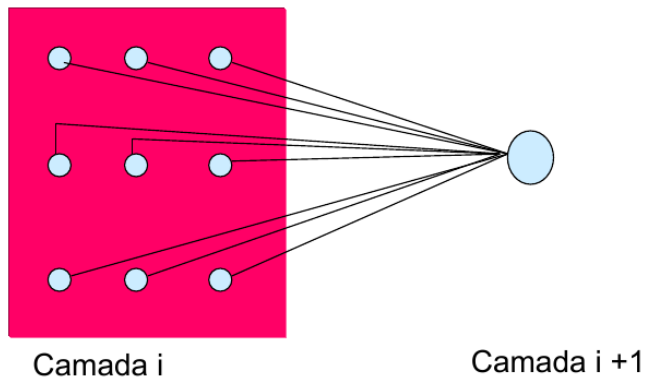


Aprendizado de Máquina: RNAs Profundas (Deep Learning)

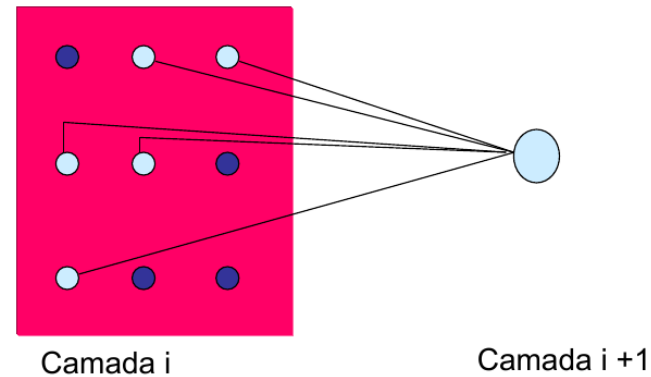
Prof. Arnaldo Candido Junior
UTFPR – Medianeira

Redes convolucionais

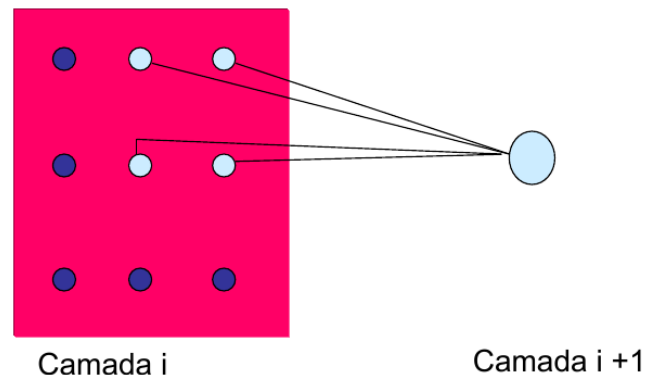
- Completamente conectada:



- Parcialmente conectada:



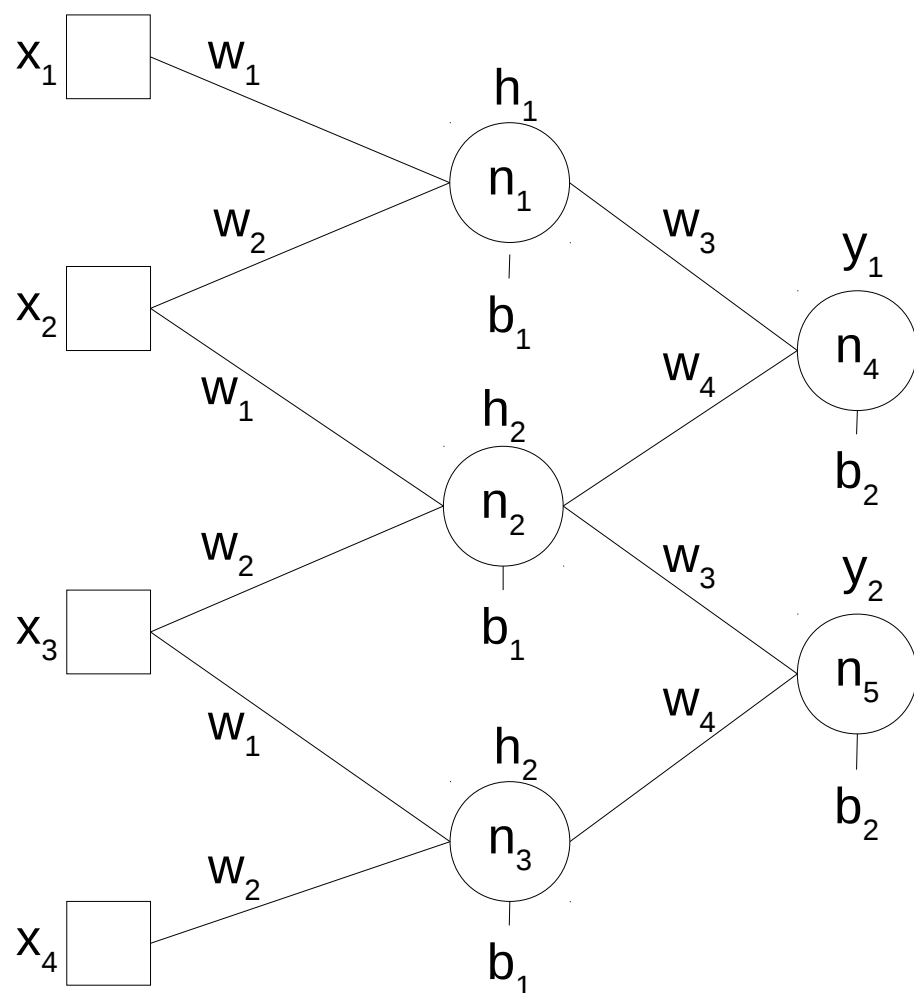
- Localmente conectada:



Redes convolucionais ⁽²⁾

- Até o momento estudamos redes MLP completamente conectadas
 - Os neurônios de cada camada se comunicam com todos os neurônios das camadas vizinhas
- Redes Convolucionais (CNNs) são redes MLP parcialmente conectadas (localmente conectadas)
 - Estão revolucionando a área de Visão Computacional

Redes convolucionais ⁽³⁾



- Exemplo: convolução 1D

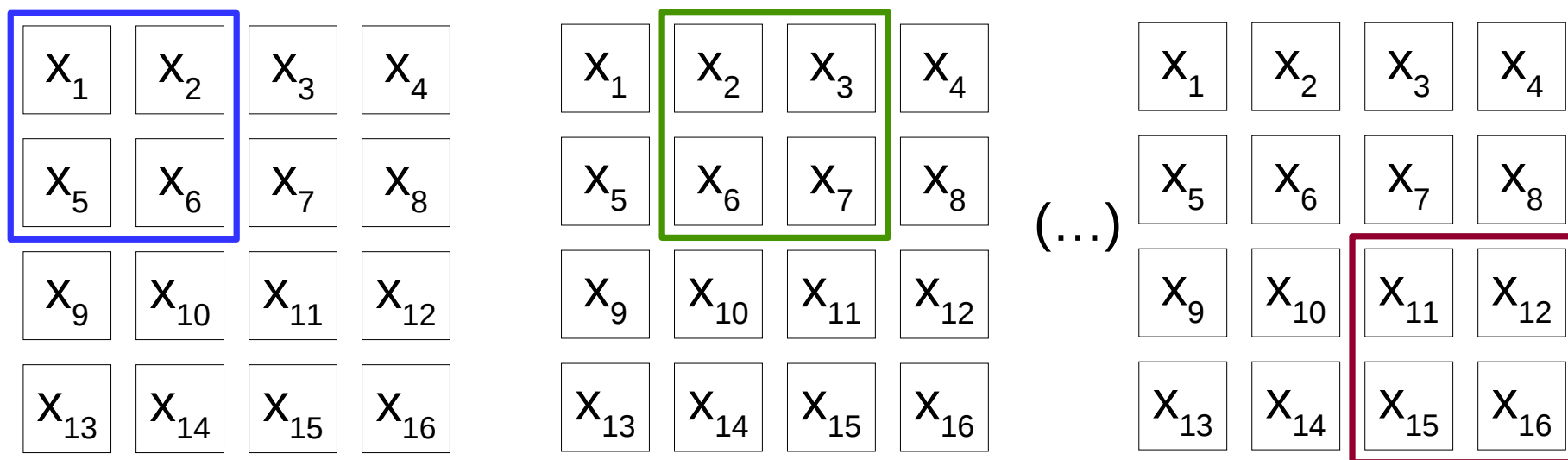
$$\begin{aligned}
 v_1 &= w_1 x_1 + w_2 x_2 + b_1; h_1 = f(v_1) \\
 v_2 &= w_1 x_2 + w_2 x_3 + b_1; h_2 = f(v_2) \\
 v_3 &= w_1 x_3 + w_2 x_4 + b_1; h_3 = f(v_3) \\
 v_4 &= w_3 h_1 + w_4 h_2 + b_2; y_1 = f(v_4) \\
 v_5 &= w_3 h_2 + w_4 h_3 + b_2; y_2 = f(v_5)
 \end{aligned}$$

Redes convolucionais ⁽⁴⁾

- Vantagens da camada convolucional em relação a totalmente conectada:
 - Menos pesos (localmente conectada)
 - Repetição (compartilhamento de pesos)
 - Resultado: treinamento mais eficiente
- n_1 , n_2 e n_3 podem ser pensados como clones que analisam diferentes partes da entrada
 - Um objeto pode estar em qualquer parte da imagem

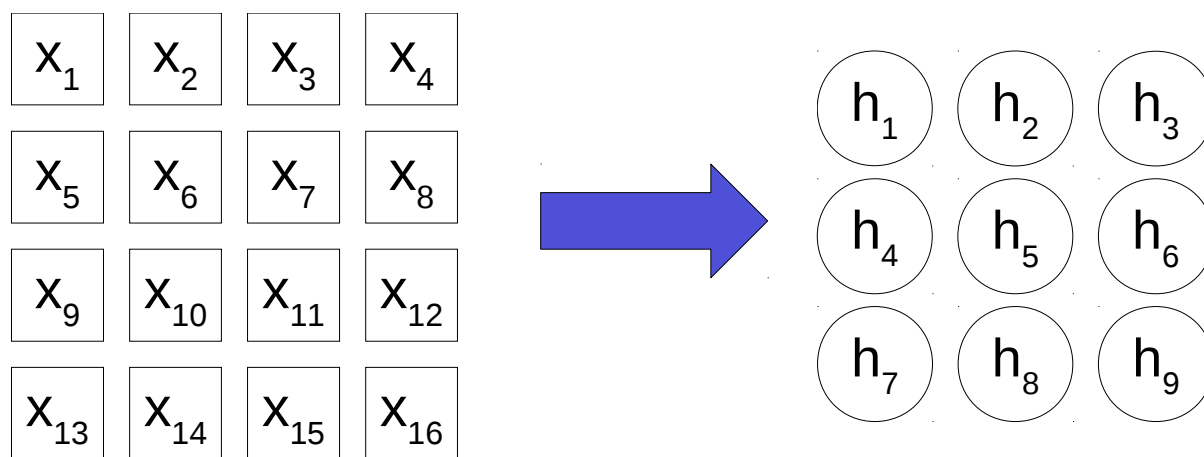
Redes convolucionais ⁽⁵⁾

- Convolução 2D:



Redes convolucionais ⁽⁶⁾

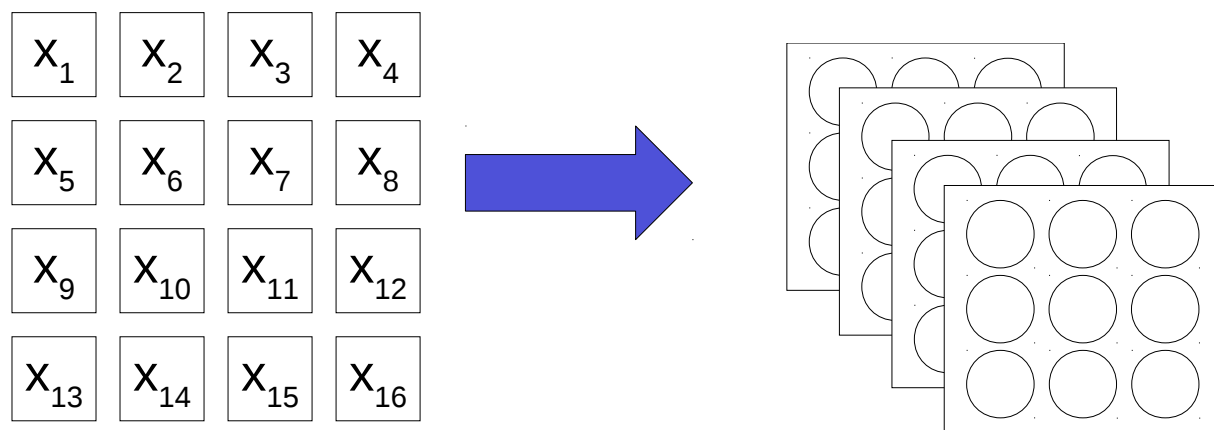
- Camadas convolucionais 2D preservam a estrutura bidimensional da entrada (ideal para imagens)



- Importante: a camada de exemplo usa um **kernel** 2x2 (tamanho do quadrado)

Redes convolucionais ⁽⁷⁾

- A camada de exemplo usa apenas um **filtro**: faz apenas uma operação de convolução
- Camadas reais usam vários filtros. Exemplo com 4 filtros:

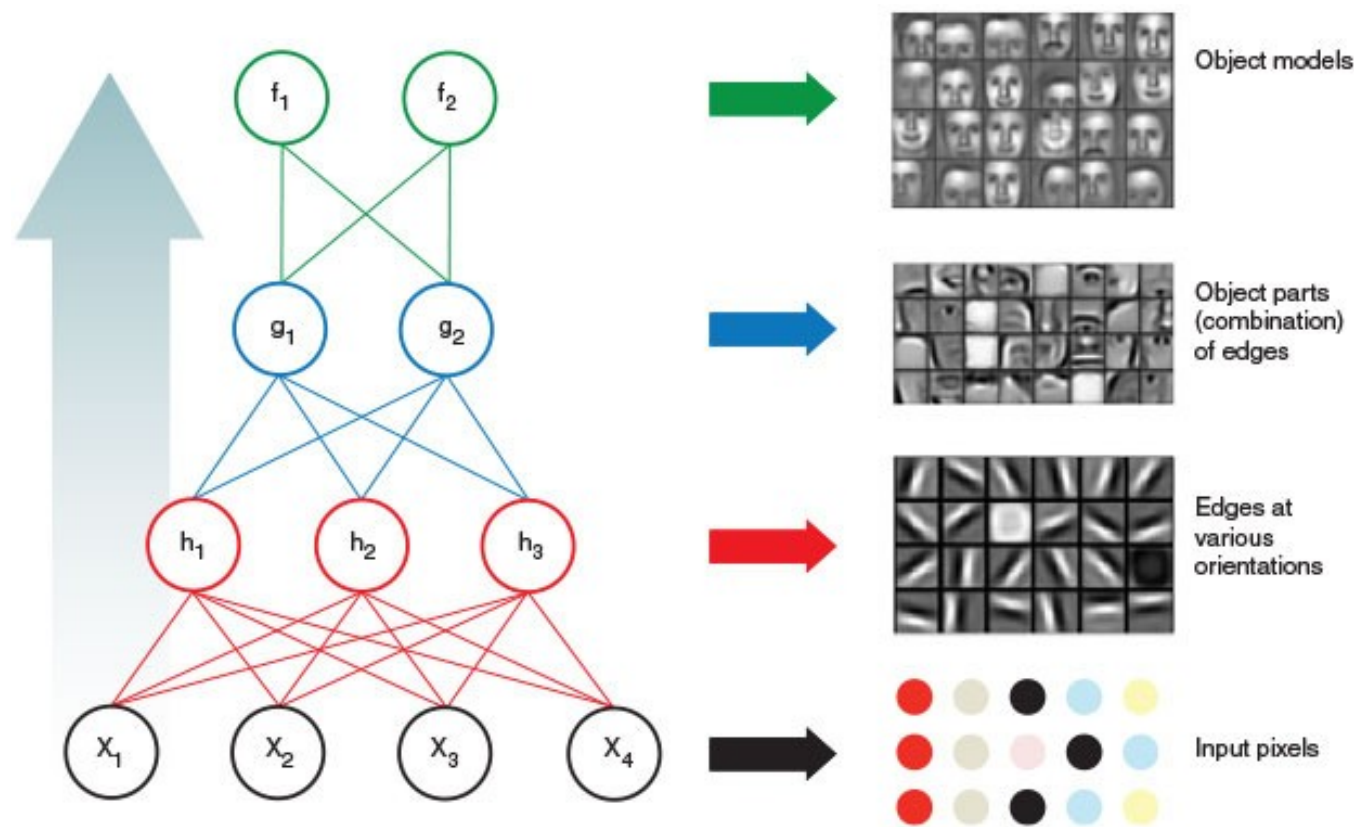


- Obs: não há pesos repetidos em filtros diferentes

Redes convolucionais ⁽⁸⁾

- Exemplos de tarefas dos filtros por camada (**hierarquias de conceitos** das CNNs):

- bordas
- formas simples
- (...)
- olhos, naris, boca
- faces



Redes convolucionais ⁽⁹⁾

- Os kernels em cada filtro
 - Simplificam a entrada
 - Extraem características
 - Exemplo:
kernel 3x3

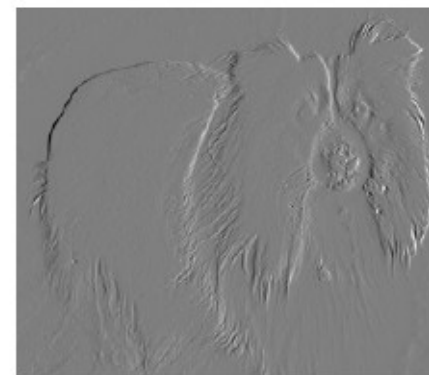
Input image



Convolution
Kernel

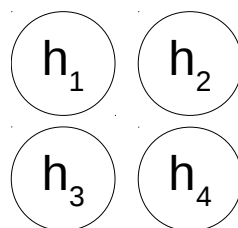
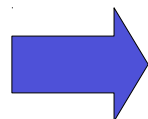
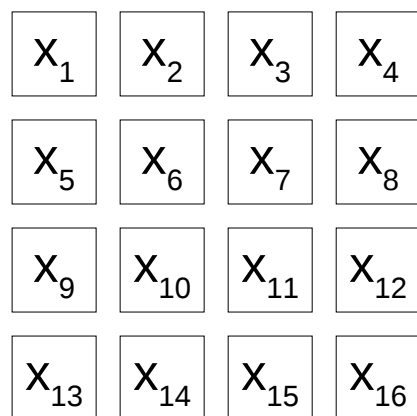
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map



Redes convolucionais ⁽¹⁰⁾

- O parâmetro **stride** (passo) determina o pulo do kernel. No primeiro exemplo foi usado stride 1
- Exemplo: stride 2 (com kernel 2x2)



$$v_1 = w_1 x_1 + w_2 x_2 + w_3 x_5 + w_4 x_6 + b$$

$$v_2 = w_1 x_3 + w_2 x_4 + w_3 x_7 + w_4 x_8 + b$$

$$v_3 = w_1 x_9 + w_2 x_{10} + w_3 x_{13} + w_4 x_{14} + b$$

$$v_4 = w_1 x_{11} + w_2 x_{12} + w_3 x_{15} + w_4 x_{16} + b$$

Redes convolucionais ₍₁₁₎

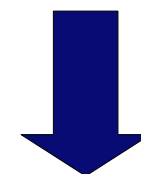
- **Padding:** consiste em aumentar a borda
- Faz a camada focar mais nas bordas
- Útil para preservar o tamanho da entrada na saída da camada
 - Exemplo: zero padding (kernel 2x2 e stride 1)

0	0	0	0	0	0
0	x_1	x_2	x_3	x_4	0
0	x_5	x_6	x_7	x_8	0
0	x_9	x_{10}	x_{11}	x_{12}	0
0	x_{13}	x_{14}	x_{15}	x_{16}	0
0	0	0	0	0	0

Redes convolucionais (12)

- **Pooling**: reduz o volume de dados que uma camada seguinte precisa processar
 - Combate a maldição da dimensionalidade
 - Dá uma ideia aproximada da posição de uma forma dentro da imagem
 - Exemplo: max-pooling 2x2 com stride 2

0.7	0.8	0.2	0.2
0.6	0.9	0.1	0.1
0.1	0.2	0.0	0.0
0.0	0.0	0.0	0.1



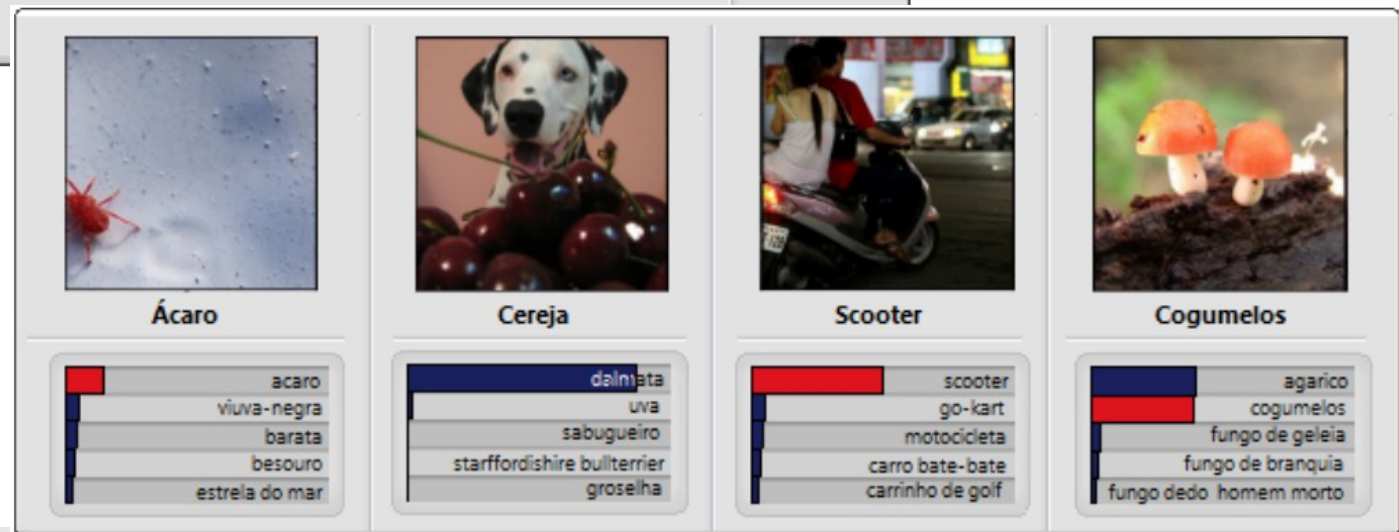
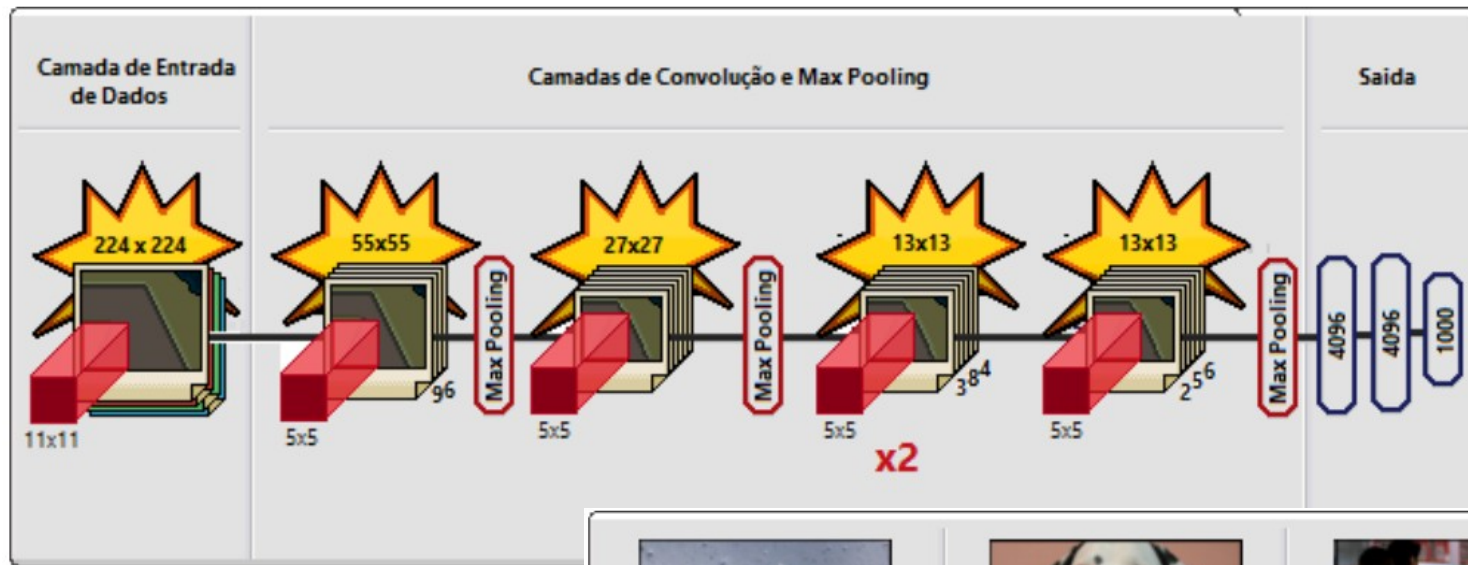
0.9	0.2
0.2	0.1

Redes convolucionais ₍₁₃₎

- No exemplo, o padrão de interesse do filtro está na segunda linha, na segunda coluna
 - Por isso, o neurônio (2, 2) é o que reage com mais intensidade
 - Contudo, dentro do filtro, todos os neurônios da vizinhança reagem fortemente ao padrão de interesse

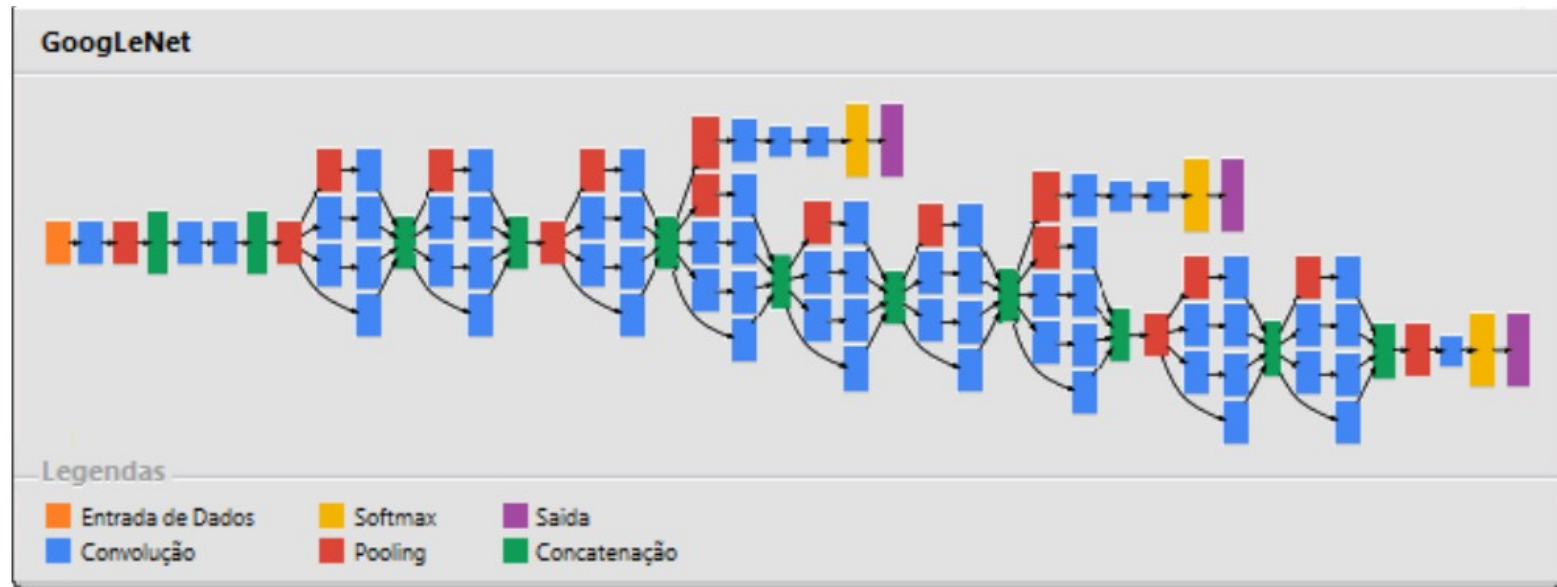
Avançado: redes famosas

- Avançado: AlexNet (classificação)



Avançado: redes famosas (2)

- Avançado: GoogLeNet ou Inception (classificação)



- Repete a camada de saída 3 vezes para lidar com o gradiente que desaparece (roxo)
- Usa vários tamanhos de kernel e concatena o resultado (vermelho)

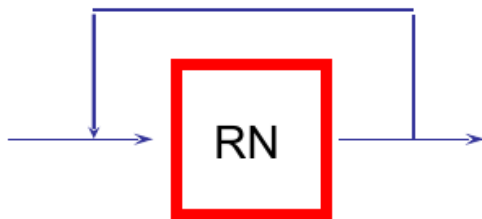
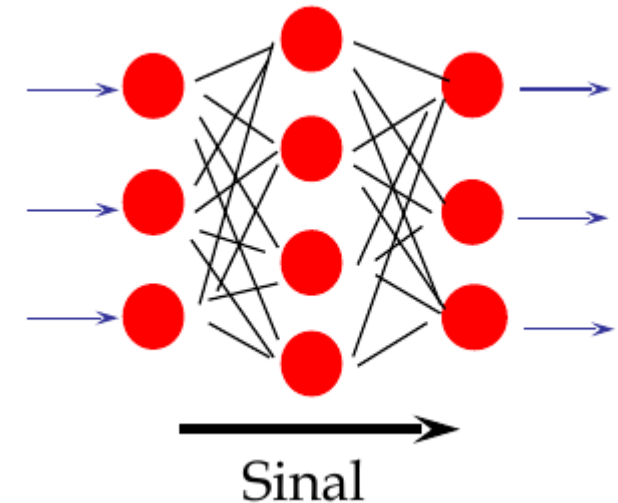
Redes convolucionais (16)

- Avançado: outras
 - Highway Network (classificação)
 - DenseNet (classificação)
 - Google Deep Dream (gerativa)
 - YOLO (identificação de objetos)

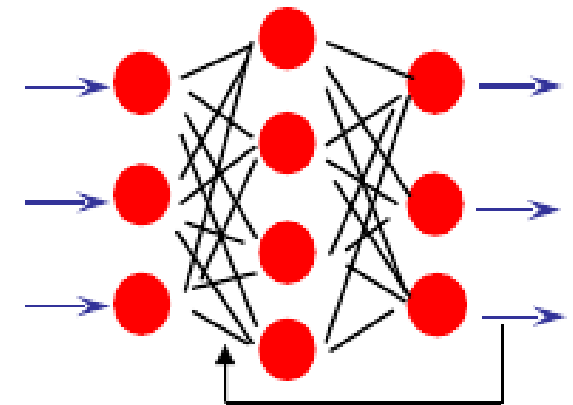
Redes recorrentes



- Rede feedforward: não existem *loops* de conexões

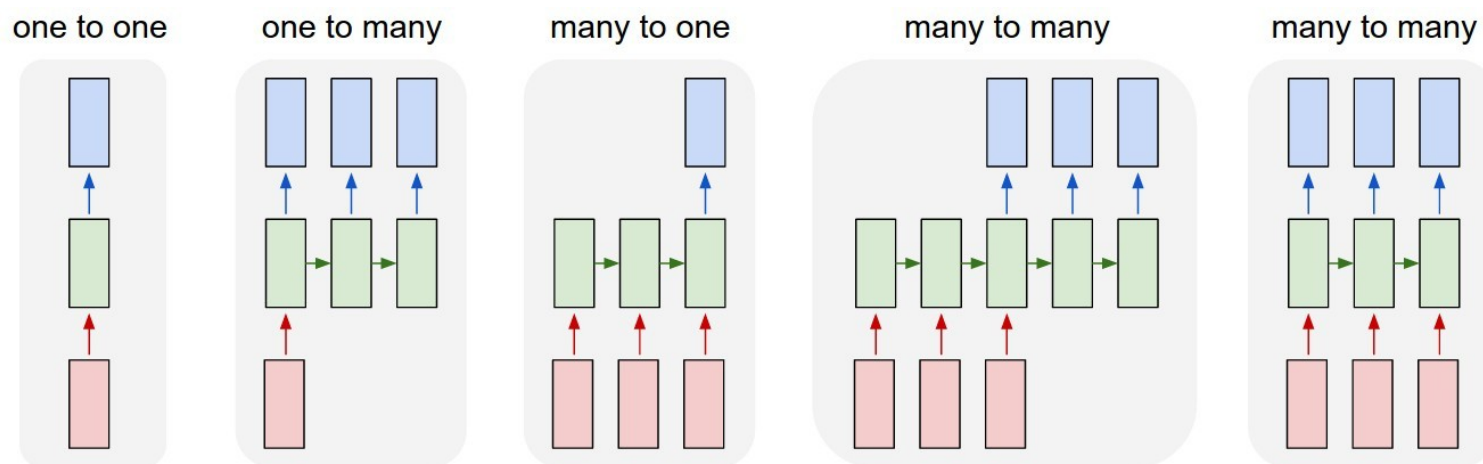


- Rede recorrentes: loops conferem uma forma de memória à rede



Redes recorrentes (2)

- $1 \rightarrow 1$: redes MLP sem recorrência
- $n \rightarrow 1$: rede vista no exemplo
- $1 \rightarrow n$: a seguir
- $m \rightarrow n$: combina as duas anteriores

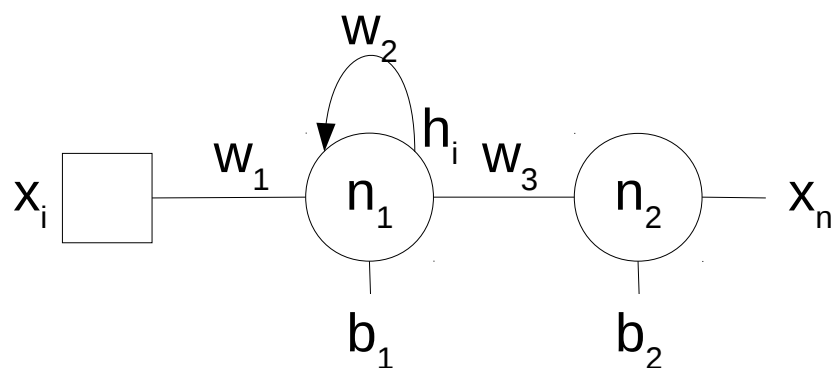


Redes recorrentes ⁽³⁾

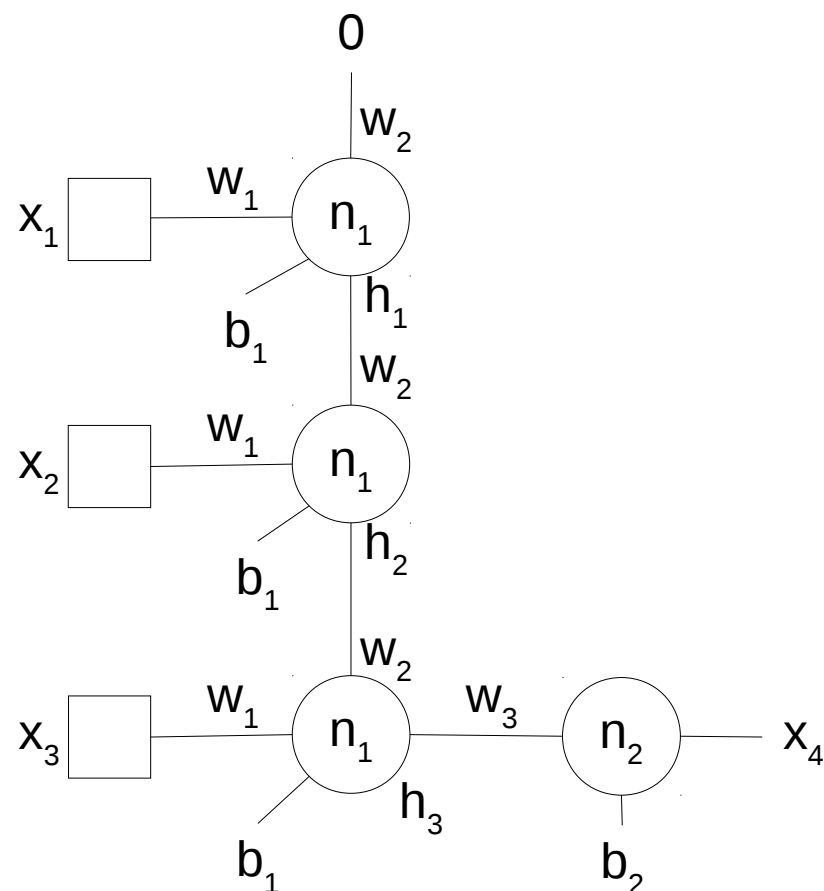
- Estão revolucionando as áreas de:
 - Séries temporais
 - Processamento de sinais
 - Processamento de língua natural
 - Entre outras
- Exemplo: análise na bolsa de valores das três últimas cotações (x_1 , x_2 e x_3) de uma ação para prever o preço (x_4)

Redes recorrentes (4)

- Representação compacta:

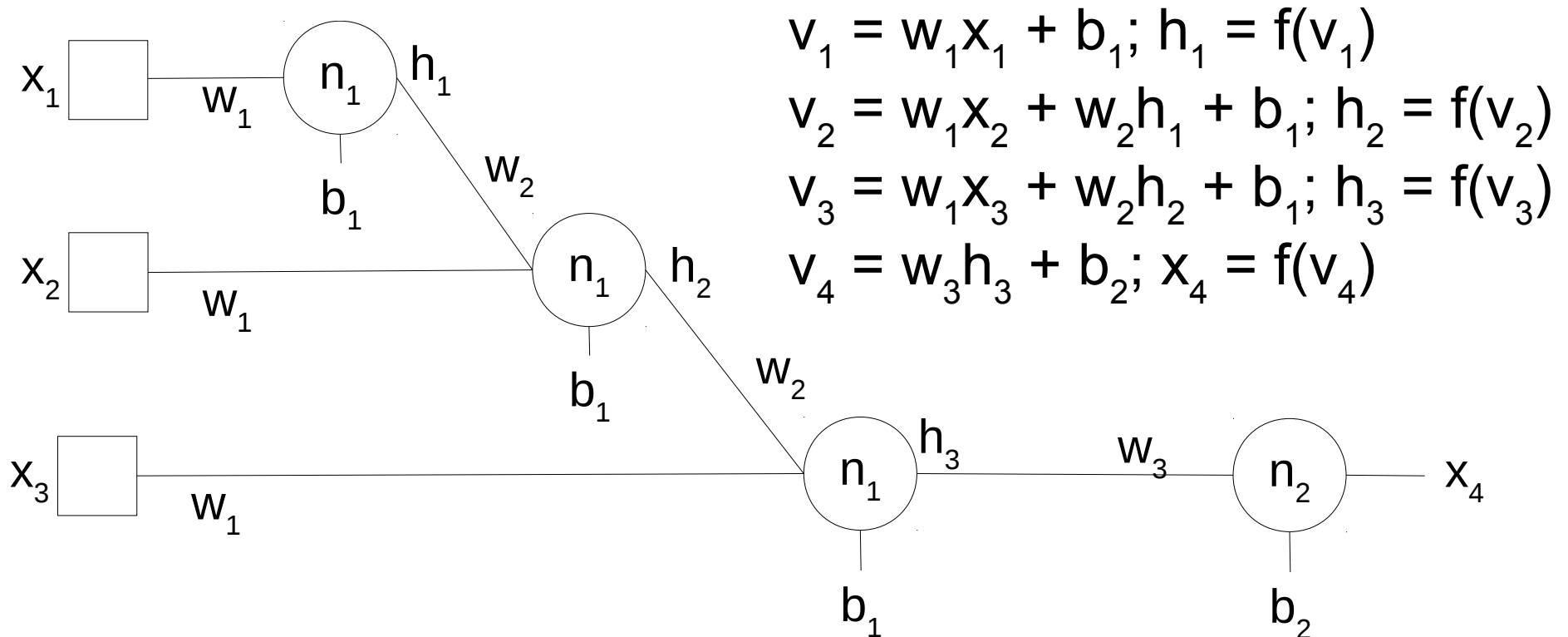


- Representação detalhada:



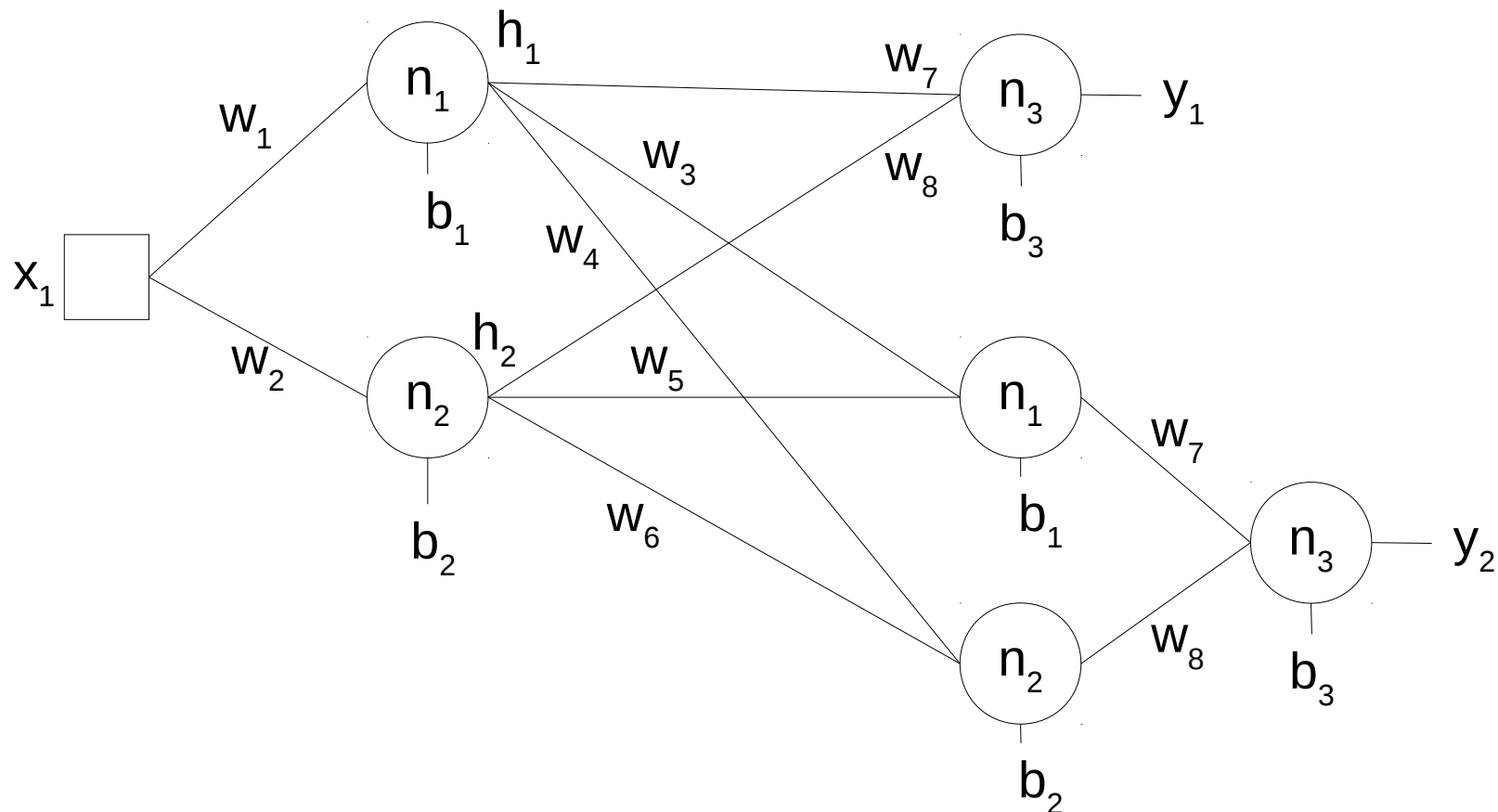
Redes recorrentes (5)

- A rede recorrente pode ser pensada como: feedforward + skip connections + compartilhamento de pesos



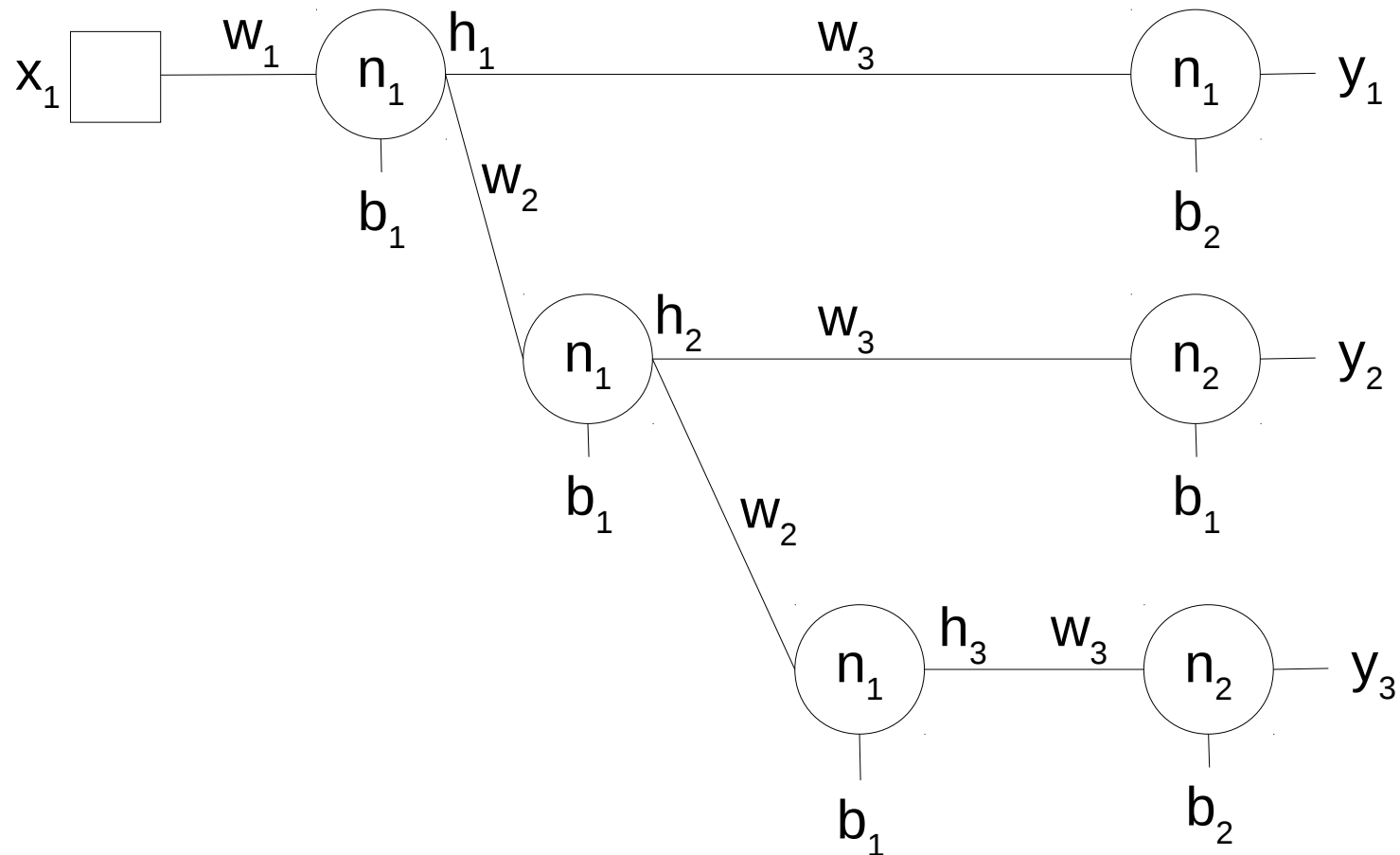
Redes recorrentes (6)

- Avançado: como fica uma rede com dois neurônios ocultos e memória de duas posições?



Redes recorrentes (7)

- Exemplo de uma rede 1-n



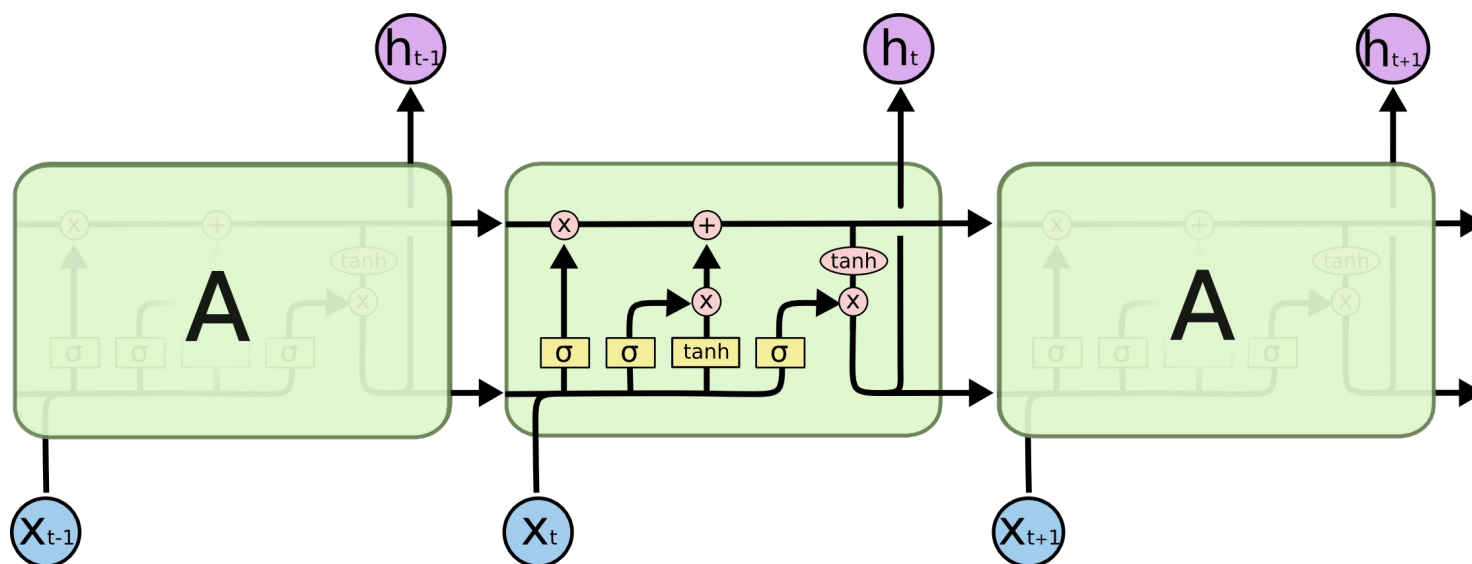
Redes recorrentes ⁽⁸⁾

- Redes reais costumam ter vários neurônios ocultos e podem ter várias entradas e saídas
- Além disso, redes podem ter saída de tamanho variável
 - Nelas, um neurônio de saída especial “fim de sequência” é usado
 - Quando a rede deseja terminar a saída, ativa esse neurônio

Redes recorrentes ⁽⁹⁾

- Redes recorrentes sofrem do problema da dissipação do gradiente. Quanto maior a memória, mais grave o problema
- Redes com portas surgiram para resolver esse problema
 - São mais inspiradas em circuitos digitais do que no cérebro
 - LSTM – Long Short Term Memory (a seguir): uma das mais populares

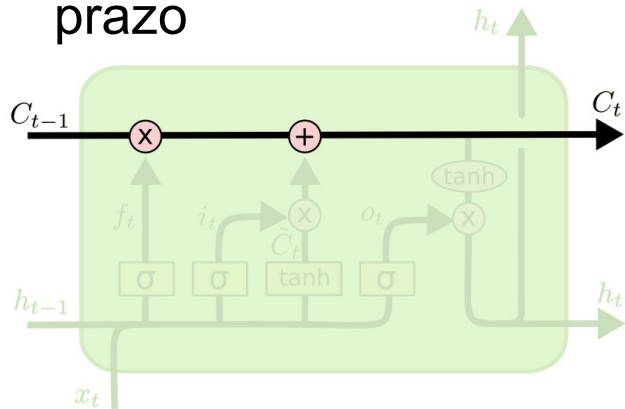
Redes recorrentes ₍₁₀₎



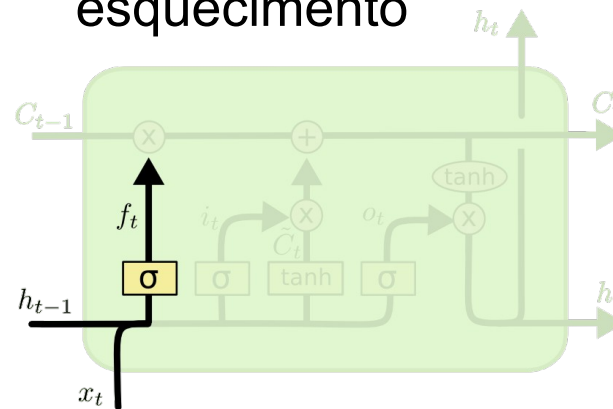
- σ : ativação sigmoide logística
- \tanh : tangente hiperbólica
- $+$: soma
- \times : produto

Redes recorrentes (11)

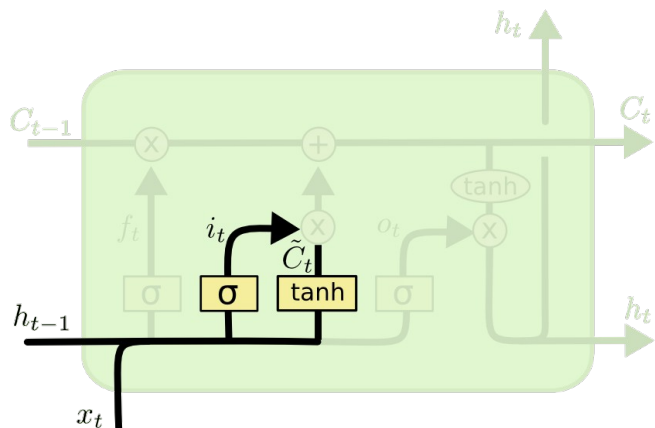
Memória de longo prazo



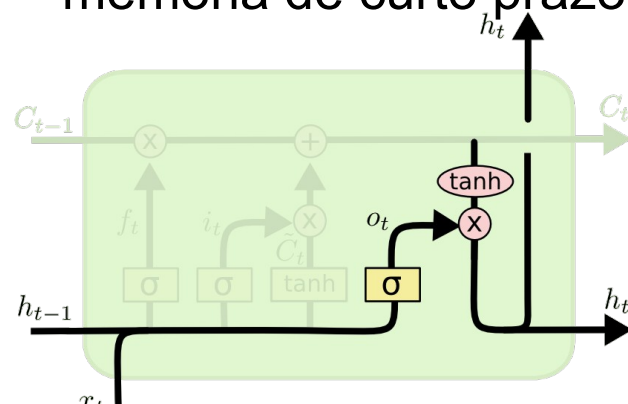
Porta do esquecimento



Porta de entrada



Porta de saída / memória de curto prazo



Redes recorrentes ⁽¹²⁾

- Exemplo com tradução neural para criar intuição:
“Mary is very smart”
 - Entrada “Mary” (embedding): neurônio recorrente memoriza “gênero feminino” e continua a tradução normalmente
 - Entrada “smart”: memória ajuda a escolher tradução adequada
 - “esperta” (correta)
 - “esperto” (incorreto)

Redes recorrentes ⁽¹³⁾

- No exemplo intuitivo:
 - Um valor positivo na memória pode indicar “feminino”
 - Um valor negativo pode indicar “masculino”
 - Um valor próximo a zero pode indicar “gênero neutro” / “desconhecido”

Redes recorrentes ⁽¹⁴⁾

- Porta do esquecimento usa multiplicação: valor 0 (esquecer tudo) ou 1 (não esquecer nada) ou intermediário (esquecer parcialmente)
- Porta da entrada usa soma: 0 é para não armazenar nada; outros valores permitem armazenar informações.
- Porta de saída: repassa a análise para a próxima camada