Day 21

深度學習與電腦視覺

學習馬拉松

cupoy 陪跑專家：楊哲寧

CUPOY

深度學習理論與實作

CNN Transfer Learning
(遷移式學習)

重要知識點

- 了解 Transfer Learning 的優勢
- 了解如何使用 Keras 做Transfer Learning

遷移式學習

CUPOY

Transfer Learning (遷移式學習)為深度學習中常見的做法，其概念是利用過去訓練過的**結構**與**學習到的權重**來加快這次的訓練任務。

遷移式學習

CUPOY

而之所以能夠這麼做是因為不同的分類任務間仍有許多**共享的特徵**，如類別貓與花瓶間也可能有相似的輪廓，因此過去訓練過分類貓的分類器，其權重也能使用來加速這次訓練花瓶分類器的速度。

參考來源： Read01

遷移式學習

通常淺層的卷積核學到比較粗略、泛用的特徵，因此在做 Transfer Learning 時，當新的 Dataset 與原本的 Dataset 相近時，可以考慮不更新淺層 Kernel 的權重(freeze)，又或是資料不足，擔心 Overfitting 時也可以使用。

Freeze or fine-tune?

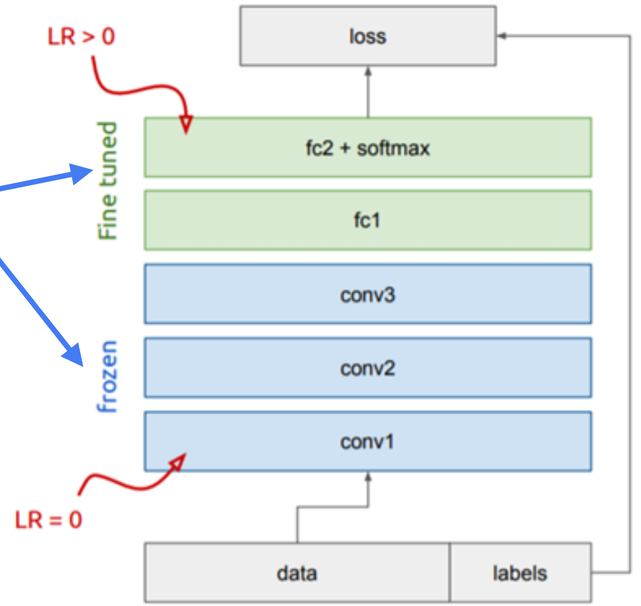
Bottom *n* layers can be frozen or fine tuned.

- **Frozen**: not updated during backprop
- **Fine-tuned**: updated during backprop

Which to do depends on target task:

- **Freeze**: target task labels are scarce, and we want to avoid overfitting
- **Fine-tune**: target task labels are more plentiful

In general, we can set learning rates to be different for each layer to find a tradeoff between freezing and fine tuning



參考來源： Towards Data Science

遷移式學習

CUPOY

- Freeze 的另一個好處是能加快訓練，然而當訓練資料的特徵差異太大時，還是建議可以全部開啟或**只 freeze 少數層**，而要 freeze 具體的層數，並沒有一定的規範。

- 在 Keras 中具體作法相當簡單，可以參考下方程式碼：

Freeze 前 100 層

```
for layer in model.layers[:100]:
    layer.trainable = False
for layer in model.layers[100:]:
    layer.trainable = True
```

遷移式學習

CUPOY

- 具體實作第一步，我們可以從 Keras 讀入特定模型架構，在這裡我們用 ResNet50 示範，所有可以使用的架構可以參考官網
- 其中 weight= ‘imagenet’ 代表我們要輸入 ImageNet 的 pretrained weight，用來做 Transfer Learning。

定義輸入影像大小

```
input_tensor = Input(shape=(32, 32, 3))
```

代表不要導入Fully connected Layer，主要是由於輸出類別數量不同(原本為1000)。

```
model=keras.applications.ResNet50(include_top=False,
weights='imagenet',input_tensor=input_tensor,pooling=None, classes=10)
```

遷移式學習

CUPOY

第二步：可以自己在原本架構後面再新增幾層，尤其當沒有導入Fully connected layers時，是必要自己加上 FC 做分類。

接上原本的output

```
x = model.output
x = GlobalAveragePooling2D()(x)
x = Dense(output_dim=128, activation='relu')(x)
x=Dropout(p=0.1)(x)
predictions = Dense(output_dim=10,activation='softmax')(x)
model = Model(inputs=model.input, outputs=predictions)
```

重新定義輸出類別數量

遷移式學習

CUPOY

第三步：定義要 Freeze 的層數，之後就跟訓練一般模型一樣了。

可以先看看模型深度決定鎖定層數

print('Model深度：', len(model.layers))

```
for layer in model.layers[:100]:
    layer.trainable = False
for layer in model.layers[100:]:
    layer.trainable = True
```

解題時間 Let’s Crack It

CUPOY

請跳出 PDF 至官網 Sample Code & 作業開始解題