Gurjus Singh

MSDS 458 Artificial Intelligence and Deep Learning

February 7th, 2021

Week 5: A.2 Second Research/Programming Assignment

**Abstract**

In this research I explored the CIFAR 10 Dataset which was used to train Convolutional Neural Networks and Deep Neural Networks. The CIFAR Dataset consists of 60,000 images of 10 classes. This was used for multiclass classification supervised learning. In the assignment I did a total of 15 experiments and compared the results of each experiment. I also did T-SNDE on the best model and explored how Convolutional Neural Networks learn.

**Introduction**

In this research that I conducted; I used the CIFAR10 dataset. The dataset consists ten classes labeled airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. With this dataset the goal was to experiment using Convolutional Neural Networks and do multiclass classification. The dataset is composed of 60,000 images, and I wanted to see the outputs of the layers of the Convolutional Neural Network as well as perform T-SNDE. After experimenting CNNs I then compared each Neural Network and made a management recommendation.

**Literature Review**

Several Researchers have used the CIFAR-10 Dataset. The data was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton [1].

One study done by Akwasi Darkwah Akwaboah titled “Convolutional Neural Network for CIFAR-10 Dataset Image Classification” explores the use of CNNs in classification of CIFAR-10 dataset [2]. Akwaboah mentions Alex Krizhevsky who developed the AlexNet architecture which obtained record performance metrics for the CIFAR-10 dataset [2]. He then explains the magic behind CNNs saying that they extract “higher level representation of image features” [2]. They are also good at narrowing network parameters [2] .

The researcher went on to play around with CNNs with the CIFAR 10 dataset. He made 3 Convolutional networks and 2 of them were prone to overfitting [2]. For this third experiment he used L2 Regularizer and dropout of 50 percent and trained on more epochs to 40 to raise accuracy which increased to 75 percent to test data [2].

**Methods**

In this research I first started out by importing packages such as Sklearn for providing metrics for each model, Tensorflow and Keras for making the CNNs and DNNs. I also imported Numpy for using arrays, pandas for DataFame purposes, and matplotlib, seaborn for data visualizations. The packages used are seen in 1-1. Graphical user interface, text, application

Description automatically generated

*Import Packages 1-1*

I first checked the versions of the Tensorflow packages to make sure I was using 2.0 or above of Tensorflow. Next, I downloaded the CIFAR-10 dataset and split it into train, and test. I check out how many images in Train and Test which were 50,000 and 10,000 respectively. I also looked at the labels to see what they were of. I then plotted some of the pictures and got their labels as seen in 1-2. After I looked at the dataset, I normalized the dataset pixels by dividing by 255 so each pixel value was between 0 and 1. Once I normalized the dataset, I split up the dataset further into the validation set. The validation set contained 10,000 images. *Graphical user interface, website

Description automatically generated*

*Picture Previews 1-2*

Once I was done it was time to create each model and experiment. I created a total of 15 experiments. I used techniques such as L1 and L2 regularizers, dropout and early stopping which are all considered regularization techniques. What was special about this research was that I created CNNs which is good at finding features from images. CNNs involve max pooling and convolution layers. Max pooling help in reducing the number of parameters in the network by “downsizing feature maps” [3]. The convolutional layer functions in that tries to extract local patterns from a window [3]. The window is called a filter which goes over each part of the image extracting the features in the image [3].

For the models I used 64 neurons for my first two experiments using Deep Neurla Networks while my 3rd and 4th experiments were CNNs where there were 2 Max Pooling/Convolution Layers and 3 Max Pooling/Convolution Layers respectively. I then applied regularization techniques such as Early Stopping, L1/L2 Regularizers, and Dropout after the first 4 experiments.

Later after implementing the models I took a look at the accuracies to find the best model. I believe that Experiment 16 performed the best by looking at the results. I deep dived into Experiment 16 by also looking at other metrics such as Confusion Matrix and Precision Score. I also performed T-SNDE on this model shown in 1-3. For Experiment 3, I wanted to know how the CNN worked so I extracted the outputs shown in 1-3, 1-4. What I got out of 1-3, 1-4 was it was extracting stripe like features which were diagonal lines.

*A picture containing text, sign, clock, says

Description automatically generated*

*Output of Convolution Layer Experiment 3 1-3*

*A picture containing text, sign, crossword puzzle, colorful

Description automatically generated*

*Output of Max Pooling Layer Experiment 3 1-4*

Graphical user interface, application, table, Excel

Description automatically generated**Results**

*Results from 16 Experiments 1-5*

I ran 16 experiments, where in the first 4 experiments I did not use regularization. I also ran 20 epochs for each experiment. Experiment One involved 64 neurons, Two layers, Experiment Two involved 64 neurons Three layers, Experiment Three involved Two Max Pooling and Convolution Layers, while Experiment Four involved Three Max Pooling and Convolution Layers. After the first four experiments, I then increased neurons involved and use regularization such as L1-L2 Regularizers, Dropout, and Early Stopping. One can see the results and architecture from 1-5. After running each model, I compared the results, and found Experiment 16 with 20 percent Dropout and 3 Max Pooling, Convolution Layers to be the best model. It had a 93.7 percent Training Accuracy, and 75.35 percent Testing Accuracy. One can see from this that the model was still overfitting, but I examined the other 15 experiments and saw all of them were either overfitting, or underfitting, so I chose a model that was overfitting the least which was Experiment 16.

After choosing what I thought was the best model, I then also looked at the Train, Test Confusion Matrix. The Confusion Matrix in 1-6 shows that the Training Data has the higher accuracy, compared to 1-7 which balanced columns which means that accuracy is less. When the Confusion Matrix has more numbers on the diagonals it means that the model is performing well.

A picture containing table

Description automatically generated

*Training Confusion Matrix 1-6*

Table

Description automatically generated

*Testing Confusion Matrix 1-7*

After looking at the Confusion Matrix I also looked at the plot which shows the loss and accuracy as seen in 1-8. I saw that there was a huge gap between Train Loss and Val Loss, it shows that the experiment is overfitting. I also looked at the precision score and it showed that the Precision score for Training Data was significantly higher than Testing Data seen in 1-9. Lastly, I performed T-SNE which is dimensionality reduction unsupervised learning technique to visualize higher dimension data as seen in 1-10 [4].

Chart, line chart

Description automatically generated

*1-8 Experiment 16 Plot*

*Graphical user interface

Description automatically generated with medium confidence*

*Precision Score 1-9*

*Chart, scatter chart

Description automatically generated*

*1-10 T-SNE*

**Conclusion**

In this research I used CIFAR-10 data to do supervised learning multiclass classification. I explored using CNNs to do the experiments. I then examined the accuracies for each of the experiments and also examined how the Convolution and Max Pooling layers learn. After exploring CNNs. I chose the best experiment which was Experiment 16 and examined the confusion matrices, precision scores, and the loss, accuracy plots. After examining the metrics, I then performed dimensionality reduction with T-SNE. After doing this research **for my management recommendation, I choose the Experiment 16 which was the 3 Max/Conv Layers with 20% Dropout architecture.** I chose this model as it had a higher accuracy than the other experiments with accuracies 93.7 and 75.35 percent as seen in 1-5, and it also overfitted less. Overall, my expectations were lowered for this research as I noticed the accuracies were not that high. I was extremely disappointed in my results, and for further research would like to try to obtain 85-90 percent Testing accuracy, by examining other ways to improve the models.

References

*[1] CIFAR-10 and CIFAR-100 datasets*. (n.d.). CS Toronto. Retrieved February 5, 2021, from <https://www.cs.toronto.edu/%7Ekriz/cifar.html>

*[2]* Akwaboah, A. D. (2019, November). *Convolutional Neural Network for CIFAR-10 Dataset Image Classification*. Research Gate. <https://www.researchgate.net/publication/337240963_Convolutional_Neural_Network_for_CIFAR-10_Dataset_Image_Classification>

*[3]* Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Company.

*[4] Introduction to t-SNE*. (n.d.). Data Camp. Retrieved February 6, 2021, from https://www.datacamp.com/community/tutorials/introduction-t-sne