

Proyecto Fin de Carrera

Análisis de datos y evaluación mediante
Redes Neuronales para predicción
meteorológica

Data analysis and evaluation with Neural
Networks for meteorological forecast

Autor

Martín Dueso Molina

Directores

José Ramón Beltrán Blázquez
Julio David Buldain Pérez

Ingeniería de Telecomunicación
Escuela de Ingeniería y Arquitectura (EINA)
Universidad de Zaragoza
2017

RESUMEN

El proyecto realizado consiste en la evaluación y comparación de diversas arquitecturas de redes neuronales, tanto con redes clásicas (con perceptrón multicapa únicamente) como con deep learning (autoencoders con perceptrón multicapa y máquinas de Boltzmann con deep belief networks) para su uso en la predicción meteorológica de determinados umbrales de temperatura en un plazo entre 3 y 7 días.

Para ello se ha utilizado una base de datos con datos temporales de un número de estaciones de Alemania. De estos datos ha habido que seleccionar los considerados más importantes para tratar. Posteriormente estos datos de entrada se tienen que preprocesar antes de introducirlos a nuestra red.

Para poder trabajar con las configuraciones de red neuronal que se quería hubo que recurrir a una toolbox de Matlab que contuviera todas ellas. Tras ello se entrena la red con la mayor parte de estos datos y se deja una pequeña parte para su testeo. Después de evaluar con distintos parámetros para cada arquitectura neuronal se ha tratado de discutir cuál es el mejor de ellos.

ÍNDICE

1.	Capítulo 1. Introducción	5
	1.1. Objetivo y Alcance.....	6
	1.2. Estado del Arte.....	6
	1.3. Descripción de la Memoria.....	8
2.	Capítulo 2. Base de Datos	10
	2.1. Variables de la Base de Datos.....	11
	2.2. Localización de las Estaciones.....	12
3.	Capítulo 3. Metodología	16
	3.1. Arquitecturas de Redes Neuronales Artificiales usadas.....	17
	3.1.1. Arquitectura 1: Autoencoders con Perceptrón Multicapa.....	17
	3.1.2. Arquitectura 2: Perceptrón Multicapa.....	19
	3.1.3. Arquitectura 3: Máquinas de Boltzmann con Deep Belief Network.....	20
	3.2. Variables de Entrada a la Red.....	21
4.	Capítulo 4. Análisis de los Resultados	26
	4.1. Entrenamiento y Validación de la Red Neuronal.....	27
	4.2. Pruebas realizadas.....	28
	4.3. Análisis de los Resultados.....	30
	4.3.1. Resultados para la Arquitectura 1.....	30
	4.3.2. Resultados para la Arquitectura 2.....	34
	4.3.3. Resultados para la Arquitectura 3.....	37
	4.3.4. Comparación y Discusión de los Resultados.....	40
5.	Capítulo 5. Conclusiones	44
	5.1. Conclusiones.....	45
	5.2. Líneas Futuras.....	46
6.	Bibliografía	47

ANEXOS

ANEXO I. Modelos meteorológicos	51
ANEXO II. Introducción a las Redes Neuronales Artificiales	56
ANEXO III. Ampliación Base de Datos	65
ANEXO IV. Funciones y Código usado	68

Capítulo 1

Introducción

En este capítulo se resumirá en qué consiste el proyecto y cuál es su objetivo o finalidad, se abordará el marco en el que está basado y su estado del arte y se describirá brevemente el contenido de la memoria.

1.1. Objetivo y Alcance

El objetivo principal será la predicción probabilística de 3 eventos meteorológicos como son temperatura mínima menor de 0°C, temperatura máxima menor de 0°C y temperatura máxima mayor de 25°C. Se trata de ver si dentro de las limitaciones de nuestro modelo y de la base de datos usada, es capaz de modelizar bien la superación o no de esos umbrales de temperatura.

Debido a que el uso de modelos probabilísticos cada vez es mayor en la meteorología, otro objetivo de este proyecto es si el uso de técnicas neuronales como el Deep Learning puede aportar mejoras a lo existente. Y como el uso de este tipo de arquitecturas hasta ahora es muy poco común en meteorología, se intenta avanzar en ese aspecto y poder aportar algo más en ese sentido.

Se buscará modelizar la probabilidad de que tenga lugar uno de dichos eventos meteorológicos en cada una de las arquitecturas de redes neuronales aquí planteadas, es decir: por un lado los algoritmos clásicos (un perceptrón multicapa), y por otro lado deep learning machines (autoencoders con perceptrón multicapa y máquinas de Boltzmann con deep belief networks), además de la comparación de unas con otras para ver cuál proporciona mejores resultados.

1.2. Estado del Arte

Como al trabajar con redes neuronales es importante tener una gran cantidad de datos, se ha tenido que recurrir a una base de datos proporcionada por el Grupo de Meteorología de Santander [1] que contenía los datos de 12 estaciones de Alemania a lo largo de 30 años, tomando un dato diario.

Esta base de datos usada está enmarcada dentro de VALUE [2] para Europa, que combinado con el programa EURO-CORDEX [3], está inmerso en el uso de diversas series temporales de varios observatorios de distintos países para utilizarlos en la predicción meteorológica y ver su funcionamiento en diversos puntos del continente europeo.

- **Trabajos previos que no usan Deep Learning**

Sin embargo, las técnicas usadas para estos proyectos, como se puede ver en [4], las llamadas técnicas de Downscaling, ver anexo I, no son exactamente lo que buscamos, ya que a pesar de que se usan para una determinada variable como en

nuestro caso es la temperatura, no usan técnicas de Deep Learning para la predicción, y solo en algunos casos utilizan métodos con redes neuronales ya que a veces usan métodos propiamente estadísticos. Además, aparte de datos de estaciones suelen usar datos de modelos globales de circulación, ver anexo I, mientras que en este proyecto solo se usan datos de estaciones.

Con esta base de datos también se han intentado hacer experimentos multi-variable para predecir varias variables para la misma salida pero de momento no ha habido ningún resultado.

Otros trabajos o tesis como esta [5] utilizan métodos puramente estadísticos de redes bayesianas para predecir una variable que en su caso es la precipitación.

En otro trabajo previo realizado en esta tesis [6] se hacen comparativas entre métodos puramente estadísticos y algunos métodos neuronales para algunos eventos meteorológicos, pero no se entra en un resultado propiamente dicho de predicción, sino que se hacen validaciones de parámetros de las redes que usan.

- **Trabajos previos que usan Deep Learning**

Como se pretende avanzar en el terreno de la probabilidad introduciendo técnicas de Deep Learning para poder comparar con técnicas de redes neuronales clásicas se buscan trabajos previos que sí hayan usado estas técnicas.

En diversos artículos en los que se usa Deep Learning para la predicción meteorológica, o de algún índice relacionado con ella, se suelen indicar sobre todo los valores del error cuadrático medio (MSE) o el error medio absoluto (MAE) de la red, además del error entre lo previsto y el suceso que se dio finalmente.

En cuanto a un índice normalizado como es el SPI [7], que es un índice para determinar la precipitación que ha caído y el grado de sequía que hay en un territorio, se han usado comparaciones entre dos modelos de redes neuronales que en este proyecto se usan, como son perceptrón multicapa y deep belief network, comparando resultados entre ellos.

El otro modelo de red neuronal usado aquí, que usa autoencoders y perceptrón multicapa, es utilizado para calcular predicción de lluvia [8] con una serie de entradas combinando los datos de días anteriores, nuestro proyecto complementarían bastante a este, puesto que aquí la predicción está referida a umbrales de temperatura, añadiendo además otras configuraciones de redes neuronales.

Otros trabajos realizados [9] [10] con Deep Learning han hecho previsiones, pero a intervalos de tiempo menores, puesto que tenían más datos al día y de más variables, lo que hace poder tener previsiones más precisas. En uno de ellos se prueban Autoencoders usando una capa con SVR (Support Vector Regression) [9].

Hay otro trabajo que usa también técnicas de Deep Learning para calcular en este caso velocidad de viento [11] pero a muy corto plazo, para los próximos minutos u horas.

En la mayor parte de estos trabajos lo que se hace es comparar la predicción con el fenómeno observado finalmente para ver cuánto se equivoca el modelo y evaluar el MSE (Mean Squared Error) o RMSE (Root Mean Squared Error) y algunas veces el MAE (Mean Absolute Error) de la red neuronal en el tiempo.

En el caso de este proyecto, la manera de evaluar la red es si acierta que la temperatura ha superado un determinado umbral o no, por lo que en lugar de modelizar una serie temporal de la variable meteorológica, en realidad tratamos con un problema de clasificación binaria: se produce el evento o no se produce.

1.3. Descripción de la Memoria

La memoria de este proyecto está formada por 5 capítulos incluido este de introducción, además de 4 anexos complementarios.

En el segundo capítulo se describe la base de datos con todas las variables que vienen en ella además de la localización de las estaciones elegidas.

En el tercer capítulo se describe la metodología empleada de una forma global; es decir, cada una de las arquitecturas de redes neuronales que se han usado. En cada una de ellas se incluyen los valores entre los que se han probado los distintos parámetros que tiene cada arquitectura. También se describen las variables de entrada a la red.

El cuarto capítulo de la memoria es en el que se explican las pruebas realizadas y se analizan los resultados obtenidos para cada arquitectura de red neuronal usada. Se discute cuáles han sido los mejores.

En el quinto capítulo es en el que se exponen las conclusiones que se han sacado de este proyecto en función de los resultados. Se discuten cuáles han sido las mayores dificultades que se han encontrado, y se abordarán posibles mejoras para un futuro.

En el anexo I se describen los modelos meteorológicos de una manera global, los tipos que hay y la incursión de la probabilidad en ellos.

En el anexo II se hace una introducción a las redes neuronales en cuanto a sus

ventajas e inconvenientes, su organización y se describen los tipos de modelos neuronales usados en el proyecto.

En el anexo III se amplía la información sobre la base de datos usada para los datos de entrada y la toolbox usada para las distintas redes neuronales probadas.

En el anexo IV se muestran código y algunas funciones que se han tenido que usar para este proyecto.

Capítulo 2

Base de Datos

En este capítulo se describe la base de datos que se ha utilizado con todas las variables que vienen en ella y la localización de las estaciones elegidas.

2.1. Variables de la Base de Datos

La base de datos que se nos ha proporcionado desde el Grupo de Meteorología de Santander [1] consta de varias series temporales de 30 años de 12 estaciones de Alemania. Las series temporales van desde el 1 enero de 1979 al 31 de diciembre de 2008 tomando un dato diario de cada variable durante todo este tiempo, en total 10958 datos. Las variables meteorológicas de estas series temporales son las mostradas a continuación, al lado de cada variable viene la nomenclatura con la que es usada la variable en la base de datos, ver anexo III para más detalle.

- Temperatura máxima (tmax)
- Temperatura mínima (tmin)
- Temperatura media (tmean)
- Precipitación (precip)
- Estado del cielo (cloud)
- Humedad relativa (hurs)
- Velocidad del viento (wss)
- Velocidad máxima del viento (wssmax)

Cabe destacar que han sido usadas aquellas variables que no tienen falta de datos, ya que hay varias de ellas que no contienen toda la serie temporal completa. Lo mismo ocurría con una estación, en la que la variable del estado del cielo (variable que resulta interesante como entrada a la red) no la contenía en su totalidad y por ello se decidió prescindir de ella y quedarnos con 11 estaciones.

Finalmente se decidió que las variables de las series temporales de las que se toman datos, para las que posteriormente serán variables de entrada, serán temperatura máxima, temperatura mínima, precipitación y estado del cielo. Con estas variables se hacen una serie de combinaciones para tener las variables de entrada a nuestro sistema (tomar datos de los 3 días anteriores o hacer medias de los 5 días anteriores), además se han introducido más variables como el mes en el que se toma el dato o la localización geográfica (altitud, latitud, longitud) de cada estación, puesto que esto también nos lo proporciona la base de datos, ver anexo III. Por lo tanto, las variables finalmente usadas como entrada a nuestro sistema, detalladas más en profundidad en el apartado 3.2, son las siguientes:

- Lluvia de últimos 3 días
- Media de lluvia últimos 5 días
- Temperatura máxima de los últimos 3 días
- Media de las máximas de los últimos 5 días
- Temperatura mínima de los últimos 3 días
- Media de las mínimas de los últimos 5 días
- Diferencia entre máxima y mínima
- Estado del cielo de los últimos 3 días
- Media del estado del cielo de los últimos 5 días
- Mes del dato de entrada
- Latitud, longitud y altitud

Los valores de estas variables han tenido que ser preprocesados normalizando su valor entre 0 y 1 para que las entradas “pesen” lo mismo y nuestra red pueda tratar adecuadamente los datos. La fórmula que se ha usado para la normalización de los datos en bruto ha sido la siguiente:

$$\text{dato procesado} = \frac{\text{dato original} - \text{dato mínimo}}{\text{dato máximo} - \text{dato mínimo}} \quad (1)$$

Al tener un dato diario durante los 30 años de la serie temporal, contamos con 10958 datos en cada una de las estaciones. Los primeros 5 se desechan para poder hacer las medias correspondientes de los 5 días y los últimos 9 días también se han decidido no cogerlos para poder tener varios días futuros para la predicción. Por tanto, tenemos 10944 datos en cada estación. De los cuales, la mayoría se usan para el entrenamiento y el resto para testear.

2.2. Localización de las Estaciones

En la base de datos aparece la localización de todas las estaciones, tanto altitud como latitud y longitud. A continuación en la siguiente tabla vemos la localización de cada una de las estaciones, y la ciudad o el accidente geográfico en el que se encuentran.

	Latitud	Longitud	Altitud	Ciudad/Zona
Estación 1	53,0464	8,7992	4	Bremen
Estación 2	47,8017	11,0117	977	Peißenberg
Estación 3	52,3833	13,0639	81	Postdam
Estación 4	47,4219	10,9867	2964	Zugspitze
Estación 5	54,1764	7,8931	4	Helgoland
Estación 6	51,1292	13,7558	227	Dresde
Estación 7	51,8000	10,6200	1142	Brocken
Estación 8	48,9733	8,3308	116	Karlsruhe
Estación 9	54,6817	13,4367	42	Putgarten
Estación 10	47,3989	10,2767	806	Oberstdorf
Estación 11	49,0433	12,1031	365	Ratisbona

Tabla 1. Coordenadas y altitud de las estaciones

Gracias a una herramienta en Matlab llamada Meteolab [12], se pueden representar, las estaciones que queremos dentro de Europa. La siguiente figura corresponde a la representación geográfica de las estaciones tratadas en este proyecto.

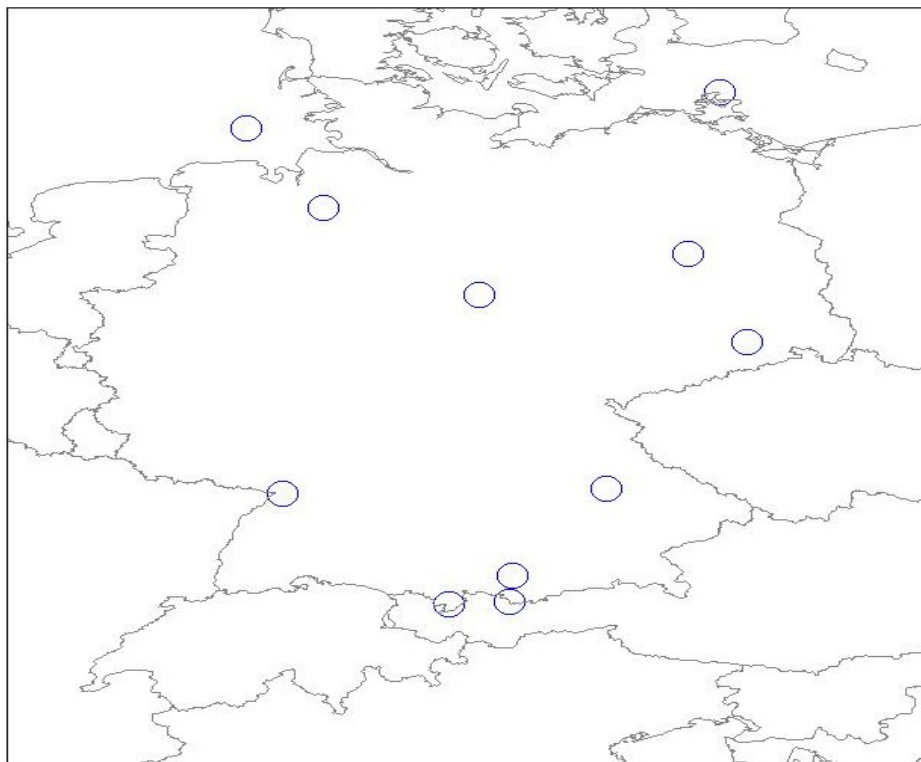


Figura 1. Mapa de Alemania con las estaciones

A pesar de que tenemos tanto las coordenadas como la altitud, para tener una mejor representación de la zona geográfica que abordamos, se muestra en la siguiente figura un mapa físico de Alemania [13] con la situación de las 11 estaciones para observar mejor si se encuentran en una zona llana o de montaña.

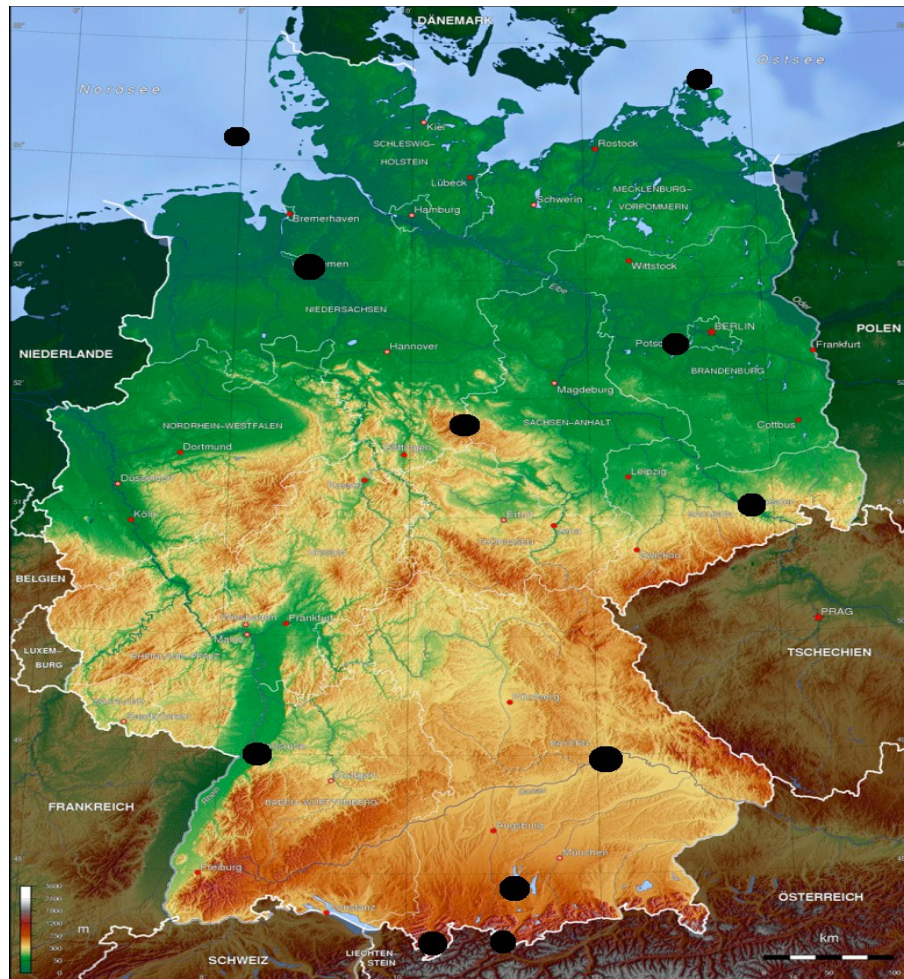


Figura 2. Mapa físico de Alemania con la situación de las estaciones [13]

Estas son las 11 estaciones que se usan para la predicción meteorológica en el proyecto. A continuación se describe a grandes rasgos el clima para las zonas en las que están situadas, puesto que Alemania no tiene un clima homogéneo.

En las 4 estaciones situadas más al norte se espera un clima oceánico con inviernos fríos, más cuanto más al este, y veranos suaves. Las precipitaciones serán más abundantes cuanto más al oeste y menos en el este, la zona noreste de Alemania es la zona que menos precipitación recibe del país. En la estación situada en la zona central (Estación 7 de la Tabla 1), y en las 3 estaciones más meridionales el clima es

predominantemente de montaña debido a su altitud, con inviernos muy fríos y veranos suaves, además de precipitaciones abundantes. Y en las 3 estaciones restantes el clima es de transición entre oceánico y continental, siendo más continentalizado cuanto más al este, con veranos algo cálidos e inviernos fríos. Las precipitaciones también son intermedias en estas estaciones, no tan abundantes como en zonas de montaña pero mayores que en las estaciones del noreste. Cabe destacar por tanto, la problemática de tener distintos tipos de clima a tratar dependiendo de dónde se sitúe la estación.

Capítulo 3

Metodología

En este capítulo se describen las arquitecturas de redes neuronales que se han usado de una manera global, con los parámetros utilizados para cada una, además de las variables de entrada a la red.

3.1. Arquitecturas de Redes Neuronales Artificiales usadas

En este proyecto vamos a tratar con tres arquitecturas de redes neuronales artificiales que incluyen varios modelos neuronales como son: Stacked Denoising Autoencoders, Perceptrón Multicapa, Restricted Boltzmann Machines y Deep Belief Networks [14] (ver anexo II).

Como el programa utilizado es Matlab, se ha tenido que buscar una librería en Matlab que contuviera los modelos neuronales requeridos para este proyecto. Finalmente se ha usado una toolbox [15] que contenía configuraciones de Deep Learning y además nos permitía probar con un perceptrón multicapa para poder comparar el funcionamiento con una configuración con algoritmos clásicos. Para ver archivos de la toolbox ver anexo III.

En este apartado, toca entrar a describir las arquitecturas de las redes neuronales usadas en este proyecto junto con los rangos en los parámetros que se han tomado en cada una de ellas (número de iteraciones, tamaño de lote o capas ocultas usadas).

3.1.1. Arquitectura 1: Autoencoders con Perceptrón Multicapa

La primera arquitectura que se ha probado es la que usa Stacked Denoising Autoencoders [10] más un perceptrón multicapa, se entrenan dos redes autoencoders sin supervisión, pues la salida es la propia entrada, de forma que el segundo autoencoder usa como entradas las salidas de la capa oculta del primer autoencoder. Una vez generadas, se apilan para implementar las dos primeras capas procesadoras de la arquitectura de la red, a cuya salida se coloca un perceptrón multicapa (Feed Forward Neural Network o Multilayer Perceptrón), también con varias capas ocultas, como aproximador de la salida deseada. La parte de deep learning la introducirían en esta arquitectura los autoencoders. En la siguiente figura se puede ver la configuración que se tiene:

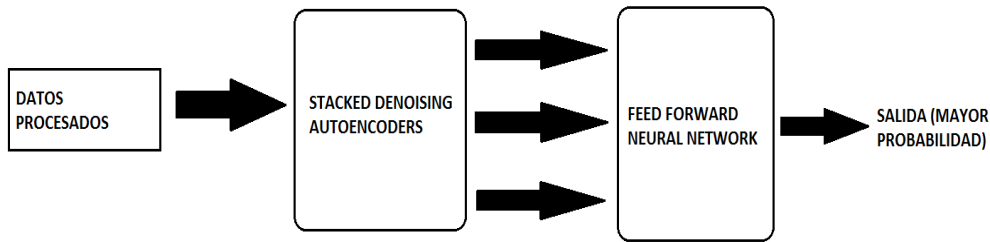


Figura 3. Esquema genérico Stacked Denoising Autoencoders más Perceptrón Multicapa

A la salida del perceptrón multicapa se tiene una o unas determinadas salidas y lo que hará la red es elegir la más probable de ellas. En la siguiente figura [8] se puede ver cómo es esta arquitectura.

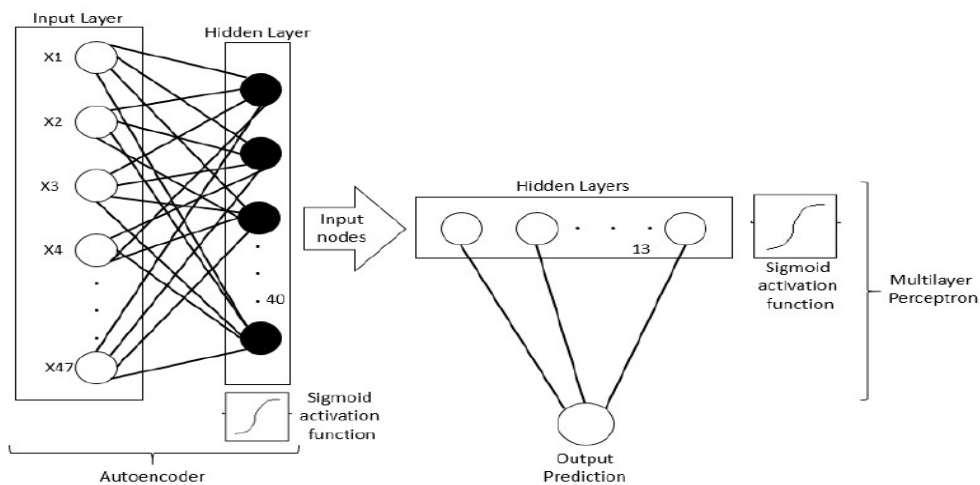


Figura 4. Una sola capa de autoencoder y un MLP con dos capas [8]

Para cada tipo de red neuronal usada se han tomado una serie de parámetros necesarios para especificar el entrenamiento de la red: función de activación, ruido enmascarado, learning rate, número de iteraciones, número de capas de la red y número de neuronas en cada capa. Para los autoencoders más el perceptrón multicapa se han utilizado los siguientes valores:

Función de activación autoencoders	sigmoide
Learning rate autoencoders	De 0.2 a 1
Ruido enmascarado	De 0 a 0.5
Número de iteraciones autoencoders	De 20 a 300
Tamaño de lote autoencoders	De 50 a 500
Número de capas autoencoders	De 2 a 4
Función de activación perceptrón	sigmoide
Learning rate perceptrón	De 0.2 a 1
Número de iteraciones perceptrón	De 15 a 125
Tamaño de lote perceptrón	De 50 a 500
Número de capas perceptrón	De 3 a 6

Tabla 2. Valores probados en la configuración de autoencoders más perceptrón multicapa

3.1.2. Arquitectura 2: Perceptrón Multicapa

La segunda configuración usada es básicamente la misma que la anterior quitando los autoencoders; es decir, el perceptrón multicapa no necesitará el entrenamiento por parte de una red de aprendizaje profundo previa.

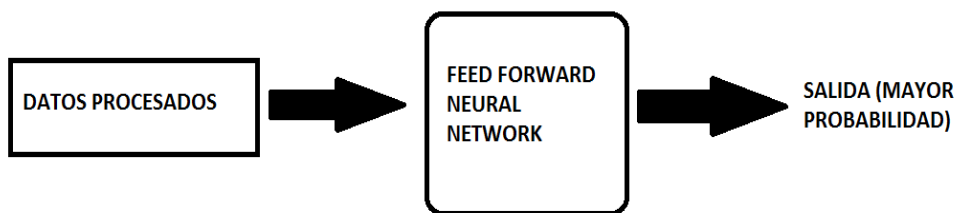


Figura 5. Esquema genérico con el perceptrón multicapa únicamente

En la siguiente tabla podemos ver los parámetros que hemos introducido a la red junto con las capas con las que se ha probado esta configuración.

Función de activación perceptrón	sigmoide
Learning rate perceptrón	De 0.2 a 1
Número de iteraciones perceptrón	De 15 a 125
Tamaño de lote perceptrón	De 50 a 500
Número de capas perceptrón	De 3 a 6

Tabla 3. Valores probados para la configuración del perceptrón multicapa solo

3.1.3. Arquitectura 3: Máquinas de Boltzmann con Deep Belief Network

Será la compuesta por las llamadas Máquinas de Boltzmann o en inglés Restricted Boltzmann Machines (RBM) [14] y una Deep Belief Network (DBN) [14]. En el siguiente esquema se puede ver el diagrama de bloques con esta configuración.

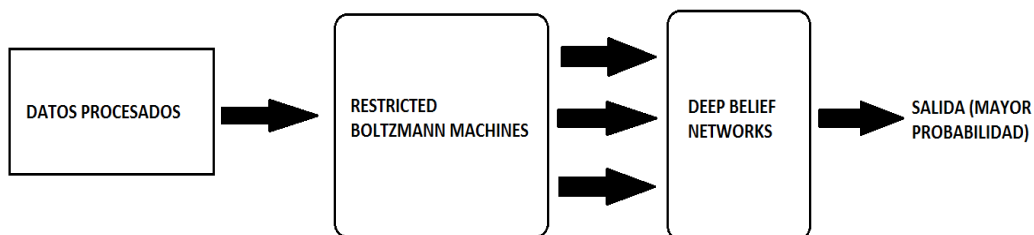


Figura 6. Esquema global Restricted Boltzmann Machines con Deep Belief Networks

En nuestra red, debido a que la toolbox [15] lo hace así, primero se entrenarán las máquinas de Boltzmann, una vez entrenadas estas se entrenará la DBN sin la capa de salida, para posteriormente entrenar todo el sistema (incluida la capa de salida) y testarlo.

Los parámetros usados para esta topología han sido los siguientes:

Función de activación	sigmoide
Learning rate	1
Número de iteraciones RBM	De 10 a 25
Tamaño de lote RBM	De 50 a 200
Número de capas RBM	2 o 3
Número de iteraciones DBN	De 10 a 25
Tamaño de lote DBN	De 50 a 500
Número de capas DBN	De 3 a 5
Número de iteraciones DBN+salida	De 10 a 250
Tamaño de lote DBN+salida	De 50 a 200

Tabla 4. Valores probados para la configuración de Máquinas de Boltzmann con Deep Belief Network

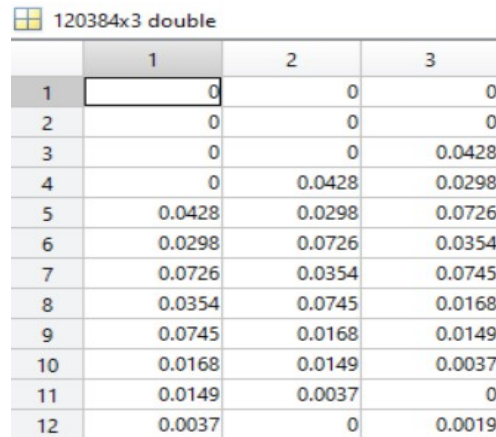
3.2. Variables de Entrada a la Red

Las redes neuronales se nutren de numerosos datos para su entrenamiento así que con las variables que se decidieron importantes, mencionadas en el apartado 2.1, se han hecho una serie de combinaciones para tener un gran número de datos y de información para intentar obtener los mejores resultados posibles.

A continuación vamos a nombrar más detalladamente las variables que se usan como entrada a la red neuronal, que están ordenadas por columnas en la matriz de entrada a la red como se describe más adelante. Después de haber aplicado el preproceso necesario con la fórmula (1), se han cogido las variables una a una para luego concatenarlas en la matriz de entrada ocupando las columnas que vienen entre paréntesis. Para consultar el código usado para obtener estas variables ver anexo IV.

- **Lluvia de los últimos 3 días (Columnas 1 a la 3)**

En cada una de las estaciones debemos de ir ordenando los datos de lluvia registrados de tal manera que en la columna 1 de nuestra matriz de entrada quede el dato de hace 3 días, en la columna 2 el dato de hace 2 días y en la columna 3 el dato de hace 1 día. De tal manera que el dato de hace 1 día cambia de columna en la siguiente fila, puesto que ya ha pasado a ser un dato de hace 2 días. Esto se puede ver mejor en la siguiente figura.



	1	2	3
1	0	0	0
2	0	0	0
3	0	0	0.0428
4	0	0.0428	0.0298
5	0.0428	0.0298	0.0726
6	0.0298	0.0726	0.0354
7	0.0726	0.0354	0.0745
8	0.0354	0.0745	0.0168
9	0.0745	0.0168	0.0149
10	0.0168	0.0149	0.0037
11	0.0149	0.0037	0
12	0.0037	0	0.0019

Figura 7. Parte de la matriz de lluvia de los últimos 3 días

- **Media lluvia últimos 5 días (Columna 4)**

En esta variable lo que se hace es la media de la lluvia caída en los 5 días anteriores. Es decir, se hace un promediado de los últimos 5 días para cada uno de los 10944 datos de cada estación, de ahí que se hayan dejado los primeros 5 datos de cada estación sin coger para el entrenamiento, porque se necesitaban para hacer estos promediados.

- **Temperatura máxima de los últimos 3 días (Columnas 5 a 7)**

Se ordenan de manera análoga a lo descrito en la lluvia de los 3 últimos días, se tienen los datos de las 11 estaciones en orden cada una de ellas con sus respectivas máximas en los 3 días anteriores.

- **Media de las temperaturas máximas de los últimos 5 días (Columna 8)**

Análogamente a lo hecho con la media de las lluvias, se hace el promedio de las temperaturas máximas de los 5 días anteriores para cada una de las estaciones.

- **Temperatura mínima de los últimos 3 días (Columnas 9 a 11)**

De la misma manera descrita con las máximas, se tienen ahora los valores normalizados de las mínimas de los 3 días anteriores, un día en cada columna, para cada estación.

- **Media de las temperaturas mínimas de los últimos 5 días (Columna 12)**

Mismo método descrito que con la media de las máximas de los últimos 3 días, se tiene en esta columna el promedio de las mínimas de los 5 últimos días para cada estación.

- **Diferencia entre temperatura máxima y mínima (Columna 13)**

En esta columna tenemos la diferencia entre la máxima y la mínima, primero se hace la resta entre los propios valores absolutos y luego se les aplica el preprocesado (1) para tener un 1 en la mayor amplitud térmica y un 0 en la menor.

- **Estado del cielo en los 3 últimos días (Columnas 14 a 16)**

Se tiene el estado del cielo en octas (valores del 0 al 8) de tal manera que el 0 está asignado a un cielo despejado y conforme el valor es más alto el cielo estará más nublado hasta que el 8 está asignado a un cielo totalmente cubierto. Cabe destacar que el valor que se tiene es el valor medio del día, ya que podría haber horas en los que el cielo esté cubierto y otras en los que esté despejado. Normalizando estos valores quedaría el 8 asignado a un valor de 1 para nuestra red, el código usado para tener en las 3 columnas estos datos es el mismo que para máximas, mínimas y lluvia de los últimos 3 días, ver anexo IV.

- **Media del estado del cielo en los 3 últimos días (Columna 17)**

Con las mismas consideraciones explicadas anteriormente para el estado del cielo, se hace un promedio con los datos de los 5 días anteriores del estado del cielo para cada estación al igual que se ha hecho con la lluvia, las máximas y las mínimas.

- **Mes del último dato de entrada (Columnas 18 a 29)**

Como se considera importante el mes en el que estamos puesto que el clima no es el mismo en verano que en invierno, se decide crear esta variable representada por 12 columnas, una para cada mes, estando ordenadas por meses. Por lo que finalmente en la matriz de entrada la columna 18 corresponde a enero, la 19 a febrero, la 20 a marzo y así sucesivamente hasta la 29 que será diciembre. Para una mejor idea de cómo está representado, en la siguiente matriz (2) vemos como es la representación ordenada para cada mes, desde la primera fila que sería enero hasta la última que sería diciembre.

$$\begin{pmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix} \quad (2)$$

- **Latitud y longitud (Columnas 30 y 31)**

En los modelos meteorológicos, ver anexo I, se considera importante introducir la localización geográfica del sitio del que se toman datos, además de delimitar el área geográfica tratada. En el caso de las estaciones que se usan es introducir las coordenadas indicadas en el apartado 2.2 aplicando el mismo preprocesado que en el resto de variables. Siendo en la fórmula (1) dato mínimo el de la longitud más occidental y la latitud más meridional y como dato máximo el de la longitud más oriental y la latitud más septentrional.

- **Altitud (Columna 32)**

Por la misma razón que para la latitud y longitud, se decidió introducir esta variable puesto que no es lo mismo el clima para diferentes altitudes.

Una vez ya con todas las variables tenemos la matriz de 120384x32 con las variables en las columnas que se han indicado entre paréntesis, de aquí se sacan los datos para entrenar la red y para testarla.

Como se explica en el capítulo 4 se van a hacer dos tipos de pruebas: una con datos de entrenamiento de todas las estaciones, y testeo también con datos de todas las estaciones, y otra con datos de entrenamiento de todas las estaciones menos una, la cual se utiliza para testear. Para la primera prueba se usan 104500 datos para entrenamiento y 15884 para testeo; para la segunda se usan 109440 datos para entrenamiento (los de 10 estaciones) y 10944 (los de la estación restante) para testear.

Cada arquitectura neuronal de las usadas en este proyecto, y cada prueba de las detalladas en el apartado 4.2 tendrá las mismas variables de entrada; es decir, una matriz de 104500x32 en un caso y una matriz de 109440x32 en otro caso.

Luego para testear se tendrán las matrices de 15884x32 en el caso de testeo con datos de todas las estaciones y de 10944x32 para el testeo con los datos de una estación.

- **Salida de las redes**

Además de la entrada, también las salidas de las redes tienen que estar tratadas adecuadamente. La capa de salida de todas las redes neuronales va a estar formada por dos neuronas de salida, que representan; una, si se supera el umbral determinado y la otra si no se supera; es decir, cada una dará una probabilidad y la que sea mayor, o lo que es lo mismo, la que supere el 50 %, será la salida decidida por la red. Luego con la matriz de testeo de salida se comprueba si ha acertado en la decisión. Cabe destacar que por la configuración de la toolbox elegida se ha tenido que optar por una salida con dos neuronas en vez de una con una neurona que hubiera sido más cómoda.

Para el entrenamiento de las redes neuronales se han usado, como se ha mencionado, 104500 datos para un caso y 109440 para otro, al tener dos neuronas de salida, necesitamos que las matrices de salida sean de dos columnas. De tal manera que se tendrá en cada columna el resultado de la predicción, representado por un 1 en la primera columna si se supera el umbral o un 0 en la segunda columna si no se supera el umbral.

La generación de las matrices de testeo se hace de manera análoga, de tal forma que se tendrán dependiendo de la prueba que hagamos, una matriz de 15884x2 para testeo con todas las estaciones y una matriz de 10944x2 para testeo con una estación.

Capítulo 4

Análisis de los Resultados

En este capítulo se detallan las pruebas hechas, los resultados obtenidos y cuáles son los que más pueden interesar comparando unos resultados con otros.

4.1. Entrenamiento y Validación de la Red Neuronal

Como se ha mencionado ya en el apartado 2.1 tenemos un total de 10944 datos de cada estación para introducir de cada una de las 32 variables de entrada, esto hace que tengamos entre las 11 estaciones 120384 datos de cada variable.

Las redes neuronales necesitan un entrenamiento previo para asimilar y aprenderse los datos, y para ello se usan la mayoría de datos que tenemos. Por otro lado, una parte de los datos se tiene que dejar para el testeo, para comprobar que el funcionamiento de la red es el adecuado. De ahí que como se ha indicado en el apartado 3.2 se tengan por un lado 104500 datos o 109400 datos para entrenar mientras que el resto en ambos casos se dejarán para testeo.

Lo que buscamos es la minimización del error de la red en la predicción meteorológica en cada configuración probada. Lo que se hace ahora es en cada arquitectura usada ver cómo evoluciona el error con el conjunto de iteraciones que hemos probado y compararlo con la función de validación para ver si nuestra red va a estar infradimensionada, bien dimensionada o sobredimensionada, que es el problema que se ha tenido aquí. A continuación podemos ver un ejemplo en una gráfica para un ejemplo de la tercera arquitectura para 40 iteraciones en el que se comprueba como el error cuadrático medio de la red y el error cuadrático medio de validación se quedan planos en la gráfica desde aproximadamente 10 iteraciones lo que hace que la red esté demasiado infradimensionada, no habiendo podido resolver este problema.

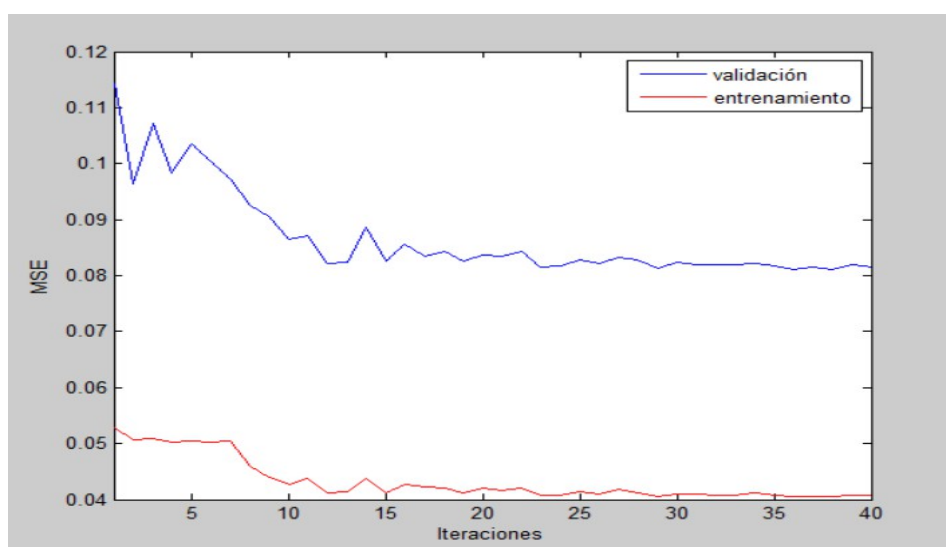


Figura 8. Comparación entre el error cuadrático medio de entrenamiento de la red y el de validación para la tercera arquitectura

En el caso de las otras arquitecturas neuronales ocurre exactamente lo mismo, así que lo que intentaremos buscar a partir de ahora es el menor error en la predicción posible probando diversas parametrizaciones de las arquitecturas neuronales propuestas.

4.2. Pruebas realizadas

Lo que pretendemos es tener el menor error en decisión de la red posible en cada configuración de red neuronal. La predicción se decidió que fuera entre 3 y 7 días puesto que en meteorología es el plazo en el que más puede interesar cómo funcionan las redes neuronales, ya que en un plazo menor las previsiones suelen acertar bastante como para intentar mejorar lo ya existente, y en un plazo mayor todavía queda mucho estudio aún con los modelos meteorológicos. Así pues, la red neuronal dependiendo de la probabilidad de salida, tomará una decisión sobre el umbral de temperatura que se esté probando para determinar si se supera o no ese umbral. A la salida de la red se tiene la probabilidad de que se supere y de que no se supere, la de mayor probabilidad será la elegida y se comprobará si la red ha acertado la previsión.

Debido a que en la base de datos proporcionada no tenemos dato de presión atmosférica, y el tenerlo para la predicción de lluvia es bastante importante, los resultados obtenidos para predecir lluvia no eran satisfactorios. Así que se optó solo por determinar la predicción para temperaturas tanto máximas como mínimas. Se eligieron unos umbrales en 0°C para máxima y mínima y 25°C para máxima. El umbral en 0°C para la temperatura mínima fue elegido porque es común un umbral para heladas nocturnas, el de 0°C para temperatura máxima puesto que es interesante para posibles olas de frío en el que esté helando durante todo el día algo que es común en Alemania en ciertas situaciones, y 25°C como umbral de temperatura máxima se eligió para determinar posibles episodios de calor ya que allí superar esa temperatura no es muy habitual. A continuación se puede observar más claramente las pruebas que se han decidido hacer.

- **Pruebas con datos de todas las estaciones**
 1. Temperatura mínima menor de 0°C
 2. Temperatura máxima menor de 0°C
 3. Temperatura máxima mayor de 25°C

- Pruebas con todos los datos de 10 estaciones y testeo en una.
 1. Temperatura mínima menor de 0°C
 2. Temperatura máxima menor de 0°C
 3. Temperatura máxima mayor de 25°C

Cada una de estas pruebas se ha hecho para las arquitecturas mencionadas en el apartado 3.1 entre los valores de los parámetros indicados en ese apartado también. El esquema global de las pruebas hechas en el proyecto es el que se muestra en la siguiente figura:

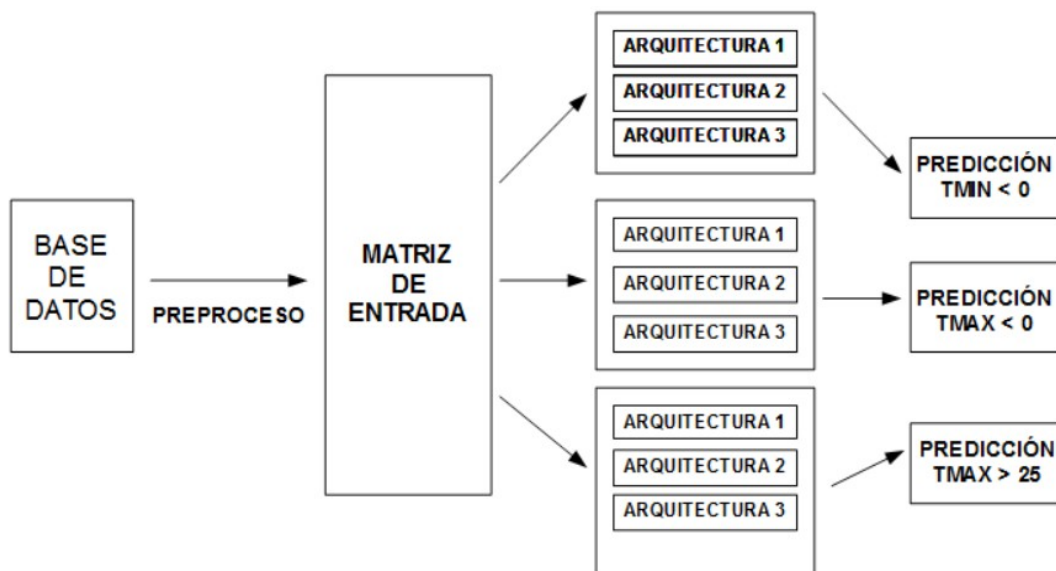


Figura 9. Esquema general del proyecto y las pruebas hechas

Siendo la matriz de entrada para las pruebas de datos con todas las estaciones de 104500x32 y de 109440x32 para las pruebas con testeo en una estación.

En cuanto a los datos testeados en una estación se ha elegido que de las 11 sea en la estación de Ratisbona, debido a que su altitud es intermedia con respecto a las estaciones que se tratan, y la zona en la que está hace que tenga un clima intermedio también en cuanto a temperatura entre las estaciones tratadas. Su localización la vemos en el siguiente mapa:

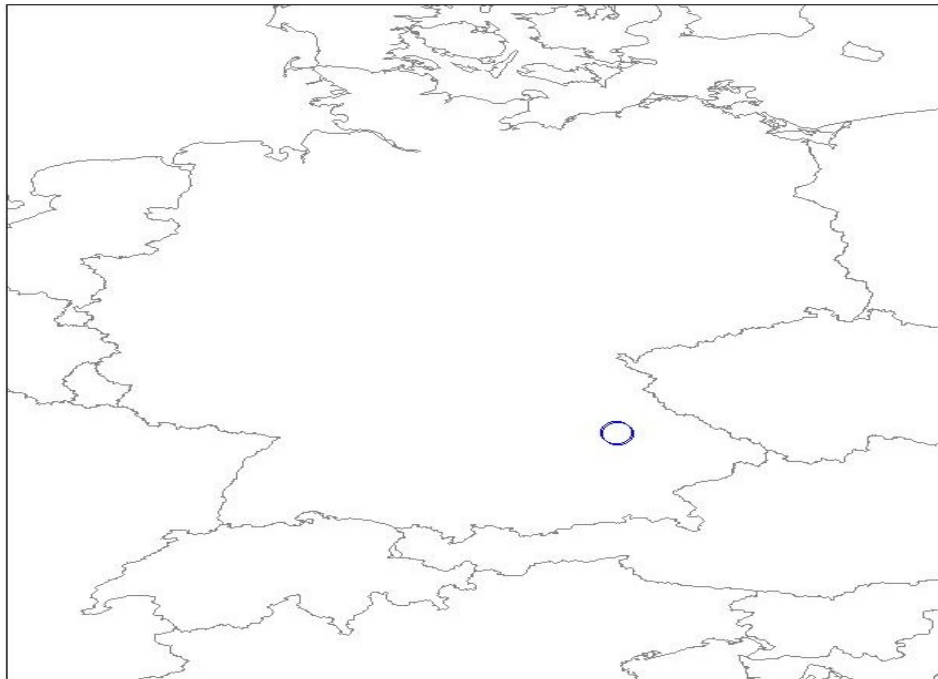


Figura 10. Localización de la estación de Ratisbona

4.3. Análisis de los Resultados

Una vez realizadas las pruebas correspondientes buscamos los mejores resultados de los que se han obtenido para cada arquitectura de red usada en el plazo entre 3 y 7 días, lo que quiere decir que se mostrarán los menores errores que se han obtenido para cada tipo de red.

4.3.1. Resultados para la Arquitectura 1

I. Testeo con datos de todas las estaciones

- Predicción para temperatura mínima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	11,83%	12,79%	13,36%	13,59%	13,61%
Error cuadrático medio en entrenamiento	0,087	0,093	0,099	0,097	0,097

Número de capas Autoencoders	2	2	2	2	2
Número de iteraciones Autoencoders	250	250	250	250	250
Número de capas Perceptrón Multicapa	5	5	5	5	5
Número de iteraciones Perceptrón Multicapa	70	70	70	70	70

Tabla 5. Resultados de la red para una mínima menor de 0°C

- **Predicción para temperatura máxima menor de 0°C**

	3 días	4 días	5 días	6 días	7 días
Error en decisión	6,47%	6,76%	6,90%	7,08%	7,25%
Error cuadrático medio en entrenamiento	0,051	0,053	0,055	0,057	0,057
Número de capas Autoencoders	2	2	2	2	2
Número de iteraciones Autoencoders	250	250	250	250	250
Número de capas Perceptrón Multicapa	6	6	6	5	5
Número de iteraciones Perceptrón Multicapa	75	75	75	75	75

Tabla 6. Resultados de la red para una máxima menor de 0°C

- Predicción para temperatura máxima mayor de 25°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	7,03%	7,10%	7,37%	9,36%	7,57%
Error cuadrático medio en entrenamiento	0,039	0,041	0,042	0,042	0,042
Número de capas Autoencoders	2	2	2	2	2
Número de iteraciones Autoencoders	250	250	250	250	250
Número de capas Perceptrón Multicapa	4	5	5	4	4
Número de iteraciones Perceptrón Multicapa	75	25	25	25	75

Tabla 7. Resultados de la red para una máxima menor de 25°C

II. Testeo con datos de la estación de Ratisbona

- Predicción para temperatura mínima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	14,99%	14,98%	15,41%	15,86%	15,95%
Error cuadrático medio en entrenamiento	0,086	0,091	0,093	0,095	0,096
Número de capas Autoencoders	2	2	2	2	2
Número de iteraciones Autoencoders	250	250	250	250	250
Número de capas Perceptrón Multicapa	4	4	4	4	4
Número de iteraciones Perceptrón Multicapa	75	75	75	75	75

Tabla 8. Resultados de la red para una mínima menor de 0°C

- Predicción para temperatura máxima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	7,29%	7,30%	7,33%	7,38%	7,48%
Error cuadrático medio en entrenamiento	0,051	0,053	0,053	0,056	0,057
Número de capas Autoencoders	2	2	2	2	2
Número de iteraciones Autoencoders	20	20	20	20	20
Número de capas Perceptrón Multicapa	4	4	4	4	4
Número de iteraciones Perceptrón Multicapa	75	75	75	75	75

Tabla 9. Resultados de la red para una máxima menor de 0°C

- Predicción para temperatura máxima mayor de 25°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	11,48%	11,56%	12,40%	12,61%	12,82%
Error cuadrático medio en entrenamiento	0,036	0,039	0,039	0,039	0,040
Número de capas Autoencoders	2	2	2	2	2
Número de iteraciones Autoencoders	250	20	20	20	250
Número de capas Perceptrón Multicapa	4	4	4	4	4
Número de iteraciones Perceptrón Multicapa	75	75	75	75	75

Tabla 10. Resultados de la red para una máxima menor de 25°C

Como se puede observar en las tablas, cuando no se tiene ningún dato de entrenamiento de la estación de testeo el resultado es peor. Aun con todo los resultados son buenos en ambos casos, sobre todo para predecir las máximas menores de 0°C.

4.3.2. Resultados para la Arquitectura 2

I. Testeo con datos de todas las estaciones

- Predicción para temperatura mínima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	11.92 %	12.79 %	13.37 %	13.66 %	13.23 %
Error cuadrático medio en entrenamiento	0,088	0,093	0,096	0,097	0,088
Número de capas Perceptrón Multicapa	5	5	5	5	5
Número de iteraciones	75	75	75	75	75

Tabla 11. Resultados de la red para una mínima menor de 0°C

- Predicción para temperatura máxima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	6.53 %	8.8 %	7,00%	7.08 %	9.9 %
Error cuadrático medio en entrenamiento	0,051	0,055	0,056	0,057	0,057
Número de capas Perceptrón Multicapa	5	5	5	5	5
Número de iteraciones	75	75	75	75	75

Tabla 12. Resultados de la red para una máxima menor de 0°C

- Predicción para temperatura máxima mayor de 25°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	7.09 %	7.34 %	7.68 %	9.35 %	7.59 %
Error cuadrático medio en entrenamiento	0,039	0,041	0,043	0,042	0,043
Número de capas Perceptrón Multicapa	4	4	4	4	4
Número de iteraciones	75	75	75	75	75

Tabla 13. Resultados de la red para una máxima menor de 25°C

II. Testeo con datos de la estación de Ratisbona

- Predicción para temperatura mínima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	14,63%	15,40%	15,96%	15,96%	16,76%
Error cuadrático medio en entrenamiento	0,087	0,091	0,094	0,095	0,098
Número de capas Perceptrón Multicapa	4	4	4	4	4
Número de iteraciones	75	75	75	75	75

Tabla 14. Resultados de la red para una mínima menor de 0°C

- **Predicción para temperatura máxima menor de 0°C**

	3 días	4 días	5 días	6 días	7 días
Error en decisión	6,97%	7,39%	7,40%	7,43%	7,50%
Error cuadrático medio en entrenamiento	0,051	0,054	0,055	0,056	0,057
Número de capas Perceptrón Multicapa	4	4	4	4	4
Número de iteraciones	75	75	75	75	75

Tabla 15. Resultados de la red para una máxima menor de 0°C

- **Predicción para temperatura máxima mayor de 25°C**

	3 días	4 días	5 días	6 días	7 días
Error en decisión	11,46%	12,22%	12,55%	12,68%	12,76%
Error cuadrático medio en entrenamiento	0,037	0,038	0,039	0,040	0,040
Número de capas Perceptrón Multicapa	4	4	4	4	4
Número de iteraciones	75	75	75	75	75

Tabla 16. Resultados de la red para una máxima menor de 25°C

Los resultados que se observan sin tener la parte de deep learning de los autoencoders vemos que son muy parecidos o algo peores a cuando la tenemos, aunque hay en algún caso en el que funciona mejor. Sin embargo, lo que sí parece que pueden introducir los autoencoders es una mayor linealidad en los resultados sin que haya diferencias tan grandes entre el error en la predicción para un día y por ejemplo el día siguiente a ese. Aunque esto se ve mejor al comparar unos resultados con otros en gráficas (apartado 4.3.4).

4.3.3. Resultados para la Arquitectura 3

Aquí, como se dijo en el apartado 3.1.3, la manera de entrenamiento con la que se trabaja es: primero entrenar las máquinas de Boltzmann, luego la DBN sin la capa de salida y luego toda la red entera. Por eso, en las siguientes tablas con los resultados hay 3 filas con las iteraciones que se han hecho en cada paso de la red.

I. Testeo con datos de todas las estaciones

- Predicción para temperatura mínima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	11.83 %	13.82 %	13.30 %	13.74 %	13.77 %
Error cuadrático medio en entrenamiento	0,087	0,093	0,094	0,097	0,099
Número de capas Máquinas de Boltzmann	2	2	2	2	2
Número de iteraciones Máquinas de Boltzmann	10	10	10	10	10
Número de capas DBN	4	4	4	4	4
Número de iteraciones DBN menos capa de salida	10	10	10	10	10
Número de iteraciones red entera	10	200	200	10	200

Tabla 17. Resultados de la red para una mínima menor de 0°C

- Predicción para temperatura máxima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	6.94 %	6.69 %	7.22 %	7.24 %	9.86 %
Error cuadrático medio en entrenamiento	0,057	0,057	0,058	0,059	0,057
Número de capas Máquinas de Boltzmann	2	2	2	2	2
Número de iteraciones Máquinas de Boltzmann	10	10	10	10	10
Número de capas DBN	4	4	4	4	4
Número de iteraciones DBN menos capa de salida	10	10	10	10	10
Número de iteraciones red entera	200	10	10	10	200

Tabla 18. Resultados de la red para una máxima menor de 0°C

- Predicción para temperatura máxima mayor de 25°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	7.08 %	7.25 %	7.51 %	7.53 %	7.37 %
Error cuadrático medio en entrenamiento	0,039	0,041	0,042	0,042	0,043
Número de capas Máquinas de Boltzmann	2	2	2	2	2
Número de iteraciones Máquinas de Boltzmann	10	10	10	10	10
Número de capas DBN	4	4	4	4	4
Número de iteraciones DBN menos capa de salida	10	10	10	10	10
Número de iteraciones red entera	200	200	200	200	200

Tabla 19. Resultados de la red para una máxima menor de 25°C

II. Testeo con datos de la estación de Ratisbona

- Predicción para temperatura mínima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	14,77%	15,49%	15,59%	15,90%	16,38%
Error cuadrático medio en entrenamiento	0,087	0,091	0,094	0,095	0,096
Número de capas Máquinas de Boltzmann	2	2	2	2	2
Número de iteraciones Máquinas de Boltzmann	10	10	10	10	10
Número de capas DBN	4	4	4	4	4
Número de iteraciones DBN menos capa de salida	10	10	10	10	10
Número de iteraciones red entera	250	10	200	10	200

Tabla 20. Resultados de la red para una mínima menor de 0°C

- Predicción para temperatura máxima menor de 0°C

	3 días	4 días	5 días	6 días	7 días
Error en decisión	7,16%	6,69%	7,22%	7,24%	7,48%
Error cuadrático medio en entrenamiento	0,051	0,056	0,058	0,059	0,057
Número de capas Máquinas de Boltzmann	2	2	2	2	2
Número de iteraciones Máquinas de Boltzmann	10	10	10	10	10
Número de capas DBN	4	4	4	4	4
Número de iteraciones DBN menos capa de salida	10	10	10	10	10
Número de iteraciones red entera	200	10	10	10	200

Tabla 21. Resultados de la red para una máxima menor de 0°C

- **Predicción para temperatura máxima mayor de 25°C**

	3 días	4 días	5 días	6 días	7 días
Error en decisión	11,52%	12,29%	12,61%	12,69%	12,55%
Error cuadrático medio en entrenamiento	0,036	0,038	0,041	0,040	0,040
Número de capas Máquinas de Boltzmann	2	2	2	2	2
Número de iteraciones Máquinas de Boltzmann	10	10	10	10	10
Número de capas DBN	4	4	4	4	4
Número de iteraciones DBN menos capa de salida	10	10	10	10	10
Número de iteraciones red entera	200	200	10	200	200

Tabla 22. Resultados de la red para una máxima menor de 25°C

Con esta configuración también se puede ver que a pesar de que hay cierta linealidad en cuanto a que en los días más cercanos el error que se comete es menor que a más plazo, también hay más fluctuaciones que en el caso de la arquitectura 1.

A continuación se van a comparar algunos de los resultados entre las 3 arquitecturas para poder observar mejor las diferencias que hay entre una y otra y cuál puede ser mejor para predecir a un determinado plazo.

4.3.4. Comparación y Discusión de los Resultados

Anteriormente se han mostrado los resultados para cada arquitectura de red usado, ahora se van a comparar aquellos que se han considerado que mejor funcionan y en el que mejor se puede apreciar las diferencias entre cada tipo de arquitectura.

La primera figura que vemos es la de predicción de temperatura menor que 0°C y testeó con todos los resultados.

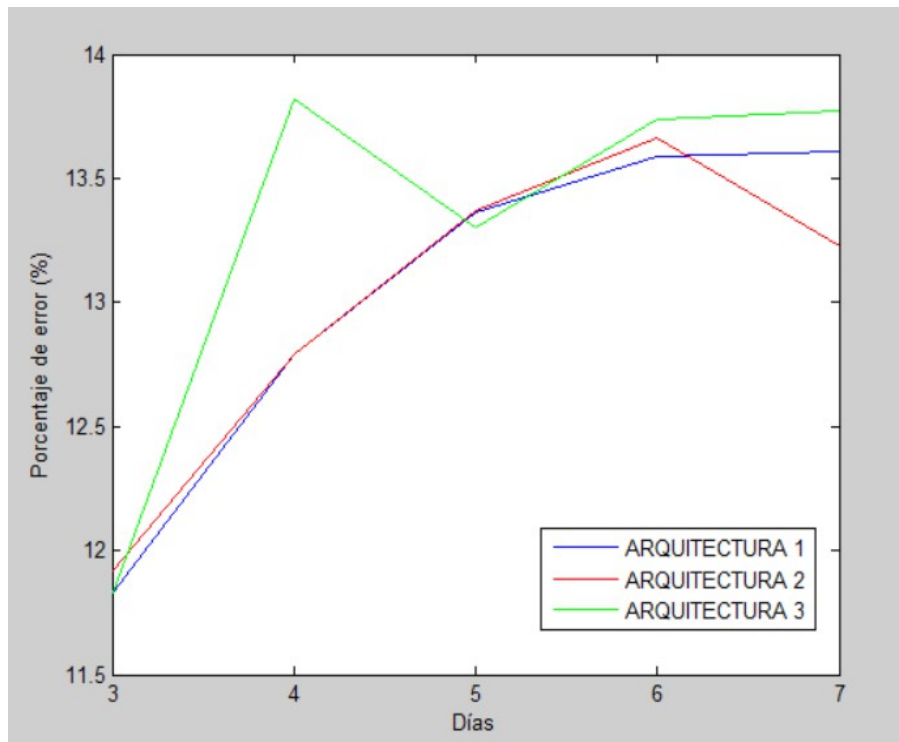


Figura 11. Comparación para la predicción de mínima menor de 0°C y testeó con todas las estaciones.

Vemos como para predecir mínimas inferiores a 0°C la arquitectura 3 no parece muy conveniente que se use puesto que muestra menor linealidad en el error en la predicción, mientras que si usamos las otras dos configuraciones el error sigue una tendencia más lineal. En la predicción a 7 días el que baje el error para la arquitectura número 2 puede ser indicativo de que no funciona tan bien ya a un plazo más largo, puesto que el error debería seguir aumentando conforme avanza el tiempo.

Ahora se muestra la predicción para determinar que la temperatura mínima sea menor de 0°C pero esta vez con testeó en la estación de Ratisbona para observar diferencias con la manera anterior de testeó.

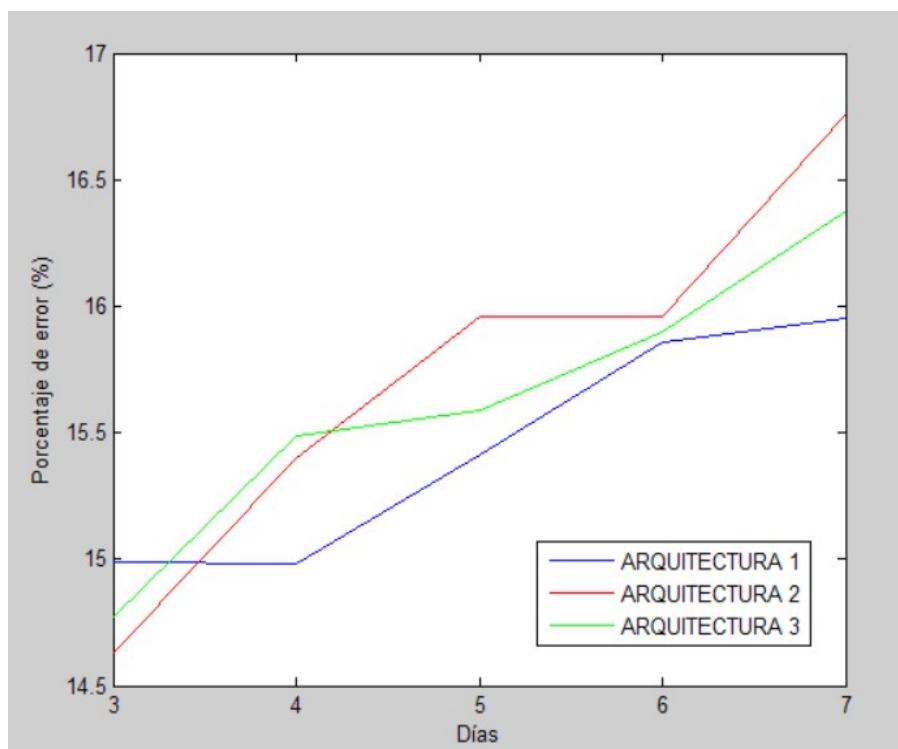


Figura 12. Comparación para la predicción de mínima menor de 0°C y testeo con una única estación

Una diferencia que se puede observar es que el error es mayor, aproximadamente un 3 % mayor que en un testeo con todas las estaciones. Esto puede ser normal puesto que la red no tiene datos de entrenamiento de la estación en la que se va a probar, con lo que a pesar de que pueda tener un clima parecido a otras estaciones no va a ser lo mismo que si tuviera datos propiamente de ella. Luego vemos que las 3 configuraciones muestran linealidad en cuanto al aumento del porcentaje de error en el tiempo y se puede determinar al igual que antes, que a pesar de haber poca diferencia entre ellas la mejor configuración parece que vuelve a ser la primera arquitectura neuronal.

Otra comparación que es interesante por ver cómo el error aumenta en función del tiempo para todos los modelos neuronales elegidos es la de predicción para una temperatura máxima mayor de 25°C testeando con los datos de la estación de Ratisbona.

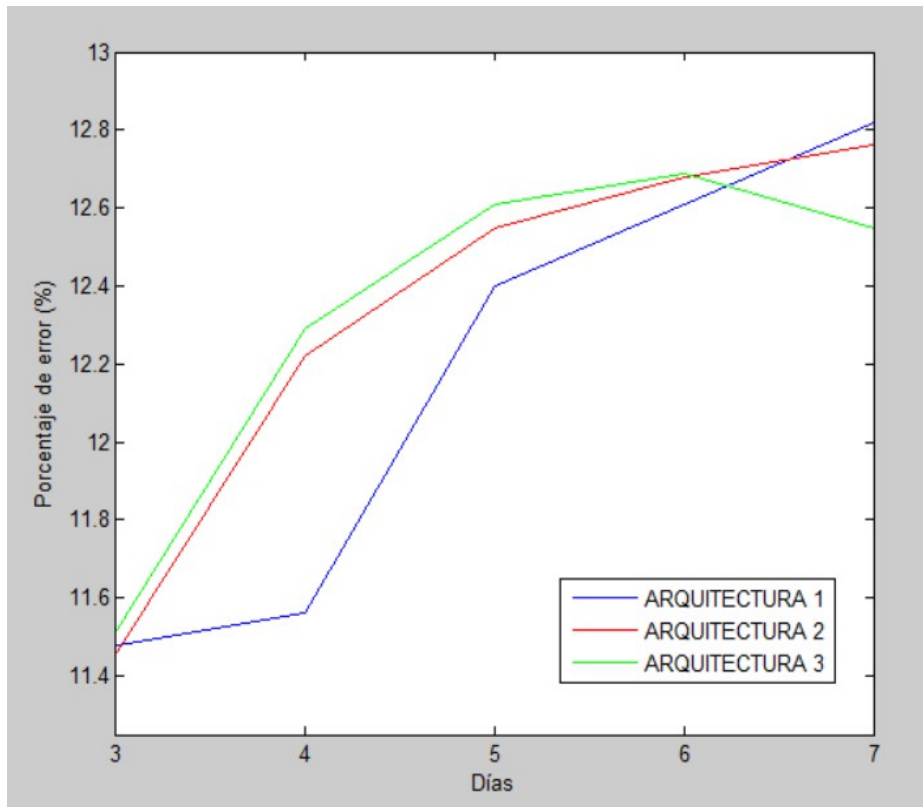


Figura 13. Comparación para la predicción de máxima mayor de 25°C y testeo con una única estación

De nuevo tenemos que la arquitectura neuronal 1 es la que menor error tendría aunque esta vez no sigue la linealidad de los casos mostrados anteriormente. Aquí las otras arquitecturas neuronales tienen un comportamiento muy parecido.

En general se podría decir que la arquitectura neuronal que funciona ligeramente mejor es la primera, la de autoencoders con perceptrón multicapa, tanto por ser generalmente el que menos error tiene, como por ofrecer una mayor linealidad en el error conforme nos alejamos en el plazo de predicción, lo que quiere decir que introducir Deep Learning sí que puede ser interesante para realizar estas predicciones.

Capítulo 5

Conclusiones

En este capítulo se habla de si se han cumplido los objetivos que se establecieron al principio del proyecto, las mayores dificultades que ha habido y se abordará cuáles pueden ser las mejoras y ampliaciones para un futuro.

5.1. Conclusiones

Tras terminar este proyecto se puede determinar que la introducción de técnicas de Deep Learning en la predicción meteorológica puede servir de ayuda para mejorar las previsiones, al menos la introducción de Autoencoders aporta una leve mejoría. Cabe destacar que como se ha comentado, la base de datos aquí es limitada en cuanto a algunas variables y en cuanto a no tener varios datos diarios. Esto ha llevado a no poder hacer algo que en un principio se quería haber hecho y es también predecir eventos meteorológicos de precipitación. Pero se tuvo que desistir al no obtener buenos resultados. Porque con un mayor número de datos de entrada probablemente los resultados hubieran sido mejores en las tres arquitecturas neuronales probadas.

Otro problema encontrado es que la mayoría de fallos en la red se dan en cambios bruscos de temperatura de un día a otro, en ese aspecto la red tarda más tiempo en aprenderse esas transiciones abruptas.

Otra conclusión es que los modelos generados para la segunda prueba, prediciendo resultados de una estación que no entrenó a la red, han resultado tener peor error, como se esperaba, pero no tanto como para descartar la generación de un modelo capaz de predecir el resultado en cualquier punto del territorio para el que se tengan medidas de varios días. Disponiendo de una red de estaciones más densa sería de esperar una considerable mejora en este tipo de modelo.

Con todo, la componente de Deep Learning que tienen las arquitecturas neuronales usadas en este proyecto no es mucha, ya que se usan pocas capas e interesaría usar más para comprobar mejor si su uso aporta más mejora, aquí con la introducción de más capas el error aumentaba. Esto es indicativo de que seguir aumentando la profundidad de la red lleva a su sobredimensionamiento y que la representación generada por la red se sobreajusta a los datos de entrenamiento (los memoriza) en lugar de generar representaciones que generalizan bien. Si se quiere explotar un modelo deep learning con muchas capas se hace necesario disponer de millones de datos y preferiblemente cientos de estaciones, de forma que la memorización de los datos no sea posible.

5.2. Líneas Futuras

El proyecto realizado deja abierta para un futuro la posibilidad de seguir con su desarrollo introduciendo bases de datos más completas para poder comprobar su funcionalidad en la predicción con otras variables como pudieran ser la lluvia o el viento e introducir mayor número de umbrales. Además se podría introducir en un futuro datos de reanálisis de modelos globales de circulación general en las coordenadas que tengan los nodos de su rejilla para intentar hacer un modelo de área limitada pero con una predicción a un plazo mayor de lo que lo hacen estos.

Otra posible utilización de cara al futuro sería cambiando la región geográfica. Aquí se han tratado datos de estaciones de Alemania, pero se podría comprobar si este modelo también tiene un funcionamiento adecuado en España dónde la variedad climática es mayor. Se podrían introducir datos históricos de estaciones repartidas por todo el país y ver su comportamiento tanto con el número de iteraciones y de capas usadas aquí como con otras distintas. También se podría seguir trabajando sin cambiar la zona geográfica, simplemente añadiendo más estaciones de Alemania, de ahí nuevamente la importancia de tener una red amplia de observatorios para poder tratar sus datos.

6. Bibliografía

- [1] <http://www.meteo.unican.es>
- [2] <http://www.value-cost.eu/>
- [3] <http://euro-cordex.net/>
- [4] CASANUEVA, Anna, et al. Daily precipitation statistics in a EURO-CORDEX RCM ensemble: added value of raw and bias-corrected high-resolution simulations. *Climate dynamics*, 2016, vol. 47, no 3-4, p. 719-737.
- [5] ANCELL TRUEBA, Rafael, et al. Aportaciones de las redes bayesianas en meteorología. Predicción probabilística de precipitación. 2013.
- [6] COFIÑO, D. Técnicas estadísticas y neuronales de agrupamiento adaptativo para la predicción probabilística de fenómenos meteorológicos locales. Aplicación en el corto plazo y en la predicción estacional. En *Aplicación en el corto plazo y en la predicción estacional*. Universidad de Cantabria, 2004.
- [7] CHEN, Junfei; JIN, Qiongji; CHAO, Jing. Design of deep belief networks for short-term prediction of drought index using data in the Huaihe river basin. *Mathematical Problems in Engineering*, 2012, vol. 2012.
- [8] PALANCA, Javier; DUQUE, Néstor. Rainfall Prediction: A Deep Learning Approach. En *Hybrid Artificial Intelligent Systems: 11th International Conference, HAIS 2016, Seville, Spain, April 18-20, 2016, Proceedings*. Springer, 2016. p. 151.
- [9] LIU, James NK, et al. Deep neural network based feature representation for weather forecasting. En *Proceedings on the International Conference on Artificial Intelligence (ICAI)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2014. p. 1.
- [10] HOSSAIN, Moinul, et al. Forecasting the weather of Nevada: A deep learning approach. En *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015. p. 1-6.

- [11] HU, Qinghua; ZHANG, Rujia; ZHOU, Yucan. Transfer learning for short-term wind speed prediction with deep neural networks. *Renewable Energy*, 2016, vol. 85, p. 83-95.
- [12] <http://www.meteo.unican.es/es/software/meteolab>
- [13] <http://www.guiadealemania.com/mapa-de-alemania/>
- [14] {Goodfellow-et-al-2016, title={Deep Learning}, author={Ian Goodfellow and Yoshua Bengio and Aaron Courville}, publisher={MIT Press}, note={\url{http://www.deeplearningbook.org}}, year={2016}}
- [15] <https://github.com/rasmusbergpalm/DeepLearnToolbox>
PALM, Rasmus Berg. Prediction as a candidate for learning deep hierarchical models of data. *Technical University of Denmark*, 2012, vol. 5.
- [16] <https://www.ecmwf.int/>
- [17] <https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forecast-system-gfs>
- [18] http://www.cpc.ncep.noaa.gov/products/international/browse_graphics/gfs_37.5km_24h_fcst.png
- [19] <http://www.ecmwf.int/en/about/media-centre/news/2011/new-seasonal-forecasting-system-operation>
- [20] <http://hirlam.org/>
- [21] <https://www.tethys.cat/antics/num02/articles/art0205esp.htm>
- [22] <http://www2.mmm.ucar.edu/mm5/>
- [23] http://www.aemet.es/es/idi/prediccion/prediccion_numerica
- [24] <http://www.meteofrance.fr/prevoir-le-temps/la-prevision-du-temps/le-modele-a-maille-fine-arome>
- [25] https://es.wikipedia.org/wiki/Red_neuronal_artificial
- [26] Apuntes de la asignatura de Neurocomputación. Universidad de Zaragoza. Julio David Buldain Pérez.

- [27] IRIGOYEN MANCHO, Miguel, et al. Reconocimiento de patrones meteorológicos mediante técnicas neuronales y estadísticas. 2011.
- [28] https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/x38.html
- [29] LARRANAGA, Pedro; INZA, Inaki; MOUJAHID, Abdelmalik. Tema 8. Redes Neuronales. *Redes Neuronales, U. del P. Vasco*, 1997, p. 12-17.
- [30] https://metacademy.org/roadmaps/rgrosse/deep_learning/version/22
- [31] <http://www.ecad.eu/>