Applications of Data Analysis and Machine Learning

Instructors: Wang Wenguan & Pradeep Ravikumar

Abstract

This document serves as a comprehensive set of course notes derived from the lectures of Professors Wang Wenguan and Pradeep Ravikumar. It covers the history and fundamentals of Artificial Intelligence, Deep Learning architectures (CNN, RNN, Transformers), and the statistical foundations of Data Analysis (MLE, Bayesian Inference, Hypothesis Testing). Key concepts are highlighted in red.

Contents

Part I

Machine Learning and Deep Learning

Based on the lectures of Prof. Wang Wenguan

1 Introduction to AI (Week 1-3)

1.1 History and Paradigms

The "Year of AI" is widely considered to be 1956. Historically, there have been two main approaches:

- **Symbolism**: Logic-based approaches.
- **Connectionism**: Network-based approaches (the foundation of modern Deep Learning).
- **Machine Language**: It's the intersection of Symbolism and Connectionism.

1.2 Neural Network Fundamentals

A neural network is essentially a method to define a set of functions and pick the best one.

- **Machine Learning**: Moving from random guesses to learned parameters.
- **Loss Function (Goodness of Function)**: We need a metric to measure "goodness" to pick the best function (often referred to as minimizing the Loss).

1.3 Activation Functions

Standard Linear Regression cannot solve all problems, such as XOR problems. To introduce non-linearity, we use Activation Functions.

The general form of a neuron activation is:

$$a = \sigma(a_1 w_1 + \cdots + a_k w_k + b)$$

Common activation functions:

1. **Sigmoid**: Compresses output between 0 and 1.
2. **Tanh**: Compresses output between -1 and 1.
3. **ReLU (Rectified Linear Unit)**:

$$f(x) = \max(0, x)$$

   ReLU is generally good for most tasks.

Architecture Note: In network design, Deep is better than fat (wide).

2 Training and Optimization (Week 4)

2.1 Task Types

- **Regression**: Predicting a value Y using a set of predictors X (e.g., predicting tomorrow's temperature: 27°C).
- **Classification**: Identifying which category an input belongs to (e.g., predicting if tomorrow is "Hot").

2.2 Gradient Descent

To create a "Good Function Maker," we must make the Loss as small as possible.

1. Pick an initial value (often random).
2. Calculate the differential (gradient).
3. Decrease or increase parameters based on the result to minimize error.

Note: The final result may vary depending on the starting point.

3 Convolutional Neural Networks (Week 6)

CNNs are designed for processing grid data like images.

- **Padding**: Adding border pixels (usually size 1) allows the convolution to capture more information, especially at the edges of the image.
- **Filter/Kernel**: The core component that slides over the input to detect features. One filter corresponds to one output.
- **Pooling**: A subsampling technique. It reduces the image size and calculation load without changing the object semantics.

4 Natural Language Processing (Week 9)

4.1 Word Representation

- **One-hot Vector**: Sparse representation. Only one element is 1, others are 0. Each vector is unique and has the same dimension.
- **Distributional Representation**: Rich in semantic info. The meaning of a word is defined by other contexts, which helps computers understand natural languages.

4.2 Word Embeddings

Embeddings build upon the Distributional Hypothesis.

- **Count-based**: Based on frequency of word co-occurrence.
- **Inference-based**: Based on the words on both sides.

  - **CBOW (Continuous Bag-of-Words)**: Predicts one word based on surrounding context words.
  - **Skip-gram**: Predicts multiple context words based on one input word. Skip-gram is harder and slower to train, but produces better distributional representations.

5 Sequence Models (Week 10)

5.1 RNN Limitations

Standard Recurrent Neural Networks (RNNs) have a major flaw: they cannot handle long-term dependencies and are hard to train.

5.2 LSTM (Long Short-Term Memory)

LSTMs introduce a "Gate" mechanism to control the flow of information and memory.

- **Gates**: Control the flow of information (three gates to update and control memory).
- **Memory Cell**: Runs through the top of the diagram, maintaining long-term state.

The Steps of LSTM:

1. Decide what to throw away from memory (Forget Gate).
2. Decide what to store in memory (Input Gate).
3. Update the memory cell.
4. Decide the output (Output Gate).

This solves the long-term dependency problem.

6 Advanced Deep Learning (Week 12)

6.1 Attention and Transformers

- **Attention Mechanism**: Allows the model to focus on relevant info even with long-term dependencies.
- **Seq2Seq**: Solves the long-term dependency issue effectively. Used in Transformer models (Encoder-Decoder architecture).
- **Models**: BERT, GPT.

6.2 LLM Training Pipeline

The modern pipeline for Large Language Models (LLMs):

1. Find good model parameters with massive datasets.
2. Update with training data for downstream tasks.
3. Align with human values.

Part II

Foundations of Data Analysis

Based on the lectures of Prof. Pradeep Ravikumar

7 The Data Science Landscape (Lect 1)

Data Analysis operates on three axes: **Data**, **Tasks**, and **Algorithms**. The fundamental flow of information is:

$$\text{Data} \xrightarrow[\text{Learning}]{\text{Model}} \text{Model} \xrightarrow[\text{Inference}]{\text{Model}} \text{Knowledge}$$

.

7.1 Task Types

- **Prediction (Supervised)**: Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$.
- **Unsupervised Learning**: Given $X \in \mathcal{X}$, learn $f(x)$.

The goal is to devise a model, draw independent samples that are identically distributed (i.i.d.), and quantify the results.

8 Maximum Likelihood Estimation (MLE) (Lect 2-3)

8.1 The Framework

1. Assume a model that captures the data distribution.
2. Choose parameters θ that maximize the probability of the observed data.

8.2 Likelihood Function

The likelihood function is defined as the product of individual probabilities:

$$P_X(X_1, \ldots, X_n; \theta) = \prod P_{X_i}(X_i; \theta)$$

In log form: $\log P_X(X; \theta) = \sum \log P_{X_i}(X_i; \theta)$. Maximizing $f(\theta)$ is equivalent to maximizing $\log f(\theta)$.

The **Maximum Likelihood Estimator (MLE)** is:

$$\hat{\theta}_{\text{MLE}} = \arg\max P(X|\theta)$$

8.3 Bias

- **Unbiased Estimator**: $E[\hat{\theta}] = \theta$.
- **Unbiasedness** is a desirable property.

The expectation of the estimator $\hat{p}$ for a Bernoulli trial is $E(\hat{p}) = p$, the "true" probability.

9 Bayesian Data Analysis (Lect 4)

9.1 Frequentist vs. Bayesian

The difference lies in whether the parameter is constant or random.

- **Frequentist (Classical)**: θ is a constant. Data is random.
- **Bayesian**: θ is a random variable.

9.2 Bayes Rule

The core formula for Bayesian inference:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

Where: $P(\theta|D)$ is the posterior, $P(D|\theta)$ is the likelihood, and $P(\theta)$ is the prior. *Motto*: "The more the merrier" regarding data.

10 Estimators and Inference (Lect 5)

10.1 Point Estimates

We want a single numerical value that represents our best guess of θ. This is the point estimate $\hat{\theta} = g(x)$.

- **MAP (Maximum A Posteriori) Estimator**:

$$\hat{\theta}(x) = \arg\max P(\theta|x)$$

- **Conditional Expectation**:

$$\hat{\theta}(x) = E(\theta|x)$$

11 Hypothesis Testing (Lect 6)

Data scientists need to help ensure the results of data analysis aren't false discoveries.

11.1 Significance Testing

1. Analyze data.
2. Compute the probability of the result R (p-value).
3. Determine if the p-value is sufficiently small or not.

11.2 Neyman-Pearson Hypothesis Testing

- Requires: (1) A statement to disprove (H_0); (2) A random variable (test statistic).
- Specifies the **alternative hypothesis** (H_1).

- Defines a critical region; if the test statistic falls in, H_0 is rejected.

Error Definitions:

- **Significance level** α: $P(R \in \text{critical region} | H_0)$ (Type I error).
- β: $P(R \notin \text{critical region} | H_1)$ (Type II error).