HomeWork No.6 - Neural Networks & Deep Learning

Name: Ajay Reddy kudumula
Euid: 11580520

Q.1 Give the Weights and structure of a neural network with a Sigmoid output activation and one hidden layer with a ReLU activation, that can represent the exclusive - or function ($\oplus$) of two Booleans...

Ans:- First, We need to express the XOR function in terms of other logical operations. XOR can be expressed as (A AND (NOT B)) OR ((NOT A) AND B).

Next We need to design a neural network that can represent this function. We will need a hidden layer with two neurons (one for each term in the OR operation), and an output layer with a single neuron.

Finally, We need to assign weights to the connections in network. The weights for the connections from the input layer to the hidden layer should be set to implement the AND and NOT operations, and the weights for the connections from the hidden layer to the output layer should be set to the OR operation.

Input to Hidden layer:

- Neuron 1: [20, -20], bias: -10
- Neuron 2: [-20, 20], bias: 10

Hidden to output layer:

- Neuron: [20, +20], bias: -10

The neural network that can represent the XOR function has a hidden layer with two neurons using ReLU function and an output layer with one neuron using sigmoid function. The weights for the connections from the input layer to the hidden layer are $[20, -20]$ and $[-20, 20]$, with biases of -10 and 10 respectively. The weights for the connections from the hidden layer to the output layer are $[20, +20]$ with the bias of -10.

Q.2) Give the Pseudocode for Conv1D, for one-dimensional convolutions (the one dimension version of Fig 8.9). What hyperparameters are required? The Pseudocode does not include all the hyperparameters of Keras of PyTorch. For two of the hyperparameters of one of these, show how the Pseudocode can be extended to include this.

Ans Pseudocode for Conv1D is as follows:

- Initialize the filter weights and bias randomly
- For each position in the input, compute the dot product between the input and the filter weights, then add the bias.
- Apply a non-linear activation function to the result of the dot product and bias addition.
- Repeat steps 2 and 3 for each filter in the layer.

The Pseudocode for Conv1D is as follows:

The hyperparameters which we use are:

1. input_data : The input one-dimensional assay
2. kernel : The one-dimensional convolutional kernel.
3. stride : The step size used when sliding the kernel over the input.
4. padding : The number of zeros added to the input data before appling the convolution.