Assignment 4 Report

Garrett Kinman

1 Model Implementation

The model chosen for this assignment was the Sequential CNN from Keras. The model was built with convolutional, max pooling, and softmax layers. This was a relatively simple model that was implemented with only 8 3×3 filters, making for only 72 weights. It also has a 2×2 pool size and a softmax layer with 10 nodes (one for each category in the classification problem). The training process for this model uses an Adam optimizer and a categorical cross-entropy loss function, and it was set to train over 50 epochs.

2 Results

When training, 1 epoch was used initially. This produced an accuracy of about 76% on the training set. When the model was trained on the training set for 50 epochs, the accuracy converged to approximately 88% accuracy on the training set. Using the model weights from after 50 epochs, a testing set accuracy of approximately 70% was achieved.

3 Challenges

The two foremost challenges in this assignment were as follows: 1) getting the environment and the necessary packages all up and running, and 2) learning the Keras API and all the options available for this particular CNN model.

For the first, it was decided to use Jupyter Notebook on a local runtime (as opposed to Google Collaboratory), as it made it easy to work with particular files without having to use Google Drive for the (relatively) large data files (> 200 MB). However, two challenges cropped up with this. First, using the Kaggle API to download the dataset did not work until a token was downloaded off of the Kaggle website. Second, attempting to import TensorFlow gave an error, as the installed version of Python was 3.7.4, which is as of yet unsupported by TensorFlow. Python 3.7.0 as well as Pip and Jupyter Notebook had to be reinstalled and the PATH variables modified to fix this.

For the second, several existing kernels were studied to see how the Keras API worked. Once that was completed, it was significantly faster to implement a model and make it work.

4 Conclusion

One takeaway from this assignment is the power of CNNs. With only a very basic model with rather few weights to optimize, it was still able to to achieve a test accuracy of 70%. Additionally, Each epoch of the training took only 5 or 6 seconds to complete on a CPU. This seeming efficiency and power suggests that,

with a GPU and more complexity to the layers and hyperparameters of the model, one could achieve a very accurate model that computes fairly quickly.

Another takeaway is the value of Anaconda. Where setting up the environment and packages took some difficulty with the various Python versions and Pip and PATH variables, use of Anaconda in future endeavors might greatly help streamline the process.

5 Individual Contribution

Garrett Kinman performed all parts of this assignment.