4033/5033 Assignment: Neural Network

Sasank Sribhashyam

In this assignment, we will implement gradient descent and stochastic gradient descent, and derive the update rule for neural network weights.

Task 1. Implement gradient descent from scratch and apply it to find a (local) minimum point for function $f(x) = e^x - x^3$. Show a curve of $f(x)$ versus update numbers in Fig 1. In this figure, x-axis is the number of updates and y-axis is $f(x)$ with the updated x's. (Basically, after every update of x, you should evaluate $f(x)$ and get a point in the figure. Connecting all points gives a curve.) Pick the learning rate and the maximum number of updates yourself, but try to show some convergence in your curve.

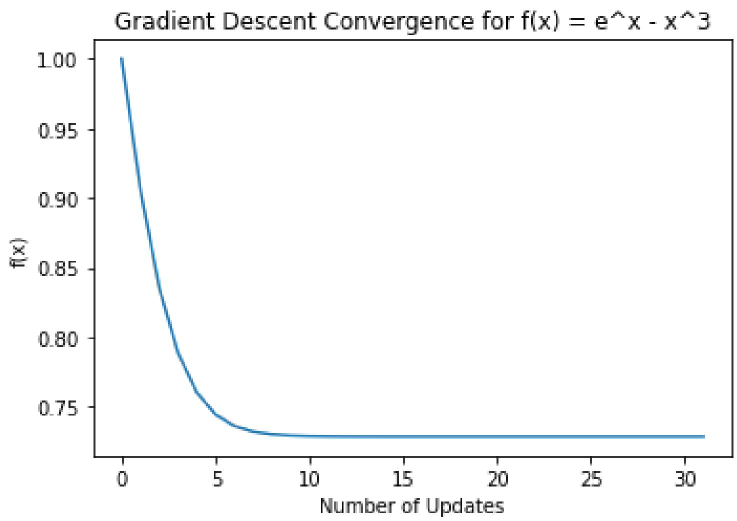

Fig. 1. A Small Neural Network

Task 2. Below is a small neural network model with a differentiable activation function σ and a set of weights w_{ij}'s and β_j's. Suppose we apply back-propagation to train this model on one instance (x, y) by minimizing the least square error i.e.,

$$L = (\hat{y}(x) - y)^2. \tag{1}$$

Derive the update rule for w_{21}. You need to evaluate $\frac{\partial L}{\partial w_{21}}$ using the following notations: $-\frac{\partial \sigma(w_{1i}x_1 + w_{2i}x_2)}{\partial(w_{1i}x_1 + w_{2i}x_2)} = \sigma'_{z_i}$. $-\frac{\partial \sigma(\beta_1 z_1 + \beta_2 z_2)}{\partial(\beta_1 z_1 + \beta_2 z_2)} = \sigma'_y$.

1 Deriving the Update Rule for w_{21} with Backpropagation

In this section, we will go through the step-by-step process of deriving the update rule for w_{21} using backpropagation. We will make use of the chain rule and the provided notations.

Step 1: Define the Loss Function We start with the loss function L:

$$L = (\hat{y}(x) - y)^2$$

Step 2: Compute the Output of the Network Calculate the output of the network:

$$\hat{y}(x) = \sigma(w_{11}x_1 + w_{21}x_2)$$

Step 3: Calculate the Error Term To calculate the error term δ, which represents the gradient of the loss with respect to the network's output:

$$\delta = \frac{\partial L}{\partial \hat{y}(x)} = 2(\hat{y}(x) - y)$$

Step 4: Calculate $\frac{\partial \hat{y}(x)}{\partial w_{21}}$ Now, let's calculate the gradient of the output with respect to w_{21} using the chain rule:

$$\frac{\partial \hat{y}(x)}{\partial w_{21}} = \frac{\partial \sigma(w_{11}x_1 + w_{21}x_2)}{\partial w_{21}}$$

Using the provided notation: $\frac{\partial \sigma(w_{11}x_1 + w_{21}x_2)}{\partial (w_{11}x_1 + w_{21}x_2)} = \sigma'(z_1)$, where $z_1 = w_{11}x_1 + w_{21}x_2$.

$$\frac{\partial z_1}{\partial w_{21}} = x_2 \quad \text{(partial derivative of } z_1 \text{ with respect to } w_{21}\text{)}$$

Now, by applying the chain rule:

$$\frac{\partial \hat{y}(x)}{\partial w_{21}} = \sigma'(z_1) \cdot \frac{\partial z_1}{\partial w_{21}} = \sigma'(z_1) \cdot x_2$$

Step 5: Update Rule for w_{21} Finally, update w_{21} using the gradient descent algorithm:

$$w_{21_{\text{new}}} = w_{21} - \text{learning\_rate} \cdot \delta \cdot \frac{\partial \hat{y}(x)}{\partial w_{21}}$$

Therefore, the update rule for w_{21} is:

$$w_{21_{\text{new}}} = w_{21} - \text{learning\_rate} \cdot 2(\hat{y}(x) - y) \cdot \sigma'(z_1) \cdot x_2$$

Where $\sigma'(z_1)$ represents the derivative of the activation function with respect to z_1, and z_1 is the weighted sum of the inputs to neuron 1.