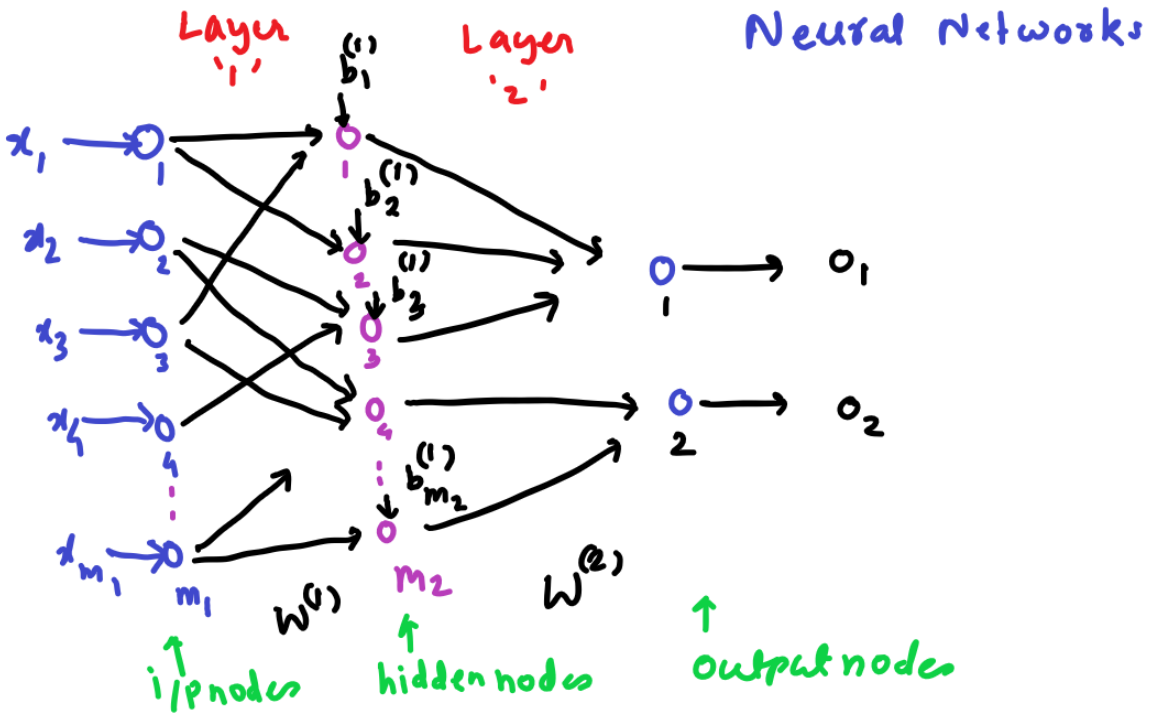
Programming Assignment 2: EE-5180

March 1, 2024

1 Instructions

1. Submission deadline: Monday (11/03/2024) 3.00 PM
2. You can discuss ideas, but you must write your own program.
3. **You are allowed to use Chat GPT. You must mention in your submission that you have used Chat GPT f if you have used it.**

2 Non-Linear Regression: Neural Networks



We consider non-linear model $y(x, \theta)$ is non-linear in both parameters θ and input x. The non-linear model can be described using neural networks.

Suppose that we are given data $\mathcal{D} = \{x_i, \tilde{y}_i\}_{i=1}^{N}$. There are N training examples. The loss function for a given training example is $L_i(\theta) = (y(x_i, \theta) - \tilde{y}_i)^2$.

The Mean square loss function for dataset $\mathcal{D}$ is

$$L(\theta) = \frac{1}{2N} \sum_{i=1}^{N} L_i(\theta)$$

Consider two layer neural network, there are m_1 input nodes, m_2 hidden nodes and m_3 output nodes. Let $W^{(1)}$ be the weight matrix between input nodes m_1 and hidden nodes m_2. Let $W^{(2)}$ be the weight matrix between hidden nodes m_2 and output nodes m_3. There can a bias at each hidden node and output node. Let $b^{(1)} = [b_1^{(1)}, b_2^{(1)}, \cdots, b_{m_2}^{(1)}]^T$ be biased at hidden nodes. Let $b^{(2)} = [b_1^{(2)}, b_2^{(2)}, \cdots, b_{m_3}^{(2)}]^T$ be biased at output nodes.

Note that hidden layer performs a non-linear operation on input data. This non-linear operation performed at the hidden unit is referred to as activation function. Use activation function as ReLU.

1. Write program in python to compute the output of feed-forward network. Assume $m_1 = 30$, $m_2 = 10$ and $m_3 = 1$. Initialize $W^{(1)}$, $W^{(2)}$, $b^{(1)}$ and $b^{(2)}$ suitably. For simplicity assume weights and bias are non-negative. Hence compute the output at the output node for suitable input vector.

2. Extend this code for output nodes $m_3 = 2$.

3. Write a back-propagation algorithm for output $m_3 = 1$. Write a a gradient computation for weights of layer 2, $w_{jk}^{(2)}$ and update rule for weight, Similar to layer 1 and update rule for $w_{ij}^{(1)}$. Similarly write a update rule for bias.

4. Combine feed-forward network and back-propagation algorithm, write algorithm similar to gradient descent.

5. In each iteration, step one :there is feed forward computation. Step two: backpropagation and update of weights and biases. Write a program for optimization of weights similar to gradient descent and compute the cost function as function of iteration.

Note: In the above programming you must not use inbuilt python library of neural network.

3 Support vector Machines (SVM)

Write a program of SVM for binary classification.

Note: You can use library functions from python on svm.

You are allowed to take help of ChatGPT for programming.

Define input data suitably, and plot cost function as function of iteration and show the output of classification with supporting hyperplane.