Lesson 9: Recurrent Networks

Table of Contents

9.1 Introduction

In this lesson, we will be exploring different types of neural networks and their applications. We will start by discussing the Hopfield network, which is a type of recurrent neural network that is used for pattern recognition and optimization problems. Next, we will delve into stochastic machines, which are neural networks that make use of randomness to improve their performance. Finally, we will look at simulated annealing, which is a technique used to find the global minimum of a function through a process of random sampling. Throughout the lesson, we will discuss the key concepts, algorithms, and applications of these neural network models.

9.2 Hopfield network.

The Hopfield network is a type of recurrent neural network (RNN) that was first introduced by John Hopfield in 1982. It is designed to store and recall patterns, and is often used in applications such as pattern recognition, optimization, and associative memory.The Hopfield network consists of a set of nodes, also known as neurons, that are connected by edges or weights. Each node represents a binary state, either 1 or -1, and the weights between the nodes represent the strength of the connection. The network is trained by presenting it with a set of patterns, and the weights are adjusted such that the network can recall these patterns later on.

The way the Hopfield network works is by updating the state of each neuron based on the states of the other neurons in the network. This is done by applying a rule called the Hebbian rule, which states that the weight between two neurons should be increased if the neurons are both activated at the same time. This means that the network will become more sensitive to patterns that are frequently presented to it.The Hopfield network is also able to solve optimization problems by using a process called energy minimization. The energy of the network is a function of the states of the neurons and the weights between them, and the goal is to find the state of the network that corresponds to the minimum energy. This can be done by updating the state of the neurons in a random order until the energy reaches a minimum.

9.3 Stochastic machines

Stochastic machines, also known as stochastic neural networks, are a type of neural network that make use of randomness to improve their performance. They are based on the idea that incorporating randomness into the learning process can help the network to escape local minima and find the global minimum of the error function.

One of the most common types of stochastic machines is the Boltzmann machine, which was first introduced by Geoffrey Hinton and Terrence Sejnowski in 1983. A Boltzmann machine is a type of energy-based model that consists of a set of binary nodes, or neurons, that are connected by edges or weights. Each node represents a binary state, either 1 or 0, and the edges represent the strength of the connection between the nodes.

The way a Boltzmann machine works is by using a process called simulated annealing. This is a technique that is inspired by the process of annealing in metallurgy, where a material is heated and then cooled slowly to improve its properties. In the case of a Boltzmann machine, the network is initialized with random weights and the state of the nodes is updated in a random order. The state of each node is determined by the probabilities of the node being in the state of 1 or 0 based on the current energy of the network. The energy of the network is a function of the states of the nodes and the weights between them, and the goal is to find the state of the network that corresponds to the minimum energy. As the network is updated, the temperature of the simulated annealing process is gradually decreased, which allows the network to converge to a state that corresponds to the global minimum of the energy function.

Another type of stochastic machine is the Restricted Boltzmann Machine (RBM). An RBM is a type of Boltzmann machine that is composed of two layers of neurons, a visible layer and a hidden layer. The visible layer represents the input data, and the hidden layer represents the features of the data. The neurons in the visible layer are connected to the neurons in the hidden layer by edges or weights. The goal of an RBM is to learn a set of features that can be used to reconstruct the input data.

Stochastic machines have been used in a variety of applications such as image recognition, natural language processing, and recommendation systems. In image recognition, for example, an RBM can be used to learn a set of features that can be used to classify images based on their content. In natural language processing, an RBM can be used to learn a set of features that can be used to generate text. In recommendation systems, an RBM can be used to learn a set of features that can be used to predict the preferences of users.

In conclusion, stochastic machines are a type of neural network that make use of randomness to improve their performance. They are based on the idea that incorporating randomness into the learning process can help the network to escape local minima and find the global minimum of the error function. Boltzmann machines and Restricted Boltzmann Machines are the most commonly used types of stochastic machines, and they have been applied in a variety of fields such as image recognition, natural language processing, and recommendation systems.

9.4 Simulated annealing

Simulated annealing is a optimization algorithm that is inspired by the process of annealing in metallurgy, where a material is heated and then cooled slowly to improve its properties. The goal of simulated annealing is to find the global minimum of a function through a process of random sampling.The basic idea behind simulated annealing is to start with a random initial state and then make small changes to the state in a random direction. The change in the state is determined by a probability function, called the acceptance probability, which depends on the current state, the new state, and the temperature of the process. At high temperatures, the acceptance probability is high, which means that the algorithm is likely to accept a new state even if it is worse than the current state. As the temperature is gradually decreased, the acceptance probability becomes smaller, which means that the algorithm is more likely to accept a new state only if it is better than the current state.The simulated annealing algorithm can be used to find the global minimum of a function by starting with a high temperature and gradually decreasing the temperature over time. As the temperature is decreased, the algorithm becomes more selective in the states that it accepts, which allows it to converge to the global minimum.

One of the key advantages of simulated annealing is that it is able to escape local minima, which are states that are not the global minimum but are still better than the surrounding states. This is because the algorithm is able to make random moves even if they result in a worse state, which allows it to explore the entire solution space.Simulated annealing has been applied in a variety of fields such as image processing, speech recognition, and control systems. For example, it can be used to solve the travelling salesman problem, which is an NP-hard problem of finding the shortest route through a set of cities. In image processing, simulated annealing can be used to optimize the parameters of image restoration algorithms. In speech recognition, it can be used to optimize the parameters of speech recognition systems

9.5 Lesson 9 Questions

1. What is the main goal of a Hopfield network?
2. How does a Boltzmann machine work?
3. What is the difference between a Boltzmann machine and a Restricted Boltzmann Machine?
4. How does simulated annealing help to find the global minimum of a function?
5. Give an example of an application of simulated annealing.
6. How does the temperature of the simulated annealing process affect the acceptance probability?
7. How does randomness help in the learning process of stochastic machines?
8. Why are Hopfield networks considered recurrent neural networks?
9. How does the Hebbian rule affect the weights of a Hopfield network?
10. How can simulated annealing help in solving the travelling salesman problem?