Deep Learning Assignment one Report

Mariam Ali

November 2019

1 Introduction

Artificial neural networks are computational units inspired from the main unit of the human brain, aiming to learn a method from a number of layers of data points or what's referred to as hidden neurons. An artificial neural network is composed of input layers, layers of hidden neurons (Mainly for the learning procedure) and output layers. The main idea behind this assignment is to calculate the loss function of a convoluted estimated output, In other words, our Model is written as follows:

$$Y = W_o\Phi(W_{h2}\Phi(W_{h1}X)) \tag{1}$$

Where the W's in the previous equation are the weights that usually calculated in any regression problem through solving a minimization problem with respect to these weights, and Φ is the activation function that's being fed the input during the learning process.

2 Methods

A network class is created to initialize the network, which is a very good way to represent our network because it will contain the given parameters that we have for the network such as input and output layers and number of hidden neurons in the inner layers and also will have the needed functions to pass the information through the network and also train the network given few parameters that can be tuned through the training process. The last point to be mentioned about the network class is the calculated quantities that are typically the derivatives of the loss function and in out case the loss functions is the estimated output from the model I previously mentioned in Equation one and the test output data points which basically the training process, these derivatives are many quantities because of the convoluted function.

3 Parameters

What's believed to be the most challenging point in this exercise is to decide what are the most suitable parameters to be fed to the network and to the training process, So the first challenge was to decide how many hidden neurons are to be in the first and second hidden layers, so when I applied a brute force "grid search", it's between quotes because I guess it's not the most efficient grid search approach, I found that very decent loss trend is correlated with values between 240 and 360. Another crucial factor is the number of epochs and the learning rate, which is the number that the gradient descent changes with in order to reach a minimum, either local or if the trainer is lucky and smart enough choosing it, it will be a global minimum.

4 Conclusion and results

I found that a good guess for the number of the hidden neurons in the first and second layer is to be double or triple the number of input neurons but less and more than that lead to horrible errors as encountered in the notebook. Finally the learning rate needs to be between 0.01 and 0.001 for the loss to converge and the training error not to be close to zero or in other words not to over fit.

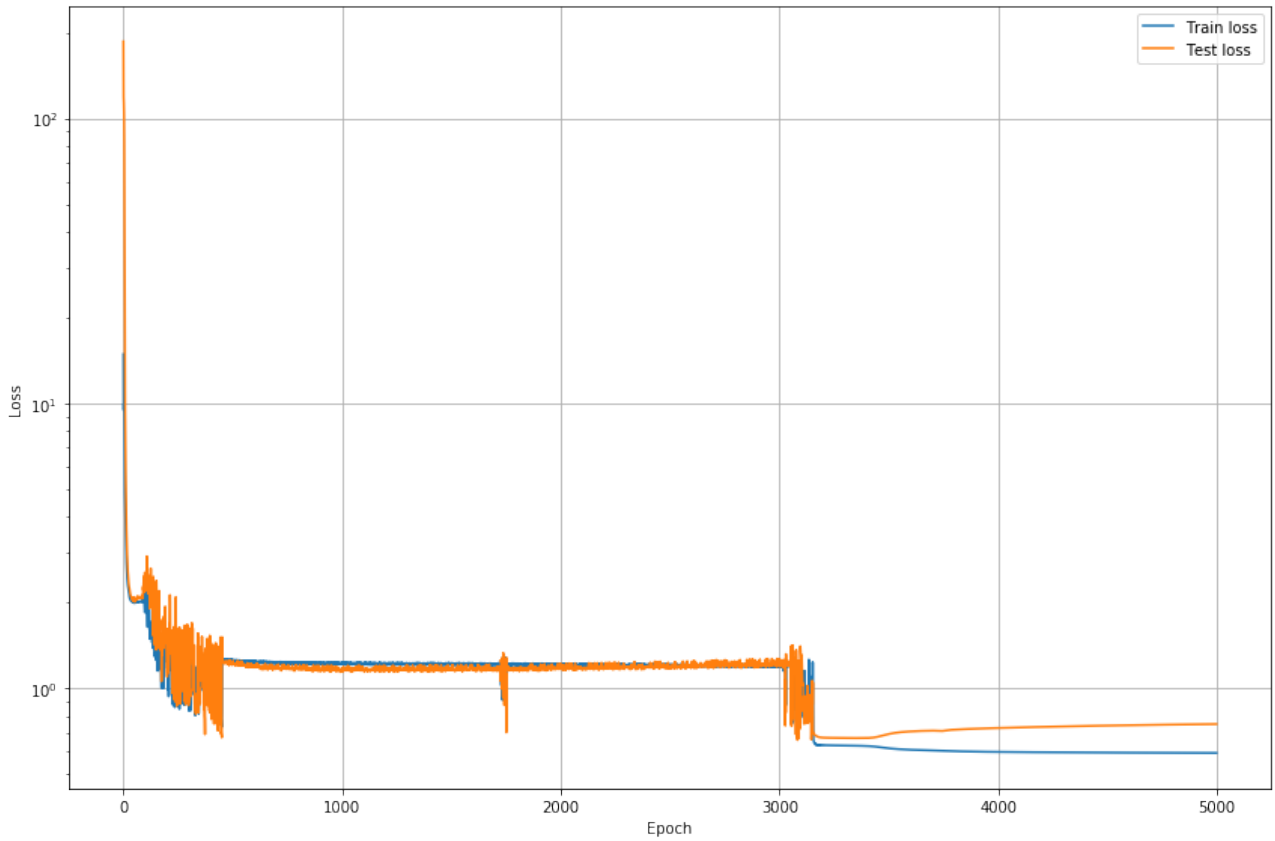

Figure 1: 240 hidden neurons in the first layer and 360 in the second

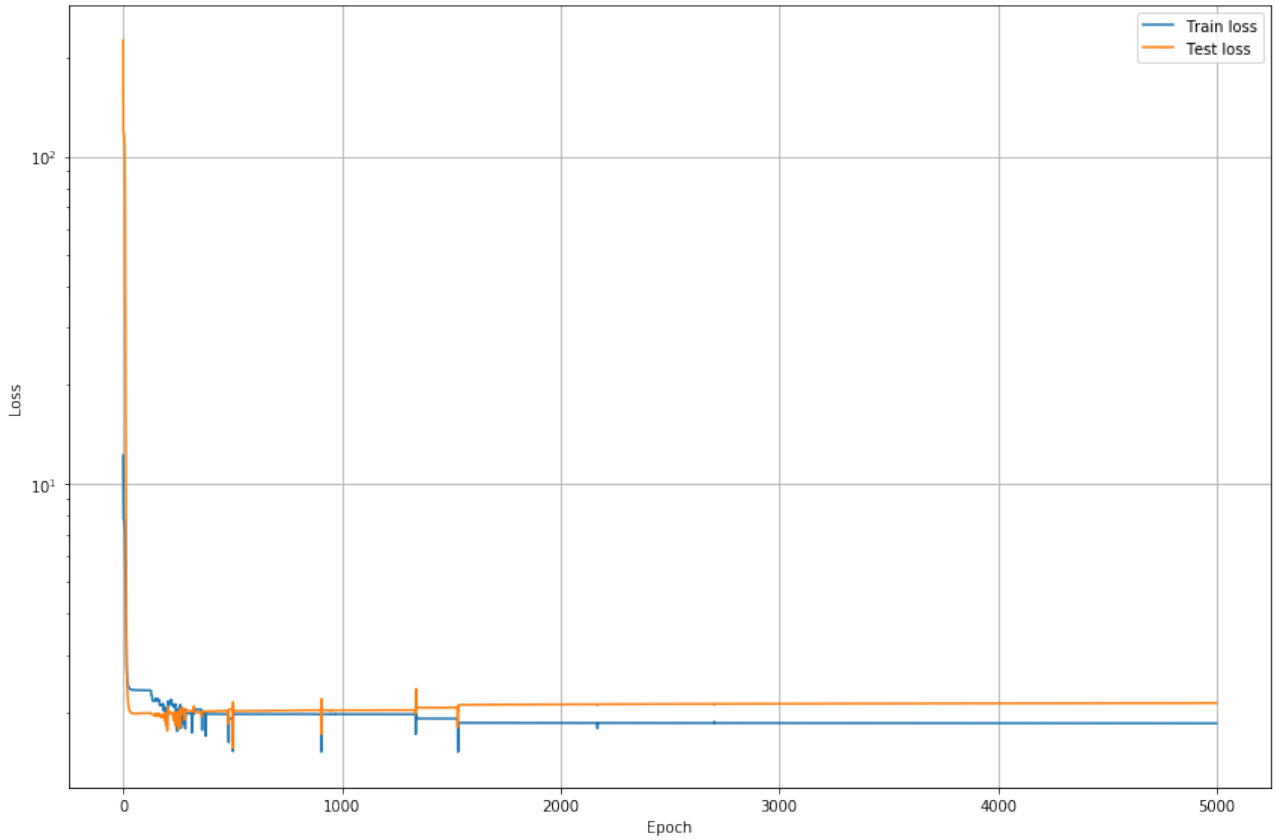

Figure 2: 360 hidden neurons in the second layer and 240 in the second

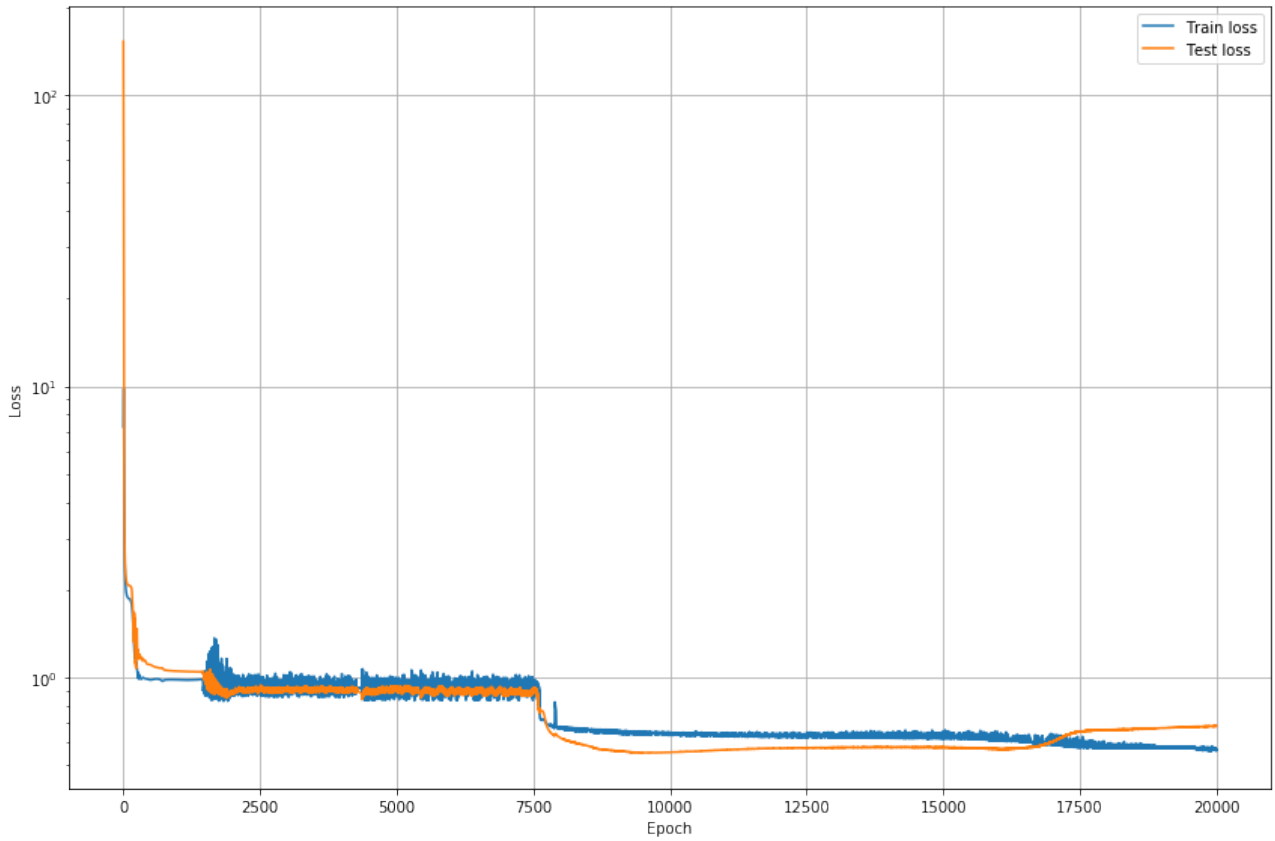

Figure 3: Training and testing loss trends according to training cycles

5 final parameters

Num. Hidden neurons in first layer	260
Num. Hidden neurons in second layer	330
Num of epochs	20000
Learning rate	0.001