Comprehensive Study Guide for Advanced Machine Learning

Eshaan Arora

Contents

1 Overview and Recap

1.1 Data Science Lifecycle

- The lifecycle involves iterative steps: Problem Definition → Data Collection → Preprocessing → Exploratory Data Analysis → Model Building → Evaluation → Deployment.
- Emphasis on feedback loops for continuous model improvement.
- Maximum Likelihood (ML) Principle:

  $$\hat{\theta} = \arg\max_{\theta} L(\theta|X) = \arg\max_{\theta} \prod_{i=1}^{N} P(x_i|\theta),$$

  where $L(\theta|X)$ is the likelihood function for parameter θ given data X.
- No Free Lunch Theorem: There is no universally best model; tradeoffs must be understood.

1.2 Predictive Analytics

- **Descriptive Analytics:** Summarizes past data to find patterns and insights.
- **Predictive Analytics:** Models like regression and classification to predict unknown or future data points.
- **Prescriptive Analytics:** Suggests actions, often involving reinforcement learning.

2 Advanced Multivariate Regression

2.1 Basis Function Expansion

- Allows for non-linear relationships by transforming input features.
- Common examples:

$$\phi_1(x) = x,\ \phi_2(x) = x^2,\ \phi_3(x) = \sin(x).$$

2.2 Regularization Techniques

- **Ridge Regression:**

$$\min_w \left\{ \sum_{i=1}^{N} (y_i - w^T x_i)^2 + \lambda \|w\|_2^2 \right\}.$$

- **Lasso Regression:**

$$\min_w \left\{ \sum_{i=1}^{N} (y_i - w^T x_i)^2 + \lambda \|w\|_1 \right\}.$$

- **Elastic Net Regularization:** Combines Ridge and Lasso penalties to balance feature selection and coefficient shrinkage.

2.3 Bias-Variance Tradeoff

$$\text{Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}.$$

- High bias: Underfitting.
- High variance: Overfitting.
- Learning curves can help detect high bias (both training and validation errors are high) or high variance (training error is low, but validation error is high).

2.4 Polynomial Regression

- Fits data using polynomial terms (e.g., x, x^2, x^3) to capture non-linear relationships.

3 Neural Networks for Regression

3.1 Activation Functions

- Rectified Linear Unit (ReLU): Efficient, but prone to the dying ReLU problem.
- Leaky ReLU: Addresses dying ReLU by allowing a small slope for negative inputs.
- Sigmoid and Tanh: Useful for smooth outputs but suffer from vanishing gradients.

3.2 Stochastic Gradient Descent (SGD)

- Iterative method to minimize loss functions.
- Weight update:

  $$w_{t+1} = w_t - \eta \frac{\partial L(w_t)}{\partial w},$$

  where η is the learning rate.
- Momentum-based SGD accelerates convergence by incorporating previous updates.

3.3 Backpropagation

- Efficiently computes gradients for multi-layer networks using the chain rule.
- Modular view using Jacobian matrices for error propagation through layers.
- Steps:
  1. Forward pass to compute predictions.
  2. Compute the error at the output.
  3. Propagate the error backward to update weights.

3.4 Multi-Layer Perceptrons (MLPs)

- Universal approximators capable of modeling any continuous function with sufficient complexity.
- Use non-linear activation functions in hidden layers for expressiveness.
- Softmax function in output layers ensures probabilities sum to 1 in classification tasks.

3.5 Transfer Learning

- Involves reusing a pre-trained model on a new, related task.
- Steps:
  - Pre-training: Train on a large dataset for a general task.
  - Fine-tuning: Adjust model weights for the specific task using a smaller dataset.

3.6 Multi-task Learning

- Simultaneously learns multiple related tasks, sharing knowledge across tasks.
- Shared layers capture general features, while task-specific layers refine predictions.

4 Data Pre-Processing

4.1 Data Transformations

- Logarithmic, scaling, and normalization transformations for improving model performance.

4.2 Imputation

- Strategies include mean, median, or mode imputation and more advanced techniques like k-Nearest Neighbors imputation.
- Missingness patterns (MCAR, MAR, MNAR) must be analyzed to decide imputation strategies.

4.3 Dimensionality Reduction

- **Principal Component Analysis (PCA):** Minimizes mean squared error (MSE) while retaining maximum variance.
- **t-SNE:** Projects high-dimensional data into 2D or 3D for visualization; preserves local structure.

4.4 Handling Outliers

- Probability-based methods (e.g., Parzen windows).
- Rule-based thresholds (e.g., 3 standard deviations from the mean).
- Distance-based methods (e.g., k-Nearest Neighbors).

5 Classification Theory and Methods

5.1 Decision Theory

- Bayes Decision Rule:

$$\text{Classify } x \text{ as } C_i \text{ if } P(C_i|x) > P(C_j|x), \forall j \neq i.$$

- Reject Option: Classify x as unknown if posterior probabilities are too close to minimize expected loss.

5.2 Bayesian Classifier

- Relies on prior probabilities $P(C)$ and likelihood $P(x|C)$ to compute posterior probabilities using Bayes Rule.
- Optimal classifier minimizes expected error.

5.3 Probabilistic Generative Models

- **Naive Bayes:** Assumes feature independence for simplicity.
- **LDA/QDA:** Assumes Gaussian distributions for each class.

5.4 Performance Metrics

- Accuracy: $\frac{\text{TP+TN}}{\text{Total}}$.
- Precision: $\frac{\text{TP}}{\text{TP+FP}}$.
- Recall: $\frac{\text{TP}}{\text{TP+FN}}$.
- F1-Score: Harmonic mean of precision and recall.
- ROC and AUC: Trade-off between true positive and false positive rates.
- Calibration: Measures whether predicted probabilities match actual frequencies.

6 Ensemble Methods

6.1 Bagging and Boosting

- **Bagging:** Reduces variance through model averaging; deeper trees reduce correlation.
- **Boosting:** Sequentially builds models focusing on prior errors; aggressively reduces bias and variance.
- AdaBoost: Adjusts weights iteratively to focus on harder examples.
- Gradient Boosted Decision Trees (GBDT): Reduces both bias and variance.

6.2 Random Forests

- Combines decision trees with random feature selection at each split.
- Evaluates feature importance and uses out-of-bag error for validation.

6.3 Mixture of Experts (MoE)

- Uses gating networks to assign data points to specific expert models.

6.4 Stacking Ensembles

- Uses meta-learners to combine base learners by training a second model on base predictions.
- Combines diverse models for improved generalization.

7 Deep Learning and Transformers

7.1 Convolutional Neural Networks (CNNs)

- Convolution operation:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(t)g(x-t)\,dt.$$

- Pooling operations reduce spatial dimensions.

7.2 Transformers

- Attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V.$$