EXPERIMENT 7

Build an Artificial Neural Network (ANN) using Keras/TensorFlow

Aim of the Experiment

To build and evaluate an Artificial Neural Network (ANN) model using TensorFlow/Keras for predicting whether a patient is diabetic or non-diabetic based on medical attributes.

Dataset Source

Dataset Name: **Pima Indians Diabetes Dataset**
Source: Kaggle

Link:
https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

Dataset Description

The Pima Indians Diabetes Dataset is a medical dataset used for binary classification.

Dataset Characteristics:

- Total Records: 768
- Total Features: 8 input features
- Target Variable: Outcome (0 or 1)
- Type: Binary Classification

Features:

1. Pregnancies – Number of times pregnant
2. Glucose – Plasma glucose concentration
3. BloodPressure – Diastolic blood pressure
4. SkinThickness – Triceps skin fold thickness
5. Insulin – 2-Hour serum insulin

6. BMI – Body Mass Index
7. DiabetesPedigreeFunction – Genetic diabetes likelihood
8. Age – Age of patient

Target Variable:

- 0 → Non-diabetic
- 1 → Diabetic

The dataset contains medical diagnostic measurements and is moderately imbalanced.

Theory of Artificial Neural Network (ANN)

An Artificial Neural Network is a supervised machine learning algorithm inspired by the biological neural network of the human brain.

It consists of:

- Input Layer
- Hidden Layers
- Output Layer

Each neuron performs:

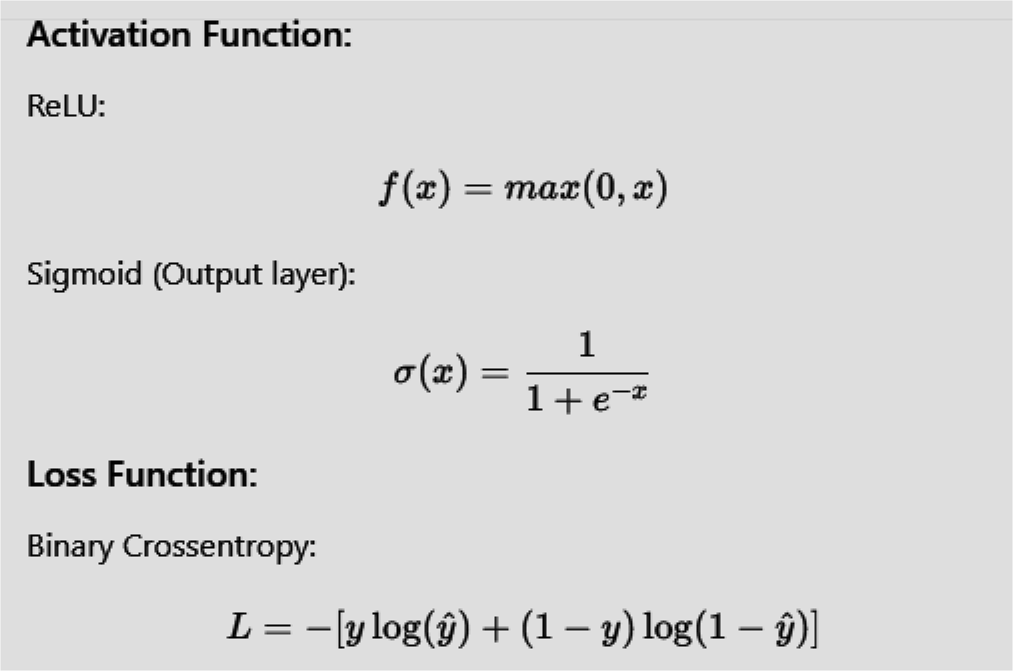
Weighted Sum:

$$Z = W^T X + b$$

Where:
W = Weights
X = Input
b = Bias

Activation Function:

ReLU:



Activation Function:

ReLU:

$$f(x) = max(0, x)$$

Sigmoid (Output layer):

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Loss Function:

Binary Crossentropy:

$$L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

The network updates weights using Gradient Descent to minimize the loss.

Algorithm Limitations

1. Requires large dataset for better performance
2. Can suffer from overfitting
3. Computationally expensive
4. Difficult to interpret (Black-box model)
5. Sensitive to hyperparameters
6. Performance decreases with highly imbalanced data

ANN may not perform well on:

- Very small datasets
- Highly noisy data
- Linearly separable simple problems (simpler models work better)

Methodology / Workflow

Step-by-Step Process:

1. Dataset Collection from Kaggle
2. Data Preprocessing
   - Handling features
   - Feature scaling using StandardScaler
3. Train-Test Split (80%-20%)
4. ANN Model Building
5. Model Compilation
6. Model Training
7. Model Evaluation
8. Performance Analysis
9. Hyperparameter Tuning

Workflow Diagram:

Dataset → Preprocessing → Train-Test Split → ANN Model → Training → Evaluation → Performance Analysis

Performance Analysis

The improved ANN model achieved:

- Test Accuracy: 75.32%
- Precision (Diabetic): 0.65
- Recall (Diabetic): 0.65
- F1-score: 0.65

Confusion Matrix:

- True Negative: 80
- True Positive: 36
- False Positive: 19
- False Negative: 19

The model performs better in predicting non-diabetic cases compared to diabetic cases. The accuracy and loss curves indicate stable convergence with slight overfitting. Overall, the model demonstrates moderate classification performance

Hyperparameter Tuning

The following hyperparameters were tuned:

1. Number of hidden layers
2. Number of neurons (32 → 16)
3. Dropout layers added (0.3 and 0.2)
4. Epochs increased (up to 200)
5. EarlyStopping callback used
6. Batch size = 32

Impact of Tuning:

- Reduced overfitting
- Improved generalization
- Improved test accuracy from 72% to 75%
- Reduced false negatives

Hyperparameter tuning improved overall model stability.

Performance Observations :

The improved ANN model achieved a **test accuracy of 75.32%**.

The training and validation accuracy curves show stable learning with only slight overfitting. Both loss curves decrease steadily and then stabilize, indicating proper convergence of the model.

From the confusion matrix:

- 80 samples were correctly classified as non-diabetic.
- 36 samples were correctly classified as diabetic.
- 19 false positives and 19 false negatives were observed.

The model performs better in predicting non-diabetic cases (precision = 0.81) compared to diabetic cases (precision = 0.65). Although performance is moderate, the model demonstrates reasonable generalization ability.

Conclusion

The Artificial Neural Network model was successfully implemented for diabetes prediction. After applying dropout and early stopping, the model achieved 75.32% test accuracy with improved generalization. Although performance is moderate, the model demonstrates the effectiveness of ANN for binary classification problems. Further improvement can be achieved using larger datasets or advanced architectures.