

REPORT

Method description

這次作業我使用由 tensorflow 團隊提供，已經在 ImageNet 上訓練好的 VGG16 模型，並在上面 fine-tune Dog Breed 分類器。

VGG16 pretrained weights: http://download.tensorflow.org/models/vgg_1620160828.tar.gz

VGG16 定義程式碼: <https://github.com/tensorflow/models/blob/master/research/slim/nets/vgg.py>

我只使用預訓練的卷積層 (Convolutional Layer)，把所有的全連接層 (Fully Connected Layer) 拿掉，並固定預訓練的卷積層權重，最後加上一層用來輸出分類結果的全連接層，只訓練這層全連接層。

簡而言之，我使用預訓練的 VGG16 權重，但是不去訓練已經訓練好的部分，只訓練最後輸出結果的全連接層。

訓練時，使用隨機套用的左右鏡射、旋轉、Shearing、Padding、Cropping、Translate、Rescale、Gaussian Blur、Median Blur、Average Blur、高斯噪聲、Dropout，並隨機調整圖片色調、彩度，再加上隨機的 Perspective Transform。

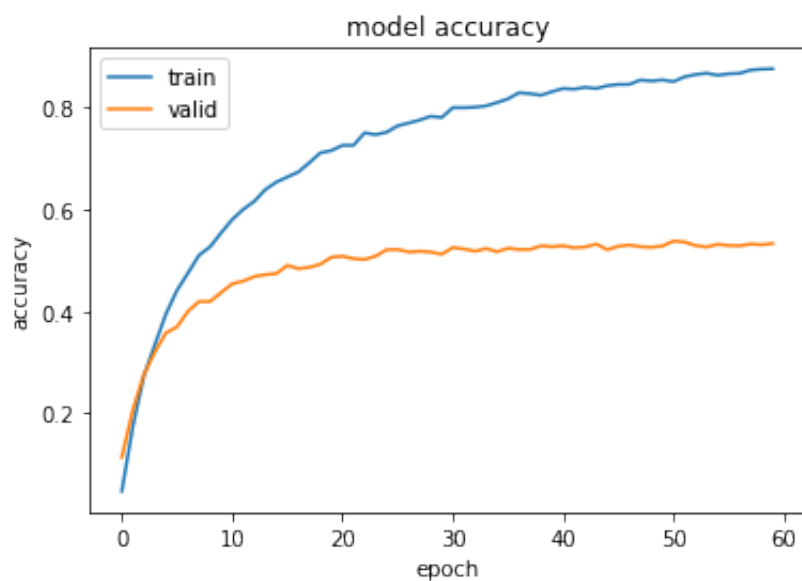
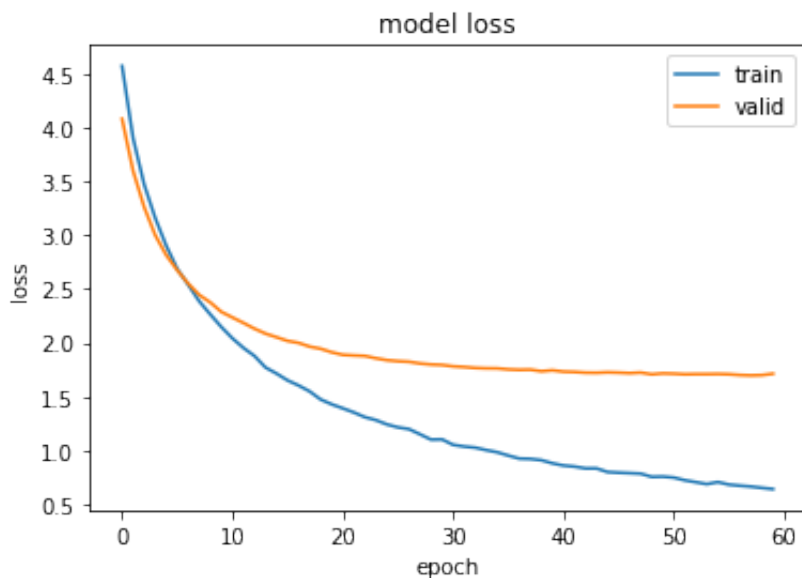
用這樣的操作，來作為這次訓練的 Data Augmentation。

訓練時，每一個 Epoch 開始時，程式會將圖片讀取順序打亂，增加一點隨機性，能讓模型學得更好。

Experimental results

使用 pretrained weights 再加上 Data Augmentation 的效果不錯，原先如果只使用 random 初始值，從頭訓練 VGG16 的話，validation accuracy 只有 2% 左右，而 loss 大概也只有 4.x ~ 5.x。而使用 ImageNet pretrain weights 後，validation accuracy 可以達到 53% 左右，loss 1.7132，差距非常大。

下面兩張分別是訓練時，training 和 validation 的 loss plot 與 accuracy plot。



訓練了 60 個 epochs，最後得到訓練結果：

Accuracy: 53.35%

Loss: 1.7132

Discussion of difficulty or problem encountered

剛開始 fine-tune 助教提供的 pretrained weights 時，發現效果比 random 初始化還差，後來只好自己去下載在 ImagenetNet 上訓練好的 pretrained weights。

雖然沒有明顯的 overfitting 發生，不過 training 和 validation loss 之間的 gap 非常大，也許一般性還不夠好？這可能要再試試更多 fine-tune 的技巧，使得模型的一般性能夠更好。

或者，可以試著使用別的模型的 pretrain weights，例如 Xception, ResNet 等較新一點的模型，搞不好會有一些驚喜。

