AIDS - ANN - Unit 4 - Competitive learning Neural Network

Unit 4 : Competitive Learning Neural Network

Unit IV	Competitive learning Neural Network	07(Hours)
Components of CL network.Pattern clustering and feature mapping network, ART networks, Features of ART models, character recognition using ART network. Self-Organization Maps (SOM): Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification		

Introduction

A competitive learning neural network is a type of artificial neural network that is trained using unsupervised learning techniques. It is designed to find the most similar pattern or cluster in a set of input data. This type of network is also known as a self-organizing map (SOM) or Kohonen network, named after its inventor, Teuvo Kohonen.

The competitive learning neural network consists of two layers: the input layer and the competitive layer. The input layer is connected to the competitive layer, where the neurons in the competitive layer compete with each other to become active based on the input data. The winning neuron, also known as the "best matching unit" (BMU), is the neuron with the weights that are most similar to the input data.

The learning process in the competitive learning neural network is based on the Hebbian learning rule, which states that the strength of the connection between two neurons should increase when they are active at the same time. When an input is presented to the network, the weights between the input and the winning neuron are strengthened, and the weights between the input and the other neurons are weakened. This process helps to improve the accuracy of the network in clustering and classification tasks.

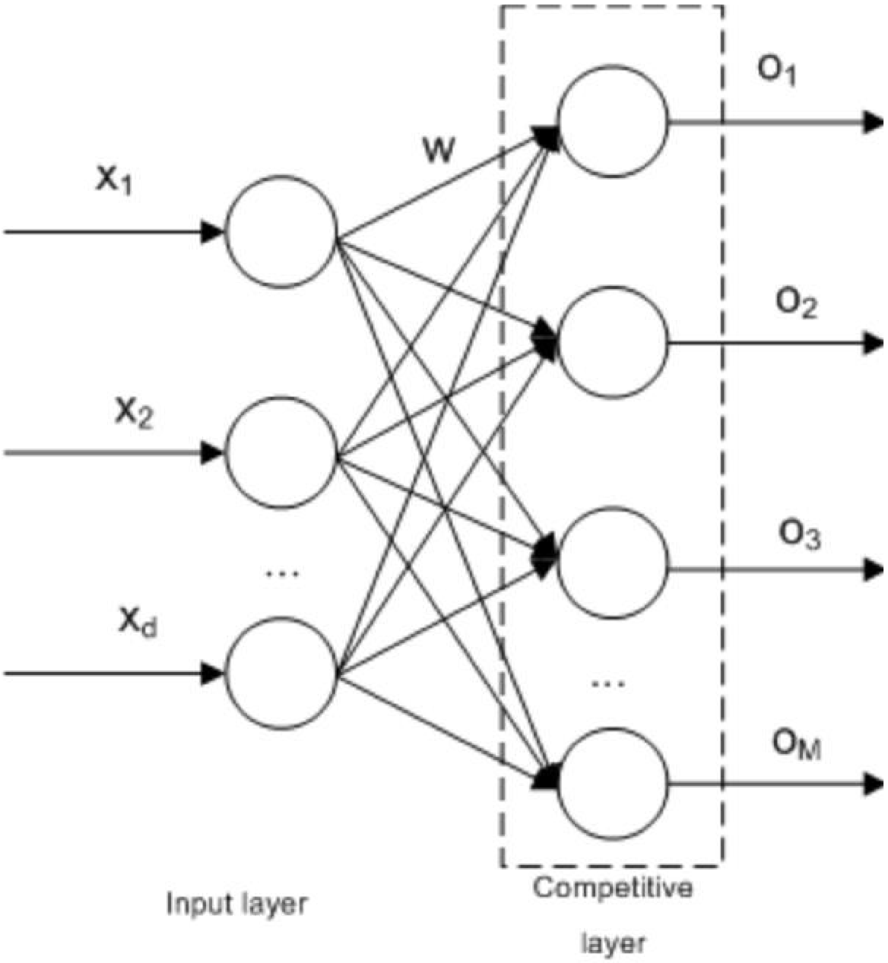
The competitive learning neural network is widely used in pattern recognition, image processing, and data analysis applications. It can be used for clustering data, detecting outliers, and performing classification tasks where the number of classes is not known in advance. The network is particularly useful when dealing with large, high-dimensional data sets, as it can reduce the dimensionality of the data and provide a compact representation of the input data.

Components of CL network

The network consists of three main components:

Input Layer: The input layer of the competitive learning neural network receives the input data that is to be processed by the network. Each input unit corresponds to one feature of the input data.

Competitive Layer: The competitive layer is the main component of the network, where the competition and learning take place. It consists of a set of neurons, each representing a cluster or class.



The neurons in the competitive layer compete with each other to become active based on the input data. The neuron that is the most similar to the input data becomes active and inhibits the activity of the other neurons. The activation of the winning neuron represents the cluster or class that the input data belongs to.

Output Layer: The output layer of the competitive learning neural network is responsible for producing the output of the network. It consists of one or more units that correspond to the output classes or clusters of the network. The output of the network is produced based on the activity of the neurons in the competitive layer. The winning neuron activates the corresponding output unit, indicating the class or cluster that the input data belongs to.

Competitive Neurons and Similarity Measure

Every competitive neuron is described by a vector of weights:

$$\mathbf{W}_i = (W_{i1}, \ldots, W_{id})^T, \quad i = 1, \ldots, M$$

where:

- $\mathbf{W}_i$ is the weight vector of the i-th neuron,
- W_{ij} represents the weight connecting the j-th input to the i-th neuron,
- d is the dimensionality of the input data,
- M is the total number of competitive neurons.

Each neuron calculates the similarity measure between the input data:

$$\mathbf{X}_n = (X_{n1}, \ldots, X_{nd})^T \in \mathbb{R}^d$$

where:

- $\mathbf{X}_n$ is the n-th input data vector,
- X_{nj} represents the j-th feature of the n-th input data,
- $\mathbb{R}^d$ denotes the d-dimensional real vector space.

The similarity measure is typically computed using a distance metric (e.g., Euclidean distance) or a dot product, depending on the specific competitive learning algorithm.

The learning in the competitive learning neural network takes place through the modification of the weights between the input and the competitive layer.

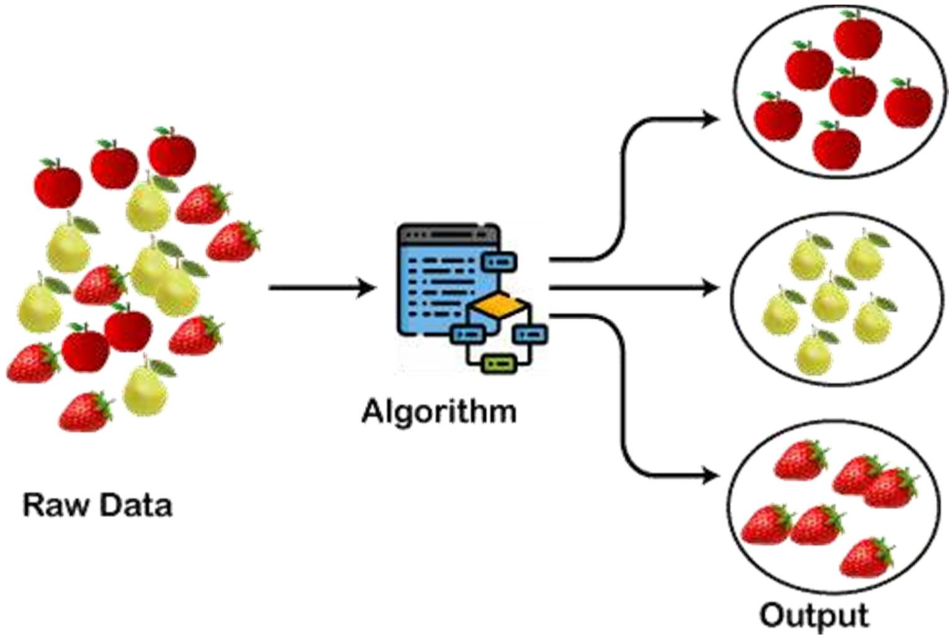
When an input is presented to the network, the weights between the input and the winning neuron are strengthened, and the weights between the input and the other neurons are weakened. This process helps to improve the accuracy of the network in clustering and classification tasks.

Competitive learning neural networks are widely used in pattern recognition, image processing, and data analysis applications. They can be used for clustering data, detecting outliers, and performing classification tasks where the number of classes is not known in advance.

Pattern clustering and feature mapping network

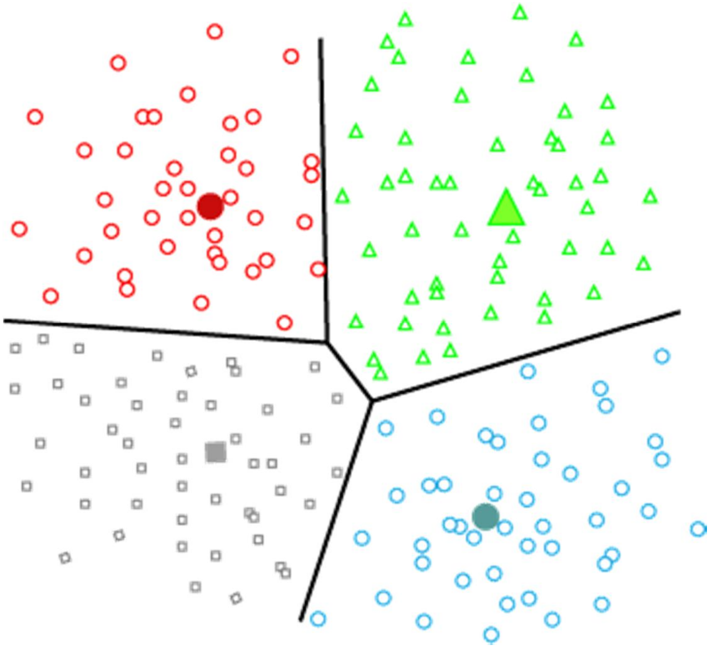
Pattern Clustering

Pattern clustering is a process in artificial intelligence that involves grouping a set of data points or patterns into clusters based on their similarity. It is an unsupervised learning technique that does not require a labeled dataset, and is commonly used in data analysis, image processing, and pattern recognition applications.



The goal of pattern clustering is to identify groups or clusters of patterns that share similar characteristics, such as shape, color, texture, or other features. The patterns in each cluster are more similar to each other than to the patterns in other clusters. The clustering algorithm uses a distance metric to measure the similarity between the patterns, and assigns each pattern to the closest cluster.

There are many different algorithms that can be used for pattern clustering, including k-means, hierarchical clustering, density-based clustering, and fuzzy clustering. Each algorithm has its own strengths and weaknesses, and is suited to different types of data and applications.



Pattern clustering is used in a wide range of applications, including image segmentation, customer segmentation, bioinformatics, and anomaly detection. It can help to identify hidden patterns in data, simplify complex data sets, and provide insights into the underlying structure of the data.

Feature Mapping Network

A feature mapping network is a type of artificial neural network that is used for dimensionality reduction and feature extraction.
It is a type of unsupervised learning algorithm that maps high-dimensional input data to a lower-dimensional space, while preserving the underlying structure and relationships in the data.

The feature mapping network consists of two main layers:
The input layer and the output layer.

The **input layer** contains the input data, which can be a high-dimensional vector or matrix.
The **output layer** contains a lower-dimensional representation of the input data, which can be a vector or matrix of reduced dimensionality.

The feature mapping network uses a set of non-linear transformation functions, such as sigmoid or Gaussian functions, to map the input data to the output layer.
These transformation functions can be trained using various unsupervised learning techniques, such as principal component analysis (PCA), autoencoders, or self-organizing maps (SOM).

Note: The primary goal of a feature mapping network is to reduce the dimensionality of the input data while preserving the important features and relationships in the data.

This can be useful for visualizing high-dimensional data, reducing noise in the data, and improving the performance of other machine learning algorithms that operate on the reduced-dimensional feature space.

Feature mapping networks are widely used in many applications as below:

1. Image processing
2. Speech recognition
3. Natural language processing.
4. They are particularly useful in applications where high-dimensional data is common, and where it is important to identify and extract the most important features and relationships in the data.

Advantages of Feature Mapping:

1. **Improved Model Performance:** Good feature mapping can greatly improve the performance of a machine learning model by providing it with a more suitable representation of the data.
2. **Reduced Dimensionality:** Feature mapping can help to reduce the dimensionality of the data, making it easier to visualize and process.
3. **Improved Interpretability:** By transforming the raw data into a more interpretable form, feature mapping can help to provide insight into the underlying structure of the data and the relationships between features.

Disadvantages of Feature Mapping:

1. **Time-consuming:** Feature mapping can be time-consuming, especially for large and complex datasets, as it requires careful consideration of the data and the selection of appropriate techniques.
2. **Expertise required:** Creating good features requires domain expertise and an understanding of the underlying data and problem.

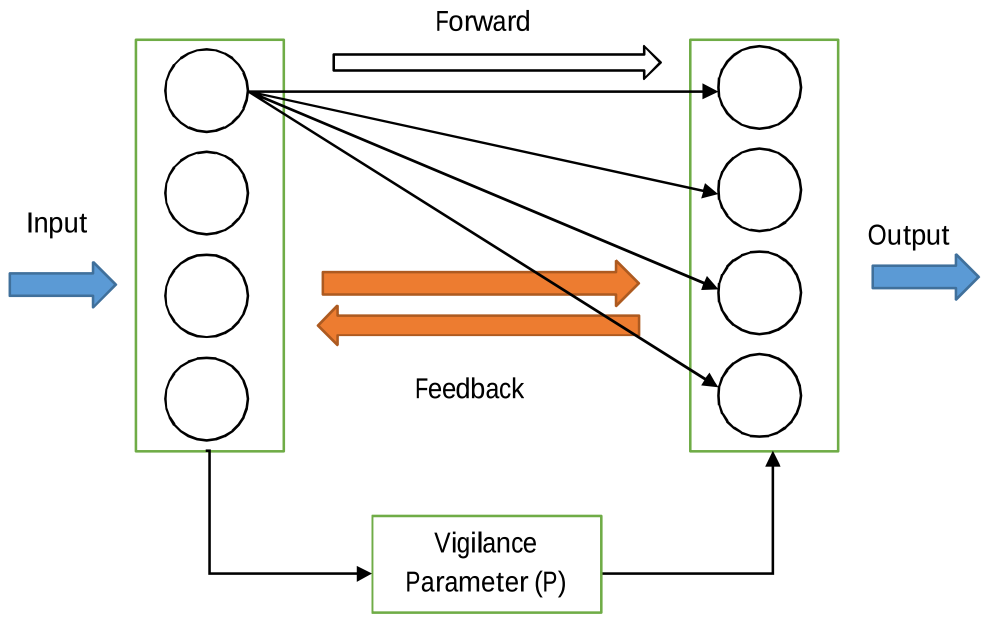
ART networks

Adaptive Resonance Theory (ART) networks are a family of artificial neural networks that are used for pattern recognition and classification tasks in artificial intelligence.

The term “adaptive” and “resonance” used in this suggests that they are open to new learning(i.e. adaptive), where the network needs to maintain stable memories while also being able to learn and adapt to new information.

ART networks are based on a two-layer architecture: **Input layer and Output layer.**

The input layer receives the input data and the output layer generates the response. The key innovation in ART networks is the use of a vigilance parameter, which determines the degree of match between the input data and the stored memories in the network.



This parameter ensures that the network can maintain stable memories while still being able to learn and adapt to new information.

Types of Adaptive Resonance Theory (ART)

1. ART1 – It is the simplest and the basic ART architecture. It is capable of clustering binary input values.
2. ART2 – It is extension of ART1 that is capable of clustering continuous-valued input data.
3. Fuzzy ART – It is the augmentation of fuzzy logic and ART.
4. ARTMAP – It is a supervised form of ART learning where one ART learns based on the previous ART module. It is also known as predictive ART.
5. FARTMAP – This is a supervised ART architecture with Fuzzy logic included.

Advantage of Adaptive Resonance Theory (ART)

A. It exhibits stability and is not disturbed by a wide variety of inputs provided to its network.
B. It can be integrated and used with various other techniques to give more good results.
C. It can be used for various fields such as mobile robot control, face recognition, land cover classification, target recognition, medical diagnosis, signature verification, clustering web users, etc.

D. It has got advantages over competitive learning (like bpnn etc). The competitive learning lacks the capability to add new clusters when deemed necessary.
E. It does not guarantee stability in forming clusters.

Limitations of Adaptive Resonance Theory
Some ART networks are inconsistent (like the Fuzzy ART and ART1) as they depend upon the order in which training data, or upon the learning rate.

Features of ART models

Adaptive Resonance Theory (ART) is a neural network model that is designed to learn and recognize patterns in a dynamic and noisy environment. Some features of the ART model are:

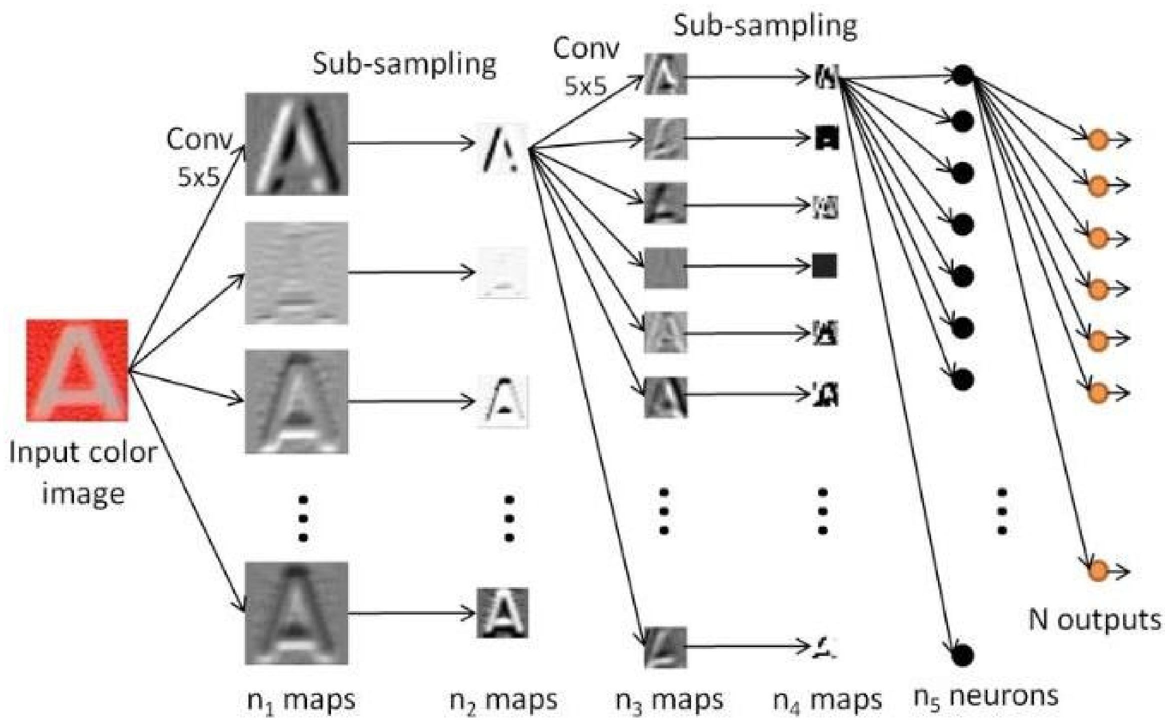
1. **Dynamic Learning:** The ART model can learn and adapt to new patterns in real-time without retraining the entire network. It can dynamically adjust its weights and activation levels to match new input patterns, while preserving the previously learned knowledge.
2. **Top-Down and Bottom-Up Processing:** The ART model incorporates both top-down and bottom-up processing. The top-down signals are used to guide the network's attention and focus on the most salient features of the input pattern, while the bottom-up signals provide sensory input and drive the network's learning.
3. **Vigilance Control:** The ART model includes a vigilance parameter that controls the network's sensitivity to input patterns. The vigilance parameter determines how similar a new input pattern needs to be to a previously learned pattern before it can be recognized as a match.
4. **Stable Category Learning:** The ART model can learn stable categories that can accommodate variations in input patterns. It achieves this by using a Winner-Takes-All (WTA) mechanism that allows it to select the most relevant category for a given input pattern.
5. **Robustness to Noise and Ambiguity:** The ART model is designed to handle noisy and ambiguous input patterns. It can tolerate variations and noise in the input patterns and still recognize them as belonging to a particular category.
6. **Unsupervised Learning:** The ART model is capable of unsupervised learning, which means that it can learn from unlabeled data. This makes it useful in scenarios where labelled data is scarce or not available.

Character recognition using ART network.

The steps involved in using the ART network for character recognition are:

1. **Data Preprocessing:** The input images are preprocessed by normalizing the image size, converting them to grayscale, and resizing them to a standard size. This step is necessary to ensure that the input images are in a consistent format that can be fed into the ART network.
2. **Encoding the Input:** The preprocessed images are encoded into a vector representation that can be fed into the ART network. This is typically done using a feature extraction algorithm such as edge detection, which extracts the relevant features of the image.
3. **ART Network Training:** The encoded vectors are fed into the ART network, which learns to recognize the patterns in the input data. The network is trained using an unsupervised learning algorithm, which means that the network learns to recognize the patterns without any explicit labels.
4. **Testing the Network:** After the network is trained, it can be tested on a set of unseen images to evaluate its performance. The input images are encoded and fed into the network, and the network's output is compared to the expected output to evaluate its accuracy.
5. **Fine-Tuning the Network:** If the network's performance is not satisfactory, it can be fine-tuned by adjusting the network's parameters or by providing additional training data.

Overall, the ART network provides a robust and efficient approach to character recognition tasks, especially in scenarios where labelled data is not available.



Self-Organization Maps (SOM)

Self-Organizing Maps (SOM), also known as Kohonen maps, are a type of artificial neural network that is used for unsupervised learning and data visualization. SOMs are designed to represent high-dimensional data in a lower-dimensional space while preserving the topological relationships between the input data.

The SOM network consists of an input layer, a layer of neurons, and an output layer. The input layer receives the input data, which is typically a high-dimensional vector, while the output layer consists of a two-dimensional grid of neurons. Each neuron in the output layer is connected to every input in the input layer via a set of weights. During the training phase, the SOM network learns to adjust its weights to map the input data onto the output layer while preserving the topological relationships between the input data.

The training process involves two main steps: competition and cooperation. In the competition phase, the input data is presented to the network, and each neuron in the output layer calculates its activation level based on its distance from the input data. The neuron with the highest activation level is known as the winner neuron and is the one that is closest to the input data. In the cooperation phase, the weights of the winner neuron and its neighboring neurons are adjusted to pull the winner neuron closer to the input data.

Once the SOM network is trained, it can be used for data visualization and clustering. The SOM network maps the input data onto the output layer, which can be visualized as a two-dimensional grid of nodes. Similar input data is mapped to neighboring nodes in the output layer, which allows for easy identification of clusters of similar data.

Overall, SOMs are a powerful tool for unsupervised learning and data visualization. They are widely used in a variety of applications, including image processing, data mining, and pattern recognition.

Two Basic Feature Mapping Models

Two basic feature mapping models in artificial neural networks (ANNs) are:

Self-Organizing Maps (SOMs): SOMs, also known as Kohonen maps, are a type of ANN that is used for unsupervised learning and data visualization. SOMs are designed to represent high-dimensional data in a lower-dimensional space while preserving the topological relationships between the input data. During training, the SOM network learns to adjust its weights to map the input data onto the output layer while preserving the topological relationships between the input data. Once the SOM network is trained, it can be used for data visualization and clustering.

Radial Basis Function Networks (RBFNs): RBFNs are a type of ANN that is used for function approximation and classification. RBFNs are composed of three layers: an input layer, a hidden layer, and an output layer. The hidden layer contains radial basis functions, which are used to transform the input data into a high-dimensional feature space. The output layer performs a linear combination of the transformed data to produce the final output. RBFNs are particularly useful for approximating complex non-linear functions.

Overall, SOMs and RBFNs are powerful tools in ANNs for unsupervised learning, data visualization, function approximation, and classification.

Self-Organization Map Algorithm

The SOM algorithm consists of the following steps:

1. **Initialization:** The algorithm begins by initializing a two-dimensional grid of neurons. Each neuron in the grid is assigned a weight vector randomly.
2. **Input data presentation:** The input data is presented to the network one at a time. Each input data point is a high-dimensional vector, typically represented by a row in a data matrix.
3. **Neuron selection:** For each input data point, the neuron with the closest weight vector is selected as the "winning" neuron. The similarity between the input data point and the weight vector of each neuron is typically measured using the Euclidean distance.
4. **Update weights:** The weights of the winning neuron and its neighbors are updated according to a learning rule that pulls them closer to the input data point. The learning rule is typically based on a neighborhood function that determines the extent to which neighboring neurons are updated.
5. **Repeat:** Steps 2-4 are repeated for all input data points, with the weights of the neurons updated after each iteration.
6. **Visualization:** Once the SOM algorithm is complete, the two-dimensional grid of neurons can be visualized as a map of the input data. Similar input data points are mapped to neighboring neurons, allowing for easy identification of clusters of similar data.

The SOM algorithm is widely used in data visualization and clustering applications due to its ability to reduce high-dimensional data into a two-dimensional space while preserving the topological relationships between the input data. It is particularly useful in applications where the structure of the data is unknown, as it allows for the identification of underlying patterns and relationships.

Properties of Feature Map

In the context of artificial neural networks, a feature map is a layer of neurons that receives input from the previous layer and performs a specific type of feature extraction on that input. Here are some properties of feature maps:

1. **Hierarchical feature representation:** Feature maps can be used to extract hierarchical features from the input data. Each layer in the feature map hierarchy extracts more abstract and high-level features from the previous layer.
2. **Spatial relationship preservation:** Feature maps can preserve the spatial relationships between the input features. For example, in a convolutional neural network (CNN), the convolutional layer applies a sliding window over the input data, which preserves the spatial relationship between the pixels in the image.

3. **Non-linear transformations:** Feature maps can perform non-linear transformations on the input data, allowing for the extraction of complex features that would be difficult or impossible to extract using linear transformations.
4. **Parameter sharing:** Feature maps can share parameters between neurons, reducing the number of parameters that need to be learned during training. For example, in a CNN, the same filter is applied to multiple regions of the input data, allowing for parameter sharing and reducing the number of parameters that need to be learned.
5. **Dimensionality reduction:** Feature maps can be used to reduce the dimensionality of the input data, making it easier to process and analyze. For example, in a CNN, the pooling layer reduces the spatial dimensionality of the input data while preserving the feature representation.

Overall, feature maps are a powerful tool in artificial neural networks for extracting hierarchical and non-linear features from input data while preserving spatial relationships and reducing dimensionality. They are widely used in applications such as computer vision, speech recognition, and natural language processing.

Computer Simulations

Computer simulations are models that use computational methods and algorithms to imitate and analyze real-world phenomena, systems, or processes. They are powerful tools for scientific inquiry, engineering design, and decision-making in a wide range of fields, including physics, chemistry, biology, engineering, economics, social sciences, and more.

There are several types of computer simulations, including:

1. **Discrete event simulations:** These simulations model systems where events occur at discrete times, such as queueing systems, transportation networks, or manufacturing processes.
2. **Agent-based simulations:** These simulations model the behavior of individual agents or entities and how they interact with each other and their environment. Examples include ecosystems, social networks, or financial markets.
3. **Continuous simulations:** These simulations model systems that change continuously over time, such as fluid dynamics or weather patterns.
4. **Monte Carlo simulations:** These simulations use random sampling techniques to model complex systems and estimate probabilities or expected outcomes. They are commonly used in finance, risk analysis, and engineering design.

Computer simulations are valuable because they can be used to explore complex systems, test hypotheses, and generate predictions that would be difficult or impossible to obtain through experimentation or observation alone. They can also be used to optimize system performance, improve design, or inform policy decisions.

However, computer simulations also have limitations and uncertainties.

- The accuracy and reliability of a simulation depend on the quality of the data and assumptions used to build the model.

- Complexity and variability of the system being modeled. Therefore, it is important to carefully validate and verify simulations and consider their limitations and uncertainties when interpreting results.

Learning Vector Quantization

Learning Vector Quantization (LVQ) is a supervised learning algorithm used in artificial intelligence for classification and pattern recognition tasks.

It is a type of artificial neural network that learns to classify input data into predefined categories based on labeled training data.

LVQ works by building a set of prototype vectors, which represent each category of input data. During training, the algorithm updates these prototype vectors based on the similarity between each input data point and the prototypes. The similarity between the input data and the prototypes is typically measured using the Euclidean distance.

The LVQ algorithm consists of the following steps:

1. **Initialization:** The algorithm begins by randomly initializing a set of prototype vectors, one for each category of input data.
2. **Input data presentation:** The input data is presented to the network one at a time, along with its corresponding label or category.
3. **Neuron selection:** For each input data point, the prototype vector that is closest to the input data point is identified.
4. **Prototype update:** The prototype vector that is closest to the input data point is updated to be more similar to the input data point, while the prototype vectors for the other categories are updated to be less similar to the input data point. This update is typically performed using a learning rate that decreases over time and a correction term that depends on the difference between the input data point and the prototype vector.
5. **Repeat:** Steps 2-4 are repeated for all input data points, with the prototype vectors updated after each iteration.
6. **Classification:** Once the LVQ algorithm is trained, it can be used to classify new input data points into categories based on the distance between the input data point and the prototype vectors.

LVQ is a simple and effective algorithm for classification and pattern recognition tasks, particularly in applications where the number of input data points is relatively small and the number of categories is relatively large. It is also computationally efficient, making it suitable for real-time applications. However, it is sensitive to the initial position of the prototype vectors and the choice of learning rate and correction term, which can affect the accuracy and convergence of the algorithm.

Adaptive Pattern Classification

Adaptive pattern classification is a type of machine learning in artificial intelligence that allows a system to learn and adapt to new patterns and input data over time. It is particularly useful for tasks such as image recognition, speech recognition, and natural language processing, where the input data may vary in complexity and may not always be fully predictable.

Adaptive pattern classification algorithms typically involve the use of artificial neural networks, which are designed to mimic the structure and function of the human brain. These networks consist of interconnected nodes or neurons, each of which receives input signals and processes them using a mathematical function to produce an output signal. The output signal from each neuron is then transmitted to other neurons in the network, allowing the network to perform complex computations and learn from input data.

Adaptive pattern classification algorithms typically involve the following steps:

1. **Training:** The algorithm is trained on a set of labeled input data, where each data point is associated with a particular output label or category. During training, the algorithm adjusts the weights and connections between the neurons in the network to minimize the error between the predicted output and the actual output for each input data point.
2. **Testing:** Once the algorithm is trained, it is tested on a separate set of input data to evaluate its performance. The algorithm uses the weights and connections learned during training to predict the output label or category for each input data point in the test set.
3. **Adaptation:** Over time, the input data may change or new patterns may emerge, requiring the algorithm to adapt to new input data. Adaptive pattern classification algorithms can adjust the weights and connections between the neurons in the network in response to new input data, allowing the network to learn and recognize new patterns over time.

Adaptive pattern classification algorithms can be implemented using various types of artificial neural networks, including feedforward networks, recurrent networks, and convolutional networks. They can also incorporate other techniques such as reinforcement learning, unsupervised learning, and deep learning to improve performance and adaptability. Overall, adaptive pattern classification is a powerful approach to machine learning that allows systems to learn and adapt to new input data and patterns over time.