Neural networks Final project

Project-1 : Predictions for Tabular form dataset (CSV file) using Neural network model

Project Steps:

Problem Definition

- Identify the task (classification or regression)
- Define the input data and desired output
- Explain why a neural network is suitable

Data Acquisition

- Load the dataset from a CSV file
- Describe the dataset source and context
- Inspect the raw format of the data

Exploratory Data Analysis (EDA)

- Show dataset dimensions (rows, columns)
- Show basic statistics (mean, std, min, max)
- Visualize distributions of important features
- Detect missing values or anomalies

Data Cleaning

- Handle missing values (remove, impute, etc.)
- Handle outliers if needed
- Remove duplicate or irrelevant data
- Ensure consistent formatting of numeric values

Data Preprocessing

- Encode categorical variables if present (one-hot or label encoding)
- Normalize / standardize numerical features
- Split dataset into:
    - Training set
    - Validation set (optional)
    - Test set
- Shuffle data

Neural Network Architecture Design

- Determine number of input neurons (based on features)
- Choose number of hidden layers and neurons
- Choose activation functions (ReLU, Sigmoid, etc.)
- Determine number of output neurons and output activation
- Present a diagram of the architecture

Example architecture:

`Input → Hidden Layer → Activation → Hidden Layer → Activation → Output`

Parameter Initialization

- Initialize weights and biases for each layer
- Use random initialization (Gaussian/Uniform)
- Mention possible improvements

Forward Propagation

For each layer compute:

1. Linear transformation:
   $Z=W \cdot X+bZ$

2. Activation:
   $A=f(Z)A$

Store the intermediate values (Z, A) for backpropagation.

Loss Function Computation

Select an appropriate loss function:

For classification:

- Binary Cross-Entropy
- Categorical Cross-Entropy

For regression:

- Mean Squared Error (MSE)

Compute loss over each batch or the whole dataset.

Backpropagation

Compute gradients of the loss with respect to:

- Output layer activation
- Hidden layer activations
- Weights and biases

Use:

- Chain rule
- Derivatives of activation functions

Compute:

dW,db

for each layer.

Parameter Update

Apply gradient descent or an optimization method:

$W = W - \alpha \cdot dW$

Where:

$b = b - \alpha \cdot db$

- α= learning rate

Training Loop

Repeat for a number of epochs:

1. Forward propagation
2. Loss computation
3. Backpropagation
4. Update weights

Track:

- training loss over epochs
- validation loss

Model Evaluation

For classification:

- Accuracy
- Precision, Recall, F1
- Confusion Matrix

For regression:

- MSE, MAE, RMSE
- R^2 score

Assess the model on the test dataset.

Model Testing

- Apply the trained model to new, unseen data
- Compare predicted outputs vs true labels
- Discuss performance

Experiments and Hyperparameter Tuning

Experiment with:

- Different number of hidden layers
- Different number of neurons
- Learning rates
- Activation functions
- Batch sizes

Compare performance results.

Visualization and Reporting

Include visualizations:

- Training loss curve
- Accuracy curve
- Confusion matrix
- Feature distributions

Examples for datasets : you can use any appropriate dataset you want

1. Predict Student Performance (Regression or Classification)

CSV features:

- study hours
- attendance
- assignments
- previous grades

Output: pass/fail or final score

2. Bank Customer Churn Prediction

CSV features:

- age, salary, balance, products, tenure, credit score
- churn label

Task: binary classification (leave/not leave)

---

3. Breast Cancer Prediction (Medical Diagnosis)

CSV: UCI Breast Cancer Dataset

Inputs: cell size, smoothness, uniformity, etc.
Output: malignant / benign

---

4. Predict House Prices (Regression)

CSV features:

- rooms, area, location score, age of building

Task: numeric prediction

---

5. Credit Card Fraud Detection

CSV features: many numerical fields
Problem: extremely imbalanced data

Task: binary classification