Machine Learning HW3 Report

學號：B06901007 系級： 電機二 姓名：戴子宜

2019.4.6

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:)

(a) CNN model 的模型架構是由 8 層 convolution layer 最後再接上 256 x 512 x 7 的 fully connected feedforward layer。每層 convolution layer 之後都會再經過 batch normalization 和 leakyReLU。每經過兩次 convolution 就會經過 maxpool 和 dropout，而 dropout 的機率會隨著層數後面越大。訓練參數的部分：batch size 為 256，learning rate 是 0.01，參數更新的方式是 Adam，總共訓練 200 個 epoch。最終的準確率在 kaggle public score 為 0.68264，在 train set 的準確率為 0.98 左右，在 validation set 的準確率為 0.67 左右。

(b) DNN model 的模型架構是 1024 x 1024 x 1024 x 1024 x 512 x7 的 fully connected feedforward layer，每層 layer 都會經過 leakyReLU 和 batch normalization。訓練參數的部分：batch size 為 256，learning rate 是 0.01，參數更新的方式是 Adam，總共訓練 200 個 epoch。

由上述的結果可以知道，Cnn 的模型可以在參數量相同的情況下，獲得更高的準確率。

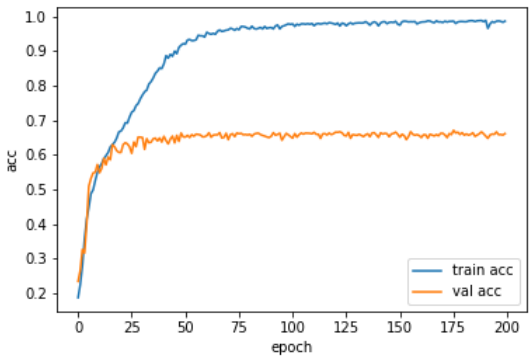
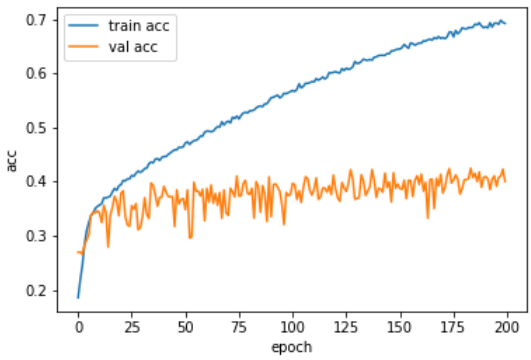
```
model.add(Conv2D(64, (3, 3), padding='same', input_shape=(48,48,1),data_format='channels_last' ))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=0.05))
model.add(Conv2D(64, (3, 3), padding='same',data_format='channels_last' ))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=0.05))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(0.25))


model.add(Conv2D(128, (3, 3), padding='same',data_format='channels_last' ))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=0.2))
model.add(Conv2D(128, (3, 3), padding='same',data_format='channels_last' ))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=0.2))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(0.3))


model.add(Conv2D(256, (3, 3), padding='same',data_format='channels_last' ))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=0.05))
model.add(Conv2D(256, (3, 3), padding='same',data_format='channels_last' ))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=0.05))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(0.35))

model.add(Conv2D(512, (3, 3), padding='same',data_format='channels_last' ))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=0.05))
model.add(Conv2D(512, (3, 3), padding='same',data_format='channels_last' ))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=0.05))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(0.35))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(7))
model.add(Activation('softmax'))
```

CNN model

```
model = Sequential()

model.add(Flatten(input_shape=(48,48,1)))
model.add(Dense(1024))
model.add(LeakyReLU(alpha=0.03))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(1024))
model.add(LeakyReLU(alpha=0.03))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(1024))
model.add(LeakyReLU(alpha=0.03))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(1024))
model.add(LeakyReLU(alpha=0.03))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(512))
model.add(LeakyReLU(alpha=0.03))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(7))
model.add(Activation('softmax'))
```

DNN model

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:)

CNN model	DNN model
	

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:)

(a) Normalization:
在加入 data normalization 以前（第一題的 cnn model 把 batch normalization 的部分拿掉），原本的 model 的 validation set 上的準確率大約是 0.55 左右，加入 batch normalization，同時一開始就將所有的 pixel 的值 normalize 到 0 和 1 之間，訓練的來的 model 在 validation set 上就可以獲得 0.64 左右的準確率。

(b) Data Normalization + Data Augmentation:
利用 keras 的 image generator 可以實作 data augmentation，利用 rotation, width_shift, height_shift, shear, zoom, horizontal range，增加 data 的數量，平均每個 epoch 多看 10 倍的數量的 data，這樣可以讓 model 的準確率提升至 0.67 左右。

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:)

利用 validation set 可以找出 confusion matrix。從 confusion matrix 可以看出，angry 和 disgust、sad 和 fear、neutral 和 sad 容易搞混。fear 本身只有 0.41 的準確率會被判斷成 fear，有約 0.15 的機率被判斷成 angry 或 sad 或 surprise。