

Inteligência Artificial

Relatório 4 - Redes neuronais

Grupo 8

Luís Pinto, n.º 201704025
Sónia Almeida, n.º 201811293
Miguel Lançóis, n.º 201506342

2 de Julho de 2019

Conteúdo

1	Introdução	3
2	Redes neuronais	3
2.1	O que são?	3
2.2	Neurônio computacional	3
2.3	Tipos de redes neuronais	4
2.4	Multi Layer Perceptrons (MPR)	4
3	Algoritmo de redes neuronais	5
4	Implementação	5
5	Resultados	5
6	Conclusão	5
7	Bibliografia/webgrafia	5

1 Introdução

Neste trabalho, vamos criar uma rede neuronal composta por duas camadas de neurónios.

2 Redes neuronais

2.1 O que são?

As redes neuronais são compostas por neurónios e permitem, graças a um conjunto de treino, "aprender". Os neurónios de uma rede neuronal são inspirados por os neurónios presentes na natureza mas o modelo utilizado nas redes neuronais é simplificado já que os neurónios do nosso cérebro não são bem conhecidos. No nosso cérebro, um neurônio é composto por um corpo celular, dendritos e axônios, como se pode ver na figura 1 (retirada do site https://www.researchgate.net/figure/Figura-2-Representacao-esquematica-da-estrutura-do-neur1_fig1_230640478):

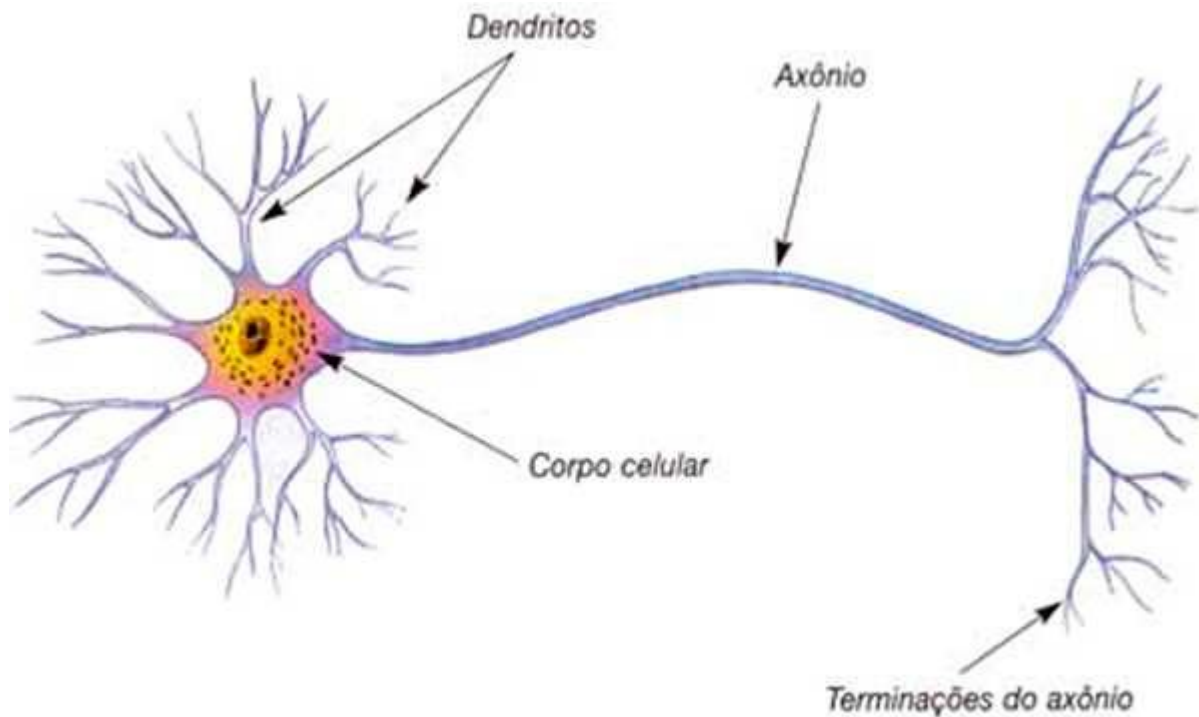


Fig.1: neurônio

Em redes neuronais, um neurônio é um elemento aritmético simples e uma rede neuronal é um conjunto de neurónios interligados.

2.2 Neurônio computacional

Um neurônio computacional é composto por um conjunto de entradas, que têm um peso associado, por um conjunto de saídas e por uma função de ativação. Cada entrada tem associada um peso aleatório $w_{i,j}$ (entre -1 e 1 no nosso caso), onde j representa cada link de entrada e i o número do neurônio considerado.

Para o neurônio i , deve calcular-se o somatório das entradas a_j multiplicadas por o peso $w_{i,j}$: $v_i = \sum_j a_j w_{i,j}$

O valor da saída a_i do neurônio i é dada por $g(v_i)$ onde g é a função de ativação. No nosso caso, a função de ativação é dada pela seguinte expressão: $g(v_i) = \frac{1}{1+e^{-v_i}}$. Esta função é uma sigmoide que é diferenciável, o que é muito importante para poder minimizar o erro de classificação durante o treinamento da rede. A função de ativação deve permitir a comparação entre a entrada e a saída e depende então dos dados.

2.3 Tipos de redes neuronais

Existem vários tipos de redes neuronais:

- feed-forward: não há ciclos, o grafo é direcionado e os links são unidirecionais. Não há links entre neurônios da mesma camada. Dados que os links são unidirecionais, a informação não retorna para os nós das camadas anteriores.
- recorrente: os links podem formar topologias arbitrárias.
- Hopfield: as conexões são bidirecionais entre os nós com pesos simétricos. Todos os nós podem ser entrada ou saída. A função de ativação produz -1 ou +1. Não encontra ótimos globais.
- Máquinas de Boltzmann: também usa pesos simétricos mas inclui unidades (escondidas) que não são nem entrada nem saída. A função de ativação é estocástica, onde a probabilidade de ser 1 é função dos pesos da entrada.

No nosso caso, são Perceptrons que vão ser usados. Os perceptrons são o exemplo mais simples de redes feed-forward com apenas uma camada.

2.4 Multi Layer Perceptrons (MPR)

Multi Layer Perceptrons são Perceptrons de várias camadas. Neste trabalho, vamos usar um MPR com duas camadas. No nosso modelo, os neurônios de uma camada só estão ligados a neurônios de outra camada e não a neurônios da mesma camada:

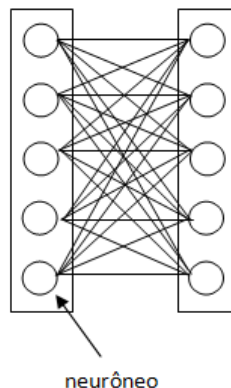


Fig.2: rede de perceptron

Os pesos são atualizados com a seguinte fórmula: $w^{(k+1)} = w^{(k)} + \lambda[v_i - g(w^k, a_i)]a_i$ onde $w^{(k)}$ representa o peso para uma dada entrada do neurônio i ao instante k .

3 Algoritmo de redes neuronais

Para saber se o nosso modelo funciona bem, de vemos utilizar um conjunto de treino e outro de teste.

4 Implementação

Linguagem: Para a implementação do código decidimos utilizar a linguagem *Python 3.7*.

Estrutura de Dados:

5 Resultados

Execução dos testes:

- Executámos iterações até termos um erro inferior a 0.05, ou até obter 10^6 iterações;
- Testámos vários learning rates diferentes e, para cada um deles, testámos 4 inputs exemplos diferentes: 1111,1010,0011,0111;
- Verificámos qual dos 4 exemplos obteve o maior erro e considerámos esse o erro para a nossa tabela de resultados.

Testes:

Learning Rate	Número de Iterações	Erro	Tempo de execução (ms)
0.05	999999	0.200313	120793
0.10	247659	0.049998	29223
0.15	297898	0.049998	35058
0.20	84830	0.049991	10004
0.25	228214	0.037433	27106
0.30	331040	0.049970	38869
0.35	329790	0.049958	38211
0.40	999999	0.800029	116724
0.45	999999	0.750028	117297

6 Conclusão

Ao avaliar cuidadosamente os resultados que obtivemos relativamente aos 4 exemplos referidos na secção 5, determinámos que um learning rate entre 0.20 e 0.25 fornece resultados significativamente mais concretos do que qualquer outro valor do learning agreement. Para além dos resultados apresentados neste relatório, testámos com mais neurónios "fantasma", com nove, e os resultados foram semelhantes. Decidimos ocultar esses testes do relatório uma vez que não nos foi pedido tal tipo de teste mas estes podem ser encontrados no ficheiro *testes.txt* na parte dos testes com 9 neurónios fantasma.

7 Bibliografia/webgrafia

- <https://artint.info/2e/html/ArtInt2e.Ch7.S5.html>

- <http://www.dcc.fc.up.pt/~ines/aulas/1718/IA/neunets.pdf>