MSBD6000B Deep Learning Project 2

QIU Feng (fqiuaa, 20398324)

1. Data pre-processing

   By using the Image package in PIL library, I import images through their paths stored in train.txt, val.txt and test.txt. I also import labels by splitting them from their paths. Changing all the jpg files into Numpy array and using One-Hot-Encoding for labels are my final steps for data pre-processing.

2. Experiment deployment

   In this project, I am using 2D Convolutional Neural Network with 4 Conv2D layers, 2 MaxPooling2D layers and 2 Dense layers to train a deep model. I will use 2569 data points for training and 550 data points for validation.

3. Experiment results

   - In order to avoid overfitting, I add a Batch Normalization into the 2D CNN to train and also train the model without the function. When comparing the validation accuracy results, adding the batch normalization decreases the test accuracy while increasing the training time. It normalizes the network input weights between 0 and 1.
   - I also compare the validation accuracy results by adding Dropout layer to the 2D CNN to avoid overfitting. Adding the dropout layer increases the test accuracy when comparing to the 2D CNN. Dropout layer adds regularization to the network by preventing weights to converge at the same position. During forward propagation, nodes are turned off

randomly while all nodes are turned on during forward propagation.

4. Conclusion

In terms of accuracy and stability, 2D CNN model with Dropout layer is the best. I trained for 70 epochs and the final stable result is below.

```
Epoch 65/70
2569/2569 [==============================] - 67s - loss: 0.0695 - acc: 0.9809 - val_loss: 1.8
137 - val_acc: 0.6364
Epoch 66/70
2569/2569 [==============================] - 74s - loss: 0.0665 - acc: 0.9809 - val_loss: 1.7
120 - val_acc: 0.6218
Epoch 67/70
2569/2569 [==============================] - 73s - loss: 0.0683 - acc: 0.9790 - val_loss: 1.7
313 - val_acc: 0.6200
Epoch 68/70
2569/2569 [==============================] - 80s - loss: 0.0693 - acc: 0.9817 - val_loss: 1.8
857 - val_acc: 0.6418
Epoch 69/70
2569/2569 [==============================] - 71s - loss: 0.0589 - acc: 0.9844 - val_loss: 2.0
150 - val_acc: 0.6145
Epoch 70/70
2569/2569 [==============================] - 80s - loss: 0.0660 - acc: 0.9798 - val_loss: 1.7
771 - val_acc: 0.6418
```