Intro to Neural Networks + Overview of AI Architectures

Introduction to Neural Networks

Lesson Objectives

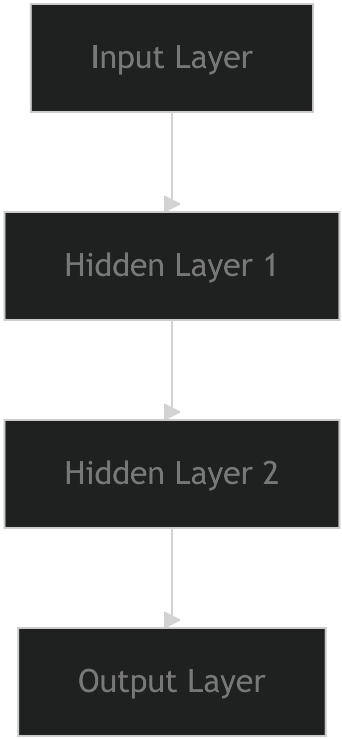
By the end of this lesson, students will be able to:

- Understand the fundamental structure of a neural network.
- Explain how neural networks learn from data.
- Identify why data quality is critical for AI model performance.
- Recognize common data-related issues that impact neural networks (bias, noise, augmentation, and synthetic data).

What is a Neural Network?

Neural networks are a type of machine learning model inspired by the human brain. They consist of interconnected layers of artificial neurons that process input data to identify patterns and make predictions.

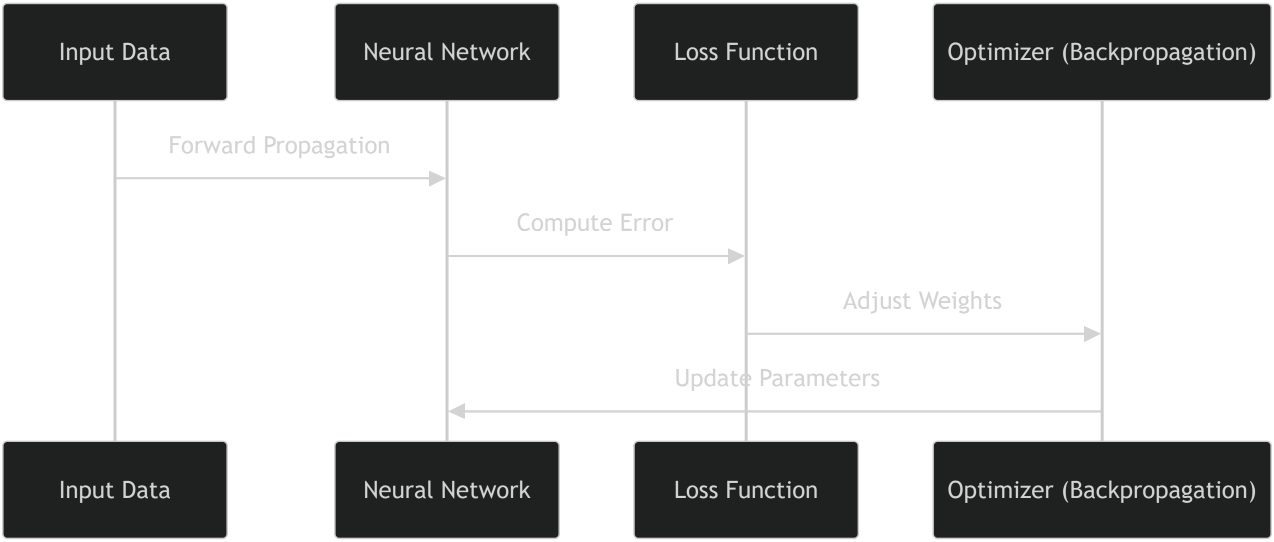
Basic Components of a Neural Network



Key Elements:

- **Neurons (Nodes):** Process input values and pass information forward.
- **Layers:**
  - **Input Layer**: Accepts raw data.
  - **Hidden Layers**: Perform computations and transformations.
  - **Output Layer**: Produces predictions or classifications.
- **Weights & Biases:** Adjust to minimize error during training.
- **Activation Functions:** Determine neuron firing (e.g., ReLU, Sigmoid, Softmax).

How Neural Networks Learn

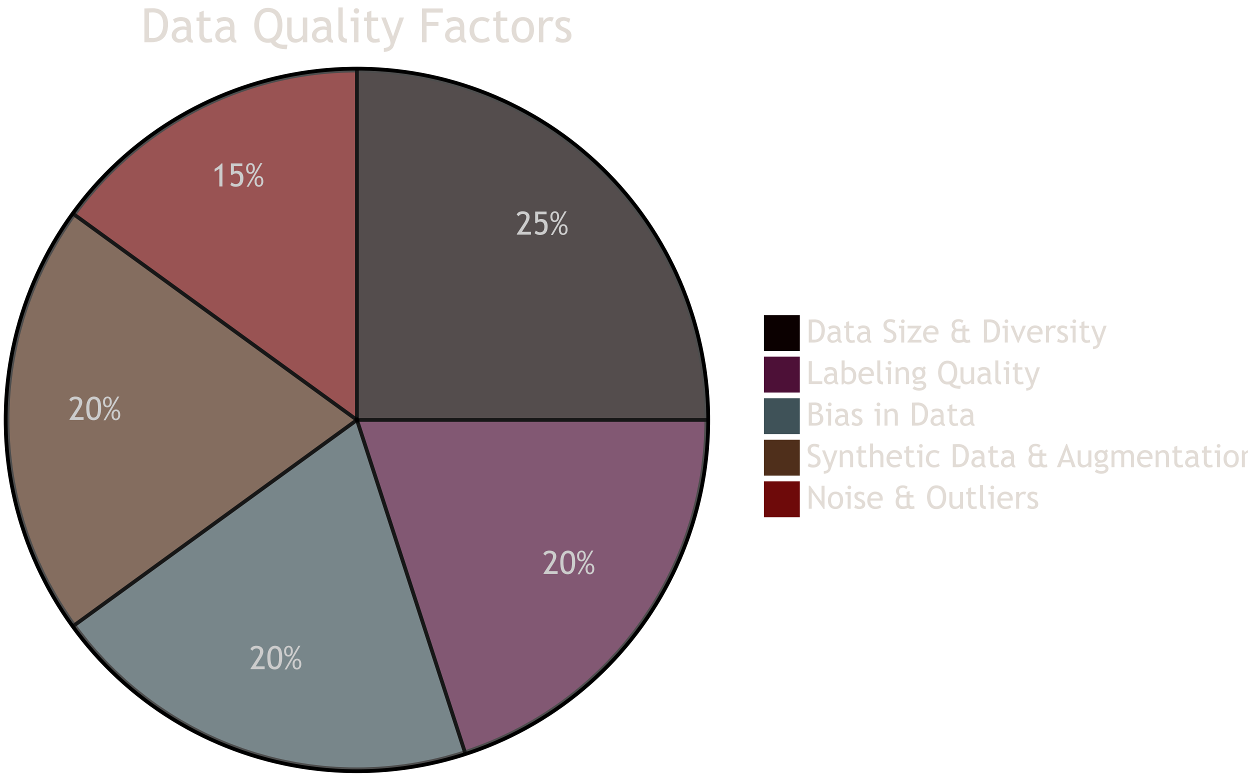


> **Key Takeaway:** Neural networks **learn from data**, and the quality of that data directly affects their ability to generalize and make accurate predictions.

Importance of Data Quality in Neural Networks

Neural networks rely on data to learn patterns and make predictions. Poor data quality can lead to **overfitting, bias, and poor generalization** to real-world scenarios.

Key Factors Affecting Data Quality



Data Size & Diversity: Avoids overfitting, ensures generalization.

Labeling Quality: Inaccurate labels introduce noise and reduce model accuracy.

Bias in Data: Reinforces discrimination in AI models.

Noise & Outliers: Leads to unstable model predictions.

Synthetic Data & Augmentation: Expands datasets to improve learning.

1. The instructor will introduce the Python script, explaining its key components:
    - Generating synthetic datasets with different data quality issues.
    - Training a neural network using TensorFlow/Keras.
    - Evaluating how accuracy changes based on dataset quality.
2. As the instructor runs the script, be sure to pay close attention to the following:
    - How noise, imbalance, and missing labels affect model accuracy.
    - Why data quality is as important as model architecture.
3. Be prepared to discuss the reflection questions with the class once the walkthrough is complete!

Python Code:

Copy

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split

# Generate synthetic dataset
def generate_data(size, noise_level=0, missing_labels=False):
    X = np.random.rand(size, 10)
    y = (X.sum(axis=1) > 5).astype(int)

    # Add noise
    if noise_level > 0:
        X += np.random.normal(0, noise_level, X.shape)

    # Remove labels
    if missing_labels:
        y[:int(size * 0.2)] = -1

    return X, y

# Create datasets
X1, y1 = generate_data(5000)
```

```
X2, y2 = generate_data(5000, noise_level=0.5)
X3, y3 = generate_data(500, missing_labels=True)

# Train a simple model on each dataset
def train_model(X, y):
    # Filter out samples with negative labels first
    mask = y >= 0
    X_filtered = X[mask]
    y_filtered = y[mask]

    # Now split the filtered data
    X_train, X_test, y_train, y_test = train_test_split(X_filtered, y_filtered, t

    model = keras.Sequential([
        keras.layers.Dense(32, activation='relu', input_shape=(10,)),
        keras.layers.Dense(16, activation='relu'),
        keras.layers.Dense(1, activation='sigmoid')
    ])
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accurac
    model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=0)
    loss, acc = model.evaluate(X_test, y_test, verbose=0)
    return acc

# Evaluate models
print("Dataset 1 Accuracy:", train_model(X1, y1))
print("Dataset 2 Accuracy:", train_model(X2, y2))
print("Dataset 3 Accuracy:", train_model(X3, y3))
```

Reflection Questions:

- How did noise and missing labels affect model accuracy?
- Which dataset produced the best-performing model, and why?
- What techniques could improve the lower-performing models?
- How does this reinforce the importance of data quality in AI development?

Summary & Key Takeaways

- Neural networks mimic the brain's structure and learn from **layered computations.**

- **Data quality is as important as model architecture**—bias, noise, and poor labels degrade performance.
- **AI models are only as good as the data they are trained on**—garbage in, garbage out.
- Data augmentation and synthetic data can **enhance training** when used correctly.

< Previous

Next >