Glossary

Neural Networks

Refer to “Exploring the Potential of AI for Control Design” (page 8)

SHIBAHARA Anna, Motion Control R&D Sect., Basic Technology R&D Center, Engineering Div.

1 What is a Neural Network?

1. 1 Definition

In artificial intelligence (AI), a neural network (NN) is a machine learning approach that mimics the learning mechanisms of living organisms. Some of the functions of the neural networks in the biological brain are mimicked [Note 1)] as mathematical models on computers to implement learning functions.

There are many types of NN configurations. This article only explains feedforward NNs (NNs in which data flows in only one direction). Please note that the configurations of the various elements of the NNs described below are just examples, and there are other configurations as well. The explanation mostly covers a general description. Some elements related to computation are omitted here.

Note 1) Often, practical NNs are not exact mimics of biological brains. The following treats NNs as practical mathematical models, not strictly as biological brains.

1. 2 NN Mechanism

This section explains how an NN can act as a substitute for a control system that receives input from two sensors. Fig. 1 shows an example of an NN configuration. Each element indicated by a circle represents a computation unit, or node. A group of nodes arranged vertically in this figure is called a layer. The first layer is called the input layer, and the final layer is called the output layer. Layers between these two are called hidden layer. Any number of hidden layer can exist.

A sensor signal input to the input layer passes

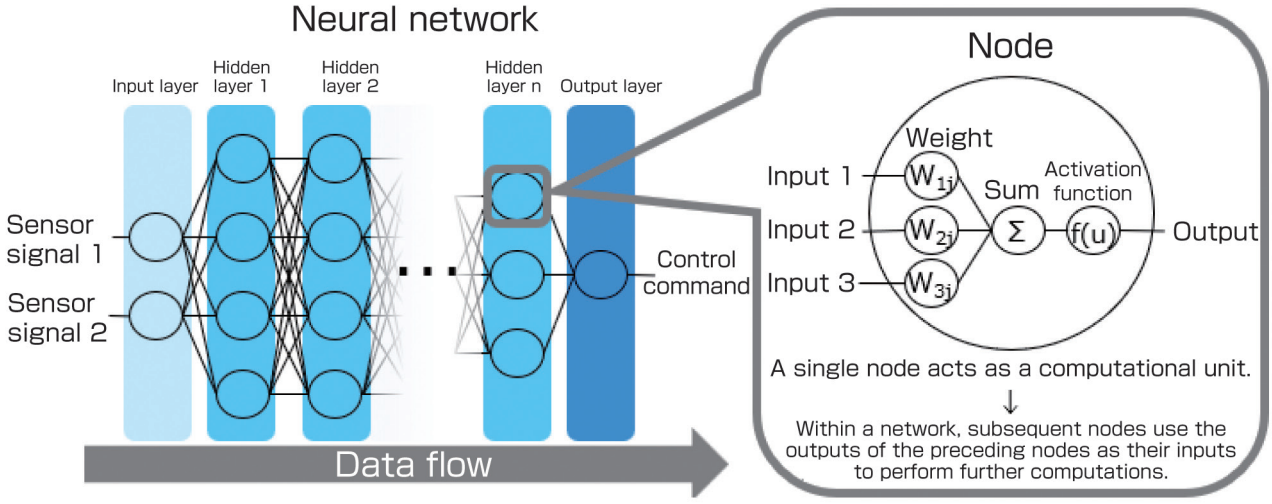

Fig. 1 NN mechanism

through the hidden layer before being output from the output layer. During this process, calculations are performed in individual nodes. These calculations, repeated in each node, transform the sensor signal into a control command by the time it reaches the output layer.

Each node multiplies the value of each input by its respective weight and sums them. The result is then subjected to an activation function (a function that determines the output of the NN). Using a non-linear activation function enables the NN to produce non-linear outputs in response to the inputs.

1. 3 Learning

NN learning refers to updating the weights of individual nodes. This process often uses a back propagation (BP) method. This section explains BP. BP updates the weights of the NN's nodes sequentially, starting from the last layer and working backwards. This method enables the NN to learn even if it has many hidden layer.

Let me explain using the example shown in Section 1.2. The difference between the control command output by the NN and the optimal control command is called a loss. Based on this value, the NN updates the weights of the nodes in each layer sequentially to achieve learning, as shown in Fig. 2.

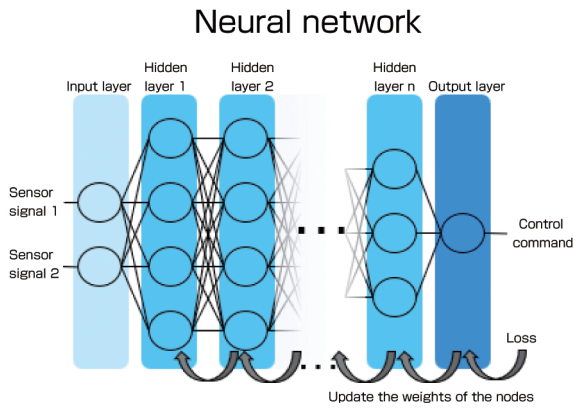

Fig. 2 Illustration of NN learning

The loss calculation method depends on the NN application and other factors. Some methods may enable NN learning even when a direct comparison with the optimal control command is impossible. One example of such a case is the use of NNs for reinforcement learning.

This section explains loss calculation using Deep Q Network (DQN), one of the most famous reinforcement learning algorithms, as an example of using an NN for reinforcement learning. In general, DQN calculates a loss using the equation below:

$$\mathbb{E}[(r+\gamma max_{a'}Q'-Q)^2]$$

($\mathbb{E}[\ ]$ indicates the expected value of [ ]; r is the reward; γ is a parameter that controls the length of the interval over which the reward is considered; $max_{a'}Q'$ is the maximum expected reward based on the current input; and Q is the expected reward based on recent control results.)

Some elements have been omitted from the equation above to facilitate explanation. In any case, calculating the loss using an equation based on this concept enables the NN to learn independently of direct comparison with optimal control command.

2 Applications of NNs

2. 1 Functional Diversity

An NN can be configured in various ways. Connecting the appropriate layers for a given application allows the NN to perform different functions. For instance, convolutional neural networks (CNNs) are well-suited to image processing, whereas recurrent neural networks (RNNs) are ideal for time-series processing.

2. 2 Typical Applications

NNs can be used for a variety of applications. Typical applications include:

(1) Supervised learning: classification problems, prediction problems
(2) Unsupervised learning: dimension reduction, clustering, data generation
(3) Reinforcement learning: control, planning

NNs can be used for a single application, but they can also be combined to achieve even more complex operations.