Comparing the DataSet V/s Loss on EMNIST DataSet

DEEPAK MITTAL

Machine Learning Engineer Intern

AI Technology and System

deepakmittalbutjevra@gmail.com

www.ai-techsystems.com

***Abstract*—Handwriting recognition is the practical application of Machine Learning and it is a widely used application of it. If the model is trained using lots of images of handwritten digits, it can be generalized for unseen patterns. Modern Technologies like IOT, Big Data, Computer Vision, etc. are helping our society to collect data and find useful patterns out of it. Using that data, it is easy to find patterns, and useful insights using Machine Learning and Artificial Intelligence.**

INTRODUCTION

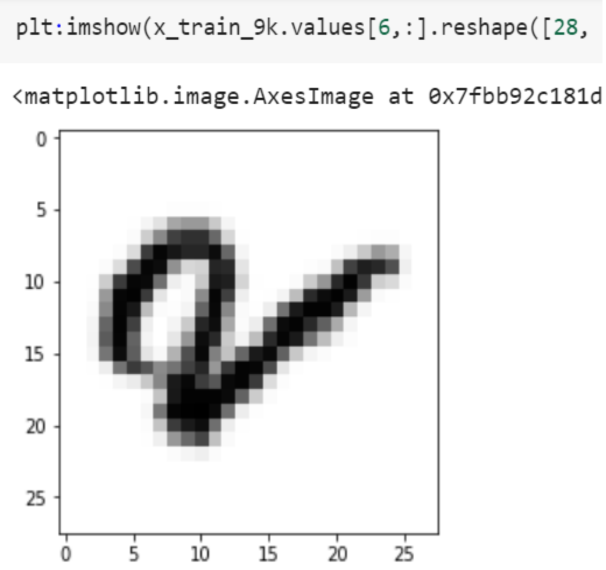
In recent years, the Machine Learning application is being widened to each corner of our society. Machine Learning is benefitting our society in many aspects. Handwriting recognition is the useful application of Machine Learning, Computer Vision and Artificial Intelligence and many parts of our society is continuously benefitting of it. The neural network uses the examples to automatically infer rules for recognizing handwritten digits and by increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy. Modern Technologies like IOT, Big Data, Computer Vision and many more are helping in collecting data and of a good quality. Cloud Computing is benefitting our society in storing data and to compute result into it.

Advances are taking place everyday in Artificial Intelligence, Computer Vision, Machine Learning, Deep Learning and many more. Recognizing handwriting is an easy task for humans but a daunting task for computers and hence computers require lots and lots of data to train first and then make prediction.

DATASET AND LIBRARY

There are many types of characters like Capital symbols, Small symbols, special symbols and many more and in order to achieve a good accuracy, computers need to train either on large dataset or a good volume of dataset with deep neural networks. The dataset used to determine dataset size v/s loss is EMNIST and it is an extension of MNIST. It can be used to train the systems for the classification and the computer vision. It contains hand written digits of size 28x28. EMNIST dataset is an open-source dataset at Kaggle. It contains lakhs of sample to properly train the system and hence test the system.

The size of the EMNIST dataset is about 1.2 GB for the complete dataset. I had loaded it into the Google Collaboratory directly from the Kaggle using API token generated. Once the API token is generated it can be link to the Collaboratory by uploading a .json file. After that, I had unzipped the file and selected two files namely *emnist-byclass-train.csv* and *emnist-byclass-test.csv* in which one is used for training and other for test the model.



```
plt:imshow(x_train_9k.values[6,:].reshape([28,
```

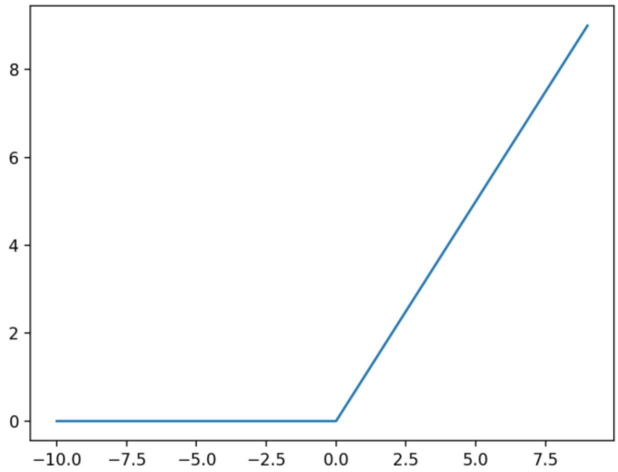
```
<matplotlib.image.AxesImage at 0x7fbb92c181d
```

The dataset contains 697k+ samples for training. This large dataset is split into 9K, 27K, 80K,240K, and 697K in order to compare the Dataset V/s Loss for various Dataset sample size. The dataset is split using pandas .iloc method. Then the Datasets are normalized by dividing with 255. The shape of the training features Datasets are now (9000, 784), (27000, 784), (80000, 784), (240000, 784), and (697931, 784) respectively and for labels are (9000, 47), (27000, 47), (80000, 47), (240000, 47), and (697931, 47) respectively.

ACTIVATION FUNCTION

ReLU

Rectified Linear Activation Function is use to train deep neural networks, an activation function is needed that looks and acts like a linear function, but is, in fact, a nonlinear function allowing complex relationships in the data to be learned. A node or unit that implements this activation function is referred to as a **rectified linear activation unit**, or *ReLU* for short. Often, networks that use the rectifier function for the hidden layers are referred to as rectified networks.



SOFTMAX

Softmax is a function that takes as input a vector of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. Some vector components could be negative, or greater than one; and might not sum to 1 but after applying softmax, each component will be in the interval (0,1), and the components will add up to 1, so that they can be interpreted as probabilities.

```
softmax = np.exp(z)/np.sum(np.exp(z))
```

LOSSES

Categorical cross entropy

Categorical crossentropy will compare the distribution of the predictions (the activations in the output layer, one for each class) with the true distribution, where the probability of the true class is set to 1 and 0 for the other classes. To put it in a different way, the true class is represented as a one-hot encoded vector, and the closer the model's outputs are to that vector, the lower the loss.

LIBRARY USED AND PROGRAMMING LANGUAGE

Libraries used are as follows-

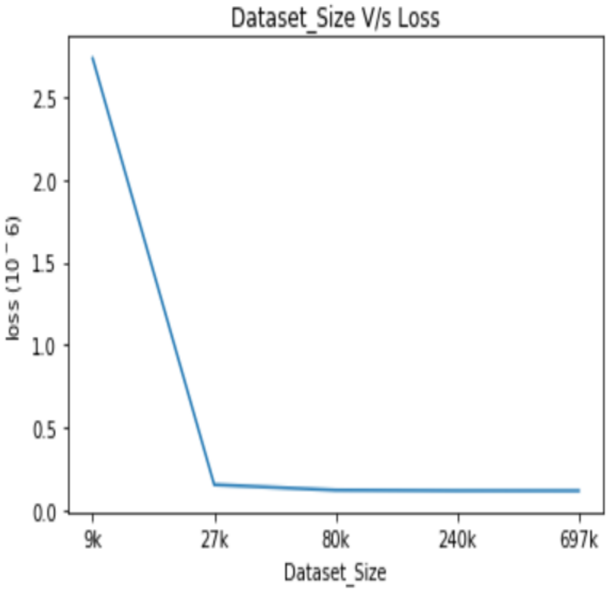
- Matplotlib.pyplot
- Numpy
- Pandas
- Keras
- Google.colab.files

Programming language-

- Python 3

FIGURES AND TABLES

Dataset_Size	**Loss**
9K	2.732704e-06
27K	1.561236e-07
80K	1.229859e-07
240K	1.197924e-07
697K	1.193159e-07



OPTIMIZER

Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models.

NUMBER OF EPOCHS

epochs = 10

CONCLUSION

Generally, it is found that the accuracy of a model is less because of training it with little data. However, if more data is used to train the same model, a significant improvement is observed in its accuracy for predicting the result on test data.
If the model is trained on a very small dataset, it may result in under-fitting. However, if the model is trained on very large dataset, it may result in over-fitting. Hence, a optimal amount of data should be used to train the model.
In the above task, the dataset is splited into 9k, 27k, 80k, 240k, and 697k samples for training of the model. It is found that the value of the loss decreases with increases in the dataset-size.

REFERENCES

[1] https://ieeexplore.ieee.org/document/8473291

[2] http://cs231n.stanford.edu/reports/2017/pdfs/810.pdf

[3] https://keras.io/models/model/

[4] https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[5] http://neuralnetworksanddeeplearning.com/chap1.html

[6] https://en.wikipedia.org/wiki/Softmax_function

[7] https://keras.io/losses/

[8] https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy

[9] https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/