ML Homework 2 Solution

Han Wang (hw2435@nyu.edu)

September 29, 2019

Problem 1

Prove that the set of classifiers $F = \{f : f(x) = sign(sin(\omega x)), \omega \geq 0\}$ operating in one dimension with input space $X = [0, 2\pi]$ is infinite.

Which, accoding to the definition of VC-dimension, can be translated into prove that the VC-dimension(F) = $infinite$.

Proof Consider input data $x_i = 2\pi 10^{-i}$ and output label as y_i. Set the parameter ω to

$$\omega = \frac{1}{2}(1 + \sum_{i=1}^{n} \frac{1-y_i}{2} 10^i) \tag{1-1}$$

Case 1: Correctly predicting all postive labels(When y_i=1) In those cases, rewrite ω as,

$$\omega = \frac{1}{2}(1 + \sum_{y_i=1} 10^i) \tag{1-2}$$

For any point $x_j = 2\pi 10^{-j}$,

$$\begin{aligned}
\omega x_j &= \pi 10^{-j}(1 + \sum_{y_i=-1} 10^i) \\
&= \pi 10^{-j}(1 + \sum_{y_i=-1, i\neq j} 10^i) \\
&= \pi(10^{-j} + \sum_{y_i=-1, i\neq j} 10^{i-j}) \\
&= \pi(10^{-j} + \sum_{y_i=-1, i<j} 10^{i-j} + \sum_{y_i=-1, i>j} 10^{i-j})
\end{aligned} \tag{1-3}$$

When $i > j$, terms $\sum 10^{i-j}$ are even integer,which can be rewritten into $2k$, for some $k \in \mathbb{N}$.

$$\omega x_j = \pi(10^{-j} + \sum_{y_i=-1, i<j} 10^{i-j}) + 2k\pi \tag{1-4}$$

The coefficient here on π is $10^{-j} + \sum_{y_i=-1,i<j} 10^{i-j}$ which is clearly smaller than $10^{-1} + 10^{-1}$ which is also smaller than 1. Denote also

$$0 < \delta = 10^{-j} + \sum_{y_i=-1,i<j} 10^{i-j} < 1 \tag{1-5}$$

Then, ωx_j meets the inequality of

$$2k\pi < \omega x_j = \pi\delta + 2k\pi < (2k+1)\pi \tag{1-6}$$

Which means $sin(\omega x_j) > 0$, in other words, the data is correctly classified.

Case 2: Correctly predicting all negtive labels(When y_i=1) On the other hand, when $y_j = -1$, for all data point $x_j = 2\pi 10^{-j}$,

$$\begin{aligned}
\omega x_j &= \pi 10^{-j}(1 + \sum_{y_i=-1} 10^i) \\
&= \pi 10^{-j}(1 + 10^j + \sum_{y_i=-1,i\neq j} 10^i) \\
&= \pi(10^{-j} + 1 + \sum_{y_i=-1,i\neq j} 10^{i-j}) \\
&= \pi(10^{-j} + 1 + \sum_{y_i=-1,i<j} 10^{i-j} + \sum_{y_i=-1,i>j} 10^{i-j}) \\
&= \pi(10^{-j} + \sum_{y_i=-1,i<j} 10^{i-j}) + \pi \sum_{y_i=-1,i>j} 10^{i-j} + \pi \\
&= \pi\delta + (2k+1)\pi
\end{aligned} \tag{1-7}$$

Where, $(2k+1)\pi < \omega x_j < (2k+2)\pi$, which leads to $sin(\omega x_j) < 0$, in other words, all negative data points are correctly classified.

In conclusion, all labeled data points can be correctly classified by $f(x)$ with paticular ω, meaning $F = f : f(x)$ can shatter any labeling data points. Thus, the proof for VC-dim$(F) = +\infty$ is complete.

Problem 2

Problem 2.a

First, consider back-propogation between the output layer and the hidden layer. Denote the sigmoid activation function as $\sigma(x) = \frac{1}{1+e^{-x}}$, denote loss function as

$$E = L(x_i) = -\sum_i (t_i log(x_i) + (1 - t_i) log(1 - x_i)) \tag{2.a-1}$$

Denote the weight between the output layer and hidden layer as w_{ji}. Denote the output for hidden layer as y_j, the output for the output layer is x_i. Perform backpropogation on w_{ji}:

$$\begin{aligned}
\frac{\partial E}{\partial w_{ji}} &= \frac{\partial L(x_i)}{\partial x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}} \\
&= \frac{\partial}{\partial x_i}(-\sum_i (t_i log(x_i) + (1 - t_i) log(1 - x_i)))\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}} \\
&= (\frac{1 - t_i}{1 - x_i} - \frac{t_i}{x_i})\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}} \\
&= (\frac{1 - t_i}{1 - x_i} - \frac{t_i}{x_i})\frac{\partial}{\partial s_i}\sigma(s_i)\frac{\partial s_i}{\partial w_{ji}} \\
&= (\frac{1 - t_i}{1 - x_i} - \frac{t_i}{x_i})\sigma'(s_i)\frac{\partial s_i}{\partial w_{ji}} \\
&= (\frac{1 - t_i}{1 - x_i} - \frac{t_i}{x_i})x_i(1 - x_i)\frac{\partial s_i}{\partial w_{ji}} \\
&= (x_i - t_i)\frac{\partial}{\partial w_{ji}}\sum_j y_j w_{ji} \\
&= (x_i - t_i)y_j
\end{aligned} \tag{2.a-2}$$

Where, we can denote term $x_i - t_i$ as δ.

Denote the weight between the hidden layer and the input layer as w_{kj}, perform backpropogation on w_{kj}:

$$
\begin{aligned}
\frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i \frac{\partial s_i}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i \frac{\partial}{\partial s_j}(\sum_j y_j w_{ji})\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i \frac{\partial}{\partial s_j}(\sum_j \sigma(s_j) w_{ji})\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i w_{ji} \sigma'(s_j)\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i w_{ji} y_j (1 - y_j)\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i w_{ji} y_j (1 - y_j)\frac{\partial}{\partial w_{kj}}\sum_j w_{kj} z_k \\
&= \sum_i \delta_i w_{ji} y_j (1 - y_j) z_k \\
&= \sum_i (x_i - t_i) w_{ji} y_j (1 - y_j) z_k
\end{aligned}
\tag{2.a-3}
$$

Problem 2.b

Denote the sigmoid activation function as $\sigma_1(x) = \frac{1}{1+e^{-x}}$. Denote softmax activation function as $\sigma_2(x_i) = \frac{e^{x_i}}{\sum_i e^{s_i}}$. Denote loss function as

$$
E = L(x_i) = -\sum_i t_i log(x_i) \tag{2.b-1}
$$

Perform backpropogation on w_{ji}:

$$
\begin{aligned}
\frac{\partial E}{\partial w_{ji}} &= \frac{\partial L(x_i)}{\partial x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}} \\
&= \frac{\partial}{\partial x_i}(-\sum_i t_i log(x_i))\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}} \\
&= -\frac{t_i}{x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}} \\
&= -\frac{t_i}{x_i}\frac{\partial}{\partial s_i}\sigma_2(s_i)\frac{\partial s_i}{\partial w_{ji}} \\
&= -\frac{t_i}{x_i}\frac{\partial}{\partial s_i}(\frac{e^{s_i}}{\sum_i e^{s_i}})\frac{\partial s_i}{\partial w_{ji}} \\
&= -\frac{t_i}{x_i}\frac{(e^{s_i})'(\sum_i e^{s_i}) - e^{s_i}(\sum_i e^{s_i})'}{(\sum_i e^{s_i})^2}\frac{\partial s_i}{\partial w_{ji}} \\
&= -\frac{t_i}{x_i}(\frac{e^{s_i}(\sum_i e^{s_i})}{(\sum_i e^{s_i})^2} - \frac{e^{x_i}e^{x_i}}{(\sum_i e^{s_i})^2})\frac{\partial s_i}{\partial w_{ji}} \\
&= -\frac{t_i}{x_i}(\frac{e^{s_i}}{\sum_i e^{s_i}} - \frac{e^{s_i}}{\sum_i e^{s_i}}\frac{e^{s_i}}{\sum_i e^{s_i}})\frac{\partial s_i}{\partial w_{ji}} \\
&= -\frac{t_i}{x_i}(x_i - x_i^2)\frac{\partial s_i}{\partial w_{ji}} \\
&= -t_i(1 - x_i)\frac{\partial}{\partial w_{ji}}\sum_j y_j w_{ji} \\
&= -t_i(1 - x_i)y_j
\end{aligned}
\tag{2.b-2}
$$

Where, we can denote term $-t_i(1 - x_i)$ as δ.

Now, perform backpropogation on $w_j i$:

$$\begin{aligned}
\frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i (\frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial s_j})\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i \frac{\partial s_i}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i \frac{\partial}{\partial s_j}(\sum_j y_j w_{ji})\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i \frac{\partial}{\partial s_j}(\sum_j \sigma_1(s_j) w_{ji})\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i w_{ji} \sigma_1'(s_j)\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i w_{ji} y_j (1 - y_j)\frac{\partial s_j}{\partial w_{kj}} \\
&= \sum_i \delta_i w_{ji} y_j (1 - y_j)\frac{\partial}{\partial w_{kj}}\sum_j w_{kj} z_k \\
&= \sum_i \delta_i w_{ji} y_j (1 - y_j) z_k \\
&= -\sum_i t_i (1 - x_i) w_{ji} y_j (1 - y_j) z_k
\end{aligned} \tag{2.b-3}$$

Problem 3

In this problem, the empirical risk of linear perceptron would be:

$$R(\theta) = \frac{1}{N}\sum_{i=1}^{N} step(-y_i\theta^T x_i) \tag{3-1}$$

The binary error are defined as:

$$E(\theta) = \frac{N_{misclassified}}{N_{total}} \tag{3-2}$$

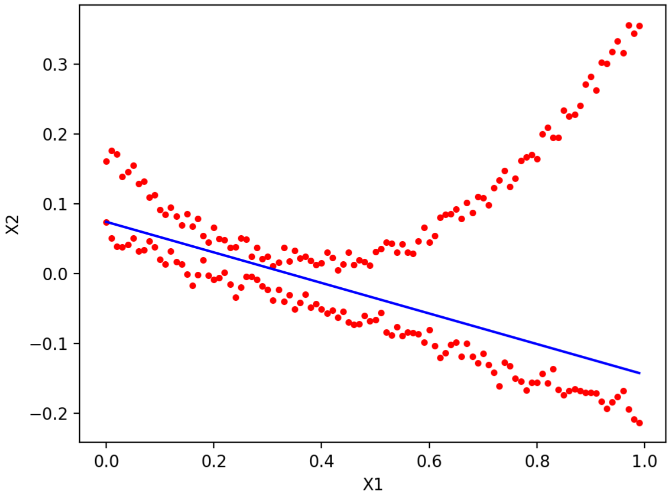
In this particular problem, I apply SGD(Stochastic Gradient Descent) to optimize perceptron model. With that being said, if data point (x_i, y_i) is misclassified,

$$\theta^{t+1} = \theta^t - \eta\nabla_\theta R^{per}|_{\theta_t} = \theta^t + y_i x_i \tag{3-3}$$

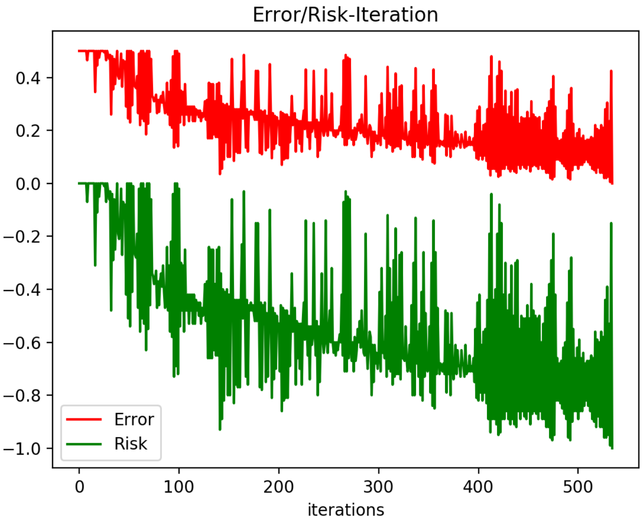
Which, set η to 1 without loss of generality.

Meanwhile, in terms of traning, the weight is initiated randomly. I set the maximum iterations to 5000, which normaly wouldn't take as many. Apply SGD optimization to perceptron model untill convergence or reach of maximum iteration.

The plot of decision boundary is as followed:

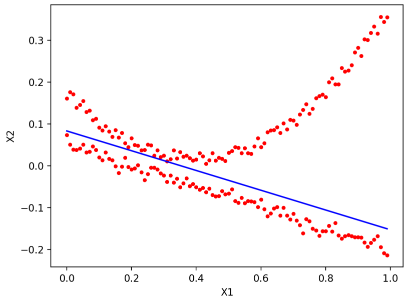


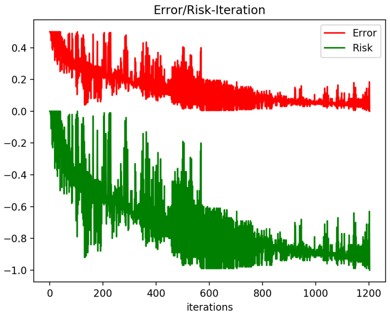
The Risk/Error to iterations plot is as followed:



It took total 534 iterations unitll convergence.

While with another set random initialization of weights, it tooks 1203 iterations untill convergence, which is longer. In this case, the plots are given below:





Problem 4

Our objective is to maximize the value of function H, which can rewrite into minimizing the value of H', which is:

$$min_p H' = min_p \sum_{k=1}^{N} p_k log(p_k) \tag{4-1}$$

Consider distribution $\{p_k | k = 1, 2, ..., N\}$ is a N-dimensional vector $\mathbf{x}$ which meets the constrain of $\sum_{i=1}^{D} x_i = 1$. Again, this can be rewritten into:

$$min_{\mathbf{x}} H'(x) = min_{\mathbf{x}} \mathbf{x}^T log(\mathbf{x}) \tag{4-2}$$

Rewrite the constrain using vectors by introducing $\mathbf{I}$ as a N-dimension vector with each entry set to 1:

$$\mathbf{I}^T \mathbf{x} = 1 \tag{4-3}$$

Using Lagrange multiplier λ to optimize under the above constrain, this problem can be rewritten into:

$$\begin{aligned} min_{\mathbf{x}}\, max_{\lambda} f(\mathbf{x}) &= min_{\mathbf{x}}\, max_{\lambda} H' - \lambda(equlity = 0) \\ &= min_{\mathbf{x}}\, max_{\lambda} \mathbf{x}^T log(\mathbf{x}) - \lambda(\mathbf{I}^T \mathbf{x} - 1) \end{aligned} \tag{4-4}$$

Take the derivative of f(p):

$$\begin{aligned} \frac{\partial}{\partial p} f(p) &= \frac{\partial}{\partial p} \mathbf{x}^T log(\mathbf{x}) - \lambda(\mathbf{I}^T \mathbf{x} - 1) \\ &= (\mathbf{I} + log(\mathbf{x})) - \lambda \mathbf{I} \end{aligned} \tag{4-5}$$

Set the derivative to 0, then we get

$$\begin{aligned} (\mathbf{I} + log(\hat{\mathbf{x}})) - \lambda \mathbf{I} &= 0 \\ log(\hat{\mathbf{x}}) &= (\lambda - 1)\mathbf{I} \\ \hat{\mathbf{x}} &= e^{\lambda - 1} \mathbf{I} \end{aligned} \tag{4-6}$$

For $\hat{\mathbf{x}}$ meet the constrain that $\mathbf{I}^T \mathbf{x} = 1$, so that

$$\begin{aligned} \mathbf{I}^T e^{\lambda - 1} \mathbf{I} &= 1 \\ e^{\lambda - 1} \cdot N &= 1 \\ \lambda &= 1 - log\, N \end{aligned} \tag{4-7}$$

Put back the value of λ to $\hat{\mathbf{x}}$, we get

$$\hat{\mathbf{x}} = e^{\lambda-1}\mathbf{I} = e^{1-log\,N-1}\mathbf{I} = e^{-log\,N}\mathbf{I} = \frac{1}{N}\mathbf{I} \tag{4-8}$$

Which means, the distribution $\{p_k|k = 1, 2, ..., N\}$ to maximize the entropy H is $\{p_k = \frac{1}{N}|k = 1, 2, ..., N\}$.