CSCI 567, Spring'18
Michael Shindler

Problem Set #2

Due: 11:59 pm, February 21, 2018

Instructions

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

You can submit multiple files, but only the last one submitted counts. That means if you finish some problems and want to submit something now, and then a later file when you finish, that's fine. If I were taking the class, that's what I'd do: that way, if I forget to finish the homework or something happens (remember Murphy's Law), I still get credit for what I finished and turned in. Remember, there are no grace days on problem sets, just on programming assignments!

Problem sets must be typewritten or neatly handwritten when submitted; if the grader cannot read your handwriting on a problem, they may elect to grade it as a zero. If it is handwritten, your submission must still be submitted as a PDF.

It is strongly recommended that you typeset with LaTeX and use that to generate a PDF file.

- The file should be named as `firstname_lastname_USCID.pdf` (e.g., Jenny_Tutone_8675309.pdf).
- Do not have any spaces in your file name when uploading it.
- Please include your name and USCID in the header of the report as well.

There are many free integrated LaTeX editors that are convenient to use. Choose the one(s) you like the most. This http://www.andy-roberts.net/writing/latex seems like a good tutorial.

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.
- The bias term is subsumed in the input vector, so the input vector is actually $x = [x', 1]^T$, unless mentioned otherwise.
- $\|.\|$ means L2-norm unless specified otherwise i.e. $\|.\| = \|.\|_2$

Problem 1 Neural networks (error-backpropagation, initialization, and nonlinearity)

[Recommended maximum time spent: 1 hour]

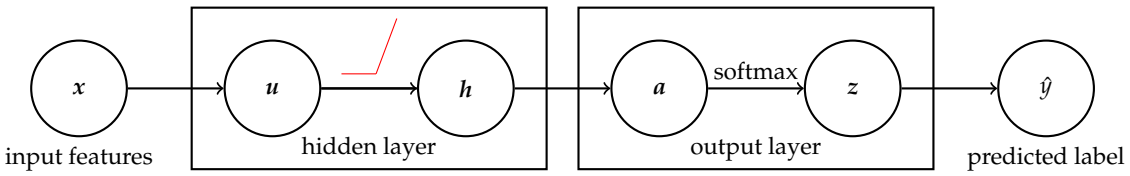

Figure 1: A diagram of a 1 hidden-layer multi-layer perceptron (MLP). *The edges mean mathematical operations, and the circles mean variables.* Generally we call the combination of a linear (or affine) operation and a nonlinear operation (like element-wise sigmoid or the rectified linear unit (relu) operation as in eq. (3)) as a hidden layer.

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 question, you are going to (Q1.1) practice error-backpropagation, (Q1.2) investigate how initialization affects optimization, and (Q1.3) the importance of nonlinearity.

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

$$\boldsymbol{x} \in \mathbb{R}^D \tag{1}$$

$$\boldsymbol{u} = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)} \quad , \boldsymbol{W}^{(1)} \in \mathbb{R}^{M \times D} \text{ and } \boldsymbol{b}^{(1)} \in \mathbb{R}^M \tag{2}$$

$$\boldsymbol{h} = \max\{0, \boldsymbol{u}\} = \begin{bmatrix} \max\{0, u_1\} \\ \vdots \\ \max\{0, u_M\} \end{bmatrix} \tag{3}$$

$$\boldsymbol{a} = \boldsymbol{W}^{(2)}\boldsymbol{h} + \boldsymbol{b}^{(2)} \quad , \boldsymbol{W}^{(2)} \in \mathbb{R}^{K \times M} \text{ and } \boldsymbol{b}^{(2)} \in \mathbb{R}^K \tag{4}$$

$$\boldsymbol{z} = \begin{bmatrix} \frac{e^{a_1}}{\sum_k e^{a_k}} \\ \vdots \\ \frac{e^{a_K}}{\sum_k e^{a_k}} \end{bmatrix} \tag{5}$$

$$\hat{y} = \arg\max_k z_k. \tag{6}$$

For K-class classification problem, one popular loss function for training is the cross-entropy loss,

$$l = -\sum_k \mathbf{1}[y == k] \log z_k, \tag{7}$$

$$\text{where } \mathbf{1}[\text{True}] = 1; \text{ otherwise, } 0. \tag{8}$$

For ease of notation, let us define the one-hot (i.e., 1-of-K) encoding

$$\boldsymbol{y} \in \mathbb{R}^K \text{ and } y_k = \begin{cases} 1, \text{ if } y = k, \\ 0, \text{ otherwise.} \end{cases} \tag{9}$$

so that

$$l = -\sum_k y_k \log z_k = -\boldsymbol{y}^T \begin{bmatrix} \log z_1 \\ \vdots \\ \log z_K \end{bmatrix} = -\boldsymbol{y}^T \log \boldsymbol{z}. \tag{10}$$

Q1.1 Assume that you have computed $\boldsymbol{u}$, $\boldsymbol{h}$, $\boldsymbol{a}$, $\boldsymbol{z}$, given $(\boldsymbol{x}, \boldsymbol{y})$. Please first express $\frac{\partial l}{\partial \boldsymbol{u}}$ in terms of $\frac{\partial l}{\partial \boldsymbol{a}}$, $\boldsymbol{u}$, $\boldsymbol{h}$, and $\boldsymbol{W}^{(2)}$.

$$\frac{\partial l}{\partial \boldsymbol{u}} = ?$$

Then express $\frac{\partial l}{\partial \boldsymbol{a}}$ in terms of $\boldsymbol{z}$ and $\boldsymbol{y}$.

$$\frac{\partial l}{\partial \boldsymbol{a}} = ?$$

Finally, compute $\frac{\partial l}{\partial \boldsymbol{W}^{(1)}}$ and $\frac{\partial l}{\partial \boldsymbol{b}^{(1)}}$ in terms of $\frac{\partial l}{\partial \boldsymbol{u}}$ and $\boldsymbol{x}$. Compute $\frac{\partial l}{\partial \boldsymbol{W}^{(2)}}$ in terms of $\frac{\partial l}{\partial \boldsymbol{a}}$ and $\boldsymbol{h}$.

$$\frac{\partial l}{\partial \boldsymbol{W}^{(1)}} = ?$$
$$\frac{\partial l}{\partial \boldsymbol{b}^{(1)}} = ?$$
$$\frac{\partial l}{\partial \boldsymbol{W}^{(2)}} = ?$$

You are encouraged to use matrix/vector forms to simplify your answers. Note that $\max\{0, u\}$ is not differentiable w.r.t. u at $u = 0$. Please note that

$$\frac{\partial \max\{0, u\}}{\partial u} = \begin{cases} 1, \text{ if } u > 0, \\ 0, \text{ if } u \leq 0, \end{cases} \tag{11}$$

which stands for the Heaviside step function. You can use

$$\frac{\partial \max\{0, \boldsymbol{u}\}}{\partial \boldsymbol{u}} = H(\boldsymbol{u}) \tag{12}$$

in your derivation of $\frac{\partial l}{\partial \boldsymbol{u}}$.

You can also use $\cdot*$ to represent element-wise product between two vectors or matrices. For example,

$$\boldsymbol{v} \cdot * \boldsymbol{c} = \begin{bmatrix} v_1 \times c_1 \\ \vdots \\ v_I \times c_I \end{bmatrix} \in \mathbb{R}^I, \text{ where } \boldsymbol{v} \in \mathbb{R}^I \text{ and } \boldsymbol{c} \in \mathbb{R}^I. \tag{13}$$

What to submit: No more than 5 lines of derivation for each of the 5 partial derivatives.

(In the nominator layout)

By the chain rule,

$$\frac{\partial l}{\partial \boldsymbol{h}} = \frac{\partial l}{\partial \boldsymbol{a}} \boldsymbol{W}^{(2)},$$
$$\frac{\partial l}{\partial \boldsymbol{u}} = \frac{\partial l}{\partial \boldsymbol{h}} H(\boldsymbol{u}),$$
$$\frac{\partial l}{\partial \boldsymbol{u}} = \frac{\partial l}{\partial \boldsymbol{a}} \boldsymbol{W}^{(2)} H(\boldsymbol{u}).$$

$$\frac{\partial l}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial l}{\partial \boldsymbol{u}},$$
$$\frac{\partial l}{\partial \boldsymbol{W}^{(1)}} = \boldsymbol{x} \frac{\partial l}{\partial \boldsymbol{u}},$$
$$\frac{\partial l}{\partial \boldsymbol{W}^{(2)}} = \boldsymbol{h} \frac{\partial l}{\partial \boldsymbol{a}}.$$

By the chain rule, $\frac{\partial l}{\partial \boldsymbol{a}_i} = \frac{\partial l}{\partial \boldsymbol{z}_j} \cdot \frac{\partial \boldsymbol{z}_j}{\partial \boldsymbol{a}_i}$. One can easily verify that

$$\frac{\partial l}{\partial \boldsymbol{z}_j} = -\delta_{yj} \frac{1}{\boldsymbol{z}_j},$$

$$\frac{\partial \boldsymbol{z}_j}{\partial \boldsymbol{a}_i} = \boldsymbol{z}_j (\delta_{ij} - \boldsymbol{z}_i),$$

where $\delta_{ij} = \mathbf{1}[i == j]$. Combine these two equations and remove the δ_{ij}, δ_{jy} (by setting $j = y$),

$$\frac{\partial l}{\partial \boldsymbol{a}_i} = -\delta_{yj} \frac{1}{\boldsymbol{z}_j} \boldsymbol{z}_j (\delta_{ij} - \boldsymbol{z}_i) = -(\delta_{iy} - \boldsymbol{z}_i),$$

which implies that

$$\frac{\partial l}{\partial \boldsymbol{a}} = -(\boldsymbol{y} - \boldsymbol{z})^T.$$

(In the denominator layout)

$$\frac{\partial l}{\partial \boldsymbol{u}} = H(\boldsymbol{u}) \boldsymbol{W}^{(2)T} \frac{\partial l}{\partial \boldsymbol{a}}.$$

$$\frac{\partial l}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial l}{\partial \boldsymbol{u}},$$
$$\frac{\partial l}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial l}{\partial \boldsymbol{u}} \boldsymbol{x}^T,$$
$$\frac{\partial l}{\partial \boldsymbol{W}^{(2)}} = \frac{\partial l}{\partial \boldsymbol{a}} \boldsymbol{h}^T.$$

$$\frac{\partial l}{\partial \boldsymbol{a}} = -(\boldsymbol{y} - \boldsymbol{z}).$$

Q1.2 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 with a concise mathematical 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 and momentum if the initial velocity vectors/-matrices are set to zero.

What to submit: No submission for the verification question. Your concise mathematical statement in one sentence for the explanation question.

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 are zero, so no updates will be made.

Q1.3 As mentioned in the lecture, nonlinearity is very important for neural networks. With nonlinearity (e.g., eq. (3)), 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. (3) and setting eq. (4) 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 $\boldsymbol{b}^{(2)}$

$$\boldsymbol{U} = ?$$
$$\boldsymbol{v} = ?$$

What to submit: No more than 2 lines of derivation for each of the question mark.

By combining the equations, we can get:

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

Problem 2 Logistic Regression

[Recommended maximum time spent: 45 minutes]
In the lectures we talked about logistic regression. Recall, that the decision boundary for logistic regression is found by maximizing log-likelihood:

$$log(P(D)) = \sum_{n=1}^{N} y_n log(\sigma(\mathbf{w}^{\mathbf{T}} x_n)) + (1 - y_n) log(1 - \sigma(\mathbf{w}^{\mathbf{T}} x_n)) \tag{14}$$

Where x_n are the observations, $y_n \in \{0, 1\}$ are their true labels. There is no a closed-form solution for logistic regression maximization, so we use iterative methods.

Q2.1 Does logistic regression always yield a unique solution? If yes, explain why, if no, bring an example and explain why the solution will not be unique.
What to submit: Your answer, your less-than-5-lines explanation and an example, if applicable.

No. If the features are not linear-independent, for example $x_{i1} = 2x_{i2}$, for all training samples x_i, then if $\beta_1 = \alpha_1, \beta_2 = \alpha_2$ is a valid solution, $\beta_1' = \alpha_1 + \frac{1}{2}\alpha_2, \beta_2' = 0$ is a valid solution too.

Q2.2 Consider a binary classification problem, where each observation x_n belongs to one of two classes: $\{0, 1\}$. However, suppose that the true label y_n of the observation x_n is not known, instead, we are given a value π_n that represents the probability that true label $y_n = 1$, i.e. $p(y_n = 1|w) = \pi_n$. Show that for such data D the log-likehood function of logistic regression model with parameters w will be given by:

$$log(P(D)) = \sum_{n=1}^{N} \pi_n log(\sigma(w^T x_n)) + (1 - \pi_n) log(1 - \sigma(w^T x_n)) \tag{15}$$

where σ is the sigmoid function:

$$\sigma(a) = \frac{1}{1 + e^{-a}} \tag{16}$$

What to submit: Your less-than-5-lines proof of Equation 15

By definition, we know

$$P(y_n = 1|\boldsymbol{x}_n) = \pi_n,$$
$$P(\hat{y}_n = 1|\boldsymbol{x}_n) = \sigma(\boldsymbol{w}^T \boldsymbol{x}_n).$$

So,

$$P(D|\boldsymbol{w}) = \prod_{n=1}^{N} P(\hat{y}_n = 1|\boldsymbol{x}_n, \boldsymbol{w})^{y_n} P(\hat{y}_n = 0|\boldsymbol{x}_n, \boldsymbol{w})^{1-y_n}.$$

$$\log P(D) = \sum_{n=1}^{N} y_n \log P(\hat{y}_n = 1|\boldsymbol{x}_n, \boldsymbol{w}) + (1 - y_n) \log P(\hat{y}_n = 0|\boldsymbol{x}_n, \boldsymbol{w})$$

Consider each data point $(\boldsymbol{x}_n, y_n)$ as a random variable, by taking the expectation over the log likelihood function, we can get

$$\begin{aligned}
\mathrm{E}[\log P(D)] &= \sum_{n=1}^{N} \mathrm{E}[y_n] \log P(\hat{y}_n = 1|\boldsymbol{x}_n, \boldsymbol{w}) + \mathrm{E}[(1 - y_n)] \log P(\hat{y}_n = 0|\boldsymbol{x}_n, \boldsymbol{w}) \\
&= \sum_{n=1}^{N} \pi_n \log P(\hat{y}_n = 1|\boldsymbol{x}_n, \boldsymbol{w}) + (1 - \pi_n) \log P(\hat{y}_n = 0|\boldsymbol{x}_n, \boldsymbol{w}) \\
&= \sum_{n=1}^{N} \pi_n \log \sigma(\boldsymbol{w}^T \boldsymbol{x}_n) + (1 - \pi_n) \log(1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_n)).
\end{aligned}$$