An Overview of Machine Learning Models

Engr. Dr. Muhammad Farooq-i-Azam

Department of Computer Engineering
COMSATS University Islamabad, Lahore Campus

fazam@cuilahore.edu.pk

Introduction

- Machine Learning (ML) enables computers to learn patterns from data and make decisions or predictions.
- ML models are integral to applications like predictive text, ride-sharing ETAs, and content recommendations.
- This presentation explores key ML model categories, their mechanisms, and applications.

Supervised Learning

- Involves training models on labeled datasets, where each input is paired with a known output.
- The model learns to predict the output for new, unseen inputs.
- Common algorithms:
  - Linear Regression
  - Logistic Regression
  - Decision Trees
  - Support Vector Machines (SVMs)
  - Neural Networks
- Applications:
  - Email spam detection
  - Image recognition
  - Predictive analytics

Unsupervised Learning

- Deals with unlabeled data; the model seeks to identify inherent patterns or groupings.
- Types:
  - **Clustering:** This finds the natural groupings for all data.
  - **Association:** The dependencies or interesting relationships between various data are determined.
  - **Dimensionality Reduction:** Dimensions of data are reduced by finding the intrinsic components that represent certain data.
- Common algorithms:
  - K-Means Clustering
  - Hierarchical Clustering
  - Principal Component Analysis (PCA)
  - Autoencoders
- Applications:
  - Customer segmentation
  - Anomaly detection
  - Data compression

Semi-Supervised Learning

- Combines supervised and unsupervised learning by using a small amount of labeled data alongside a larger set of unlabeled data.
- The model leverages the labeled data to guide the learning process and improve accuracy.
- Applications:
  - Web content classification
  - Speech recognition
  - Protein sequence classification

Self-Supervised Learning

- A subset of unsupervised learning where the data itself provides the supervision.
- The model generates pseudo-labels from the input data and learns to predict these labels.
- Commonly used in:
  - Natural Language Processing (e.g., word embeddings)
  - Computer Vision (e.g., image colorization)
- Bridges the gap between unsupervised and supervised learning by creating supervisory signals from the data.

Reinforcement Learning

- Models learn by interacting with an environment, receiving rewards or penalties for actions taken.
- The goal is to develop a policy that maximizes cumulative rewards.
- Key components:
  - Agent: The learner or decision-maker.
  - Environment: Everything the agent interacts with.
  - Actions: Choices the agent can make.
  - Rewards: Feedback from the environment based on actions.
- Applications:
  - Game AI (e.g., AlphaGo)
  - Robotics control
  - Personalized recommendations

Q-Learning

- **Type:** Model-free, off-policy algorithm.
- **Goal:** Learn the optimal action-selection policy.
- **Method:**
  - Learns Q-values: estimates of the total future rewards for taking an action in a given state.
  - Updates Q-values using the Bellman equation.
  - Chooses actions that maximize the Q-value, regardless of the current policy.
- **Advantages:**
  - Converges to optimal policy under certain conditions.
  - Simple to implement and widely used.
- **Use Case:** Grid world pathfinding, decision-making tasks.

SARSA (State-Action-Reward-State-Action)

- **Type:** Model-free, on-policy algorithm.
- **Goal:** Learn Q-values by following the current policy.
- **Method:**
  - Updates Q-values using the actual action taken under the current policy.
  - Update rule: $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s', a') - Q(s,a)]$
  - Accounts for the policy's behavior during learning.
- **Differences from Q-Learning:**
  - Q-Learning is off-policy (targets optimal action), SARSA is on-policy (uses actual action).
  - SARSA can be more cautious and stable in noisy environments.
- **Use Case:** Situations where risk-aware learning is important.

Temporal Difference (TD) Learning

- **Type:** Model-free prediction method.
- **Goal:** Estimate value functions by learning from incomplete episodes.
- **Method:**
  - Uses the difference (TD error) between predicted and actual rewards over time steps.
  - TD Update: $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
  - Balances the benefits of Monte Carlo methods and dynamic programming.
- **Advantages:**
  - Can learn online and incrementally.
  - Doesn't require waiting for final outcomes like Monte Carlo.
- **Use Case:** Policy evaluation in RL tasks.

Deep Q-Network (DQN)

- **Type:** Deep reinforcement learning algorithm.
- **Goal:** Extend Q-learning to handle high-dimensional state spaces.
- **Method:**
  - Uses a deep neural network to approximate the Q-function.
  - Inputs a state; outputs Q-values for each possible action.
  - Trained using experience replay and fixed target networks to improve stability.
- **Advantages:**
  - Works well on image and complex input data.
  - Capable of human-level performance on Atari games.
- **Use Case:** Game AI, robotics, autonomous systems.

Transfer Learning: Overview

- **Definition:** Transfer learning is a technique where knowledge from one task is reused to improve learning on a different, but related, task.
- **Why it Matters:**
  - Training deep learning models from scratch requires large datasets and computational resources.
  - Transfer learning allows leveraging pre-trained models to save time and improve performance.
- **Key Idea:** Use a model trained on a source task to boost performance on a target task.

Transfer Learning

- **Typical Workflow:**
  1. **Pretraining:** Train a model on a large, general-purpose dataset (e.g., ImageNet, Wikipedia).
  2. **Feature Reuse:** Keep the earlier layers (features) of the pretrained model.
  3. **Fine-Tuning:** Replace and retrain the final layers on the target task dataset.
- **Layer Strategy:**
  - Freeze low-level layers (generic features).
  - Retrain high-level layers (task-specific features).

Transfer Learning

- **Applications:**
  - **Computer Vision:** Using pretrained CNNs for tasks like facial recognition or medical imaging.
  - **Natural Language Processing:** Adapting models like BERT, GPT for tasks like sentiment analysis, chatbots, summarization.
- **Benefits:**
  - Reduces training time and computational cost.
  - Requires less labeled data for the new task.
  - Often improves performance on small or domain-specific datasets.

Deep Learning Architectures

- Deep Learning involves neural networks with multiple layers (deep neural networks) that can model complex patterns in data.
- Common architectures:
  - Convolutional Neural Networks (CNNs): Specialized for processing grid-like data such as images.
  - Recurrent Neural Networks (RNNs): Designed for sequential data, capturing temporal dependencies.
  - Transformers: Utilize self-attention mechanisms, excelling in tasks like language modeling.
- Applications:
  - Image and speech recognition
  - Natural language processing
  - Autonomous vehicles

Ensemble Methods

- Combine predictions from multiple models to improve accuracy and robustness.
- Common techniques:
  - Bagging: Builds multiple independent models and averages their predictions (e.g., Random Forests).
  - Boosting: Builds models sequentially, each correcting errors of the previous one (e.g., Gradient Boosting Machines).
- Applications:
  - Competition-winning solutions in machine learning contests
  - Risk assessment in finance
  - Medical diagnosis

Conclusion

- Understanding various ML models is crucial for selecting the appropriate approach for a given problem.
- While foundational models like supervised and unsupervised learning are widely known, emerging techniques like self-supervised learning are gaining prominence.
- Advanced architectures and ensemble methods further enhance the capabilities of ML systems.
- Continuous learning and adaptation are key in the evolving field of machine learning.

References

- Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer
- Will Jennings, ML Models: Understanding the Fundamentals, Gretel
- Stephen DeAngelis, Algorithms: Machine Learning's Secret Sauce, Enterra Solutions