Neural Network Assignment 5 Report

Sreeraj Rimmalapudi

Algorithm 1 Neural Network Function Approximation

```
function MAIN
    Initialize a uniformly chosen random weight vector $\mathbf{w}$ of size 3N + 1 where N is the number of hidden neurons.
    Let $x_1..x_n$ be the training samples
    Let $d_1..d_n$ be the desired output
    for $epoch = 1$ to 100000 do
        for each training sample $x_i$ do
            Perform forward propagation and get output $y_i$
            Perform back propagation to get gradients with respect to $\mathbf{w}$
            Update the weights $\mathbf{w}$ using gradient descent
        Perform forward propagation with updated weights to get output $\mathbf{y}$
        $Meanloss = \frac{1}{n} \sum_{i=1}^{n} (d_i - y_i)^2$

function FORWARD PROPAGATION
    Let $x$ be the input sample
    Let $\mathbf{w1}$ be $w_1..w_N$, $\mathbf{b1}$ be $w_{N+1}..w_{2N}$
    Let $\mathbf{w2}$ be $w_{2N+1}..w_{3N}$, b2 be $w_{3N+1}$
    $\mathbf{hiddenout} = \tanh(\mathbf{w1} * x + \mathbf{b1})$; where $\mathbf{hiddenout}$ (output of the hidden neurons) is a vector of size N
    $y = \sum_{n=1}^{N} (\text{hiddenout}_i * w2_i) + b2$
    Store $\mathbf{w2}$,$x$, $\mathbf{hiddenout}$, $y$ for backprop

function BACK PROPAGATION
    Get $\mathbf{w2}$,$x$ $\mathbf{hiddenout}$, $y$ from forward prop
    Let d be the desired sample output
    $cost\_derivative = 2 * (y - d)$                ▷ derivative of the cost wrt to output $y$
    $delta2 = cost\_derivative * 1$
    $\mathbf{gradw2} = \mathbf{hiddenout} * delta2$                ▷ gradient of loss wrt $\mathbf{w2}$
    $gradb2 = delta2$                ▷ gradient of loss wrt b2
    $\mathbf{delta1} = \mathbf{w2} * \text{delta2} \odot (1 - \mathbf{hiddenout}^2)$
    $\mathbf{gradw1} = x * \mathbf{delta1}$                ▷ gradient of loss wrt $\mathbf{w1}$
    $\mathbf{gradb1} = \mathbf{delta1}$                ▷ gradient of loss wrt $\mathbf{b1}$
    Append $\mathbf{gradw1}$, $\mathbf{gradb1}$,$\mathbf{gradw2}$, $\mathbf{gradb2}$ to form vector $\mathbf{gradw}$

function GRADIENT DESCENT
    Let $\eta$ be the learning rate
    for each $w_i$ in $\mathbf{w}$ do
        $w_i = w_i - \eta * \text{grad}w_i$
```