Machine Learning and Pattern Recognition Practice
Session V: Neural Networks

Martin Palazzo

Universite de Technologie de Troyes
Universidad Tecnologica Nacional Buenos Aires

mpalazzo@frba.utn.edu.ar

December 15, 2022

Overview

Perceptron

Parametrized supervised learning

optimization problem in supervised learning

$$\hat{y} = f_w(x)$$

$$\hat{w} = \underset{w}{\operatorname{argmin}} \; L(y, \hat{y}) = \underset{w}{\operatorname{argmin}} \; L(y, f_w(x))$$

Perceptron classifier

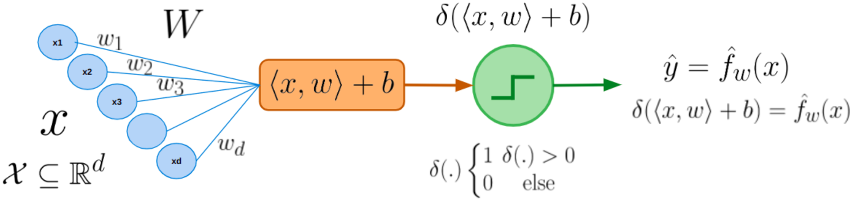


Figure: The perceptron pipeline

The perceptron model

The model get d-dimensional input vectors where each input variable is linearly combined to a weight vector w. After the linear combination $\langle x, w \rangle$ an activation function is used to discretize the output value.

Activation function

Activation function

activation function

$$f_i = \sigma_i(W_i X_{i-1} + b_i)$$

where σ is the activation function or "neuron" which input is a linear combination of weights and random variable X. There are multiple activation functions such as

- Identity
- Relu
- Sigmoid
- TanH

Activation function

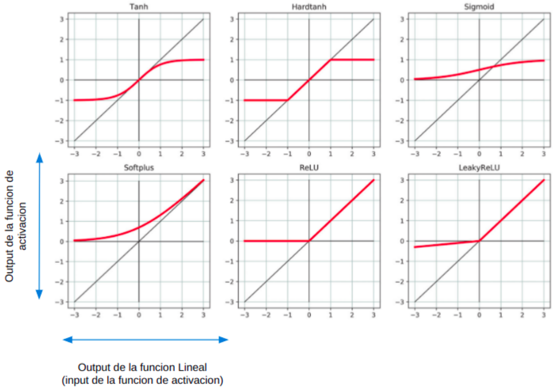


Figure: The landscape of activation functions

Some activation functions saturate when $x \to \infty$ and others not.

Activation function

sigmoid

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

TanH

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1-e^{-2x}}{1+e^{-2x}}$$

Softmax

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for \ i = 1, 2, \dots, K$$

Softmax

$$Relu(z) = max(0, z)$$

Network architectures

single neuron and single layer neural network

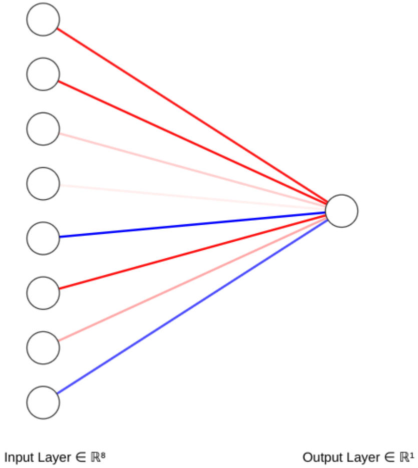


Figure: Neural network of 1 neuron and 1 layer

two neuron and single layer neural network

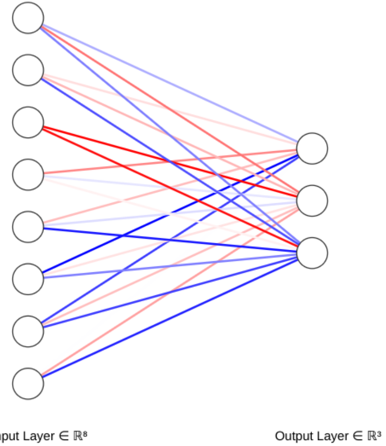
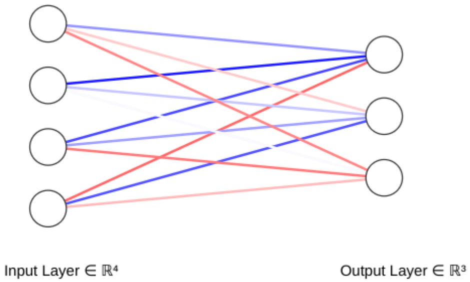


Figure: Neural network of 2 neuron and 1 layer

Matrix notation to Neural Nets

if $x \in \mathbb{R}^d$ with $d = 4$ and a output label $z \in \mathbb{Z}^p$ with $p = 3$ then
$\mathbf{W}X + b = z$

$$\sigma\left(\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right) = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$



Multilayer perceptron

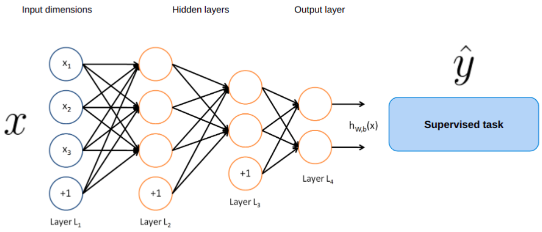


Figure: Neural network of multiple hidden MLP layers

Multilayer perceptron stacked layers equation

$$y = \sigma(W_n(\ldots\sigma(W_2(\sigma(W_1 x + b_1) + b_2))\cdots + b_n))$$

Loss functions

Loss functions

Classification: cross entropy

Binary classification

$$-(y \log(p) + (1 - y) \log(1 - p))$$

Multiclass:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

Mean Squared Error

For regression

$$\sum_{i=1}^{N} |x_i - y_i|$$

Loss landscape

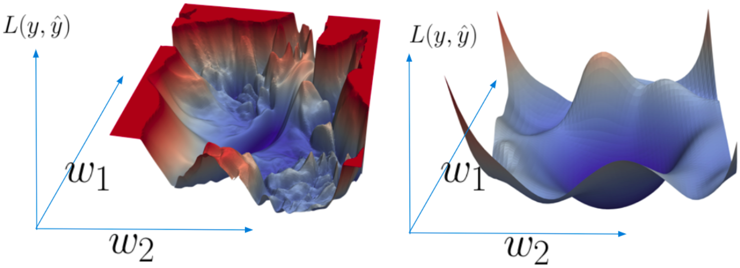


Figure: Loss landscape. Source: https://www.jeremyjordan.me

The loss landscape is conditioned to the dataset **X**, the model architecture and the type of loss function. Via a non-convex optimization problem using Stochastic Gradient Descent a $\hat{w}$ coordinate weight parameter is determined to solve the optimization problem in a local minima.

Loss across training epochs

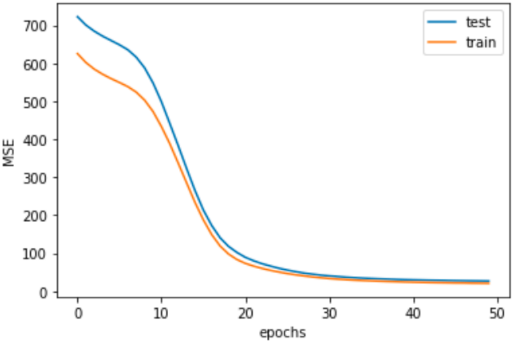


Figure: Loss

At each epoch the weight parameters are updated in order to minimize the value of the objective loss function.

- Stevens, E., Antiga, L., & Viehmann, T. (2020). Deep learning with PyTorch. Manning Publications.
- Shawe-Taylor, J., Cristianini, N. (2004). Kernel methods for pattern analysis. Cambridge university press.