Neural Network

Epoch IIT Hyderabad

Ankit Saha
AI21BTECH11004

30 Jul 2022

1. Introduction

Neural networks are the basic paradigm of deep learning, which is a subset of machine learning. The objective of a neural network is to learn the pattern between a given set of input and output, so that it can predict the output for any unseen input data. Neural networks accomplish this through a network of artificial neurons arranged in layers with connections between them.

A neural network tries to mimic the working of a human brain. The artificial neurons in a neural network, just like biological neurons, receive and transmit signals from and to other neurons through the connections between them. The output at each neuron is calculated using a linear combination of all the inputs received by that neuron using a set of weights that is fixed for every neuron in the network.

2. Perceptrons

Perceptrons were the earliest models of artificial neurons used. They are binary classifiers. Suppose we are given m labelled samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)$ where $x_i \in \mathbb{R}^n, y_i \in \{-1, 1\} \;\; \forall i \in [m]$.
The output of a perceptron with weights $\mathbf{w} \in \mathbb{R}^n$ and bias $b \in \mathbb{R}$ is given by

$$y = \operatorname{sgn}\left(\mathbf{w}^\top \mathbf{x} + b\right) \tag{2.1}$$

The goal of training the perceptron is to find $\mathbf{w}$ and b such that

$$y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 0 \qquad \forall i \in [m] \tag{2.2}$$

This is possible if the data is linearly separable using the following perceptron training algorithm:

i) Initialize $\mathbf{w} = \mathbf{w}^{(0)}$ and $b = b^{(0)}$
ii) Iterate over all inputs in the training set and check whether $y_i \left(\mathbf{w}^{(t)^\top} \mathbf{x}_i + b^{(t)}\right) \geq 0$
iii) If yes, then continue to the next data point
iv) Else, update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \delta y_i \mathbf{x}_i \tag{2.3}$$

$$b^{(t+1)} = b^{(t)} + \delta y_i \tag{2.4}$$

where δ is the step size

3. Multilayer Perceptron Networks

A simple perceptron works when the data is linearly separable. However, if the data is not linearly separable, multilayer perceptrons (MLP) are used. The neurons in a MLP network (MLPN) need not necessarily be perceptrons. A combination of different types of artificial neurons can be used.

A MLPN consists of several neurons arranged in at least three layers. The first layer is called the input layer where the initial inputs are received. The last layer is called the output layer which gives the final output. The layers in between them are called hidden layers.

A neuron in a layer is connected to one or more neurons in the adjacent layers. Signals are transmitted from layer to layer through these connections. The output of one neuron becomes the input of the next neuron in the next layer.

The goal is to find a neural network architecture that best fits the data, i.e., minimizes the generalization error. A common choice for the error function in regression is the average least squares error. Let the ouput of the MLPN with weights $\mathbf{w}$ for input $\mathbf{x}$ be denoted by $f(\mathbf{w}, \mathbf{x})$. Then the average least squares error is given by

$$J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - f(\mathbf{w}, \mathbf{x}_i))^2 \tag{3.1}$$

The optimal $\mathbf{w}$ that minimizes the above cost function can be found by using the steepest descent algorithm where the gradients at each step are calculated using a technique known as backpropagation. Backpropagation involves the repeated use of the chain rule for calculating gradients at each level.

$$z = \mathbf{w}^{\top} \mathbf{x}_i = \sum_{j=1}^{n} w_j x_{ij} \tag{3.2}$$

$$\implies \frac{\partial z}{\partial w_j} = x_{ij} \tag{3.3}$$

$$\implies \frac{\partial f(z)}{\partial w_j} = \frac{\partial f(z)}{\partial z} \frac{\partial z}{\partial w_j} = f'(z) x_{ij} \tag{3.4}$$

Note that the output function f needs to be differentiable

4. Types of Neural Networks

Deep Neural Network (DNN)

Deep neural networks are neural networks with a very high number of hidden layers and connections. Deep neural networks generally need to be regularized because even if one neuron has a near-zero derivative, then all subsequent neurons will have near-zero derivatives as well while backpropagating. This is known as the vanishing gradient problem.

$$J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - f(\mathbf{w}, \mathbf{x}_i))^2 + \lambda \|\mathbf{w}\|^2 \tag{4.1}$$

This prevents the weights from becoming too large and the derivatives from becoming too small.

Recurrent Neural Network (RNN)

Unlike feedforward neural networks, recurrent neural networks exhibit temporal dynamic behaviour. The output at a particular time step is used as an input for the next time step. In other words, the outputs of a neuron are being fed back to it. They are trained using an algorithm known as backpropagation through time.

$$J(\mathbf{w}) = \frac{1}{mT} \sum_{i=1}^{m} \sum_{t=1}^{T} (y_i(t) - f(\mathbf{w}, \mathbf{x}_i(1:t)))^2 + \lambda \|\mathbf{w}\|^2 \tag{4.2}$$

Convolutional Neural Network (CNN)

Convolutional neural networks are multilayer perceptron networks with low connectivity. The output is given by the convolution of an input and a kernel, hence the name. CNNs are mostly used for image and audio processing. A CNN mimics the human visual cortex where each neuron in the convolutional layer only processes a specific region in the input data.

5. Questions

i) List some artifical neurons other than perceptrons.
ii) Write the hyperbolic tangent function in terms of sigmoid.
iii) Which of the above mentioned neuron functions are not differentiable?
iv) What happens to the perceptron training algorithm when the data is not linearly separable?
v) Give a practical example where neural networks have been used successfully.

6. Answers

i)
- Sigmoid

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

- Hyperbolic tangent

$$\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$

- Rectified Linear Unit (ReLU)

$$f(z) = \max(0, z)$$

- Leaky Rectified Linear Unit (Leaky ReLU)

$$f(z) = \max(\alpha z, z) \qquad \alpha \ll 1$$

- Exponential Linear Unit (ELU)

$$f(z) = \begin{cases} z & z \geq 0 \\ \alpha\left(e^{z}-1\right) & z < 0 \end{cases} \qquad \alpha > 0$$

ii)

$$\begin{aligned}
\tanh(z) &= \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} && (6.1)\\
&= \frac{e^{2z}-1}{e^{2z}+1} && (6.2)\\
&= \frac{e^{2z}}{e^{2z}+1} - \frac{1}{e^{2z}+1} && (6.3)\\
&= \frac{1}{1+e^{-2z}} - \frac{1}{1+e^{-(-2z)}} && (6.4)\\
&= \sigma(2z) - \sigma(-2z) && (6.5)\\
&= \sigma(2z) - (1-\sigma(2z)) && (6.6)\\
&= 2\sigma(2z) - 1 && (6.7)
\end{aligned}$$

iii) The ReLU, the Leaky ReLU and the ELU (for $\alpha \neq 1$) are all not differentiable at $z = 0$
iv) When the data is not linearly separable, the perceptron training algorithm ends up in an infinite loop, which is a major drawback of the algorithm.
v) Neural networks have been very successful in handwritten character recognition.