Assignment-3

Convolutional neural network

Indian Institute of Information Technology, Allahabad

Members:

IIT2015109 – K Jaya Pavan

IIT2015110 – Eshwar

IIT2015111 - Naga Teja

INTRODUCTION:

In this task, we will take in the working of CNNs and their distinctive models. Furthermore, discover which demonstrate gives better outcomes. Convolutional neural networks(CNN) are deep feed-forward artificial neural systems. They are generally utilized in deep learning. Convolutional neural networks works on the basic fact that a vision system will use the similar knowledge at every position in the given image. This process is accomplished by giving the weights of feature detectors so we can use the features that are learned at one location, at different locations. This procedure is proficient by giving the weights of highlight identifiers so we can utilize the highlights that are found out at one area, at various areas. Convolutional neural systems utilizes Gradient drop , back-engendering, max pooling and so forth. Convolutional neural systems comprises of an info and a yield layer, and also various hidden layers. The hidden layers of a Convolutional neural systems ordinarily comprise of convolutional layers, pooling layers, completely associated layers and normalization layers.

ARCHITECTURE:

We are using MNIST dataset and we are changing activation function. MNIST dataset is of manually written digits that is regularly utilized for preparing different frameworks. It comprises of 60,000 trained images and 10,000 testing pictures. There are 10 classes in MNIST.

.

Following is the architecture of CNN used:-

We have created a simple Convolutional Neural Network.

It takes a 28×28 dimension image as input from MNIST dataset.

*In -> [[Conv2D->relu]*2 -> MaxPool2D -> Dropout]*2 -> Flatten -> Dense -> Dropout -> Out*

Our architecture consists of two convolutional layers with 32 filters and each filter of size 5x5. And then we apply ReLU activation function to the convoluted image in both the convolutional layers and then we use Swish activation function and then we will absove which will give good results. Max pooling with filter size of 2×2 and with stride of 2.

Then we will take dropout layer with probability of 0.25. Then, we will flatten the output of dropout layer to convert it to vector form. Then it will be taken as input for dense layer which has 128 neurons and we apply ReLU function. And then output is sent to another dropout layer with probability of 0.50 and apply ReLU function. Again output will be sent to the dense layer which have 10 neurons and after applying the softmax activation function we get the desired output and class will be identified for the image.

EXPERIMENTS:

We have done several experiments with the architecture. We changed the the activation function.

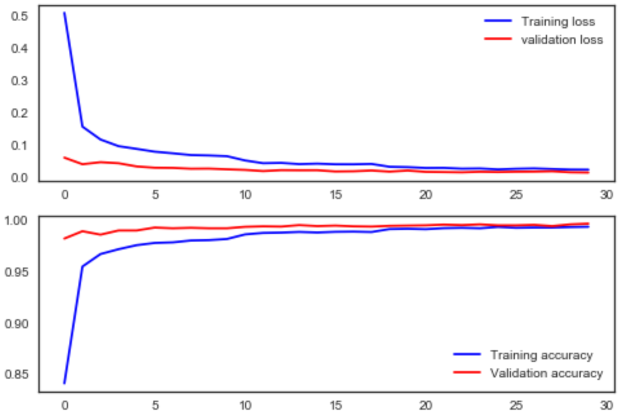
Following is the experiment with architecture:-

In -> [[Conv2D->swish]*2 -> MaxPool2D -> Dropout]*2 -> Flatten -> Dense -> Dropout -> Out

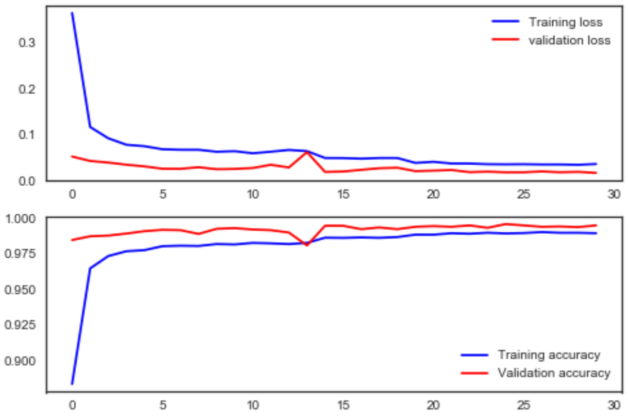
ANALYSIS:

Accuracy comparison

In this architecture, we change the activation function from Relu to Swish to the same architecture. In term of accuracy i found that Swish is doing better than ReLU. Make 90 position up in kaggle.



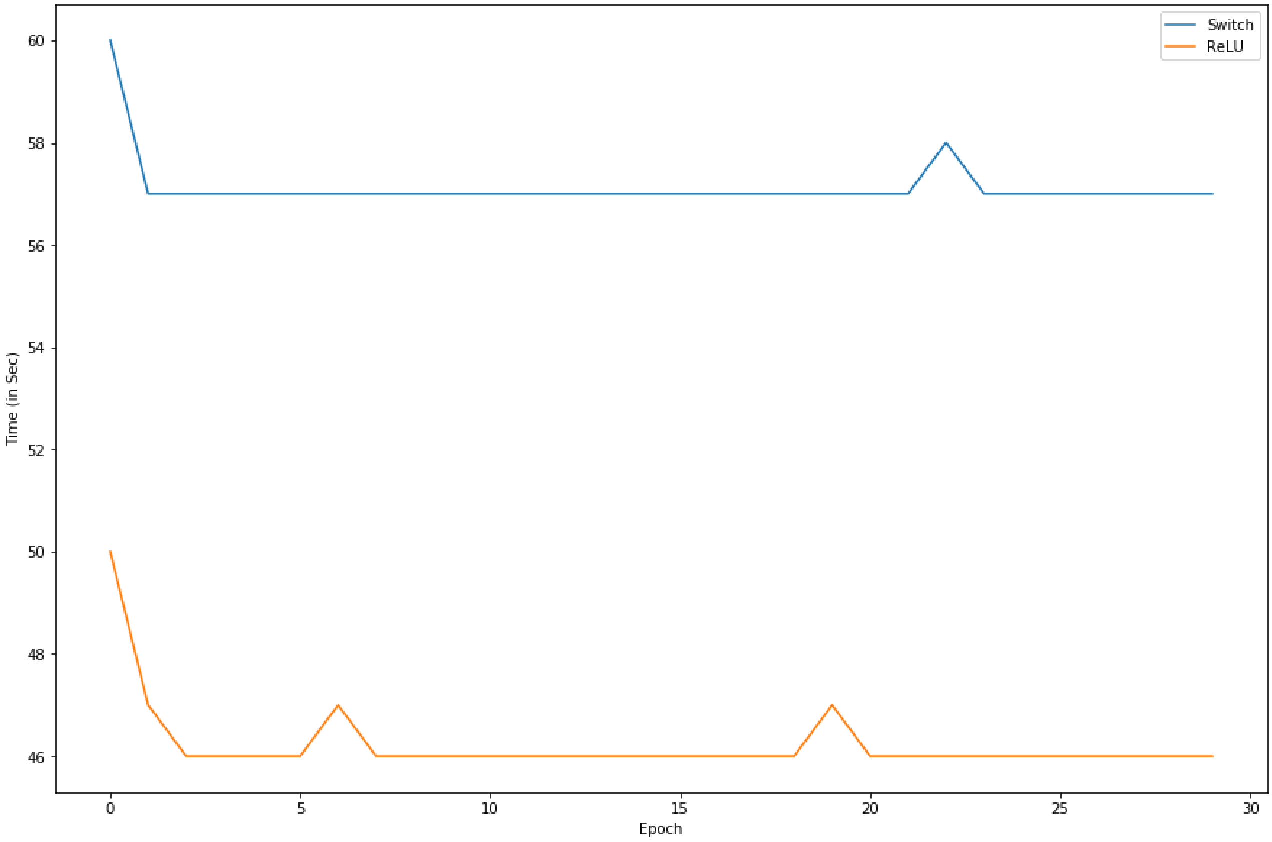
Switch



ReLU

Time Comparsion:

ReLU is more than 10 second faster than Swish in each epoch . Swish calculate the sigmoid of input and multiply with the input. May be that take extra 10 second for each epoch.



CONCLUSION:

After submission Swish 90 position up my position. But that was just 0.002 more. And it's take 20%–30% more time per epoch in GPU. There is another version of Swish activation known as Swish_beta (f(x) = 2x*sigmoid(beta*x)). I hope that will work better than Normal Swish.