ANN Assignment Solution

This document provides the solution for the Artificial Neural Network (ANN) assignment based on the specified requirements.

Task Requirements

- **Activation Function**: `tanh` (Hyperbolic Tangent)
- **Weights**: Randomly chosen from the interval `[-0.5, 0.5]`
- **Biases**: `b1 = 0.5`, `b2 = 0.7`
- **Output**: Print the final network output.

Python Implementation

Python

```
import numpy as np

def tanh(x):
    """Hyperbolic Tangent activation function."""
    return np.tanh(x)

def run_ann_assignment():
    # 1. Define inputs (using example inputs: 0.05, 0.10)
    inputs = np.array([0.05, 0.10])

    # 2. Choose weights randomly from interval [-0.5, 0.5]
    # Weights for Input to Hidden (2x2 matrix)
    weights_input_hidden = np.random.uniform(-0.5, 0.5, (2, 2))
    # Weights for Hidden to Output (2x2 matrix)
    weights_hidden_output = np.random.uniform(-0.5, 0.5, (2, 2))

    # 3. Biases b1, b2 = 0.5, 0.7 respectively
    bias_hidden = 0.5
    bias_output = 0.7

    print("--- ANN Assignment Task ---")
    print(f"Inputs: {inputs}")
    print(f"Random Weights (Input to Hidden):\n{weights_input_hidden}")
    print(f"Random Weights (Hidden to Output):\n{weights_hidden_output}")
    print(f"Biases: b1={bias_hidden}, b2={bias_output}")
    print("-" * 30)

    # Forward Pass: Input to Hidden Layer
```

```
    net_hidden = np.dot(inputs, weights_input_hidden) + bias_hidden
    out_hidden = tanh(net_hidden)

    print(f"Hidden Layer Net Inputs: {net_hidden}")
    print(f"Hidden Layer Outputs (after tanh): {out_hidden}")

    # Forward Pass: Hidden to Output Layer
    net_output = np.dot(out_hidden, weights_hidden_output) + bias_output
    out_output = tanh(net_output)

    print("-" * 30)
    print(f"Output Layer Net Inputs: {net_output}")
    print(f"Final Network Output: {out_output}")

if __name__ == "__main__":
    run_ann_assignment()
```

Explanation of the Network Structure

1. **Input Layer**: Two inputs ($x_1 = 0.05, x_2 = 0.10$).
2. **Hidden Layer**: Two neurons (H_3, H_4) with a bias $b_1 = 0.5$.
3. **Output Layer**: Two neurons (O_1, O_2) with a bias $b_2 = 0.7$.
4. **Activation**: The `tanh` function is applied to the net input of each neuron in both the hidden and output layers.

Sample Output

Plain Text

```
--- ANN Assignment Task ---
Inputs: [0.05 0.1 ]
Random Weights (Input to Hidden):
[[-0.045553    0.30079943]
 [ 0.43280671  0.41861237]]
Random Weights (Hidden to Output):
[[0.29883808 0.18496082]
 [0.40709721 0.290317  ]]
Biases: b1=0.5, b2=0.7
------------------------------
Hidden Layer Net Inputs: [0.54100302 0.55690121]
Hidden Layer Outputs (after tanh): [0.49374684 0.50567464]
------------------------------
```

```
Output Layer Net Inputs: [1.0534091  0.93812976]
Final Network Output: [0.78312821 0.73436179]
```