CSE 676: Assignment #1

Due data: 11:59 PM, March 15th, 2019

0.1 Softmax [2 points]

1) [1 point] Prove that softmax is invariant to constant sifts in the input, *i.e.*, for any input vector $\mathbf{x}$ and a constant scalar c, the following holds:

$$\text{softmax}(\mathbf{x}) = \text{softmax}(\mathbf{x} + c) ,$$

where $\text{softmax}(\mathbf{x})_i \triangleq \frac{e^{x_i}}{\sum_{i'} e^{x_{i'}}}$, and $\mathbf{x} + c$ means adding c to every dimension of $\mathbf{x}$.

2) [1 point] Let $\mathbf{z} = \mathbf{W}\,\mathbf{x} + \mathbf{c}$, where $\mathbf{W}$ and $\mathbf{c}$ are some matrix and vector, respectively. Let

$$J = \sum_i \log \text{softmax}(\mathbf{z})_i \ .$$

Calculate the derivatives of J w.r.t. $\mathbf{W}$ and $\mathbf{c}$, respectively, *i.e.*, calculate $\frac{\partial J}{\partial \mathbf{W}}$ and $\frac{\partial J}{\partial \mathbf{c}}$.

0.2 Logistic Regression with Regularization [2 points]

1) [1 point] Let the data be $(\mathbf{x}_i, y_i)_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. Logistic regression is a binary classification model, with the probability of y_i being 1 as:

$$p(y_i; \mathbf{x}_i, \boldsymbol{\theta}) = \sigma\left(\boldsymbol{\theta}^T \mathbf{x}_i\right) \triangleq \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}_i}} ,$$

where $\boldsymbol{\theta}$ is the model parameter. Assume we impose an L_2 regularization term on the parameter, defined as:

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{\lambda}{2} \boldsymbol{\theta}^T \boldsymbol{\theta}$$

with a positive constant λ. Write out the final objective function for this logistic regression with regularization model.

2) [1 point] If we use gradient descent to solve the model parameter. Derive the updating rule for $\boldsymbol{\theta}$. Your answer should contain the derivation, not just the final answer.

0.3 Derivative of the Softmax Function [3 points]

1) [1 point] Define the loss function as

$$J(\mathbf{z}) = -\sum_{k=1}^{K} y_k \log \tilde{y}_k ,$$

where $\tilde{y}_k = \frac{e^{z_k}}{\sum_{k'} e^{z_{k'}}}$, and $(y_1, \cdots, y_K)$ is a known probability vector. Derive the $\frac{\partial J(\mathbf{z})}{\partial \mathbf{z}}$. Note $\mathbf{z} = (z_1, \cdots, z_K)$ is a vector so $\frac{\partial J(\mathbf{z})}{\partial \mathbf{z}}$ is in the form of a vector. Your answer should contain the derivation, not just the final answer.

2 [1 point] Assume the above softmax is the output layer of an FNN. Briefly explain how the derivative is used in the backpropagation algorithm.

3) [1 points] Let $\mathbf{z} = \mathbf{W}^T \mathbf{h} + \mathbf{b}$, where $\mathbf{W}$ is a matrix, $\mathbf{b}$ and $\mathbf{h}$ are vectors. Use the chain rule to calculate the gradient of $\mathbf{W}$ and $\mathbf{b}$, *i.e.*, $\frac{\partial J}{\partial \mathbf{W}}$ and $\frac{\partial J}{\partial \mathbf{b}}$, respectively.

0.4 MNIST with FNN [3 points]

1) [3 points] Design an FNN for MNIST classification. Implement the model and plot two curves in one figure: *i*) training loss vs. training iterations; *ii*) test loss vs. training iterations.

   - You can use code from websites. However, you must reference (cite) the code in your answer.
   - Submission includes the plot of the two curves and the runnable code (with a ReadMe file containing instructions on how to run the code).