Sign Language Recognition

A project from the course Deep Learning and Natural language processing by Dr. Amos Azaria.

Dolev Abuhazira [1], Egor Kononov [2]

Abstract :

American Sign Language is the primary language of many North Americans who are deaf or hard of hearing and identify as part of the Deaf community. Sign languages are complete natural languages that having their own syntax and grammar she emerged as a language in the US School for the Deaf , founded by Thomas Gallaudet in 1817.

Through sign language recognition systems, those people will be able to communicate much better with people outside the deaf community and individuals within the community.
In this article we will review and compare between different ways of sign language recognition by different algorithms and models based on deep learning process.

1. Introduction :

Sign language is based on hand expressions that express the opinions and thoughts of people with disabilities. Most of the people today do not understand sign language and therefore there is a big gap in communication between people with disabilities and those people without disabilities.

In these days when technological advancement is increasing, there is a need to also integrate communities that are left behind due to their difficulties. If there is a way for people and deaf-mute people to communicate, the deaf-mute people can easily live like a normal person. And the only way to communicate is using the sign language.

On this article we will present different models based on deep learning whose purpose is to recognize sign language, using these models we can build systems that use the same models.
In addition, we will explore and delve into the features of the various models and compare them in order to get the best model approach.

Related Work :

G. Anantha Rao et. al. discusses the results of the application of deep convolutional neural network for sign language recognition, with an accuracy of 92.88% recognition on self-constructed dataset using OpenCV and Keras.

Sebastien Marcel et. al. uses a neural network to interpret hand postures ar in image. The dataset was self-constructed which contains uniform and complex backgrounds with a recognition accuracy of 93.7%.

Srinath S, Ganesh Krishna Sharma uses a classification approach for sign language recognition , their system recognizes 24 American Sign Language alphabets gestures with accuracy of 86.67%.

2.Sign Language Data Set :

The MNIST dataset is a large collection of handwritten digits that is used for trainning various image processing system. This data contains set of 28x28 images of all the alphabet except J and Z because they require a movement for representation. Source : Kaggle.

Each example in the dataset contains the lable which corresponds to their class, and also 784 features that represent a numerical value (pixels) for each example. There are 27455 exmaples in the tranning set and 7172 in the test set.
A sampled image set can be observed in Fig. (1) And the dataset distibuition observed in Fig. (2).

Fig. (1)

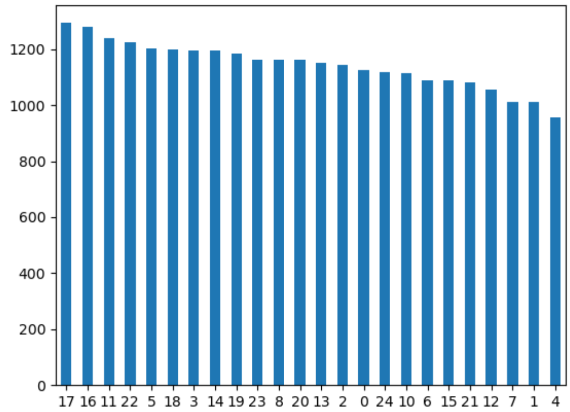

Fig. (2)

3. Experminetal Results :

For this work we implemented three different models and feed them with the same inputs so we can see superiorities and drawbacks of networks.
Each one of the models train and test on normalized data. We divide each numerical value that represent an image pixel by 255. Because computation of high numerical values my become more complex.

3.1 Logistic Regression Model :

This model used to describe the relationship between a response variable and more explantory variables.
Because of the fact that our problem is classification , we use the Softmax activation function to classify each example in the dataset. This function scaling the output values and converting them into probabilities such that all values in the returned list of size 25 have sum to 1.0.
In addition , we use the Adam method for stochastic optimization that will be optimize our loss function which defiend as cross entropy that used to describe a predected outcome compare to the true outcome.
The trainning was completed in 15 epochs with 250 batch size. Training and validation loss and accuracy values in each epoch were shown in Fig. (3,4)

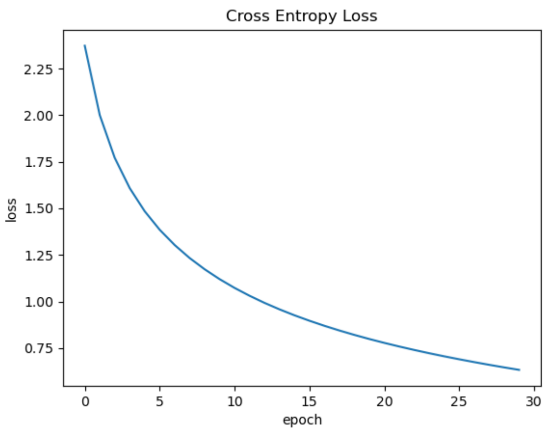

Fig. (3)

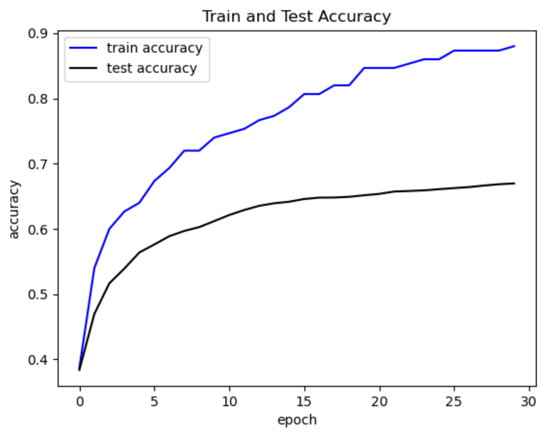

Fig. (4)

3.2 Multi-Layer Neural Network :

This model have multiple layers between the input and output layers. The neural network finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. Our neural network contains two hidden layers when their activiation function is Relu which more faster to compute because Relu just needs to pick max (0,x) and not perform expensive exponential operations as in Sigmoids and Softmax activition functions. The activation function of the output layer defiend as Softmax because our problem is classification. Also we use the Adam method for stochastic optimization that will be optimize our loss function which defiend as category cross entropy (Keras Library).

In Addition we use the dropout regularization technique on each iteration, we randomly shut down 20% neurons of the second hidden layer and don't use those neurons in both forward propagation and back-propagation.
Since we drop some neurons on the second hidden layer, this will lead to smaller network which in turns means simpler network.

The trainning was completed in 30 epochs with 50 batch size. While the trainning process , 20% of the trainning data was randomly split to validation set.
Training and validation loss and accuracy values in each epoch were shown in Fig. (5,6)

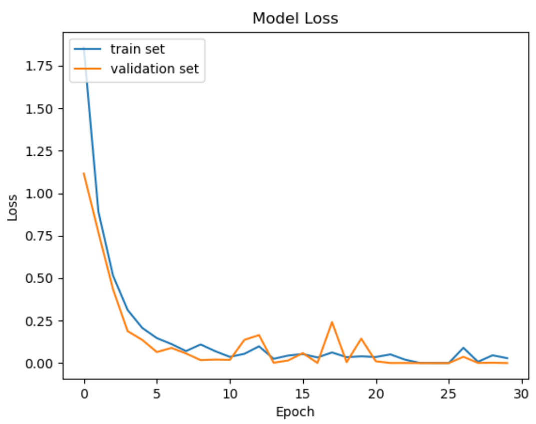

Fig. (5)

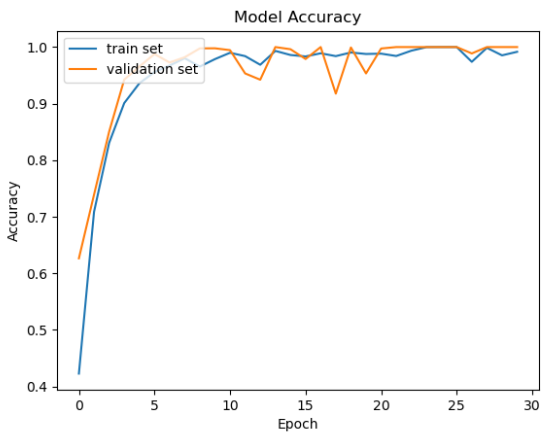

Fig. (6)

3.3 Convolutional Neural Network:

Convolutional Neural Network is similar to an ordinary feedforward neural network as the output of each layer is a combination of the input, weight matrix, and the bias vector followed by a non-linear transformation.
it takes advantages of the convolution operation between the filters (kernels) and the input

Convolutional Neural Network normally consisted of several types of layers, including convolutional layer, pooling layer, and fully-connected layer. Fig. (7) represented an input image to the Convolutional Neural Network.

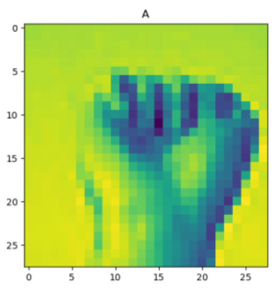

Fig. (7)

In this experiment our convolutional neural network model developed consists 9 different layers :

- Three convolutional layer
- Three Pooling (Max) layer
- Three Dropout layers
- Two Dense connected layers

The input layer accepts single-channel gray-scale image with size of (28,28).
Each convolutional layer have kenels of size (3,3) And each max pooling layer consists (2,2) pool size with the Relu activation function. In addition each dropout layer drops out 20% of the hidden and visible neurons from the connected layers. The last obtained layer is flattened and connected to a fully connected layer with 128 features. This layer is connected to the output layer of size 25.

Also we use the Adam method for stochastic optimization that will be optimize our loss function which defiend as category cross entropy (Keras Library).
The trainning was completed in 30 epochs with 50 batch size. While the trainning process ,Training and validation loss and accuracy values in each epoch were shown in Fig. (8,9).
Test results of this study were shown in Fig.(10) Confusion matrix of classification.

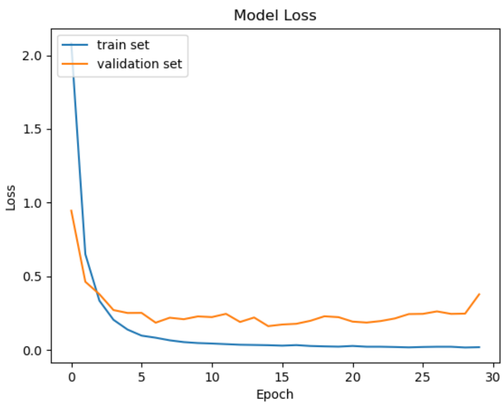

Fig. (8)

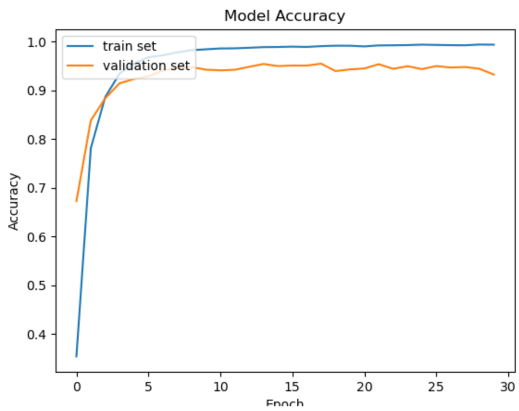

Fig. (9)

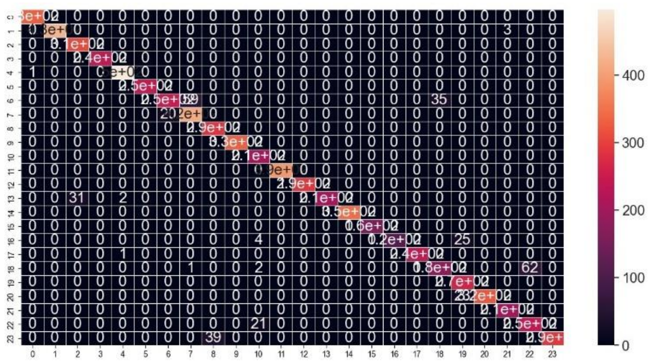

Fig. (10)

4. Conclusions :

In this article we compared between three different models based on deep learnning , As a result the most efficient model is the Convolutional Neural Network which get 96% of accuracy compare to Multi layer neural network which get 86.4%.