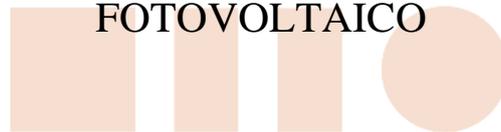


GRADO EN INGENIERÍA ELECTRÓNICA DE
COMUNICACIONES



Trabajo Fin de Grado

PREDICCIÓN MEDIANTE REDES NEURONALES DE LA
PRODUCCIÓN DE ENERGÍA DE UN PARQUE
FOTOVOLTAICO



ESCUELA POLITECNICA

Autor: Fernando Rodríguez Páler

Tutor: Dr. Francisco Javier Rodríguez Sánchez

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

**GRADO EN INGENIERÍA ELECTRÓNICA DE
COMUNICACIONES**

Trabajo Fin de Grado

**PREDICCIÓN MEDIANTE REDES NEURONALES DE LA
PRODUCCIÓN DE ENERGÍA DE UN PARQUE FOTOVOLTAICO**

Autor: Fernando Rodríguez Paler

Tutor: Dr. Francisco Javier Rodríguez Sánchez

TRIBUNAL:

Presidente: Santiago Cóbreces Álvarez

Vocal 1º: Almudena López Dorado

Vocal 2º: Francisco Javier Rodríguez Sánchez

CALIFICACIÓN:

FECHA:

“Ponte de frente al sol y las sombras quedarán detrás de ti.”

Proverbio Mahorí

A mi tutor, Javier:
Por toda la ayuda que me ha ofrecido durante estos meses.

A mis padres, mi hermano y Laura:
Por estar siempre a mi lado.

ÍNDICE GENERAL

1. Resumen	13
2. Abstract	15
3. Resumen extendido	17
4. Energía solar fotovoltaica.....	19
4.1. Introducción	19
4.2. El efecto fotovoltaico	19
4.3. Ventajas del uso de la energía solar fotovoltaica.....	24
4.4. Desventajas del uso de la energía solar fotovoltaica	26
4.5. Desarrollo de la energía solar fotovoltaica en el mundo.....	26
4.6. Elementos de una instalación fotovoltaica.....	28
4.7. Factores que afectan al rendimiento de una instalación fotovoltaica	34
5. Técnicas de estimación.....	37
5.1. Introducción	37
5.2. Modelos de estimación	37
6. Redes neuronales artificiales	45
6.1. Introducción	45
6.2. Elementos de una red neuronal artificial	46
6.3. Topología de las redes neuronales artificiales	48
6.4. Aprendizaje en las redes neuronales artificiales	50
6.5. Sobreentrenamiento en las redes neuronales	51
6.6. El perceptrón multicapa	52
7. Diseño e implementación de la red neuronal para predicción de la producción de un parque solar	55
7.1. Introducción	55
7.2. Localización del parque fotovoltaico.....	55
7.3. Diseño de la red neuronal	59
7.4. Herramienta de desarrollo.....	65
7.5. Creación de la red neuronal	66
7.6. Entrenamiento de la red neuronal	68
7.7. Pruebas sobre la red neuronal entrenada.....	75
7.8. Pruebas de la red neuronal	79
7.9. Modelo teórico de Osterwald.....	89
7.10. Interfaz gráfica de usuario.....	94
8. Conclusiones	101
9. Anexo	103
10. Presupuesto.....	109
11. Manual de usuario	113
12. Bibliografía y referencias	115

ÍNDICE DE FIGURAS

Figura 1. Circuito equivalente de una célula solar real (Fuente: Universidad de Jaén) .	21
Figura 2. Curva I-V de una célula solar (Fuente: EMSD).....	22
Figura 3. Modelado de un panel fotovoltaico.....	23
Figura 4. Huella de carbono en China y en Europa (Fuente: EcoSiglos).....	25
Figura 5. Capacidad y adiciones anuales de energía solar FV, 2005-2015 (Fuente: REN21).....	27
Figura 6. Capacidad y adiciones de energía solar FV, países líderes, 2015 (Fuente: REN21).....	27
Figura 7. Irradiación global horizontal en Europa (Fuente: SolarGIS).....	28
Figura 8. Esquema de una instalación fotovoltaica conectada a la red (Fuente: Paneles Solares PR).....	29
Figura 9. Módulo fotovoltaico (Fuente: Soliclima).....	30
Figura 10. Módulo de silicio monocristalino (Fuente: Jardin-solar).....	31
Figura 11. Módulo de silicio policristalino (Fuente: Jardin-solar).....	32
Figura 12. Módulo de silicio amorfo (Fuente: SoloStocks).....	33
Figura 13. Inversor fotovoltaico (Fuente: ABB).....	33
Figura 14. Hiperplano en una máquina de soporte vectorial (Fuente: OpenCV).....	39
Figura 15. Funcionamiento del algoritmo k-nn (Fuente: iie).....	42
Figura 16. Esquema de los elementos de una neurona biológica (Fuente: Principia Málaga).....	45
Figura 17. Elementos de una red neuronal artificial (Fuente: [25]).....	46
Figura 18. Función de activación lineal (Fuente: [26]).....	47
Figura 19. Función de activación sigmoideal logarítmica (Fuente: [26]).....	48
Figura 20. Función de activación sigmoideal tangente hiperbólica (Fuente: [26]).....	48
Figura 21. Topología de una red neuronal típica (Fuente: Ibermatica RTD).....	49
Figura 22. Método de validación cruzada (Fuente: [27]).....	52
Figura 23. Datos linealmente separables (Fuente: [28]).....	52
Figura 24. Esquema del perceptrón multicapa (Fuente: [29]).....	53
Figura 25. Climograma de Monteroni di Lecce (Fuente: Climate-Data).....	56
Figura 26. Vista aérea del parque fotovoltaico (Fuente: Google Maps).....	57
Figura 27. Sistema fotovoltaico de la universidad de Salento (Fuente: [30]).....	58
Figura 28. Diagrama de la red neuronal diseñada.....	60
Figura 29. Creación del perceptrón multicapa.....	66
Figura 30. Interfaz del entrenamiento.....	69
Figura 31. MSE vs. número de iteraciones.....	70
Figura 32. Estado del entrenamiento.....	71
Figura 33. Gráficas de regresión lineal.....	72
Figura 34. Histograma de los errores.....	73
Figura 35. Esquema de los pesos sinápticos de la red neuronal (Fuente: [31]).....	73
Figura 36. Herramienta HELIOCLIM 3 (Fuente: SoDa).....	76
Figura 37. Prueba de la semana 1.....	81
Figura 38. Prueba de la semana 2.....	82
Figura 39. Prueba de la semana 3.....	83
Figura 40. Prueba de la semana 4.....	84

Figura 41. Prueba de la semana 5	85
Figura 42. Prueba de la semana 6	86
Figura 43. Prueba de la semana 7	87
Figura 44. Prueba de la semana 8	88
Figura 45. Gráfica comparativa de la energía real producida con los dos modelos desarrollados	94
Figura 46. Aspecto de la herramienta GUIDE	96
Figura 47. Aspecto final de la interfaz gráfica de usuario diseñada.....	97
Figura 48. Ejecución de la interfaz gráfica de usuario	99
Figura 49. Guía del manual de usuario.....	114

ÍNDICE DE TABLAS

Tabla 1. Especificaciones técnicas de los módulos fotovoltaicos	57
Tabla 2. Entradas de la base de datos de entrenamiento pre-estandarizado	63
Tabla 3. Entradas de la base de datos de entrenamiento post-estandarizado	64
Tabla 4. Salida de la base de datos de entrenamiento pre-estandarizado	65
Tabla 5. Salida de la base de datos de entrenamiento post-estandarizado	65
Tabla 6. Pesos sinápticos ajustados entre las entradas y la capa oculta	74
Tabla 7. Pesos sinápticos ajustados entre la capa oculta y la salida	74
Tabla 8. MAPE en función del número de neuronas de la capa oculta	80
Tabla 9. Valores estimados de la semana 1	81
Tabla 10. Valores estimados de la semana 2	82
Tabla 11. Valores estimados de la semana 3	83
Tabla 12. Valores estimados de la semana 4	84
Tabla 13. Valores estimados de la semana 5	85
Tabla 14. Valores estimados de la semana 6	86
Tabla 15. Valores estimados de la semana 7	87
Tabla 16. Valores estimados de la semana 8	88
Tabla 17. Pérdidas mensuales del sistema fotovoltaico	91
Tabla 18. Valores de energía estimados mediante el modelo de Osterwald	93
Tabla 19. MAPE semanal promedio calculado para el modelo teórico de Osterwald ...	93
Tabla 20. Costes de hardware	109
Tabla 21. Costes de adquisición del material	109
Tabla 22. Costes de personal	110
Tabla 23. Coste de ejecución material	110

1. Resumen

El presente Trabajo Fin de Grado trata de un estudio sobre la predicción de la producción de energía de un parque fotovoltaico. Para realizar la simulación se ha elegido un parque situado en Monteroni di Lecce (Italia), cuya potencia nominal es de 960 kW.

Se han desarrollado diferentes modelos basados en redes neuronales así como un modelo matemático, con el objetivo de predecir con el menor error posible la energía producida en el parque fotovoltaico. Para el desarrollo de las redes neuronales ha sido necesaria una extensa base de datos que recoge diversos parámetros meteorológicos y técnicos del parque fotovoltaico.

Palabras clave: Energía solar fotovoltaica, redes neuronales, perceptrón multicapa, predicción.

2. Abstract

This Final Project is a study about the forecast of energy production of a photovoltaic farm. In order to carry out the simulation a farm has been chosen located in Monteroni di Lecce (Italy), whose nominal power is 960 kW.

Different models based on neural networks have been developed as well as a mathematical model, with the goal of forecasting the energy produced in the photovoltaic farm with the least error. For the development of the neural networks has been necessary a large data base that collects diverse meteorological and technical parameters of the photovoltaic farm.

Key words: Photovoltaic solar energy, neural networks, multilayer perceptron, forecasting.

3. Resumen extendido

La estimación de la producción de energía solar fotovoltaica permite gestionar y, en caso necesario, almacenar la energía producida en los parques fotovoltaicos de manera eficiente. Uno de los motivos de la importancia que están adquiriendo los diferentes modelos de predicción fotovoltaica es que la capacidad anual de energía solar fotovoltaica en el mundo se ha incrementado exponencialmente en los últimos diez años. En la actualidad, se trata del tipo de energía más barata en un gran número de países debido a que se han abaratado sustancialmente los costes de producción de módulos fotovoltaicos.

Por ello, se ha realizado un estudio sobre los diferentes modelos de estimación existentes para predecir la producción de energía de un parque fotovoltaico. Entre estos modelos de predicción, destaca el modelo basado en redes neuronales. Por ello, se van a estudiar las posibilidades que ofrece dicho modelo, y se diseñará e implementará un modelo basado en el perceptrón multicapa.

A la hora de diseñar un perceptrón multicapa, se debe tener especial cuidado al elegir su topología (número de capas, número de entradas, número de salidas, técnicas de entrenamiento, etc.). También será necesario disponer de una base de datos adecuada que contenga los registros capturados del parque fotovoltaico para poder llevar a cabo el entrenamiento de la red neuronal. El número de parámetros que debe contener la base de datos depende de la aplicación. En el caso de la predicción de energía fotovoltaica, interesa que la base de datos contenga registros como irradiancia y temperatura, ya que son los principales factores que afectan a la producción de energía.

Se ha elegido un parque fotovoltaico situado en Monteroni di Lecce (Italia), para llevar a cabo las pruebas de la red neuronal. Este parque tiene una potencia nominal de 960 kW, y su base de datos satisface todos los requerimientos deseables.

Una vez procesados los registros de la base de datos del sistema fotovoltaico, se lleva a cabo el entrenamiento y las pruebas de la red neuronal. Para obtener los mejores resultados posibles, se han hecho numerosas pruebas variando la topología de la red, hasta encontrar el modelo de menor error.

Además del modelo basado en redes neuronales, se ha empleado un modelo matemático a los efectos de comparación, para evaluar cuál de los dos modelos es más adecuado para llevar a cabo tareas de predicción.

Por último, se ha desarrollado una interfaz gráfica de usuario en la que se ha implementado el modelo de estimación basado en el perceptrón multicapa, para que cualquier usuario sin conocimientos de programación ni de redes neuronales pueda llevar a cabo estimaciones de producción de energía en el parque fotovoltaico.

4. Energía solar fotovoltaica

4.1. Introducción

La energía solar es aquella obtenida mediante la captación de la radiación electromagnética procedente del sol.

La potencia de la radiación variará a lo largo del día en función de diferentes parámetros, como las condiciones meteorológicas y atmosféricas, la rotación de la tierra, la inclinación de los rayos solares, etc. La magnitud utilizada para indicar el valor de la radiación solar incidente sobre una superficie recibe el nombre de irradiancia, la cual se mide en W/m^2 , siendo la irradiancia máxima procedente del Sol en la superficie terrestre de aproximadamente $1000 W/m^2$ en condiciones óptimas.

4.2. El efecto fotovoltaico

El efecto fotovoltaico fue demostrado primeramente por Edmond Becquerel en 1839, pero no fue hasta 1883 cuando Charles Fritts construyó la primera célula fotovoltaica de estado sólido, con un 1% de eficiencia.

La energía solar fotovoltaica se obtiene mediante la transformación directa de la radiación solar en energía eléctrica [1]. Para ello son necesarias las células fotovoltaicas, que cuales son dispositivos electrónicos que están formados por diferentes láminas de material semiconductor, las cuales han sido dopadas de manera diferente (tipo "P" y tipo "N").

Para dopar dichas láminas, se debe añadir al material cantidades medidas de ciertas impurezas. De esta forma, los semiconductores presentarán conductividad controlada y relativamente constante dentro de un margen de temperatura amplio.

Se produce un semiconductor tipo "P" si se introduce una pequeña concentración de impurezas aceptoras de valencia 3 dentro de la estructura de la red cristalina. Se conocen como impurezas aceptoras puesto que cuando la red cristalina adquiere energía térmica, uno de los electrones de los átomos de silicio será aceptado por un átomo de impureza.

Por otro lado, para producir un semiconductor tipo “N” se introduce una pequeña concentración de impurezas donadoras de valencia 5 dentro de la estructura de la red. Se conocen como impurezas donadoras porque donan electrones libres.

Al colocarse una parte del semiconductor tipo “P” junto con otra parte del semiconductor tipo “N”, los electrones de la zona “N” tienden a dirigirse a la zona “P”, mientras que los huecos de la zona “P” tienden a dirigirse a la zona “N”. Este fenómeno se conoce como corriente de difusión, el cual provoca que en la zona de la unión se produzca una estructura estable debido a la neutralización electrón-hueco.

Puesto que la zona “P” tiende a perder huecos, se volverá cada vez más negativa mientras que la zona “N”, como tiende a perder electrones, se volverá cada vez más positiva, lo cual dejará al descubierto algunos iones fijos en la red cristalina (iones aceptores en la zona “P” e iones donadores en la zona “N”). La presencia de estas cargas fijas generará un campo eléctrico (barrera de potencial) cuyas líneas de campo se dirigen de la zona “N” a la zona “P”, es decir, de manera opuesta a la corriente de difusión.

Para que se produzca el efecto fotovoltaico debe cumplirse la siguiente condición:

$$E_{Fotón} = \frac{h * c}{\lambda} \geq E_g \quad (1)$$

Donde h es la constante de Planck, c es la velocidad de la luz y λ es la longitud de onda de la radiación solar. Por otro lado, la energía del fotón dependerá de la variable λ y presentará la siguiente forma:

$$E_{Fotón}(eV) = \frac{1240}{\lambda(nm)} \quad (2)$$

La radiación solar está compuesta por fotones (partículas responsables de las manifestaciones cuánticas del fenómeno electromagnético) que, cuando inciden sobre una célula fotovoltaica, pueden transferir su energía a un electrón de un átomo de la célula fotovoltaica, si impactan con él. Para ello, la energía

del fotón debe ser mayor que un valor mínimo determinado el cual depende del material, denominado GAP de energías prohibidas (E_g). Cuando los fotones disponen de la energía suficiente, inciden en la superficie de la capa "P", liberando electrones de los átomos de silicio. Estos electrones generarán un movimiento a través de la capa de semiconductor. El campo eléctrico creado mediante la unión P-N permite el flujo de corriente en un único sentido a través de la unión. En la Figura 1 se muestra el circuito equivalente de una célula solar real.

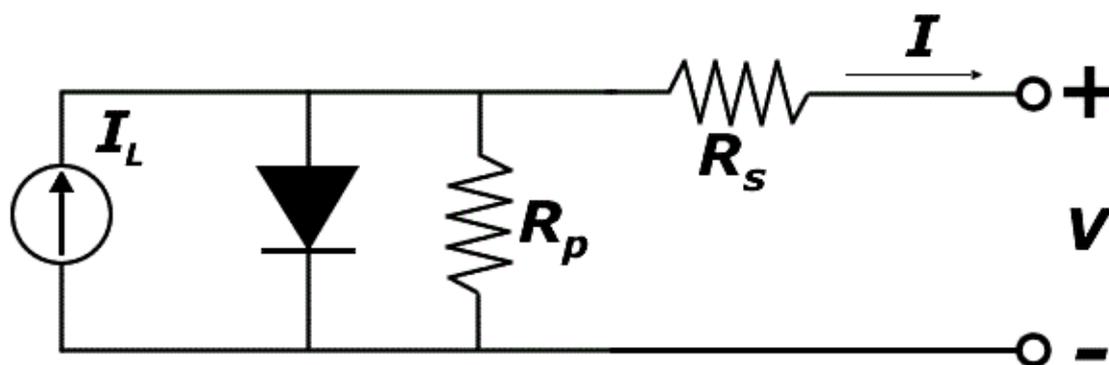


Figura 1. Circuito equivalente de una célula solar real (Fuente: Universidad de Jaén)

La curva I-V (Figura 2) permite caracterizar el comportamiento eléctrico de las células solares, y se obtiene a partir del circuito equivalente de las mismas. En la curva se pueden destacar los siguientes parámetros:

- Corriente de cortocircuito: Se trata de la corriente existente en la célula cuando la tensión entre sus terminales es nula. Se expresa como I_{sc} . La corriente de cortocircuito depende de factores como el número de fotones y las propiedades ópticas de la célula, etc.
- Tensión de circuito abierto: Se trata de la tensión existente en la célula ante ausencia de carga e intensidad nula. Se expresa como V_{oc} . Depende principalmente de la corriente I_o .
- Punto de máxima potencia: Se trata del producto de la tensión en el punto de máxima potencia (V_{mp}) por la corriente en el punto de máxima potencia (I_{mp}). Se expresa como PMP.

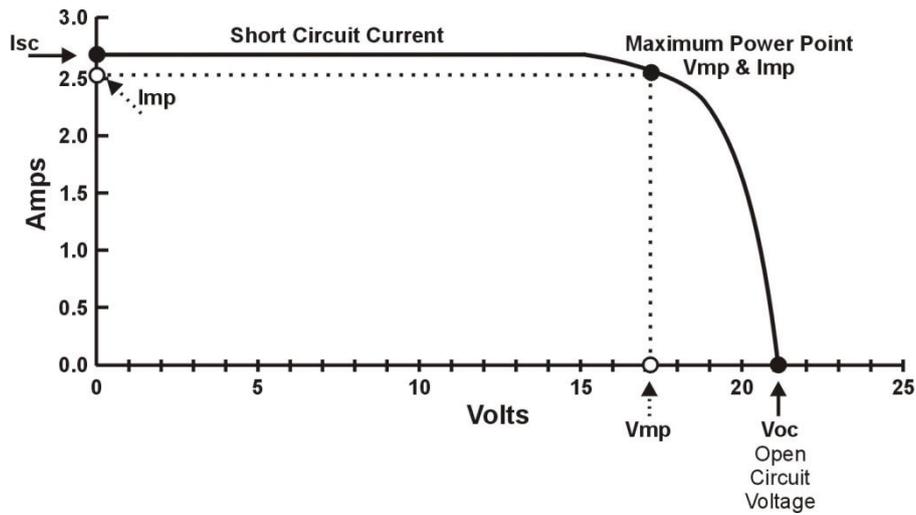


Figura 2. Curva I-V de una célula solar (Fuente: EMSD)

En la siguiente expresión (3) se representa la curva tensión-corriente de una célula fotovoltaica:

$$I = I_{sc} * \left(1 - e^{\frac{v-V_{oc}}{n_s * V_T}} \right) \quad (3)$$

Donde n_s es el número de células en serie y V_T se trata de la tensión térmica ($V_T = \frac{K_B * T}{e}$, siendo K la constante de Boltzman, T la temperatura y e la carga del electrón.)

Un aumento de la temperatura producido en una célula solar, reducirá el GAP del semiconductor, afectando negativamente a todos los parámetros de funcionamiento de la célula. El parámetro más afectado será la tensión de circuito abierto (V_{oc}) que disminuirá aproximadamente 2,3 mV/°C. Además, aumentará ligeramente la corriente de cortocircuito (I_{sc}) y, en conjunto, disminuirá el rendimiento de la célula.

Las expresiones (4), (5) y (6) establecen la dependencia existente entre la corriente de cortocircuito, la tensión de circuito abierto y la potencia, con la temperatura.

$$I_{sc} = I_{sc,25} * (1 + \alpha_I * \Delta T) \quad (4)$$

$$V_{oc} = V_{oc,25} * (1 + \alpha_V * \Delta T) \quad (5)$$

$$P = P_{25} * (1 + \alpha_p * \Delta T) \tag{6}$$

Donde ΔT es el incremento de temperatura (temperatura [°C] – 25°C), y los coeficientes de temperatura α_I , α_V y α_P son proporcionados por el fabricante de la célula.

Las células solares se pueden agrupar en paneles fotovoltaicos mediante la interconexión de varias células entre sí. Para construir un panel fotovoltaico, primeramente se deben asociar las células en serie hasta conseguir el nivel de tensión deseado. A continuación, se deben realizar asociaciones en paralelo de varias asociaciones en serie hasta conseguir el nivel de corriente deseado. En la Figura 3 se muestra un posible modelado de un panel fotovoltaico.

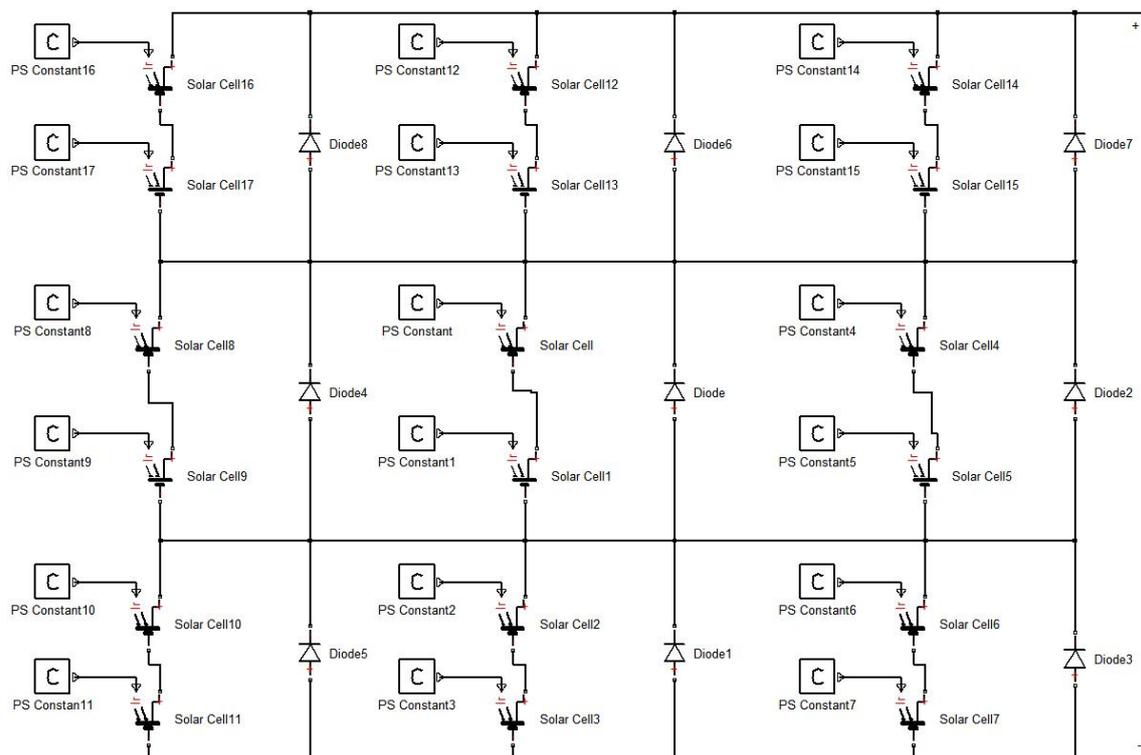


Figura 3. Modelado de un panel fotovoltaico

4.3. Ventajas del uso de la energía solar fotovoltaica

La energía solar fotovoltaica es una de las grandes alternativas existentes para hacer frente a la excesiva dependencia de los combustibles fósiles. La principal ventaja del uso de la energía solar fotovoltaica respecto a otras fuentes de energía convencionales es que se trata de una energía “limpia”, no consume combustibles fósiles de manera directa (aunque puede darse el caso de que para producir los paneles fotovoltaicos sí que se empleen combustibles fósiles), y durante su vida útil no genera residuos (una vez finalizada, los paneles fotovoltaicos se vuelven un residuo reciclable en vidrio y otros materiales, mediante procesos que deben investigarse todavía para mejorarse). El hecho de que se trate de una fuente de energía “limpia” se traduce en un menor impacto ambiental y una menor contribución al cambio climático (generado por las excesivas emisiones de CO_2), aunque la huella de carbono de las células fotovoltaicas no es cero puesto que se necesita energía para llevar a cabo el proceso de fabricación, transporte e instalación.

En la Figura 4 se muestran la cantidad de gramos de CO_2 generados por kWh en China y en Europa debidos a la energía solar fotovoltaica. Como se puede observar, la huella de carbono de los paneles fotovoltaicos producidos en China es mayor que la huella de los paneles producidos en Europa. Esto es debido a que las fábricas de los paneles fotovoltaicos en China dependen en gran medida de la energía eléctrica procedente en su mayoría del carbón, por lo que generan más CO_2 .

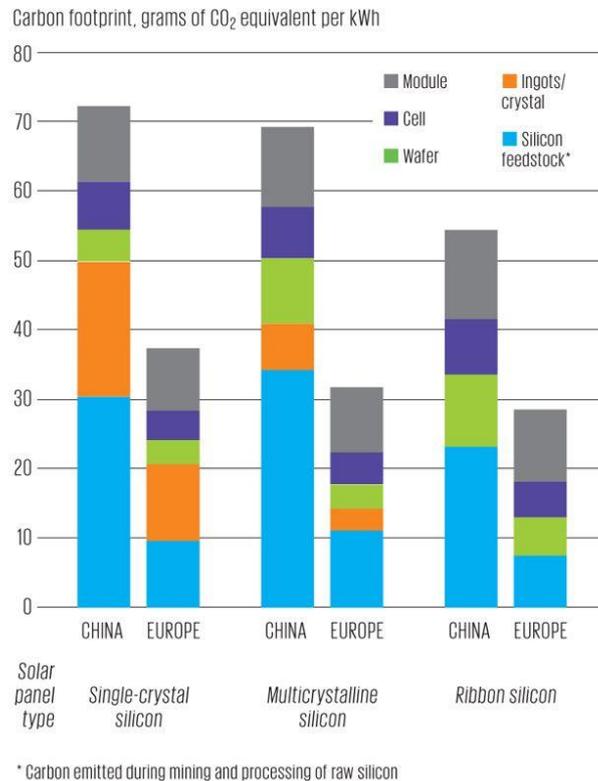


Figura 4. Huella de carbono en China y en Europa (Fuente: EcoSiglos)

La energía solar fotovoltaica se trata de una fuente de energía con un potencial ilimitado. El Sol irradia aproximadamente $3,8 \times 10^{26}$ W cada segundo, de los cuales la superficie terrestre recibe 120.000 TW de irradiación solar como consecuencia de la distancia entre la Tierra y el Sol, es decir, 1,37 kW por metro cuadrado en el exterior de la atmósfera.

Además de las mencionadas anteriormente, tiene otras ventajas ambientales como la ausencia de ruido (debido a que no existen partes móviles que generen contaminación acústica).

La energía solar fotovoltaica también tiene ventajas económicas, como el bajo mantenimiento (los paneles fotovoltaicos instalados para uso doméstico solamente precisan una o dos limpiezas anuales, y suelen tener garantías de hasta 25 años). También puede incrementarse el tamaño de la instalación añadiendo módulos fotovoltaicos y en el caso de un sistema fotovoltaico conectado a red, se puede mejorar la calidad del servicio de la energía suministrada por la red, debido a que la máxima producción del sistema

fotovoltaico suele coincidir con las horas en las que hay más problemas de suministro eléctrico por parte de las compañías eléctricas.

4.4. Desventajas del uso de la energía solar fotovoltaica

A pesar de las ventajas existentes del uso de la energía solar fotovoltaica, también existen algunos inconvenientes. El principal inconveniente económico radica en la técnica de fabricación de los paneles fotovoltaicos; aunque la principal materia prima usada (silicio) es muy abundante, requiere técnicas complejas y caras. Además, puesto que las células fotovoltaicas no tienen una gran eficiencia, el rendimiento es bastante bajo (en la mayoría de los casos inferior al 20%).

Otra de las desventajas es la dependencia de la producción con la climatología del lugar de instalación, así como el impacto visual de los parques fotovoltaicos, que suelen ocupar una gran superficie de terreno.

Por último, la energía solar fotovoltaica presenta otro inconveniente y es la necesidad de instalar un sistema de almacenamiento de energía en los sistemas fotovoltaicos autónomos.

4.5. Desarrollo de la energía solar fotovoltaica en el mundo

La energía solar fotovoltaica ha experimentado un crecimiento exponencial a lo largo de los años, en parte debido a la necesidad de potenciar las energías renovables respecto al uso de combustibles fósiles, lo que se ha traducido en mecanismos de fomento en muchos países. En [2] se presenta un informe anual del año 2016 sobre la situación de las energías renovables en el mundo.

A finales del año 2005, como se puede ver en la Figura 5, la potencia total de energía fotovoltaica instalada en el mundo era de aproximadamente 5,1 GW. Este dato fue aumentando rápidamente, alcanzando los 40 GW en 2010 (8 veces más que en 2005) y los 227 GW que hay instalados aproximadamente en la actualidad (último registro del año 2015).

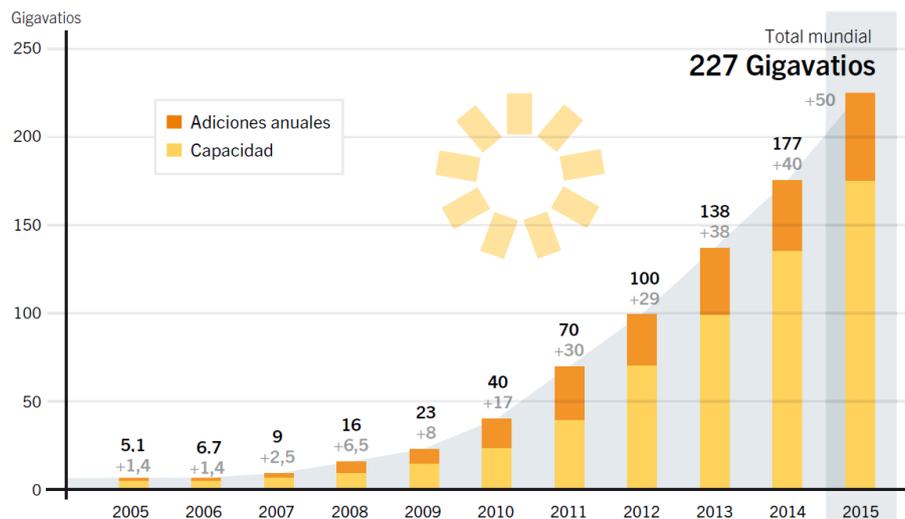


Figura 5. Capacidad y adiciones anuales de energía solar FV, 2005-2015 (Fuente: REN21)

En el año 2015, la capacidad mundial de energía solar fotovoltaica se incrementó cerca de un 25% con respecto al 2014 (aproximadamente se incrementó la capacidad 50 GW). Los países que más incrementaron su capacidad en el año 2015 fueron China, Japón y Estados Unidos respectivamente (más de la mitad de la capacidad instalada durante dicho año corresponde a estos tres países).

En la Figura 6 se muestra el panorama actual de los diez países líderes del mundo en cuanto a capacidad de energía solar fotovoltaica instalada.

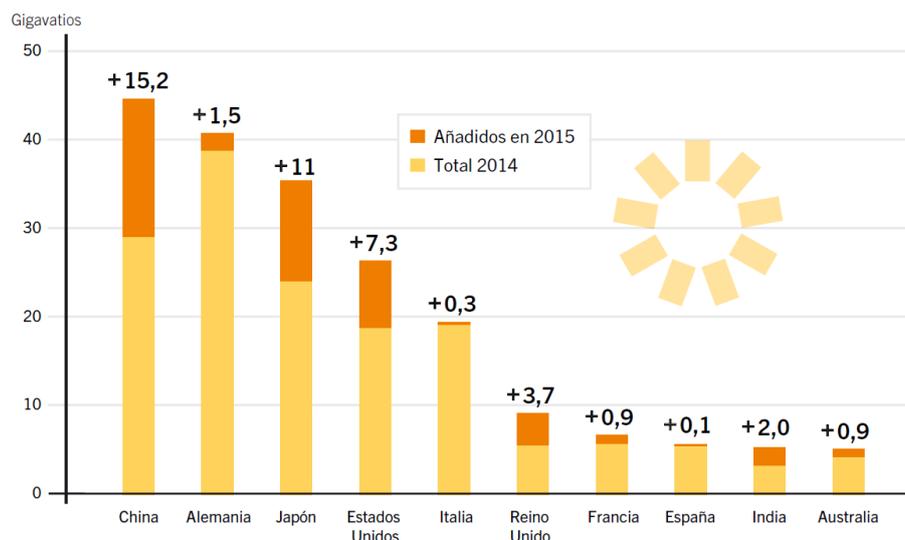


Figura 6. Capacidad y adiciones de energía solar FV, países líderes, 2015 (Fuente: REN21)

Como se puede observar, existen países que a priori tienen menor potencial de energía solar fotovoltaica (como por ejemplo, Alemania) que están

entre los países con mayor capacidad instalada, y sin embargo países como España o Italia que debido a sus condiciones climatológicas podrían ser de los países con mayor producción fotovoltaica, se quedan atrás.

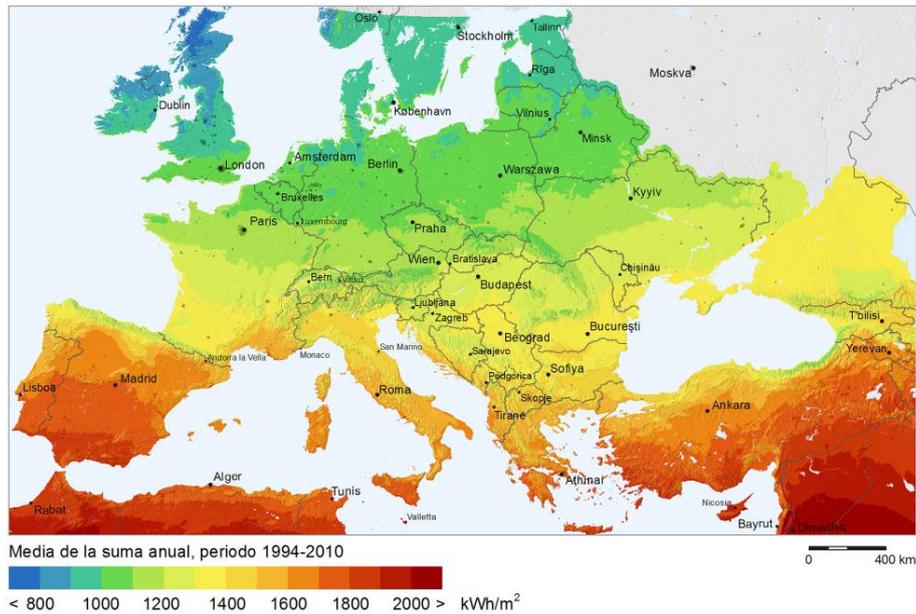


Figura 7. Irradiación global horizontal en Europa (Fuente: SolarGIS)

En el caso de España, en el año 2008 se instaló el 40% de la capacidad de energía solar fotovoltaica instalada en todo el mundo, a día de hoy la cifra es mucho menor. En el año 2015 solamente se instalaron en España 49 MW, apenas un 0,09% de la capacidad instalada a nivel mundial durante dicho año. Para cambiar esta tendencia se deben tomar medidas para que este tipo de energía sea respaldada por las leyes y se debe facilitar el autoconsumo.

4.6. Elementos de una instalación fotovoltaica

Existen dos tipos de instalaciones fotovoltaicas, los sistemas fotovoltaicos autónomos y los sistemas fotovoltaicos conectados a la red.

Los sistemas fotovoltaicos autónomos están formados por, los módulos fotovoltaicos, un sistema de almacenamiento de energía (baterías), un regulador, un inversor y la carga. El objetivo de este tipo de instalaciones es cubrir todas las necesidades de las cargas instaladas mediante la combinación de generación y almacenamiento de energía. Para ello, además de los paneles solares se dispone de un sistema de baterías donde se acumula el exceso de

generación, para que pueda emplearse posteriormente en los periodos en los que la generación eléctrica sea menor a la demanda energética. La finalidad del regulador es la de cerrar el paso de corriente cuando las baterías se encuentren completamente cargadas; por su parte, el inversor, es el encargado de convertir la corriente continua que entregan los módulos fotovoltaicos en corriente alterna.

En la Figura 8 se muestra la estructura típica de un sistema fotovoltaico conectado a la red. En este tipo de instalaciones el sistema opera en paralelo con la red eléctrica. Los elementos principales que lo forman son los módulos fotovoltaicos y un inversor, el cual se encuentra conectado entre los módulos fotovoltaicos y la red eléctrica. La peculiaridad de este tipo de sistemas es que se puede mejorar la calidad del servicio suministrado por la red eléctrica, ya que normalmente la producción máxima del sistema fotovoltaico suele coincidir con los periodos en los que existen mayores problemas de suministro por parte de las compañías eléctricas.

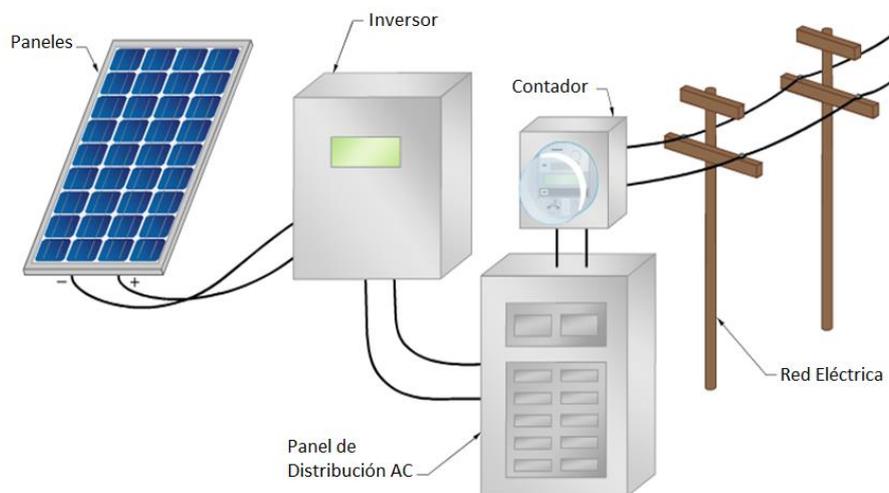


Figura 8. Esquema de una instalación fotovoltaica conectada a la red (Fuente: Paneles Solares PR)

A continuación, se explicarán en detalle los elementos que forman un sistema fotovoltaico conectado a red.

4.6.1. Módulo fotovoltaico

Se conoce como módulo fotovoltaico al conjunto de células fotovoltaicas conectadas en serie y en paralelo hasta conseguir la potencia nominal

necesaria para satisfacer los requerimientos energéticos deseados. Aproximadamente el 84% de la energía solar que incide sobre un módulo fotovoltaico se pierde en forma de calor, es por ello por lo que se requieren grandes instalaciones de módulos fotovoltaicos para conseguir potencias elevadas.

Lo más común es encontrar paneles de 1.5, 6, 12, 24 y 48 voltios, pudiéndose conseguir cualquier requerimiento de potencia mediante la asociación de éstos.



Figura 9. Módulo fotovoltaico (Fuente: Soliclima)

Una vez elegido el número de células que van a formar el panel, se ensamblan y se encapsulan adecuadamente para proteger los contactos entre las células que forman el panel. Para llevar a cabo el encapsulamiento se deben buscar siempre materiales que proporcionen impermeabilidad, protección al polvo y aislamiento eléctrico, como por ejemplo el etilvinilacetato (EVA).

Puesto que las células son dispositivos frágiles, será necesario colocar dos capas de protección, una superior y una inferior. La protección superior tiene la finalidad de proteger el módulo de las inclemencias atmosféricas. Además, el material debe ser resistente a impactos y debe facilitar la transmisión de la radiación solar. Uno de los materiales con mejores prestaciones para la fabricación de la protección superior es el cristal templado con bajo contenido en hierro.

La protección inferior tiene como objetivo proporcionar al módulo protección frente a la humedad, y para ello se utilizan materiales acrílicos o polímeros (como el *Tedlar*).

4.6.1.1. Tipos de módulos fotovoltaicos

En la actualidad existen diferentes tipos de módulos fotovoltaicos que se clasifican según su proceso de fabricación y composición. A continuación, se explicarán en detalle los principales:

- Silicio monocristalino: El silicio puro monocristalino se obtiene del silicio puro fundido y dopado con boro. Se caracteriza por tener un color azul homogéneo y por la conexión de las células individuales entre sí. Bajo ensayos de laboratorio el rendimiento de este tipo de módulos es del 24%, sin embargo, una vez instalados suelen alcanzar un rendimiento real del 18%. Se trata del tipo de módulo con mayor rendimiento y el que tiene una mayor vida útil, pero también es el más caro debido a su tecnología de fabricación y a su alta pureza.



Figura 10. Módulo de silicio monocristalino (Fuente: Jardin-solar)

- Silicio policristalino: El silicio policristalino al igual que el monocristalino, se obtiene del silicio puro fundido. La principal diferencia es que el policristalino reduce el número de fases de cristalización. Se caracteriza por su superficie estructurada en cristales y las diferentes tonalidades azules que lo forman. En laboratorio, el rendimiento de este tipo de paneles es de

aproximadamente un 20%, sin embargo, en el caso real suele alcanzar un rendimiento del 14%. Puesto que tiene un método de fabricación más simple, resulta más barato que el monocristalino, pero tiene el inconveniente de que es menos resistente al calor que los monocristalinos por lo que a altas temperaturas funcionará peor que un módulo monocristalino.

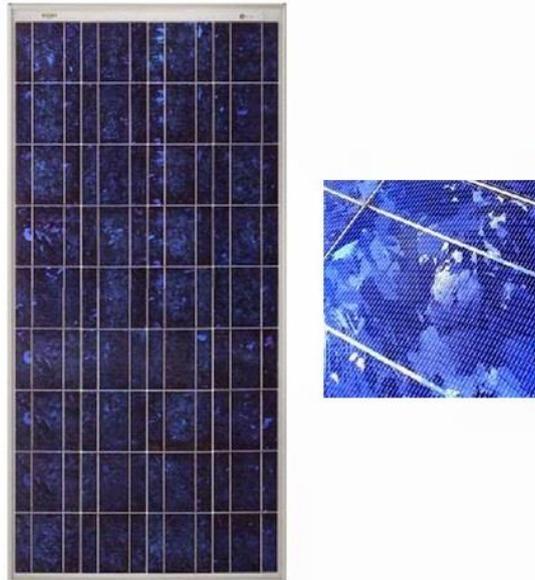


Figura 11. Módulo de silicio policristalino (Fuente: Jardin-solar)

- **Silicio amorfo**: El silicio amorfo no tiene estructura cristalina, si no que se deposita en forma de lámina sobre un sustrato. Se caracteriza por su color homogéneo marrón y la ausencia de conexión visible entre las células. Tiene un 16% de rendimiento en laboratorio, y en el caso real una vez instalado no llega al 10%. Debido a su bajo precio, se utiliza sobre todo para pequeña electrónica de consumo.



Figura 12. Módulo de silicio amorfo (Fuente: SoloStocks)

4.6.2. Inversor

El inversor fotovoltaico es el dispositivo encargado de transformar la energía generada por los módulos fotovoltaicos, para adaptarla a las necesidades de la carga o para inyectarla a la red eléctrica. Para ello, el inversor convierte la corriente continua producida por los módulos en corriente alterna. Al hacerlo debe tener en cuenta las características específicas de la red eléctrica (50Hz de frecuencia y 220V eficaces de tensión en España).



Figura 13. Inversor fotovoltaico (Fuente: ABB)

A la hora de planificar la instalación fotovoltaica, es muy importante elegir el inversor fotovoltaico adecuadamente ya que una gran parte de las pérdidas energéticas que se van a producir en la instalación provendrán del inversor. Para ello hay que buscar un inversor que tenga alta eficiencia (de

modo que pueda funcionar correctamente para un gran rango de potencias), así como alta fiabilidad. Además, se deberá buscar una buena regulación de tensión, para adaptar la energía generada a las características de la red eléctrica, lo que permitirá inyectar dicha energía en la red. Otra característica deseable de los inversores fotovoltaicos es que el consumo sea pequeño ante la ausencia de cargas conectadas.

Se debe tener especial cuidado a la hora de elegir la potencia nominal del inversor fotovoltaico. Siempre se deberá elegir un inversor cuya potencia nominal sea aproximadamente un 20% superior a la suma de todas las potencias demandadas por todas las cargas que forman parte de la instalación en un instante.

4.7. Factores que afectan al rendimiento de una instalación fotovoltaica

Existen una serie de factores que pueden provocar una disminución del rendimiento de una instalación fotovoltaica. Entre los más importantes están los siguientes:

4.7.1. Pérdidas debidas al polvo

Este tipo de pérdidas tienen lugar debido a la existencia de suciedad y polvo en la superficie de los módulos fotovoltaicos. Si la suciedad está localizada uniformemente, se producirá una disminución en la tensión y la corriente entregadas por el generador. Por el contrario, si la suciedad está localizada de manera puntual se producirá un aumento de pérdidas de *mismatch*, explicadas en el siguiente apartado.

4.7.2. Pérdidas de conexionado (mismatch)

Este tipo de pérdidas se produce por la interconexión de células solares con características eléctricas diferentes. Si se conectan dos módulos fotovoltaicos en serie, y ambos tienen diferentes corrientes (una causa puede ser la existencia de polvo, como se ha visto anteriormente), la corriente total de la asociación serie será la menor de las corrientes. De la misma forma, si se conectan dos módulos fotovoltaicos en paralelo y ambos tienen diferentes

tensiones, la potencia total del sistema fotovoltaico será menor que la suma de las potencias de cada uno de los módulos que componen la instalación.

Puesto que la potencia está directamente relacionada con la tensión y la corriente, cuando se produzcan pérdidas de *mismatch* se producirá una pérdida de potencia.

Las pérdidas de *mismatch* se pueden reducir mediante la utilización de diodos de “*bypass*”, que permiten que el módulo de menores prestaciones sea cortocircuitado hasta que sus parámetros eléctricos se igualen con el del resto de módulos.

4.7.3. Pérdidas debidas al sombreado

El sombreado es uno de los mayores problemas que hay que tener en cuenta a la hora de emplazar una instalación fotovoltaica. En entornos urbanos resulta muy complicado evitar este problema. Cuando se produce el sombreado de una célula que forma parte de una asociación serie, dicha célula se comportaría como una resistencia y aparecería una tensión inversa en ella, consumiendo potencia generada por las demás células de la asociación (y produciendo un calentamiento de la célula sombreada). Esto se conoce como efecto “*hot spot*” (“punto caliente”).

4.7.4. Pérdidas debidas a la temperatura

Las pérdidas debidas a la temperatura dependen directamente de factores como la irradiancia y la temperatura normal de operación de la célula (NOCT, que se obtiene a partir de unos valores normalizados de temperatura ambiente e irradiancia). En el caso práctico, la potencia que es capaz de entregar el módulo fotovoltaico disminuye aproximadamente 0,5% por cada grado centígrado que aumenta la temperatura de la célula a partir de 25 °C.

4.7.5. Pérdidas debidas al inversor

Tal y como se ha mencionado anteriormente, es importante elegir un inversor con un rendimiento alto en condiciones nominales de operación, ya que cuanto más alto sea el rendimiento menor serán las pérdidas asociadas al inversor.

Además, es importante elegir correctamente la potencia nominal del inversor, ya que, si la potencia nominal del inversor es demasiado grande comparada con la potencia nominal de los módulos fotovoltaicos, el sistema trabajará en un rango de valores de rendimiento bajo.

4.7.6. Pérdidas angulares

Cuando se habla de la potencia nominal de un módulo fotovoltaico, se suele referir a la potencia máxima que puede entregar dicho módulo bajo determinadas condiciones estándar (irradiancia de 1000 W/m² y temperatura de la célula de 25 °C, con una incidencia normal), pero cuando en la práctica la incidencia de los rayos no es perpendicular al módulo fotovoltaico tienen lugar una serie de pérdidas que variarán en función del grado de inclinación de la radiación incidente.

4.7.7. Pérdidas debidas a la tolerancia

Debido al proceso de fabricación y a los diferentes elementos que forman un módulo fotovoltaico, la potencia nominal del mismo puede variar ligeramente respecto a la indicada en la ficha técnica del mismo debido a la tolerancia que presenta.

Normalmente la potencia nominal de los módulos fotovoltaicos se encuentra en la zona negativa de la tolerancia, por lo que lo más común será que la potencia real de los módulos fotovoltaicos sea menor que la que indica su hoja de características.

5. Técnicas de estimación

5.1. Introducción

Debido al carácter errático de producción que tienen las energías renovables en general, y la energía solar fotovoltaica en particular, es importante encontrar un sistema que permita estimar de manera certera la cantidad de energía que puede producir un parque fotovoltaico.

La estimación fotovoltaica permite llevar a cabo una planificación energética apropiada, lo que ayudará a realizar un cálculo de la reserva de generación, lo cual tendrá consecuencias económicas positivas. Además, otro de los motivos de la importancia de la estimación es poder dar solución a los problemas que pueden surgir en la red eléctrica.

Existen numerosas técnicas de estimación que se pueden clasificar de diversas formas. Uno de los métodos de clasificación sería aquel que se divide en técnicas que utilizan únicamente valores pasados para llevar a cabo la estimación, y técnicas que utilizan como entradas otras predicciones que tienen su origen en herramientas de predicción meteorológica (además de poder utilizar valores pasados).

5.2. Modelos de estimación

A continuación, se verán en detalle cinco de los principales modelos de estimación, así como las características de cada uno de ellos.

5.2.1. Modelos basados en redes neuronales

Los modelos basados en redes neuronales se desarrollaron a partir de 1940 gracias al trabajo de McCulloch y Pitts [3].

Las redes neuronales artificiales consisten en un modelo matemático basado en el sistema nervioso biológico, formado por un gran número de elementos simples que procesan información mediante su estado dinámico como respuesta a entradas externas.

Las unidades básicas del modelo son las neuronas. Cada una de estas neuronas estará interconectada con las entradas de la red neuronal y/o con los

diferentes elementos del modelo, además de tener un peso asociado. La respuesta a los impulsos de entrada será la salida de la red neuronal.

Las etapas principales del modelado basado en redes neuronales son: elección de las variables de entrada, tipo de red y número de capas, preparación del conjunto de datos, creación de la red neuronal, entrenamiento de la red neuronal y validación de los resultados.

Las series de tiempo de la carga eléctrica son usualmente funciones no lineales de variables externas, por lo que debido a esta no linealidad, las redes neuronales artificiales reciben gran atención en la solución de problemas como el precio de la electricidad. Un ejemplo de esto es el estudio presentado en [4] donde se utilizan las redes neuronales (ANN) para predecir a corto plazo los precios de la electricidad por hora (para el día siguiente o dos días después).

En [5], se desarrolla un método para determinar la generación de un panel fotovoltaico usando redes neuronales artificiales para obtener un valor adecuado de la potencia suministrada por la instalación fotovoltaica. Específicamente este método se ha utilizado para obtener la potencia proporcionada por una instalación en particular.

Más adelante se explicará en detalle todo lo relacionado con este modelo puesto que ha sido el modelo elegido para llevar a cabo gran parte del estudio realizado.

5.2.2. Modelos basados en máquinas de soporte vectorial

Las máquinas de soporte vectorial constituyen una técnica de predicción asociada al campo de la inteligencia artificial. El algoritmo fue desarrollado por Vladimir Vapnik [6] y está altamente ligado a las redes neuronales.

El objetivo de esta técnica se basa en construir un hiperplano en un espacio de dimensionalidad muy alta que sea capaz de separar las clases existentes. La separación entre el punto más próximo de cada clase debe ser la máxima posible, por lo que será primordial encontrar el hiperplano adecuado.

Se deberán diferenciar dos tipos de casos, aquellos en los que las clases son linealmente separables y los que no lo son.

En aquellos casos en los que las muestras son linealmente separables, existirá el concepto de margen, el cual se define como la suma de las distancias de los puntos más cercanos al hiperplano. El hiperplano óptimo puede ser descrito como la combinación de varios puntos llamados vectores de soporte. El objetivo en este tipo de casos será el de maximizar dicho margen.

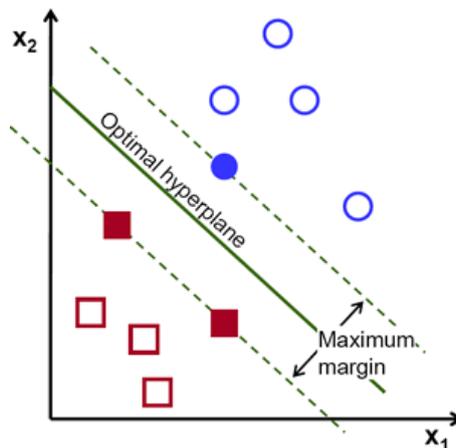


Figura 14. Hiperplano en una máquina de soporte vectorial (Fuente: OpenCV)

Cuando los casos no sean linealmente separables, no existirá un hiperplano que separe linealmente los datos. En estos casos se deberá asumir cierto error que permita llevar a cabo la clasificación. Este concepto se conoce como holgura (ξ). Además, existirá un parámetro de regularización (C) que no deberá ser ajustado en la formulación de la máquina de soporte vectorial. El objetivo de este parámetro es realizar un balance entre la maximización del margen y la violación de la clasificación. Una vez asumido esto, se deberá encontrar el máximo margen como si se tratase de un caso cuyas clases son linealmente separables.

En [7] se presenta un estudio en el que se desarrolla un modelo de predicción fotovoltaica basado en máquinas de soporte vectorial.

En [8] se obtiene una estimación de la radiación solar mensual mediante métodos basados en máquinas de soporte de vectores que fueron entrenados con datos de temperatura del aire.

5.2.3. Modelos ARIMA

Los modelos ARIMA, creados por Box y Jenkins [9], son modelos estadísticos desarrollados para llevar a cabo la estimación de modelos dinámicos de series temporales en los que existe una gran dependencia entre los datos. El nombre de ARIMA proviene de “modelo autorregresivo integrado de medias móviles”.

La expresión del modelo ARIMA (p,d,q) es la siguiente:

$$(1 - \beta_1 B - \beta_2 B^2 - \dots - \beta_p B^p) X_t = (1 - \alpha_1 B - \alpha_2 B^2 - \dots - \alpha_q B^q) e_t \quad (7)$$

Como se puede observar, el modelo ARIMA está formado por tres componentes de órdenes (p,d,q). La componente p hace referencia a la componente autorregresiva de orden polinómico. La componente d indica el número de veces que la serie debe ser restada para que sea estacionaria. Por último, la componente q indica la componente de medias móviles de orden polinómico.

La componente autorregresiva del modelo indica que las observaciones en un determinado momento se pueden predecir mediante las observaciones previas del modelo (añadiendo un término de error). Mientras que la componente de medias móviles del modelo indica que el valor actual puede estimarse a partir de la componente aleatoria de ese momento.

El modelo ARIMA está formado por cuatro etapas:

- Identificación. Usando los datos ordenados cronológicamente se intenta identificar el posible modelo ARIMA (p,d,q). El objetivo es determinar las transformaciones necesarias para convertir la serie en estacionaria y calcular los valores adecuados para la parte autorregresiva y la media móvil.
- Estimación. Considerando el modelo elegido para la serie temporal, se estiman los parámetros y se obtiene el error y los residuos.
- Validación. Esta etapa sirve para validar si el modelo seleccionado es el adecuado, en caso de no ser así, habría que

repetir la primera etapa. Esto se realizará cuando los residuos muestren dependencia.

- Predicción. Una vez seleccionado el modelo ARIMA (p,d,q) se puede llevar a cabo la predicción.

Un ejemplo de estimación basada en modelos ARIMA se puede ver en [10], donde se utiliza para el cálculo de la radiación solar media.

Por lo general, el modelo basado en redes neuronales ofrece mejores resultados de predicción de generación a corto plazo con respecto a los modelos ARIMA, como se demuestra en [11].

5.2.4. Modelos basados en lógica difusa

La lógica difusa fue desarrollada por Loffi A. Zadeh en 1965 como un mecanismo para modelar sistemas no lineales, y aprender de los datos haciendo uso de determinados algoritmos de aprendizaje [12]. La lógica difusa permite tratar información imprecisa, en términos de conjuntos difusos, que se combinan en reglas para definir acciones.

Las tres etapas fundamentales de la lógica difusa son: la elección de las entradas del proceso difuso, la designación de las funciones y conjuntos de pertenencia, y la elección de las reglas difusas y de la variable de salida.

Los conjuntos difusos aceptan que un elemento puede pertenecer parcialmente a un conjunto, midiendo el grado de pertenencia de dicho elemento al conjunto con un valor entre 0 y ,1 el cual se calculará mediante la función de pertenencia. Existen muchos tipos de funciones de pertenencia, siempre y cuando sean continuas en todo su rango y tomen valores entre 0 y 1.

Las operaciones básicas que se llevan a cabo con los conjuntos difusos son unión (*AND*) e intersección (*OR*). Mediante el uso de estas operaciones se puede interconectar y los diferentes conjuntos para, a su vez, obtener nuevos conjuntos de pertenencia.

Las reglas difusas comprenden las relaciones entre las entradas y las salidas a través de las operaciones de conjuntos.

En [13], se presenta un modelo de predicción fotovoltaica a corto plazo basado en lógica difusa utilizando la irradiancia solar medida.

5.2.5. Modelos basados en k-nn

El método k-nn (también llamado k-vecinos más cercanos, *k-nearest neighbors*) se trata de un método de clasificación supervisado muy utilizado puesto que es muy intuitivo. Este método permite estimar una función de densidad $F(x/C_j)$ que predice el valor de x para la clase C_j .

El algoritmo consiste en clasificar el punto de estudio en la clase a la cual pertenezca la mayor cantidad de vecinos cercanos del conjunto. Para ello, habrá que tener especial cuidado a la hora de elegir el valor de la variable k . Si k toma un valor muy pequeño, la predicción será extremadamente sensible a puntos ruidosos. Por otro lado, si k es demasiado grande, la vecindad del punto a predecir elegida incluirá muchos puntos de diferentes clases.

Por lo tanto, k debe ser lo suficientemente grande para disminuir la probabilidad de una clasificación errónea, pero pequeño en comparación con la cantidad de puntos de estudio.

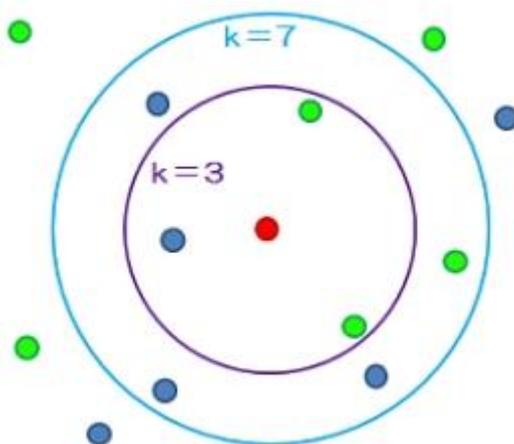


Figura 15. Funcionamiento del algoritmo k-nn (Fuente: iie)

En caso de que se produzca un empate entre clases se debe definir una regla heurística para llevar a cabo el desempate, como por ejemplo seleccionar la clase que tenga menor distancia media, o aquella que contenga al vecino más próximo.

Se dice que es un algoritmo vago ("*lazy algorithm*") puesto que no existe fase de entrenamiento para crear el modelo si no que hasta que no llega un dato a clasificar no se pone en marcha el algoritmo de aprendizaje.

Un ejemplo de modelo de estimación basado en k-nn se puede estudiar en [14], donde se demuestra la correlación existente entre la generación fotovoltaica y la irradiación solar.

6. Redes neuronales artificiales

6.1. Introducción

Las redes neuronales artificiales consisten en un modelo matemático formado por un gran número de elementos simples, que procesan información mediante su estado dinámico como respuesta a entradas externas. Este tipo de modelos matemáticos simulan el comportamiento del cerebro humano, puesto que son capaces de procesar información, de aprender, y de generalizar igual que las neuronas biológicas.

El cerebro está formado por un gran número de elementos interconectados llamados neuronas. Las neuronas son células excitables, que acumulan energía eléctrica en forma de potencial de membrana, para posteriormente liberarla como potencial de acción, y conducir así señales eléctricas. Dichas neuronas están formadas por tres elementos principales: las dendritas, el cuerpo celular y el axón. El punto de contacto entre el axón de una neurona y la dendrita de otra neurona se conoce como sinapsis. La sinapsis puede ser eléctrica o química y su función es integrar señales y elaborar respuestas adecuadas, que se traducen en una conducta externa. La función de la red neuronal depende de la estructura de la misma y del tipo de sinapsis que presente.

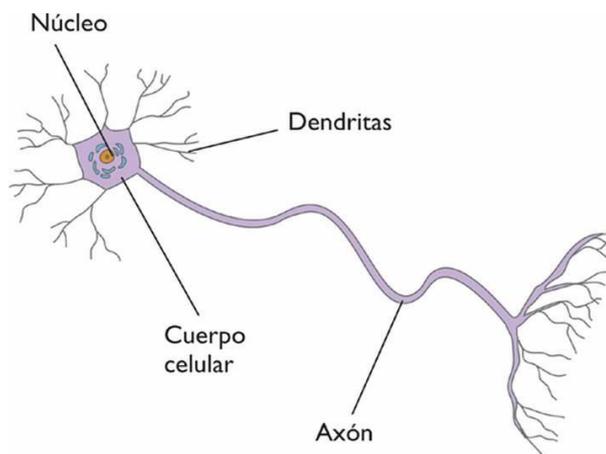


Figura 16. Esquema de los elementos de una neurona biológica (Fuente: Principia Málaga)

Las redes neuronales artificiales no llegan a alcanzar la complejidad del cerebro, sin embargo, existen dos grandes similitudes entre las redes neuronales biológicas y las artificiales. Por un lado, las dos son sistemas computacionales altamente interconectados. Por otro lado, las conexiones entre las diferentes neuronas determinan la función de la red.

Aunque las neuronas biológicas son más lentas comparadas con los circuitos electrónicos, el cerebro puede llevar a cabo tareas mucho más rápido que cualquier ordenador convencional, debido a su estructura en paralelo.

El concepto moderno de red neuronal artificial comenzó en 1940 con el trabajo de McCulloch y Pitts [3], quienes mostraron que las redes neuronales podían llevar a cabo operaciones lógicas y aritméticas. La primera aplicación práctica surgió en 1957 cuando Rosenblatt implementó el perceptrón y demostró su utilidad para reconocer patrones [15]. A partir de este momento y en las décadas posteriores, se produjo un gran interés por este campo, desarrollándose nuevos algoritmos de aprendizaje y de entrenamiento, lo que ha facilitado la implementación de aplicaciones en sectores como el aeroespacial, automoción, defensa, electrónica, robótica, medicina, etc.

6.2. Elementos de una red neuronal artificial

Las redes neuronales artificiales están formadas principalmente por cuatro elementos: entradas, pesos sinápticos, función de activación y salida. A continuación, se explicará en detalle cada uno de los elementos.

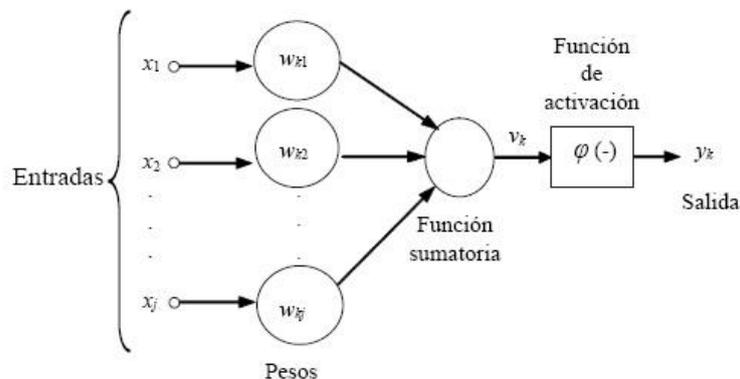


Figura 17. Elementos de una red neuronal artificial (Fuente: [25])

6.2.1. Entradas

Las entradas de la red neuronal son señales que pueden proceder del exterior o bien ser la salida de otra neurona.

6.2.2. Pesos sinápticos

Los pesos sinápticos son valores numéricos que multiplican a las señales de entrada. Estos valores se ajustan en la fase de aprendizaje de la red pudiendo ser positivos, negativos o cero. Si un peso sináptico toma un valor positivo producirá una señal excitadora, puesto que cuando el valor positivo se multiplique por la señal de entrada, y se lleve a cabo el sumatorio, tendrá influencia positiva. Si por el contrario un peso sináptico toma un valor negativo producirá una señal inhibitoria por el mismo motivo. Si un peso sináptico toma valor cero quiere decir que la conexión entre esas dos neuronas es inexistente.

6.2.3. Función de activación

El sumatorio de todas las entradas de la neurona, una vez que han sido multiplicadas por sus pesos sinápticos, se transforma mediante la función de activación para obtener la salida de la neurona. Las funciones de activación más empleadas son:

- Función de activación lineal: En este tipo de función de transferencia, la salida de la función es directamente proporcional a su entrada.

$$a = n \quad (8)$$

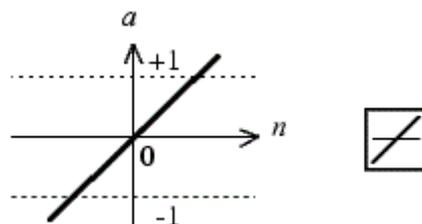


Figura 18. Función de activación lineal (Fuente: [26])

- Función de activación sigmoideal logarítmica: En este tipo de función de transferencia, si los valores de entrada son

ligeramente mayores a cero, su salida se aproxima a la unidad, mientras que si los valores de entrada son ligeramente menores que cero, su salida tenderá a cero.

$$a = f(n) = \frac{1}{1 + e^{-n}} \quad (9)$$

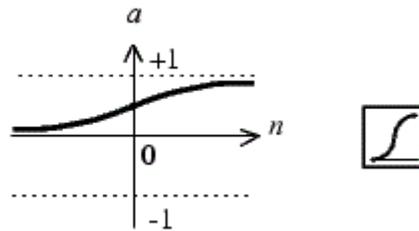


Figura 19. Función de activación sigmoide logarítmica (Fuente: [26])

- Función de activación sigmoide tangente hiperbólica: En este tipo de función de transferencia, la salida estará comprendida entre 1 y -1 dependiendo de si la entrada es mayor que cero o menor que cero.

$$a = f(n) = \frac{1 - e^{-n}}{1 + e^{-n}} \quad (10)$$

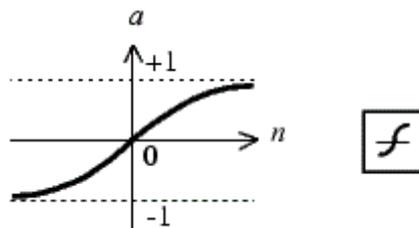


Figura 20. Función de activación sigmoide tangente hiperbólica (Fuente: [26])

6.2.4. Salida

La función de salida se obtiene de la función de activación de la neurona. Esta salida puede utilizarse como salida de la red neuronal, o bien tomarse como entrada de otra neurona.

6.3. Topología de las redes neuronales artificiales

Se conoce como topología al término utilizado para referirse a la forma de organizar las neuronas dentro de la red neuronal. Las neuronas de una red

se agrupan en capas. Los parámetros principales de la red son: el número de capas, el número de neuronas en cada capa y el tipo de conexión que se lleva a cabo entre las diferentes neuronas de la red.

Las capas de una red neuronal se pueden clasificar en función de su situación dentro de la red, pudiendo ser capa de entrada, capas ocultas o capa de salida. Las capas ocultas se sitúan entre la entrada de la red y la capa de salida, mientras que la capa de salida está compuesta por las neuronas que proporcionan la respuesta de la red neuronal.

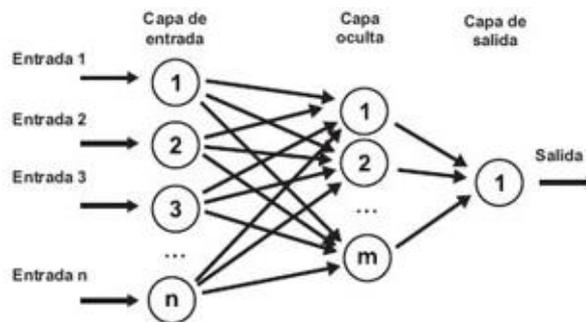


Figura 21. Topología de una red neuronal típica (Fuente: Ibermatica RTD)

En función del número de capas que las forman, las redes neuronales pueden considerarse monocapa o multicapa. Las redes neuronales monocapa están formadas únicamente por el conjunto de entradas y la capa de salida. Por otra parte, las redes neuronales multicapa están formadas por el conjunto de entradas, la capa de salida, y un determinado número de capas ocultas (como mínimo una).

Por último, se pueden clasificar atendiendo al flujo de información en el interior de la red.

6.3.1. Redes de propagación hacia delante

Si las señales y las conexiones entre las distintas neuronas se transmiten desde las entradas hacia la salida atravesando las diferentes capas siguiendo únicamente ese sentido (hacia delante), se tratarán de redes *feed-forward*.

Las redes *feed-forward* son un tipo de red muy utilizado en aplicaciones de clasificación de patrones. Algunos ejemplos de este tipo de red son: perceptrón, perceptrón multicapa, red Adaline, red Kohonen, etc.

6.3.2. Redes de propagación realimentadas

Si las señales y las conexiones entre las distintas neuronas se transmiten tanto hacia delante como hacia atrás a lo largo de la red, se tratará de redes *feed-back*.

Este tipo de redes son muy utilizadas para aplicaciones de control de procesos dinámicos. Algunos ejemplos de red *feed-back* serían: red Hopfield, red Elman y red Jordan.

6.4. Aprendizaje en las redes neuronales artificiales

Para que las redes neuronales puedan desempeñar la función para la que han sido creadas, previamente debe llevarse a cabo un proceso de aprendizaje a través de un conjunto de datos de entrenamiento. La finalidad de este proceso de aprendizaje es ajustar los pesos sinápticos de la red neuronal para conseguir que la respuesta de la red sea la correcta.

Los distintos métodos de aprendizaje se pueden dividir en dos grupos, que se verán en detalle a continuación:

Aprendizaje supervisado: Para llevar a cabo un aprendizaje de tipo supervisado, es necesario disponer de un conjunto de datos de entrenamiento que contenga tanto entradas como salidas. Los datos de entrada son procesados por la red uno a uno, y de la misma forma se va calculando las salidas. Las salidas calculadas se van comparando con las salidas deseadas y según sea el error existente entre las salidas calculadas y las deseadas, se van modificando los pesos sinápticos de las neuronas evaluando el error existente. De esta manera se consigue reproducir los patrones de manera muy precisa.

Aprendizaje no supervisado: Cuando se opta por un aprendizaje no supervisado, solamente es necesario un conjunto de datos que

contenga las entradas (no existe información disponible sobre las salidas). La red se auto-organizará para obtener las salidas en base a la correlación existente entre las entradas presentadas anteriormente, sin ningún mecanismo externo para modificar el valor de los pesos sinápticos.

6.5. Sobreentrenamiento en las redes neuronales

Cuando la red se entrena demasiado, es posible que tenga lugar una pérdida de generalización. Esto es debido a que la red predice de manera muy eficaz la respuesta con los datos que ya se le han presentado en el entrenamiento, pero no es capaz de generalizar cuando se le presentan datos nuevos, por lo que no aprende correctamente el patrón de entrenamiento.

Existen diferentes métodos para evitar que se produzca sobreentrenamiento a la hora de entrenar la red neuronal.

Una de las opciones sería reducir el tamaño de la red neuronal ajustándola en función del objetivo para el que ha sido diseñada. Se debe tener especial cuidado a la hora de utilizar esta técnica ya que si se reduce demasiado el tamaño de la red es posible que no sea capaz de aprender adecuadamente, y la respuesta de la red neuronal no se aproxime al objetivo deseado.

Otra de las técnicas utilizadas para evitar el sobreentrenamiento sería la validación cruzada. Este método consiste en dividir el conjunto de datos de entrenamiento en dos. Una de las partes se utiliza para llevar a cabo el entrenamiento propiamente dicho, y la otra parte que se llama conjunto de validación, se utiliza para calcular el error de validación (es el error que se obtiene al utilizar datos diferentes a los del proceso de entrenamiento). De esta forma se entrenará la red hasta que el error de validación alcance un mínimo, puesto que después volverá a aumentar.

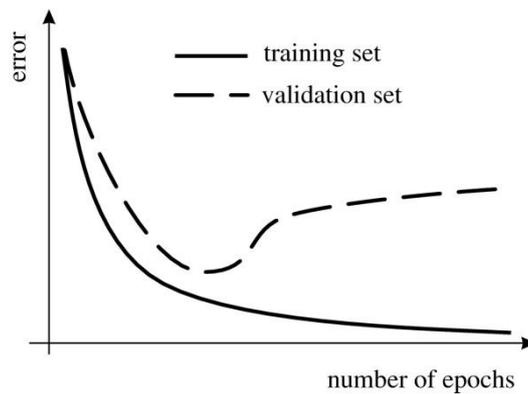


Figura 22. Método de validación cruzada (Fuente: [27]; Error! No se encuentra el origen de la referencia.)

6.6. El perceptrón multicapa

Como se ha visto anteriormente, el primer perceptrón fue diseñado en 1957 por Rosenblatt [15]. Cuando se investigó a fondo este tipo de red neuronal, se descubrieron algunas limitaciones en el modelo.

Uno de los problemas del perceptrón simple es que debido a su estructura sencilla (está formado únicamente por una capa) tiene una capacidad muy limitada. Por este motivo, se dice que el perceptrón simple solamente es útil cuando los datos de estudio son linealmente separables, como se observa en la Figura 23.

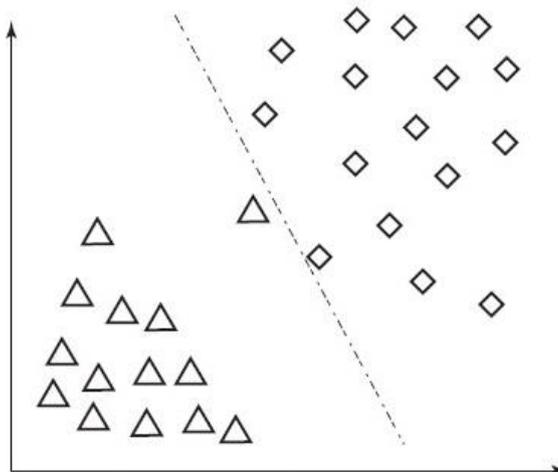


Figura 23. Datos linealmente separables (Fuente: [28])

El perceptrón multicapa surgió para poner fin a las limitaciones del perceptrón simple. Para ello, se optó por modificar la estructura básica del

perceptrón simple añadiendo varias capas. El hecho de añadir más capas a la estructura permite que la red opere en paralelo, reduciendo mucho el tiempo de procesamiento que necesita la red para conseguir la respuesta adecuada.

El perceptrón multicapa se trata de una red *feed-forward*, es decir, con propagación hacia delante. Normalmente, cuando se utiliza este tipo de red neuronal se eligen funciones de tipo sigmoidal como función de activación. En la Figura 24 se muestra la estructura típica de un perceptrón multicapa.

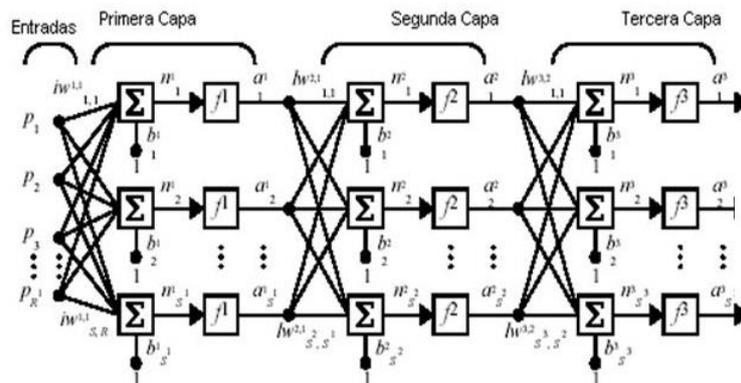


Figura 24. Esquema del perceptrón multicapa (Fuente: [29])

Anteriormente se ha visto en detalle en qué consiste el sobreentrenamiento y cuáles son las principales técnicas para evitarlo. Puesto que se están comentando las características del perceptrón multicapa, es necesario señalar qué método de entrenamiento utiliza este tipo de red.

El perceptrón multicapa utiliza el algoritmo *backpropagation* (propagación hacia atrás) para llevar a cabo el entrenamiento.

Este algoritmo consiste en llevar a cabo el aprendizaje de un conjunto de datos como ejemplo, para obtener la respuesta de la red y comparar el resultado con la salida que se desea obtener. De esta forma se calcula el error en la capa de salida. A continuación, el error se transmitirá hacia atrás pasando por todas las capas intermedias, calculando el error en cada una de ellas en función del error total, hasta llegar a la entrada. En este momento se actualizarán los valores de los pesos sinápticos de cada capa en función del error generado en dicha capa. Así, la próxima vez que se lleve a cabo el proceso, la salida generada estará más próxima a la salida deseada. En [16] se

puede ver una descripción detallada del funcionamiento y de las expresiones matemáticas del algoritmo *backpropagation*.

7. Diseño e implementación de la red neuronal para predicción de la producción de un parque solar

7.1. Introducción

En este capítulo se explica el diseño e implementación de la red neuronal para realizar la predicción de la producción de energía de un parque fotovoltaico. El parque fotovoltaico elegido para llevar a cabo el entrenamiento y las pruebas de la red neuronal está situado en Monteroni di Lecce (Italia), y tiene una potencia nominal de 960 kW.

A la hora de diseñar la red, se ha elegido un perceptrón multicapa. La elección de este modelo de red neuronal respecto a otros modelos se debe a que el perceptrón multicapa presenta cierta tolerancia a fallos, y a que se trata de una estructura altamente no lineal. En [17] se puede ver una comparación entre distintos modelos de redes neuronales. Para configurar el perceptrón multicapa, habrá que tener en cuenta la elección de ciertos parámetros. Como por ejemplo el número de capas, el número de neuronas en la capa oculta, las entradas y salidas a utilizar en la red neuronal, el tipo de entrenamiento, etc.

Una vez implementada la herramienta, se diseñará una interfaz gráfica para poder realizar todo el proceso de manera cómoda y poder observar los resultados de la predicción pudiendo escoger entre diferentes conjuntos de datos de prueba organizados en periodos semanales.

7.2. Localización del parque fotovoltaico

El parque fotovoltaico elegido para llevar a cabo las pruebas de la red se encuentra situado en Monteroni di Lecce, Italia (40°20'05.8"N 18°06'45.9"E).

Monteroni di Lecce es una pequeña localidad de 13000 habitantes situada a 8 kilómetros de Lecce, al sur de Italia. Se encuentra a 53 metros sobre el nivel del mar. Se trata de una región con un clima templado-cálido, donde la temperatura media anual es de 16,4°C (las temperaturas son más cálidas en agosto con una media de 24,7°C, y las más frías en enero con una media de 9,2°C). Las precipitaciones son escasas en los meses de verano, con

una media de 15mm en julio, siendo noviembre el mes más lluvioso históricamente, con unas precipitaciones de 91mm.

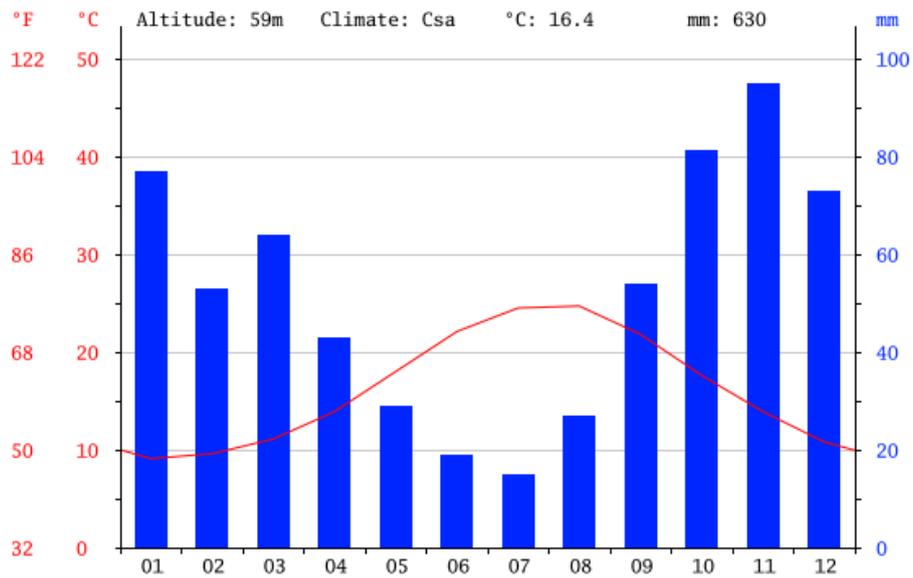


Figura 25. Climograma de Monteroni di Lecce (Fuente: Climate-Data)

El parque fotovoltaico está instalado en la Universidad de Salento, en concreto en el campus universitario Ecotekne. Los módulos fotovoltaicos se encuentran situados en el tejado de varios aparcamientos.



Figura 26. Vista aérea del parque fotovoltaico (Fuente: Google Maps)

A continuación, en la Tabla 1 se muestran las características técnicas de los módulos fotovoltaicos instalados:

Tipo	Silicio monocristalino
Potencia nominal	320 W
Tensión máxima	54,7 V
Corriente máxima	5,86 A
Tensión de circuito abierto	64,8 V
Corriente de cortocircuito	6,24 A
Peso	18,6 kg
Superficie del módulo	1,57 m ²

Tabla 1. Especificaciones técnicas de los módulos fotovoltaicos

El sistema fotovoltaico completo está compuesto por dos bloques. Por un lado, hay un bloque con 1104 módulos fotovoltaicos orientados con un azimutal de -10° y una inclinación de 3° que en conjunto conforman una potencia nominal de 353,3 kW. Por otro lado, el otro bloque está formado por

1896 módulos fotovoltaicos orientados con un azimutal de -10° y una inclinación de 15° cuya potencia nominal resulta 606,7 kW.

Si se suman las potencias nominales de ambos bloques, resulta una potencia nominal total del sistema fotovoltaico de 960 kW. El hecho de orientar los dos bloques con distintas inclinaciones tiene como objetivo maximizar la producción de energía del parque.



Figura 27. Sistema fotovoltaico de la universidad de Salento (Fuente: [30])

Para monitorizar los parámetros del sistema fotovoltaico, se ha integrado un sistema de adquisición de datos formado por un conjunto de sensores que se utilizan para medir la temperatura ambiente, la temperatura de los módulos fotovoltaicos y la irradiación solar existente. Los datos son recogidos por un sistema *SCADA SIMATIC WinCC* cada minuto en el caso de tratarse de la irradiación solar (se obtienen los valores de irradiación para una inclinación de 3° y 15°), o cada 10 minutos en el caso de tratarse de la temperatura ambiente y de la temperatura de los módulos fotovoltaicos.

Se ha escogido este sistema fotovoltaico para llevar a cabo el entrenamiento y las pruebas de la red neuronal diseñada debido a que es muy complicado encontrar bases de datos completas de parques fotovoltaicos que contengan todos los parámetros necesarios para poder realizar una predicción adecuada. En otras instalaciones fotovoltaicas estudiadas o bien faltaban parámetros o el número de registros de la base de datos no era lo suficientemente grande como para realizar un entrenamiento adecuado. Para

poder realizar una predicción acertada, es necesario disponer como mínimo de un año de registros. Además, puesto que en la producción de energía solar fotovoltaica uno de los factores más influyentes es la radiación solar, conviene que la base de datos contenga este registro. Por último, también es importante conocer el valor de la temperatura existente en los paneles fotovoltaicos, ya que dicha temperatura está directamente relacionada con la eficiencia de los paneles. El parque fotovoltaico de Monteroni di Lecce satisface todos los requerimientos deseables.

7.3. Diseño de la red neuronal

Como se ha visto anteriormente, se ha optado por diseñar e implementar un modelo de red neuronal basado en el perceptrón multicapa.

A la hora de desarrollar el modelo del perceptrón multicapa, se deben fijar ciertos parámetros como el tipo de red, la función de entrenamiento, la función de aprendizaje, etc.

El perceptrón multicapa diseñado está formado por tres capas: capa de entrada, capa oculta y capa de salida. Por lo tanto, se puede afirmar que el diseño implementado tiene topología multicapa. Se ha elegido un modelo con una sola capa oculta puesto que salvo casos que presenten gran complejidad, no se justifica el uso de dos capas ocultas. El coste computacional de la topología con dos capas ocultas es mucho mayor que con una sola capa oculta y además no se mejoran los resultados.

En cuanto a las entradas de la red neuronal, como se puede ver en la Figura 28, son seis las entradas de la red neuronal implementada: fecha en formato seno, fecha en formato coseno, temperatura ambiente, temperatura del módulo fotovoltaico, irradiancia a 3° de inclinación e irradiancia a 15° de inclinación.

No existe una regla exacta para calcular el número de neuronas necesario en la capa oculta, pero una aproximación acertada sería decir que el número de neuronas debe ser menor o igual al doble del número de entradas. Es decir, puesto que existen 6 entradas en la red, el número máximo de

neuronas en la capa oculta deberían ser 12. De superar dicho número de neuronas, la salida se volvería inestable. Más adelante, se realizarán numerosas pruebas para averiguar el número de neuronas en la capa oculta con el que se obtienen mejores resultados.

La capa de salida está formada por una única salida, que será la energía producida por el sistema fotovoltaico.

A la hora de elegir un algoritmo de entrenamiento, hay que tener en cuenta que el perceptrón multicapa se trata de una red *feed-forward backpropagation*, por lo tanto, se debe escoger un algoritmo de entrenamiento de tipo *backpropagation*. De entre todos los algoritmos disponibles, el algoritmo de Levenberg-Marquardt es el más rápido, además de ser altamente recomendable para llevar a cabo un aprendizaje de tipo supervisado. En [18] se muestra un estudio acerca del algoritmo de Levenberg-Marquardt así como una comparación entre distintos tipos de algoritmos.

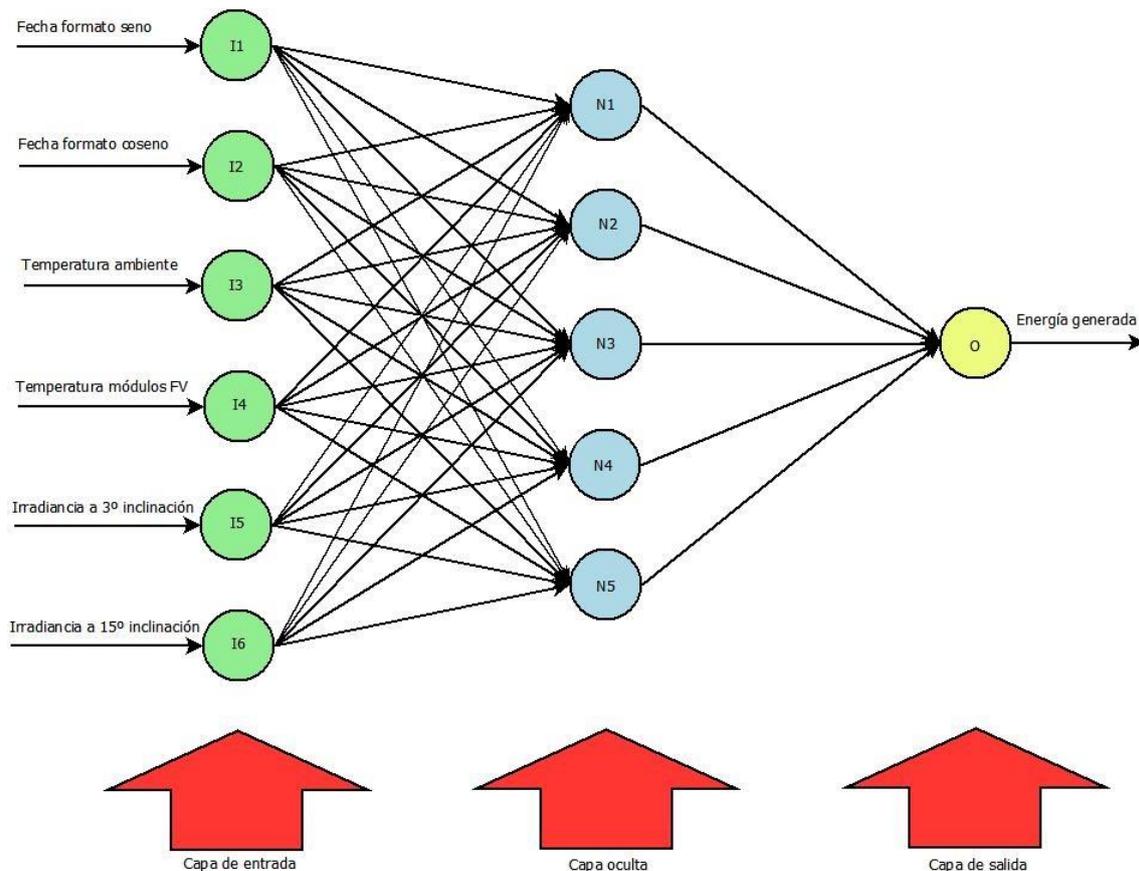


Figura 28. Diagrama de la red neuronal diseñada

7.3.1. Entradas de entrenamiento

La base de datos de entrenamiento del sistema fotovoltaico se trata de un fichero compuesto por los datos recogidos por el sistema de adquisición mencionado anteriormente. Se debe recordar que estos datos han sido capturados con una frecuencia de un minuto o de 10 minutos en función del parámetro del que se tratase, pero a la hora de crear la base de datos, estos datos adquiridos se han procesado para ofrecer las muestras tomadas con frecuencia horaria. La base de datos comienza el día 1 de marzo del 2012, y finaliza el día 28 de febrero del 2013, por lo que se trata de un fichero de 8760 registros (la base de datos tiene 8760 muestras).

Se ha escogido un año entero como base de datos para llevar a cabo el entrenamiento, ya que al tener una muestra de cada día del año la red podrá generalizar, ofreciendo unos resultados de entrenamiento aceptables sin llegar al sobreentrenamiento.

7.3.1.1. Definición de las entradas

Se ha escogido una topología formada por seis entradas. A la hora de escoger las entradas, se deben tener en cuenta los factores que más influyen en la producción de energía solar fotovoltaica. Es importante conocer la fecha puesto que se trata de la referencia que posee la red neuronal para predecir la energía generada en un momento determinado. Por otro lado, la temperatura es un factor que afecta directamente a la potencia que entregan los módulos fotovoltaicos, por lo que será adecuado que forme parte de las entradas de la red. Por último, la irradiancia es un factor clave para la generación fotovoltaica, por lo que es altamente recomendable que forme parte de las entradas de la red. Por tanto, las entradas de la red neuronal son:

1. *Fecha en formato seno*: Muestra el día al que corresponden el resto de entradas, en formato seno. La muestra 1 corresponde al día 1 de marzo del 2012 y la muestra 365 corresponde al día 28 de febrero del 2013.
2. *Fecha en formato coseno*: Muestra el día al que corresponden el resto de entradas, en formato coseno. La muestra 1 corresponde

al día 1 de marzo del 2012 y la muestra 365 corresponde al día 28 de febrero del 2013.

3. Temperatura ambiente: Muestra la temperatura ambiente media diaria, medida en grados centígrados.
4. Temperatura de los módulos: Muestra la temperatura media diaria de los módulos fotovoltaicos, medida en grados centígrados.
5. Irradiancia a una inclinación de 3°: Muestra la irradiancia media diaria de los paneles cuya inclinación es de 3°, medida en W/m².
6. Irradiancia a una inclinación de 15°: Muestra la irradiancia media diaria de los paneles cuya inclinación es de 15°, medida en W/m².

Estas entradas son preprocesadas (como se verá a continuación) y suministradas en un fichero formado por un año entero de registros. De esta forma se garantiza que el usuario pueda acceder a dichos registros fácilmente.

7.3.1.2. Preprocesamiento de los datos

Para hacer este fichero más manejable a la hora de trabajar con la red neuronal, se ha realizado sobre la base de datos un promedio, resultando un fichero formado por el promedio diario de las muestras horarias obtenidas por el sistema de adquisición, con lo que se consigue un fichero de 365 filas (el promedio de cada uno de los días que forman la base de datos).

Para una correcta implementación de la red neuronal, es necesario estandarizar sus entradas. Se debe partir de la premisa de que pequeños cambios en la entrada dan lugar a pequeños cambios en la salida. Si se cumple esto, se consigue mejorar considerablemente el entrenamiento de la red neuronal, con lo que se obtienen mejores resultados en las pruebas. Todos los datos deberían estar dentro del intervalo $[-1,1]$ o bien $[0,-1]$, por lo que se han utilizado técnicas para conseguir estos valores.

Una de las técnicas utilizadas es la que se conoce como técnica de los "máximos". Con esta técnica, se debe buscar el máximo del vector de datos a evaluar. Una vez hallado el máximo, se dividirá cada uno de los datos del vector entre su máximo, obteniéndose un nuevo vector estandarizado en el que

todas las muestras estarán dentro del rango [0,1] (puesto que todos los datos del vector original son positivos). Esta técnica se ha utilizado para procesar las entradas: temperatura ambiente, temperatura de los módulos, irradiancia a una inclinación de 3° e irradiancia a una inclinación de 15°.

Otra de las técnicas usadas es la estandarización mediante la función seno y coseno. Esta técnica se utiliza en dos de las entradas, las que corresponden a la fecha en formato seno y en formato coseno. Para llevarla a cabo, hay que tener en cuenta el número de muestras que componen un año (365 muestras) y utilizar las siguientes expresiones:

$$\text{sen} \left(\left(2 * \frac{\pi}{365} \right) * n \right) \quad (11)$$

$$\text{cos} \left(\left(2 * \frac{\pi}{365} \right) * n \right) \quad (12)$$

Siendo 'n' el número de muestra. Con esto se consigue que la entrada oscile en un rango entre -1 y 1.

El hecho de utilizar dos entradas fecha diferentes (la misma muestra multiplicada por seno y coseno), se debe a que de esta forma se consigue que a cada muestra (de 1 a 365) le corresponda un par de datos seno-coseno únicos para cada muestra (no se repetirán). Mediante esta técnica se consigue que la entrada oscile entre -1 y 1, además de que no se confundan ninguna de las 365 muestras.

A continuación, en la Tabla 2 y Tabla 3 se observa un ejemplo de la base de datos de entradas de entrenamiento original y su aspecto una vez llevado a cabo el estandarizado:

Fecha	Tª ambiente	Tª de los módulos	Irradiancia a 3°	Irradiancia a 15 °
07/03/2012	12,49	14,20	169,15	184,42
08/03/2012	13,94	16,43	181,74	196,26
09/03/2012	14,17	15,94	135,28	142,54
10/03/2012	12,70	13,08	122,25	124,62
11/03/2012	12,44	12,91	174,33	187,78
12/03/2012	10,85	12,21	252,69	254,57
13/03/2012	10,90	12,26	323,15	362,56

Tabla 2. Entradas de la base de datos de entrenamiento pre-estandarizado

Fecha	Fecha	Tª ambiente	Tª de los módulos	Irradiancia a 3°	Irradiancia a 15 °
0,120208	0,992749	0,356267	0,321123	0,354614	0,390468
0,137279	0,990532	0,397627	0,371553	0,381008	0,415537
0,154309	0,988023	0,404188	0,360472	0,283607	0,301797
0,171293	0,98522	0,362257	0,295795	0,25629	0,263855
0,188227	0,982126	0,354841	0,291951	0,365473	0,397582
0,205104	0,97874	0,309488	0,276121	0,529751	0,538995
0,221922	0,975065	0,310914	0,277252	0,677466	0,76764

Tabla 3. Entradas de la base de datos de entrenamiento post-estandarizado

7.3.2. Salida de entrenamiento

La salida de la base de datos de entrenamiento de la red neuronal se trata de un fichero formado por los datos de salida de la instalación fotovoltaica, que han sido recogidos por el sistema de adquisición de datos.

Este fichero recoge la energía horaria total (se debe recordar que existen dos bloques de generación, uno con una orientación de los módulos a 3° y el otro bloque con una orientación a 15°) generada por el sistema fotovoltaico, medida en kWh.

Al igual que ocurría con la base de datos de entradas, la base de datos de salida ha sido procesada para facilitar su manejo, de tal modo que se ha hecho el acumulado diario de la energía horaria que forma la base de datos, resultando una nueva base de datos compuesta por la energía diaria generada por el sistema fotovoltaico, medida en kWh/día.

Para mejorar la predicción, se debe llevar a cabo un estandarizado de los datos de salida, como ya se hizo con los datos de entrada. En este caso, para obtener unos resultados aceptables en el entrenamiento y en las pruebas basta con dividir los datos de salida de la base de datos de entrenamiento entre cien, no siendo necesario que estos datos queden comprendidos entre -1 y 1 como ocurría con las entradas.

A continuación, en la Tabla 4 y Tabla 5 se muestra un ejemplo de la base de datos de salida de entrenamiento original y su aspecto una vez llevado a cabo el estandarizado:

Fecha	Energía diaria
07/03/2012	2761,1
08/03/2012	2842,98
09/03/2012	2227,88
10/03/2012	1980,84
11/03/2012	2778,4
12/03/2012	3959,7
13/03/2012	5117,5

Tabla 4. Salida de la base de datos de entrenamiento pre-estandarizado

Fecha	Energía diaria
07/03/2012	27,611
08/03/2012	28,4298
09/03/2012	22,2788
10/03/2012	19,8084
11/03/2012	2778,4
12/03/2012	39,597
13/03/2012	51,175

Tabla 5. Salida de la base de datos de entrenamiento post-estandarizado

7.4. Herramienta de desarrollo

Se ha diseñado un modelo basado en el perceptrón multicapa para diseñar la herramienta de predicción. Para desarrollar dicho modelo, lo más sencillo es utilizar un software matemático que sea capaz de llevar a cabo la creación, entrenamiento y prueba de la red neuronal.

El software matemático elegido para la realización de este proyecto es Matlab R2012a. Matlab se trata de una herramienta de cálculo científico muy potente, capaz de llevar a cabo numerosas tareas en diferentes áreas (procesamiento de señales, sistemas de control, redes neuronales, etc.) debido a la existencia de varias librerías especializadas (*toolboxes*).

La librería utilizada para el desarrollo de las redes neuronales se conoce como “*Neural Network Toolbox*” (*nntool*), y permite la creación, entrenamiento y simulación de numerosos modelos de redes neuronales. En [19] se puede encontrar ayuda acerca del funcionamiento de esta *toolbox*.

7.5. Creación de la red neuronal

El perceptrón multicapa puede crearse de dos formas, ya que Matlab ofrece la posibilidad al usuario de implementar la red mediante la programación de scripts, o mediante la herramienta de la librería de redes neuronales *nntool*. Se va a utilizar esta última forma para poder ver mediante la interfaz de Matlab el desarrollo del proceso.

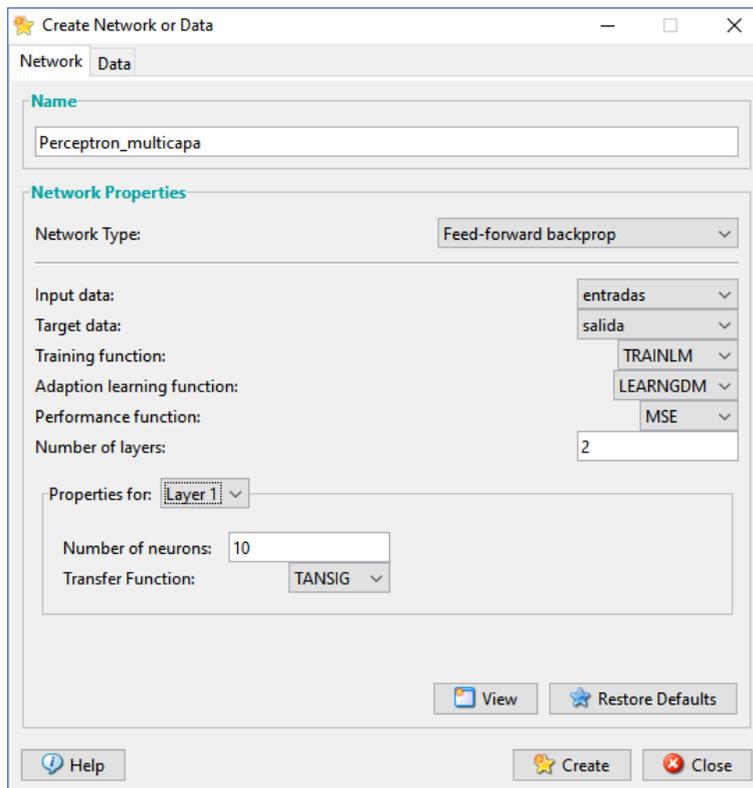


Figura 29. Creación del perceptrón multicapa

Los parámetros utilizados a la hora de crear el perceptrón multicapa han sido los siguientes:

- ***Network type:*** *Feed-forward backpropagation*. Como se ha visto anteriormente, para configurar un perceptrón multicapa es necesario que la propagación de la red sea hacia delante.
- ***Input data:*** En este campo se seleccionará la base de datos de entrada de entrenamiento.
- ***Target data:*** En este campo se seleccionará la base de datos de salida de entrenamiento.

- Training function: TRAINLM. Se trata del algoritmo de entrenamiento de Levenberg-Marquardt [20][21]. Este algoritmo requiere un gran uso de memoria pero se trata del algoritmo más rápido para entrenar redes *feed-forward*. Además, posee excelentes propiedades de convergencia y robustez por lo que se ha convertido en uno de los algoritmos más utilizados.
- Adaption learning function: LEARNNGDM. Se trata de la función de aprendizaje de los pesos, la cual utiliza el gradiente descendente con *momentum* (lo que acelera la convergencia de los pesos). En [22] se puede ver una explicación acerca del mismo.
- Performance function: MSE. Se trata de la función error cuadrático medio entre la salida de la red y la salida deseada. La expresión del MSE es la siguiente:

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] \quad (13)$$

- Number of layers: Es el número de capas del que consta la red, siendo el número mínimo 1, que correspondería a la capa de salida. Normalmente, con utilizar dos capas es suficiente (una capa oculta y una de salida) para llevar a cabo la gran mayoría de las tareas. En este caso se han elegido 2 capas, la capa 1 corresponde a la capa oculta y la capa 2 se trata de la capa de salida.

Los parámetros para finalizar la configuración de la red son:

- Properties for layer 1: Se llevará a cabo una comparación de los resultados de las pruebas variando el número de neuronas en la capa oculta. Se probará desde 1 hasta 12 neuronas. La función de transferencia utilizada será la sigmoideal tangente hiperbólica (*tansig*).
- Properties for layer 2: En la capa de salida no se puede elegir el número de neuronas, por lo que solamente se tendrá que escoger el tipo de función de transferencia. Debido a las características de

la respuesta de la red, se ha optado por una función de activación lineal (*purelin*).

7.6. Entrenamiento de la red neuronal

Una vez diseñada la red neuronal, se llevará a cabo su entrenamiento. Para ello, se deberán volver a confirmar los datos de las entradas y las salidas (objetivos) de la red. Además, se pueden fijar ciertos parámetros referentes al algoritmo de entrenamiento de Levenberg-Marquardt, los más importantes son los siguientes:

- El número máximo de iteraciones (*epochs*) que tendrán lugar, por defecto son 1000.
- El tiempo (*time*) que puede durar el entrenamiento de la red, por defecto es infinito.
- El máximo error permitido (*goal*) para finalizar el entrenamiento, por defecto fijado en 0 aunque es complicado alcanzar este valor.
- El gradiente mínimo (*min_grad*), que representa el valor mínimo que debe alcanzar el gradiente para detener el entrenamiento.
- El número de iteraciones máximo (*max_fail*) en los que no disminuye el error de validación. Por defecto es 6, y si se alcanza este valor será un indicador de que la red neuronal está perdiendo su capacidad de generalización y se detendrá el entrenamiento.

Tal y como se explicó anteriormente, la base de datos de entrenamiento está formada por 365 muestras, de las cuales el 75% (255 muestras) se utiliza para el entrenamiento propiamente dicho, un 15% (55 muestras) se utilizará para validación, y el 15% restante (55 muestras) para llevar a cabo el testeo de la red. La división de datos en cada uno de los tres conjuntos es completamente aleatoria.

Cuando se da por finalizada la edición de estos parámetros, puede comenzar el entrenamiento de la red, momento en el que aparecerá una interfaz que muestra información acerca del estado del entrenamiento.

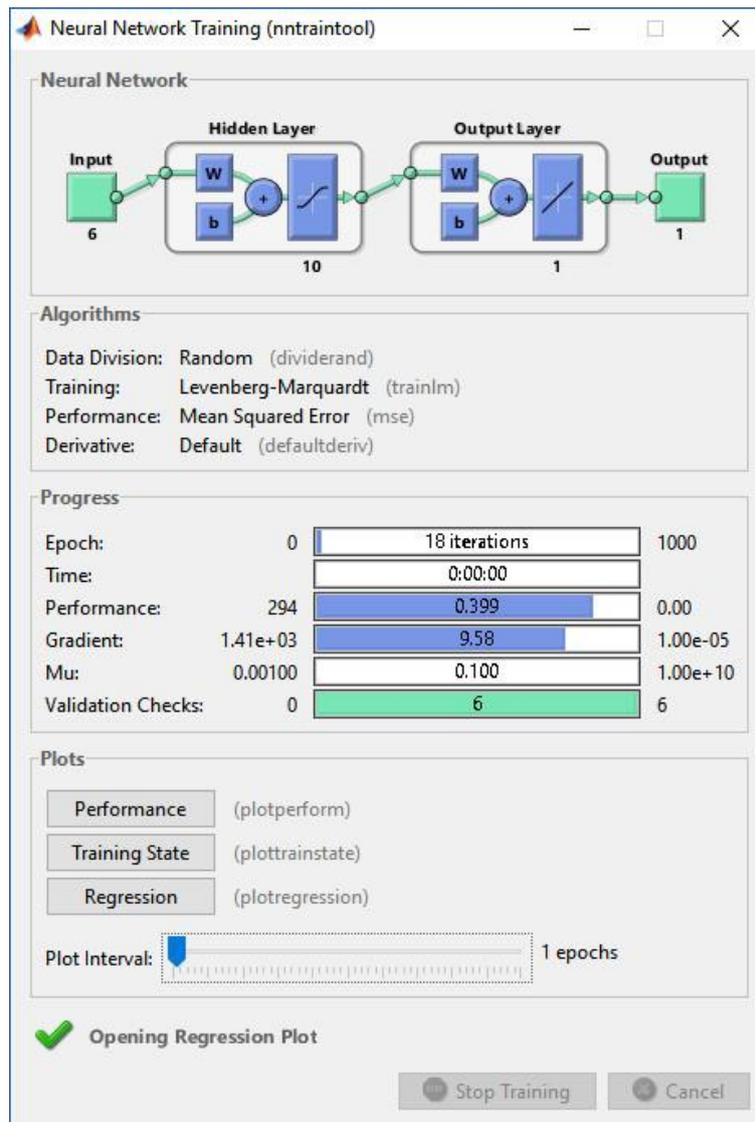


Figura 30. Interfaz del entrenamiento

En esta interfaz se puede observar cómo varían los parámetros en función del estado del entrenamiento. Como se puede ver en la Figura 30, el entrenamiento ha concluido debido a que ha habido 6 iteraciones en las que el error de validación no ha disminuido, y se ha dado por finalizado.

Una vez finalizado el entrenamiento, hay disponibles tres gráficas. En la primera de ellas, disponible en la Figura 31, se muestra el error cuadrático medio en función del número de iteraciones llevadas a cabo durante el entrenamiento. La red finalizó el entrenamiento después de 12 iteraciones, con un error $MSE=0,60418$. Puesto que después de otras 6 iteraciones el error no disminuyó, se detuvo el entrenamiento.

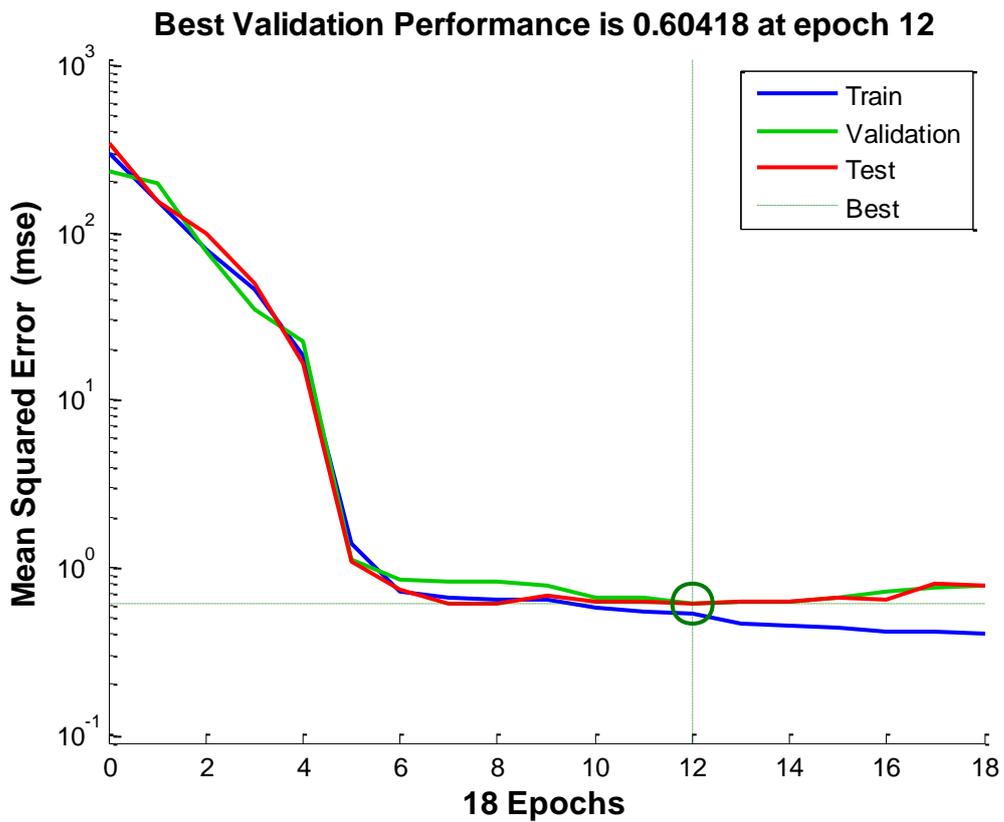


Figura 31. MSE vs. número de iteraciones

En la siguiente gráfica, Figura 32, se observan los valores que van tomando los diferentes parámetros asociados al algoritmo de entrenamiento en función del número de iteraciones durante el entrenamiento.

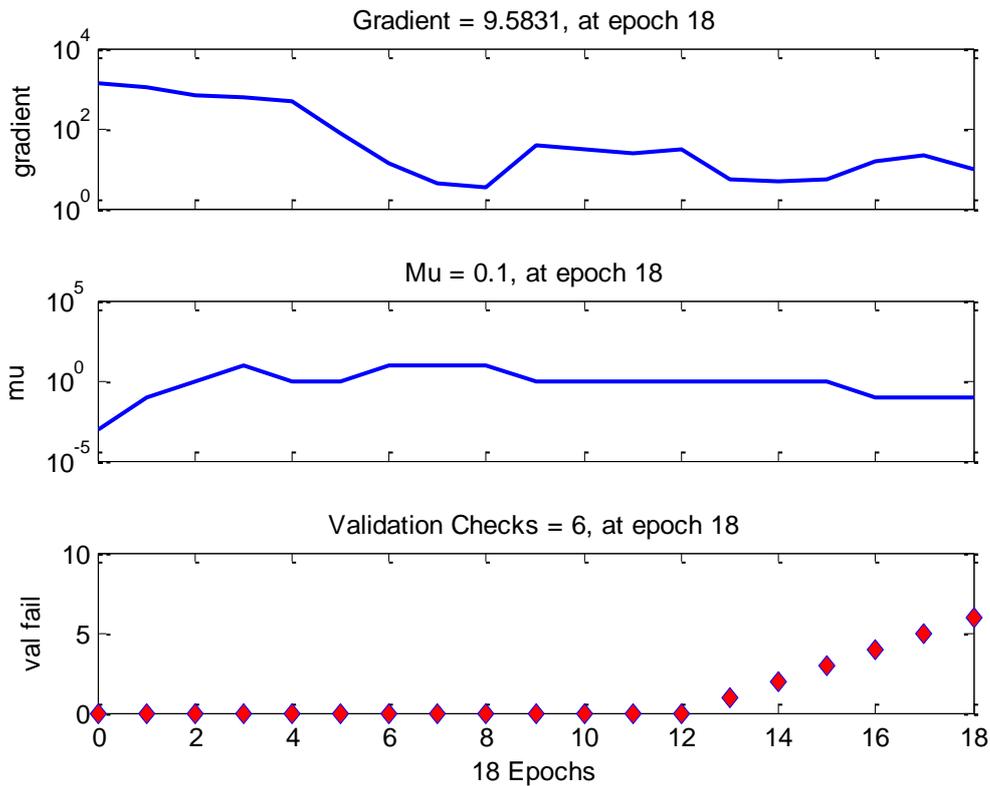


Figura 32. Estado del entrenamiento

Existe una última gráfica disponible en la interfaz de entrenamiento. En la Figura 33 se puede observar la recta de regresión lineal de los objetivos respecto a las salidas durante el proceso de entrenamiento, de validación y de testeó. La regresión lineal es un modelo matemático utilizado para aproximar la relación de dependencia entre dos valores (en este caso, el objetivo y la salida).

Cuanto más próximo a 1 sea el coeficiente de correlación R, mayor será la dependencia entre salida y objetivo.

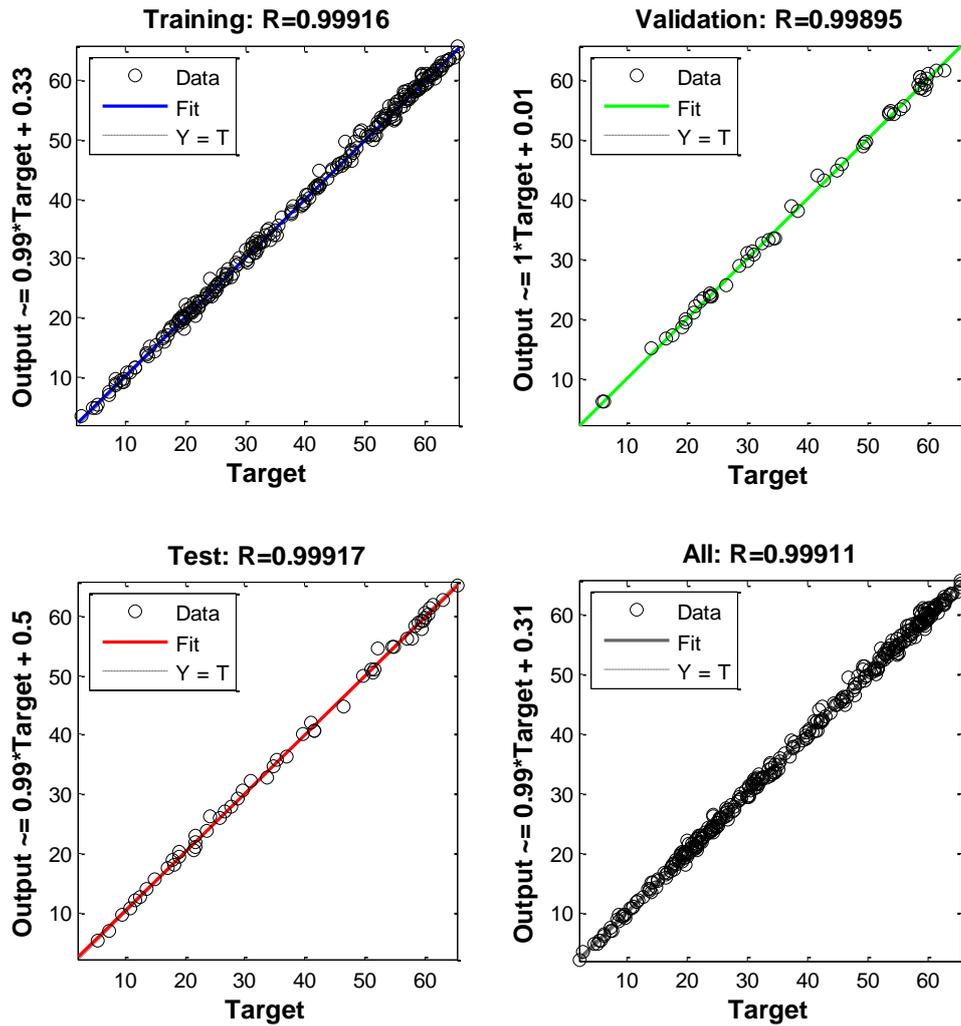


Figura 33. Gráficas de regresión lineal

En la Figura 34, se muestra el histograma de los errores que tienen lugar durante la fase de entrenamiento de la red.

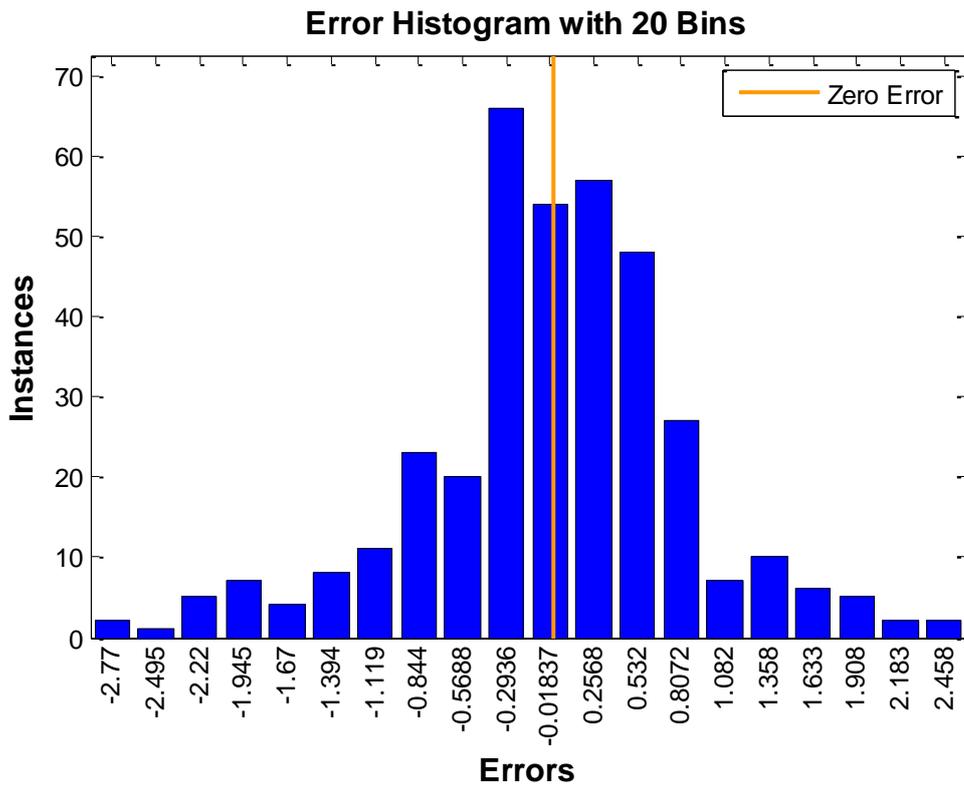


Figura 34. Histograma de los errores

7.6.1. Pesos sinápticos

A continuación, se van a mostrar los pesos sinápticos ajustados una vez finalizado el entrenamiento de la red neuronal para el modelo con el que se han obtenido mejores resultados, el que contiene 5 neuronas en la capa oculta.

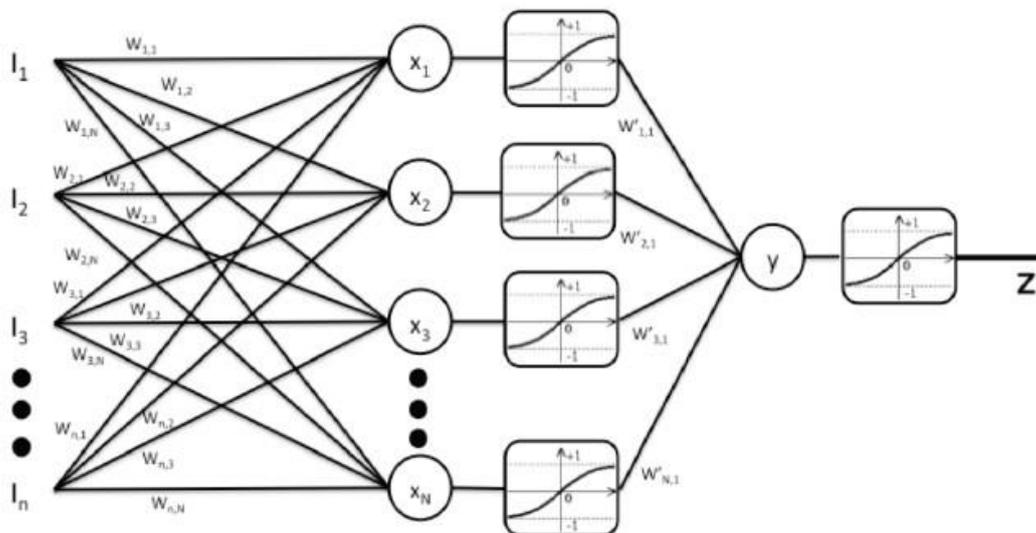


Figura 35. Esquema de los pesos sinápticos de la red neuronal (Fuente: [31])

Los valores de los pesos sinápticos se establecen tanto en las conexiones entre las entradas y las neuronas de la capa oculta, como en las conexiones entre las neuronas de la capa oculta y la salida de la red.

Cada una de las entradas establece una conexión con cada una de las neuronas de la capa oculta, y puesto que hay seis entradas y cinco neuronas en la capa oculta, existirán treinta pesos sinápticos asociados a dichas conexiones. En la Tabla 6 se muestra el valor de los pesos sinápticos ajustados.

	1ª entrada	2ª entrada	3ª Entrada	4ª entrada	5ª entrada	6ª entrada
1ª neurona	-0,7583418	3,6175161	0,3203282	-0,9682135	-3,2814494	3,7933974
2ª neurona	0,0010871	-0,021069	0,0130879	0,0044161	-0,1229723	-0,0641576
3ª neurona	-0,4794759	1,2415876	0,3568773	1,607765	3,3210578	-0,8335878
4ª neurona	3,5440064	-0,2107106	1,631437	2,124276	1,7962381	-0,2832010
5ª neurona	2,716445	4,2110776	-2,3171937	2,1882341	1,5842950	-0,6179481

Tabla 6. Pesos sinápticos ajustados entre las entradas y la capa oculta

Se debe recordar que cuando los pesos sinápticos sean positivos, dicho valor producirá una señal excitadora. Cuando los pesos sean negativos, el valor producirá una señal inhibidora. Por último, cuando el peso sináptico tome valor cero, la conexión considerada será inexistente. En este caso, se observa que todas las conexiones llevadas a cabo entre las entradas y las neuronas de la capa oculta tienen su influencia en la red, puesto que ningún peso sináptico toma valor cero. Pero sí que se observa que las conexiones que se realizan entre las entradas y la segunda neurona de la capa oculta toman valores cercanos a cero. Esto es debido a que la influencia de dicha neurona en la red es menor que la del resto de neuronas.

Por último, en la Tabla 17 se muestra el valor de los pesos sinápticos ajustados en las conexiones entre las neuronas de la capa oculta y la salida.

	1ª neurona	2ª neurona	3ª neurona	4ª neurona	5ª neurona
Salida	-0,05390955	-5,34287727	0,06620768	0,03061591	-0,05327788

Tabla 7. Pesos sinápticos ajustados entre la capa oculta y la salida

7.7. Pruebas sobre la red neuronal entrenada

En este apartado, se lleva a cabo todo lo relacionado con el procesamiento de los datos de prueba para llevar a cabo las pruebas sobre la red neuronal entrenada.

7.7.1. Entradas de los datos de prueba

Una vez entrenada la red neuronal, se analiza su funcionamiento ante datos no entrenados. Para ver los resultados que puede ofrecer a la hora de realizar una predicción, se debe utilizar un conjunto de datos diferente al usado durante el entrenamiento para ver la capacidad de generalización que ha adquirido la red neuronal.

En este proyecto se han elegido conjuntos de prueba con periodicidad semanal. Se debe recordar que la base de datos de entrenamiento albergaba datos desde el 1 de marzo del 2012 hasta el 28 de febrero del 2013. Para realizar la prueba se han escogido 8 conjuntos de datos, cada uno de 7 días, por lo que se puede llevar a cabo la prueba de dos meses. Los conjuntos de datos de prueba están repartidos de la siguiente manera:

- Semana 1: Del 1 de marzo del 2013 al 7 de marzo del 2013.
- Semana 2: Del 8 de marzo del 2013 al 14 de marzo del 2013.
- Semana 3: Del 15 de marzo del 2013 al 21 de marzo del 2013.
- Semana 4: Del 22 de marzo del 2013 al 28 de marzo del 2013.
- Semana 5: Del 29 de marzo del 2013 al 4 de abril del 2013.
- Semana 6: Del 5 de abril del 2013 al 11 de abril del 2013.
- Semana 7: Del 12 de abril del 2013 al 18 de abril del 2013.
- Semana 8: Del 19 de abril del 2013 al 25 de abril del 2013.

Puesto que para probar la red neuronal, será necesario el mismo número de entradas en la red que cuando se quería llevar a cabo el entrenamiento, se debe pensar un método para ofrecer dichas entradas a la red sin ofrecer ningún valor futuro de temperatura o de irradiancia que falsee la predicción.

Para ello, ha sido necesario encontrar más muestras que añadir a la base de datos para hacerla más amplia. En [23], se puede encontrar una herramienta llamada HELIOCLIM 3, proporcionada por el departamento de sistema de adquisición de datos de MINES Paris Tech - Transvalor en la cual podemos introducir ciertos parámetros como la localización, fecha, inclinación del plano, albedo, etc. Esto generará un fichero descargable formado por una base de datos que contiene las muestras de irradiaciones a la inclinación del plano deseada, y durante el período seleccionado.

Uno de los problemas de esta herramienta es que, al tratarse de una versión gratuita, tiene sus limitaciones. Para los usuarios de pago, la herramienta puede proporcionar muestras desde el 1 de febrero del 2004 hasta la actualidad. Sin embargo, en la versión gratuita el período de muestras cubre desde el 1 de febrero del 2004 al 31 de diciembre del 2005. A pesar de que es suficiente, sería recomendable obtener más bases de datos procedentes de otras fuentes, pero resulta muy complicado localizarlas, por lo que el procedimiento se ha llevado a cabo con lo existente.

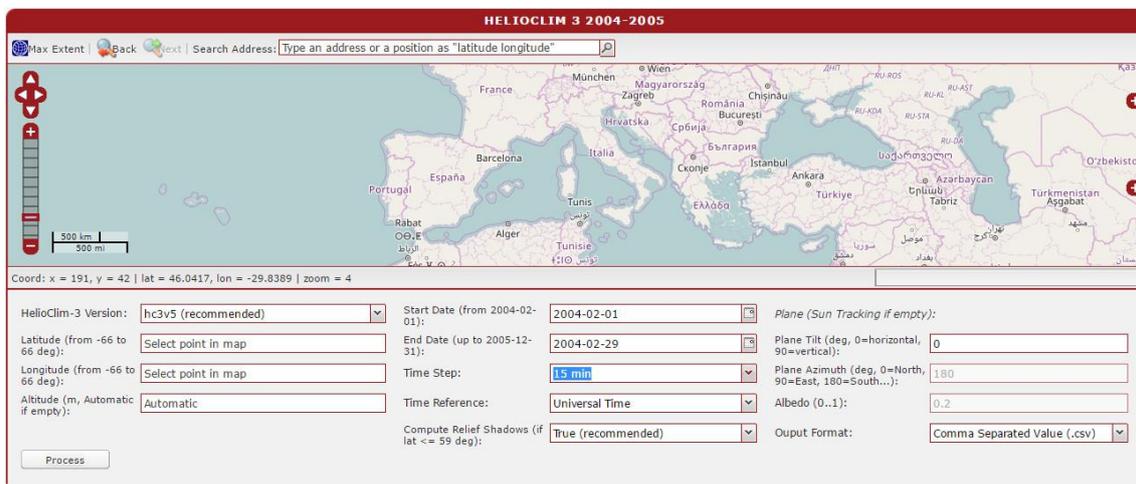


Figura 36. Herramienta HELIOCLIM 3 (Fuente: SoDa)

El procedimiento para obtener las entradas de prueba consiste en, a partir del histórico de muestras formado por la base de datos original del sistema fotovoltaico y la obtenida a partir de la aplicación HELIOCLIM 3, hallar el promedio diario histórico de cada uno de los datos que forman el conjunto de datos de prueba (es decir, cada una de las entradas de la red). Con esto se

obtendrá el valor medio diario histórico de la temperatura ambiente, de la temperatura de los módulos fotovoltaicos, de la irradiancia con un plano de inclinación de 3°, y de la irradiancia con un plano de inclinación de 15° tal y como puede verse en el siguiente ejemplo:

$$Irr. 3^\circ \text{ histórica}|_{01/03} = \frac{Irr. 3^\circ|_{01/03/04} + Irr. 3^\circ|_{01/03/05} + Irr. 3^\circ|_{01/03/12}}{3} \quad (14)$$

Estos valores promedio obtenidos serán los que se introduzcan en las entradas de la red neuronal en el caso de la temperatura ambiente y la temperatura del módulo fotovoltaico. Para introducir las muestras de irradiancia, habrá que calcular la varianza y la desviación típica.

La varianza (σ^2) es el promedio aritmético de las desviaciones respecto a la media.

$$\sigma^2 = \frac{1}{n} * \sum_{i < j} (X_i - X_j)^2 \quad (15)$$

Por otro lado, la desviación típica se define como la raíz cuadrada de la varianza (σ).

$$\sigma = \sqrt{\sigma^2} \quad (16)$$

Planteado de otra forma, la desviación típica expresa lo que tienden a alejarse los valores concretos del promedio en un conjunto de datos. Se debe calcular la desviación típica de cada uno de los conjuntos diarios históricos correspondientes a la irradiancia a 3° y a la irradiancia a 15°.

Puesto que anteriormente se ha calculado el promedio diario histórico, si a dicho promedio se le suma o se le resta la desviación típica de ese conjunto histórico, existirán dos límites (superior e inferior) al que tenderán a alejarse los valores de las irradiancias.

Para elegir entre uno de los dos límites se deben observar dichos límites y compararlos con las muestras históricas. Todo parece indicar que lo correcto es elegir el límite que se encuentre más próximo a la tendencia histórica.

Eventualmente, puede suceder que la elección no haya sido la adecuada (históricamente un día puede tener muy buenas condiciones de irradiancia y un año en concreto puede no ser así), pero estos casos son impredecibles, de ahí la importancia de que la base de datos sea lo más amplia posible puesto que cuantas más muestras formen la base de datos, mejor se podrá generalizar.

Se deberá llevar a cabo este proceso con cada uno de los días que formen el conjunto de datos de prueba, con lo que se creará una base de datos de prueba con los valores de temperatura ambiente, temperatura del módulo, irradiancia a 3° e irradiancia a 15° que han sido calculados.

Para facilitar la prueba, esta base de datos se dividirá semanalmente tal y como se ha explicado previamente, formando así los diferentes conjuntos de entradas que se presentarán a la red.

7.7.2. Salida de los datos de prueba

Puesto que se han elegido entradas de prueba de periodicidad semanal, los conjuntos de salidas de prueba también deben tener la misma longitud. Al igual que ocurría con las entradas, las salidas también van del 1 de marzo del 2013 al 25 de abril del 2013.

Se debe comentar la diferencia entre salida y objetivo en el proceso de prueba. La salida se trata de la respuesta de la red a una determinada entrada, es decir, es la predicción propiamente dicha. La salida como tal no requiere ninguna base de datos porque precisamente es lo que se genera cuando se lleva a cabo la prueba. Si para facilitar el entrenamiento, como se vio anteriormente la salida de entrenamiento se dividía entre cien, en este caso una vez que se obtenga la respuesta de la red se debe multiplicar la salida por cien para obtener los valores de producción de energía estimados. Cada muestra de salida representará la producción diaria de energía eléctrica en el parque fotovoltaico, medida en kWh, para un determinado día (el día que se haya elegido en el conjunto de datos de entrada). Puesto que las entradas están organizadas semanalmente, la respuesta de la red estará formada por un conjunto de datos de siete muestras (una correspondiente a cada día).

El objetivo, es la energía diaria real producida en el parque fotovoltaico en un día determinado. De forma similar a lo que ocurre con las entradas de prueba, el objetivo alberga muestras desde el 1 de marzo del 2013 al 25 de abril del 2013, por lo que será necesario organizar dichos valores semanalmente, en ocho conjuntos de datos tal y como se hace con las entradas de prueba.

La función del objetivo es llevar a cabo la comparación entre dicho valor y la salida estimada para poder calcular el error de predicción existente.

7.8. Pruebas de la red neuronal

A continuación se llevarán a cabo diferentes pruebas y se estudiarán los resultados obtenidos. El criterio escogido para cuantificar el error de predicción se trata del error porcentual absoluto medio (MAPE), el cual mide la desviación en términos porcentuales. Su expresión es la siguiente:

$$MAPE = \left(\frac{1}{n} * \sum \frac{|E_{real} - E_{estimada}|}{|E_{real}|} \right) * 100 \quad (17)$$

En la Tabla 8, se muestra el error de la prueba MAPE para cada semana que forma la base de datos de prueba, en función del número de neuronas elegidas para la capa oculta de la red.

Se debe recordar que el número de neuronas de la capa oculta pueden ser entre 1 y 12 (el número máximo de neuronas no debe superar el doble de entradas de la red).

	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 5	Sem. 6	Sem. 7	Sem. 8
1 neurona	16,05%	11,40%	6,21%	13,25%	17,58%	9,07%	5,63%	11,81%
2 neurona	15,66%	11,50%	6,33%	13,04%	18,42%	10,55%	5,87%	12,39%
3 neurona	17,61%	11,78%	6,76%	12,60%	17,96%	10,91%	5,79%	13,62%
4 neurona	17,49%	12,25%	5,83%	14,41%	17,54%	10,49%	6,44%	12,89%
5 neurona	16,32%	11,42%	7,00%	12,50%	17,21%	8,05%	6,11%	13,36%
6 neurona	17,91%	12,01%	6,33%	12,57%	16,78%	10,73%	6,20%	13,11%
7 neurona	17,43%	12,07%	6,67%	11,52%	17,33%	10,65%	5,88%	13,19%
8 neurona	17,37%	11,85%	6,66%	12,84%	17,40%	10,12%	5,95%	12,84%
9 neurona	16,05%	11,26%	6,13%	12,18%	17,25%	10,55%	6,82%	12,56%
10 neurona	18,18%	12,69%	5,59%	12,02%	17,35%	10,58%	6,11%	13,34%
11 neuronas	18,68%	12,54%	5,90%	11,15%	16,77%	9,65%	6,39%	12,52%
12 neuronas	17,31%	11,83%	9,67%	12,73%	18,29%	10,35%	5,95%	13,05%

Tabla 8. MAPE en función del número de neuronas de la capa oculta

Como se observa, aparentemente los resultados no varían demasiado en función de si se elige un número de neuronas u otro. Observando el conjunto de las ocho semanas, con una sola neurona en la capa oculta se obtienen buenos resultados. Esto es debido a que en algunos casos, al estar la capa oculta formada por una sola neurona, el entrenamiento no se lleva a cabo correctamente (no converge, y se alcanza el gradiente máximo por lo que el entrenamiento finaliza). Los resultados obtenidos con una neurona en la capa oculta se descartan.

Para mostrar los resultados de las pruebas se ha elegido el siguiente modelo con menor error, es decir, el modelo con 5 neuronas en la capa oculta de la red.

A continuación, se mostrarán las gráficas de las pruebas y los valores estimados de la energía producida por el sistema fotovoltaico de cada una de las semanas que forman el conjunto de datos de prueba.

7.8.1. Semana 1

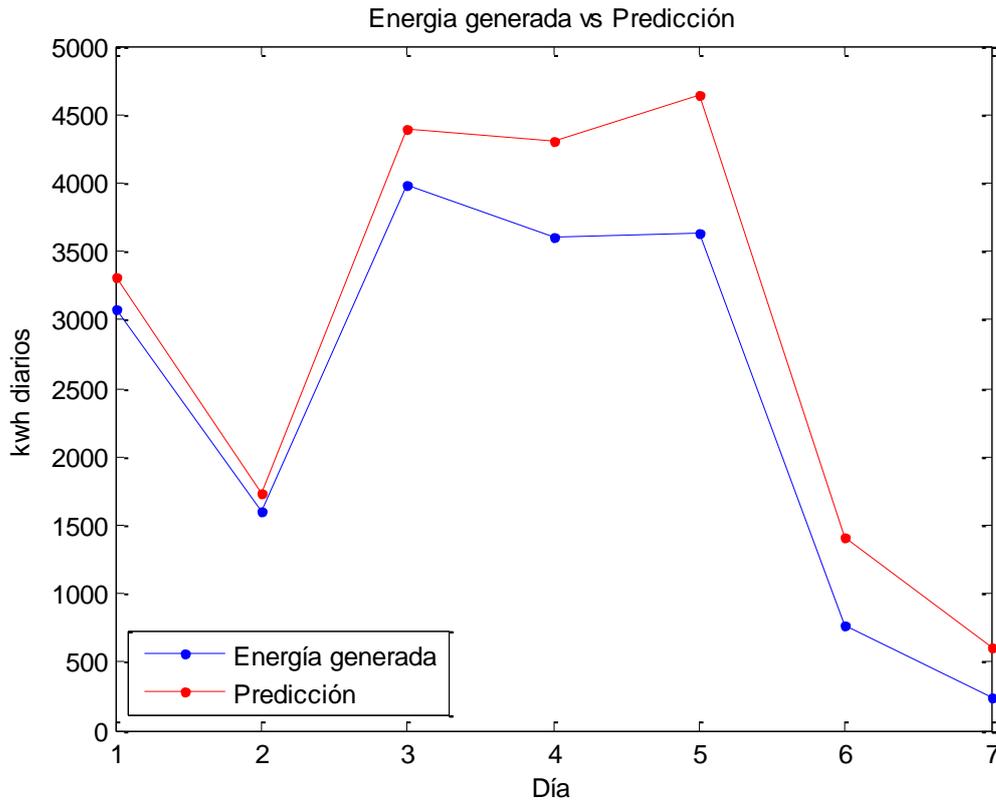


Figura 37. Prueba de la semana 1

Día	Valores reales (kWh diarios)	Valores estimados (kWh diarios)
1	3075,83	3275,95518
2	1599,44	1802,44746
3	3985,06	4340,26975
4	3597,74	4263,35246
5	3632,45	4615,93649
6	769,31	1470,55513
7	237,19	511,378897
MAPE=16,32%		

Tabla 9. Valores estimados de la semana 1

En la Figura 37 se muestra la gráfica comparativa entre la energía real producida en el parque fotovoltaico, y la predicción llevada a cabo por la red neuronal correspondiente a la semana 1 de la base de datos de prueba (1 de marzo del 2013 – 7 de marzo del 2013). Con esta prueba se obtiene un error promedio semanal MAPE del 16,32%.

7.8.2. Semana 2

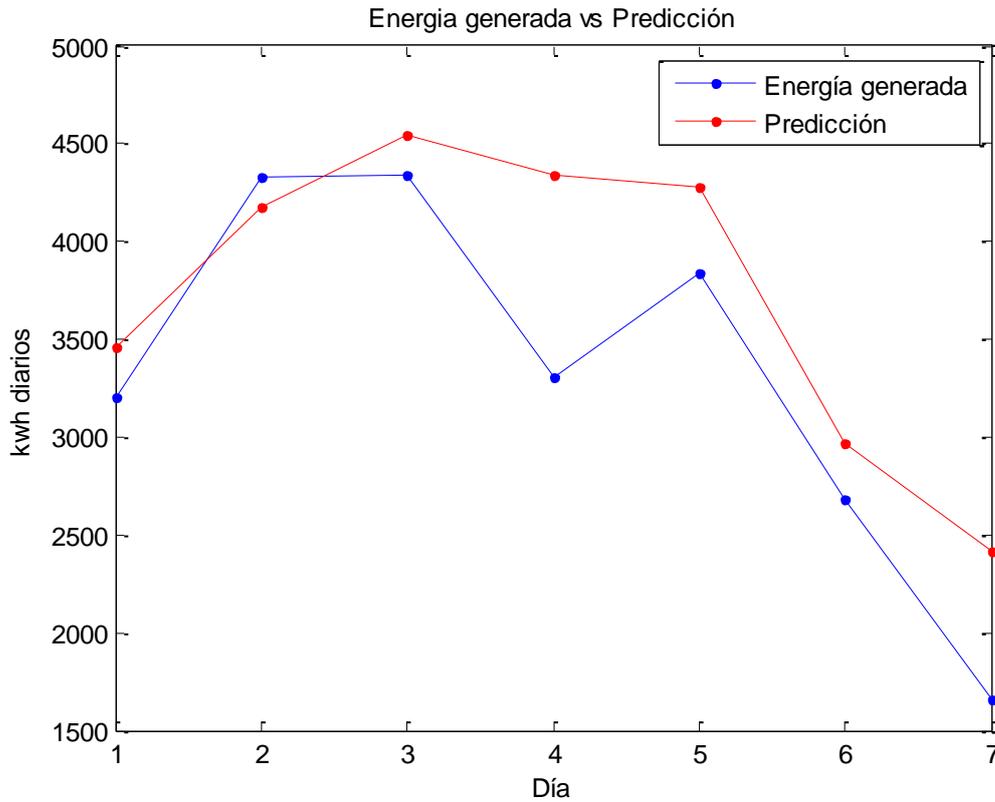


Figura 38. Prueba de la semana 2

Día	Valores reales (kWh diarios)	Valores estimados (kWh diarios)
1	3203,11	3463,94983
2	4334,32	4173,53476
3	4343,96	4540,14392
4	3303,94	4339,0509
5	3842,79	4280,53758
6	2678,63	2972,7521
7	1656,07	2416,48037
MAPE=11,42%		

Tabla 10. Valores estimados de la semana 2

En la Figura 38 se puede observar la gráfica comparativa entre la energía real generada en el parque fotovoltaico, y la estimación llevada a cabo por la red neuronal correspondiente a la semana 2 de la base de datos de prueba (8 de marzo del 2013 – 14 de marzo del 2013), obteniéndose un error promedio semanal MAPE del 11,42%.

7.8.3. Semana 3

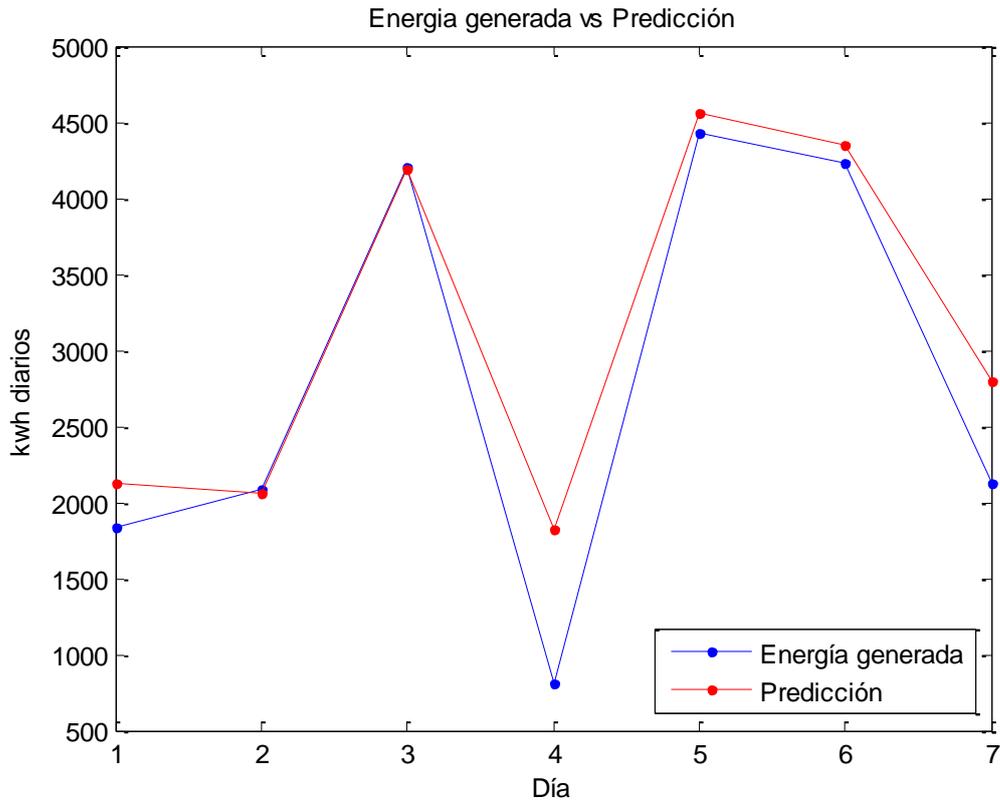


Figura 39. Prueba de la semana 3

Día	Valores reales (kWh diarios)	Valores estimados (kWh diarios)
1	1834,7	2130,62237
2	2089,5	2069,05019
3	4199,64	4287,91018
4	808,89	1858,24556
5	4422,62	4565,18308
6	4227,9	4430,75175
7	2130,22	2819,87625
MAPE=7,00%		

Tabla 11. Valores estimados de la semana 3

En la Figura 39 se observa la gráfica comparativa entre la energía real generada, y la predicción llevada a cabo por la red neuronal correspondiente a la semana 3 de la base de datos de prueba (15 de marzo del 2013 – 21 de marzo del 2013). Con esta prueba se obtiene un error promedio semanal MAPE del 7,00%.

7.8.4. Semana 4

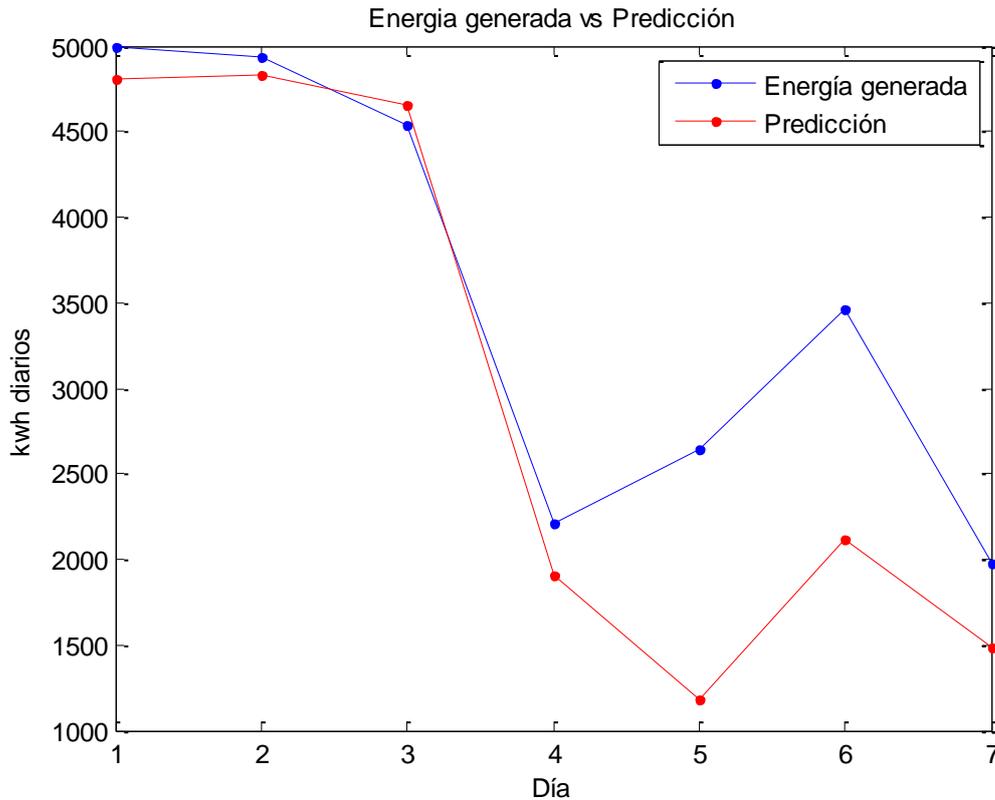


Figura 40. Prueba de la semana 4

Día	Valores reales (kWh diarios)	Valores estimados (kWh diarios)
1	4996,33	4806,35362
2	4940,2	4831,92774
3	4533,23	4657,67095
4	2206,45	1906,08651
5	2648,97	1182,38839
6	3456,48	2115,69969
7	1975,97	1480,94825
MAPE=12,50%		

Tabla 12. Valores estimados de la semana 4

En la Figura 40 se representa la gráfica comparativa entre la energía real producida en el parque fotovoltaico, y la predicción llevada a cabo por la red neuronal correspondiente a la semana 4 de la base de datos de prueba (22 de marzo del 2013 – 28 de marzo del 2013), obteniéndose un error promedio semanal MAPE del 12,50%.

7.8.5. Semana 5

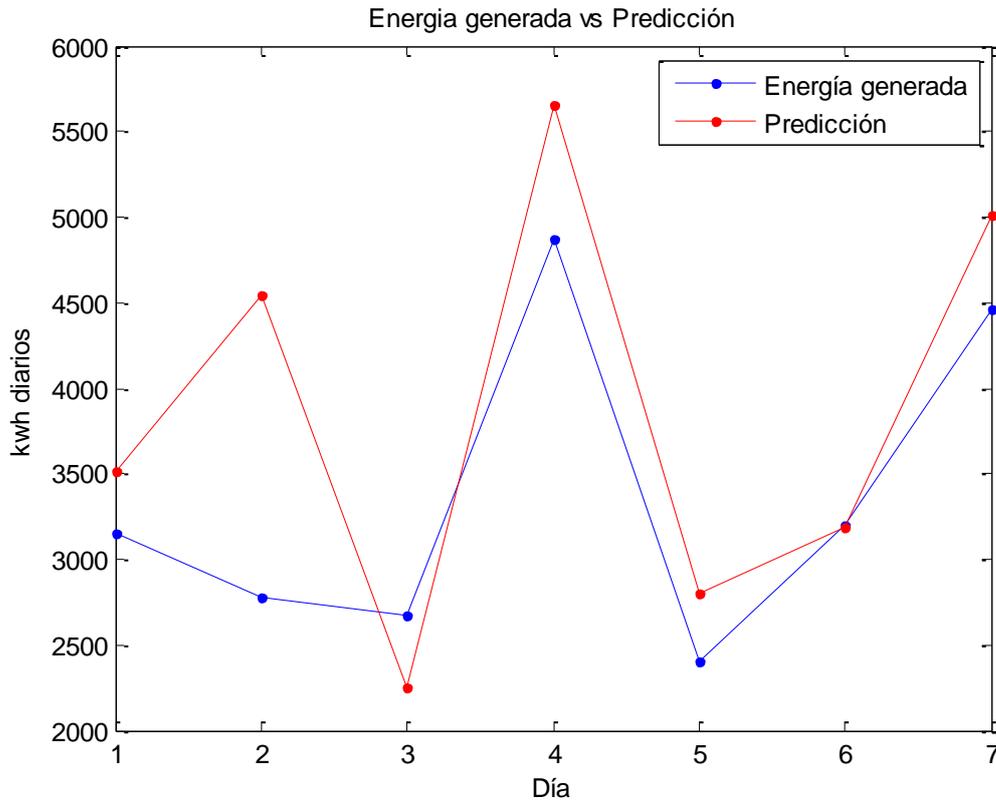


Figura 41. Prueba de la semana 5

Día	Valores reales (kWh diarios)	Valores estimados (kWh diarios)
1	3148,53	3511,48721
2	2774,54	4545,26831
3	2677,91	2248,23154
4	4868,34	5656,59304
5	2402,8	2795,92815
6	3199,01	3182,83804
7	4458,95	5015,63827
MAPE=17,21%		

Tabla 13. Valores estimados de la semana 5

En la Figura 41 se muestra la gráfica comparativa entre la energía real generada y la predicción llevada a cabo por la red neuronal correspondiente a la semana 5 de la base de datos de prueba (29 de marzo del 2013 – 4 de abril del 2013). Con esta prueba se obtiene un error promedio semanal MAPE del 17,21%.

7.8.6. Semana 6

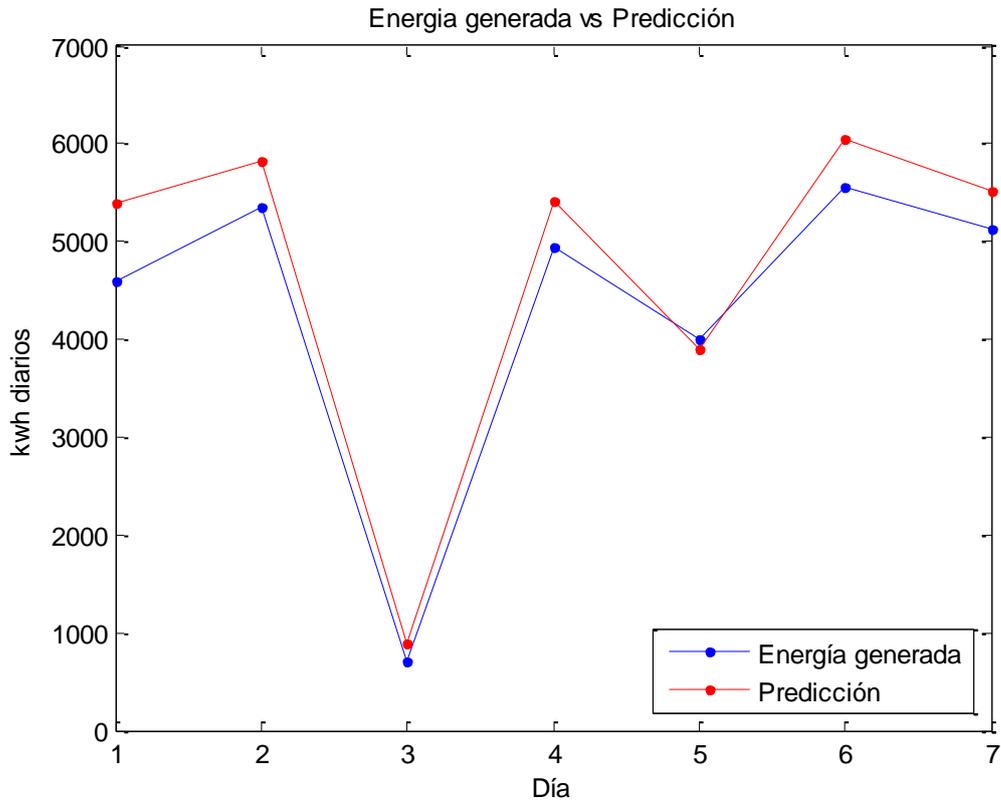


Figura 42. Prueba de la semana 6

Día	Valores reales (kWh diarios)	Valores estimados (kWh diarios)
1	4596,73	5354,92286
2	5348,91	5864,15756
3	696,87	853,322214
4	4941,27	5370,84266
5	4006,23	3925,47195
6	5558,28	5796,2479
7	5120,8	5467,68595
MAPE=8,05%		

Tabla 14. Valores estimados de la semana 6

En la Figura 42 se puede observar la gráfica comparativa entre la energía real producida por el parque fotovoltaico, y la estimación llevada a cabo por la red neuronal correspondiente a la semana 6 de la base de datos de prueba (5 de abril del 2013 – 11 de abril del 2013), obteniéndose un error promedio semanal MAPE del 8,05%.

7.8.7. Semana 7

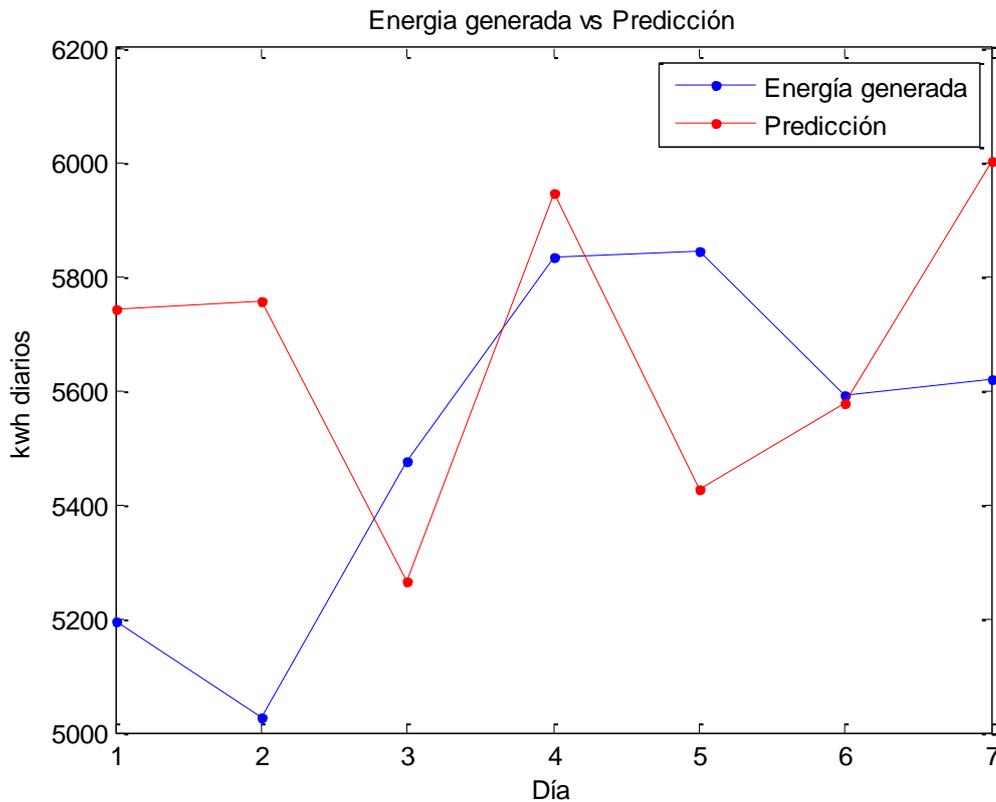


Figura 43. Prueba de la semana 7

Día	Valores reales (kWh diarios)	Valores estimados (kWh diarios)
1	5193,89	5722,42999
2	5025,87	5712,90919
3	5475,21	5221,82944
4	5832,2	5919,35556
5	5845,19	5389,91259
6	5590,6	5545,20639
7	5619,32	5978,92827
MAPE=6,11%		

Tabla 15. Valores estimados de la semana 7

En la Figura 43 se muestra la gráfica comparativa entre la energía real generada en el parque fotovoltaico y la predicción llevada a cabo por la red neuronal correspondiente a la semana 7 de la base de datos de prueba (12 de abril del 2013 – 18 de abril del 2013). Con esta prueba se obtiene un error promedio semanal MAPE del 6,11%.

7.8.8. Semana 8

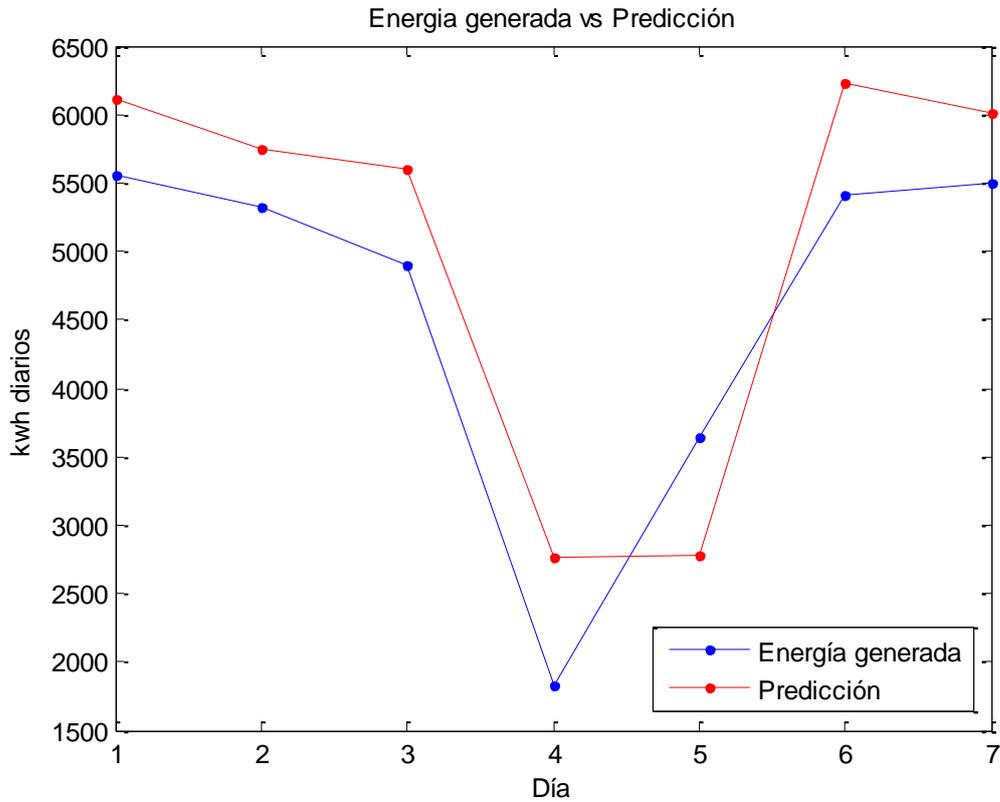


Figura 44. Prueba de la semana 8

Día	Valores reales (kWh diarios)	Valores estimados (kWh diarios)
1	5563,52	6116,81552
2	5327,08	5800,88771
3	4903,04	5638,44424
4	1834,44	2803,62159
5	3646,28	2840,17144
6	5403,52	6233,58918
7	5505,49	6028,66293
MAPE=13,36%		

Tabla 16. Valores estimados de la semana 8

En la Figura 44 se puede observar la gráfica comparativa entre la energía real producida, y la estimación llevada a cabo por la red neuronal correspondiente a la semana 8 de la base de datos de prueba (19 de abril del 2013 – 25 de abril del 2013). Con esta prueba se obtiene un error promedio semanal MAPE del 13,36%.

7.8.9. Conclusiones de las pruebas

Como se observa en las pruebas, en ninguno de los casos el error promedio semanal es mayor del 20%. A pesar de ser unos resultados muy aceptables, cabe destacar que seguramente podrían mejorarse en el caso de que existiese una base de datos más amplia. No para ampliar el período de entrenamiento, puesto que con un año de datos la red se entrena satisfactoriamente, si no para ajustar el conjunto de datos de prueba. De esta forma, si existiesen más años de muestras, se podría generalizar más con los datos de entrada de la red neuronal para llevar a cabo la prueba. Esto haría que las estimaciones llevadas a cabo por la red fuesen más fiables.

A pesar de todo, cabe destacar la imposibilidad de realizar una estimación buena en días atípicos. Aunque históricamente un día determinado siempre haya tenido buenas condiciones de irradiación, si el día a evaluar es diametralmente opuesto la predicción siempre va a ser mala. Por esto es importante evaluar los resultados semanales en conjunto, para suavizar el error que puede surgir al predecir la energía producida en estos días atípicos.

7.9. Modelo teórico de Osterwald

Además del modelo de predicción basado en redes neuronales, se ha desarrollado otro modelo teórico conocido como modelo de Osterwald, para poder comparar ambos modelos y ver cuál de ellos obtiene mejores resultados.

El modelo de Osterwald, además de su simpleza, es uno de los modelos teóricos con los que mejores resultados se obtienen. Su expresión general es la siguiente:

$$P_m = P_{n|STC} * \frac{G}{G_{STC}} * [1 - \gamma * (T_c - 25)] \quad (18)$$

Puesto que los intervalos existentes de temperatura e irradiancia tienen frecuencia horaria, se puede definir la energía generada con el modelo de Osterwald mediante la siguiente expresión:

$$E = P_{n|STC} * \frac{G}{G_{STC}} * [1 - \gamma * (T_c - 25)] \quad (19)$$

Donde G es la irradiación horaria en el plano al que están situados los módulos fotovoltaicos, G_{STC} es la irradiación en condiciones estándar de funcionamiento (1000 W/m^2), γ es el coeficiente de variación de potencia, T_c es la temperatura de la célula fotovoltaica y $P_{n|STC}$ es la potencia nominal del módulo fotovoltaico en condiciones estándar de funcionamiento.

El coeficiente de variación de potencia (γ) normalmente toma valores entre $-0,005 \text{ }^\circ\text{C}^{-1}$ y $-0,003 \text{ }^\circ\text{C}^{-1}$ cuando los módulos están formados por silicio cristalino. Ante la falta de información del valor que toma este parámetro en el parque fotovoltaico estudiado se asume $\gamma = -0,004 \text{ }^\circ\text{C}^{-1}$.

En [36] se puede ampliar la información acerca de este modelo matemático.

Teniendo en cuenta que en el sistema fotovoltaico en cuestión hay situados módulos fotovoltaicos a una inclinación de 3° y 15° , al hacer el cálculo teórico de la energía generada en el parque habrá que tener en cuenta ambas aportaciones, resultando la expresión de la energía total del sistema fotovoltaico:

$$E|_{TOTAL} = (P_{n|STC} 3^\circ * \frac{G|_{3^\circ}}{G_{STC}} + P_{n|STC} 15^\circ * \frac{G|_{15^\circ}}{G_{STC}}) * [1 - \gamma * (T_c - 25)] \quad (20)$$

La única diferencia entre esta expresión y la anterior, es que al multiplicar por la potencia nominal de los módulos fotovoltaicos, hay que tener en consideración que parte de ellos tienen una inclinación de 3° (353,3 kW) y la otra parte tiene una inclinación de 15° (606,7 kW).

Al tratarse de un modelo teórico, hay que tener en cuenta que dicho modelo no tiene en cuenta las pérdidas que se producen en el sistema fotovoltaico. Como se vio anteriormente, existen numerosos factores que afectan al rendimiento del sistema. Para conocer las pérdidas de la instalación fotovoltaica, puesto que se dispone de la base de datos que contiene la energía generada, se puede llevar a cabo un cálculo del promedio mensual de pérdidas

de la instalación fotovoltaica (a partir de la energía generada teórica, y puesto que se dispone de la energía real generada) con ayuda de la expresión:

$$\eta = \frac{E_{\text{obtenida}}}{E_{\text{nominal}}} \quad (21)$$

En la Tabla 17 se muestran las pérdidas del sistema fotovoltaico para cada mes del año.

	Pérdidas
Enero	7,96 %
Febrero	9,75 %
Marzo	15,89 %
Abril	17,45 %
Mayo	23,32 %
Junio	25,94 %
Julio	24,23 %
Agosto	23,03 %
Septiembre	18,45 %
Octubre	13,54 %
Noviembre	9,28 %
Diciembre	7,65 %

Tabla 17. Pérdidas mensuales del sistema fotovoltaico

Con estos datos, se puede utilizar la expresión de la energía total del sistema fotovoltaico (16) para calcular la energía total máxima en función de las irradiancias y de la temperatura de la célula. Una vez calculado este valor, habrá que tener en cuenta las pérdidas promedio del sistema fotovoltaico ese determinado mes para obtener la energía real producida en el parque fotovoltaico.

A continuación, en la Tabla 18 se muestran los valores obtenidos mediante la expresión de la energía total del sistema fotovoltaico, la energía real producida en el parque y la estimación mediante el modelo teórico teniendo en cuenta las pérdidas, para cada día que forma el conjunto de datos de prueba.

*Predicción mediante redes neuronales de la
producción de energía de un parque fotovoltaico*

Fecha	Energía real producida [kWh/día]	Energía teórica [kWh/día]	Estimación (E. teórica - Pérdidas) [kWh/día]
01-mar-13	3075,83	3653,85	3073,253
02-mar-13	1599,44	1854,003	1559,402
03-mar-13	3985,06	4952,373	4165,441
04-mar-13	3597,74	4822,372	4056,097
05-mar-13	3632,45	5243,396	4410,22
06-mar-13	769,31	1552,156	1305,519
07-mar-13	237,19	538,8152	453,1974
08-mar-13	3203,11	3933,454	3308,428
09-mar-13	4334,32	4907,732	4127,893
10-mar-13	4343,96	5315,02	4470,464
11-mar-13	3303,94	5020,063	4222,375
12-mar-13	3842,79	4856,452	4084,761
13-mar-13	2678,63	3280,533	2759,257
14-mar-13	1656,07	2617,504	2201,582
15-mar-13	1834,7	2210,519	1859,268
16-mar-13	2089,5	2153,497	1811,306
17-mar-13	4199,64	4817,898	4052,334
18-mar-13	808,89	1999,261	1681,578
19-mar-13	4422,62	5452,17	4585,82
20-mar-13	4227,9	5259,267	4423,57
21-mar-13	2130,22	3069,388	2581,662
22-mar-13	4996,33	5604,52	4713,962
23-mar-13	4940,2	5825,984	4900,235
24-mar-13	4533,23	5619,606	4726,65
25-mar-13	2206,45	1987,741	1671,889
26-mar-13	2648,97	1164,668	979,6026
27-mar-13	3456,48	2273,206	1911,994
28-mar-13	1975,97	1507,269	1267,764
29-mar-13	3148,53	4072,667	3425,52
30-mar-13	2774,54	5474,721	4604,788
31-mar-13	2677,91	2242,301	1885,999
01-abr-13	4868,34	6813,539	5624,577
02-abr-13	2402,8	3044,44	2513,185
03-abr-13	3199,01	3605,378	2976,24
04-abr-13	4458,95	6064,795	5006,488
05-abr-13	4596,73	6380,317	5266,952
06-abr-13	5348,91	7102,833	5863,389
07-abr-13	696,87	955,3015	788,6014
08-abr-13	4941,27	6429,963	5307,934
09-abr-13	4006,23	4612,172	3807,348
10-abr-13	5558,28	7178,229	5925,628

11-abr-13	5120,8	6429,816	5307,813
12-abr-13	5193,89	7021,972	5796,638
13-abr-13	5025,87	6916,64	5709,687
14-abr-13	5475,21	6198,935	5117,221
15-abr-13	5832,2	7141,882	5895,624
16-abr-13	5845,19	6250,114	5159,469
17-abr-13	5590,6	6583,984	5435,079
18-abr-13	5619,32	7097,551	5859,029
19-abr-13	5563,52	7393,947	6103,703
20-abr-13	5327,08	6946,149	5734,046
21-abr-13	4903,04	6745,869	5568,715
22-abr-13	1834,44	3097,355	2556,867
23-abr-13	3646,28	3099,211	2558,399
24-abr-13	5403,52	7692,539	6350,191
25-abr-13	5505,49	7315,362	6038,831

Tabla 18. Valores de energía estimados mediante el modelo de Osterwald

Agrupando los valores de energía estimados en conjuntos semanales, tal y como se hizo con el modelo basado en redes neuronales, se puede calcular también el error porcentual absoluto medio (MAPE). Los resultados son los que se muestran en la Tabla 19:

	MAPE
Semana 1	12,58%
Semana 2	7,76%
Semana 3	6,50%
Semana 4	18,52%
Semana 5	10,65%
Semana 6	6,60%
Semana 7	1,01%
Semana 8	8,47%

Tabla 19. MAPE semanal promedio calculado para el modelo teórico de Osterwald

En la Figura 45 se muestra una gráfica comparativa que abarca las ocho semanas que forman el conjunto de datos de prueba y representan la energía real producida en el parque fotovoltaico, la predicción llevada a cabo mediante el modelo basado en redes neuronales y la estimación realizada mediante el modelo matemático de Osterwald.

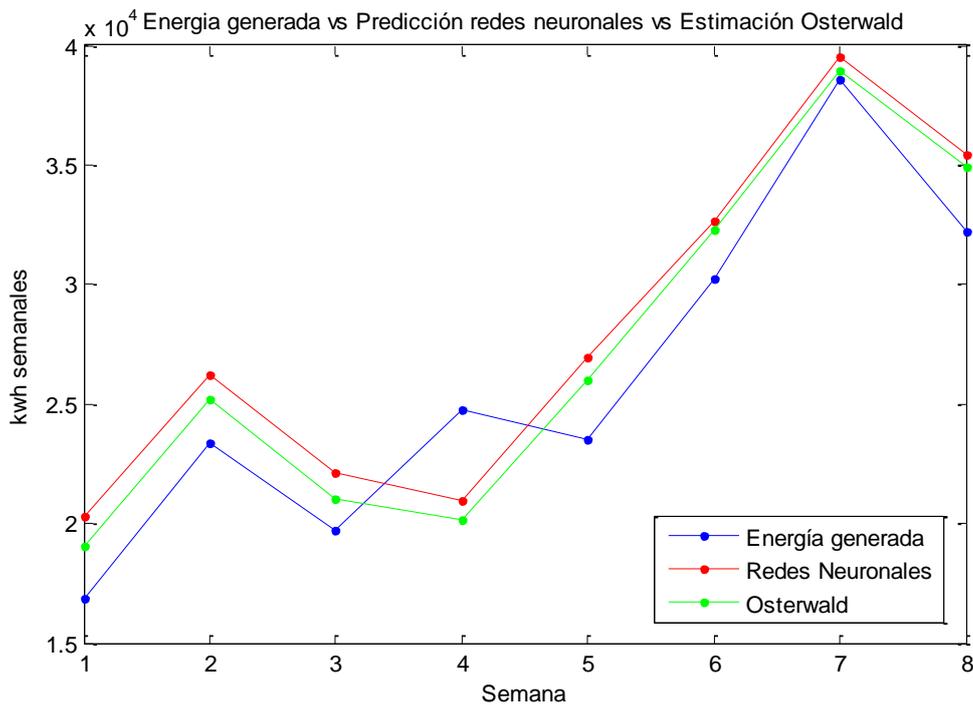


Figura 45. Gráfica comparativa de la energía real producida con los dos modelos desarrollados

Tal y como se observa en la Figura 45, la diferencia entre utilizar el modelo basado en redes neuronales, y el modelo teórico de Osterwald es prácticamente insignificante en términos de estimación de la producción. Ambos modelos presentan un gran comportamiento, y constituyen una gran herramienta para poder realizar predicciones acertadas sobre la producción de energía de un parque fotovoltaico.

Si se debe destacar algo, es que una vez desarrollado el modelo basado en redes neuronales tiene un manejo y un funcionamiento muy sencillo para el usuario final, mientras que el modelo teórico de Osterwald requiere llevar a cabo numerosas operaciones matemáticas, lo que conlleva un proceso mucho más tedioso. Este es el principal motivo por el que se considera más práctico el uso del modelo basado en redes neuronales.

7.10. Interfaz gráfica de usuario

En este apartado, se ha desarrollado una interfaz gráfica de usuario (GUI) mediante la herramienta Matlab para que el uso de la herramienta de predicción desarrollada sea lo más sencilla posible de cara al usuario final. La

intención del desarrollo de esta herramienta es que cualquier persona con poco conocimiento sobre el uso del entorno Matlab sea capaz de utilizar la herramienta de predicción mediante redes neuronales satisfactoriamente.

Se puede obtener información acerca de la creación de interfaces gráficas en [24].

La interfaz gráfica de usuario permite desarrollar visualmente un gran número de elementos (botones, menús, etiquetas, etc.) y situarlos dentro de ventanas para poder llevar a cabo diferentes acciones durante el transcurso del programa. La particularidad de esta herramienta del entorno Matlab es que mientras que un *script* se ejecuta en orden de la primera a la última línea, la interfaz gráfica de usuario se ejecutará en función de los deseos del usuario, pudiendo verse alterado el orden de ejecución del programa.

Dentro de la ventana “*Command window*” de Matlab, tecleando *guide* se iniciará la herramienta, pudiendo elegir entre diferentes estructuras predefinidas de interfaces gráficas. El nombre con el que se guardará el fichero es importante puesto que una vez guardado, para ejecutar la interfaz gráfica bastará con escribir en la ventana “*Command window*” el nombre del fichero creado.

Una vez elegido el tipo de estructura, se abre la ventana de edición (*GUIDE*), en la que se podrá empezar a diseñar la herramienta. El aspecto que presenta la herramienta se puede observar en la Figura 46.

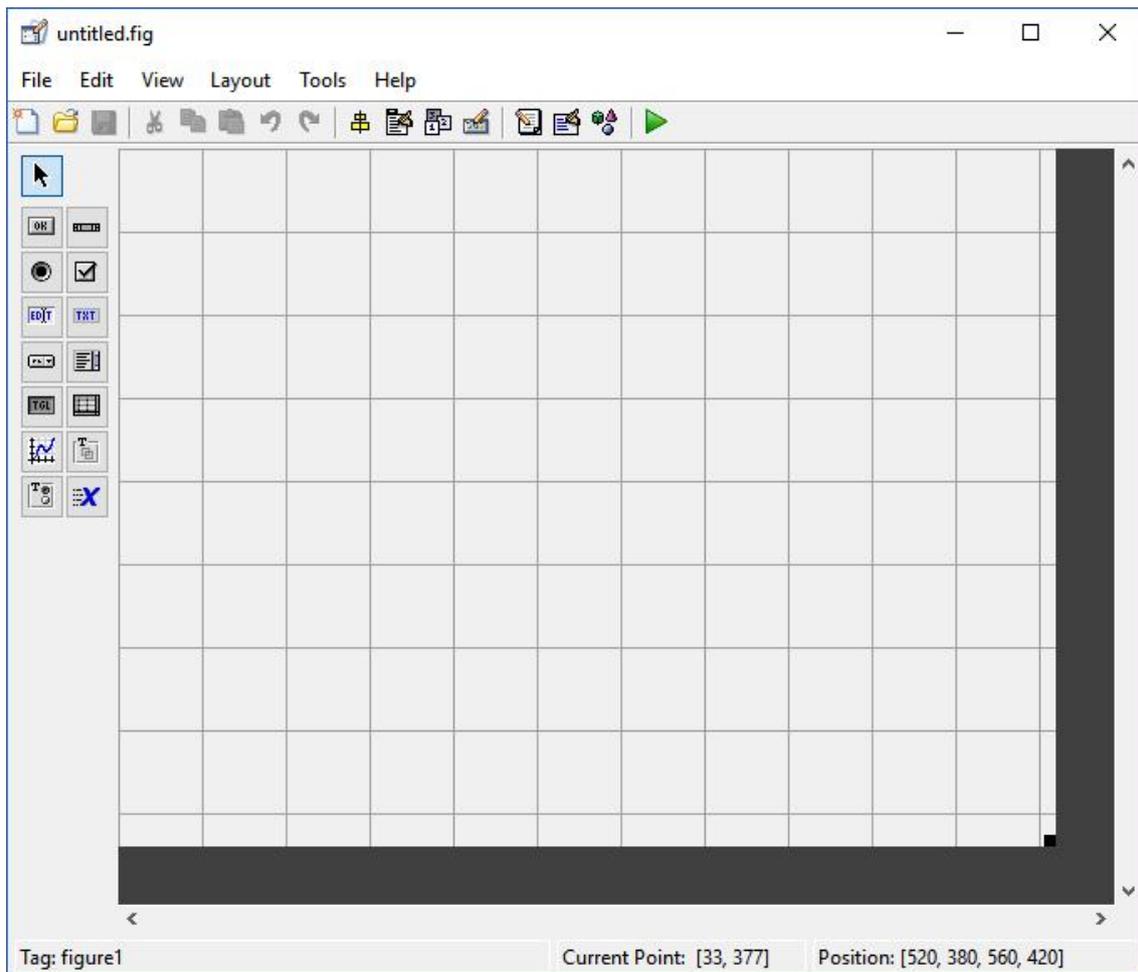


Figura 46. Aspecto de la herramienta GUIDE

Hay que tener en cuenta las diferentes posibilidades que presenta la herramienta. En la columna de la izquierda se muestran diferentes órdenes. Para la elaboración de esta interfaz gráfica de usuario, se han utilizado principalmente cuatro comandos: *Push Button*, *Static Text*, *Axes* y *Panel*.

Cada uno de los objetos gráficos diseñados, disponen de un identificador único para dicho objeto, de esta forma los elementos se podrán referenciar a lo largo del programa con dicho identificador y permitirán diferenciarlos del resto de elementos.

A continuación, se van a explicar los principales comandos utilizados para la creación de la interfaz:

- **Push Botton**: Son objetos cuadrados que se sitúan en la pantalla y llevan a cabo una acción cuando se pulsa sobre ellos. Su tamaño, color y posición es editable por el usuario.
- **Static text**: Este comando sirve para mostrar mensajes en pantalla, El texto a mostrar se debe especificar en el apartado “String” dentro de las propiedades del comando.
- **Axes**: Se tratan de ejes de coordenadas. Dentro de estos ejes se representará la gráfica que compara la energía producida en el parque fotovoltaico con la predicción llevada a cabo. El tamaño y la posición de los ejes puede ser editado.
- **Panel**: Sirve para agrupar distintos elementos de una ventana gráfica.

En la Figura 47 se muestra el aspecto de la interfaz gráfica de usuario una vez situados todos los elementos requeridos.

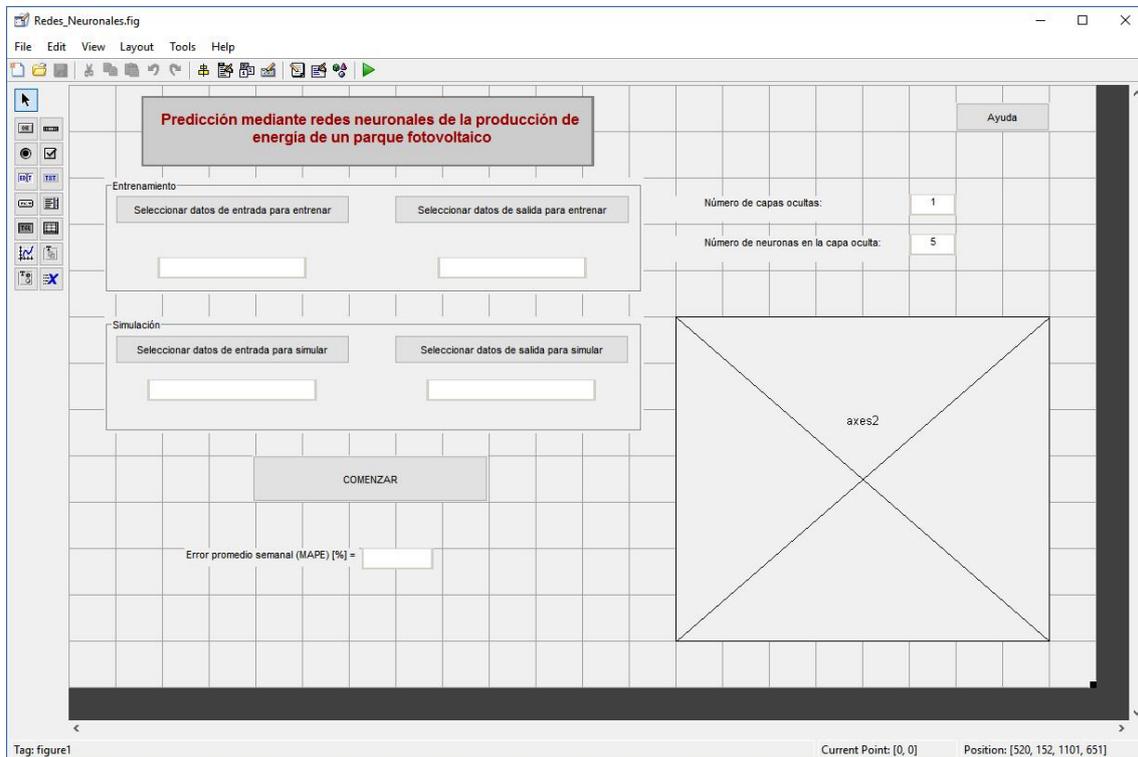


Figura 47. Aspecto final de la interfaz gráfica de usuario diseñada

Una vez creada la interfaz gráfica, se puede llevar a cabo la simulación de la herramienta. Para ello, se debe teclear “Redes_Neuronales” en la ventana “*Command window*” del entorno Matlab. Una vez hecho esto se presentará la ventana principal del programa. Como se observa, la ventana está formado por seis botones. Presionando el botón de la esquina superior derecha, “Ayuda”, se desplegará un mensaje de ayuda que mostrará al usuario las nociones básicas del uso del programa.

Los dos botones que están dentro del panel “Entrenamiento”, sirven para seleccionar los dos ficheros necesarios (entradas y salida) para llevar a cabo el entrenamiento de la red neuronal. Estos ficheros se encontrarán dentro de la carpeta “Entrenamiento” que puede verse una vez que se pulsa cualquiera de los dos botones superiores. El fichero de entradas, se deberá seleccionar con el botón izquierdo mientras que el fichero de salida se deberá seleccionar con el botón derecho. El nombre de los ficheros de entradas y salida de entrenamiento es respectivamente “entradas_12_meses_norm” y “salida_12_meses_norm”. Una vez seleccionados los dos ficheros, se observa que en la ventana “*Workspace*” de Matlab aparecen las variables “Entrada_train” y “Salida_train” puesto que ya se han cargado en la memoria del programa.

Debajo del panel “Entrenamiento”, se encuentra el panel “Simulación” en el que se observan otros dos botones. Estos botones se utilizarán para seleccionar los datos de prueba de la red neuronal, es decir, sirven seleccionar los datos sobre los que se va a realizar la predicción. El botón izquierdo sirve para seleccionar los datos de entradas a probar. Una vez que es pulsado, se debe seleccionar la carpeta “Simulación”, y seleccionar una de las ocho carpetas disponibles (cada una de ellas corresponde a una semana). Una vez abierta una de las carpetas, se debe seleccionar el fichero de entrada de prueba deseado, que tiene la forma “entrada_semana_X_test_norm”. Una vez hecho esto, con el botón derecho del panel se debe realizar el mismo proceso pero esta vez con el fichero de salida, de la forma “salida_semana_X_test_norm”. Es importante seleccionar los ficheros de

entradas y salidas de la misma carpeta de simulación, ya que si no los resultados serían erróneos.

En los recuadros blancos debajo de cada botón, se puede observar el nombre del fichero seleccionado. De esta forma se puede comprobar si se han producido errores a la hora de seleccionar los ficheros.

Finalmente, una vez elegidos los ficheros de entrenamiento y prueba correctamente, se presiona el botón “COMENZAR” para llevar a cabo el entrenamiento y prueba de la red neuronal. Durante el proceso aparecerán algunas gráficas referentes al proceso de entrenamiento, como por ejemplo el esquema de la red diseñada, el gráfico del MSE, etc. Además, la apariencia del programa principal cambiará y se mostrará una gráfica comparando la energía real producida por el parque fotovoltaico la semana seleccionada y la estimación llevada a cabo por la red neuronal, tal y como se muestra en la Figura 48.

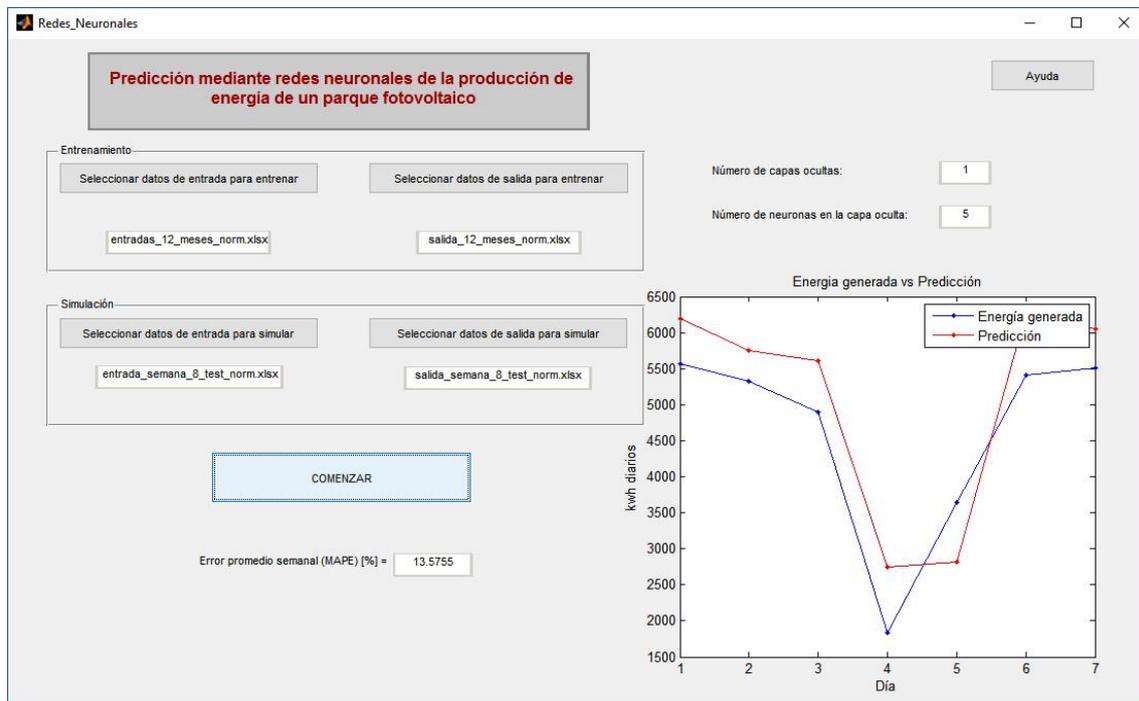


Figura 48. Ejecución de la interfaz gráfica de usuario

Además, debajo del botón “COMENZAR” aparecerá el error promedio porcentual de la semana seleccionada, para poder ver el error de predicción cometido.

Este proceso se puede llevar a cabo tantas veces como se desee seleccionando otros datos de simulación distintos.

8. Conclusiones

A lo largo de este Trabajo Fin de Grado se ha llevado a cabo un análisis de las diferentes técnicas existentes para la predicción de la producción de energía fotovoltaica, y se ha implementado un modelo de predicción basado en redes neuronales.

Para ello, ha sido necesario obtener una visión general sobre el panorama fotovoltaico actual. La liberalización del sector eléctrico, ha ayudado a entender la importancia de conseguir modelos de estimación que sean capaces de predecir con alta fiabilidad la energía fotovoltaica generada en un sistema. A día de hoy, el coste de la energía solar fotovoltaica está en su punto mínimo, y se ha consolidado como la energía más barata en más de 50 países. Para continuar con la expansión de este tipo de energía se deberán seguir estudiando técnicas para mejorar su eficiencia y almacenamiento, así como técnicas de estimación que ayuden a la planificación y gestión de la energía.

Los modelos de estimación basados en redes neuronales han demostrado ser uno de los mejores métodos existentes para llevar a cabo la predicción fotovoltaica. Hay que destacar la importancia que tiene la base de datos del sistema a estimar. Si la base de datos está incompleta o no contiene muchos registros, llevar a cabo una estimación acertada se vuelve muy complicado. Por ello, se debe fomentar la instalación de sistemas de adquisición de datos en los parques fotovoltaicos puesto que cuanto más completa sea la base de datos, mejores resultados de predicción se obtendrán.

En este caso en particular, la base de datos del parque fotovoltaico analizado contenía un año de registros. Ha sido suficiente para llevar a cabo un buen entrenamiento, pero para realizar las pruebas de la red neuronal, ha sido necesario recurrir a aplicaciones de terceros para obtener más registros y así poder realizar un procesado adecuado de las entradas y salidas para probar la red neuronal. A la hora de procesar las entradas y salidas de prueba, se debe partir de la premisa de que cuantos más registros existan, mejor será la predicción. Esto es debido al carácter impredecible de las condiciones

meteorológicas, cuantos más registros existan en la base de datos mejor se podrá generalizar.

Se ha implementado un modelo de redes neuronales basado en el perceptrón multicapa y se han llevado a cabo numerosas pruebas variando la topología de la red neuronal (en concreto, el número de neuronas de la capa oculta).

Los resultados de las pruebas ofrecen un error promedio semanal del 6,11% en el mejor de los casos (correspondiente a la semana 7), y un error del 17,21% en el peor de los casos (correspondiente a la semana 5). Por lo que se tratan de unos resultados muy satisfactorios teniendo en cuenta la base de datos existente. De haber tenido acceso a más registros de la base de datos los resultados posiblemente habrían mejorado.

En conclusión, se considera que la herramienta desarrollada es de gran utilidad para llevar a cabo la predicción de la energía producida en un parque fotovoltaico debido a su facilidad de uso y a los buenos resultados que ofrece.

9. Anexo

```
function varargout = Redes_Neuronales(varargin)
% REDES_NEURONALES MATLAB code for Redes_Neuronales.fig
% REDES_NEURONALES, by itself, creates a new REDES_NEURONALES or
% raises the existing
% singleton*.
%
% H = REDES_NEURONALES returns the handle to a new
REDES_NEURONALES or the handle to
% the existing singleton*.
%
% REDES_NEURONALES('CALLBACK', hObject, eventData, handles,...)
calls the local
% function named CALLBACK in REDES_NEURONALES.M with the given
input arguments.
%
% REDES_NEURONALES('Property','Value',...) creates a new
REDES_NEURONALES or raises the
% existing singleton*. Starting from the left, property value
pairs are
% applied to the GUI before Redes_Neuronales_OpeningFcn gets
called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to Redes_Neuronales_OpeningFcn via
varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Redes_Neuronales

% Last Modified by GUIDE v2.5 13-Dec-2016 15:57:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Redes_Neuronales_OpeningFcn, ...
                  'gui_OutputFcn', @Redes_Neuronales_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Redes_Neuronales is made visible.
```

```
function Redes_Neuronales_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Redes_Neuronales (see VARARGIN)

%axis off
% Choose default command line output for Redes_Neuronales
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Redes_Neuronales wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Redes_Neuronales_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu2

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in Entradas_entrenamiento.
function Entradas_entrenamiento_Callback(hObject, eventdata, handles)
% hObject    handle to Entradas_entrenamiento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Entrada_train;
[nombre direc]=uigetfile({'*.xlsx'},'Abrir Archivo');
Entrada_train=strcat(direc, nombre);
Entrada_train = xlsread (Entrada_train);
set(handles.text5, 'String', nombre)
assignin('base','Entrada_train',Entrada_train)

% --- Executes on button press in Salida_entrenamiento.
function Salida_entrenamiento_Callback(hObject, eventdata, handles)
% hObject    handle to Salida_entrenamiento (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Salida_train;
[nombre direc]=uigetfile({'*.xlsx'},'Abrir Archivo');
Salida_train=strcat(direc, nombre);
Salida_train = xlsread (Salida_train);
set(handles.text6, 'String', nombre)
assignin('base','Salida_train',Salida_train)

% --- Executes on button press in Comenzar.
function Comenzar_Callback(hObject, eventdata, handles)
% hObject    handle to Comenzar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% This script assumes these variables are defined:
%
%   entrada - input data.
%   salida  - target data.

global Entrada_train;
global Salida_train;
global Entrada_sim;
global Salida_sim;

inputs = Entrada_train;
targets = Salida_train;

% Create a Fitting Network
hiddenLayerSize = 5;
net = fitnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
```

```
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% For help on training function 'trainlm' type: help trainlm
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainlm'; % Levenberg-Marquardt

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse'; % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression','plotfit'};

% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);

%Weights
wi = net.IW{1,1};
assignin('base','Pesos_entradas',wi)
whl = net.LW{2,1};
assignin('base','Pesos_capa_oculta',whl)

% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets .* tr.valMask{1};
testTargets = targets .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs);
valPerformance = perform(net,valTargets,outputs);
testPerformance = perform(net,testTargets,outputs);

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotregression(targets,outputs)
figure, ploterrhist(errors)

%DATASET DE PRUEBA
```

```
inputs1 = Entrada_sim;
targets1 = Salida_sim;

% Test the Network - Dataset de Prueba
outputs1 = net(inputs1);
errors1 = gsubtract(targets1,outputs1);
performancel = perform(net,targets1,outputs1);

%Plots
%figure, plotperform(tr) %Es el mismo que para el entrenamiento
Salida_sim = Salida_sim*100;
outputs1 = outputs1*100;

assignin('base','Salida_estimada',outputs1)
assignin('base','Salida_real',Salida_sim)

axes(handles.axes2);
plot(Salida_sim, '-b.') % vector energía real
hold on
plot(outputs1, '-r.') % vector energía predicha
hold off
title('Energia generada vs Predicción')
legend('Energía generada', 'Predicción')
xlabel('Día') % Etiqueta el eje x
ylabel('kwh diarios') % Etiqueta el eje y

S_Salida_sim = sum(Salida_sim);
S_outputs1 = sum(outputs1);

MAPE_sim = (abs(Salida_sim - outputs1)/Salida_sim)*100;

set(handles.mape2,'String',num2str(MAPE_sim));

% --- Executes on button press in Ayuda.
function Ayuda_Callback(hObject, eventdata, handles)
% hObject    handle to Ayuda (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

helpdlg({'Pasos para ejecutar la aplicación:', ' - Pulse el botón
superior izquierdo y seleccione el fichero
"entradas_12_meses_norm".', ' - Pulse el botón superior derecho y
seleccione el fichero "salida_12_meses_norm".', ' - Con el botón
inferior izquierdo, seleccione el fichero de entrada semanal que
desee.', ' - Con el botón inferior derecho, seleccione el fichero de
salida semanal que deseese.', ' - Pulse COMENZAR.'});

% --- Executes on button press in Entradas simulacion.
function Entradas_simulacion_Callback(hObject, eventdata, handles)
% hObject    handle to Entradas_simulacion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Entrada_sim;
[nombre direc]=uigetfile({'*.xlsx'},'Abrir Archivo');
Entrada_sim=strcat(direc, nombre);
Entrada_sim = xlsread (Entrada_sim);
set(handles.text7, 'String', nombre)
```

```
assignin('base','Entrada_sim',Entrada_sim)

% --- Executes on button press in Salida_simulacion.
function Salida_simulacion_Callback(hObject, eventdata, handles)
% hObject    handle to Salida_simulacion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Salida_sim;
[nombre direc]=uigetfile({'*.xlsx'},'Abrir Archivo');
Salida_sim=strcat(direc, nombre);
Salida_sim = xlsread (Salida_sim);
set(handles.text8, 'String', nombre)
assignin('base','Salida_sim',Salida_sim)

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox1
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
listbox1

% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

10. Presupuesto

En este apartado, se va a llevar a cabo una estimación del coste total que supone la realización de este proyecto.

Coste del material:

Se incluyen los costes de equipamiento hardware y software. Se incluye el equipo informático necesario para la realización de este proyecto, tanto los dispositivos físicos como las licencias necesarias. En la Tabla 20 se muestra en detalle el coste del material hardware mientras que en la Tabla 21 se muestran los costes por adquisición del material necesario:

Equipo	Precio	Duración	Uso	TOTAL
Ordenador portátil Asus F541UA, i7- 6500, 8GB RAM	649 €	3 años	5 meses	70 €

Tabla 20. Costes de hardware

SUBTOTAL POR USO DE EQUIPOS 70 €

Material	TOTAL
Licencia Matlab R2012a	2.000 €
Licencia Neural Network Toolbox	1.000 €
Licencia Microsoft Office 2016	539 €

Tabla 21. Costes de adquisición del material

SUBTOTAL POR ADQUISICIÓN DEL MATERIAL 3.539 €

Total gastos por material 3.609 €

Coste de personal:

Por otro lado, los costes de mano de obra se han dividido en las distintas fases de desarrollo del proyecto. Se han empleado aproximadamente 300 horas para la finalización del proyecto. De estas 300 horas, aproximadamente unas 200 horas han consistido en labores de investigación y desarrollo software, mientras que el resto del tiempo empleado se ha dedicado a trabajos de documentación. En la Tabla 22 se muestran los costes generados por gastos de personal:

Trabajo	Nº horas	Coste por hora	TOTAL
Ingeniero	200	30 €	6.000 €
Mecanógrafo	100	10 €	1.000 €

Tabla 22. Costes de personal

Total gastos por empleados 7.000 €

Coste de ejecución material:

En este apartado se incluyen los gastos totales de realización del proyecto, tal y como se puede ver en la Tabla 23:

Coste	TOTAL
Coste del material	3.609 €
Coste del personal	7.000 €

Tabla 23. Coste de ejecución material

Coste de ejecución material 10.609 €

Gastos generales y beneficio industrial:

En este apartado se incluyen gastos producidos por el uso de las instalaciones donde se desarrolla el proyecto más el beneficio industrial. Se estima como recargo por estos conceptos un 25% del coste de ejecución material:

Gastos generales y beneficio industrial 2.582 €

Presupuesto de ejecución por contrata:

Se calcula sumando el presupuesto de ejecución material más los gastos generales y de beneficio industrial:

COSTE DE EJECUCIÓN MATERIAL 10.609 €

GASTOS GENERALES Y BENEFICIOS INDUSTRIALES 2.582 €

Total de ejecución por contrata 13.191 €

Honorarios:

Los honorarios facultativos por la ejecución del presente proyecto se determinan de acuerdo a las tarifas de los honorarios de los ingenieros en trabajos particulares, vigentes a partir del 1 de Septiembre de 1997 y dictadas por el Colegio Oficial de Ingenieros de Telecomunicación. Se aplica el coeficiente correspondiente al grupo XII, aplicaciones especiales de electrónica y aparatos de Telecomunicación, que es el 7%.

Total por redacción del proyecto	923 €
---	--------------

Importe total:

IMPORTE EJECUCIÓN POR CONTRATA	13.191 €
--------------------------------	----------

HONORARIOS DE REDACCIÓN	923 €
-------------------------	-------

Subtotal	14.114 €
-----------------	-----------------

IVA 21%	2964 €
---------	--------

TOTAL	17.078 €
--------------	-----------------

11. Manual de usuario

En la ventana principal de la aplicación *Redes_Neuronales*, tiene a su disposición seis botones (ver Figura 49).

1. **Presione el botón “Ayuda”**. En cualquier momento, si lo desea, puede acceder al menú de ayuda donde se explicará detalladamente el funcionamiento de la aplicación. Una vez consultado, presione el botón para cerrar.

Para llevar a cabo una prueba de la red neuronal debe seguir las siguientes instrucciones:

2. **Presione el botón “Seleccionar datos de entrada para entrenar”**.

Una vez hecho, se abrirá una ventana en la que deberá seleccionar el fichero de entradas de entrenamiento. La ruta del fichero es:

`\Entrenamiento\entradas_12_meses_norm.xls`.

Una vez seleccionado, presione el botón “Abrir” para cargarlo en la memoria de la aplicación.

3. **Presione el botón “Seleccionar datos de salida para entrenar”**.

De manera similar, una vez pulsado se abrirá una ventana en la que deberá seleccionar el fichero de salida de entrenamiento. La ruta del fichero es:

`\Entrenamiento\salida_12_meses_norm.xls`.

Una vez seleccionado, presione el botón “Abrir” para cargarlo en la memoria de la aplicación.

A partir de este momento, puede elegir entre ocho semanas distintas para probar la red neuronal implementada. Debe tener claro la semana sobre la que quiere hacer la prueba y escoger los datos de entradas y salida de prueba de la misma semana, de la siguiente manera:

4. **Presione el botón “Seleccionar datos de entrada para simular”**.

Una vez pulsado se volverá a abrir una ventana en la que deberá seleccionar el fichero de entrada de la semana a probar. La ruta del fichero es:

\Simulación\Semana x\entrada_semana_x_test_norm.xls.

Una vez seleccionado, presione el botón “Abrir” para cargarlo en la memoria de la aplicación.

5. Presione el botón “Seleccionar datos de salida para simular”.

Una vez pulsado se abrirá nuevamente una ventana en la que deberá seleccionar el fichero de salida de la semana a probar. La ruta del fichero es:

\Simulación\Semana x\salida_semana_x_test_norm.xls.

Una vez seleccionado, presione el botón “Abrir” para cargarlo en la memoria de la aplicación.

Una vez seleccionados estos parámetros, se puede llevar a cabo la prueba de la red neuronal:

- 6. Presione el botón “COMENZAR”.** De esta manera se iniciará el proceso de entrenamiento y prueba de la red neuronal. Durante la prueba, aparecerán diversas ventanas mostrándole diferentes gráficas que hacen referencia al proceso de entrenamiento de la red neuronal. Además, se mostrarán gráficamente los resultados de la prueba así como el porcentaje de error promedio semanal.



Figura 49. Guía del manual de usuario

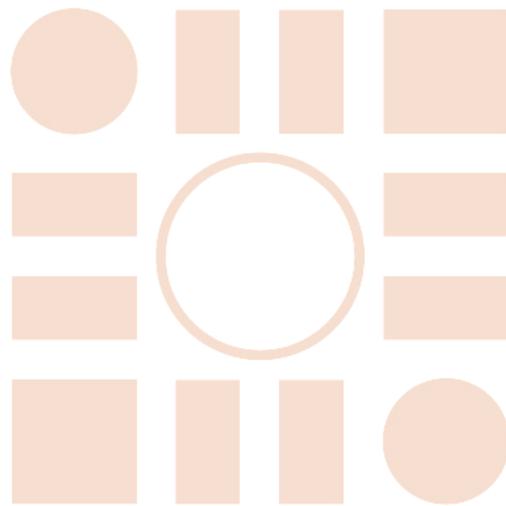
12. Bibliografía y referencias

- [1] Fernández M. "Energía Solar: Electricidad Fotovoltaica". Liber Factory, 2010.
- [2] "Annual Reports - REN21", REN21, 2017. [Online]. Disponible:
<http://www.ren21.net/about-ren21/annual-reports/>
- [3] McCulloch W, Pitts W. (1943). "A logical calculus of the ideas immanent in nervous activity". [Online]. Disponible:
<https://pdfs.semanticscholar.org/5272/8a99829792c3272043842455f3a110e841b1.pdf>
- [4] Gareta R, Romeo LM, Gil A. "Forecasting of electricity prices with neural networks". Energy Convers Manage 2006;47(13–14):1770–8.
- [5] Almonacid F, Rus C, Pérez-Higueras P, Hontoria L. "Estimation of the energy of a PV generator using artificial neural network". Renewable Energy 2009;34:2743–50.
- [6] Vapnik V. "The Nature of Statistical Learning Theory". Springer, 1995.
- [7] De Giorgi MG, Congedo PM, Malvoni M. (2016). "Data on Support Vector Machines (SVM) model to forecast photovoltaic power". [Online]. Disponible:
<http://www.sciencedirect.com/science/article/pii/S235234091630525X>
- [8] Chen JH, Wu W, Xie D. "Estimation of monthly solar radiation from measured temperature using support vector machines e a case study". Renewable Energy 2011;36:413–20.
- [9] Box G, Jenkins G. "Time Series Analysis: Forecasting and Control". Wiley, 1970.
- [10] Santos JM, Pinazo JM, Cañada J. (2003). "Methodology for generating daily clearness index index values Kt starting from the monthly average daily value Kt. Determining the daily sequence using stochastic models". [Online]. Disponible:
<http://www.sciencedirect.com/science/article/pii/S0960148102002173>
- [11] Cao Q, Ewing B, Thompson M. "Forecasting wind speed with recurrent neural networks". Eur J Oper Res 2012;221(1):148–54.
- [12] Zadeh L. (1965). "Fuzzy Sets". [Online]. Disponible:
<http://www.sciencedirect.com/science/article/pii/S001999586590241X>
- [13] Chugh A, Chaudhary P, Rizwan M. (2015). "Fuzzy logic approach for short term solar energy forecasting". [Online]. Disponible:
<http://ieeexplore.ieee.org/document/7443206/>

- [14] Liu Z, Zhang Z. (2016). "Solar forecasting by K-Nearest Neighbors method with weather classification and physical model". [Online]. Disponible: <http://ieeexplore.ieee.org/document/7747859/>
- [15] Rosenblatt F. (1961). "Principles of neurodynamics. Perceptrons and the theory of brain mechanisms" (No. VG-1196-G-8). CORNELL AERONAUTICAL LAB INC BUFFALO NY.
- [16] Valencia MA, Yáñez C, Sánchez L. (2006). "Algoritmo Backpropagation para Redes Neuronales: conceptos y aplicaciones". [Online]. Disponible: <http://www.repositoriodigital.ipn.mx/handle/123456789/8628>
- [17] Polo D, Pedraza L, Martínez E. "Comparison of Neural Network applied to prediction of times Series". Prospect, Vol 13, N° 2, 88-95, 2015.
- [18] Yu H, Wilamowsky B. (2010). "Levenberg-Marquardt Training". [Online]. Disponible: <https://pdfs.semanticscholar.org/708a/e4da2eabd4b6a19471cfa7fd072537d499a8.pdf>
- [19] "Neural network toolbox User Guide". Matlab. [Online]. Disponible: https://www.mathworks.com/help/pdf_doc/nnet/nnet Ug.pdf
- [20] Levenberg K. "A method for the solution of certain problems in least squares". Quarterly of Applied Mathematics, 5, 164–168, 1944
- [21] Marquardt D. "An algorithm for least-squares estimation of nonlinear parameters". SIAM Journal on Applied Mathematics, 11(2), 431–441, June 1963.
- [22] "Unsupervised Feature Learning and Deep Learning Tutorial", Ufldl.stanford.edu, 2017. [Online]. Disponible: <http://ufldl.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/>
- [23] "HelioClim-3 for Free - www.soda-pro.com", Soda-pro.com, 2017. [Online]. Disponible: <http://www.soda-pro.com/web-services/radiation/helioclim-3-for-free>
- [24] "App Building". Matlab. [Online]. Disponible: https://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf
- [25] Santiago-Rodríguez S, Arteaga-Ramírez R, Sangerman-Jarquín D, Cervantes-Osornio R, Navarro A. "Evapotranspiración de referencia estimada con Fao-Penman-Monteith, Priestley-Taylor, Hargreaves y RNA". Scielo.org.mx, 2017. [Online]. Disponible: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-09342012000800005

- [26] "Department of Radio Engineering FEE CTU Prague - Home Page", Radio.feld.cvut.cz, 2017. [Online]. Disponible: <http://radio.feld.cvut.cz/>
- [27] Gurney K. (2007). "Neural networks for perceptual processing: from simulation tools to theories". [Online]. Disponible: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2323553/>
- [28] Sánchez N. (2015). "Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento USD/COP spot intradiario". [Online]. Disponible: <http://revistas.uexternado.edu.co/index.php/odeon/article/view/4414>
- [29] Khatchaturian O, Padilha F. (2008). "Reconhecimento de variedades de soja por meio do processamento de imagens digitais usando redes neurais artificiais". [Online]. Disponible: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-69162008000400016
- [30] De Giorgi MG, Congedo PM, Malvoni M, Laforgia D. (2015). "Error analysis of hybrid photovoltaic power forecasting models: A case study of mediterranean climate". [Online]. Disponible: <http://www.sciencedirect.com/science/article/pii/S0196890415004422>
- [31] Tapia G, Glaría A. (2015). "Red neuronal artificial para detectar esfuerzo físico desde planos de fase de onda de pulso". [Online]. Disponible: <http://www.scielo.org.co/pdf/rinbi/v9n17/v9n17a02.pdf>
- [32] Hagan M. "Neuronal network Design". Thomson, 1996.
- [33] De Giorgi MG, Congedo PM, Malvoni M, Tarantino M. (2013). "Short-term power forecasting by statistical methods for photovoltaic plants in south Italy". [Online]. Disponible: <http://www.imeko.org/publications/tc19-2013/IMEKO-TC19-2013-034.pdf>
- [34] De Giorgi MG, Congedo PM, Malvoni M. (2016). "Data on photovoltaic power forecasting models for Mediterranean climate". [Online]. Disponible: <http://www.sciencedirect.com/science/article/pii/S2352340916302773>
- [35] Rus-Casas C, Aguilar J, Rodrigo P, Almonacid F, Pérez-Higueras P. (2014). "Classification of methods for annual energy harvesting calculations of photovoltaic generators". [Online]. Disponible: <http://www.sciencedirect.com/science/article/pii/S0196890413007279>
- [36] Gopalakrishnan K. "Soft Computing in Green and Renewable Energy Systems". Springer, 2011

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá