

1 Introduction

1.1 What are neural networks?

Neural networks are functions implemented as software. This means that they are not physical objects, but rather a set of algorithms that are programmed into a computer. These algorithms are inspired by the way that neurons in the human brain work, but they are not a perfect imitation.

As said, neural networks are functions

$$y = f(x, w), \quad (1)$$

where x is the input data, y is the output and w are parameters, called weights.

Neural networks are trained iteratively using a set of examples x_t, y_t , adjusting the weights w until the network can accurately predict the output for a given input. This process is called supervised learning.

$$w_{k+1} = g(w_k, x_t, y_t), \quad (2)$$

where w_0 is a random value.

2 Example: ADALINE

ADALINE (Adaptive Linear Neuron or later Adaptive Linear Element) is a single layer neural network with multiple nodes where each node accepts multiple inputs and generates one output. Given the following variables as:

- x is the input vector
- w is the weight vector
- n is the number of inputs
- θ is a constant
- y is the output of the model, defined as

$$y = \left(\sum_{i=1}^n x_i w_i \right) + \theta \quad (3)$$

If we further assume that $w_0 = \theta$ and $x_0 = 1$, then we can define

$$y = \sum_i x_i w_i. \quad (4)$$

2.1 Learning rule

The learning rule used by ADALINE is the LMS ("least mean squares") algorithm, a special case of gradient descent.

Define

1. ϵ is the learning rate

2. o is the desired output or target
3. $E = (o - y)^2$ is the square of error.

The LMS algorithm updates the weights by

$$w^{k+1} = w^k + \epsilon (o - y) x, \quad (5)$$

where w^0 is a random value. This learning rule minimizes E , and is the stochastic gradient descent update for linear regression.

3 Frameworks

PyTorch and TensorFlow are two popular deep learning frameworks that can be used to build and train neural networks. They provide a variety of tools and features that make it easier to develop and deploy neural network models.

Here are some of the benefits of using frameworks like PyTorch and TensorFlow:

1. Ease of use: These frameworks provide high-level APIs that make it easy to define and train neural networks. This can save a lot of time and effort, especially for beginners.
2. Flexibility: These frameworks are very flexible and can be used to build a wide variety of neural network architectures. This makes them suitable for a variety of tasks, such as image classification, natural language processing, and speech recognition.
3. Performance: These frameworks are designed to be efficient and can be used to train large neural networks on large datasets.
4. Community support: There is a large and active community of users and developers for these frameworks. This means that there is plenty of documentation and support available, and it is easy to find help if you need it.

Overall, PyTorch and TensorFlow are powerful tools that can be used to build and train neural networks. They are a good choice for beginners and experienced developers alike.

Ultimately, the best framework for you will depend on your specific needs and preferences. If you are a beginner, PyTorch is a good choice because it is easier to learn. If you need a framework that is highly efficient for training large neural networks, TensorFlow is a good choice.