

Iván Fernández Navarro

NIA: 100383564

Carlos García Corral

NIA: 100383380

Grupo 84

Práctica 2 - Parte 1:

Problema de clasificación con el Perceptrón Multicapa

Redes de Neuronas Artificiales

ÍNDICE:

1.	Introducción	2
2.	Preproceso de los datos	2
3.	Modelo Perceptrón multicapa	3
3.1.	Experimentación realizada	3
3.2.	Resultados obtenidos	3
3.3.	Análisis de los resultados	6
4.	Conclusiones	6

1. Introducción

En esta práctica comprobaremos la eficacia del modelo no lineal Perceptrón Multicapa a la hora de clasificar el tipo de cielo a partir de imágenes tomadas por una cámara. La resolución de este problema ocupa un lugar importante en la estimación de la radiación solar, ya que la cantidad de radiación depende de si hay nubes o no, o incluso del tipo de nubes.

Existen tres tipos de cielos: Cielo Despejado, Un solo tipo de nube (Nube) y Varios tipos de nube (Multinube).

Cada tipo de cielo tendrá 12 atributos que lo identifiquen:

- Media del canal azul
- Media del canal rojo
- Desviación típica del canal azul
- Sesgo del canal azul
- Diferencia de medias:
 - Rojo – Verde
 - Rojo – Azul
 - Verde - Azul
- Entropía del canal azul
- Energía del canal azul
- Contraste del canal azul
- Homogeneidad del canal azul
- Cobertura

Usando esos 12 valores, nuestro modelo deberá intentar predecir el tipo de cielo que se muestra en la imagen, habiendo las tres posibilidades comentadas anteriormente.

2. Preproceso de los datos

Antes de entrenar y poner a prueba los modelos de regresión, es importante tratar los datos de forma que aumente la exactitud de las predicciones.

Es recomendable normalizar los inputs de manera que sus valores estén en el rango de 0 a 1 para cada atributo. Es decir, en cada atributo, el valor máximo será 1 y el mínimo 0, y el resto de valores tendrán un valor proporcional a estos.

Otro paso del preproceso es la aleatorización de los datos, es decir, cambiar el orden aleatoriamente de los datos, para evitar posibles patrones que se encuentren según la colocación de los datos.

El último paso consiste en la separación de los datos mediante el método de Validación Cruzada con 4 folds.

Se divide el conjunto total de datos en 4 partes iguales (P1, P2, P3 Y P4) de forma que cada parte conste del siguiente número de instancias de cada clase:

- Clase 1 (Cielo Despejado) → 48 instancias
- Clase 2 (Multinube) → 156 instancias
- Clase 3 (Nube) → 513 instancias

Una vez se han separado las diferentes partes se combinarán dichas partes de tal forma que se tengan 4 parejas de ficheros formados por dos datasets: entrenamiento y testing.

Las combinaciones serán las siguientes:

- P1 como conjunto de test y (P2+P3+P4) como conjunto de entrenamiento
- P2 como conjunto de test y (P1+P3+P4) como conjunto de entrenamiento
- P3 como conjunto de test y (P1+P2+P4) como conjunto de entrenamiento
- P4 como conjunto de test y (P1+P2+P3) como conjunto de entrenamiento

Con estas combinaciones se realizarán una serie de experimentos para comprobar la eficacia de los distintos modelos a crear.

3. Modelo Perceptrón Multicapa

3.1. Experimentación realizada

Para ver qué parámetros resultan mejor para el modelo de perceptrón multicapa, se va a ir experimentando con el número de capas ocultas y sus respectivos nodos, y el ratio de aprendizaje.

El número máximo de ciclos quedará constante, siendo así este parámetro igual a 10,000 ciclos.

En cuanto al ratio de aprendizaje analizaremos los valores 0.01, 0.1 y 0.5. De esta manera, tendremos 3 ratios de aprendizaje con los que experimentar. Se han elegido estos valores porque se ha querido comprobar si con valores pequeños (de 2 decimales) y más grandes (de 1 decimal), el error mínimo mejoraba o empeoraba, pudiendo así hacer comparaciones.

Si hablamos sobre las capas ocultas y sus nodos, se ha querido hacer experimentos con 2 y 3 capas ocultas cambiando la distribución de nodos. Se han escogido como nodos los valores 10, 15, 20, 30 y algunas de sus diferentes combinaciones (15-30, 10-20-10, 10-30-10, 30-15-30). De esta manera, tendremos 4 topologías diferentes con cada una de las parejas formadas en el [punto 2](#).

Como se ha experimentado con 48 modelos diferentes (12 por cada pareja), y cada uno se ejecuta con 10,000 ciclos, se ha modificado el código del *script* de tal forma que se automatice la experimentación mediante bucles.

Los atributos *topología* y *razón* se han convertido en listas, en las que se inicializarán las 4 topologías y los 3 ratios de aprendizaje. Mediante el bucle se crearán los 48 modelos combinando las topologías y los ratios.

3.2. Resultados obtenidos

Train / Test	# neuronas ocultas	Razón de aprendizaje	# ciclos	Error training	% aciertos training	Error testing	% aciertos testing
Parejas1	15-30	0.01	10000	0,18	0,90	0,39	0,77
Parejas1	15-30	0.1	10000	0,03	0,98	0,50	0,72

Parejas1	15-30	0.5	10000	0,02	0,99	0,58	0,69
Parejas1	10-20-10	0.01	10000	0,25	0,83	0,34	0,79
Parejas1	10-20-10	0.1	10000	0,04	0,98	0,53	0,71
Parejas1	10-20-10	0.5	10000	0,25	0,80	0,44	0,74
Parejas1	10-30-10	0.01	10000	0,22	0,85	0,38	0,76
Parejas1	10-30-10	0.1	10000	0,01	0,99	0,55	0,70
Parejas1	10-30-10	0.5	10000	0,28	0,80	0,44	0,73
Parejas1	30-15-30	0.01	10000	0,18	0,88	0,41	0,77
Parejas1	30-15-30	0.1	10000	0,04	0,98	0,53	0,73
Parejas1	30-15-30	0.5	10000	0,01	0,99	0,49	0,49
Parejas2	15-30	0.01	10000	0,08	0,87	0,39	0,78
Parejas2	15-30	0.1	10000	0,04	0,98	0,45	0,76
Parejas2	15-30	0.5	10000	0,17	0,87	0,33	0,74
Parejas2	10-20-10	0.01	10000	0,27	0,83	0,40	0,71
Parejas2	10-20-10	0.1	10000	0,05	0,98	0,56	0,69
Parejas2	10-20-10	0.5	10000	0,21	0,87	0,43	0,71
Parejas2	10-30-10	0.01	10000	0,26	0,84	0,37	0,75
Parejas2	10-30-10	0.1	10000	0,02	0,99	0,47	0,76
Parejas2	10-30-10	0.5	10000	0,19	0,86	0,51	0,69
Parejas2	30-15-30	0.01	10000	0,21	0,85	0,39	0,74
Parejas2	30-15-30	0.1	10000	0,02	0,99	0,49	0,74
Parejas2	30-15-30	0.5	10000	0,03	0,98	0,53	0,69
Parejas3	15-30	0.01	10000	0,21	0,88	0,36	0,76
Parejas3	15-30	0.1	10000	0,05	0,97	0,53	0,72
Parejas3	15-30	0.5	10000	0,16	0,89	0,42	0,73
Parejas3	10-20-10	0.01	10000	0,27	0,83	0,33	0,77
Parejas3	10-20-10	0.1	10000	0,16	0,83	0,42	0,74
Parejas3	10-20-10	0.5	10000	0,30	0,77	0,38	0,74
Parejas3	10-30-10	0.01	10000	0,25	0,81	0,39	0,73
Parejas3	10-30-10	0.1	10000	0,10	0,95	0,43	0,75
Parejas3	10-30-10	0.5	10000	0,26	0,83	0,36	0,73
Parejas3	30-15-30	0.01	10000	0,21	0,87	0,37	0,75
Parejas3	30-15-30	0.1	10000	0,01	0,99	0,54	0,71
Parejas3	30-15-30	0.5	10000	0,27	0,87	0,38	0,75
Parejas4	15-30	0.01	10000	0,19	0,87	0,38	0,74
Parejas4	15-30	0.1	10000	0,02	0,99	0,46	0,74
Parejas4	15-30	0.5	10000	0,02	0,98	0,62	0,67

Parejas4	10-20-10	0.01	10000	0,27	0,81	0,37	0,74
Parejas4	10-20-10	0.1	10000	0,12	0,95	0,52	0,71
Parejas4	10-20-10	0.5	10000	0,16	0,85	0,42	0,74
Parejas4	10-30-10	0.01	10000	0,24	0,83	0,44	0,70
Parejas4	10-30-10	0.1	10000	0,06	0,98	0,48	0,73
Parejas4	10-30-10	0.5	10000	0,16	0,86	0,38	0,77
Parejas4	30-15-30	0.01	10000	0,21	0,86	0,48	0,72
Parejas4	30-15-30	0.1	10000	0,01	0,99	0,59	0,67
Parejas4	30-15-30	0.5	10000	0,00	1,00	0,62	0,68

Min error training 1	0,00
Min error testing 1	0,33
Max aciertos training 1	1,00
Max aciertos testing 1	0,79

Min error training 2	0,01
Min error testing 2	0,33
Max aciertos training 2	0,99
Max aciertos testing 2	0,78

Min error training 3	0,01
Min error testing 3	0,34
Max aciertos training 3	0,99
Max aciertos testing 3	0,77

Min error training 4	0,01
Min error testing 4	0,36
Max aciertos training 4	0,99
Max aciertos testing 4	0,77

Min error training 5	0,01
Min error testing 5	0,36
Max aciertos training 5	0,99
Max aciertos testing 5	0,77

Como podemos observar se han subrayado los 5 mejores resultados por colores.

La mayoría de los mejores resultados se han obtenido con el modelo generado con 3 capas ocultas de 30, 15 y 30 nodos respectivamente. Aunque el ratio de aprendizaje del mejor modelo varía entre 0.1 y 0.5, escogeremos 0.1 porque balanceando parece que es el que mejor resultados da tanto en entrenamiento como en testing.

3.3. *Análisis de los resultados*

Como se ha comentado anteriormente el modelo escogido es el generado con 3 capas ocultas de 30, 15 y 30 nodos respectivamente, y un ratio de aprendizaje de 0.1.

Analizar nuestro modelo a base de gráficas es muy difícil, puesto que hay 2151 patrones (717 instancias por cada fold y 3 folds por cada set de entrenamiento), y por cada uno de ellos 2 salidas (obtenida y deseada). Por tanto, es mejor fijarse en los datos numéricos que en gráficas.

Los datos numéricos a evaluar serán las evoluciones de los errores de entrenamiento y test a lo largo del aprendizaje.

Por razón de espacio y de simplicidad, se muestra la evolución del error de entrenamiento y de testing de los ciclos 1, 5000 y 10000.

Train / Test	# neuronas ocultas	Razón de aprendizaje	Error training ciclo 1	Error training ciclo 5000	Error training ciclo 10000	% aciertos training	Error testing ciclo 1	Error testing ciclo 5000	Error testing ciclo 10000	% aciertos testing
Parejas1	15-30-15	0.5	0.45022	0.03908	0.03905	0,90	0.43734	0.52296	0.52765	0,77
Parejas2	15-30-15	0.5	0.44923	0.04504	0.02055	0,98	0.43740	0.47258	0.48987	0,72
Parejas3	15-30-15	0.5	0.44989	0.02312	0.01136	0,99	0.43735	0.54826	0.53563	0,69
Parejas4	15-30-15	0.5	0.45156	0.02939	0.01307	0,83	0.44002	0.51435	0.59259	0,79

Como podemos observar los errores de entrenamiento a medida que aumentamos de ciclo disminuyen considerablemente, en cambio, pasa todo lo contrario con los errores de testing. Esto es normal puesto que con el testing evaluamos nuevos datos que el modelo nunca ha visto, y en el entrenamiento se puede sobreajustar a los datos dados.

4. Conclusiones

Como conclusión podríamos decir que el perceptrón multicapa es un modelo que se ajusta bastante bien en el uso de Deep Learning, puesto que un porcentaje medio de 0.925 en aciertos de entrenamiento y un porcentaje medio de 0.7425 en aciertos de testing es muy buen resultado.

En la parte 2 de la práctica podremos comprobar si las redes convolucionales dan mejor resultado que el perceptrón multicapa.

Práctica 2 - Parte 1:

Problema de clasificación con el Perceptrón Multicapa

