Problem Set #4

Joe Michail
DATA SCI 423 – Machine Learning
Northwestern University

May 17, 2020

Problem 7.2

See attached page for code. I was able to get down to 9 mis-classifications total.

Problem 7.3

See attached page for code. I was able to get no mis-classifications as stated in the problem.

Problem 7.4

Starting with equation (7.20):

$$g(\mathbf{w}_0, ..., \mathbf{w}_{C-1}) = \frac{1}{P}\sum_{p=1}^{P} \max_{j=0,...,C-1_{j \neq y_p}} (0, \mathring{\mathbf{x}}_p^T(\mathbf{w}_j - \mathbf{w}_{y_p})).$$

For $C = 2$:

$$g(\mathbf{w}_0, \mathbf{w}_1) = \frac{1}{P}\sum_{p=1}^{P} \max_{j \neq y_p}(0, \mathring{\mathbf{x}}_p^T(\mathbf{w}_j - \mathbf{w}_{y_p})).$$

From the argument in chapter 6: $\mathring{\mathbf{x}}_p^T\mathbf{w} > 0$ $(y_p = 1)$; $\mathring{\mathbf{x}}_p^T\mathbf{w} < 0$ $(y_p = -1)$. Combining the two gives: $-y_p\mathring{\mathbf{x}}_p^T\mathbf{w} < 0$, so:

$$g(\mathbf{w}_0, \mathbf{w}_1) = \frac{1}{P}\sum_{p=1}^{P} \max_{j \neq y_p}(0, -y_p\mathring{\mathbf{x}}_p^T(\mathbf{w}_j - \mathbf{w}_{y_p})).$$

In the binary case: $y_p\mathring{\mathbf{x}}_p^T\mathbf{w}_{y_p} = 0$. In addition, in the binary case $\mathbf{w}_0 = \mathbf{w}_1 = \mathbf{w}$ (since there is only one boundary and set of weights), so:

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

QED.

Problem 7.6

Starting with equation (7.24):

$$g(\mathbf{w}_0, ..., \mathbf{w}_{C-1}) = \frac{1}{P}\sum_{p=1}^{P}\log\left(1 + \sum_{j=0;j\neq y_p}^{C-1} e^{\mathring{\mathbf{x}}_p^T(\mathbf{w}_j - \mathbf{w}_{y_p})}\right).$$

Plugging in $C = 2$:

$$g(\mathbf{w}_0, \mathbf{w}_1) = \frac{1}{P}\sum_{p=1}^{P}\log\left(1 + \sum_{j=0;j\neq y_p}^{1} e^{\mathring{\mathbf{x}}_p^T(\mathbf{w}_j - \mathbf{w}_{y_p})}\right).$$

In the binary case $\mathbf{w}_0 = \mathbf{w}_1 = \mathbf{w}$ and $\mathring{\mathbf{x}}_p^T\mathbf{w}_{y_p} = 0$ so,

$$g(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P}\log\left(e^0 + e^{\mathring{\mathbf{x}}_p^T\mathbf{w}}\right).$$

The softmax is defined as $\text{softmax}(s_0, s_1) = \log(e^{s_0} + e^{s_1})$, therefore it obviously follows that:

$$g(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P}\log\left(e^0 + e^{\mathring{\mathbf{x}}_p^T\mathbf{w}}\right) = \frac{1}{P}\sum_{p=1}^{P}\log(e^0 + e^{\mathring{\mathbf{x}}_p^T\mathbf{w}}) = \frac{1}{P}\sum_{p=1}^{P}\text{softmax}(0, \mathring{\mathbf{x}}_p^T\mathbf{w}).$$

QED.

Problem 7.8

Start with softmax:

$$g(\mathbf{w}_0, ..., \mathbf{w}_{C-1}) = \frac{1}{P}\sum_{p=1}^{P}\log\left(\sum_{j=0}^{C-1} e^{\mathring{\mathbf{x}}_p^T\mathbf{w}_j}\right) - \mathring{\mathbf{x}}_p^T\mathbf{w}_{y_p}.$$

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

$$\nabla_{\mathbf{w}_c} g = \frac{1}{P}\sum_{p}\nabla_{\mathbf{w}_c}\log\left(\sum_{j} e^{\mathring{\mathbf{x}}_p^T\mathbf{w}_j}\right) - \nabla_{\mathbf{w}_c}\mathring{\mathbf{x}}_p^T\mathbf{w}_{y_p}.$$

The second term is a constant and applying the differentiation to the first term yields:

$$\nabla_{\mathbf{w}_c} g = \frac{1}{P}\sum_{p}\frac{e^{\mathring{\mathbf{x}}_p^T\mathbf{w}_c}}{\sum_d e^{\mathring{\mathbf{x}}_p^T\mathbf{w}_d}}\mathring{\mathbf{x}}_p^T.$$

Taking the gradient again with respect to $\mathbf{w}_c$ (to get the diagonal):

$$\begin{aligned}
\nabla^2 g &= \frac{1}{P}\sum_p \left[\frac{e^{\mathring{\mathbf{x}}_p^T \mathbf{w}_c}}{\sum_d e^{\mathring{\mathbf{x}}_p^T \mathbf{w}_d}} - \left(\frac{e^{\mathring{\mathbf{x}}_p^T \mathbf{w}_c}}{\sum_d e^{\mathring{\mathbf{x}}_p^T \mathbf{w}_d}}\right)^2\right] \mathring{\mathbf{x}}_p \mathring{\mathbf{x}}_p^T \\
&= \frac{1}{P}\sum_p \left[\frac{e^{\mathring{\mathbf{x}}_p^T \mathbf{w}_c}}{\sum_d e^{\mathring{\mathbf{x}}_p^T \mathbf{w}_d}} \left(1 - \frac{e^{\mathring{\mathbf{x}}_p^T \mathbf{w}_c}}{\sum_d e^{\mathring{\mathbf{x}}_p^T \mathbf{w}_d}}\right)\right] \mathring{\mathbf{x}}_p \mathring{\mathbf{x}}_p^T
\end{aligned}$$

Since all terms are positive, this means the sum of the eigenvalues (and the eigenvalues themselves) are positive, so the softmax is always convex.
Now the perceptron:

$$g(\mathbf{w}_0, ..., \mathbf{w}_{C-1}) = \frac{1}{P}\sum_p \max_{j=0,...,C-1}(0, \mathring{\mathbf{x}}_p^T(\mathbf{w}_j - \mathbf{w}_{y_p})).$$

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

$$\begin{aligned}
\nabla_{\mathbf{w}_c} g &= \frac{1}{P}\sum_p \max_{j=0,...,C-1}(0, \nabla_{\mathbf{w}_c}\mathring{\mathbf{x}}_p^T(\mathbf{w}_j - \mathbf{w}_{y_p})) \\
&= \frac{1}{P}\sum_p \max_{j=0,...,C-1}(\mathbf{0}, \mathring{\mathbf{x}}_p).
\end{aligned}$$

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

$$\begin{aligned}
\nabla^2 g &= \frac{1}{P}\sum_p \max_{j=0,...,C-1}(\mathbf{0}, \mathbf{0}) \\
&= \mathbf{0}.
\end{aligned}$$

Since the eigenvalues are all non-negative, this implies the perceptron cost function is always convex.

Problem 9.2

See attached page for code. In general, I was able to get the same results as in the textbook. The edge-based method classified about 2,000-3,000 more letters correctly than the pixel-based one after 20 iterations. Therefore, the edge-based detector reigns supreme here.