Scholar's Digest- Journal of Multidisciplinary Studies
Volume 02, Issue 10, October, 2023
ISSN (E): 2949-8856
Scholarsdigest.org

REGULATION OF TRANSPORT SYSTEM BASED ON HEBB'S ALGORITHM

Djurayev Sherzod Sobirjonovich
Namangan Institute of Engineering and Technology
sherzoddjurayev1989@gmail.com

Fayzullayev Dedaxuja Zokirjon o`g`li
Namangan Institute of Engineering and Technology
fdedaxuja@gmail.com

Abstract:
The regulation of transport systems is crucial for effective traffic management, logistics, and supply chain optimization. This article explores the application of Hebb's algorithm, inspired by synaptic plasticity in the human brain, to control and optimize transport systems. We delve into the mathematical expression and initial conditions of Hebb's algorithm, discussing its potential in traffic management, route optimization, and resource allocation. We also address the challenge of unbalanced datasets in digit classification and propose a solution using Hebb's algorithm. Furthermore, we compare Hebb's algorithm with Kohonen's algorithm, highlighting their respective advantages. Ultimately, Hebb's algorithm offers a promising approach to enhance transport system regulation and traffic optimization based on neural network principles.
.

Keywords: Regulation of transport systems, Traffic management, Logistics, Supply chain optimization, Hebb's algorithm, Synaptic plasticity, Mathematical expression, Initial conditions, Learning rate, Neuron representation, Weight initialization, Applications in transport system regulation, Unbalanced dataset, Classification of handwritten digits, Hebb's rule, Kohonen's algorithm, Supervised learning, Unsupervised learning, Neuron representation, Weight update, Output generation, Advantages of Hebb's algorithm, Advantages of Kohonen's algorithm, Linear relationship, Traffic flow estimation.

INTRODUCTION

The regulation of transport systems plays a vital role in various domains such as traffic management, logistics, and supply chain optimization. Efficiently controlling the flow of vehicles or goods requires intelligent algorithms that can adapt and learn from the underlying patterns in the system. One such algorithm is Hebb's algorithm, which is inspired by the concept of synaptic plasticity in the human brain. In this article, we will explore the mathematical expression and initial conditions of Hebb's algorithm and discuss its potential applications in regulating transport systems.

Hebb's Algorithm: Hebb's algorithm, proposed by Donald Hebb in 1949, is a learning rule that strengthens or weakens the connections between neurons based on their activity. It is often used to model associative learning and memory formation in neural networks. The algorithm can be adapted to regulate transport systems by considering the vehicles or goods as "neurons" and the connections between them as "synapses."

Mathematical Expression: The mathematical expression of Hebb's algorithm can be represented as follows:

$$\Delta w_{ij} = \eta * x_i * x_j$$

Where:

- Δw_{ij} represents the change in the weight (connection) between neurons i and j.
- η is the learning rate, which determines the speed at which the weights are updated.
- x_i and x_j are the activities or states of neurons i and j, respectively.

The algorithm updates the weights between neurons based on the product of their activities. If two neurons are often active simultaneously, the weight between them is strengthened, indicating a strong connection. On the other hand, if two neurons are rarely active together, the weight between them is weakened, suggesting a weak or non-existent connection.

Initial Conditions: To apply Hebb's algorithm to regulate transport systems, certain initial conditions need to be defined. These conditions include:

1. Neuron Representation: Each vehicle or good in the transport system is considered as a neuron. The state or activity of a neuron can be defined based on various factors such as speed, location, or availability.
2. Weight Initialization: The initial weights between neurons represent the initial connections or relationships between vehicles or goods. These weights can be randomly initialized or set based on prior knowledge of the system.
3. Learning Rate Selection: The learning rate (η) determines the magnitude of weight updates and should be carefully selected. A high learning rate may lead to rapid changes in the system, while a low learning rate may result in slow adaptation.

Applications in Transport System Regulation: Hebb's algorithm can be applied to various aspects of transport system regulation, including traffic management, route optimization, and resource allocation. By considering vehicles or goods as neurons and their connections as weights, the algorithm can learn from the patterns and behaviors observed in the system.

For example, in traffic management, the algorithm can adaptively adjust traffic light timings based on the activities of vehicles at different intersections. By strengthening the weights between intersections experiencing high traffic flow simultaneously, the algorithm can optimize the overall traffic flow and reduce congestion.

In logistics and supply chain optimization, Hebb's algorithm can learn from the historical data of goods movement and adjust the routing or allocation of resources accordingly. By strengthening the weights between efficient routes or allocation strategies, the algorithm can enhance the overall efficiency and reduce costs.

Conclusion: Hebb's algorithm provides a promising approach for regulating transport systems based on the principles of synaptic plasticity in the human brain. By leveraging the mathematical expression and initial conditions of the algorithm, we can adaptively learn from the underlying patterns and optimize the flow of vehicles or goods. With further research and development, Hebb's algorithm holds great potential to revolutionize the efficiency and effectiveness of transport system regulation, leading to improved traffic management, logistics, and supply chain optimization.

Problem: Consider a scenario where you have a dataset consisting of images of handwritten digits (0-9) and you want to develop a system that can accurately classify these digits. However, the dataset is unbalanced, meaning that there are significantly more samples for certain digits compared to others. This poses a challenge as the algorithm may become biased towards the majority class and perform poorly on the minority classes.

Solution using Hebb's Algorithm: To address this problem, we can leverage Hebb's algorithm and its main theorem to develop a solution. The main theorem of Hebb's algorithm states that "neurons that fire together, wire together." In other words, if two neurons are frequently active simultaneously, the connection between them is strengthened.

Here's how we can apply Hebb's algorithm to solve the unbalanced digit classification problem:

1. Initialize the weights:
- Initialize the weights between neurons (representing pixels) randomly or with small values close to zero.
2. Iterate through the dataset:
- For each image in the dataset, calculate the activation of each neuron by multiplying the pixel values with their corresponding weights.
- Update the weights between active neurons using Hebb's rule: $\Delta w_{ij} = \eta * x_i * x_j$, where Δw_{ij} is the change in weight between neurons i and j, η is the learning rate, x_i and x_j are the activities of neurons i and j, respectively.
- Repeat this process for all images in the dataset.
3. Classify new digits:
- Once the weights have been updated using Hebb's algorithm, we can use them to classify new digits.
- For each new digit, calculate the activation of each neuron using the updated weights.
- Determine the class of the digit based on the neuron with the highest activation.

By applying Hebb's algorithm and its main theorem, the weights between neurons representing pixels are adjusted based on their co-occurrence. This allows the algorithm to learn the patterns and relationships between pixels in the dataset. Consequently, the algorithm can better discriminate between different digits, even in the case of an unbalanced dataset.

It is important to note that while Hebb's algorithm can help address the unbalanced digit classification problem, it may not be sufficient on its own. Additional techniques such as data augmentation, oversampling or undersampling, or using more advanced machine learning algorithms can be combined with Hebb's algorithm to further improve the accuracy and performance of the classification system.

Differences between Hebb's Algorithm and Kohonen's Algorithm:

1. Learning Paradigm:

- Hebb's Algorithm: Hebb's algorithm is a supervised learning algorithm, meaning it requires labeled training data to learn the patterns and relationships between inputs and outputs.

- Kohonen's Algorithm: Kohonen's algorithm is an unsupervised learning algorithm, meaning it does not require labeled training data. It learns the underlying structure and patterns of the data solely based on input patterns.

2. Neuron Representation:

- Hebb's Algorithm: In Hebb's algorithm, each neuron typically represents a feature or attribute of the input data.
- Kohonen's Algorithm: In Kohonen's algorithm, each neuron represents a prototype or cluster center that captures the characteristics of a group of similar input patterns.

3. Weight Update:

- Hebb's Algorithm: Hebb's algorithm updates the weights between neurons based on their co-occurrence or simultaneous activation. The weight update is proportional to the product of the activities of the connected neurons.
- Kohonen's Algorithm: Kohonen's algorithm updates the weights between neurons based on their distance or similarity to the input pattern. The weight update is proportional to the difference between the input pattern and the weight vector of the neuron.

4. Output Generation:

- Hebb's Algorithm: Hebb's algorithm generates an output based on the neuron with the highest activation or activity level.
- Kohonen's Algorithm: Kohonen's algorithm generates an output based on the neuron with the closest weight vector to the input pattern.

Advantages of Hebb's Algorithm:

- Hebb's algorithm is relatively simple and computationally efficient.
- It can learn and adapt to patterns and relationships in the data.
- It can be applied to both classification and regression problems.
- It does not require a large amount of labeled training data.

Advantages of Kohonen's Algorithm:

- Kohonen's algorithm is unsupervised, making it suitable for exploratory data analysis and clustering tasks.
- It can discover the underlying structure and patterns in the data without the need for labeled training data.
- It can handle high-dimensional and complex data.
- It can visualize the data and provide insights into the clusters and prototypes.

Conclusion:

Hebb's algorithm is a supervised learning algorithm that learns patterns and relationships between inputs and outputs. On the other hand, Kohonen's algorithm is an unsupervised learning algorithm that discovers the underlying structure and patterns in the data. Both algorithms have their own advantages and can be applied to different types of problems.

Hebb's algorithm can be used to estimate traffic flow based on the activity of sensors or detectors installed along the road. The algorithm learns patterns and relationships between sensor readings and traffic flow. Here are the traffic flow estimation equations based on the Hebb algorithm:

Initialization:

- Initialize the weights between sensors and estimated traffic flow values with random or small values close to zero.

Iterate over the data set:

- For each t step, calculate the activation of each sensor by multiplying the sensor readings by their corresponding weights.
- Estimate the traffic flow at stage t by summing the weighted activations of all sensors: flow_estimate(t) = S (activation(sensor_i) * weight_i)Og'irliklarni yangilang:
- Update weights between sensors and traffic flow estimates using Hebb's rule: weight_i(t+1) = weight_i(t) + η * activation(sensor_i) * (flow_actual(t) - flow_estimate(t)) where weight_i(t+) 1) updated weight between sensor i and the traffic flow estimate at time step t+1, η is the learning rate, activation(sensor_i) is the activation of sensor i, flow_true(t) is the actual traffic flow at time step t, and flow_estimate(t) - estimated traffic flow at time step t.

By applying the Hebb algorithm and updating the weights based on the difference between the actual traffic flow and the estimated traffic flow, the algorithm studies the relationship between the sensor readings and the traffic flow. Over time, weights are adjusted to increase the accuracy of traffic flow estimation.

The Hebb algorithm assumes a linear relationship between sensor readings and traffic flow. In practice, the relationship may be more complex and may require the use of more advanced machine learning algorithms or additional features to improve the accuracy of traffic flow estimation. However, it is important to note that the Hebb algorithm assumes a linear relationship between the sensor readings and the traffic flow, which may not always hold true in real-world scenarios. In such cases, more advanced machine learning algorithms or additional features may be required to enhance the accuracy of traffic flow estimation.

Overall, the Hebb algorithm provides a solid foundation for traffic flow estimation and can be a valuable tool for traffic management and planning. By leveraging the power of machine learning, it enables the extraction of meaningful insights from sensor data and contributes to the optimization of traffic flow in urban environments.

References:

1. Барский А.Б. Нейронные сети: распознавание, управление, принятие решений. – М.: Финансы и статистика, 2004. – 176 с.
2. Каллан Роберт. Основные концепции нейронных сетей: Пер. с англ. – М.:Издательский дом «Вильямс», 2001.
3. Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры: Учеб. пособие для вузов. – 2-е изд.,перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. – 400 с. Технические науки — от теории к практике № 11 (47), 2015 г. www.sibac.info
4. Мелихова О.А., Чумичев В.С., Джамбинов С.В., Гайдуков А.Б. Некоторые аспекты криптографического взлома и повышения надежности алгоритмов шифрования// Молодой ученный. – Казань, № 11(91), 2015. –С. 392–394.

5. Мелихова О.А. Приложение матлогики к проблемам моделирования// Известия ЮФУ. Технические науки. – Таганрог: Изд-во ТТИ ЮФУ, 2014. № 7(156). – С. 204–214.
6. Madaliyev , X. «Creation of interface through app design of matlab software for automatic determination of loads on roller machine worker shaft». Interpretation and Researches, т. 1, вып. 10, июнь 2023 г.,
7. Fayzullayev Dedakhuja Zokirjon oglu. (2022). Mexbios development studio software package for developing control programs and modeling electric drive systems. Web of Scientist: International Scientific Research Journal, 3(5), 1964–1967. https://doi.org/10.17605/OSF.IO/G9BF8
8. Ismonovich K. A., Abdusamatugli I. M. Modeling the Method of Linear Approximation of Signals in SPLC (Sensor Programmable Logic Controller) //International Journal on Orange Technologies. – 2021. – Т. 3. – №. 10. – С. 55-59.
9. Ismonovich K. A., Muhammadziyo I. Mathematical Modeling of Heat Flux Distribution in Raw Cotton Stored in Bunt //Engineering. – 2020. – Т. 12. – №. 8. – С. 591-599.