Nicholas Lichtsinn

IST 718

Lab 9

**Introduction**

The Fashion-MNIST dataset allows the use of machine learning algorithms to aid in the improvement of computer vision and deep learning. As the traditional MNIST dataset to some is too simple and can easily be modeled with different algorithms with a high success rate.

As the use of convolutional neural networks for image classification has improved this dataset has been used to test different models and to teach people to evaluate model performance for these visual-based classifications.

**Analysis**

First I read in the dataframe fashion-MNIST data with a csv for the machine learning models such as random forest, and boosted forest. For the Keras CNN model I imported the dataset directly from Keras.

I then mapped the categories to the corresponding digit values 0-9. And looked for NA’s.

I split the dataset into train and validation sets, and standardized the data using the StandardScaler in python, then I plotted the PCA Variance before running the models.

Graphical user interface, chart, line chart

Description automatically generated

.

The Random Forest Model was built with the RandomForestClassifier and here are the results:

Text

Description automatically generated

Next, I built a boosted forest model with XGBClassifier resulting in:

A screenshot of a computer

Description automatically generated with medium confidence

The Keras model was built using 20 epochs and the Conv2D package to create a convolution kernel with the layer, Activation was set to ‘relu’ with the input shape of 28, 28, 1 to improve performance. Here is the resulting performance on the test validation.



**Conclusion**

The Keras neural network performed the best with an accuracy of 98% with the boosted forest in second with an accuracy of 87% and the random forest coming in last with an accuracy of 85%. There was not much improvement in the boosted forest, but the CNN Keras model really improved the accuracy after tuning the model.

The CNN model is great at learning accurate patterns and insights from the provided data and can provide better outcomes than other machine learning models if tuned but requires well structured and clean data to do so and uses a large amount of computational power. Random forest models work well to reduce overfitting and reduce variance to improve the accuracy of the model but require the training of many trees using a lot of computational power and can have a long training time to decide even though here it was the fastest model. The Extreme Gradient Boosting model that was used is great for increased performance and can provide parallelization in tree building and out of core computing but sacrifices scalability and speed to train the model.

The compute performance varied with the random forest only taking 47 seconds, and the other two models taking significantly longer to compute with the boosted forest taking 1052 seconds and the Keras model taking 740 seconds.