CSCI 567, Spring 2019
Prof. Victor Adamchik

Theory Assignment 2

Due: 11:59 pm, Feb 15th, 2019

Instructions

Submission: Assignment submission will be via `courses.uscden.net`. By the submission date, there will be a folder named 'Theory Assignment 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 integrated LaTeX editors that are convenient to use (e.g Overleaf, ShareLaTeX). Choose the one(s) you like the most. This tutorial Getting to Grips with LaTeX is a good start if you do not know how to use LaTeX yet.

Please also follow the rules below:

- The file should be named as `Firstname_Lastname_USCID.pdf` e.g., `Don_Quijote_de_la_Mancha_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 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. Review the rules on academic conduct in the syllabus: a single instance of plagiarism can adversely affect you significantly more than you could stand to gain.

Notes 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.
- $\|.\|$ means L2-norm unless specified otherwise *i.e.* $\|.\| = \|.\|_2$

Problem 1 Logistic Regression (3 points)

Review Recall that the logistic regression model is defined as:

$$p(y=1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)$$
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

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

$$\min_{\boldsymbol{w},b} \ell(\boldsymbol{w},b) = \min_{\boldsymbol{w},b} -\sum_n \{y_n \log[p(y_n=1|\boldsymbol{x}_n)] + (1-y_n)\log[p(y_n=0|\boldsymbol{x}_n)]\} \quad (1)$$
$$= \min_{\boldsymbol{w},b} -\sum_n \{y_n \log\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n+b) + (1-y_n)\log[1-\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n+b)]\} \quad (2)$$

Question Bias solution Consider if one does not have access to the feature $\boldsymbol{x}$ of the data, and is given a training set of $\mathcal{D} = \{y_n\}_{n=1}^N$, where $y_n \in \{0,1\}$. What would be the optimal logistic regression classifier in that case? What is the probability that a test sample is labeled as 1?
Hint: write out the objective function as in Eqn. 2, and solve for the optimal bias term b^*.

What to submit: 1) fewer-than-5-line derivation and the formula for the optimal bias b^*. 2) the probability that a test sample is labeled as 1.
Ans:

$$b^* = \min_b -\sum_n \{y_n \log\sigma(b) + (1-y_n)\log[1-\sigma(b)]\} \quad \text{cross entropy objective} \quad (3)$$
$$\sum_n y_n(1-\sigma(b^*)) - (1-y_n)\sigma(b^*) = 0 \quad \text{Taking derivatives w.r.t } b \text{ 1 points} \quad (4)$$
$$\sigma(b^*) = \frac{\sum_n y_n}{N} \quad \text{solve for } b^* \text{ 1 points} \quad (5)$$
$$b^* = \log\left(\frac{\sum_n y_n}{\sum_n (1-y_n)}\right) \quad \text{1 points} \quad (6)$$

$\sigma(b^*)$ is the optimal logistic regression classifier, also the probability that a test sample is labeled as 1, which is the fraction of label 1 samples in the training data.

Problem 2 Linear Classifiers (10 points)

In this problem, you are going to use linear classifiers to solve the famous XOR problem. In the XOR problem, there are two binary input features $x_1, x_2 \in \{0, 1\}$, and the label $y = 0$ if $x_1 = x_2$ and 1 otherwise. See Table 1 for an illustration.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: XOR problem

2.1 Logistic Regression Assume we use the logistic regression model

$$p(y = 1|x) = \sigma(b + w_1 x_1 + w_2 x_2), \tag{7}$$

and the loss function as in Eq. 2. The training set contains 4 data points, one for each row in Table 1.

Question I greedy algorithm We define a greedy training algorithm as follows: we start with no features, and optimize the loss ℓ w.r.t. b; then we fix b and add feature x_1 and optimize w_1; finally we fix both b, w_1, and optimize w_2. Please write out the final logistic regression classifier.
What to submit: the logistic regression classifier, i.e. b, w_1 *and* w_2.
Ans: $b = w_1 = w_2 = 0$ **(3 points)**

Question II error rate What is the best classification error rate on the training examples by the logistic regression model (Eq. 7)?
What to submit: best error rate.
Ans: 0.25 **(1 points)**

Question III feature design Suppose we can design another feature x_3 using x_1 and x_2, to include in the logistic regression model, *i.e.* $p(y = 1|x) = \sigma(b + w_1 x_1 + w_2 x_2 + w_3 x_3)$. Which of the following features, if any, can allow us to correctly classify all the training examples by logistic regression?

(a) $x_3 = x_1 - x_2$

(b) $x_3 = x_1 x_2$

(c) $x_3 = x_2^2$

(d) $x_3 = x_1^2 + x_2^2$

What to submit: select all that satisfies, or none if none of them satisfies.
Ans: b **(2 points)**

2.2 Perceptron Assume we use the perceptron algorithm (Alg. 1) to solve the XOR problem. We change the class label 0 to -1, and denote $\boldsymbol{x} = [1, x_1, x_2]$, $\boldsymbol{w} = [b, w_1, w_2]$. And $\text{sign}(x) = \begin{cases} 1, & \text{if } x > 0, \\ -1, & \text{otherwise.} \end{cases}$

Algorithm 1: Perceptron algorithm

```
Initialize: w = 0
while not converged do
    randomly pick (x, y), make prediction ŷ = sign(w^T x)
    if ŷ ≠ y then
        w ← w + yx
```

Question I weight Table 2 shows the number of times each point is misclassified during a run of the perceptron algorithm. Write down the final output of the algorithm, *i.e.* the weight vector $\boldsymbol{w}$.

(x_1, x_2)	y	Times misclassified
(0, 0)	-1	1
(0, 1)	1	3
(1, 0)	1	1
(1, 1)	-1	2

Table 2: Times misclassified in perceptron algorithm

What to submit: weight vector $\boldsymbol{w}$.
Ans: By the algorithm, the weight is a linear combination of data points weighted by the number of times they are misclassified and their label. That is

$$\boldsymbol{w} = -1 \times (1,0,0) + 3 \times (1,0,1) + 1 \times (1,1,0) - 2 \times (1,1,1) = (1,-1,1).$$

(3 points)

Question II convergence Does Alg. 1 converge on the XOR data?
What to submit: answer Yes *or* No.
Ans: No **(1 points)**

Problem 3 Neural network (12 points)

In the lecture, we have talked about error-backpropagation, a way to compute partial derivatives (or gradients) w.r.t the parameters of a neural network. We have also mentioned that optimization is challenging and nonlinearity is important for neural networks. In this problem, you are going to (Question I) practice error-backpropagation, (Question II) investigate how initialization affects optimization, (Question III) study the importance of nonlinearity, and (Question IV) design a neural network to solve XOR problem.

For Question I to III, you are given the following 1-hidden layer multi-layer perceptron (MLP) for a K-class classification problem (see Fig. 1 for illustration and details).

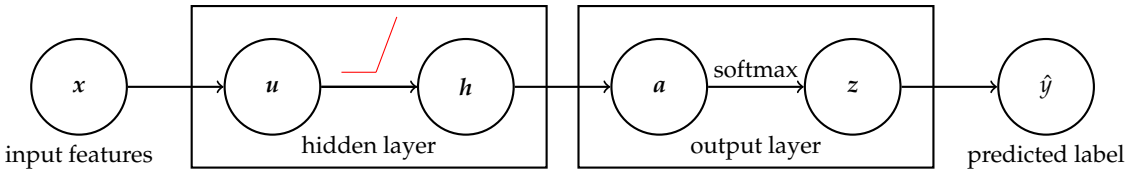

Figure 1: A diagram of a 1 hidden-layer multi-layer perceptron (MLP). *The edges mean mathematical operations, and the circles mean variables.* Generally we call the combination of a linear (or affine) operation and a nonlinear operation (like element-wise sigmoid or the rectified linear unit (ReLU) operation as in eq. (10)) as a hidden layer.

$(\boldsymbol{x}, y)$ is a labeled instance, where $\boldsymbol{x} \in \mathbb{R}^D$ and $y \in \{1, 2, \cdots, K\}$.

input features $\quad \boldsymbol{x} \in \mathbb{R}^D$ (8)

hidden layer $\quad \boldsymbol{u} = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}, \quad \boldsymbol{W}^{(1)} \in \mathbb{R}^{M \times D}$ and $\boldsymbol{b}^{(1)} \in \mathbb{R}^M$ (9)

$$\boldsymbol{h} = \max\{0, \boldsymbol{u}\} = \begin{bmatrix} \max\{0, u_1\} \\ \vdots \\ \max\{0, u_M\} \end{bmatrix} \tag{10}$$

output layer $\quad \boldsymbol{a} = \boldsymbol{W}^{(2)}\boldsymbol{h} + \boldsymbol{b}^{(2)}, \quad \boldsymbol{W}^{(2)} \in \mathbb{R}^{K \times M}$ and $\boldsymbol{b}^{(2)} \in \mathbb{R}^K$ (11)

softmax
$$\boldsymbol{z} = \begin{bmatrix} \frac{e^{a_1}}{\sum_k e^{a_k}} \\ \vdots \\ \frac{e^{a_K}}{\sum_k e^{a_k}} \end{bmatrix} \tag{12}$$

predicted label $\quad \hat{y} = \arg\max_k z_k.$ (13)

For K-class classification problem, a popular loss function for training is the cross-entropy loss. We denote the cross-entropy loss with respect to the training example (x, y) by l :

$$l = -\ln(z_y) = -\ln\left(\frac{e^{a_y}}{\sum_k e^{a_k}}\right) = \ln\left(1 + \sum_{k \neq y} e^{a_k - a_y}\right),$$

where z_y is the y-th coordinate of the softmax output $\boldsymbol{z}$. Note l is a function of the parameters of the network, that is, $\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{W}^{(2)}$, and $\boldsymbol{b}^{(2)}$. Before you proceed to the questions, you are encouraged to check the dimensionality of each intermediate results $\boldsymbol{u}, \boldsymbol{h}, \boldsymbol{a}, \boldsymbol{z}$, to make sure you understand Eq. 8-13.

Question I Error-backpropagation Assume that you have computed $\boldsymbol{u}$, $\boldsymbol{h}$, $\boldsymbol{a}$, $\boldsymbol{z}$, given $(\boldsymbol{x}, \boldsymbol{y})$. Follow the four steps below to find out the derivatives of l with respect to all the four parameters $\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{W}^{(2)}$ and $\boldsymbol{b}^{(2)}$. You are encouraged to use matrix/vector forms to simplify your answers. Note that we follow the convention that the derivative with respect to a variable is of the same dimension of that variable. For example, $\frac{\partial l}{\partial \boldsymbol{W}^{(1)}}$ is in $\mathbb{R}^{M \times D}$. (This is called the denominator layout.)

1. First express $\frac{\partial l}{\partial \boldsymbol{a}}$ in terms of $\boldsymbol{z}$ and y. You may find it convenient to use the notation $\boldsymbol{y} \in \mathbb{R}^K$ whose k-th coordinate is 1 if $k = y$ and 0 otherwise.
2. Then express $\frac{\partial l}{\partial \boldsymbol{W}^{(2)}}$ and $\frac{\partial l}{\partial \boldsymbol{b}^{(2)}}$ in terms of $\frac{\partial l}{\partial \boldsymbol{a}}$ and $\boldsymbol{h}$.
3. Next express $\frac{\partial l}{\partial \boldsymbol{u}}$ in terms of $\frac{\partial l}{\partial \boldsymbol{a}}$, $\boldsymbol{u}$, and $\boldsymbol{W}^{(2)}$. You will need to use the (sub)derivative of the ReLU function $\max\{0, u\}$ denoted by $H(u)$, which is

$$H(u) = \begin{cases} 1, \text{ if } u > 0, \\ 0, \text{ if } u \leq 0, \end{cases}$$

   Also, you may find it convenient to use the notation $\boldsymbol{H}(\boldsymbol{u}) \in \mathbb{R}^{M \times M}$ which stands for a diagonal matrix with $H(u_1), \ldots, H(u_M)$ on the diagonal.
4. Finally, express $\frac{\partial l}{\partial \boldsymbol{W}^{(1)}}$ and $\frac{\partial l}{\partial \boldsymbol{b}^{(1)}}$ in terms of $\frac{\partial l}{\partial \boldsymbol{u}}$ and $\boldsymbol{x}$.

What to submit: Write down the final answers to the 6 partial derivatives in blue.
Ans:

$$\frac{\partial l}{\partial \boldsymbol{a}} = \boldsymbol{z} - \boldsymbol{y} \qquad \text{1 points}$$

$$\frac{\partial l}{\partial \boldsymbol{W}^{(2)}} = \frac{\partial l}{\partial \boldsymbol{a}} \boldsymbol{h}^T \qquad \text{1 points}$$

$$\frac{\partial l}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial l}{\partial \boldsymbol{a}} \qquad \text{1 points}$$

$$\frac{\partial l}{\partial \boldsymbol{u}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{h}} \frac{\partial l}{\partial \boldsymbol{a}} = \boldsymbol{H}(\boldsymbol{u}) \boldsymbol{W}^{(2)T} \frac{\partial l}{\partial \boldsymbol{a}} \qquad \text{1 points}$$

$$\frac{\partial l}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial l}{\partial \boldsymbol{u}} \boldsymbol{x}^T \qquad \text{1 points}$$

$$\frac{\partial l}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial l}{\partial \boldsymbol{u}} \qquad \text{1 points}$$

Question II Initialization Suppose we initialize $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}, \boldsymbol{b}^{(1)}$ with zero matrices/vectors (i.e., matrices and vectors with all elements set to 0), please first verify that $\frac{\partial l}{\partial \boldsymbol{W}^{(1)}}, \frac{\partial l}{\partial \boldsymbol{W}^{(2)}}, \frac{\partial l}{\partial \boldsymbol{b}^{(1)}}$ are all zero matrices/vectors, irrespective of $\boldsymbol{x}, \boldsymbol{y}$ and the initialization of $\boldsymbol{b}^{(2)}$.

Now if we perform stochastic gradient descent for learning the neural network using a training set $\{(\boldsymbol{x}_i \in \mathbb{R}^D, \boldsymbol{y}_i \in \mathbb{R}^K)\}_{i=1}^N$, please explain in a concise statement (in one sentence) why no learning will happen on $\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$, $\boldsymbol{b}^{(1)}$ (i.e., they will not change no matter how many iterations are run). Note that

this will still be the case even with weight decay (L_2 regularization) and momentum if the initial velocity vectors/matrices are set to zero.

What to submit: No submission for the verification question. Your one-sentence statement explains why no learning will happen.
Ans: Since $\boldsymbol{W}^{(2)}$ is all zero, $\frac{\partial l}{\partial \boldsymbol{u}}$ is all zero. So $\frac{\partial l}{\partial \boldsymbol{W}^{(1)}}, \frac{\partial l}{\partial \boldsymbol{b}^{(1)}}$ are all zero. Since $\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}$ are all zero, $\boldsymbol{h}$ is all zero. So $\frac{\partial l}{\partial \boldsymbol{W}^{(2)}}$ is all zero. In each iteration, all gradients with respective to these three parameters are zero, so no updates will be made. **(2 points)**

Question III Nonlinearity As mentioned in the lecture, non-linearity is very important for neural networks. With non-linearity (e.g., eq. (10)), the neural network shown in Fig. 1 can bee seen as a nonlinear basis function $\boldsymbol{\phi}$ (i.e., $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{h}$) followed by a linear classifier f (i.e., $f(\boldsymbol{h}) = \hat{y}$).

Please show that, by removing the nonlinear operation in eq. (10) and setting eq. (11) to be $\boldsymbol{a} = \boldsymbol{W}^{(2)}\boldsymbol{u} + \boldsymbol{b}^{(2)}$, the resulting network is essentially a linear classifier. More specifically, you can now represent $\boldsymbol{a}$ as $\boldsymbol{U}\boldsymbol{x} + \boldsymbol{v}$, where $\boldsymbol{U} \in \mathbb{R}^{K\times D}$ and $\boldsymbol{v} \in \mathbb{R}^K$. Please write down the representation of $\boldsymbol{U}$ and $\boldsymbol{v}$ using $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}, \boldsymbol{b}^{(1)}$, and $\boldsymbol{b}^{(2)}$.
What to submit: the representation of $\boldsymbol{U}$ and $\boldsymbol{v}$.
Ans:

$$\boldsymbol{U} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}$$
$$\boldsymbol{v} = \boldsymbol{W}^{(2)}\boldsymbol{b}^{(1)} + \boldsymbol{b}^{(2)}$$

(2 points)

Question IV network design In this question, we design a 1-hidden layer neural network to solve the XOR problem given in Table 1 using two units with ReLU nonlinear activation. The network is illustrated in Fig. 2. We use the superscript $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}$ to distinguish the weights from the first layer (hidden layer), and the second layer (output layer).

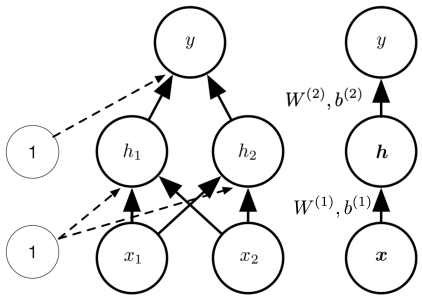

Figure 2: A small neural network. Expanded version (left) and compact version (right).

$$h_1 = \max\left\{0,\ W_{11}^{(1)}x_1 + W_{21}^{(1)}x_2 + b_1^{(1)}\right\},$$
$$h_2 = \max\left\{0,\ W_{12}^{(1)}x_1 + W_{22}^{(1)}x_2 + b_2^{(1)}\right\},$$
$$y = \text{sign}[W_1^{(2)}h_1 + W_2^{(2)}h_2 + b^{(2)}], \qquad \text{where sign}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{otherwise.} \end{cases}$$

We set $W_{11}^{(1)} = -1, W_{21}^{(1)} = -1, b_1^{(1)} = 1, W_{12}^{(1)} = 1, W_{22}^{(1)} = 1, b_2^{(1)} = 0$, and $W_2^{(2)} = -1$. Please write down a feasible solution for $W_1^{(2)}$ and $b^{(2)}$, which achieves 0 error rate on XOR problem.

What to submit: $W_1^{(2)}$ *and* $b^{(2)}$.

Ans: The answer is not unique. One possible answer $W_1^{(2)} = -2, b^{(2)} = 1.5$. **(2 points)**

For grading, we can write a script which inputs $W_1^{(2)}$ and $b^{(2)}$, and returns whether it's a feasible solution or not.