaje, que son descritos, como procesos que consisten en el ajuste de pesos específicos para minimizar la función error entre la que debiera ser la salida de la red, que es un dato conocido y la salida producida por los pesos específicos actuales

- Métodos basados en lógica difusa.

Este tipo de técnica permite considerar la imprecisión de los datos a través de reglas de extracción de variables de entrada y de modelos estadísticos. En general, esta técnica es usada para el tratamiento de datos combinándose con otro método para lograr una predicción completa y satisfactoria (Chen, 2012).

2.3.2. MÉTODOS PARAMÉTRICOS

Se basan en el hecho de que la demanda se puede modelar matemáticamente. Los parámetros del modelo son estimados usando técnicas estadísticas sobre los datos históricos de la demanda y sobre los factores que afectan a estos datos.

(Yan, 2014) Utiliza procesos Gaussianos, en los cuales asume que todas las variables siguen distribuciones Gaussianas. Todas las variables disponibles son utilizadas para poder predecir de una manera más exacta el consumo de energía. Una de las ventajas de la predicción mediante procesos Gaussianos consiste en que además de ofrecer la estimación más probable, también describe la distribución de las nuevas predicciones, hecho que puede llegar a ser muy beneficioso en el desarrollo de estrategias de control. La estimación mediante procesos Gaussianos garantiza una estimación robusta incluso cuando el número de datos es limitado o esta desequilibrado.

- Métodos de regresión

- Lineal

Los métodos de regresión también son usados para la predicción de consumo y demanda eléctrica en sistemas sencillos. Esta regresión

consiste en la predicción mediante una ecuación formada por variables independientes y coeficientes que se ajustan al modelo de consumo de electricidad del sistema. La ecuación proporciona una salida con el valor de la potencia eléctrica predicha (Kaytez, 2015).

- Complejo

Relacionados con los métodos de regresión, también se han desarrollado otros modelos de predicción como los *Support Vector Machine*(SVM) o *Least Squares Support Vector Machine* (LS-SVM)(Kaytez, 2015) donde se utiliza la función de pérdida del método y un conjunto de restricciones ecuacionales “LS-SVM” para hallar la solución a través de una serie lineal de ecuaciones. El método SVM se basa en resolver un problema cuadrático, este método necesita normalmente una alta potencia computacional.

- *Métodos de predicción de series temporales*

Una serie temporal es un conjunto de observaciones ordenadas en el tiempo, que pueden representar la evolución de una variable (económica, física, etc.) a lo largo de él. El objetivo del análisis de una serie temporal es el conocimiento de su patrón de comportamiento, para así prever su evolución futura, suponiendo que las condiciones no variarán. Dado que no se trata de fenómenos deterministas, sino sujetos a una aleatoriedad, el estudio del comportamiento pasado ayuda a inferir la estructura que permita predecir su comportamiento futuro, pero es necesaria una gran cautela en la previsión debido a la inestabilidad del modelo.

- Modelos univariados.

El análisis univariado de series temporales mediante los modelos ARIMA permite estimar adecuadamente un conjunto de coeficientes lineales de la misma serie atrasada varias unidades temporales, que pueden utilizarse para interpretar datos extraídos de distintos procesos estocásticos para efectuar predicciones válidas, sobre todo a corto plazo

- Modelos causales.

El modelado causal temporal intenta descubrir relaciones causales clave en datos de series temporales. En el modelado causal temporal el procedimiento crea un modelo de serie temporal autorregresivo para cada objetivo y solo incluye las entradas que tienen una relación causal con el objetivo.

- Métodos de modelado dinámico con estimación de parámetros

Las representaciones en el espacio de estados también tienen su aplicación para la predicción de la potencia eléctrica demandada (Yanxia-lu, 2012). Utiliza un espacio de estados gaussiano para poder predecir el consumo de energía eléctrica de la hora siguiente.

Algunos autores también han desarrollado métodos heurísticos para las predicciones de consumo de energía como (Escrivá, 2014) donde dividen la instalación eléctrica total en

subinstalaciones con consumo eléctrico independiente; esas subinstalaciones tienen un consumo dependiente de diferentes variables como la temperatura o humedad entre otras. El método consiste en elegir los cuatro días más parecidos de cada subinstalación y a partir de los consumos de esos 4 días, mediante métodos heurísticos, hallar la potencia que se consumirá a corto plazo.

3. REDES NEURONALES

3.1. INTRODUCCIÓN A LAS REDES NEURONALES

Las Redes Neuronales (NN o *Neural Networks*) se crearon para simular los sistemas nerviosos biológicos, constituidos por neuronas y nodos que interconectan las neuronas. El primer modelo de red neuronal fue propuesto en 1943 por McCulloch y Pitts. Era un modelo de neuronas binarias, donde cada neurona podía adoptar dos estados. Este modelo fue la base para el desarrollo de diferentes modelos a posteriori. Una primera clasificación de redes neuronales podría ser la siguiente.

- **Modelos basados en la Biología:** Son un conjunto de redes que simulan los sistemas neuronales biológicos y funciones como la vista o la audición.
- **Modelos artificiales aplicados:** Diseñados para resolver algún problema en concreto que no tiene necesariamente que ver con los sistemas biológicos.

3.2. REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales suele estar inspiradas en las redes neuronales biológicas pero con unas estructuras de conexión y unas funcionalidades diferentes, se puede decir que las NN (*Neural Networks*) tienen tres características principales.

- **Auto-Organización y Adaptabilidad:** utilizan algoritmos de aprendizaje adaptativo y de auto-organización ofreciendo mejores posibilidades de procesamiento robusto.
- **Procesado no Lineal:** aumenta la capacidad de la red para la clasificación de patrones, la aproximación de funciones, y el aumento de su inmunidad frente al ruido.
- **Procesado Paralelo:** normalmente se usa un gran número de nodos de procesamiento, con alto nivel de interconectividad.

Al elemento básico de computación se llama nodo o unidad Figura 3.1, el cuál recibe una entrada desde la salida de otros nodos o del exterior. Cada entrada posee un peso específico asociado “w”, que se aumentará o disminuirá en el proceso de aprendizaje. Cada nodo aplica una función dada f de la suma de las entradas ponderadas mediante los pesos:

$$y_i = \sum_j w_{ij} y_j \quad (3.1)$$

Existen dos fases en el modelado de una red neuronal.

- **Fase de entrenamiento:** se usa un conjunto de datos o patrones de entrenamiento para determinar los pesos (parámetros) que definen el modelo de red neuronal. Se calculan de manera iterativa, de acuerdo los valores de entrenamiento, con el objeto de minimizar el error cometido entre la salida obtenida por la red neuronal y la salida deseada.
- **Fase de Prueba:** en la fase anterior, el modelo puede que se ajuste demasiado a las particularidades presentes en los patrones de entrenamiento, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos. Esto se denomina sobreajuste. Para evitar este problema, es aconsejable utilizar un segundo grupo de datos diferentes a los de entrenamiento, el grupo de validación, que permita controlar el proceso de aprendizaje.

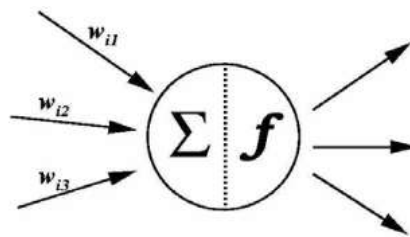


Figura 3.1 Nodo o unidad

Normalmente, los pesos óptimos se obtienen optimizando (minimizando) alguna función de energía. Por ejemplo, minimizar el error cuadrático medio entre el valor de salida y el valor real esperado.

3.2.1. REDES NEURONALES SUPERVISADAS Y NO SUPERVISADAS

Las redes neuronales se clasifican en función de sus métodos de entrenamientos, donde se pueden distinguir tres tipos de redes neuronales: **redes de pesos fijos**, en las cuales no existen ningún entrenamiento, **redes no supervisadas**, y **redes de entrenamiento supervisado**.

Reglas de entrenamiento Supervisado. Este tipo de red neuronal es la más popular. Los datos para el entrenamiento los constituyen varios patrones conocidos de entrada y salida, el entrenamiento será supervisado por un maestro que conoce la salida. La adaptación de pesos en la etapa ($m + 1$) se basa en la siguiente formula.

$$w_{ij}^{m+1} = w_{ij}^m + \Delta w_{ij}^m \quad (3.2)$$

Reglas de Entrenamiento No Supervisado. Los datos de entrenamiento consisten únicamente en patrones de entrada. La red aprende a adaptarse según las experiencias de los patrones de entrenamiento anteriores.

3.2.2. FUNCIONES BASE Y DE ACTIVACIÓN

Una red neuronal típica se puede caracterizar por la función de base y la función de activación. Cada nodo (unidad de proceso), suministra un valor y_j a su salida. Este valor se propaga a través de la red mediante conexiones unidireccionales hacia otros nodos de la red.

Asociada a cada conexión hay un peso sináptico denominado $\{w_{ij}\}$, que determina el efecto del nodo j -ésimo sobre el nodo i -ésimo. Las entradas al nodo i -ésimo que provienen de los otros nodos son acumuladas junto con el valor umbral θ_i , y se aplica la función base f , obteniendo u_i . La salida final y_i se obtiene aplicando la función de activación sobre u_i .

Función Base

La función base tiene dos formas típicas.

Función lineal de tipo hiperplano: El valor de red es una combinación lineal de las entradas.

$$u_i(w, x) = \sum_{j=1}^n w_{ij}x_j \quad (3.3)$$

Función radial de tipo hiperesférico: es una función de base de segundo orden no lineal. El valor de red representa la distancia a un determinado patrón de referencia.

$$u_i(w, x) = \sqrt{\sum_{j=1}^n (x_j - w_{ij})^2} \quad (3.4)$$

Función de Activación (Función de neurona)

El valor de red, expresado por la función de base, $u(w, x)$, se transforma mediante una función de activación no lineal. Las funciones de activación más comunes son la función sigmoideal y gaussiana:

Función sigmoideal

$$f(u_i) = \frac{1}{1 + e^{-\frac{u_i}{\sigma^2}}} \quad (3.5)$$

Función gaussiana

$$f(u_i) = ce^{-\frac{u_i^2}{\sigma^2}} \quad (3.6)$$

Estructuras de conexión de atrás hacia adelante

Una red neuronal se determina por las neuronas y la matriz de pesos. Hay tres tipos de capas de neuronas:

- la capa de entrada.
- la capa oculta.

- la capa de salida.

Las capas de neuronas se interconectan mediante una red de pesos que puede ser de los siguientes tipos

- Hacia adelante:
- Hacia atrás.
- Lateral.
- De retardo.

Conexiones hacia delante: los valores de las neuronas de una capa inferior son propagados hacia las neuronas de la capa superior

Conexiones hacia atrás: estas conexiones llevan los valores de las neuronas de una capa superior a otras de la capa inferior.

Conexiones laterales: a la neurona de salida que da el valor más alto se le asigna el valor total (por ejemplo, 1), mientras que a todas las demás se le da un valor de 0

Conexiones con retardo: los elementos de retardo se incorporan en las conexiones para implementar modelos dinámicos y temporales, es decir, modelos que precisan de memoria.

3.2.3. TAMAÑO DE LAS REDES NEURONALES

En una red multicapa de propagación hacia delante, puede haber una o más capas ocultas entre las capas de entrada y salida. El tamaño de las redes depende del número de capas y del número de neuronas ocultas por capa. El número de unidades ocultas está directamente relacionado con las capacidades de la red. Para que el comportamiento de la red sea correcto, se tiene que determinar apropiadamente el número de neuronas de la capa oculta.

3.3. PERCEPTRÓN MULTICAPA

Es una red neuronal formada por varias capas; actúa como un aproximador universal de funciones mediante el algoritmo de aprendizaje *backpropagation*, obligando a la red neuronal a contener al menos una capa oculta, con suficientes unidades no lineales. Es capaz de aproximar cualquier tipo de función o relación continua entre un grupo de variables de entrada y salida.

Por eso se considera a las redes neuronales tipo perceptrón multicapa una herramienta de propósito general, no lineal y flexible.

Rumelhart et al. (1986) presentaron un algoritmo de aprendizaje conocido como "Regla Delta Generalizada" por el que las redes perceptrón multicapa podían aprender la relación existente entre un conjunto de patrones de entrada y sus salidas correspondientes.

El también conocido como algoritmo “*backpropagation error*” (propagación del error hacia atrás) tiene la capacidad de dotar a la red de generalización de tal modo que el sistema obtiene una salida correcta para un conjunto de datos de entrada que no se habían usado antes

3.3.1. ARQUITECTURA

Las redes Perceptrón multicapa están formadas por

- **Capa de entrada**
- **Capa de salida**
- **Una o más capas ocultas**

La mayoría de problemas se podrán resolver utilizando una única capa oculta. En la Figura 3.2 se puede observar el esquema de una red neuronal perceptrón multicapa.

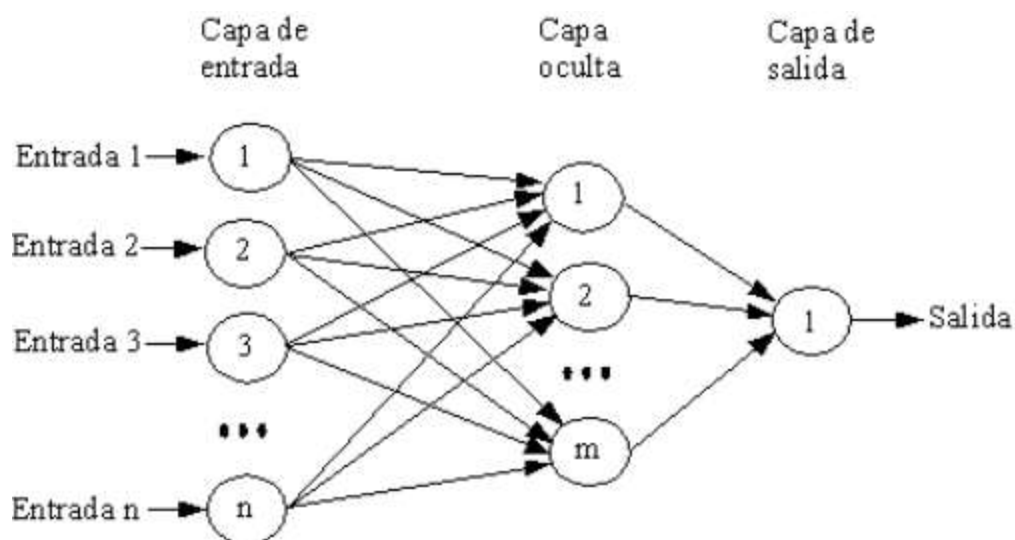


Figura 3.2. Arquitectura Perceptrón multicapa

Las conexiones de las neuronas son siempre hacia adelante, no hay conexiones laterales, ni conexiones hacia atrás, la información siempre se dirige desde la entrada hacia la salida.

Se considera w_{ji} como el peso de conexión entre la neurona de entrada i y la neurona oculta j . v_{kj} es el peso de conexión entre la neurona oculta j y la neurona de salida k .

3.3.2. ALGORITMO DE APRENDIZAJE BACKPROPAGATION

Se realiza mediante dos pasos. Primeramente se considera una **etapa de funcionamiento** donde se presenta un patrón de entrada y éste se transmite a través de las sucesivas capas de neuronas hasta obtener una salida. Después se procede a ejecutar una **etapa de entrenamiento o aprendizaje** donde se modifican los pesos de la red hasta que la salida deseada por el usuario coincida con la salida obtenida por la red en todos los patrones de entrada.

Etapa de funcionamiento

Cuando se presenta un patrón p de entrada $X^p: x_1^p \dots x_i^p \dots x_N^p$, éste se transmite a través de los pesos w_{ji} desde la capa de entrada hacia la capa oculta. Las neuronas de esta capa intermedia transforman las señales recibidas mediante la aplicación de una función de activación proporcionando, de este modo, un valor de salida. Este se transmite a través de los pesos v_{kj} hacia la capa de salida, donde aplicando la misma operación que en el caso anterior, las neuronas de esta última capa proporcionan la salida de la red. Este proceso se puede representar matemáticamente de la siguiente forma:

La entrada total que recibe una neurona oculta j , net_j^p es:

$$net_j^p = \sum_{i=1}^n w_{ij} x_i^p + \theta_j \quad (3.7)$$

En la cual θ_j es el umbral de la neurona que se considera como un peso asociado a una neurona ficticia con valor de salida 1.

La salida de la neurona oculta j , y_j^p se obtiene aplicando una función sobre su entrada.

$$y_j^p = f(net_j^p) \quad (3.8)$$

De la misma forma la entrada que recibe una neurona de salida es:

$$net_k^p = \sum_{j=1}^H v_{kj} y_j^p + \theta_k \quad (3.9)$$

También se le aplica una función que genera una salida

$$y_k^p = f(net_k^p) \quad (3.10)$$

Etapa de entrenamiento

En la etapa de aprendizaje se busca un objetivo: hacer mínimo el error entre la salida producida y la salida deseada del grupo de entrenamiento. En las redes perceptrón multicapa se tiene un entrenamiento supervisado dirigido por un maestro. La función error a minimizar es la siguiente.

$$E^p = 0,5 * \sum_{K=1}^M (d_k^p - y_k^p)^2 \quad (3.11)$$

Donde d_k^p es la salida deseada para la neurona de salida k y patrón p

El error se podrá medir de la siguiente forma

$$E = \sum_{y=1}^P E^p \quad (3.12)$$

Aplicando la técnica del gradiente decreciente, se podrá saber hacia dónde ajustar el peso para ir minimizando el error, el error decrecerá más rápido hacia la dirección negativa del gradiente.

$$-\sum_{y=1}^P \frac{dE^p}{dw_{ji}} \quad (3.13)$$

El principal peligro es la convergencia en un mínimo local, sin embargo, a nivel práctico la forma de modificar los pesos iterativamente consiste en aplicar la regla de la cadena al gradiente añadiendo una tasa de aprendizaje η , así el incremento en una neurona de salida será:

$$\Delta v_{kj}(n+1) = \eta \sum_{y=1}^P \delta_k^p y_j^p \quad (3.14)$$

En la cual

$$\delta_k^p = (d_k^p - y_k^p) f'(net_k^p) \quad (3.15)$$

En una neurona oculta

$$\Delta w_{ji}(n+1) = \eta \sum_{y=1}^P \delta_j^p x_i^p \quad (3.16)$$

Donde

$$\delta_i^p = f'(net_j^p) \sum_{K=1}^M \delta_k^p v_{kj} \quad (3.17)$$

El error de una neurona oculta, viene determinado por la suma de los errores que se cometen en las k neuronas de salida que reciben como entrada la salida de esa neurona oculta j , de ahí que se conozca al algoritmo con el nombre de *backpropagation*.

Para la modificación de los pesos hay dos maneras:

- La actualización se realiza después de haber presentado todos los patrones
- Actualización tras la presentación de cada patrón de entrenamiento.

Para acelerar la convergencia Rumelhart et al. (1986) sugirieron añadir un factor conocido como momento que tiene en cuenta la dirección del incremento del paso anterior.

$$\Delta v_{kj}(n+1) = \left(\eta \sum_{y=1}^P \delta_k^p y_j^p \right) + \alpha \Delta v_{kj}(n) \quad (3.18)$$

3.3.3. APLICACIONES DE LA RED NEURONAL PERCEPTRÓN MULTICAPA

Las redes perceptrón multicapa se han utilizado para resolver dos tipos de problemas:

- **Problemas de predicción** Estimación de una variable de salida a partir de un conjunto de variables de entrada.
- **Problemas de clasificación** Asignación de patrones a una categoría

Selección de las variables relevantes y procesamiento de datos

Para obtener una aproximación funcional óptima, se deben elegir cuidadosamente las variables a emplear.

Incluir en el modelo las variables predictoras que realmente predigan la variable dependiente o de salida. Pero que a su vez no tengan relaciones entre sí.

Las variables deben seguir una distribución normal o uniforme, y el rango de posibles valores debe ser aproximadamente el mismo. Las variables de entrada y salida suelen acotarse en valores comprendidos entre 0 y 1 ó entre -1 y 1.

Si la variable es discreta, se utiliza la codificación *dummy*. Por ejemplo, la variable sexo podría codificarse como: 0 = hombre, 1 = mujer; estando representada por una única neurona. La variable nivel social podría codificarse como: 100 = bajo, 010 = medio, 001 = alto; estando representada por tres neuronas. Por su parte,

Si la variable es de naturaleza continua, ésta se representa mediante una sola neurona, como, por ejemplo, el CI de un sujeto.

Entrenamiento de la red neuronal

Los pesos iniciales deberán ser bajos y aleatorios.

El número de neuronas de la capa de entrada está predeterminado por el número de variables predictoras, así como el número de neuronas de salida está determinado por el número de variables que se quieran conocer.

En la capa oculta, se deben usar el mínimo número de neuronas con los que se pueda resolver el problema, para ello, se deberá evaluar de forma experimental el rendimiento de las diferentes arquitecturas con los grupos de validación.

El valor de la tasa de aprendizaje η controla el cambio de los pesos en cada iteración deberá ser el mayor posible, sin que este provoque grandes oscilaciones, suele tomar valores de entre 0,05 y 0,5.

El factor α que es el encargado de acelerar la convergencia, toma valores cercanos a 1 como 0,95.

Como funciones de activación, hay dos formas básicas:

- Función lineal

$$y = Kx + X \quad (3.19)$$

- Función sigmoïdal

$$y = \frac{1}{1 + e^{-x}} \quad (3.20)$$

Para aprovechar la capacidad de las redes neuronales de aprender relaciones complejas o no lineales entre variables, se recomienda la utilización de funciones no lineales al menos en las neuronas de la capa oculta.

Para la capa de salida, en tareas de clasificación las neuronas toman como función de activación la función sigmoïdal, mientras que en tareas de predicción, la función lineal.

Evaluación del modelo.

Una vez seleccionados el conjunto de parámetros que han obtenido el mejor rendimiento ante el conjunto de validación, se deberá evaluar la capacidad de generalización de la red a través de un tercer conjunto de datos, el conjunto **test**, para ello se utilizará la función cuadrática.

$$MC_{error} = \frac{\sum_{y=1}^P \sum_{k=1}^M (d_k^p - y_k^p)^2}{P \cdot M} \quad (3.21)$$

Interpretación de los pesos obtenidos

Se trata de interpretar los pesos de la red, el método más utilizado es el análisis de sensibilidad. Para ello se analiza el efecto producido en una salida por el cambio en una entrada. Normalmente se varía una única entrada permaneciendo el resto con unos valores fijos.

4. ORGANIZACIÓN DE LOS DATOS.

En este apartado se da comienzo a la explicación del caso práctico, antes de mostrar todas las simulaciones y conclusiones es necesario explicar los cambios que se han introducido a la forma inicial de los datos para que estos fueran interpretables por la toolbox de Matlab. Para comenzar se escribirá una pequeña explicación sobre los tipos de datos que se manejan en este proyecto.

Como datos de entrada a la red se pueden manejar dos tipos de datos, variables exógenas y variables inherentes al sistema. En este proyecto ambas serán fundamentales para poder realizar una buena predicción

- **Variables exógenas:** variables externas al sistema como pueden ser la temperatura, la humedad, la velocidad del aire, etc. Dentro de este grupo se pueden distinguir tres subcategorías
 - Variables climáticas: Solo se han de utilizar aquellas variables que afecten directamente al sistema. Además hay que tener en cuenta que estas variables pueden estar interrelacionadas por lo que realizar una predicción con varias variables que estén relacionadas entre sí puede perjudicar al sistema.
 - Variables de calendario: Permiten distinguir entre los distintos tipos de día existentes y su relación con el modelo. Así en el caso del consumo eléctrico del hospital, en un día laboral habrá más consumo que en un día no laboral, también permite distinguir entre las horas dentro del mismo día (el consumo nocturno es menor que el consumo diurno).
 - Variables económicas: Este tipo de variables tienen influencia en sistemas grandes y con la intención de predecir a largo plazo. No se suelen incluir en el modelo.
- **Variables inherentes al sistema:** Estas variables ayudan a predecir buscando información en condiciones similares que sucedieron en el pasado.

Según se puede encontrar en bibliografía, existen multitud de combinaciones de variables según el autor. Lo ideal es estudiar las relaciones existentes en los candidatos a variables de entrada para no incluir correlaciones que puedan afectar a los resultados. Además, el conocimiento del sistema a modelar, la experiencia en el tema, etc., dan una ventaja en la elección de las variables de entrada. (Lu, 1993) realiza un estudio previo estadístico para identificar qué instantes de la demanda anterior afectan a la previsión de la demanda actual.

En este sistema, inicialmente, se ha decidido utilizar como variables de entrada:

- Potencia consumida hace 1 semana
- Potencia consumida hace 1 día
- Potencia consumida hace 3 horas
- Potencia consumida hace 2 horas
- Potencia consumida hace 1 hora
- Distinción entre día laborable o festivo
- Hora del día
- Temperatura horaria o temperatura máxima y mínima del día

Se debe aclarar, que otro de los puntos a desarrollar en este proyecto, es la capacidad de predicción de la red neuronal sin alguna de las entradas anteriores, de tal modo que se medirá la importancia que tiene esa entrada y su necesidad o no para una correcta predicción.

Para la aplicación desarrollada en este proyecto ha hecho falta adecuar el conjunto de datos de entrada a una determinada forma que fuera interpretable por la *toolbox* en el entrenamiento de la red y en la predicción de la demanda eléctrica.

Antes de dar forma a los datos para que sean interpretables por la *toolbox* hay que asegurarse de que cumplen las siguientes generalidades, que garantizarán un buen funcionamiento de la red neuronal.

- Eliminación de la componente de la tendencia de la demanda: las redes neuronales no son capaces de modelar la tendencia a medio o largo plazo, por lo tanto, habría que modificar los datos para que esa tendencia no aparezca.
- Cambio de escala: En muchos casos, resulta conveniente normalizar las variables, para que tomen valores entre cero y uno. De esta forma, se pueden eliminar sesgos debidos a las unidades de medida. Además, las prestaciones de las redes neuronales aumentan considerablemente al trabajar con valores acotados entre cero y uno. El motivo de tener que realizar esta normalización es debido a las funciones utilizadas durante el proceso de entrenamiento, donde se aplicará una modificación del método del gradiente. Si las entradas no están normalizadas, se tendrán valores muy altos, donde las funciones de activación estarán “saturadas” y el gradiente será prácticamente cero, lo que dificulta el uso de estas técnicas.
- Codificado de los datos: Existirán en muchos casos variables categóricas, binarias, horas del día, etc. Dichas variables se codificarán de tal forma que sus valores estén acotados entre cero y uno como se ha comentado anteriormente

Filtrado: las técnicas de identificación suelen ser bastante sensibles a la aparición de anomalías en los datos. Estos datos atípicos pueden deberse a fallos en los instrumentos de medida, cortes en el suministro eléctrico, etc., que hacen que tengan una relevancia

nula en el proceso de identificación. Por otro lado, existen otro tipo de datos anormales debidos a días con un comportamiento anómalo (vacaciones, fiestas, etc.), que perjudicarían al modelo si se tuvieran en cuenta como días normales

4.1. CODIFICACIÓN DE LOS DATOS DE POTENCIA.

Inicialmente se disponen de 24 hojas Excel (una por mes) con los datos cuartos horarios de potencia consumida del hospital de León. Como ejemplo se muestra la potencia consumida en el mes de Abril de 2008 (Figura 4.1), al trabajar con datos de potencia cada cuarto de hora, se cuenta con un total de 2880 datos (30 días 24 horas, 4 datos/hora).

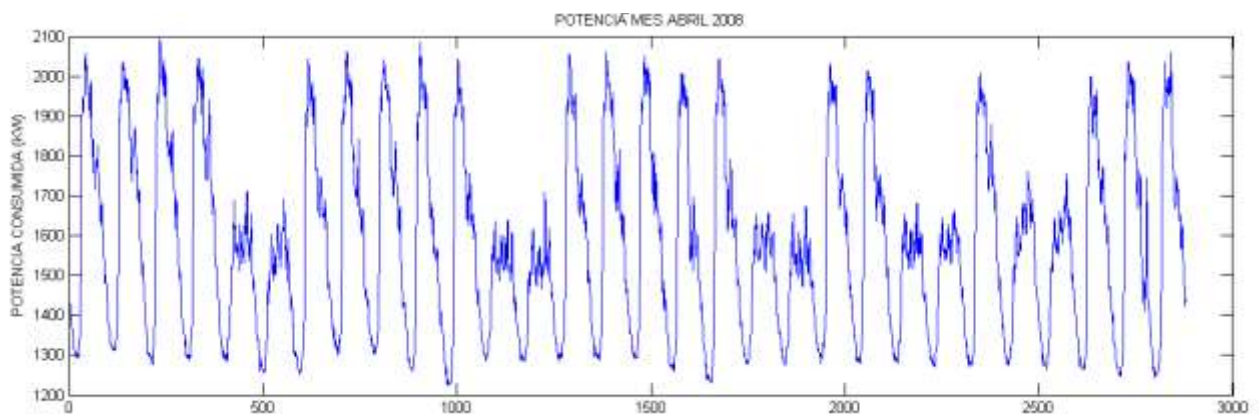


Figura 4.1 Potencia consumida Abril 2008 (datos cuarto-horarios)

La estimación que se necesita hacer, es una estimación horaria, por lo que por comodidad se ha decidido trabajar con datos horarios en vez de con datos cuarto horarios. El primer cambio a realizar es pasar los datos cuarto-horarios a datos horarios, para ello se hace una media de los 4 datos de potencia que hay por hora y el resultado se coge como dato de potencia consumida en esa hora. El resultado se muestra en la Figura 4.2, que representa exactamente el mismo mes que la Figura 4.1.

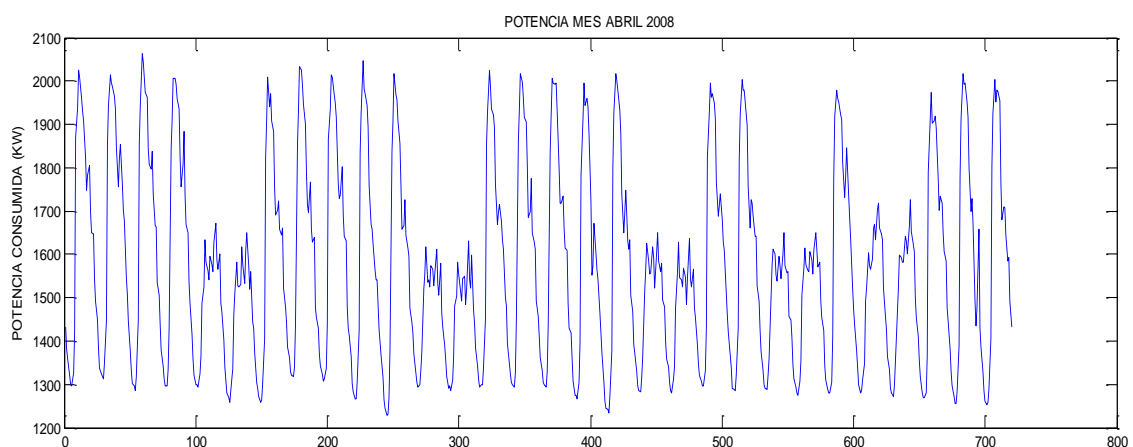


Figura 4.2 Potencia consumida mes de Abril

Para eliminar los puntos anómalos, que puedan ser debidos a fallos en los instrumentos de medida o sucesos puntuales poco corrientes, será necesario aplicar un filtrado o suavizado de los datos de potencia consumida. Además, con el suavizado de la curva se eliminarán también pequeñas fluctuaciones que no aporten información relevante al estudio, pero que, sin embargo, pueden producir un mayor tiempo de cálculo o incluso, una predicción errónea de la demanda. En definitiva, se quiere conseguir que los picos de potencia consumida que se pueden ver en las figuras anteriores, distorsionen de la menor forma posible el entrenamiento de la red neuronal y eliminar aquellos puntos anómalos que hagan fallar a la red neuronal.

En este proyecto, se trabajan con dos tipos de algoritmos para el suavizado de curvas. Se compararán los métodos y se determinará cuál es el mejor de estos algoritmos para el suavizado de este modelo. Los dos algoritmos utilizados son:

- Suavizado mediante Splines Cúbicos
- Suavizado mediante el filtro de Savitzky-Golay.

4.1.1. SUAVIZADO DE LA CURVA DE POTENCIA CONSUMIDA MEDIANTE SPLINES CÚBICOS.

4.1.1.1. Introducción

La interpolación mediante **splines lineales** (Figura 4.3) consiste en dividir el intervalo a interpolar en una serie de intervalos más pequeños llamados subintervalos, y en cada uno de estos subintervalos construir un polinomio que se aproxime a la forma original. Esta forma de aproximar por medio de funciones es conocida como aproximación polinómica fragmentaria, y consiste en unir una serie de puntos $\{(x_0, f(x_0)), \dots, (x_n, f(x_n))\}$ mediante una serie de polinomios de primer grado.

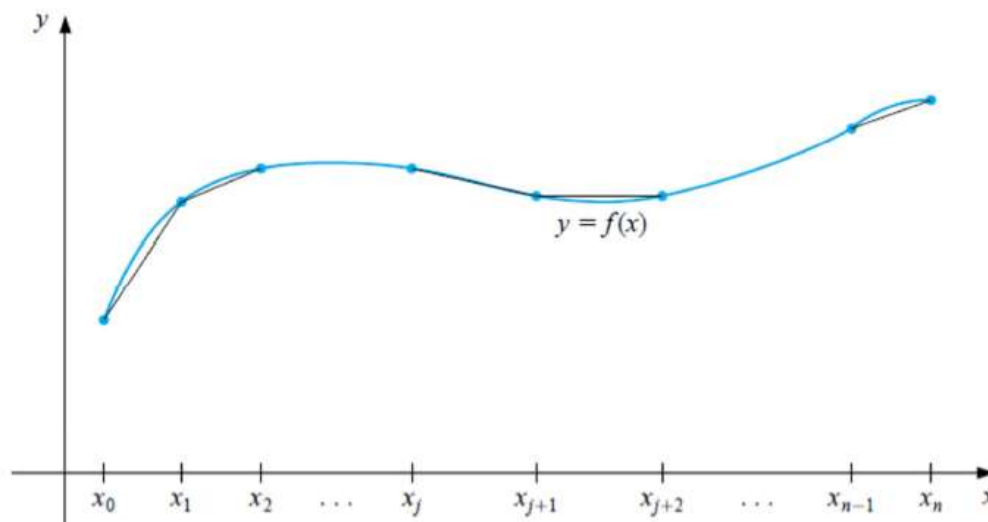


Figura 4.3 Interpolación mediante segmentos de recta

La aproximación por funciones lineales muestra una desventaja: no se tiene la seguridad de que haya diferenciabilidad en los extremos de los subintervalos, lo cual dentro de un contexto geométrico significa que la función suavizada tiene una forma brusca en los puntos iniciales y finales de cada intervalo. A menudo se requiere una forma suave en los extremos de cada subintervalo suavizado y por lo tanto la función que aproxima la curva inicial debe ser continuamente diferenciable.

La aproximación polinómica fragmentaria más común utiliza polinomios entre cada par consecutivo de nodos y recibe el nombre de **spline cúbico** (Figura 4.4). Un polinomio cúbico general contiene cuatro constantes; así pues, el procedimiento del spline cúbico ofrece suficiente flexibilidad para garantizar que la función resultado de la interpolación no sólo sea continuamente diferenciable en el intervalo, sino que además tenga una segunda derivada continua en el intervalo.

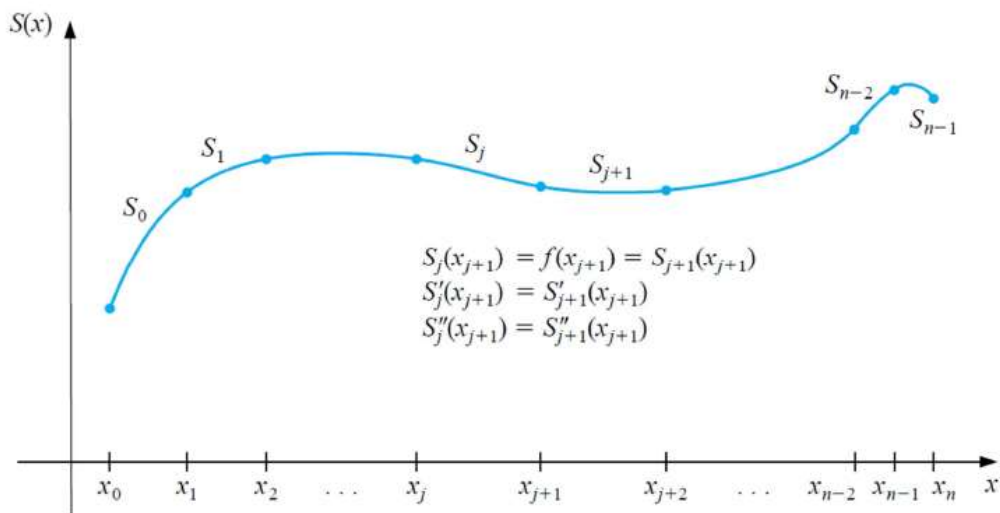


Figura 4.4 Interpolación mediante Splines Cúbicos

Definición. [Spline cúbico.]

Dado un conjunto de datos $\{x_1 = a, x_2, \dots, x_n, x_{n+1} = b\}$ del intervalo $[a,b]$, diremos que la función S_Δ es un spline cúbico asociado a Δ si cumple las siguientes condiciones:

1. La restricción S_i de S_Δ a cada intervalo $[x_i, x_{i+1}]$, para $i = 1, 2, \dots, n$ es un polinomio de grado no superior a tres
2. $S(x) \in C^2([a, b])$, es decir, S_Δ es una función continua, dos veces derivable y con derivadas continuas en el intervalo $[a,b]$.

Definición. [Spline cúbico interpolante.] Diremos que S_Δ es un spline cúbico interpolante para el conjunto de nodos Δ , si :

1. S_Δ es un spline cúbico asociado a Δ .
2. $S_\Delta(x_i) = f(x_i) = y_i$ para $i = 1, 2, \dots, n + 1$, es decir cumple las condiciones del intervalo de interpolación

Antes de construir un spline cúbico se van a ver cuántas condiciones ha de cumplir y cuantas incógnitas van a hacer falta.

Si en cada subintervalo de Δ se intenta construir un polinomio de grado tres que aproxime a la función, se deberán calcular cuatro incógnitas (los cuatro coeficientes del polinomio de grado tres) por subintervalo, es decir $4n$ incógnitas. Por otro lado, estos polinomios deben cumplir, en cada uno de los nodos las condiciones.

$$S_i(x_{i+1}) = S_{i+1}(x_{i+1}) \quad (4.1)$$

$$S'_i(x_{i+1}) = S'_{i+1}(x_{i+1}) \quad (4.2)$$

$$S''_i(x_{i+1}) = S''_{i+1}(x_{i+1}) \quad (4.3)$$

Es decir, se deben cumplir un total de $3(n - 1)$ condiciones además de las $n + 1$ de interpolación:

$$S_i(x_i) = f(x_i) = y_i \quad (4.4) \quad i = 1, 2, \dots, n + 1 \quad (4.5)$$

Dado que tenemos un total de $4n$ incógnitas para $4n - 2$ condiciones, debemos imponer dos nuevas condiciones para poder determinar los coeficientes de la función spline. Dependiendo de las condiciones que imponamos, obtendremos un tipo de spline u otro.

- Si exigimos que la derivada segunda se anule en los extremos, es decir, si:

$$S''_{\Delta}(a) = S''_{\Delta}(b) = 0, \quad (4.6)$$

diremos que $S_{\Delta}(x)$ es el **spline natural** asociado al conjunto de nodos Δ .

- Si exigimos que la derivada primera tome un determinado valor en los extremos, es decir, si

$$S'_{\Delta}(a) = y'_1, \quad (4.7) \quad S'_{\Delta}(b) = y'_{n+1} \quad (4.8)$$

diremos que $S_{\Delta}(x)$ es el **spline sujeto** asociado al conjunto de nodos Δ .

- Si, suponiendo que $y_1 = y_{n+1}$, exigimos que.

$$S'_{\Delta}(a) = S'_{\Delta}(b), \quad (4.9) \quad S''_{\Delta}(a) = S''_{\Delta}(b) \quad (4.10)$$

diremos que se trata de un **spline periódico**.

4.1.1.2. Solución adoptada.

La *toolbox* de MATLAB “Curve fitting toolbox” da la opción de utilizar la función ‘*spline*’ para el filtrado o suavizado de las curvas mediante splines cúbicos. El comando ‘*yy = spline(x,y,xx)*’ realiza la interpolación del conjunto de puntos dados por los vectores ‘*x, y*’

mediante el uso de splines cúbicos. Como salida (variable 'yy') se obtienen los valores suavizados evaluados en los puntos 'xx'.

Si 'y' es una matriz, el spline cúbico se aplicará a cada fila 'y'. Esta función impone las condiciones "not-a-knot" en los extremos de cada intervalo. Si el vector 'y' contiene dos valores más que el 'x', el primer y el último valor del vector 'y' se usan como valores de la derivada en los nodos extremos, y el spline construido es el sujeto.

Ya se mostró anteriormente que se quiere ajustar el nivel de suavizado de la curva aproximada para ver cual se adapta mejor tomando la decisión de que datos son relevantes y cuales anómalos, se decidió utilizar otra función que pertenece a una *toolbox* llamada "*Spline Toolbox*".

La función en cuestión es 'csaps'. El comando 'pp = csaps(x,y,p)' devuelve la forma polinómica a trozos (ppform) de un spline cúbico suavizado 'f' para los datos dados 'x,y', con el valor de 'f' en el lugar de los datos 'x(j)' aproximando el valor de los datos 'y(:,j)', para 'j=1:length(x)'. Los valores pueden ser escalares, vectores o matrices. Los puntos de datos del mismo sitio se sustituirán por su media, ponderada por la suma de los pesos correspondientes. El parámetro 'p' permite fijar el nivel de suavizado.

El parámetro de suavizado determina el peso relativo, según las exigencias contradictorias, de que 'f' sea muy suavizado o de que 'f' esté cerca de los datos. Para 'p=0', 'f' es una línea recta ajustada por mínimos cuadrados a los datos, mientras que, en el otro extremo, es decir, para 'p=1', 'f' es el spline cúbico interpolador natural.

A medida que 'p' varía entre 0 y 1, el spline suavizado varía de un extremo al otro. Se puede poner 'p > 1', pero esto lleva a una spline suavizado aún más duro que el spline cúbico interpolador natural. Si el valor de 'p' en la entrada es negativo o vacío, entonces Matlab utiliza el valor por defecto.

Se ha optado por establecer un valor de 0.95 para este parámetro después de ver los resultados que ofrecían varios valores probados.

Con este valor en el parámetro de suavizado, se consigue que la curva suavizada siga totalmente la tendencia de la curva sin suavizar, a su vez se evita una diferencia grande entre los picos de potencia que impedirían el buen funcionamiento de la red neuronal.

En la Figura 4.5 se muestra una pequeña justificación del valor elegido, en el apartado 6.4 se entrará más en detalle sobre el porqué de esta elección

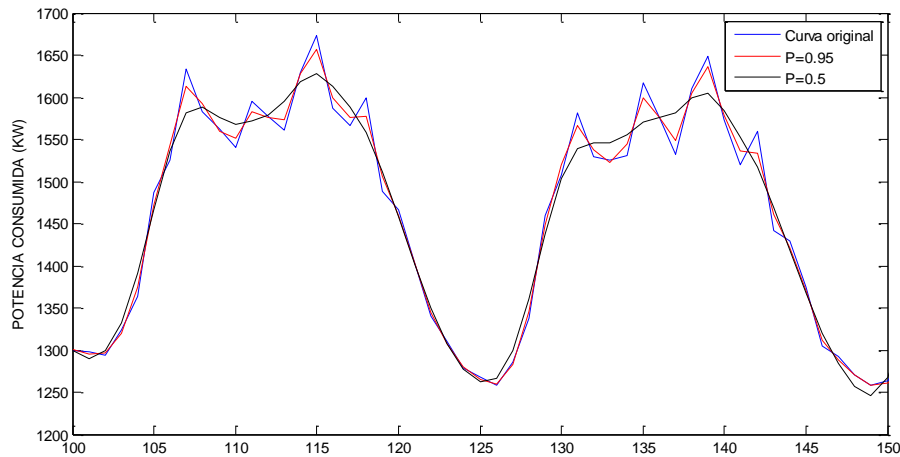


Figura 4.5 Suavizado de la curva de potencia mediante splines cúbicos

4.1.2. SUAVIZADO DE LA CURVA DE POTENCIA CONSUMIDA MEDIANTE EL FILTRO DE SAVITZY Y GOLAY.

4.1.2.1. Introducción

El filtro de Savitzky-Golay es un procedimiento de suavizado de curvas que proporciona una aproximación a la derivada primera. Su forma analítica permite hacer un aproximamiento satisfactorio de la curva en el punto suavizado y en su derivada. Los coeficientes del filtro de Savitzky-Golay se obtienen mediante una aproximación local de mínimos cuadrados en las inmediaciones del punto, estos valores se pueden encontrar en unas tablas tabuladas por los propios Savitzky y Golay. *El número de puntos con respecto a los que se efectúa la aproximación y el grado del polinomio generalizado de aproximación son los factores que condicionan el proceso de suavizado:*

La principal ventaja de esta aproximación es que tiende a preservar características de la curva inicial tales como los mínimos y máximos relativos, que es precisamente la forma de suavizado que se está buscando, así como el ancho de los picos, que normalmente desaparecen con otras técnicas de filtrado y suavizado.

Como se ha comentado, el filtro Savitzky-Golay se basa en ajustar los puntos medidos en una ventana de ancho $2N+1$ (esta ventana hace referencia al número de puntos respecto a los que se efectúa la aproximación) a un polinomio. Es decir, el filtro Savitzky-Golay es un filtro polinomial que se puede describir según la siguiente fórmula:

$$x(k) = \sum_{i=-N}^N a(i)z(k+i) \quad (4.11)$$

Los valores $a(i)$ se pueden calcular mediante la siguiente ecuación.

$$a(n) = \sum_{m=1}^{M+1} \{(G^Y G)^{-1}\}_{1,m} n^{m-1} \quad (4.12)$$

Dónde $G=i_j$ siendo $i=-N....N$ el índice que hace referencia al ancho de la ventana y $j=0....M$ el índice que hace referencia al orden del polinomio, siendo M el máximo orden del polinomio y $2N+1$ el ancho de la ventana.

En el caso de una ventana de 7 puntos ($N=3$) y un polinomio de grado 2 ($M=2$) los coeficientes, serían los siguientes:

$$G = \begin{bmatrix} -3^0 & -3^1 & -3^2 \\ -2^0 & -2^1 & -2^2 \\ -1^0 & -1^1 & -1^2 \\ 0^0 & 0^1 & 0^2 \\ 1^0 & 1^1 & 1^2 \\ 2^0 & 2^1 & 2^2 \\ 3^0 & 3^1 & 3^2 \end{bmatrix}$$

Por lo tanto:

$$(G^T G)^{-1} = \begin{pmatrix} 0.3333 & 0 & -0.047619 \\ 0 & 0.03570 & 0 \\ -0.047619 & 0 & 0.011905 \end{pmatrix}$$

Y sustituyendo en la ecuación 1 se pueden determinar los valores de los coeficientes:

$$a_3 = -0.0952$$

$$a_2 = 0.1429$$

$$a_1 = 0.2857$$

$$a_0 = 0.3333$$

$$a_{-1} = 0.2857$$

$$a_{-2} = 0.1429$$

$$a_{-3} = -0.0952$$

Sustituyendo en la ecuación de filtros polinomiales, se puede hallar el valor del dato suavizado.

No obstante los valores $a(i)$ fueron tabulados para distintos anchos de ventana y grado de polinomio. Estos valores, se pueden encontrar en tablas listas para sustituir en la siguiente ecuación.

$$x(k) = \frac{1}{NORM} \sum_{i=-N}^N a(i)z(k+i) \quad (4.13)$$

Dónde $1/NORM$ es el factor normalización de los coeficientes y $a(i)$ son los valores que toman los coeficientes del polinomio del filtro.

Tabla 1 Valores tabulados Filtro de Golay-Savitzky

Degree	2/3 (quadratic/cubic)			4/5 (quartic/quintic)	
Window size	5	7	9	7	9
-4			-21		15
-3		-2	14	5	-55
-2	-3	3	39	-30	30
-1	12	6	54	75	135
0	17	7	59	131	179
1	12	6	54	75	135
2	-3	3	39	-30	30
3		-2	14	5	-55
4			-21		15
Normalisation	35	21	231	231	429

4.1.2.2. Solución adoptada

Como se ha podido leer en el párrafo anterior la principal ventaja es que se tienen dos variables que modifican las características del suavizado de la curva, el grado del polinomio, y el ancho de ventana. Matlab posee una función que permite realizar el filtrado de datos pudiendo variar las dos características mencionadas.

Se han realizado dos pruebas para determinar la variación de la curva suavizada respecto del grado del polinomio a escoger y el ancho de ventana que mejor se adaptan a las características requeridas en el suavizado

Prueba 1: Variación del ancho de ventana para un polinomio de grado 2.

El primer ancho de ventana escogido fue el mínimo ancho de ventana posible:3.

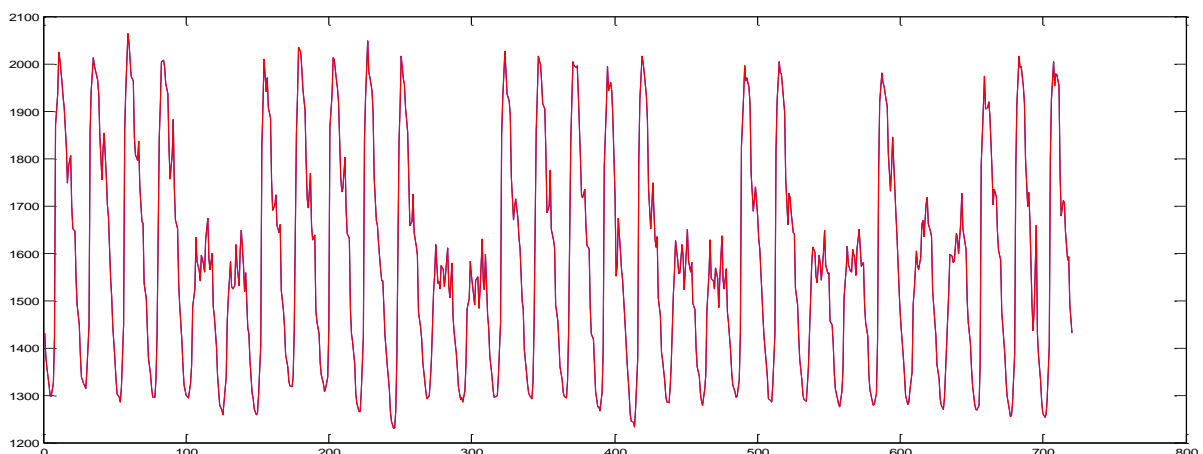


Figura 4.6 Savitzky-Golay Grado =2 Ancho de ventana =3El

El resultado que se muestra en la **¡Error! No se encuentra el origen de la referencia.** s una superposición de curvas, es decir la curva original es igual a la curva suavizada, esto es debido a que el ancho de ventana es muy pequeño, el mínimo posible y por lo tanto el suavizado es mínimo.

A posteriori, se realizaron distintas simulaciones con distintos anchos de ventana, el resultado se muestra en la Figura 4.7

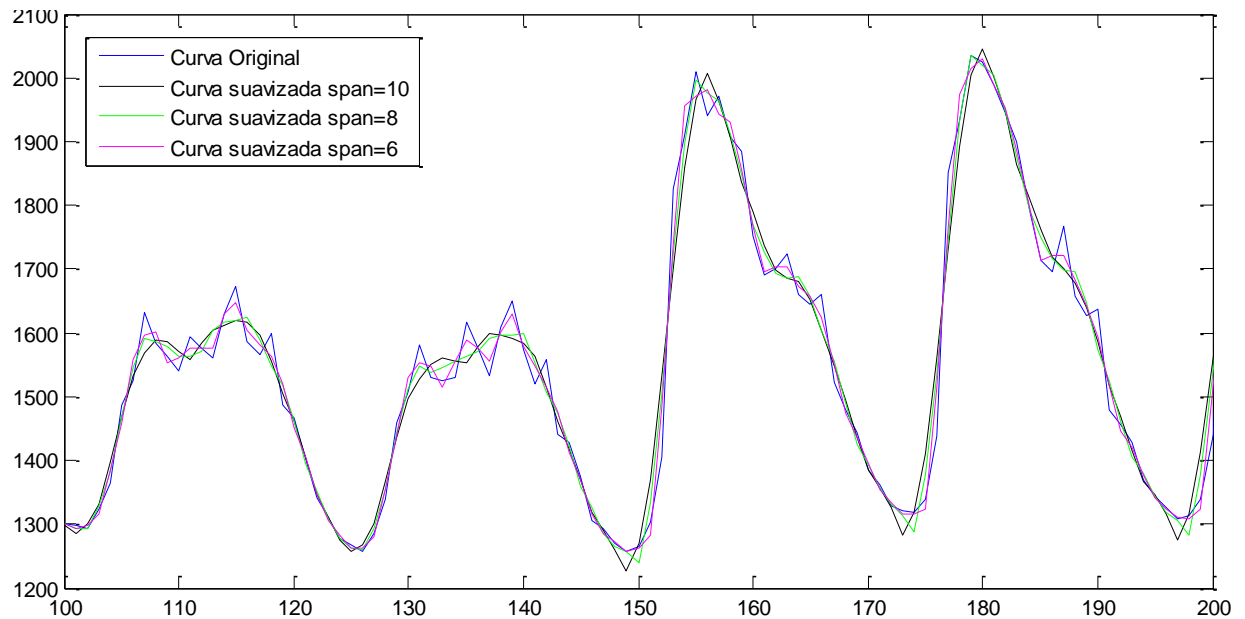


Figura 4.7 Savitzky-Golay Grado 2

Dónde:

Span 10= Ancho de ventana 10

Span 8= Ancho de ventana 8

Span 6= Ancho de ventana 6

Como se observa en la ilustración anterior, cuanto mayor es el ancho de ventana mayor es el suavizado obtenido. Por tanto, la elección del ancho de la ventana es un factor muy importante puesto que si el ancho de ventana es mínimo no se consigue el suavizado deseado, mientras que si el ancho de ventana es excesivo se corre el riesgo de perder información.

Prueba 2: Variación del grado del polinomio para un ancho de ventana de 6.

En la Figura 4.8 se observa una superposición de curvas, la curva suavizada mediante polinomio de grado 2 y grado 3 es la misma, mientras que, la curva original, la curva suavizada mediante polinomio de grado 4 y la curva suavizada de grado 5 son idénticas.

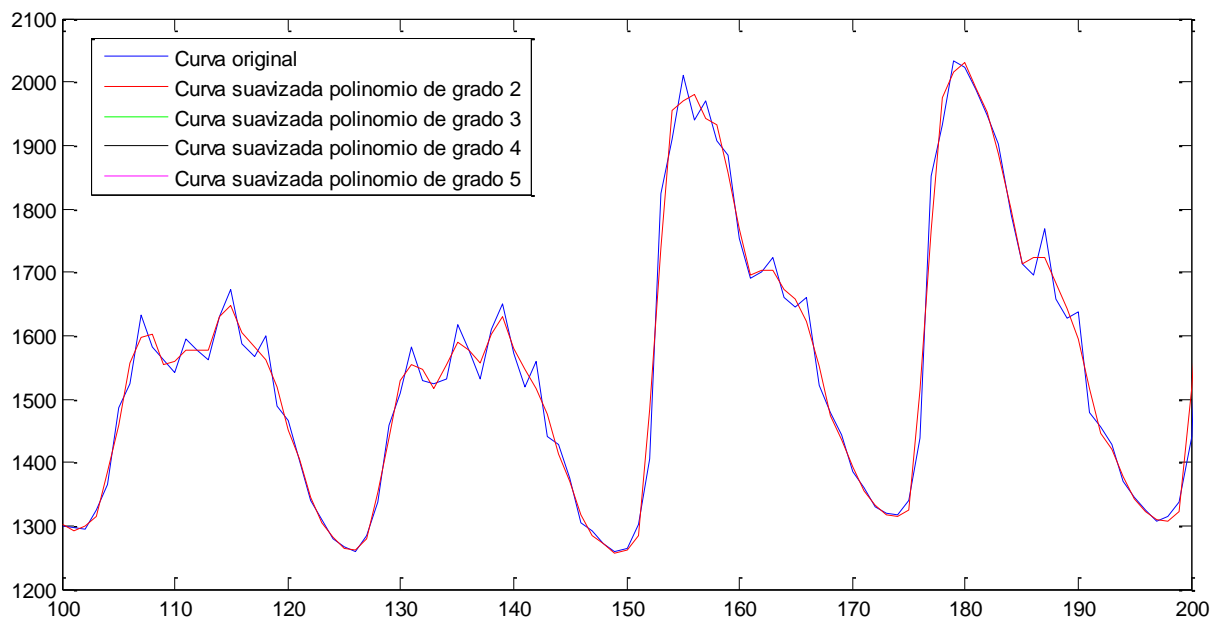


Figura 4.8 Savitzky-Golay diferentes grados de polinomio

Es decir, ocurre lo opuesto que ocurriría al modificar el ancho de ventana, cuanto mayor sea el grado del polinomio usado, menor será el suavizado, y menor la información perdida y viceversa.

Debido a las características de la curva suavizada que se quiere obtener, donde no importan tanto los picos de potencia consumidos y sí que se siga la tendencia de la curva original, se opta por elegir un **polinomio de grado 3 y un ancho de ventana de 6 datos**. Al igual que en el caso anterior, se entrará más en detalle en el apartado 6.4

4.1.3. ESCALADO DE LOS VALORES DE POTENCIA

La última modificación a realizar con los datos de potencias, es escalarlos con valores entre 0 y 1, necesario para un mejor funcionamiento de la red neuronal como se ha mencionado anteriormente.

Todos los datos de potencia necesarios para las predicciones que se quieran realizar se guardan consecutivamente en un vector. El primer dato del vector será la potencia escalada del consumo de potencia de la primera hora necesaria para realizar la predicción de la primera semana elegida y el último dato, será la potencia de la última hora de la última semana a predecir.

4.2. CODIFICACIÓN DE LOS DATOS DE TEMPERATURA.

La temperatura está directamente relacionada con la demanda eléctrica, en caso de temperaturas frías será necesario un sistema de calefacción, y en caso de temperaturas calurosas se utiliza el sistema de refrigeración como el aire acondicionado. Los dos sistemas son unos de los principales consumidores de energía por lo que no se puede despreciar su uso. Si bien se podría considerar un modelo de todo o nada en el que o se enciende el sistema de calefacción o se enciende el sistema de refrigeración, y se asume que se produce cierto equilibrio en el consumo y que éste es constante a lo largo del año. Se verá que es en las estaciones de transición como son el otoño y la primavera cuando se tienen los mayores problemas a la hora de estimar la demanda eléctrica. La razón de estos problemas de estimación surge en que, si la demanda real sufre variaciones a lo largo del período a estimar, dichas variaciones serán muy difíciles de ajustar sin información extra para la red que no dependa solamente del histórico de carga.

Se poseen en Excel los datos horarios de temperatura horaria formando una matriz de 24 horas por el número de días del mes. Al igual que antes escalaremos los datos de temperatura entre 0 y 1.

En el algoritmo que se propone, no se va a trabajar con la temperatura horaria directamente, sino que para un mismo día se van a necesitar únicamente dos temperaturas: la máxima y la mínima. Por lo tanto hacen falta dos vectores, en un vector se almacenarán por día 24 valores iguales correspondientes a la máxima temperatura, y en el otro vector los 24 valores correspondientes a la mínima temperatura.

4.3. CODIFICACIÓN DE LOS DIAS LABORABLES Y DIAS NO LABORABLES.

Como se observa en la Figura 4.1 y en la Figura 4.2 se puede distinguir claramente que el consumo de potencia los fines de semana es inferior al de los días laborables al igual que ocurriría en cualquier otro día no laboral. Por este motivo es necesario distinguir entre días laborables y días no laborables, para ello como se ha mencionado antes será necesario codificar esta información. En este caso se codifica mediante la siguiente regla.

Día laboral = 1

Día no laboral o fin de semana= 0

Mediante esta norma, se tiene un método que permite considerar este factor tan influyente. Al igual que en el resto de los casos se almacenará en un vector todos los 1 y 0 necesarios para definir el periodo de tiempo necesario para la predicción.

4.4. CODIFICACIÓN DE LA HORA.

Al colocar todos los datos anteriores ordenadamente en vectores se puede informar a la red neuronal de la hora exacta a la que pertenece ese dato. La hora también es un dato relevante en el consumo eléctrico del hospital, siendo mayor el consumo durante el día que durante la noche.

Se considera la hora como una variable categórica por lo tanto habrá que codificarla en binario. Los valores de las horas en números naturales van desde 0 hasta 23 por lo que hará falta una columna de 5 dígitos binarios para codificarlas.

4.5. ORGANIZACIÓN DE LOS DATOS PARA EL ENTRENAMIENTO DE LA RED NEURONAL.

Una vez que se tienen los datos de potencia, temperaturas y días festivos ordenados y codificados en vectores se puede proceder a formar las matrices necesarias para el entrenamiento de la red neuronal. Para el **método de máxima y mínima temperatura** propuesto como ya se ha visto, se utilizan las temperaturas máximas y mínimas de cada día.

Para el entrenamiento de la red neuronal es necesario introducir a la red neuronal un conjunto de datos que después del procedimiento aplicado en la red neuronal darán como salida la potencia consumida en esa hora. En nuestro caso inicial, las entradas serán 5 variables indicando la potencia consumida en las 3 horas previas, el día anterior y la potencia consumida hace justo una semana, una variable binaria indicando si el día es laborable o no, 5 variables binarias indicando la hora y por último 2 variables con la temperatura máxima y mínima del día.

La salida y objetivo de la red neuronal, será la potencia consumida en la hora correspondiente. La explicación de estos detalles se ofrecerá en el funcionamiento de la red neuronal.

4.5.1. MÉTODO MÁXIMO Y MÍNIMO DE TEMPERATURA.

En este método cada columna de datos de entrada a la red neuronal será la siguiente: La matriz se aumentara hasta completar todos los datos necesarios para el entrenamiento.

Potencia consumida hace 1 semana(i), ..., Potencia consumida hace 1 semana(n)
Potencia consumida hace 1 día(i), ..., Potencia consumida hace 1 día(n)
Potencia consumida hace 3 horas(i), ..., Potencia consumida hace 3 horas(n)
Potencia consumida hace 2 horas(i), ..., Potencia consumida hace 2 horas(n)
Potencia consumida hace 1 hora(i), ..., Potencia consumida hace 1 hora(n)
Dia laboral o no(i), ..., Dia laboral o no(n)
Hora 1(i), ..., Hora 1(n)
Hora 2(i), ..., Hora 2(n)
Hora 3(i), ..., Hora 3(n)
Hora 4(i), ..., Hora 4(n)
Hora 5(i), ..., Hora 5(n)
Temperatura máxima(i), ..., Temperatura máxima(n)
Temperatura mínima(i), ..., Temperatura mínima(n)

Donde (i) representa el primer dato necesario y (n) el último dato necesario.

En nuestro caso se ha decidido realizar el entrenamiento con 28 días, por lo que serán necesarios todos los datos de esos 28 días. En total trabajaremos con $28 \times 24 = 672$ columnas de datos.

4.5.2. TARGETS.

La red neuronal deberá entrenarse de tal modo que cada columna, que tendrá el aspecto de la mostrada anteriormente tendrá que dar como resultado o acercarse al valor de la potencia real consumida de la hora correspondiente. Esta potencia real estará guardada en un vector que al igual que las matrices anteriores tendrá un total de 672 columnas.

(potencia consumida(i) ... potencia consumida (n))

5. DISEÑO DE LA RED NEURONAL.

Para realizar la implementación de la red neuronal se ha utiliza una *toolbox* de Matlab que permite manejar redes neuronales con gran facilidad. Sin embargo habrá que definir una serie de parámetros, como el tipo de red neuronal, (Perceptrón multicapa en este caso) el número de capas, el número de neuronas, el método de entrenamiento...

5.1. NÚMERO DE CAPAS

La primera característica a completar, es definir el número de capas que van a formar la red neuronal, como se mencionó con anterioridad una red neuronal está compuesta por una capa de entrada, una capa de salida y una o varias capas ocultas en el medio de las dos anteriores. La mayoría de los problemas se pueden resolver con una capa intermedia y por lo tanto una única capa intermedia será la elección. La red neuronal se puede representar de la siguiente manera (Figura 5.1).

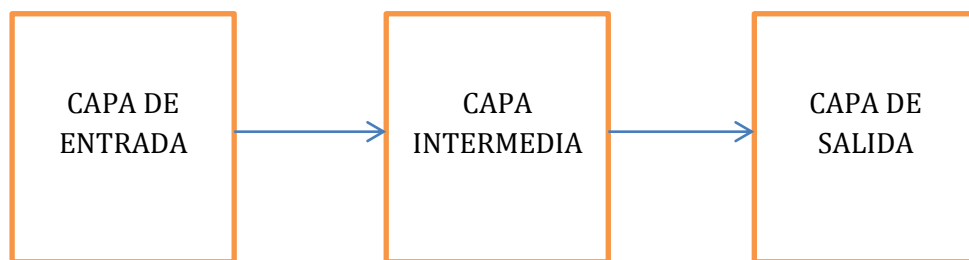


Figura 5.1 Representación número de capas de la red neuronal

5.2. NÚMERO DE NEURONAS

Capa de entrada

El número de neuronas de la capa de entrada viene definido por el número de entradas a la red neuronal por lo tanto inicialmente se tendrán 13 neuronas en la capa de entrada, aunque después este número de neuronas se verá reducido debido a la eliminación de alguna de las entradas.

Capa intermedia

No existe ningún procedimiento específico que indique de una manera clara el número de neuronas de la capa intermedia, aunque si existen dos vertientes que son bastante utilizadas para hallar el número de neuronas de la capa oculta:

- Prueba y error: Este método consiste en entrenar la red varias veces con un número de neuronas diferentes en la capa intermedia. Se seleccionará el número de neuronas con un menor error sobre los datos de validación.

- Mediante reglas heurísticas: Se realiza una optimización de los parámetros de la red mediante algoritmos genéticos. Esto presenta el inconveniente de que se necesita mucho tiempo para dar con la solución al problema. La determinación del número óptimo de neuronas de la capa oculta está estrechamente ligada con la determinación de la complejidad óptima del modelo.

En este proyecto se han trabajado con distintos números de variables de entrada, que es un parámetro que también afecta a la elección del número de neuronas de la capa intermedia, como a lo largo del proyecto se modifica el número de variables de entrada, también será necesario modificar el número de neuronas de la capa intermedia.

Para el caso inicial en el que se trabajan con 13 variables de entrada, se han utilizado 15 neuronas en la capa intermedia, que fue el número de neuronas óptimo utilizados en proyectos anteriores de este departamento. En el resto de casos se justificará más adelante.

Capa de Salida

Únicamente se utilizará una neurona en la capa de salida, ya que solo se quiere conocer la potencia estimada.

5.3. ALGORITMO DE ENTRENAMIENTO

El entrenamiento de una red neuronal consiste en la determinación de los pesos sinápticos de cada una de las neuronas.

Se emplean múltiples métodos para determinar los pesos. El método más conocido y del que ya se ha hablado en el punto 3.3.2 es el algoritmo *backpropagation*, sin embargo, también existen otros métodos que determinan el valor de los pesos sinápticos minimizando la suma de cuadrados del error (mse) de predicción en los datos de entrenamiento y que también son perfectamente válidos

El método *backpropagation* o método del gradiente, es un método clásico de optimización, en el cual se calcula del gradiente mediante un proceso de “propagación” del error de atrás hacia adelante, es decir, el error se propaga desde las neuronas de salida hasta las neuronas de entrada. Este algoritmo, varía los pesos sinápticos minimizando el error de la salida real y la salida predicha. El **método de Levenberg-Marquardt**, que es el que se utilizará en la *toolbox*, es una mezcla entre el método *backpropagation* y el algoritmo de Gauss-Newton.

El algoritmo *backpropagation* como ya se vio está controlado por los siguientes parámetros:

- **Tasa de aprendizaje:** constante que relaciona la variación de los pesos sinápticos con el gradiente del error respecto a dicho peso sináptico.

- **Momento** Acelera el proceso de descenso del algoritmo hacia el punto de convergencia,
- **Número de iteraciones:** Este término se conoce como “épocas” y determina el número de veces que cambian los pesos sinápticos. Es un parámetro muy importante ya que influye mucho en la calidad del entrenamiento, por esta razón, debe seleccionarse una duración del entrenamiento adecuada.
 - Un sobre-entrenamiento, puede provocar que la red neuronal “se aprenda de memoria” los datos de entrenamiento y la soluciones, y no sea capaz de “razonar” ante el resto de datos de entrada, ofreciendo unas salidas incorrectas.
 - Un entrenamiento corto es insuficiente para que la red neuronal sea capaz de “asimilar” el algoritmo que modela el sistema.

Para evitar el problema del sobre-entrenamiento se utilizan técnicas de validación durante el entrenamiento. Matlab, evita este problema mediante el algoritmo de Levenberg-Marquardt.

La selección de todos estos parámetros se suele realizar mediante prueba y error, salvo en el uso de redes neuronales en programas como Matlab en los cuales dicha selección no se puede configurar y viene determinada automáticamente por el algoritmo de entrenamiento. Las iteraciones dependerán de unas condiciones de parada de dicho entrenamiento. Las variables que se puede imponer en Matlab son: valores límite de iteraciones, tiempo de entrenamiento, valor del gradiente máximo y mínimo, etc.

5.3.1. SELECCIÓN DE DATOS EN LOS DISTINTOS CONJUNTOS DE ENTRENAMIENTO

Como se comentó en secciones pasadas, en el entrenamiento de una red neuronal actúan tres subconjuntos de datos: el conjunto de entrenamiento (*training set*), el conjunto de validación (*validation set*) y el conjunto de prueba (*test set*).

Con el conjunto de entrenamiento se calcula la pendiente del gradiente y se actualizan los pesos sinápticos de la red.

El conjunto de validación, que ofrece una primera evaluación del modelo en la misma fase de entrenamiento, es usado para minimizar el sobreajuste del modelo, con este conjunto se verifica que la modificación de los pesos sinápticos debido a la minimización del error de entrenamiento conlleva una mejora en el conjunto de datos y no solo sobre los datos de entrenamiento. Para que no haya un sobre entrenamiento, tanto el error de los datos de entrenamiento como el error de los datos de validación deben reducirse.

El *test set* comprueba la validez o no del modelo una vez finalizada la fase de entrenamiento. Si este error es parecido al error de validación podría significar que el reparto de datos entre los diferentes conjuntos no ha sido el adecuado.

La *toolbox* de Matlab ofrece cuatro funciones diferentes para asignar de distinta manera a cada subconjunto (*training set*, *validation set*, *test set*) los datos de entrenamiento.

- **dividerand**: Divide los datos al azar (por defecto) en tres subgrupos usando la división de parámetros `net.divideParam.trainRatio`, `net.divideParam.valRatio` y `net.divideParam.testRatio`. La fracción de datos que se coloca en el conjunto de entrenamiento es $\text{trainRatio} / (\text{trainRatio} + \text{valRatio} + \text{testRatio})$, con una fórmula similar para los otros dos conjuntos. Las ratios por defecto para el entrenamiento, prueba y validación son 0,7, 0,15 y 0,15, respectivamente.
- **divideblock**: Los datos se dividen en tres subgrupos usando tres bloques contiguos de los datos originales (*training* utiliza el primer bloque, *validation* el segundo y *testing* el tercero). La fracción de los datos originales que se dedica a cada subgrupo es determinada por los mismos parámetros usados para `dividerand`.
- **divideint**: Los datos se dividen por un método entrelazado, como en el barajeo de un mazo de cartas. Esto se hace para que diferentes porcentajes de los datos entren en los tres subgrupos. La fracción de los datos originales que se dedica a cada subgrupo es determinada por los mismos parámetros usados para `dividerand`.
- **divideind**: Los datos son divididos por su índice. Los índices de los tres subconjuntos se definen por la entrada de los parámetros usando previamente `net.divideParam.trainInd`, `net.divideParam.valInd` y `net.divideParam.testInd`. La asignación estándar de estos índices es la matriz nula, por lo que se debe establecer los índices cuando se usa esta opción.

En el proyecto se ha optado por usar la función de `divideind` para poder asignar los datos a cada grupo “manualmente”. Esta elección es debida a la curva que tiene la potencia consumida formadas por picos y subidas y bajadas de unos picos a otros, con este método conseguimos que los datos se repartan de una forma equivalente y que cada subgrupo posea datos de todas las zonas de la curva.

5.3.2. VENTAJAS E INCONVENIENTES DEL USO DE REDES NEURONALES ARTIFICIALES.

Una red neuronal artificial requiere una menor experiencia, ya que es el propio algoritmo de aprendizaje el que determina cuales son los pesos sinópticos, a partir de los datos históricos. Sin embargo, sí que es necesario tener un buen conocimiento de las redes neuronales

Los más importante y a la vez complicado en una red neuronal es el proceso de selección de variables exógenas como precisamente se ve en este proyecto. La precisión de las

predicciones de las redes neuronales depende fuertemente de las variables escogidas. De hecho, muchos artículos se dedican únicamente a discutir que variables se han de escoger y cuáles no. Muchos de estos escritos llegan incluso a defender posiciones totalmente distintas. En el caso estudiado para este proyecto, por ejemplo, hay autores que defiende la inclusión de variables a largo plazo y otros autores que solo incluyen variables a corto plazo.

Las redes neuronales son modelos de caja negra. Es decir, es imposible interpretar el significado de los pesos sinápticos y su relación con las variables. De esta forma, si se utiliza una red neuronal artificial para predecir una variable, no será posible averiguar qué camino a llevado a la red neuronal a ofrecer esa salida. Es decir, por muy buena que sea una predicción no se sabrá porque ha sido buena y si ha sido una predicción errónea, no se sabrán los errores que han llevado a la red neuronal a predecir erróneamente

La red neuronal requiere que el modelo a predecir no presente ninguna tendencia a medio o largo plazo. Lo que puede obligar a acondicionar algunos datos pasados. Sí el sistema a predecir tiene una ligera tendencia la red neuronal será capaz de interpolar con un acierto determinado, pero será incapaz de extrapolar con el mismo acierto. De esta manera hará falta un experto que ayude a interpretar los resultados

Como ya se ha comentado antes, no existe ningún método específico para determinar de una manera precisa los parámetros de la red neuronal quedando la determinación de estos parámetros expuestos a los criterios de prueba error o a métodos heurísticos que conllevan un gran gasto de tiempo.

Las redes neuronales presentan un mal comportamiento ante datos erróneos o anómalos. Por ello existe la necesidad de una etapa de filtrado y suavizado, así como de escalado. Las redes neuronales son vulnerables ante una mala elección de volumen de datos, pero como en casos anteriores, será la experiencia y los métodos de prueba y error los que verifiquen la validez del conjunto de datos.

Respecto a la funcionalidad de las redes neuronales en el ámbito de la predicción eléctrica existen muchos estudios tanto de predicciones a largo plazo como de predicciones a corto plazo, obteniéndose mejores resultados en este último tipo de predicciones.

6. CASO PRÁCTICO

En este apartado del proyecto se abordará el estudio inicial de la red neuronal propuesta, se mostrarán las simulaciones practicadas para verificar el buen funcionamiento de la red y sus ventajas e inconvenientes, y se intentará dar una justificación a los resultados obtenidos, así como distinguir y comentar los resultados que da una misma simulación con distintos métodos.

Se cuentan con los datos de dos años de potencia consumida del hospital de León por lo que todas las simulaciones deberán corresponder a ese periodo de tiempo (Abril 2008 - Abril 2010). Como ya se ha escrito antes, el método de simulación será el siguiente. Entrenamiento de la red neuronal con un conjunto de 28 días, predicción de la semana siguiente a esos 28 días.

La red neuronal utilizará datos horarios de potencia y temperatura por lo que tanto en las curvas predichas, como las curvas de datos disponibles se encontraran rectas uniendo los distintos datos horarios.

En este proyecto, se ha decidido realizar el entrenamiento y la predicción de la siguiente manera: 28 días de entrenamiento, predicción de los 7 días posteriores. Anteriormente en el departamento, se habían realizado 2 tipos de predicciones, la primera, una red neuronal con 3 meses de entrenamiento y 1 mes de de predicción, la segunda, 28 días de entrenamiento, 7 de predicción. La decisión de realizar la predicción mediante este último formato es simple. La simulación precisa de menos datos, es más rápida y con una mucha mayor libertad para elegir la semana a simular además, al predecir un menor periodo de tiempo, se espera que los resultados sean más precisos.

6.1. CARACTERÍSTICAS DE LA RED NEURONAL DE MATLAB

La red neuronal se creará con la *toolbox*: “*Neuronal Network Toolbox*” que tiene el aspecto mostrado en la Figura 6.2, tiene las siguientes características:

- Numero de capas: 3
- Numero de Neuronas:
 - Capa de entrada = Numero de variables de entrada.
 - Capa oculta = 15 para el caso inicial, después se modifica.
 - Capa de salida = 1 neurona indicando la potencia predicha.
- Algoritmo de entrenamiento: Lanverberg-Manquardt backpropagation.
- División de los datos: Divideind. Se escribirá un apartado específico para aclarar esta parte. Como ejemplo se muestra la Figura 6.1 donde se observa cómo se dividen los datos en conjuntos de 10 datos, para posteriormente distribuirles entre los distintos grupos de entrenamiento.

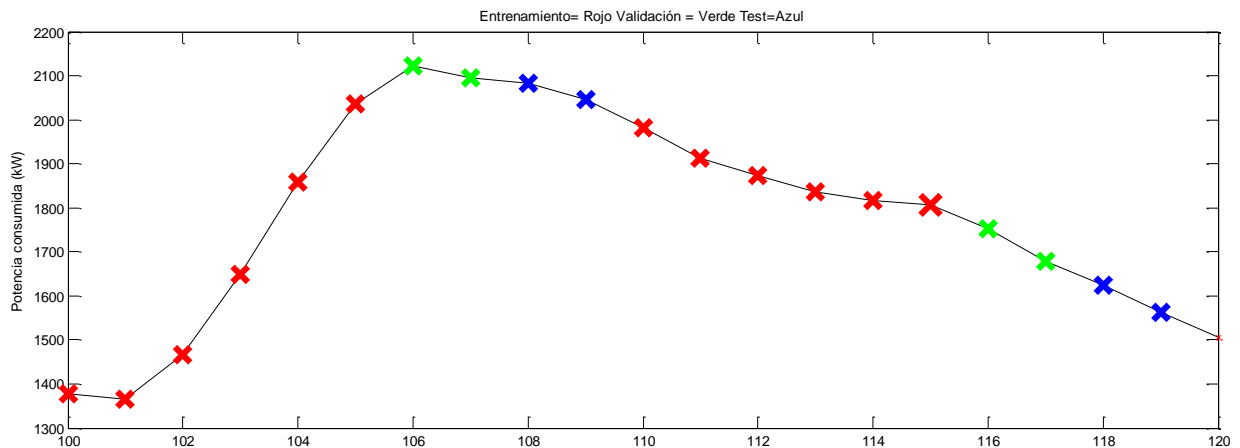


Figura 6.1 Datos de entrenamiento (rojo), validación (verde) y test (azul)

- Función de Activación neuronas de la capa oculta: Debido a que las variables exógenas están acotadas entre 0 y 1 se usa la función LogSig
- Función de Activación de la neurona de la capa de salida: Función lineal, debido a tiene que ofrecer como salida

A la hora de entrenar las redes neuronales, uno de los problemas que se debe tener en cuenta es el llamado sobreajuste (o sobre-entrenamiento). El error en el conjunto de entrenamiento es de un valor pequeño, pero cuando se introducen nuevos datos en la red el error se magnifica. La red ha memorizado los ejemplos del entrenamiento, pero no ha aprendido a generalizar ante nuevas situaciones. Hay dos métodos para mejorar la generalización de la red implementados en Neural Network Toolbox de Matlab: la regularización y la detención temprana. En este proyecto se usará el método de la detección temprana, pero con el objetivo de estudiar la robustez de la red, ya que el tamaño de esta ya se había determinado previamente en otro estudio.

-Método de la detención temprana (early stopping) El método por defecto para mejorar la generalización se llama detención temprana. Esta técnica se equipa por defecto en todas las funciones de creación de redes, incluyendo las que utilizan el algoritmo de retropropagación (o backpropagation) como la función newff (feed-forward backpropagation network).

Este método se utiliza para evitar el sobreajuste usando la validación cruzada, en la que el conjunto de entrenamiento se divide en un conjunto de estimación (*training set*) y otro de validación (*validation set*).

El subconjunto de estimación es usado para entrenar la red del modo habitual, salvo por una ligera modificación: la sesión de entrenamiento se detiene periódicamente para que la red sea testeada en el subconjunto de validación.

El algoritmo se detendrá ante cualquiera de estos casos:

- El error de salida es menor o igual al deseado.
- Sobrepaso de un cierto tiempo.
- Sobrepaso del número de épocas.
- Diferencia significativa entre el error de validación y el error de prueba.
- El error de validación se mantiene igual o crece.

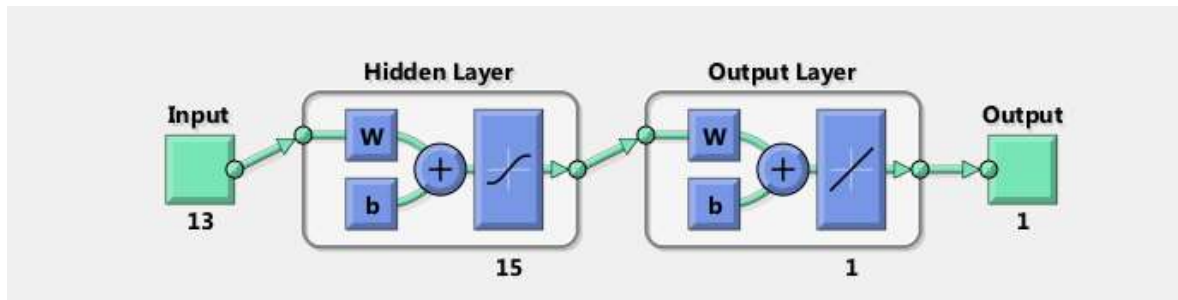


Figura 6.2 Aspecto de la red neuronal en la Toolbox

6.2. VARIABLES DE ENTRADA INICIALES.

Como ya se ha comentado antes, inicialmente se trabajan con 13 variables. Estas variables no han sido elegidas al azar, sino que ya han sido previamente estudiadas en distintos proyectos y en distintas publicaciones (Murto, 2008)

En estos proyectos y publicaciones se afirma la necesidad de trabajar y predecir con la ayuda de datos pasados del consumo de energía eléctrica. La necesidad de introducir la hora como variable y diferenciar entre días festivos y no festivos. Además, se ha decidido introducir la variable temperatura para que la red pueda encontrar un motivo para asimilar las distintas variaciones de potencia durante los distintos periodos de tiempo.

Las variables exógenas deberán estar acotadas entre 0 y 1.

Potencia consumida hace 1 semana
Potencia consumida hace 1 día
Potencia consumida hace 3 horas
Potencia consumida hace 2 horas
Potencia consumida hace 1 hora
Día laboral o no
Hora 1
Hora 2
Hora 3
Hora 4
Hora 5
Temperatura máxima
Temperatura mínima

Un ejemplo de la estructura anterior se puede encontrar en la Tabla 2, extraída de Matlab, que contiene los primeros datos de la última semana de Agosto de 2008.

Tabla 2 Aspecto datos de entrada a la red neuronal.

	Hora 1 Día X	Hora 2 Día X	Hora 3 Día X	Hora 4 Día X	Hora 5 Día X	Hora 6 Día X	Hora 7 Día X
Potencia consumida hace 1 semana	0,23	0,21	0,18	0,16	0,15	0,14	0,20
Potencia consumida hace 1 día	0,22	0,20	0,17	0,16	0,15	0,15	0,18
Potencia consumida hace 3 horas	0,36	0,32	0,27	0,22	0,19	0,17	0,16
Potencia consumida hace 2 hora	0,32	0,27	0,22	0,19	0,17	0,16	0,15
Potencia Consumida hace 1 hora	0,27	0,22	0,19	0,17	0,16	0,15	0,14
Día festivo	1	1	1	1	1	1	1
Dígito 1 Hora	0	0	0	0	0	0	0
Dígito 2 Hora	0	0	0	0	0	0	0
Dígito 3 Hora	0	0	0	0	1	1	1
Dígito 4 Hora	0	0	1	1	0	0	1
Dígito 5 Hora	0	1	0	1	0	1	0
Temperatura máxima	0,79	0,79	0,79	0,79	0,79	0,79	0,79
Temperatura mínima	0,50	0,50	0,50	0,50	0,50	0,50	0,50

6.3. ELECCIÓN DE LOS DATOS DE ENTRENAMIENTO.

Ya se ha escrito en esta memoria sobre los cuatro métodos que tiene la *toolbox* para seleccionar los datos entre los subgrupos de entrenamiento validación y test. Estas funciones son: *dividerand*, *divideblock*, *divideint*, *divideind*, la diferencia entre estas funciones se puede consultar en el punto 5.3.1. En el proyecto se ha optado por usar la función de *divideind* para poder asignar los datos a cada grupo “manualmente”, esto es debido a la curva que tiene la potencia consumida formadas por picos y subidas y bajadas de unos picos a otros, con este método conseguimos que los datos se repartan de una forma equivalente y que cada subgrupo posea datos de todas las zonas de la curva.

La función `divideind` tiene la siguiente forma:

```
[trainInd, valInd, testInd] = divideind(Q, trainInd, valInd, testInd)
```

Donde Q es el conjunto de datos de entrenamiento, `trainInd` los datos que van a parar al conjunto de entrenamiento introducidos manualmente, de igual manera a `valInd` se le asignan los datos del subgrupo validación y a `testInd` se le asignan los datos del subgrupo test.

Lo primero que se ha hecho para poder seleccionar los datos de cada grupo es dividir el total de los datos de entrenamiento en grupos de 10 puntos. Dentro de cada grupo se dividirán los datos de la siguiente manera:

- 6 datos seguidos al *training set*.
- Los 2 siguientes datos irán a parar al *validation set*.
- Los últimos 2 datos al *test set*.

Se quieren que todos los datos del grupo de entrenamiento pasen por todas las posiciones del grupo de 10 datos, de modo que un mismo dato esté 6 veces en el *training set*, 2 veces en el *validation test* y 2 veces en el *test set*. De esta forma se asegurará que si simulamos 10 veces todos los datos pasan por todas las posiciones y se evitan errores que puedan ser causados debido a una situación específica. Para poder seleccionar cada uno de las 10 opciones de entrenamiento, al ejecutar el programa, Matlab pedirá una variable llamada desplazamiento, que podrá tener unos valores comprendidos entre 0 y 9.

A continuación, se exponen unos ejemplos que justifican esta solución adoptada. Los tres casos mostrados a continuación pertenecen a la tercera semana de Julio, una de las semanas con más errores a la hora de predecir. Se puede observar, que existe una variación en la potencia predicha en función de los grupos de entrenamiento. Si solo se hubiera utilizado un grupo de entrenamiento, y este hubiera sido el grupo de entrenamiento número 6, el error hubiera sido mucho mayor que el error proporcionado por la predicción de otros grupos de entrenamiento. Lo que se busca al simular 10 veces, es reducir al mínimo los errores de predicción debido a situaciones específicas del grupo de entrenamiento.

Para mostrar estos ejemplos se utilizan 6 figuras, las Figuras 6.3, 6.5 y 6.7 muestran como varía la asignación de datos a los diferentes subconjuntos en los distintos grupos de entrenamiento, mientras que las Figuras 6.4, 6.6 y 6.8 muestran los resultados obtenidos mediante estos conjuntos de entrenamiento. Como se comentó en el párrafo anterior, el error de la Figura 6.8 es mucho mayor que el mostrado por las Figuras 6.4 y 6.6. de ahí la necesidad de realizar las 10 simulaciones para evitar situaciones específicas.

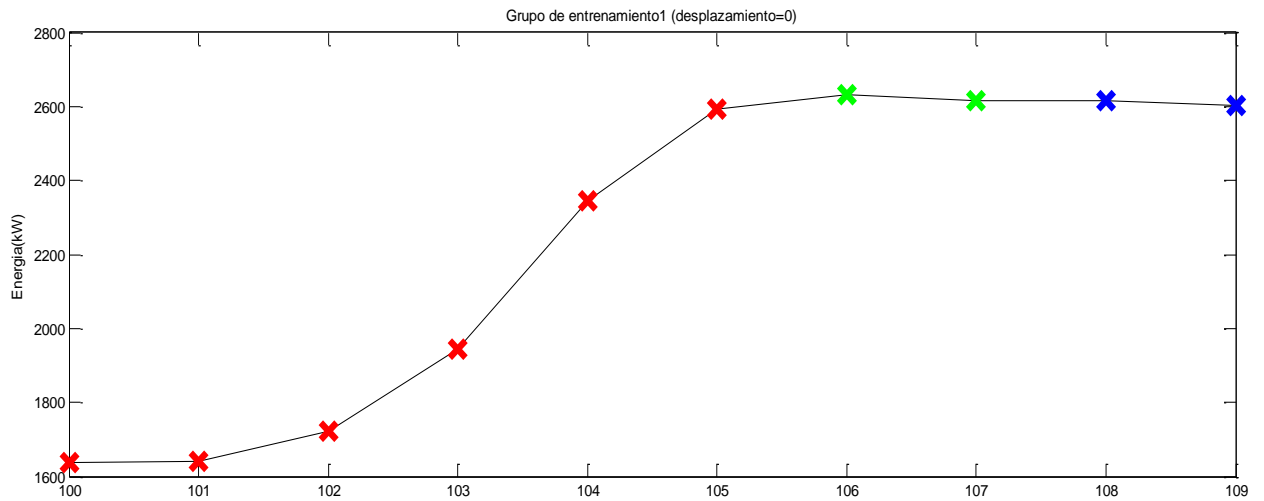


Figura 6.3 Grupo 1 de entrenamiento. Rojo (Training set), Verde (Validation Set) Azul (Test set)

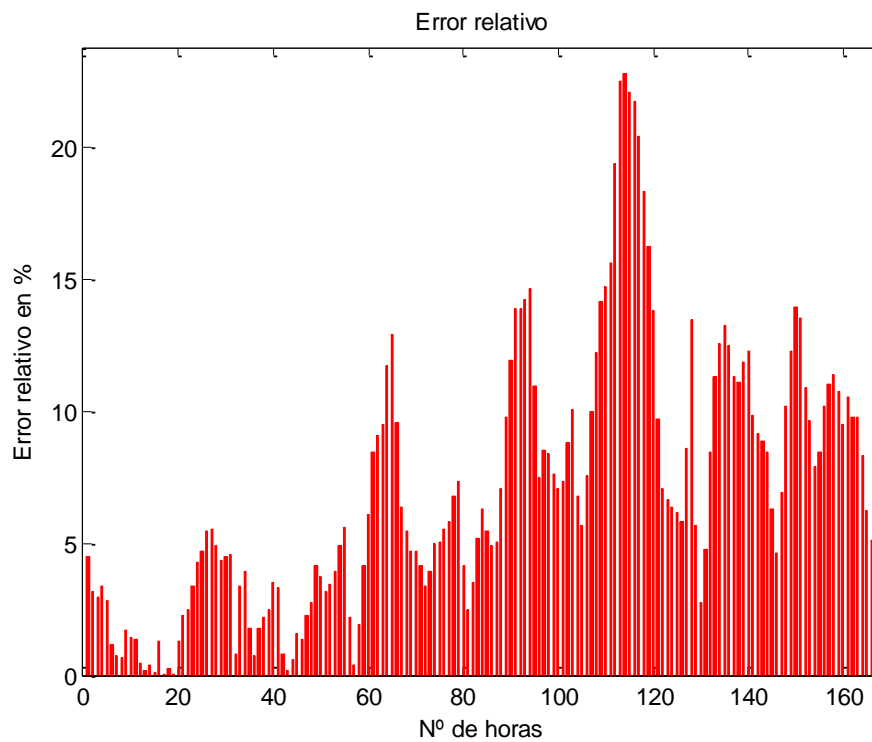


Figura 6.4 Error relativo con el grupo de entrenamiento 1

Error medio: 7.0121

Desviación típica de la estimación: 151.6995 kW

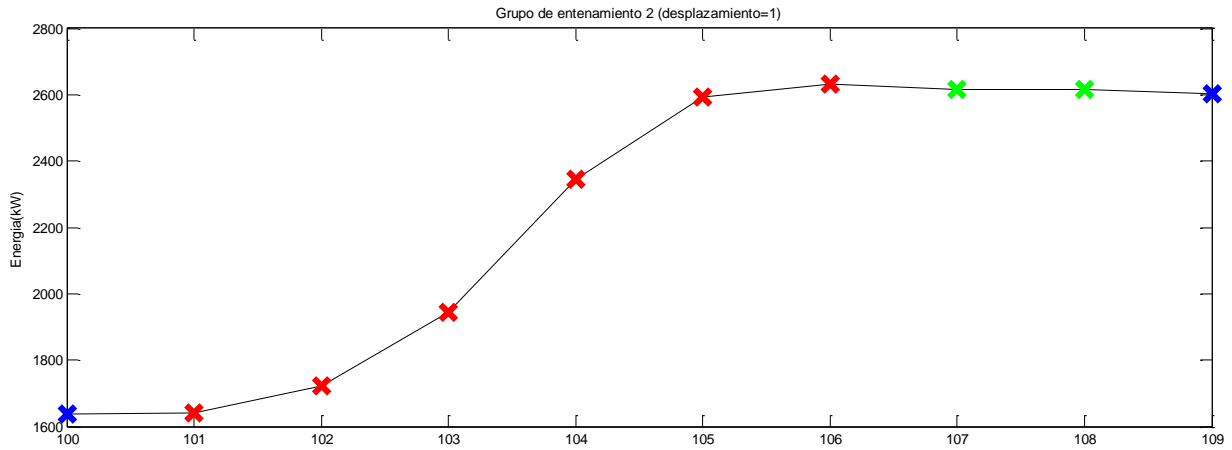


Figura 6.5 Grupo 2 de entrenamiento. Rojo (Training set), Verde (Validation Set) Azul (Test set)

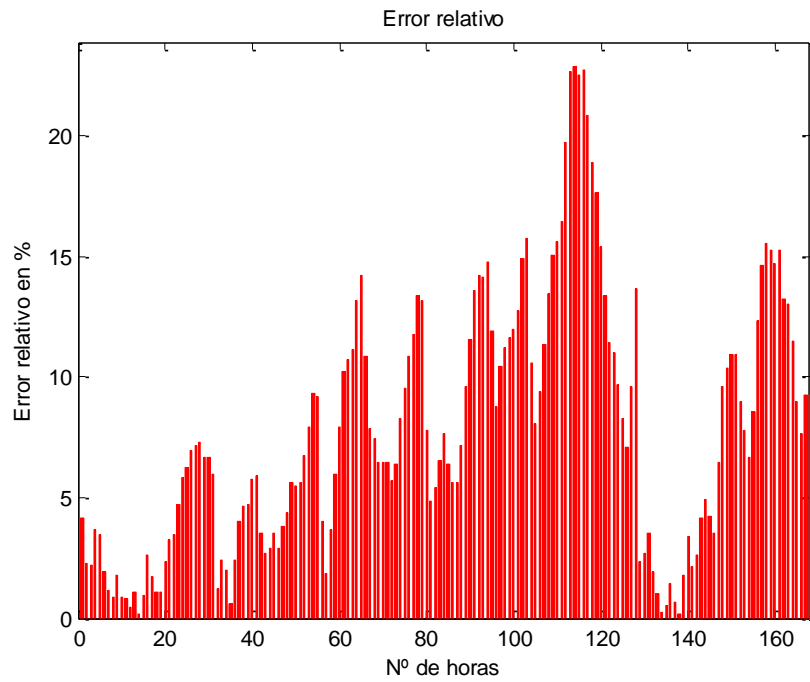


Figura 6.6 Error relativo con el grupo de entrenamiento 2

Error medio: 7.7891

Desviación típica de la estimación: 164.5397 kW

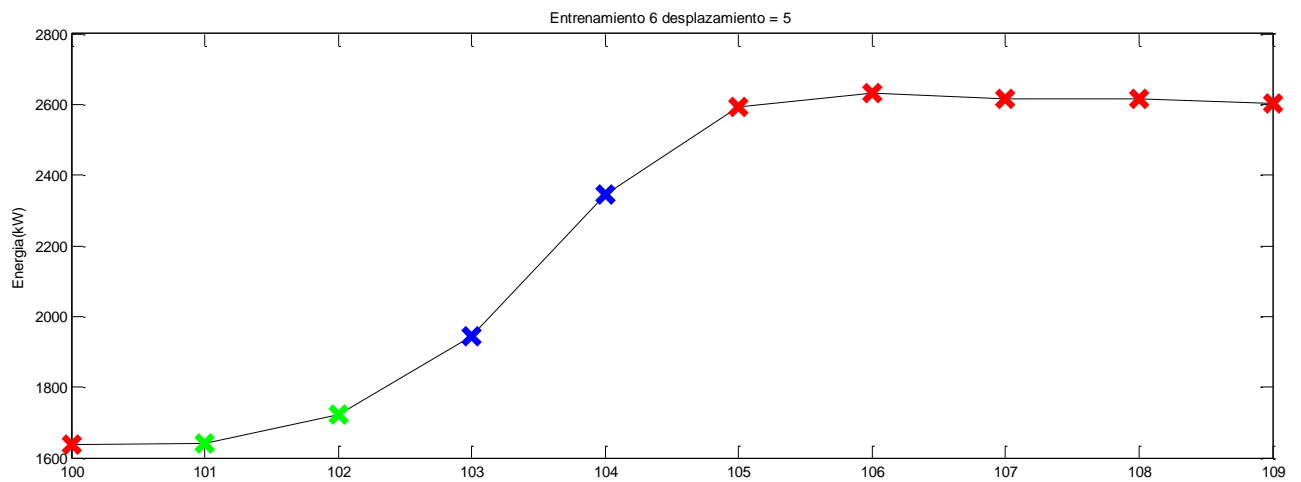


Figura 6.7 Grupo 6 de entrenamiento. Rojo (Training set), Verde (Validation Set) Azul (Test set)

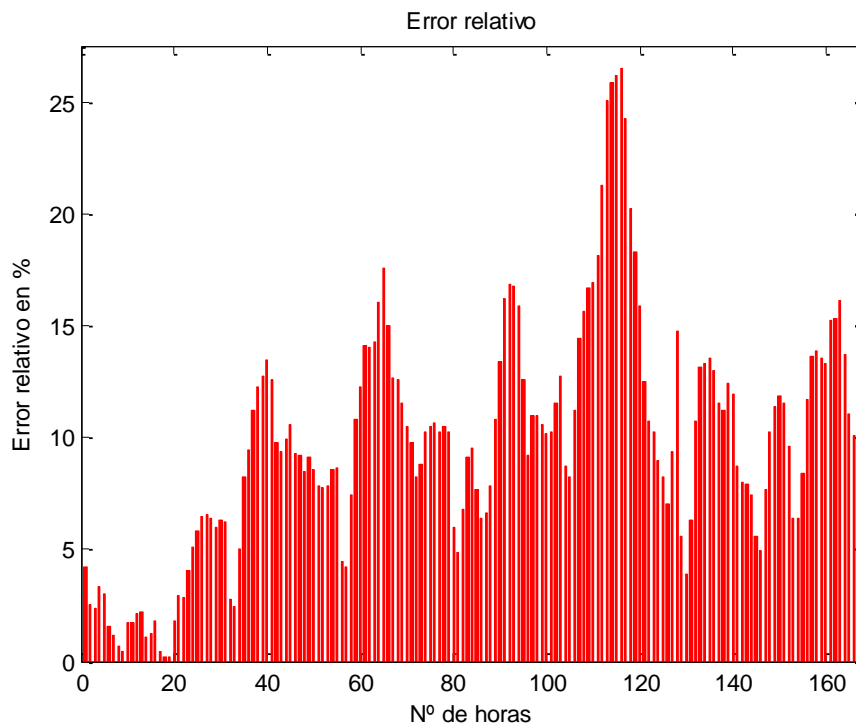


Ilustración 6.3-1 Error relativo con el grupo de entrenamiento 6

Error medio: 9.8024

Desviación típica de la estimación: 200.4662 kW

Existe un pequeño problema con los datos horarios. Como viene siendo habitual desde hace años, en marzo y octubre se adelanta y retrasa la hora respectivamente. Esto se traduce en que en marzo falta una hora de demanda.

Además, podemos pensar que dicho cambio afectaría al consumo o a la configuración de los datos de entrada o la salida. Sin embargo, el mero hecho de que se cambie la hora no puede modificar en gran medida la curva de la demanda ya que, por ejemplo, nos levantaremos a la misma “hora” se atrase o se adelante la misma, o la actividad industrial (en este caso hospitalaria) se adelante o retrase una hora (las actividades se desplazan en tiempo absoluto, pero no en el relativo). Siempre se empezará a la misma hora numérica aunque sea una hora “real” antes o después que el día anterior.

Para solucionar este problema se ha decidido modificar los datos de los meses de octubre y marzo manualmente, en el primer caso se elimina un dato, y en el segundo caso se añade otro, los datos introducidos, aunque en principio pudiera parecer un problema el hecho de añadir o quitar datos, el problema queda subsanado con el filtrado previo de los datos de potencia, que realizará una aproximación de los datos que faltan, lo que nos ahorrará el tener que crear funciones especiales solo para solucionar este inconveniente. A continuación, explicaremos en que consiste este filtrado.

6.4. SIMULACIONES.

6.4.1. DIFERENCIAS ENTRE EL SUAVIZADO DE SPLINES Y EL SUAVIZADO MEDIANTE EL FILTRO DE GOLAY-SAVITZKY.

Con el filtrado y suavizado de las curvas se busca que la curva de entrada sea más uniforme, con este suavizado se eliminarán los puntos anómalos que puedan ser debidos a sucesos puntuales, poco corrientes o a fallos en los aparatos de medida. También se eliminan las pequeñas fluctuaciones que no aportan ninguna ayuda y que incluso, pueden llegar a provocar un mayor tiempo de cálculo o una predicción errónea del consumo.

Como ya se ha tratado antes en este proyecto se utilizan dos algoritmos para el suavizado de la curva de potencia original:

- Splines Cúbicos
- Algoritmo de Savitzky-Golay

6.4.1.1. *Splines Cúbicos.*

En el punto 4.1.2 se puede encontrar toda la teoría y el desarrollo de la elección del suavizado mediante el spline cúbico. En ese punto se hará una pequeña demostración del porque el grado de suavizado escogido. Se ha de recordar que el suavizado variará en función de un parámetro “p” que puede variar entre 0 y 1. Para ello se muestran a

continuación una serie de figuras en las que se puede observar la variación del suavizado en función del parámetro p y el error entre la curva real y la curva suavizada.

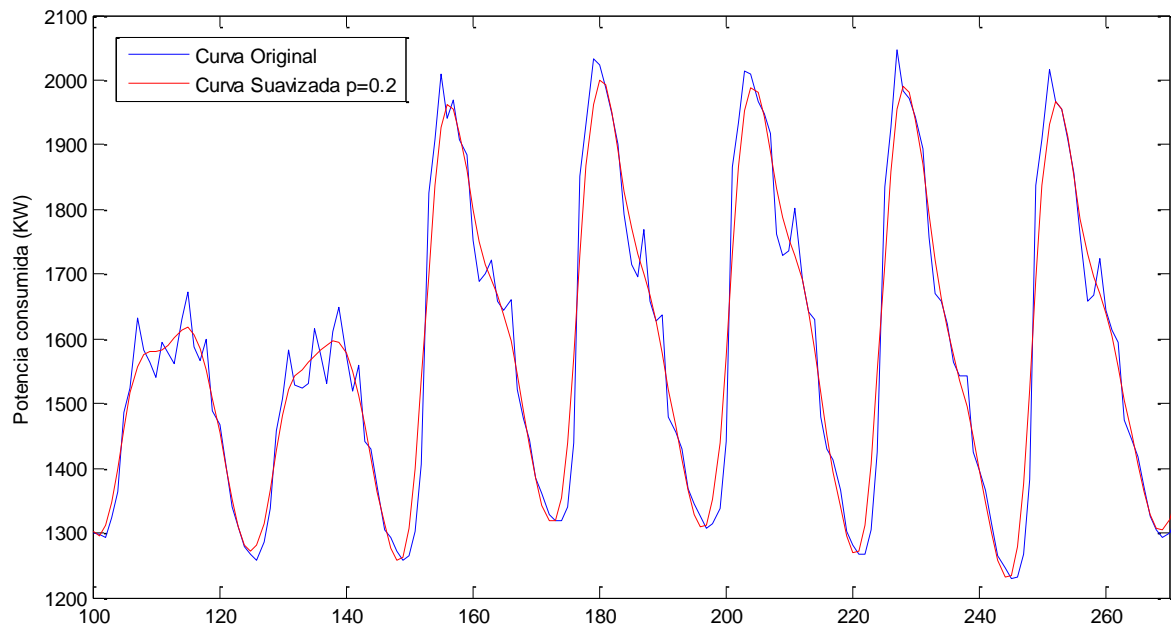


Figura 6.8 Suavizado de la Curva con $p=0.2$

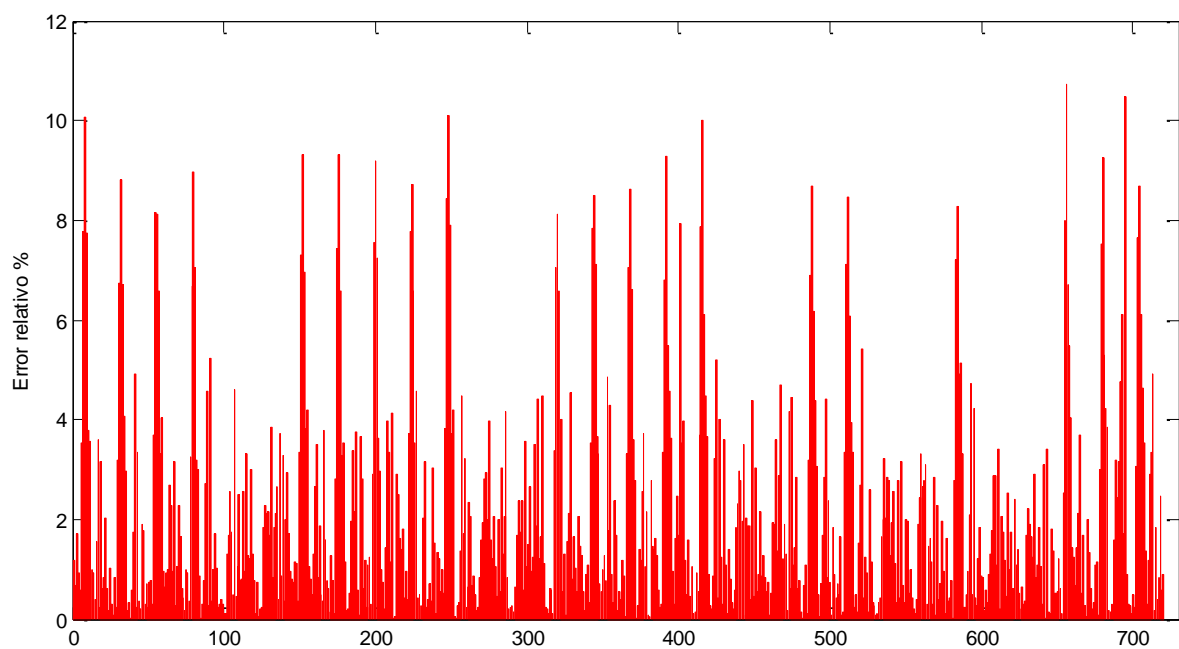


Figura 6.9 Error del suavizado de la curva con $p=0.2$

En Figura 6.8 ($p=0.2$) se puede ver que el suavizado es bastante grande eliminando muchos de los picos característicos. La curva filtrada solo conserva de la curva original una tendencia general, que para este caso se considera que no es suficiente, porque además de la tendencia también se quieren conservar puntos significativos de la curva

de potencia consumida, en la Figura 6.9 se puede observar como en los picos de potencia se alcanzan errores de hasta un 10 %. Correspondientes a la eliminación de los picos.

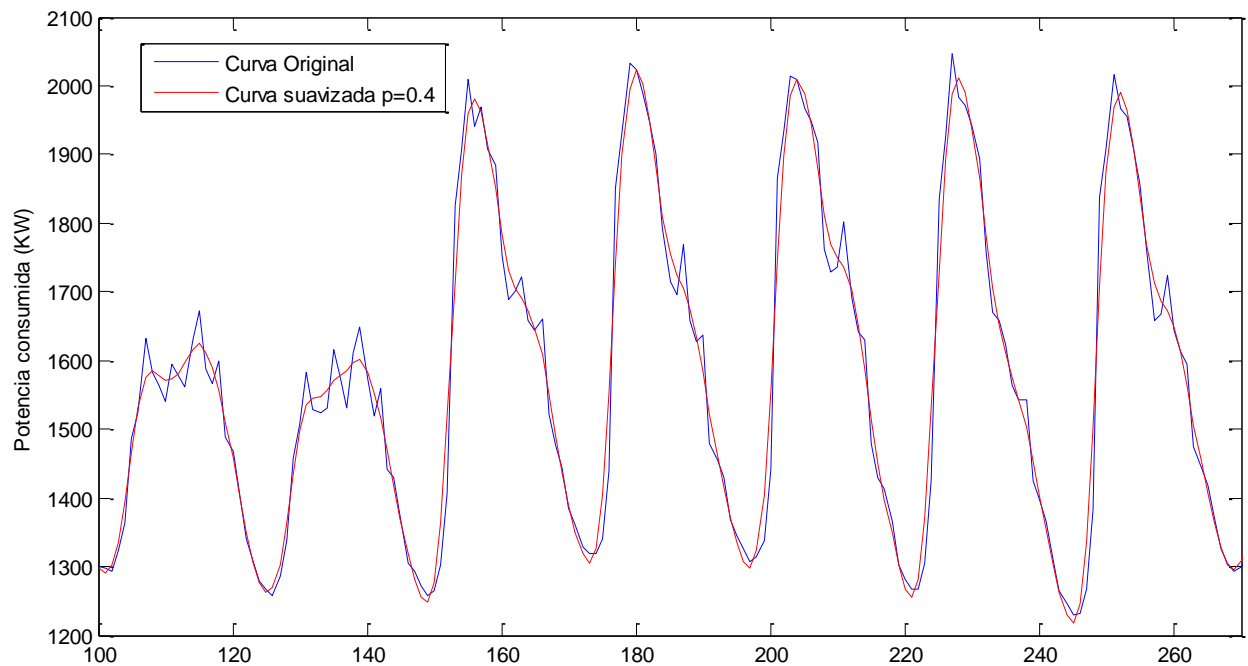


Figura 6.10 Suavizado de la curva con $p=0.4$

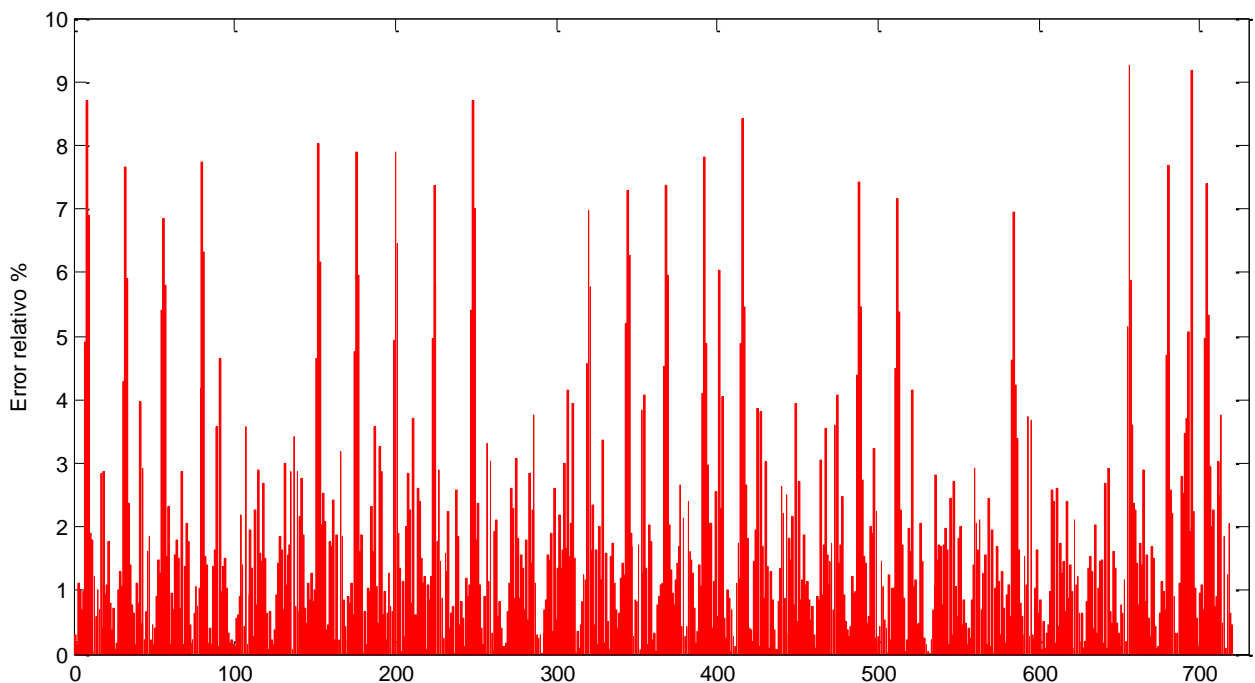


Figura 6.11 Error del suavizado de la curva con $p=0.4$

Como se ve en las Figuras 6.10 y 6.11 ($p=0.4$) se encuentra un suavizado muy similar al que ocurría con $p=0.2$ y por lo tanto se tienen los mismos problemas que

en el caso anterior. Esta vez el error disminuye mínimamente alcanzó una cota de un 9% en los picos máximos de error.

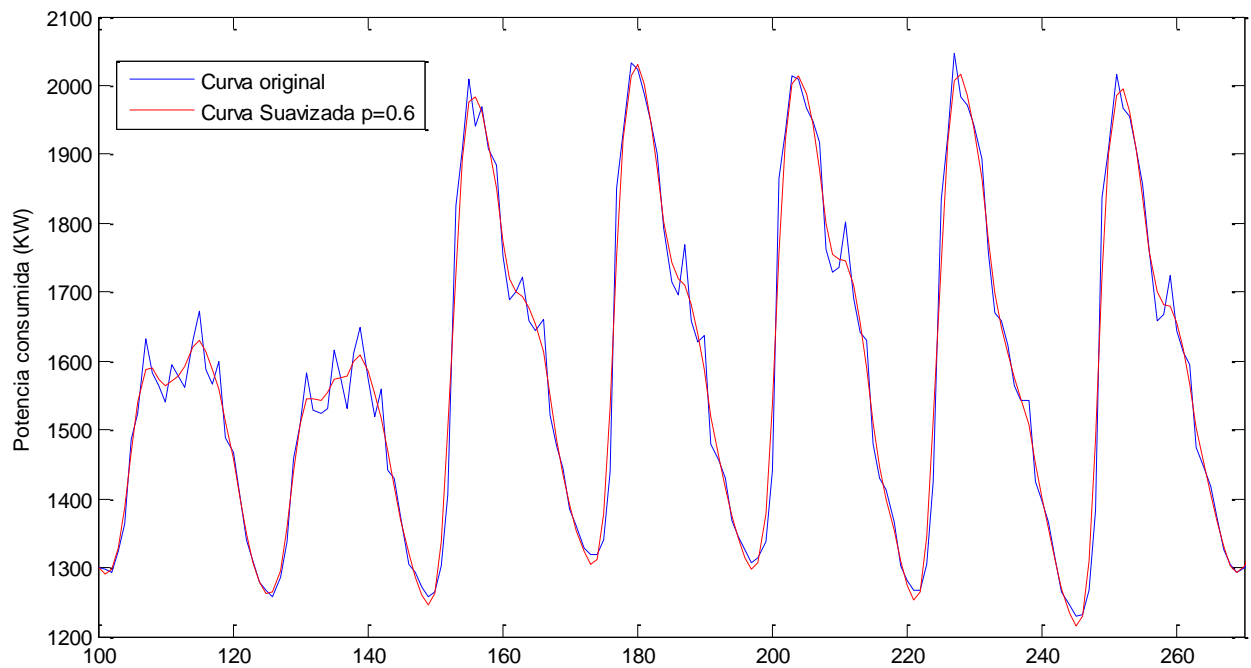


Figura 6.12 Suavizado de la curva con $p=0.6$

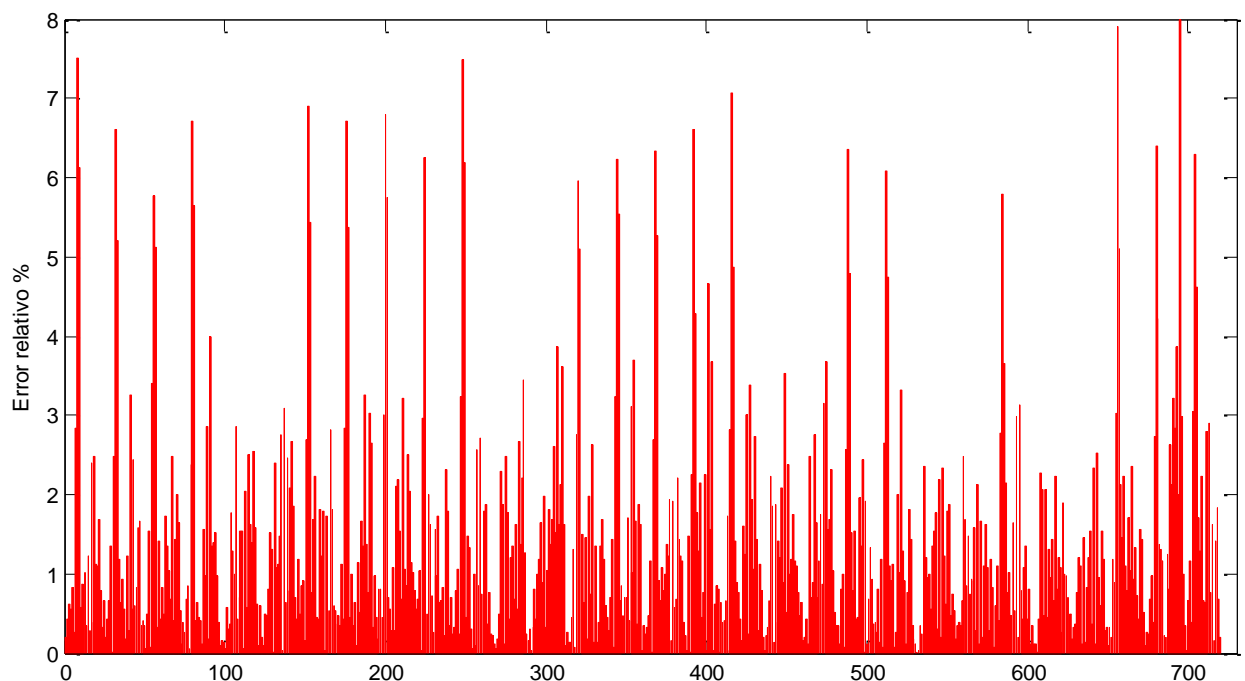


Figura 6.13 Error del suavizado de la curva con $p=0.6$

El suavizado mostrada en las Figuras 6.12 y 6.13 ($p=0.6$) sí que se acerca bastante al suavizado necesario, de hecho se han hecho algunas pruebas con este suavizado y el resultado ha sido satisfactorio, pero con el suavizado de Savitzky-Golay se obtiene una

curva muy parecida y por lo tanto se ha querido utilizar una curva un poco distinta con la técnica de splines.

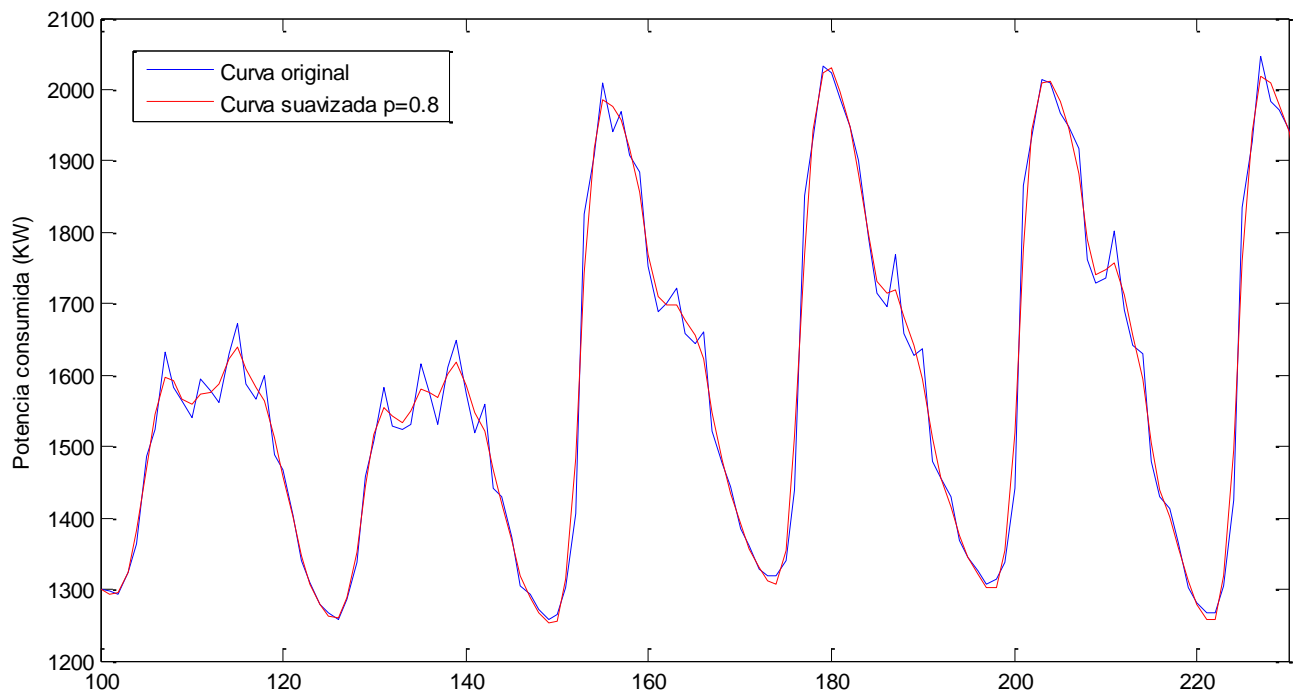


Figura 6.14 Suavizado de la curva con $p=0.8$

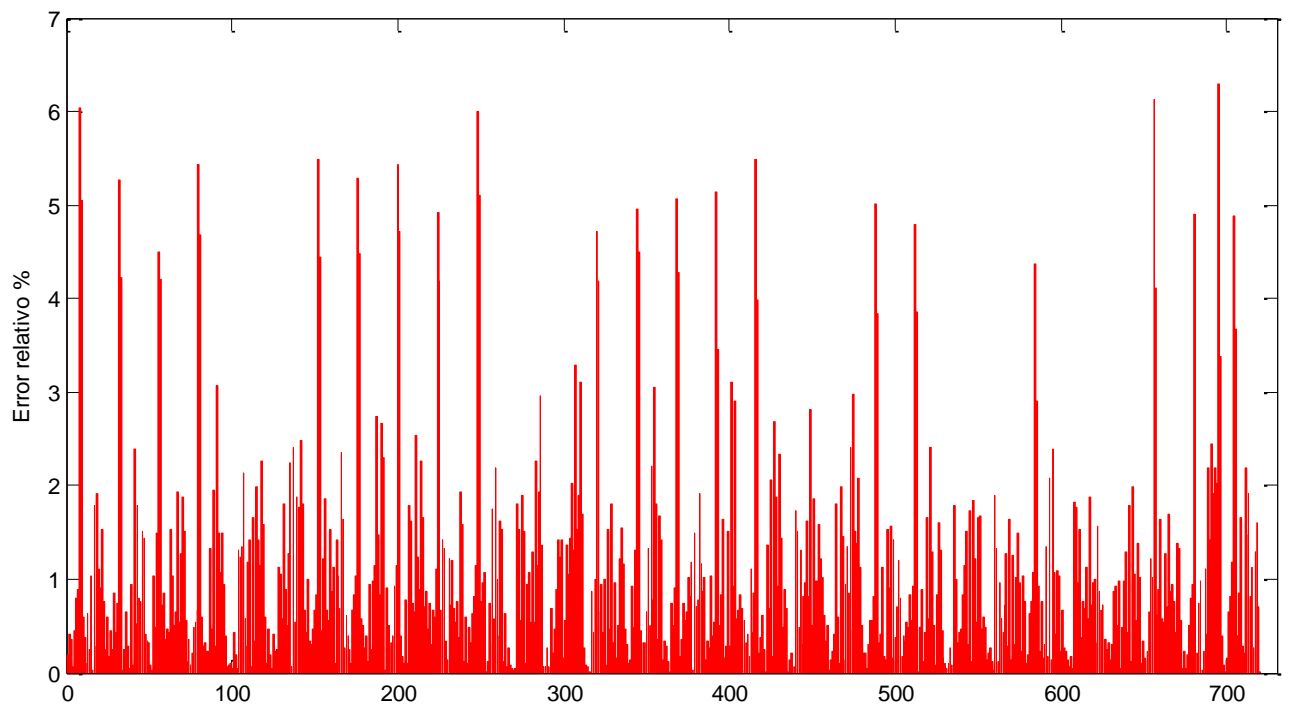


Figura 6.15 Error del suavizado de la curva con $p=0.8$

En las Figuras 6.14 y 6.15 ($p=0.8$) se aprecian unos errores relativos un poquito más altos a los obtenidos con la técnica de Savitzky-Golay, también utilizado en algunas pruebas con excelentes resultados. Perfectamente válido para suavizar la curva original.

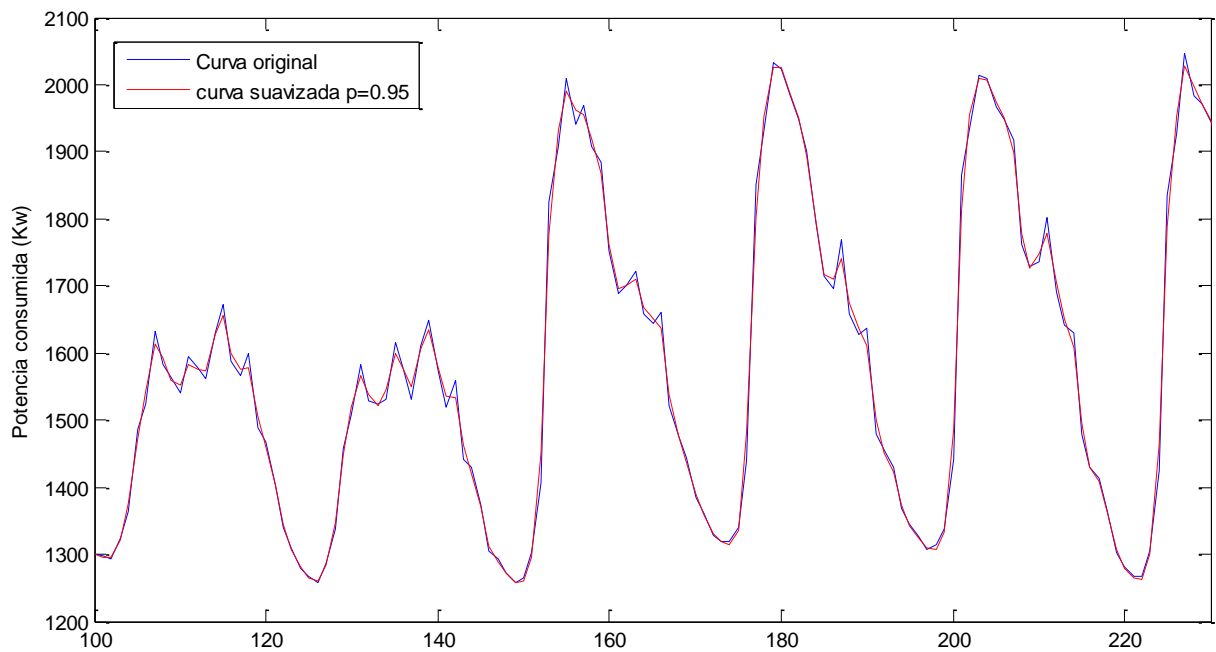


Figura 6.16 Suavizado de la curva con $p=0.95$

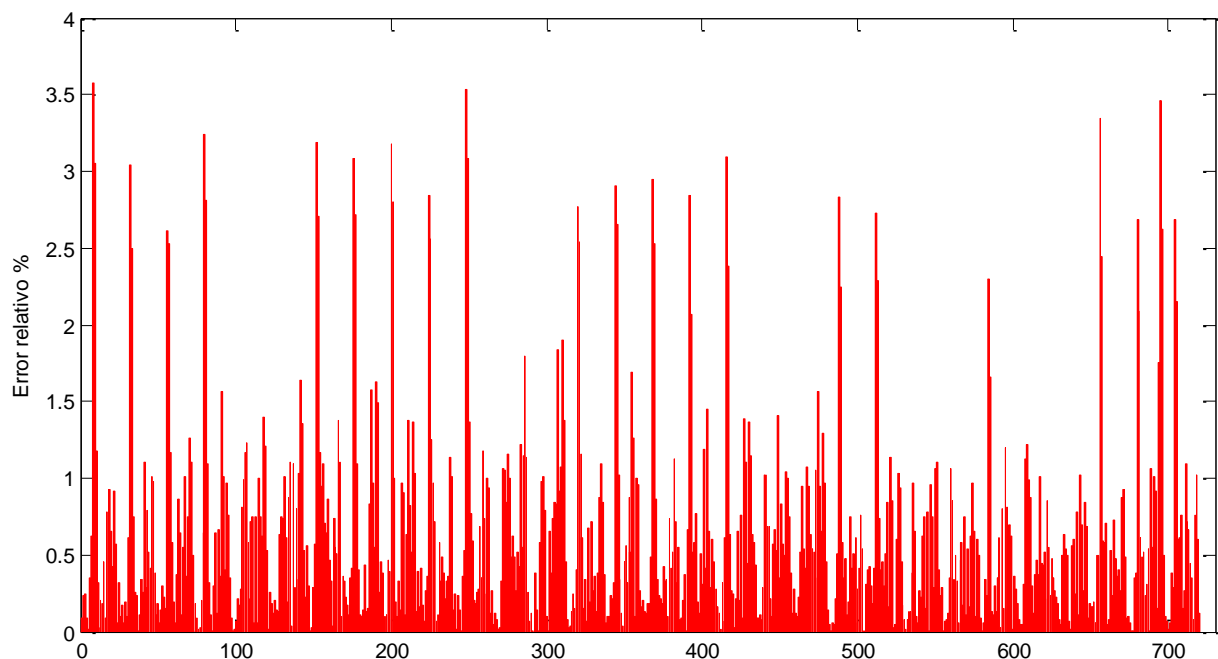


Figura 6.17 Error del suavizado de la curva con $p=0.95$

El suavizado mostrado en las Figuras 6.16 y 6.17 ($p=0.95$) ha sido el elegido como el suavizado a realizar mediante la técnica de interpolación con Splines, como ya se justificó en el apartado 4.1.1 se consigue eliminar gran parte de la diferencia que existe

entre picos de potencia de extremos opuestos y por lo tanto beneficia de gran manera al funcionamiento de la red neuronal.

6.4.1.2. Savitzky-Golay

En el 4.1.2 se justificó la elección del suavizado del filtro de Savitzky-Golay con un polinomio de grado 3 y un ancho de ventana 6. Esta elección se ha basado en el método de prueba-error simulando varias semanas, de varias estaciones distintas, con distintos grados de polinomio y distintos anchos de ventana. La solución de compromiso óptima encontrada entre la carga computacional y el error fue la mencionada anteriormente. Se ha optado por exponer únicamente la solución escogida y recordar un poco como variaba el suavizado en función del ancho de ventana.

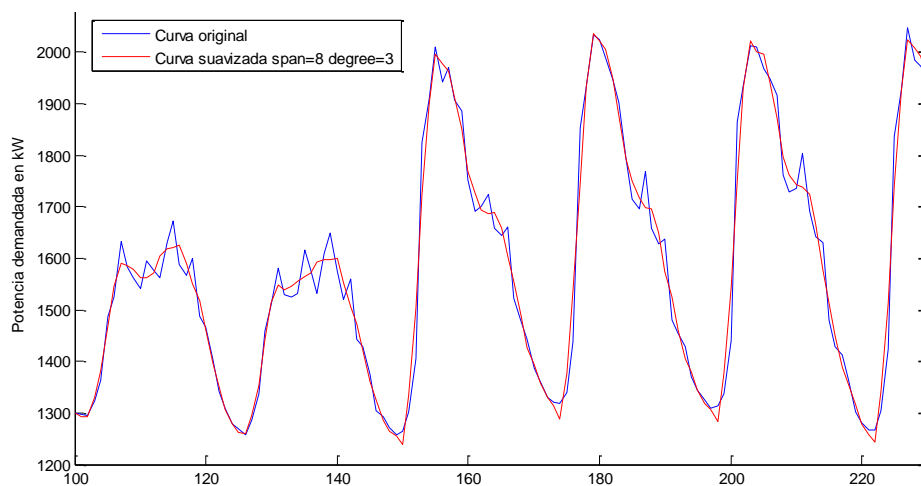


Figura 6.18 Suavizado de la curva Ancho de ventana=8 Grado del polinomio =3

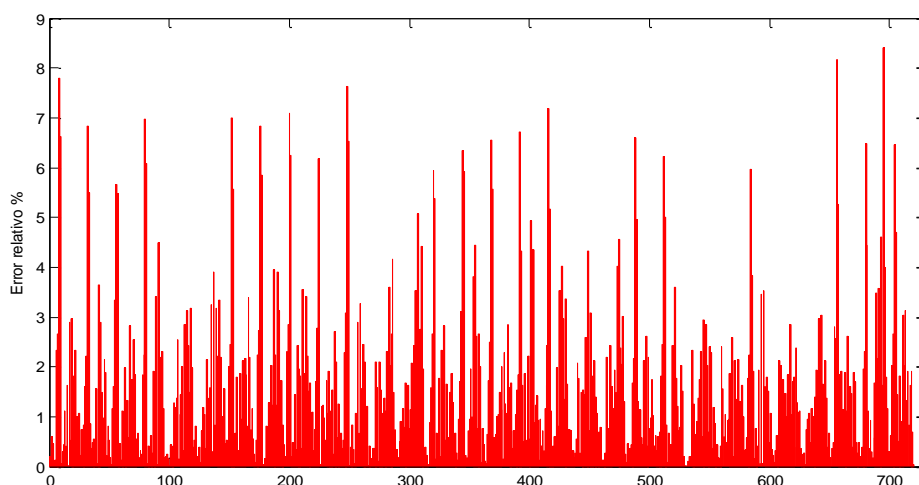


Figura 6.19 Error del suavizado de la curva Ancho de ventana=8 Grado del polinomio =3

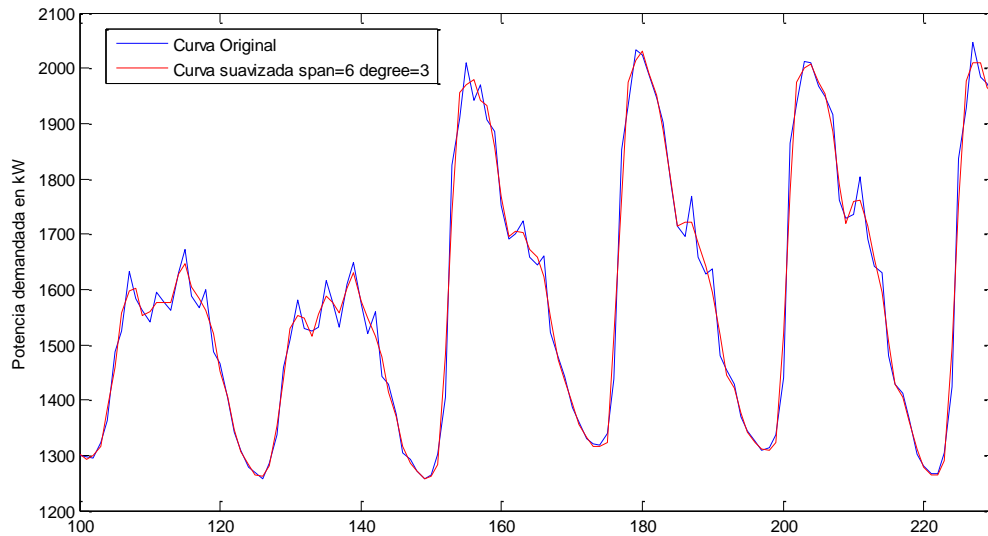


Figura 6.20 Suavizado de la curva Ancho de ventana=6 Grado del polinomio =3

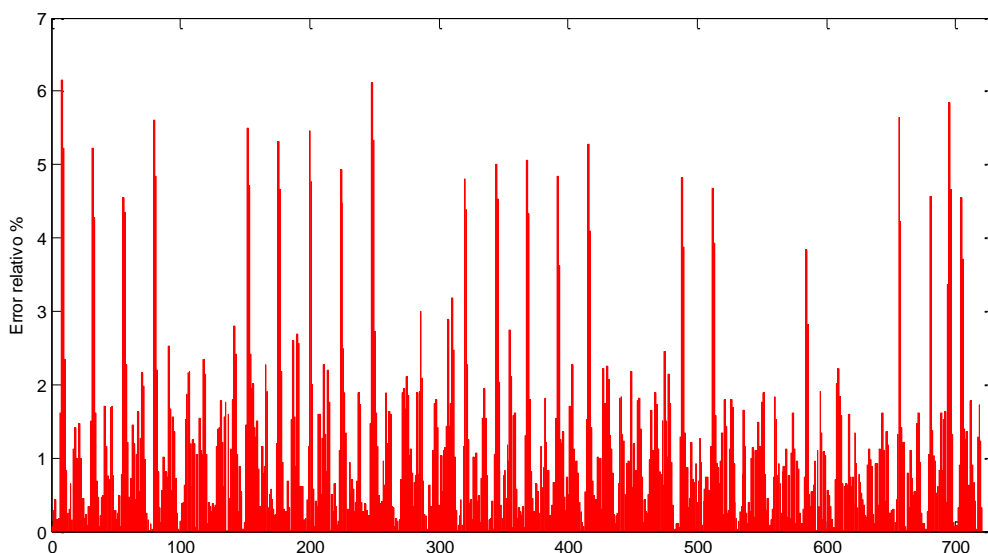


Figura 6.21 Error del suavizado de la curva Ancho de ventana=6 Grado del polinomio =3

Las figuras anteriores muestran el resultado de dos suavizados distintos mediante el método de Savitzky-Golay, la primera simulación se ha realizado con un ancho de ventana de 6 datos y un polinomio de segundo grado, mientras que en la segunda simulación se ha vuelto a utilizar la misma ventana de datos pero esta vez con un polinomio cúbico

Se puede ver como las curvas de las Figuras 6.18 (Ancho de ventana 6 grado del polinomio 2) y 6.20 (Ancho de ventana 6 , grado de polinomio 3) obtenidas mediante el suavizado de Savitzky Golay se asemejan mucho a las curvas obtenidas mediante el

suavizado de Splines. Entre las dos simulaciones se diferencian en que se consigue un mayor suavizado con un ancho de ventana menor, como ya se expuso anteriormente.

Cualquiera de los dos suavizados valdría perfectamente, pero se ha escogido suavizar mediante ancho de ventana 6 y grado de polinomio 3 ya que como se observar en la Figura 6.21 (Ancho de ventana 6 , grado de polinomio 3) se obtienen unos picos de errores y en general un menor área de error que en la Figura 6.19 (Ancho de ventana 6 grado del polinomio 3).

Se espera que mediante Savitzky-Golay se obtengan mejores resultados, si se comparan las predicciones realizadas con la curva suavizada, si se comparan con la curva real, se esperan unos mejores resultados con el método de suavizado de splines cúbicos:

6.5. SIMULACIONES CON SUAVIZADO DE SPLINES Y CON SUAVIZADO MEDIANTE EL FILTRADO DE SAVITZKY GOLAY.

Antes de mostrar los resultados se ha de aclarar que los errores obtenidos variarán en función de la estación que se esté prediciendo. Aquellas estaciones donde las variaciones de temperatura sean mayores obtendrán unos resultados mucho más dispersos, es decir, simulando y prediciendo la misma semana con grupos de entrenamiento diferentes, se obtendrán bastantes diferencias entre los errores, de ahí que sea crucial simular y predecir la mismas semana con distintos grupos de entrenamiento.

Para poder evaluar los errores se ha optado por mostrarlos en boxplot, cuyas características se observan en la siguiente imagen.

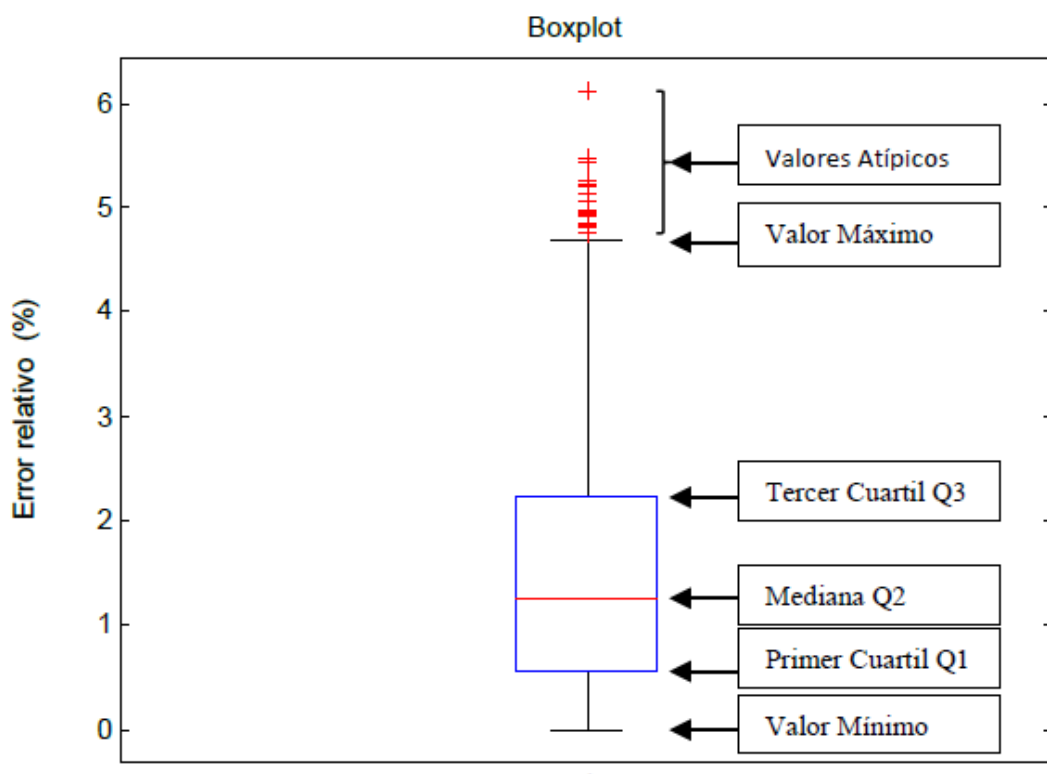


Figura 6.22 Boxplot

De esta forma se mostraran los 10 boxplot correspondientes a los 10 grupos de entrenamiento en la predicción de una semana, y se podrá evaluar la diferencia entre ellos y una valoración general.

El error relativo se ha considerado respecto de la curva suavizada, aunque también se ha hallado numéricamente el error relativo respecto a la curva real. Estos errores de definen a continuación:

$$\text{Error relativo} = \frac{\text{Potencia Predicha} - \text{Potencia de la Curva suavizada}}{\text{Potencia de la curva suavizada}}$$

$$\text{Error relativo real} = \frac{\text{Potencia Predicha} - \text{Potencia de la curva sin suavizar}}{\text{Potencia de la curva sin suavizar}}$$

Además se calcula la desviación media de la Potencia predicha respecto a la real de la curva suavizada:

$$\text{Desviación}(KW) = \sqrt{\frac{\sum_{i=1}^n (\text{Demanda predicha}(i) - \text{Demanda real}(i))^2}{n}}$$

Dada la aleatoriedad de las estimaciones presentada por la red neuronal cada vez que es entrenada, aunque el conjunto de entrenamiento sea el mismo, para hacer las estimaciones, se repetirá el proceso de entrenar y estimar **doce veces** y se tomará como estimación la media de las doce estimaciones obtenidas. De esta forma, aunque la red presente resultados estables y regulares, se minimizarán las posibles desviaciones o estimaciones extraordinarias debidas a la aleatoriedad de la elección del conjunto real de entrenamiento y la minimización numérica de la función del error.

En total se van a mostrar 4 semanas estimadas, una de otoño, una de primavera, una semana de Invierno, y otra de Verano, donde se quiere evidenciar las diferencias de exactitud obtenidas dependiendo de la estación y de sus condicionantes.

Tabla 3 Semanas simuladas

Semana estimada	Semana anterior	Mes de entrenamiento	Semana a estimar	Año
Otoño	25 -31 Agosto	1-28 Septiembre	29 Septiembre 5 Octubre	2008
Invierno	5 -11 Enero	12 Enero 14 febrero	9 – 15Febrero	2009
Primavera	9-15 Marzo	16 Marzo	13-19 Abril	2009

		12 Abril		
Verano	22-28 Junio	29 Junio	27 Julio	2009
		26 Julio	2 Agosto	

6.6. RESULTADOS GOLAY –SPLINES

Como ya se ha comentado se mostrarán los resultados de la simulación de cuatro semanas. El formato será el siguiente. Inicialmente se hará un pequeño comentario de los resultados obtenidos mediante las simulaciones en las 4 estaciones del año. Se harán referencia a las temperaturas, errores, etc. A continuación se muestran las simulaciones obtenidas en cada semana mostrada para cada estación, primero se mostrará una tabla con los datos numéricos y a continuación se mostrará una imagen donde se muestren los boxplot de cada grupo de entrenamiento.

6.6.1. OTOÑO.

Por las simulaciones que se han realizado se han obtenido unos resultados dispares ,en la mayoría de semanas los resultados han sido relativamente buenos con errores relativos bajos <5-7%,. También hubo semanas con errores relativos bastante por encima de la media de otoño. Probablemente estos resultados vienen provocados por las condiciones climatológicas de otoño con semanas con bruscos ascensos y descensos de temperaturas, como se pueden observar en las Figuras 6.23 y 6.24.

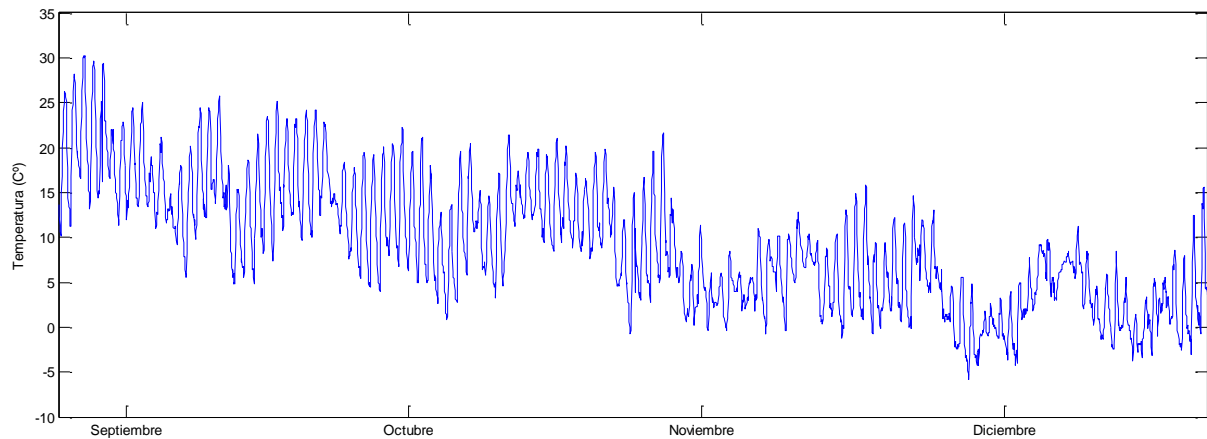


Figura 6.23 Temperaturas otoño

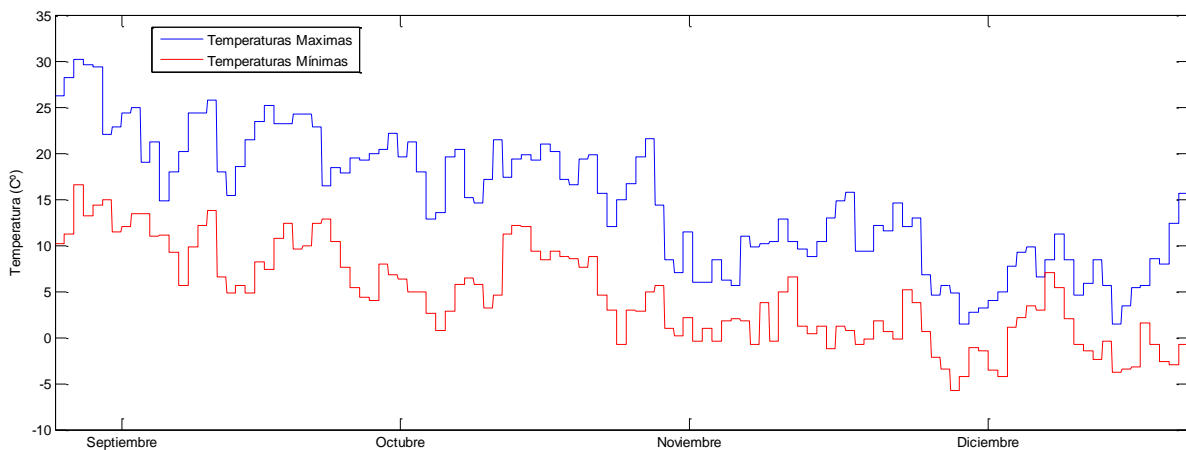


Figura 6.24 Temperaturas máximas y mínimas otoño

6.6.1.1. Splines Cúbicos

A continuación se muestran las simulaciones correspondientes a la semana mostrada de Otoño cuyo suavizado se ha realizado mediante la interpolación de splines cúbicos.

Tabla 4 Errores obtenidos con el suavizado de splines en la semana del 29 de Septiembre al 6 de Octubre

Grupo entrenamiento	1	2	3	4	5	6	7	8	9	10
Error relativo respecto a datos suavizados (%)	1,52	1,46	1,56	1,51	1,39	1,41	1,58	1,40	1,50	1,49
Error relativo respecto a datos reales (%)	1,62	1,666	1,66	1,52	1,57	1,70	1,61	1,65	1,66	1,62
Desviación típica (Kw)	34,84	34,07	37,63	35,88	32,96	33,12	36,60	30,49	33,53	34,45

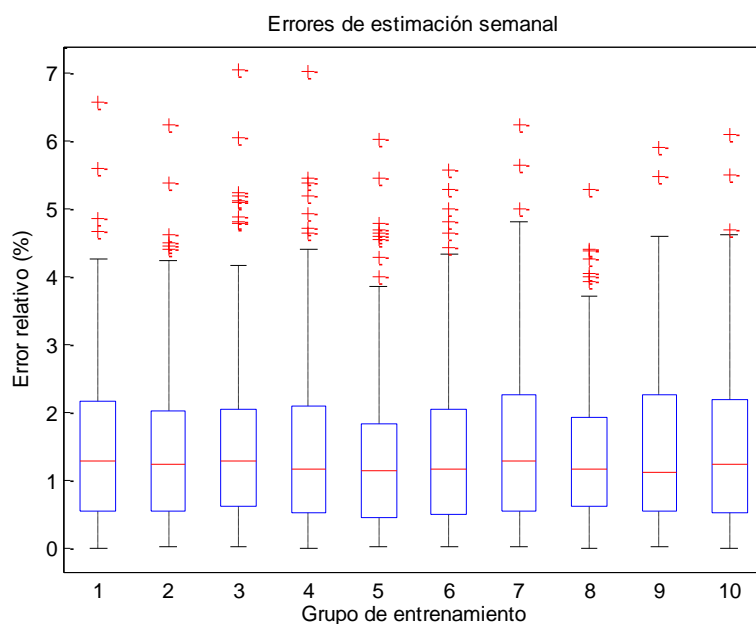


Figura 6.25 Resultados obtenidos mediante el suavizado de splines en la semana del 29 Septiembre al 5 de Octubre

Como se ve en la Figura 6.25 y en la Tabla 4 los errores son bastante bajos. Existen bastantes desviaciones típicas en cada uno de los grupos de entrenamiento obteniéndose para cada grupo un resultado bastante similar. Es una de las mejores semanas de otoño.

6.6.1.2. Savitzky-Golay

A continuación se muestran las simulaciones correspondientes a la semana mostrada de Otoño cuyo suavizado se ha realizado mediante la interpolación de Savitzky-Golay, como se observará en la Figura 6.26 y en la Tabla 5, los resultados obtenidos son muy similares a los que se mostraron en el apartado anterior.

Tabla 5 Errores obtenidos con el suavizado de Golay en la semana del 29 de Septiembre al 6 de Octubre

Grupo entrenamiento	1	2	3	4	5	6	7	8	9	10
Error relativo respecto a datos suavizados (%)	1,81	1,67	1,58	1,54	1,74	1,57	1,68	1,71	1,67	1,61
Error relativo respecto a datos reales (%)	2,06	1,96	1,85	1,82	1,95	1,87	1,92	1,97	1,95	1,95
Desviación típica (Kw)	42,70	39,46	37,21	37,27	43,75	37,47	38,22	39,64	38,87	36,21

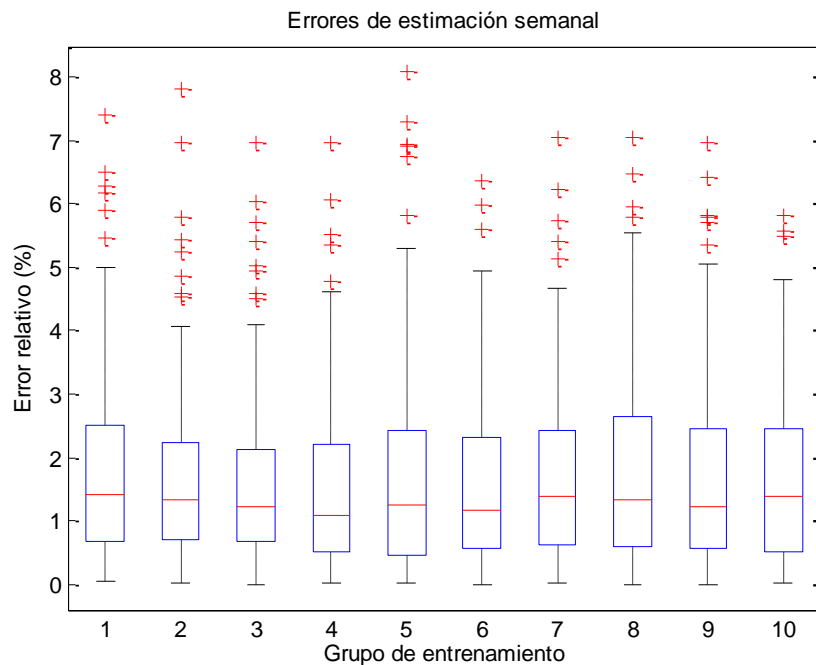


Figura 6.26 Resultados obtenidos mediante el suavizado de splines en la semana del 29 Septiembre al 5 de Octubre

Los resultados que muestra la Figura 6.26 son parecidos al caso de la Figura 6.25 , y al contrario de como se preveía la Tabla 5 (Suavizado Golay) muestra un error mayor en el error relativo respecto los datos suavizados que el error mostrado en la Tabla 4 (Suavizado mediante splines)

6.6.2. INVIERNO

Las predicciones arrojan errores relativos bajos, todos los resultados de las estimaciones que se han hecho durante invierno son muy parecidos al contrario que sucedía en otoño, donde si había mucha más variedad, en la época de invierno además, se obtienen muy pocas desviaciones típicas. Durante el mes de Enero, sí que se han obtenido unos resultados peores que durante el resto de la época. Lo resultados, en general pueden ser debidos a la constancia de las temperaturas durante el periodo como se aprecian en las Figuras 6.27 y 6.28.

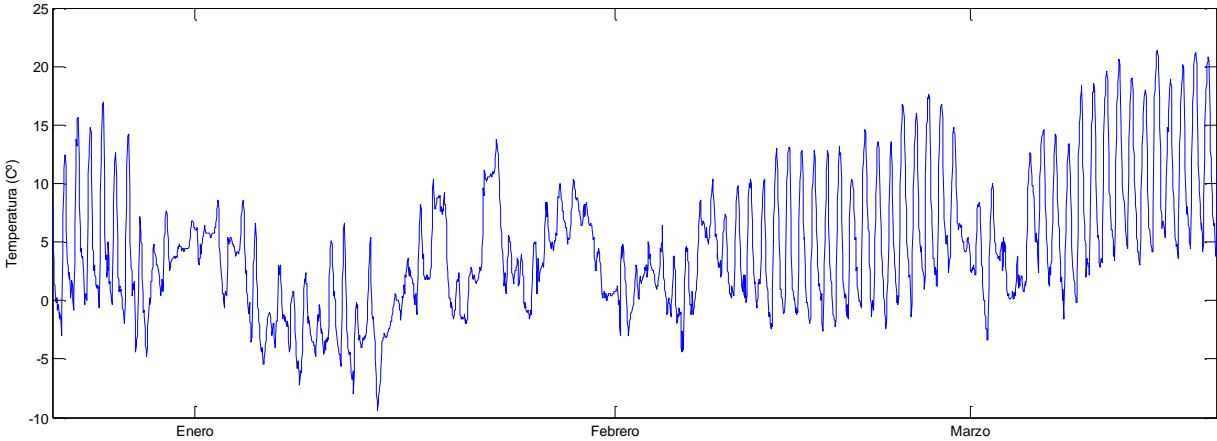


Figura 6.27 Temperaturas Invierno

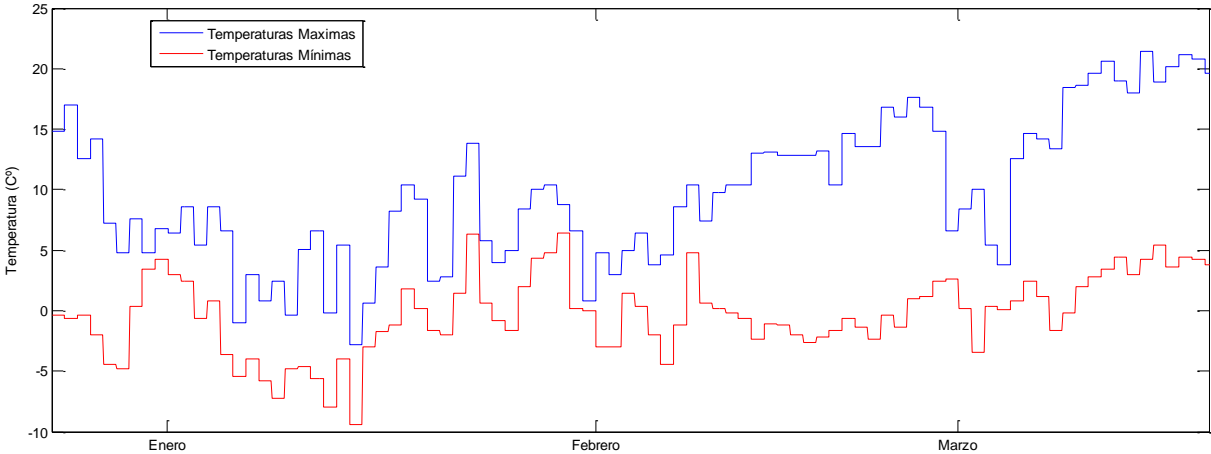


Figura 6.28 Temperaturas máximas y mínimas invierno

Si se comparan estas gráficas con las de otoño se puede ver como son unas gráficas mucho más suaves y horizontales que es uno de los motivos por el cual se esperan unos resultados muy regulares a lo largo del Invierno.

6.6.2.1. Splines Cúbicos

A continuación se muestran los resultados obtenidos para la semana de invierno (9-15 Febrero) con el suavizado de Splines.

Tabla 6 Errores obtenidos con el suavizado de Splines del 9 al 15 de Febrero

Grupo entrenamiento	1	2	3	4	5	6	7	8	9	10
Error relativo respecto a datos suavizados (%)	2,55	2,71	2,32	2,24	2,23	3,06	2,44	2,28	2,28	2,35
Error relativo respecto a datos reales (%)	2,63	2,79	2,44	2,35	2,34	3,27	2,61	2,39	2,39	2,43
Desviación típica (Kw)	55,13	58,47	50,52	47,72	55,29	67,49	52,86	57,20	55,50	56,71

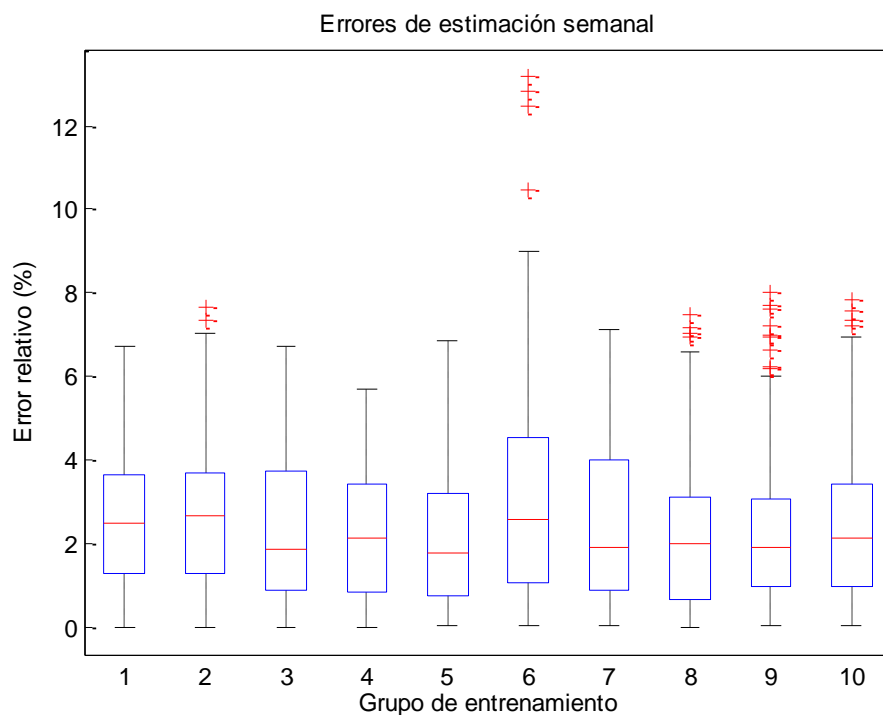


Figura 6.29 Resultados obtenidos mediante el suavizado de splines en la semana del 9 al 15 de Febrero

En la Figura 6.29 y Tabla 6 se puede observar lo hablado anteriormente, unos errores relativos bajos < 3% y muy pocas desviaciones típicas, se puede afirmar que la red neuronal trabaja bien en el periodo de invierno.

6.6.2.1. Savitzky-Golay

Ahora se muestran los resultados obtenidos para la semana de invierno con el suavizado de Savitzky-Golay.

Tabla 7 Errores obtenidos con el suavizado de Golay del 9 al 15 de Febrero

Grupo entrenamiento	1	2	3	4	5	6	7	8	9	10
Error relativo respecto a datos suavizados (%)	2,55	2,78	1,62	2,01	2,62	2,30	1,84	1,92	2,19	2,36
Error relativo respecto a datos reales (%)	2,82	3,09	1,85	2,298	2,95	2,60	2,23	2,20	2,43	2,69
Desviación típica (Kw)	56,46	56,30	34,80	45,06	56,44	53,53	44,54	46,58	49,24	49,01

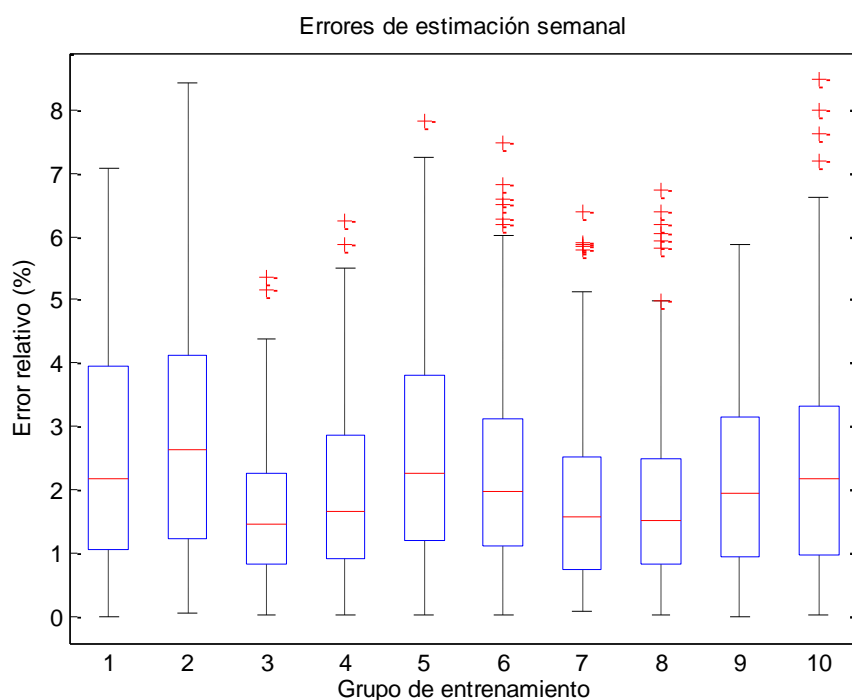


Figura 6.30 Resultados obtenidos mediante el suavizado de Golay en la semana del 9 al 15 de Febrero

Los resultados de la Figura 6.30 son muy parecidos a los obtenidos en el caso de suavizado mediante Splines cúbicos. En alguna caso como con el grupo de entrenamiento 3, sí que se observa un resultado mucho mejor que el dado en el caso anterior, no deja de ser una caso puntual en el que se habrán dado unas circunstancias adecuadas para que la red neuronal trabaje mejor. En este caso los errores relativos respecto a la curva sí que son más bajos los mostrados en la Tabla 7 (Golay) que en la Tabla 6 (splines).

6.6.3. PRIMAVERA

Es un periodo en el que en todas las simulaciones realizadas arrojan unos resultados un poco confusos e inconcluyentes para determinar la validez de la red neuronal. Los errores relativos son altos comparados con las predicciones realizadas hasta ahora, hay bastante variedad de resultados entre las distintas semanas e incluso dentro de la misma semana dependiendo del grupo de entrenamiento, se obtienen resultados bastantes diferentes. También se dan muchas desviaciones típicas con valores de errores elevados. Una de las características que pueden justificar estos resultados son las grandes variaciones de temperatura existentes que se observan en las Figuras 6.31 y 6.32. Una de las conclusiones que se pueden sacar en este periodo es la falta de alguna variable mas para poder realizar una mejor predicción de la potencia consumida.

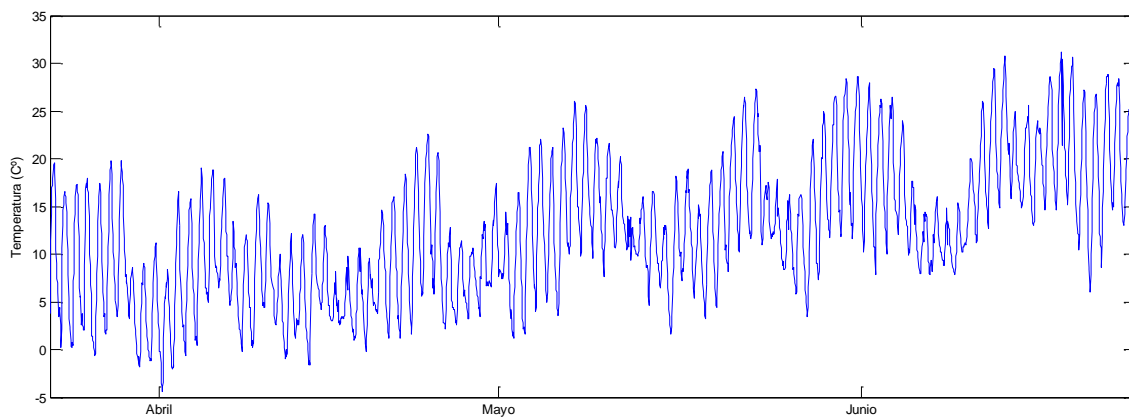


Figura 6.31 Temperaturas primavera

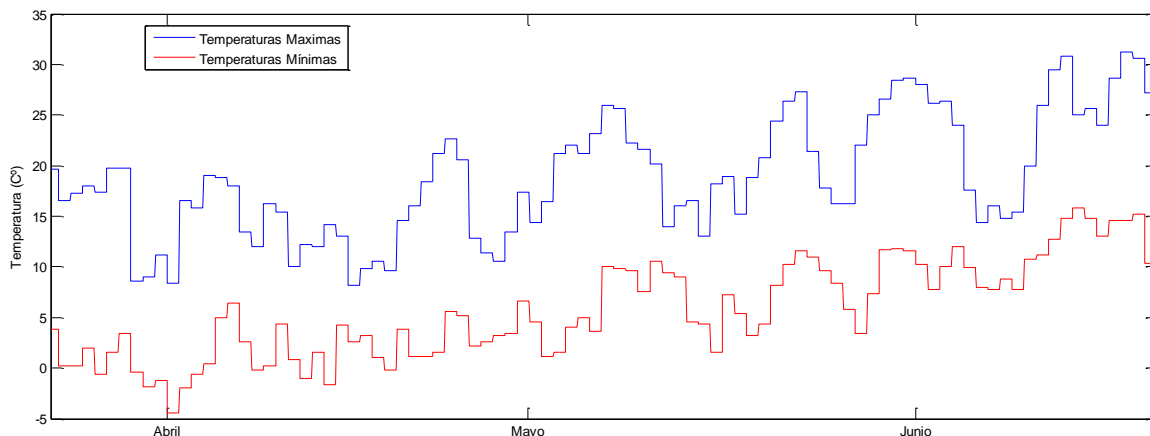


Figura 6.32 Temperaturas máximas y mínimas primavera

En comparación con las gráficas anteriores de otoño e invierno, estas son menos lineales y existe un mayor rango entre las distintas temperaturas máximas y mínimas

6.6.3.1. Splines

En la Tabla 8 y Figura 6.33 se muestran los resultados obtenidos al predecir la semana del 13 al 19 de Abril. El suavizado de las curvas de potencia ha sido realizado mediante la interpolación por splines.

Tabla 8 Errores obtenidos con el suavizado de splines del 13 al 19 de Abril

Grupo entrenamiento	1	2	3	4	5	6	7	8	9	10
Error relativo respecto a datos suavizados (%)	2,83	4,05	3,35	2,99	2,88	2,47	1,77	2,02	2,25	2,25
Error relativo respecto a datos reales (%)	2,90	4,11	3,38	3,04	2,94	2,54	1,86	2,11	2,34	2,36
Desviación típica (Kw)	56,98	76,12	63,35	57,26	56,00	48,71	40,88	44,15	42,04	47,77

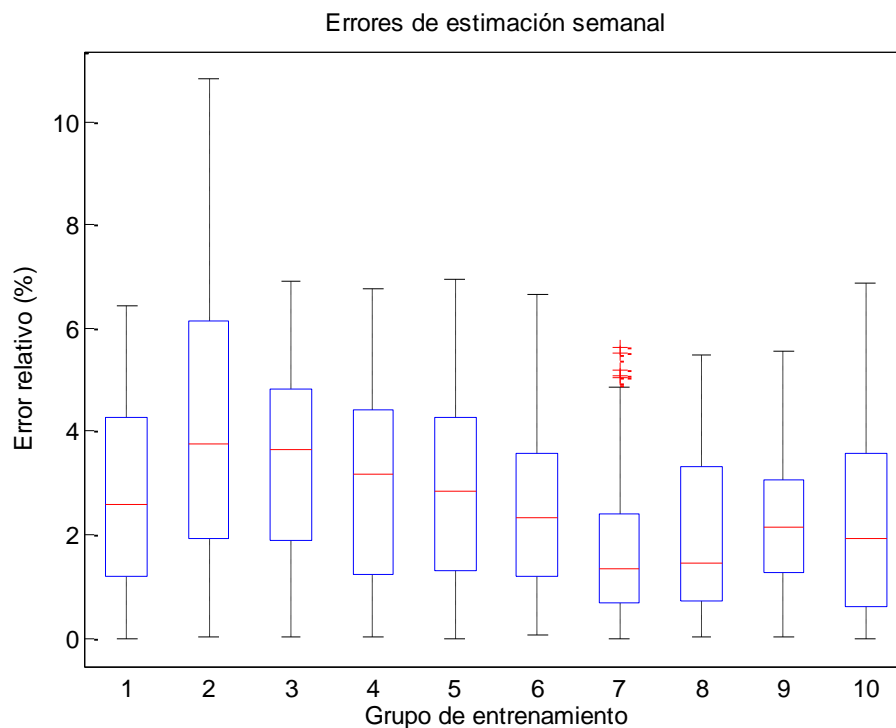


Figura 6.33 Resultados obtenidos mediante el suavizado de splines en la semana del 13 al 19 de Abril

En la Figura 6.33 se aprecia en general un aumento del error relativo en comparación con las dos épocas pasadas, en la misma gráfica se puede apreciar la diferencia de resultados en función de los grupos de entrenamiento. La Tabla 8 muestra unos

resultados dispares en función del grupo de entrenamiento. Como ya se ha comentado, puede ser debido a las variaciones de temperatura de esta estación.

6.6.3.2. Savitzky-Golay

En la Tabla 9 y Figura 6.34 se muestran los resultados obtenidos al predecir la semana del 13 al 19 de Abril. El suavizado de las curvas de potencia ha sido realizado mediante la interpolación por Golay.

Tabla 9 Errores obtenidos con el suavizado de Golay del 13 al 19 de Abril

Grupo entrenamiento	1	2	3	4	5	6	7	8	9	10
Error relativo respecto a datos suavizados (%)	2,89	3,28	2,96	2,45	2,57	2,31	1,98	1,99	2,44	2,65
Error relativo respecto a datos reales (%)	3,02	3,40	3,09	2,57	2,71	2,58	2,24	2,25	2,69	2,85
Desviación típica (Kw)	66,23	66,74	57,52	48,94	55,62	48,09	50,81	44,52	49,35	58,06

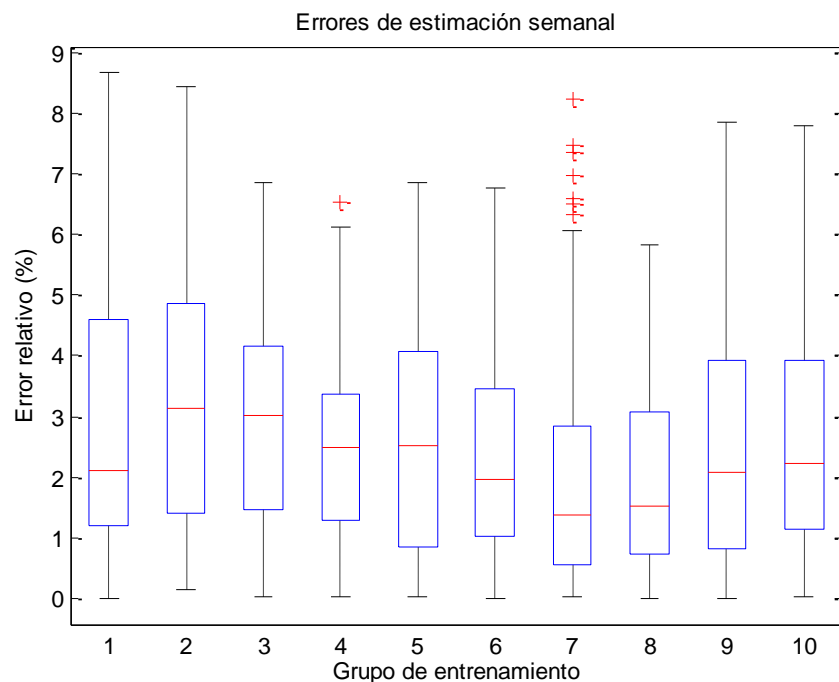


Figura 6.34 Resultados obtenidos mediante el suavizado de Golay en la semana del 13 al 19 de Abril

Los resultados mostrados en la Tabla 9 y en la Figura 3.34 son como siempre, parecidos a los obtenidos con el método anterior, aunque como se ve en la imagen anterior, se

obtiene un rango de errores más elevado que los obtenidos con el algoritmo de suavizado se Savitzky-Golay. En este caso el error relativo respecto de la curva suavizada en algunos grupos de entrenamiento es más bajo mediante el suavizado de splines (Tabla 8) y en otros es más bajo en el suavizado de Golay (Tabla 9).

6.6.4. VERANO

En este periodo en las simulaciones realizadas se obtienen unos errores mayores de los esperados, con mucha variedad de resultados dependiendo de las semanas, aunque dentro de la misma semana sí que se obtienen unos resultados parecidos.

La variación de temperaturas que se pueden observar en las Figuras 6.35 y 6.36 no es ni mucho menos igual que en los meses correspondientes a las épocas de otoño y primavera, pero los resultados sí que son parecidos en muchos casos en cuanto a lo elevado del error. Por lo que se extrae una conclusión: aunque la temperatura pueda ayudar a mejorar una predicción para lograr una mejor predicción, hacen falta más variables para el entrenamiento de la red neuronal si se quieren lograr mejores resultados.

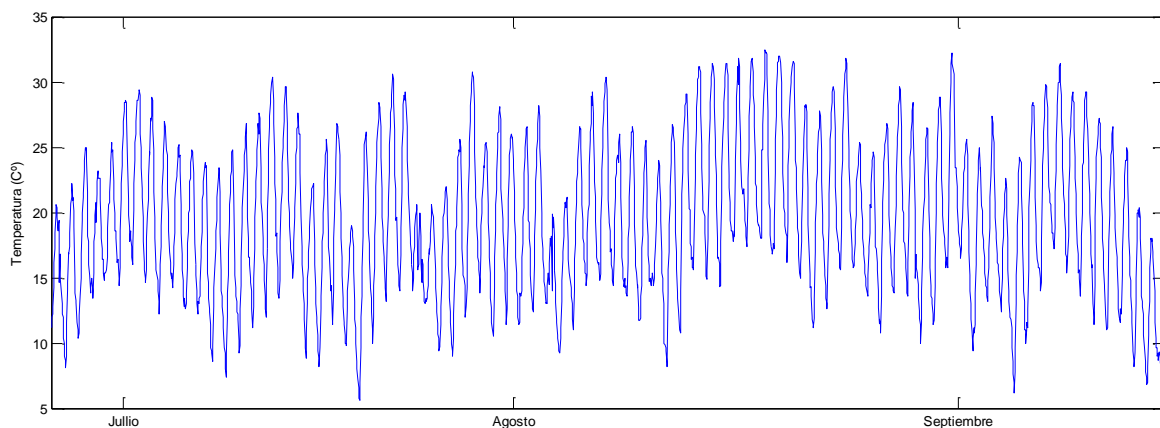


Figura 6.35 Temperaturas verano

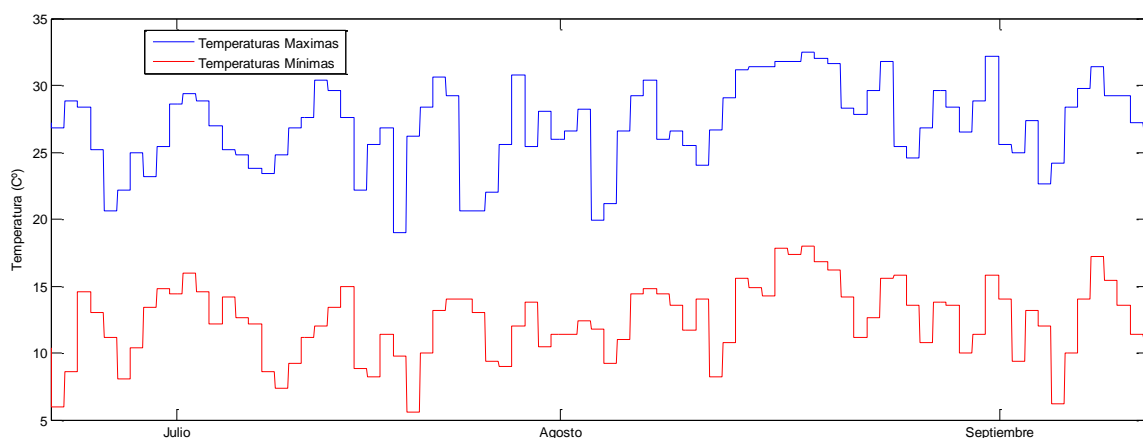


Figura 6.36 temperatura máxima y mínima verano

En las figuras 6.35 y 6.36 las variaciones de temperatura son mas o menos constantes, con periodos de subida de temperaturas y con intervalos de bajada. La diferencia entre la temperatura máxima y mínima es bastante constante a lo largo de todo el periodo.

6.6.4.1. Splines Cúbicos

Los resultados mostrados a continuación pertenecen a la semana que va desde el 27 de Julio al 2 de Agosto. Las curvas de potencia se suavizan mediante el uso de splines cúbicos

Tabla 10 Errores obtenidos con el suavizado de splines en la semana del 29 de Septiembre al 6 de Octubre

Grupo entrenamiento	1	2	3	4	5	6	7	8	9	10
Error relativo respecto a datos suavizados (%)	3,06	2,52	2,38	2,58	2,43	2,99	2,37	2,08	2,13	3,09
Error relativo respecto a datos reales (%)	3,20	2,66	2,46	2,69	2,53	3,11	2,55	2,29	2,33	3,20
Desviación típica (Kw)	67,57	62,01	54,18	58,94	57,81	68,87	56,21	47,74	50,025	66,45

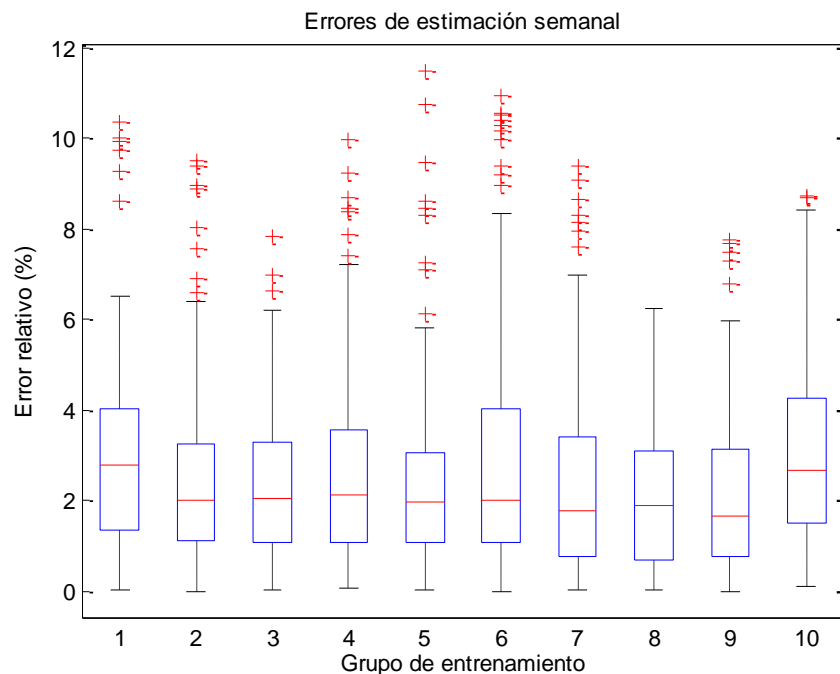


Figura 6.37 Resultados obtenidos mediante el suavizado de splines en la semana del 27 Julio al 2 de Agosto

La semana mostrada, es una en la que se obtienen mejores resultados como se ve en la Tabla 10 los errores son bastante bajos, aunque si se puede observar en la Figura 6. 38 una gran cantidad de desviaciones típicas.

6.6.4.2. Savitzky-Golay

Los resultados mostrados a continuación pertenecen a la semana que va desde el 27 de Julio al 2 de Agosto. Las curvas de potencia se suavizan mediante el uso de la técnica de Savitzky-Golay.

Tabla 11 Errores obtenidos con el suavizado de Golay en la semana del 29 de Septiembre al 6 de Octubre

Grupo entrenamiento	1	2	3	4	5	6	7	8	9	10
Error relativo respecto a datos suavizados (%)	2,86	2,24	2,298	2,49	2,51	2,70	2,27	2,05	2,41	3,09
Error relativo respecto a datos reales (%)	3,09	2,48	2,50	2,71	2,66	2,94	2,59	2,31	2,71	3,30
Desviación típica (Kw)	62,47	54,53	53,316	57,01	57,98	60,81	51,85	44,92	52,29	66,80

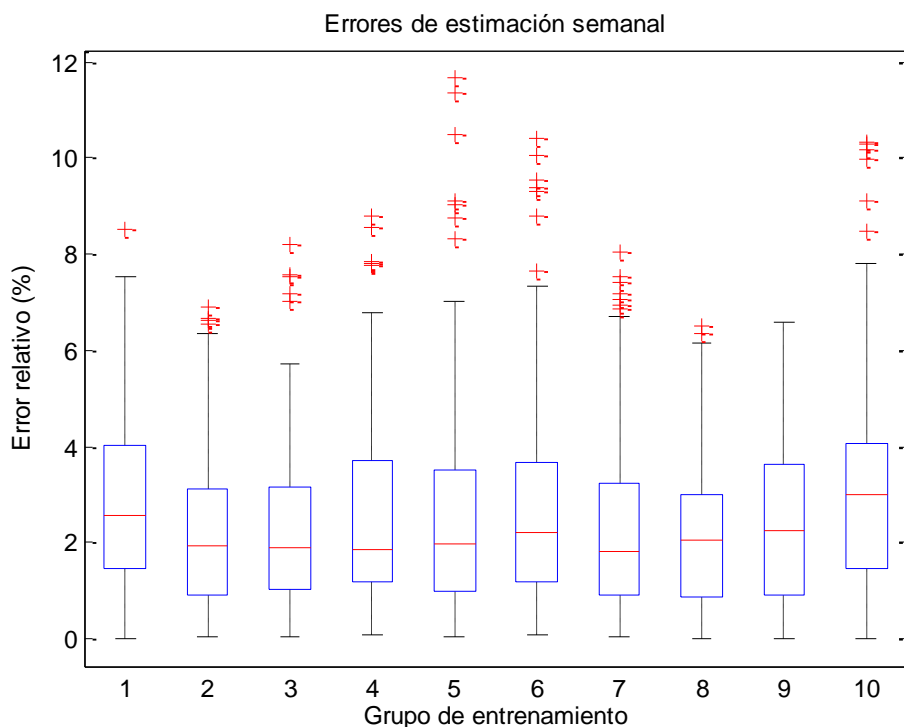


Figura 6.38 Resultados obtenidos mediante el suavizado de Golay en la semana del 27 Julio al 2 de Agosto

Los resultados mostrados en la Figura 6.39 son prácticamente idénticos a los obtenidos con el suavizado mediante splines cúbicos (Figura 6.38). Los errores relativos respecto de la curva suavizada con Golay (Tabla 11) también son muy parecidos a los obtenidos en el suavizado de splines (Tabla 10).

6.7. CONCLUSIONES

Las conclusiones ya se han ido comentado a lo largo de los resultados mostrados por la simulación de las distintas semanas de las distintas estaciones del año. En general, se obtienen resultados que pueden llegar a ser válidos en la mayoría de las predicciones.

En el resto de semanas donde se obtienen unos resultados bastante peores, muestran una carencia del modelo. Esa carencia puede residir en la falta de alguna variable endógena de entrada que pueda ayudar aún mas en la predicción de las distintas potencias consumidas aunque, si que es cierto, que la temperatura es un claro indicador de como puede funcionar el modelo, se echan en falta mas variables que consigan un modelo de predicción aún más válido.

En cuanto a los dos métodos de suavizado utilizados, ambos son válidos, se esperaba que el método de Savitzky-Golay respondiera mejor comparando las predicciones respecto de la curva suavizada pero no ha habido apenas diferencia comparandolo con el caso de suavizado mediante splines, y en este último método además se obtienen mejores resultados respecto de la curva real. Por ello en el siguiente apartado, se utilizara únicamente el suavizado mediante splines.

Los dos métodos contienen variables que marcan el nivel de suavizado de la curva, quizá para alguna situación específica sea mejor suavizar con Savitzky-Golay debido a que tiene dos parámetros para escoger el grado de suavizado.

6.8. IMPORTANCIA DE LAS VARIABLES DE POTENCIA

Ya se ha escrito sobre las distintas opiniones que existen a favor y en contra de la inclusión como entradas de las variables de consumo de potencia en tiempos pasados para ayudar a predecir la potencia consumida en un instante de tiempo predeterminado. Se han visto modelos extremos que incluyen variables de potencia de incluso años anteriores.

En el modelo inicial de este proyecto sí que se han incluido variables de consumo de potencia de meses anteriores y semanas anteriores además de valores de potencia de consumo previo a corto plazo.

Algunos autores defienden la teoría de que las variables a largo plazo en el pasado más que ayudar, equivocan a la red neuronal ya que las condiciones que se dan en una situación presente pueden ser muy distintas a las que se dieron hace un mes, dos meses o hace un años.

En la siguiente parte del proyecto se quiere estudiar cómo afectan las variables pasadas a “largo y medio plazo” en el entrenamiento de la red neuronal y si esta misma funcionaría mejor con ellas o sin ellas. Es decir si hasta ahora para el entrenamiento de la red neuronal, se usaban la potencia consumida hace un semana, hace un día y hace tres, dos y una hora respectivamente, se van a ir eliminando estas variables del

entrenamiento de la red neuronal y se van a evaluar los resultados comparándolos con los mismos obtenidos en el apartado anterior. Se van a realizar tres pruebas:

- Eliminando la potencia consumida a largo plazo.
- Eliminando la potencia consumida a medio plazo.
- Únicamente utilizando la potencia a muy corto plazo.

Es decir, nuestras variables necesarias para el entrenamiento de la red neuronal, quedan como se ven a continuación para los tres casos.

El primer paso a realizar tal y como se anuncio en el apartado 6.1 es la modificación de las neuronas de la capa oculta, para ello se ha utilizado el método de prueba y error. Este método ha consistido en simular dos veces 4 semanas diferentes con distintos números de neuronas y hacer una evaluación del error y de las situaciones específicas encontradas.

1 Caso. Eliminando la potencia consumida a largo plazo

Las variables de entrada son las siguientes.

Potencia consumida hace 1 día
Potencia consumida hace 3 horas
Potencia consumida hace 2 horas
Potencia consumida hace 1 hora
Dia laboral o no
Hora 1
Hora 2
Hora 3
Hora 4
Hora 5
Temperatura máxima
Temperatura mínima

Despues de las pruebas, se considera oportuno dejar el número de neuronas de la capa intermedia en 15.

2 Caso. Eliminando la potencia consumida a largo y corto plazo

En este caso, las variables de entrada son las siguientes:

Potencia consumida hace 3 horas
Potencia consumida hace 2 horas
Potencia consumida hace 1 hora
Dia laboral o no
Hora 1
Hora 2
Hora 3
Hora 4
Hora 5
Temperatura máxima
Temperatura mínima

Despues de las pruebas, se considera oportuno dejar el número de neuronas de la capa intermedia en 14.

3 Caso. Utilizando para el entrenamiento únicamente la potencia a muy corto plazo.

En este último caso, las variables de entrada son:

Potencia consumida hace 1 hora
Dia laboral o no
Hora 1
Hora 2
Hora 3
Hora 4
Hora 5
Temperatura máxima
Temperatura mínima

El número de neuronas de la capa intermedia desciende hasta 13.

Hay que recordar que el número de neuronas de la capa de entrada se modifica para que coincida con el número de variables de entrada.

A continuación se exponen las simulaciones y predicciones obtenidas en las mismas semanas que en el caso inicial. El suavizado de las curvas se obtienen mediante la interpolación por splines cúbicos.

6.8.1. OTOÑO

A continuación se mostrarán una serie de gráficas mostrando las simulaciones de los 3 casos para la semana del 29 de Septiembre al 5 de Octubre, Como se observará se obtienen unos resultados muy buenos al igual que pasaba en el apartado 6.6.1, al igual que se explicó en ese mismo apartado, no todas las semanas de otoño responden tan bien

6.8.1.1. Caso 1. Eliminando la potencia consumida a largo plazo

El resultado de la simulación ha sido el siguiente.

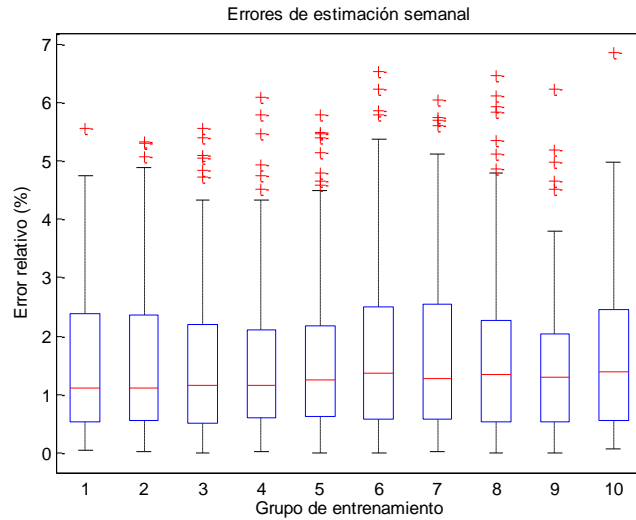


Figura 6.39 Simulación otoño caso 1

Como se observa en la Figura 6.39, los errores medios están cercanos al 1%, lo que muestra una gran respuesta de la red ante el caso 1.

6.8.1.2. Caso 2. Eliminando la potencia consumida a largo y corto plazo

La Figura 6.40 muestra el resultado obtenido.

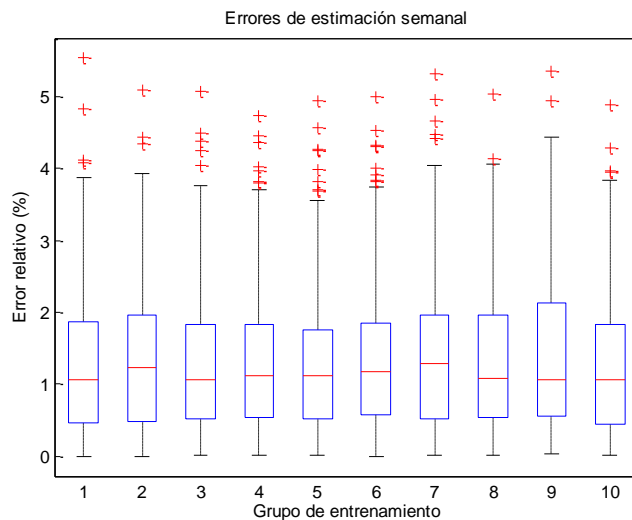


Figura 6.40 Simulación otoño caso 2

Al igual que en el caso1, los errores medios no sobrepasan el 1,5%, tambien se obtiene una muy buena simulación.

6.8.1.3. Caso 3. Utilizando para el entrenamiento únicamente la potencia a muy corto plazo

La figura 6.41, muestra los errores de la predicción obtenidos para la semana de otoño (29 Sep-5 Oct) en el caso 3. Si se compara este resultado con el resultado de los casos 1,2 y 3, se observa, que tenemos menos desviaciones típicas, pero en general, muy similar a las 2 simulaciones anteriores.

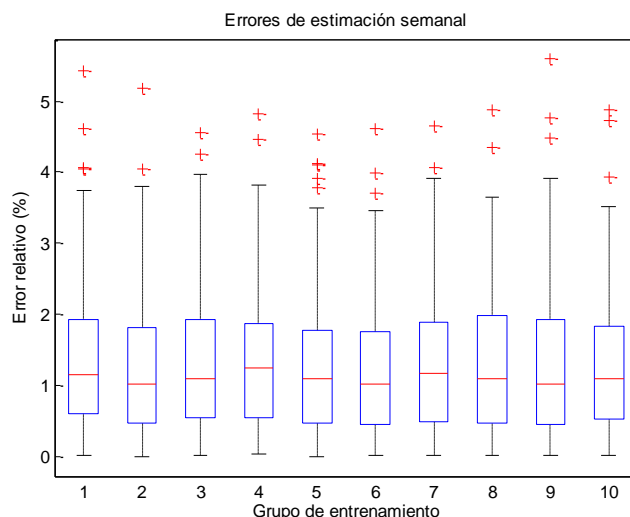


Figura 6.41 Simulación otoño caso 3

Tabla 12 Comparación errores predicción 29 Septiembre - 5 Octubre

Error relativo %	1	2	3	4	5	6	7	8	9	10
Spline	1,52	1,46	1,56	1,51	1,39	1,41	1,58	1,40	1,50	1,49
Golay	1,81	1,67	1,58	1,54	1,74	1,57	1,68	1,71	1,67	1,61
Caso 1	1,34	1,27	1,32	1,31	1,39	1,43	1,34	1,45	1,30	1,34
Caso 2	1,37	1,23	1,29	1,35	1,28	1,24	1,31	1,31	1,34	1,31
Caso 3	1,38	1,299	1,29	1,33	1,34	1,27	1,28	1,33	1,32	1,37

La Tabla 12 muestra una mejoría de los resultados en los tres casos propuestos, aunque, si bien es cierto que no hay mucha diferencia entre los errores de los tres casos y el error del caso inicial mediante el suavizado por splines cúbicos, si se obtienen unos resultados que pueden llevar a pensar que hay ciertas variables de entrada del caso inicial que solo aportan carga computacional. Como se observa en las Figuras 6.39,6.40 y 6.41, cuantas menos variables de entrada de potencia se utilicen, se reduce el número de desviaciones típicas. Puede ser debido, a que con una variable de potencia la red

neuronal ya sea capaz de extraer toda la información necesarias, y el resto de variables en vez de aportar información aporten unicamente carga computacional y error.

6.8.2. INVIERNO

Se mostrarán los resultados obtenidos en la semana que va desde el 9 al 15 de Febrero para los 3 casos planteados, se observará que durante esta época, los resultados obtenidos, mejoran a los resultados que se produjeron en el caso inicial (uso de todas las potencias, semana anterior, día anterior y las tres horas anteriores)

6.8.2.1. Caso 1. Eliminando la potencia consumida a largo plazo

En la Figura 6.42 se observar que los resultados mejoran incluso a los obtenidos en la Figura 6.29 y Figura 6.30, lo que parece indicar, que las variables de potencia a largo plazo, no son tan necesarias.

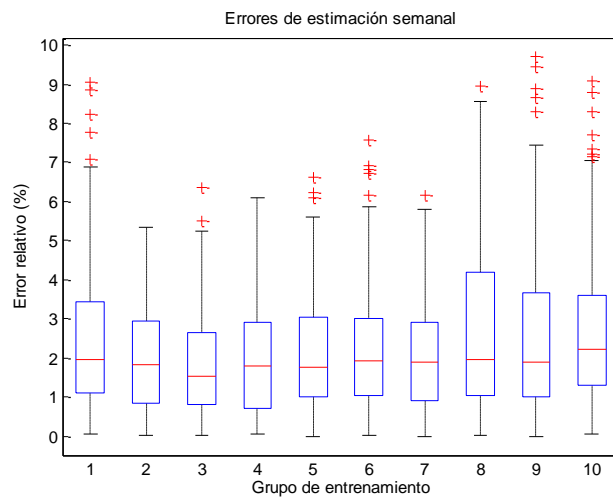


Figura 6.42 Resultados simulación semana invierno caso 1

6.8.2.2. Caso 2. Eliminando la potencia consumida a largo y corto plazo

La Figura 6.43 corrobora lo que se escribió en el apartado anterior, y además se obtienen menos desviaciones típicas.

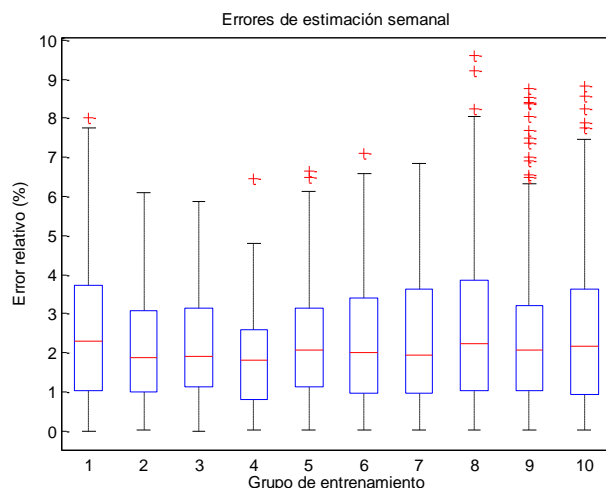


Figura 6.43 Resultados simulación invierno caso 2

6.8.2.3. Caso 3. Utilizando para el entrenamiento únicamente la potencia a muy corto plazo

La Figura 6.44 ofrece unos resultados mejores que en los 2 casos anteriores, con menores desviaciones típicas, y un error medio un poco mas pequeño. Se vuelve a estar en el mismo caso que en la semana simulada de otoño.

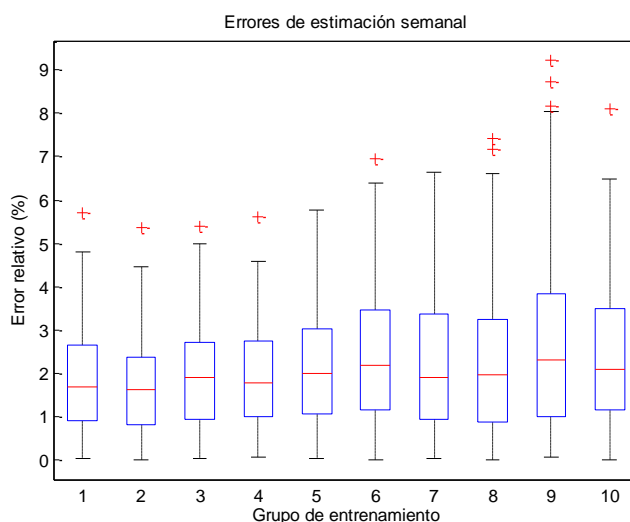


Figura 6.44 Resultados simulación invierno caso 3

Tabla 13 Comparación errores predicción 9-15 Febrero

Error relativo %	1	2	3	4	5	6	7	8	9	10
Spline	2,55	2,71	2,32	2,24	2,23	3,06	2,44	2,28	2,28	2,35
Golay	2,55	2,78	1,62	2,01	2,62	2,30	1,84	1,92	2,19	2,36
Caso 1	2,48	2,03	1,89	1,98	2,10	2,24	2,10	2,69	2,53	2,64
Caso 2	2,24	1,62	1,97	1,85	2,01	2,33	2,24	2,367	2,72	2,44
Caso 3	1,85	1,68	1,96	1,88	2,14	2,34	2,26	2,22	2,66	2,36

La Tabla 13 vuelve a mostrar resultados similares a los obtenidos en la semana de otoño, se obtiene un mejor resultado (error mas pequeño) en los tres casos nuevos. También se repite el patrón de menores desviaciones a medida que se eliminan variables de entrada.

Desde luego con estos dos casos analizados se puede empezar a cuestionar si merece la pena introducir mas carga computacional para obtener peores resultados, a continuación se estudiarán casos mas desfavorables, para ver como responden estos tres casos ante esas situaciones.

6.8.3. PRIMAVERA

Al igual que pasaba en el punto 6.6.3 las siguientes predicciones son mas irregulares que las obtenidas en otoño e invierno, además al contrario que venía pasando en las dos estaciones anteriores, los resultados de los 3 nuevos casos, son peores que los dados en el caso inicial, donde se usaban las variables de potencia de hace 1 semana, 1 día y las 3 últimas horas.

6.8.3.1. Caso 1. Eliminando la potencia consumida a largo plazo

La Figura 6.45 muestra al igual que la Figura 6.33 y Figura 6.34 unos peores resultados que en estaciones anteriores, donde además se produce mucha diferencia de error entre los distintos grupos de entrenamiento.

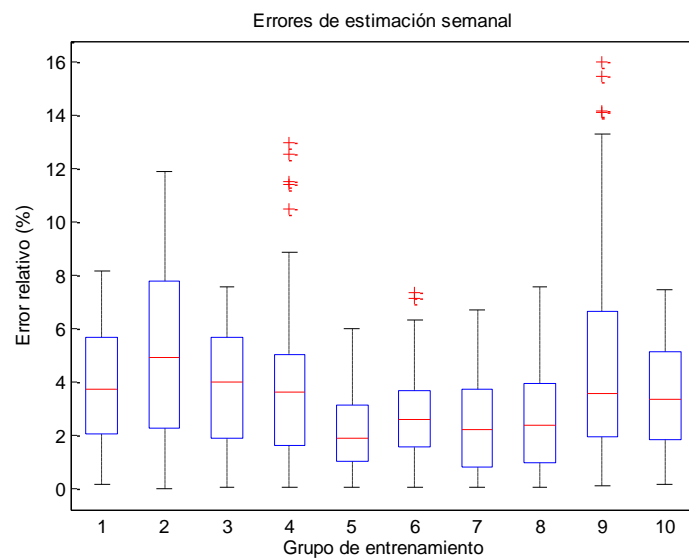


Figura 6.45 Simulación semana de primavera caso 1

6.8.3.2. Caso 2. Eliminando la potencia consumida a largo y corto plazo

En la Figura 6.46 se pueden ver unos resultados parecidos al caso anterior, esta gráfica vuelve a reflejar unos resultados peores que los dados en el caso inicial.

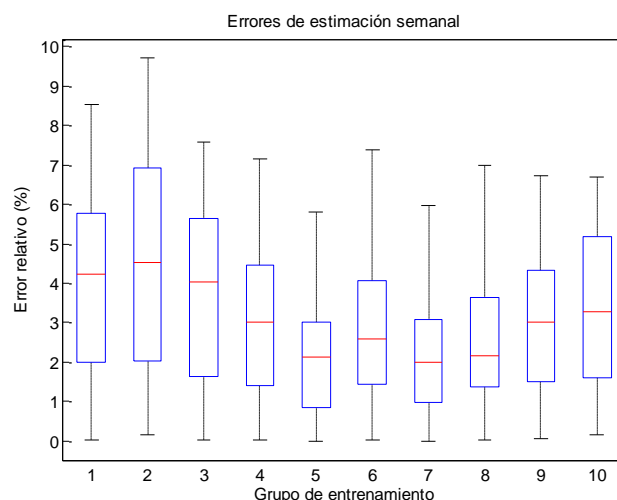


Figura 6.46 Simulación semana de primavera caso 2

6.8.3.3. Caso 3. Utilizando para el entrenamiento únicamente la potencia a muy corto plazo

De los 3 nuevos casos, este tercer caso cuyo resultado de la simulación se observa en la Figura 6.47 es el peor en cuanto a lo elevado del error medio y al número de desviaciones típicas de alguno de sus grupos de entrenamiento.

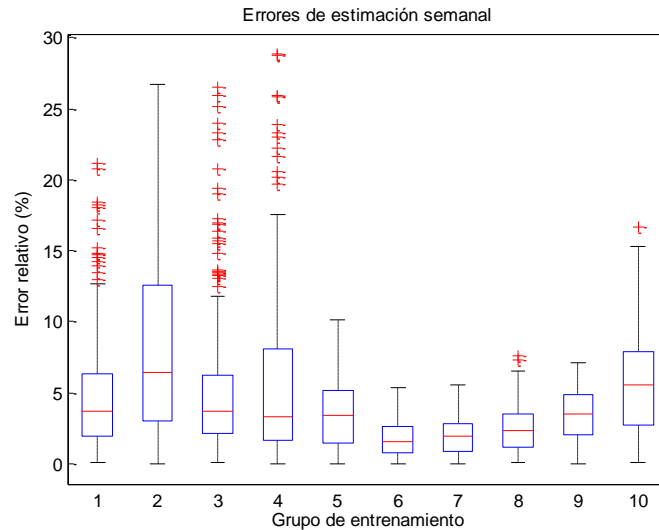


Figura 6.47 Simulación semana de primavera caso 3

Tabla 14 Comparación errores predicción 13-19 abril

Error relativo %	1	2	3	4	5	6	7	8	9	10
Spline	2,83	4,05	3,35	2,99	2,88	2,47	1,77	2,02	2,25	2,25
Golay	2,893	3,28	2,96	2,45	2,57	2,31	1,98	1,99	2,44	2,65
Caso 1	3,87	5,10	3,85	3,60	2,14	2,70	2,37	2,57	4,63	3,40
Caso 2	3,96	4,52	3,68	2,99	2,12	2,84	2,16	2,48	2,95	3,32
Caso 3	5,31	8,20	5,86	5,91	3,65	1,80	1,98	2,45	3,46	5,64

Los resultados de la predicción esta semana son peores si los comparamos con los anteriores, aunque si bien es cierto que en algunos grupos de entrenamiento se obtienen mejores resultados, en la mayoría de ellos se obtienen peores resultados.

Comparando los tres casos entre sí, el peor parado, es el caso en el que solo se utiliza como variable pasada, la potencia consumida hace 1 hora. Esta simulación parece indicar que para aquellas semanas donde se han obtenido peores resultados, no será tan clara la ventaja en favor de los nuevos casos. Puede ser debido a las diferencias bruscas de temperatura características de esta estación, donde al tener más datos de potencia disponibles, puede seguir como guía a la red neuronal de la tendencia de los cambios bruscos de temperatura y por lo tanto llegar a predecir mejor cuantos mas datos de potencia pasada se tengan.

6.8.4. VERANO

En este apartado se muestran las predicciones obtenidas de los 3 casos en la semana del 27 de julio al 2 de agosto. A parte de que los errores son mayores de los esperados como ya se vio en el apartado 6.6.4 se vuelve a la tónica general de las estaciones de otoño e invierno obteniéndose mejores resultados en los 3 casos que en el caso inicial.

6.8.4.1. Caso 1. Eliminando la potencia consumida a largo plazo

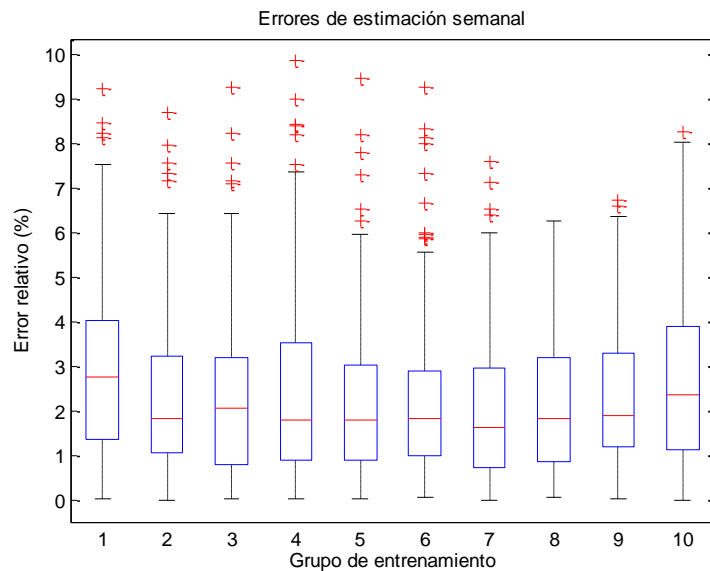


Figura 6.48 Simulación verano caso 1

6.8.4.2. Caso 2. Eliminando la potencia consumida a largo y corto plazo

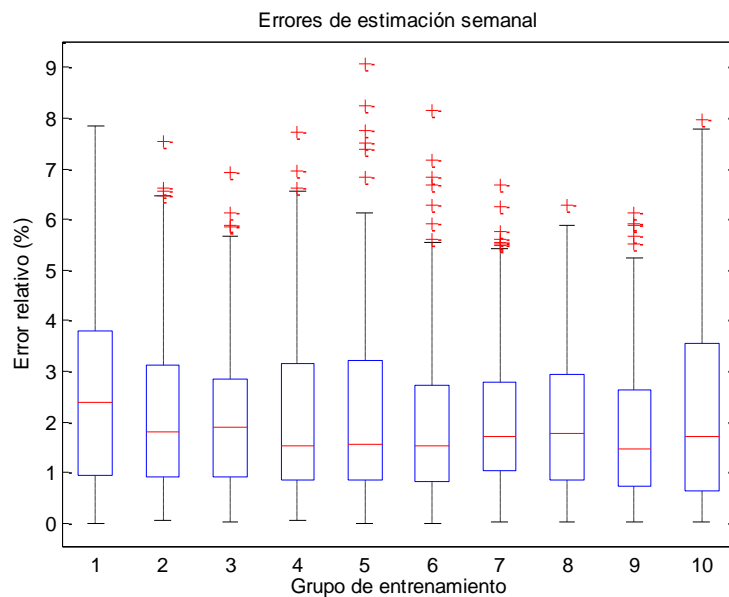


Figura 6.49 Simulación verano caso 2

6.8.4.3. Caso 3. Utilizando para el entrenamiento únicamente la potencia a muy corto plazo

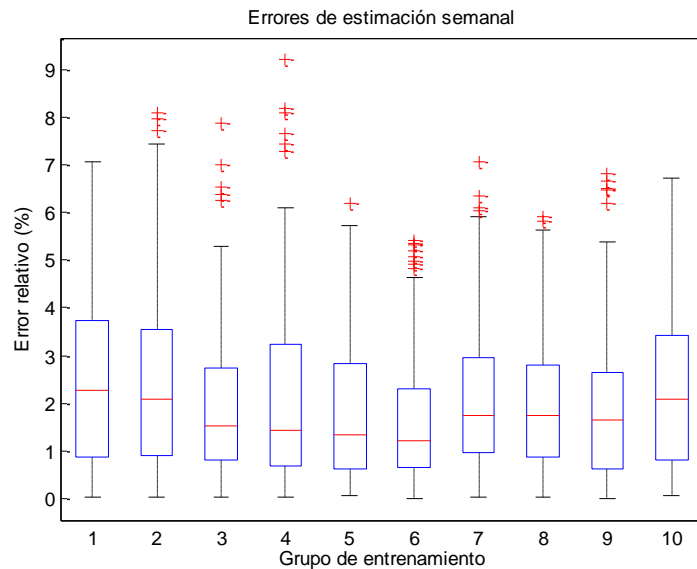


Figura 6.50 Simulación verano caso 3

Las Figuras 6.48 , 6.49 y 6.50 muestran unos resultados muy similares, como se puede comprobar también en la Tabla 15, los 3 casos mejoran los resultados del caso inicial en casi todos los grupos de entrenamiento, de hecho, cuanto menos variables de potencia se utilizan para predecir, mejores son los resultados.

Tabla 15 Comparación errores predicción 27 julio-2 agosto

	1	2	3	4	5	6	7	8	9	10
Spline	3,06	2,52	2,38	2,58	2,43	2,99	2,37	2,08	2,13	3,09
Golay	2,86	2,24	2,298	2,49	2,51	2,70	2,27	2,05	2,41	3,09
Caso 1	2,98	2,40	2,41	2,43	2,21	2,23	2,07	2,14	2,37	2,69
Caso 2	2,55	2,21	2,14	2,16	2,22	2,03	2,08	2,09	1,86	2,27
Caso 3	2,44	2,41	1,99	2,12	1,83	1,63	2,07	2,02	1,89	2,24

Esta semana vuelve a mostrar unas simulaciones que arrojan unos mejores resultados par los tres casos, incluso como ya paso en las simulaciones de otoño e invierno, los mejores resultados se obtienen para el caso 3. Al haber menos variaciones de temperatura las variables de potencia son menos necesarias para modelar esa variación de temperatura y por lo tanto no son tan utiles como en la primavera.

6.8.5. OTRAS SIMULACIONES

Como con los ejemplos anteriores no ha quedado del todo claro como son de válidos los tres casos donde se reduce el número de variables de entrada, se mostrarán tres simulaciones que pueden ayudar a esclarecer resultados. Si algo queda claro de las predicciones anteriores, es que ante semanas donde se han obtenido buenos resultados en el caso inicial con errores relativos pequeños, la potencia consumida en el pasado aporta mas bien poco a la red neuronal.

Como en este apartado, estamos comparando los casos 1, 2 y 3. Solo se va a mostrar la simulación y los errores de caso inicial con el suavizado mediante splines comparando los resultados con los casos 1 2 y 3.

6.8.5.1. Semana 13-19 de Octubre.

En el otoño al igual que en la primavera, es donde mayores variaciones de temperatura existen, por lo tanto, se espera que en este periodo, los 3 nuevos casos analizados tengan problemas para mejor los resultados del caso inicial.

Tabla 16 Comparación de errores semana 13-19 octubre

Error relativo %	1	2	3	4	5	6	7	8	9	10
Spline	7,04	5,70	6,79	7,94	7,16	7,66	9,44	7,38	7,54	6,87
1 caso	7,30	7,57	7,77	7,78	7,59	7,40	6,30	7,74	8,03	7,90
2 caso	7,53	8,06	8,11	7,77	7,56	7,59	7,75	6,88	6,99	8,49
3 caso	6,86	7,17	7,71	7,54	7,88	7,41	7,42	7,56	8,04	7,15

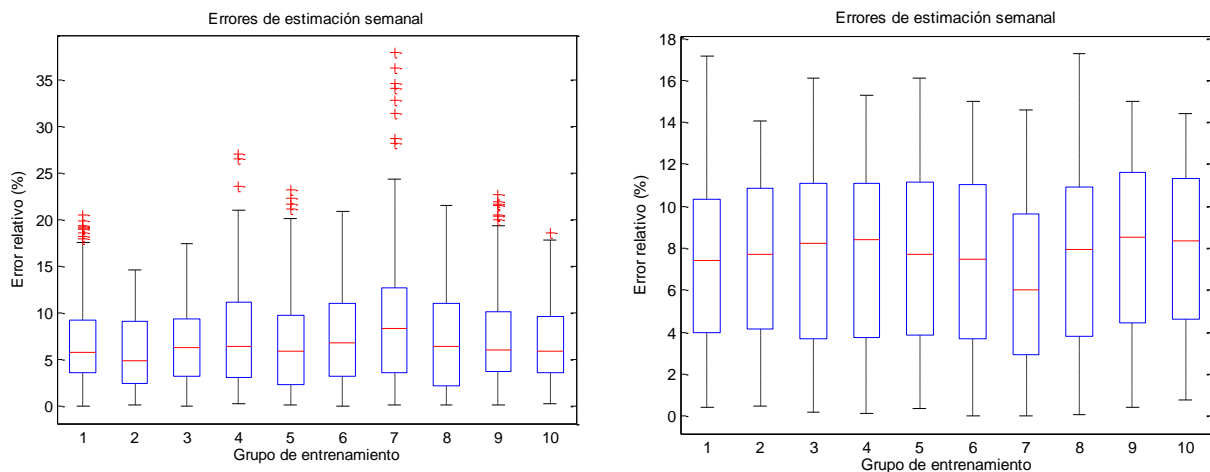


Figura 6.51 Caso inicial (spline) (izquierda) y Caso1 (derecha)

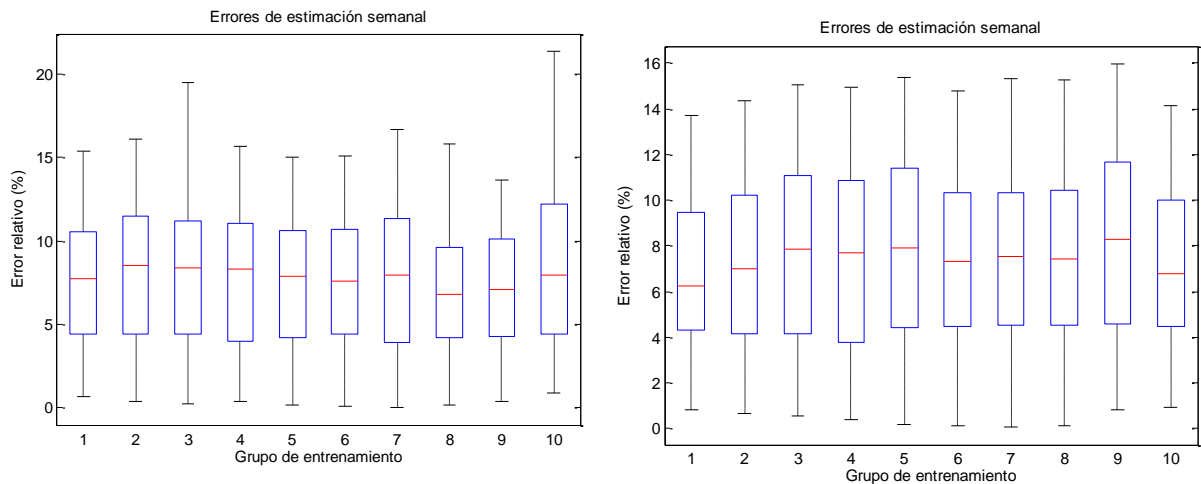


Figura 6.52 Caso 2 (izquierda) Caso 3 (derecha)

Teniendo en cuenta los errores relativos medios de la Tabla 16, los 4 resultados son bastante parecidos, en algunos grupos de entrenamiento las predicciones son mejores para el caso inicial (uso de variables de potencia de hace 1 semana, 1 día y 3 horas anteriores) y en otros grupos las mejores predicciones se dan en los otros 3 casos. Esto nos muestra que la supremacía de mejores resultados de los tres nuevos casos en aquellas semanas y periodos donde las variaciones de temperatura no son bruscas, no se cumplen en aquellas semanas donde si existen esas grandes variaciones, generalmente en las semanas de otoño y primavera.

En cambio si se observan las desviaciones típicas de las Figuras 6.51 y 6.52, se ve como en el caso de suavizado con splines y 13 variables de entrada (caso inicial), se producen errores puntuales mucho mayores, no dejan de ser características que se cumplen a lo largo del otoño debido a las variaciones de temperatura. La semana simulada en el apartado 6.8.1 si que mostraba unos mejores resultados para los tres casos nuevos que en esta nueva semana, pero tambien se observa como los errores son mucho menores que la media del otoño. Se puede extraer como conclusión que aunque esa semana pertenezca al otoño, sus variaciones de temperatura no han sido tan bruscas como suelen ser en esta estación.

6.8.5.2. Semana 28 Diciembre-4 Enero

Durante el Invierno no existe esas variaciones bruscas de temperatura, por lo que se espera según lo visto y explicado en el apartado 6.8.2 que los 3 nuevos casos, respondan mejor que el caso inicial, se ha escogido además una semana un poco peculiar en cuanto al número de días no laborables que hay para exponer el buen funcionamiento de la red neuronal ante este tipo de casos.

Tabla 17 Comparación de errores semana 28 diciembre-4 enero

Error relativo %	1	2	3	4	5	6	7	8	9	10
Spline	3,70	4,18	4,90	3,59	4,93	5,30	5,45	6,07	3,79	3,41
1 caso	4,09	3,15	3,24	2,90	3,14	3,45	3,20	3,98	3,40	1,60
2 caso	4,78	3,01	4,91	4,15	3,63	3,26	4,47	4,02	3,29	2,82
3 caso	3,11	4,34	3,45	3,18	4,52	3,03	3,39	4,19	3,49	4,01

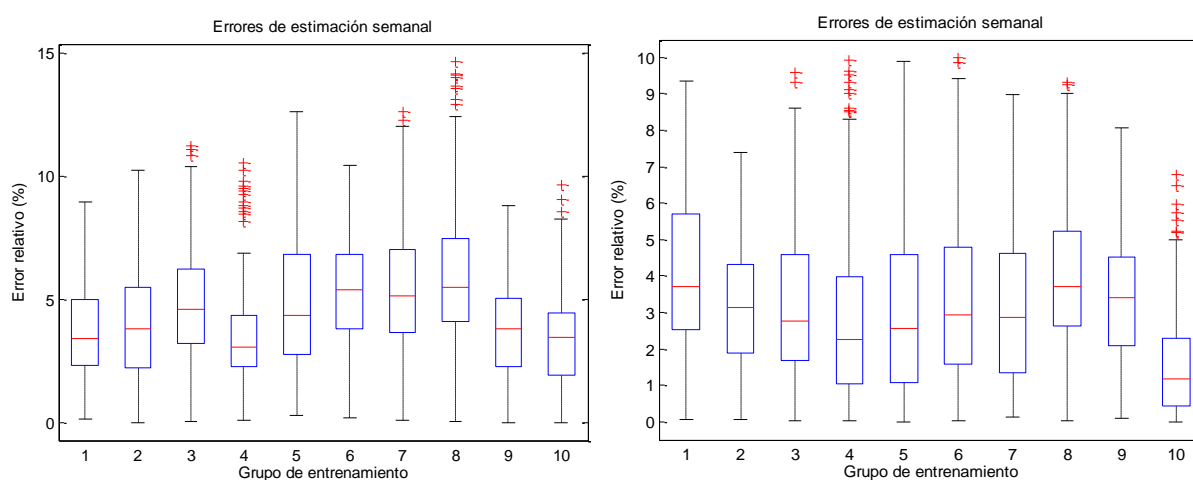


Figura 6.53 Caso inicial (izquierda) Caso 1 (derecha)

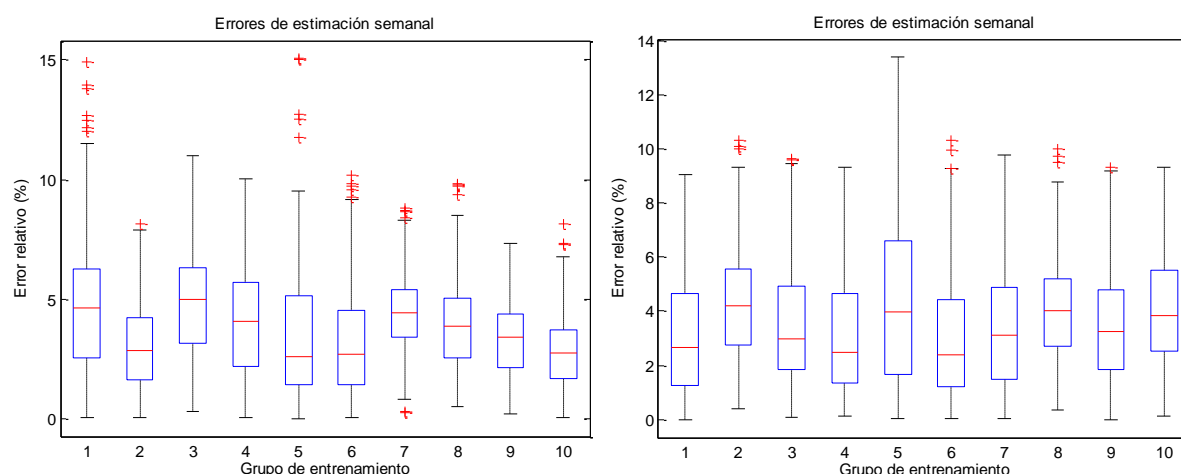


Figura 6.54 Caso 2 (izquierda) Caso 3 (derecha)

Al contrario que pasaba en la semana simulada en el apartado 6.8.5.1 como se puede extraer de conclusión en la Tabla 17 los resultados de los 3 casos mejoran al caso inicial, las Figuras 6.53 y 6.54 corroboran los resultados de la Tabla, y se puede observar en ellas una mejora de los resultados al eliminar variables de potencia consumida en el pasado.

6.8.5.3. Semana 20-26 de Abril

En la anterior semana de primavera predicha en el apartado 6.8.3 se obtuvieron unos resultados muy malos para los 3 casos nuevos, estos resultados se justificaron en base a las variaciones bruscas de temperatura características de primavera, con esta nueva simulación se pretende mostrar que el resultado que se obtuvo no es fruto de la casualidad, sino que tiene un fundamento justificado.

Tabla 18 Comparación de errores semana 20-26 Abril

Error relativo %	1	2	3	4	5	6	7	8	9	10
Spline	4,27	4,623	4,27	5,84	5,61	3,62	2,777	3,49	3,71	2,95
1 caso	6,09	7,34	7,14	7,38	6,23	4,90	3,72	4,63	4,57	4,85
2 caso	6,13	7,72	7,51	9,32	7,01	6,02	4,47	5,42	5,76	5,36
3 caso	6,03	8,27	8,56	8,69	6,29	5,96	4,31	4,38	5,12	5,04

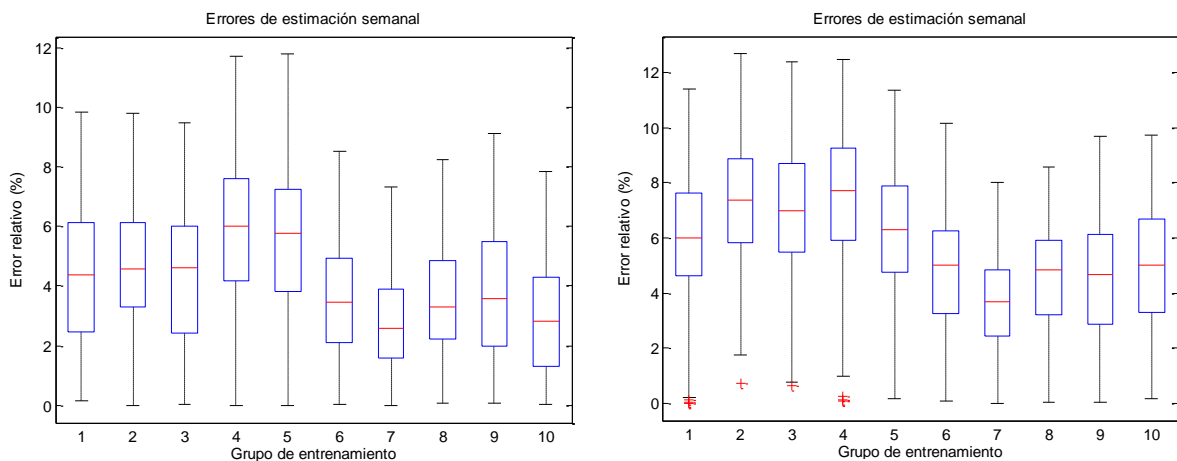


Figura 6.55 Caso inicial (izquierda) Caso 1 (derecha)

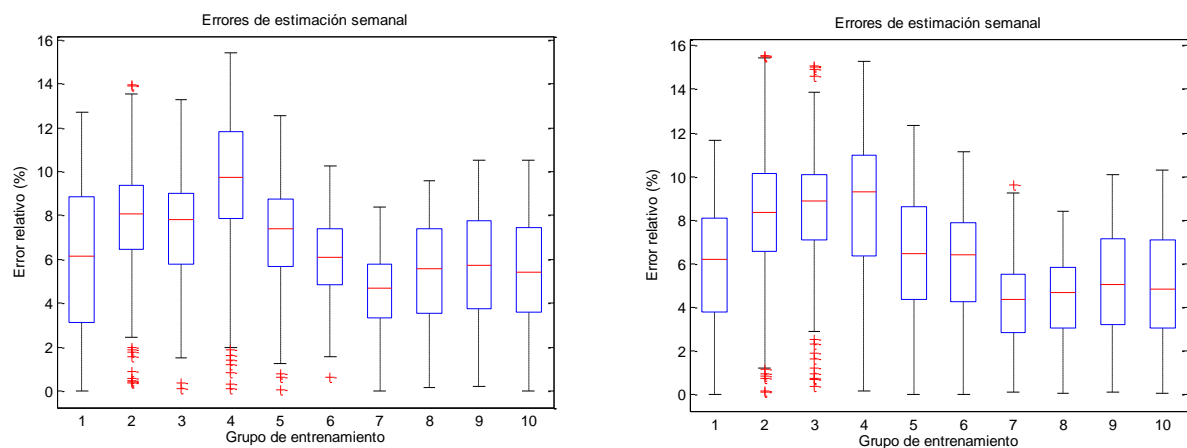


Figura 6.56 Caso 2 (izquierda) Caso 3 (derecha)

La Tabla 18 y las Figuras 6.55 y 6.56 muestran los resultados que ya se ha justificado anteriormente, esta vez los resultados del caso inicial, son mejores en todos los grupos de entrenamiento.

6.8.6. CONCLUSIONES

Después de realizar las simulaciones, se llega a la conclusión de que en la mayoría de los casos, si se eliminan de las entradas las variables de potencia consumida en el pasado no tienen ninguna repercusión, no solo eso, sino que además, en casi todos los casos se mejora la exactitud de la predicción.

Es sobre todo en la época de primavera y en la época central de otoño donde funcionan peor este tipo de simulaciones en las que se eliminan las variables de potencia consumida. Esta característica puede ser debida a las variaciones de temperatura existentes. Se debe recordar, que en esta época era cuando peor funcionaban las simulaciones con todas las variables de potencia consumida.

Como ya se dijo anteriormente, se sigue evidenciando la falta de alguna variable endógena que consiga hacer funcionar la red neuronal en estas semanas. Desde luego si se encuentra esa variable de entrada, se podrán eliminar las variables de potencia consumida en el pasado (salvo la de la hora anterior) y obtener mejores resultados con menor carga computacional.

7. RESUMEN RESULTADOS CONCLUSIONES Y PROPUESTAS DE CONTINUACIÓN

7.1. RESUMEN RESULTADOS

Se han mostrado a lo largo del proyecto una serie de figuras y tablas con una serie de resultados que se han ido justificando, en este apartado se quiere resaltar y resumir las gráficas más importantes al igual que se expondrán unas conclusiones generales. El primer estudio que se realizó, fue la comparación entre dos métodos de suavizado. El suavizado mediante la interpolación de splines-cúbicos y el filtrado de Savitzky-Golay.

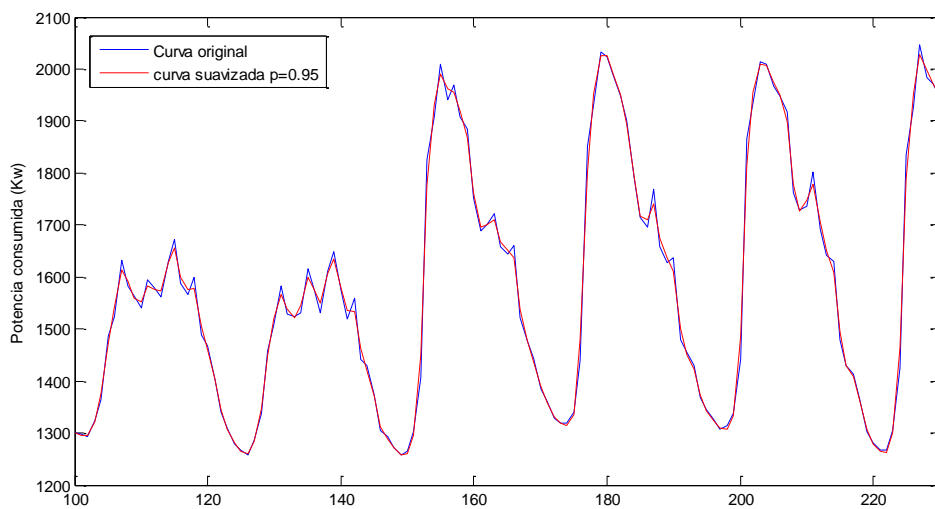


Figura 7.1 Splines

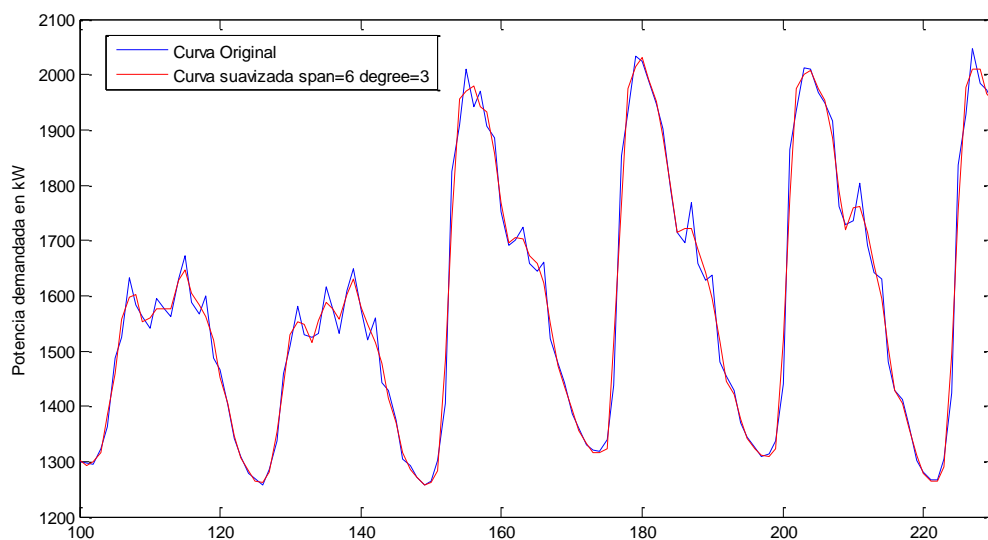


Figura 7.2 Savitzky-Golay

Aunque la Figura 7.2 (Savitzky-Golay) muestra un mayor suavizado lo cual implica una mayor eliminación de puntos anómalos y por lo tanto la red neuronal debería responder

mejor ante los errores relativos respecto a la curva suavizada, la Tabla 19 correspondiente a la simulaciones del apartado 6.6 arroja unos resultados dispares en cuanto a que método es mejor. Aunque en general el error relativo respecto de la curva suavizada es menor en Golay, hay muchos casos en los que la simulación con el suavizado de la curva de potencia mediante el método de splines obtiene errores relativos más bajos.

Como respecto a la curva real, el error relativo es menor con el suavizado de splines, en el resto del proyecto se decidió suavizar las curvas de potencia mediante interpolación por splines.

Se debe aclarar, que se hubiera podido conseguir una curva menos suavizada con Savitzky y que respondiera igual de bien que el caso de suavizado con Splines. Pero la misión era comprobar que ante una curva más suavizada, la red neuronal respondía mejor, el resultado no ha sido satisfactorio, se decidió utilizar Savitzky-Golay debido a que tiene dos parámetros que permite ajustar de una mejor manera el grado de suavizado.

Tabla 19 Comparación de Errores Spline-Golay

Error relativo (%)	1	2	3	4	5	6	7	8	9	10	
Otoño	Spline	1,52	1,46	1,56	1,51	1,39	1,41	1,58	1,40	1,50	1,49
	Golay	1,81	1,67	1,58	1,54	1,74	1,57	1,68	1,71	1,67	1,61
Invierno	Spline	2,55	2,71	2,32	2,24	2,23	3,06	2,44	2,28	2,28	2,35
	Golay	2,86	2,24	2,29	2,49	2,51	2,70	2,27	2,05	2,41	3,09
Primavera	Spline	2,83	4,05	3,35	2,99	2,88	2,47	1,77	2,02	2,25	2,25
	Golay	2,89	3,28	2,96	2,45	2,57	2,31	1,98	1,99	2,44	2,65
Verano	Spline	3,06	2,52	2,38	2,58	2,43	2,99	2,37	2,08	2,13	3,09
	Golay	2,86	2,24	2,29	2,49	2,51	2,70	2,27	2,05	2,41	3,09

El siguiente paso fue hacer un estudio sobre la dependencia de los buenos resultados de las predicciones respecto al número de variables de potencia consumida en el pasado utilizadas como variables de entrada.

Aunque en las estaciones en las que las variaciones de temperaturas son pequeñas como en verano y en invierno, cuanto menos variables de potencia se utilicen, mejores son los resultados (Tabla 20).

Se debe recordar:

- Caso 1. Se elimina como variable de entrada la potencia consumida a largo plazo (hace 1 semana)

- Caso 2. Se elimina la potencia consumida a largo y corto plazo (hace 1 semana y 1 día)
- Caso 3. Solo se utiliza como variable la potencia a muy corto plazo (potencia consumida hace 1 hora)
- Spline. Caso inicial con el suavizado mediante interpolación cúbica

Tabla 20 Errores relativos Invierno y Verano

Errores relativos %		1	2	3	4	5	6	7	8	9	10
Invierno	Spline	2,55	2,71	2,32	2,24	2,23	3,06	2,44	2,28	2,28	2,35
	Caso 1	2,48	2,03	1,89	1,98	2,10	2,24	2,10	2,69	2,53	2,64
	Caso 2	2,241	1,62	1,97	1,85	2,01	2,33	2,24	2,367	2,72	2,44
	Caso 3	1,85	1,68	1,96	1,88	2,14	2,34	2,26	2,22	2,66	2,36
Verano	Spline	3,06	2,52	2,38	2,58	2,43	2,99	2,37	2,08	2,13	3,09
	Caso 1	2,98	2,40	2,41	2,43	2,21	2,23	2,07	2,14	2,37	2,69
	Caso 2	2,55	2,21	2,14	2,16	2,22	2,03	2,08	2,09	1,86	2,27
	Caso 3	2,44	2,41	1,99	2,12	1,83	1,63	2,07	2,02	1,89	2,24

En las estaciones cuyas variaciones de temperatura son mayores, el resultado es el contrario (Tabla 21), aunque si es cierto que se evidencia más en primavera que en otoño, en esta última estación hay grupos de entrenamiento donde se obtienen mejores resultados en el caso inicial (spline), mientras que en verano e invierno, nunca pasaba esto y una vez más es debido a los cambios bruscos de temperatura.

Tabla 21 Errores relativos Otoño y Primavera

Errores relativos %		1	2	3	4	5	6	7	8	9	10
Otoño	Spline	7,04	5,70	6,79	7,94	7,16	7,66	9,44	7,38	7,54	6,87
	Caso 1	7,30	7,57	7,77	7,78	7,59	7,40	6,30	7,74	8,03	7,90
	Caso 2	7,53	8,06	8,11	7,77	7,56	7,59	7,75	6,88	6,99	8,49
	Caso 3	6,86	7,17	7,71	7,54	7,88	7,41	7,42	7,56	8,04	7,15
Primavera	Spline	4,27	4,623	4,27	5,84	5,61	3,62	2,777	3,49	3,71	2,95
	Caso 1	6,09	7,34	7,14	7,38	6,23	4,90	3,72	4,63	4,57	4,85
	Caso 2	6,13	7,72	7,51	9,32	7,01	6,02	4,47	5,42	5,76	5,36
	Caso 3	6,03	8,27	8,56	8,69	6,29	5,96	4,31	4,38	5,12	5,04

7.2. CONCLUSIONES

Antes de enunciar las conclusiones, se quiere aclarar que cuando se escribe sobre las estaciones de otoño y primavera, el texto se refiere a épocas con mucha variación de temperatura, es decir el 90% de las semanas de estas estaciones, no todas. El mismo caso en invierno y verano, donde las semanas suele ser muy estables

La influencia de las condiciones climatológicas externas es un factor muy importante en la predicción del consumo eléctrico de un hospital, aquellas estaciones como son el otoño y la primavera con unas temperaturas muy variables son sin duda las más difíciles de predecir y en las que hay mayor variedad de resultados entre las distintas semanas. Por el contrario las épocas “planas” de temperaturas como el invierno y el verano son las que mejores resultados obtienes, siendo esos resultados una constante a lo largo de las semanas, es decir, al simular una semana se pueden anticipar todos los resultados del resto de semanas de la estación ya sea con un error mayor (verano) o con un error menor (invierno).

Se puede afirmar que la potencia consumida en el pasado no es un factor tan importante como se pensaba, la carga computacional usada al utilizar variables de potencia consumida en el pasado puede ser perfectamente eliminada sin que la red neuronal note apenas diferencia, por lo menos en la estaciones con temperaturas estables. Respecto a las estaciones de otoño y primavera, si es cierto que esa característica de temperaturas muy cambiantes hace que los resultados obtenidos sean mucho peores.

La conclusión que se extrae que aparte de la falta de alguna variable de entrada más como la luminosidad o la humedad, lo que más error proporciona en las semanas de malos resultados es el conjunto de datos con los que se entrena la red, es decir, para que se obtuvieran mejores resultados, habría que introducir otros datos como entrenamiento e la red. En cualquier caso, para el algoritmo planteado en este proyecto, sí que es necesario utilizar las variables de potencia de consumo pasado (en primavera, y otoño) mientras no se cambie algo de lo mencionado hace unas líneas ya que aunque los resultados sean malos, peores son al eliminar esas variables de potencia.

7.3. LINEAS FUTURAS

Estudio de las variables de entrada, como se ha podido observar aunque la temperatura es una variable que sí es influyente en la correcta predicción de una red neuronal, en aquellas estaciones donde las variaciones de temperatura son grandes, los errores de predicción aumentan de manera considerable. La introducción de nuevas variables de entrada, como puede ser la luminosidad (problema al modelar) o el número de pacientes en el hospital (que seguro que también influye en cuanto al calor irradiado por cada cuerpo y el gasto de energía por paciente) podrían aportar a la red neuronal ese dato que la falta para acabar de funcionar correctamente.

Se podría probar a dividir la red neuronal en varias redes neuronales para buscar una mayor precisión, por ejemplo, si el hospital tienen 15 plantas, realizar 15 redes neuronales, es cierto que para poder realizar este trabajo harían falta más detalles (consumo de cada planta) que tendría que aportar el hospital

En cuanto al algoritmo de entrenamiento, propongo crear un sistema de puntuación para evaluar el parecido de los días entre sí y entrenar la red con los días más parecidos al día o la semana a predecir.

8. COSTES DERIVADOS DEL PROYECTO

En todos los proyectos existe un apartado muy importante que puede marcar el desarrollo satisfactorio del proyecto. Los aspectos económicos de un proyecto han provocado en numerosas ocasiones la debacle de lo que inicialmente parecía un gran proyecto.

El apartado económico permite hacer un análisis de la inversión necesaria, así como el análisis de viabilidad y rentabilidad del proyecto y las formas de financiación de este. Todos estos análisis ayudarán a tomar una decisión sobre la ejecución y/o continuidad del proyecto.

En el presente proyecto únicamente se ha podido estimar un análisis de costes y gastos una vez finalizado el proyecto.

El presente trabajo, es un proyecto de investigación, por lo que el desglose de costes y gastos no será muy profundo, pero sí dará una idea de lo que puede llegar a costar económicamente un trabajo de estas características.

Recordando la diferencia entre costes y gastos: “Coste es aquel tipo de gasto que añade valor al producto o sin el cual no se podría obtener el mismo. Todo coste es un gasto, pero no todo gasto es un coste”. Se considera que todo gasto que se ha tenido en cuenta en este proyecto, es un coste, por lo que aún queda más simplificado el estudio económico del proyecto. Se pueden distinguir dos tipos de costes:

- Costes Directos: “Aquellos que se pueden imputar directa y fácilmente con un producto o servicio vendible”. Es decir aquellos sin los cuales no se podría obtener un producto
- Costes Indirectos. “Aquellos que no afectan realmente a la obtención del producto o servicio final”.

8.1. COSTES DIRECTOS

Dentro de esta categoría podemos distinguir tres tipos de costes:

- Costes de Materiales
- Costes de mano de obra directa
- Gastos directos

Antes de iniciar la contabilización de los costes, se hará un balance de las horas trabajadas en los distintos puntos del proyecto

8.1.1. CALCULO DE COSTE TEMPORAL

En este apartado se pretende hacer un estudio de las horas dedicadas a la realización del proyecto. Lo primero que se ha hecho es dividir el proyecto en 5 etapas.

- Fase I: Estudio inicial
- Fase II: Programación
- Fase III: Simulación
- Fase IV: Análisis de resultados
- Fase V: Informe.

Tabla 22 Desglose en horas del trabajo realizado

FASE	ACTIVIDAD	HORAS
ESTUDIO INICIAL	Estudio Redes Neuronales	15
	Estudio Caso específico	10
	Estudio Algoritmos Suavizado	10
TOTAL HORAS FASE I		35
PROGRAMACIÓN	Estudio Casos Anteriores	25
	Planteamiento nuevas necesidades	5
	Programación nuevas funciones	5
TOTAL HORAS FASE II		35
SIMULACIÓN	Simulaciones	60
TOTAL HORAS FASE III		60
ANÁLISIS DE RESULTADOS	Análisis	10
TOTAL HORAS FASE IV		15
INFORME	Redacción de la memoria	50
TOTAL HORAS FASE V		50
TOTAL HORAS PROYECTO		195

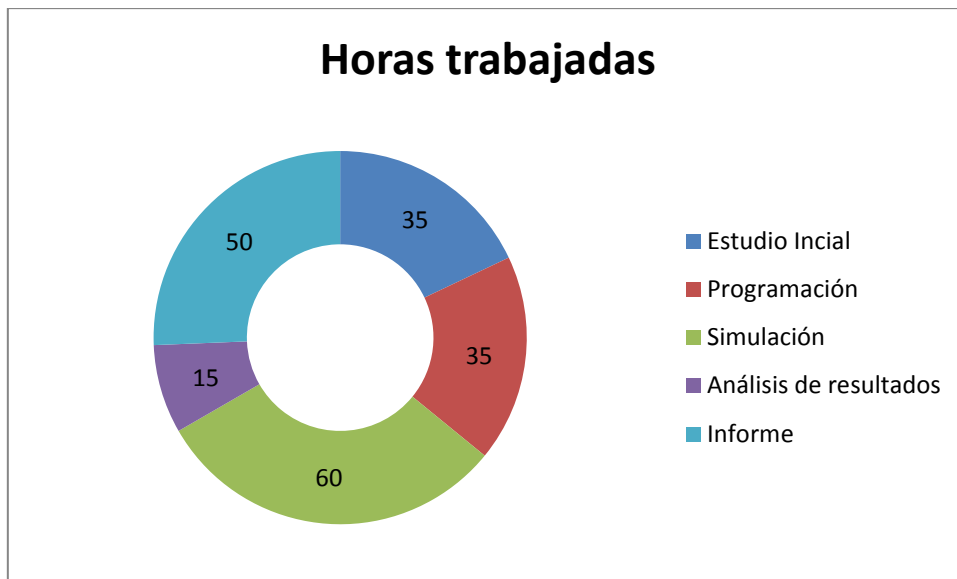


Figura 8.1 Horas trabajadas

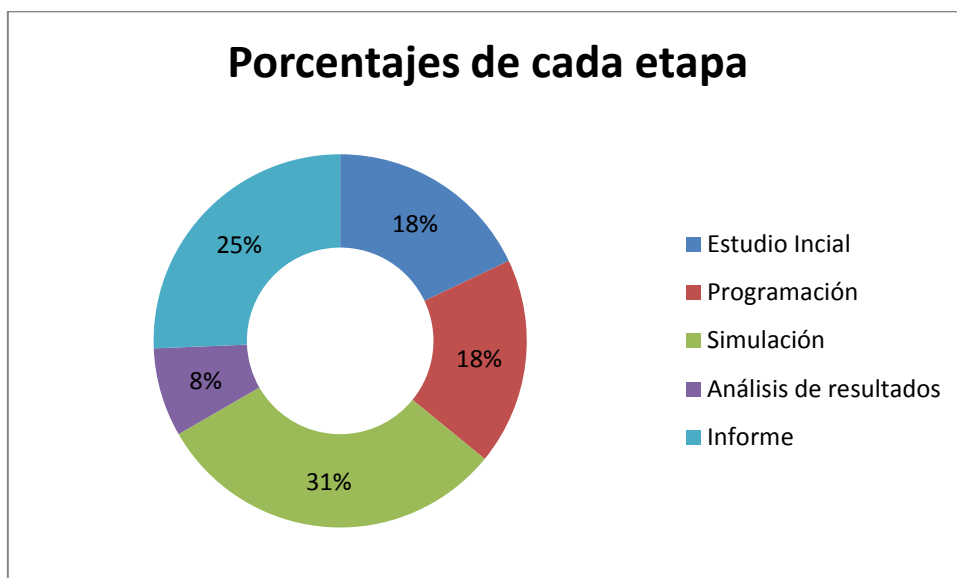


Figura 8.2 Porcentaje del tiempo total dedicado

8.1.2. COSTE DE MANO DE OBRA DIRECTA

El cálculo del coste de personal, se ha elaborado en base a que la persona que ha trabajado en el proyecto tiene una categoría de Ingeniero Industrial.

Los costes que se enuncian a continuación se hacen considerando que el Ingeniero trabaja para una empresa como contratado y no como autónomo. Por supuesto, las cifras que aparecen a continuación son cifras aproximadas que se aproximan a una situación real.

Para empezar, se calcula el coste anual del Ingeniero. Se supone un sueldo bruto anual de 40.000 euros en 14 pagas, es decir, 2857.14 euros por paga, si se supone un IRPF del 19 % y una seguridad social a pagar por el trabajador del 8%.

Tabla 23 Coste mano de obra

CONCEPTO	DESGLOSE	COSTE (€)
Sueldo Bruto	Seguridad social Ingeniero	2.742,28
	IRPF	7.600
	Sueldo Neto	29.657,14
	Total	40.000
Seguridad Social Empresa	SS empresa	12.000
TOTAL	TOTAL	52.000

Para el cálculo del coste de la hora de trabajo del Ingeniero, hace falta saber el número de horas anuales trabajadas.

Calculo de las horas anuales:

Tabla 24 Días anuales trabajados

Concepto	Nº Días
Días totales	365
Fines de Semana	104
Días Festivos	14
Vacaciones	30
Libre disposición	5
Días totales trabajados	212

$$\text{Horas anuales trabajadas} = 212 * 8 = 1696 \text{ Horas/año}$$

El coste horario del Ingeniero será de:

$$\text{Coste horario} = \frac{48.000 \text{ euros}}{1696 \text{ Horas trabajadas}} = 30.66 \text{ euros/hora}$$

Por lo que ahora se puede hacer una estimación del coste de cada etapa.

Tabla 25 Coste de las fases del proyecto

FASE	ACTIVIDAD	HORAS	COSTE (€)
ESTUDIO INICIAL	Estudio Redes Neuronales	459,9	459,9
	Estudio Caso específico	306,6	306,6
	Estudio Algoritmos Suavizado	306,6	306,6
TOTAL HORAS FASE I		35	1073,1
PROGRAMACIÓN	Estudio Casos Anteriores	766,5	766,5
	Planteamiento nuevas necesidades	153,3	153,3
	Programación nuevas funciones	153,3	153,3
TOTAL HORAS FASE II		35	1073,1
SIMULACIÓN	Simulaciones	1839,6	1839,6
TOTAL HORAS FASE III		60	1839,6
ANÁLISIS DE RESULTADOS	Análisis	306,6	306,6
TOTAL HORAS FASE IV		15	459,9
INFORME	Redacción de la memoria	1533	1533
TOTAL HORAS FASE V		50	1533
TOTAL HORAS PROYECTO		195	5978,7

8.1.3. COSTES MATERIAL

En ese apartado se expone la inversión inicial del hardware y software usados, así como sus años de amortización y su valor residual. Una vez estipulada la inversión inicial y los años de amortización, se puede calcular la amortización correspondiente al periodo de tiempo de cada uno de los elementos.

Tabla 26 Amortizaciones

CONCEPTO	INVERSION INICIAL (€)	AÑOS DE AMORTIZACION	VALOR RESIDUAL (%)	VALOR RESIDUAL (€)
HARDWARE				
Ordenador personal+ Hardware	1000	5	10%	100
SOFTWARE				
Windows 7	125	3	-	-
Microsoft Office 2013	150	3	-	-
Matlab R2010a	6000	3	-	-

Como amortización se ha escogido una amortización lineal con el máximo porcentaje posible 20% (5 años) en el hardware y 33% (3 años) en el software.

El coste de amortización por hora se calcula mediante la siguiente fórmula:

$$ca(\text{€/hora}) = \frac{\text{Inversión inicial} - \text{Valor residual}}{1696 (\text{horas}) * \text{Años de amortización}}$$

Tabla 27 Coste de las amortizaciones

CONCEPTO	INVERSION INICIAL	AÑOS DE AMORTIZACION	VALOR RESIDUAL	HORAS DE USO	CA	COSTEL TOTAL AMORTIZACIÓN (€)
HARDWARE						
Ordenador personal+ Hardware	1000	5	100	195	0.106	20.67
SOFTWARE						
Windows 7	125	3	-	195	0.0245	4.77
Microsoft Office 2013	150	3	-	70	0.0295	2.06
Matlab R2010a	6000	3	-	100	1.179	117.92

El coste total de las amortizaciones durante el periodo asciende a **145.42€**

8.1.4. GASTOS DIRECTOS.

El único gasto directo imputable a este proyecto, es la compra de los datos de temperatura al AEMET por un coste de **100 €**.

8.2. COSTES INDIRECTOS.

Los servicios de limpieza, costes derivados de la administración, gastos dirección y gastos de explotación como pueden ser el uso de luz, calefacción, internet... no han sido incluidos en este proyecto debido a la dificultad de imputar estos costes sin prácticamente ningún dato.

8.3. COSTES TOTALES

El resumen de los costes totales del proyecto se puede consultar en la Tabla 28. Se puede observar que al ser un proyecto de investigación realizado íntegramente con ordenador, la mayoría del coste es por el concepto de mano de obra.

Tabla 28 Coste total del proyecto

CONCEPTO	COSTE €
Mano de obra	5978,7
Amortizaciones	145,42
Gastos directos	100
Total	6224,12

9. BIBLIOGRAFÍA

D. Carnicero. Desarrollo de una red neuronal para la estimación de la curva de consumo eléctrico de una instalación hospitalaria empleando datos meteorológicos y splines cúbicos para el pretratamiento de los datos de entrenamiento, Universidad de Valladolid, 2011.

Chen, Toly; and Wang, Yu-Cheng. Long-Term Load Forecasting by a Collaborative Fuzzy-Neural Approach. *International Journal of Electrical Power & Energy Systems*, 12, 2012, vol. 43, no. 1. pp. 454-464. ISSN 0142-0615.

T. W. S. Chow; and C. T. Leung. Neural Network Based Short-Term Load Forecasting using Weather Compensation. *IEEE Transactions on Power Systems*, 1996, vol. 11, no. 4. pp. 1736-1742. ISSN 0885-8950.

Apuntes “Control de la gestión empresarial” Master Ingeniería Industrial Universidad de Valladolid

Escrivá-Escrivá, Guillermo; ROLDÁN-BLAY, Carlos and ÁLVAREZ-BEL, Carlos. Electrical Consumption Forecast using Actual Data of Building End-use Decomposition. *Energy and Buildings*, 10, 2014, vol. 82. pp. 73-81. ISSN 0378-7788.

OECD/IEA, International Energy Agency. [Http://www.iea.org/aboutus/faqs/Energyefficiency/](http://www.iea.org/aboutus/faqs/Energyefficiency/), 2014.

Kaytez, Fazil, et al. Forecasting Electricity Consumption: A Comparison of Regression Analysis, Neural Networks and Least Squares Support Vector Machines. *International Journal of Electrical Power & Energy Systems*, 5, 2015, vol. 67. pp. 431-438. ISSN 0142-0615.

C. N. Lu; H. T. Wu and S. Vemuri. Neural Network Based Short Term Load Forecasting. *IEEE Transactions on Power Systems*, 1993, vol. 8, no. 1. pp. 336-342. ISSN 0885-8950.

Mahmoud, Thair S., et al. Modelling Self-Optimised Short Term Load Forecasting for Medium Voltage Loads using Tuning Fuzzy Systems and Artificial Neural Networks. *Energy Conversion and Management*, 12, 2015, vol. 106. pp. 1396-1408. ISSN 0196-8904.

P. Murto. *Neural Network Models for Short-Term Load Forecasting*, 2008.

Xiao, Liye, et al. A Combined Model Based on Data Pre-Analysis and Weight Coefficients Optimization for Electrical Load Forecasting. *Energy*, 3/15, 2015, vol. 82. pp. 524-549. ISSN 0360-5442.

Yan, Juan, et al. Time Series Wind Power Forecasting Based on Variant Gaussian Process and TLBO. *Neurocomputing*. ISSN 0925-2312.

Yanxia-Lu; and H. f. Shi. *The Hourly Load Forecasting Based on Linear Gaussian State Space Model*. , 2012. ISBN 2160-133X.