Winter 2017 Exercise **James and Siyadong**
Text and Visual Classification {cy443, sx225}@cornell.edu

1 Text Classifier

1.1 Problem Definition

To classify the 20 newsgroup in the 20 newsgroups dataset.

1.2 Model Structure

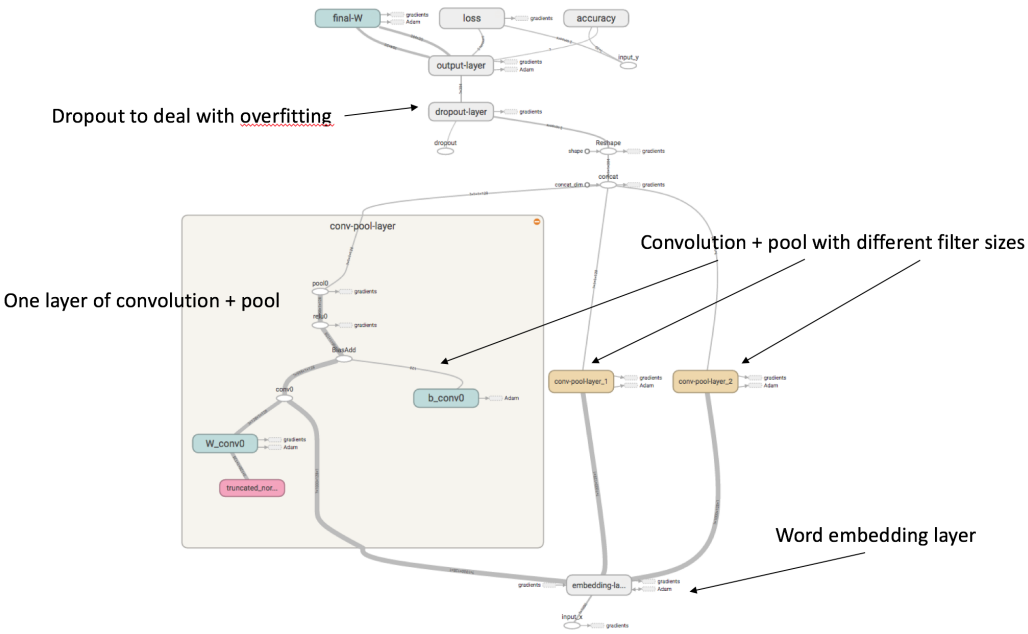

Figure 1: Model structure.

This model architecture is from Convolutional Neural Networks for Sentence Classification[1]. We also looked at the tutorial at [2].

1.3 Experiment

We developed one model in `model.py`, and methods to train and test them in `main.py`. Please refer to `https://github.com/sidxiong/TensorFlowTask`

for source code. Typically, we used `AdamOptimizer` and mini-batch size 128. Optimizer's initial learning rate is fine tuned after several epochs.

We first built a baseline using logistic regression with Tf-Idf Bag-of-Words. We then tried a simplified model mentioned in Sec.1.2. Finally we tried the full model in 1.2. We also tried changing the parameters used in the model described in 1.2.

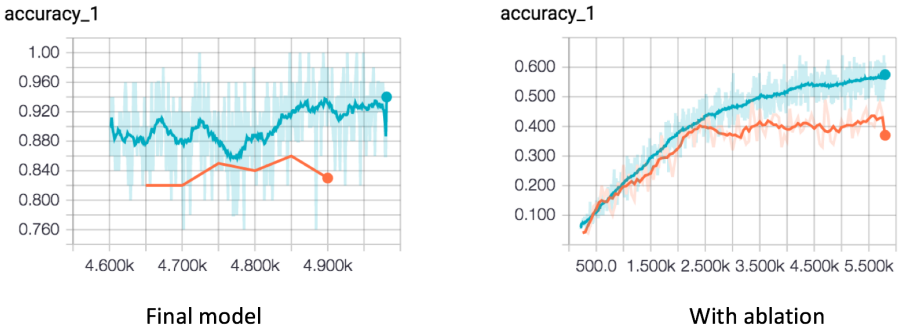

Figure 2: Accuracy comparisons.

Results Using TensorBoard, we can easily monitor the process of training and validation. Train and Validation accuracy are shown in Fig.2.

We tested our models on test set. Results are shown in Table.1.

Table 1: Test set results

Model	Logistic Regression	1 filter size and 10 filters	final model
Test accuracy	72.3%	58.5%	96.9%

2 Image Classifier

2.1 Problem Definition

In this visual classification exercise, we were required to develop simple Convolution Neural Networks using TensorFlow[3] to classify two classes of images from ImageNet[4]. Specifically, we downloaded two categories of images that contain `birds` and `fish` respectively and develop a two class classifier.

Figure 3: Sample images.

2.2 Model Structure and Dataset

Models We tried three models: 1). a simple CNN model; 2). AlexNet's conv layers + three customized fc layers with dropout; 3). AlexNet[6]'s conv layers + direct output with dropout. Model structures details are shown in Fig.4.

For simple CNN, we trained it from scratch; for the latter two AlexNet based models, we fixed the conv layers weights and biases and only fine tuned the fully connected layers parameters. AlexNet based models' parameters were initialized with a pre-trained model downloaded from caffe's website[5].

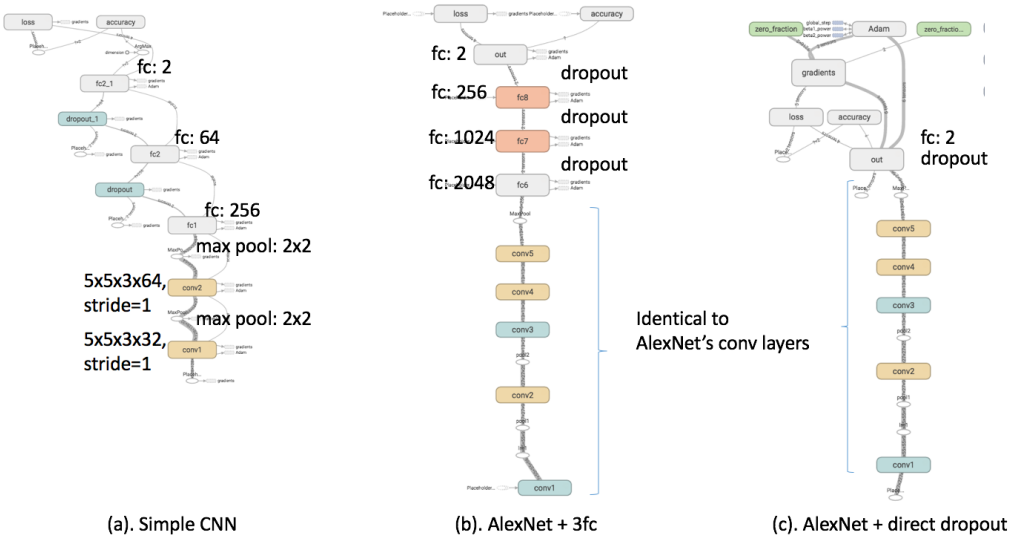

Figure 4: Models structures.

Data set We in total used 2600 images and we divided them into train set, validation set, and test set that contain 1600, 600, and 400 images respectively. We ensured that each set has equal numbers of positive and negative samples. All the images were re-sized towards same size as 128×128 for simple CNN model and 227×227 for AlexNet based models. RGB channels were retained. It is noteworthy that we used data augmentation: for each image, we used flips and rotations to generate five more images. Also, we subtracted mean for every image.

2.3 Experiment

We developed three models in `model.py`, and methods to train and test them in `main.py`. Please refer to `https://github.com/sidxiong/TensorFlowTask` for source code. Typically, we used `AdamOptimizer` and mini-batch size 128. Optimizer's initial learning rate is fine tuned after several epochs.

Results Using TensorBoard, we can easily monitor the process of training and validation. Train and Validation accuracy are shown in Fig.5. We tested

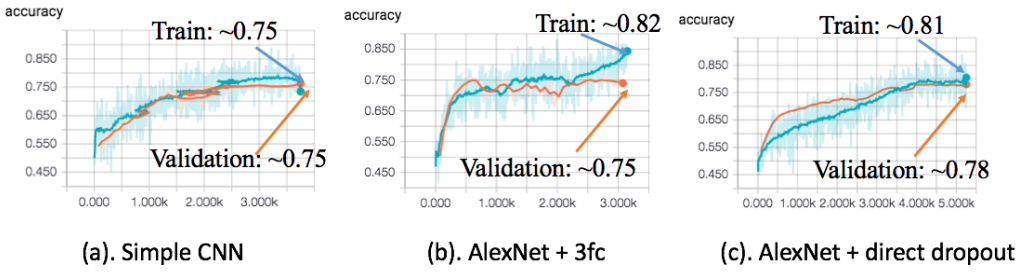

Figure 5: Accuracy comparisons.

our models on test set. Results are shown in Table.2.

Table 2: Test set results

Model	Simple CNN	AlexNet + 3FC	AlexNet + Direct dropout
Test accuracy	72.50%	81.25%	81.50%

2.4 Analysis

We firstly tried a simple CNN trained from scratch. It turned out that it lacked enough representation ability to classify birds and fish. We then utilized pre-trained AlexNet model and fine tuned its last few layers to adapt it

to our problem. For three fully connected layers, because of lack of training images, although we used data augmentation, the model was easily overfitted. Hence, we used a directly dropout model to combat overfitting, which consequently resulted in acceptable performance.

References

[1] Kim Y. (2014). Convolutional Neural Networks for Sentence Classification

[2] http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/

[3] https://www.tensorflow.org/

[4] http://imagenet.org/

[5] http://caffe.berkeleyvision.org/model_zoo.html

[6] Krizhevsky A., Sutskever I., and Hinton G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems.