CSE 575 - Statistical Machine Learning Classification Using Neural Networks and Deep Learning Ira A. Fulton School of Engineering, Arizona State University

AJAY KANNAN
ASU ID : 1219387832

***Abstract*—The primary goal of this project is to create a convolution neural network with a deep architecture. The proposed structure of the neural network uses multiple convolutions with a convolutional kernel of 5x5 and a dense layers of fully connected network to predict the output. In between the convolutions, two pooling layers 2x2 with a stride of 2x2 is given for to reduce the dimensions of the feature maps. In the entire network, the activation function is rectified linear unit, except for the final activation function, which is going to be softmax. The data set is the SVHN which consists of printed digits cropped from the house plate pictures belonging to 10 classes. This structure yields 91.3% accuracy for the testing set. This portfolio is an individual project where I have selected the third project for this course.**

Keywords — Convolutional neural networks, pooling layer, activation layer, Saturation

I. INTRODUCTION

From late twentieth century, neural networks are heavily researched on. The world is at the point where virtually all technologies rely on computing. And nowhere is the importance of computing more pronounced than in the arena of artificial intelligence. The adoption of GPU (graphic processing units) in general purpose processing enabled AlexNet[2], a convolutional neural network (CNN)[3] running on GPUs implemented in CUDA, to win the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge, aka the Olympics of computer vision), by a huge advantage in accuracy. And now, everything is AI-ed. The cutting edge technology came of age in 2016, and its impact has intensified this year. Even Microsoft admitted the technology is not yet mature enough. Why? Current computing capabilities cannot keep up with the cost and energy demands made by increasingly complicated AI operations. However, the AI revolution still has a long way to go.

This project is a simplified version of the very famous AlexNet - convolution neural network.

II. ABOUT THE DATA

A. Data set

The Street View House Numbers (SVHN) Dataset[1] is a real-world image data set for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. It is similar to MNIST dataset[4] (e.g., the images are of small cropped digits), but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem (recognising digits and numbers in natural scene images). SVHN is obtained from house numbers in Google Street View images. See Fig 1 to view the sample of the data.

B. Input and Output data

The total number of samples is split into training and testing set. Training images are 73,257 in number alone with the output classes. The testing images are 26,032 in number and comes alone with its output classes for verification of the algorithm. Each image is off resolution of, 32 x 32 x 3, where 3 represents with three separate channels of RGB. Output is encoded in a one-hot fashion for better performances.

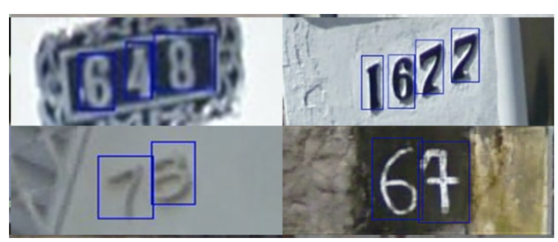

Fig. 1: Sample data points

III. METHODOLOGY

The image classification is done through a supervised learning approach. We use deep learning and convolutional neural network (CNN) to train and test the network. In this project, the task is to build a CNN for doing a classification. The data set contains SVHN which are images of digits cropped from the house plate pictures belonging to 10 classes.

A. Convolutional Layer

The convolutional layer (Conv2D) is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels). During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the filter entries and the input, producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

The filter which is used here is 5x5 and it learns as the network trains. The activation function here, is the rectified linear unit or ReLu. The equation 1 gives the ReLu function,

$$f(x) = max(0, x) \tag{1}$$

where, x acts as a switch.
Also, Stride controls how depth columns around the width and height are allocated. The stride here is 2x2, which means greater overlap of receptive fields and greater spatial dimensions of the output volume.

B. Pooling layer

One important concept of CNN is pooling, which is a form of non-linear down-sampling. There are several non-linear functions to implement pooling, where max pooling is the most common and this architecture uses this. It partitions the input image into a set of rectangles and, for each such sub-region, outputs the maximum. The equation of the set of rectangles is defined as,

$$f_{X,Y}(S) = max^{1}_{a,b=0} S_{(2X+a,2Y+b)} \tag{2}$$

C. Fully connected network

After flattening, the network becomes easier to work with the nodes. Then the most common approach is to predict the output after passing it through a fully connected network. Here too, ReLu is used to make the network less denser. The output is calculated through a softmax function which is the last layer of the network. The loss is calculated through categorical cross-entropy.

D. Parameters

1) Number of filter are five before the fully connected network. 3 layers of conv2D with two output layers having 64 output nodes and the last layer having 128 nodes.
2) Filter size is another property which determines the granularity of the data set. The challenge is to find the right level of granularity so as to create abstractions at the proper scale, given a particular data set, and without overfitting. 5x5 is chosen as the filter size.
3) Learning rate is also a variable and different learning rates will give different values.

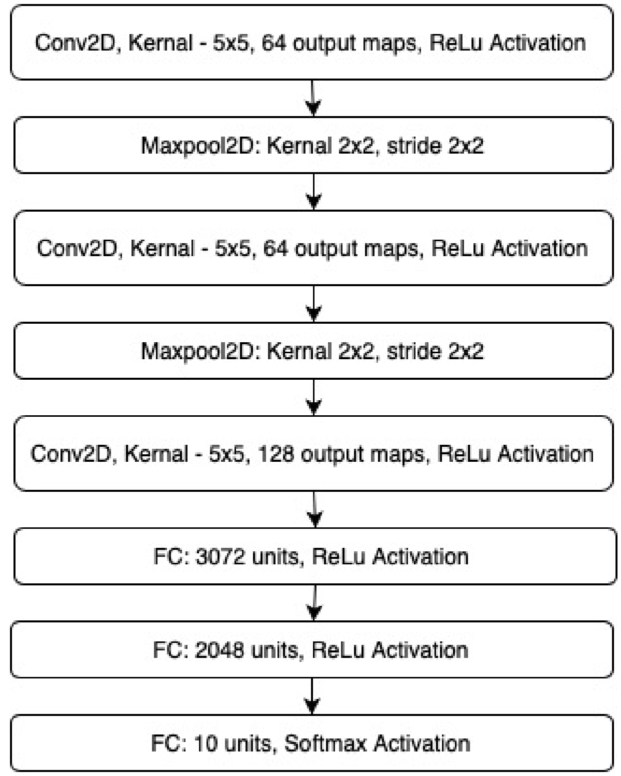

Fig. 2: System Flowchart

E. Working of the network

In the internal working of the network, they are being split. The first layer is trained with a convolution, Conv2d of shape of 5x5 and an activation layer which is ReLu unit. This is followed by max pooling layer with stride 2. The process is repeated and lastly a convulsion which is mapped to 128*16 nodes flattened layer. This fully connected layer is periodically reduced to give 10 nodes with the last Softmax layer. The output is mapped with a Stochastic gradient descent optimiser and the loss is calculated based on categorical cross-entropy (using one-hot vector for the output nodes). The system flowchart is given in Fig 2.

IV. Observations

The system is trained and tested for 20 epochs. The graphs are plotted for the loss incurred in the system and the accuracy of the both training & testing data. The model is graphed with each epoch versus loss and accuracy in Fig 3 and 4.

A. Loss

When one increases the training, the loss is also increased with respect to the testing line. As mentioned earlier, this

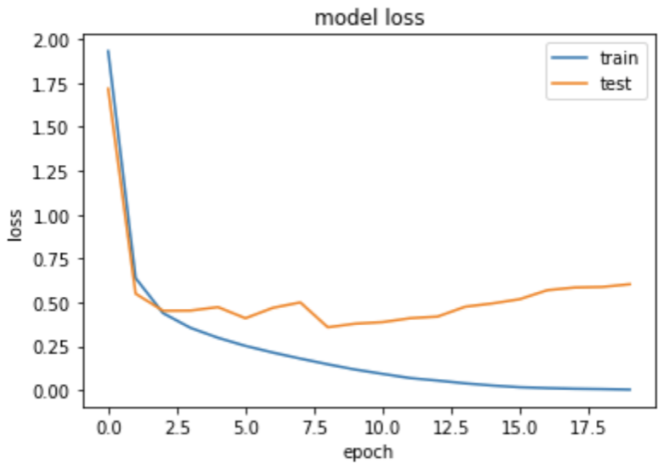

Fig. 3: System Loss Graph

symbolises overfitting which can clearly cause error in the system (Fig 3).

B. Saturation

The accuracy graph (Fig 4) shows a static line while testing towards the end. This symbolises the saturation of the network.

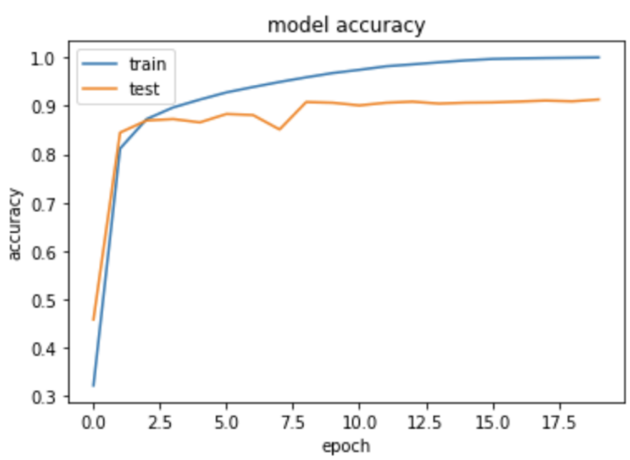

Fig. 4: System Accuracy Graph

C. Alternative parameters

1) Learning rate: Many number of learning rates were tested. The learning rate is most accurate at 1e-2. Some of the alternate learning parameters which were tested are 1e-3, 1e-4 and 1e-1.

2) Optimisers: Many optimisers are there and out of which, SGD and Adadelta give positive results. The results are tested for 2 to 50 epochs for both SGD and adadelta. The best results were for SGD with testing accuracy of 91.3%.

3) Dropout: Normally one uses dropout to regularise the network and to avoid overfitting quickly. As this is a smaller dataset, the dropout is not required.

D. Results

The Training Evaluation for the training twenty epoch had come as 99.99%. The loss was a minimum of .23%, which means that the system has over-trained the data set.

Interestingly the same cannot be said for the testing set as an accuracy of 91.30685% was verified. The loss of the testing set was 61.30%, which means, the data is learnt, such that it is over fitting.

V. My Learnings

This is the final project on statistical machine learning where we have to individually develop a convolution neural network to classify an image data set with efficiency. It is a self learning network which can classify images specifically numeric data set. The data set was given to us and I had to view the data to get an idea on how the data works. First I learnt that I should normalise the data to reduce the data redundancy and improve the data integrity. To put it simple, the data must be under one roof.

Then, I changed the encoding of the data output. An one-hot encoding allows the representation of categorical data to be more expressive. Many machine learning algorithms cannot work with categorical data directly. The categories must be converted into numbers. This is required for both input and output variables that are categorical. Then we build the model as per the architecture mentioned above.

Doing this project I learnt quite a number of things. For instance, in the neural network context, the phenomenon of saturation refers to the state in which a neuron predominantly outputs values close to the asymptotic ends of the bounded activation function. Saturation damages both the information capacity and the learning ability of a neural network. But the saturation limit help us to identify the number of epochs where the training phase must stop.

I played with the various parameters of the system. For example, as it is said in the report, I changed the optimisers and learning rates. For learning rate, you have to find the global maximum or minimum to get the maximum accuracy. Also I learnt that some networks take days to run. So you have to be careful to select the appropriate parameters which comes with practice.

Many different types of parameters are there in the neural network and Machine Learning in general. We have to find the right balance of those parameters to find the best possible efficiency and predict the right output. In the class we learnt concepts like logistic regression, sigmoid function, artificial neural networks, the basics of CNN, PCA, Unsupervised learning etc. I hope to learn more about machine learning and deep learning in the future.

ACKNOWLEDGMENT

I should take this opportunity to thank Professor Baoxin Li and the course instructor Nupur Thakur for guiding me to learn this subject and helping me to finish the projects.

REFERENCES

[1] Data set - http://ufldl.stanford.edu/housenumbers/

[2] AlexNet - https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541–551, Winter 1989.

[4] MNIST Dataset - http://yann.lecun.com/exdb/mnist/