Michael Eng

EE628 Final

- Control
  - No data augmentation, no dropout, freeze first 15 convolution layers.
  - Traing and valiadtion:
    - 140s - acc: 0.9586 - val_acc: 0.9183
  - Testing:
    - acc: 0.9196
  - The result is pretty good, but it I slightly overfit and I can do better.
- Data Augmentation
  - Training and validation:
    - 163s - acc: 0.9819 - val_acc: 0.9781
  - Testing:
    - acc: 0.9776
  - Including data augmentation increases all of the accuracies by a substantial amount, however it takes a bit longer to train.
- Dropout
  - Adding a dropout of 50%, the
  - Training and validation:
    - 140s - acc: 0.9771 - val_acc: 0.9772
  - Testing:
    - acc: 0.9772
  - The result of adding dropout completely eliminates overfitting, and increases the testing accuracy.
- Freeze layers
  - Freezing vgg16 layers will reduce time to train each epoch. Since the input data is similar to Imagenet, the accuracy will not reduce too much. In this example, only 10 layers are frozen so the tpp 6 are retrained
  - Training and validation:
    - 145s - acc: 0.9621 - val_acc: 0.9590
  - Testing:
    - acc: 0.9596
  - The result of retraining more layers is a slightly better performance, while also increasing training time.
- Best
  - The best accuracy was achieved with a combination of data augmentation, 50% dropout, and 10 frozen layers.
  - Training and validation:
    - 159s - acc: 0.9782 - val_acc: 0.9799
  - Testing:
    - acc: 0.9816
  - The result is well performing network with little overfitting.

- Takeaway
  - Because I used the vgg16 network, there were over 17 million trainable parameters. This meant that only 3 epochs could be trained or the network would get overfit to the training data. This also means that the improvements from data augmentation, dropout and pretraining are very hard to see as the network is already very good. A good benchmark

would be to not use a vgg16 and to create my own convolution layers. This way I can reduce the trainable parameters and enable the network to be able to generalize to new data.