Homework 5

Jian Huang

2023/10/15

Task 1: Show the multi-class Softmax reduces to two-class Softmax cost when C = 2

The multi-class Softmax cost function:

$$g(w_0, ..., w_{C-1}) = \frac{1}{P}\sum_{p=1}^{P}[log(\sum_{j=0}^{C-1} e^{\dot{x}_p^T w_j}) - \dot{x}_p^T w_{yp}] \tag{1}$$

When C = 2, the equation will be simplified to eq(4):

$$g(w_0, w1) = \frac{1}{P}\sum_{p=1}^{P}[log(\sum_{j=0}^{1} e^{\dot{x}_p^T w_j}) - \dot{x}_p^T w_{yp}] \tag{2}$$

$$= \frac{1}{P}\sum_{p=1}^{P}[log(e^{\dot{x}_p^T w_0} + e^{\dot{x}_p^T w_1}) - \dot{x}_p^T w_{yp}] \tag{3}$$

$$= \frac{1}{P}\sum_{p=1}^{P}[log(e^{\dot{x}_p^T w_0} + e^{\dot{x}_p^T w_1}) - log(e^{\dot{x}_p^T w_{yp}})] \tag{4}$$

$$= \frac{1}{P}\sum_{p=1}^{P}[log(\frac{e^{\dot{x}_p^T w_0} + e^{\dot{x}_p^T w_1}}{e^{\dot{x}_p^T w_{yp}}})] \tag{5}$$

Given that $y_p \in \{1, -1\}$, we denote the two classes in labels of 1 and -1.
For points with the label $y_p = 1$, they all have $w_{yp} = w_1$ and each one of them has a softmax cost as:

$$g_p^{y_p=1} = log(\frac{e^{\dot{x}_p^T w_0} + e^{\dot{x}_p^T w_1}}{e^{\dot{x}_p^T w_1}}) \tag{6}$$

$$= log(1 + e^{\dot{x}_p^T (w_0 - w_1)}) \tag{7}$$

$$\tag{8}$$

While for points with the label $y_p = -1$, they all have $w_{yp} = w_0$ and each one of them has a softmax cost as:

$$g_p^{y_p=-1} = log(\frac{e^{\dot{x}_p^T w_0} + e^{\dot{x}_p^T w_1}}{e^{\dot{x}_p^T w_0}}) \tag{9}$$

$$= log(1 + e^{\dot{x}_p^T (w_1 - w_0)}) \tag{10}$$

$$\tag{11}$$

If we denote $\mathbf{w} = w_1 - w_0$, then the above two cases could be unified with their labels (y_p):

$$g(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P}[log(1 + e^{-y_p \dot{x}_p^T \mathbf{w}})] \tag{12}$$

which is exactly the **two-class softmax cost**s equation.

Task 2: Show the multi-class Softmax is equivalent to two-class Cross Entropy cost when C = 2

When C = 2 and taking advantage of the derivations in **Taks1**, the equation could be simplified to eq(4):

$$g(w_0, w1) = \frac{1}{P}\sum_{p=1}^{P}[log(\frac{e^{\dot{x}_p^T w_0} + e^{\dot{x}_p^T w_1}}{e^{\dot{x}_p^T w_{yp}}})]$$

Given that $y_p \in \{0, 1\}$, we denote the two classes in labels of 0 and 1.
For points with the label $y_p = 0$, they all have $w_{yp} = w_0$ and each one of them has a softmax cost as:

$$\begin{aligned} g_p^{y_p=0} &= log(\frac{e^{\dot{x}_p^T w_0} + e^{\dot{x}_p^T w_1}}{e^{\dot{x}_p^T w_0}}) && (13)\\ &= log(1 + e^{\dot{x}_p^T (w_1 - w_0)}) && (14)\\ & && (15) \end{aligned}$$

Again, if we denote $\mathbf{w} = w_1 - w_0$:

$$\begin{aligned} g_p^{y_p=0} &= log(1 + e^{\dot{x}_p^T \mathbf{w}}) && (16)\\ & && (17) \end{aligned}$$

For points with the label $y_p = 1$, they all have $w_{yp} = w_1$ and each one of them has a softmax cost as:

$$g_p^{y_p=1} = log(1 + e^{-\dot{x}_p^T \mathbf{w}}) \quad (18)$$

We can also express the above two cases together by using their labels $y_p \in \{0, 1\}$.

$$g(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P}[y_p log(1 + e^{-\dot{x}_p^T \mathbf{w}}) + (1 - y_p) log(1 + e^{\dot{x}_p^T \mathbf{w}})] \quad (19)$$

Also, notice that:

$$\begin{aligned} \sigma(\dot{x}_p^T \mathbf{w}) &= \frac{1}{1 + e^{-\dot{x}_p^T \mathbf{w}}} && (20)\\ \sigma(-x) &= 1 - \sigma(x) && (21) \end{aligned}$$

Then,

$$\begin{aligned} g(\mathbf{w}) &= \frac{1}{P}\sum_{p=1}^{P}[y_p log(\sigma(\dot{x}_p^T \mathbf{w})^{-1}) + (1 - y_p) log(1 - \sigma(\dot{x}_p^T \mathbf{w}))^{-1}] && (22)\\ &= -\frac{1}{P}\sum_{p=1}^{P}[y_p log(\sigma(\dot{x}_p^T \mathbf{w})) + (1 - y_p) log(1 - \sigma(\dot{x}_p^T \mathbf{w}))] && (23) \end{aligned}$$

which is exactly the equation for the **two-class cross entropy cost** equation.

Task 3: Implement the multi-class Softmax for a 4-class classification task

1. Choices of hyper-parameters

Hyperparameters	Value
Local optimization method	Adam
Initial weights	Random $\in$ (0, 1)
alpha	0.1
max iterations	200
lambda	0.01

Table 1: Caption

Reasonings:

1. Adam, a variation of gradient descent works very efficient with limited max iterations.
2. Since there is no singularities in Softmax cost function, thus random values to initialize weights works fine.
3. Depend on cost history plot and also iterations, combinations of alpha values of 0.1, 0.01 and 0.001, lambda values of 0.001 and 0.01 and iteration numbers of 200, 500 and 2000 were tested. The current choice of using alpha = 0.1, lambda = 0.001, iterations = 200 can reach convergence perfectly.

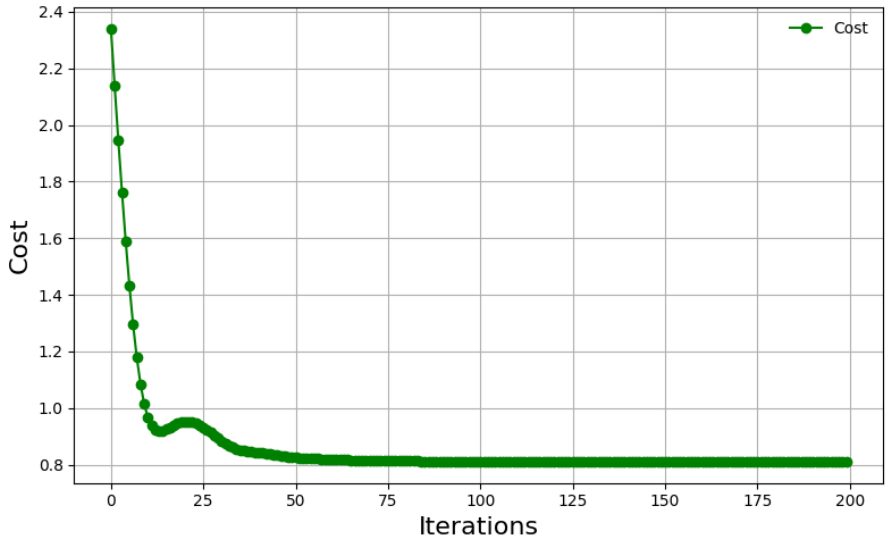

Figure 1: Cost history

2. The final cost and accuracy of your solution to the 4-class classification task

Final cost	0.8115
Misclassifications	11
Overall accuracy	72.5%

Table 2: Model performance

3. A figure showing the original data and regions of your model solution

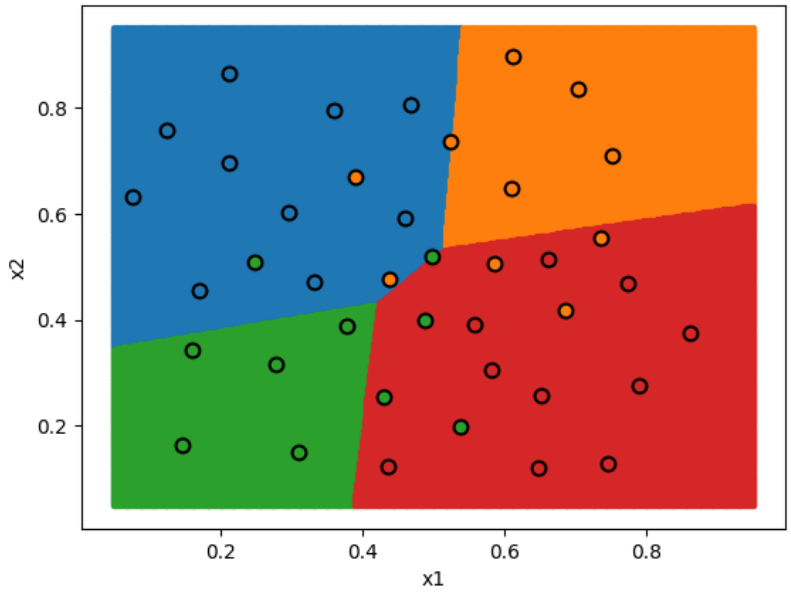

Figure 2: Final Model