Pusat
Sains Matematik

CENTRE OF MATHEMATICAL SCIENCES
UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

BSD2513 ARTIFICIAL INTELLIGENCE
SEM I 2025/2026

LAB REPORT 4

TITLE	Chapter 4: Neural Networks
LECTURER'S NAME	Dr. Ku Muhammad Na'im Ku Khalif
NAME	Sara Khadija binti Saidin
ID NUMBER	SD23061
SUBMISSION DATE	5/1/2026

<u>Question 1</u>
General Knowledge
Describe the component in networks that are built in one neural network.

(5 Marks)
(CO1 PO1)

A neural network is built from several fundamental components that work together to process information and learn patterns from data. The main components are:

1. **Neurons (Nodes/Units)**
   Neurons are the basic processing units of a neural network. Each neuron receives input values, performs a weighted sum, applies an activation function, and produces an output. Neurons are inspired by biological neurons and are responsible for learning representations of data.

2. **Layers**
   A neural network is organized into layers of neurons:

   - **Input layer**: Receives raw data or features.

   - **Hidden layers**: Perform intermediate computations and extract complex patterns.

   - **Output layer**: Produces the final prediction or classification result.
     The number of layers and neurons determines the depth and capacity of the network.

3. **Weights**
   Weights are parameters that control the strength of connections between neurons. They determine how much influence an input has on a neuron's output. During training, weights are adjusted to minimize prediction error.

4. **Biases**
   Biases are additional parameters added to the weighted sum to allow the model to fit data more flexibly. They help shift the activation function and improve learning, especially when inputs are zero.

5. **Activation Functions**
   Activation functions introduce non-linearity into the network, enabling it to model complex relationships. Common activation functions include ReLU, sigmoid, tanh, and softmax.

6. **Loss (Cost) Function**
   The loss function measures the difference between the network's predictions and the actual target values. It guides the learning process by indicating how well the model is performing.

7. **Optimizer (Learning Algorithm)**
   The optimizer updates the weights and biases based on the loss function. Examples include Gradient Descent, Adam, and RMSprop. The optimizer determines how quickly and efficiently the network learns.

8. **Connections (Edges)**
   Connections link neurons between layers and carry signals weighted by their corresponding weights. These connections define the network's structure.

<u>Question 2</u>

Python: Neural Network

Create three separate web applications to visualise the following activation functions:

1. Rectified Linear Unit (ReLU),
2. Sigmoid,
3. Hyperbolic Tangent (Tanh)

Coding:

ReLu Activation Function

```
import streamlit as st
import numpy as np
import matplotlib.pyplot as plt

st.set_page_config(page_title="ReLU Activation Function",
layout="centered")

st.title("ReLU Activation Function")
st.write("Rectified Linear Unit (ReLU): f(x) = max(0, x)")

# Slider for x range
x_min, x_max = st.slider(
    "Select range of x values",
    min_value=-20,
    max_value=20,
    value=(-10, 10)
)
x = np.linspace(x_min, x_max, 400)
relu = np.maximum(0, x)

# Plot
fig, ax = plt.subplots()
ax.plot(x, relu)
ax.set_xlabel("x")
ax.set_ylabel("ReLU(x)")
ax.set_title("ReLU Activation Function")
ax.grid(True)

st.pyplot(fig)
```

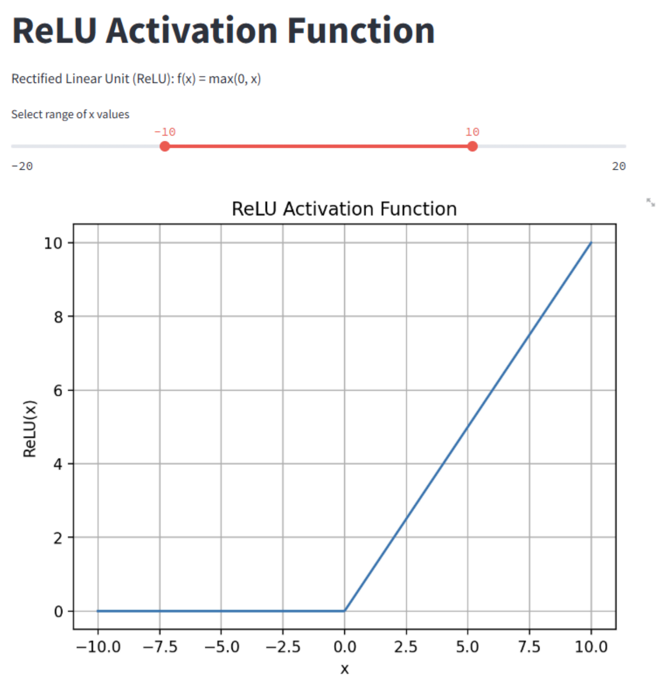
Sigmoid Activation Function

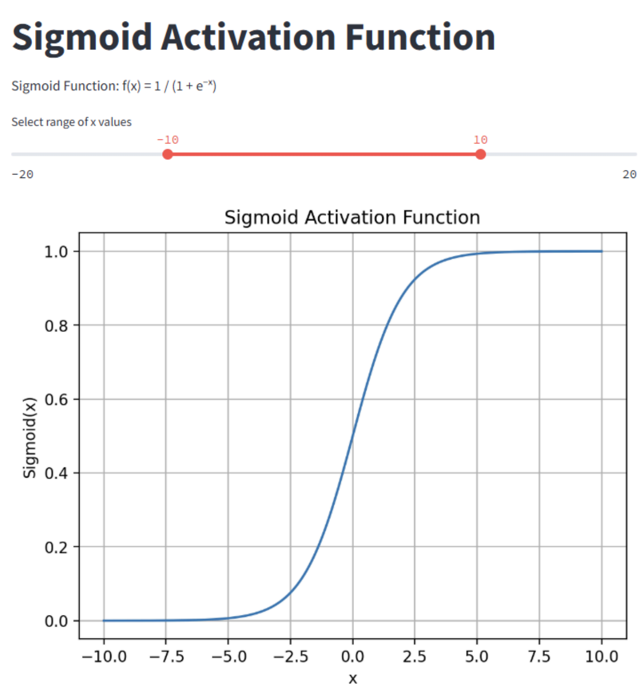
```
import streamlit as st
import numpy as np
import matplotlib.pyplot as plt

st.set_page_config(page_title="Sigmoid Activation Function",
layout="centered")

st.title("Sigmoid Activation Function")
st.write("Sigmoid Function: f(x) = 1 / (1 + e⁻ˣ)")

# Slider for x range
x_min, x_max = st.slider(
    "Select range of x values",
    min_value=-20,
    max_value=20,
    value=(-10, 10)
)

x = np.linspace(x_min, x_max, 400)
sigmoid = 1 / (1 + np.exp(-x))

# Plot
fig, ax = plt.subplots()
ax.plot(x, sigmoid)
ax.set_xlabel("x")
ax.set_ylabel("Sigmoid(x)")
ax.set_title("Sigmoid Activation Function")
ax.grid(True)

st.pyplot(fig)
```

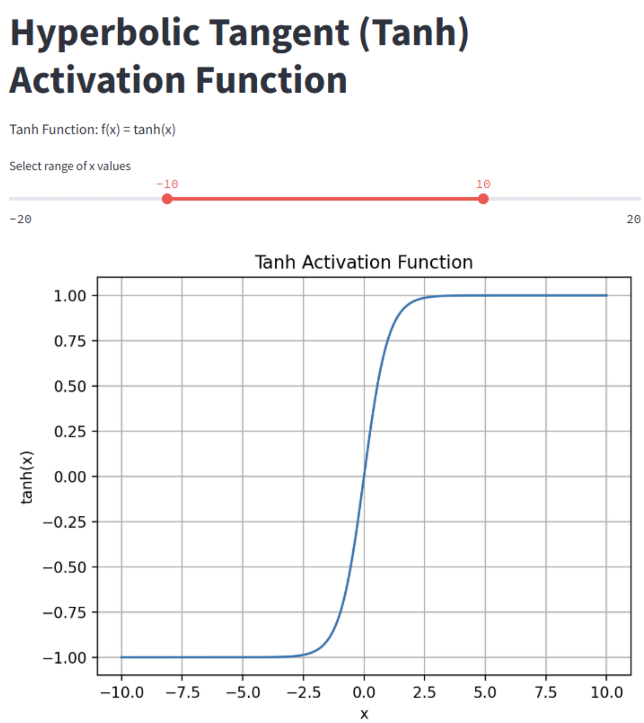
Tanh Activation Function

```
import streamlit as st
import numpy as np
import matplotlib.pyplot as plt

st.set_page_config(page_title="Tanh Activation Function",
layout="centered")

st.title("Hyperbolic Tangent (Tanh) Activation Function")
st.write("Tanh Function: f(x) = tanh(x)")

# Slider for x range
x_min, x_max = st.slider(
    "Select range of x values",
    min_value=-20,
    max_value=20,
    value=(-10, 10)
)

x = np.linspace(x_min, x_max, 400)
tanh = np.tanh(x)

# Plot
fig, ax = plt.subplots()
ax.plot(x, tanh)
ax.set_xlabel("x")
ax.set_ylabel("tanh(x)")
ax.set_title("Tanh Activation Function")
ax.grid(True)

st.pyplot(fig)
```

Output:


ReLU Activation Function
Rectified Linear Unit (ReLU): f(x) = max(0, x)
Select range of x values
-10
10
-20
20
ReLU Activation Function
ReLU(x)
x
10
8
6
4
2
0
−10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 10.0


Sigmoid Activation Function
Sigmoid Function: f(x) = 1 / (1 + e⁻ˣ)
Select range of x values
-10
10
-20
20
Sigmoid Activation Function
Sigmoid(x)
x
1.0
0.8
0.6
0.4
0.2
0.0
−10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 10.0



Deployment:

GitHub: https://github.com/SaraKhadija/sd23061_lab4_intro_ai

Streamlit:

ReLu Activation Function: https://reluactivationfunctionapplab4.streamlit.app/

Sigmoid Activation Function: https://sigmoidactivationfunctionlab4.streamlit.app/

Tanh Activation Function: https://tanhactivationfunctionlab4.streamlit.app/