Assignment 2 Report

IEORE 4742 Deep Learning for OR and FE

Ahmad Shayaan[1]
as5948

{ahmad.shayaan@columbia.edu}@columbia.edu
Columbia University
October 14, 2019

[1]All authors contributed equally to this work.

Question 1 Solution

Question 1 Part 1

The total number of trainable parameters is the sum of all the wights and biases of each layer. In the network that we are training the total number of parameters are given by.

$$\text{Total Number of Parameters} = (784 \times 400) + 400 + (400 \times 200) + 200 + (200 \times 100) + 100 + (100 \times 50) + 50 + (50 \times 25) + 25 + (25 \times 10) + 10$$

$$\text{Total Number of Parameters} = 420885$$

Question 1 Part 2

Table 1: Change in accuracy with change in architecture

Layers	Architecture 1	Architecture 2	Custom Architecture 1	Custom Architecture 2
Hidden Layer 1	Sigmoid (784x400)	Tanh (784x400)	ReLu (784x500)	ReLu (784x1000)
Hidden Layer 2	Tanh (400x200)	ReLu (400x200)	ReLu (500x400)	ReLu (1000x500)
Hidden Layer 3	ReLu (200x100)	ReLu (200x100)	ReLu (400x200)	ReLu (500x400)
Hidden Layer 4	Sigmoid (100x50)	Tanh (100x50)	ReLu (200x100)	ReLu (400x200)
Hidden Layer 5	Leaky ReLu (50x25)	Leaky ReLu (50x25)	ReLu (100x50)	ReLu (200x100)
Hidden Layer 6	None	None	ReLu (50x25)	ReLu (100x50)
Hidden Layer 7	None	None	None	ReLu (50x25)
Accuracy	97.93	98.16	98.4	98.62

It can be observed from the table that having ReLU activation and more number of layers increases the accuracy. This is because ReLU activation does not saturate the gradient which help the optimizer to quickly converge to the optimum. Increase in the depth of the network helps the network learn more complex relationship between the input features and the objective function.

Question 1 Part 3

Table 2: Accuracy for different optimizers

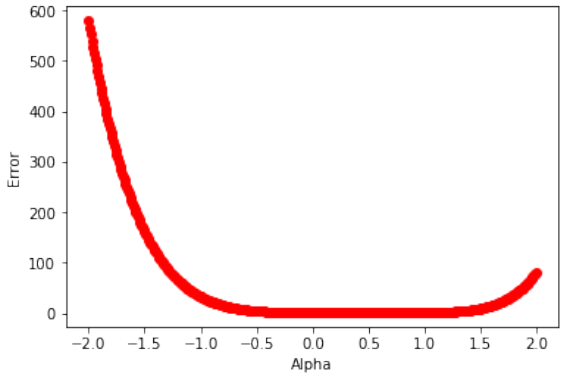
Optimizer Used	Accuracy in percentage
Gradient Descent Optimizer	98.12
Adam Optimizer	10.09
Adagrad Optmizer	98.27
RMSprop Optimizer	10.1

It can be seen from the table that Adagrad optimizer does the best with the same network architecture. This is because this algorithm adaptively scales the learning rate for each step it takes towards the optimum. Adam and RMSprop algorithm don't do as well because the learning rate is not sufficient. On reducing the learning rate I observed that Adam and RMS prop do better than other optimization algorithms. It has been shown that Adam and RMSProp are highly sensitive to certain values of the learning rate and they can catastrophically fail to converge. Gradient Descent optimizer has also provided us with good results.

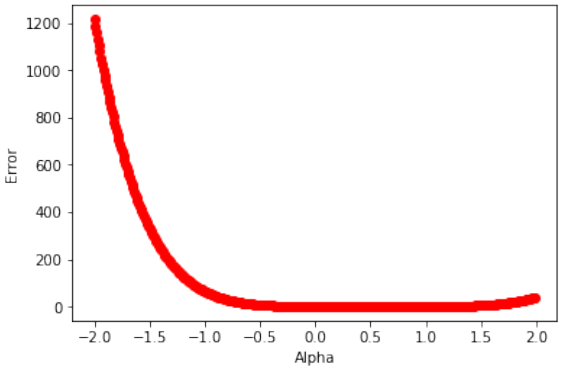
Question 2

Question 2 Part 2

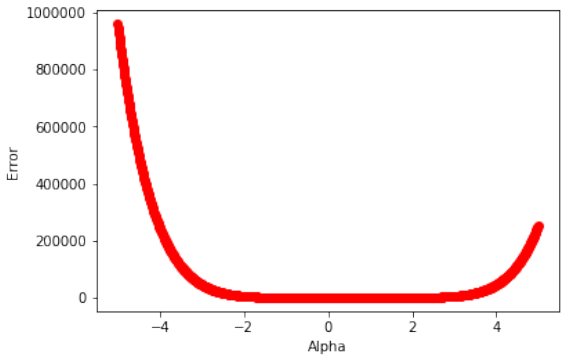
The linearly interpolated loss surfaces of the different networks are shown below. To plot the surface I modified the linear interpolation code that was provided to us.



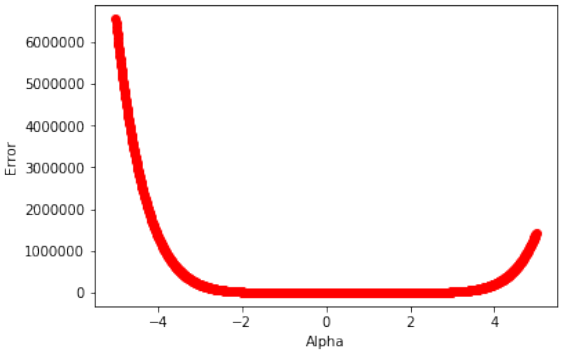
(a) Gradient optimzer



(b) Adam optimizer



(c) Adagrad Optimizer



(d) RMSProp Optimizer

Figure 1: Loss Function for various optimizers

Figure 1a shows the loss surface for the first network in question 1. It can be observed from the figure that the loss surface is flat near the optimum learned by our network and tends to rise as we move further away from the center.

Figure 1b shows the loss surface for the second network in question 1. The curve again

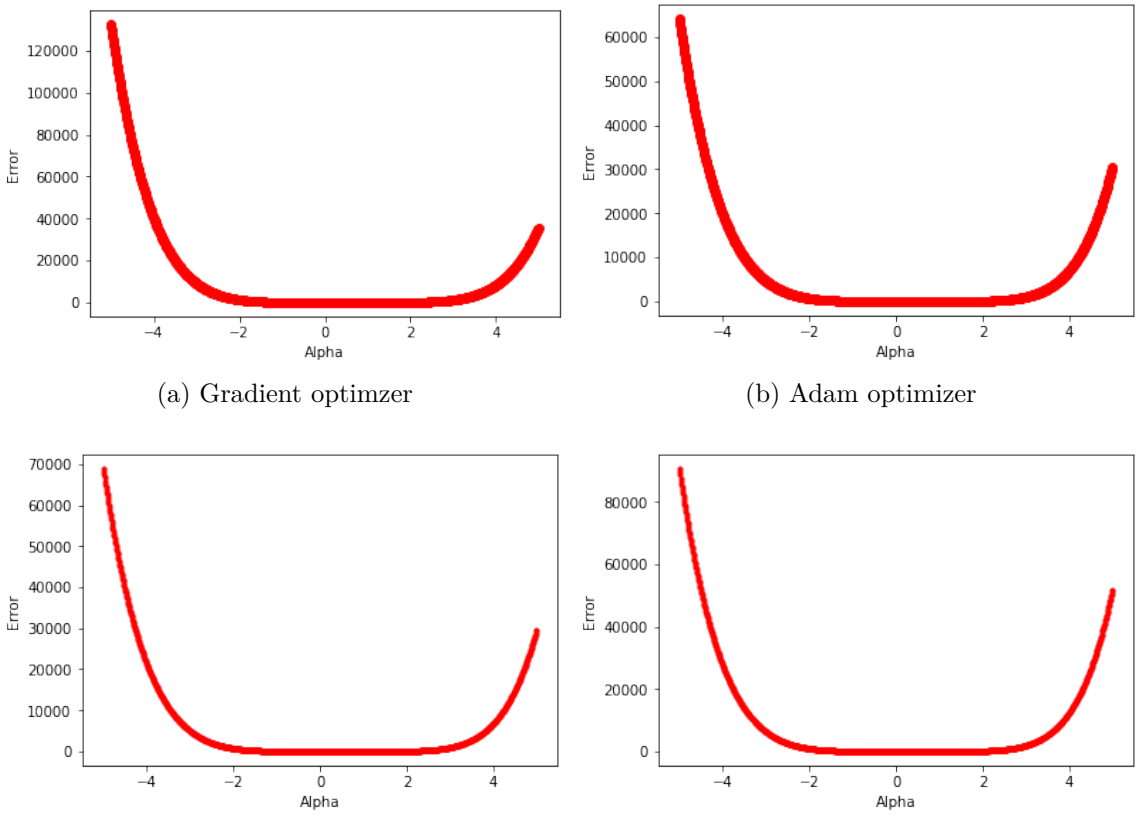
is flat at the optimum learned by the network and rises as we move away from the center. However, the rate at which the cost rise smaller as compared to the loss surface with previous parameters.

Figure 1c shows the loss surface for the network with six layers in question 1. There is no change in the loss surface as we wary the interpolation factor from -2 to 2. We can only observe the changes in the loss surface if we increase the range of the interpolation factor. On increasing the rage we can observe that the loss rapidly start rising as we move away from the optimum.

Figure 1d shows the loss surface for the network with seven layers. It can be observed that the rate of change of the loss as we vary the interpolation parameter is smaller compared to the loss surface for the six layer network.

Question 2 Part 2

In this question we had to plot the change in the loss surface as we change the optimization algorithm.



(a) Gradient optimzer

(b) Adam optimizer

(c) Adagrad Optimizer

(d) RMSProp Optimizer

Figure 2: Loss Function for various optimizers

Figure 2 shows the loss surfaces for different optimizers. The loss surfaces mostly differ on

the rate with which the loss changes as we change interpolation factor. The rate of change on of the loss is maximum in case of gradient descent optimizer and minimum in case Adagrad optimizer

Question 3

Question 3 Part 1

The total number of trainable parameters is the sum of all the wights and biases of each layer. In the network that we are training the total number of parameters are given by.

$$\text{Total Number of Parameters} = (3072 \times 400) + 400 + (400 \times 200) + 200 + (200 \times 100) + 100 + (100 \times 50) + 50 + (50 \times 25) + 25 + (25 \times 10) + 10$$

$$\text{Total Number of Parameters} = 1336085$$

Question 3 Part 2

We had to train a network to correctly classify the images in CIFAR-10 data set. The dataset consist of 60000 images each with dimension $32 \times 32 \times 3$. The images presented in the dataset belong to 10 different classes and all the classes are mutually exclusive. There are 50000 training images and 10000 test images.

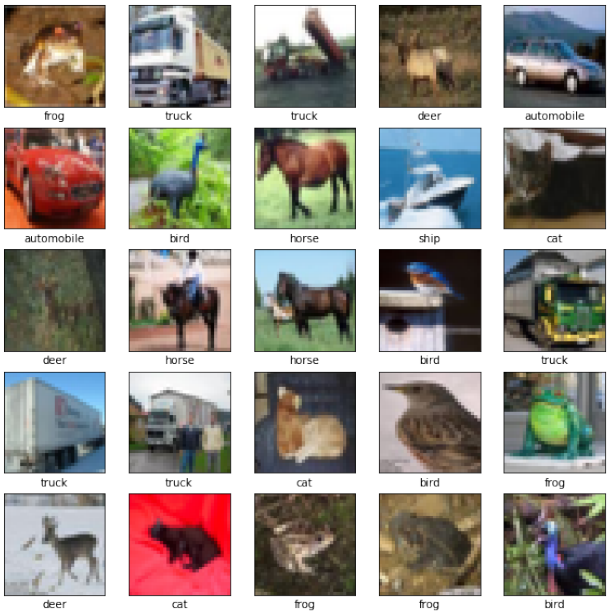


Figure 3: Images present in CIFAR-10 dataset

Table 3: Change in accuracy with change in architecture

Layers	Architecture 1	Architecture 2	Custom Architecture 1	Custom Architecture 2
Hidden Layer 1	Sigmoid (3072x400)	Tanh (3072x400)	ReLu (3072x1000)	ReLu (3072x1000)
Hidden Layer 2	Tanh (400x200)	ReLu (400x200)	ReLu (1000x500)	ReLu (1000x500)
Hidden Layer 3	ReLu (200x100)	ReLu (200x100)	ReLu (500x200)	ReLu (500x400)
Hidden Layer 4	Sigmoid (100x50)	Tanh (100x50)	ReLu (200x100)	ReLu (400x200)
Hidden Layer 5	Leaky ReLu (50x25)	Leaky ReLu (50x25)	ReLu (100x50)	ReLu (200x100)
Hidden Layer 6	None	None	ReLu (50x25)	ReLu (100x50)
Hidden Layer 7	None	None	None	ReLu (50x25)
Accuracy	26.94	27.4	40	47.63

It can be observed from table 3 that accuracy increases with the increase in the depth of the network and the number of neurons. This is because a deeper network is able to capture more complex relationship between the input features and the objective function. The accuracy also increases with the increase in the number of neurons at each layer as with more number of neurons per layer we will be loosing very less information when we reduce the dimensionality of the input.

Question 3 Part 3

Table 4: Accuracy for different optimizers

Optimizer Used	Accuracy in percentage
Gradient Descent Optimizer	22.4
Adam Optimizer	16
Adagrad Optmizer	26.94
RMSprop Optimizer	6.12

The results replicate the findings in question one part 3 that Adagrad optimizer outperform all the other optimization function.