Assignment 1: Neural network architecture and forward pass

1. Neural Network with 1 Hidden Layer

Objective

Implement a neural network with specified architecture and loss function.

Specifications

- **Input:** 3 units
- **Hidden Layer:** 1 layer with 3 units, ReLU activation
- **Output:** Squared loss function

Instructions

1. Initialize weights and biases randomly for each layer.
2. Implement a forward pass function to compute the output of the neural network.
3. Use squared loss as the loss function.

Your Task

Write Python code to implement the neural network described above. Include comments to explain each step of your implementation. Test your implementation with sample input data. **Don't use packages like PyTorch or TensorFlow.**

Hint: Remember the universal dot product formula which works for the forward pass. np.dot(X, W) + B. This formula will be your friend!

2. Neural Network with 2 Hidden Layers

Objective

Implement a neural network with specified architecture and loss function.

Specifications

- **Input:** 2 units
- **Hidden Layers:** 2 layers with 3 units each, ReLU activation in the first layer, Softmax activation in the last layer
- **Output:** Cross entropy loss function

Instructions

1. Initialize weights and biases randomly for each layer.
2. Implement a forward pass function to compute the output of the neural network.
3. Use cross entropy loss as the loss function.

Your Task

Write Python code to implement the neural network described above. Explain briefly how Softmax activation differs from ReLU. Test your implementation with sample input data.

Don't use packages like PyTorch or TensorFlow.

Hint: Remember the universal dot product formula which works for the forward pass. np.dot(X, W) + B. This formula will be your friend!