Assignment 4

Jenny Zheng, Jack Wei

Project Description:

The goal of this assignment is to use a convolutional neural network (CNN) to identify the highest of 3 integers in a series of black-and-white photos with different backgrounds.

Implementation of the Model:

A convolutional neural network (CNN) with 4 convolutional layers and 3 fully connected layers was used. To implement the CNN, the Keras and TensorFlow Python libraries were used. The first convolutional layer used 64 filters, and for each subsequent convolutional layer the number of filters increased as per convention by powers of 2. Relu activation was used for each of the convolutional layers, and dropout was implemented for each convolutional layer to prevent overfitting. MaxPooling was also utilized for each convolutional layer to reduce the size of the data and control overfitting. To make the CNN faster and more stable, batch normalization was used for each convolutional layer.

Three fully connected layers were implemented, also with batch normalization and dropout. Softmax activation was used for the output layer.

Preliminary inspection on the provided data reveals that the images in the dataset contains many unwanted inputs for classification, namely, the image background. Therefore, it is important to segment the numbers from the background to ensure that the input data is relevant to the objective of classification. A binary threshold with manually tuned threshold value was applied in the preprocessing routine. The thresholded images were then used to train the aforementioned model.

15 epochs were used when training the model, as after 15 epochs, the changes in validation loss and training loss were minimal.

Results:

After tuning the threshold value for preprocessing and applying the CNN, the model yields a loss of 0.1049 and a label classification accuracy of 0.9649 for the training data. The final accuracy for evaluation is 92.0%.

Challenges:

One of the challenges that we encountered was the crashing of the Google Colab notebook due to maxing out of RAM. To account for this problem, we had to split the project into many parts, saving the computing results of each part into Google Drive so that instead of computing all the variables each time, we just had to load the results from Google Drive.

For data preprocessing, we encountered challenges when deciding the threshold value. If the threshold is too low, then the images will result in having a lot of false positives, white specs or blobs that suppress the desirable features. On the other hand, if the threshold is too high, some input features

which are the pixel values of the digits will simply disappear. To decide on the final value, we checked the histogram of various random images in the dataset and determined the average pixel value of the digits. We eventually decided on using 210, a value that is sufficient to segment the digits while not introducing too many false positives.

Conclusion:

From this assignment, we learned how to pre-process data so that it is optimal for model fitting. We also learned how to work with convolutional neural networks and process images. Through this assignment, we became more familiar with useful libraries such as TensorFlow, Keras, cv2, and NumPy.

Individual Contributions

Jenny:

- Loaded and reshaped data so that it can be fit into the CNN model
- Created the convolutional neural network sequential model
- Formatted outputs for Kaggle submission

Jack:

- Inspected the input data by applying histogram on the images
- Proposed binary thresholding to segment the digits
- Manually tuned the hyperparameter for segmentation and preprocessing