



Memoria de Redes Neuronales

FUNDAMENTOS DE LOS SISTEMAS INTELIGENTES

AARÓN HERNÁNDEZ ÁLVAREZ

JUAN JOSÉ BELLO SANTANA

Índice

| | | |
|-------------|---|----------|
| I. | Disposiciones previas..... | 2 |
| 1. | Objetivos del estudio | 2 |
| 2. | Preparación del entorno de trabajo | 3 |
| II. | Obtención del modelo óptimo..... | 4 |
| 1. | Resultados de pruebas preliminares | 4 |
| 2. | Reentrenando los modelos candidatos..... | 5 |
| 3. | Selección del candidato | 6 |
| III. | Analizando el modelo seleccionado | 7 |
| 1. | Comportamiento en primer entrenamiento | 7 |
| 2. | Comportamiento en segundo entrenamiento | 7 |
| 3. | Matriz de confusión del candidato seleccionado..... | 8 |
| IV. | Anexo | 9 |
| 1. | Funciones empleadas en el modelo | 9 |

I.Disposiciones previas

1. Objetivos del estudio

El objetivo de estudio de este informe se basa en la creación de una red neuronal la cual sea capaz de diferenciar imágenes de coches de diferentes colores. Para obtener las imágenes, se ha partido principalmente del dataset localizado en el siguiente enlace: https://ai.stanford.edu/~jkrause/cars/car_dataset.html

La red neuronal desarrollada será capaz de distinguir coches entre los siguientes colores: blanco, negro, azul, rojo y amarillo. Para ello, se ha utilizado para cada color un total de 80 imágenes destinadas al entrenamiento de la red y 20 imágenes pertenecientes al conjunto de validación de la misma.

De igual manera, se han creado diferentes modelos mediante la modificación de los hiperparámetros de la red, supresión y adición de capas convolutivas, destacando en el informe que se presenta el mejor de los todos los modelos propuestos.

2. Preparación del entorno de trabajo

Nomenclatura empleada para la denominación de los Modelos:

Para mejor comprensión de la arquitectura de los modelos creados, ésta viene definida en el propio nombre, de forma que tomará los parámetros especificados en el código base si éstos no se mencionan en éste.

Se indican en esta tabla los términos empleados y su abreviación:

| Capas | |
|---------------------|---|
| C | Capa convolutiva de neuronas |
| D | Capa Densa (Fully Connected) de neuronas |
| F | Capa de aplanamiento (Flatten) |
| Propiedades de Capa | |
| k | Tamaño de la muestra de cada neurona (kernel_size) |
| d | Porcentaje respecto de la unidad de neuronas desactivadas aleatoriamente. Ayuda a evitar los efectos de memorización en la red. |

Por ejemplo, para el siguiente nombre de archivo:

Modelo_C32k33_C64k33_d025_D128_d05_D5

Tendríamos que se trata de un modelo de dos capas convolutivas con muestras de 3x3 cada una, un dropout primario del 0,25, una capa densa de 128 y un dropout secundario del 0,5, antes de acabar en la capa final densa de 5 salidas (las 5 categorías).

II.Obtención del modelo óptimo

1. Resultados de pruebas preliminares

Para garantizar la equivalencia de todos los modelos probados se realizaron las construcciones colocando a todos un tiempo de entrenamiento total de 100 épocas, de forma que éste quede descartado como elemento de discreción.

Para un total de 100 épocas, los resultados comparativos respecto del factor de precisión han sido los siguientes:

| ID | Modelo | Pérdida | Precisión | Precisión (%) |
|----|--|---------|-----------|---------------|
| 1 | C16k44_C32k44_d025_D128_d05_D5 | 1.0156 | 0.6939 | 69,4% |
| 2 | C32k33_C64k33_C128k33_d025_D256_d05_D5 | 0.7476 | 0.7959 | 79,6% |
| 3 | C32k33_C64k33_d025_D128_d05_D5 | 0.5977 | 0.8265 | 82,7% |
| 4 | C32k33_C128k33_C256k33_d05_D256_d05_D5 | 0.7188 | 0.7755 | 77,6% |
| 5 | C32k44_C128k44_d075_D256_d05_D5 | 0.4994 | 0.8980 | 89,8% |
| 6 | C32k55_C64k55_C128k55_d05_D128_d05_D5 | 1.0721 | 0.7959 | 79,6% |
| 7 | C32k55_C64k55_d025_D128_d05_D5 | 0.4993 | 0.8878 | 88,9% |
| 8 | C64k33_C128k33_d025_D256_d05_D5 | 0.4283 | 0.8776 | 87,8% |

Mientras que si nos fijamos en la evolución de la pérdida se observa los siguientes resultados:

| ID | Modelo | Pérdida Inicial | Perdida Final |
|----|--|-----------------|---------------|
| 1 | C16k44_C32k44_d025_D128_d05_D5 | ~1.60 | 1.0156 |
| 2 | C32k33_C64k33_C128k33_d025_D256_d05_D5 | ~1.60 | 0.7476 |
| 3 | C32k33_C64k33_d025_D128_d05_D5 | ~1.60 | 0.5977 |
| 4 | C32k33_C128k33_C256k33_d05_D256_d05_D5 | ~1.60 | 0.7188 |
| 5 | C32k44_C128k44_d075_D256_d05_D5 | ~1.60 | 0.4994 |
| 6 | C32k55_C64k55_C128k55_d05_D128_d05_D5 | ~1.60 | 1.0721 |
| 7 | C32k55_C64k55_d025_D128_d05_D5 | ~1.60 | 0.4993 |
| 8 | C64k33_C128k33_d025_D256_d05_D5 | ~1.60 | 0.4283 |

Por ende, podemos concluir que los modelos 5,7,8 Disponen de una arquitectura mucho más adecuada para la función que se ha planteado para la práctica, así como para el dataset propuesto.

Sin embargo, se ha de destacar el curioso fenómeno que se ha dado en los modelos candidatos. Mientras que en el modelo 5, que es el que más precisión ofrece, la pérdida es de 0.4994, en el que peor precisión ofrece de los tres, el 8, la desviación existente en los valores reales y los predichos es la menor de todas, puntuando un 0.4283

Hay mencionar además que las funciones de ajuste y pérdida no han variado en ninguno de los casos, de forma que el factor diferencial es la arquitectura de cada modelo en sí, de manera que se garantiza que aquel modelo que salga seleccionado otorgará los mejores resultados usando los mismos recursos que los demás (ver anexo para más información sobre las funciones empleadas). Si es cierto que se han cambiado los valores por defecto de pérdida, para verificar el comportamiento de los prototipos con diferentes valores asignados.

2. Reentrenando los modelos candidatos

Siguiendo con el proceso anterior, se procedió a realizar una segunda ronda de entrenamiento, manteniendo el mismo número de épocas y sin realizar ningún cambio.

Se muestran a continuación los resultados de la variación de la precisión y la pérdida después de reentrenar los tres candidatos. En el caso de la precisión se puede observar que el crecimiento ha sido estable en todos los modelos, aumentando esta medida en cerca de un 3% para todos los casos.

Actualmente, los tres candidatos seleccionados son estables al 90% de precisión, por lo cual, el seleccionado de éstos tres debería otorgar 9 de 10 resultados positivos de promedio.

| ID | Modelo | Precisión | Precisión final | Variación (%) |
|----|---------------------------------|-----------|-----------------|---------------|
| 5 | C32k44_C128k44_d075_D256_d05_D5 | 0.8980 | 0.9286 | +3,06% |
| 7 | C32k55_C64k55_d025_D128_d05_D5 | 0.8878 | 0.9184 | +3,06% |
| 8 | C64k33_C128k33_d025_D256_d05_D5 | 0.8776 | 0.9082 | +3,06% |

Por su parte, el factor de pérdida ha disminuido mucho más en el candidato 5, con una reducción de cerca del 20% respecto de la unidad y cerca de un 6% de promedio a sus competidores.

Es de reseñar además la brecha que tenía respecto al modelo 8, que había resultado ser el mejor en esta categoría en la primera fase se ha reducido notablemente. Es de suponer que tarde o temprano su rendimiento se igualará en términos de error.

| ID | Modelo | Pérdida | Perdida Final | Variación (%) |
|----|---------------------------------|---------|---------------|---------------|
| 5 | C32k44_C128k44_d075_D256_d05_D5 | 0.4994 | 0.2973 | -20,72 % |
| 7 | C32k55_C64k55_d025_D128_d05_D5 | 0.4993 | 0.3209 | -17,84% |
| 8 | C64k33_C128k33_d025_D256_d05_D5 | 0.4283 | 0.2849 | -14,34% |

3. Selección del candidato

En vista de los resultados, hemos decidido optar por el modelo número 5, pues es el que ha demostrado tener un comportamiento mucho más positivo después de entrenar 200 épocas. Para este candidato, los valores de pérdida y precisión son aceptables, y no demuestran un comportamiento estabilizado (pueden seguir decreciendo un poco más si se continuase entrenando).

La estructura se compone en primer lugar por una capa convolutiva de 32 filtros y un kernel-size de 4x4. A continuación, se encuentra otra capa convolutiva de 128 filtros y kernel-size de 4x4.

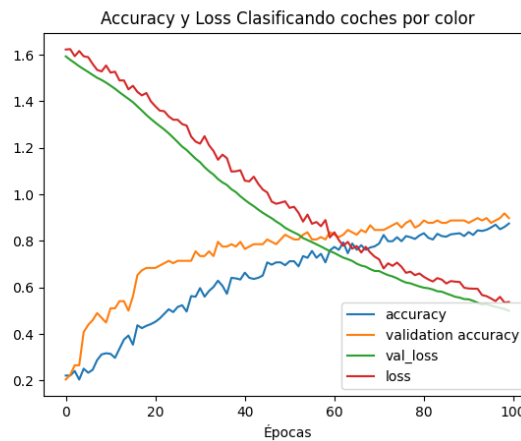
Luego se realiza un dropout del 75% de las neuronas de dicha capa, se pasa a una capa fully-connected de 256 filtros, se realiza otro dropout del 5% de las neuronas de esta capa y por último se forma una capa fully-connected de 5 capas, pues la red clasifica entre 5 categorías diferentes.

III. Analizando el modelo seleccionado

1. Comportamiento en primer entrenamiento

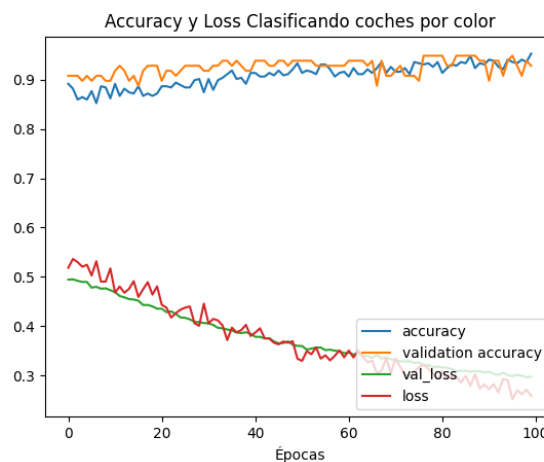
Se puede observar que el comportamiento entre la precisión del conjunto de validación y la precisión principal, así como de los errores correspondientes, no varía ni muestra anomalías significativas durante el desarrollo del entrenamiento.

Es plausible descartar la existencia del *overfitting* (sobreajuste), debido a que las trayectorias de las precisiones no se alejan y siguen un comportamiento común.



2. Comportamiento en segundo entrenamiento

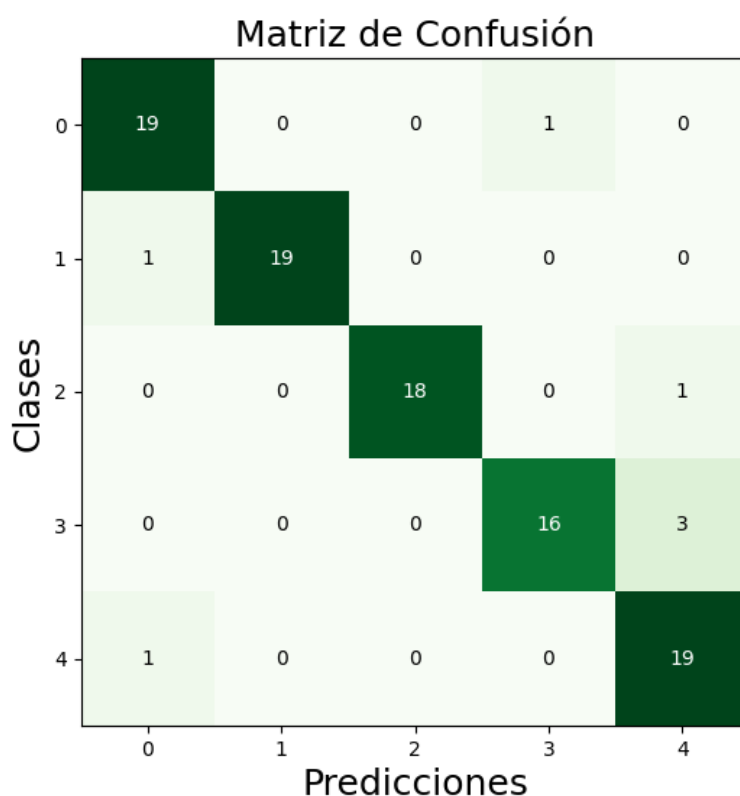
En el segundo entrenamiento el comportamiento se mantiene similar al primero, si bien la escala es mucho menor debido a que partimos del mismo modelo ya entrenado. En esta segunda etapa, se continúa observando el descenso progresivo de la pérdida y un aumento continuado de la precisión, aunque ambas ya comienzan a estabilizarse.



3. Matriz de confusión del candidato seleccionado

A continuación, se muestra la matriz de confusión resultante del modelo seleccionado. Como se puede observar, las tasas de acierto son superiores al 90% en todos los casos, lo cual corrobora la precisión obtenida durante las etapas de entrenamiento previas.

Estas conclusiones se extraen a primera vista, debido a que en la figura los valores más oscuros, que son aquellos en los que el modelo demuestra mayor dificultad para clasificar, se encuentran en la diagonal principal de la matriz, lo cual tiene sentido si se entiende que tiene mayor dificultad para clasificar a una imagen de la clase 1 como de su misma clase.



IV. Anexo

1. Funciones empleadas en el modelo

Softmax: Es parecida a la sigmoide, puesto que ésta también convierte cualquier valor a un valor entre cero y uno. La diferencia está en que la función softmax no trabaja sobre un valor sino sobre un vector. De esta forma, convierte todas las componentes de un vector en valores entre cero y uno, pero, además, garantiza que la suma de todos estos valores sea 1.

La salida de la función softmax conforma una distribución de probabilidad.

Su expresión es:

$$\text{Softmax}(\vec{v})_i = \frac{e^{v_i}}{\sum_{j=1}^n e^{v_j}}$$

Fuente: Apuntes de clase.

Categorical Crossentropy: Es una función de pérdida que evalúa el funcionamiento de un modelo de clasificación cuya salida es un valor entre 0 y 1, como ocurre con la función Softmax previamente mencionada. Debido a que se en la pérdida orienta de forma categórica, es adecuada para supuestos en los que se desea situar un elemento en una clase dentro de un conjunto de posibles clases candidatas.

Los valores de pérdida que se obtienen en una predicción serán más cercanos a 0 cuanto más se acerque el resultado de la misma al valor del vector de tipo one-hot.

Fuente: <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy>