CSCI 567, Spring 2023
Yan Liu

Homework 2

Date: Feb 19, 2023
Due: 11:59 PM, Mar 1, 2023 (AoE)
Total scores: 80 points

Problem 1 Generalized Linear Models (26 points)

We have already introduced linear regression, logistic regression, and multinomial logistic regression. Now we discuss a broader family of models - Generalized Linear Models (GLMs).

1.1 We begin with defining a special family of distributions - the exponential family distributions. If a distribution can be written in the form

$$p(y;\boldsymbol{\eta}) = b(y)\exp(\boldsymbol{\eta}^T\boldsymbol{t}(y) - a(\boldsymbol{\eta})), \tag{1}$$

it then belongs to the exponential family. In this problem, we always have $y, b(y), a(\boldsymbol{\eta})$ as scalars. $\boldsymbol{\eta}, \boldsymbol{t}(y) \in \mathbb{R}^K$ are K-dim vectors, but the definition also applies to $K = 1$.)

We can easily find that the Bernoulli distribution $y \sim \text{Bernoulli}(q) \Rightarrow p(y) = q^y(1-q)^{1-y}$ is in in the exponential family:

$$\begin{aligned}
p(y) &= \exp(y\log q + (1-y)\log(1-q)) \\
&= 1 \cdot \exp(\log\frac{q}{1-q}\cdot y + \log(1-q)), \text{ where} \\
b(y) &= 1 \\
\eta &= \log\frac{q}{1-q} \\
t(y) &= y \\
a(\eta) &= -\log(1-q).
\end{aligned}$$

Show that the categorical distribution is also in the exponential family, and write down its $b(y), \boldsymbol{\eta}, \boldsymbol{t}(y), a(\boldsymbol{\eta})$. **(8 points)**

(Hint: You may consider using the following form of categorical distribution:

$$p(y;\boldsymbol{q}) = (Cq_1)^{1\{y=1\}}(Cq_2)^{1\{y=2\}}\dots(Cq_K)^{1\{y=K\}},$$

where q_k is a non-negative scalar. $C = 1/\sum_{k=1}^K q_k$ for normalization so that $Cq_1, \dots, Cq_K$ are probabilities. $1\{y=k\} = 1$ if $y = k$, otherwise $1\{y=k\} = 0$. Notice that $\boldsymbol{\eta}$ can be some expression of $\boldsymbol{q}$.)

1.2 Now we give the steps to construct a GLM:

(1) Given the input feature $\boldsymbol{x} \in \mathbb{R}^D$, find a proper distribution belonging to the exponential family as the distribution of the label y conditioning on $\boldsymbol{x}$: $p(y|\boldsymbol{x}) \sim \text{ExponentialFamily}(\boldsymbol{\eta})$. $\boldsymbol{\eta} \in \mathbb{R}^K$.

(2) To make the model linear, we let $\boldsymbol{\eta} = \boldsymbol{W}\boldsymbol{x}$. $\boldsymbol{W} \in \mathbb{R}^{K\times D}$. (When $K = 1$ we usually write it as $\boldsymbol{w}^T\boldsymbol{x}$.)

(3) We select $\boldsymbol{h}(\boldsymbol{x};\boldsymbol{W}) = \mathbb{E}_{y\sim p(y|\boldsymbol{x};\boldsymbol{W})}\boldsymbol{t}(y)$ as our predicted value.

If we select the conditional distribution in Step (1) as the Bernoulli distribution, please finish the remaining steps to construct a GLM and show $\boldsymbol{h}(\boldsymbol{x};\boldsymbol{w})$. **(6 points)**

1.3 If we select the conditional distribution in Step (1) as the categorical distribution in the previous question, please finish the remaining steps to construct a GLM and show $h(\boldsymbol{x};\boldsymbol{W})$. **(6 points)**

1.4 Now let's construct a GLM to predict values that are most likely to follow the Poisson distribution (e.g. daily number of visitors in a store). Show that Poisson distribution is in the exponential family and finish the steps to construct a GLM by deriving $h(\boldsymbol{x}; \boldsymbol{w})$. (A slight difference in these steps is that η in this question is now a scalar.) **(6 points)**
(Hint: You may consider using the following form of Poisson distribution:

$$p(y; \lambda) = \frac{\lambda^y \exp(-\lambda)}{y!},$$

where $\lambda > 0$ and is a scalar. Notice that η can be some expression of λ.)

Problem 2 Neural Networks (24 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 to optimize using gradient descent. In this question, you are going to practice (Q2.1) error-backpropagation, (Q2.2) how initialization affects optimization, and (Q2.3) the importance of nonlinearity.

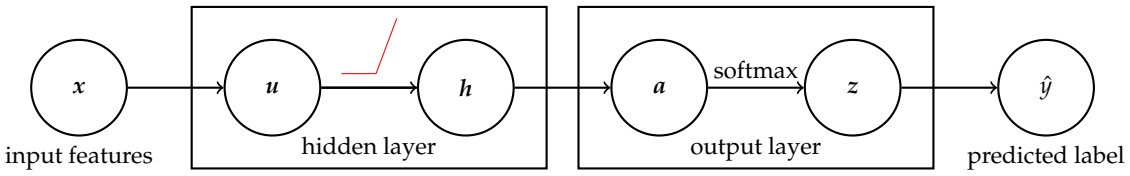

Figure 1: A diagram of a 1-hidden layer neural net. *The edges mean mathematical operations, and the circles mean variables.* Generally we call the combination of a linear (or affine) operation and a nonlinear operation (like element-wise sigmoid or the rectified linear unit (relu) operation as in eq. (4)) as a hidden layer. Note the two slight differences compared to the diagram used in the lecture : 1) one circle represents a vector and thus an array of neurons here and 2) the activation operations are also explicitly represented as edges here.

Specifically, you are given the following 1-hidden layer neural net 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,

$$\boldsymbol{x} \in \mathbb{R}^D \tag{2}$$

$$\boldsymbol{u} = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}, \quad \boldsymbol{W}^{(1)} \in \mathbb{R}^{M \times D} \text{ and } \boldsymbol{b}^{(1)} \in \mathbb{R}^M \tag{3}$$

$$\boldsymbol{h} = \max\{0, \boldsymbol{u}\} = \begin{bmatrix} \max\{0, u_1\} \\ \vdots \\ \max\{0, u_M\} \end{bmatrix} \tag{4}$$

$$\boldsymbol{a} = \boldsymbol{W}^{(2)}\boldsymbol{h} + \boldsymbol{b}^{(2)}, \quad \boldsymbol{W}^{(2)} \in \mathbb{R}^{K \times M} \text{ and } \boldsymbol{b}^{(2)} \in \mathbb{R}^K \tag{5}$$

$$\boldsymbol{z} = \begin{bmatrix} \frac{e^{a_1}}{\sum_k e^{a_k}} \\ \vdots \\ \frac{e^{a_K}}{\sum_k e^{a_k}} \end{bmatrix} \tag{6}$$

$$\hat{y} = \arg\max_k z_k. \tag{7}$$

For K-class classification problem, one popular loss function for training is the cross-entropy loss. Specifically we denote the cross-entropy loss with respect to the training example $(\boldsymbol{x}, y)$ by l:

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

Note that l is a function of the parameters of the network, that is, $\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{W}^{(2)}$ and $\boldsymbol{b}^{(2)}$.

2.1 Error Back-propagation 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. **(4 points)**
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 $\boldsymbol{h}$. **(4 points)**
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 $H(u)$ and is 1 if $u > 0$ and 0 otherwise. 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 points)**
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 $\boldsymbol{x}$. **(4 points)**

2.2 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, please explain with a concise statement why no learning will happen with the this initialization. **(4 points)**

2.3 Non-linearity As mentioned in the lecture, non-linearity is very important for neural networks. With non-linearity (e.g., eq. (4)), 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. (4) and setting eq. (5) 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 $\boldsymbol{b}^{(2)}$. **(4 points)**

Problem 3 Regularized Linear Regression With Kernels (15 points)

In class, we derive the closed-form solution of regularized linear regression with kernels. Now we discuss its gradient descent solution.
For the following regularized linear regression with feature mapping $\phi \in \mathbb{R}^D \to \mathbb{R}^M,\ M \gg D$

$$L(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - y_i\right\|_2^2 + \frac{1}{2}\lambda\left\|\boldsymbol{w}\right\|_2^2,\ \lambda > 0,$$

3.1 Write down $\boldsymbol{w}_{t+1}$ after one step gradient descent (using all examples) from $\boldsymbol{w}_t$ with learning rate $\alpha > 0$. **(3 points)**

3.2 What will be the problem if we directly conduct gradient descent? **(2 points)**

3.3 Denote as K the corresponding kernel of ϕ: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$.

(1) Prove that if we start from $\boldsymbol{w}_0 = \mathbf{0}$, for each t during gradient descent, we can always find scalars $\beta_i^{(t)},\ i = 1, \ldots, n$ such that $\boldsymbol{w}_t = \sum_{i=1}^{n}\beta_i^{(t)}\phi(\boldsymbol{x}_i)$. In other words, each $\boldsymbol{w}_t$ is a linear combination of $\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_n)$. (Hint: use induction from $t = 0$ to $1, 2, \ldots$) **(8 points)**

(2) Write down $\beta_1^{(t+1)}, \ldots, \beta_n^{(t+1)}$ after one step gradient descent from $\beta_1^{(t)}, \ldots, \beta_n^{(t)}$. Note that you should not have $\boldsymbol{w}$ in your final result. **(2 points)**

Problem 4 Direction of Linear Discriminant Hyperplane (15 points)

Consider linear discriminant analysis for a two-class classification problem on a dataset of N inputs $\{\mathbf{x}_1 \ldots \mathbf{x}_N\}$ and corresponding labels $\{y_1 \ldots y_N\}$, $y_i \in \{-1, 1\}\ \forall i \in \{1 \ldots N\}$. We say input $\mathbf{x}_i$ belongs to class $\mathcal{C}_1$ if its label y_i is 1 and it belongs to class $\mathcal{C}_{-1}$ if its label is -1. Mathematically, $\mathcal{C}_1 = \{(\mathbf{x}_i, y_i) : i \in [N], y_i = 1\}$ and $\mathcal{C}_{-1} = \{(\mathbf{x}_i, y_i) : i \in [N], y_i = -1\}$

We aim to find a separating hyperplane $\mathbf{w}$ such that if input $\mathbf{x}_i$ belongs to $\mathcal{C}_1$ then $\mathbf{w}^T\mathbf{x}_i \geq 0$ and if it belongs to $\mathcal{C}_{-1}$ then $\mathbf{w}^T\mathbf{x}_i \leq 0$. However, this might not be always possible. Instead, one way to relax the goal is to find a hyperplane $\mathbf{w}^*$ that maximizes $f(\mathbf{w}) = \sum_{i=1}^{N} y_i\mathbf{w}^T\mathbf{x}_i$ under the constraint $\|\mathbf{w}\| = 1$. Note that $f(\mathbf{w})$ can be arbitrarily maximized by increasing the magnitude of $\mathbf{w}$ and thus the constraint $\|\mathbf{w}\| = 1$ (or equivalently, $\|\mathbf{w}\|^2 = 1$) is important. We also assume that $\sum_{i=1}^{N} y_i\mathbf{x}_i \neq \mathbf{0}$ otherwise the objective $f(\mathbf{w})$ is always 0.

This can be written as a well-defined optimization problem using Lagrange multipliers (you do not have to know what this is to solve this problem). More concretely, there exists $\lambda \neq 0$ such that the hyperplane $\mathbf{w}^*$ we are looking for satisfies:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}\in\mathbb{R}^D} \sum_{i=1}^{N} y_i\mathbf{w}^T\mathbf{x}_i - \lambda(\mathbf{w}^T\mathbf{w} - 1) \tag{8}$$

4.1 Prove the following **(8 points)**

$$\mathbf{w}^* = \frac{1}{2\lambda}\left(\sum_{i:x_i\in\mathcal{C}_1}\mathbf{x}_i - \sum_{j:x_j\in\mathcal{C}_{-1}}\mathbf{x}_j\right).$$

4.2 Find the value of λ. **(4 points)**

4.3 In terms of minimizing the training error, can you think of one issue of our objective, i.e. maximizing $f(\mathbf{w})$? **(3 points)**