Pattern Association, Pattern recognition and Function Approximation

By

Mausam Chouksey

Pattern association

- An *associative memory* is a brainlike distributed memory that learns by *association.*
- Association takes one of two forms:
  - *Autoassociation :*In autoassociation, a neural network is required to *store* a set of patterns (vectors) by repeatedly presenting them to the network. The network is subsequently presented a partial description or distorted (noisy) version of an original pattern stored in it, and the task is to *retrieve* (*recall)* that particular pattern.
  - *heteroassociation* :Heteroassociation differs from autoassociation in that an arbitrary set of input patterns is *paired* with another arbitrary set of output patterns. Autoassociation involves the use of unsupervised learning, whereas the type of learning involved in heteroassociation is supervised.

Cont'd…

- Let x_k denote a *key pattern* (vector) applied to an associative memory and y_k denote a *memorized pattern* (vector). The pattern association performed by the network is described by

$$\mathbf{x}_k \rightarrow \mathbf{y}_k, \quad k = 1, 2, \ldots, q$$

- where q is the number of patterns stored in the network.The key pattern x_k acts as a stimulus that not only determines the storage location of memorized pattern y_k, but also holds the key for its retrieval.

Cont'd...

- In an autoassociative memory, $y_k = x_k$, so the input and output (data) spaces of the network have the same dimensionality.

- In a heteroassociative memory, $y_k \neq x_k$, hence, the dimensionality of the output space in this second case may or may not equal the dimensionality of the input space.

Cont’d…

- There are two phases involved in the operation of an associative memory:
  - *Storage phase*, which refers to the training of the network
  - *Recall phase*, which involves the retrieval of a memorized pattern in response to the presentation of a noisy or distorted version of a key pattern to the network.

Cont’d…

- Let the stimulus (input) x represent a noisy or distorted version of a key pattern x_j.
- This stimulus produces a response (output) y.
- For perfect recall, we should find that $y = y_j$, where y_j is the memorized pattern associated with the key pattern x_j
- When $y \neq y_j$ for $x = x_j$, the associative memory is said to have made an *error* in recall.

Pattern Recognition

- Humans are good at pattern recognition. We receive data from the world around us via our senses and are able to recognize the source of the data.
- For example, we can recognize the familiar face of a person even though that person has aged since our last encounter, identify a familiar person by his or her voice on the telephone despite a bad connection, and distinguish a boiled egg that is good from a bad one by smelling it.
- Humans perform pattern recognition through a learning process; so it is with neural networks.

Cont'd…

- *Pattern recognition* is formally defined as *the process whereby a received pattern signal is assigned to one of a prescribed number of classes (categories).*
- A neural network performs pattern recognition by first undergoing a training session, during which the network is repeatedly presented a set of input patterns along with the category to which each particular pattern belongs. Later, a new pattern is presented to the network that has not been seen before, but which belongs to the same population of patterns used to train the network.
- The network is able to identify the class of that particular pattern because of the information it has extracted from the training data. Pattern recognition performed by a neural network is statistical in nature, with the patterns being represented by points in a multidimensional *decision space*.
- The decision space is divided *into regions, each one of which is associated* with a class. The decision boundaries are determined by the training process. The construction of these boundaries is made statistical by the inherent variability that exists within and between classes.

Function Approximation

It is a concept used in machine learning and mathematics that can be explained in simple terms

How Does It Work?

1. **Data Collection:** First, you gather data. For example, you might collect data on daily temperatures and the number of ice creams sold each day.
2. **Finding a Pattern:** Next, you look for a pattern in that data. You might notice that on hotter days, more ice creams are sold. However, the exact relationship might not be a straight line; it could be curved or have ups and downs.
3. **Creating a Model:** Function approximation involves creating a model (like a mathematical equation) that tries to mimic or approximate the relationship between temperature and ice cream sales. This model won't be perfect, but it will give you a good idea of how one affects the other.
4. **Making Predictions:** Once you have your model, you can use it to make predictions. For example, if you know the temperature for a future day, you can use your model to estimate how many ice creams will be sold.

Why is it Useful?

Function approximation is useful because it helps us make sense of complex relationships in data. Instead of needing to know every detail about how two things interact, we can use a simpler model to make predictions and decisions. This is widely used in various fields, such as economics, engineering, and even everyday life, like predicting how much gas you might need for a trip based on distance.