CSCI 567, Fall'19
Victor Adamchik

Problem Set #2

Due: Oct 9, 2019 11:59 pm

Instructions

Submission: Assignment submission will be via courses.uscden.net. By the submission date, there will be a folder named `Homework 2` set up in which you can submit your files. Please be sure to follow all directions outlined here.

You can submit multiple times, but only the last submission counts. That means if you finish some problems and want to submit something first and update later when you finish, that's fine. In fact you are encouraged to do this: that way, if you forget to finish the homework on time or something happens (remember Murphy's Law), you still get credit for whatever you have turned in.

Problem sets must be typewritten or neatly handwritten when submitted. In both cases, your submission must be a single PDF. It is strongly recommended that you typeset with LaTeX. There are many free online LaTeX editors that are convenient to use (e.g Overleaf). You can also use offline editor such as TeXShop.

Please also follow the rules below:

- The file should be named as Firstname_Lastname_USCID.pdf e.g., Jeff_Dean_8675309045.pdf.
- Do not have any spaces in your file name when uploading it.
- Please include your name and USCID in the header of your report as well.

Collaboration: You may discuss with your classmates. However, you need to write your own solutions and submit separately. Also in your written report, you need to list with whom you have discussed for each problem. Please consult the syllabus for what is and is not acceptable collaboration.

Note on notation: Unless stated otherwise, scalars are denoted by small letter in normal font, vectors are denoted by small letters in bold font and matrices are denoted by capital letters in bold font.

Problem 1 Convergence of Perceptron Algorithm (30 points)

In this problem you need to show that when the two classes are linearly separable, the perceptron algorithm will converge. Specifically, for a binary classification dataset of N data points, where every x_i has a corresponding label $y_i \in \{-1, 1\}$ and is normalized: $\|\mathbf{x}_i\| = \sqrt{\mathbf{x}_i^T \mathbf{x}_i} = 1, \forall i \in \{1, 2, \ldots, N\}$, the perceptron algorithm proceeds as below:

```
while not converged do
    Pick a data point $(\mathbf{x}_i, y_i)$ randomly
    Make a prediction $y = \text{sign}\left(\mathbf{w}^T \mathbf{x}_i\right)$ using current $\mathbf{w}$
    if $y \neq y_i$ then
        $\mathbf{w} \leftarrow \mathbf{w} + y_i \mathbf{x}_i$
    end
end
```

Algorithm 1: Perceptron

In other words, weights are updated right after the perceptron makes a mistake (weights remain unchanged if the perceptron makes no mistakes). Let the (classification) margin for a hyperplane $\mathbf{w}$ be $\gamma(\mathbf{w}) = \min_{i \in [N]} \frac{|\mathbf{w}^T \mathbf{x}_i|}{\|\mathbf{w}\|}$ (convince yourself that $\gamma(\mathbf{w})$ is the smallest distance of any data point from the hyperplane). Let $\mathbf{w}_{opt}$ be the optimal hyperplane, i.e. it linearly separates the classes with maximum margin. Note that since data is linearly separable there will always exist some $\mathbf{w}_{opt}$. Let $\gamma = \gamma(\mathbf{w}_{opt})$.

Following the steps below, you will show that the perceptron algorithm makes a finite number of mistakes that is at most γ^{-2}, and therefore the algorithm must converge.

1.1 Show that if the algorithm makes a mistake, the update rule moves it towards the direction of the optimal weights $\mathbf{w}_{opt}$. Specifically, denoting explicitly the updating iteration index by k, the current weight vector by $\mathbf{w}_k$, and the updated weight vector by $\mathbf{w}_{k+1}$, show that, if $y_i \mathbf{w}_k^T \mathbf{x}_i < 0$, we have

$$\mathbf{w}_{k+1}^T \mathbf{w}_{opt} \geq \mathbf{w}_k^T \mathbf{w}_{opt} + \gamma \left\|\mathbf{w}_{opt}\right\| \tag{1}$$

(5 points)

Hint: *Consider* $\left(\mathbf{w}_{k+1} - \mathbf{w}_k\right)^T \mathbf{w}_{opt}$ *and consider the property of* $\mathbf{w}_{opt}$.

Solution:

$$\begin{aligned} \mathbf{w}_{k+1} &= \mathbf{w}_k + y_i \mathbf{x}_i \\ \mathbf{w}_{k+1}^T \mathbf{w}_{opt} &= \mathbf{w}_k^T \mathbf{w}_{opt} + y_i \mathbf{x}_i^T \mathbf{w}_{opt} \quad \textbf{(3 points)} \end{aligned} \tag{2}$$

Since, $y_i \mathbf{x}_i^T \mathbf{w}_{opt} = \left|\mathbf{x}_i^T \mathbf{w}_{opt}\right|$($\mathbf{W}_{opt}$ perfectly separates the data), by the definition of γ

$$\mathbf{w}_{k+1}^T \mathbf{w}_{opt} \geq \mathbf{w}_k^T \mathbf{w}_{opt} + \gamma \left\|\mathbf{w}_{opt}\right\| \textbf{ (2 points)} \tag{3}$$

Note: In this case, we cannot say that the update moves $\mathbf{w}$ to exactly the optimal direction, but the magnitude of projection of $\mathbf{w}$ on $\mathbf{w}_{opt}$ has increased, which means apart from other things, the component of weights in optimal direction increases.

1.2 Show that the length of updated weights does not increase by a large amount. Mathematically show that, if $y_i {\mathbf{w}_k}^T x_i < 0$

$$\|\mathbf{w}_{k+1}\|^2 \leq \|\mathbf{w}_k\|^2 + 1 \tag{4}$$

. (5 points)

Hint: *Consider* $\|\mathbf{w}_{k+1}\|^2$ *and substitute* $\mathbf{w}_{k+1}$.

Solution:

$$\begin{aligned} \|\mathbf{w}_{k+1}\|^2 &= \mathbf{w}_{k+1}^T \mathbf{w}_{k+1} = (\mathbf{w}_k + y_i \mathbf{x}_i)^T (\mathbf{w}_k + y_i \mathbf{x}_i) \\ &= \|\mathbf{w}_k\|^2 + 2 y_i \mathbf{w}_k^T \mathbf{x}_i + y_i^2 \mathbf{x}_i^T \mathbf{x}_i \end{aligned} \textbf{(3 points)} \tag{5}$$

Input x_i has norm 1 and the algorithm has made a mistake so $y_i \mathbf{w}_k^T \mathbf{x}_i \leq 0$ **(2 points)**

$$\|\mathbf{w}_{k+1}\|^2 = \|\mathbf{w}_k\|^2 + 2 y_i \mathbf{w}_k^T \mathbf{x}_i + y_i^2 \mathbf{x}_i^T \mathbf{x}_i \leq \|\mathbf{w}_k\|^2 + 1 \tag{6}$$

1.3 Assume that the initial weight vector $\mathbf{w}_0 = \mathbf{0}$ (an all-zero vector). Using results from Problem 1.1 and 1.2 , show that for any iteration $k+1$, with M being the total number of mistakes the algorithm has made for the first k iterations, we have

$$\gamma M \leq \|\mathbf{w}_{k+1}\| \leq \sqrt{M} \tag{7}$$

Hint: *use Cauchy-Schwartz inequality* $\mathbf{a}^T \mathbf{b} \leq \|\mathbf{a}\| \|\mathbf{b}\|$ *and telescopic sum.* (15 points)

Solution: By repeatedly applying results from Problem 2.1 for 1 to M mistakes and summing them up.

$$\mathbf{w}_{k+1}^T \mathbf{w}_{opt} \geq \mathbf{w}_0^T \mathbf{w}_{opt} + M\gamma \|\mathbf{w}_{opt}\| \textbf{ (3 points)} \tag{8}$$

since $w_0 = 0$

$$\mathbf{w}_{k+1}^T w_{opt} \geq M\gamma \|\mathbf{w}_{opt}\| \textbf{ (3 points)} \tag{9}$$

Due to Cauchy-Schwartz inequality,

$$\begin{aligned} M\gamma \|\mathbf{w}_o pt\| \leq & \mathbf{w}_{k+1}^T \mathbf{w}_{opt} \leq \|\mathbf{w}_{k+1}\| \|\mathbf{w}_{opt}\| \\ M\gamma \leq & \|\mathbf{w}_{k+1}\| \end{aligned} \textbf{(3 points)} \tag{10}$$

Similarly, use results of Problem 2.2 repeatedly and sum to them all to conclude

$$\|\mathbf{w}_{k+1}\|^2 \leq \|\mathbf{w}_0\|^2 + M \textbf{(3 points)} \tag{11}$$

Since, initial weights are 0, conclude that

$$\|\mathbf{w}_{k+1}\|^2 \leq M \textbf{(3 points)} \tag{12}$$

1.4 Using result of Problem 1.3, conclude $M \leq \gamma^{-2}$. (5 points)

Solution: Sovle $\gamma M \leq \sqrt{M}$ for M

Problem 2 Logistic Regression (30 points)

Recall that the logistic regression model is defined as:

$$p(y=1|\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x}+b\right) \tag{13}$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{14}$$

Given a training set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbb{R}^{K\times 1}$ and $y_n \in \{0,1\}$, we will minimize the cross-entropy error function to solve $\mathbf{w}$.

$$\begin{aligned}\min_{\mathbf{w},b} L(\mathbf{w},b) &= \min_{\mathbf{w},b} -\sum_n \{y_n \log\left[p\left(y_n=1|\mathbf{x}_n\right)\right] + (1-y_n)\log\left[p\left(y_n=0|\mathbf{x}_n\right)\right]\} \\ &= \min_{\mathbf{w},b} -\sum_n \left\{y_n \log\left[\sigma\left(\mathbf{w}^T\mathbf{x}_n+b\right)\right] + (1-y_n)\log\left[1-\sigma\left(\mathbf{w}^T\mathbf{x}_n+b\right)\right]\right\}\end{aligned} \tag{15}$$

2.1 Please derive the update rule for w using Gradient Descent (GD) method. **(10 points)**

Solution:

$$\begin{aligned}\mathbf{w}^{(t+1)} &\leftarrow \mathbf{w}^{(t)} - \eta\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} \\ &= \mathbf{w}^{(t)} - \eta\sum_n \left[\sigma\left(\mathbf{w}^T\mathbf{x}_n+b\right) - y_n\right]\mathbf{x}_n\end{aligned} \tag{16}$$

2.2 Suppose we have four training samples $(x_1,y_1) = (1,0)$, $(x_2,y_2) = (1,1)$, $(x_3,y_3) = (1,1)$ and $(x_4,y_4) = (1,1)$. Suppose our logistic regression model is $p(y=1|x) = \sigma(wx)$ We initialize this model with $w=0$ and use learning rate = 0.001. When using GD to optimize this model, after one batch iteration, what's the training accuracy? **(15 points)**

Solution: Calculate the gradient,

$$\frac{\partial L(w)}{\partial w} = \sum\{[\sigma(wx_n) - y_n]\,x_n\}\ \textbf{(5 points)} \tag{17}$$

Substitute the data points and then set w = 0,

$$\frac{\partial L(w)}{\partial w} = [\sigma(w)-0] + [\sigma(w)-1] + [\sigma(w)-1] + [\sigma(w)-1] = -1 \tag{18}$$

Therefore, according to the GD update rule, $w = 0 - 0.001 * (-1) = 0.001$ **(5 points)**

Predictions on training data:

$$\begin{aligned}\hat{y}_1 &= \mathbb{I}\left[\sigma\left(w^* x_1\right) > 0.5\right] = 1 \neq y_1 \\ \hat{y}_2 &= \mathbb{I}\left[\sigma\left(w^* x_2\right) > 0.5\right] = 1 = y_2 \\ \hat{y}_3 &= \mathbb{I}\left[\sigma\left(w^* x_3\right) > 0.5\right] = 1 = y_3 \\ \hat{y}_4 &= \mathbb{I}\left[\sigma\left(w^* x_4\right) > 0.5\right] = 1 = y_3\end{aligned} \tag{19}$$

The training accuracy is $\frac{3}{4}$. **(5 points)**

2.3 Based on the model we get in problem 2.2, if we have a test dataset containing three samples: $(x_1,y_1) = (-1,0)$, $(x_2,y_2) = (1,1)$, $(x_3,y_3) = (1,0)$, what is the test accuracy? **(5 points)**

Solution: The test accuracy is $\frac{2}{3}$.

Problem 3 Backpropagation (40 points)

Suppose we have a Multi-Class Neural Networks defined below. An illustration is provided in Fig. 1. Please answer the following questions.

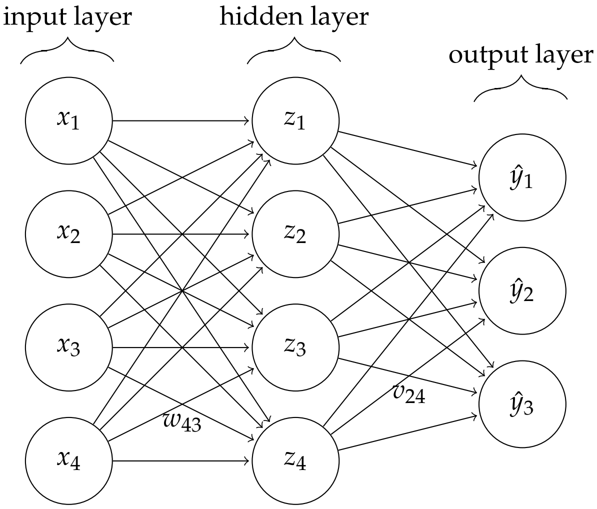

Figure 1: A neural network with one hidden layer.

Forward Propagation. For multi-class classification, we use softmax layer with cross-entropy loss as output. In the hidden layer, we use *tanh* activation function. The forward propagation can be expressed as:

$$\text{input layer} \qquad x_i, \tag{20}$$

$$\text{hidden layer} \qquad z_k = tanh\left(\sum_{i=1}^{4} w_{ki}x_i\right), tanh(\alpha) = \frac{e^{\alpha} - e^{-\alpha}}{e^{\alpha} + e^{-\alpha}} \tag{21}$$

$$\text{output layer} \qquad \hat{y}_j = softmax(o_j) = \frac{\exp(o_j)}{\sum_{i=1}^{3} \exp(o_i)}, \text{ where } o_j = \sum_{k=1}^{4} v_{jk}z_k \tag{22}$$

$$\text{loss function} \qquad L(y, \hat{y}) = -\sum_{j=1}^{3} y_j \log \hat{y}_j, \text{ where } \hat{y}_j \text{ is prediction, } y_j \text{ is ground truth} \tag{23}$$

Backpropagation Please write down $\frac{\partial L}{\partial v_{jk}}$ and $\frac{\partial L}{\partial w_{ki}}$ in terms of only x_i, z_k, o_j, y_j, and/or $\hat{y}_j$ using backpropagation.**(40 points)**

Solution:

There are TWO methods to solve this question. We demonstrate any method will get the same result. If the student's answer follow anyone below, it should be correct.

The first solution for $\frac{\partial L}{\partial v_{jk}}$:

$$\frac{\partial L}{\partial v_{jk}} = \sum_{i=1}^{3} \frac{\partial L}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial o_j} \frac{\partial o_j}{\partial v_{jk}} \qquad \text{3 point}$$

$$\begin{aligned}\frac{\partial L}{\partial \hat{y}_i} &= \frac{\partial}{\partial \hat{y}_i} \left[- \sum_{j=1}^{3} y_i \log \hat{y}_i \right] \\ &= -\frac{y_i}{\hat{y}_i} \qquad \text{3 points}\end{aligned}$$

when $i = j$

$$\begin{aligned}\frac{\partial \hat{y}_i}{\partial o_j} &= \frac{\partial}{\partial o_j} \left[\frac{\exp(o_i)}{\sum_{m=1}^{3} \exp(o_m)} \right] \\ &= \frac{g'(x)h(x) - h'(x)g(x)}{[h(x)]^2}, \text{where } g(x) = \exp(o_j), h(x) = \sum_{i=1}^{3} \exp(o_i) \\ &= \frac{\exp(o_j) \sum_{m=1}^{3} \exp(o_m) - \exp(o_j)\exp(o_j)}{[\sum_{m=1}^{3} \exp(o_m)]^2} \\ &= \hat{y}_j(1 - \hat{y}_j) \qquad \text{3 points}\end{aligned}$$

when $i \neq j$

$$\begin{aligned}\frac{\partial \hat{y}_i}{\partial o_j} &= -\frac{\exp(o_i)\exp(o_j)}{[\sum_{m=1}^{3} \exp(o_m)]^2} \\ &= -\hat{y}_j \hat{y}_i \qquad \text{3 points}\end{aligned}$$

$$\begin{aligned}\frac{\partial o_j}{\partial v_{jk}} &= \frac{\partial \sum_{k=1}^{4} v_{jk} z_k}{\partial v_{jk}} = z_k \qquad \text{3 points} \\ \rightarrow \frac{\partial L}{\partial v_{jk}} &= \frac{y_j}{\hat{y}_j} \hat{y}_j (y_j - 1) z_k + \sum_{i=1, i \neq j}^{3} \frac{y_i}{\hat{y}_i} \hat{y}_j \hat{y}_i z_k \\ &= y_j(\hat{y}_j - 1) z_k + \sum_{i=1, i \neq j}^{3} y_i \hat{y}_j z_k \qquad \text{3 points}\end{aligned}$$

For multi-classification, the ground truth vector is an one hot vector. Thus, $\sum_{i=1}^{3} y_i = 1$. We then simplify the above result as:
(To be fair to those students know this simplification, we credit 3 points for this step.)

$$\begin{aligned}\rightarrow \frac{\partial L}{\partial v_{jk}} &= y_j(\hat{y}_j - 1) z_k + \sum_{i=1, i \neq j}^{3} y_i \hat{y}_j z_k \\ &= y_j(\hat{y}_j - 1) z_k + (1 - y_j) \hat{y}_j z_k \\ &= (\hat{y}_j - y_j) z_k \qquad \text{3 points}\end{aligned}$$

The second solution for $\frac{\partial L}{\partial v_{jk}}$:

(We found some students follow this solution. However, in the lecture, Prof. Victor didn't teach one hot vector in multi-classification problem. Other students may not know this assumption to simplify the calculation. Thus, we provide this solution for students to better understand the back propagation in the multi-classification problem.)

For multi-classification, the ground truth vector is an one hot vector. Thus, $\sum_{i=1}^{3} y_i = 1$. We assume the ground truth $y_j = 1$ (**3 points**).

$$\begin{aligned}\frac{\partial L}{\partial v_{jk}} &= \sum_{i=1}^{3} \frac{\partial L}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial o_j} \frac{\partial o_j}{\partial v_{jk}} \\ &= \frac{\partial L}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial o_j} \frac{\partial o_j}{\partial v_{jk}} \qquad \text{3 point}\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial \hat{y}_j} &= \frac{\partial}{\partial \hat{y}_j}\left[-\sum_{j=1}^{3} y_i \log \hat{y}_j\right] \\ &= -\frac{y_j}{\hat{y}_j} \qquad \text{3 points}\end{aligned}$$

$$\begin{aligned}\frac{\partial \hat{y}_i}{\partial o_j} &= \frac{\partial}{\partial o_j}\left[\frac{\exp(o_i)}{\sum_{m=1}^{3}\exp(o_m)}\right] \\ &= \frac{g'(x)h(x) - h'(x)g(x)}{[h(x)]^2}, \text{where } g(x) = \exp(o_j), h(x) = \sum_{i=1}^{3}\exp(o_i) \\ &= \frac{\exp(o_j)\sum_{m=1}^{3}\exp(o_m) - \exp(o_j)\exp(o_j)}{[\sum_{m=1}^{3}\exp(o_m)]^2} \\ &= \hat{y}_j(1-\hat{y}_j) \qquad \text{3 points}\end{aligned}$$

$$\frac{\partial o_j}{\partial v_{jk}} = \frac{\partial \sum_{k=1}^{4} v_{jk} z_k}{\partial v_{jk}} = z_k \qquad \text{3 points}$$

$$\begin{aligned}\rightarrow \frac{\partial L}{\partial v_{jk}} &= \frac{y_j}{\hat{y}_j}\hat{y}_j(1-\hat{y}_j) \\ &= y_j(\hat{y}_j - y_j)z_k \\ &= (\hat{y}_j - y_j)z_k \qquad \text{6 points}\end{aligned}$$

Regarding the grading, we strictly follow the above rubric. For those students who still have more confusion in this question, please talk to TA (Chaoyang He) during the OH.

The solution for $\frac{\partial L}{\partial w_{ki}}$:

$$\frac{\partial L}{\partial w_{ki}} = \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial w_{ki}} \qquad \text{3 point}$$

$$\frac{\partial L}{\partial z_k} = \sum_{m=1}^{3}\frac{\partial L}{\partial o_m}\frac{\partial o_m}{\partial z_k}$$

$$= \sum_{m=1}^{3}\left(\sum_{j=1}^{3}\frac{\partial L}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial o_m}\right)\frac{\partial o_m}{\partial z_k} \qquad \text{3 point}$$

Some students wrote another form as follows. It's also correct.

$$\frac{\partial L}{\partial z_k} = \sum_{j=1}^{3}\frac{\partial L}{\partial \hat{y}_j}\sum_{m=1}^{3}\frac{\partial \hat{y}_j}{\partial o_m}\frac{\partial o_m}{\partial z_k}$$

For the computing inside the above parentheses, we can reuse the result from the previous step, which follows the idea of back propagation. Therefore, we now have:

$$\frac{\partial L}{\partial z_k} = \sum_{m=1}^{3}(\hat{y}_m - y_m)v_{mk} \qquad \text{6 point}$$

$$\frac{\partial z_k}{\partial w_{ki}} = \frac{\partial}{\partial w_{ki}} tanh\left(\sum_{i=1}^{4} w_{ki}x_i\right)$$

$$= (1 - {z_k}^2)x_i \qquad \text{3 points}$$

$$\rightarrow \frac{\partial L}{\partial w_{ki}} = \sum_{m=1}^{3}(\hat{y}_m - y_m)v_{mk}(1 - {z_k}^2)x_i \qquad \text{4 point}$$