Problem Set #3

Joe Michail
DATA SCI 423 – Machine Learning
Northwestern University

May 2, 2020

Question 6.5

The Cross Entropy cost is defined as:

$$g(\mathbf{w}) = -\frac{1}{P}\sum_{p=1}^{P} y_p \log(\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})) + (1-y_p)\log(1-\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})).$$

Making the substitution that $\sigma(-x) = 1-\sigma(x)$:

$$g(\mathbf{w}) = -\frac{1}{P}\sum_{p=1}^{P} y_p \log(\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})) + (1-y_p)\log(\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})).$$

Splitting the second term and combining like terms:

$$g(\mathbf{w}) = -\frac{1}{P}\sum_{p=1}^{P} y_p \log(\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})) - y_p\log(\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})) + \log(\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})).$$

Using the property of logs:

$$g(\mathbf{w}) = -\frac{1}{P}\sum_{p=1}^{P} y_p \log\left(\frac{\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})}{\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})}\right) + \log(\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})).$$

The log-quotient of sigmoid functions in the first term reduces to the argument:

$$g(\mathbf{w}) = -\frac{1}{P}\sum_{p=1}^{P} y_p\mathring{\mathbf{x}}_p^T\mathbf{w} + \log(\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})).$$

Taking the gradient with respect to $\mathbf{w}$:

$$\begin{aligned}\nabla_{\mathbf{w}} g(\mathbf{w}) &= -\frac{1}{P}\sum_{p=1}^{P} \nabla(y_p\mathring{\mathbf{x}}_p^T\mathbf{w}) + \nabla(\log(\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})) \\ &= -\frac{1}{P}\sum_{p=1}^{P} y_p\mathring{\mathbf{x}}_p + \frac{\nabla(\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w}))}{\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})}.\end{aligned}$$

Using the property of sigmoids:

$$\frac{d\sigma(y(x))}{dx} = \sigma(y(x))(1-\sigma(y(x))\frac{dy}{dx} = \sigma(y(x))\sigma(-y(x))\frac{dy}{dx}.$$

Therefore:

$$\begin{aligned}\nabla g(\mathbf{w}) &= -\frac{1}{P}\sum_{p=1}^{P} y_p \mathring{\mathbf{x}}_p + \frac{\nabla(\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w}))}{\sigma(-\mathring{\mathbf{x}}_p^T\mathbf{w})} \\ &= -\frac{1}{P}\sum_{p=1}^{P} y_p \mathring{\mathbf{x}}_p - \sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})\mathring{\mathbf{x}}_p.\end{aligned}$$

$$\boxed{\nabla g(\mathbf{w}) = -\frac{1}{P}\sum_{p=1}^{P}\left(y_p - \sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})\right)\mathring{\mathbf{x}}_p}$$

To find the Hessian, we take the gradient again:

$$\begin{aligned}\nabla^2 g(\mathbf{w}) = \nabla_{\mathbf{w}}^T \nabla g(\mathbf{w}) &= -\frac{1}{P}\sum_{p=1}^{P} \nabla_{\mathbf{w}}^T(y_p\mathring{\mathbf{x}}_p) - \nabla_{\mathbf{w}}^T(\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})\mathring{\mathbf{x}}_p) \\ &= \frac{1}{P}\sum_{p=1}^{P} \nabla_{\mathbf{w}}^T(\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w}))\mathring{\mathbf{x}}_p.\end{aligned}$$

Again using the property of sigmoids above:

$$\boxed{\nabla^2 g(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P}\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w})(1-\sigma(\mathring{\mathbf{x}}_p^T\mathbf{w}))\mathring{\mathbf{x}}_p\mathring{\mathbf{x}}_p^T}$$

Question 6.10

The perceptron cost is defined as:

$$g(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P}\max(0, -y_p\mathring{\mathbf{x}}_p^T\mathbf{w}).$$

The geometric definition of concavity is:

$$g(\lambda\mathbf{w_1} + (1-\lambda)\mathbf{w_2}) \leq \lambda g(\mathbf{w_1}) + (1-\lambda)g(\mathbf{w_2}), \lambda \in [0,1].$$

Combining them we get:

$$\frac{1}{P}\sum_{p=1}^{P}\max(0, -y_p\mathring{\mathbf{x}}_p^T(\lambda\mathbf{w_1} + (1-\lambda)\mathbf{w_2})) \leq \frac{\lambda}{P}\sum_{p=1}^{P}\max(0, -y_p\mathring{\mathbf{x}}_p^T\mathbf{w_1}) + \frac{1-\lambda}{P}\sum_{p=1}^{P}\max(0, -y_p\mathring{\mathbf{x}}_p^T\mathbf{w_2}).$$

Working with the left hand side, the second argument of max can be written as:

$$-y_p\mathring{\mathbf{x}}_p^T(\lambda\mathbf{w_1}+(1-\lambda)\mathbf{w_2}) = -y_p\mathring{\mathbf{x}}_p^T(\lambda(\mathbf{w_1}-\mathbf{w_2})+\mathbf{w_2})$$

If $\mathbf{w_1}$ and $\mathbf{w_2}$ lie just before and after the point of concavity, respectively, then $||\mathbf{w_1}-\mathbf{w_2}||_2 \sim \epsilon$. We can also write $\mathbf{w_1}-\mathbf{w_2}=\vec{\epsilon}$; since λ is bounded between 0 and 1:

$$-y_p\mathring{\mathbf{x}}_p^T(\lambda(\mathbf{w_1}-\mathbf{w_2})+\mathbf{w_2}) = -y_p\mathring{\mathbf{x}}_p^T(\lambda\vec{\epsilon}+\mathbf{w_2}) \approx -y_p\mathring{\mathbf{x}}_p^T\mathbf{w_2}.$$

Going back to the original equation:

$$\sum_{p=1}^{P}\max(0,-y_p\mathring{\mathbf{x}}_p^T\mathbf{w_2}) \leq \lambda\sum_{p=1}^{P}\max(0,-y_p\mathring{\mathbf{x}}_p^T\mathbf{w_1}) + (1-\lambda)\sum_{p=1}^{P}\max(0,-y_p\mathring{\mathbf{x}}_p^T\mathbf{w_2}).$$

Shifting terms:

$$\lambda\sum_{p=1}^{P}\max(0,-y_p\mathring{\mathbf{x}}_p^T\mathbf{w_2}) \leq \lambda\sum_{p=1}^{P}\max(0,-y_p\mathring{\mathbf{x}}_p^T\mathbf{w_1})$$

Since the initial constraint that was put on the problem was $\mathbf{w}_1-\mathbf{w}_2=\vec{\epsilon}$ where $||\epsilon|| << 1$, so we can write $\mathbf{w}_1 \approx \mathbf{w}_2$. Therefore, the terms within the sum are equivalent leaving:

$$\lambda \leq \lambda$$

which holds for all values of $\mathbf{w}_1$ and $\mathbf{w}_2$, so ReLU is a convex function.

Question 6.13

Please see attached code for the cost history plots, and classification info.

Question 6.15

Please see attached code for the perceptron fit using the `scikit-learn` Python package. The confusion matrix is:

	Predicted	Values
—	Good	Bad
Good	576	124
Bad	133	167

Question 6.16

Although half of example 6.12 is cut off in the text by a picture, I was generally able to get the same accuracy. In the case where all weights are equal, the total accuracy of both classes is about 95% and accuracy of the red class is 60%. When the red class has a weight 5x higher than the others, the accuracy decreases to 93% but the red accuracy increases to 80%. In the case where the red weights are 10x higher, the total accuracy decreases to 91% but the red accuracy is now 100%.