

Machine (Deep) Learning

Course:
INFO-6145 Data Science and Machine Learning



Revised by:
Mohammad Noorchenarboo

October 31, 2024

Contents

- 1 Machine Learning Paradigms in "The Master Algorithm"
 - Symbolists
 - Evolutionaries
 - Bayesians
 - Analogizers
 - Connectionists
- 2 Data Transformation
 - Importance of Data Transformation in Machine Learning
 - Challenges in Manual Data Transformation
 - Automating Data Transformation in Machine Learning
- 3 Deep Learning
 - Introduction to Deep Learning
 - How Deep Learning Works
 - Role of Weights and Backpropagation
 - Metaphor: "Uncrumpling" Data

Current Section

1 Machine Learning Paradigms in "The Master Algorithm"

- Symbolists
- Evolutionaries
- Bayesians
- Analogizers
- Connectionists

2 Data Transformation

- Importance of Data Transformation in Machine Learning
- Challenges in Manual Data Transformation
- Automating Data Transformation in Machine Learning

3 Deep Learning

- Introduction to Deep Learning
- How Deep Learning Works
- Role of Weights and Backpropagation
- Metaphor: "Uncrumpling" Data

Introduction to the Five Tribes of Machine Learning

Pedro Domingos categorizes machine learning researchers into five "tribes," each representing a unique approach to creating algorithms. These are:

- Symbolists: Learning through logic and rule-based systems.
- Evolutionaries: Learning through natural selection and optimization.
- Bayesians: Learning through probability and inference.
- Analogizers: Learning through similarity and case-based reasoning.
- Connectionists: Learning through neural networks and pattern recognition.

We will explore each tribe's theory, methods, and limitations.

Symbolists: Logic-Based Learning

Symbolists believe that intelligence is based on **logical reasoning** and knowledge is represented as symbols. They use **inductive logic programming** to create rules and relationships.

Primary Focus

The goal is to discover general rules from specific instances. Symbolists use deductive reasoning to make predictions or classify new cases.

Key Techniques

Algorithms like decision trees, expert systems, and rule-based systems represent Symbolists' approach.

Symbolists: Logic-Based Learning

Limitations

- Symbolist algorithms struggle with uncertainty and are not robust against noisy data.
- They require a significant amount of human input for defining rules and interpreting data.

Example: Diagnosing Diseases

An expert system for medical diagnosis that uses specific symptoms to derive logical conclusions about potential diseases.

Evolutionaries: Learning through Optimization

Evolutionaries model learning as a process of **optimization**, where algorithms evolve solutions over generations by selecting the fittest candidates.

"Survival of the fittest"

Primary Focus

The main idea is to use evolutionary strategies, inspired by natural selection, to "breed" better solutions over time.

Key Techniques

Techniques include genetic algorithms, genetic programming, and swarm intelligence. These methods iterate and improve upon solutions, mimicking biological evolution.

Evolutionaries: Learning through Optimization

Limitations

- Evolutionary algorithms are computationally expensive.
- Convergence to an optimal solution may be slow and can get stuck in local optima.

Example: Evolving Traffic Patterns

An evolutionary algorithm could optimize traffic light timings by simulating various scenarios to minimize congestion.

Bayesians: Learning through Probabilistic Inference

Bayesians approach learning as **probabilistic inference**. They use data to update beliefs, with a foundation in Bayes' theorem for calculating posterior probabilities.

Primary Focus

Bayesian methods aim to model uncertainty by assigning probabilities to events or outcomes and updating these as new data arrives.

Key Techniques

Bayesian networks, Markov models, and probabilistic programming represent Bayesian methods. These models quantify uncertainty and are used in areas requiring high accuracy.

Bayesians: Learning through Probabilistic Inference

Limitations

- Bayesian methods can be computationally intense, especially with complex models or large data sets.
- They require careful selection of prior distributions, which can impact outcomes.

Example: Weather Forecasting

Bayesian models in weather forecasting update predictions as new information (temperature, pressure) becomes available, adjusting probabilities of rain or sunshine.

Analogizers: Learning through Case-Based Reasoning

Analogizers emphasize learning by **drawing parallels between cases**, using similarity-based algorithms to solve problems.

Primary Focus

Analogizers find connections between similar cases and use these to generalize or solve new problems. They view learning as matching patterns.

Key Techniques

Algorithms such as k-nearest neighbors (KNN), support vector machines (SVMs), and kernel methods are often used by Analogizers. These methods focus on measuring similarity between data points.

Analogizers: Learning through Case-Based Reasoning

Limitations

- Memory-intensive, as they store large datasets for case comparison.
- Struggle with high-dimensional data and can be sensitive to irrelevant features.

Example: Product Recommendation Systems

In a product recommendation system, Analogizers recommend items to users based on similarities to other users with similar preferences.

Connectionists: Learning through Neural Networks

Connectionists focus on learning by **mimicking the brain**, using artificial neural networks to recognize patterns through multiple layers of processing.

Primary Focus

Connectionist algorithms learn through adjusting connections (weights) between "neurons" in a network to improve predictions or classifications.

Key Techniques

Common methods include deep learning, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), enabling applications in image recognition and language processing.

Connectionists: Learning through Neural Networks

Limitations

- Neural networks require large datasets and computational resources to train effectively.
- They are often considered "black boxes" due to the lack of interpretability in decision-making.

Example: Image Classification

A CNN trained on labeled images can identify objects within photos, such as animals, buildings, or vehicles.

Current Section

1 Machine Learning Paradigms in "The Master Algorithm"

- Symbolists
- Evolutionaries
- Bayesians
- Analogizers
- Connectionists

2 Data Transformation

- Importance of Data Transformation in Machine Learning
- Challenges in Manual Data Transformation
- Automating Data Transformation in Machine Learning

3 Deep Learning

- Introduction to Deep Learning
- How Deep Learning Works
- Role of Weights and Backpropagation
- Metaphor: "Uncrumpling" Data

Data Transformation: The Core Challenge in Machine Learning

The central problem in machine learning and deep learning is to **transform data meaningfully**. This involves creating representations of the data that help models get closer to the desired output.

Objective of Data Transformation

To find useful representations of input data that make the learning process more effective, enabling models to understand patterns and relationships.

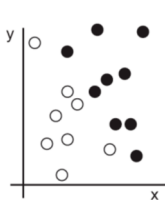
Data Transformation: The Core Challenge in Machine Learning

Example: Image Encoding

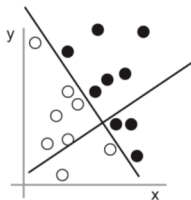
A color image can be encoded in different formats:

- RGB format: **Red-Green-Blue** values represent color intensity.
- HSV format: **Hue-Saturation-Value** captures color in a perceptually meaningful way.

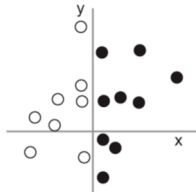
1: Raw data



2: Coordinate change



3: Better representation



Transformations for Suitable Representations

Machine learning models need to find representations that capture essential patterns in data. Proper transformations can make data more suitable for the specific task at hand.

Transformations Help To

- Simplify data by highlighting relevant features.
- Improve model performance by aligning data closer to the expected output.
- Enable models to focus on informative aspects, reducing noise.

Challenge

Without appropriate transformations, models may fail to capture critical patterns, leading to poor performance.

Examples of Data Transformation

There are many ways to transform data, each aiming to make the data more useful for a model. For example:

- **Normalizing** data to a standard range.
- **Encoding categorical variables** as numerical values.
- **Creating new coordinate systems** to better separate data points by class.

Visual Data Transformation

By transforming data points visually, it is possible to separate classes effectively:

- Plotting data in a 2D or 3D space to reveal natural groupings.
- Applying transformations that highlight differences between classes.

The Difficulty of Manually Transforming Data

Transforming data manually can be straightforward for simple tasks but becomes challenging for complex problems, such as:

- Classifying handwritten digits where different handwriting styles add variability.
- Identifying objects in images where lighting, angles, and backgrounds vary significantly.

Challenge

Manually designing transformations for high-dimensional data, like images or text, is labor-intensive and requires domain expertise.

Automating Data Transformation

The goal of machine learning is to **automate the process of transforming data** to create representations that maximize model performance.

Machine Learning's Role

Machine learning models learn to identify and apply transformations to data, discovering useful representations without manual intervention.

Example: Deep Learning for Image Recognition

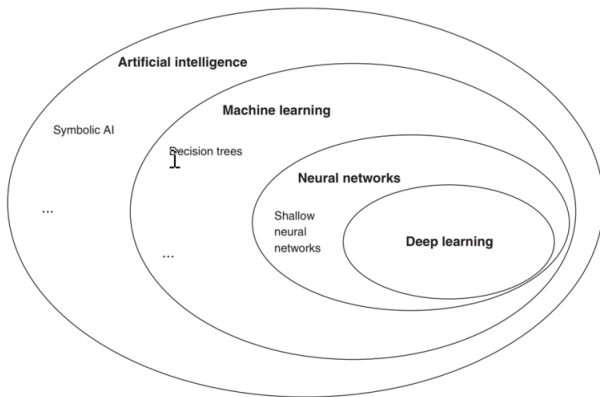
In deep learning, convolutional layers automatically learn features like edges, shapes, and textures, which are helpful for object recognition.

Current Section

- 1 Machine Learning Paradigms in "The Master Algorithm"
 - Symbolists
 - Evolutionaries
 - Bayesians
 - Analogizers
 - Connectionists
- 2 Data Transformation
 - Importance of Data Transformation in Machine Learning
 - Challenges in Manual Data Transformation
 - Automating Data Transformation in Machine Learning
- 3 Deep Learning
 - Introduction to Deep Learning
 - How Deep Learning Works
 - Role of Weights and Backpropagation
 - Metaphor: "Uncrumpling" Data

Deep Learning: A Subfield of Machine Learning

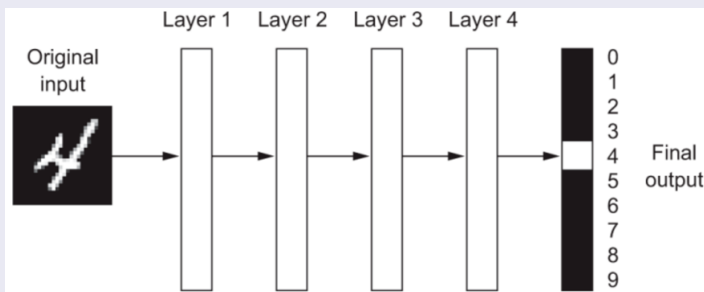
Deep Learning is a specific subfield of machine learning that focuses on learning successive layers of representations.



Deep Learning: A Subfield of Machine Learning

Definition

Deep learning models are structured with multiple layers, each layer learning a progressively more meaningful representation of the input data.

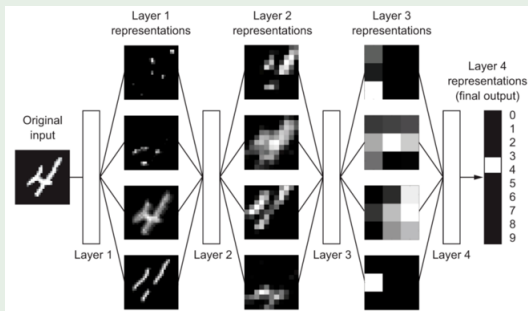


Deep Learning: A Subfield of Machine Learning

Layered Learning

Each layer captures features at increasing levels of abstraction:

- Early layers may capture edges in an image.
- Middle layers capture shapes.
- Final layers capture complex patterns, like objects or scenes.



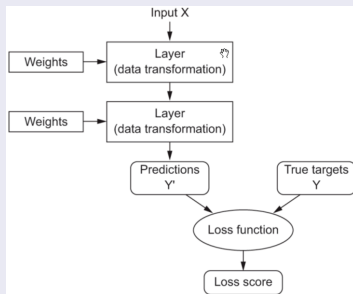
Understanding the Deep Learning Process

Deep learning models perform input-to-target mapping by progressively transforming data through layers.

Understanding the Deep Learning Process

Core Process

Each layer applies transformations to the data using **weights**, or layer parameters, which are adjusted based on feedback from a **loss function**.

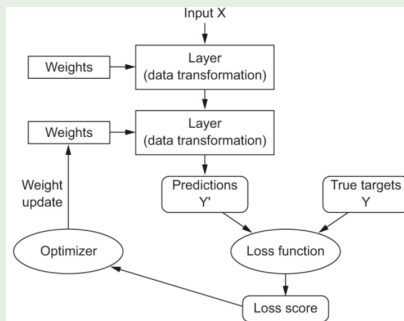


Understanding the Deep Learning Process

Feedback Loop

The training process is a continuous loop:

- 1 Calculate output predictions based on current weights.
- 2 Measure error (loss) between predictions and actual target values.
- 3 Adjust weights to reduce error using **backpropagation**.



Weights and Backpropagation

Weights are parameters that define the strength of connections between neurons. They are refined through **backpropagation**, a process that propagates error backward to update weights.

Backpropagation

- Calculates gradients of the loss function with respect to weights.
- Adjusts weights to minimize loss, making the model predictions closer to the target.

Challenge

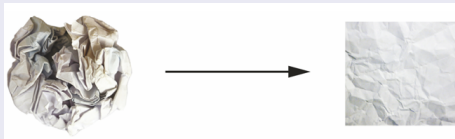
Training deep networks can be computationally intensive and requires large amounts of data to avoid overfitting.

Visualizing Deep Learning: The "Uncrumpling" Process

One way to understand deep learning is through the metaphor of **uncrumpling paper**:

The Metaphor

- Imagine classes of data points as **crumpled colored paper** representing mixed categories.
- Deep learning "uncrumples" the paper, layer by layer, to separate and identify these classes.



Visualizing Deep Learning: The "Uncrumpling" Process

Example

In image recognition, deep learning "uncrumples" the data by adjusting layers until patterns representing specific objects, like cats and dogs, are clearly separated.

More examples:

- [cs.stanford](https://cs.stanford.edu/)
- [poloclub.github](https://poloclub.github.io/)

Conclusion: Why Deep Learning Matters

Deep learning's layered structure allows it to automatically learn and extract complex patterns from data, making it highly effective for tasks like image and speech recognition.

Key Insight

By automating the process of feature extraction and learning meaningful representations, deep learning transforms how we approach complex data problems.