ECSE 4850/6850

Introduction to Deep Learning

Spring, 2020

Homework #3

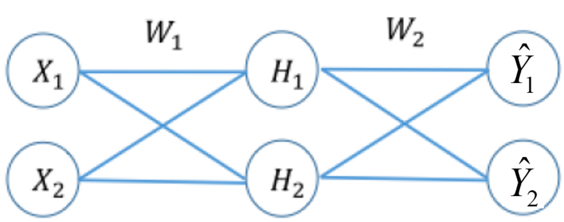
Due date: 2:00 pm, Feb. 27th

1. Problem 1 (20 points)

   A NN contains three layers: input layer ($\mathbf{X}^{N\times 1}$), a hidden layer ($\mathbf{H}^{N_1\times 1}$), and an output layer ($\hat{\mathbf{Y}}^{K\times 1}$). Let $\mathbf{W}^1$ and $\mathbf{W}_0^1$ respectively represent the weight matrix and weight bias vector for the hidden layer, and $\mathbf{W}^2$ and $\mathbf{W}_0^2$ be the weight matrix and weight bias vector for the output layer respectively. Assuming the ReLu activation function for the hidden layer and softmax function for the output layer, derive the expression for $\frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{W}^1}$ and $\frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{W}^1_{ij}}$, where $\mathbf{W}^1_{ij}$ represents ith row and jth column element of $\mathbf{W}^1$. Show the derivation process and intermediate results.

2. Problem 2 (30 points)

   The structure of a Neural Network is given below.



   The input value $\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and the desired output value is $\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. The initial weight matrix for the first layer W^1:

$$W^1 = \begin{bmatrix} W_1^1 & W_2^1 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 4 & 5 \end{bmatrix} \tag{1}$$

The bias weight matrix $W_0^1 = \begin{bmatrix} 1 \\ -6 \end{bmatrix}$. The initial weight matrix for the second layer W^2:

$$W^2 = \begin{bmatrix} W_1^2 & W_2^2 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 3 \end{bmatrix} \tag{2}$$

The bias weight matrix $W_0^2 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$.

Perform the following tasks:

- Forward propagation: calculate the value of the hidden nodes using sigmoid function as the activation function, and obtain the predicted value for output nodes $\hat{\boldsymbol{Y}}$ using softmax as the output function. Compute the gradient of the output $\nabla\hat{\mathbf{Y}}$ using cross-entropy loss function.
- Back propagation: Given $\nabla\hat{\mathbf{Y}}$, obtain the gradients for the weight matrix and bias vector for each layer.
- Update the weight matrix and bias vectors for each layer with their estimated gradients, using a learning rate of 0.5, and then compute the output value $\hat{\mathbf{Y}}$ using the updated weight matrices. Verify that the updated weight matrices have reduced the output loss function, as compared to those obtained using the initial weight matrices

For each task above, write out its equations and the estimated values.