ISTANBUL TECHNICAL UNIVERSITY, COMPUTER ENGINEERING

BLG561E Deep Learning HW1 Math Questions

Anıl Öztürk

504181504

October 30, 2019

1 HINGE LOSS

We can assume this linear hypothesis function as a one-layer network. The input of the problem is a sixth order polynomial. We can write it as;

$$x_i = ax_i^6 + bx_i^5 + cx_i^4 + dx_i^3 + ex_i^2 + fx_i^1 + gx_i^0$$

$$x_{i0} = ax_i^6, x_{i1} = bx_i^5, x_{i2} = cx_i^4, x_{i3} = dx_i^3, x_{i4} = ex_i^2, x_{i5} = fx_i^1, x_{i6} = gx_i^0$$

The only linear feature in the input is $fx^1(x_5)$. We must process only the non-linear features of the input. So we should mask it in our linear hypothesis function. A typical linear hypothesis function would be like as follows;

$$h_\theta(X_i) = X_i^T\theta$$

1.1 HYPOTHESIS FUNCTION

We can define our linear hypothesis function to mask our linear features in input, as follows;

$$h_\theta(X_i) = X_i^T \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 = 0 \\ \theta_6 \end{bmatrix}$$

$$h_\theta(X_i) = x_{i0}\theta_0 + x_{i1}\theta_1 + x_{i2}\theta_2 + x_{i3}\theta_3 + x_{i4}\theta_4 + x_{i5}\theta_5 + x_{i6}\theta_6$$

1.2 DERIVED LOSS FUNCTION

Our loss function is **Hinge Loss** which is as follows;

$$l(\theta) = \sum_{i=1}^{k} max(1 - y_i h_\theta(x_i), 0)$$

We can derive the term like;

$$\frac{d[1 - y_i h_\theta(x_i)]}{dh_\theta(x_i)} = -y_i$$

We can derive the conditional upstream gradient equation **per data sample** as follows;

$$\frac{\partial l(\theta)}{\partial h_\theta(x_i)} = \begin{cases} 0, & \text{if } y_i h_\theta(x_i) \geq 1 \\ -y_i, & \text{otherwise} \end{cases}$$

We can derive the gradient equation for thetas wrt hinge loss as follows;

$$\frac{\partial h_\theta(x_i)}{\partial \theta} = \begin{bmatrix} \frac{\partial h_\theta}{\partial \theta_0}(x_i) \\ \frac{\partial h_\theta}{\partial \theta_1}(x_i) \\ \frac{\partial h_\theta}{\partial \theta_2}(x_i) \\ \frac{\partial h_\theta}{\partial \theta_3}(x_i) \\ \frac{\partial h_\theta}{\partial \theta_4}(x_i) \\ \frac{\partial h_\theta}{\partial \theta_5}(x_i) \\ \frac{\partial h_\theta}{\partial \theta_6}(x_i) \end{bmatrix} = \begin{bmatrix} x_{i0} \\ x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ x_{i5} \\ x_{i6} \end{bmatrix} = X$$

$$\frac{\partial h_\theta(x_i)}{\partial \theta_j} = x_{ij}$$

$$\nabla l(\theta_j) = \frac{\partial l(\theta)}{\partial h_\theta(x_i)} \frac{\partial h_\theta(x_i)}{\partial \theta_j}$$

$$\nabla l(\theta_j) = \begin{cases} 0, & \text{if } y_i h_\theta(x_i) \geq 1 \\ -y_i x_{ij}, & \text{otherwise} \end{cases}$$

1.3 Optimization Algorithm

The **i's** are for the **sample index**, **j's** are for the **theta and corresponding polynomial part index**.

```
i=0
while True do
    output = value from forward-pass
    foreach j do
        if (ground_truth[i] * output >= 1) then
            gradient = -ground_truth[i] * x[i,j]
        end
        else
            gradient = 0
        end
        thetas[j] -= learning_rate * gradient
    end
    i += 1
end
```

Algorithm 1: Pseudo-Code for Optimization

2 Derive the Gradients

2.1 Answer

$$\frac{\partial L}{\partial z_4} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_4}$$

$$l = -y log(\hat{y}) - (1-y) log(1-\hat{y})$$

$$\frac{\partial l}{\partial \hat{y}} = \frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$$

$$\frac{\partial \hat{y}}{\partial z_4} = \sigma(z_4)(1-\sigma(z_4))$$

$$\frac{\partial L}{\partial z_4} = \left(\frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}}\right)\sigma(z_4)(1-\sigma(z_4))$$

2.2 Answer

$$\delta_{z_4} = \frac{\partial L}{\partial z_4}$$

$$z_4 = W_{14} a_1 + b_4$$

$$\frac{\partial z_4}{\partial W_{14}} = a_1$$

$$\frac{\partial L}{\partial W_{14}} = \delta_{z_4} a_1$$

2.3 Answer

$$\delta_{a_1} = \frac{\partial L}{\partial a_1}$$

$$\frac{\partial L}{\partial W_{11}} = \delta_{a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial W_{11}}$$

$$a_1 = ReLU(z_1)$$

$$\frac{\partial a_1}{\partial z_1} = \begin{cases} 1, & \text{if } z_1 > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$z_1 = W_{11} x_1 + b_1$$

$$\frac{\partial z_1}{\partial W_{11}} = x_1$$

$$\frac{\partial L}{\partial W_{11}} = \begin{cases} \delta_{a_1} x_1, & \text{if } z_1 > 0 \\ 0, & \text{otherwise} \end{cases}$$

2.4 Answer

The addition that comes from the L2 Regularization can be defined as;

$$\frac{\lambda}{2} W^2$$

In backprop with every weight, we also should implement this addition's gradient;

$$\frac{d}{dW}\left(\frac{\lambda}{2} W^2\right) = \lambda W$$

For 2.1

Since there are no weight term between z_4 and loss output, the gradient should remain the same.

$$\delta_{z_4} = \left(\frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}}\right)\sigma(z_4)(1-\sigma(z_4))$$

For 2.2

$$\delta_{W_{14}} = \delta_{z_4} a_1 + \lambda W_{14}$$

For 2.3

$$\frac{\partial L}{\partial W_{11}} = \begin{cases} \partial_{a_1} x_1 + \lambda W_{11}, & \text{if } z_1 > 0 \\ \lambda W_{11}, & \text{otherwise} \end{cases}$$