1 Loss Function in Forward-Forward Algorithm

In the novel approach of neural network training, known as the Forward-Forward Algorithm, an elegantly designed objective function is at its heart. The objective function concurrently maximizes the favorability of positive data and minimizes that of negative data. Formally, the loss function is defined as follows:

$$\text{loss} = \frac{1}{N} \sum_{i=1}^{N} \left(-\log\left(1 + \exp\left(-\left(g_{\text{pos},i} - \text{threshold}\right)\right)\right) + \log\left(1 + \exp\left(g_{\text{neg},i} - \text{threshold}\right)\right)\right)$$

Where:

- N signifies the total count of elements in the tensors g_{pos} and g_{neg}.
- $g_{\text{pos},i}$ and $g_{\text{neg},i}$ denote the i-th elements of the tensors g_{pos} and g_{neg}, respectively.
- threshold represents a scalar value modulating the 'goodness' measure.
- The functions log and exp apply element-wise, ensuring the proper computation of the overall loss value.
- The symbol $\sum$ indicates a summation over all elements resulting from the log operations, followed by division by N to derive the mean loss.

2 Assessing Neuronal Activity

The Forward-Forward Algorithm grants each layer of the network an individual objective function. The mean squared output value of the neurons, denoted by y_j, serves to calculate the average activity within each layer:

$$g = \frac{1}{M} \sum_{j=1}^{M} y_j^2$$

Where:

- M stands for the total number of neurons within a layer.
- y_j refers to the output value of the j-th neuron.
- The symbol $\sum$ signifies a summation of all the squared y_j values, calculating the mean squared activity of the layer.