The source code is at the end of this document

1 Classification: Convolutional Neural Networks

1.1 Design and Implementation Choices of your Model

Inspired by VGG and ResNet, almost all convolution layers are 3×3 and almost all pooling layers are 2×2.

The following CNN models are tested.

CNN1 is a basic CNN with one convolutional layer; CNN2 is similar to the network in the link given in the assignment; CNN3 is CNN2 with one more convolutional layer and max pooling layer to check if CNN2 is deep enough; CNN4 is a VGG like network given in this link. In addition, all models in this blog are evaluated, but they have very similar performance with the previous ones so I will not show their structure below. Meanwhile, VGG19 model with imagenet pretrained weight and transfer learning of InceptionV3 were tested. Finally, I tried training a Resnet 50 and an Inception model from scratch, and the tutorial can be found here(Coursera CNN week 2 programming assignments) and here respectively.

CNN1

CNN1 is a sample CNN with only one convolutional layer. Dropout layer is used, which is the easiest way to prevent overfitting.

CNN2

CNN2 is similar to the network in the link given in the assignment. It has four convolutional layers, each followed by a batch normalization layer, which outputs the previous layer by sub-

stracting the batch mean and dividing it by batch standard deviation. Batch normalizatoin could speed up training and increase the stability of the network.

CNN3

CNN3 is just CNN2 with an extra block(a Conv2D, a Batch normalization and a Dropout layer), because I wanted to check if a deeper network could do a better job.

CNN4

CNN4 is a VGG like network given in this link.