

在這次作業中，我利用 keras 搭建了一個卷積神經網路模型(CNN model)來當作一個分類器，用來判別 10 種不同動物的類別。

圖一為模型的架構圖。第一個卷積層的 filter 數為 16、kernel_size=[5,5]、activation function 是”relu”；接著是 size=[2,2]的最大池化層；第二個卷積層僅 filter 數量與第一個不同(36 個)，後面一樣也接著最大池化層，並多了一層 dropout 隨機丟去 25%神經元。再來是一個扁平層以及一個有 128 個神經元的全連接層(activation function 一樣是”relu”)；最後經過一個 dropout 層(丟去 50%)之後，輸出 10 個神經元的 output layer。

圖一、

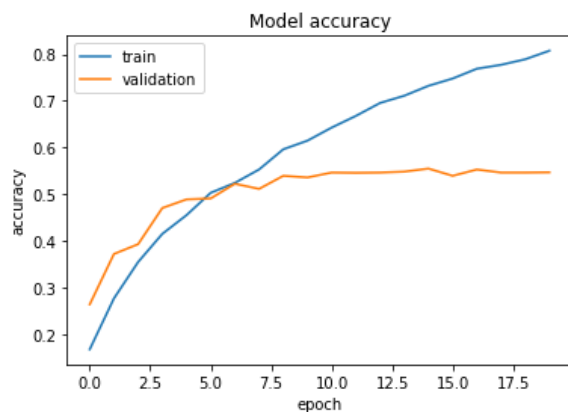
```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 64, 64, 16)	1216
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 16)	0
conv2d_2 (Conv2D)	(None, 32, 32, 36)	14436
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 36)	0
dropout_1 (Dropout)	(None, 16, 16, 36)	0
flatten_1 (Flatten)	(None, 9216)	0
dense_1 (Dense)	(None, 128)	1179776
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 10)	1290

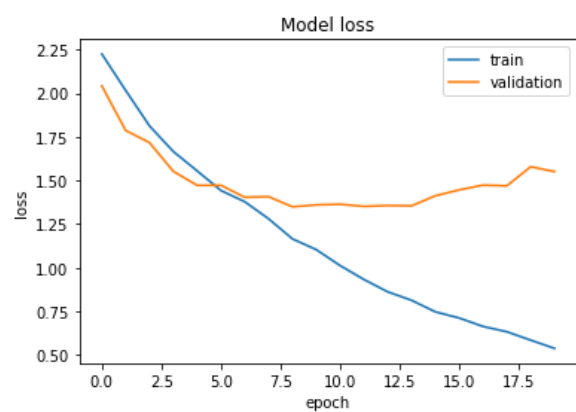
Total params: 1,196,718
Trainable params: 1,196,718
Non-trainable params: 0

定義完模型架構之後開始進行訓練，訓練的 accuracy 和 loss 趨勢圖顯示於圖二及圖三。雖然 training data 的準確率能達到 80%，但 validation set 只有 55%；另外在 validation 的 loss 在第 13 個 epochs 之後有微幅的上升，因此以上兩點可以知道已經 overfitting 了。

圖二、



圖三、



這次作業我針對許多參數進行調整，像是：卷積層的數量、卷積層 filter 的數量、卷積層的 kernel_size、dropout 的比例、全連接層的數量、全連接層的神經元數目、batch_size 等等，做了多次訓練，validation data 的 accuracy 都在 50% 左右，只好選了一個最好的 (55%)，但最後回去試助教給的 model，才發現 validation data 的 accuracy 竟然可以到 60%！不過 train accuracy 就沒有我的高。總之，我覺得深度學習沒有我想像中的輕鬆，雖然只需要套現成的模型，但最難的就是要調整這些參數，以及資料前處理了。