Foundations of Machine Learning in Python

Moritz Wolter

August 29, 2022

High-Performance Computing and Analytics Lab

Overview

Neural networks

Estimation, Overfitting and Regularization

Classification

Neural networks

The wonders of the human visual system

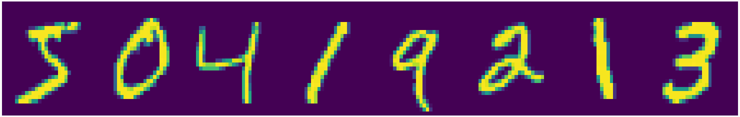
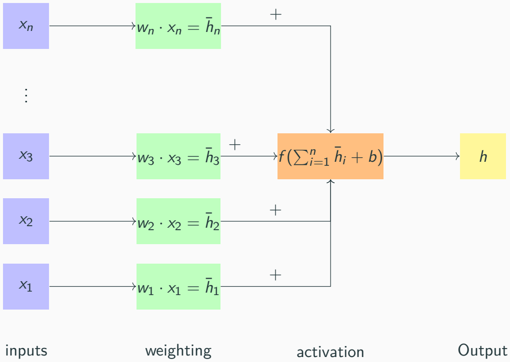


Figure: Most humans effortlessly recognize the digits 5 0 4 1 9 2 1 3.

The perceptron

How can we teach computers to recognize digits?

Use perceptrons to mimic biological neurons loosely.

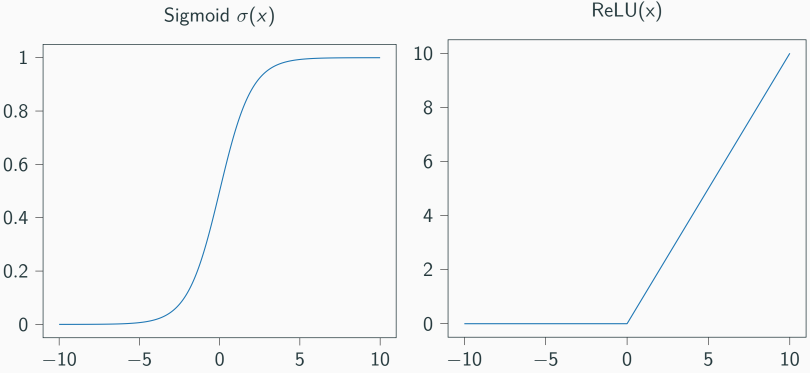


Formally a single perceptron is defined as:

$$f(\mathbf{w}^T \mathbf{x}) = h \tag{1}$$

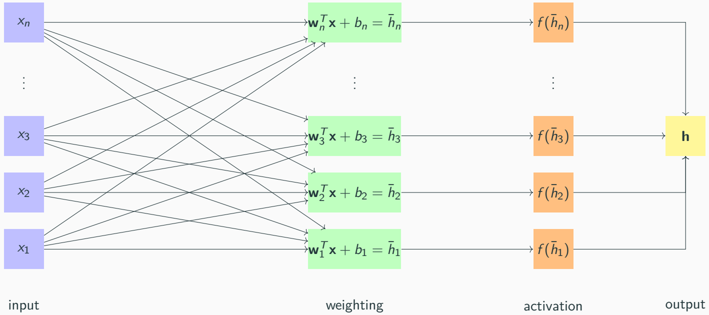
The activation function f

Two popular choices for the activation function f.



Arrays of perceptrons

Let's extend the definition to cover an array of perceptrons:



Every input is connected to every neuron, in matrix language, this turns into

$$\bar{\mathbf{h}} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{2}$$

$$\mathbf{h} = f(\bar{\mathbf{h}}). \tag{3}$$

The loss function

To choose weights for the network, we require a quality measure. We already saw the mean squared error cost function,

$$C_{\mathsf{mse}} = \frac{1}{2} \sum_{k=1}^{n} (\mathbf{y}_k - \mathbf{h}_k)^2 = \frac{1}{2} (\mathbf{y} - \mathbf{h})^T (\mathbf{y} - \mathbf{h}) \tag{4}$$

This function measures the squared distance from each desired output. $\mathbf{y}$ denotes the desired labels, and $\mathbf{h}$ the network output.

The gradient of the mse-cost-function

Both the mean squared error loss function and our dense layer are differentiable.

$$\frac{\partial C_{\text{mse}}}{\partial \mathbf{o}} = \mathbf{o} - \mathbf{y} = \triangle_{\text{mse}} \tag{5}$$

The $\triangle$ symbol will re-appear. It always indicates incoming gradient information from above.

The gradient of a dense layer

The chain rule tells us the gradients for the dense layer[Nie15]

$$\delta \mathbf{W} = [f'(\bar{\mathbf{h}})]\mathbf{x}^T, \ldots \tag{6}$$

Good news! Jax can take care of these computations for you!

The gradient of a dense layer

The chain rule tells us the gradients for the dense layer[Nie15]

$$\delta \mathbf{W} = [f'(\bar{\mathbf{h}})]\mathbf{x}^T, \ldots \tag{6}$$

Good news! Jax can take care of these computations for you!

You can choose to optimally verify these equations by completing the deep learning project.

Derivatives of our activation functions

TODO

Perceptrons can learn functions

TODO

Multi-layer networks

TODO

Backpropagation

TODO

Estimation, Overfitting and Regularization

Denoising a signal

TODO

Classification

The cross-entropy loss

TODO

MNIST digit

Modified National Institute of Standards and Technology database [**dumoulin2016guide**]

References

[Nie15] Michael A Nielsen. *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, USA, 2015.