# Machine Learning Homework 1

### A. Derive the forward and backward schemes for the Regression Problem

forward propagation: this question is a 3-layer MLP then $z = Wx$ and $y = Wz$ where z is the hidden layer, x is the input,W is the weight matrix,y is the output.
activation use Relu(if $x < 0$,then $x = 0$)
backward propagation:Gradient of Relu is if $x > 0, x = 1; x < 0, x = 0$,and the other function is same as teacher said
cost function:$\sum_i \|\hat{y}^i - y^i\|$

### B. Prepare the house data set, and then do the preprocessing to the data

$X\_train$ using the housing data except column['RAD'] and column['MEDV'],and use StandardScaler() to normalize $X\_train$
$y\_train$ using the column['MEDV'],the processing of y is divide by 10,then the values of y will not by to large

### C. Implement the 3-layer MLP)

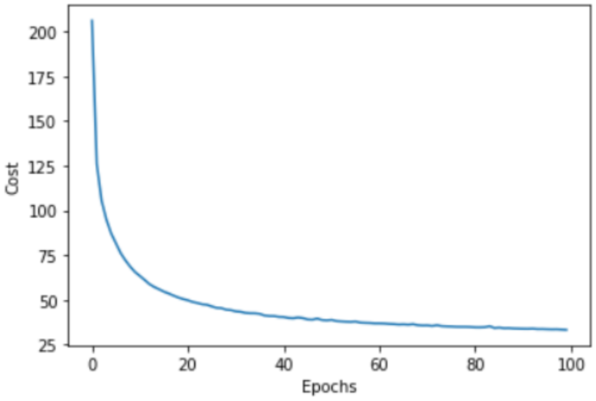


Fig. 1. cost of the 3-layer MLP

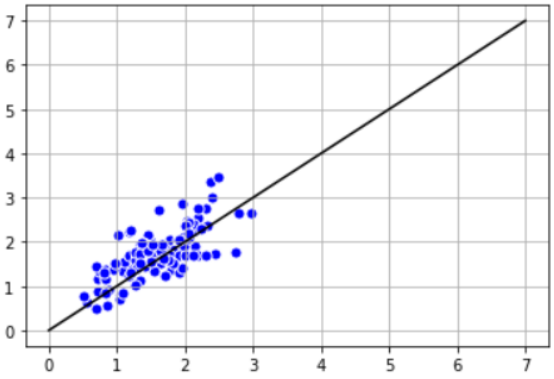
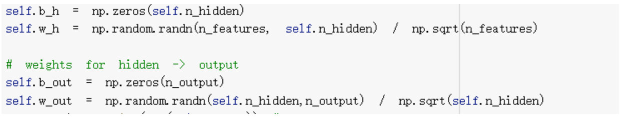


Fig. 2. similarity of y_test and y_pred

### D. Implement the Xavier initialization

the original model's initial weight is a zero matrix.if adding the Xavier initialization, it will give the weight values such that the cost decrease faster



```
self.b_h  =  np.zeros(self.n_hidden)
self.w_h  =  np.random.randn(n_features,  self.n_hidden)  /  np.sqrt(n_features)

#  weights  for  hidden  ->  output
self.b_out  =  np.zeros(n_output)
self.w_out  =  np.random.randn(self.n_hidden,n_output)  /  np.sqrt(self.n_hidden)
```

Fig. 3. Xavier initialization funcion

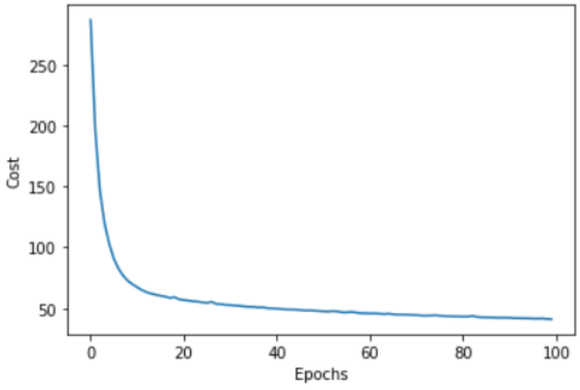


Fig. 4. cost of the 3-layer MLP with Xavier initialization

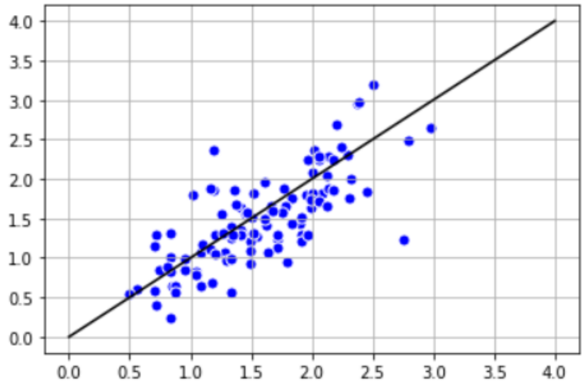
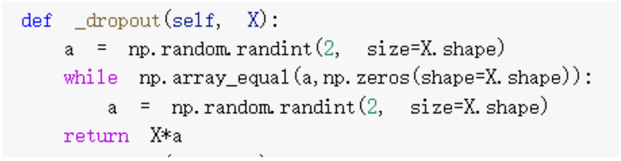


Fig. 5. similarity with Xavier initialization

### *E. Implement the Dropout*



```
def  _dropout(self,  X):
    a  =  np.random.randint(2,  size=X.shape)
    while  np.array_equal(a,np.zeros(shape=X.shape)):
        a  =  np.random.randint(2,  size=X.shape)
    return  X*a
```

Fig. 6. Dropout function

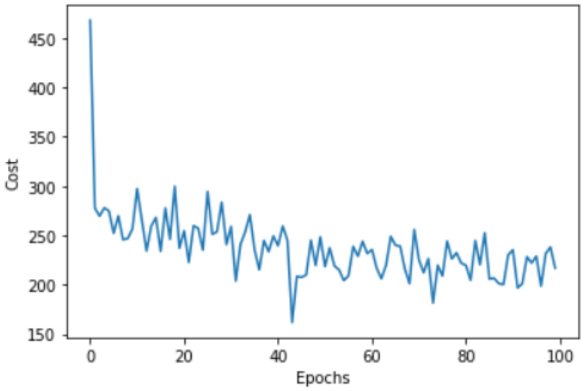


Fig. 7. cost of the 3-layer MLP with Dropout

### *F. Comparison*

the original model is almost like the model with Xavier initialization,but the cost of the model with dropout is not continuously decrease.I think the reason is the dropout function i wrote.because it is randomly drop nodes,so if it drop too many nodes the cost will increase

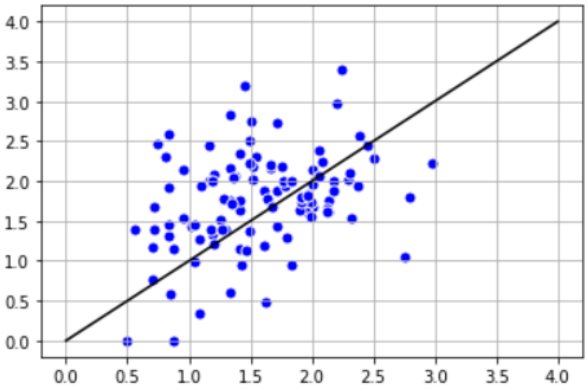


Fig. 8. similarity with Dropout