

# Murillo da Silveira Grübler

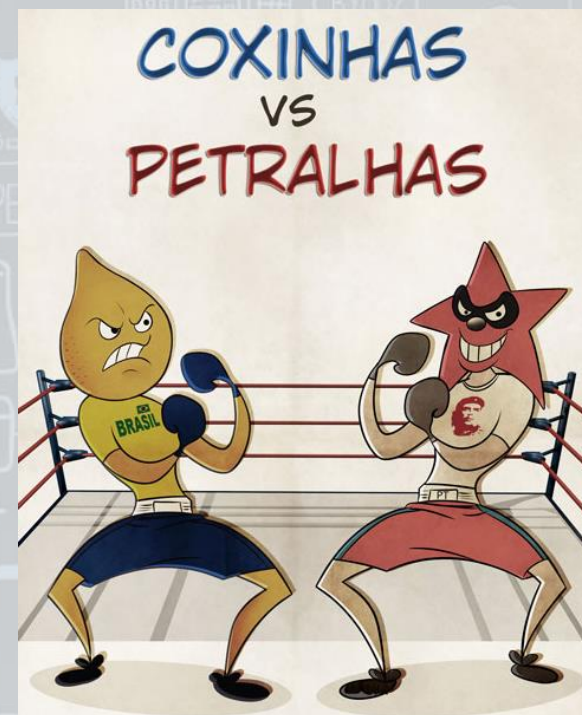
Vamos iniciar com a teoria...



# Em quais problemas as Redes Neurais atuam?

Revista Preferida	Comediante Favorito	A favor das privatizações	Definição
Veja	Danilo Gentili	Sim	Coxinha
Carta Capital	Gregorio Duvivier	Não	Petralha

- Verificar se uma pessoa pode receber um empréstimo, analisando o histórico;
- Prever a duração da estadia de um paciente admitido em um hospital;
- Diagnosticar doenças;





# O que é uma Rede Neural Artificial?

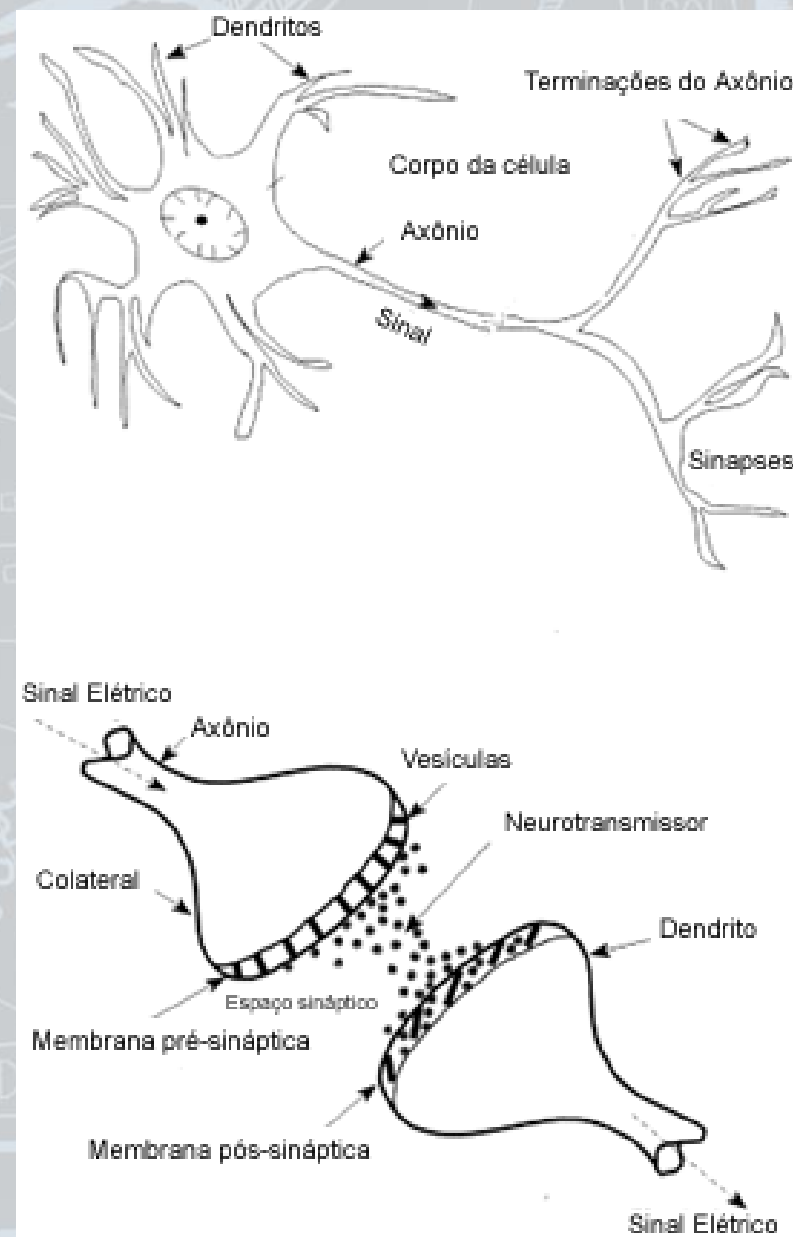
Uma Rede Neural Artificial (RNA) pode ser definida, segundo Basheer e Hajmeer (2000), como uma estrutura complexa densamente interligada por elementos de processamento simples (neurônio), adaptativos que possuem a capacidade de realizar operações como cálculos em paralelo, para processamento de dados e representação de conhecimento.

# O que é uma Rede Neural Artificial?

- A RNA possui uma série de equações matemáticas que simulam o processo biológico de aprendizado e memorização e pode ser desenvolvido utilizando diferentes algoritmos de treinamento (TU, 1996);
- A RNA é uma abstração da rede neural biológica, servindo de modelo para o aprendizado de máquinas e resoluções de problemas complexos (Rojas, 1996);

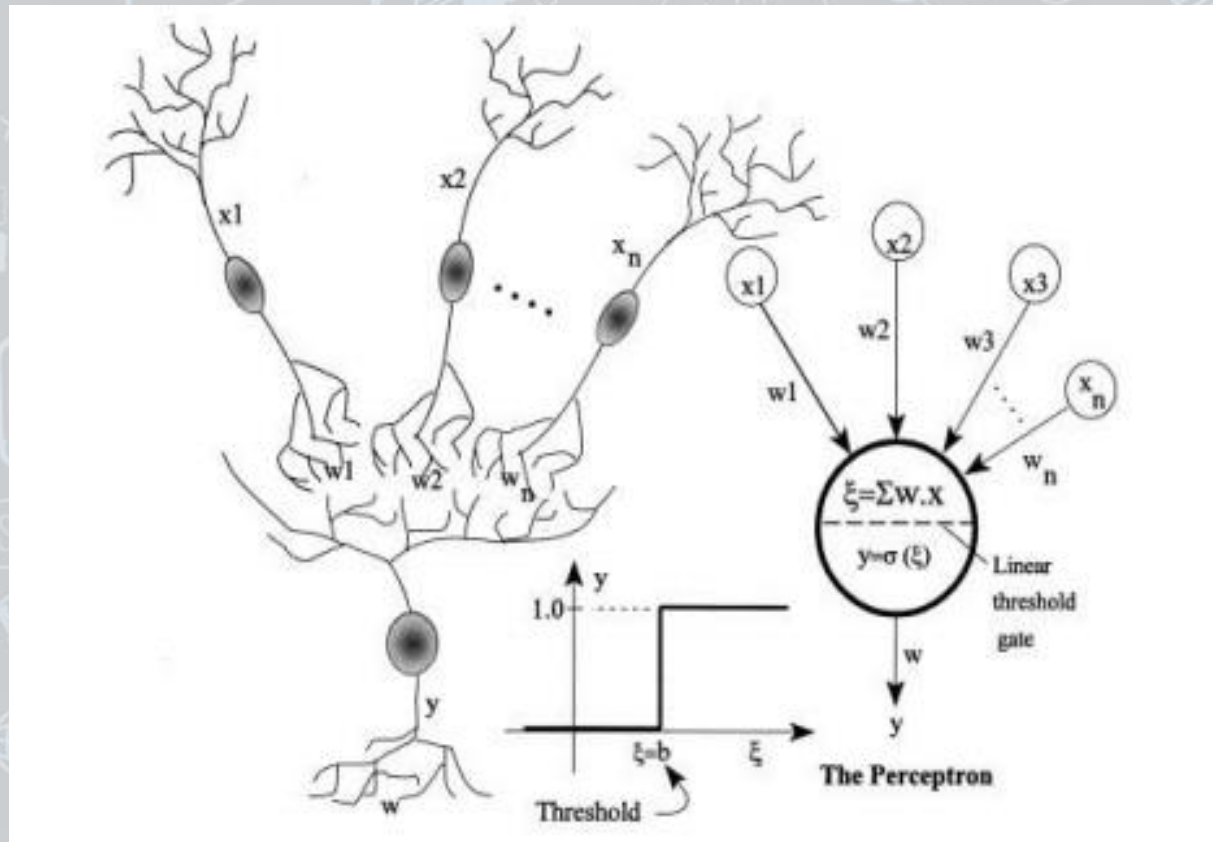
# Neurônio biológico

O sistema nervoso possui arquiteturas globais das mais diversas complexidades. Todavia, todos são compostos de módulos de base semelhantes, chamados de células neurais ou neurônios, desempenhando funções diferentes (ROJAS, 1996).



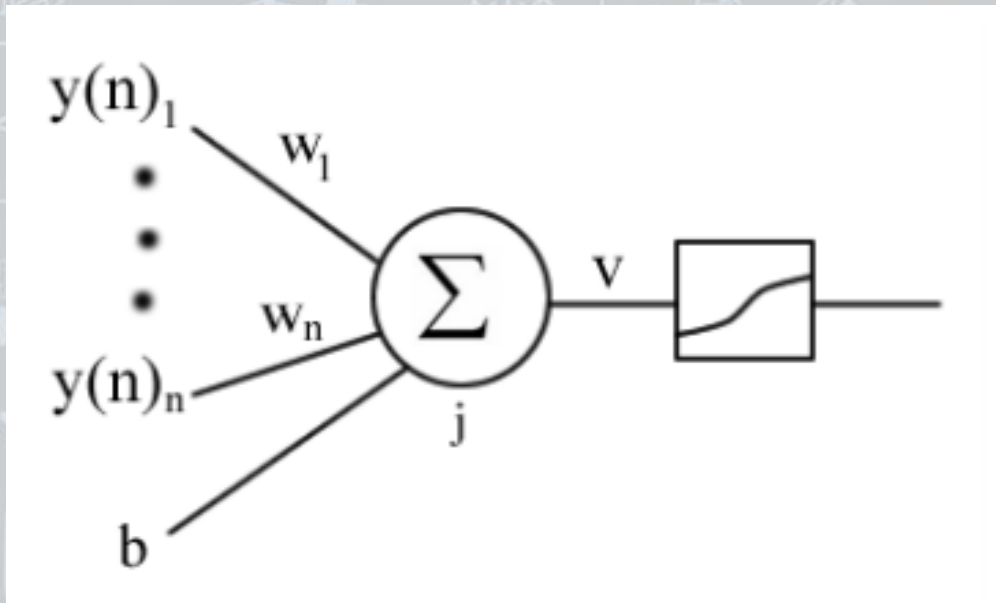


# Relação entre o neurônio biológico e artificial

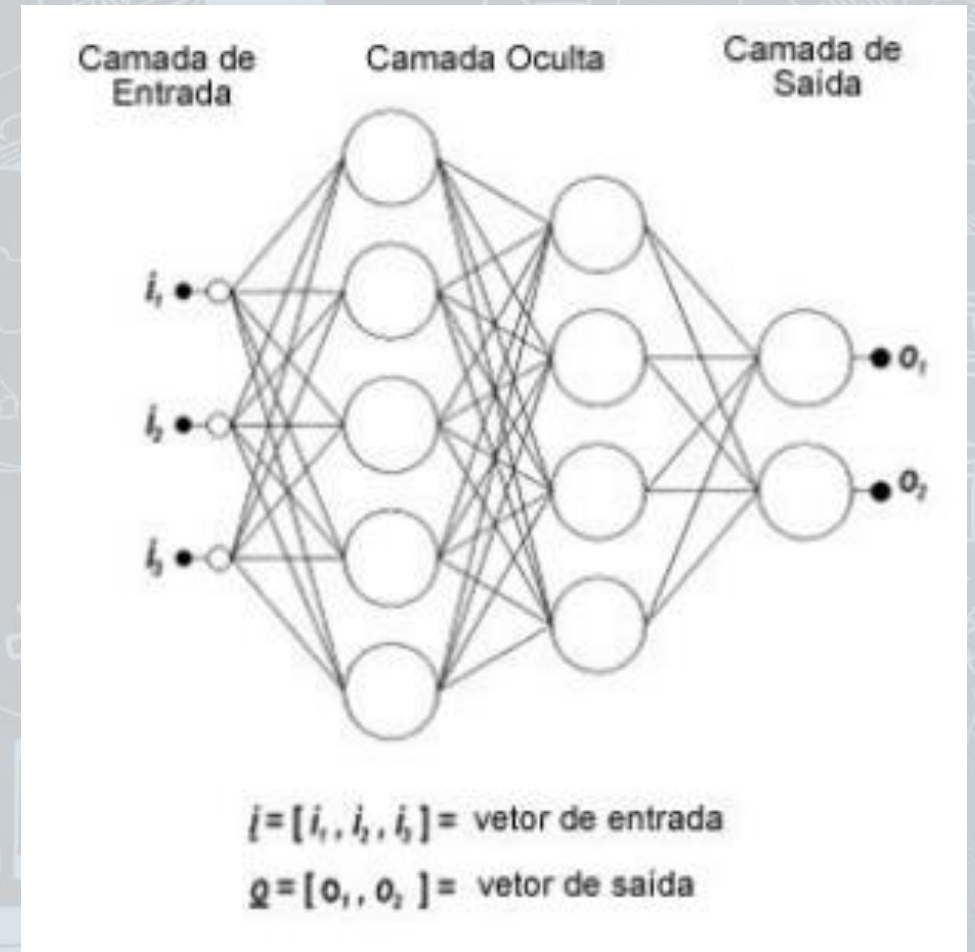


- A relação entre as redes é que ambas possuem axônio, dendrito e comunicam-se por sinapses (BASHEER; HAJMEER, 2000)

# RNA Multilayer Perceptron



$$v_j = \sum_{i=0}^m w_i y_i + b$$





# Função de Ativação

- Sigmóide Logística

- Saída entre [0, 1]

$$y_j = \frac{1}{1 + e^{-v_j}}$$

- Tangente Hiperbólica

- Saída entre [-1, +1]

$$\phi(x) = \tanh(x)$$

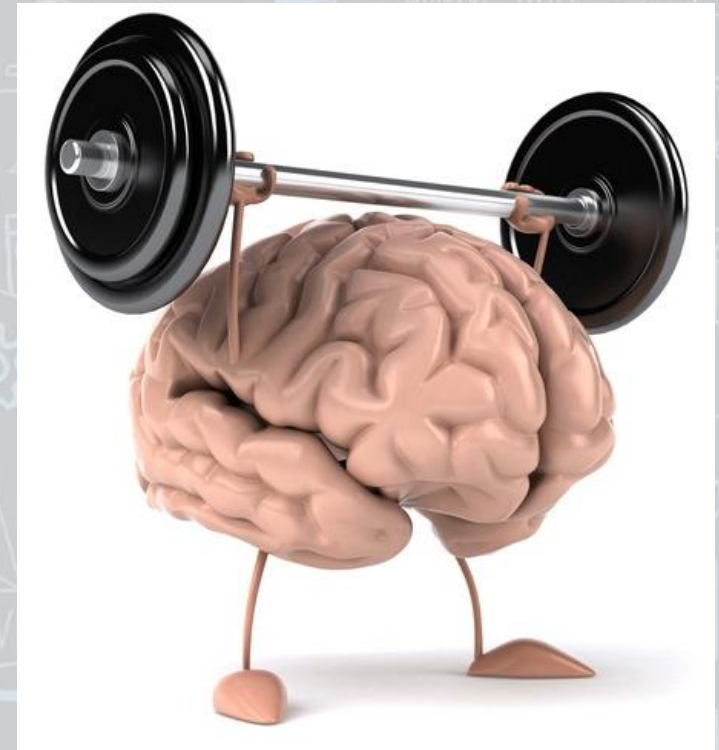
- Softmax

- Saída entre [0, 1] e a soma de todos é igual a 1

$$\phi_i = \frac{e^{z_i}}{\sum_{j \in \text{group}} e^{z_j}}$$

# Treinamento

- Backpropagation;
- Algoritmo Genético;
- Particle Swarm Optimization

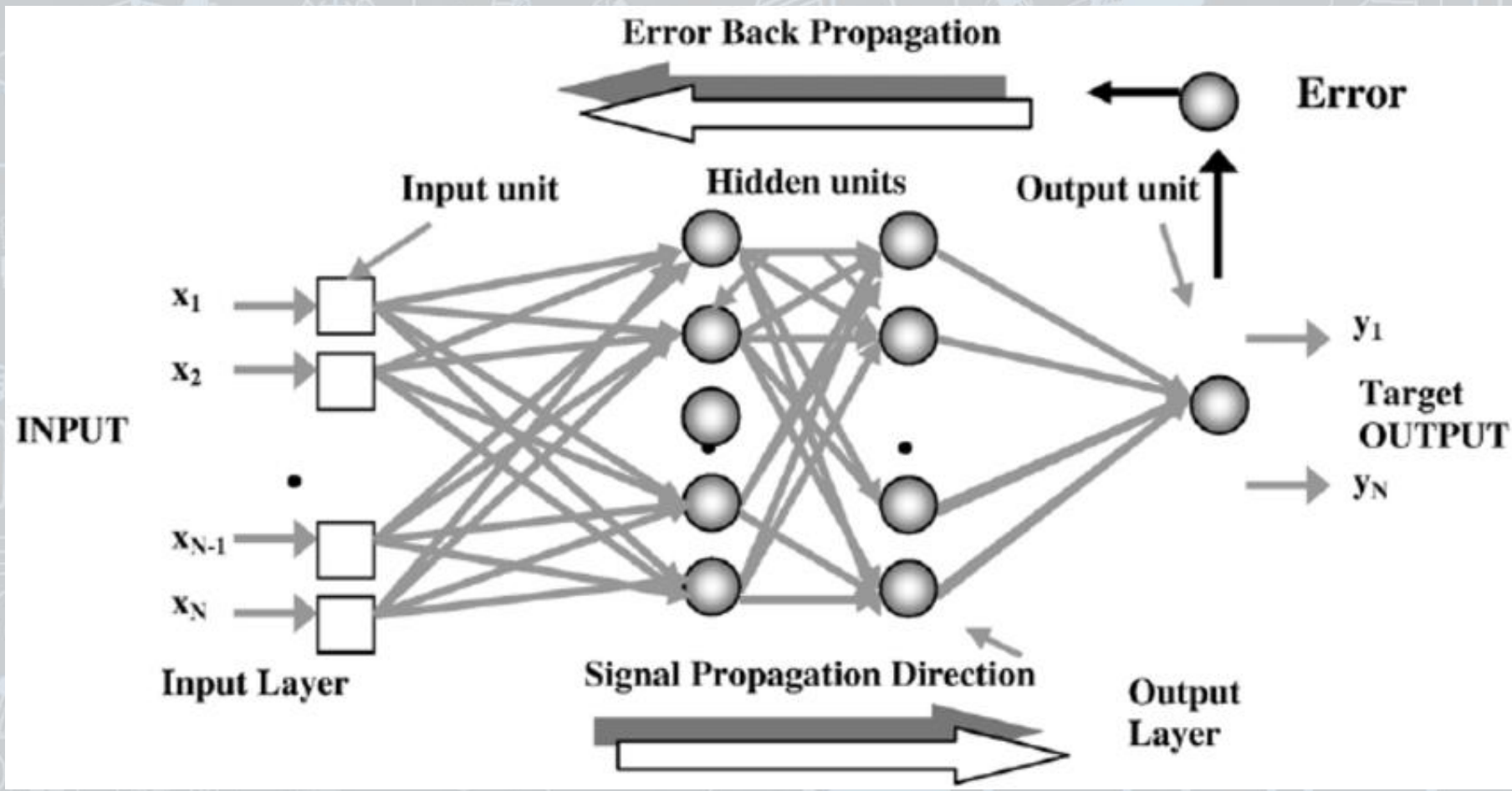


# Backpropagation

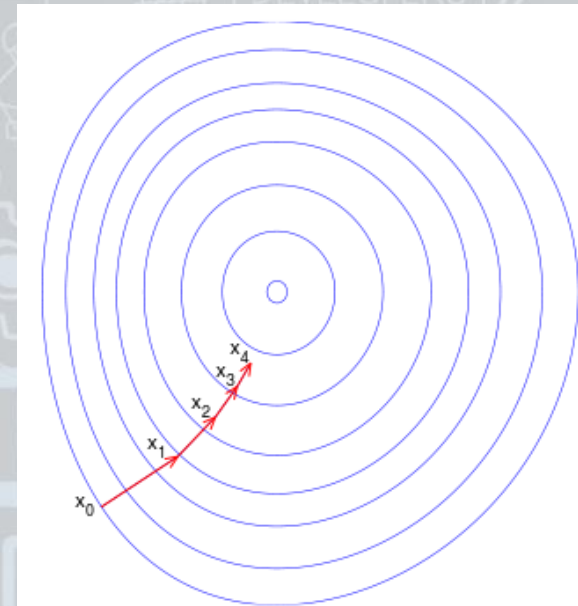
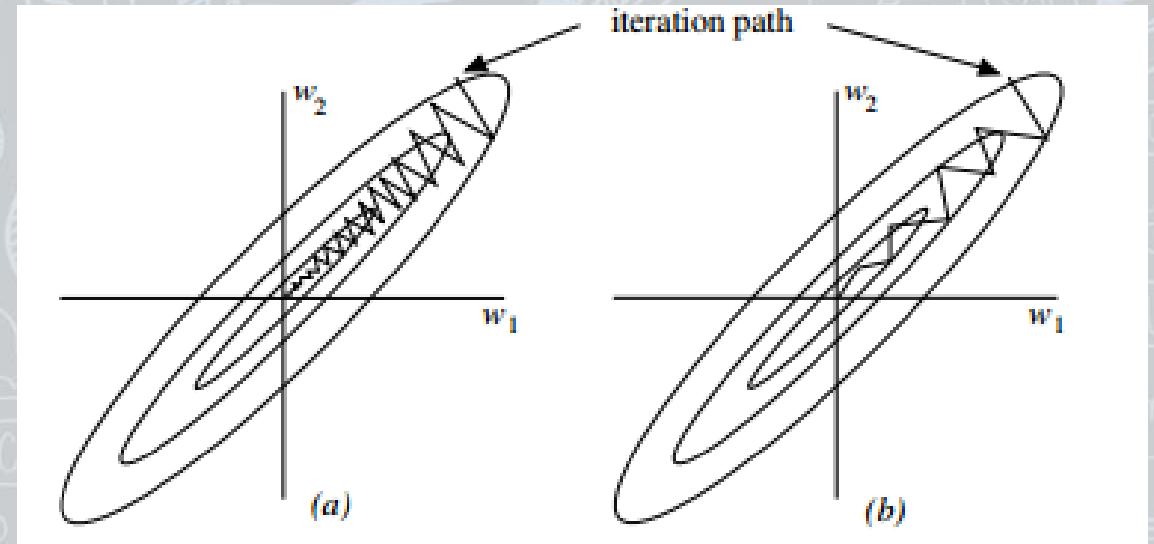
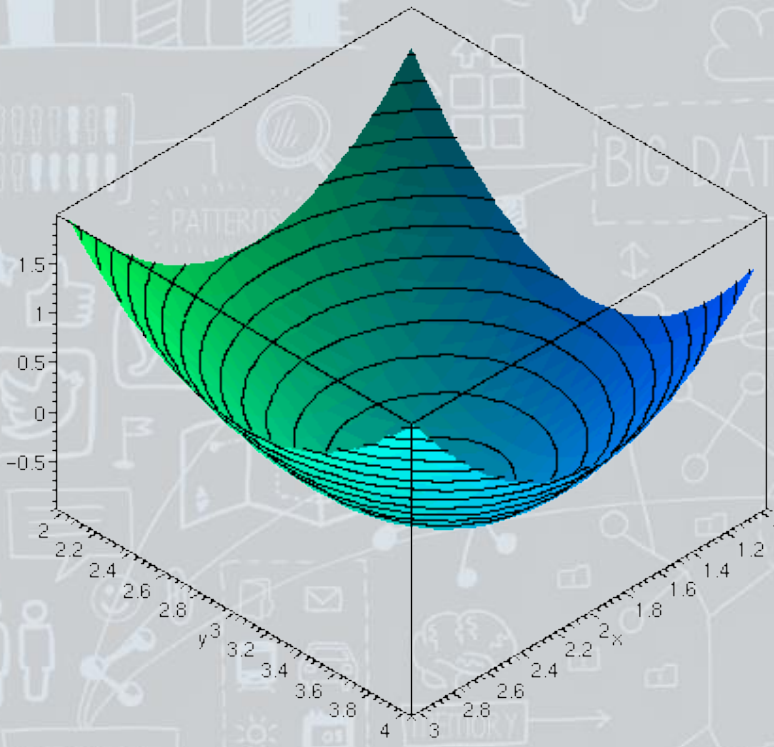
- O algoritmo realiza o ajuste dos pesos da rede para minimizar as diferenças entre a saída atual e a saída desejada;
- É uma técnica rápida que necessita apenas da taxa de aprendizagem e o momentum.
- O backpropagation é um método baseado no gradiente descendente, o que significa que este algoritmo não garante encontrar um mínimo global e pode estagnar em soluções de mínimos locais, onde ficaria preso indefinidamente;



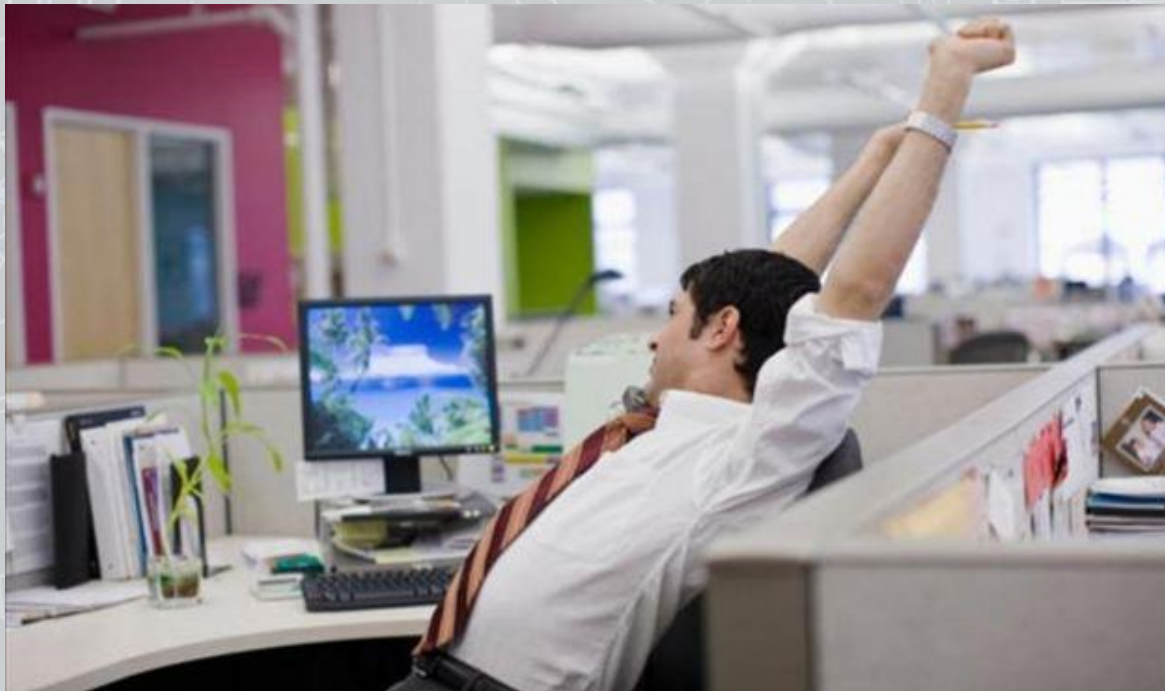
# Backpropagation



# Backpropagation

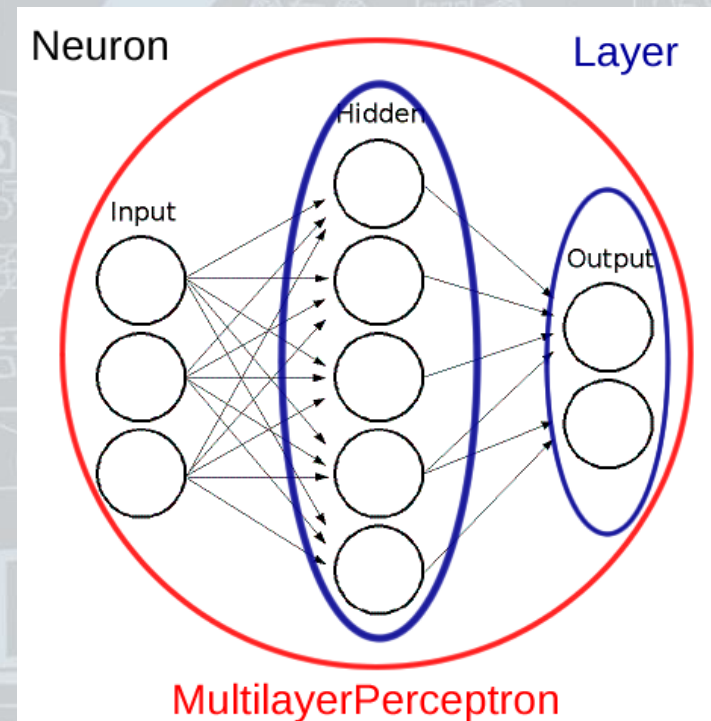
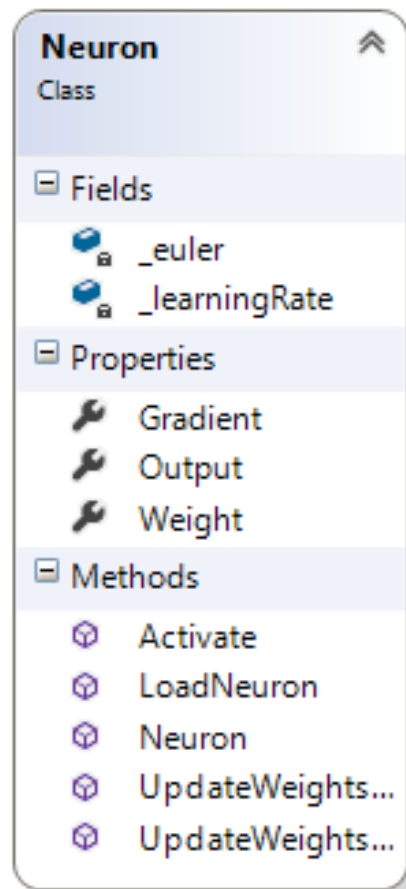
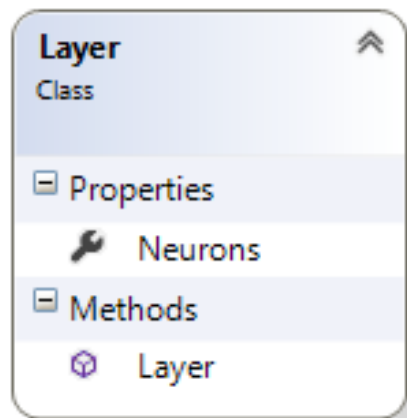
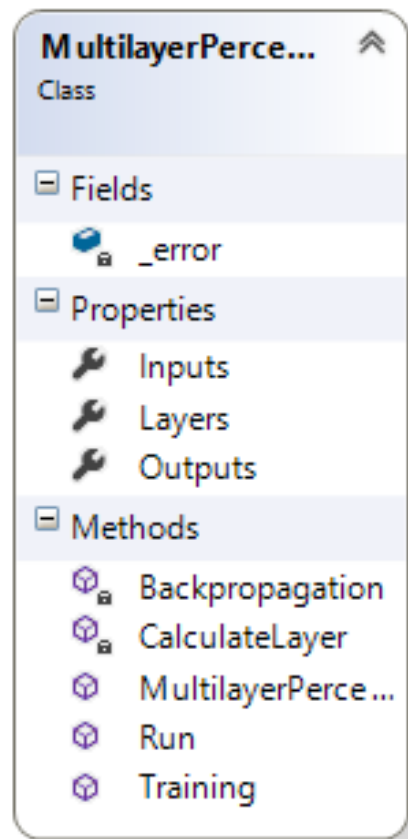


Hora do código...





# Diagrama de Classe



# Classe MultilayerPerceptron

void Training(double[,] inputs, double[] outputs)  
-> Backpropagation

```
public void Training(double[,] inputs, double[] outputs)
{
    long iteration = 0;
    double error;
    do
    {
        error = 0;
        int numberTrain = inputs.Length / inputs.GetLength(1);
        for (int i = 0; i < numberTrain; i++)
        {
            double tagert = outputs[i];
            double[] inputTest = new double[inputs.GetLength(1)];
            for (int j = 0; j < inputs.GetLength(1); j++)
                inputTest[j] = inputs[i, j];

            double output = Run(inputTest).FirstOrDefault(k => k > 0);
            Backpropagation(inputTest, output, tagert);

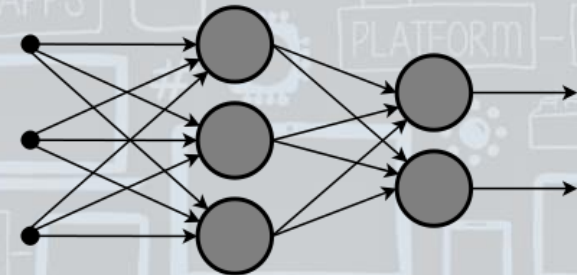
            double delta = tagert - output;
            error += Math.Pow(delta, 2);
            System.Diagnostics.Debug.WriteLine(Math.Round((output * 100) / tagert, 2) + "%");
        }

        iteration++;
    } while (error >= _error);

    System.Diagnostics.Debug.WriteLine("Quantidade de iterações: " + iteration);
}
```

double[] Run(double[] inputs)  
-> Feedforward

```
public double[] Run(double[] inputs)
{
    foreach (Layer layer in Layers)
        inputs = CalculateLayer(layer, inputs);
    return inputs;
}
```





# Demonstração

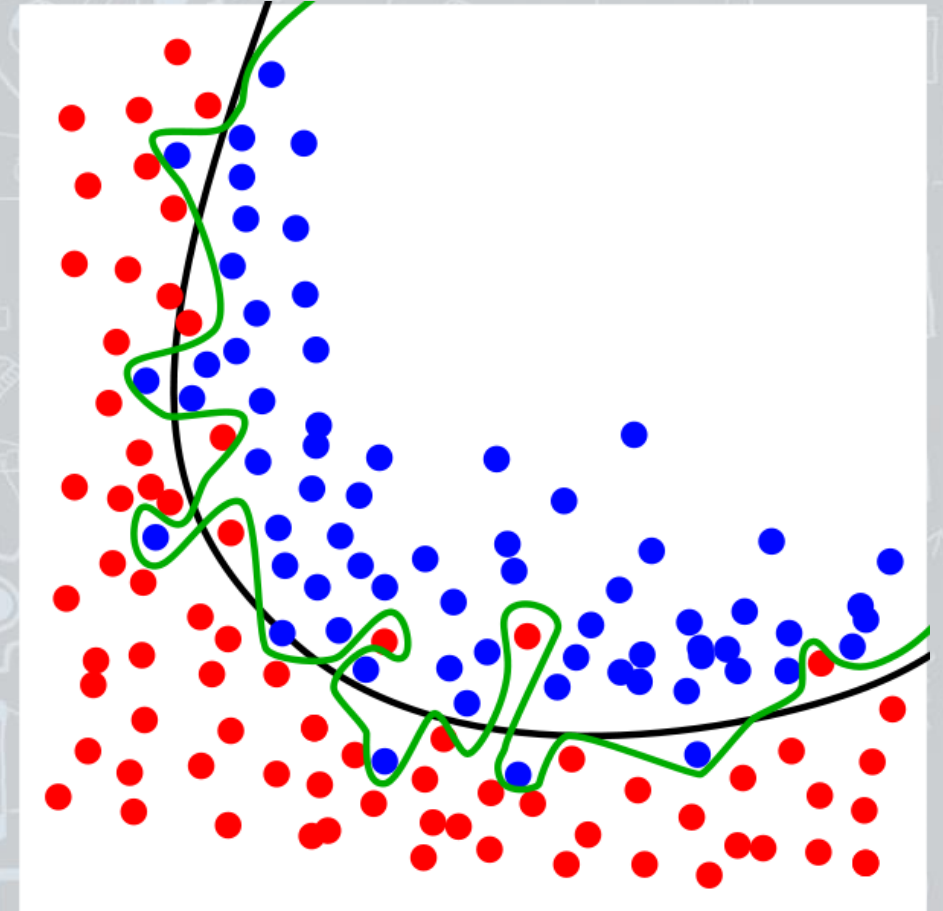
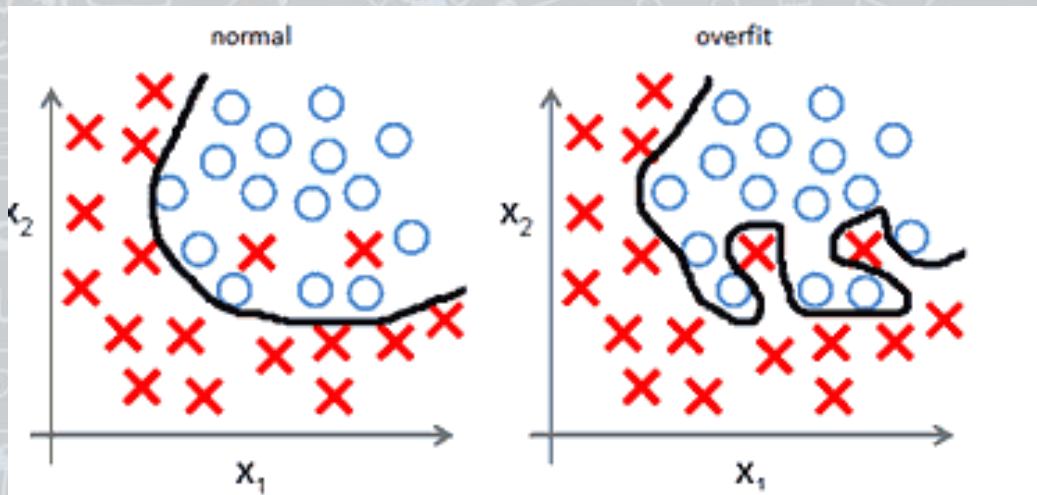


# Bibliotecas

- [AForge.NET](#)
- [Weka with C#](#)
- [Infer.NET](#)
- [Accord.NET Framework](#)

# Overfitting

Overfitting é o termo, em aprendizagem de máquina, utilizado quando o modelo estatístico se ajusta ao conjunto de dados/amostra.



# Underfitting





# Contato

- Email: [msgrubler@gmail.com](mailto:msgrubler@gmail.com)
- Twitter: [@msgrubler](https://twitter.com/msgrubler)
- Github: <https://github.com/Murillo>

# Referências

- BASHEER, I. A.; HAJMEER, M. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods, [S.l.], v. 43, n. 1, p. 3-31, 2000.
- TU, J. V. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of clinical epidemiology, [S.l.], v. 49, n. 11, p. 1225-1231, 1996.
- ROJAS, R. Neural Networks: A Systematic Introduction. 1. ed. [S.l.]: Springer, 1996.
- TISSOT, Hegler C.; CAMARGO, Luiz C.; POZO, A. T. Treinamento de redes neurais feedforward: comparativo dos algoritmos backpropagation e differential evolution. In: **Brazilian Conference on Intelligent Systems**. 2012.
- MONTANA, David J.; DAVIS, Lawrence. Training Feedforward Neural Networks Using Genetic Algorithms. In: **IJCAI**. 1989. p. 762-767.