**ASSIGNMENT -5**

**TEXT CLASSIFICATION**

Trilok Palla

Northeastern University [College of Professional Studies](https://cps.northeastern.edu/)

**ALY 6020- Predictive Analytics**

Justin Grosz

**INTRODUCTION:**

Character processing handwriting is a pattern. A person’s handwriting is defined by several factors. Analysis of hand-written notes will require several features in consideration like spacing between characters, a tilt of characters to right or left, style of writing, font, the background of the handwritten notes i.e., ruled paper, plain paper, etc, the color of the notes i.e., red, black, or blue, etc.

**About K-Nearest Neighbour Algorithm:**

KNN is a machine learning algorithm that doesn’t need any training of data. The KNN classifier relies on the distance between feature vectors and also, classifies unknown data points by finding the most common class among the k closest examples.

Each data point in the k closest cast a vote and the highest category number of votes wins.

**About Neural Networks:**

A convolutional neural network (CNN) is a type of artificial neural network used primarily for image recognition and processing, due to its ability to recognize patterns in images. A CNN is a powerful tool but requires millions of labeled data points for training. CNNs must be trained with high-power processors, such as a GPU or an NPU if they are to produce results quickly enough to be useful.

**About the data:**

This is an MNIST dataset. It consists of 42,000 data points with 46 features. These 46 features consist of 1 label and the rest of the pixels. With the help of these features, we are going to test the accuracy of predicting handwriting using the KNN model and Neural Networks

**Libraries Used:**

* Numpy
* Pandas
* Matplotlib
* Tensorflow
* Sklearn methods
* Keras
* CV2

**Data Cleaning/Pre-processing:**

* There were no NULL values and duplicates found on the given dataset.
* 25 percent of the data is used for testing and the rest is used for training models.

**Data Visualization:**

* A Count plot of each of the digits(label) has been plotted to get a better understanding of the target variable.
* A few sample plots of the MNIST data have also been plotted to get a basic idea of handwritten digit data.
* All the plots are attached in the appendix.

**KNN Model:**

* The ‘label’ variable has been set as the target variable.
* An additional 10 percent of training data is used for validation.
* Given below is the summary of data points,
* training data points: 28350
* validation data points: 3150
* testing data points: 10500
* Testing and training data have been normalized before fitting the model
* KNN model has been trained with k values varying from 1 to 30, with a step of 2.
* Given below are the 15 accuracy values that have been observed alongside their corresponding K values,

Text

Description automatically generated with medium confidence

* The KNN model is found to have its highest model accuracy **at k=13, with 66%.**
* A few predictions made by the model have also been made which are depicted below,

Graphical user interface, application

Description automatically generatedA picture containing chart

Description automatically generated

A picture containing chart

Description automatically generatedA picture containing graphical user interface

Description automatically generated

A picture containing waterfall chart

Description automatically generated

**KNN model challenges:**

* KNN is not a Machine Learning based model, hence it is easier as it only requires the number of categories and the distance. The limitation of this model is that it cannot make predictions for rare or unique values in the whole population.
* Although KNN produces good accuracy on the testing set, the classifier remains slower and costlier in terms of time and memory. It requires large memory for storing the entire training dataset for prediction.
* Furthermore, Euclidean distance is very sensitive to magnitudes, hence features in the dataset that has high magnitudes will always weigh more than their counterparts with low magnitudes.
* Finally, we should keep in mind that KNN isn’t suited for large-dimensional datasets with all we’ve mentioned above.

**Neural Network Model:**

* Similar to KNN, the target variable has been set as ‘label’ and the dataset is split into a 75:25 split ratio with 10 percent of the training data set for validation purposes.
* Data should be normalized before interpreting any neural network model so that it speeds up learning and leads to faster convergence. The same has been done by dividing the train and test data by 255.0
* Ideally, to use convolution neural networks, the complete pixelated data of 784 (28\*28) columns would be required. Since our data set consists of only 45 pixelated columns, the data has been reshaped accordingly as given below,

Graphical user interface, text, application

Description automatically generated

* Data needs to be prepared in specific ways before applying it to a neural network model. As such, one hot encoding is used here to target variables to convert numerical categorical variables into binary vectors.
* A sequential Neural Network model is built with few layers. The first layer is the Convolution layer followed by a max pooling layer. These 2 are the first basic layers that can help the model for feature extraction of image and the pooling layer helps in reducing dimensions of feature maps. Pooling also helps summarize the features present in a region as it reduces the amount of computation performed.
* The next layer is a dropout layer with 0.25 random nodes dropping out. This is done so that the model would not be dependent so much on only a few sets of nodes in the hidden layer. Then we flatten the intermediate outputs (fully connected CNN) before adding 2 dense activation layers of ‘Relu’ and ‘SoftMax’ activation functions.
* Given below is a basic summary of the network,

Table

Description automatically generated

* Adams optimizer is used for compiling and a categorical cross-entropy is used to measure loss with a learning rate of 0.001 and accuracy as its metrics.
* Our handwritten digit dataset needs to be artificially expanded to prevent the overfitting issue.
* To replicate the digit variances, minor changes have been made to the training data in form of data augmentation.

Text, letter

Description automatically generated

* The model has been fitted with 25 epochs and with an ideal batch size of 30. Given below are the results of this implementation,

Table

Description automatically generated

* The best **accuracy** score is observed to be **57 percent** with a validation score of up to **63 percent.**
* Given below plots give us a better understanding of loss and accuracy parameters,

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

* The above accuracy graph clearly depicts how accuracy is steadily increasing with an increasing number of epochs
* The Loss graph depicts how the loss function decreases steadily with an increasing number of epochs.
* Given below depicts the validation loss. It can be seen how much decline can be seen with increasing epochs.

Chart, line chart

Description automatically generated

* A confusion matrix has also been plotted to depict the true and predicted label,

Table

Description automatically generated

**Neural Networks (CNN) model Challenges:**

* A large part of the dataset is needed for training data is needed for the CNN to be effective and they fail to encode the position and orientation of objects.
* They have a hard time classifying images with different positions.
* CNN's tend to be much slower because of operations like Maxpool.
* In case the convolutional neural network is made up of multiple layers, the training process could take a particularly long time if the computer does not have a good GPU.
* Convolutional neural networks will recognize the image as clusters of pixels that are arranged in distinct patterns. They don’t understand them as components present in the image.

**MODEL COMPARISON:**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Time | Loss |
| KNN Classifier | 66% | 18.75 Seconds (for all K values) |  |
| Convolution Neural Network | 57% | 400 seconds (25 Epochs) | 1.184 |

It is clear from these comparisons that **KNN Classifier** is the ideal model fit for this data set. I would suggest the school go ahead with the KNN classifier, since it is a much simpler model, and with low data points and variables it has proven to be more accurate. Usually, Neural networks have an upper hand in any handwritten recognition but the pixel data which is being produced is very limited for CNNs to show their full potential.

**Appendix:**

Chart, bar chart

Description automatically generated

*Fig1: Count plot of total digits*

Shape

Description automatically generated Icon

Description automatically generated with low confidence

*Fig2: Digit 1 display from pixel Fig3: Digit 4 display from pixel data*

**References:**

<https://medium.com/@muhammadshoaibali/flattening-cnn-layers-for-neural-network-694a232eda6a>

<https://towardsdatascience.com/optical-character-recognition-with-knn-classifier-10fd220ed797>

Thank you,

Trilok Palla