Universidad
Carlos III de Madrid

Grado en Ingeniería Informática

Curso 2020/2021

Redes de Neuronas Artificiales

Problema de Clasificación: Parte II

Clasificación de imágenes con Redes Convolucionales

Autores:

Alba Reinders Sánchez	*100383444*
Alejandro Valverde Mahou	*100383383*

Índice

1. Introducción

El problema consiste en clasificar imágenes donde las entradas de la red son directamente los píxeles de cada imagen. Se utiliza el conjunto de datos *CIFAR10*, compuesto por **60000** imágenes en color (3 canales, *RGB*) de **32x32** píxeles. El conjunto de datos se divide en 50000 imágenes para entrenamiento y 10000 para test.

Hay un total de **10 clases** con 6000 imágenes por clase, por lo que en este caso las clases sí están balanceadas, las diferentes clases son:

- 0 → *airplane*
- 1 → *automobile*
- 2 → *bird*
- 3 → *cat*
- 4 → *deer*
- 5 → *dog*
- 6 → *frog*
- 7 → *horse*
- 8 → *ship*
- 9 → *truck*

El objetivo de la práctica es entrenar diferentes arquitecturas de **Perceptrón Multicapa** y **Redes de Neuronas Convolucionales** para analizar cómo influyen sus hiperparámetros en la resolución del problema de clasificación. Además de comparar sus resultados para comprobar cuál de las dos arquitecturas es más efectiva.

2. Diseño, entrenamiento y evaluación del PM

Inicialmente se intenta resolver este problema con el método de 'fuerza bruta'. Es decir, se aplana la información de los píxeles de las imágenes de entrada en un único vector y se entrena un **Perceptrón Multicapa** con estas entradas.

Para estudiar la eficacia de este acercamiento se realiza una pequeña experimentación probando distintas arquitecturas de red:

Arquitectura	***epochs***	***Accuracy* entrenamiento**	***Accuracy* test**	***Loss* entrenamiento**	***Loss* test**
(50)	25	0.5552	0.4743	1.2606	1.5040
(25)	21	0.5039	0.4493	1.4128	1.5678
(100)	16	0.5543	0.4922	1.9684	1.4483
(100,100)	30	0.6442	0.5045	0.9995	1.4618
(100, 100, 100)	23	0.5900	0.5081	1.1478	1.4276

Tabla Experimentos PM

En cada uno de los experimentos se ajusta manualmente el número de *epochs* óptimos para encontrar los mejores resultados posibles.

Se parte de la arquitectura dada en el tutorial y se prueba a modificar sus hiperparámetros a partir de ella. Primero se prueba a disminuir el número de neuronas, lo que no genera mejores resultados. Por tanto se prueba a aumentarlas y el *accuracy* mejora, entonces se decide añadir una capa oculta más y también genera mejores resultados.

Como último experimento se configura una red con 3 capas ocultas de 100 neuronas cada una, este obtiene los mejores resultados. Sin embargo, el *accuracy* en test que alcanza se queda estancado entorno a 0,5.

Tras esta experimentación se puede concluir que aumentar la complejidad de la arquitectura de la red, tanto en capas ocultas como en neuronas, hace que genere mejores resultados. Aunque no consigue superar un umbral, por lo que la utilización del *PM* para resolver este problema no es del todo eficaz.

Real \ Predicho	**Airplane**	**Automobile**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane										
Automobile										
Bird										
Cat										
Deer										
Dog										
Frog										
Horse										
Ship										
Truck										

Matriz de confusión mejor experimeto *PM*

Real \ Predicho	**Airplane**	**Automobile**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane										
Automobile										
Bird										
Cat										
Deer										
Dog										
Frog										
Horse										
Ship										
Truck										

Matriz de confusión mejor experimeto *CNN*

3. Diseño, entrenamiento y evaluación de la CNN

4. Comparación PM y CNN

5. Conclusión