



UNIVERSIDAD DE GRANADA

VISIÓN POR COMPUTADOR
GRADO EN INGENIERÍA INFORMÁTICA

PRÁCTICA 2

REDES NEURONALES CONVOLUCIONALES

Autor

Vladislav Nikolov Vasilev

Rama

Computación y Sistemas Inteligentes



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

CURSO 2019-2020

Índice

1. BASENET EN CIFAR100	2
2. MEJORA DEL MODELO	3
3. TRANSFERENCIA DE MODELOS Y AJUSTE FINO CON RESNET50 PARA LA BASE DE DATOS CALTECH-UCSD	3
Referencias	4

1. BASENET EN CIFAR100

Antes de empezar con la traducción de la arquitectura proporcionada de BaseNet, hace falta establecer la forma de la entrada de la primera capa de la red. Esto es necesario, ya que el modelo necesita conocer dicho tamaño para poder ser compilado sin ningún tipo de error. Como las imágenes de *CIFAR100* tienen un tamaño de 32×32 píxeles, y tienen 3 canales, la dimensión de la entrada va a ser la siguiente:

```
1 # Tamaño de la entrada
2 input_shape = (32, 32, 3)
```

Una vez definida la forma de la entrada, ya se puede empezar a hacer la traducción a código. El resultado es el siguiente:

```
1 # Creacion del modelo
2 model = Sequential()
3 model.add(Conv2D(6, kernel_size=(5, 5), padding='valid',
4                 input_shape=input_shape))
5 model.add(Activation('relu'))
6 model.add(MaxPooling2D(pool_size=(2, 2)))
7
8 model.add(Conv2D(16, kernel_size=(5, 5), padding='valid'))
9 model.add(Activation('relu'))
10 model.add(MaxPooling2D(pool_size=(2, 2)))
11
12 model.add(Flatten())
13 model.add(Dense(units=50))
14 model.add(Activation('relu'))
15 model.add(Dense(units=25))
16 model.add(Activation('softmax'))
```

BaseNet es un modelo secuencial, así que empezamos indicando eso. A continuación, añadimos el primer módulo convolucional. Este se compone de una convolución 2D con un *kernel* de 5×5 , una función de activación no lineal (RELU en este caso) y un MaxPooling de tamaño 2×2 . El parámetro *padding = valid* de *Conv2D* indica que solo se tiene que aplicar la convolución allá donde se pueda ajustar el *kernel*; es decir, como en las regiones de los bordes no se puede, se van a ignorar estas zonas, lo cual implica que la salida no va a tener el mismo tamaño que la entrada. Este módulo convolucional se repite otra vez. Después de eso, nos encontramos con las capas densas, las cuáles van a actuar como clasificador. La capa *Flatten* es necesario ponerla, ya que coge la salida de la anterior y la aplanar, dejándola como un vector de características que sirve como entrada al modelo denso. La última capa, la de activación *softmax* es la que va a dar la salida, clasificando las entradas.

2. MEJORA DEL MODELO

3. TRANSFERENCIA DE MODELOS Y AJUSTE FINO CON RESNET50 PARA LA BASE DE DATOS CALTECH-UCSD

Referencias

- [1] Texto referencia
<https://url.referencia.com>