INF721

2023/2

UFV

Aprendizado em Redes Neurais Profundas

A31: Conclusão

Plano de Aula

- Retrospectiva INF721
- O que não conseguimos cobrir
- Como continuar aprendendo
- Como se manter atualizado
- Avaliação da disciplina

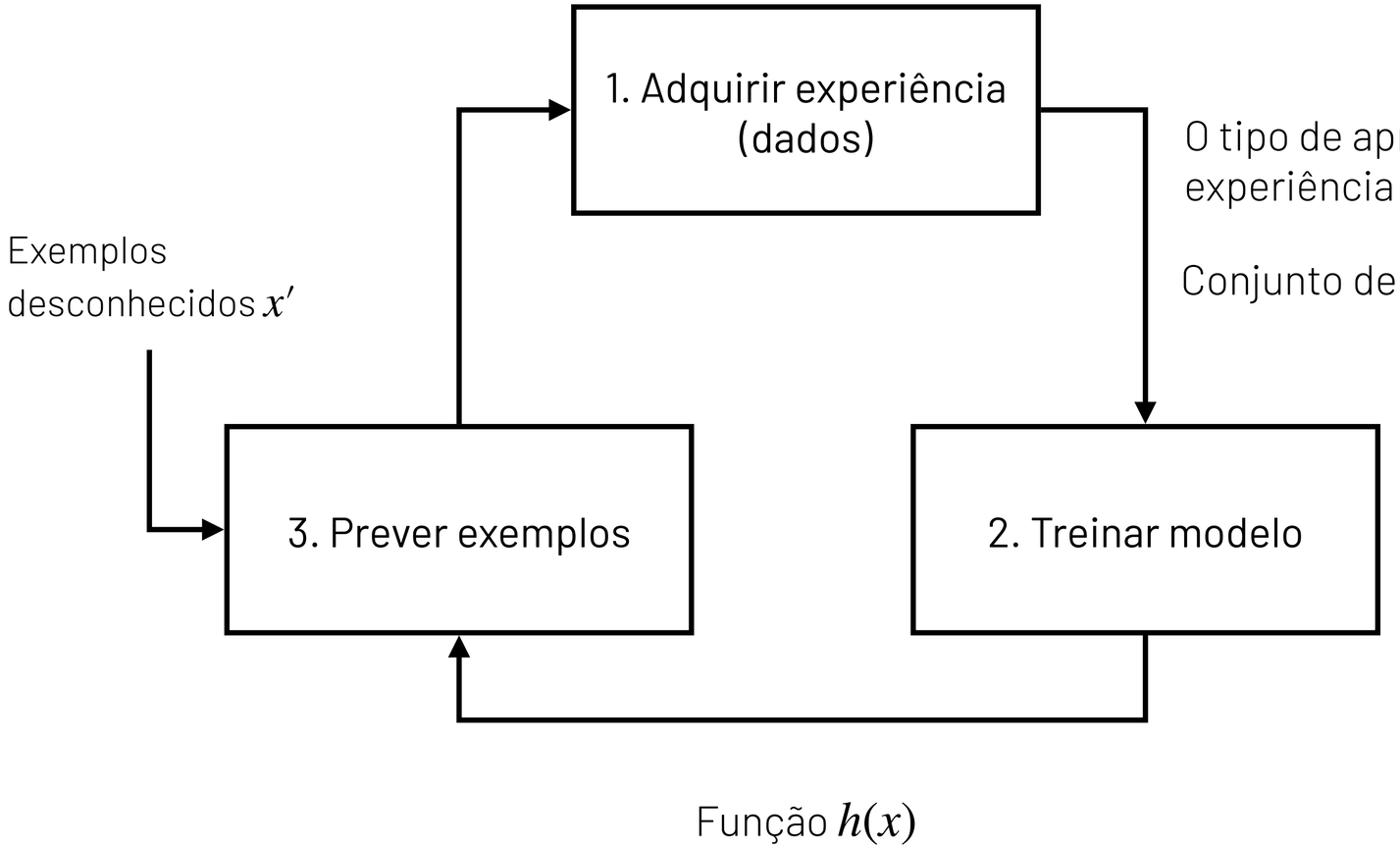
Algoritmos são tradicionalmente implementados como funções

$y = f(x)$

O objetivo de **aprendizado de máquina** é encontrar uma função, a partir de dados, para resolver um dado problema computacional

Aprendizado de Máquina

Aprender uma função $h(x)$ a partir de um conjunto de dados D para prever o rótulo de exemplos desconhecidos.



O tipo de aprendizado é definido pelo tipo de experiência (i.e., dados) observada pelo agente.

Conjunto de dados D

Rotulados

- **Aprendizado Supervisionado**

Não-rotulados

- **Aprendizado Não-supervisionado**

Valores de recompensa do ambiente

- **Aprendizado por Reforço**

Regressão Logística

Entrada

Um exemplo $x \in \mathbb{R}^d$

Saída

A probabilidade de x ser da classe $y = 1$

$\hat{y} = P(y = 1 \mid x),\ 0 \leq \hat{y} \leq 1$

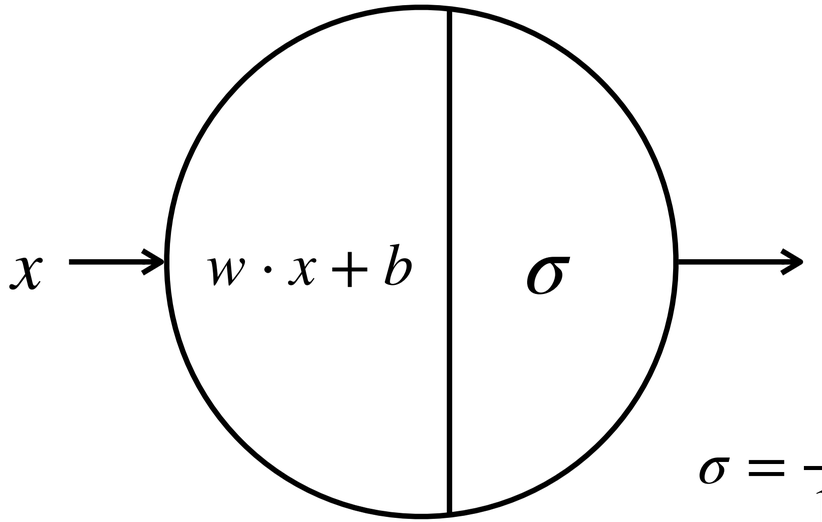
Hipótese

$h(x) = \sigma(w \cdot x + b)$

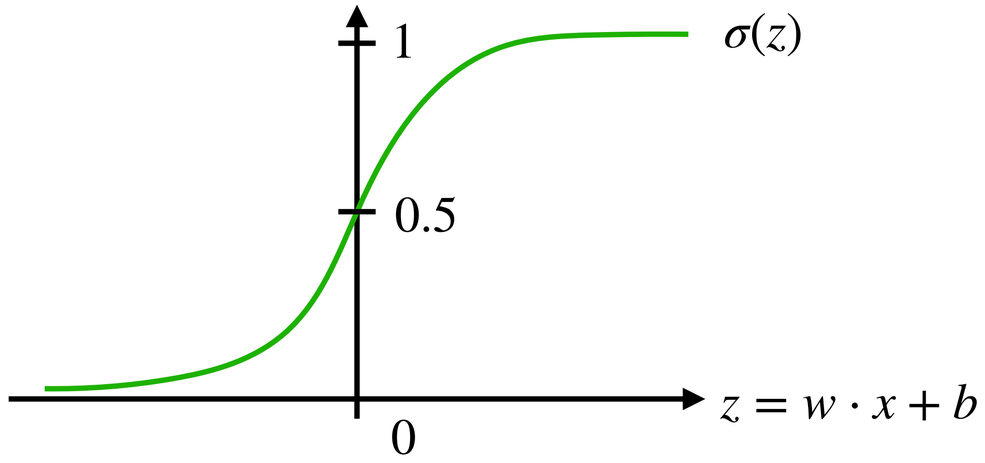
onde:

$w \in \mathbb{R}^d$ e $b \in \mathbb{R}$ são pesos

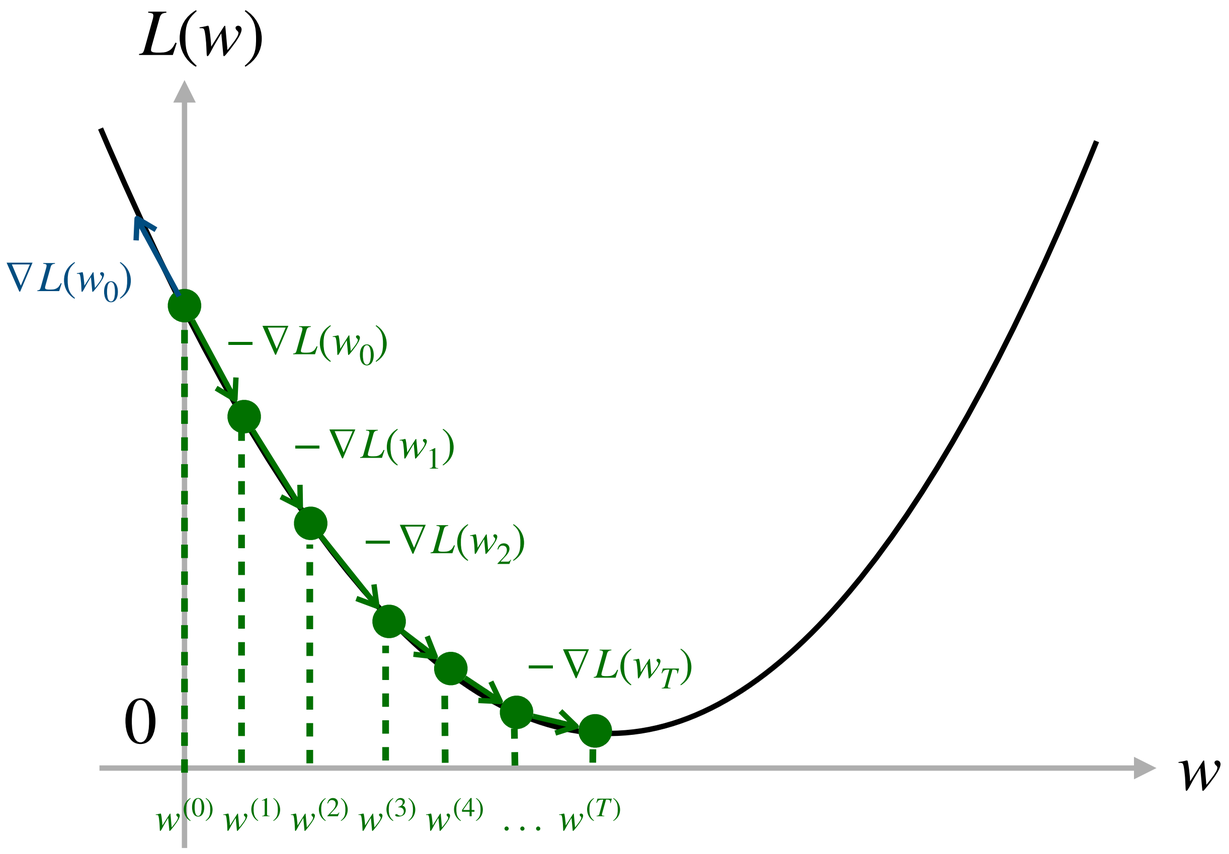
σ é a função logística (sigmoide)



$$\sigma = \frac{1}{1 + e^{-z}}$$



Gradiente Descendente



Dado um valor inicial w, atualizamos iterativamente o valor de w na direção de descida mais íngrime de L a partir do ponto $(w, L(w))$.

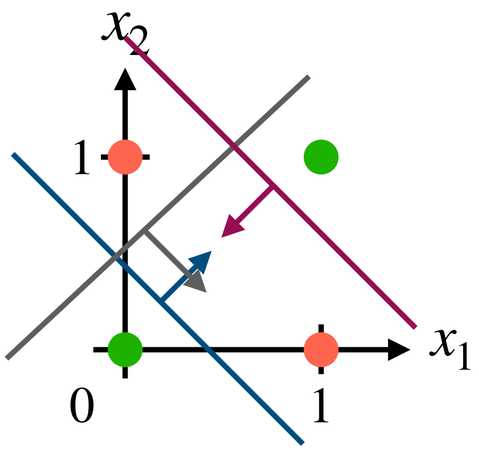
$$w_t \leftarrow w_{t-1} - \alpha \nabla L(w_{t-1})$$

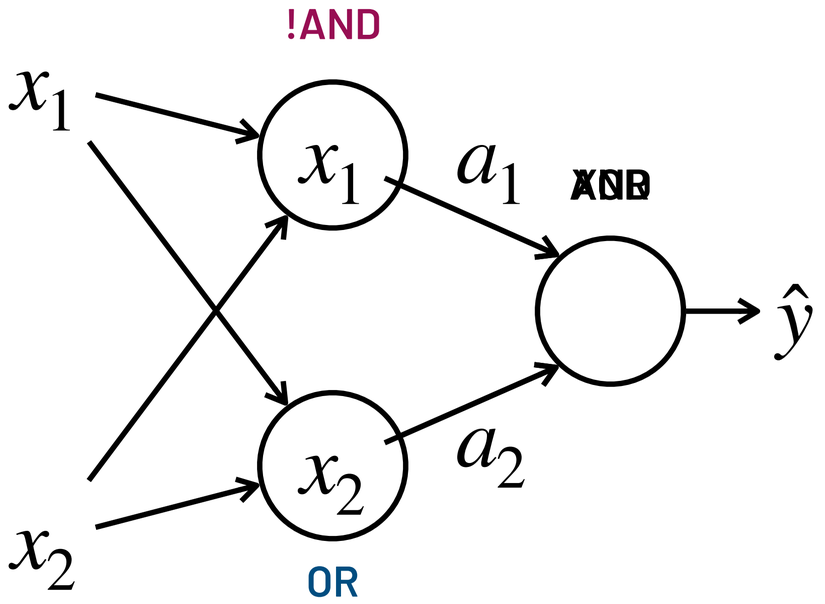
onde α é um híper-parâmetro chamado de **taxa de aprendizado** (*learning rate*), responsável por controlar o comprimento do vetor gradiente.

Multilayer Perceptron

$$f(x_1, x_2) = x_1 \text{ XOR } x_2$$

x_2 \ x_1	0	1
0	0	1
1	1	0





RNAs aprendem representações intermediárias $\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$ dos dados de entrada $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, chamadas **representações latentes**, que podem tornar um problema não-linearmente separável em linearmente separável!

Retropropagação (Backprop)

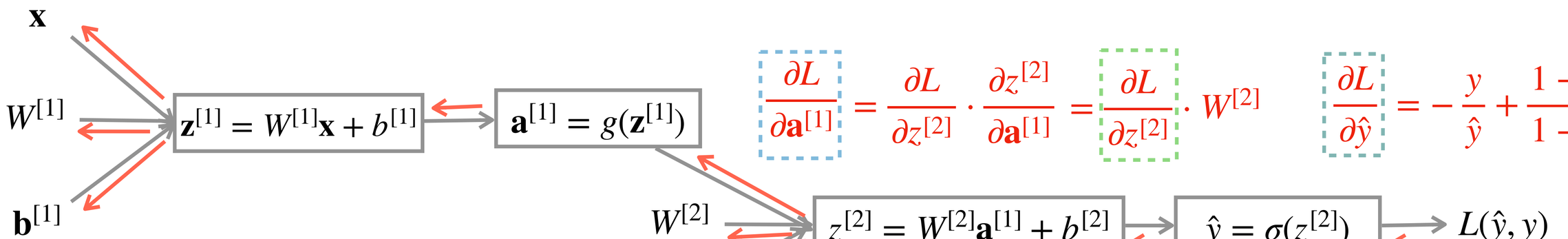
Calcular as derivadas parciais da função de perda com relação aos pesos $W^{[l]}$ e $\mathbf{b}^{[l]}$ para todas as camadas l de trás pra frente com a regra da cadeia.

MLP (2 camadas)

$$\mathbf{z}^{[1]} = W^{[1]}\mathbf{x} + \mathbf{b}^{[1]} \qquad z^{[2]} = W^{[2]}\mathbf{a}^{[1]} + b^{[2]}$$

$$\mathbf{a}^{[1]} = g^{[1]}(\mathbf{z}^{[1]}) \qquad \hat{y} = \sigma(z^{[2]})$$

$$L(\hat{y}, y) = -\,y\ log\ \hat{y} + (1-y)\ log\ (1-\hat{y}))$$



$$\frac{\partial L}{\partial \mathbf{a}^{[1]}} = \frac{\partial L}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial \mathbf{a}^{[1]}} = \frac{\partial L}{\partial z^{[2]}} \cdot W^{[2]} \qquad \frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$$

$$\frac{\partial L}{\partial \mathbf{z}^{[1]}} = \frac{\partial L}{\partial \mathbf{a}^{[1]}} \cdot \frac{\partial \mathbf{a}^{[1]}}{\partial \mathbf{z}^{[1]}} = \frac{\partial L}{\partial z^{[2]}} \cdot W^{[2]} \times \frac{dg^{[1]}}{d\mathbf{z}^{[1]}}$$

$$\frac{\partial L}{\partial z^{[2]}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z^{[2]}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}} \cdot \hat{y}(1-\hat{y}) = \hat{y} - y$$

$$\frac{\partial L}{\partial W^{[1]}} = \frac{\partial L}{\partial \mathbf{z}^{[1]}} \cdot \mathbf{x}^T$$

$$\frac{\partial L}{\partial W^{[1]}} = \frac{\partial L}{\partial \mathbf{z}^{[1]}}$$

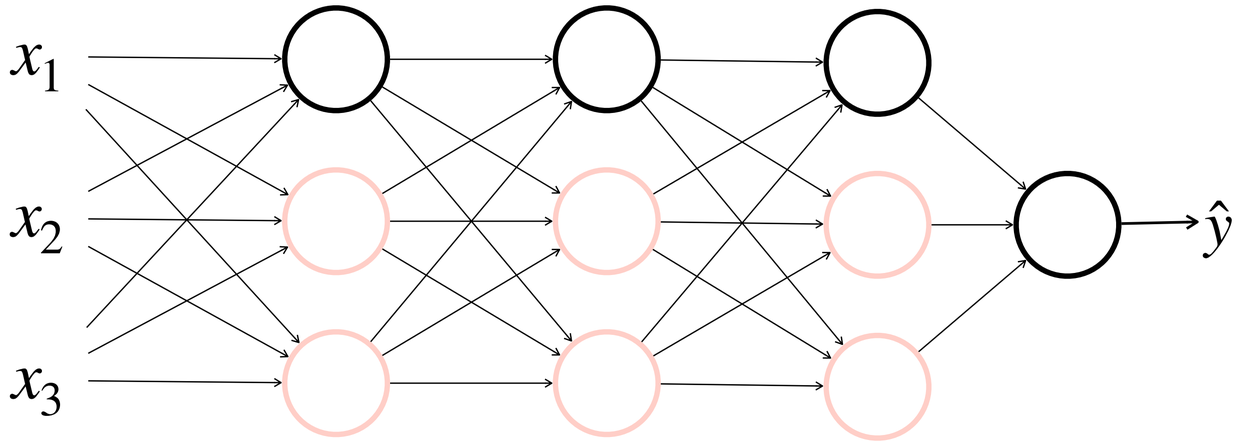
$$\frac{\partial L}{\partial W^{[2]}} = \frac{\partial L}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial W^{[2]}} = \frac{\partial L}{\partial z^{[2]}} \cdot a^{[1]^T}$$

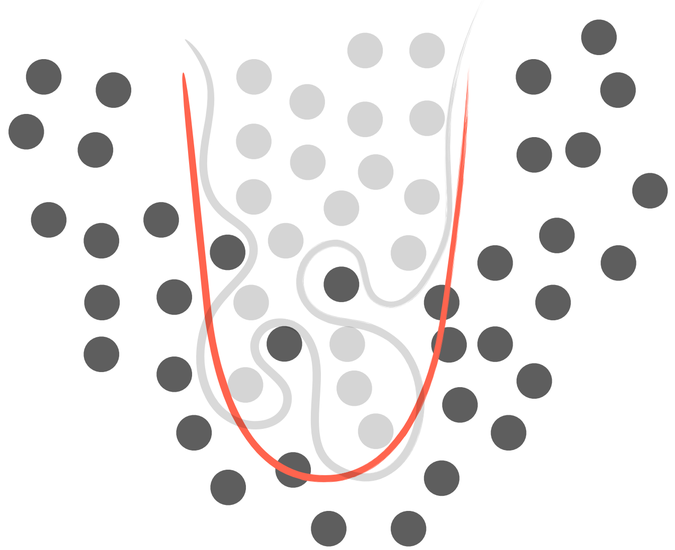
$$\frac{\partial L}{\partial b^{[2]}} = \frac{\partial L}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial b^{[2]}} = \frac{\partial L}{\partial z^{[2]}}$$

UFV

Regularização

$$L(h) = -\frac{1}{n}\sum_{i=1}^{n} L(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{2n}\|W\|_2^2 \longrightarrow W^{[l]} \approx 0$$





Ao reduzir os pesos de alguns neurônios, a regularização simplifica a hipótese de uma RNA em tempo de treinamento, tornando a fronteira de decisão mais simples também.

Adaptive Moment Estimation (Adam)

O Adam combina o RMSProp e momento

$$dw, db = backward(X^t)$$

$$Vdw = \beta_1 \cdot Vdw + (1 - \beta_1)dw, \quad Vdb = \beta_1 \cdot Vdb + (1 - \beta_1)db$$

Momento

$$Sdw = \beta_2 \cdot Sdw + (1 - \beta_2)dw^2, \quad Sdb = \beta_2 \cdot Sdb + (1 - \beta_2)db^2$$

RMSProp

$$Vdw = \frac{Vdw}{1 - \beta_1^t}, \quad Vdb = \frac{Vdb}{1 - \beta_1^t}$$

$$Sdw = \frac{Sdw}{1 - \beta_2^t}, \quad Sdb = \frac{Sdb}{1 - \beta_2^t}$$

$$w = w - \alpha \frac{Vdw}{\sqrt{Sdw}}$$

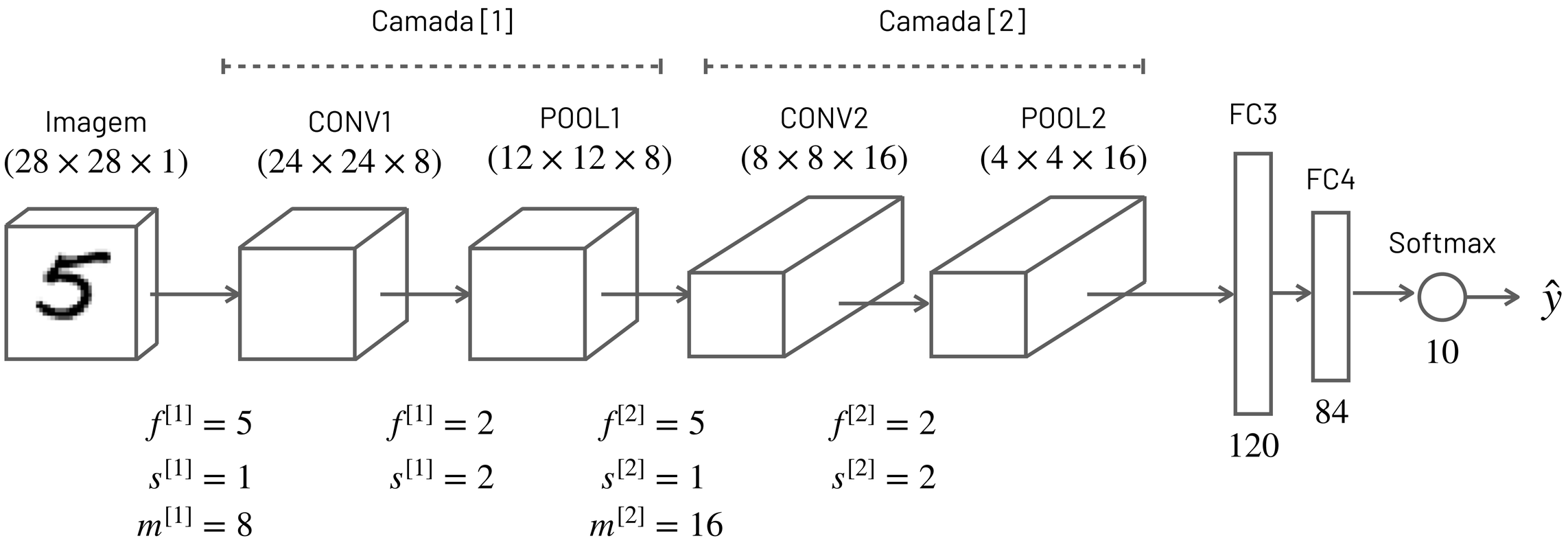
$$b = b - \alpha \frac{Vdb}{\sqrt{Sdb}}$$

Recomendações de valores para os híper-parâmetros:

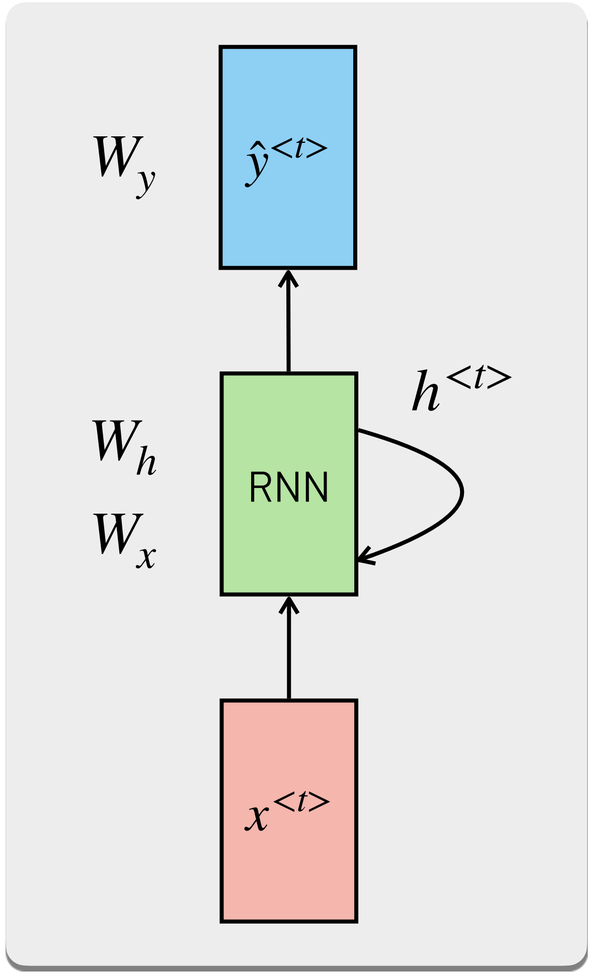
$$\beta_1 = 0.9$$

$$\beta_2 = 0.999$$

Redes Neurais Convolucionais (CNNs)



Redes Neurais Recorrentes (RNN)

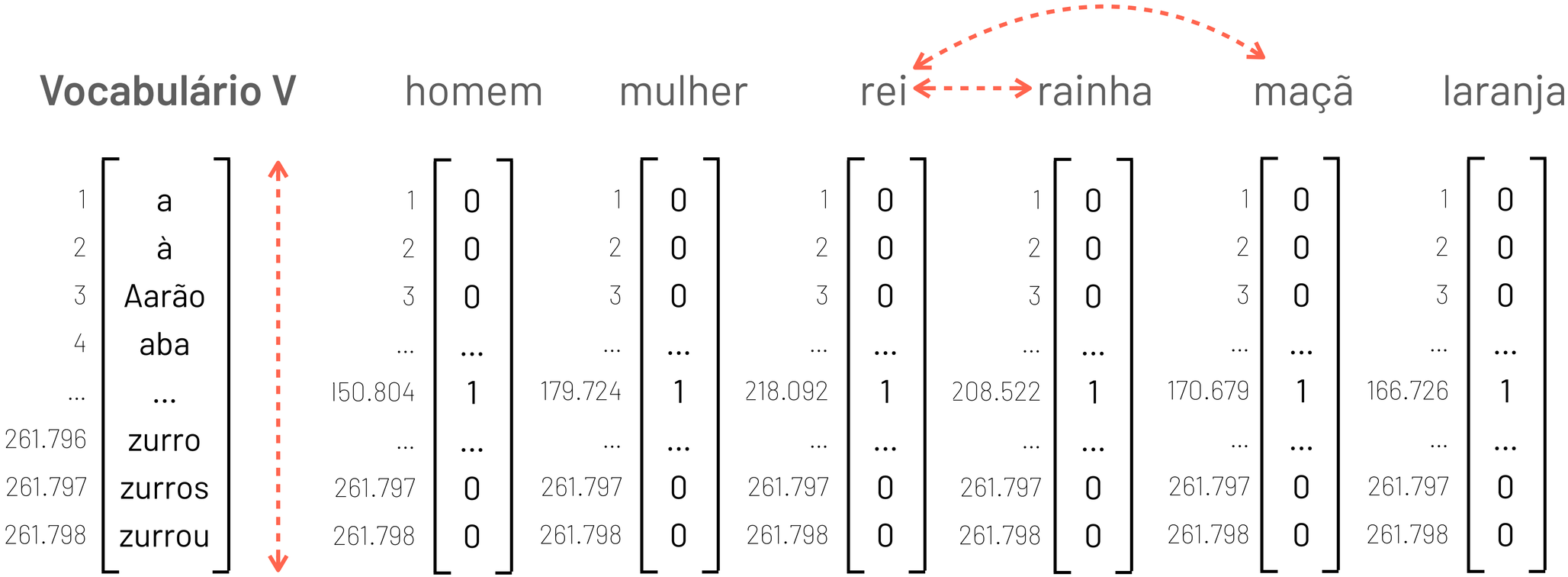


A RNN processa cada elemento da entrada $x^{<t>}$ de uma vez, mantendo um estado (vetor) $h^{<t>}$ que é atualizado a cada intervalo de tempo para gerar uma saída $y^{<t>}$

$$h^{<t>} = g_1(W_h h^{<t-1>} + W_x x^{<t>} + b_h)$$
$$\hat{y}^{<t>} = g_2(W_y h^{<t>} + b_y)$$

- g_1: função de ativação da camada escondida (tanh/relu)
- g_2: função de ativação da camada de saída (sigmoid/softmax)

Representação de Palavras (Word Embeddings)



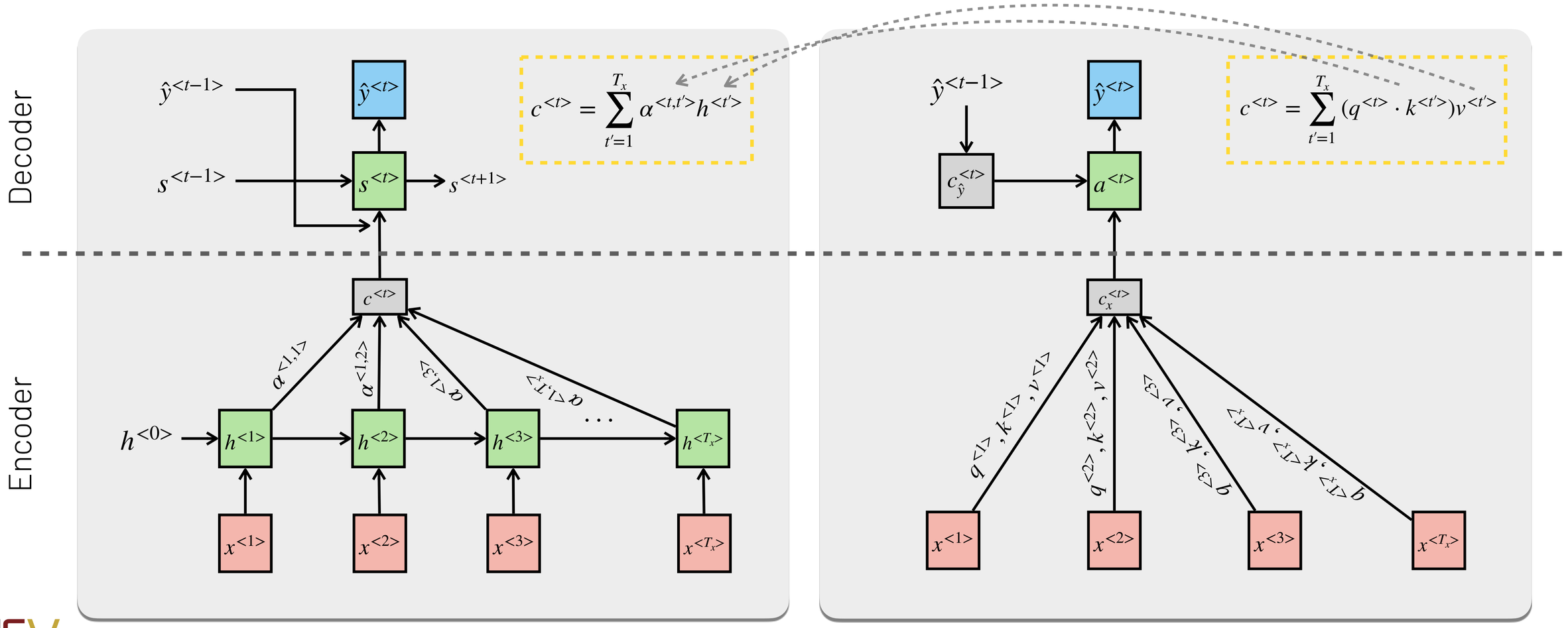
|V| = 261.798

1. O número de características das palavras é definido pelo tamanho do vocabulário |V|
2. A distância entre quaisquer duas palavras é a mesma.

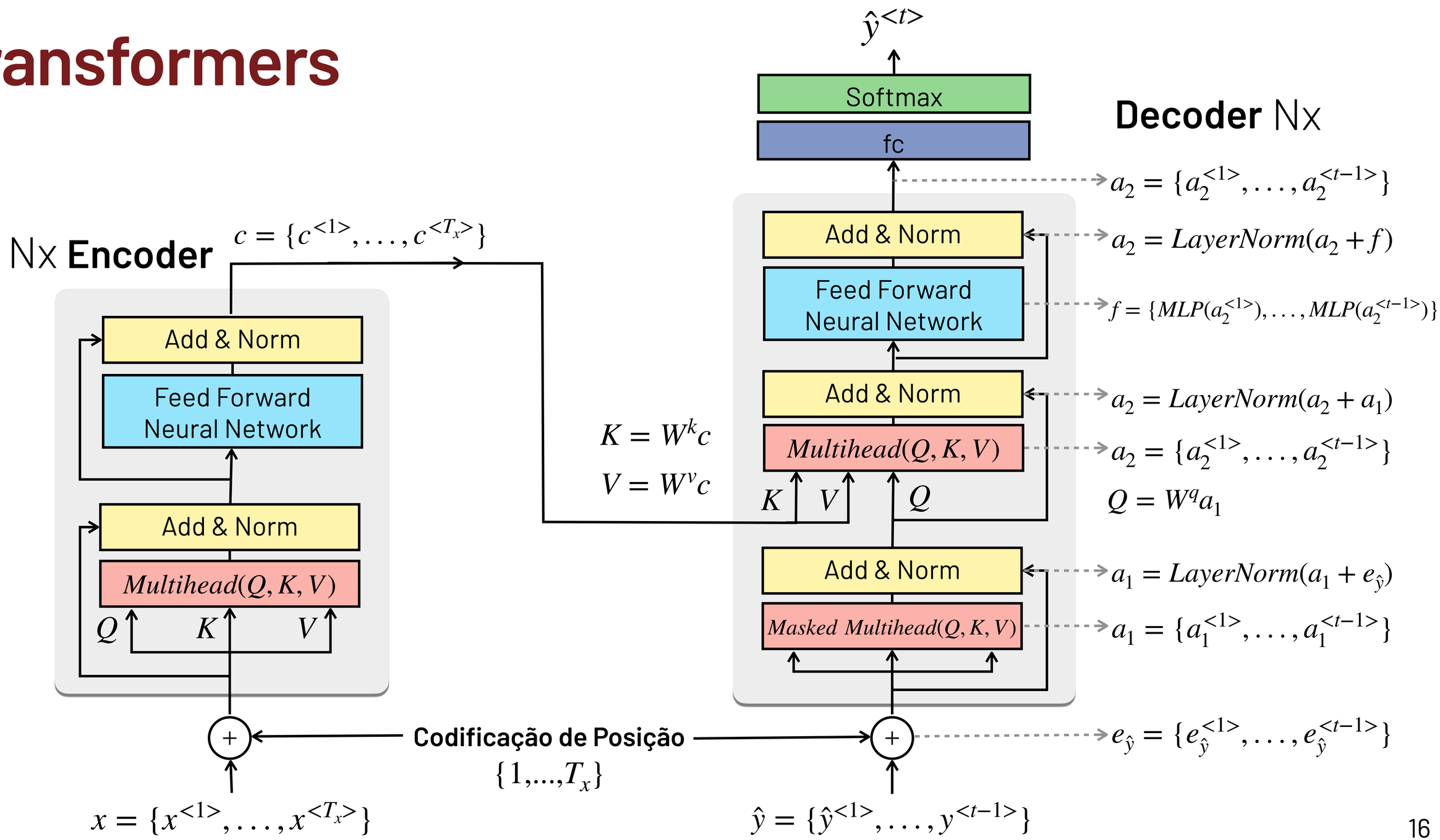
Mecanismos de Atenção

RNNs Atenção Aditiva

Transformers Auto-Atenção (Self Attention)



Transformers



O que não conseguimos cobrir...

- Ajuste de Híperparâmetros
- Normalização
  - Batch e Layer
- Modelos Generativos
  - Generative Adversarial Networks
  - Variational Autoencoders
  - Diffusion Models

Como continuar aprendendo no DPI-UFV

- INF692 - Redes Convolucionais
- INF791 - Tópicos Especiais II - Processamento de Linguagem Natural
- INF623 - Inteligência Artificial
- INF723 - Visualização de Dados
- INF493 - Tópicos Especiais III - Ciência de Dados

Como continuar aprendendo online

- Berkeley CS188 - Intro to Artificial Inteligence
- Cornell CS4780 - Machine Learning for Intelligent Systems
- Stanford CS231 - Deep Learning for Computer Vision
- Stanford CS224 - Natural Language Processing with Deep Learning
- Berkeley CS285 - Deep Reinforcement Learning
- Andrew Ng's Deep Learning Specialization

Como se manter atualizado

- Seguir os “Heroes of Deep Learning” no Twitter (X)
  - Geoff Hinton: @geoffreyhinton
  - Andrew Ng: @AndrewYNg
  - Yann LeCunn: @ylecun
  - Andrej Karpathy: @karpathy
  - Procure identificar os principais pesquisadores da sua área de interesse
- Canais com atualização de papers:
  - “AK" @_akhaliq no Twitter
  - Two Minutes Paper no Youtube

Suporte Técnico

Daqui pra frente, fique à vontade para me perguntar qualquer coisa sobre Deep Learning...

Basta enviar um e-mail para lucas.n.ferreira@ufv.br

Tentarei responder até o próximo fim de semana

Válido para vida toda 😊

Um último pedido 🙏

Reponder o questionário de avaliação da disciplina:

https://forms.gle/1HWjzPz3orLJrniF9

Fim...

Obrigado pelo interesse e participação!

Muito sucesso na carreira de vocês!

Lucas N. Ferreira

lucas.n.ferreira@ufv.br