SUDO CODE - Week 5: Core Concepts in Machine Learning

Doan Ngoc Mai

September 17, 2025

Abstract

This survey explores five core concepts in Machine Learning (ML): hidden layers, perceptron, memory-based learning, gradient descent, and loss function. Each section provides step-by-step definitions with simplified explanations that avoid unexplained jargon, followed by properties, connections to other ML concepts, and real-world applications. By bridging intuitive metaphors and technical understanding, this report aims to make these ideas accessible to beginners while still maintaining academic depth.

1 Hidden Layer

Definition

- A **layer** is a group of simple units (like boxes) that receive numbers, perform calculations, and pass results forward.
- The **hidden layer** sits between the input (raw data) and the output (final prediction).
- It is called "hidden" because we cannot directly observe its internal results; they are only used inside the model.
- Step-by-step:
  - Data (e.g., pixels of an image or words in a sentence) enters the first layer.
  - The hidden layer transforms this data into more useful features.
  - Example in image recognition:
    * The first hidden layer may detect simple edges.
    * The second may detect shapes (circles, corners).
    * Later layers combine these features to recognize full objects (cats, cars).

Properties and Relations

- Hidden layers enable **deep learning**. Without them, models can only solve simple problems.
- A hidden layer is composed of many **perceptrons**.
- Their weights are trained using **gradient descent**, guided by the **loss function**.
- Memory-based learning typically does not use hidden layers, making the two approaches contrasting paradigms.

Applications

- Image classification (abstracting features step by step).
- Natural language processing tasks such as translation or sentiment analysis.
- Speech recognition systems (detecting phonemes and patterns).

2 Perceptron

Definition

- A **perceptron** is the simplest kind of "neuron" in ML.
- It takes inputs, multiplies them by weights (importance values), sums them, and compares the result to a threshold.
- If the sum exceeds the threshold, it outputs 1 (yes); otherwise, 0 (no).
- Example: deciding whether to bring an umbrella:
  - Input 1: Weather forecast (rain = 1, no rain = 0).
  - Input 2: Looking at the sky (cloudy = 1, clear = 0).
  - Each input has a weight (how much you trust it).
  - If the weighted sum is high enough, the perceptron outputs "Yes, bring umbrella".

Properties and Relations

- Perceptrons are the **building blocks** of hidden layers.
- A single perceptron can only solve simple linear problems.
- Multiple perceptrons stacked together form neural networks.
- Weights are trained with **gradient descent** guided by the **loss function**.

Applications

- Early spam filters (spam vs. not spam).
- Simple robotics or control tasks.
- Teaching tool for neural network basics.

3 Memory-Based Learning

Definition

- "Memory" here means storing previously seen examples.
- Memory-based learning solves new problems by comparing them with stored examples.
- Example: to classify a fruit, the model finds the most similar stored fruits. If most were apples, it predicts apple.

Properties and Relations

- Does not involve complex hidden layers or weight updates.
- Instead of learning rules, it memorizes and reuses training data.
- Closely related to **nearest neighbor methods**.
- Different from perceptron-based models, which compress data into weights.

Applications

- Recommendation systems (finding similar users/items).
- Early translation systems using phrase tables.
- Medical diagnosis (comparing a patient's data to past cases).

4 Gradient Descent

Definition

- "Gradient" = slope of a curve (how steep a hill is).
- "Descent" = moving downward.
- Gradient descent is like finding the bottom of a valley by walking downhill step by step.
- In ML, the curve represents the error (loss), and the lowest point means the best model.

Properties and Relations

- Updates perceptron and hidden layer weights to minimize loss.
- Requires the **loss function** to determine the direction of descent.
- Variants include:
  - Batch gradient descent (all data each step).
  - Stochastic gradient descent (SGD: one sample at a time).
  - Mini-batch gradient descent (a balance between the two).

Applications

- Training all modern neural networks (CNNs, RNNs, Transformers).
- Optimization problems in economics, physics, and engineering.

5 Loss Function

Definition

- "Loss" = measure of how wrong a model's prediction is.
- A loss function calculates this error.
- Example: if the true answer is 10 but the model predicts 8 → loss = 2; if it predicts 3 → loss = 7.

Properties and Relations

- Guides the entire learning process: smaller loss = better model.
- Provides the feedback gradient descent uses to adjust weights.
- Different tasks use different loss functions:
  - Classification → cross-entropy loss.
  - Regression → mean squared error.

Applications

- Image recognition (cross-entropy).
- Predicting house prices (MSE).
- Reinforcement learning (policy and value loss).

6 References

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Available at https://www.deeplearningbook.org/.
- TensorFlow Tutorials. TensorFlow. Available at http://tensorflow.org/tutorials.
- 3Blue1Brown. (2017). *Deep Learning*. [YouTube Video]. https://www.youtube.com/watch?v=aircAruvnKk.
- Welch Labs. (2015). *Neural Networks Demystified*. [YouTube Video]. http://youtube.com/watch?v=IHZwWFHWa-w.
- Siraj Raval. (2017). *The Math of Intelligence*. [YouTube Video]. http://youtube.com/watch?v=OFbnpY_k7js.
- StatQuest. (2018). *Gradient Descent, Step-by-Step*. [YouTube Video]. https://www.youtube.com/watch?v=QBbC3Cjsnjg.
- Messahli, Y. (2021). *Instance-Based vs Model-Based Learning Theory*. Kaggle Notebook. Available at https://www.kaggle.com/code/yassermessahli/instance-based-vs-model-based-learning-theory.