

Redes Neurais para Processamento Temporal

Germano Vasconcelos
Centro de Informática
Universidade Federal de Pernambuco

Aprendizagem de Padrões Temporais – *Time Sequences*



- Muitas tarefas requerem aprendizagem de sequência de eventos

Reconhecimento de Sequências: produzir saída específica quando uma sequência de entrada específica é apresentada.

Exemplo: reconhecimento de voz

Reprodução de sequências: gerar o restante de uma sequência quando a rede recebe parte da sequência.

Exemplo: previsão de séries temporais (ações, índices, previsão do tempo e fluxos)

Associação temporal: produzir sequência específica de saída em resposta a uma sequência específica de entrada.

Exemplo: geração de fala



Redes Recorrentes



- ◆ Possuem 1 (um) ou mais loops de feedback que permitem a representação de informação temporal
- ◆ Representações que criam memórias de curto prazo



Motivação para Redes Recorrentes



➤ Redes Feedforward networks

- Informação flue apenas no sentido para frente
- Um padrão de entrada produz Uma saída
- Sem noção de tempo ou memória de estados anteriores

➤ Recorrência

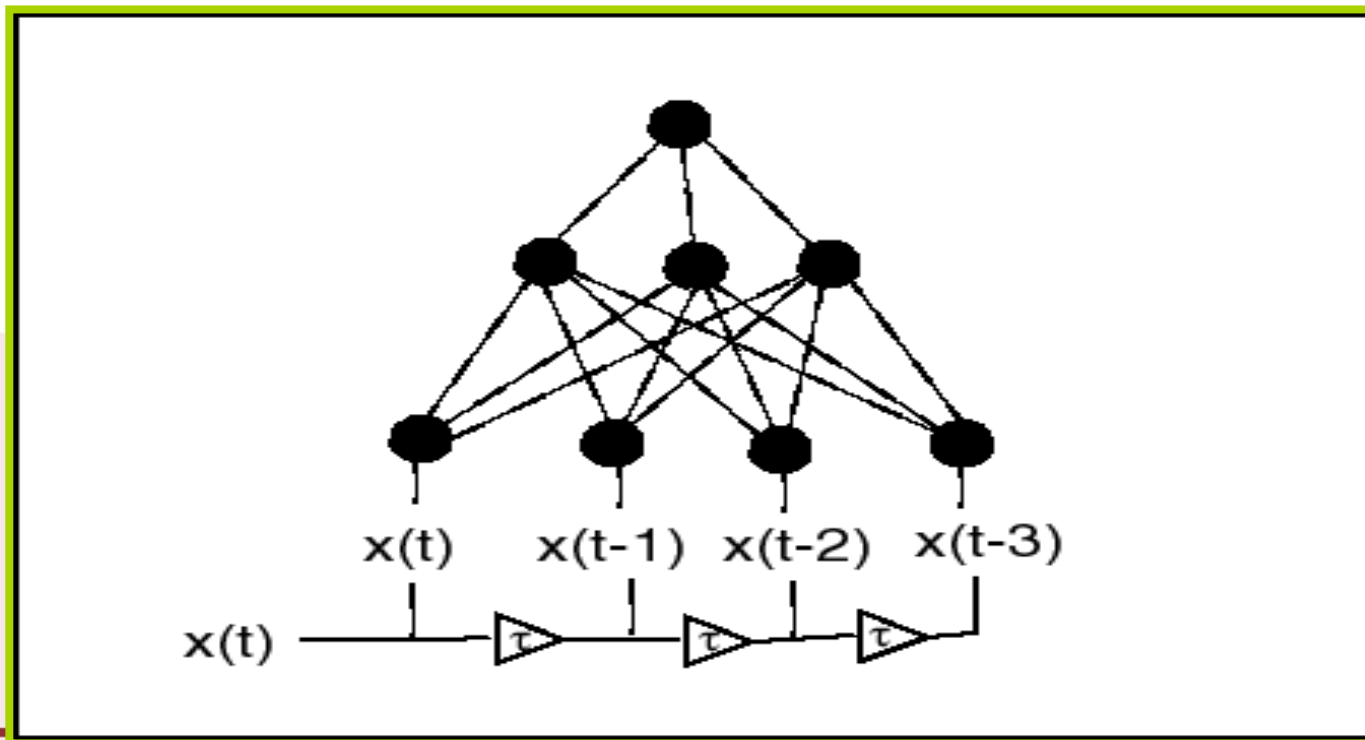
- Nós se conectam para trás com outros nós e consigo mesmos
- Fluxo de informação é multi-direcional
- Sentido de tempo e memória de estados anteriores

- Sistemas biológicos possuem recorrência (estruturas feedforward também existem)



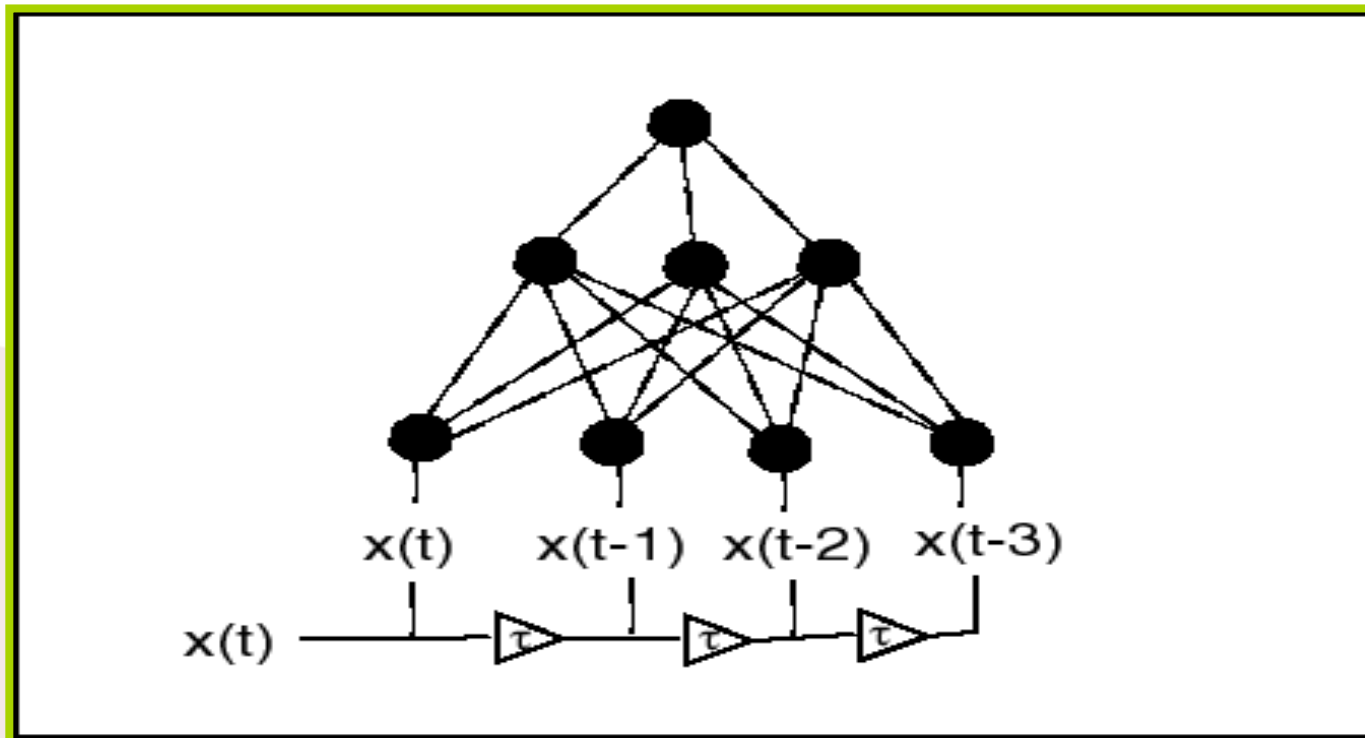
Como Representar Tempo em Redes Feedforward?

- ◆ Apresentando a informação temporal relevante ao mesmo tempo!
- ◆ Permite uso de backpropagation convencional (e outros)



Como Representar Tempo em Redes Feedforward?

- Potencial desvantagem: memória limitada ao tamanho da “linha de retardo” (número de retardos)
- Se um número grande de entradas for necessário pode tornar treinamento lento e necessidade de muitos exemplos



Previsão de Séries Temporais



- Séries Temporais = sequência de valores medidos em momentos sucessivos no tempo

- Exemplos:
 - Evolução de variações cambiais
 - Evolução do preço de ações
 - Monitoramento de sinais biológicos
 - Previsão meteorológica
 - Previsão de demanda de produtos e serviços (*bike*)

- Objetivo: prever o valor futuro em um ou vários passos

Exemplo: Previsão (*Forecasting*)

- Dado um conjunto de n valores de uma variável ($y(t_1), y(t_2), \dots, y(t_n)$) em uma sequência de tempo $t_1, t_2, \dots, t_n$,
- Prever o valor $y(t_{n+1})$ num futuro t_{n+1}

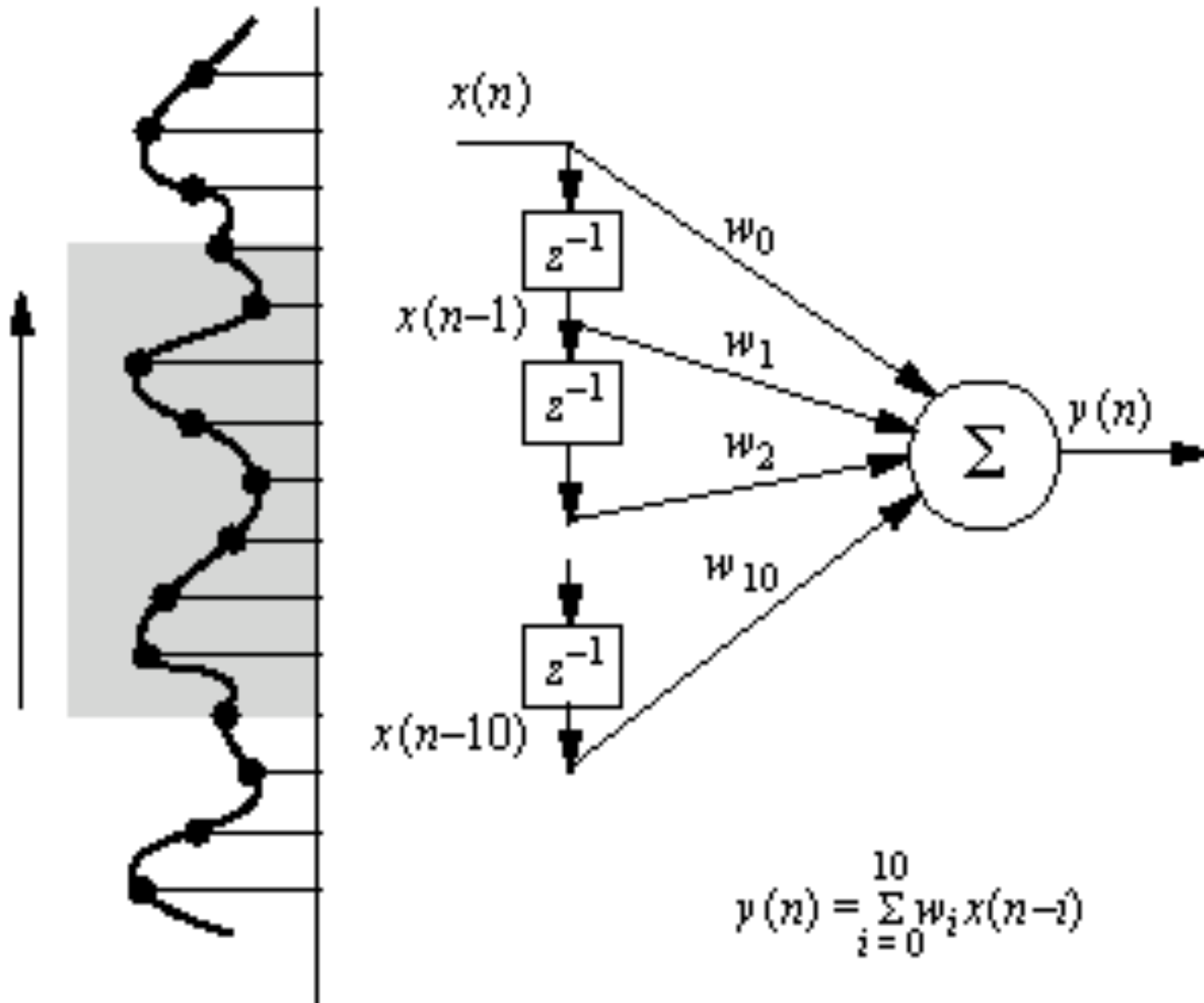
Previsão – Questões Relevantes



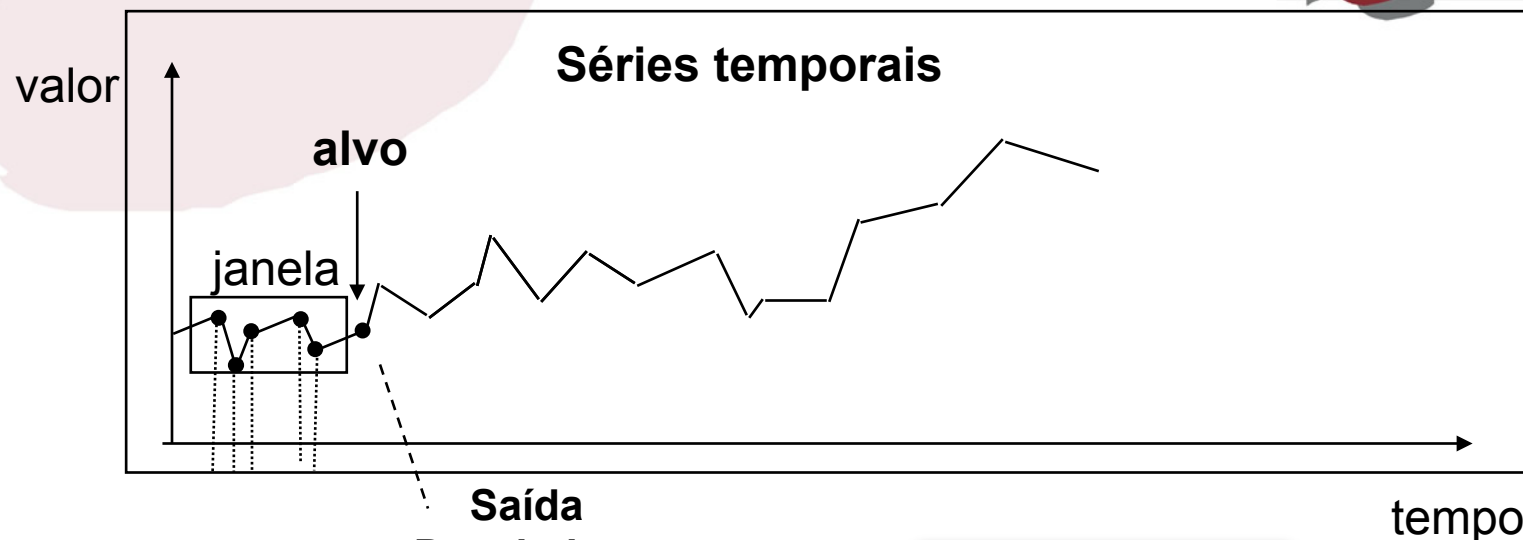
- Definição da janela de entrada
- Definição do horizonte de previsão
- Definição de outras variáveis explicativas



Previsão com uma Rede MLP



Previsão de Séries Temporais



Entradas
da rede =
 n valores
passados

Saída
Desejada =
valor da série
 k passos à
frente

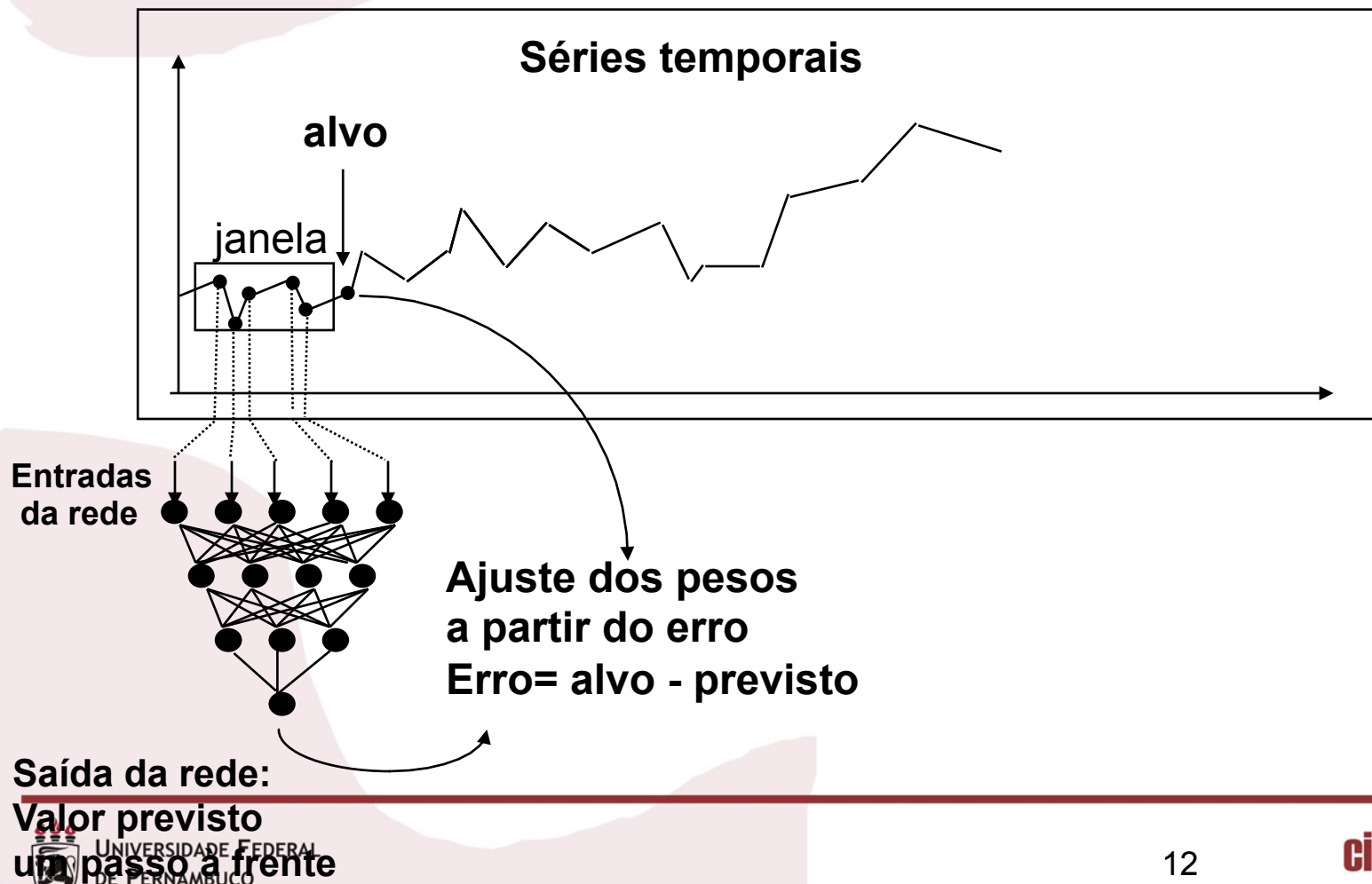
**Definição da
janela de entrada**

**Definição da
janela de saída**

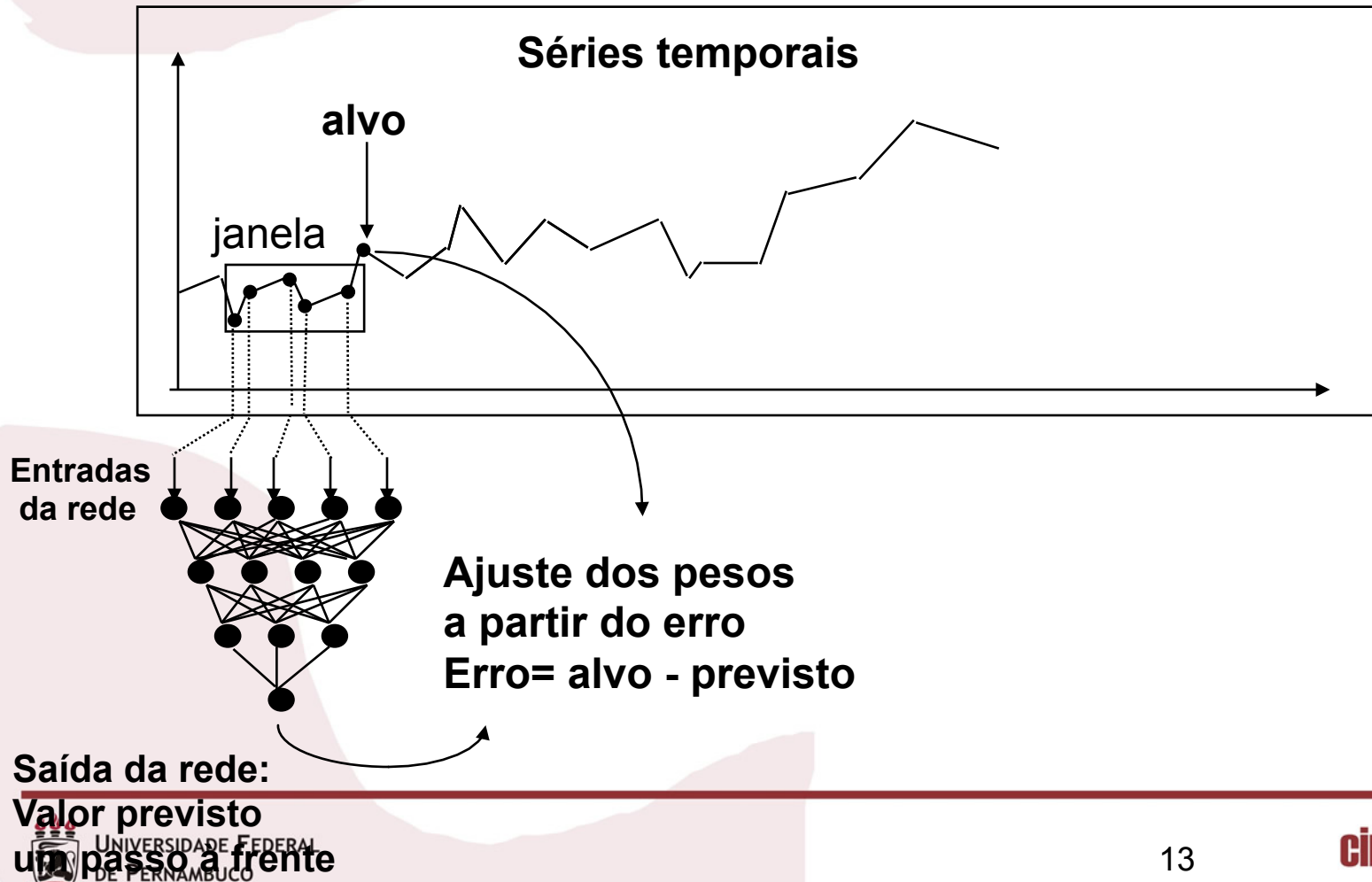
**Ex: 5 valores
passados**

**Ex: valor um
passo à frente**

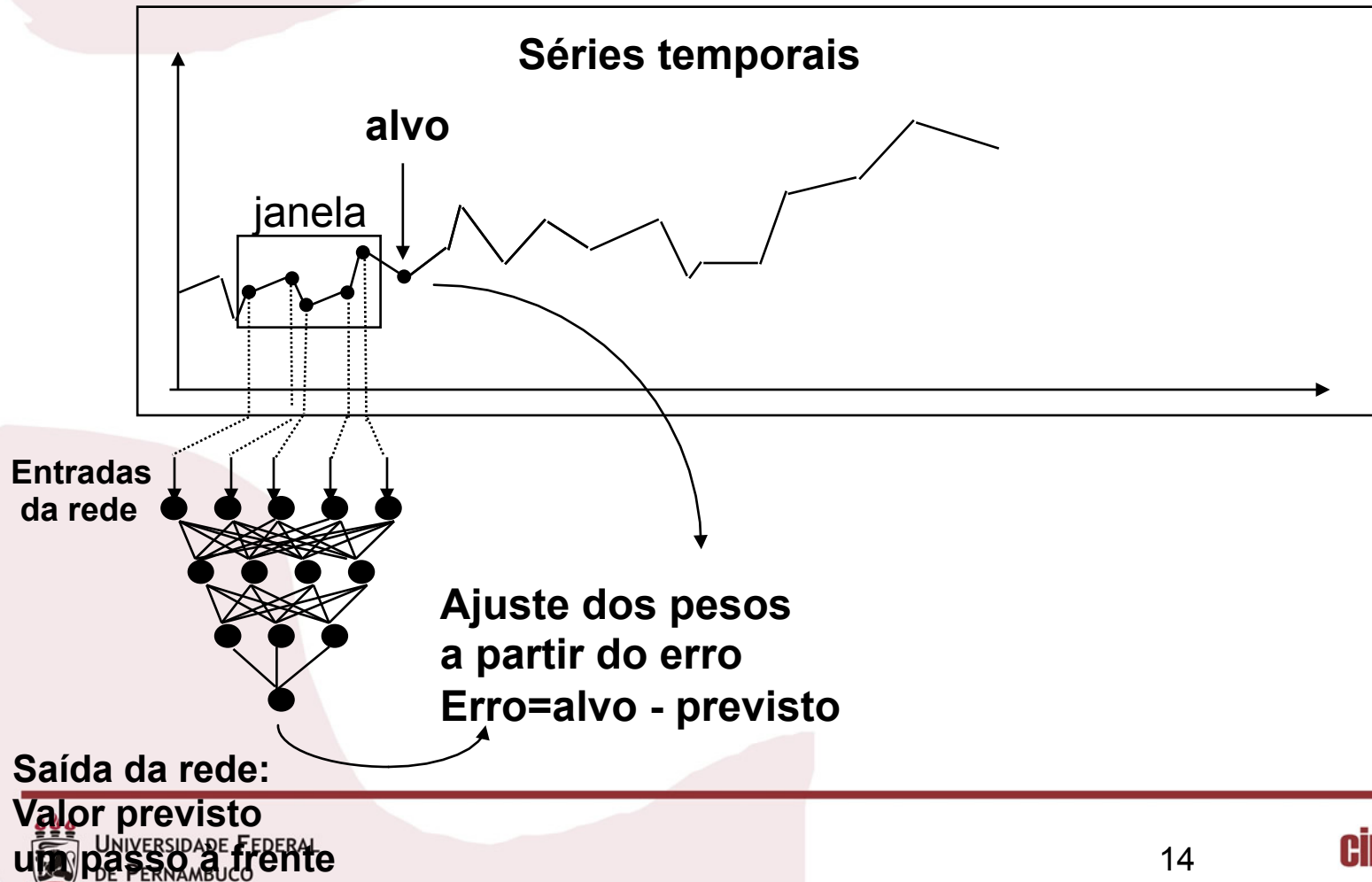
Exemplo: previsão utilizando apenas a série histórica como entrada



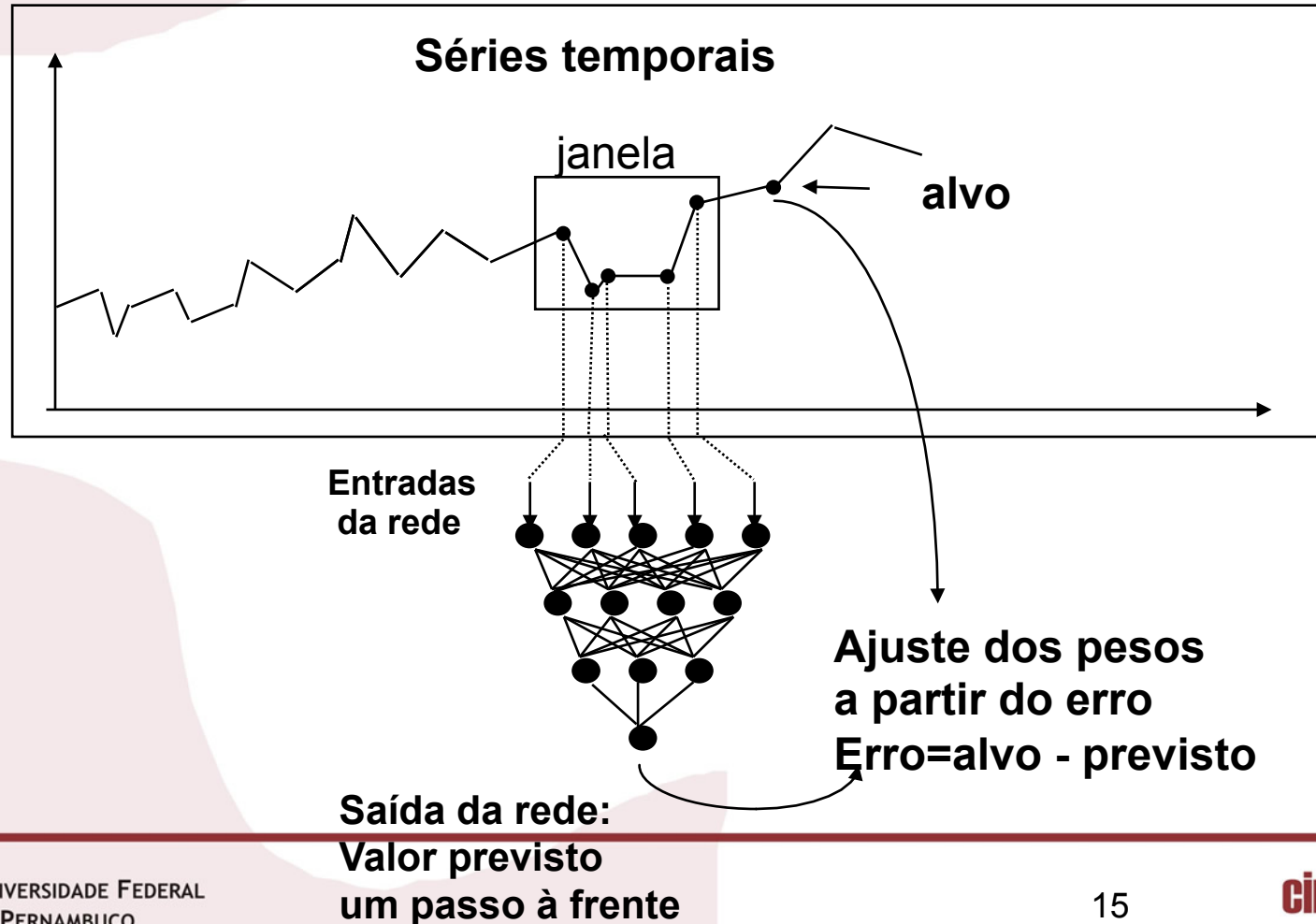
Exemplo: previsão utilizando apenas a série histórica como entrada



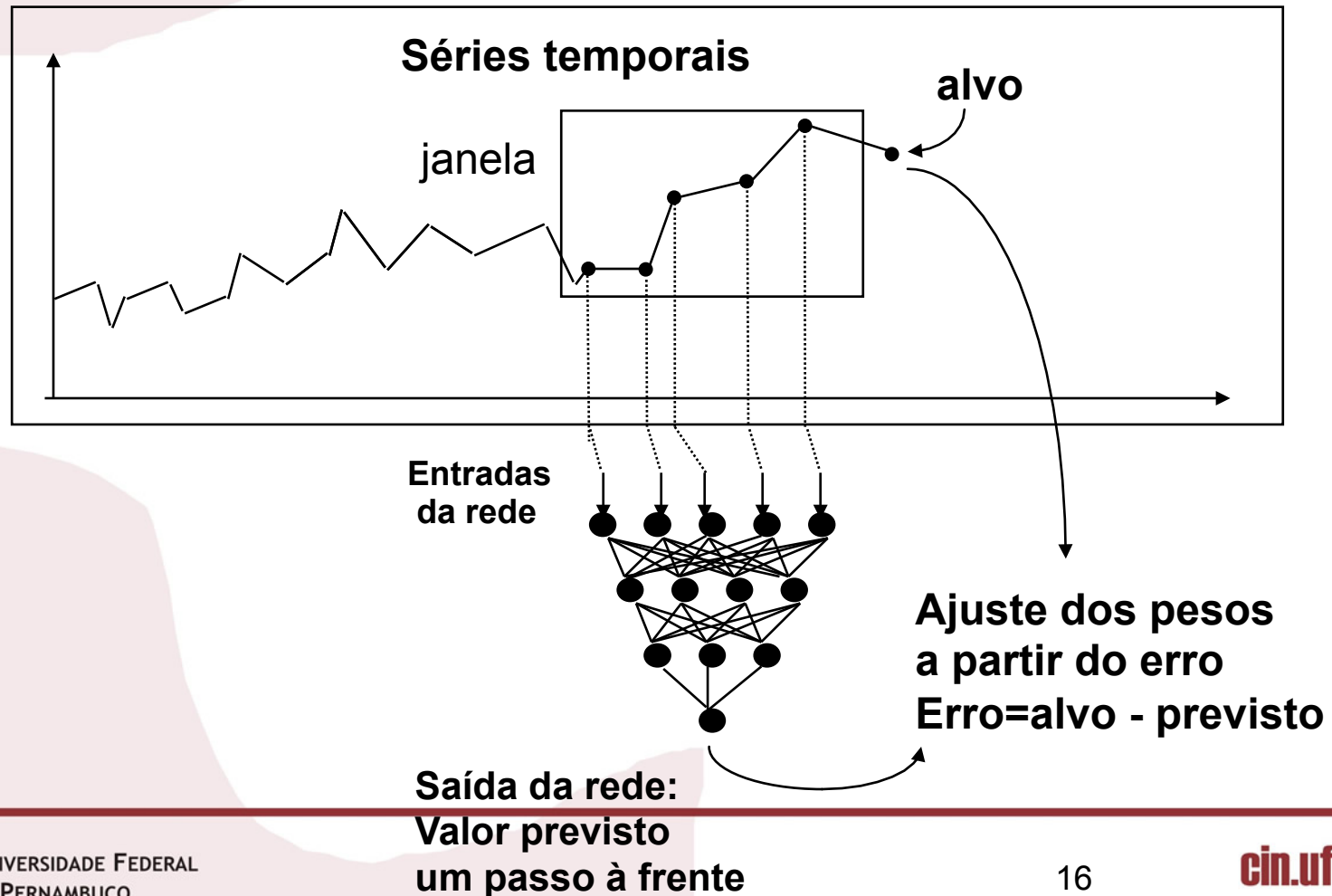
Exemplo: previsão utilizando apenas a série histórica como entrada



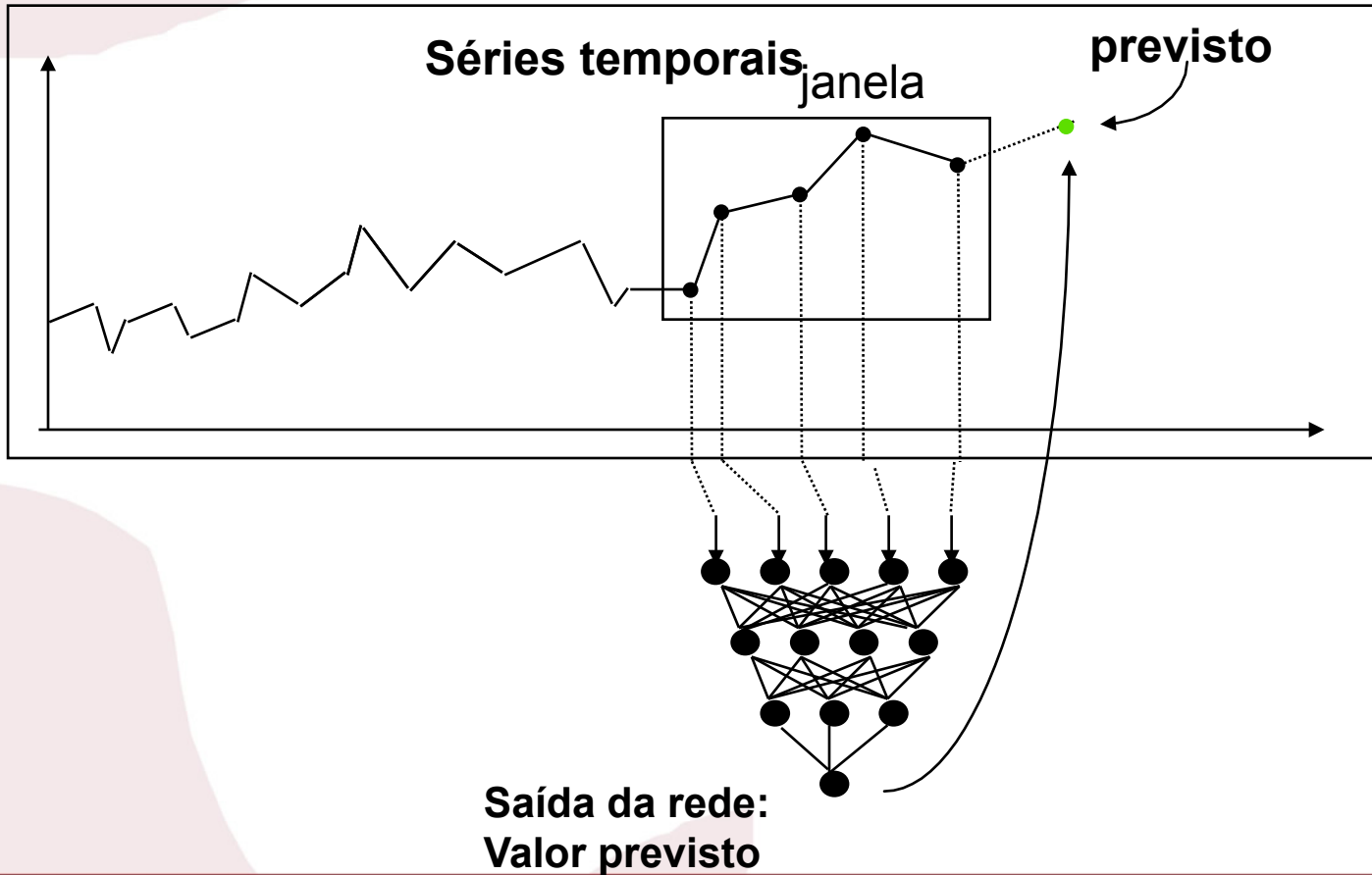
Exemplo: previsão utilizando apenas a série histórica como entrada



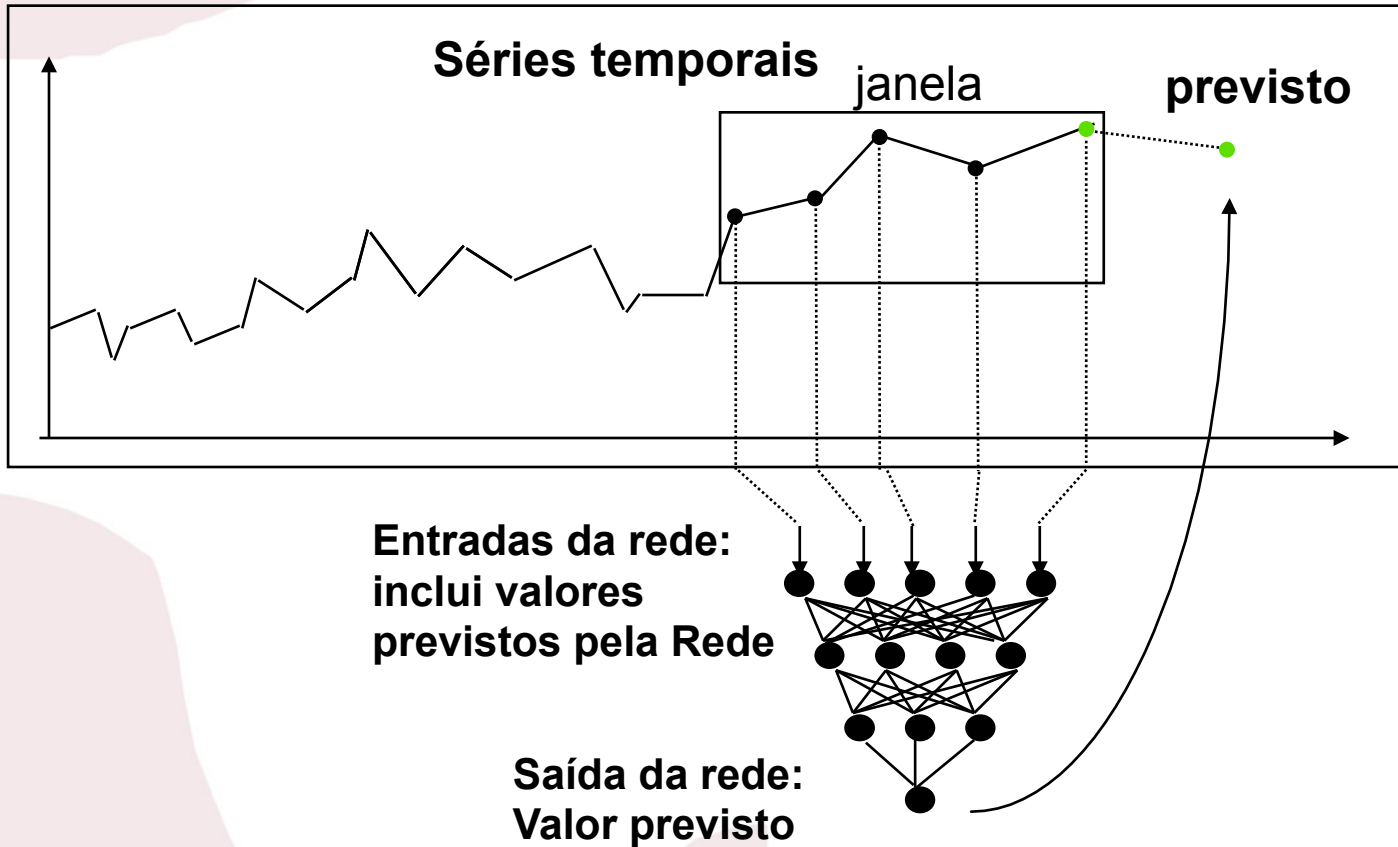
Exemplo: previsão utilizando apenas a série histórica como entrada



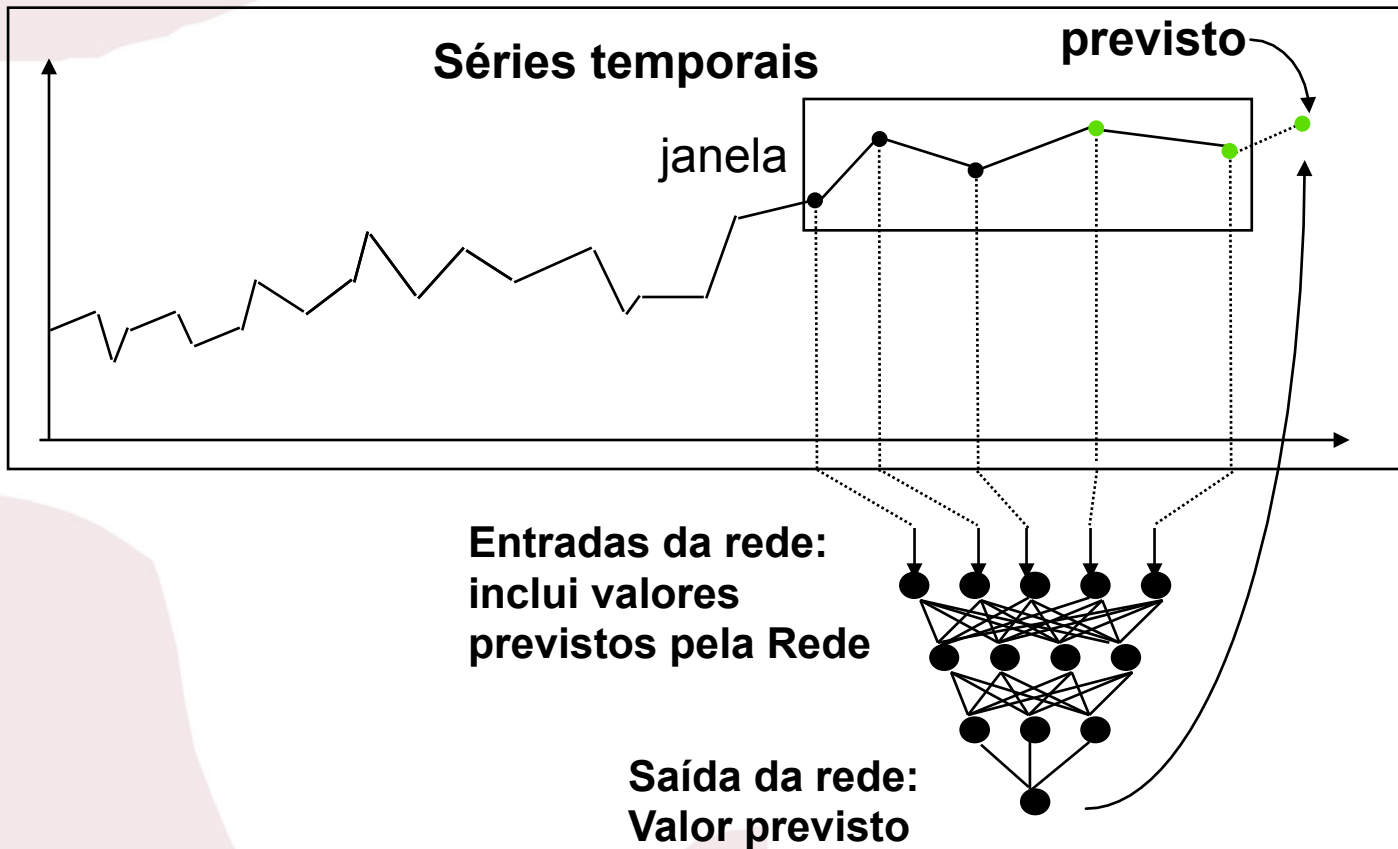
Exemplo: previsão utilizando apenas a série histórica como entrada



Exemplo: previsão utilizando apenas a série histórica como entrada



Exemplo: previsão utilizando apenas a série histórica como entrada



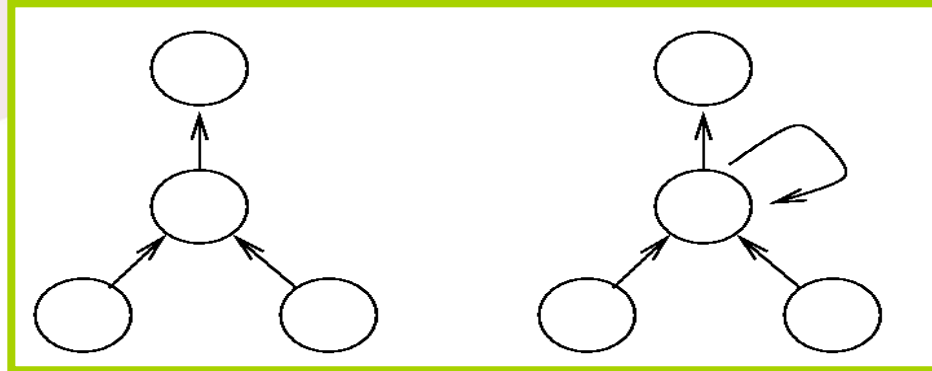
Mémoria Curta em Redes Recorrentes Simples (SRN)



- Unidades de contexto tem o papel de lembrar o estado interno anterior
- Unidades escondidas tem a tarefa de mapear ambas, a entrada externa corrente e o estado interno anterior a uma resposta desejada



Recorrente vs. Feedforward



- Entrada para a Rede Recorrente contem termo que reflete o estado da rede (ativação da unidade escondida) antes da apresentação do padrão atual
- Quando padrões seguintes são apresentados, os estados das unidades escondidas e de saídas são uma função de tudo o que a rede viu até então
- *Comportamento da rede é baseada na sua história, então apresentação e representação precisa ser pensada temporalmente*

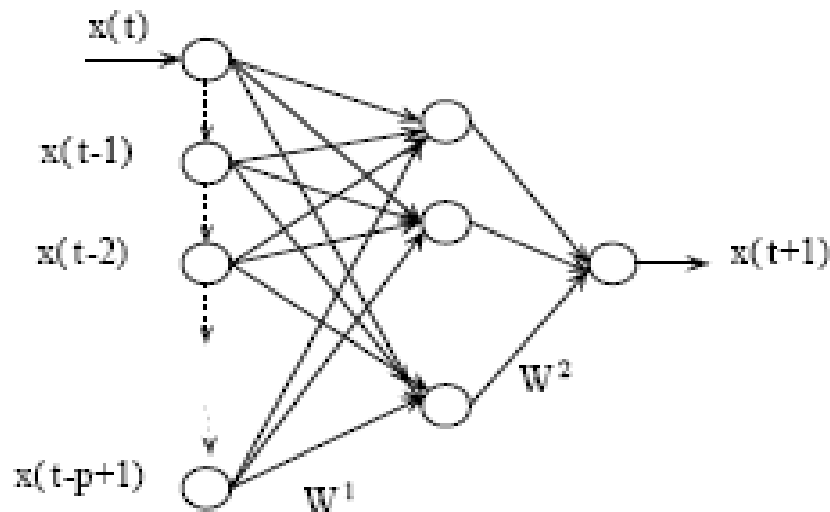

```
graph BT; INPUT[INPUT] --> HIDDEN[HIDDEN]; HIDDEN --> OUTPUT[OUTPUT]; INPUT --> HIDDEN; HIDDEN -.-> CONTEXT[CONTEXT];
```

Unidades de contexto no tempo t
guardam uma cópia das ativações
unidades escondidas no tempo $t-1$

output: years ago boy and girl ..

Redes Feedforward para Processamento Temporal

Arquitetura:



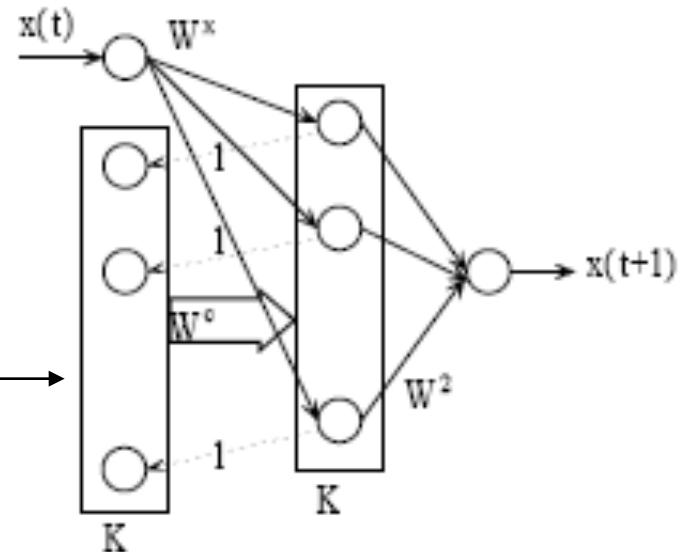
Funcionamento:

$$y = \sum_{k=1}^K w_k^2 f\left(\sum_{j=0}^{p-1} w_{kj}^1 x(t-j)\right)$$

Rede de Elman

Arquitetura:

Unidades de Contexto



Funcionamento

$$y = \sum_{k=1}^K w_k^2 h_k(t)$$

$$h_k(t) = f \left(w_k^x x(t) + \sum_{j=1}^K w_{kj}^c h_j(t-1) \right) \quad h_j(0) = 0.$$

unidades de contexto
contém cópias das
saídas das unidades
escondidas em
instantes anteriores

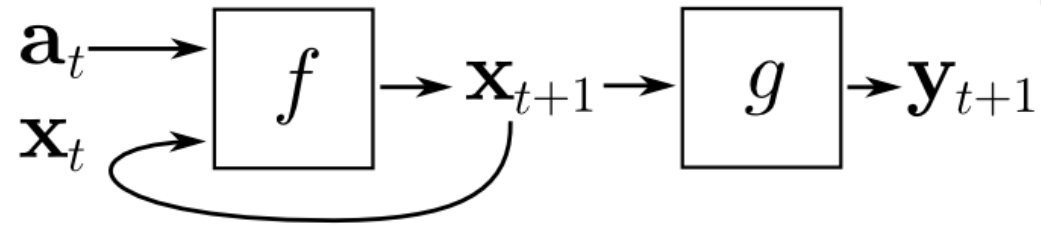
Backpropagation Through Time (BPTT)



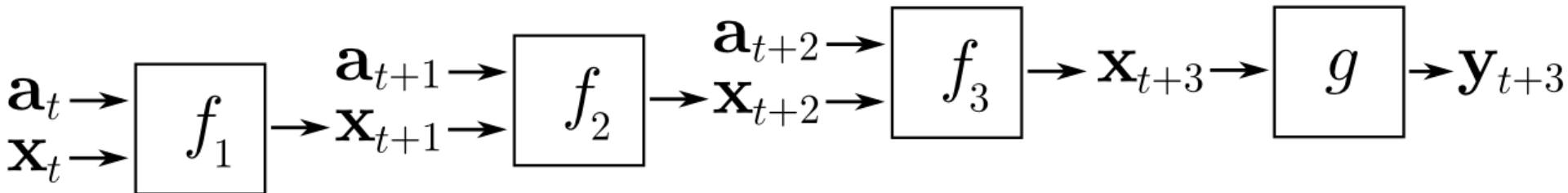
- Generalização da idéia é copiar ativações de entrada e escondidas por vários passos no tempo
- Quanto mais camadas de contexto (*copy layers*), mais história é incluída de forma explícita na computação do gradiente do erro
- Abordagem ficou conhecida como *Back Propagation Through Time (BPTT)*
- BPTT entendido como uma “aproximação” de computação do gradiente que leva em consideração não apenas as entradas mais recentes como todas vistas até o momento

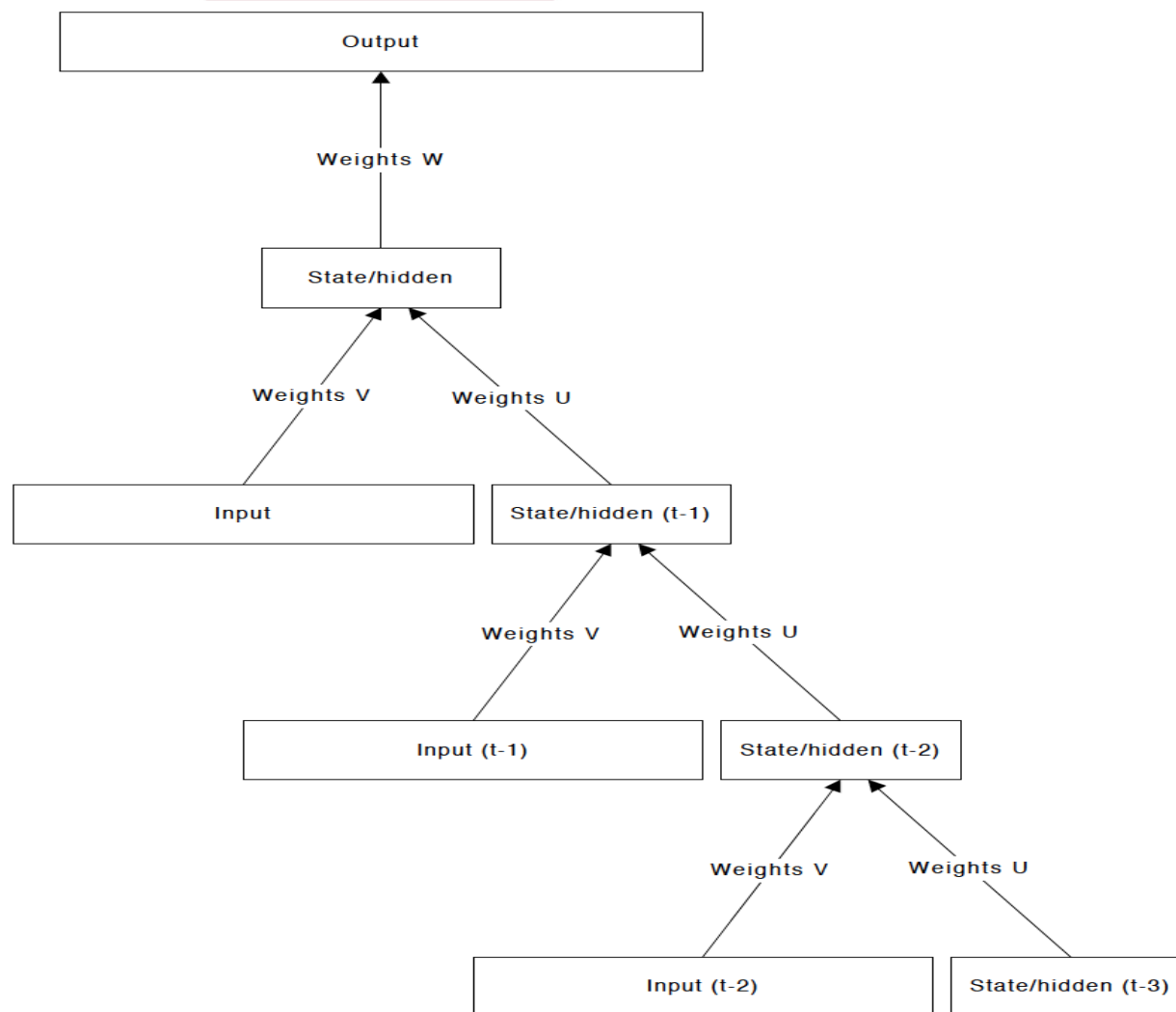


Backpropagation Through Time (BPTT)



↓ unfold through time ↓



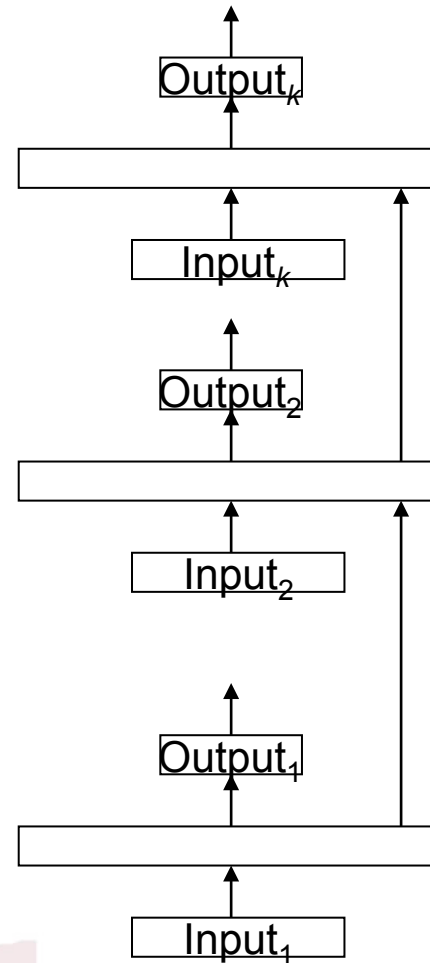
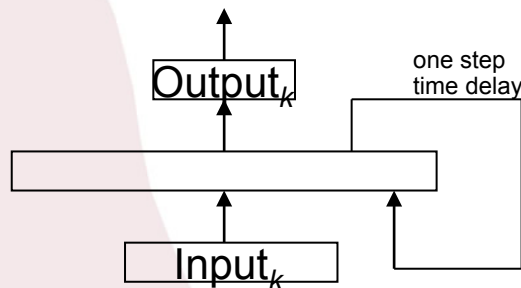


Note $k=1$
is just standard
BP with no
feedback

Figure 5: The effect of unfolding a network for BPTT ($\tau = 3$).

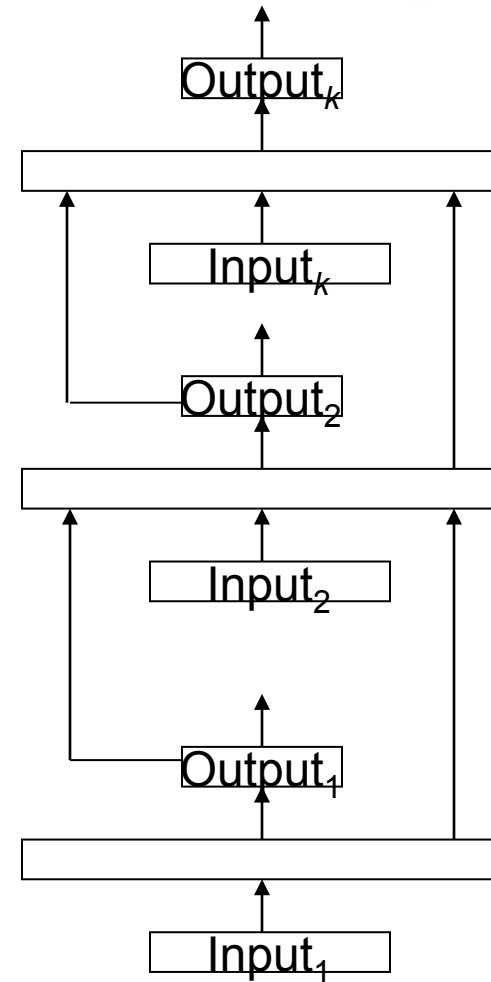
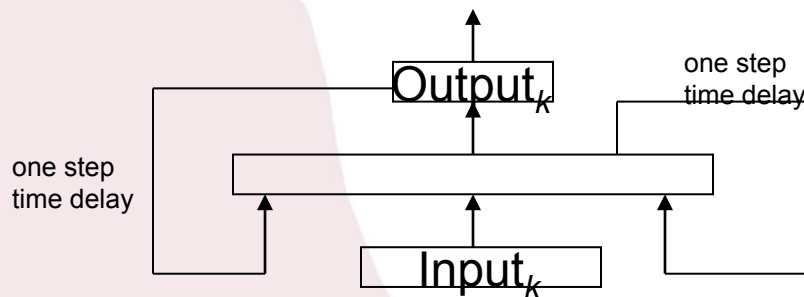
BPTT - Unfolding in Time ($k=3$) with output connections

Weights at each layer are maintained as exact copies



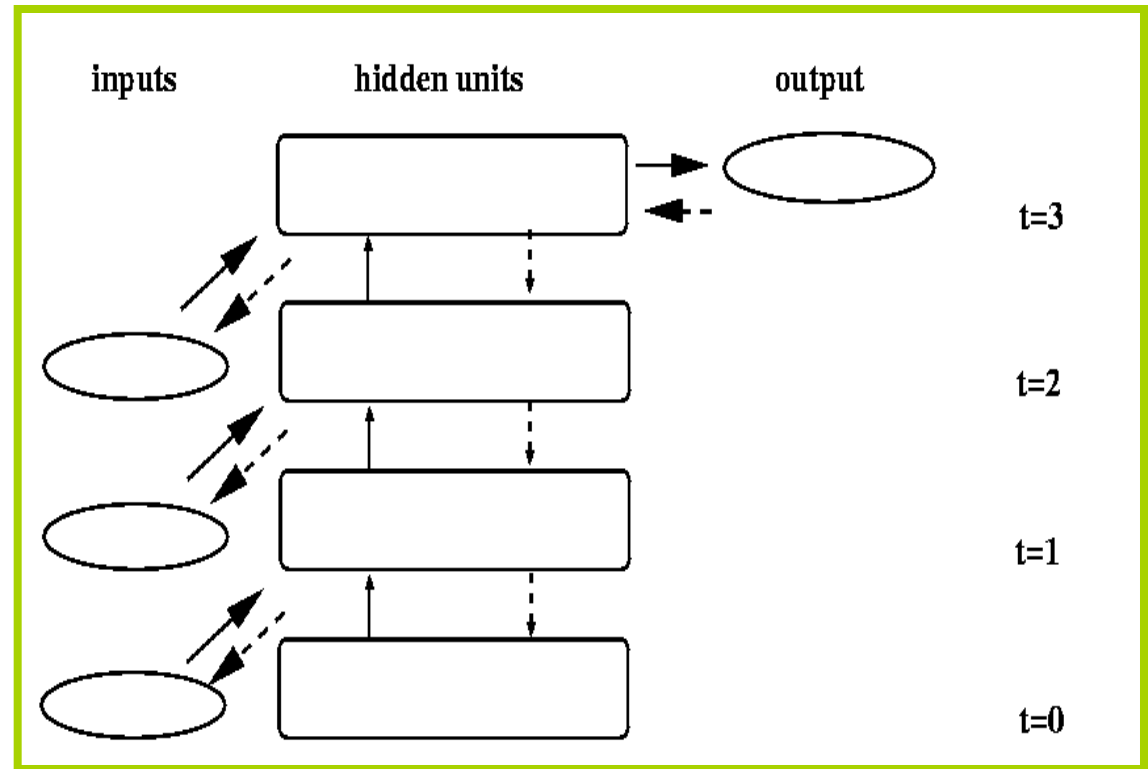
BPTT - Unfolding in Time ($k=3$) with output connections

Weights at each layer are maintained as exact copies



Backpropagation Through Time (BPTT)

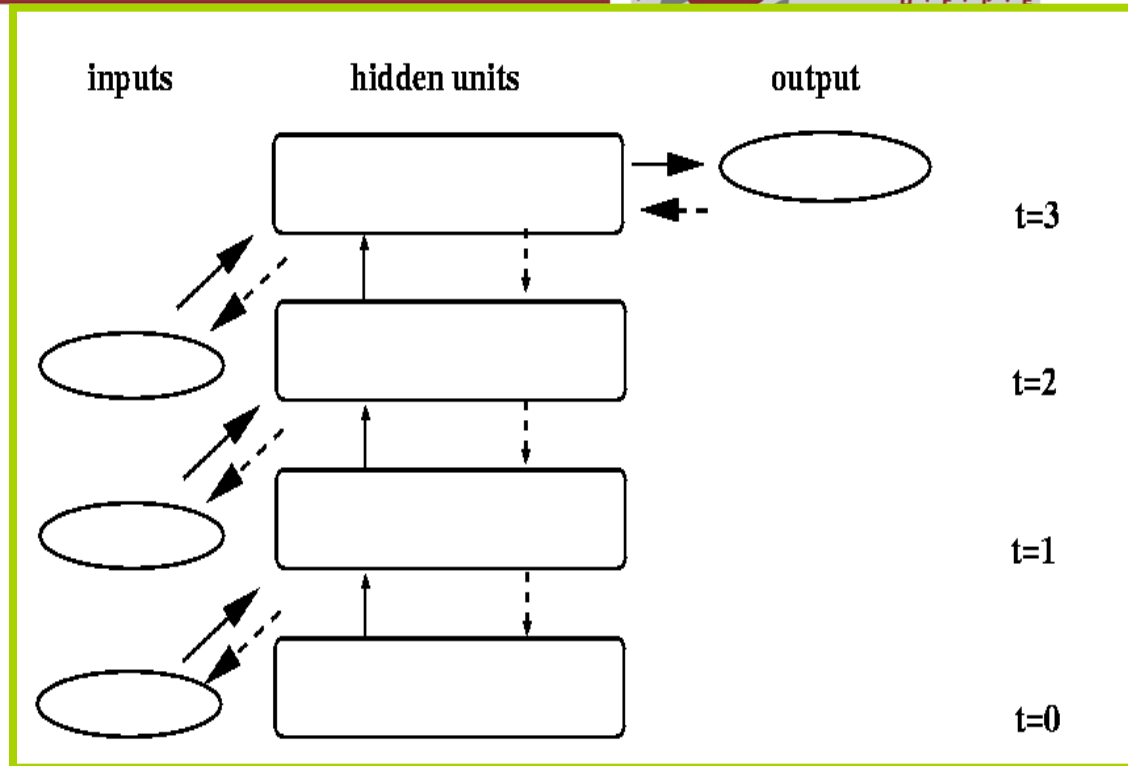
- Ativações das unidades escondidas e de entrada dos últimos 3 passos são armazenadas
- Linhas sólidas mostram que as ativações das unidades são definidas das entradas e camada escondida no passo anterior



- **Passo para trás é executado para determinar valores separados de delta (erro) para cada unidade e passo de tempo separadamente**

Backpropagation Through Time

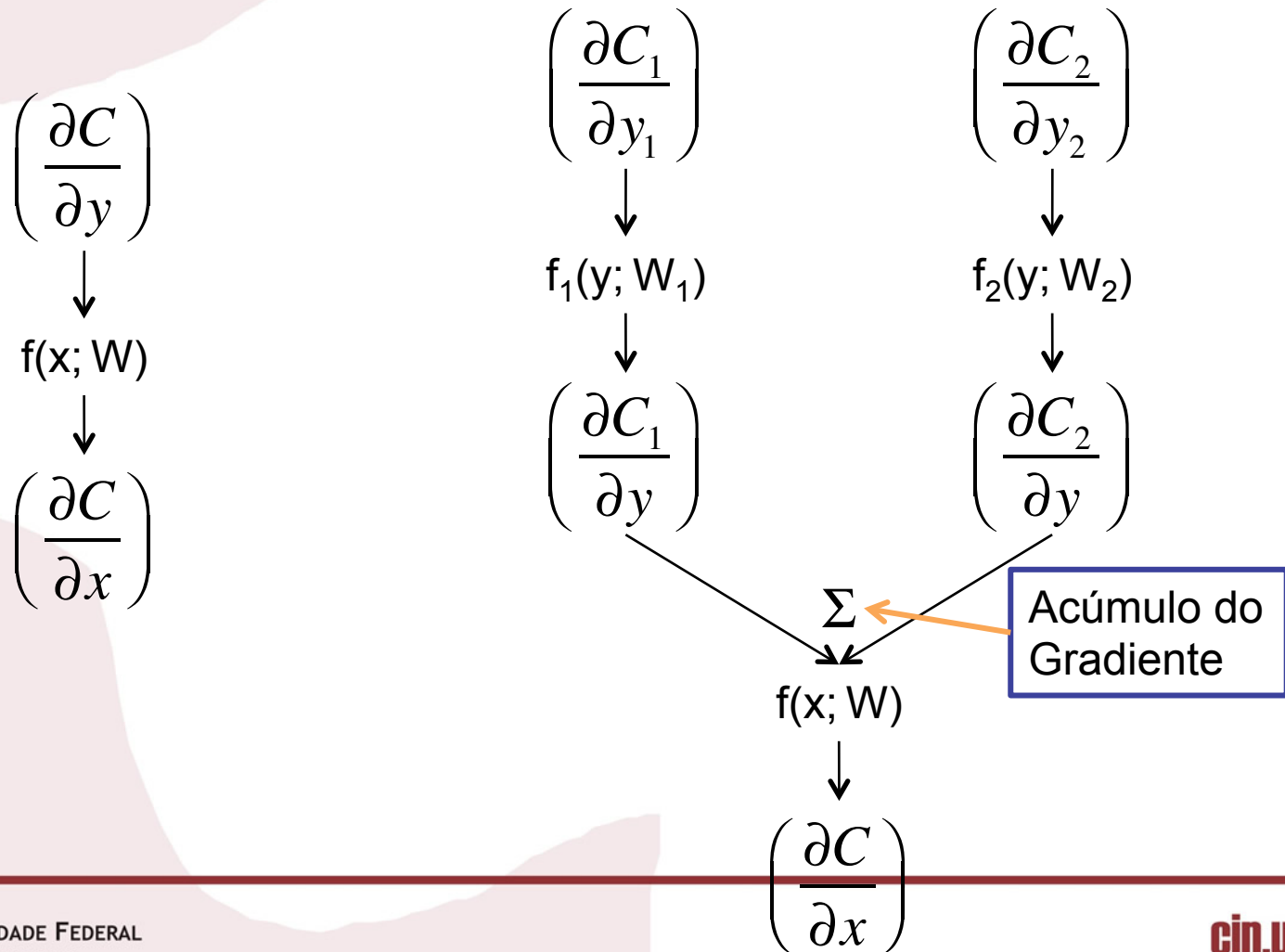
- Como cada camada anterior é uma cópia da camada um nível acima, introduzimos uma nova restrição: pesos em cada nível são idênticos
- Estamos “espalhando” o tempo em múltiplas camadas



A derivada parcial do erro com respeito a $w_{i,j}$ é a soma das derivadas parciais calculada para a cópia de $w_{i,j}$ entre cada 2 camadas

$$E_{total}(t_0, t_n) = \frac{1}{2} \sum_{t=t_0}^{t_n} \sum_{j \in A} e_j^2(t)$$

Backpropagation Through Time



Backpropagation Through Time



- A rede “desdobrada (no passo forward) é tratada como uma “grande” rede feedforward
- Recebe a série temporal inteira como entrada (ou pelo menos do tamanho da janela considerada)
- Atualização dos pesos é computada para cada cópia na rede desdobrada e somada (ou tirada a média)
- E aplicada em todos os pesos da RNN (pesos iguais)



- Rede de Elman e BPTT (generalização) buscam aproximar a computação do gradiente baseada nas entradas passadas
 - Mantendo algoritmo BP padrão
 - BPTT foi aplicado com sucesso em um número de aplicações (exemplo: modelagem de sinais de eletrocardiogramas – ECG)
 - Tarefa principal, nesse caso, é produzir uma resposta específica para uma sequência de entradas

- Principal desvantagem é que requer uma quantidade alta de recursos
 - Armazenamento – da história inteira
 - Processamento – cálculo do gradiente para todas as camadas
 - Dados – muitos exemplos para melhores representações

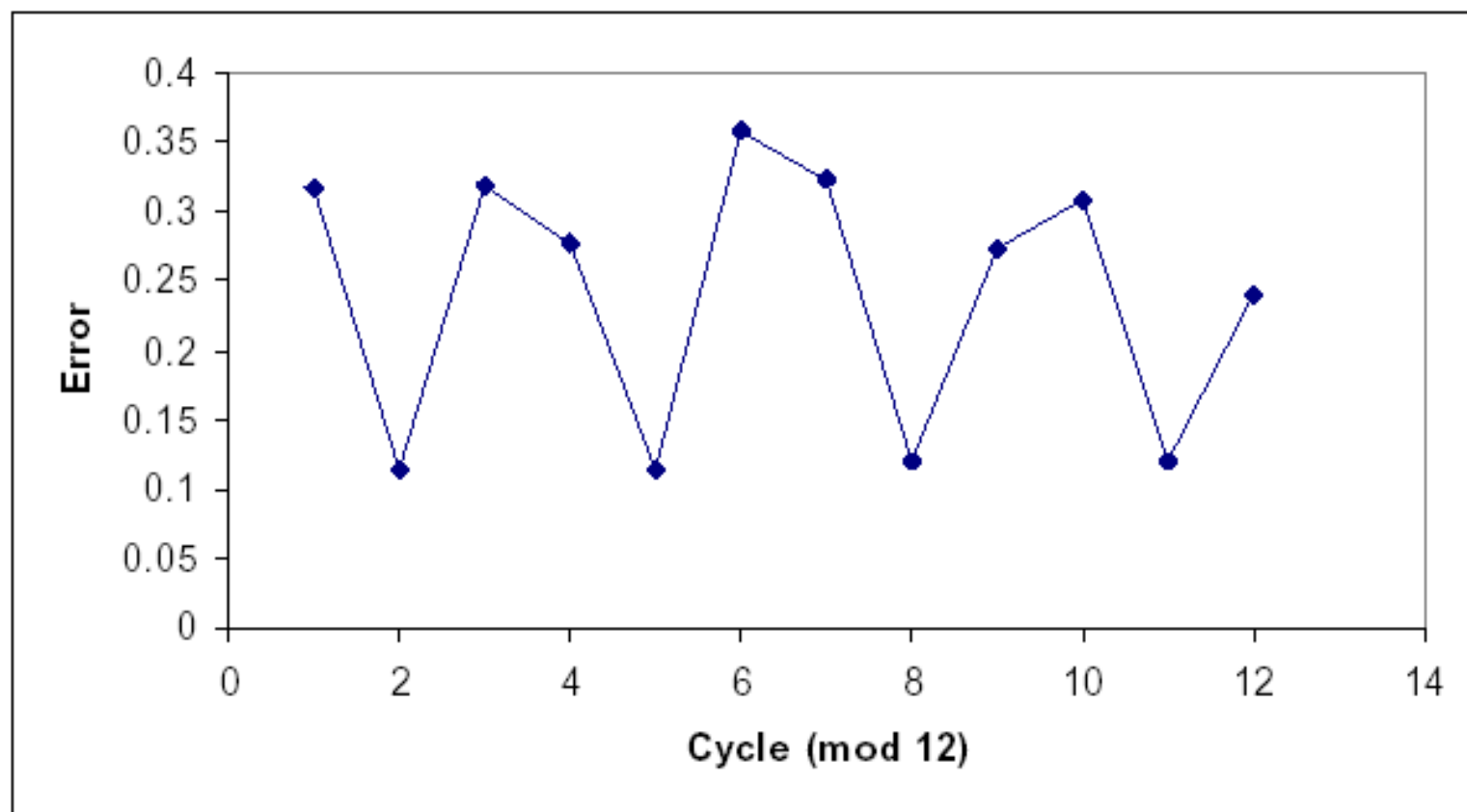
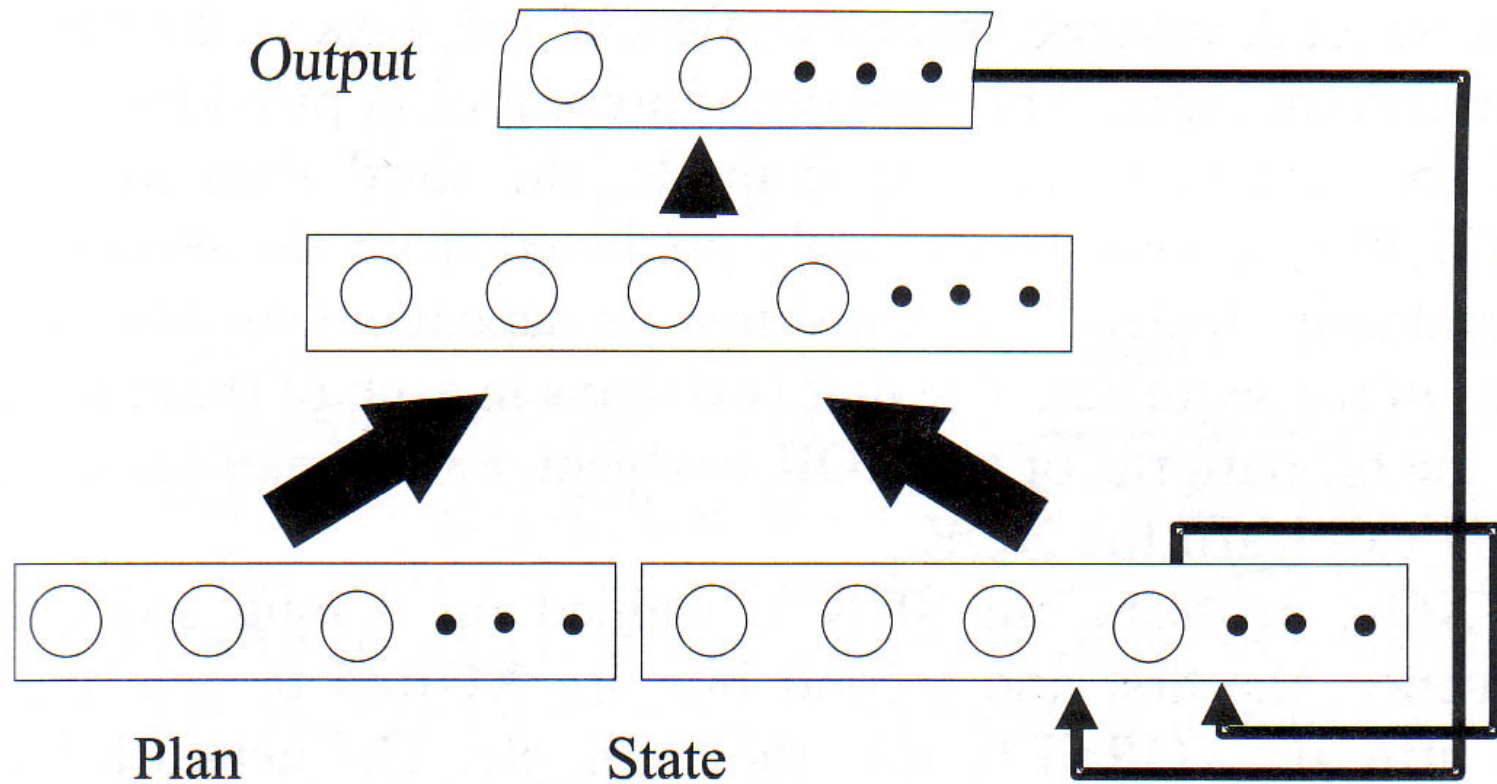


Figure 3. Graph of root mean squared error over 12 consecutive inputs in sequential XOR task. Data points are averaged over 1200 trials.

Rede de Jordan

Feedback das Unidades de Saída



Rede de Jordan

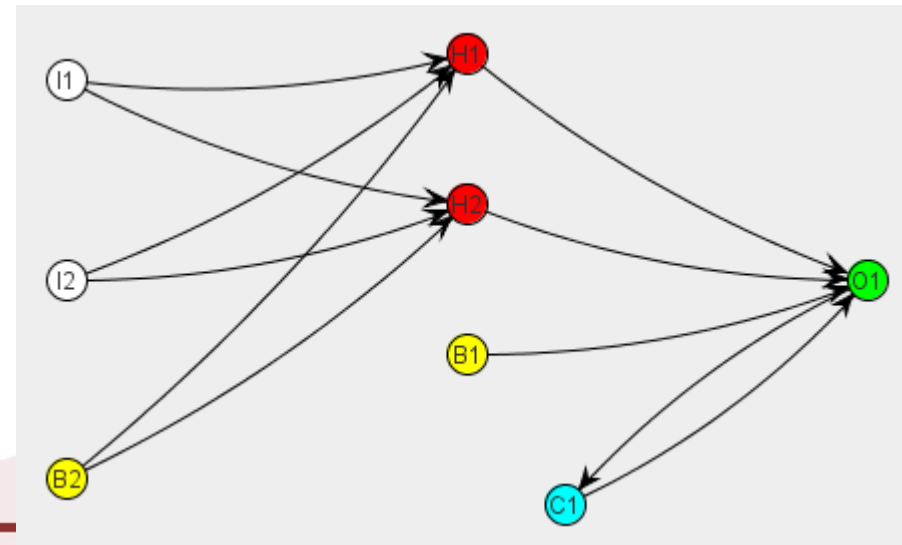
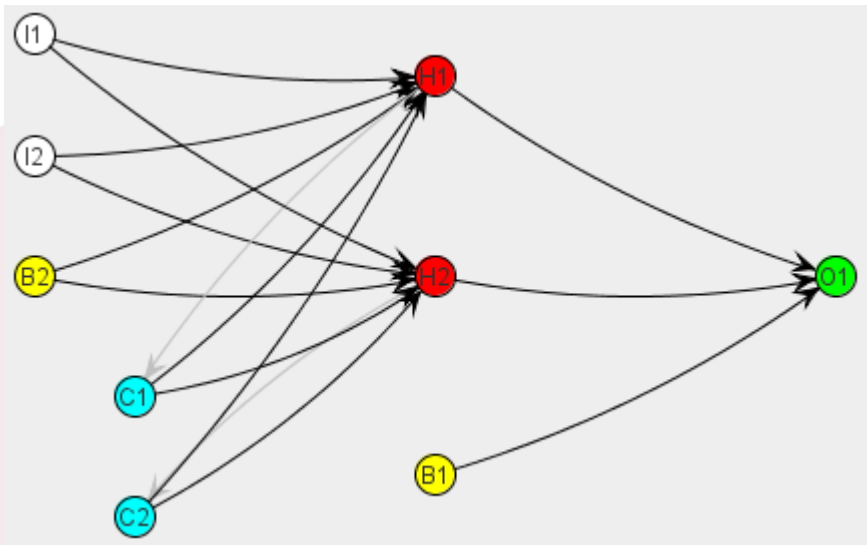


- Conexões da camada de saída para a camada de entrada e conexões das entradas para elas mesmas
- Aplicada em tarefas dependentes de sequências de estados sucessivos (padrões temporais)
- Pode ser treinada com backpropagation
- Rede possui uma forma de guardar memórias curtas (short-term memory)
- Semelhante à Rede de Elman, para treinamento com backpropagation



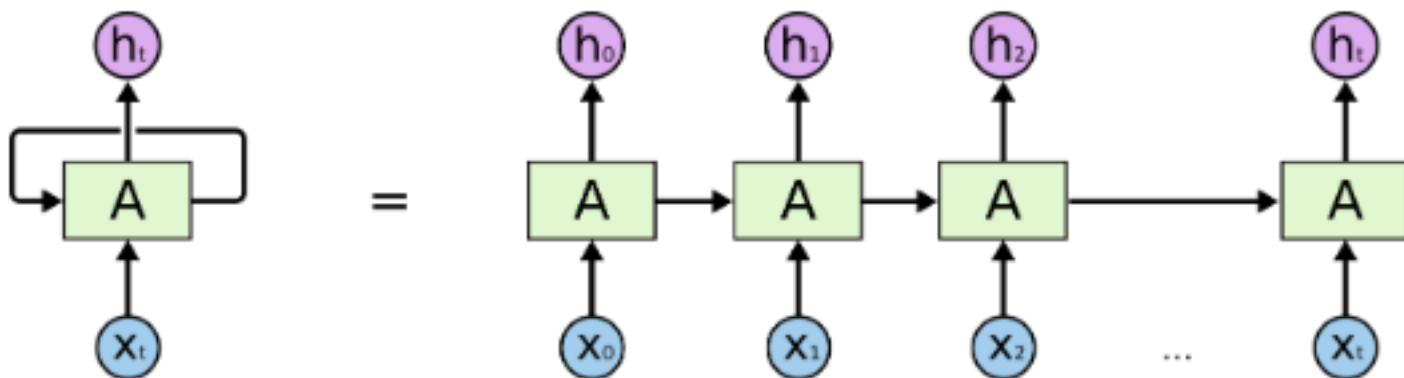
Diferenças

- Elman – guardam memória das unidades escondidas
- Jordan – guardam memória das unidades de saída
- Podem produzir desempenhos diferentes em aplicações diferentes



Long Short Term Memory Networks (LSTM Networks)

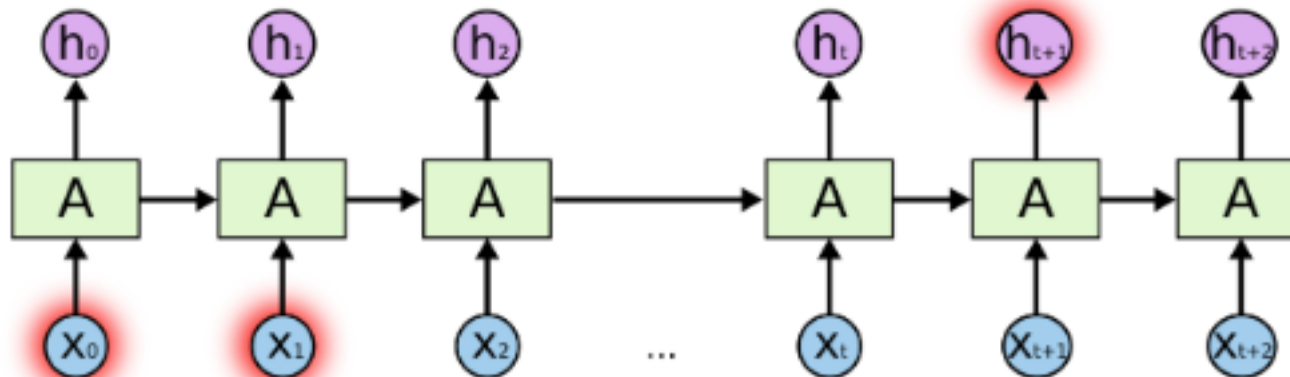
Rede Recorrente (RNN) Clássica



An unrolled recurrent neural network.

Problemas com Redes Recorrentes (RNN) Clássicas

Dependências temporais longas!!



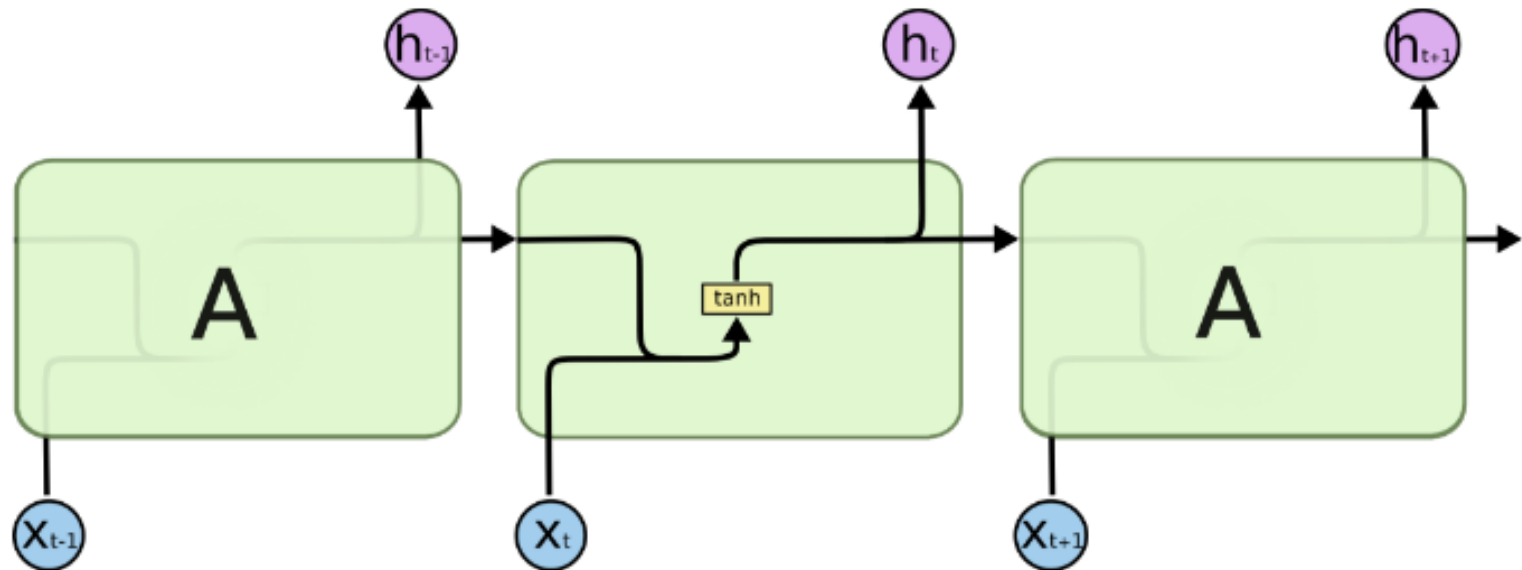
Na teoria, OK!! Na prática, “*the vanishing gradient problem...*”

LSTMs

Dependências temporais longas!!

Na teoria, OK!! Na prática, “*the vanishing gradient problem...*”

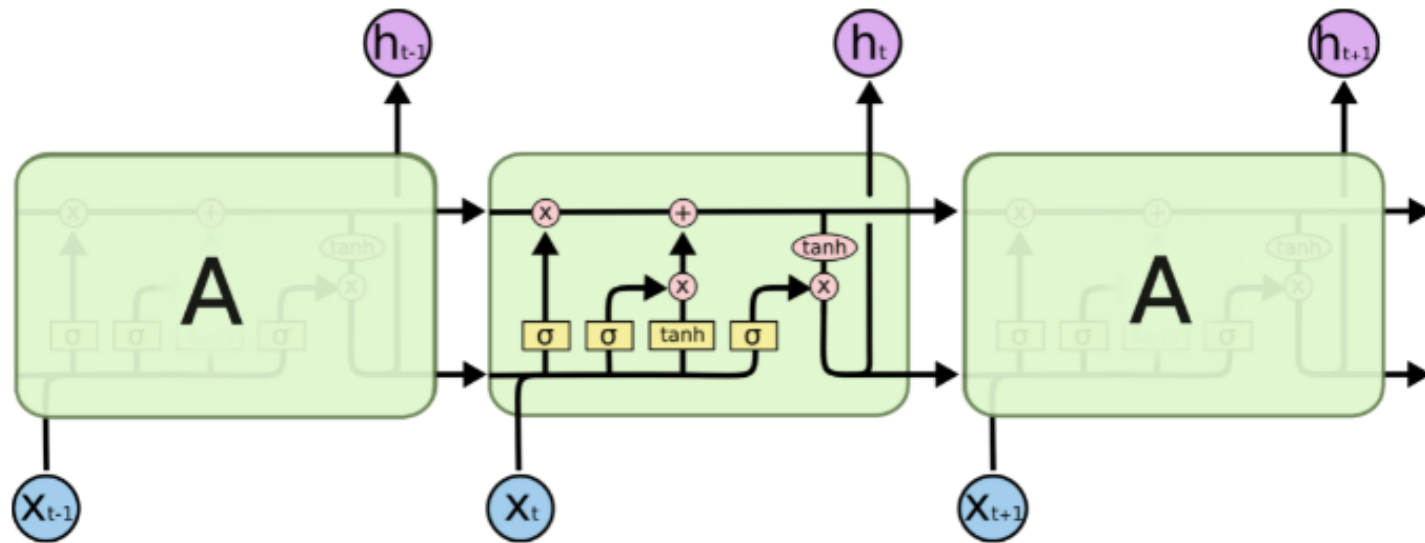
Na RNN Clássica



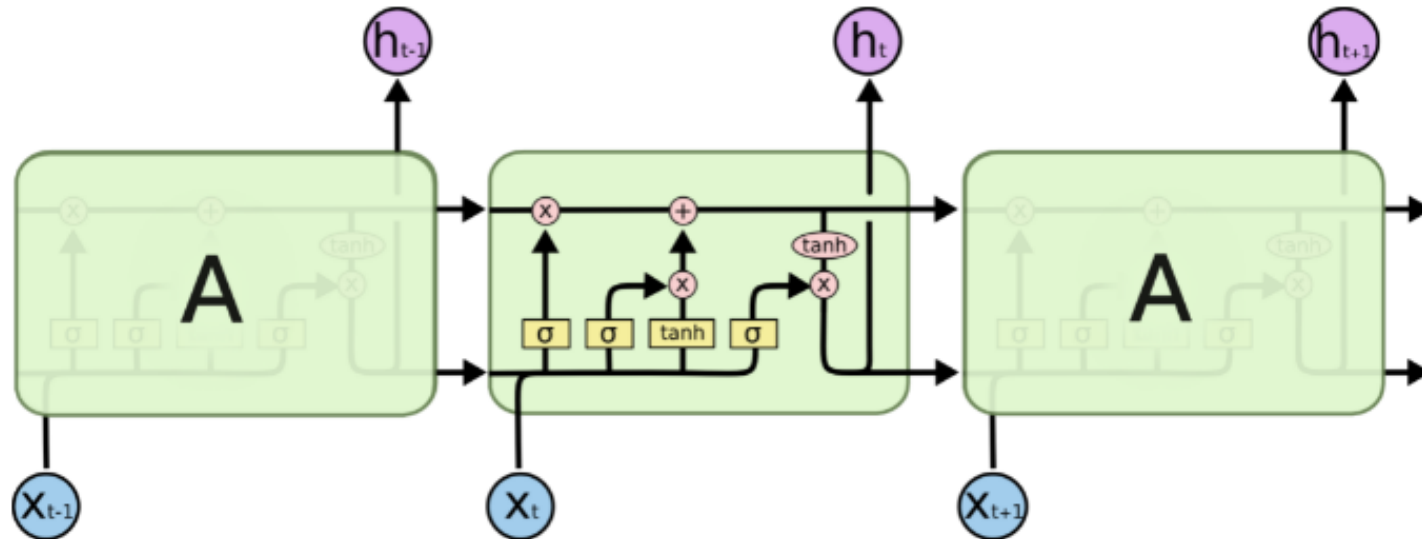
Uma rede neural simples em cada módulo

Na LSTM

Estrutura com 4 redes neurais simples em cada módulo

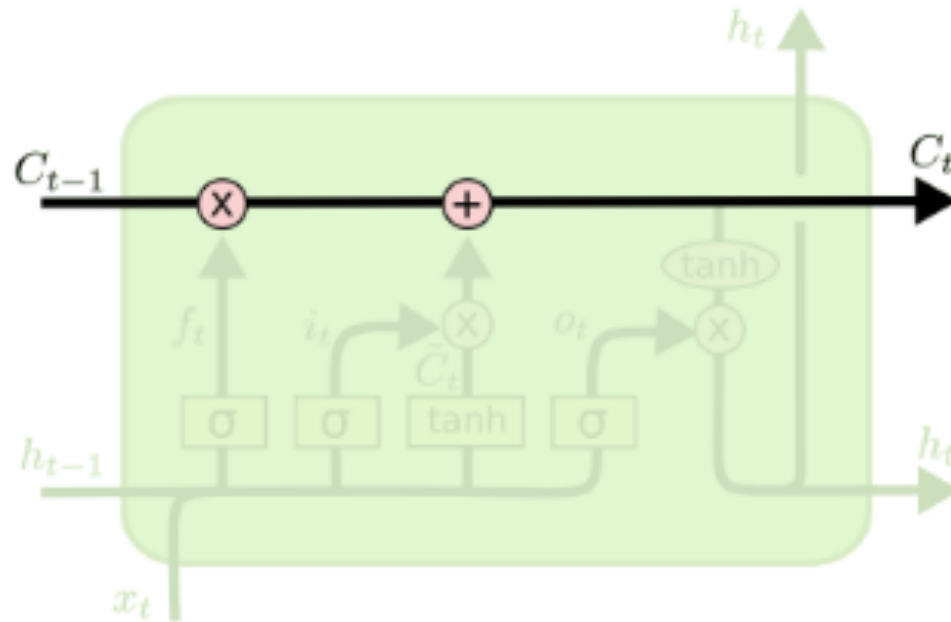


Na LSTM



Conceito de célula de memória que carrega a informação

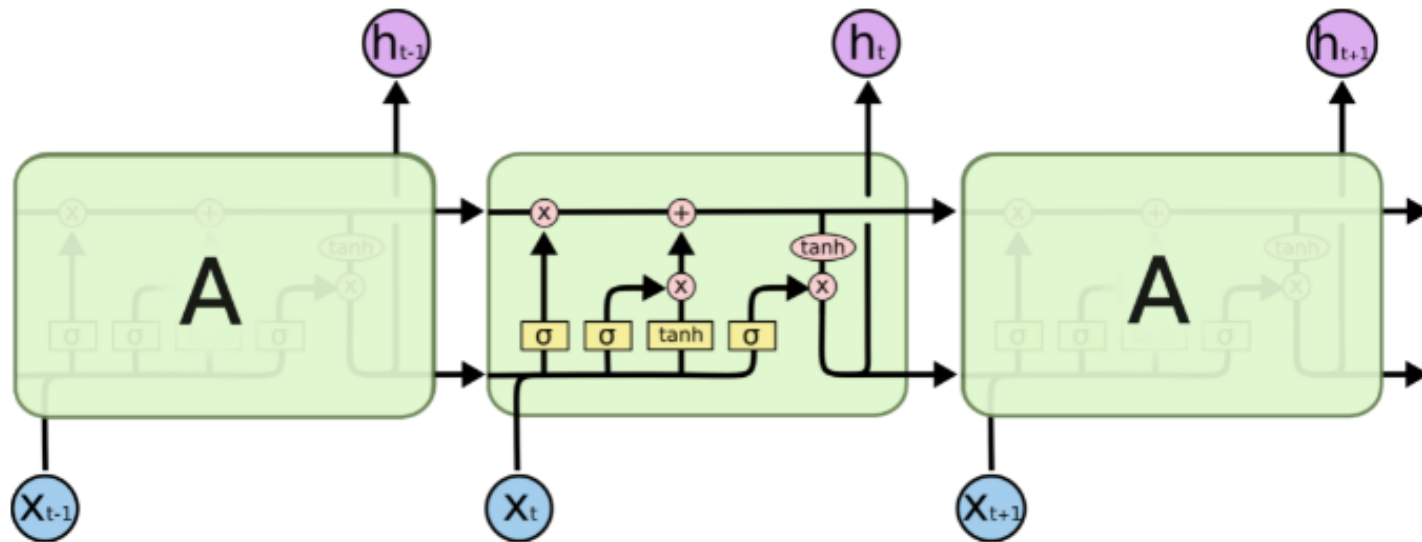
Essa é Ideia Central da LSTM



Conceito de célula de memória que carrega a informação

Na LSTM

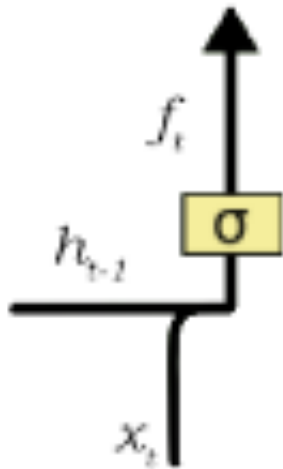
Esteira transportadora ou montagem (*conveyor belt*)



Memória é esquecida, alterada ou mantida através de filtros!

The Forget Gate (Filtro do Esquecimento)

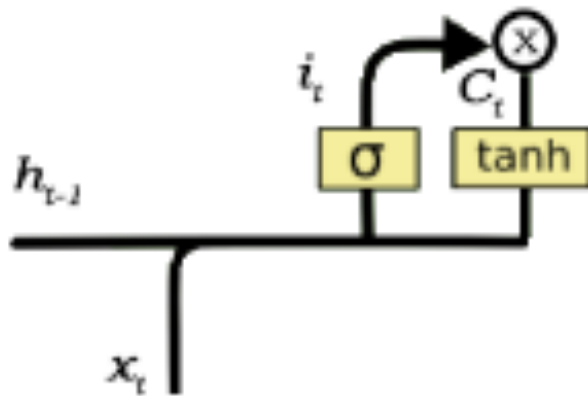
Bob is a nice person. Dan on the other hand is evil.



$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

The Input Gate (Filtro da Entrada)

Bob knows swimming. He told me over the phone that he had served the navy for 4 long years.

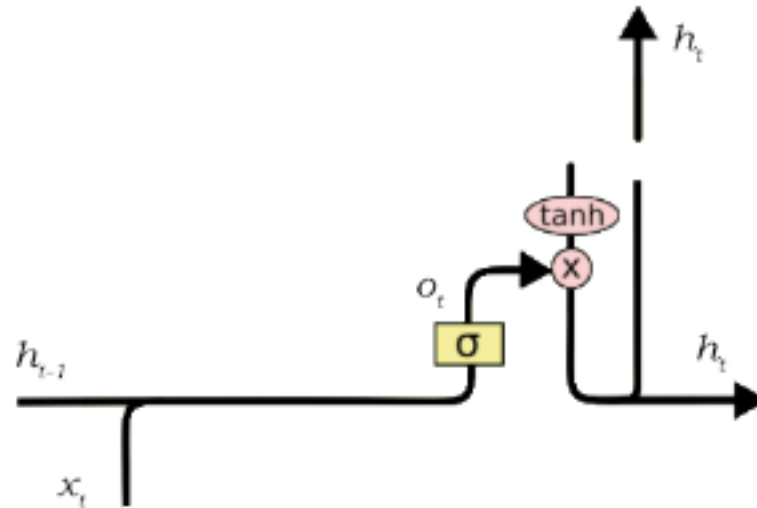


$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

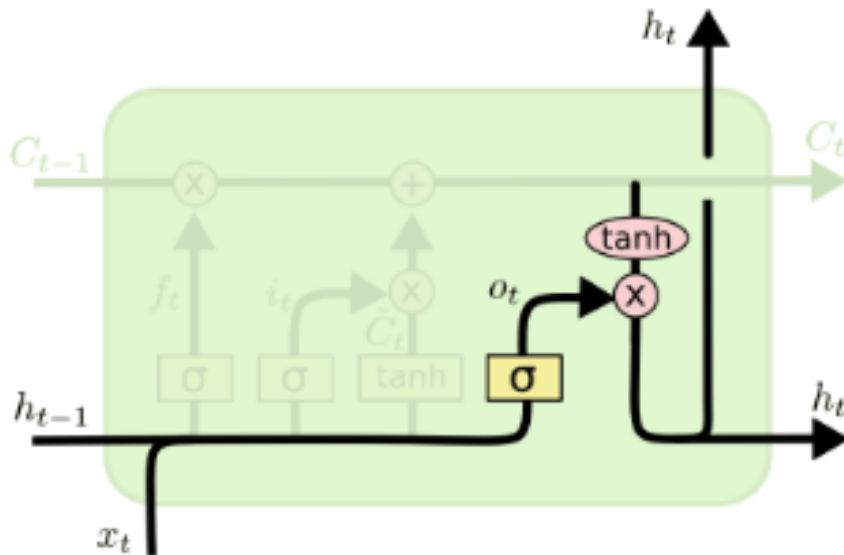
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

The Output Gate (Filtro da Saída)

Bob fought single handedly with the enemy and died for his country. For his contributions brave ____.



Saida do Módulo



$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

Sentiment Classification



- Classify a restaurant review from Yelp! OR movie review from IMDB OR ... as positive or negative
- Inputs: Multiple words, one or more sentences
- Outputs: Positive / Negative classification
- “The food was really good”
- “The chicken crossed the road because it was uncooked”



Sentiment Classification



RNN

↑
The

→
 h_1



UNIVERSIDADE FEDERAL
DE PERNAMBUCO

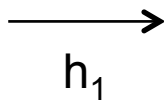
cin.ufpe.br

Sentiment Classification

RNN



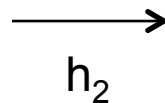
The



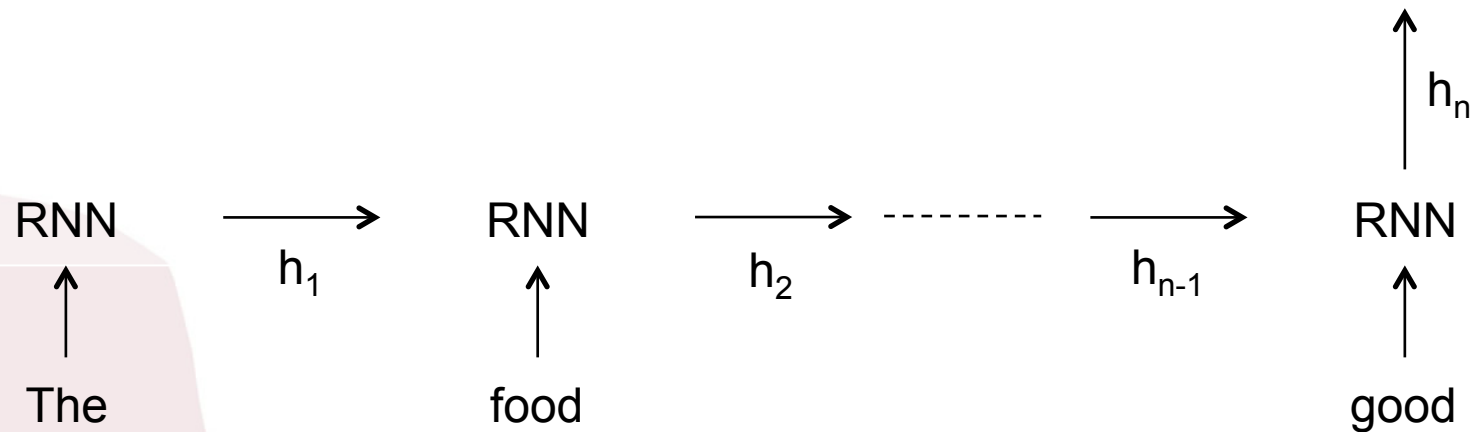
RNN



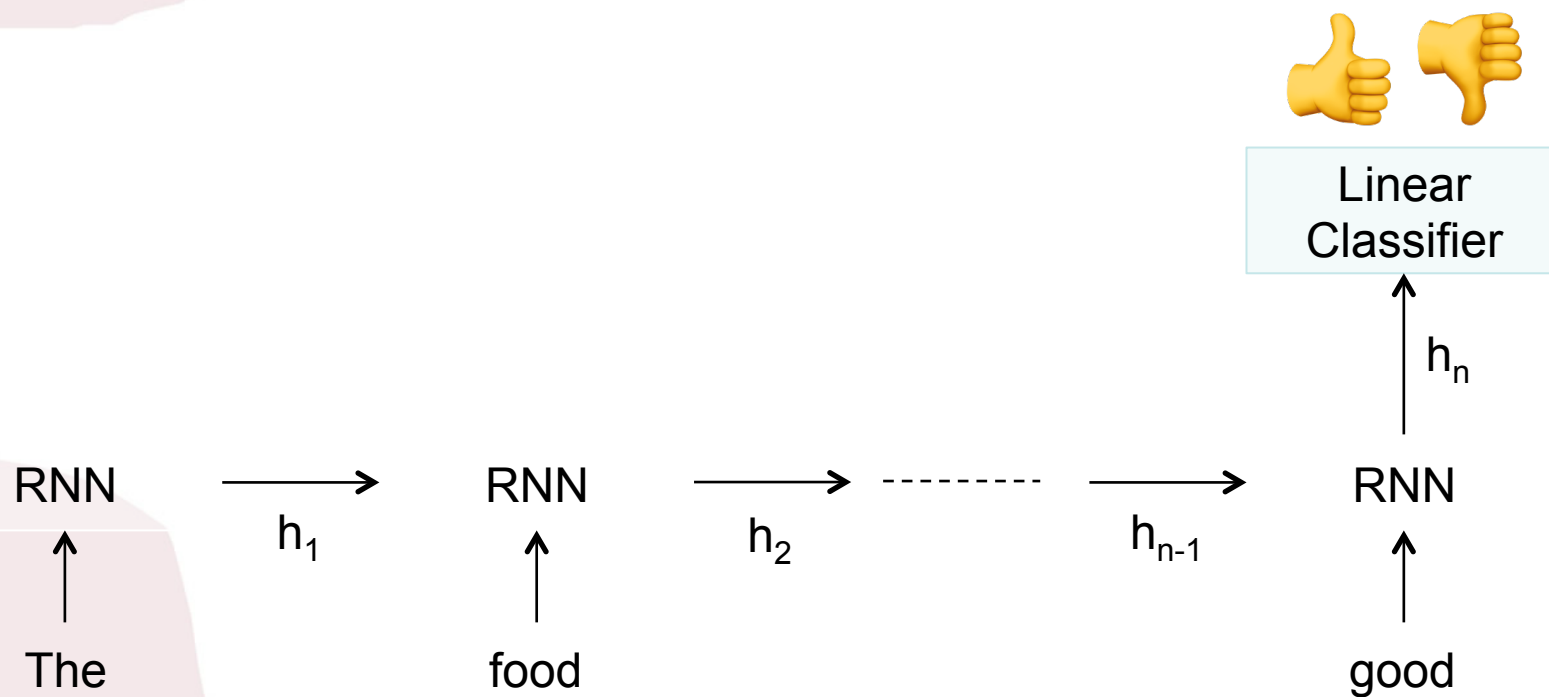
food



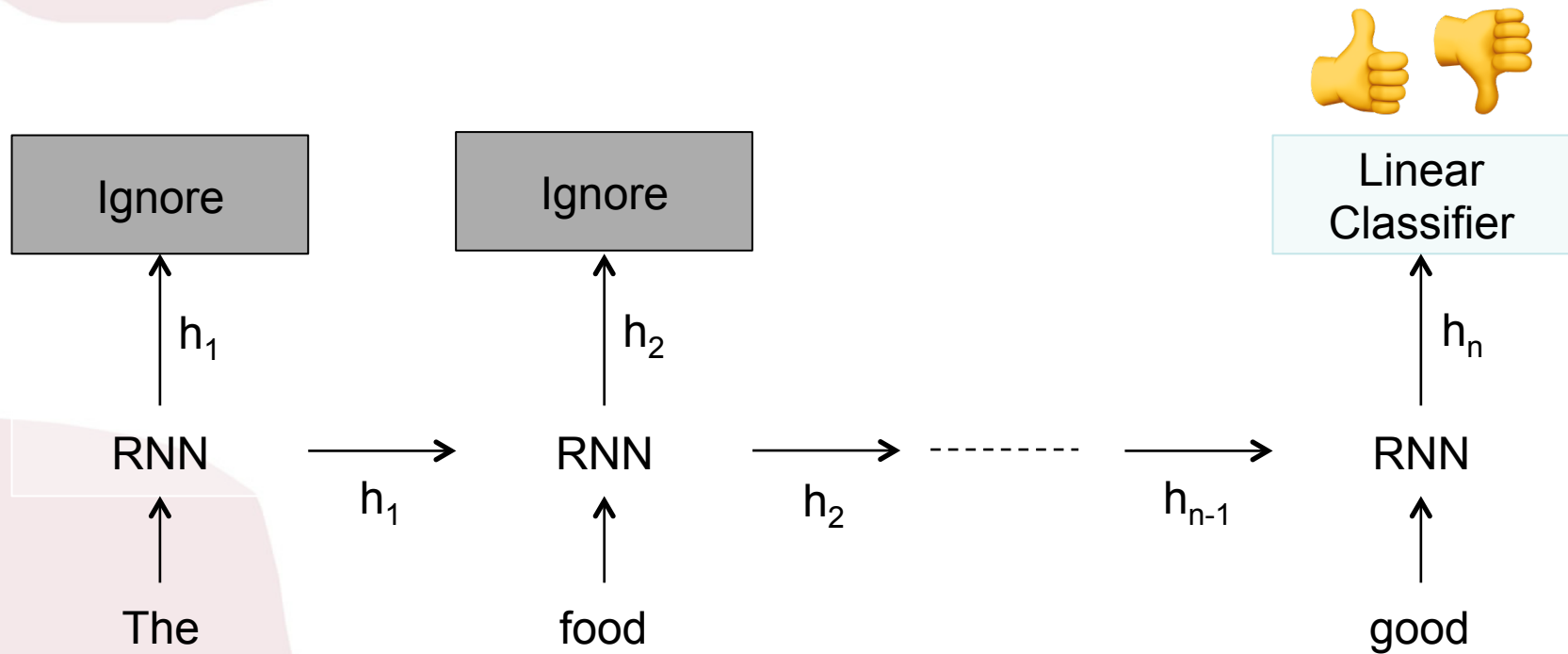
Sentiment Classification



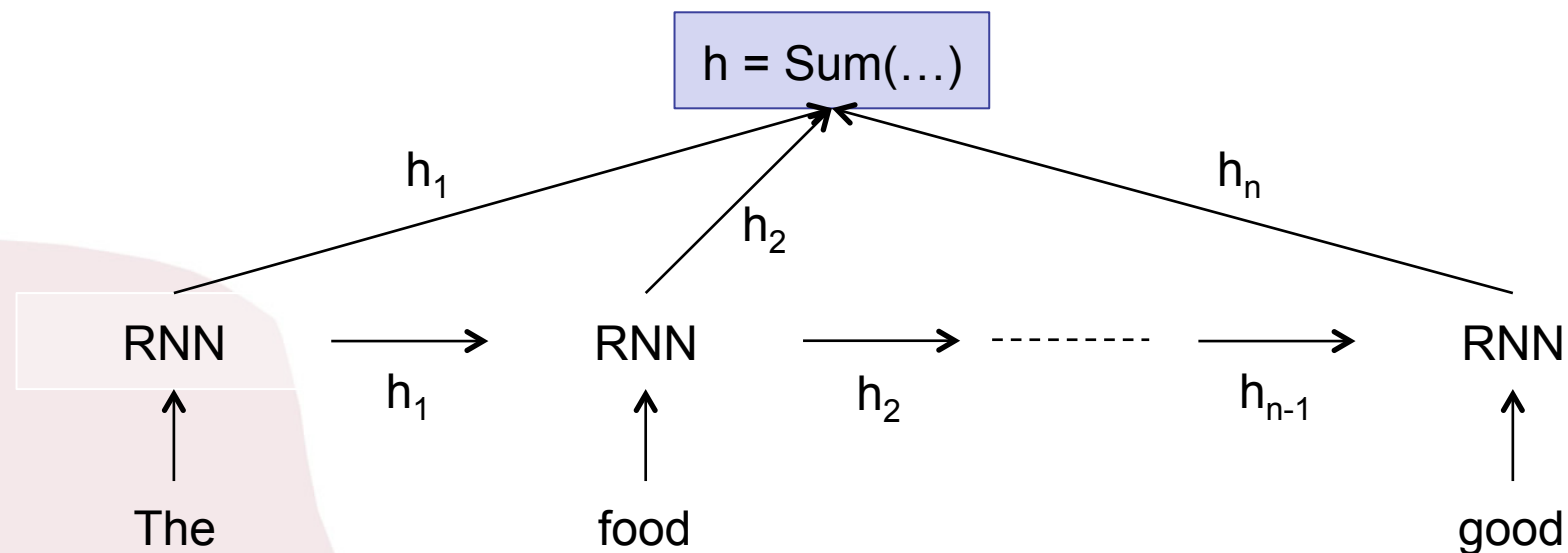
Sentiment Classification



Sentiment Classification



Sentiment Classification



Sentiment Classification

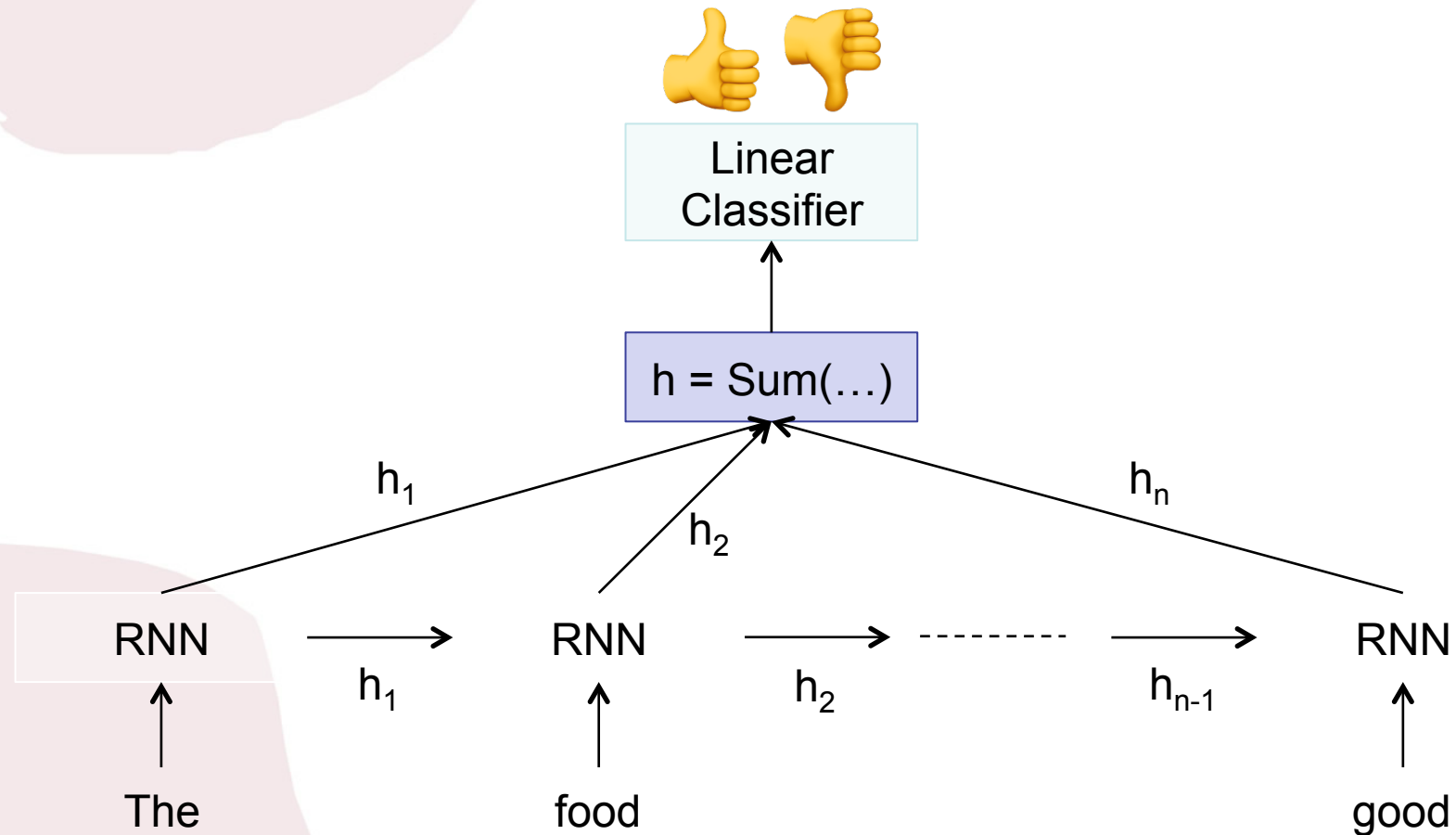


Image Captioning

- Given an image, produce a sentence describing its contents
- Inputs: Image feature (from a CNN)
- Outputs: Multiple words (let's consider one sentence)



: The dog is hiding

Image Captioning

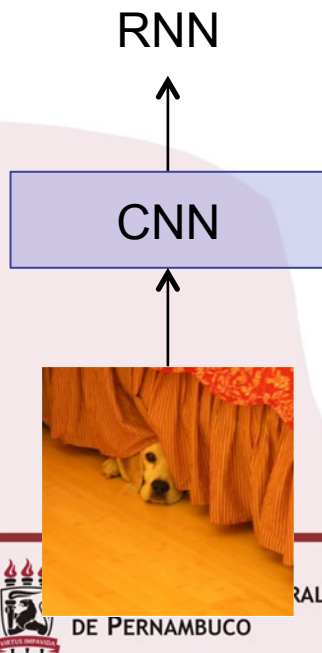


Image Captioning

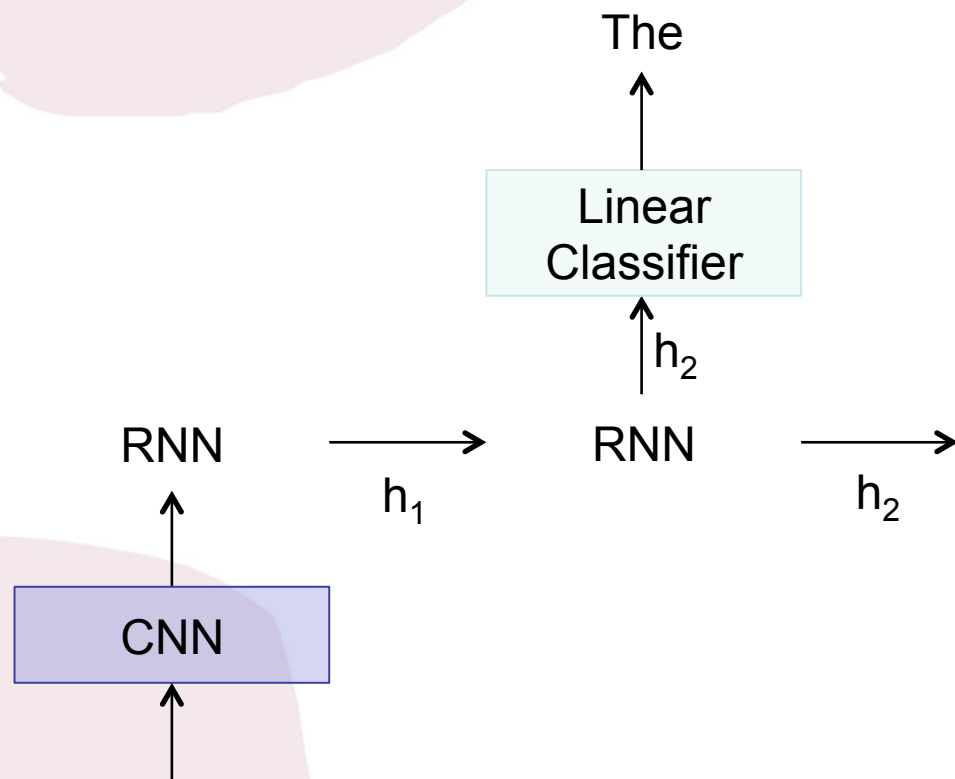
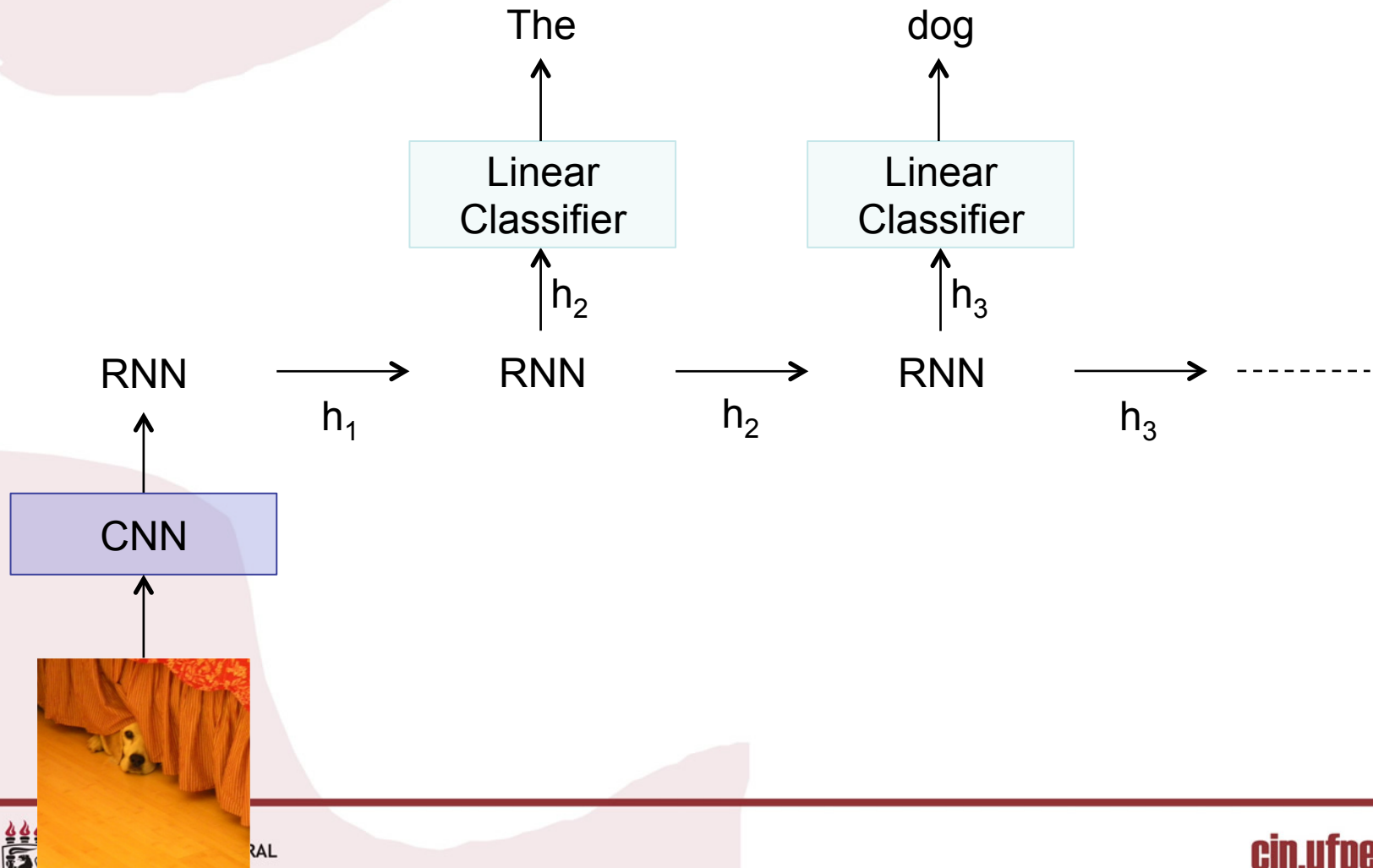


Image Captioning



RNN Outputs: Image Captions

**A person riding a
motorcycle on a dirt road.**



Two dogs play in the grass.



**A herd of elephants walking
across a dry grass field.**



**A group of young people
playing a game of frisbee.**



**Two hockey players are
fighting over the puck.**

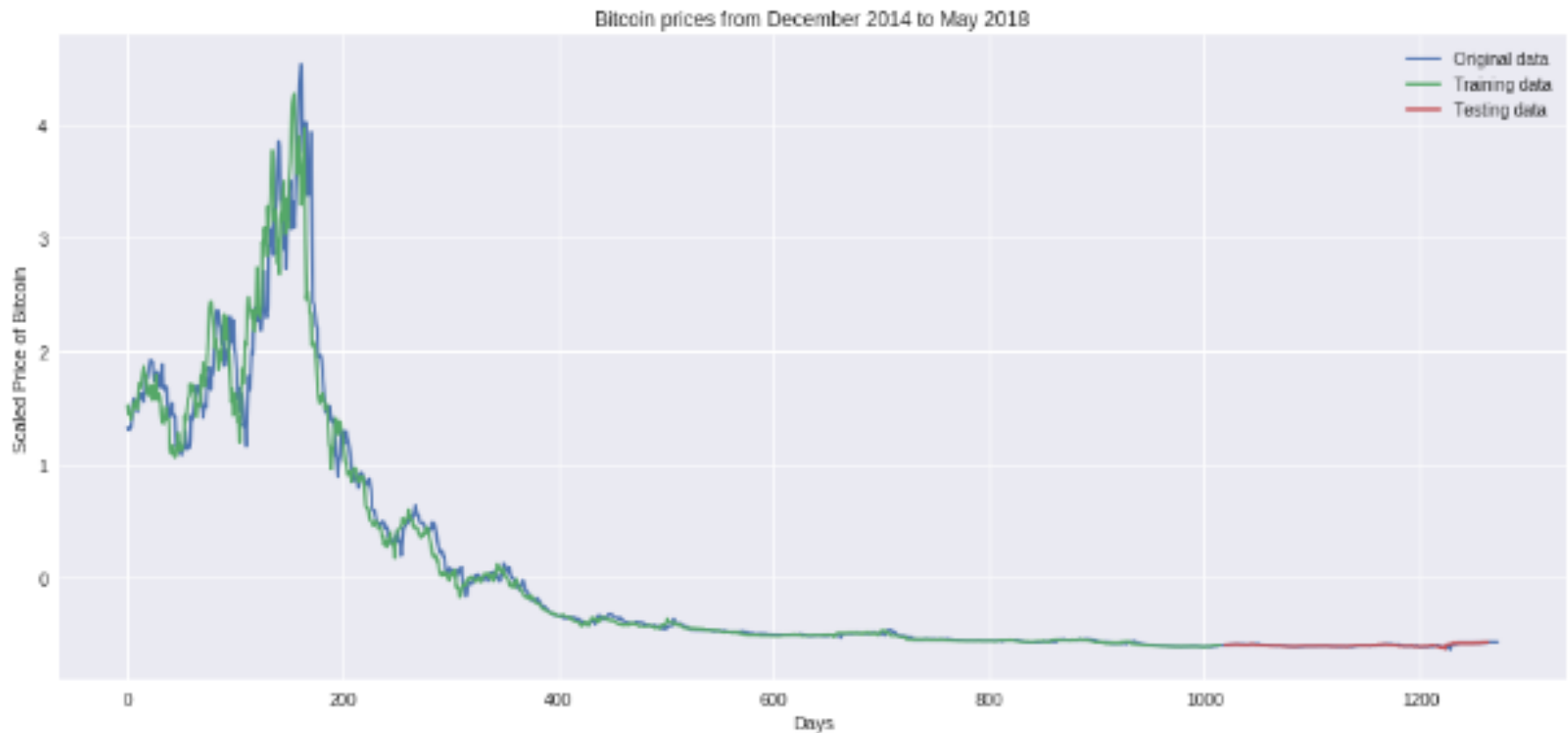


**A close up of a cat laying
on a couch.**



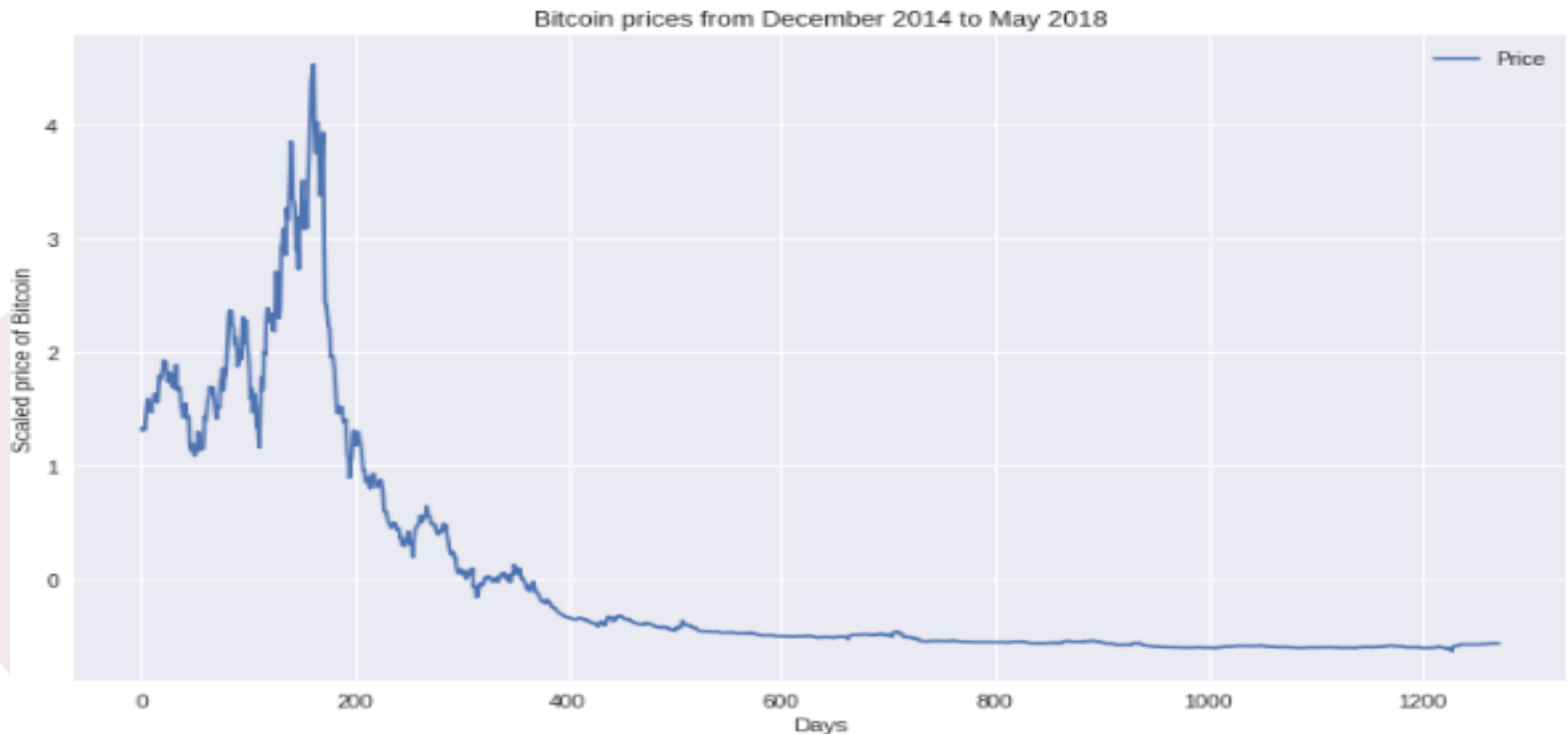
Mini-projeto com LSTMs

Usar os primeiros 80% dos registros para treinar e restante 20% para teste



Mini-projeto com LSTMs

Empregar LSTM para prever o preço do Bitcoin de dezembro de 2014 a maio de 2018!



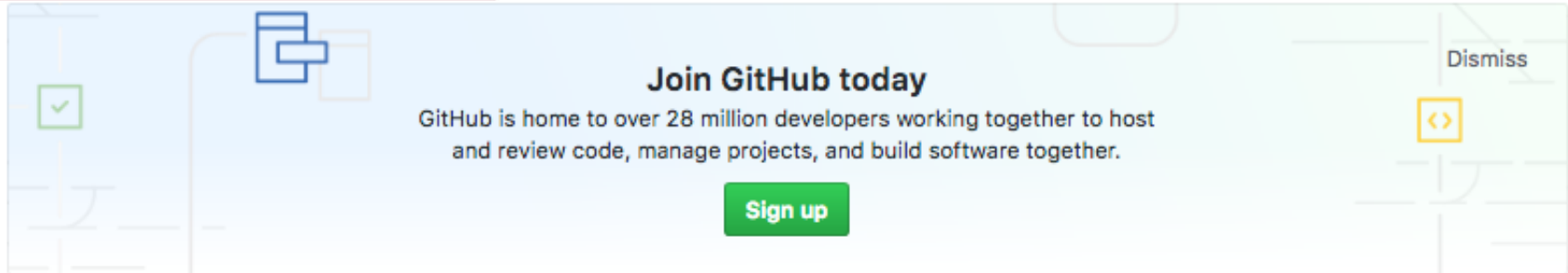
Mini-projeto com LSTMs



- <https://heartbeat.fritz.ai/a-beginners-guide-to-implementing-long-short-term-memory-networks-lstm-eb7a2ff09a27>



Mini-projeto com LSTMs (Credits Brian Mwangi)



This is a guide on how you can implement time series in RNNs using LSTMs to determine the future prices of bitcoin

5 commits1 branch0 releases1 contributor

Branch: masterNew pull requestFind fileClone or download

brynmwangy	Add files via upload	Latest commit efff086 on 27 May
README.md	Initial commit	4 months ago
btc.csv	Add files via upload	4 months ago
btc.ipynb	Created using Colaboratory	4 months ago

README.md

predicting-bitcoin-prices-using-LSTM

This is a guide on how you can implement time series in RNNs to determine the future prices of bitcoin

Mini-projeto com LSTMs (Credits Brian Mwangi)

Importing libraries

We're going to work with a variety of libraries that we'll have to install first in our Colab notebook and then import into our environment.

```
#install all the required dependancy libraries
!pip install tensorflow #LSTM implementation
!pip install numpy #for matrix multiplication
!pip install pandas #define the data structures
!pip install matplotlib #for visualization
!pip install scikit-learn #for normalizing our data(scaling)

#importing the libraries
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
import io
%matplotlib inline
```

Geração de Textos com LSTMs



- Outra aplicação com código e instruções
- Tarefa de previsão de sequências
- <https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/>

