HW Neural Networks

1 Single-Layer Neural Networks

1.1 Exercise 1: Basic Perceptron

Consider a single perceptron with weights $w_1 = 0.5$, $w_2 = -0.3$, and bias $b = 0.2$. Using the step activation function:

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

(a) Calculate the output for input $\mathbf{x} = [1.0, 2.0]^T$.

1.2 Exercise 2: Linear Regression as a Neural Network

A single-layer network without activation function can represent linear regression. Given training data:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

(a) Write the output of a single neuron as $\hat{y} = wx + b$.

(b) Define the mean squared error (MSE) loss function.

(c) Derive the gradient of the loss with respect to w and b.

(d) For the dataset $\{(1, 2), (2, 4), (3, 5)\}$, compute one gradient descent update with learning rate $\eta = 0.1$, starting from $w = 0$, $b = 0$.

1.3 Exercise 3: Binary Classification

Design a single-layer network for the logical AND function. The truth table is:

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

(a) Find weights w_1, w_2, and bias b that correctly classify all points using a step activation.

(b) Explain why the XOR function cannot be solved by a single-layer perceptron.

2 Two-Layer Neural Networks

2.1 Exercise 4: Forward Propagation

Consider a two-layer network with architecture: 2 inputs → 3 hidden units → 1 output. Given:

$$W^{(1)} = \begin{bmatrix} 0.5 & 0.2 \\ -0.3 & 0.4 \\ 0.1 & -0.6 \end{bmatrix}, \quad \mathbf{b}^{(1)} = \begin{bmatrix} 0.1 \\ -0.2 \\ 0.3 \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} 0.7 & -0.4 & 0.2 \end{bmatrix}, \quad b^{(2)} = 0.1$$

Use sigmoid activation $\sigma(z) = \frac{1}{1+e^{-z}}$ for all layers.

(a) For input $\mathbf{x} = [1, 1]^T$, compute the hidden layer activations $\mathbf{h}$.

(b) Compute the final output $\hat{y}$.

2.2 Exercise 5: XOR Problem

Solve the XOR problem using a two-layer network.

(a) Design a network with 2 inputs, 2 hidden units (with step activation), and 1 output.

(b) Manually find weights that solve XOR by constructing hidden units that compute AND and NAND operations.

(c) Verify your solution on all four input combinations.

3 Training and Testing

3.1 Extra-Credit Project

Consider the dataset, available here, in which each sample consists of an input represented as a binary number and an output label of 0, 1, or 2. Randomly divide the dataset into two subsets: one for training and the other for testing. Using Python with either PyTorch or scikit-learn, implement a classification model and evaluate its performance using accuracy, precision, recall, and F1 score.