Machine Learning Homework1

Yung-Ping Shen
Department of Applied Mathematics
National Chung Hsing University
Taichung, Taiwan
curt60079@gmail.com

I. Introduction

This is the first homework of machine learning. Our goals is use regression to predict the value of MEDV in housing dataset.

II. Related work

in this work, we use a 3-layer Perceptron to help us achieve our goals which included an input layer, a hidden layer and an output layer. And We use MSE as our loss function.

$$\mathfrak{f}(w) = \sum_i \left\| \widehat{y}^{(i)} - y^{(i)} \right\|$$

In our Perceptron, we use forward propagation to predict the result, and use backward propagation to update the weight in the Perceptron.

III. forward propagation

In the forward part, we will calculate the result according to the following equation.

$$\begin{aligned} Z^{(h)} &= (W^{(h)})^T A^{(in)} \\ A^{(h)} &= \phi(Z^{(h)}),\ which\ \phi(x) = \frac{1}{1+e^{-x}} \\ Z^{(out)} &= (W^{(out)})^T A^{(h)} \\ A^{(out)} &= \phi(Z^{(out)}) \end{aligned}$$

In the following equation $A^{(in)}$ means the input data, $(W^{(h)})^T$, $Z^{(h)}$ and $A^{(h)}$ means the parameters and data in hidden layer and $(W^{(out)})$, $Z^{(out)}$ and $A^{(out)}$ means the parameters and output in output layer.

IV. Backward propagation

In the backward part, we have to figure out how the weights are updated. The equation is as follows. First, we started at our loss function.

$$\begin{aligned} \mathfrak{f}(w) &= \sum_i \left\| \widehat{y}^{(i)} - y^{i} \right\|^2 \\ &= Tr(\left\| A^{(out)} - Y \right\|^T \cdot \left\| A^{(out)} - Y \right\|) \\ &= Tr((A^{(out)} - Y)^T (A^{(out)} - Y)) \end{aligned}$$

Next, we make a partial differentiation of $W_{i,j}^{(out)}$ in $\mathfrak{f}(w)$.

$$\begin{aligned} \frac{\partial \mathfrak{f}(w)}{\partial W_{i,j}^{(out)}} &= \frac{\partial Tr((A^{(out)} - Y)^T (A^{(out)} - Y))}{\partial (A^{(out)} - Y)} \cdot \frac{\partial (A^{(out)} - Y)}{\partial W_{i,j}^{(out)}} \\ &= 2(A^{(out)} - Y) \cdot \frac{\partial (A^{(out)} - Y)}{\partial A^{(out)}} \cdot \frac{\partial A^{(out)}}{\partial W_{i,j}^{(out)}} \\ &= 2(A^{(out)} - Y) \cdot \frac{\partial ((W_{i,j}^{(out)})^T A^h)}{\partial W_{i,j}^{(out)}} \\ &= 2A_j^h \delta_i^{out},\ which\ \delta_i^{out} = A^{(out)} - Y \end{aligned}$$

And next, we make a partial differentiation of $W_{i,j}^{(h)}$ in $\mathfrak{f}(w)$.

$$\begin{aligned} \frac{\partial \mathfrak{f}(w)}{\partial W_{i,j}^{(h)}} =& \frac{\partial Tr((A^{(out)} - Y)^T (A^{(out)} - Y))}{\partial (A^{(out)} - Y)} \cdot \frac{\partial (A^{(out)} - Y)}{\partial W_{i,j}^{(h)}} \\ =& 2(A^{(out)} - Y) \cdot \frac{\partial (A^{(out)} - Y)}{\partial A^{(out)}} \cdot \frac{\partial A^{(out)}}{\partial W_{i,j}^{(h)}} \\ =& 2(A^{(out)} - Y) \cdot \frac{\partial ((W_{i,j}^{(h)})^T A^h)}{\partial A^{(h)}} \cdot \frac{\partial A^{(h)}}{\partial W_{i,j}^{h}} \\ =& 2(A^{(out)} - Y) \cdot \frac{\partial ((W^{(out)})^T A^h)}{\partial A^h} \cdot \frac{\phi(Z^h)}{\partial Z^h} \cdot \frac{\partial Z^h}{\partial W_{i,j}^{(h)}} \\ =& 2(A^{(out)} - Y) \cdot \frac{((W^{out})^T A^h)}{\partial A^h} \\ & \cdot (\phi(Z^h) \odot (C - \phi(Z^h))) \cdot \frac{\partial ((W^h)^T A^{in})}{\partial W_{i,j}^{(h)}} \\ =& 2A_j^{in} \delta_i^h,\ which\ \delta_i^h = W^{(out)} \delta^{(out)} \odot \frac{\partial (Z^h)}{\partial Z^h} \end{aligned}$$

Finally, we can get these two equations as follow.

$$\begin{aligned} \frac{\partial \mathfrak{f}(w)}{\partial W_{i,j}^{(out)}} &= a_j^h \delta_i^{(out)} \\ \frac{\partial \mathfrak{f}(w)}{\partial W_{i,j}^{(h)}} &= a_j^{in} \delta_i^{(h)} \end{aligned}$$

V. Experimental result

The following is the result of our experiment on 3-layer Perceptron. We will adjust the three parameters of hidden unit, learning rate and minibatch, and compare the differences between them. In our experiments, we have standardized the data.

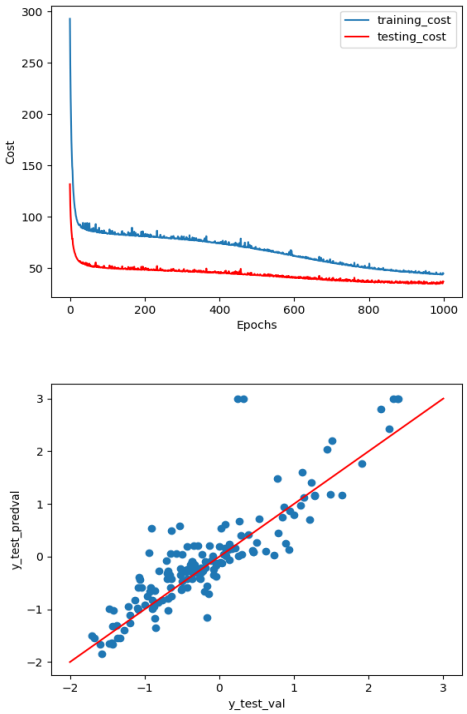

Fig. 1. hidden unit:100, learning rate:0.001, minibatch: 1

In the upper graph of Figure 1, we can see that the loss of training and testing is decreasing; in the lower diagram of Figure 1, the x-axis is the actual value of the test data, the y-axis is the predicted value of the test data, and the red line is A straight line with x=y, the closer the blue point is to the red line, the closer our prediction and actual value are.

In Figure 1, 2 and 3, we can see that adjusting minibatch has little effect on the loss of training and testing. In Figure 1, 4 and 5, we change the number of the hidden unit. We can see that when the number of hidden units is large, the loss of trining and testing will have a relatively large shock. In Figures 1, 6 and 7, we change the value of the learning rate. When the value of the learning rate is increased, the loss of training and testing will fluctuate greatly, although it can be reduced the loss value to a smaller time in a shorter time, but also has the risk of instability; and if the learning rate is adjusted, the loss of training and testing will be smoother but slower.

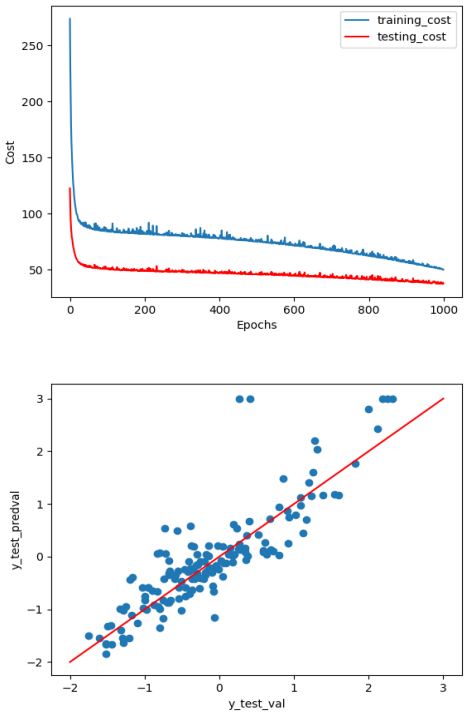

Fig. 2. hidden unit:100, learning rate:0.001, minibatch: 5

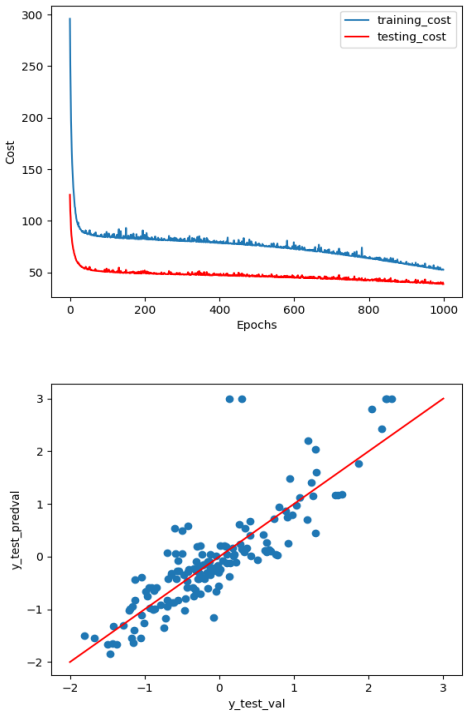

Fig. 3. hidden unit:100, learning rate:0.001, minibatch: 10

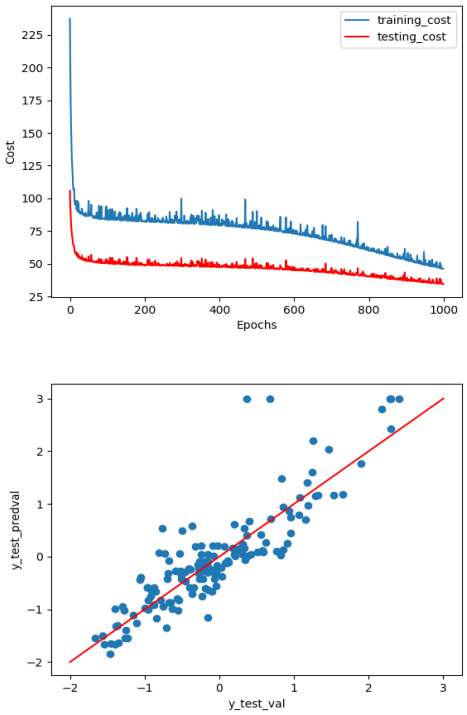

Fig. 4. hidden unit:200, learning rate:0.001, minibatch: 1

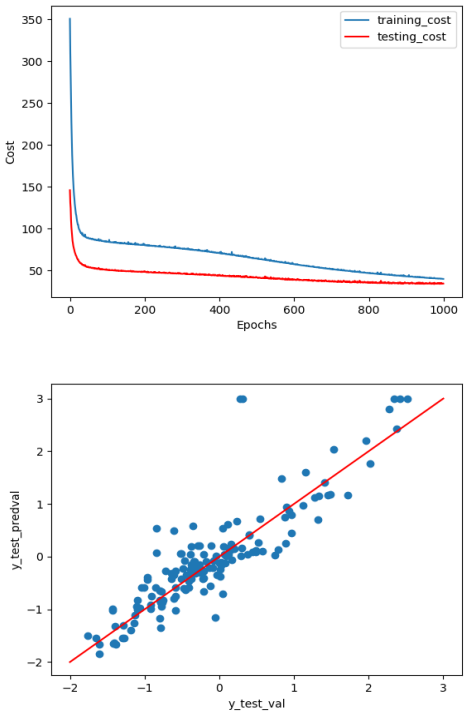

Fig. 5. hidden unit:50, learning rate:0.001, minibatch: 1

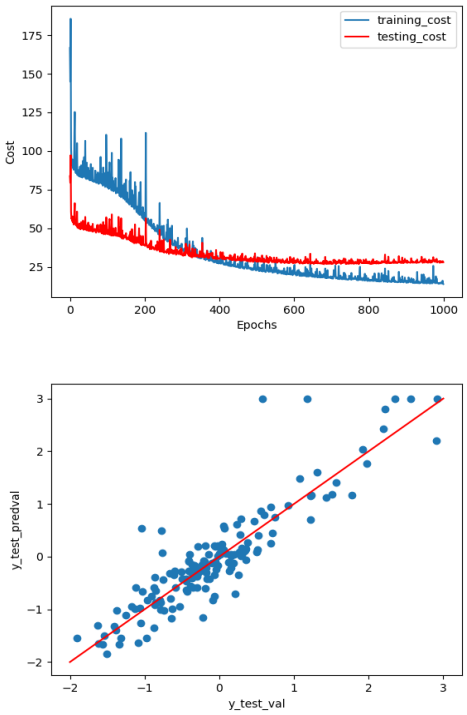

Fig. 6. hidden unit:100, learning rate:0.005, minibatch: 1

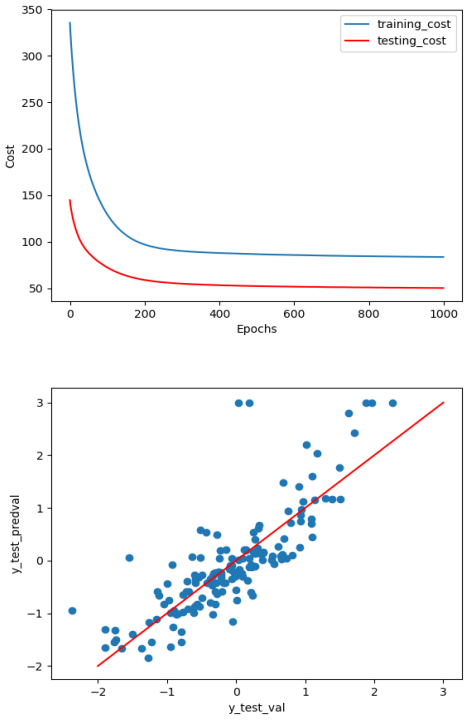

Fig. 7. hidden unit:100, learning rate:0.0001, minibatch: 1