10-707: Deep Learning, Fall 2017

Homework 1

Lecturer: Russ Salakhutdinov *Name: Yuan Liu(yuanl4)*

1 Problem 2

- K-nearest-neighbor regression:

  For knn, the estimator is given by:

  $$\hat{f}(x^*) = \frac{1}{k} \sum_{i \in N_k(x^*)} y_i$$

  $N_k(x^*)$ contains the indices of the k closest points of $x_1, \ldots, x_N$ to x^*. Then we can know:

  $$l_i(x^*; \mathcal{X}) = \begin{cases} 1, & i \in N_k(x^*) \\ 0, & otherwise \end{cases} \tag{1}$$

- Linear regression: For linear regression, the estimator is given by:

  $$\hat{f}(x^*) = {x^*}^T w$$

  where $w = (X^T X)^{-1} X^T y$, $y = (y_1, \ldots, y_N)^T$ and $X = (x_1, \ldots, x_N)^T$. Then we can get:

  $$\hat{f}(x^*) = {x^*}^T (X^T X)^{-1} X^T y$$

  So

  $$l_i(x^*; \mathcal{X}) = ({x^*}^T (X^T X)^{-1} X^T)_i$$

  $l_i(x^*; \mathcal{X})$ equals to the i_{th} component of ${x^*}^T (X^T X)^{-1} X^T$.

2 Problem 3

- Normalization: $p(x = 1|\mu) + p(x = -1|\mu) = \frac{1+\mu}{2} + \frac{1-\mu}{2} = 1$
- Mean: $E[x] = 1 * p(x = 1|\mu) + (-1) * p(x = -1|\mu) = \mu$
- Variance: $Var[x] = E[x^2] - (E[x])^2 = \frac{1+\mu}{2} + \frac{1-\mu}{2} - \mu^2 = 1 - \mu^2$
- Entropy: $Entropy = -\sum_{i \in \{-1,1\}} p(x = i|\mu) \log p(x = i|\mu) = -\frac{1+\mu}{2} \log \frac{1+\mu}{2} - \frac{1-\mu}{2} \log \frac{1-\mu}{2}$

3 Problem 4

Denote l is the correct label of x, and t is the label of x given by the dataset. So we can have the following formula:

$$p(l = 1|t = 1) = 1 - \epsilon$$

$$p(l = 1|t = 0) = \epsilon$$

$$\begin{aligned} p(l = 1|x; w) &= p(t = 1|x; w)p(l = 1|t = 1) + p(t = 0|x; w)p(l = 1|t = 0) \\ &= y(x, w)(1 - \epsilon) + (1 - y(x, w))\epsilon \end{aligned}$$

Then we can get:

$$l \sim Bernoulli\,(y(x, w)(1 - \epsilon) + (1 - y(x, w))\epsilon)$$

$$p(l|x, w) = [y(x, w)(1 - \epsilon) + (1 - y(x, w))\epsilon]^l\,[1 - y(x, w)(1 - \epsilon) - (1 - y(x, w))\epsilon]^{1-l}$$

$$\begin{aligned} \text{cost function} &= -\log p(l|x, w) \\ &= -l\log(y(x, w)(1 - \epsilon) + (1 - y(x, w))\epsilon) - (1 - l)log(1 - y(x, w)(1 - \epsilon) - (1 - y(x, w))\epsilon) \\ &= -l * \log(y - 2y\epsilon + \epsilon) - (1 - l) * \log(1 - y + 2y\epsilon - \epsilon) \end{aligned}$$

Where l is the label of training dataset, y is the output of neural network.

If $\epsilon = 0$, then

$$\text{cost function} = -l * \log y - (1 - l) * log(1 - y)$$

which is the standard negative log likelihood of binary classification.

4 Problem 5

First represent two networks in the following form:

- Sigmoid network: Input $x = (x_1, \ldots, x_p)^T$, First Layer: $a^{sig} = W_1^{sig}x + b_1^{sig}$, Activation function: $h^{sig} = \sigma(a^{sig})$, Second Layer: $o^{sig} = W_2^{sig}h^{sig} + b_2^{sig}$
- Tanh network: Input $x = (x_1, \ldots, x_p)^T$, First Layer: $a^{tanh} = W_1^{tanh}x + b_1^{tanh}$, Activation function: $h^{tanh} = tanh(a)$, Second Layer: $o^{tanh} = W_2^{tanh}h^{tanh} + b_2^{tanh}$

By observation we can have:

$$\sigma(2a) = \frac{tanh(a) + 1}{2}$$

First, we can assume:

$$W_1^{sig} = 2W_1^{tanh}, \quad b_1^{sig} = 2b_1^{tanh}$$

Then

$$a^{sig} = W_1^{sig}x + b^{sig} = 2W_1^{tanh}x + 2b_1^{tanh} = 2a^{tanh}$$

$$h^{sig} = \sigma(2a^{tanh}) = \frac{h^{tanh} + 1}{2}$$

Second, we can assume:

$$W_2^{sig} = 2W_2^{tanh}, \quad b_2^{sig} = b_2^{tanh} - W_2^{tanh} \cdot \mathbf{1}$$

Where $\mathbf{1}$ is a vector and all its component is 1.Then

$$o^{sig} = W_2^{sig} h^{sig} + b_2^{sig} = 2W_2^{tanh} \frac{h^{tanh} + \mathbf{1}}{2} + b_2^{tanh} - W_2^{tanh} \cdot \mathbf{1} = o^{tanh}$$

As a result, we can have the following equation:

$$W_1^{sig} = 2W_1^{tanh}, \quad b_1^{sig} = 2b_1^{tanh}$$

$$W_2^{sig} = 2W_2^{tanh}, \quad b_2^{sig} = b_2^{tanh} - W_2^{tanh} \cdot \mathbf{1}$$