Deep Learning Project 2

Ju Wu, Yingxue Yu

May 2021

Introduction

In this project, we design a simple neural network from scratch with basic modules like Linear(fully connected layers), Tanh, and ReLU and Sequential layers to combine several modules in basic sequential structure, loss modules like MSE and BCE and optimizer modules like SGD and Adam. And then we test the framework with a simple classification task.

Implementations

We design the framework for neural network as shown in Fig. 1, in which network class, dense layer class, activation function class and optimizer class are defined and combined together. And then we implement a test executable named *test2.py* that imports our basic framework with sequential layers that generates a training and a test set of 1000 points sampled uniformly in $[0, 1]^2$, each with a label 0 if outside the disk centered at $(0.5, 0.5)$ of radius $1/\sqrt{2\pi}$, and 1 inside, build the basic network with two input units, two output units, three hidden layers of 25 units, train the above network with MSE, logging the loss, compute and prints the final train and the test errors. And then we test the performance of the basic network with Adam optimizer.

Architecture

The architecture of our neural network is shown in Fig. 2 and we choose $Tanh$ module as $tanh(10x)$ to accelerate training.

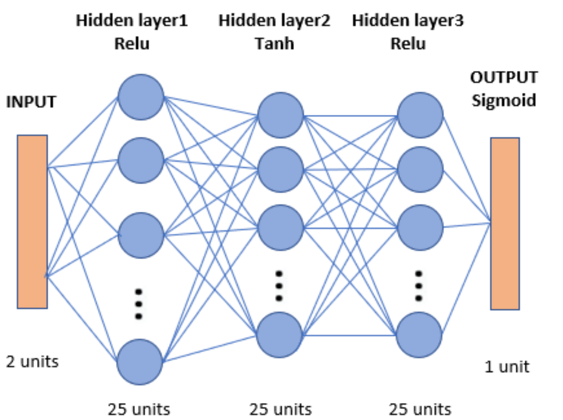


Figure 2: Architecture of basic neural network

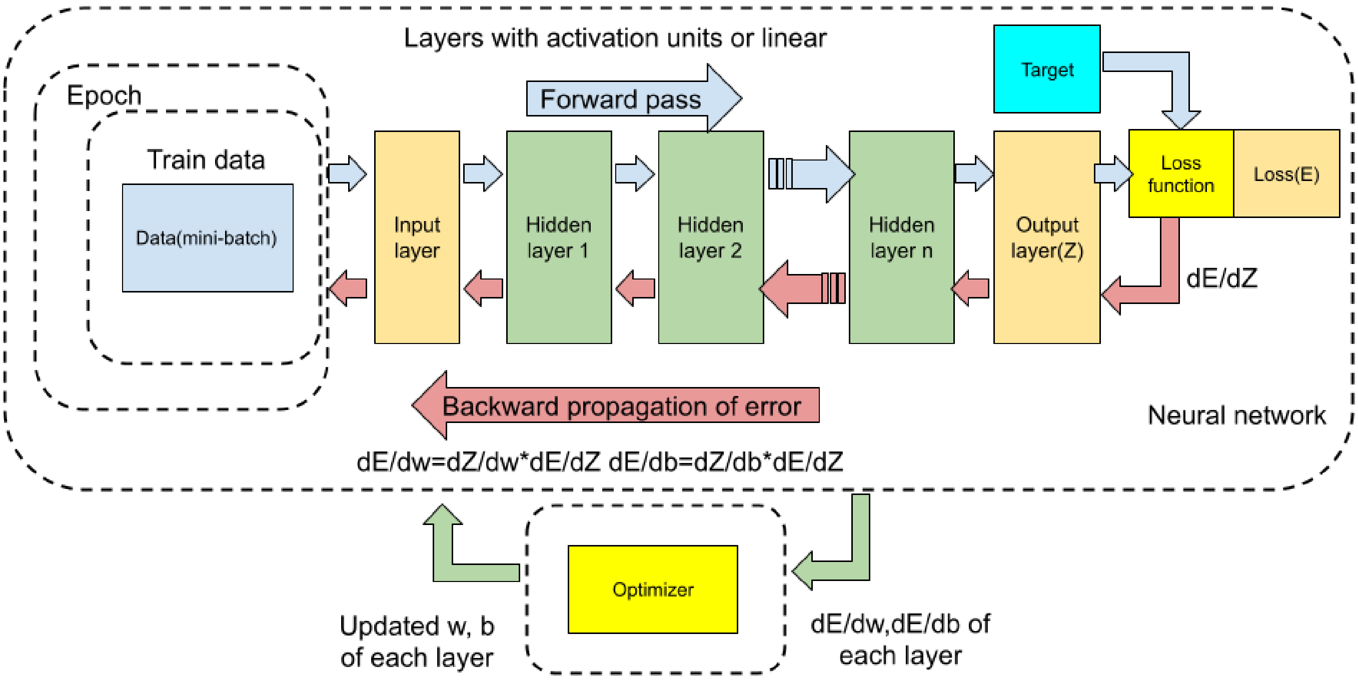
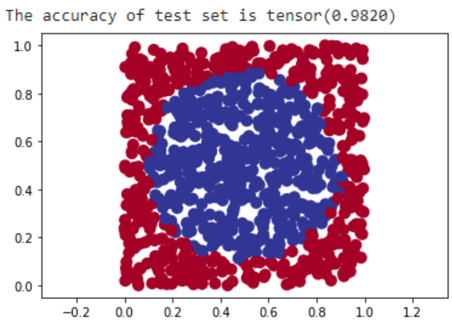
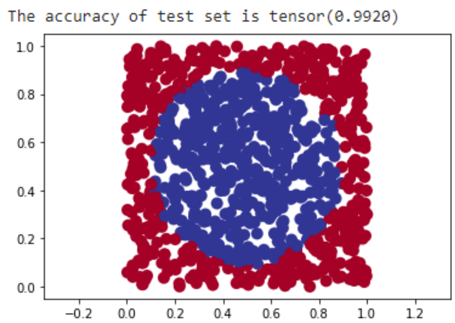


Figure 1: Architecture of built neural network



basic network with MSE and SGD



basic network with MSE and Adam

Table 1: Results of classification for test set

Experiments

We first tested the performance of the model in Fig. 2 trained with MSE loss function and SGD optimizer. We logged average loss of each epoch, computed and printed the final train and test errors and accuracy of each epoch. Model parameters are initialized in uniform distribution randomly. The detailed setting for training and testing are listed in Table. 2, we did 10 rounds and obtained curves of average loss and accuracy per epoch of each round for each configuration of network. And we show our results in Table.1 and Fig. 3.

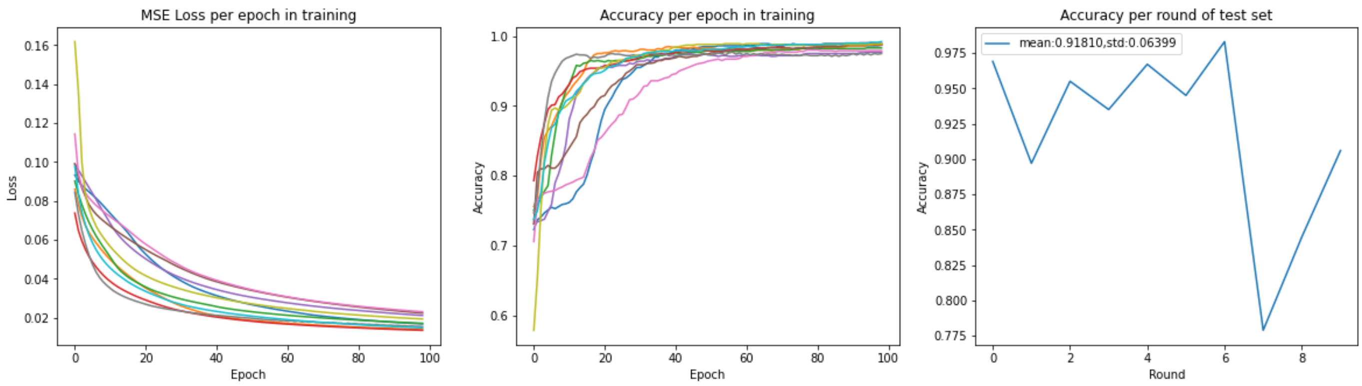


Figure 3: Loss of MSE and accuracy per epoch in 10 rounds(SGD)

And then we tested the base model trained with Adam optimizer, and recorded the results in Table.1 and Fig.4.

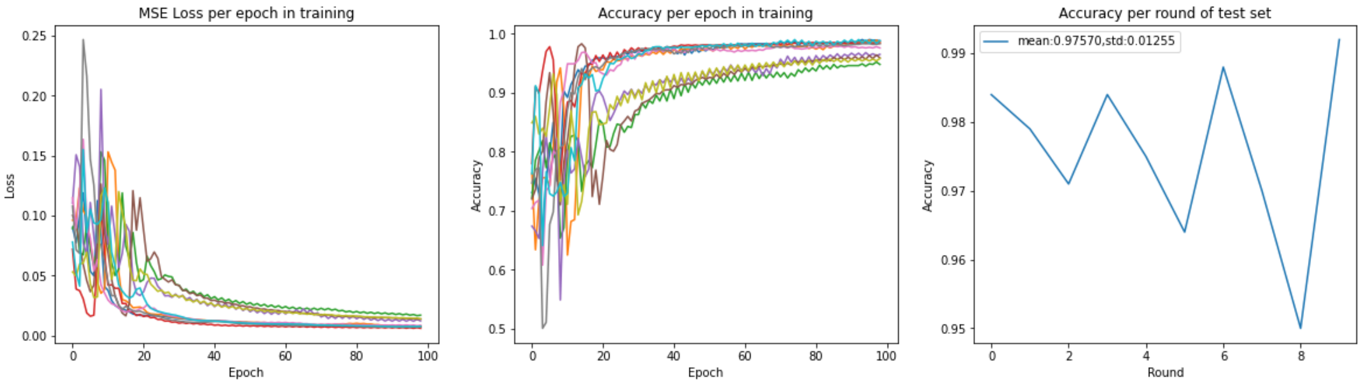


Figure 4: Loss of MSE and accuracy per epoch in 10 rounds(Adam)

Experiment	Epochs	Initialization	Learning Rate	Optimizer	Batch Size	Rounds
Basic Model(MSE)_1	100	Uniform distribution+He	1e-3	SGD	100	10
Basic Model(MSE)_2	100	Uniform distribution+He	1e-2	Adam	100	10

Table 2: Hyper-parameters of the built networks and settings of experiments

Experiment	Average Loss	Accuracy
Basic Model(MSE)_1	0.20572±0.00609	91.81±6.399
Basic Model(MSE)_2	0.22976±0.00792	97.57±1.255

Table 3: Results of experiments

Discussion

During our experiments, we observe that as the width and depth of built network increase, the gradients of parameters are susceptible to vanishing and the loss is sensitive to initialization and stuck in local minimum without further improvement of performance, so that the Relu and He initialization are employed to accelerate training. As shown in Table. 3, we obtain good training and testing performance for both experiments, and Adam achieves better average testing accuracy than SGD in our basic model.