CSCI 567, Spring 2021
Sirisha Rambhatla

Homework #2

Released: Feb 6, 2021
Due: 11:59 pm, Feb 19, 2021
Total scores: 40 points

Instructions

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 Gradient Descent **(8 points)**

In the lecture, we discussed linear regression with regularization, where we optimize the regularized residual sum of squares objective as follows,

$$w^* = \arg\min_w RSS(w) + \lambda\|\mathbf{w}\|_2^2 = \arg\min_w \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

where $\mathbf{X} \in \mathbb{R}^{N\times(D+1)}$ is the feature matrix with N instances and $D+1$ dimensional features. $\mathbf{y} \in \mathbb{R}^N$ is the target. $\mathbf{w} \in \mathbb{R}^{(D+1)}$ are the model parameters and $\lambda > 0$ is the regularization coefficient.

We solved the optimization problem by deriving the closed form solution. Here you are asked to alternatively solve the problem using gradient descent.

1.1 Derive the update step and write the gradient descent algorithm for regularized linear regression. **(5 points)**

$$\nabla L(\mathbf{w}) = 2(\mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{y}) + 2\lambda\mathbf{w}$$

```
Initialize $\mathbf{w}^{(0)}$
For $t = 0, 1, 2 \cdots$ T
    $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta\nabla L(\mathbf{w}^{(t)})$
Return $\mathbf{w}^{(T)}$
```

1.2 Rewrite the algorithm with stochastic gradient descent. **(3 points)**

$$\nabla L(\mathbf{w}) = 2(\mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{y}) + 2\lambda\mathbf{w} = 2\lambda\mathbf{w} + 2\sum_{n=1}^{N}(\mathbf{x}_n\mathbf{x}_n^\top\mathbf{w} - y_n\mathbf{x}_n)$$

```
Initialize $\mathbf{w}^{(0)}$
For $t = 0, 1, 2 \cdots$ T
    $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta\tilde{\nabla} L(\mathbf{w}^{(t)})$
Return $\mathbf{w}^{(T)}$
```

where $\tilde{\nabla} L(\mathbf{w}) = 2N(\mathbf{x}_n\mathbf{x}_n^\top\mathbf{w} - y_n\mathbf{x}_n) + 2\lambda\mathbf{w}$ is the unbiased estimate of the gradient.

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 (Q1.1) error-backpropagation, (Q1.2) how initialization affects optimization, and (Q1.3) the importance of nonlinearity.

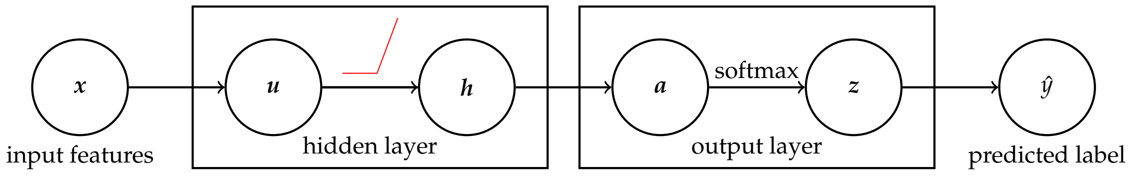

Figure 1: A diagram of a 1-hidden layer neural net. *The edges mean mathematical operations, and the circles mean variables.* Generally we call the combination of a linear (or affine) operation and a nonlinear operation (like element-wise sigmoid or the rectified linear unit (relu) operation as in eq.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

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

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

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

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

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

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

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

$$\frac{\partial l}{\partial \boldsymbol{a}} = \boldsymbol{z} - \boldsymbol{y} \qquad \text{(see derivation in Week 3 Lecture 7: Multi-class Classification (slide 20))}$$
$$\frac{\partial l}{\partial \boldsymbol{W}^{(2)}} = \frac{\partial l}{\partial \boldsymbol{a}} \boldsymbol{h}^T$$
$$\frac{\partial l}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial l}{\partial \boldsymbol{a}}$$
$$\frac{\partial l}{\partial \boldsymbol{u}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{h}} \frac{\partial l}{\partial \boldsymbol{a}} = \boldsymbol{H}(\boldsymbol{u}) \boldsymbol{W}^{(2)T} \frac{\partial l}{\partial \boldsymbol{a}}$$
$$\frac{\partial l}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial l}{\partial \boldsymbol{u}} \boldsymbol{x}^T$$
$$\frac{\partial l}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial l}{\partial \boldsymbol{u}}$$

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)**

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.

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

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 3 Direction of Linear Discriminant Hyperplane **(8 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:

$$\mathbf{w}^* = \underset{\mathbf{w}\in\mathbb{R}^D}{\arg\max} \sum_{i=1}^{N} y_i\mathbf{w}^T\mathbf{x}_i - \lambda(\mathbf{w}^T\mathbf{w} - 1) \tag{7}$$

3.1 Prove the following **(4 points)**

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

To find the maximum we set the gradient of $f(w) = \sum_{i=1}^{N} y_i\mathbf{w}^T\mathbf{x}_i + \lambda(\mathbf{w}^T\mathbf{w} - 1)$ to $\mathbf{0}$.

$$\nabla f(\mathbf{w}) = \sum_{i=1}^{N} y_i\mathbf{x}_i - 2\lambda\mathbf{w} = \mathbf{0}$$

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

3.2 Find the value of λ. **(2 points)**

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

3.3 In terms of minimizing the training error, can you think of one issue of our objective, i.e. maximizing $f(\mathbf{w})$? **(2 points)**

Maximizing this objective might lead to a solution that prefers having a large margin on some data points with the price of misclassifying others.