How Different Data Sets Affect Activation Function Error Rates

By Corey Domingos

1. Introduction

Significance: Image Recognition when specifically applied to Autonomous Driving vehicles is an extremely controversial topic. Since these applications have the power to drive a vehicle, it is imperative that the developing team utilizes the proper tools when creating such applications. If the structure of the Neural Network that drives said Image Recognition applications implemented in Autonomous Driving vehicles improperly classifies an image it could have grave consequences for those inside the vehicle and in the surrounding area. <u>The objective of this Research Study is to highlight how certain data sets affect Convolutional Neural Network Activation Function error rate percentages.</u> In regards to Autonomous Driving vehicles properly choosing an Activation Function that is proven to have the lowest error rate for a particular data set is a crucial aspect of the development process and this Research will act as a baseline for such applications.

Background:

Image Recognition software is widely implemented using Neural Networks, more specifically Convolutional Neural Networks. A Convolutional Neural Network, which will be referred to as a CNN throughout this proposal, “Consists of hidden layers having convolution and pooling functions [1]”. The main difference between a CNN and a traditional ANN (*Artificial Neural Network*) is the Convolutional Layer which simply multiplies then sums up the input data into a simpler form for the rest of the network to utilize. There are many different types of network layers which can be implemented within CNN’s, and the ones used in this Research Study will be highlighted in the *Methodology* section. Although CNN’s have “Been around for the past 50 years [2]” Computer Scientists have recently made vast advances in Image Recognition applications in respect to lowering error rate percentages. In order to effectively and efficiently lower error rate percentages when developing a CNN it is necessary to research which Activation Function best suits your data set.

In a research study done by three students at *SASTRA University, Kumbakonam, India* (P.SIBI, S.ALLWYN JONES, P.SIDDARTH) it was found that certain activation functions work best for the dataset that they tested on. An Activation Function is what tells the network to “fire” or not. In most cases the activation function will return either a 0, which means no action is taken. Otherwise the Function will return a 1, and in that instance the network has properly classified an image. In the study mentioned above their lowest error rate percentages were achieved with “Sigmoid and Linear [3]” with their error rate percentages in the last epoch being “0.0003930641 and 0.0063356720 [3]” respectively. This research study done by these students will act as a baseline for this study, but will be expanded upon by applying multiple datasets.

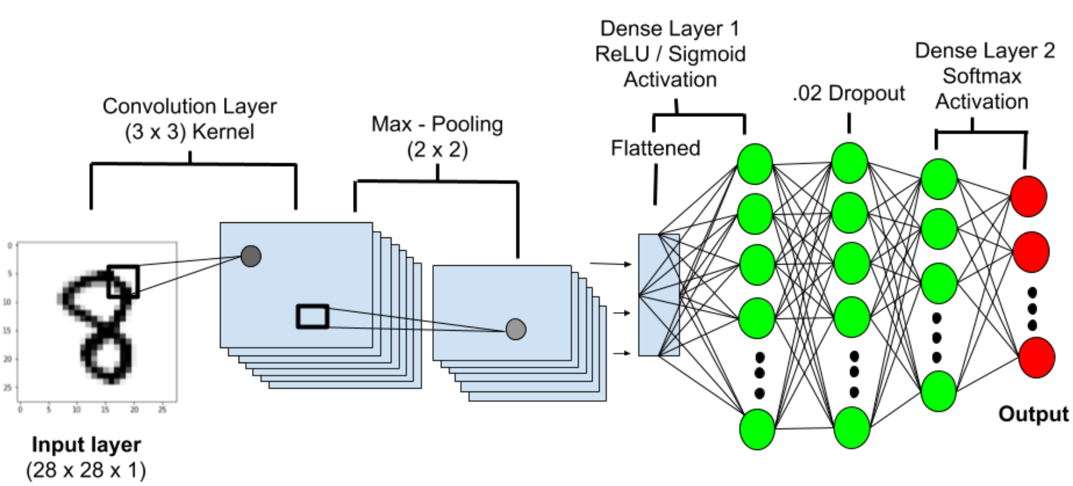
The datasets that will be experimented on are the MNIST and FASHION_MNIST datasets due to their popularity within the data science community. The FASHION_MNIST data set was chosen because of a study done by Michael McKenna where he suggests that a “Replacement for MNIST is entitled Fashion-MNIST [4]” due to the overwhelming simplicity of the MNIST dataset. These datasets will be depicted in detail in the *Methodology* section of this study. *In this Research study a comparison between the performance of the Sigmoid and Relu Activation functions when applied to different data sets will be highlighted*. The performance of each Activation function in the 10th Epoch will be measured using tables to show the difference in error rate percentages.

2. Methodology

The tutorial offered by Towards Data Science “Image Classification in 10 Minutes with MNIST Dataset [5]” was followed to create the CNN network. The structure of the CNN was not altered from the tutorital mentioned above due to time constraints, but will be explained so the reader can understand what function each layer serves. A diagram has been included to visualize the structure of the CNN, which is included in *Fig 1*. The first layer is the input layer and it will be explained further in the *Results* section of this paper. The second layer is a Convolution layer which as described in the Background section of this paper reduces the number of inputs. The reduction of the data in this network is (3 x 3) Kernels. The next layer is a pooling layer, Max-Pooling, which takes the largest value from each Kernel square. These 2D arrays are then flattened so they can create a fully connected network.

At this point, one of the main focuses of study is implemented, the Activation Function. In the first Dense layer the *Sigmoid* or *ReLU* Activation Function is used. The change in Activation Function is achieved by altering the parameters being passed through the first Dense layer. In between the two dense layers there is a Dropout Layer which helps the CNN not overfit the datasets. Lastly there is a second Dense Layer which uses the *Softmax* Activation Function. The effects of implementing and altering the first Dense Layer will be depicted in the *Results* section of this paper, and the second Dense Layer in the *Discussion* section.

Figure 1. *CNN Structure*



In the Towards Data Science tutorial referenced above the data set that was implemented was MNIST. The MNIST data set is a collection of 60,000 grayscale handwritten digits, and it was chosen to be one of the two datasets tested on in this study due to it being offered in the tutorial. Additionally the MNIST data set acts as a baseline for most Image Recognition applications due its widespread use and simplicity so it only seemed right to stick with it for this research study. The second data set that was implemented was the FASHION_MNIST, which as stated in the *Background*, acts as a good replacement to the MNIST data set. The FASHION_MNIST differs from MNIST by being a collection of 60,000 grayscale clothing images instead of digits. Both the MNIST and FASHION_MNIST data sets are offered from a library called *TensorFlow*, which is a very well known name among the Data Science community. A common trend between both of these data sets is that they are both grayscale. Training a CNN on grayscale images vs colored images is much easier, and implementing colored images would just add another variable to be considered when comparing the results. The application used for this project was developed using Python in a Jupyter Notebook, and the source code can be found at *https://github.com/coreydom/CNN-Research*.

3. Results

The MNIST dataset, as stated above, is a collection of handwritten digits. *Figure 2* is an example of one of these 60,000 digits that could have been chosen to train the CNN on. The digit can be changed within the source code by altering just one line of code *'image_index = 50555'*. Any index point up to 60,000 can be chosen which allows for easy testing of different images. *Figure 3* is an example of the FASHION_MNIST data set, and the current image can also be changed through that one line of code found above.

Figure 2. *MNIST Image*

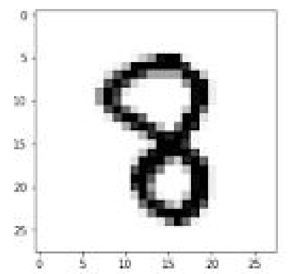
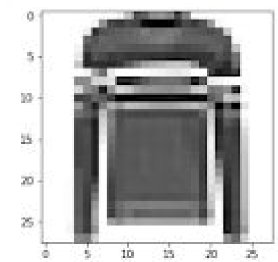


Figure 3. *FASHION_MNIST Image*



The results of this research study did prove that one Activation Function does work best for a given dataset. When testing on the image found in *Figure 2* the best performance was achieved with the RELU Activation Function. For *Figure 3*, the lowest error rate was also obtained through the RELU Activation Function. A comparison between the error rates in the 10th epoch can be viewed in *Table 1*. A comparison between the functions at the time of application creation and testing can be found at the *Github* link posted above.

Table 1. *Error Rates in 10th Epoch*

	RELU	**Sigmoid**
MNIST	0.0172	0.0179
FASHION_MNIST	0.1159	0.1350

The MNIST data set had great performance with low error rate percentages but as spoken about in the background the MNIST data set is known for being almost too simple to do research with. Without adding the Dropout layer in between the two dense layers the accuracy of the CNN was too high to be logical. In order to change the Activation Functions used in this study only one line of code needed to be changed. The parameter in the first Dense Layer line '*model.add(Dense(128,activation=tf.nn.relu))*' that needs to be changed is the '*tf.nn.relu*'. By only altering this line, and no other aspects of the CNN structure when testing the activation functions a true result was obtained with no other variables being added.

4. Discussion

The methods used in this Research Study were implemented in such a manner that it is easily repeatable and could be expanded on in future studies. There are three major areas that this study could be expanded on, the first of them being the implementation of more Activation Functions. In the study done by the students at *SASTRA University, Kumbakonam, India* (P.SIBI, S.ALLWYN JONES, P.SIDDARTH) which was referenced in the *Background* section they tested "12 different functions [3]". Testing 12 Activation Functions on multiple data sets would be a lengthy task, but in order to further solidify this research implementing as many functions as time allows would be best. If only one more Activation Function could be added to this study, the TANH function would be chosen. The TANH function "Is basically a shifted sigmoid neuron[6]" and would be chosen due to its popularity among Data Scientists. Branching off of that topic brings up the next area for expansion which is testing on more data sets. The MNIST and FASHION_MNIST data sets are relatively easy to work with and testing on more complex, or even colored data sets would help prove the results further. A good example for a data set that would be a good fit for this expansion on this study is the *Stanford_Dogs* offered by *Tensorflow*. This data set "contains images of 120 breeds of dogs from around the world [7]". To implement this data set the CNN would need to be altered to accept the colored images, but in combination with the adding more Activation Functions the results could have more of a real world application.

Lastly the third major limitation of this research study is the effect of the second Dense layer that was highlighted in *Figure 1*. In this layer the Activation Function stayed the same throughout all tests, which was the *Softmax* function. The *Softmax* function allows the "Neural network to be able to determine the probability that the dog is in the image [8]" if applied to the *Stanford_Dogs* data set. Without this function being implemented in the last layer the error rate percentages are extremely high due to the need for the "Squashing function [8]" which "limits the output of the function into the range 0 to 1 [8]". Making changes to this layer would alter the error rate percentages and would act as a good extension of this study.

In this study the thesis <u>how certain data sets affect Convolutional Neural Network Activation Function error rate percentages</u> was tested and proven to have tangible results. When testing the MNIST and FASHION_MNIST data sets across the same CNN structure while changing the Activation Functions a "best case" was achieved. The *RELU* Activation Function had the lowest error rate percentage in the 10th epoch on both data sets. By completing this research the further studies depicted above will be able to be experimented on to further solidify the results achieved through this study.

Bibliography

[1]

vibhor nigam, “Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning,” *Medium*, Feb. 11, 2020. https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90 (accessed Apr. 13, 2020).

[2]

S. Hijazi, R. Kumar, and C. Rowen, “Using Convolutional Neural Networks for Image Recognition,” p. 12.

[3]

P. Sibi, S. A. Jones, and P. Siddarth, “ANALYSIS OF DIFFERENT ACTIVATION FUNCTIONS USING BACK PROPAGATION NEURAL NETWORKS,” . *Vol.*, vol. 47, p. 5, 2005.

[4]

H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms,” *arXiv:1708.07747 [cs, stat]*, Sep. 2017, Accessed: Apr. 07, 2020. [Online]. Available: http://arxiv.org/abs/1708.07747.

[5]

O. G. Yalçın, “Image Classification in 10 Minutes with MNIST Dataset,” *Medium*, Mar. 22, 2020. https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d (accessed Apr. 09, 2020).

[6]

D. Dholakia, “Activation Functions,” *Medium*, May 08, 2018. https://towardsdatascience.com/activation-functions-b63185778794 (accessed Apr. 13, 2020).

[7]

“stanford_dogs | TensorFlow Datasets,” *TensorFlow*. https://www.tensorflow.org/datasets/catalog/stanford_dogs (accessed Apr. 13, 2020).

[8]

“Softmax Layer,” *DeepAI*, May 17, 2019. https://deepai.org/machine-learning-glossary-and-terms/softmax-layer (accessed Apr. 13, 2020).