CS6923 Machine Learning

Homework 3

Shang-Hung Tsai

Part 1

1.

(a)

$$P(C_1|x^{(1)})) = \frac{1}{1+e^{-(0.01*(-5)+0.01*3+0.01)}} = \frac{1}{1+e^{-0.01}} = 0.4975$$

(b) The predicted label is C_2 or 0 because 0.4975 < 0.5. This prediction is inconsistent with the label of this example.

(c)

$$P(C_1|x^{(2)})) = \frac{1}{1+e^{-(0.01*2+0.01*3+0.01)}} = \frac{1}{1+e^{-0.06}} = 0.515$$

The predict label is C_1 or 1 because 0.515 > 0.5. This prediction is inconsistent with the label of this example.

(d) 100%

(e)

$$Err(w, w_0|X) = -(1 * log(0.4975) + 1 * log(1 - 0.515)) = 1.4218$$

(f)

$$w_0 = 0.01 + 0.005 * (1 - 0.4975 + 0 - 0.515) = 0.0099375$$
$$w_1 = 0.01 + 0.005 * ((1 - 0.4975) * (-5) + (0 - 0.515) * 2) = -0.0077125$$
$$w_2 = 0.01 + 0.005 * ((1 - 0.4975) * 3 + (0 - 0.515) * 3) = 0.0098125$$

(g)

$$P(C_1|x^{(1)})) = \frac{1}{1+e^{-(-0.0077125*(-5)+0.0098125*3+0.0099375)}} = 0.519475$$
$$P(C_1|x^{(2)})) = \frac{1}{1+e^{-(-0.0077125*2+0.0098125*3+0.0099375)}} = 0.505987$$
$$Err(w, w_0|X) = -(1 * log(0.519475) + 1 * log(1 - 0.505987)) = 1.360130$$

(h) The cross-entropy went down after one iteration of the gradient descent. This is expected, because gradient descent should gradually improve the weights of the function and reduce error.

(i) 50%

2.

(a) The learning rate could be too high, and therefore causes oscillations and divergence.
Suggestion: decrease learning rate

(b) The learning rate could be too low, and therefore convergence becomes very slow.
Suggestion: increase learning rate

(c) The value could be stucked at a local minimum.
Suggestion: try run gradient descent again with a different initial value.

3.

Since we want to predict class C_1 if $h(x) > 0.30$, we have

$$\frac{1}{1+e^{-(w^Tx+w_0)}} > 0.3$$

$$1 > 0.3 + 0.3e^{-(w^Tx+w_0)}$$

$$\frac{7}{3} > e^{-(w^Tx+w_0)}$$

Take natural log on both side of the equation.

$$ln(\frac{7}{3}) > -(w^T x + w_0)$$

$$(w^T x + w_0) > -ln(\frac{7}{3}) = -0.847298$$

So we can predict positive if $g(x) > -0.847298$ and negative otherwise.

4.

For the original update function, the updated term is obtained by taking partial derivatives on the cross-entropy function

$$\delta w_j = -\eta \frac{\partial Err}{\partial w_j} = -\eta(-\sum_t (r^t - y^t)x_j) = \eta \sum_t (r^t - y^t)x_j$$

for j = 1,...d

$$\delta w_j = -\eta \frac{\partial Err}{\partial w_j} = -\eta(-\sum_t (r^t - y^t)) = \eta \sum_t (r^t - y^t)$$

for j = 0

If we want to use regularized version of the cross-entropy error function, we will need to also take partial derivatives on the regularized term, which gives us λw_j

Therefore, we get

$$\delta w_j = -\eta \frac{\partial Err}{\partial w_j} = -\eta(-\sum_t \left[(r^t - y^t)x_j\right] + \lambda w_j) = \eta(\sum_t \left[(r^t - y^t)x_j\right] - \lambda w_j)$$

for j = 1,...d

$$\delta w_j = -\eta \frac{\partial Err}{\partial w_j} = -\eta(-\sum_t (r^t - y^t)) = \eta \sum_t (r^t - y^t)$$

for j = 0

w_0 does not change because regularized term does not include w_0.