Computação Bio-Inspirada

Fabrício Olivetti de França

01 de fevereiro de 2020

UFABC

CMCC
Centro de Matemática, Computação e Cognição
Center for Mathematics, Computation and Cognition

Topics

1. Inspiração Biológica
2. Neurônio Artificial
3. Rede de Hopfield
4. Aprendizagem Profunda

Inspiração Biológica

Sistema Nervoso

O **cérebro** e o **sistema nervoso** atuam como um **processador de informação** nos seres vivos.

No ser humano ele é composto por cerca de 10 bilhões de neurônios.

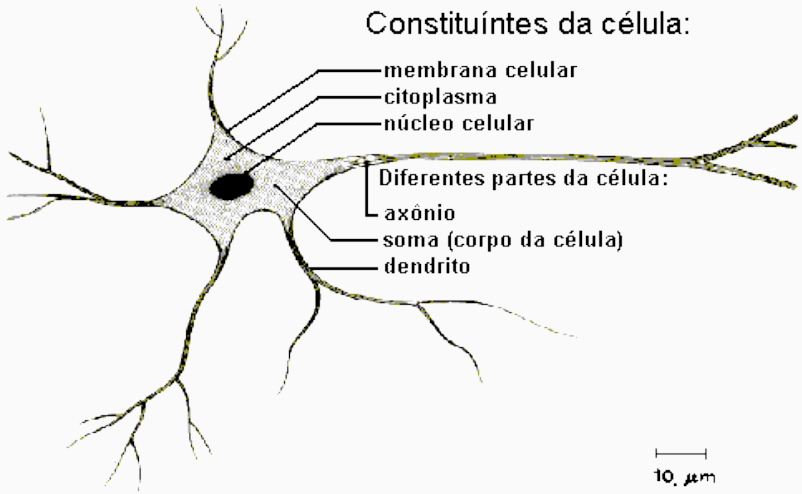
Cada neurônio está interligado a cerca de 5000 outros neurônios através de estruturas conhecidas como **sinapses** .

Neurônio

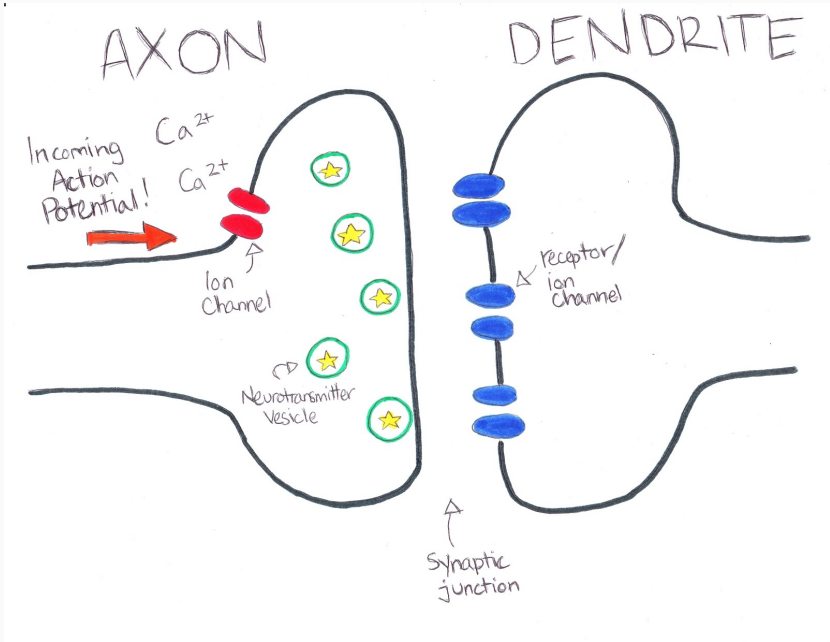
Dendritos : recebe estímulo de outros neurônios

Somma : combina as informações recebidas

Axônio : envia estímulos para outros neurônios



Sinapse



http://thebraingeek.blogspot.com.br/2012/04/synapse-2-synaptic-junction.html

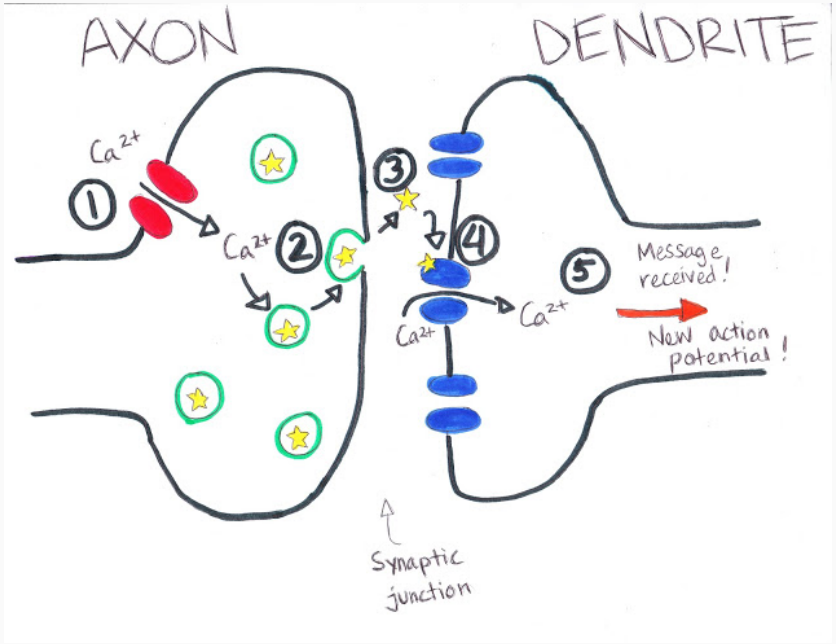
O cálcio (Ca2+) com potencial positivo

abre as portas no axônio.

Com o aumento na quantidade de Ca2+ as vesículas são abertas.

Neurotransmissores fluem até a junção.

Sinapse



Neurotransmissores se juntam aos receptores e abrem as portas deixando entrar cálcio, sódio e outros íons positivos no dendrito.

O neurônio destino fica carregado positivamente causando um impulso elétrico ativando o neurônio.

Neurônio Artificial

Neurônio Artificial

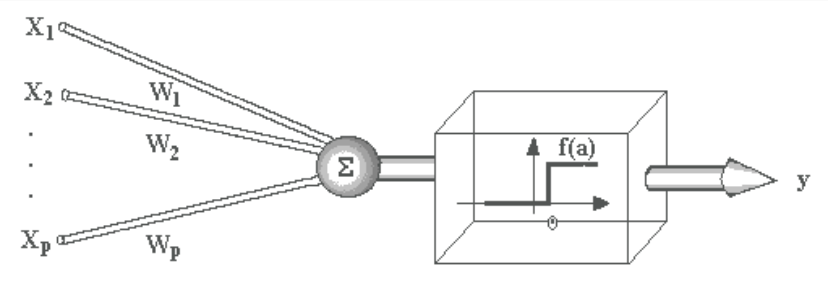
Para:

- Tentar entender o funcionamento do neurônio no processamento de informação e;
- Tentar utilizar esse modelo para processamento de informação computacional.

O neurofisiologista McCulloch e o matemático Walter Pitts criaram um modelo matemático/computacional.

Neurônio Artificial

Esse neurônio permitia múltiplas entradas de valores binários (simulando pulsos elétricos) e uma saída (resultado do processamento da informação).

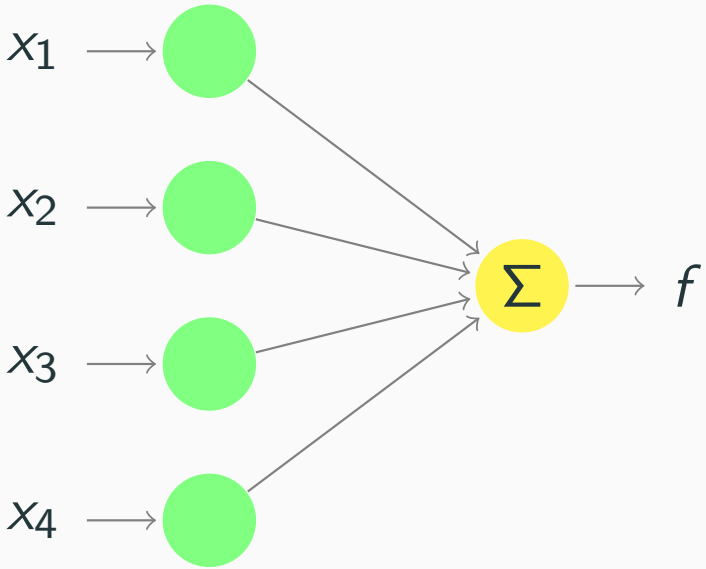


Modelo de Neurônio Artificial

Podemos criar um neurônio artificial como um *nó* de um Grafo que recebe múltiplas entradas e emite uma saída.

A saída é definida como a aplicação de uma função de ativação na soma dos valores de entrada.

Modelo de Neurônio Artificial



Modelo de Neurônio Artificial

Pensando em entradas com valores reais entre 0 e 1, o neurônio é ativado sempre que a soma das entradas for maior que um determinado τ, ou seja, a função f é:

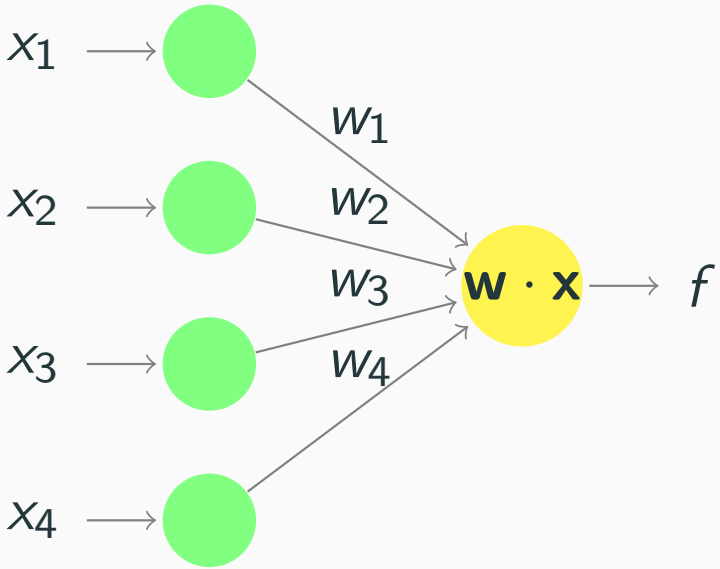
$$f(z) = \begin{cases} 0, & \text{se } z < \tau \\ g(z), & \text{c. c.} \end{cases},$$

com $g(z)$ a função de ativação que determinar o pulso a ser enviado e z a soma dos estímulos de entrada.

Modelo de Neurônio Artificial

Também é possível ponderar a importância dos estímulos de entrada do neurônio através de um vetor de pesos $\mathbf{w}$, substituindo a somatória pelo produto interno $\mathbf{w} \cdot \mathbf{x}$:

Modelo de Neurônio Artificial



Modelo de Neurônio Artificial

O Neurônio Artificial é conhecido como Percéptron. Esse tipo de modelo é dito **paramétrico** pois é descrito por um número finito de parâmetros (**w**).

Esse modelo consegue descrever apenas relações lineares.

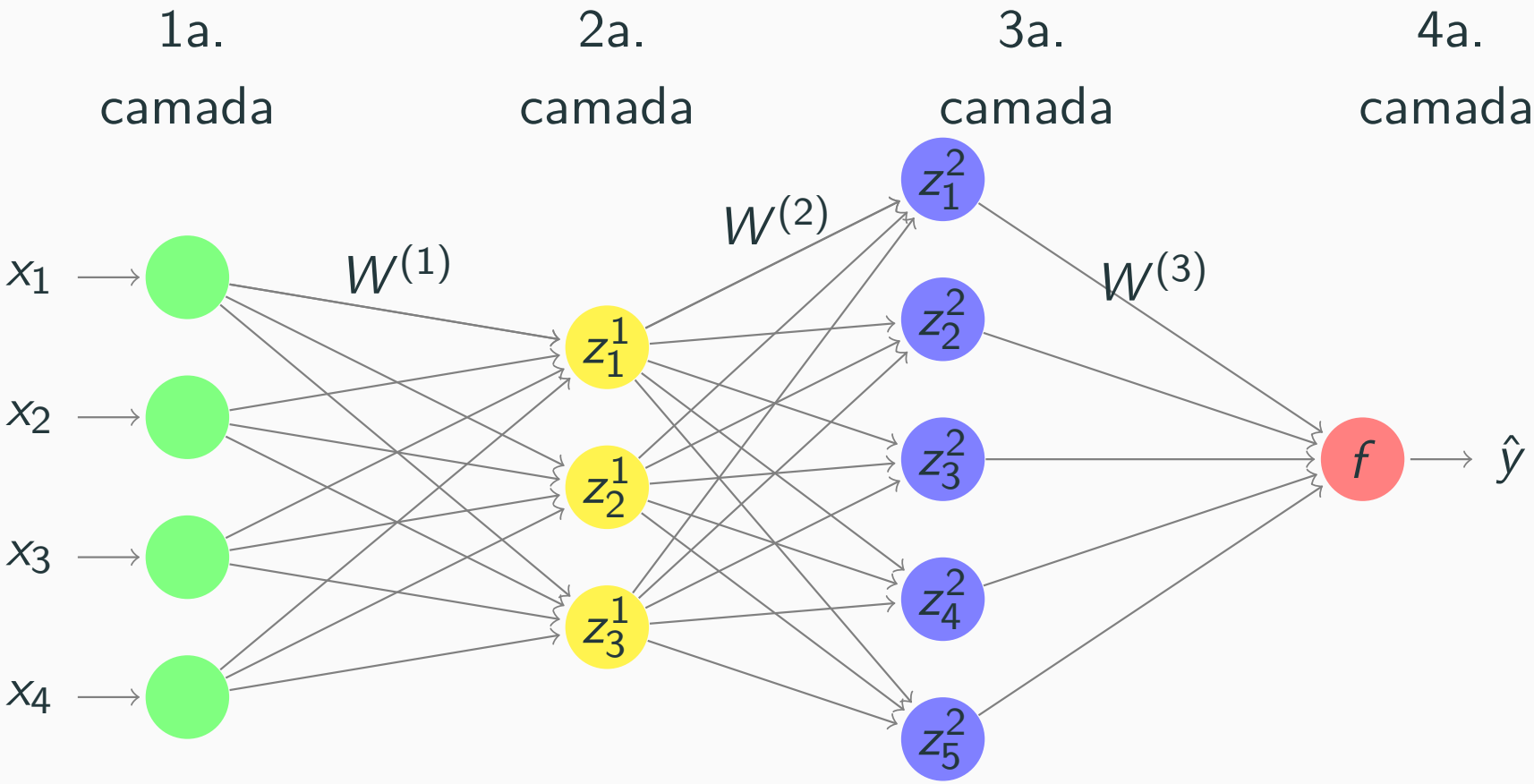
Percéptron de Múltiplas Camadas

Conhecido como **Multi-Layer Perceptron** (MLP) ou **Feedforward Neural Network**.

Rede composta de vários neurônios conectados por *camadas*.

Percéptron de Múltiplas Camadas

O uso de camadas permite aproximar funções não-lineares.



Percéptron de Múltiplas Camadas

Essa é uma forma de Aprendizagem Profunda (*Deep Learning*).

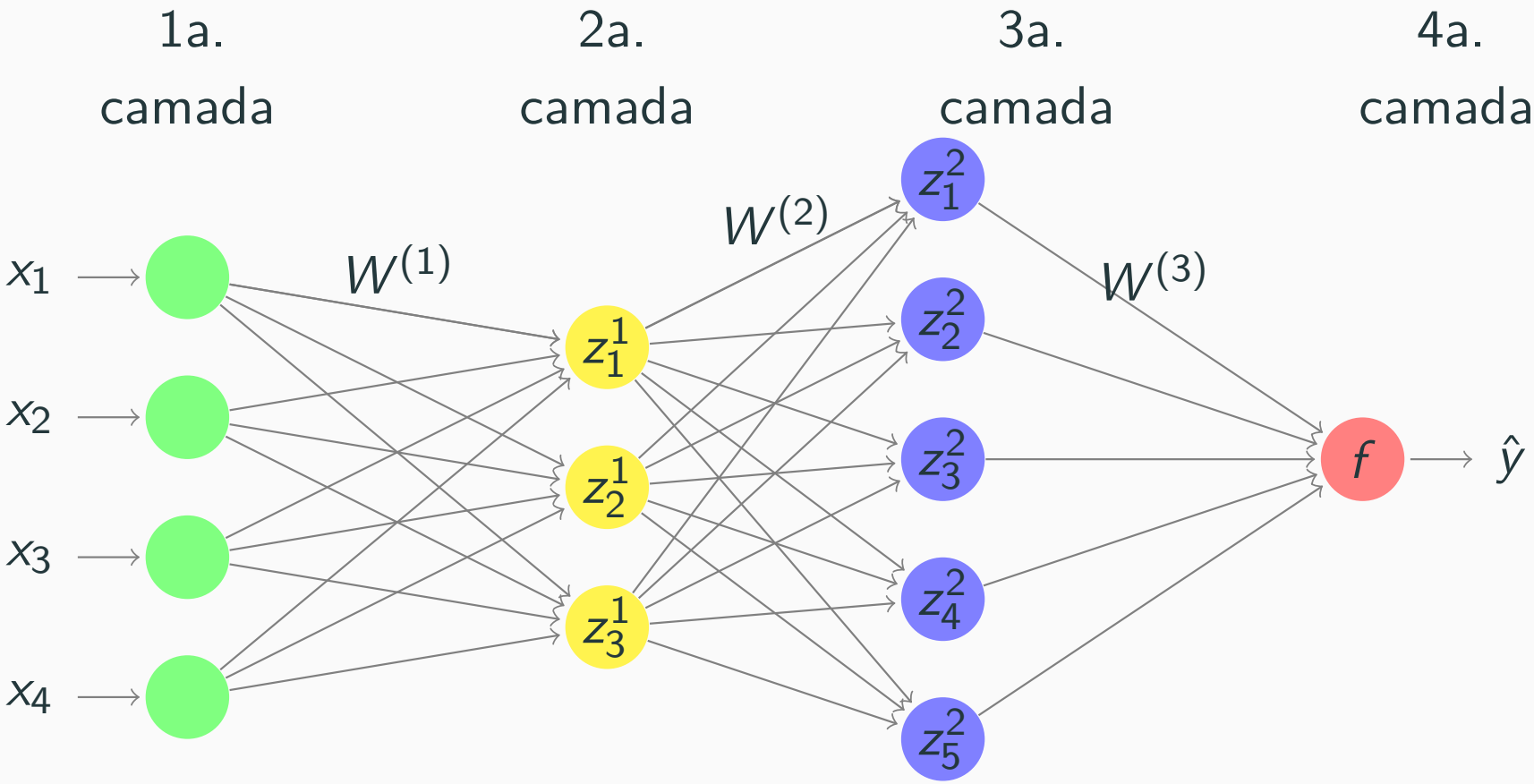
Quanto mais camadas, mais profunda é a rede.

Cada camada tem a função de criar novos atributos mais complexos.

Essa rede define um modelo computacional.

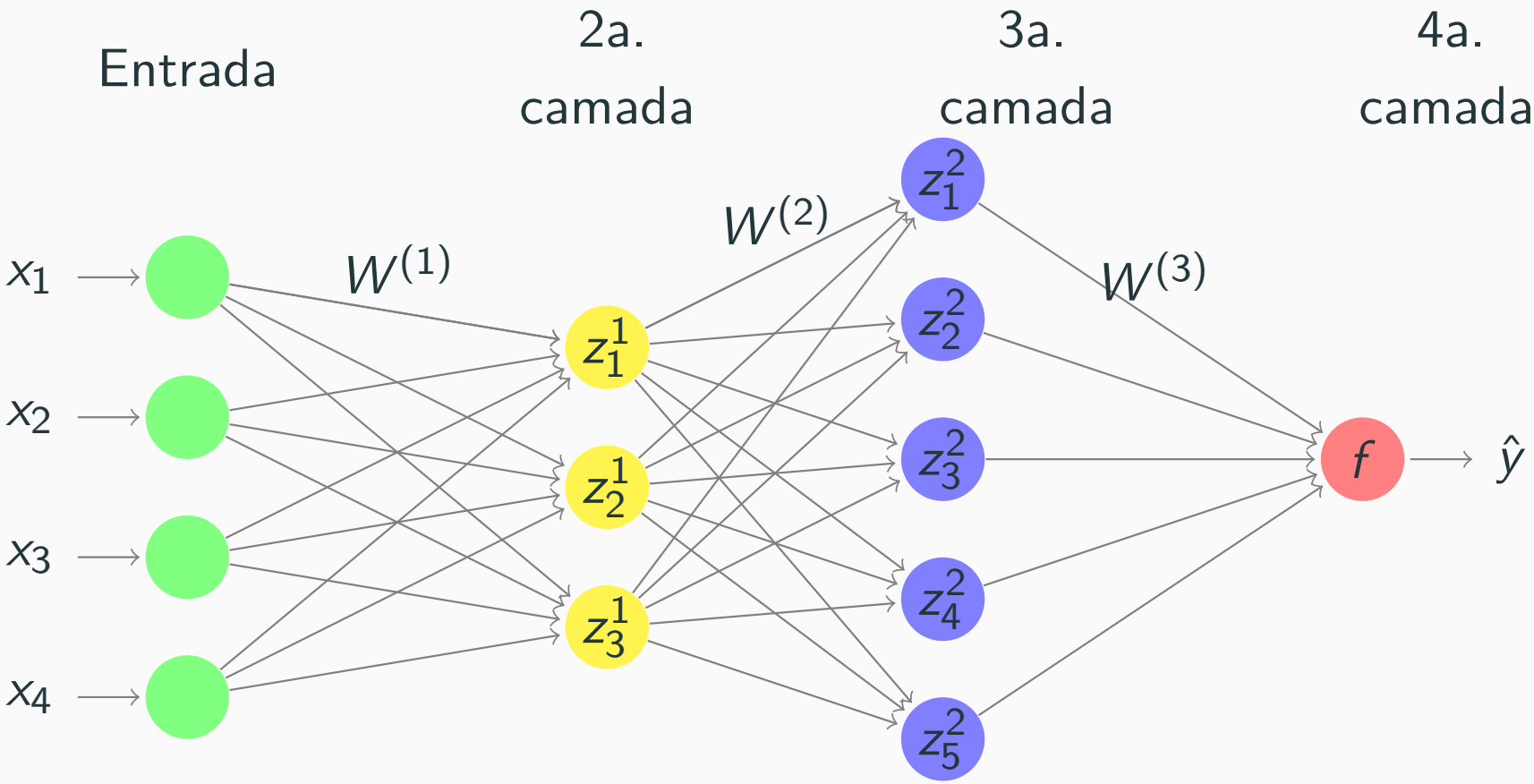
Definições

Cada conjunto de nós é denominado como uma *camada*.



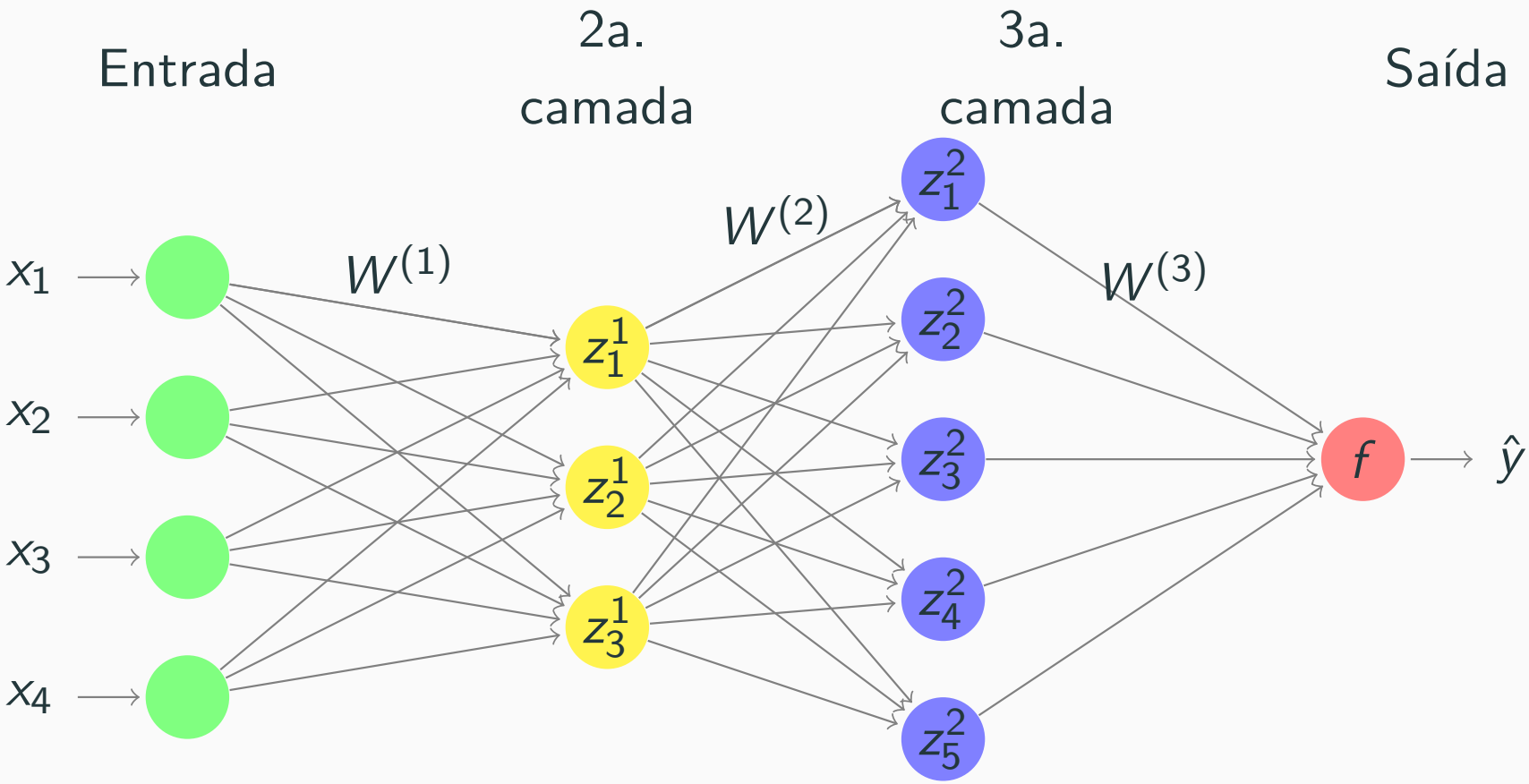
Definições

A primeira camada é conhecida como *entrada*.



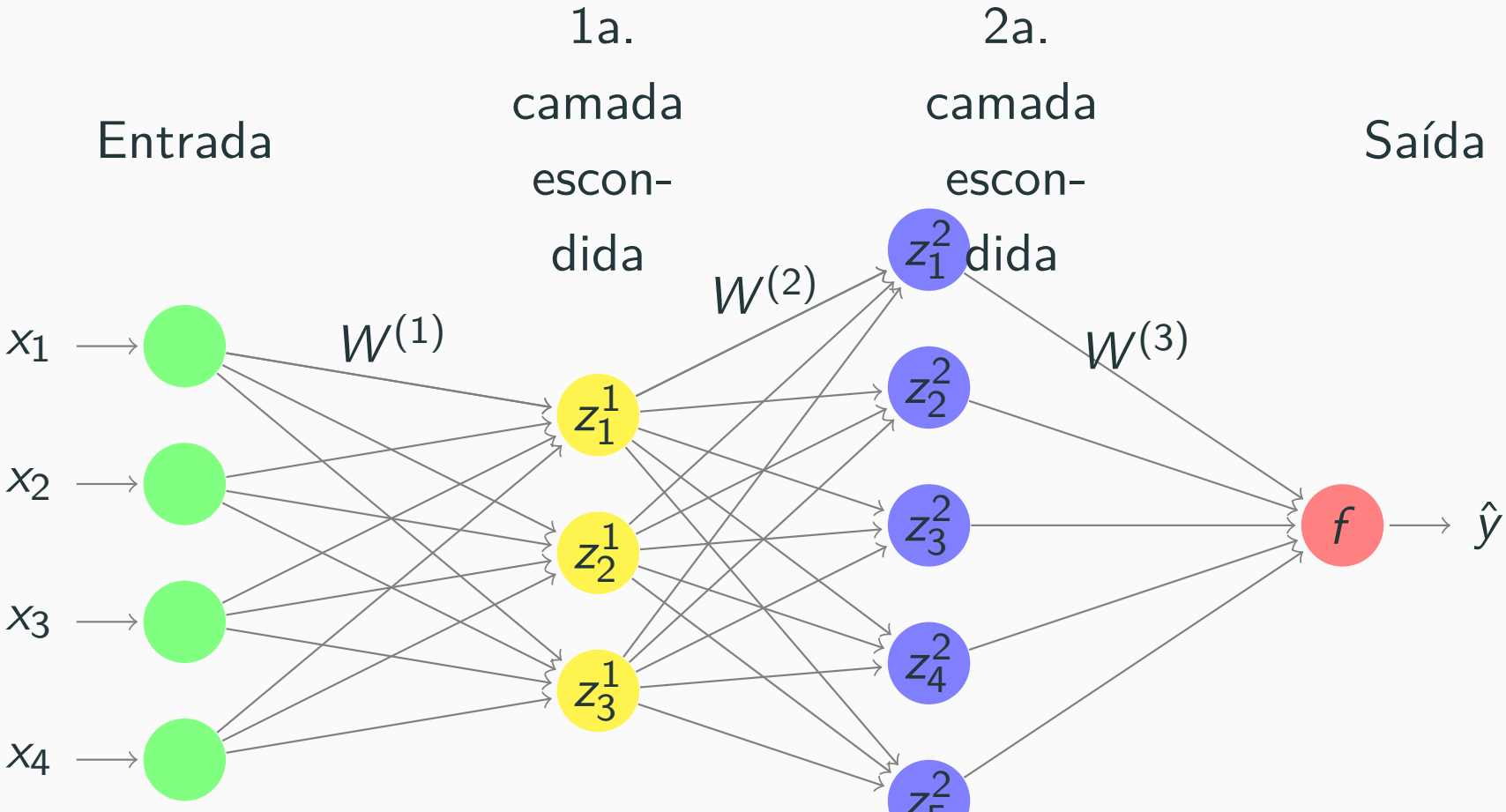
Definições

A última camada é a *saída* ou *resposta* do sistema.



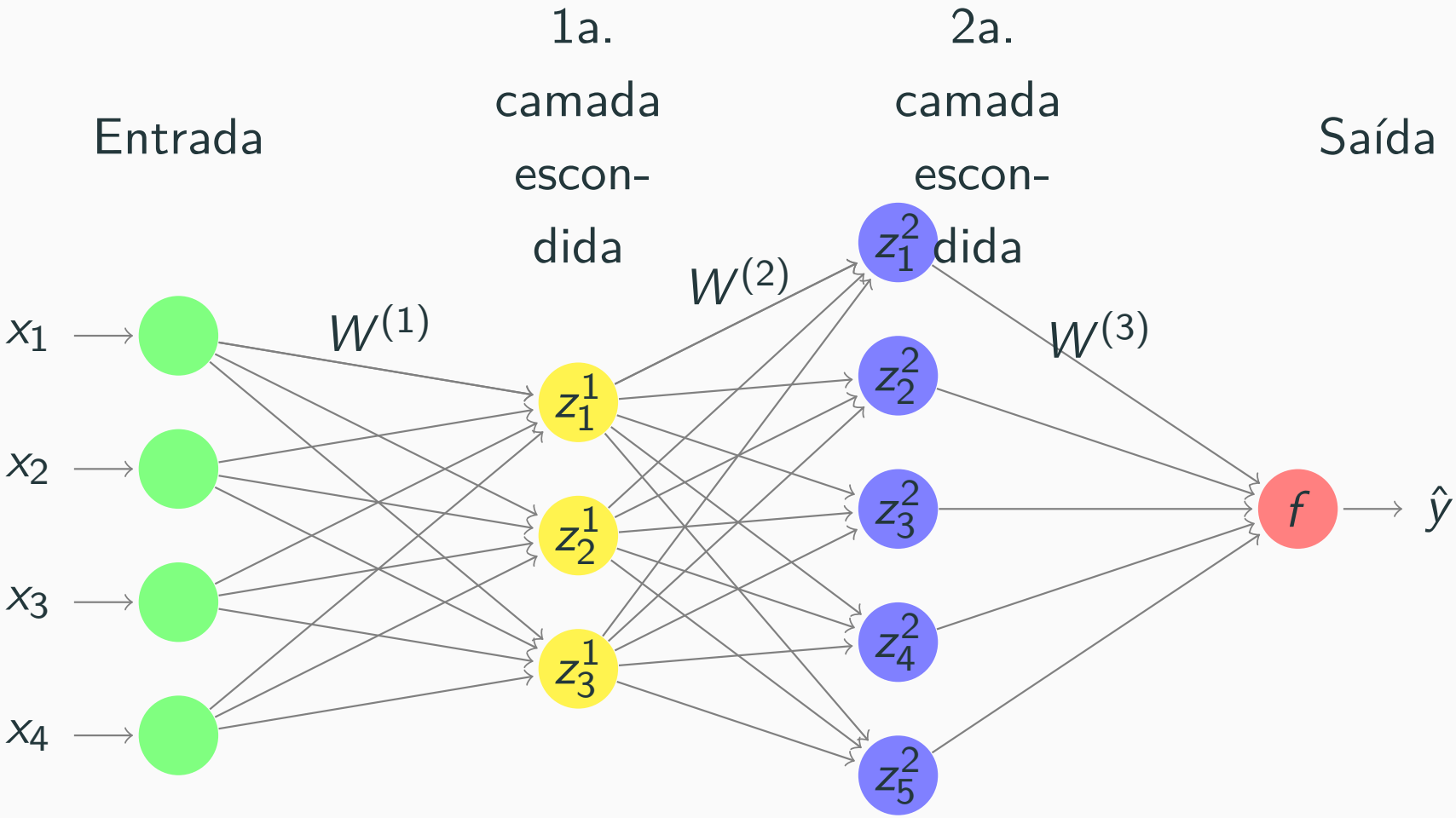
Definições

As camadas restantes são numeradas de 1 a m e são conhecidas como *camadas escondidas* ou *hidden layers*. Esse nome vem do fato de que elas não apresentam um significado explícito de nosso problema.



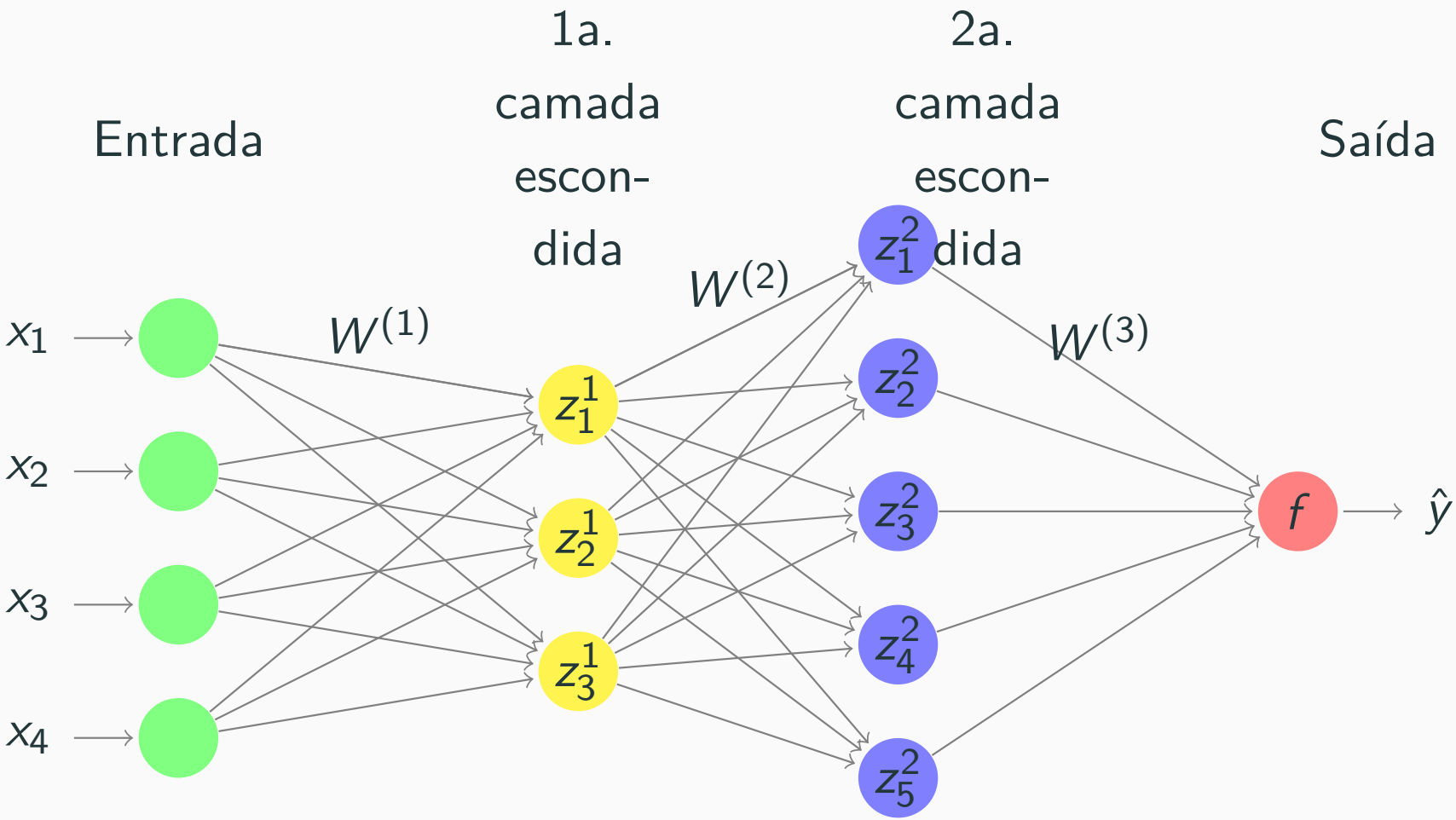
Definições

Duas camadas vizinhas costumam ser totalmente conectadas formando um grafo bipartido (em algumas variações podemos definir menos conexões).



Definições

Esse grafo é ponderado e os pesos são definidos por uma matriz W de dimensão $n_o \times n_d$, com n_o sendo o número de neurônios da camada de origem e n_d da de destino.



Definições

Assumindo que as entradas são representadas por um vetor linha $\mathbf{x}$, o processamento é definido como: $\mathbf{z^1} = f^{(1)}(\mathbf{x} \cdot W^{(1)})$.

Exercício

Dado que a camada i tem m neurônios e a camada j tem n neurônios. Determine as dimensões de z^i, $W^{(j)}$, e z^j.

Definições

A camada seguinte calcula a próxima saída como $\mathbf{z^2} = f^{(2)}(\mathbf{z^1} \cdot W^{(2)})$, e assim por diante.

Processamento em Lote

Se temos uma matriz de dados $X \in \mathbb{R}^{n \times d}$ e quiséssemos obter uma saída $\mathbf{y} \in \mathbb{R}^{n \times 1}$ em uma Rede Neural com duas camadas escondidas contendo $h1$ e $h2$ neurônios, qual seria a sequência de processamento?

(quais cálculos temos que fazer?)

Processamento em Lote

Se temos uma matriz de dados $X \in \mathbb{R}^{n \times d}$ e quiséssemos obter uma saída $\mathbf{y} \in \mathbb{R}^{n \times 1}$ em uma Rede Neural com duas camadas escondidas contendo $h1$ e $h2$ neurônios, qual seria a sequência de processamento?

$$Z^{(1)} = f^{(1)}(X \cdot W^{(1)})$$

$$Z^{(2)} = f^{(2)}(Z^{(1)} \cdot W^{(2)})$$

$$y = f^{(3)}(Z^{(2)} \cdot W^{(3)})$$

Funções de Ativação

As funções de ativação comumente utilizadas em Redes Neurais são:

- **Linear:** $f(z) = z$, função identidade.
- **Logística:** $f(z) = \frac{1}{1+e^{-z}}$, cria uma variável em um tipo sinal, com valores entre 0 e 1.
- **Tangente Hiperbólica:** $f(z) = \tanh(z)$, idem ao anterior, mas variando entre -1 e 1.
- **Rectified Linear Units:** $f(z) = \max(0, z)$, elimina os valores negativos.
- **Softmax:** $f(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$, faz com que a soma dos valores de z seja igual a 1.

Ajustando os Parâmetros

Para determinar os valores corretos dos pesos, utilizamos o Gradiente Descendente, assim como nos algoritmos de Regressão Linear e Logística.

Ajustando os Parâmetros

Note porém que o cálculo da derivada da função de erro quadrático é igual aos casos já estudados apenas na última camada.

Ajustando os Parâmetros

Para as outras camadas precisamos aplicar o algoritmo de **Retropropagação** que aplica a regra da cadeia.

Retropropagação

O algoritmo segue os seguintes passos:

- Calcula a saída para uma certa entrada.
- Calcula o erro quadrático.
- Calcula o gradiente do erro quadrático em relação a cada peso.
- Atualiza pesos na direção oposta do gradiente.
- Repita.

Retropropagação

Assumindo a função de ativação logística $\sigma(z)$, cuja derivada é $\sigma(z)(1 - \sigma(x))$, o gradiente da camada de saída é $\frac{\partial e}{\partial W^{(3)}} = (\hat{y} - y) \cdot z^2$.

Retropropagação

O gradiente da camada anterior é calculado como:
$\frac{\partial e}{\partial W^{(2)}} = (\hat{y} - y) \cdot W^{(3)} \cdot \sigma'(x \cdot W^{(1)}) \cdot z^1$. E assim por diante até a camada de entrada.

Vanishing Gradient

Determine os valores mínimos e máximos das derivadas das seguintes funções de ativação:

- Logistica: $\sigma(z) = \frac{1}{1+e^{-z}}$ cuja derivada é $\sigma'(z) = \sigma(z)(1 - \sigma(z))$.
- Tangente Hiperbólica: $\tanh(z)$ cuja derivada é $\tanh'(z) = 1 - \tanh^2(z)$.
- RELU: $relu(z) = max(0, z)$ cuja derivada é $relu'(z) = 1, 0$, se $z > 0$ ou caso contrário, respectivamente.

Vanishing Gradient

Como $0 \leq \sigma(z) \leq 1$, temos que o valor mínimo da derivada é quando $\sigma(z) = \{0, 1\}$ em que $\sigma'(z) = 0$.

O valor máximo ocorre quando $\sigma(z) = 0.5$ e $\sigma'(z) = 0.25$.

Vanishing Gradient

Como $-1 \leq \tanh(z) \leq 1$, temos que o valor mínimo da derivada é quando $\tanh(z) = \{-1, 1\}$ em que $\tanh'(z) = 0$.

O valor máximo ocorre quando $\tanh(z) = 0$ e $\tanh'(z) = 1$.

Vanishing Gradient

Como $relu'$ assume apenas dois valores, temos diretamente que o mínimo e máximo são 0 e 1, respectivamente.

Vanishing Gradient

O gradiente de cada camada quando utilizamos a função logística, terá o valor de no máximo 25% da camada seguinte, ou seja, quanto mais camadas, menores os valores de gradientes das primeiras camadas.

Vanishing Gradient

Isso é conhecido como *Vanishing Gradient* e é possível remediar utilizando outras funções de ativação como *tanh* e RELU.

Dicas para Melhorar o desempenho do ajuste

- Utilize tanh como função sigmoidal.
- Utilize *softmax* para multi-classes.
- Escale as variáveis de saída para a mesma faixa de valores da segunda derivada da função de ativação (ex.: para tanh deixe as variáveis entre -1 e 1).

Dicas para Melhorar o desempenho do ajuste

- Ajuste os parâmetros utilizando mini-batches dos dados de treinamento.
- Inicialize os pesos como valores aleatórios uniformes com média zero e desvio-padrão igual a $\frac{1}{\sqrt{m}}$, com m sendo o número de nós da camada anterior.

Rede de Hopfield

Aprendizado Hebbiano

Em 1949, Hebb escreveu um livro intitulado “Organization of Behavior”.

“Quando um axônio de uma célula A está próxima de excitar uma célula B e repetidamente ou persistentemente toma parte em ativá-la, algum processo crescente ou mudança metabólica se apossa de uma ou ambas as células de forma que a eficiência de A, assim como a de uma das células B excitadas, são aumentadas”.

Aprendizado Hebbiano

Ou seja, dado que o neurônio A ativa o neurônio B frequentemente, a sinapse (peso na rede artificial) é aumentada para estimular a ativação.

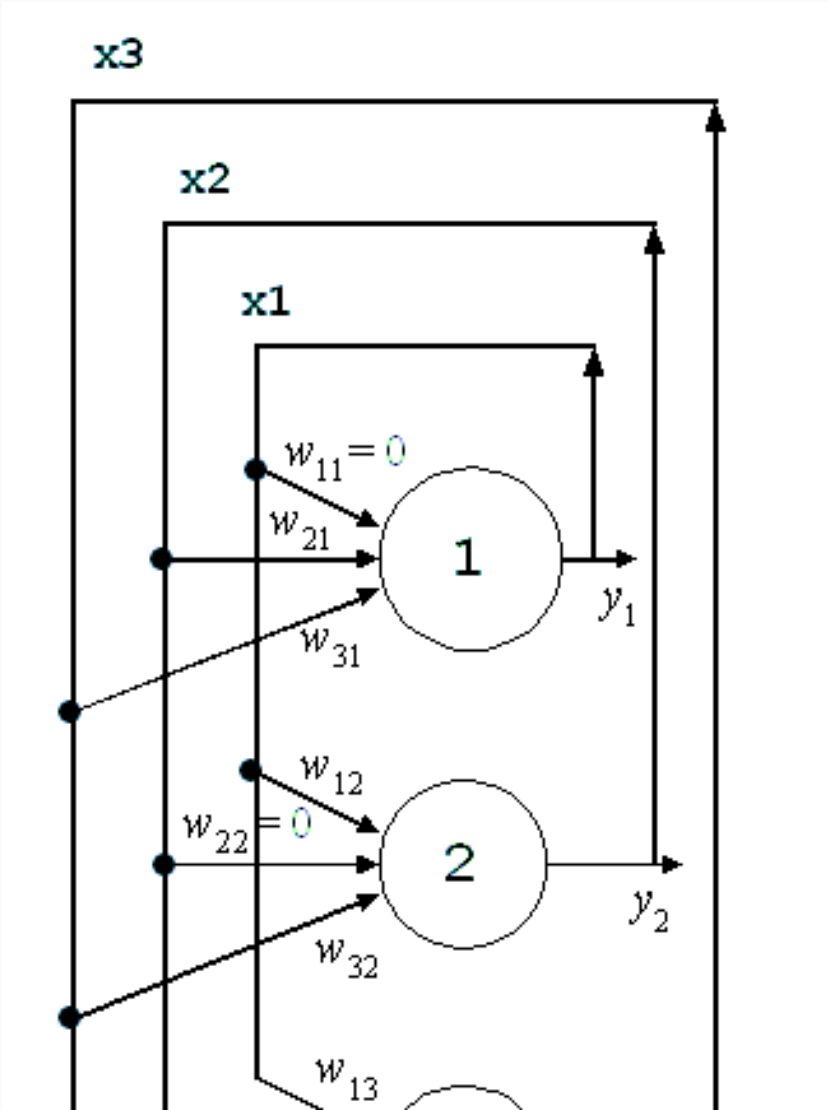
Essa teoria está relacionada ao processo de memória associativa.

Aprendizado Hebbiano

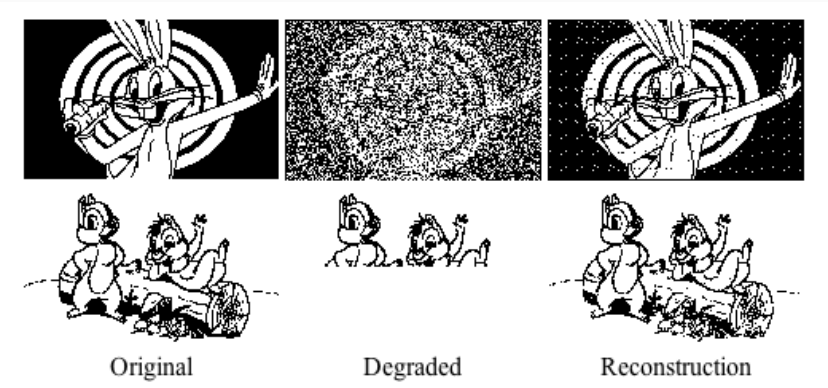
Com a memória associativa podemos resgatar memórias com apenas parte de uma informação.

Rede Hopfield

Esse efeito é simulado artificialmente através da rede de hopfield, ou rede recorrente, onde a saída dos neurônios alimentam a entrada.



Rede Hopfield



Aprendizagem Profunda

Aprendizagem Profunda

Teoria da dinâmica de aprendizado no cérebro.

Camadas de redes neurais que aprendem hierarquicamente apenas parte da informação.

Processo de auto-organização, resultado final é o aprendizado completo.

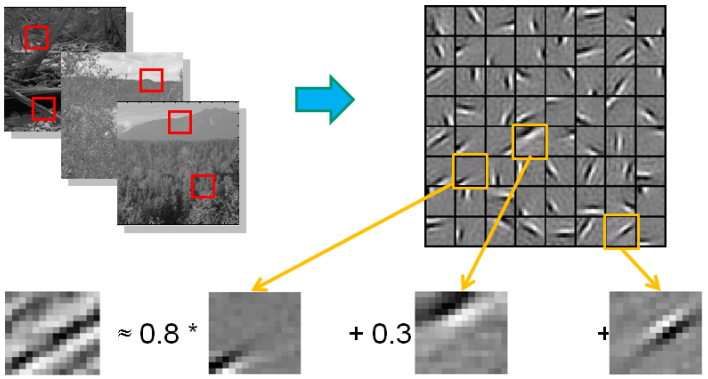
Ocorre durante o período natal e em partes do pós-natal.

Essa teoria é uma explicação para a inteligência humana em comparação com outros animais, inclusive primatas.

O período em que o aprendizado por camadas ocorre é um pouco maior do que em outras espécies, permitindo uma maior capacidade na extração e processamento de informação proveniente de estímulos externos.

Porém isso nos torna mais dependentes de nossos pais por um maior período de tempo.

Sparse coding illustration

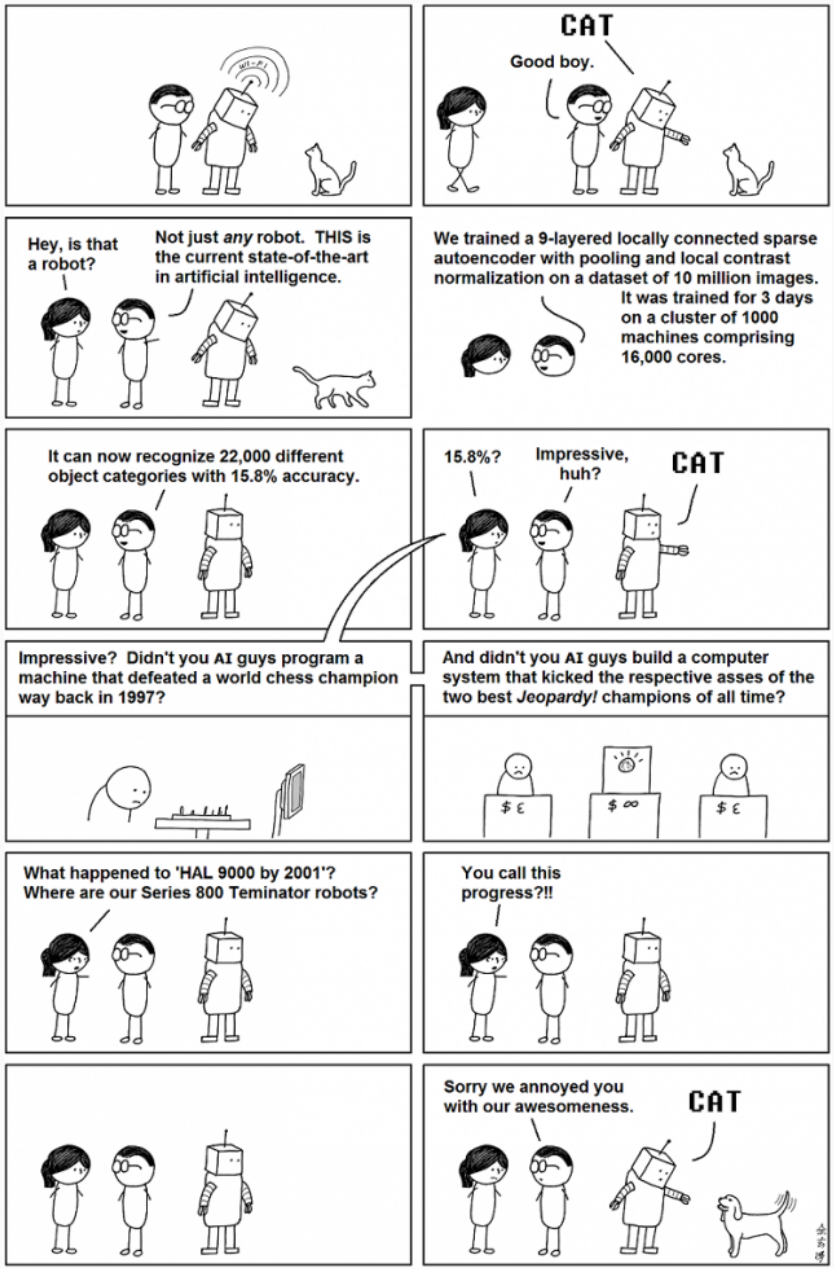


$[a_1, \ldots, a_{64}] = [0, 0, \ldots, 0, \mathbf{0.8}, 0, \ldots, 0, \mathbf{0.3}, 0, \ldots, 0, \mathbf{0.5}, 0]$
(feature representation)

Slide credit: Andrew Ng

Compact & easily interpretable

Aprendizagem Profunda



Aprendizagem Profunda