Neural Network Demo Presentation Script

Introduction

Hello everyone, today I'll be walking you through a short Python project that demonstrates how to build a simple neural network using TensorFlow and Keras. Our goal is to predict medical insurance charges based on details like age, BMI, number of children, and smoking status. This will help us understand the basic workflow of any machine learning project — from data preprocessing to training and evaluation.

Step 1: Importing Libraries

We import libraries such as pandas, NumPy, scikit-learn, and TensorFlow. Pandas handles tabular data, NumPy supports mathematical operations, scikit-learn helps with preprocessing and splitting the data, and Keras provides an easy deep-learning interface.

Step 2: Loading the Dataset

We load our dataset using pandas. It includes columns like age, sex, BMI, smoker, region, and the target variable: charges. Viewing the first few rows helps confirm that everything loaded correctly.

Step 3: Data Preprocessing

Neural networks only work with numbers. So, we convert categorical columns like sex, smoker, and region into numeric values using one-hot encoding.

Step 4: Splitting Features and Labels

We separate the dataset into X (features) and y (labels) so the model knows what input leads to what output.

Step 5: Train–Test Split

We split the data into training and testing sets so we can evaluate model performance on unseen data.

Step 6: Feature Scaling

Inputs need to be on similar scales. For example, age may range up to 60 while BMI stays between 20–40. StandardScaler normalizes them for efficient training.

Step 7: Building the Model

We build a Sequential neural network with:

- 64 neurons (ReLU activation)

- 32 neurons (ReLU activation)

- 1 output neuron (for regression)

Step 8: Compiling the Model

We compile the model using Mean Squared Error as the loss function and Adam as the optimizer. We also track Mean Absolute Error.

Step 9: Training the Model

We train the model for 100 epochs using the .fit() function while monitoring validation accuracy.

Step 10: Evaluating the Model

We evaluate the model using test data to understand how well it generalizes.

Step 11: Making Predictions

We use .predict() to generate predictions for new data and compare predicted vs actual values.

Conclusion

We cleaned and prepared the data, built and trained a neural network, evaluated it, and made predictions. The same workflow applies to more complex models like image classification or text processing. Data preprocessing is just as important as the model itself.