

Vetores densos e um estudo de caso em redes sociais

Evelin Amorim

Quem eu sou?

- Graduação em Ciência da Computação na UFES(2002-2007)
- Mestrado em Mineração de Dados na Puc-Rio (2007-2009)
- Doutorado em NLP na UFMG (11/2013- Atualmente)
- Research Engineer na Kunumi (Atualmente)

REVISTA EXAME

A pequena Kunumi vai brigar contra o Google

Depois de lançar três startups de sucesso, um professor universitário cria a Kunumi para competir no promissor mercado de inteligência artificial

Por [Letícia Toledo](#)

© 23 out 2017, 19h11 - Publicado em 20 out 2017, 05h55

Organização

- **Introdução**
- **Parte I**
 - Classificação
- **Parte II**
 - Modelo Neural da Linguagem
- **Parte III**
 - Prática

Introdução

- Processamento de Linguagem Natural (PLN) antigamente
 - Modelos lineares: SVM, regressão, etc.
 - Vetores esparsos de características.
- PLN recentemente
 - Entrada são vetores densos
 - Redes neurais não lineares

Introdução

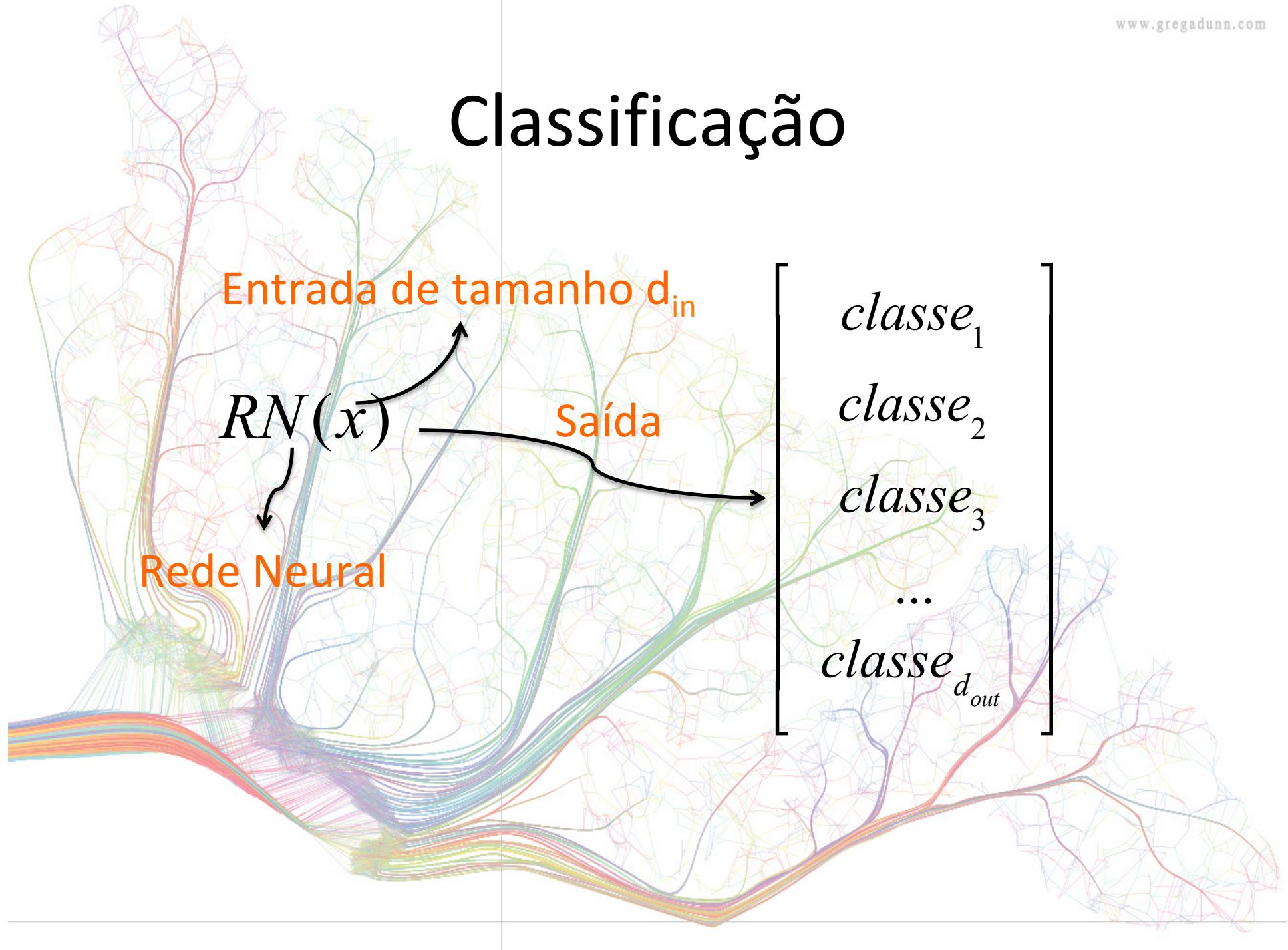
- Modelos de Linguagens
 - Como representar linguagem no computador
 - Vocabulário
 - Palavras desconhecidas do vocabulário

Introdução

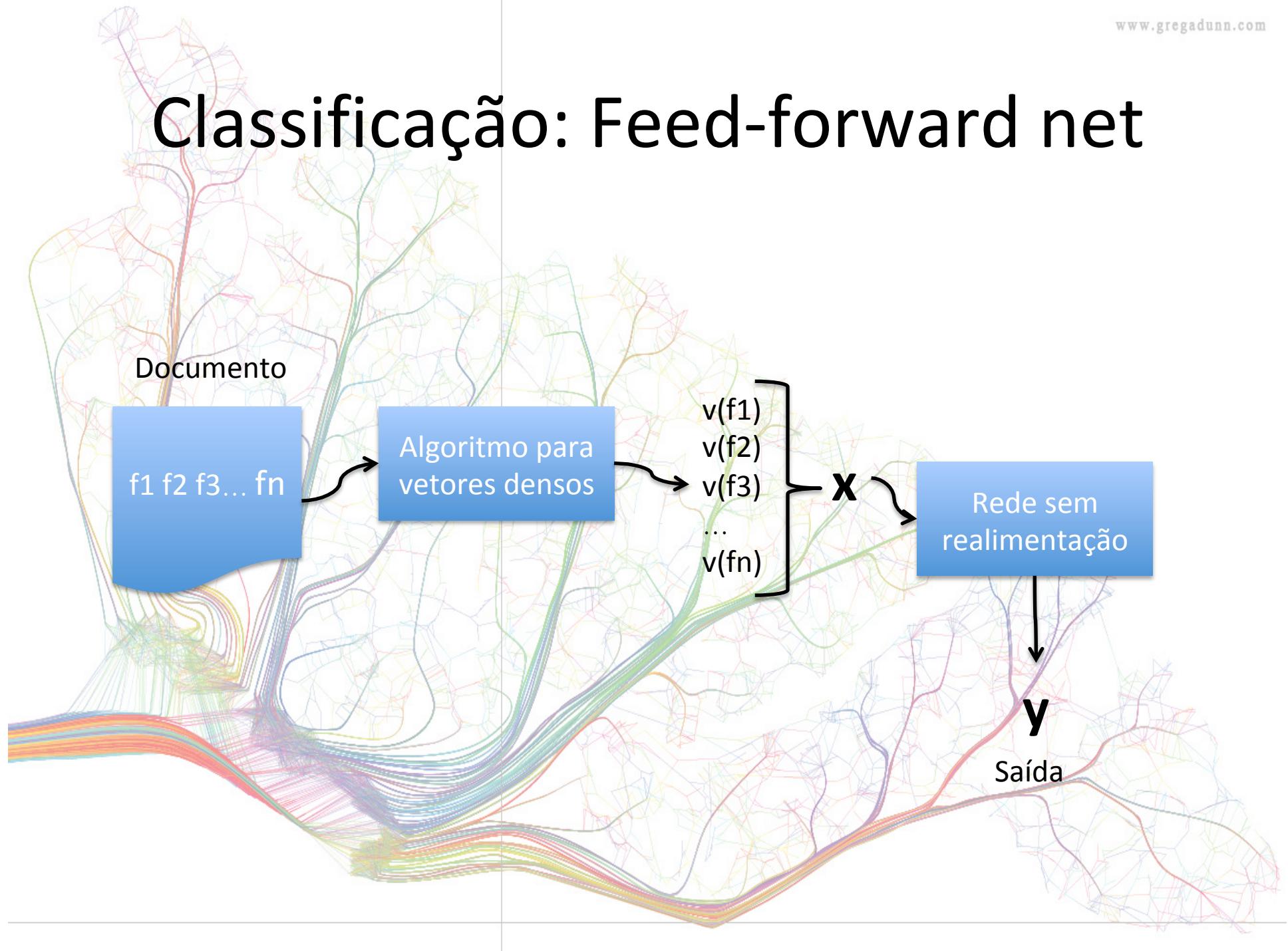
- Tipos de redes neurais
 - Classificação/Regressão
 - Redes sem realimentação (*feed-forward*)
 - Representação de dados
 - Características
 - Entrada para redes sem realimentação
 - Redes recursivas e recorrentes

PARTE I: CLASSIFICAÇÃO

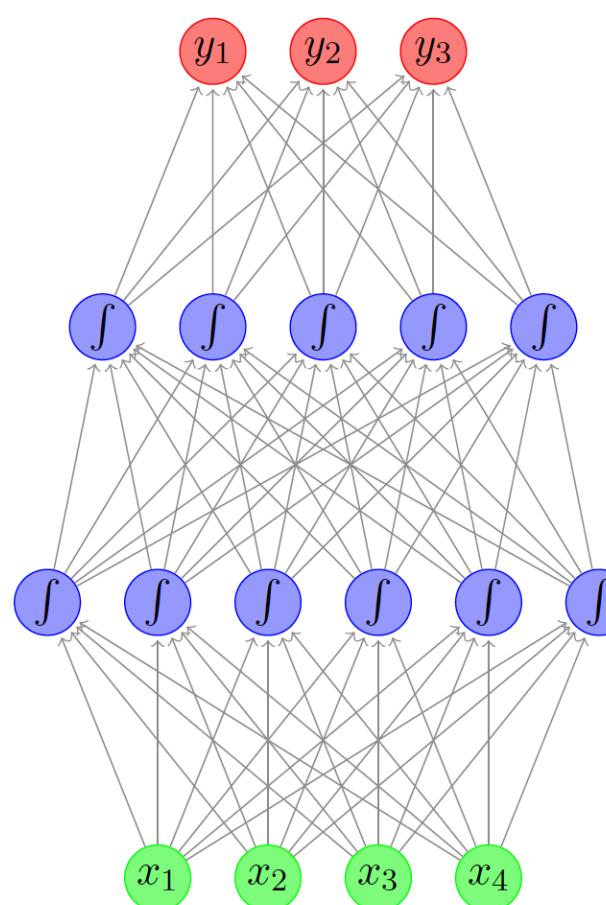
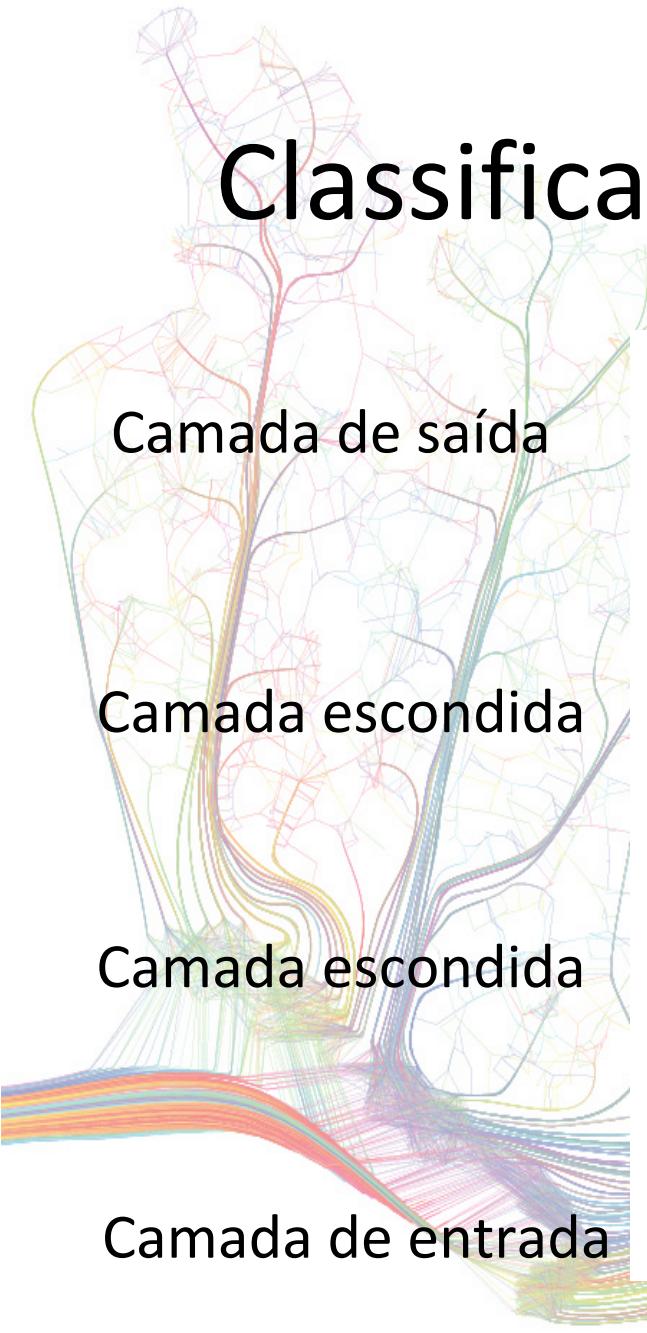
Classificação



Classificação: Feed-forward net



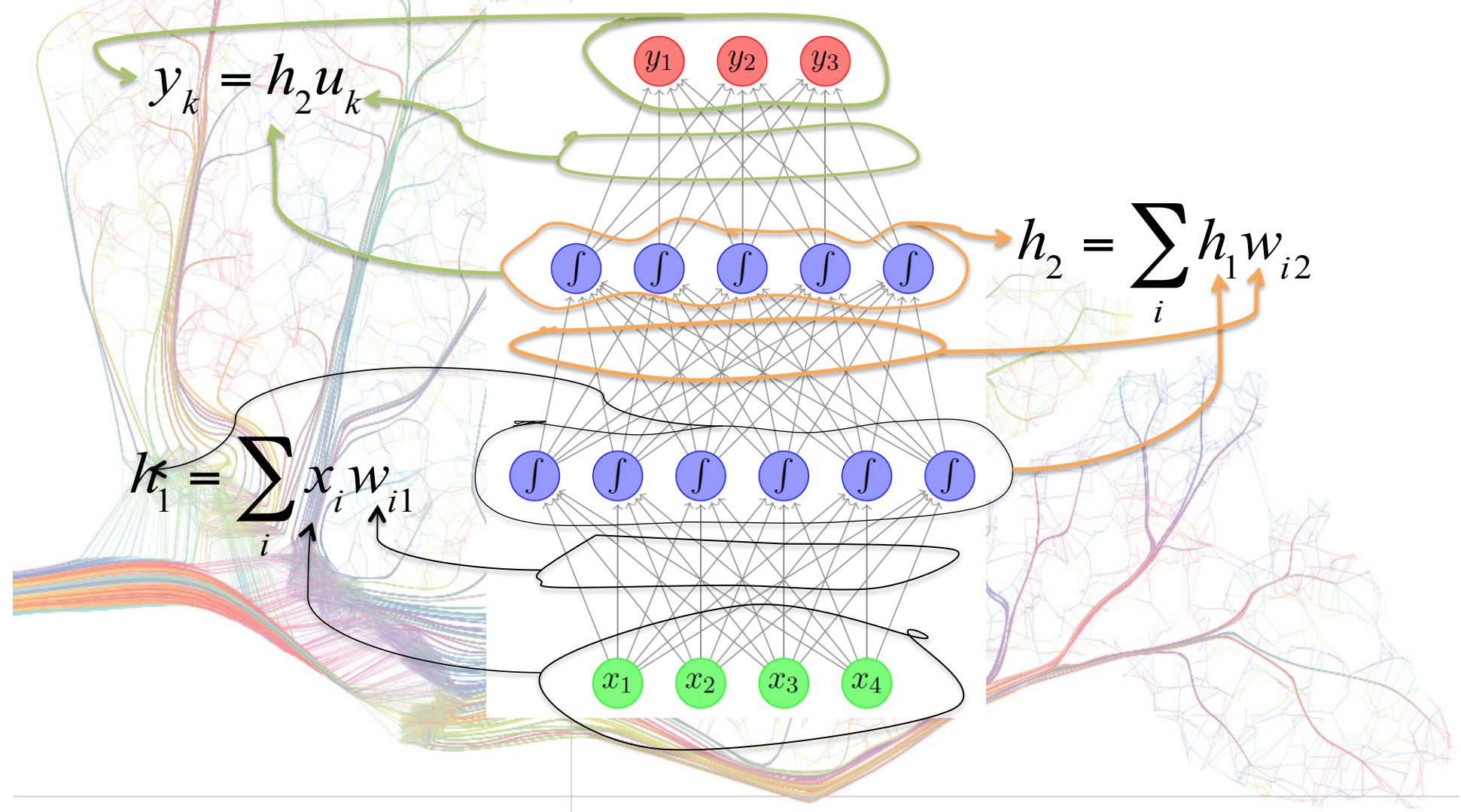
Classificação: Feed-forward net



(Goldberg, 2015)



Classificação: Feed-forward net



Classificação: Feed-forward net

- Função não linear de ativação
 - Tangente hiperbólica
 - $\tanh(x)$: saída de -1 a 1
 - Função logística
 - $\text{sigmoid}(x)$: saída de 0 a +1
 - Unidade Linear retificada
 - $\text{relu}(x)$: saída de 0 a ∞

$$y_k = \text{relu}(h_2 u_k)$$

Classificação: Feed-forward net

- E w_i como fica?
 - Poderia ser fixo?
 - Como os pesos são inicializados?
 - Podemos atualizar os pesos?
 - Gradiente descendente (contra): **minimiza o erro**
 - Backpropagation: **de trás para frente!**
 - Época: **Cada passagem nos dados**
 - Taxa de aprendizado (learning rate)

Classificação: Feed-forward net

- Código 1
- Código 2

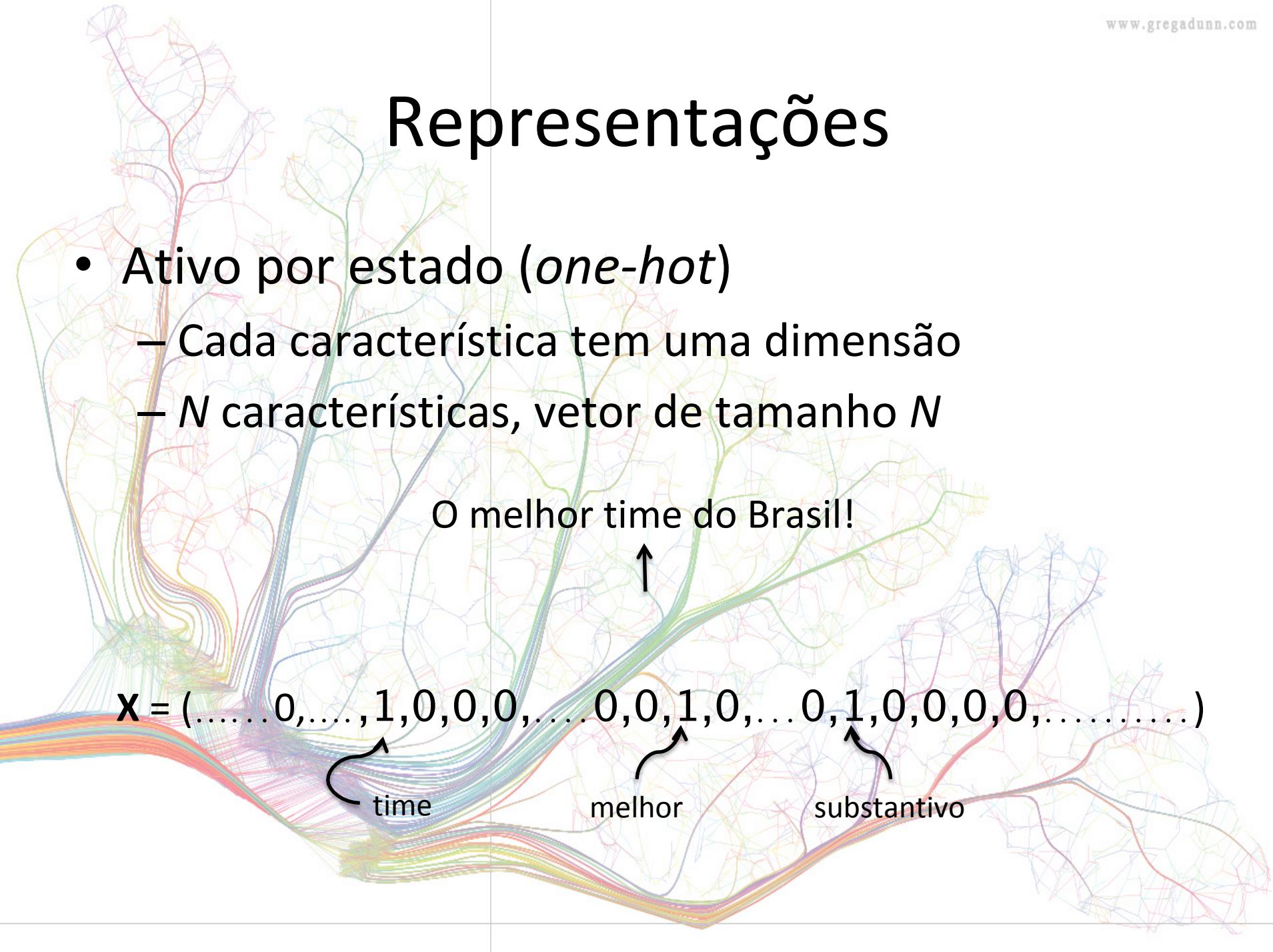
PARTE II: MODELO NEURAL DA LINGUAGEM

Representações

- Ativo por estado (*one-hot*)
 - Cada característica tem uma dimensão
 - N características, vetor de tamanho N

O melhor time do Brasil!

$X = (\dots, 0, \dots, 1, 0, 0, 0, \dots, 0, 0, 1, 0, \dots, 0, 1, 0, 0, 0, 0, \dots)$



Representações

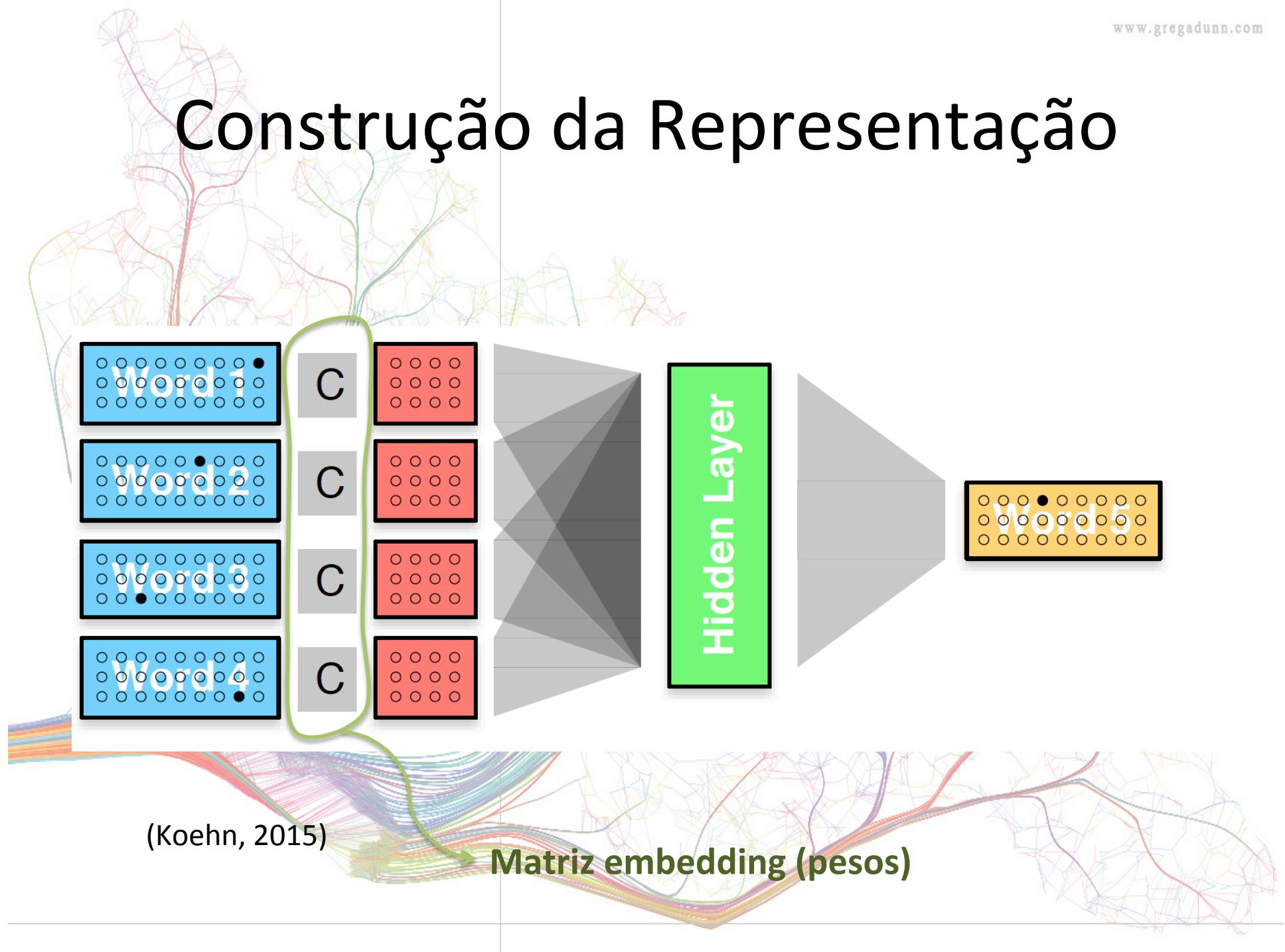
- Densa
 - Cada característica é representada por um vetor de tamanho d

O melhor time do Brasil!

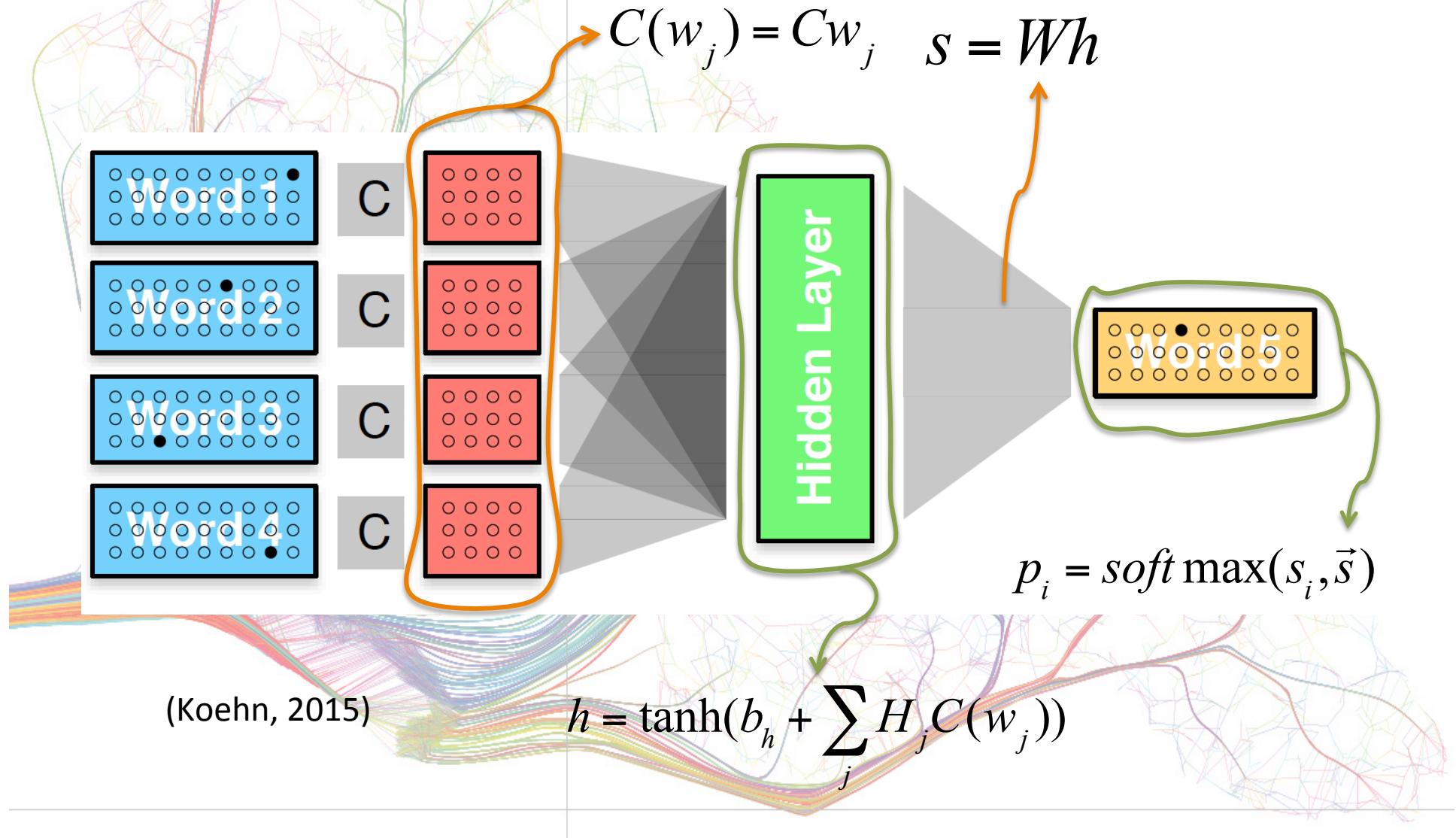
$[-0.0424, 0.00018, -0.01873, 0.07636, 0.04403]$

$[-0.00733, -0.00834, 0.04494, -0.06332, -0.03788]$

Construção da Representação



Construção da Representação



Representação Densa

- Como saber se temos uma representação boa?
- Qual será nossa função objetivo?

$$L(x, \vec{y}; W) = - \sum_k y_k \log p_k$$

Representação Densa

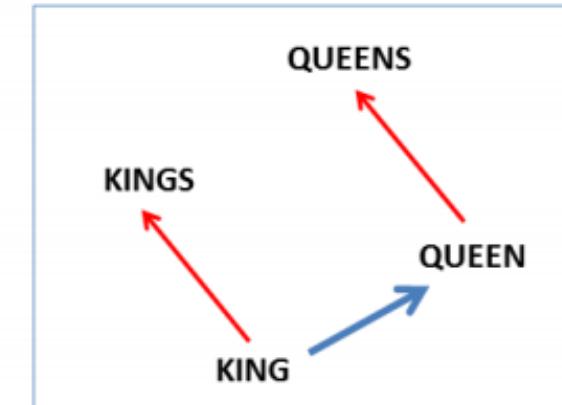
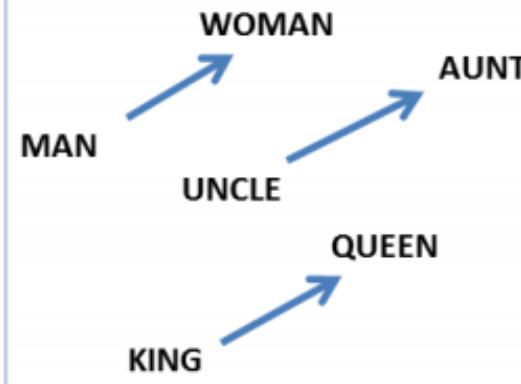
- Combinação de características
 - Soma ou média
 - Classificadores não lineares procuram pelos melhores pesos
 - Escalar com o tamanho da rede
 - Dados massivos

Representação Densa

- Dimensionalidade
 - Não existe limite teórico
 - Tamanho proporcional a quantidade de classes
 - Tamanho padrão em diversas aplicações: 300
 - O que significa uma dimensão < 300 ou > 300 ?

Representação densa

- O que podemos fazer só com a representação?

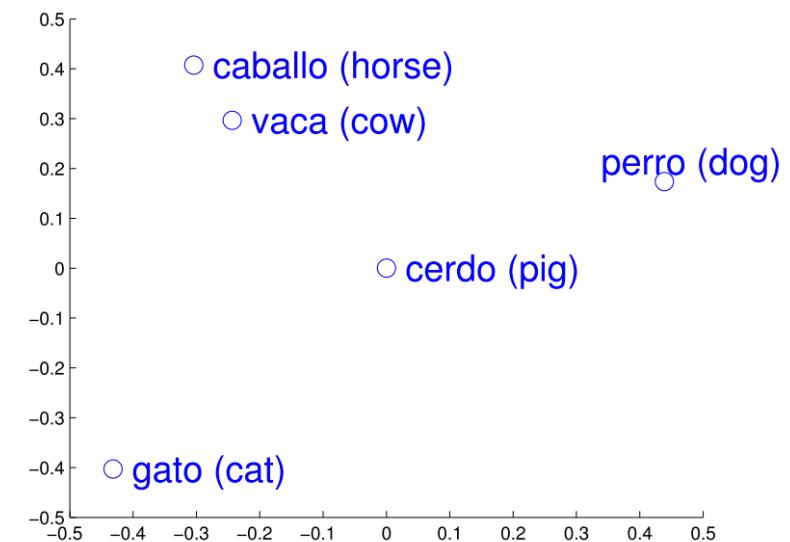


Representação densa

- Treinando seu modelo
 - Utilizar um arcabouço: [word2vec](#), Glove, etc.
 - Dados processados: text8, etc.
- Utilizando em um algoritmo de aprendizado estruturado
 - [Multi-layer perceptron](#)

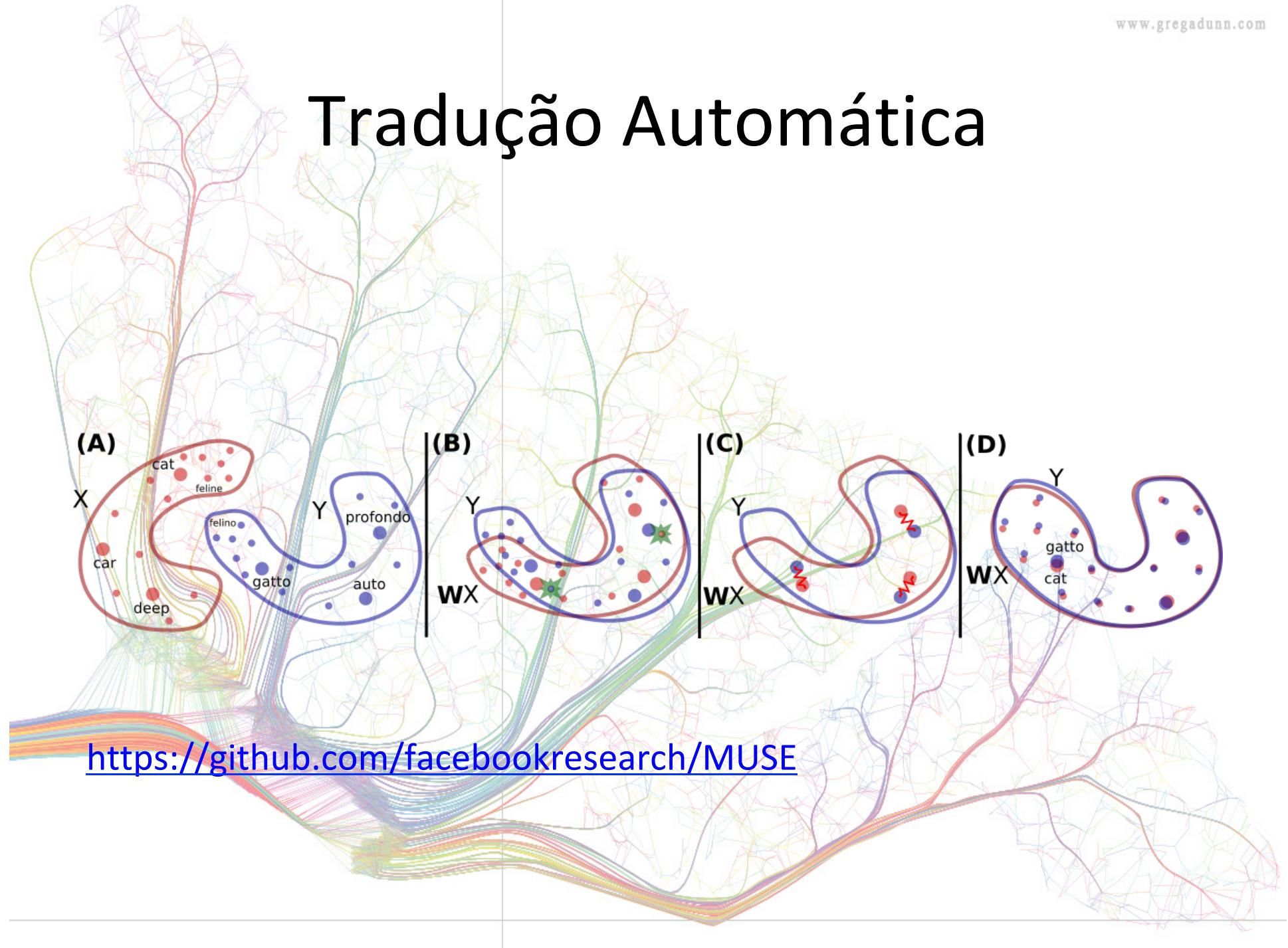
PARTE III: PRÁTICA

Tradução Automática



(Mikolov et. al, 2013)

Tradução Automática



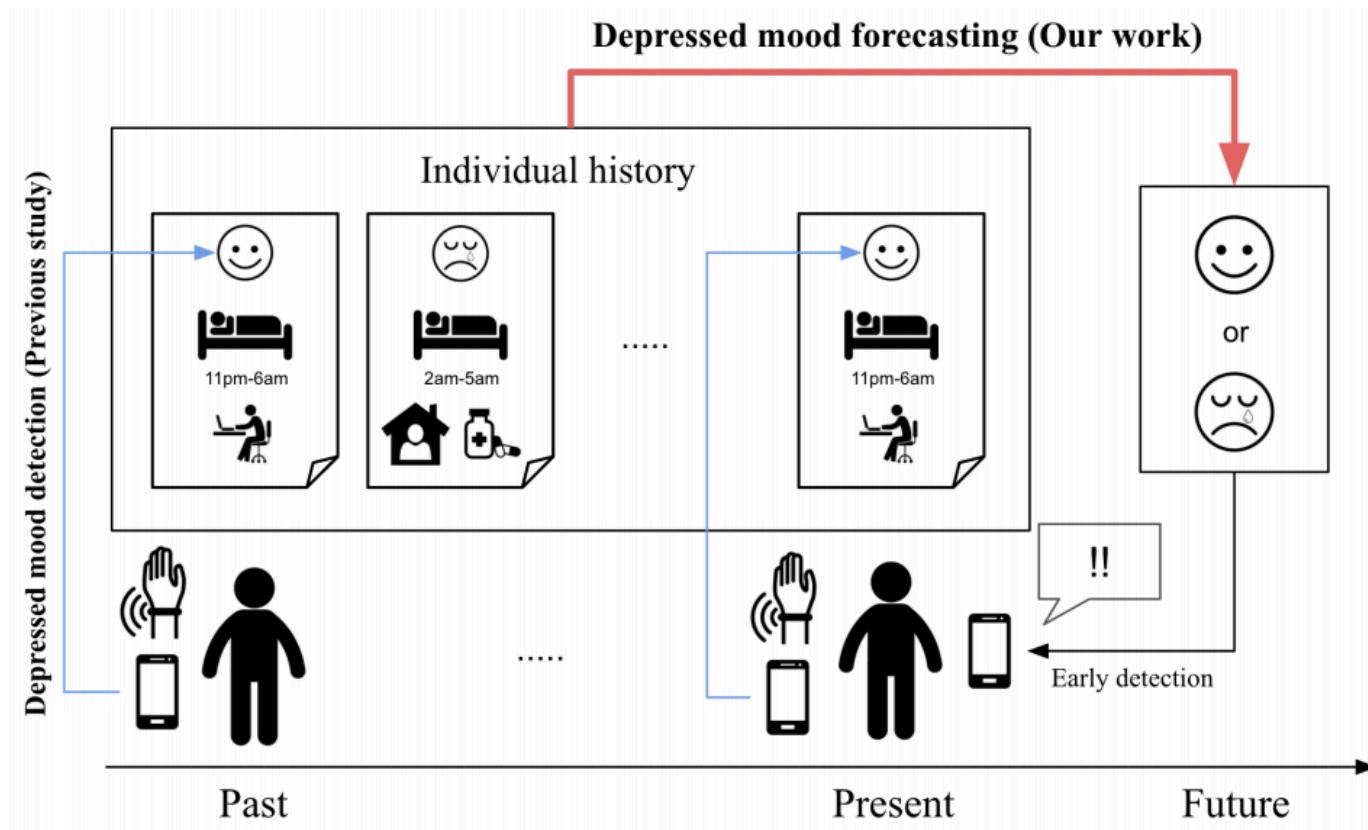
Análise de Sentimento

	Our model Sentiment + Semantic	Our model Semantic only	LSA
melancholy	bittersweet	thoughtful	poetic
	heartbreaking	warmth	lyrical
	happiness	layer	poetry
	tenderness	gentle	profound
	compassionate	loneliness	vivid
ghastly	embarrassingly	predators	hideous
	trite	hideous	inept
	laughably	tube	severely
	atrocious	baffled	grotesque
	appalling	smack	unsuspecting
lackluster	lame	passable	uninspired
	laughable	unconvincing	flat
	unimaginative	amateurish	bland
	uninspired	clichéd	forgettable
	awful	insipid	mediocre
romantic	romance	romance	romance
	love	charming	screwball
	sweet	delightful	grant
	beautiful	sweet	comedies
	relationship	chemistry	comedy

(Maas et. al, 2011)



Análise de Sentimento

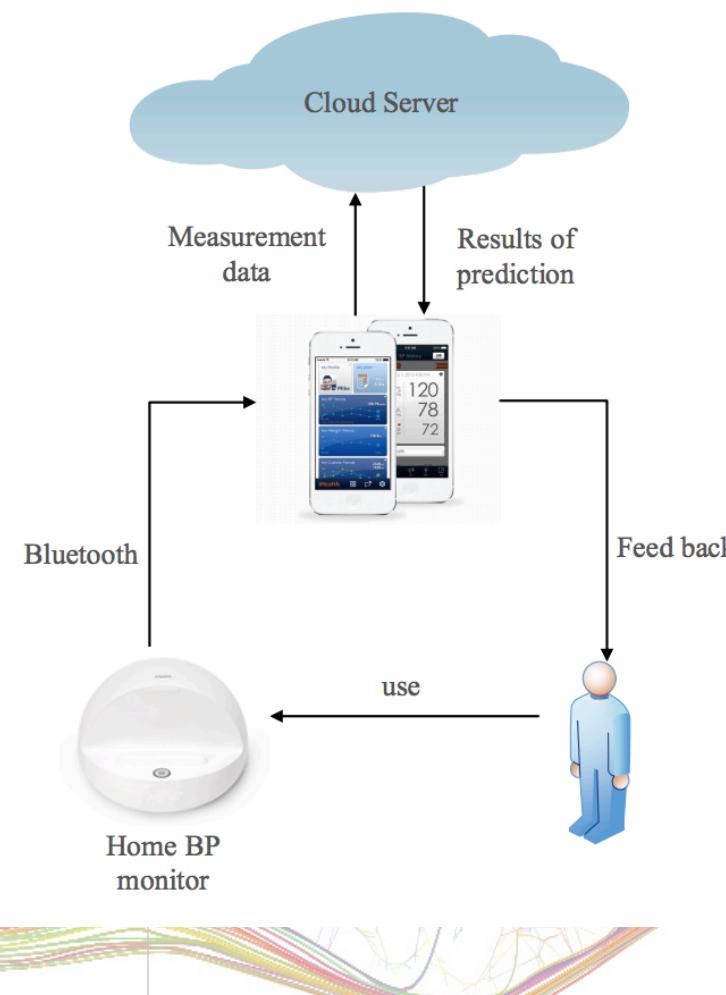


(Suhara et. al, 2017)

Computação para saúde



(Li et. Al, 2017)



Treinando um Modelo no word2vec

No diretório word2vec

- perl wikifil.pl tweets.txt > tweets_clean.txt
- time ./word2vec -train tweets_clean.txt -output vectors_tweets.bin -cbow 1 -size 200 -window 8 -negative 25 -hs 0 -sample 1e-4 -threads 20 -binary 1 -iter 15

Carregando nosso modelo

Instalar gensim (?)

Abra um editor de texto:

```
from gensim.models import KeyedVectors  
word_vectors = KeyedVectors.load_word2vec_format(  
    'vectors_tweets.bin', binary=True, unicode_errors='ign  
ore')  
print(word_vectors.most_similar(positive=['btw']))  
v = word_vectors['btw']
```

Lendo nossos dados

```
import pandas as pd  
import numpy as np  
  
df = pd.read_csv('grad_students_tweets.csv')  
y = np.zeros(df.shape[0])
```

Transformando em Embeddings

```
dim_vector = 200
X = []
for idx, r in df.iterrows():
    words = r['text'].lower().split()
    v = np.zeros(dim_vector)
    for w in words:
        v = np.add(v, word_vectors[w])
    X.append(v)
```

Plotando em um gráfico

- Scikit-learn

```
from sklearn.manifold import TSNE  
# We'll use matplotlib for graphics.  
import matplotlib.pyplot as plt  
import matplotlib.path_effects as PathEffects  
import matplotlib  
  
# We import seaborn to make nice plots.  
import seaborn as sns  
sns.set_style('darkgrid')  
sns.set_palette('muted')  
sns.set_context("notebook", font_scale=1.5,  
                rc={"lines.linewidth": 2.5},
```

Plotando em um gráfico

```
def scatter(x, colors):
    # We choose a color palette with seaborn.
    # palette = np.array(sns.color_palette("hls", 21))
    palette = np.array(sns.color_palette("hls", 1))

    # We create a scatter plot.
    f = plt.figure(figsize=(8, 8))
    ax = plt.subplot(aspect='equal')
    sc = ax.scatter(x[:,0], x[:,1], lw=0, s=40,
                    c=palette[colors.astype(np.int)])
    plt.xlim(-25, 25)
    plt.ylim(-25, 25)
    ax.axis('off')
    ax.axis('tight')

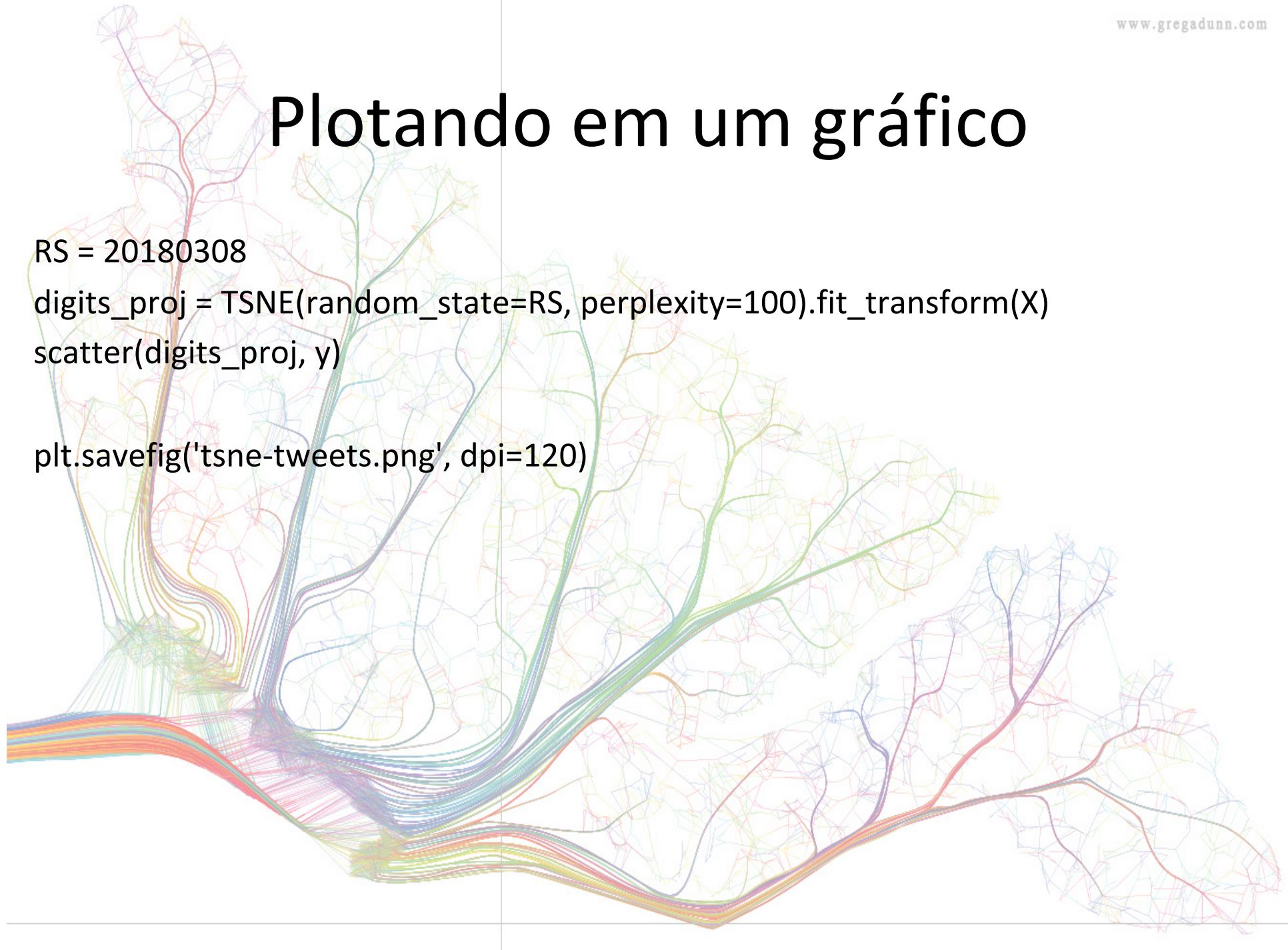
    # We add the labels for each digit.
    txts = []
    i = 0
    # while i <= 10:
    while i <= 3:
        # Position of each label.
        xtext, ytext = np.median(x[colors == i, :], axis=0)
        ax.spines['left'].adjust_location()
        i = i + 1
```

Plotando em um gráfico

RS = 20180308

```
digits_proj = TSNE(random_state=RS, perplexity=100).fit_transform(X)  
scatter(digits_proj, y)
```

```
plt.savefig('tsne-tweets.png', dpi=120)
```



Ideias????

- Quais ideias vocês podem fazer com o que você tem?

Referências

Goldberg, Yoav. "A Primer on Neural Network Models for Natural Language Processing." *J. Artif. Intell. Res.(JAIR)* 57 (2016): 345-420.

Koehn, Philipp. Chapter 13. Statistical machine translation. Cambridge University Press, 2015.

Li, Xiaohan, Shu Wu, and Liang Wang. "Blood Pressure Prediction via Recurrent Models with Contextual Layer." Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017.

Referências

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1, pp. 142-150. Association for Computational Linguistics, 2011.

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. "Exploiting similarities among languages for machine translation." arXiv preprint arXiv: 1309.4168 (2013).

Suhara, Yoshihiko, Yinzhan Xu, and Alex 'Sandy' Pentland. "Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks." Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017.

Contato