

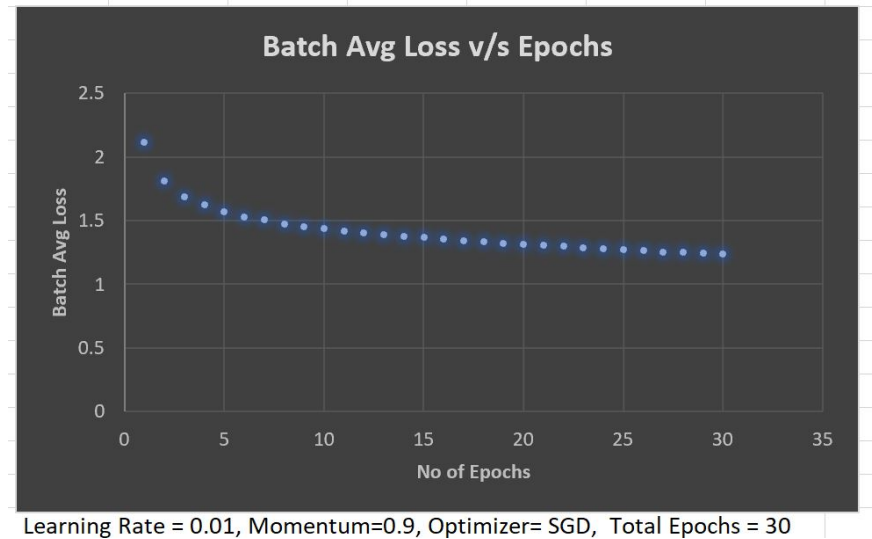
In this project I implemented 7 layer fully-connected (MLP) and 7 layer Convolutional Neural Networks (CNN) image classifier on CIFAR10 dataset. Performed multi-class classification and implemented the code using Pytorch framework. I started with data analysis, CIFAR-10 contains 60000 labeled for 10 classes images 32x32 in size for each RGB channel, train set has 50000 and test set 10000. The categories are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

0.1 7 Layer FC Neural Net (MLP)

Fully Connected Network with ReLu Activation . See below the network structure that is used

7 Layer Fully Connected Neural Network Architecture			
Layer	Input Dimension	Output Dimension	Activation
1	3072	200	ReLu
2	200	200	ReLu
3	200	200	ReLu
4	200	200	ReLu
5	200	200	ReLu
6	200	200	ReLu
7	200	10	SoftMax

The final test accuracy was **53%** (as required). Here is the plot of batch averaged loss per epochs for total 30 epochs:



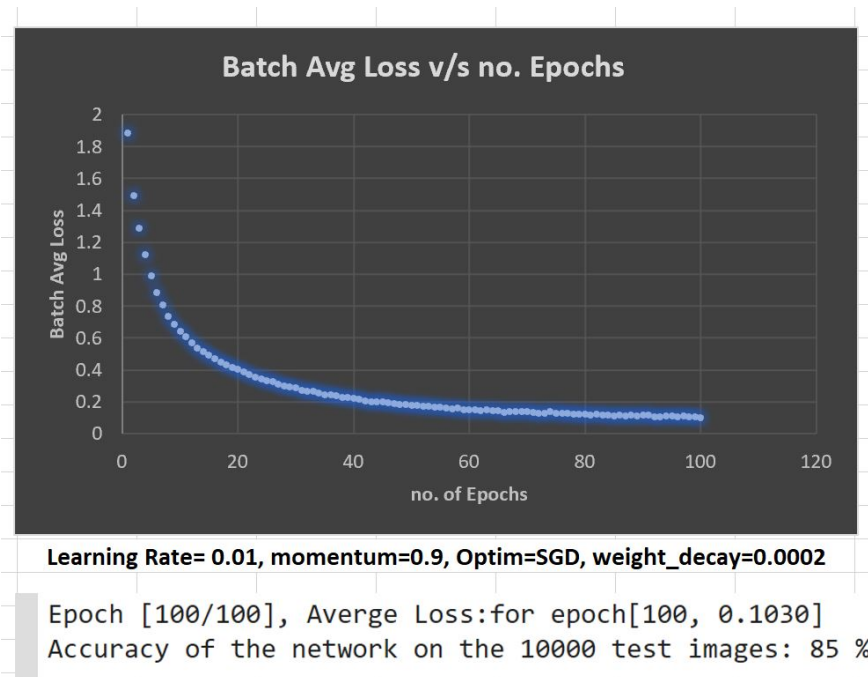
The fully connected neural network scales poorly on the CIFAR-10 dataset and achieves accuracy just over 50% . There are total **817610** learn-able parameters, This model would not scale well of images of larger sizes as parameters will explode.

0.2 7 Layer Convolutional Neural Net

7 layers convolutional neural networks, 4 convolutional layers and 3 fully- connected layers, with ReLu activation function. see the network architecture below:

7 Layer Convolutional Neural Network Architecture									
Layer	Type	Input features	Input Channel	Output Channel	Kernel Size	Stride	Padding	Output features	Activation
1	Conv2d	32*32*3	3	84	5	2	1	15.5*15.5*84	ReLu
2	Conv2d	15.5*15.5*84	84	256	5	2	1	7.25*7.25*256	ReLu
3	Conv2d	7.25*7.25*256	256	256	3	1	0	5.25*5.25*256	ReLu
4	Conv2d	5.25*5.25*256	256	256	2	1	0	4.25*4.25*256	ReLu
5	FC	4096						4096	ReLu
6	FC	4096						4096	ReLu
7	FC	4096						10	Softmax
Learning Rate = 0.01, Weight_Decay = 0.0002, Opt = SGD, momentum=0.9,#Epochs=100									

Below I've plotted batch-averaged losses for each epoch, i ran total 100 epoch.



I was able to achieve **85%** test accuracy , see the listed hyper parameters in above images. I used Google Collab GPU to increase training speed. I could have improved it further but stopped once i got 85%. We can clearly see that the accuracy of CNN is higher than MLP neural net. Also the loss drops quickly and converges nicely compare to MLP.

0.3 Comparison of 2 models

We can clearly see that the accuracy of CNN is higher than MLP neural net. I also noticed that the loss function was not dropping enough and hence not converging well for MLP. Also it is clear that if you build deep neural network for image classification, CNN will use less number of learn-able parameters, because it shares the parameters. Also the output feature of CNN depends on smaller region of inputs as conv layers are breaking the input (e.g. an image) up into common features, and the FC layers in the end of CNN are piecing those features together into e.g. objects that you want the network to recognize.

0.4 No Relu Function

I removed non-linear Relu function from both 7 layer MLP and 7 layer CNN network below are the results

- **7 layer MLP without Relu with same hyper parameters** It gave me an accuracy of just **36%**. We can clearly see that without a non-linear activation function in the first 7 layer MLP network, no matter how many layers we would add it would behave just like a single-layer perceptron, because summing these layers would give you just another linear function
- **7 layer CNN without Relu with same hyper parameters** This one gave me an accuracy of **37%**, we can clearly see that the neural network did not perform well if we remove the non-linear activation function.

It is clear that Non linearity in neural network allowed us to model a response variable (aka target variable, class label, or score) that varies non-linearly with its explanatory variables. Non-linearity also means that the output cannot be reproduced from a linear combination of the inputs (which is not the same as output that renders to a straight line).

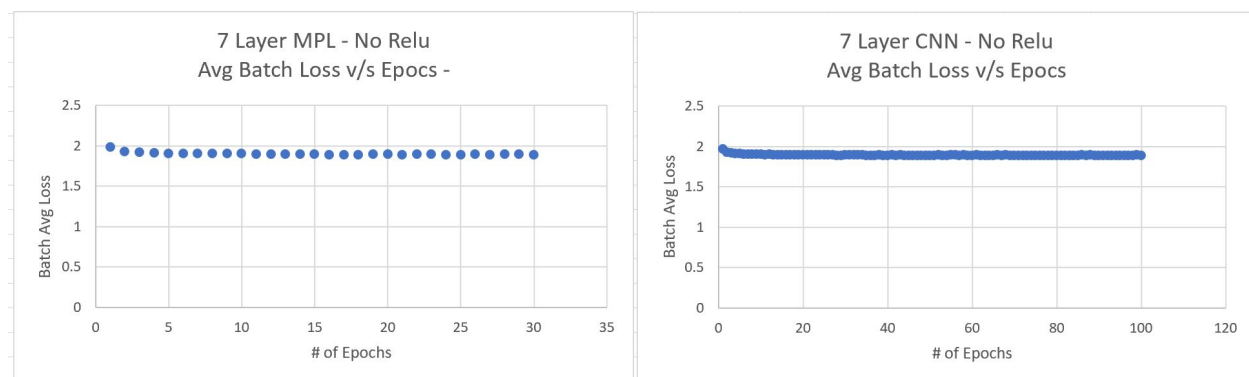


Figure : Loss Convergnace for both networks when Relu Activation was not used