Lab07: Artificial Neural Networks – Predicting Bike Rentals with Weather Data

Course: Programming for Data Science
Professor: Dr. Carlos J. Costa, PhD.

March 19, 2026

Objective

The objective of this lab is to introduce students to the fundamental concepts of Artificial Neural Networks (ANNs) and apply them to a realistic business problem. Students will learn how to model, train, and interpret neural networks using real-world data.

Business Context

You are working for a bike-sharing company. The goal is to predict the number of bike rentals based on weather conditions such as temperature, humidity, and wind speed.

This prediction helps the company:

- Optimize bike distribution
- Improve customer satisfaction
- Reduce operational costs

Part 1: ANN Concepts

Task 1: Conceptual Understanding

Answer the following questions in your own words:

1. What is an Artificial Neural Network?
2. What is a neuron?
3. Explain the concepts of weights, bias, and activation function.
4. What is forward propagation?
5. What is a loss function?
6. What is backpropagation?

Task 2: Mapping Concepts to the Problem

Explain how ANN concepts apply to the bike rental problem:

- Input layer

- Weights
- Output
- Loss function

Part 2: Data Collection (API)

Task 3: Retrieve Weather Data

Use the following Python code to collect real-time weather data:

```
import requests

url = "https://api.open-meteo.com/v1/forecast?latitude=38.72&longitude=-9.14&current_weather=true"

data = requests.get(url).json()

temp = data["current_weather"]["temperature"]
wind = data["current_weather"]["windspeed"]

humidity = 60  # simplified

print(temp, wind, humidity)
```

Task 4: Interpretation

- How does temperature affect bike rentals?
- How does wind influence demand?

Part 3: Dataset Construction

Task 5: Analyze Dataset

```
dataset = [
    [10, 80, 15, 100],
    [15, 70, 10, 150],
    [20, 60, 8, 220],
    [25, 50, 5, 300],
    [30, 40, 3, 280],
    [12, 85, 20, 90]
]
```

Explain:

- Why this dataset is realistic
- What patterns you observe

Part 4: PyTorch Implementation

Task 6: Build the Model

```
import torch
import torch.nn as nn

data = torch.tensor(dataset, dtype=torch.float32)
X = data[:, :3]
y = data[:, 3].unsqueeze(1)

class Model(nn.Module):
    def __init__(self):
        super().__init__()
        self.l1 = nn.Linear(3, 8)
        self.l2 = nn.Linear(8, 1)

    def forward(self, x):
        x = torch.relu(self.l1(x))
        x = self.l2(x)
        return x

model = Model()
```

Task 7: Train the Model

```
import torch.optim as optim

criterion = nn.MSELoss()
optimizer = optim.Adam(model.parameters(), lr=0.01)

for epoch in range(2000):
    pred = model(X)
    loss = criterion(pred, y)

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

Task 8: Make Predictions

```
new_data = torch.tensor([[temp, humidity, wind]])
print(model(new_data))
```

Task 9: Reflection
Identify in the code:

- Where are the weights?
- Where does forward propagation occur?
- Where is backpropagation implemented?

Part 5: TensorFlow Implementation

Task 10: Build and Train Model

```
import tensorflow as tf
import numpy as np

data = np.array(dataset)

X = data[:, :3]
y = data[:, 3]

model = tf.keras.Sequential([
    tf.keras.layers.Dense(8, activation='relu', input_shape=(3,)),
    tf.keras.layers.Dense(1)
])

model.compile(optimizer='adam', loss='mse')

model.fit(X, y, epochs=2000, verbose=0)
```

Task 11: Predict

```
print(model.predict([[temp, humidity, wind]]))
```

Part 6: Business Interpretation

Task 12: Decision-Making

Answer:

1. If predicted rentals = 300, what should the company do?
2. If predicted rentals = 90, what actions should be taken?
3. What are the limitations of this model?