Chapter 8: Evaluation

Introduction

In the field of Artificial Intelligence (AI), building a model is only part of the journey. Evaluating how well the model performs is crucial to ensure that it makes accurate and reliable predictions. **Evaluation** helps in determining the effectiveness of an AI model in real-world applications. This chapter will introduce the key **evaluation techniques**, **performance metrics**, and **tools** used to test AI models, ensuring they meet the expectations set during training and validation.

8.1 What is Evaluation in AI?

Evaluation in AI is the process of testing the trained model to check its accuracy and performance. The goal is to measure how well the AI system performs on unseen data (called the test set). Evaluation helps in:

- Validating the effectiveness of the model
- Avoiding underfitting and overfitting
- Selecting the best-performing model
- Fine-tuning for better results

Example:

Suppose you trained an AI model to recognize handwritten digits. Evaluation will tell how accurately it identifies new digits it hasn't seen before.

8.2 Need for Evaluation

AI models can behave differently when exposed to new data. Evaluation helps ensure:

- **Correctness**: Does the model predict accurately?
- **Robustness**: Can it handle real-world inputs?
- **Generalization**: Is it good only on training data or on new data too?

Without evaluation, you risk deploying a faulty or biased model.

8.3 Types of Datasets Used in Evaluation

1. **Training Set**

- Used to train the model.
- The model learns patterns from this data.

2. **Validation Set**
   - Used during training to tune the model parameters.
   - Helps avoid overfitting.
3. **Test Set**
   - Used after training to evaluate the final performance.
   - Never used during training.

8.4 Performance Metrics in AI

Here are key metrics used to evaluate AI models:

8.4.1 Accuracy

- Measures the percentage of correct predictions.
- Formula:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \times 100$$

Example:

If out of 100 test images, 85 were classified correctly:

$$\text{Accuracy} = \frac{85}{100} = 85\%$$

8.4.2 Precision

- Measures how many of the predicted positives are actually correct.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

8.4.3 Recall (Sensitivity)

- Measures how many actual positives the model correctly predicted.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

8.4.4 F1 Score

- Harmonic mean of precision and recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

F1 score is useful when there is class imbalance.

8.5 Confusion Matrix

A **Confusion Matrix** is a 2x2 table that helps visualize the performance of a classification model.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

This table helps calculate accuracy, precision, recall, etc.

8.6 Overfitting vs Underfitting

Overfitting

- The model performs well on training data but poorly on test data.
- Learns noise and unnecessary details.

Underfitting

- The model performs poorly on both training and test data.
- Fails to learn the patterns.

Goal: Build a model that generalizes well to new data.

8.7 Cross-Validation

Cross-validation is a method used to test the model multiple times on different subsets of the data to ensure consistent performance.

- **K-Fold Cross-Validation**: The data is divided into K parts, and the model is trained and tested K times.
- This helps reduce the variance and gives a more reliable performance estimate.

8.8 Tools for Evaluation

Common tools used for evaluating models include:

- **Scikit-learn**: Offers functions like `accuracy_score`, `confusion_matrix`, etc.
- **TensorFlow/Keras**: Built-in methods to evaluate deep learning models.

- **Google Teachable Machine**: For visual AI models in schools and simple projects.

8.9 Real-World Example: Spam Detection

Let's say you trained an AI model to detect spam emails. After training:

- You feed it 1,000 new emails.
- 800 are non-spam (ham), and 200 are spam.
- Model correctly identifies 180 spam emails (TP) but wrongly labels 20 ham emails as spam (FP).
- It misses 20 spam emails (FN).

From this data, you can compute:

- **Accuracy**, **Precision**, **Recall**, **F1 Score** using formulas.
- Evaluate if your model is reliable or needs improvement.

Summary

- **Evaluation** is a vital step in the AI model development process.
- It helps test the **performance**, **accuracy**, and **reliability** of the model.
- Key metrics: **Accuracy, Precision, Recall, F1 Score**
- Tools like **Confusion Matrix** and **Cross-validation** ensure deeper insights.
- Avoid **overfitting** and **underfitting** for a balanced model.
- Always evaluate on **unseen data** (test set) for a realistic measure of performance.
