ID : 4221386

NAME : Moaz Awad Ali

https://github.com/AmazingMoaaz/AI323-Computational-Neuroscience

```python
import numpy as np
```

[1] Python

```python
# Define tanh activation function
def tanh(x):
    return np.tanh(x)
```

[2] Python

```python
# Initialize weights randomly in the range [-0.5, 0.5]
def init_weight():
    return np.random.uniform(-0.5, 0.5)
```

[3] Python

```python
# Network structure
network = {
    'inputs': {'i1': 0.05, 'i2': 0.10},
    'biases': {'b1': 0.5, 'b2': 0.7},
    'weights': {
        'w1': init_weight(), 'w2': init_weight(),
        'w3': init_weight(), 'w4': init_weight(),
        'w5': init_weight(), 'w6': init_weight(),
        'w7': init_weight(), 'w8': init_weight()
    }
}
```

[4] Python

```python
def forward_pass(network):
    # Hidden layer computations
    h1_input = network['inputs']['i1'] * network['weights']['w1'] + network['inputs']['i2'] * network['weights']['w3'] + network['biases']['b1']
    h2_input = network['inputs']['i1'] * network['weights']['w2'] + network['inputs']['i2'] * network['weights']['w4'] + network['biases']['b1']
    h1_output = tanh(h1_input)
    h2_output = tanh(h2_input)

    # Output layer computations
    o1_input = h1_output * network['weights']['w5'] + h2_output * network['weights']['w7'] + network['biases']['b2']
    o2_input = h1_output * network['weights']['w6'] + h2_output * network['weights']['w8'] + network['biases']['b2']

    o1_output = tanh(o1_input)
```

[5]

File Edit Selection View Go Run Terminal Help

AI323-Computational-Neuroscience

Assignment (1).ipynb

Assignment (1).ipynb

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ... Python 3.11.5

```
    o1_output = tanh(o1_input)
    o2_output = tanh(o2_input)

    return {
        'h1_input': h1_input, 'h2_input': h2_input,
        'h1_output': h1_output, 'h2_output': h2_output,
        'o1_input': o1_input, 'o2_input': o2_input,
        'o1_output': o1_output, 'o2_output': o2_output
    }
```

[5] Python

```
def compute_error(results):
    target_o1 = 0.01
    target_o2 = 0.99
    error_o1 = 0.5 * (target_o1 - results['o1_output']) ** 2
    error_o2 = 0.5 * (target_o2 - results['o2_output']) ** 2
    return error_o1 + error_o2
```

[6] Python

```
# Perform forward pass
results = forward_pass(network)
print(f"Output o1: {results['o1_output']}")
print(f"Output o2: {results['o2_output']}")
```

[7] Python

```
Output o1: 0.6264098709610402
Output o2: 0.6157476755509717
```

```
# Compute error
error = compute_error(results)
print(f"Error: {error}")
```

[8] Python

```
Error: 0.2600129656868535
```

```
print("https://github.com/AmazingMoaaz/AI323-Computational-Neuroscience")
```

[1] 0.0s Python

```
https://github.com/AmazingMoaaz/AI323-Computational-Neuroscience
```

main Launchpad 0 1 Connect Ln 1, Col 74 Spaces: 8 Cell 10 of 10 Go Live