Data Mining
Redes Neurais Artificiais

Prof. Dr. Joaquim Assunção

DEPARTAMENTO DE COMPUTAÇÃO APLICADA
CENTRO DE TECNOLOGIA
UFSM
2024

www.inf.ufsm.br/~joaquim

Fair user agreement

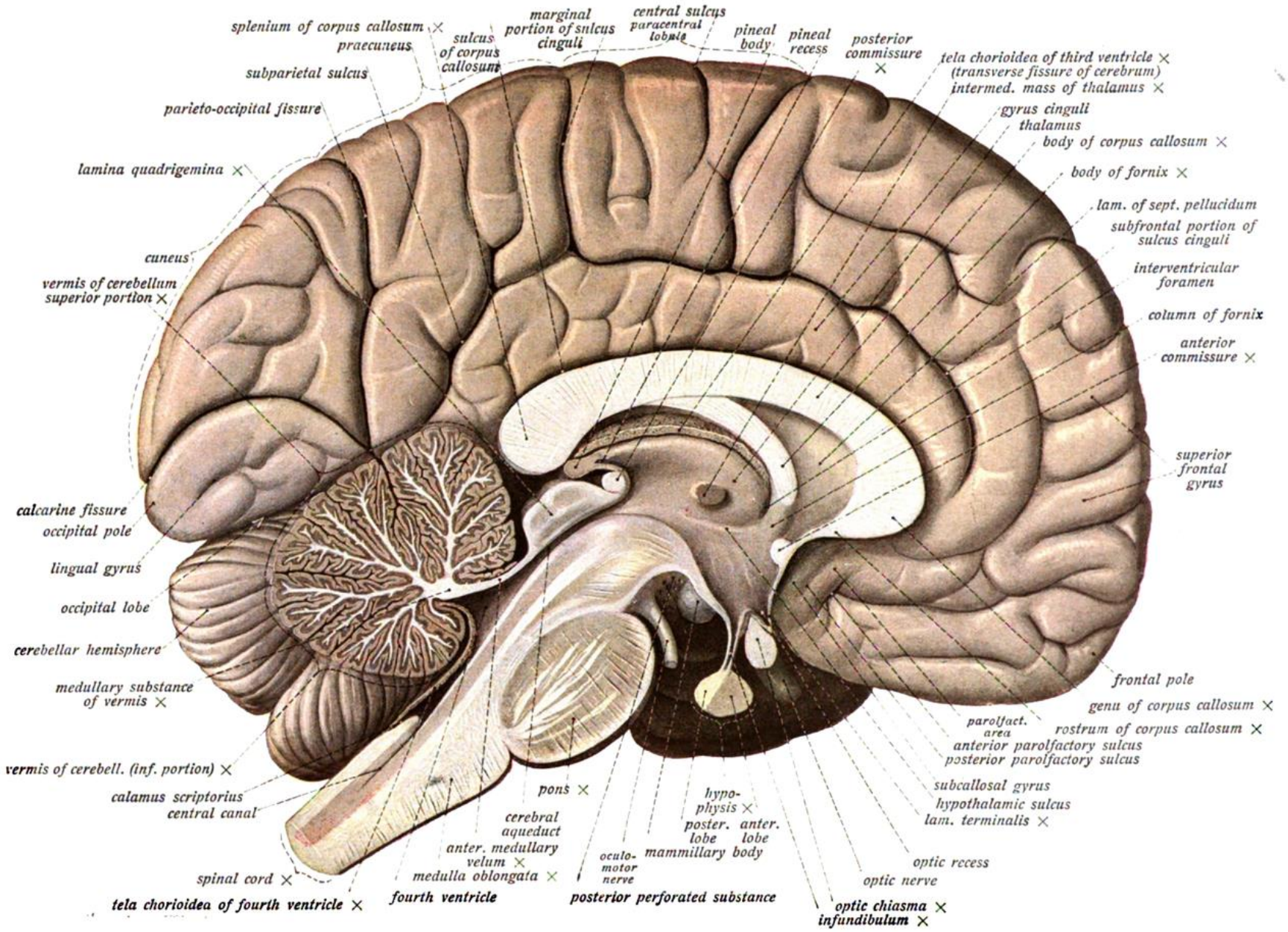
Este material foi criado para a disciplina de Mineração de Dados - Centro de Tecnologia da UFSM.

Você pode usar este material livremente*; porém, caso seja usado em outra instituição, **me envie um e-mail** avisando o nome da instituição e a disciplina.

*A maior parte deste material foi retirado do livro: “Joaquim V. C. Assunção. **Uma Breve Introdução à Mineração de Dados: Bases Para a Ciência de Dados, com Exemplos em R**. 192 páginas. Novatec. 2021. ISBN-10 : 6586057507.”

Prof. Dr. Joaquim Assunção.

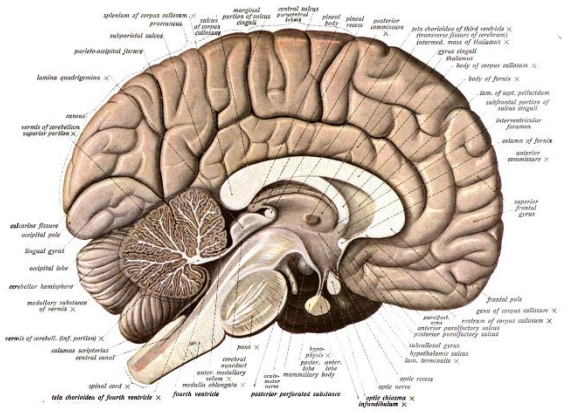
joaquim@inf.ufsm.br



https://en.wikipedia.org/wiki/Human_brain#/media/File:Sobo_1909_624.png

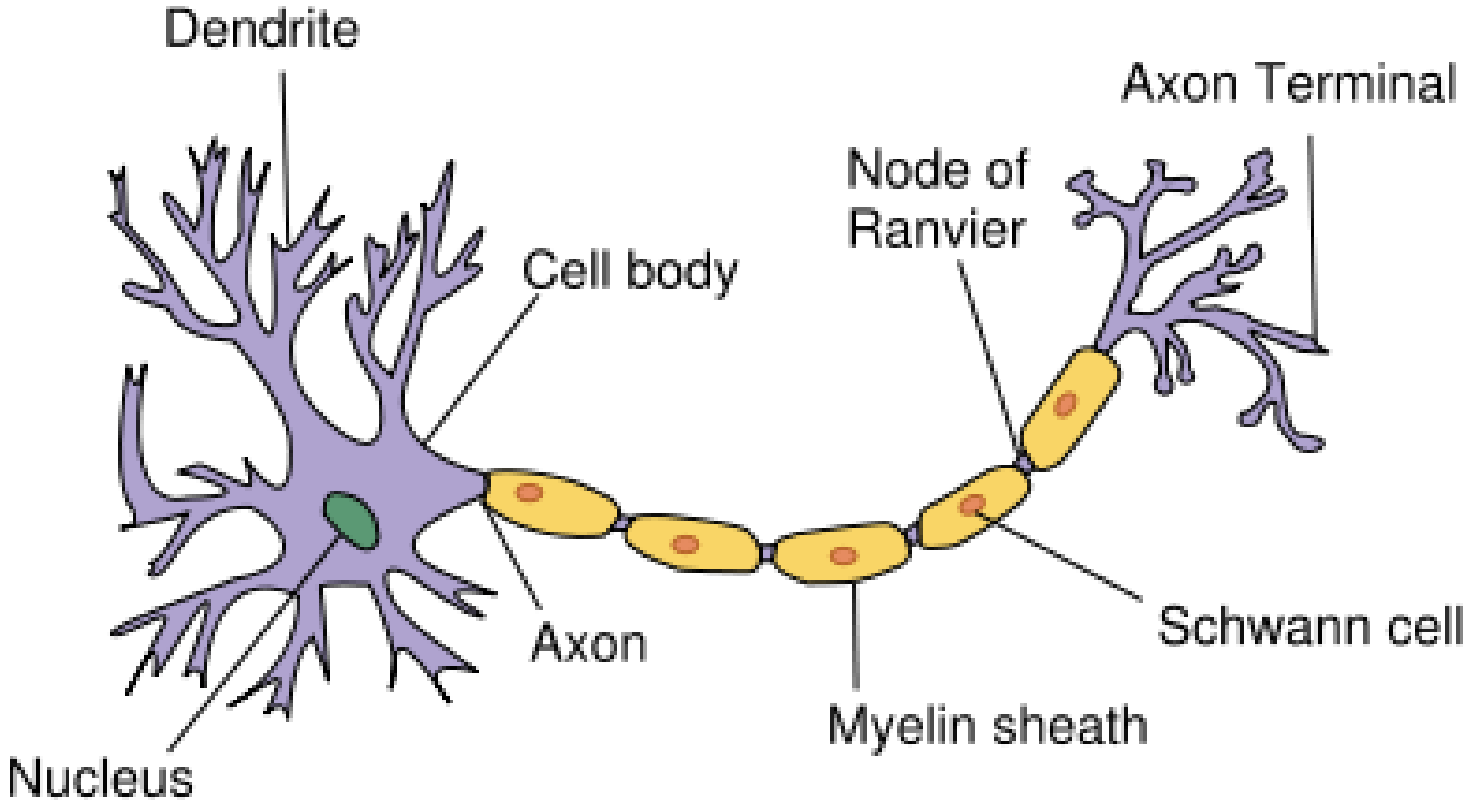
Cérebro e Neurônios

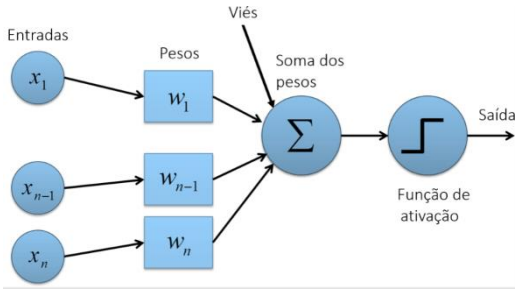
- Estima-se que o cérebro humano tenha cerca de ~86 bilhões neurônios cada um conectado em média a ~7.000 outros neurônios.
- Cada neurônio recebe sinais através de sinapses que controlam os efeitos do sinal no neurônio. Acredita-se que essas conexões sinápticas desempenham um papel fundamental no comportamento do cérebro.



https://en.wikipedia.org/wiki/Human_brain#/media/File:Sobo_1909_624.png

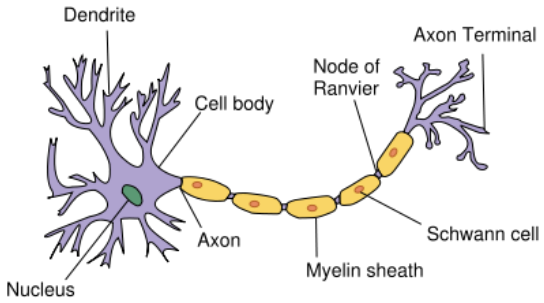
Neurônio *(en. neuron)*

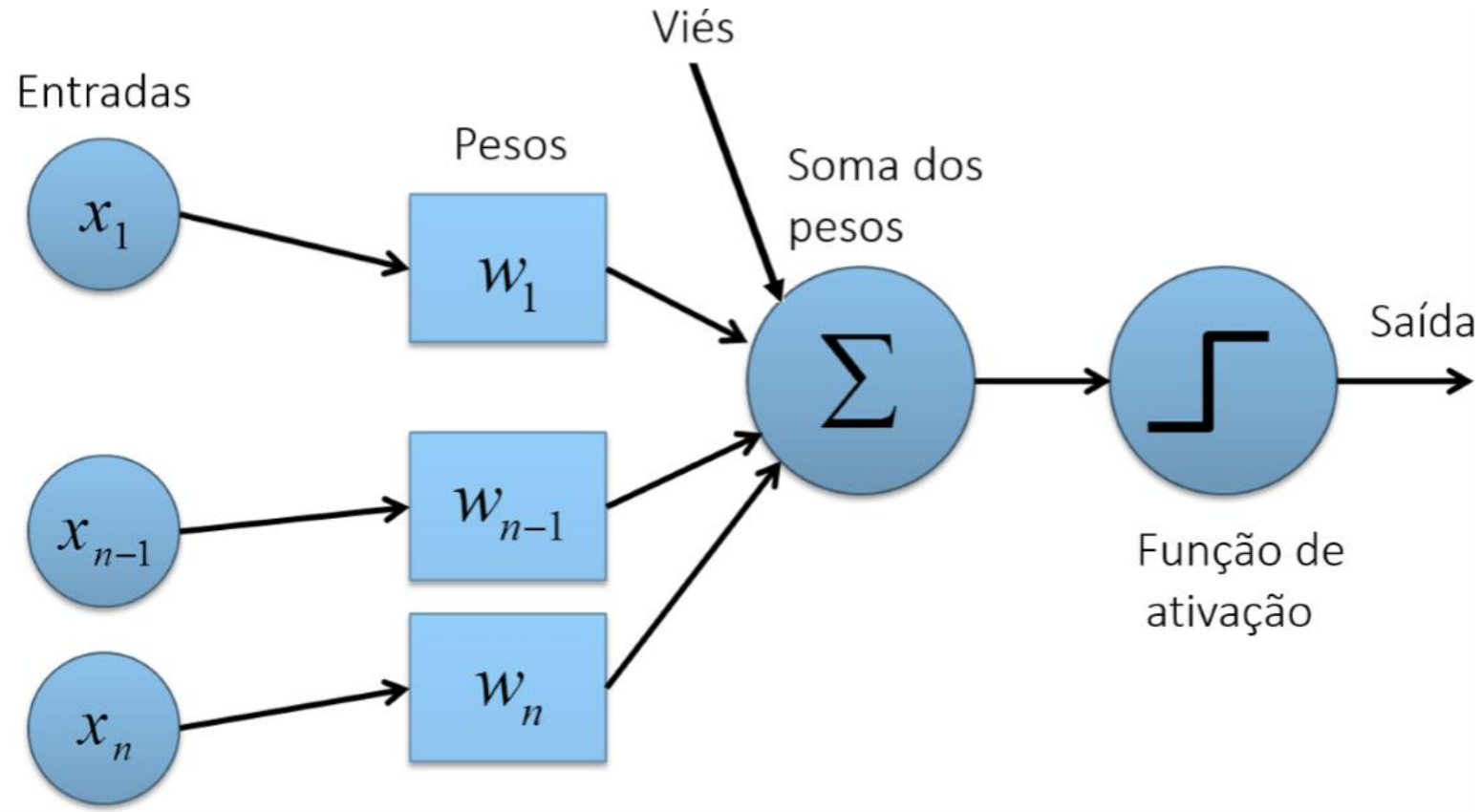




*https://simple.wikipedia.org/wiki/File:Neuron.svg

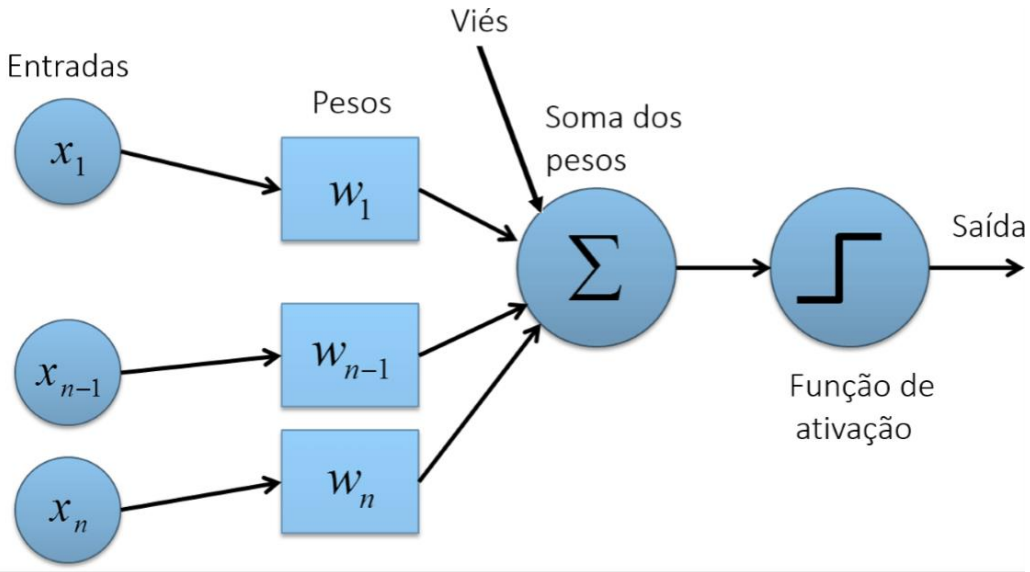
Perceptron


Dendrite
Axon Terminal
Node of Ranvier
Cell body
Schwann cell
Axon
Myelin sheath
Nucleus


Viés
Entradas
Pesos
Soma dos pesos
x_1
w_1
x_{n-1}
w_{n-1}
x_n
w_n
Σ
Saída
Função de ativação

Perceptron

- Cada entrada é associada a um peso (w) que é agregada por uma função de soma (soma 1).
- w_0 é o viés associado.

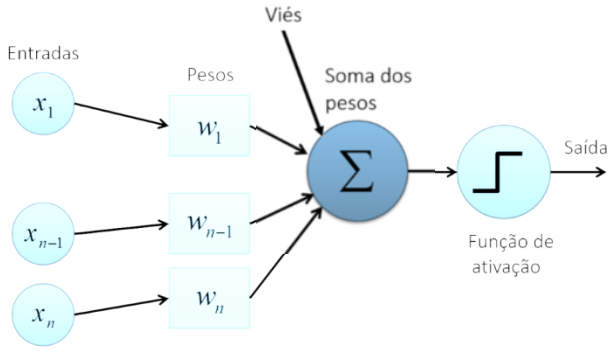


$$s = w_0 \sum_{i=1}^{n} w_i, x_i$$

Somatório → Viés

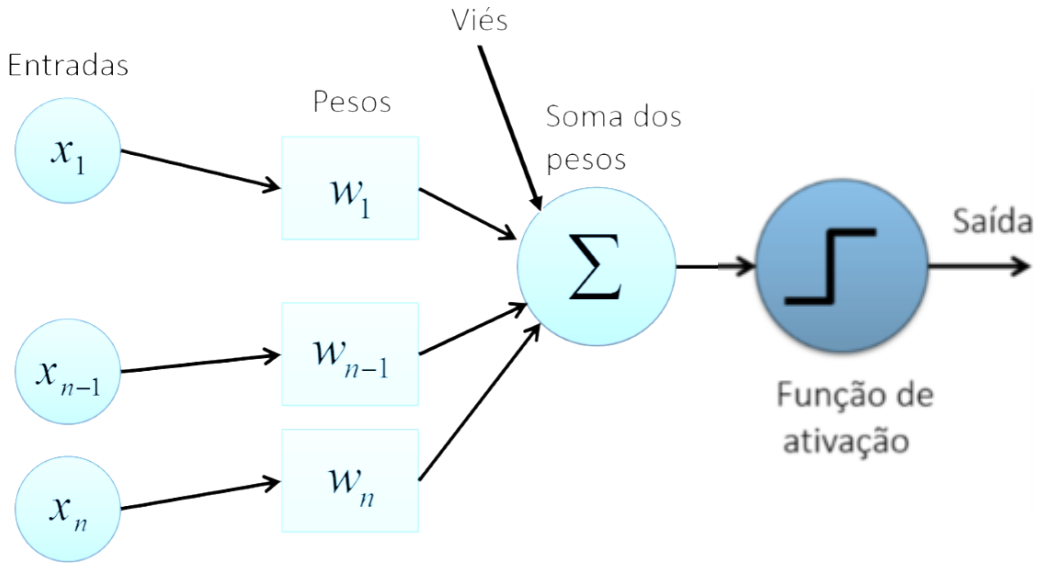
- O viés associado pode ser visto como um peso 1 para uma entrada 0. Logo, podemos simplificar a equação a seguir.

$$s = w_0 \sum_{i=1}^{n} w_i, x_i \quad = \quad s = \sum_{i=0}^{n} w_i, x_i$$



Função de ativação

- Uma função de ativação g(também chamada de função de esmagamento) que mapeia *s* para g(v) o valor de saída do neurônio.



Exemplo

- Considere o conjunto de dados abaixo.
- Usaremos um *perceptron* para classificar a variável "classe" de acordo com as variáveis "var1" e "var2".

```
> foo
  var1 var2 classe
1    1    1      1
2    1    0      0
3    0    1      0
4    0    0      0
```

Exemplo

- Use a biblioteca "`neuralnet`".
- Os parâmetros são:

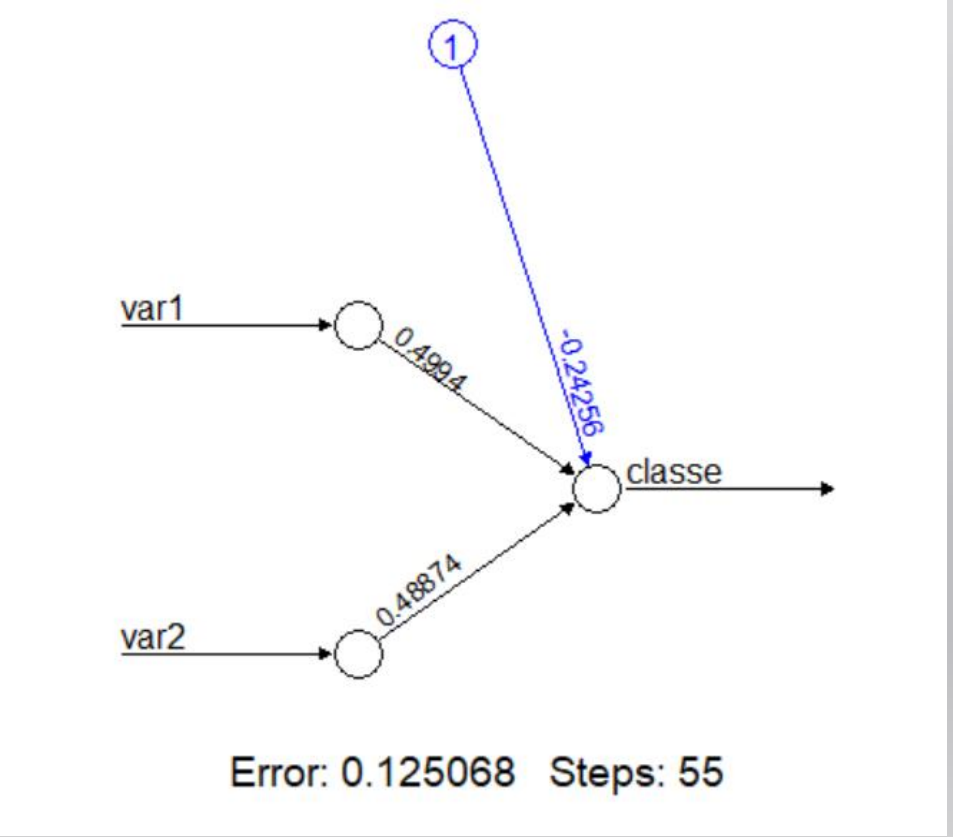
1. Classe a ser predita "~"
2. Variáveis que serão usadas como entrada, separadas por um sinal de "+"
3. O número de camadas ocultas da rede. Vamos usar zero para criar um perceptron.

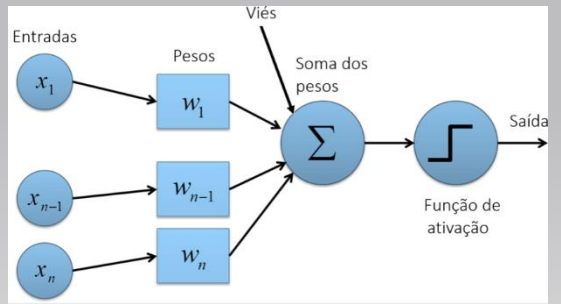
```
NN = neuralnet(classe ~ var1 + var2, foo,
     hidden = 0)
```

Exemplo

- Podemos usar a função `plot` nativa do R para gerar uma representação visual do perceptron (ou NN, ou DNN).

```
plot(NN)
```





Internamente...

Ao imprimir a variável, uma série de informações são expostas.

- `covariate` e `response` são as variáveis de entrada e a variável a ser predita, respectivamente.
- `$err.fct` mostra a função de erro que pode ser a soma do erro quadrático (SSE) ou a entropia cruzada.
- Internamente, SSE é dado por: $\frac{1}{2} * (y - x)^2$

Internamente...

...

- A função de ativação (`$act.fct`) que pode ser logística ou hiperbólica.
- Pesos iniciais, pesos finais, erro final, quantidade de passos para a convergência etc.

Usando a Rede Neural

- Use a função `compute` para obter a saída de uma rede neural com base no que foi aprendido pela mesma.

```
previsao <- compute(NN, dadosDeEntrada)
```

Note que `dadosDeEntrada` são exatamente as mesmas variáveis para as quais a rede foi treinada, na mesma ordem de entrada.

Rede Neural ADALINE

- A primeira rede neural, derivada diretamente do Perceptron.
- Adaline (*ADAptative LINear Element*) foi proposta por Widrow e Hoff.
- Essa rede é formada por apenas uma camada oculta e possui uma estrutura quase idêntica à do *perceptron*, exceto pela função de ativação que é linear e é dada pela minimização do erro quadrático médio (*Squared Mean Error*, SME).

Hands On!

1. Crie um dataframe com as operações lógicas com quatro variáveis binárias (v1, v2, v3, v4) contendo todas as combinações possíveis, adicione como classe a operação lógica: *v1 E v2 OU v3 E V4*. Crie uma rede neural Adaline para testar sua operação lógica usando 12 amostras. Teste as demais entradas.

Rede Neural de múltiplas camadas

- O poder de uma rede neural está diretamente relacionado a quantidade de camadas ocultas que ela guarda (embora outros fatores tenham influência).
- Em R, usando `neuralnet,` podemos fazer isso simplesmente alterando o parâmetro `hidden.`

Rede Neural de múltiplas camadas

- Para isso, passamos um vetor de inteiros, onde o tamanho do vetor é a quantidade de camadas ocultas e os valores correspondem a quantidade de neurônios na rede. Ex: `c(5, 7)` vai gerar uma rede com duas camadas ocultas, a primeira com 5 neurônios e a segunda com 7.

Hands On!

1. Use o *dataframe* criado (com quatro variáveis binárias) e repita o exercício anterior com uma rede de duas e três camadas. Analise o resultado.

Notas

- A biblioteca *neuralnet* não trabalha com classes nominais. Use múltiplas classes numéricas para isso (como o formato de uma matriz para o APRIORI).
- Valores de diferentes escalas podem ser problemáticos, normalize os dados primeiramente. Depois reverta o processo para obter os dados originais.