Machine Learning II – Professor Amir Jafari

Comparing Baseline and LeNet for Hand Gesture Recognition

Final Report - Arshiful Islam Shadman

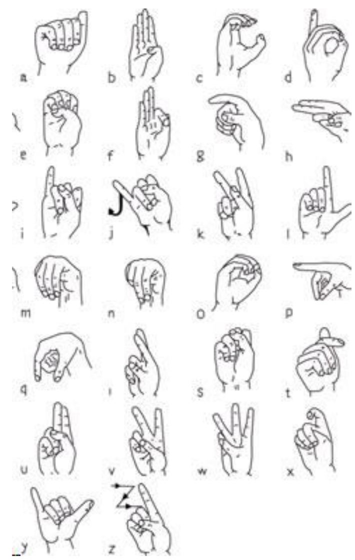
George Washington University

Date – 12/08/2020

Introduction

Hearing loss, deafness, hard of hearing, anacusis, or hearing impairment, is defined as a partial or total inability to hear. People suffering from this disability tend to communicate with sign languages. The American Sign Language (ASL) is a visual language that includes body movements, facial expression and hand gestures.

Some of these hand gestures represents the letters of the alphabets. In this project the 26 alphabets are being classified using computer vision.



Description of the data

The dataset is being downloaded from Kaggle which is a collection of images of alphabets from the American Sign Language arranged in folder marked with their respective labels. Meaning it consists of 29 different folders (i.e. A-Z, Space, Nothing and Delete). Each of this folders contain 3,000 images of each kind summing up to a total of 87,000 images. Therefore the dataset being used does not suffer from imbalances. Each of these images are of 200*200 pixels.

Methods

1. Pre-training

In the pre-training step the image data goes through the preprocessing and choose the two neural networks.

Normalization: The data was brought in using ImageFolder package. Since the data consists of images, 3 values of mean and 3 values of SD for each color channel RGB is being used. This helps to get the data within a range (-1,1) which helps to train a lot faster.

Train Test and Validation Split: The data is arranged in one parent folder including all the sub images folders. It was split it into three sets: train, validation, and test set. A ratio of 70:15:15 is used sklearn's train/test split library.

The table below summarizes the model architectures of the baseline model and LeNet:

	Baseline	**LeNet**
Number of Convolutional Layers	2 CONV(5,5)	2 CONV(5,5)
Number of fully connected layers	1 (250 neurons)	2 (120,84 neurons)
Transfer function in the output layer	LogSoftMax	Linear

2. Training

In the training step it involves the use of training algorithm and the use of performance index which is the cross entropy. Cross entropy is used to find the probability distribution of the multiclass image classification. Adam optimizer which is a robust variant of general stochastic gradient algorithm is used for training algorithm.

Here, the Xavier Normal Initialization is applied as the algorithm automatically determines the scale of initialization based on the number of input and output neurons and makes sure that the weights are not too small or too big. Moreover in order to avoid over fitting drop out nodes and early stopping is being applied. A total of 20 epoch were used for both the models while training.

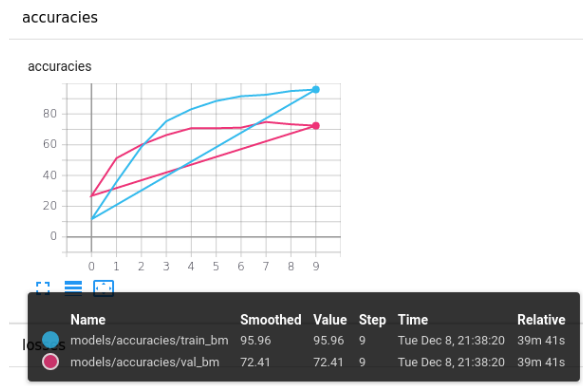
3. Post Training

Some common and classic metrics like confusion matrix, precision, f-score, AUC were used. The error parameters, accuracy trends and loss trends were also used to evaluate the model performances.

4. Results

Baseline Model

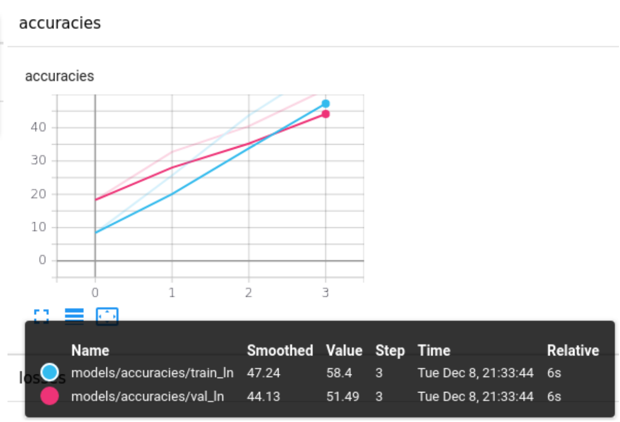
The following trends for accuracies and losses were obtained:



The trends show that the model performs well by the 9th epoch and does not learn much from that point onwards. And the accuracy for the training maxed to 95.96% whereas the accuracy of the model over validation set is 72.41%.

LeNet Model

The following trends for accuracies and losses were obtained:

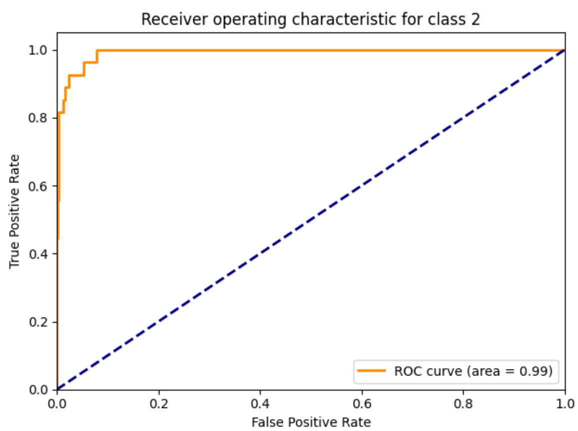


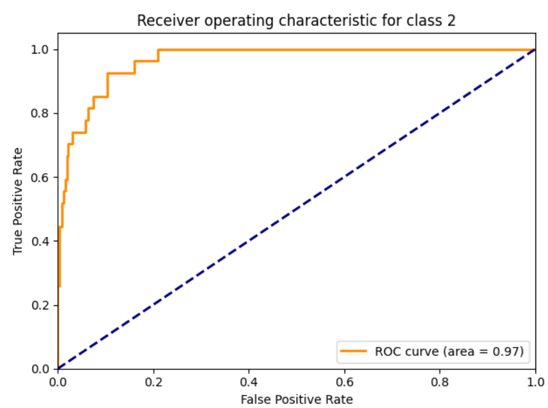
The trends show that the model didn’t performs as well as the Baseline Model However it doesn’t learn much after the 3rd epoch. And the accuracy for the training maxed to 58.4% whereas the accuracy of the model over validation set is 51.49%.

The table below shows the summary of the results of the two models:

	Baseline	LeNet
Precision	0.00807274643571692	0.5143162561498629
Training Accuracy	95.96%	58.39%
Validation Accuracy	72.41%	51.49%
F-Score	0.7209187707367711	0.48049252379868024
AUC	0.99	0.97
Error Standard Deviation	0.00807274643571692	0.00811545001871179
Error Mean	0.02018198514806813	0.03431515967708894

The figures below shows the ROC charts of the two models:





Baseline ROC ***LeNet ROC***

5. Conclusion

The baseline model seems to perform better than LeNet. However time wise LeNet is faster. The accuracy on the validation is low for LeNet but this is because of the batch size. Changing the number of neuron or increasing the batch size could improve the models. However in most cases LeNet performs better than what is obtained in this project and it can be used for hand gesture translation application where time is important. Other application for example image labeling or tagging in social media platforms can rely on Baseline Models. For future work a combination of the two could models could be used for classification.

6. References

Bheda, V., & Radpour, D. (2017). Using deep convolutional networks for gesture recognition in american sign language. CoRR, abs/1710.06836. Retrieved from http://arxiv.org/abs/1710.06836

Garcia, B., & Viesca, S. (2016). Real-time American Sign Language recognition with convolutional neural networks. In In convolutional neural networks for visual recognition. Stanford. http://cs231n.stanford.edu/reports/2016/pdfs/214_Report.pdf

Implementing Yann LeCun's LeNet-5 in Pytorch https://towardsdatascience.com/implementing-yann-lecuns-lenet-5-in-pytorch-5e05a0911320

Trigueiros, P., Ribeiro, F., & Reis, L. P. (2012, June). A comparison of machine learning algorithms applied to hand gesture recognition. In 7th iberian conference on information systems and technologies (cisti 2012) (p. 1-6). https://core.ac.uk/download/pdf/55626122.pdf