Machine Learning

SEU — 2024 Spring

Assignment 3

58122204 Xie xing *Instructor: Xu-Ying Liu*

§ Model Selection and Evaluation & Neural Networks §

Problem 1: Multi-layer Perception

(1) Given the network structure with one hidden layer and two neurons, we need to configure the weights and biases to model the XOR function $x_1 \oplus x_2$.

Neural Network Configuration:

- **First Hidden Neuron:**
  - Weights: $w_{11} = 1$, $w_{12} = 1$
  - Bias: $b_1 = -1$

  This gives the ReLU output:

$$\text{ReLU}(w_{11}x_1 + w_{12}x_2 + b_1) = \text{ReLU}(x_1 + x_2 - 1)$$

- **Second Hidden Neuron:**
  - Weights: $w_{21} = 1$, $w_{22} = 1$
  - Bias: $b_2 = -2$

  This gives the ReLU output:

$$\text{ReLU}(w_{21}x_1 + w_{22}x_2 + b_2) = \text{ReLU}(x_1 + x_2 - 2)$$

- **Output Neuron:**
  - Weights: $w_{01} = 1$, $w_{02} = -2$
  - Bias: $b_0 = 0$

  The output is computed as:

$$f(w_{01}a_1 + w_{02}a_2 + b_0)$$

Output Analysis:

- When $x_1 = x_2 = 0$: Both hidden neurons output 0, leading to an output of 0.
- When $x_1 = 1, x_2 = 0$ or $x_1 = 0, x_2 = 1$: The first hidden neuron outputs 1, the second outputs 0, and the final output is 1.
- When $x_1 = x_2 = 1$: The first hidden neuron outputs 1, the second outputs 0, leading to an output of 0.

Thus, the network successfully models the XOR function.

This assignment is due May 17, 2024 and the date of submission is May 2, 2024.

(2) Given the neural network structure with boolean inputs $x \in \{0,1\}^p$ and boolean output $y \in \{0,1\}$, we need to demonstrate that such a network can implement any arbitrary boolean function $h : \{0,1\}^p \to \{0,1\}$. We are allowed to use any finite number of neurons in the hidden layer, each followed by a threshold function.

Strategy

The strategy involves using a hidden layer that has enough neurons to represent every possible input combination. There are 2^p possible combinations for p boolean inputs.

Configuration

- **Hidden Layer:** Introduce 2^p neurons in the hidden layer. Each neuron n_i is dedicated to activating for exactly one input combination that maps to 1 in the function h.
- **Weights and Biases:** For each neuron n_i, configure the weights $\mathbf{w}_i$ and bias b_i such that:

$$n_i = \text{ReLU}(\mathbf{w}_i \cdot \mathbf{x} + b_i) \tag{1.1}$$

  where $\mathbf{w}_i$ has a large positive value for inputs that should trigger n_i and b_i is set to slightly less than the negative sum of the weights so that n_i activates only when its corresponding input pattern is present.

- **Output Layer:** The output neuron uses weights of 1 for all connections from the hidden neurons and a bias of -0.5 (assuming activation if the sum is positive). It computes the final output as:

$$y = f\left(\sum_{i=1}^{2^p} n_i - 0.5\right) \tag{1.2}$$

  where $f(v) = 1$ if $v > 0$ otherwise $f(v) = 0$.

Result

This configuration allows each neuron in the hidden layer to represent a specific pattern of the input that corresponds to an output of 1 according to the function h. The output neuron effectively sums these activations to determine if the input combination should result in a 1 or 0, thus being able to represent any arbitrary boolean function h.

Problem 2: Gradient explosion and gradient vanishing

(1)

(2)

Problem 3: CNN

(1) The total number of parameters in this neural network is 12.

(2)

(3)

Problem 4: CNN

Problem 5: RNN: BPTT