CSCI 567, Summer'20
Victor Adamchik

Problem Set #2

Due: July 15, 2020 11:59 pm

Instructions

Submission: Assignment submission will be via `courses.uscden.net`. By the submission date, there will be a folder named `Written 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 online LaTeX editors that are convenient to use (e.g Overleaf). You can also use offline editor such as TeXShop.

Please follow the rules below while submitting:

- 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 Logistic regression (25 points)

Recall that the logistic regression model is defined as:

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

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

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

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

1.1 Suppose we have three training samples $(x_1^1, x_1^2, y_1) = (0,1,1), (x_2^1, x_2^2, y_2) = (2,2,0), (x_3^1, x_3^2, y_3) = (1,0,1)$. Suppose our logistic regression model is $p(y=1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}+b)$ with $b=0$. We initialize this model with $\mathbf{w} = [0,0]^T$ and use learning rate = 0.1. When using GD to optimize this model, after one batch iteration, what's the training accuracy? (20 points)

1.2 Based on the model we got in previous question, if we have a test dataset consisting of samples: $(x_1^1, x_1^2, y_1) = (2,1,0), (x_2^1, x_2^2, y_2) = (-1,-1,1), (x_3^1, x_3^2, y_3) = (0,2,1)$, what is the testing accuracy? (5 points)

Problem 2 Equivalence of least absolute value and maximum likelihood (25 points)

In the lecture, we saw that regressions can be used with different losses for different purposes. We have looked at how minimizing least squared error $LSE = \sum_{i=1}^{n}(y_i - \mathbf{w}^{*T}\mathbf{x}_i)^2$ for linear regression leads to maximizing likelihood with Gaussian prior. In this exercise we will consider a different loss, least absolute value or least absolute error, $LAV = \sum_{i=1}^{n}|y_i - \mathbf{w}^{*T}\mathbf{x}_i|$ and show that minimizing this loss gives us w^* with maximum likelihood with Laplace distribution. Recall, that probability density function for Laplace distribution is given via:

$$P(x) = \frac{1}{2b}\exp\big(-\frac{|x-\mu|}{b}\big)$$

For that, let's assume, that the target variables $\mathbf{y}$ and input variables $\mathbf{X}$ are related through the following equation

$$y_i = \mathbf{w}^{*T}\mathbf{x}_i + \epsilon_i \tag{1}$$

where ϵ_i is the error term created due to the fact that the model does not fully represent the relationship between input and target. Let us assume that these errors ϵ_i are i.i.d. (independently and identically distributed) according to a Laplace distribution with 0 location parameter, and some scale parameter b.

2.1 Write down what is the distribution of y_i? In other words, you are asked to write down $P(y_i|\mathbf{x}_i)_{w^*}$ in terms of $y_i, \mathbf{x}_i$ and b. Keep in mind, that w^* is constant for this purpose. (10 points)

2.2 Note, that in part 2.1 we did not condition on $\mathbf{w}^*$, because it is the solution that minimizes the least absolute error, thus it is constant given $\mathbf{X}$ and $\mathbf{y}$. Instead, we used it to parametrize the conditional probability of $\mathbf{y}$. In other words, we can rewrite $P(y_i|\mathbf{x}_i)_{\mathbf{w}^*}$ as a function of $(\mathbf{y}, \mathbf{X}, \mathbf{w}^*)$; since $\mathbf{X}$ and $\mathbf{y}$ are fixed for a dataset, we may also parametrize our function with just $\mathbf{w}$. We would like to express the probability of observing the results $\mathbf{y}$ of our dataset: $P(y_i|\mathbf{x}_i)_{\mathbf{w}}$, which we will call the **likelihood function**: $L(\mathbf{w})$.

Show that $L(\mathbf{w}) = \prod_{i=1}^{n}\frac{1}{2b}\exp\big(-\frac{|y_i - \mathbf{w}^{*T}\mathbf{x}_i|}{b}\big)$ (5 points)

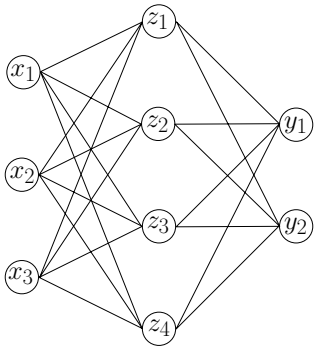
2.3 Instead of choosing a $\mathbf{w}$ that minimizes the least absolute value, we want to choose such a $\mathbf{w}$ that maximizes the likelihood of seeing our target variables $\mathbf{y}$ given the input variables $\mathbf{X}$. This is the principle of **maximum likelihood**.

Show that maximizing $L(\mathbf{w})$ is equivalent to minimizing least absolute value. This can be done via showing that two objectives are equivalent for the same optimization parameters.

Hint: Since maximizing a function is equivalent to maximizing any strictly increasing function of that function, instead of directly working with $L(\boldsymbol{w})$, try looking at $\log L(\boldsymbol{w})$. (10 points)

Problem 3 Neural Networks (25 points)

Consider the following neural network with one hidden layer.



Each neuron in the hidden layer is defined as $z_k = h(\sum_{i=1}^{3} w_{ki} x_i)$ for $k = 1, \ldots, 4$, where $h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and the outputs are defined as $y_j = \sum_{k=1}^{4} v_{jk} z_k$ for $j = 1, 2$. Suppose we choose the squared loss function for every pair, i.e. $L(\mathbf{y}, \widehat{\mathbf{y}}) = \frac{1}{2}\left((y_1 - \widehat{y}_1)^2 + (y_2 - \widehat{y}_2)^2\right)$, where y_j and $\widehat{y}_j$ represent the true outputs and our estimations, respectively.

3.1 Derive the expression of the derivative of the $h(x)$ function in terms of itself, i.e. what is $\frac{d\, h(x)}{dx} = f(h(x))$. (10 points)

3.2 Using the results from the previous question, derive the backpropagation updates (i.e. $\frac{\partial L}{\partial w_{ki}}$, $\frac{\partial L}{\partial v_{jk}}$) for estimation of w_{ki} and v_{jk}. In order to do that, first, express δ_k - the intermediate derivatives at the hidden layer in terms of $z_k, v_{jk}, \widehat{y}_j, y_j$. (5 points)

3.3 Express $\frac{\partial L}{\partial v_{jk}}$ in terms of $z_k, \widehat{y}_j, y_j$ (5 points)

3.4 Express $\frac{\partial L}{\partial w_{ki}}$ in terms of δ_k (5 points)

Problem 4 Convolutional Neural Networks (10 points)

Consider the following CNN. An $8 \times 8 \times 3$ image input, followed by a convolution layer with 2 filters of size 2×2 (stride 1, no zero padding), then another convolution layer with 4 filters of size 3×3 (stride 2, no zero padding), and finally a max pooling layer with a 2×2 filter (stride 1, no zero padding).

i. How many parameters are there in this network? Give two answer, with a bias neuron and without. (5 points)

ii. What is the final dimension of the output of this network? (5 points)

Problem 5 Valid Kernel (15 points)

$k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a valid kernel if and only if its Gram matrix, also known as Kernel Matrix, is positive semi-definite (PSD). So, to **disprove** that k is a valid kernel it is sufficient to find a set of $\mathbf{x}$ that breaks the condition of positive semi-definiteness of the Gram matrix. However, to prove that k **is** a valid kernel, it is more convenient to show that one of the following properties is true:

- it can be expressed as dot product in some transformed feature space i.e. $k(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2)$ and $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$ is some transformation, or
- it is a linear combination of some kernels with positive coefficients k_1, k_2 i.e. $k(\mathbf{x}_1, \mathbf{x}_2) = ak_1(\mathbf{x}_1, \mathbf{x}_2) + bk_2(\mathbf{x}_1, \mathbf{x}_2)$, $a, b > 0$, or
- it is a product of two kernels k_1, k_2 i.e. $k(\mathbf{x}_1, \mathbf{x}_2) = k_1(\mathbf{x}_1, \mathbf{x}_2)k_2(\mathbf{x}_1, \mathbf{x}_2)$

5.1 Prove or disprove that $k(\mathbf{x}_1, \mathbf{x}_2) = (f(\mathbf{x}_1) + f(\mathbf{x}_2))^2$ is a valid kernel. (10 points)

5.2 Show that if $k(\mathbf{x}_1, \mathbf{x}_2)$ is some valid kernel, then $f(k(\mathbf{x}_1, \mathbf{x}_2)) = \sum_{i=0}^{p} c_i k^i(\mathbf{x}_1, \mathbf{x}_2)$; $c_i \geq 0$, i.e. f is some polynomial of degree p with positive coefficients is also a kernel. (5 points)