

# Redes Neurais e Aprendizado Profundo



Nas aulas anteriores, implementamos os seguintes algoritmos de Aprendizado de Máquina:

- Regressão Linear (para problemas de regressão)
- Regressão Logística (para problemas de classificação)

**OBS:** Com isso, finalizamos a **Parte 1** da disciplina.

Nas aulas anteriores, implementamos os seguintes algoritmos de Aprendizado de Máquina:

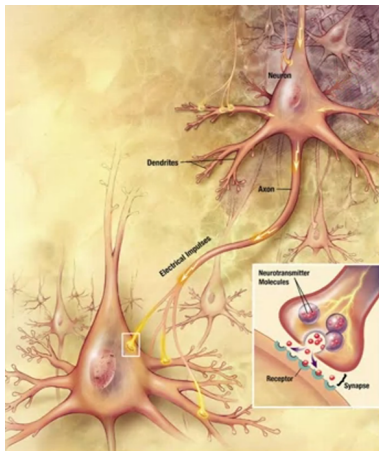
- Regressão Linear (para problemas de regressão)
- Regressão Logística (para problemas de classificação)

**OBS:** Com isso, finalizamos a **Parte 1** da disciplina.

Vamos agora entrar na **Parte 2** da disciplina, onde estudaremos algoritmos **mais avançados**, em especial, as **Redes Neurais Artificiais (RNAs)**

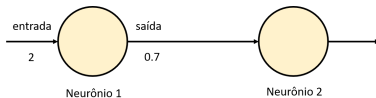
- **Origem:** Algoritmos que imitam o funcionamento do cérebro humano (redes neurais biológicas).
- **Aplicações modernas:** Reconhecimento de voz, Processamento de imagens, Processamento de linguagem natural (texto), Modelos Generativos, etc

# Mas como funcionam os neurônios biológicos?



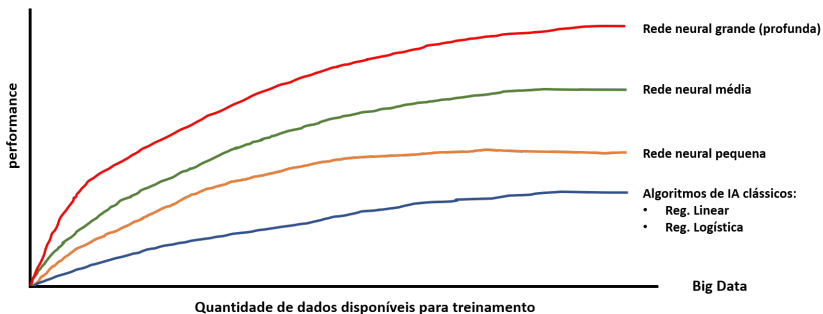
- **Possuem entradas**, que recebem sinais elétricos de outros neurônios
- Esses sinais são recebidos, processados e enviados por meio das suas **saídas** a outros neurônios.
- Os neurônios seguintes recebem esses sinais de saída como sendo suas entradas, e o ciclo continua.
- Esse raciocínio serviu de inspiração para a criação das redes neurais artificiais.

# Modelo matemático simplificado para um neurônio



- Para nós, um neurônio é um modelo (função), que recebe uma ou mais entradas, faz alguma conta com esses valores numéricos, e os disponibiliza em sua(s) saída(s)

# Necessidade de algoritmos mais complexos

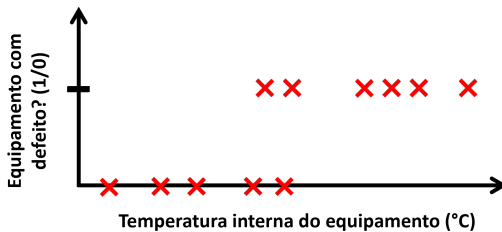


- Devido ao aumento de dados disponíveis digitalmente
- Devido à elevada capacidade de processamento a nível de hardware (GPUs, por exemplo)
- Assim, aplicações bastante complexas se tornaram possíveis (ChatGPT, por exemplo)

## Exemplo: Equipamento com defeito

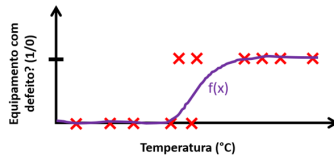
Suponha:

- Que você possui um equipamento de alto custo
- Que você deseja realizar manutenções preventivas periodicamente
- Que o fabricante do equipamento forneceu o seguinte conjunto de dados para auxiliar no diagnóstico de falhas





## Exemplo: Equipamento com defeito



Tratando como um modelo de **regressão logística**, temos:

**Entrada:**

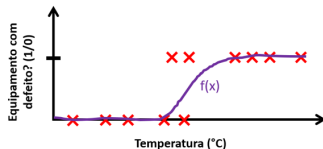
$x$ : temperatura do equipamento

**Saída:**

$f(x)$ : probabilidade do equip. com temp.  $x$  estar com defeito

$$f(x) = \frac{1}{1 + e^{-(wx+b)}} \rightarrow \text{Sigmoide}$$

## Exemplo: Equipamento com defeito



Tratando como um modelo de **regressão logística**, temos:

**Entrada:**

$x$ : temperatura do equipamento

**Saída:**

$f(x)$ : probabilidade do equip. com temp.  $x$  estar com defeito

$$f(x) = \frac{1}{1 + e^{-(wx+b)}} \rightarrow \text{Sigmoide}$$

Tratando como uma **rede neural com um único neurônio**, temos:

**Entrada:**

$x$ : temperatura do equipamento

**Saída:**

$f(x)$ : probabilidade do equip. com temp.  $x$  estar com defeito

$$a = f(x) = \frac{1}{1 + e^{-(wx+b)}}$$



- $a$  refere-se à quantidade de ativação feita pelo neurônio
- Nessa representação, note que o neurônio é uma unidade de regressão logística, ou seja, cada neurônio está sendo ativado pela função sigmoide.
- Também seria possível usar uma função de ativação linear, tal que  $a = f(x) = wx + b$ . Algumas pessoas chamam este caso de neurônio "sem ativação". Note que isso resulta num modelo de regressão linear.

Buscando tornar o problema **mais realista**, vamos considerar agora que o equipamento estar com defeito ou não depende de 4 características principais (e não apenas da sua temperatura).

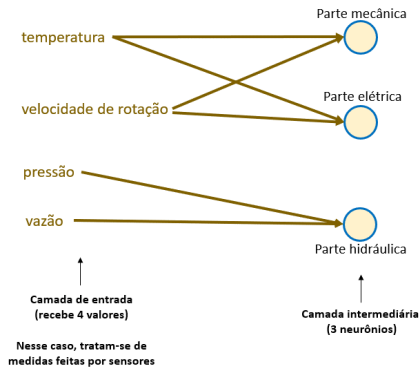
As características agora são:

- temperatura [ $^{\circ}\text{C}$ ]
- velocidade de rotação [rpm]
- pressão [Pa]
- vazão [ $\text{m}^3/\text{s}$ ]

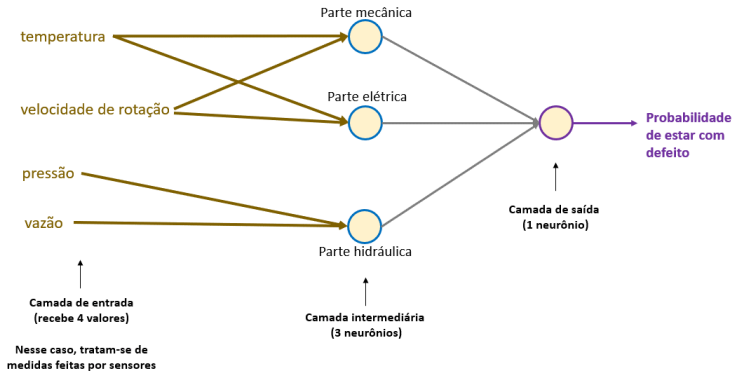
Pergunta:

Como poderíamos criar uma rede neural que descreve esse comportamento?

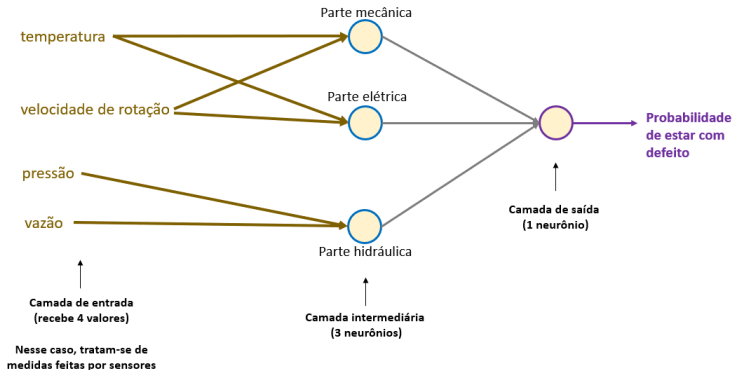
## Exemplo: Equipamento com defeito



## Exemplo: Equipamento com defeito

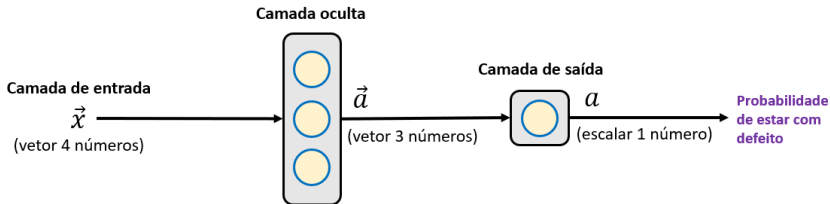
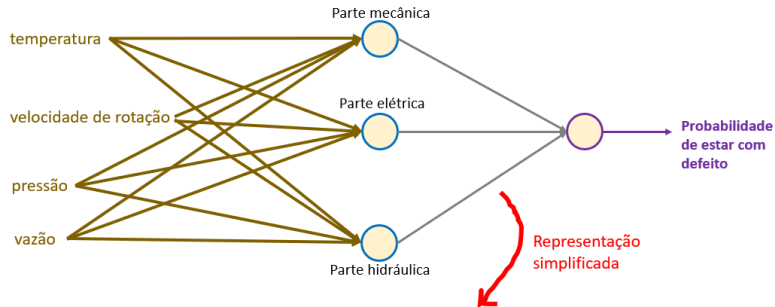


## Exemplo: Equipamento com defeito

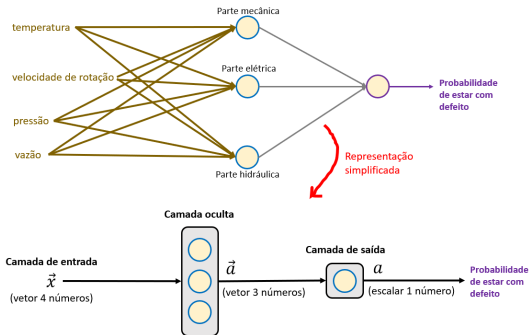


- A rede recebe 4 números por meio das suas 4 entradas
- Os três neurônios da Camada intermediária (também chamada de 'camada oculta') calculam suas 3 ativações correspondentes com base nos valores presentes em sua(s) entrada(s)
- Por fim, o neurônio da Camada de saída recebe esses 3 valores e calcula seu próprio valor único de ativação
- Cada neurônio pode ser entendido como sendo uma unidade elementar da regressão logística

## Exemplo: Equipamento com defeito



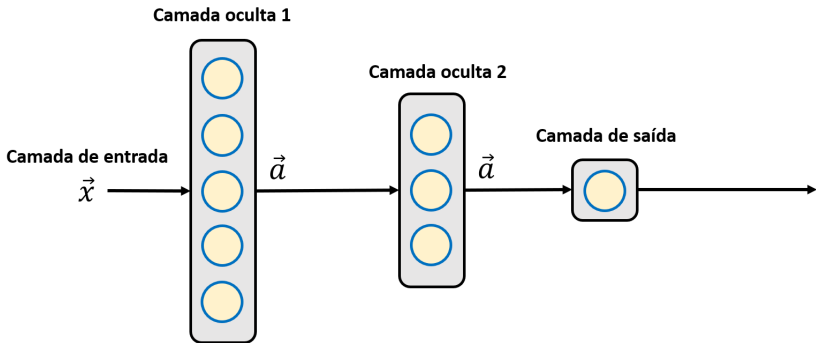
## Exemplo: Equipamento com defeito



### Simplificando a ideia:

- Na prática, é difícil identificar quais são as entradas mais relevantes para cada neurônio, igual nós fizemos para esse exemplo
- Por isso, nas redes neurais de fato usadas nós consideramos que cada neurônio tem acesso a todos os valores que vêm da camada anterior
- A ideia é que rede neural consiga aprender de forma autônoma quais características presentes nas suas entradas são relevantes e quais não são
- Ou seja, não é tão necessário fazer **Engenharia de Características** (como fizemos no caso da Regressão linear), já que a tendência é que a própria rede crie características intermediárias que ela acha que são mais apropriadas
- Isso faz com que as Redes Neurais sejam tão poderosas.





Perguntas:

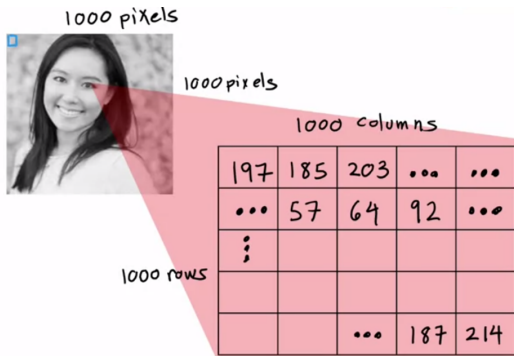
Quantas camadas escolher para a rede? Quantos neurônios colocar em cada camada?

Resposta:

Trata-se de um problema de **definição da arquitetura da rede**.

# Um outro Exemplo: Reconhecimento facial

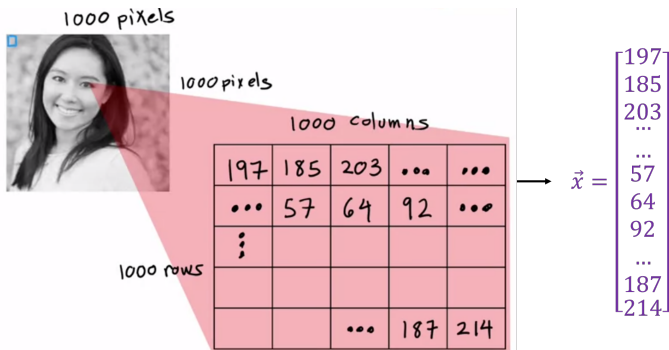
Suponha que você deseja identificar a identidade de uma pessoa a partir de uma imagem.



Fonte: **Machine Learning Specialization**, *deeplearning.ai*, Stanford Online, Coursera.org.

- Para o computador, a imagem é uma matriz de pixels ( $1000 \times 1000$ , nesse caso), onde cada pixel possui um próprio valor de intensidade de brilho (de 0 a 255, nesse caso)
- Seria possível montar um vetor de entrada para a nossa rede neural a partir da matriz?

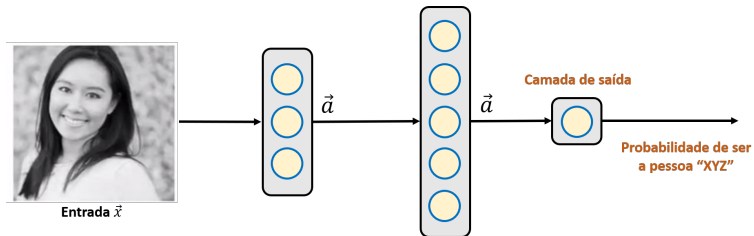
## Um outro Exemplo: Reconhecimento facial



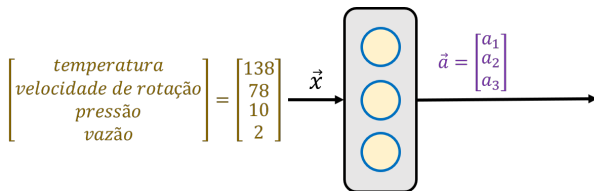
Fonte: **Machine Learning Specialization**, [deeplearning.ai](https://www.deeplearning.ai), Stanford Online, Coursera.org.

Sim, basta 'desenrolar' a matriz. Em inglês, esse procedimento é chamado de *flatten*

## Um outro Exemplo: Reconhecimento facial



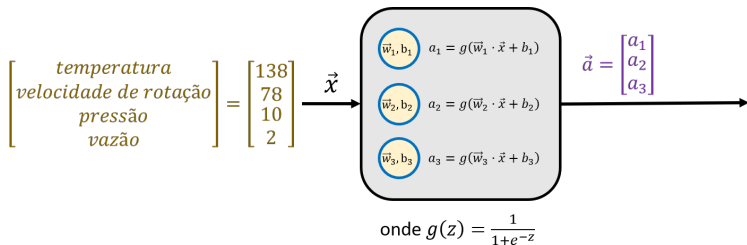
Voltando para o exemplo de defeito em equipamento (pensando apenas na camada oculta)



## Perguntas:

- Como calcular a ativação de cada neurônio?
- Quantos valores o vetor  $\vec{a}$  terá?

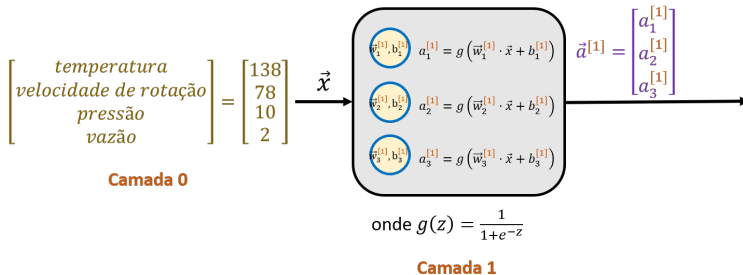
## Detalhando o funcionamento de uma das camadas de uma rede neural



Lembre-se:

- Cada neurônio consiste em uma unidade de regressão logística

## Detalhando o funcionamento de uma das camadas de uma rede neural



### Importante!

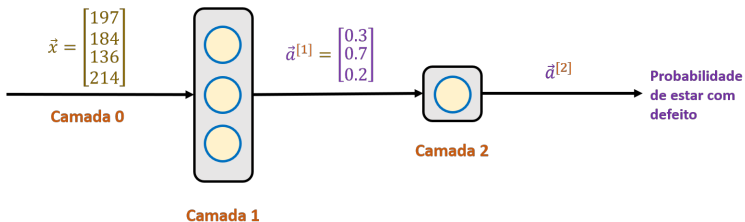
- Usaremos o sobrescrito  $[i]$  para nos referirmos a valores que correspondem à camada  $i$ .
- Pense também o seguinte: Os valores agora calculados para  $\vec{a}^{[1]}$ , por exemplo,

$$\vec{a}^{[1]} = \begin{bmatrix} 0.3 \\ 0.7 \\ 0.2 \end{bmatrix}$$

servem como valores de entrada para a camada que viria na sequência (camada 2).

# Detalhando o funcionamento de uma das camadas de uma rede neural

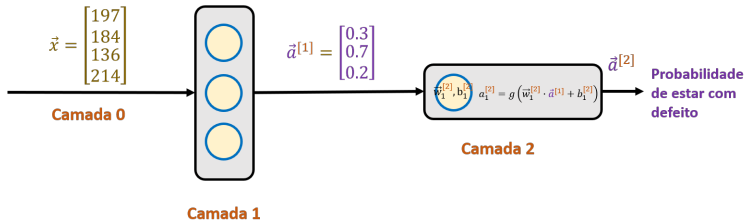
Ou seja, por enquanto nós temos:



- Como calcular a ativação  $\vec{a}^{[2]}$  do neurônio presente na Camada 2?



## Detalhando o funcionamento de uma das camadas de uma rede neural



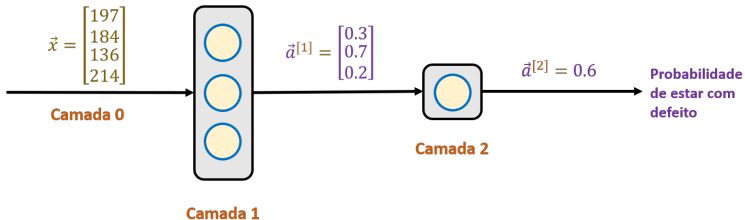
- Fazemos de forma análoga ao que havíamos feito para a Camada 1
- Note que a entrada para o neurônio da Camada 2 não é  $\vec{x}$ , mas sim  $\vec{a}^{[1]}$ .
- $\vec{a}^{[2]}$  vai resultar num valor entre 0 e 1, por exemplo,

$$\vec{a}^{[2]} = 0.6$$

Para o nosso exemplo, esse valor consiste na probabilidade do equipamento apresentar defeito.

# Detalhando o funcionamento de uma das camadas de uma rede neural

Portanto, temos o seguinte resultado final:

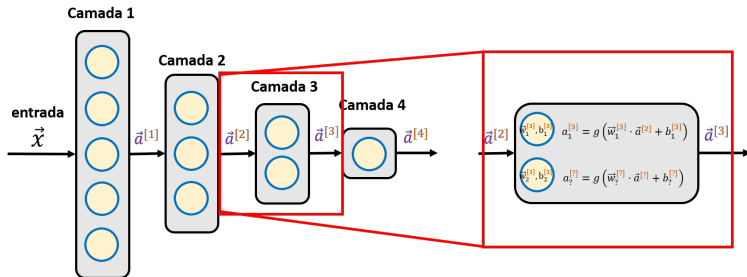


Pergunta:

E aí, esse equipamento está com defeito ou não?

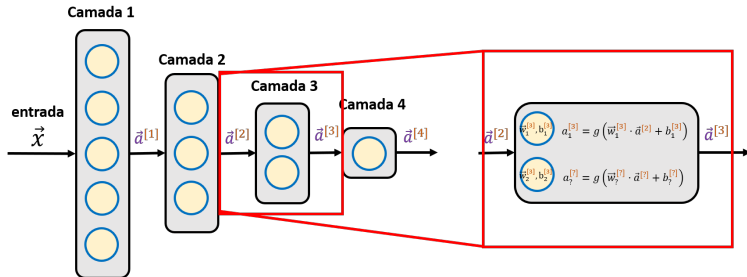
Resposta:

Depende. Se considerarmos um valor de limiar de 0.5, então podemos dizer que está sim com defeito ou na eminência de apresentar um defeito (manutenção preventiva).



## Perguntas:

- A) Qual é a camada de entrada dessa rede? E sua camada de saída?
- B) Quantas camadas ocultas essa rede possui?
- C) Com relação à Camada 3, quais seriam os subscritos e sobrescritos para o seu segundo neurônio?



Podemos definir que a ativação gerada pelo  $j$ -ésimo neurônio da  $l$ -ésima camada é dada por

$$a_j^{[l]} = g(\vec{w}_j^{[l]} \cdot \vec{a}^{[l-1]} + b_j^{[l]})$$

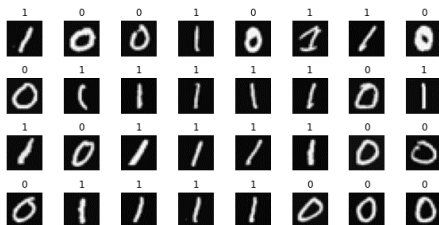
Para que essa notação possa valer também para a Camada 1, definimos também

$$\vec{x} = \vec{a}^{[0]}$$

Vamos agora sistematizar o processo de realizar previsões a partir de uma rede neural já treinada.

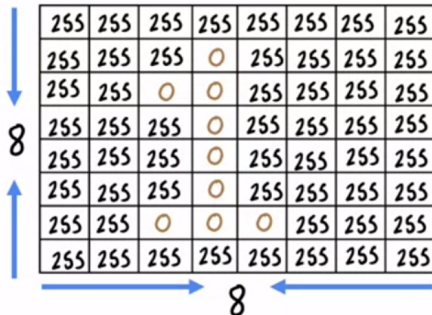
Quando usamos uma rede já treinada para fazermos previsões, trata-se de um problema de **inferência**.

Exemplo: Reconhecendo dígitos 0 e 1 escritos à mão



## Exemplo: Reconhecendo dígitos 0 e 1 escritos à mão

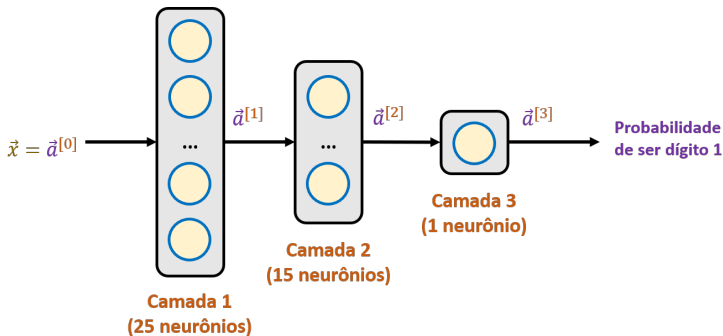
Considere que você tem uma imagem ( $8 \times 8 = 64$  pixels) que possui o dígito 0 ou 1 escrito à mão:



- Valor 0 significa cor preta
- Valor 255 significa cor branca
- Valores entre 0 e 255 implicam em diferentes tons de cinza
- $\vec{x}$  é o vetor (matriz desenrolada) contendo o valor de brilho para os 64 pixels

## Exemplo: Reconhecendo dígitos 0 e 1 escritos à mão

Usaremos uma rede neural com a seguinte arquitetura:



Pergunta:

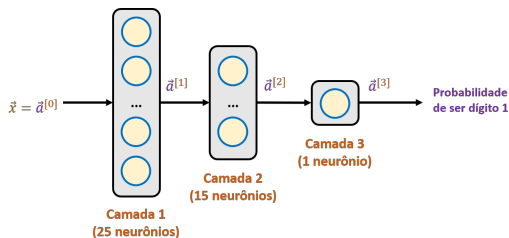
Como calcular  $\vec{a}^{[3]}$  a partir de  $\vec{a}^{[0]} = \vec{x}$ ?

Resposta:

Usaremos a estratégia que acabamos de aprender (*forward propagation*) → iremos da esquerda para a direita, propagando os cálculos!



## Exemplo: Reconhecendo dígitos 0 e 1 escritos à mão



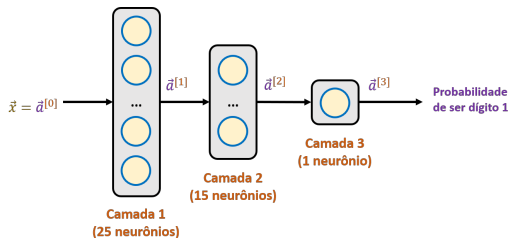
Para a Camada 1, temos:

$$\vec{a}^{[1]} = \begin{bmatrix} g(\vec{w}_1^{[1]} \cdot \vec{x} + b_1^{[1]}) \\ \vdots \\ g(\vec{w}_{25}^{[1]} \cdot \vec{x} + b_{25}^{[1]}) \end{bmatrix}$$

Para a Camada 2, temos:

$$\vec{a}^{[2]} = \begin{bmatrix} g(\vec{w}_1^{[2]} \cdot \vec{a}^{[1]} + b_1^{[2]}) \\ \vdots \\ g(\vec{w}_{15}^{[2]} \cdot \vec{a}^{[1]} + b_{15}^{[2]}) \end{bmatrix}$$

## Exemplo: Reconhecendo dígitos 0 e 1 escritos à mão



Finalmente, para a Camada 3, temos:

$$\vec{a}^{[3]} = g(\vec{w}_1^{[3]} \cdot \vec{a}^{[2]} + b_1^{[3]})$$

se  $a^{[3]} \geq 0.5$ ,  $\hat{y} = 1$  (imagem contém dígito 1).

se  $a^{[3]} < 0.5$ ,  $\hat{y} = 0$  (imagem contém dígito 0).

**De olho no código!**

Vamos agora explorar o conhecimento básico adquirido sobre neurônios e camadas usando código.

Clique no link abaixo para acessar o código:

```
https://colab.research.google.com/github/xaximpvp2/master/blob/main/codigo\_aula\_15\_neuronios\_e\_camadas.ipynb
```

**OBS:** Nesse código, vamos aproveitar para iniciar nossos estudos em cima do pacote **TensorFlow**, que consiste numa ferramenta de machine learning bastante utilizada.

## Parte 1

Rode todo o código. Responda às questões nele contidas e complete-o, se necessário.

**OBS:** Caso esse seja seu primeiro contato com **TensorFlow**, é normal que demore um tempo até que você concorde com o que está sendo feito e explicado no código.

## Parte 2

- 1 Adicione amostras tanto ao problema de regressão como ao problema de classificação (sem comprometer de forma significativa a capacidade do modelo de explicar esses dados)