

## Assignment Report: A2 , Neural Networks

Date: 6th September, 2019

Submitted By: Sumit Singh, 2016PH10575

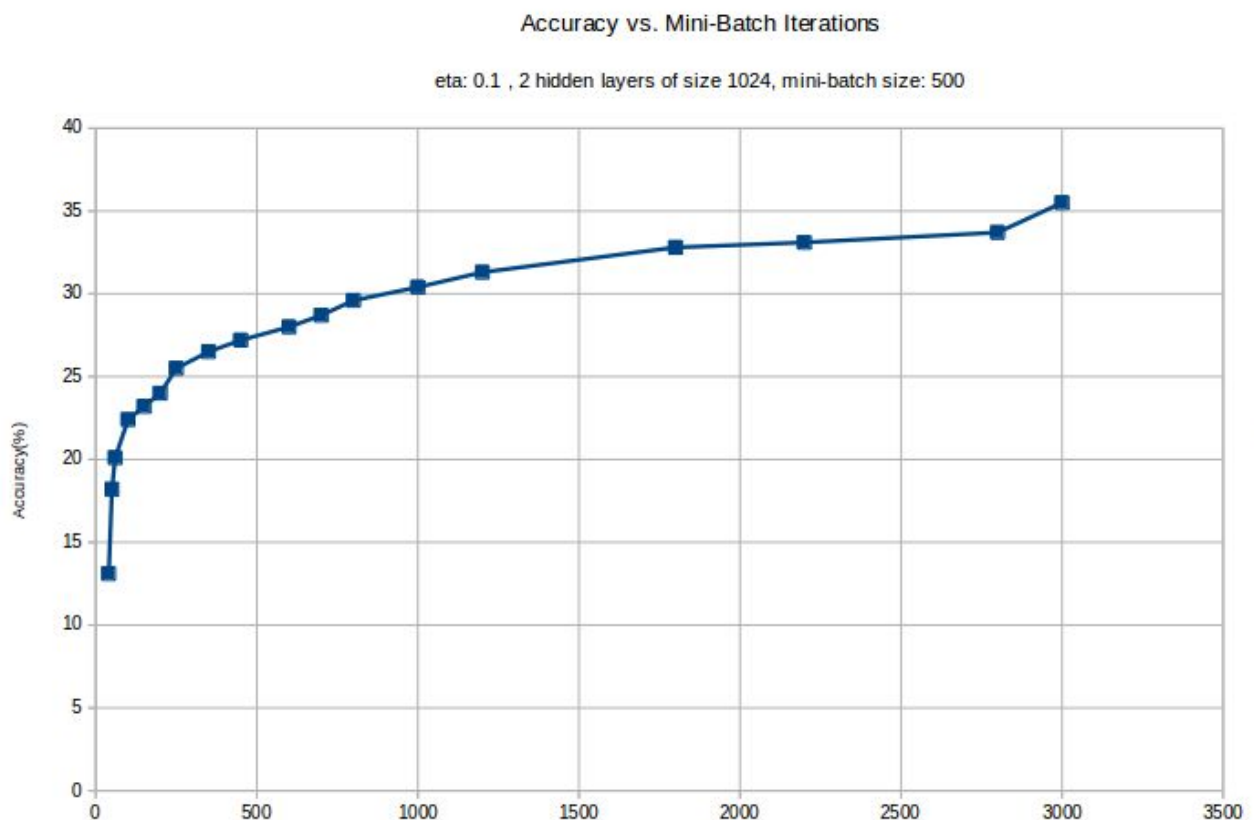
### Common Architectural Details For 2(c) and 2(d):

The best architecture is as follows:

Learning rate:	0.1
Learning strategy:	Fixed
No. of mini_batch iterations:	7000
No. of hidden layers:	2
No. of units in every hidden layer:	128 , 128
Size of mini-batch:	500
Maximum accuracy achieved:	40.24%

The experimentations done and the results obtained are reported below:

- 1. Increasing number of hidden layers:** Accuracy increases upto 2 or 3 hidden layers (depending on the number of units per layer). A single hidden layer performs poorly. Hidden layers above a total of three also perform poorly due to vanishing gradients.
- 2. No. of mini-batch iterations:** Increasing number of mini-batch iterations usually led to improvement in performance. The following bar-graph summarizes it better.



3. **Number of units in a hidden layer:** No. of units less than 100 per layer performed poorly. However when the number of units in the hidden layers (esp. the first hidden layer) was a **power of 2**, there was a **significant increase** in accuracy. For instance,

No. of units in hidden layer	Accuracy
125	23.4%
128	26.7%
130	24.6%

4. **Learning Rate:** Learning rate of **0.1** gave the best accuracy. Smaller rates took larger number of iterations to achieve similar accuracy. Larger rates performed poorly.
5. **Weight Ranges:** Best accuracy was achieved with sigmoid activation function when initial weights were taken from standard normal distribution between -0.1 and 0.1. For ReLU activation function the best accuracy was achieved with **Kaiming Initialization** where normalization factor is  $\sqrt{2} / \sqrt{\text{no. of units in the layer from weights are incoming}}$ . This took care of the 'exploding' and 'vanishing' gradients in case of ReLU activation.
6. **Activation Function:** With proper initialization, ReLU performed better than logistic sigmoid.
7. **Data Normalization:** Training data was normalized with dividing the entire set with 255.0. Performance improved due to normalization.

#### Part (d):

Implemented **Gabor filter** for frequency 0.9. The filtered data was normalized by dividing with the maximum of the image array. Experimenting with frequencies and band-widths for Gabor filter gave varying accuracies. Only the real part of the response was considered. Best accuracy achieved was about 40.24%.