Machine Learning

Ufuk DOGAN

2018/10/19

1 Write down the vectorized equation for the forward pass.

$$X * W + B$$

X is the **input** vectors.
W is the **weight** of the vectors.
B is the **bias (constant)**

In the assignment we know that input is the vector so, we just need to do vector multipication and adding the bias for the forward pass.

2 Write down the vectorized equation for the MSE of the perceptron.

$$E = (d - net)^2/2$$

E is the **error**.
d is the our **target**.
net is the **output of the forward pass**

3 Determine the derivative of the error function w.r.t weights.

$$\frac{\partial E}{\partial W}(net - d) * X$$

net is the **output of the forward pass**.
d is the **target**.
X is the **input**

4 Write down the equation of the weight update by gradient descent, and explain why it can be used as a learning algorithm. Provide a positive and negative argument.

Learning rate is the one of the important parameters for the machine learning projects. If the learning rate is low, traning is more reliable. However, the optimization process take much time because, our steps will progress slowly.

On the other hand, if the learning rate is high, the difference between our two steps can be too high thus, sometimes our training can not converge. Weight changes can be so big that the optimizer overshoots the minimum and makes the loss worse.

$$\Delta W = \eta * (d - net) * X$$

$$W^{new} = W^{old} - \Delta W$$

net is the **output of the forward pass**.
X is the **input**
n is the **learning rate**.

5 Learning in multi-layer neural networks is usualy done with help of two methods: backpropagation and gradient descent. Describe briefly what role in the learning process each of the two has.

Exercise.py

```python
input_total = 4
w = [- 0.1, - 0.3, 0.2, 2.0]
x = [6.0, 4.0, 1.0, 1.0]


def weightChangeForOneIteration(weight, x_input):
    sum = 0
    for i in range(0, input_total):
        sum += w[i] * x[i]

    delta_weight = 0.02*(1-(sum))*x_input
    return delta_weight

for i in range(0, input_total):
    final_weight = w[i] - weightChangeForOneIteration(w[i], x[i])
    print("weight " + str(i+1) + "'s new weight is: " + str(final_weight))
```

```
weight 1's new weight is: -0.17200000000000001
weight 2's new weight is: -0.348
weight 3's new weight is: 0.188
weight 4's new weight is: 1.988
```

6 Learning in multi-layer neural networks is usualy done with help of two methods: backpropagation and gradient descent. Describe briefly what role in the learning process each of the two has.

Backpropagation algorithm contains 2 important steps. The first one is the forward pass and other one is the backward pass.

In the network we have some layers which can be Linear and also we have activation functions which can be tanh, sigmoid. The main purpose of the forward pass is, obtaining the prediction value.

On the other hand, backward pass computes the gradient using the error value which we computed in the forward pass. Also, in this step weights are updated too by using gradient descent. For computing the gradient, we have to calculate the partial derivative of the error (loss) respect to the weights.

In gradient descent as I wrote above, the weights are updated. We update the weights using the learning rate, target, inputs and the output of the forward pass.

We should iterate the forward pass until we reached the end of the network, and we continue backward pass until we reached the beginning of the network.