

Introduction to Neural Network

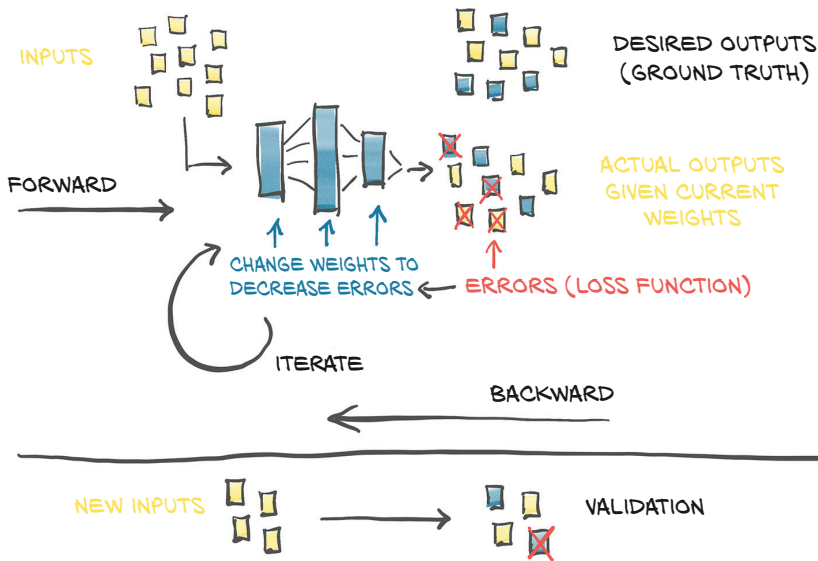
TF4063

Fadjar Fathurrahman

Program Studi Teknik Fisika
Institut Teknologi Bandung

The materials are taken from: Eli Stevens. Deep Learning with PyTorch, Chapter 6.

THE LEARNING PROCESS



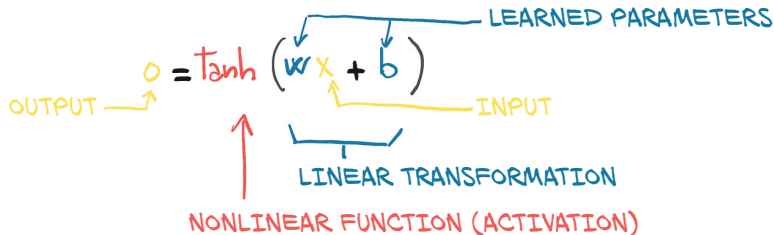
Neural network

- ▶ Neural networks are mathematical entities capable of representing complicated functions through a **composition** of simpler functions.
- ▶ The initial models of neural networks were inspired by neuroscience. Modern artificial neural networks bear only a slight resemblance to the mechanism of neurons in the brain.
- ▶ The building blocks of these complicated functions is the neuron, which is a linear transformation of the input followed by the application of a fixed nonlinear function, referred to as activation function:

$$o = f(wx + b)$$

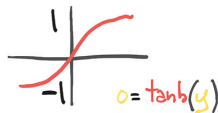
where x : input, o : output, w : weight, b : bias. The input and output can be scalar or vector, weight can be scalar or matrix, and bias can be scalar or vector.

THE "NEURON"



LEARNED

$$\begin{aligned} w &= 2 \\ b &= 6 \end{aligned}$$



$$18 \rightarrow 2 \times 18 + 6 = 42 \rightarrow \tanh(42) = 1$$

$$-2.79 \rightarrow 2 \times (-2.79) + 6 = .042 \rightarrow \tanh(.042) = 0.3969$$

$$-10 \rightarrow 2 \times (-10) + 6 = -14 \rightarrow \tanh(-14) = -1$$

Multilayer network

- ▶ A multilayer neural network, is made up of a composition of functions:

$$x_1 = f(w_0x + b_0)$$

$$x_2 = f(w_1x_1 + b_1)$$

.....

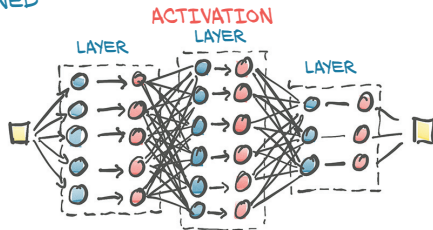
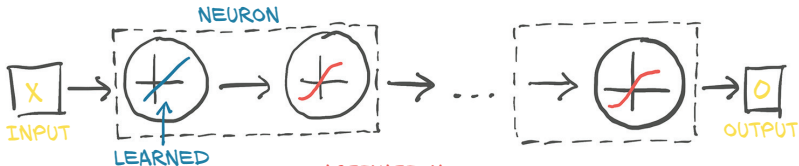
$$y = f(w_nx_n + b_n)$$

- ▶ To train the network, we need to choose a loss function and minimize it:
 - ▶ mean squared error (MSE)
 - ▶ absolute error

A NEURAL NETWORK

$$O = \tanh\left(w_n\left(\dots \tanh\left(w_2\left(\tanh\left(w_1 x + b_1\right) + b_2\right)\right)\dots + b_n\right)\right)$$

Diagram illustrating the equation for the output O of a neural network layer. The equation is a nested \tanh function. The input x is multiplied by the weight w_1 and added to the bias b_1 . This result is then passed through a \tanh activation function, and the output is added to b_2 . This process repeats for n layers. The final output O is the result of the n -th layer's \tanh function applied to the weighted sum of the previous layer's output and bias b_n . The weights w_i and biases b_i are labeled as LEARNED PARAMETERS.



Python code

```
https://github.com/deep-learning-with-pytorch/dlwpt-code  
https://github.com/Apress/  
beginning-anomaly-detection-using-python-based-dl
```