Prerak Mehta

MSDS 458 Artificial Intelligence and Deep Learning

First Research Assignment

Abstract:

In the assignment we use the MNIST dataset which consists of a set of 70,000 images (28x28 pixel) of handwritten digits. Each image is labeled with the digit it represents. We construct dense neural networks to build a image classifying models and see how various attributes of the dense neural networks such as hidden layers and hidden nodes impact the accuracy of the models. There are a total of 5 experiments being carried out in this research assignment. The goal of the first three experiments is to find the best dense neural network model with one hidden layer and also explore what the node(s) in the hidden layer are detecting and what their outputs contribute to the final classification of an image. In experiment 4, we reduce the dimensions of the best model found from the previous experiment and train it on the new lower dimensional data. And lastly in experiment 5 we use a random forest classifier to get the top 70 features from the 784 features (pixels) and then train the best model using these 70 features.

Introduction:

This research assignment addresses a hypothetical management problem that involves developing a neural network model for digit classification and the ways in which the accuracy can be improved with real data examples of handwritten digits on paper. This research assignment serves as a hands-on experience practical experience with not only designing, training and assessing a neural network but also understand how the nodes in a simple single hidden layer represent features within the input data.

Literature Review:

There has been tons of research done in relation to image classification of handwritten digits. Many have furthered this research by including image recognition of complex mathematical operators and building mathematical equation solving phone applications. Photomath app is probably the most prolific example in this case where a person can just take a photo of the mathematical equation from the app and the app would just output the answer to the equation and in most instances also breakdown the steps. This idea can be much furthered into developing an app that can be used to automate checking of a math test/exam where the whole or most of the test/exam can be checked/graded automatically and human intervention can be needed for the parts that failed to be recognized or had very low recognition accuracy. Students can be given extra credit/incentive to write their answers clearly in order to save time on checking/grading. Of course this is just a hypothetical scenario considering how rudimentary this research is, but still an idea with a lot of potential considering the kind of research and technology that already exists. Also academia, like many other sectors, is a very underdeveloped sector from a data science perspective so there is definitely a lot of opportunity for research and development of ideas like this.

Methods:

To begin, the first and foremost thing was to collect and load the MNIST data that already exists in keras. This dataset will serve as the base for all the model training, validation and testing in the assignment. The 70,000 images retrieved images will split into a set of 60,000 training images and 10,000 test images. Before training the data it needs to be preprocessed by reshaping it into the shape that the network expects and scaling it so that all the values are in the [0,1] interval. All the training images are in an array of shape (60000, 28, 28) of type unit8 with

values in the [0,255] interval. By rescaling it gets transformed into a float32 array of shape (60000, 784) with values between 0 and 1. The labels are transformed from (60000x1) to (60000,10) using one hot encoder since the labels are all categorical. The dense neural network in this research assignment will consist of 784 input nodes, a hidden layer with 1 node (experiment 1) and 2 nodes (experiment 2) and also 10 output nodes that correspond to the 10 digits. The validation data held back is around 10% of the training data (which is 6,000 images). In experiment 1, after the model 1 is trained, 60,000 activation values of the hidden node for the original set of training images are grouped by the 10 predicted classes and the sets are visualized using boxplot. The expectation here is the overlap between the range of values in the "boxes" is minimal. In experiment 2, a similar process is to be followed but this time with 2 nodes in the hidden layer. The output of the hidden nodes are plotted using a scatterplot for each of the 60,000 images. These predictions will be color coded to see which of the 10 classes they belong to. The color clusters here, again, are expected to have minimal overlap. On top of these visualizations, two more plots displaying training and validation loss and accuracy (for each epoch) are made using Matplotlib. Another visualization that is obtained is that of a confusion matrix for each experiment. Experiment 3 involves exploring the previous experiment(s) but this time with more hidden nodes. The goal of this experiment is to come up with 1 model that provides the highest accuracy. There are some key points for the first 3 experiments. The last layer of all the models is always a 10-way "softmax" layer that returns an array of 10 probability scores (summing to 1). Each score represents the probability that the label (digit) in inspection belongs to one of the 10 digit classes. To make these models ready for training, they need 1) loss function - measures how good a job the model does while training the data. Since the category labels have already been transformed using an encoder, "categorical_crossentropy" can be used as the loss function. 2)

optimizer - mechanism through which the model will update itself based on the data it sees and its loss function and 3) Metrics - to monitor during training and testing. In this research experiment the accuracy metrics will be used. To keep the comparisons between all the models fair, the parameters while training the data such as epochs and how much validation data is held will be kept constant. In experiment 3, a function is defined to create a DNN model with a given number of hidden layers and a fixed number of nodes per hidden layer. Using that function, a model was trained on all nodes from 1 to 500 one at a time. Later the model with x number of nodes that gave the highest accuracy was chosen and compared against models from previous experiments. Later this model was used in experiment 4 and 5. In experiment 4, PCA decomposition was used to reduce the number of dimensions of the training set of 28x28 dimensional MNIST images from 784 to 154. 95% of training images variance will be lying along these components. This experiment aims at exploring whether significantly reducing dimensions impact the validation accuracy significantly. In experiment 5, a random forest classifier is used to get the relative importance of the 784 features of the 28x28 dimensional images and select the tio 70 features to train the best model build from experiment 3. The performance of this model is then compared to the DNN from experiment 3 and 4 to see how well it performed. Just like experiment 1, in experiments 4 and 5 also the 60,000 activation values of the hidden node for the original set of training images are grouped by the 10 predicted classes and these groups are visualized by boxplots.

Results:

The highlights of the results from the 5 experiments were as follow:

- The test accuracy in experiment 1 was pretty low ~ 39%.. This was expected as there was only 1 hidden layer and only 1 node used to train the model. The validation loss was also

very high. The graphical version of the confusion matrix looked to be in a bad shape as well.

- In experiment 2, the test accuracy significantly improved compared to experiment 1 ~ 68%. But the validation loss was still higher than desired. This improvement in test accuracy signifies the importance of the number of nodes for improving the test accuracy.
- In experiment 3, the highest accuracy was achieved by using 413 nodes in the first hidden layer. The accuracy was over 98% which is pretty good. Not just that the validation loss was close to 0.1 which is very impressive.
- In experiment 4, the test accuracy did not decline even a little bit by reducing the number of dimensions. However the test loss did go a bit higher. The model ran much faster while not having any negative impact on the performance of an already high performing model obtained from experiment 3.
- In experiment 5, the test accuracy obtained was around 94% with an increase in validation loss. Although it is worth noting that the performance of the model did not drop significantly even after reducing the number of features to more than 1/10 of the fraction. This shows the significance of the top 10% features in models built similar to this research assignment.

Conclusion:

In conclusion this was a very useful research assignment which served multiple purposes such as exploring neural networks and seeing how the neurons/nodes with a single hidden layer impact ML models for classification of images. Future business prospects in the academia sector using this research were also explored in this assignment. This research assignment can be furthered by finding the most optimal/time saving ways to get the highest accuracy.