Experiment 9: Back Propagation Neural Network on Student Performance Dataset

---

Aim:

To implement a **Back Propagation Neural Network (BPNN)** to train and update weights and biases for predicting **Result (Pass/Fail)**.

Theory:

- **Back Propagation Neural Network (BPNN)** is a **supervised learning model**.
- It consists of **input, hidden, and output layers**.
- Weights and biases are **updated iteratively** using **error correction** to minimize prediction error.
- Useful for **numerical prediction and classification**.

Dataset (student_performance.arff)

@relation student_performance

@attribute Attendance numeric

@attribute InternalMarks numeric

@attribute AssignmentScore numeric

@attribute SemesterMarks numeric

@attribute Result {Pass, Fail}

@data

80,75,70,85,Pass

60,65,60,70,Pass

50,55,50,45,Fail

90,80,85,90,Pass

70,60,65,60,Pass

45,50,55,50,Fail

85,75,80,88,Pass

Procedure (Using WEKA):

1. Open **WEKA → Explorer**.
2. Click **Open File** → select **student_performance.arff**.
3. Go to **Classify tab**.
4. Choose **Classifier → functions → MultilayerPerceptron**.
5. Configure network parameters:
    - **Learning rate** (e.g., 0.3)
    - **Momentum** (e.g., 0.2)
    - **Training epochs** (e.g., 500)
6. Click **Start** → WEKA trains the neural network.
7. Observe **weight and bias updates** in the output.
8. Predict **Result** for new student records.

Result (Sample / Expected):

- Neural Network trained successfully.
- Weights and biases **updated iteratively** to minimize error.
- Predicted Result for new instance (Attendance=65, InternalMarks=70, AssignmentScore=60, SemesterMarks=68) → **Pass**
- Training Accuracy: 100% (for this small dataset)

Conclusion:

- Back Propagation Neural Network successfully predicted student Result.
- Updating weights and biases improves model **learning and accuracy**.
- WEKA simplifies **neural network training and prediction** without coding.