**EXPERIMENT – 3.1**

**Mapped Course Outcome**

CO1: Identify and describe soft computing techniques and their roles in building intelligent machines.

**AIM:**  
Write a program to build a neural network implementing back propagation network.

**Theory**  
Backpropagation is a supervised learning algorithm used for training artificial neural networks. It involves two phases: forward propagation, where inputs pass through the network to generate output, and backward propagation, where the network's error is calculated and minimized by adjusting the weights using gradient descent. This process iterates until the network learns to produce the correct output.

**Procedure:**

**Step 1: Setup and Installation**

1. **Install Anaconda:**
   * Visit [Anaconda Downloads](https://www.anaconda.com/products/individual).
   * Download the installer suitable for your operating system.
   * Launch the installer and follow the instructions.
   * Choose whether to add Anaconda to your PATH environment variable (recommended not to add).
   * Register Anaconda as your default Python.
   * Click the Install button.
   * After installation, click Next and then Finish.
2. **Install Required Libraries:**
   * Open Anaconda Navigator.
   * In the search box, type tensorflow and install it.
   * Similarly, search for keras and install it.
   * Ensure other necessary libraries (like numpy, pandas, matplotlib) are installed.

**Step 2: Implementing the Neural Network**

1. **Import Necessary Libraries:**

python

Copy code

import numpy as np

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

1. **Prepare the Dataset:**

Use a sample dataset like the Iris dataset. Here, we'll create a simple dataset.

python

Copy code

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import OneHotEncoder

# Load dataset

data = load\_iris()

X = data.data

y = data.target.reshape(-1, 1)

# One hot encoding

encoder = OneHotEncoder(sparse=False)

y = encoder.fit\_transform(y)

# Split dataset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

1. **Build the Neural Network Model:**

python

Copy code

model = Sequential([

Dense(10, input\_shape=(4,), activation='relu'),

Dense(10, activation='relu'),

Dense(3, activation='softmax')

])

1. **Compile the Model:**

python

Copy code

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

1. **Train the Model:**

python

Copy code

model.fit(X\_train, y\_train, epochs=50, batch\_size=5)

1. **Evaluate the Model:**

python

Copy code

loss, accuracy = model.evaluate(X\_test, y\_test)

print(f'Loss: {loss}, Accuracy: {accuracy}')

**Step 3: Running the Program**

1. Open Jupyter Notebook from Anaconda Navigator.
2. Create a new Python 3 notebook.
3. Copy and paste the above code sections into the notebook cells.
4. Execute each cell sequentially to build, train, and evaluate the neural network.

**Video Tutorial**

[How to Install TensorFlow and Keras](https://www.youtube.com/watch?v=RgO8BBNGB8w)

**Further Reading**

Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2016). Introduction to Anaconda and Python: Installation and setup. Python for research in psychology, 16(5), S5-S11.

**Prospective Viva Questions**

1. Define Soft Computing and its importance.
2. Explain how Soft Computing can be applied in traffic management.
3. Discuss the advantages of using Jupyter Notebook for data science projects.
4. Describe the purpose and functionalities of NumPy in Python.
5. Elaborate on the uses of Pandas in data manipulation and analysis.