Homework#7

Yuhang Peng

6.4

Excercise 6.4

a). the softmax using a single hidden layer basis

$$g=\sum_{p=1}^{P}\log\left(1+e^{-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)}\right)$$

$$\frac{\partial g}{\partial b}=-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)\right)y_p$$

$$\frac{\partial g}{\partial w_n}=-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)\right)a(c_n+\bar{x}_p^T\bar{v}_n)\,y_p$$

$$\frac{\partial g}{\partial c_n}=-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)\right)a'(c_n+\bar{x}_p^T\bar{v}_n)\,w_n y_p$$

$$\nabla_{\bar{v}_n}g=-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)\right)a'(c_n+\bar{x}_p^T\bar{v}_n)\,\bar{x}_p w_n y_p$$

b). $$\bar{g}=\begin{bmatrix}\sigma\left(-y_1\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_1^T\bar{v}_m)\right)\right)\\ \vdots\\ \sigma\left(-y_P\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_P^T\bar{v}_m)\right)\right)\end{bmatrix}$$

$$\bar{t}_n=\begin{bmatrix}\tanh(c_n+\bar{x}_1^T\bar{v}_n)\\ \vdots\\ \tanh(c_n+\bar{x}_P^T\bar{v}_n)\end{bmatrix}\qquad \bar{s}_n=\begin{bmatrix}\operatorname{sech}^2(c_n+\bar{x}_1^T\bar{v}_n)\\ \vdots\\ \operatorname{sech}^2(c_n+\bar{x}_P^T\bar{v}_n)\end{bmatrix}$$

$$\frac{\partial g}{\partial b}=-[1,\dots,1]\begin{bmatrix}\sigma\left(-y_1\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_1^T\bar{v}_m)\right)\right)\\ \vdots\\ \sigma\left(-y_P\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_P^T\bar{v}_m)\right)\right)\end{bmatrix}\odot\begin{bmatrix}y_1\\ \vdots\\ y_P\end{bmatrix}$$

$$=-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)\right)y_p$$

$$\frac{\partial g}{\partial w_n}=-[1,\dots,1]\left(\begin{bmatrix}\sigma\left(-y_1\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_1^T\bar{v}_m)\right)\right)\\ \sigma\left(-y_P\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_P^T\bar{v}_m)\right)\right)\end{bmatrix}\odot\begin{bmatrix}\tanh(c_n+\bar{x}_1^T\bar{v}_n)\\ \vdots\\ \tanh(c_n+\bar{x}_P^T\bar{v}_n)\end{bmatrix}\odot\begin{bmatrix}y_1\\ \vdots\\ y_P\end{bmatrix}\right)$$

$$=-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)\right)a(c_n+\bar{x}_p^T\bar{v}_n)\,y_p$$

$$\frac{\partial g}{\partial c_n}=-[1,\dots,1]\left(\begin{bmatrix}\sigma\left(-y_1\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_1^T\bar{v}_m)\right)\right)\\ \vdots\\ \sigma\left(-y_P\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_P^T\bar{v}_m)\right)\right)\end{bmatrix}\odot\begin{bmatrix}\operatorname{sech}^2(c_n+\bar{x}_1^T\bar{v}_n)\\ \vdots\\ \operatorname{sech}^2(c_n+\bar{x}_P^T\bar{v}_n)\end{bmatrix}\odot\begin{bmatrix}y_1\\ \vdots\\ y_P\end{bmatrix}\right)w_n$$

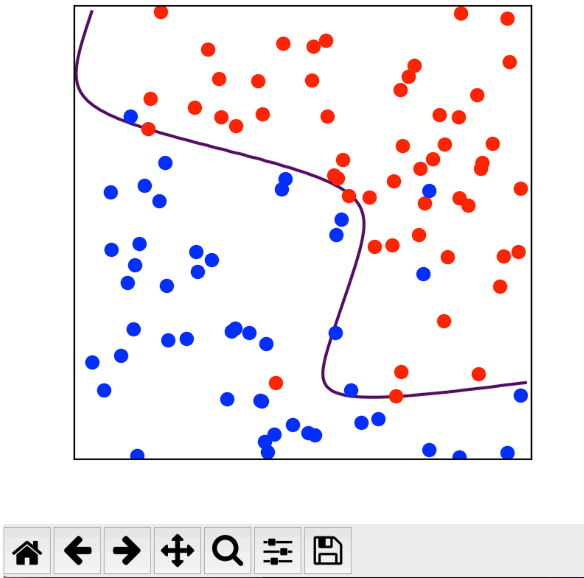
$$=-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)\right)a'(c_n+\bar{x}_p^T\bar{v}_n)\,w_n y_p$$

$$\nabla_{\bar{v}_n}g=-[x_1,\dots,x_P]\begin{bmatrix}\sigma\left(-y_1\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_1^T\bar{v}_m)\right)\right)\\ \vdots\\ \sigma\left(-y_P\left(b+\sum_{m=1}^{M}w_m\tanh(c_m+\bar{x}_P^T\bar{v}_m)\right)\right)\end{bmatrix}\odot\begin{bmatrix}\operatorname{sech}^2(c_n+\bar{x}_1^T\bar{v}_n)\\ \vdots\\ \operatorname{sech}^2(c_n+\bar{x}_P^T\bar{v}_n)\end{bmatrix}\odot\begin{bmatrix}y_1\\ \vdots\\ y_P\end{bmatrix}w_n$$

$$=-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}w_m a(c_m+\bar{x}_p^T\bar{v}_m)\right)\right)a'(c_n+\bar{x}_p^T\bar{v}_n)\,\bar{x}_p w_n y_p$$

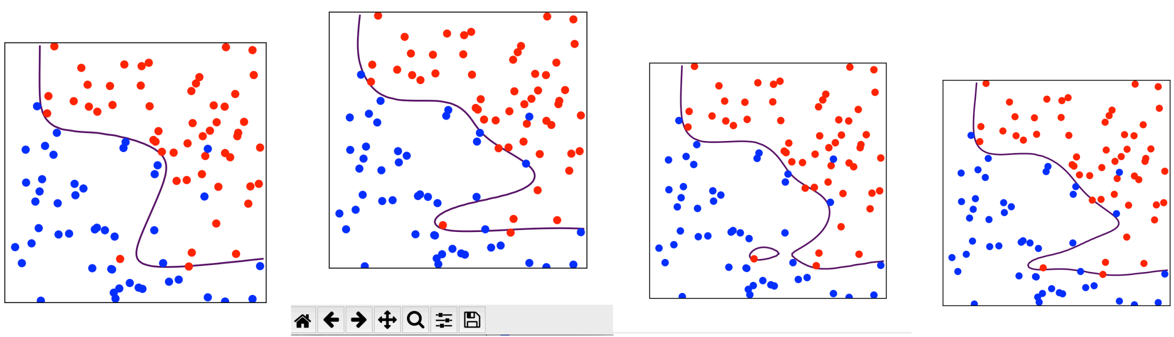
6.5

a)



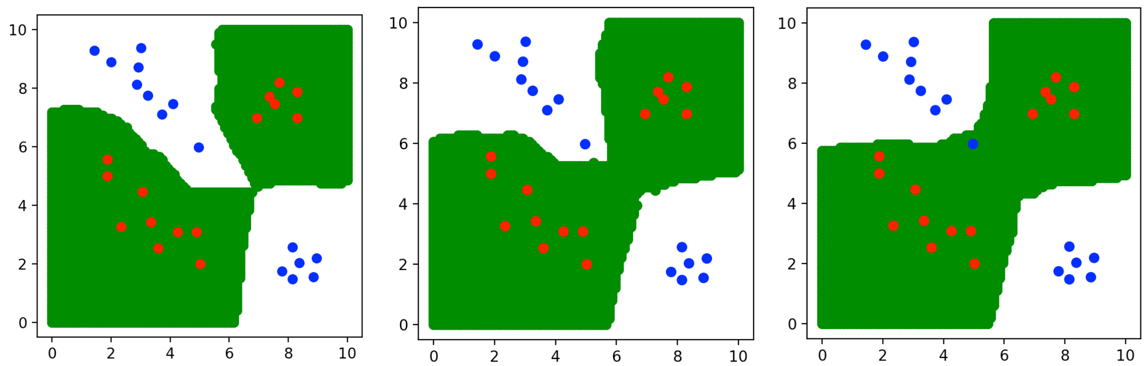
This is the result of the M = 4, from this figure, we can see that though some blue point still be divided into the red area, the function is easy to achieve. Due to the fact that I set the random initialization, each time the result will be a little be different.

b)



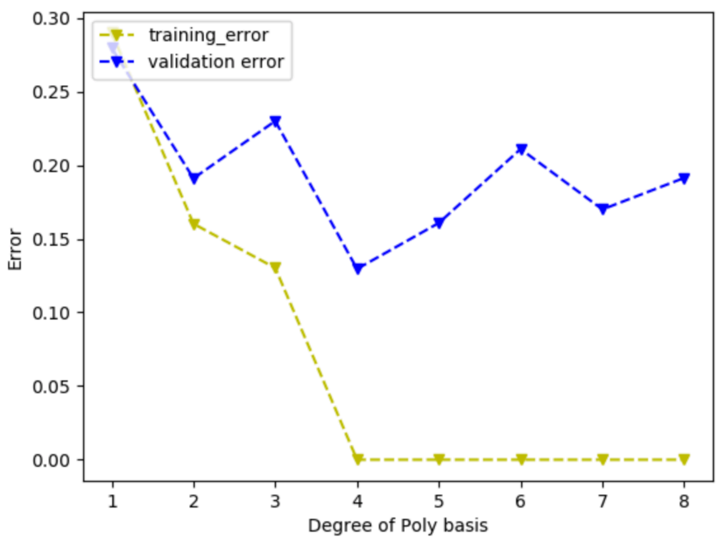
There are results of M = 5, M = 6, M = 8, M = 10 resepctively. From these figures, it is easy to find that in general instances of classification, analogous to what we saw with regression, adding more basis feature (increasing M) can result in fitting closely to the data we have while poorly to underlying function. Compare the result between each figure, the M = 4 & M = 5 will provide a better result, because the data points is clearly divided and function is fitting. But when we increasing the M, we will find overfitting here, like when M = 8, a little circle appear in the figure.

6.6



These are k = 1, k = 5, k = 10 respectively. Compare with the figure in the textbook, it will be a little different due to the size of point and plot method I select.

6.9



For this question, I am not quiet understand why the result is so unstable, I try Gradient Decent and Newton method. Both of them can produce a good curve, just like the one in the textbook, but I obtained this result by focusing on the gradient decent and running the code for nearly 20 times. For each time, I change the iteration time and the initialization values for gradient decent. I believe the iteration time must larger than 2000000 in order to obtain a curve similar with the one in the textbook. If you find some bugs inside my code, I will appreciate. Thanks.