Machine Learning Homework 2

April 11, 2025

2023141460251 Haoxiang Sun

Question 1: The Evolution of Neuron from Biology to Mathematics

Neural network models are inspired by biological neurons but have evolved into powerful mathematical constructs. This question explores the conceptual transition from biology to computational models.

1.1 Biological Neuron

A biological neuron consists of:

- **Dendrites**: Receive signals from other neurons.
- **Cell body**: Integrates inputs.
- **Axon**: Transmits outputs to downstream neurons.

It fires an action potential when the total input exceeds a threshold:

$$\text{Output} = \begin{cases} 1 & \text{if } \sum(\text{synaptic inputs}) > \theta \\ 0 & \text{otherwise} \end{cases}$$

This all-or-none behavior inspired early computational models of decision-making.

1.2 McCulloch-Pitts Neuron (1943)

This was the first formalization of an artificial neuron. Given binary inputs $x_1, x_2, \ldots, x_n \in \{0, 1\}$ and weights $w_1, w_2, \ldots, w_n$, the neuron computes:

$$z = \sum_{i=1}^{n} w_i x_i, \quad y = \begin{cases} 1 & \text{if } z \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

This neuron can represent logic gates (AND, OR, NOT) but lacks learning ability and is limited to linearly separable functions.

1.3 Perceptron (1958)

The perceptron introduces a trainable component into the model. For an input vector $\mathbf{x} \in \mathbb{R}^n$, weights $\mathbf{w}$, and bias b, the perceptron computes:

$$z = \mathbf{w}^T\mathbf{x} + b, \quad y = \phi(z) = \begin{cases} +1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Learning Rule: When a sample $(\mathbf{x}_i, y_i)$ is misclassified (i.e., $\hat{y}_i \neq y_i$), update:

$$\mathbf{w} \leftarrow \mathbf{w} + \eta(y_i - \hat{y}_i)\mathbf{x}_i, \quad b \leftarrow b + \eta(y_i - \hat{y}_i),$$

where η is the learning rate. This rule corrects the weight vector in the direction of the error.

1.4 Modern Neurons and Activation Functions

Modern neural networks use differentiable activation functions, such as:

$$\text{Sigmoid: } \phi(z) = \frac{1}{1+e^{-z}}, \quad \text{ReLU: } \phi(z) = \max(0, z),$$

allowing for gradient-based training via backpropagation:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\nabla_{\mathbf{w}}\mathcal{L}, \quad \mathcal{L} = \frac{1}{m}\sum_{i=1}^{m}\ell(y_i, \hat{y}_i).$$

This change enables modeling of complex, non-linear relationships.

1.5 Summary

The evolution from biological neurons to mathematical models illustrates the progression from mimicking nature to developing efficient computational algorithms. Early models, although simplistic, laid the foundation for modern deep learning architectures capable of complex tasks such as classification, regression, and pattern recognition.

Question 2: The Key Concepts of Perceptron and Its Limitations

The perceptron is one of the seminal models in machine learning. Here, we provide a detailed explanation of its core ideas and the shortcomings that motivate later, more sophisticated approaches.

2.1 Detailed Key Concepts

Structure:

- **Input and Weights**: Each input $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is linearly combined with a corresponding weight vector $\mathbf{w}$. The linear form $\mathbf{w}^T\mathbf{x}+b$ represents the aggregation of features.
- **Activation Function**: The perceptron employs a step function:

$$y = \phi(z) = \begin{cases} +1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

  This non-differentiable function strictly categorizes inputs.

- **Learning Mechanism**: When the predicted label $\hat{y}_i$ does not match the true label y_i, the weights and bias are updated:

$$\mathbf{w} \leftarrow \mathbf{w} + \eta(y_i - \hat{y}_i)\mathbf{x}_i, \quad b \leftarrow b + \eta(y_i - \hat{y}_i).$$

  This iterative process continues until the model converges (when data are linearly separable) or does not converge if they are not.

- **Geometric View**: The decision boundary is given by $\mathbf{w}^T\mathbf{x} + b = 0$. Geometrically, $\mathbf{w}$ is perpendicular to this hyperplane. The signed distance from any point to the hyperplane is:

$$d(\mathbf{x}) = \frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|},$$

  which indicates the confidence in classification.

2.2 Supplementary Analysis of Limitations

- **Linear Separability Requirement**: The perceptron converges only if the training data are linearly separable. For instance, consider the XOR problem:

$\mathbf{x}$	y
$(0,0)$	-1
$(0,1)$	$+1$
$(1,0)$	$+1$
$(1,1)$	-1

  No linear hyperplane can separate these points correctly, leading to perpetual updates without convergence.

- **Binary Decision and No Probabilities**: The binary step function outputs only ± 1 and does not provide a measure of confidence. Unlike logistic regression, which outputs probabilities via the sigmoid, the perceptron does not offer insights into uncertainty.

- **Sensitivity to Noise**: The model is sensitive to outliers. An erroneously labeled or noisy sample can cause large updates, destabilizing the hyperplane position.

- **Single-Layer Structure**: With only a single layer, the perceptron cannot capture hierarchical or non-linear patterns. Complex decision boundaries require multiple layers (multilayer perceptrons), which incorporate non-linear activations.

- **Lack of Margin Maximization**: The algorithm does not explicitly maximize the separation (margin) between classes—a concept later exploited by Support Vector Machines (SVMs) to enhance generalization.

2.3 Historical Extensions and Impact

Despite these limitations, the perceptron:

- Sparked interest in neural computation and led directly to the development of adaptive linear neurons (ADALINE) and multilayer networks.

- Paved the way for backpropagation and deep learning methods that overcome non-linear problems.

- Encouraged exploration of margin-based methods such as SVMs and logistic regression.

The perceptron remains a crucial pedagogical model that underscores fundamental concepts in learning theory.

Question 3: Who is Vladimir N. Vapnik?

Vladimir Naumovich Vapnik is a towering figure in the realm of statistical learning theory. His work has laid the foundation for many modern machine learning methods.

3.1 Biography and Contributions

- **Background**: Born in 1936, Vapnik's early research in the Soviet Union led to foundational work in pattern recognition and neural networks.
- **Support Vector Machines (SVMs)**: In collaboration with Alexey Chervonenkis and later with Corinna Cortes, Vapnik developed SVMs. SVMs focus on finding a hyperplane that maximizes the margin between classes:

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

  with extensions to soft margins that introduce slack variables.
- **VC Dimension**: Vapnik introduced the concept of Vapnik-Chervonenkis (VC) dimension, a measure of the capacity of a hypothesis space. This concept is essential for understanding the trade-off between model complexity and generalization performance.
- **Statistical Learning Theory**: His formulation of Empirical Risk Minimization (ERM) and Structural Risk Minimization (SRM) provided theoretical guarantees on how well a model generalizes from training data to unseen data.
- **Kernel Methods and Non-linear Classification**: Vapnik's work popularized the kernel trick, allowing SVMs to perform non-linear classification by mapping data into high-dimensional spaces.

3.2 Impact on Machine Learning

Vapnik's contributions have had far-reaching effects:

- His theories underpin many modern algorithms, influencing not only SVMs but also the design of various learning frameworks.
- The principles of margin maximization and capacity control continue to shape research in deep learning and generalization theory.
- His seminal book, *The Nature of Statistical Learning Theory* (1995), remains a key reference for theoreticians and practitioners alike.

Vapnik's work bridges theory and practice, making him one of the most influential figures in machine learning.

Question 4: Maximum Margin Principle and the Role of Support Vectors

The maximum margin principle is central to SVMs, and understanding it is key to appreciating why SVMs generalize well.

4.1 Maximum Margin Principle

SVMs are designed to find the hyperplane that maximizes the margin between two classes. The margin is defined as the distance between the hyperplane and the nearest points from any class:

$$\text{Margin} = \frac{2}{\|\mathbf{w}\|}.$$

Maximizing the margin (or equivalently minimizing $\|\mathbf{w}\|^2$) under the constraints:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i,$$

leads to improved generalization because a larger margin typically reduces model variance and increases robustness.

4.2 The Role of Support Vectors

- **Definition**: Support vectors are the training examples that lie exactly on or within the margin boundaries. They are critical because they directly define the position and orientation of the decision boundary.
- **Sparse Representation**: The optimal weight vector is expressed solely in terms of support vectors:

  $$\mathbf{w} = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i,$$

  where α_i are the Lagrange multipliers from the dual formulation.
- **Geometric Significance**: Since only the support vectors affect the model parameters, SVMs often exhibit sparsity in representation, meaning that many non-critical data points do not influence the decision function:

  $$f(\mathbf{x}) = \sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b.$$

- **Soft-Margin Extension**: In the presence of noise, support vectors include points that violate the margin (with $0 < \alpha_i \leq C$ or $\alpha_i = C$), striking a balance between classification accuracy and margin width.

4.3 Supplementary Insights

The maximum margin principle connects to Structural Risk Minimization (SRM), where one balances training error with model complexity. A larger margin reduces the VC dimension of the classifier, resulting in better generalization bounds. Thus, support vectors are not only data points that "support" the hyperplane but also represent the most informative and borderline cases in the dataset.

Question 5: How to Compute the Distance from a Point to the Decision Boundary

Computing the distance from a given point to the hyperplane is fundamental in SVMs and other linear classifiers.

5.1 Distance Formula Derivation

Given a hyperplane defined by:

$$\mathbf{w}^T\mathbf{x} + b = 0,$$

the perpendicular (shortest) distance d from a point $\mathbf{x}_0$ to the hyperplane is:

$$d = \frac{|\mathbf{w}^T\mathbf{x}_0 + b|}{\|\mathbf{w}\|}.$$

Derivation Outline:

- Identify any point $\mathbf{x}_p$ on the hyperplane (i.e., satisfying $\mathbf{w}^T\mathbf{x}_p + b = 0$).
- The vector $\mathbf{x}_0 - \mathbf{x}_p$ is decomposed along the normal direction $\frac{\mathbf{w}}{\|\mathbf{w}\|}$.
- The projection of $\mathbf{x}_0 - \mathbf{x}_p$ onto this unit vector yields the distance.

5.2 Example Calculation

For instance, let:

$$\mathbf{w} = [2, 1]^T, \quad b = -3, \quad \mathbf{x} = [4, 2]^T.$$

Then,

$$\mathbf{w}^T\mathbf{x} + b = 2 \cdot 4 + 1 \cdot 2 - 3 = 7,$$

$$\|\mathbf{w}\| = \sqrt{2^2 + 1^2} = \sqrt{5},$$

so the distance is:

$$d = \frac{7}{\sqrt{5}} \approx 3.13.$$

5.3 Extensions to Non-Linear Cases

In non-linear SVMs using kernels, the decision function is:

$$f(\mathbf{x}) = \sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b.$$

Direct calculation of geometric distances becomes more challenging. Approximate measures, such as using the value of $|f(\mathbf{x})|$ normalized appropriately, serve as proxies for distance in the transformed feature space.

Question 6: What Are the Limitations of Linear SVM?

Linear SVMs are robust classifiers; however, they face several limitations when applied to complex or large-scale data.

6.1 Limitations

- **Assumption of Linear Separability**: Linear SVMs assume that data can be separated by a straight line (or hyperplane). For complex, non-linear patterns (e.g., circular or concentric distributions), no linear separator will work well.
- **Sensitivity to Outliers**: Since the optimal hyperplane is determined by support vectors, noisy data or outliers can disproportionately influence the model, especially if they reside near the margin.

- **High-Dimensional Challenges**: In high-dimensional settings ($n \gg m$), the SVM can overfit since a hyperplane may perfectly separate the classes by chance; regularization via the parameter C is essential, yet finding the optimal C can be challenging.
- **Lack of Probabilistic Outputs**: The decision function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ only gives a class decision. Post-processing methods such as Platt scaling can estimate probabilities but are not intrinsic to the linear SVM formulation.
- **Binary Classification Constraint**: Linear SVMs natively support only binary classification. Multi-class problems require strategies like one-vs-rest or one-vs-one, which increase computational complexity and may lead to inconsistent results.
- **Feature Scaling Dependence**: The optimization assumes that features are properly normalized. Unscaled data can distort the margin estimation, leading to suboptimal decision boundaries.

6.2 Practical Solutions and Extensions

To overcome these limitations:

- **Kernel SVMs**: By mapping data into higher-dimensional spaces using kernels (e.g., Gaussian, polynomial), one can capture non-linear relationships without explicitly computing transformations.
- **Regularization and Soft Margins**: Introducing slack variables and tuning the regularization parameter C help balance margin maximization with misclassification tolerance.
- **Probabilistic Calibration**: Techniques such as Platt scaling or isotonic regression allow for extracting probability estimates from the SVM scores.
- **Multi-class Strategies**: One-vs-rest or one-vs-one techniques enable extension to multi-class tasks, though at the cost of increased model complexity.