پیش گزارش آزمایش ۵

ستایش خاصه تراش ۴۰۰۲۳۰۲۴ مهشاد اکبری سریزدی۹۹۲۳۰۹۳ ساحل نادرزاده۹۹۲۳۰۸۱

1. What about multi-layer RBFs? (reasons for lack of development, how deep can a pare RBF network be)

**Reasons for Lack of Development in Multi-Layer RBFs**

***Complexity***: Multi-layer RBF networks are more complex to design, and train compared to single-layer RBF networks1. This complexity can make them less appealing for some applications.

***Overfitting***: Adding more layers can increase the risk of overfitting, especially if the network is not properly regularized.

***Competition***: Other neural network architectures, like deep learning models (e.g., convolutional neural networks), have shown better performance in many tasks, leading to more research and development in those areas.

**Depth of a Multi-Layer RBF Network**

The depth of a multi-layer RBF network can vary depending on the specific application and the complexity of the problem. However, in practice, multi-layer RBF networks are typically not very deep. They usually consist of one or two hidden layers2. Adding more layers can increase the computational cost and risk of overfitting without necessarily providing significant performance improvements.

1. For finding the best architecture regarding a problem there are several methods, study some of the methods and reports

***1. Grid Search***

**Grid Search**is a systematic approach that involves exhaustively searching through a predefined set of hyperparameters. The concept of grid search has been around for quite some time, but it became more formalized and widely used in the context of machine learning and hyperparameter tuning in the early 2000s.

***2. Random Search***

**Random Search**was introduced earlier than grid search. The method was first described by Anderson in 1953. It involves randomly sampling hyperparameter combinations from a specified distribution, which can be more efficient than grid search, especially when dealing with a large number of hyperparameters.

***3. Bayesian Optimization***

**Bayesian Optimization** is a sequential design strategy for global optimization of black-box functions. It was popularized in the early 21st century but its roots go back to the work of Jonas Mockus in the 1970s and 1980s. Bayesian optimization uses a probabilistic model to predict the performance of hyperparameters and chooses the next set of hyperparameters to evaluate based on this model.

***4. Genetic Algorithms***

**Genetic Algorithms** are inspired by the process of natural selection and genetics. The concept was first introduced in the 1950s by researchers like Friedberg and later formalized by John Holland in the 1970s. Genetic algorithms simulate the process of natural evolution to search for optimal solutions.

***5. Hyperband***

**Hyperband** is a bandit-based approach to hyperparameter optimization that was introduced in 2016 by Lisha Li and colleagues. It focuses on efficiently allocating resources to hyperparameter configurations and uses early stopping to speed up the search process.

***6. Neural Architecture Search (NAS)***

**Neural Architecture Search (NAS)** is a technique for automating the design of neural networks. It has gained significant attention in recent years, especially with advancements in deep learning. NAS involves defining a search space, search strategy, and performance estimation strategy to find the best architecture for a given task.