

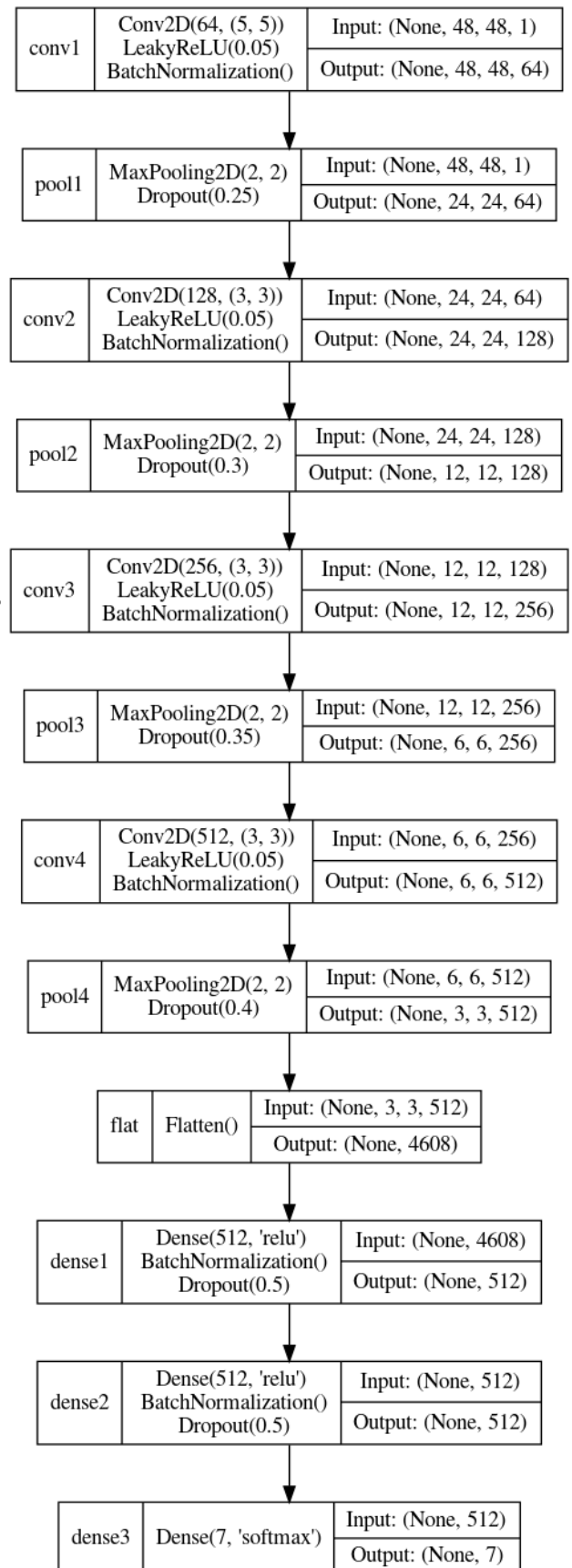
1. (2%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？並請用與上述 CNN 接近的參數量，實做簡單的 DNN model，同時也說明其模型架構、訓練參數和準確率為何？並說明你觀察到了什麼？

(Collaborators: 無)

答：

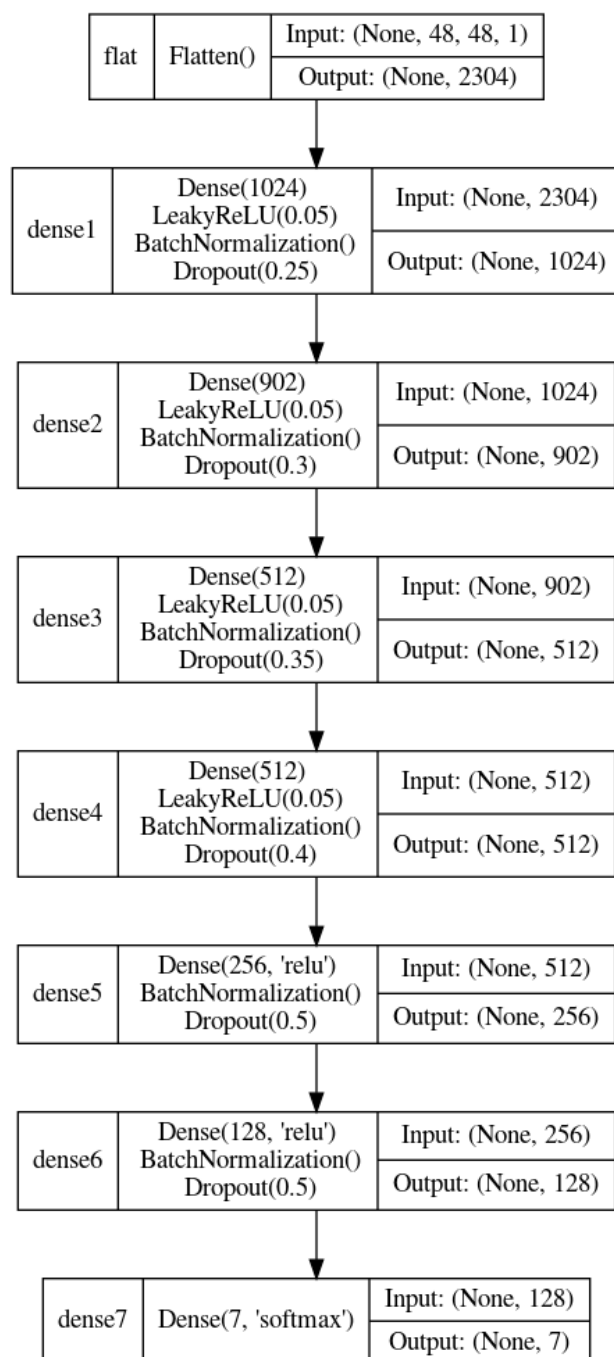
右圖是我使用的 CNN model，每個 convolution block 包含了一層 Convolution layer、一層 LeakyReLU layer 和一層 BatchNormalization layer；每個 pooling block 包含了一層 MaxPooling layer 和一層 Dropout layer。Conv2D 的一個參數是 filter 的數目，第二個參數是 filter 的大小，stride 皆設為 1，並且有使用 padding 使得 input 和 output 的圖片大小相同。MaxPooling2D 的參數代表了 filter 的大小，因為皆為(2, 2)故通過後圖片的大小會減半。交錯通過 4 個 convolution block 和 pooling block 之後通過 Flatten layer，之後通過兩層的 dense block，每個 block 包含了一層 Dense layer、一層 BatchNormalization layer 和一層 Dropout layer。Dense layer 的第一個參數為 output dimension，第二個參數為 activation function。最後再通過以 softmax 為 activation function 的 Dense layer。

在訓練時有使用 data augmentation，data 皆有 normalization(除以 255)，訓練時選擇前 22000 筆資料當作 training data，其餘資料為 validation data。optimizer 使用 adam，batch_size 設為 128，訓練了 200 個 epoch 後在 kaggle 上的正確率分別為 public: 0.68152，private: 0.69100。



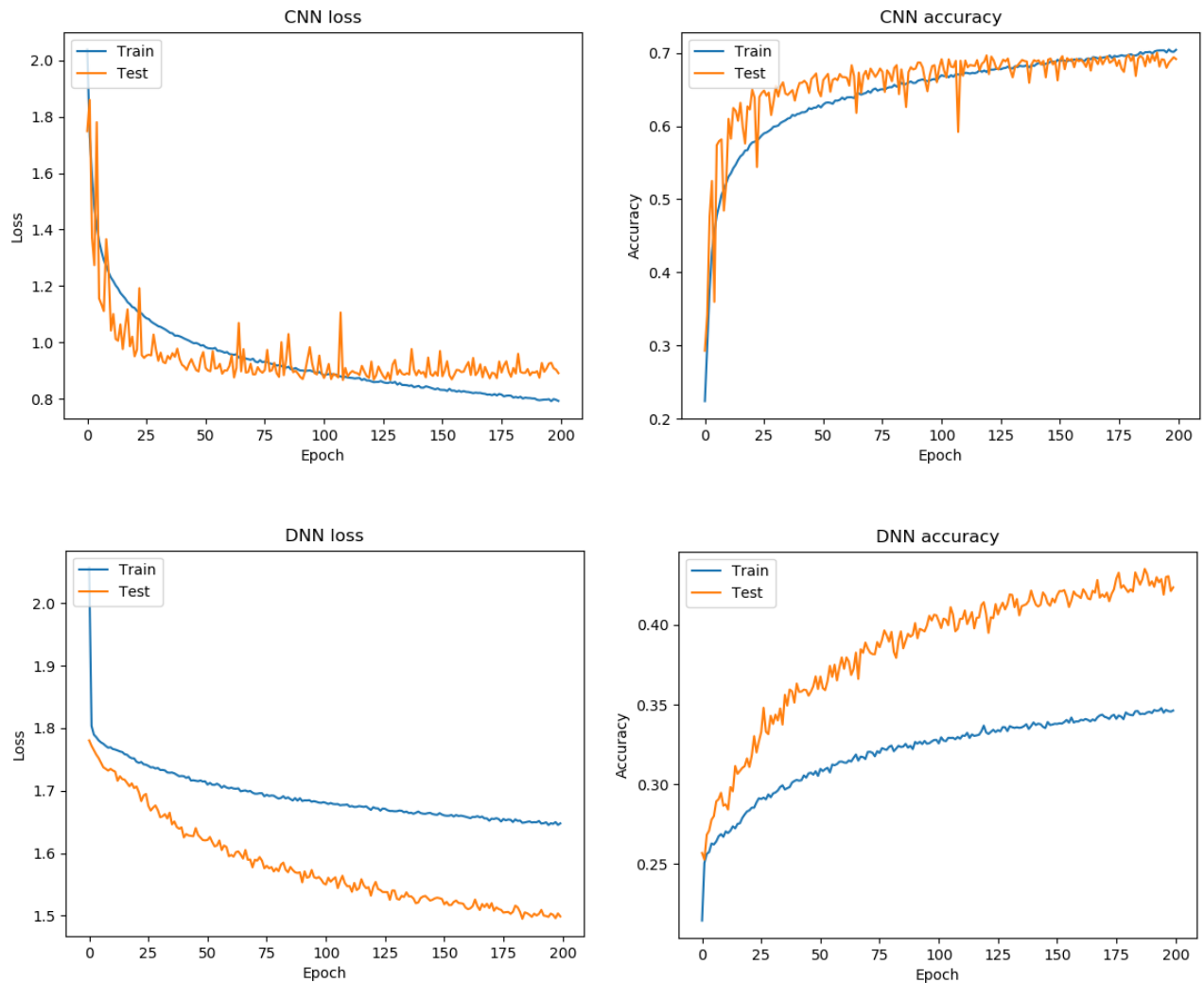
上述 CNN model 可訓練的參數共有 4180871 個，我架了一個有 4181657 個可訓練參數的 DNN model，結構如右圖。訓練時和 CNN 使用了一模一樣的條件，在 kaggle 上的正確率分別為 public: 0.40874，private: 0.42184。

我觀察到雖然使用了差不多的參數，甚至 DNN 還比 CNN 多了一點，但 DNN 的表現完全不如 CNN。此外我觀察到雖然有差不多的參數，但 DNN 的訓練時間較短。



2. (1%) 承上題，請分別畫出這兩個 model 的訓練過程 (i.e., loss/accuracy v.s. epoch)
(Collaborators: 無)

答：



上面的圖中 train 代表 training data，test 代表 validation data。至於為何在 DNN 中 train 的 loss 比 test 高，accuracy 比 test 低是因為我使用了 Dropout，以下是 keras 的官方說明：<https://keras.io/getting-started/faq/#why-is-the-training-loss-much-higher-than-the-testing-loss>

“A Keras model has two modes: training and testing. Regularization mechanisms, such as Dropout and L1/L2 weight regularization, are turned off at testing time.”

因此為了呈現 Dropout 關掉後的實際情形，我在訓練完後對兩種 model evaluate，得到以下結果：

	Train loss	Test loss	Train acc	Test acc
CNN	0.439	0.891	0.84	0.69
DNN	1.462	1.499	0.44	0.42

3. (1%) 請嘗試 data normalization, data augmentation, 說明實作方法並且說明實行前後對準確率有什麼樣的影響？

(Collaborators: 無)

答：

data normalization：將圖片除以 255。

data augmentation：使用 keras 的 ImageDataGenerator，rotation_range 設為 30，width_shift_range、height_shift_range、shear_range 設為 0.2，zoom_range 設為 [0.75, 1.25]，horizontal_flip 設為 True。

在 validation data 上的準確率如下：

raw	normalization	normalization+augmentation
0.6286	0.6308	0.6916

可以發現 normalization 和 augmentation 皆會提升準確率。

4. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]

(Collaborators: 無)

答：右圖是以 validation data 畫出的 confusion matrix，可看出 disgust 容易被誤判成 angry，但反過來卻不會；fear 容易被誤判為 sad、surprise 和 neutral；sad 容易被誤判為 neutral，但反過來也不會。

