Machine Learning Summary

Mathematical Notes

October 27, 2025

Contents

1 Supervised Learning

1.1 Problem Formulation

Definition 1.1. Given training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ are features and y_i are labels, find a function $f : \mathbb{R}^d \to \mathcal{Y}$ that generalizes well to unseen data.

1.2 Linear Regression

Definition 1.2. Linear regression assumes $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ and minimizes:

$$\mathcal{L}(\mathbf{w}, b) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T\mathbf{x}_i - b)^2$$

The closed-form solution is:

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

1.3 Logistic Regression

Definition 1.3. Logistic regression uses the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ and minimizes:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log p_i + (1 - y_i)\log(1 - p_i)]$$

where $p_i = \sigma(\mathbf{w}^T\mathbf{x}_i)$.

1.4 Regularization

- **L1 (Lasso)**: $\mathcal{L} + \lambda \sum_{j=1}^d |w_j|$
- **L2 (Ridge)**: $\mathcal{L} + \lambda \sum_{j=1}^d w_j^2$
- **Elastic Net**: $\mathcal{L} + \lambda_1 \sum_{j=1}^d |w_j| + \lambda_2 \sum_{j=1}^d w_j^2$

2 Neural Networks

2.1 Feedforward Networks

Definition 2.1. A **feedforward neural network** with L layers computes:

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)})$$

for $l = 1, \ldots, L$ where $\mathbf{h}^{(0)} = \mathbf{x}$.

2.2 Activation Functions

- **Sigmoid**: $\sigma(z) = \frac{1}{1+e^{-z}}$
- **Hyperbolic tangent**: $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$
- **ReLU**: $\mathrm{ReLU}(z) = \max(0, z)$
- **Leaky ReLU**: $\mathrm{LeakyReLU}(z) = \max(0.01z, z)$
- **Softmax**: $\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$

2.3 Backpropagation

Theorem 2.1 (Backpropagation Algorithm)**.** The gradient of the loss with respect to weights is:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_{ij}^{(l)}} = \delta_j^{(l)} h_i^{(l-1)}$$

where $\delta_j^{(l)}$ is the error signal.

2.4 Optimization Algorithms

- **SGD**: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \mathcal{L}(\mathbf{w}_t)$
- **Momentum**: $\mathbf{v}_{t+1} = \mu \mathbf{v}_t - \eta \nabla \mathcal{L}(\mathbf{w}_t)$, $\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbf{v}_{t+1}$
- **Adam**: Adaptive learning rates with momentum
- **RMSprop**: Root mean square propagation

3 Deep Learning

3.1 Convolutional Neural Networks

Definition 3.1. A **convolutional layer** applies filters $\mathbf{F}$ to input $\mathbf{X}$:

$$(\mathbf{X} * \mathbf{F})_{i,j} = \sum_{m,n} \mathbf{X}_{i+m,j+n} \mathbf{F}_{m,n}$$

3.2 Recurrent Neural Networks

Definition 3.2. An **RNN** maintains hidden state $\mathbf{h}_t$:

$$\mathbf{h}_t = \sigma(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{xh}\mathbf{x}_t + \mathbf{b}_h)$$

$$\mathbf{y}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y$$

3.3 Long Short-Term Memory (LSTM)

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad \text{(forget gate)} \tag{1}$$
$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad \text{(input gate)} \tag{2}$$
$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \quad \text{(candidate values)} \tag{3}$$
$$\mathbf{C}_t = \mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{C}}_t \quad \text{(cell state)} \tag{4}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \quad \text{(output gate)} \tag{5}$$
$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{C}_t) \quad \text{(hidden state)} \tag{6}$$

3.4 Attention Mechanisms

Definition 3.3. Self-attention computes:

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

3.5 Transformers

Definition 3.4. A **transformer** uses multi-head self-attention and feedforward layers without recurrence.

4 Unsupervised Learning

4.1 Clustering

Definition 4.1. K-means minimizes:

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} w_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

where $w_{ik} = 1$ if $\mathbf{x}_i$ belongs to cluster k, 0 otherwise.

4.2 Dimensionality Reduction

Definition 4.2. Principal Component Analysis (PCA) finds orthogonal directions of maximum variance by solving:

$$\max_{\mathbf{w}} \mathbf{w}^T \mathbf{S} \mathbf{w} \quad \text{subject to} \quad \|\mathbf{w}\| = 1$$

where $\mathbf{S}$ is the covariance matrix.

4.3 Autoencoders

Definition 4.3. An **autoencoder** learns to reconstruct input through an encoder-decoder architecture:

$$\mathbf{h} = f(\mathbf{x}), \quad \hat{\mathbf{x}} = g(\mathbf{h})$$

4.4 Generative Models

- **Generative Adversarial Networks (GANs)**
- **Variational Autoencoders (VAEs)**
- **Flow-based models**
- **Diffusion models**

5 Ensemble Methods

5.1 Random Forest

Definition 5.1. A **random forest** combines multiple decision trees trained on bootstrap samples with random feature selection.

5.2 Gradient Boosting

Definition 5.2. Gradient boosting iteratively fits weak learners to the negative gradient:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \gamma_m h_m(\mathbf{x})$$

where h_m minimizes the residual errors.

5.3 AdaBoost

Definition 5.3. **AdaBoost** adaptively weights training examples based on previous errors.

6 Model Evaluation

6.1 Classification Metrics

- **Accuracy**: $\frac{TP+TN}{TP+TN+FP+FN}$
- **Precision**: $\frac{TP}{TP+FP}$
- **Recall**: $\frac{TP}{TP+FN}$
- **F1-score**: $\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$
- **AUC-ROC**: Area under the receiver operating characteristic curve

6.2 Regression Metrics

- **Mean Squared Error**: $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- **Mean Absolute Error**: $\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- **R-squared**: $R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$

6.3 Cross-Validation

Definition 6.1. **k-fold cross-validation** splits data into k folds, trains on $k-1$ folds, and validates on the remaining fold.

7 Support Vector Machines

7.1 Linear SVM

Definition 7.1. **Linear SVM** finds the hyperplane that maximizes the margin between classes:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

7.2 Kernel Trick

Definition 7.2. The **kernel trick** allows SVMs to work in high-dimensional feature spaces without explicitly computing the transformation.

Common kernels:

- **Linear**: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$
- **Polynomial**: $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + c)^d$
- **RBF**: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$

8 Decision Trees

8.1 Splitting Criteria

- **Gini impurity**: $G = 1 - \sum_{i=1}^{c} p_i^2$
- **Entropy**: $H = -\sum_{i=1}^{c} p_i \log_2 p_i$
- **Information gain**: $\text{IG} = H(S) - \sum_{v \in \text{Values}} \frac{|S_v|}{|S|} H(S_v)$

8.2 Random Forest

Definition 8.1. Random Forest combines multiple decision trees with bagging and random feature selection.

9 Reinforcement Learning

9.1 Markov Decision Process

Definition 9.1. An **MDP** is defined by (S, A, P, R, γ) where:

- S is the state space
- A is the action space
- $P(s'|s, a)$ is the transition probability
- $R(s, a)$ is the reward function
- γ is the discount factor

9.2 Q-Learning

Definition 9.2. Q-learning updates the Q-function:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

9.3 Policy Gradient

Definition 9.3. Policy gradient methods directly optimize the policy using gradient ascent on the expected return.

10 Applications

10.1 Computer Vision

- Image classification
- Object detection
- Image segmentation
- Face recognition
- Medical imaging

10.2 Natural Language Processing

- Sentiment analysis
- Machine translation
- Question answering
- Text generation
- Named entity recognition

10.3 Speech Processing

- Speech recognition
- Speech synthesis
- Speaker identification
- Emotion recognition

10.4 Recommendation Systems

- Collaborative filtering
- Content-based filtering
- Hybrid approaches
- Matrix factorization

11 Important Theorems

11.1 Universal Approximation Theorem

Theorem 11.1. A feedforward neural network with a single hidden layer can approximate any continuous function on a compact set, given sufficient hidden units.

11.2 No-Free-Lunch Theorem

Theorem 11.2. No learning algorithm can be universally better than any other across all possible learning problems.

11.3 Bias-Variance Decomposition

Theorem 11.3. The expected prediction error can be decomposed as:

$$\mathbb{E}[(y - \hat{f}(x))^2] = \text{Bias}^2[\hat{f}(x)] + \text{Var}[\hat{f}(x)] + \sigma^2$$

11.4 VC Dimension

Definition 11.1. The **VC dimension** of a hypothesis class is the maximum number of points that can be shattered by the class.

12 Regularization Techniques

12.1 Dropout

Definition 12.1. Dropout randomly sets a fraction of input units to 0 during training to prevent overfitting.

12.2 Batch Normalization

Definition 12.2. Batch normalization normalizes the inputs to each layer to reduce internal covariate shift.

12.3 Early Stopping

Definition 12.3. Early stopping terminates training when validation performance stops improving.

13 Hyperparameter Tuning

13.1 Grid Search

Definition 13.1. Grid search exhaustively searches through a specified parameter grid.

13.2 Random Search

Definition 13.2. Random search samples hyperparameters from specified distributions.

13.3 Bayesian Optimization

Definition 13.3. Bayesian optimization uses a probabilistic model to guide the search for optimal hyperparameters.