Dois modelos de aprendizagem profunda para análise morfossintática

Marcos Vinícius Treviso
marcosvtreviso@gmail.com

Orientador: Fabio Natanael Kepler

Trabalho de Conclusão de Curso II

3 de dezembro de 2015

Universidade Federal do Pampa

Roteiro

- Introdução
- Fundamentação
- Trabalhos relacionados
- Modelo neural recursivo
- Modelo neural recorrente bidirecional
- Testes e resultados
- Considerações finais

Roteiro

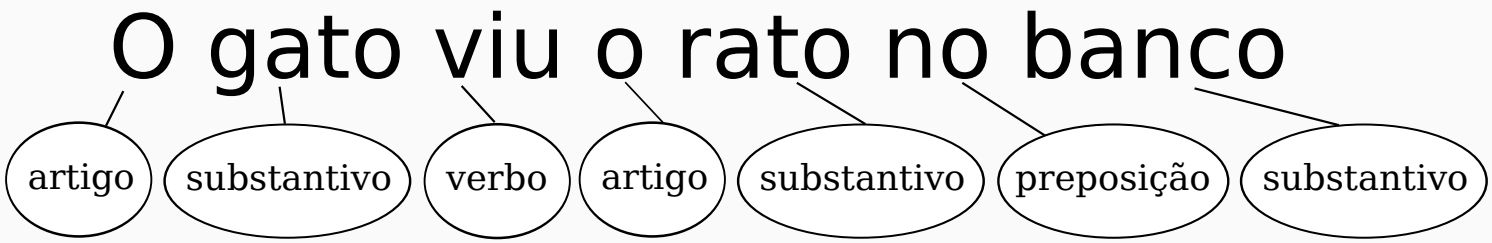
- Introdução
  - *Part-of-speech* (POS) Tagging
  - O problema
  - Objetivos
- Fundamentação
- Trabalhos relacionados
- Modelo neural recursivo
- Modelo neural recorrente bidirecional
- Testes e resultados
- Considerações finais

POS Tagging

É conhecido em Processamento de Linguagem Natural (PLN) como o ato de classificar uma palavra pertencente a um conjunto de textos em uma classe gramatical.

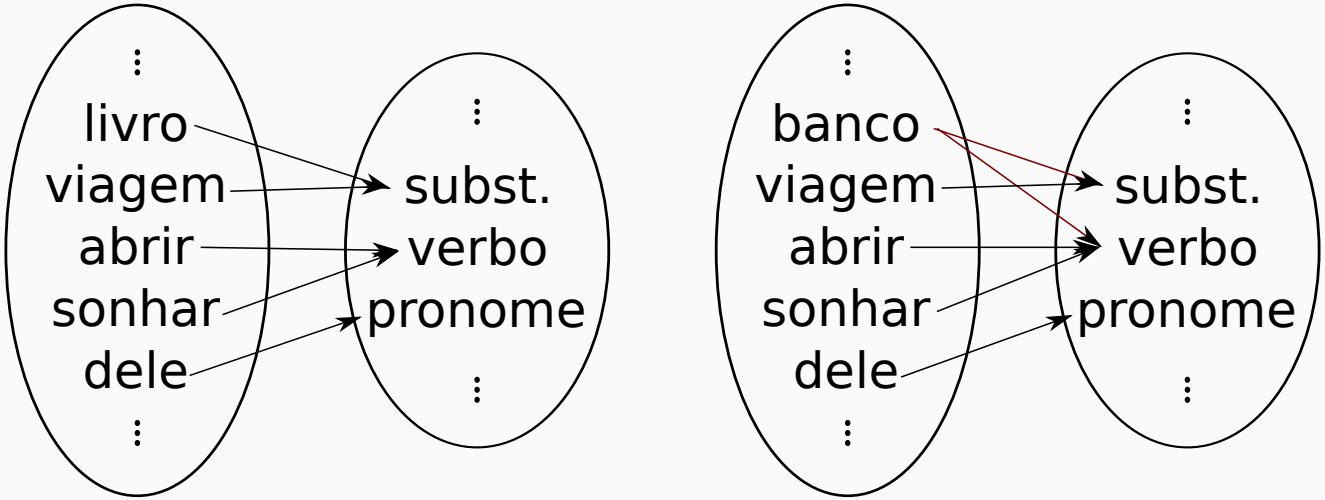
- Qual a medida de eficiência?
  - Acurácia
  - Atualmente cerca de 97%

- Quais são as aplicações?
  - Tradução automática
  - Sumarização



O problema

- Linguagens naturais são ambíguas
- Estratégia trivial não é eficaz
- Necessário analisar o contexto
- Aprendizado de máquina



Objetivos

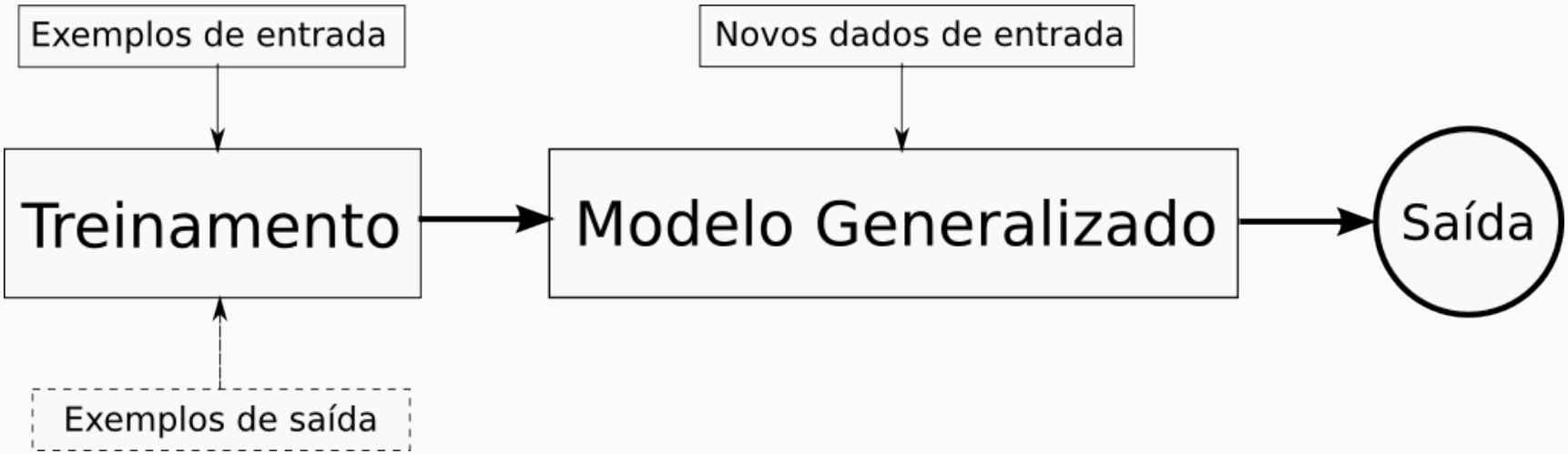
- Desenvolver novos modelos para POS Tagging
  - A princípio para o português brasileiro
- Alcançar estado da arte
  - Combinar abordagens existentes
- Analisar a acurácia
  - Palavras dentro do vocabulário
  - Palavras fora do vocabulário
  - Palavras ambíguas
  - Sentença

Roteiro

- Introdução
- Fundamentação
  - Aprendizado de máquina
  - Córpus
  - Representação de palavras
  - Redes neurais
  - Aprendizagem profunda
- Trabalhos relacionados
- Modelo neural recursivo
- Modelo neural recorrente bidirecional
- Testes e resultados
- Considerações finais

Aprendizado de máquina

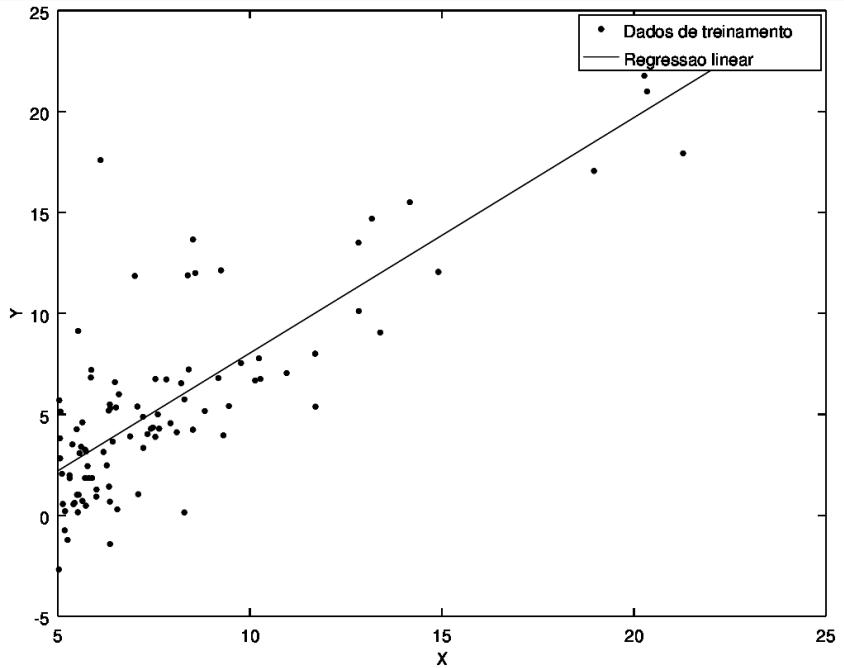
- Aprendizado supervisionado
  - Regressão
  - Classificação
- Aprendizado não supervisionado

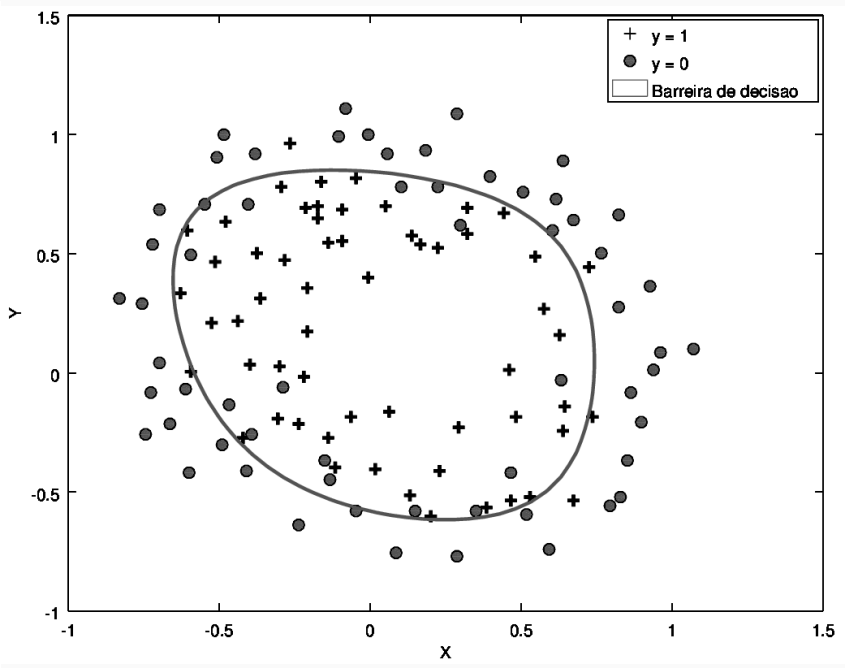


$$h_\theta(x) = \theta_0 + \theta_1 f(x_1) + \theta_2 f(x_2) + ... + \theta_n f(x_n)$$

Aprendizado de máquina

- Aprendizado supervisionado
  - **Regressão**
  - **Classificação**
- Aprendizado não supervisionado





Córpus

- Coleções de textos agrupados
- Anotação gramatical manual
- *Córpus* para o português brasileiro:

Córpus	**Sentenças**	**Palavras**	**Classes gramaticais**
Mac-Morpho original	53,374	1,221,465	41
Mac-Morpho revisado[1]	49,932	945,958	26
Tycho Brahe	55,932	1,541,654	265

- Por que não combiná-los?

1. Revisado por: Fonseca, Rosa e Aluísio (2015).

Representação de palavras

- Vetores reais valorados em um espaço multidimensional (*word embeddings*)
- Mais desempenho de aplicações em PLN e menos engenharia de *features*
- Conseguem capturar informações sintáticas e semânticas
- Geradas de maneiras diferentes dependendo da técnica utilizada
  - Word2Vec, Wang2Vec, GloVe, etc.
- Palavras fora do vocabulário de treinamento podem ter seu próprio vetor

Representação de palavras

- Palavras similares estão próximas

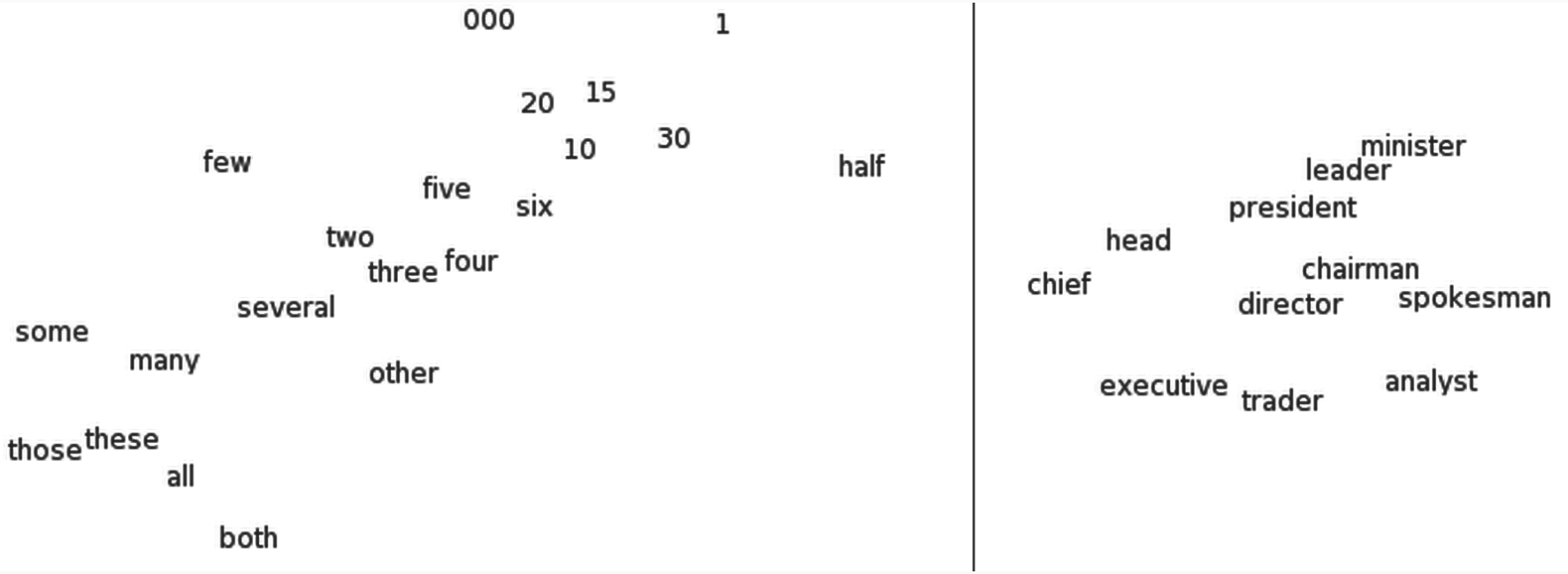
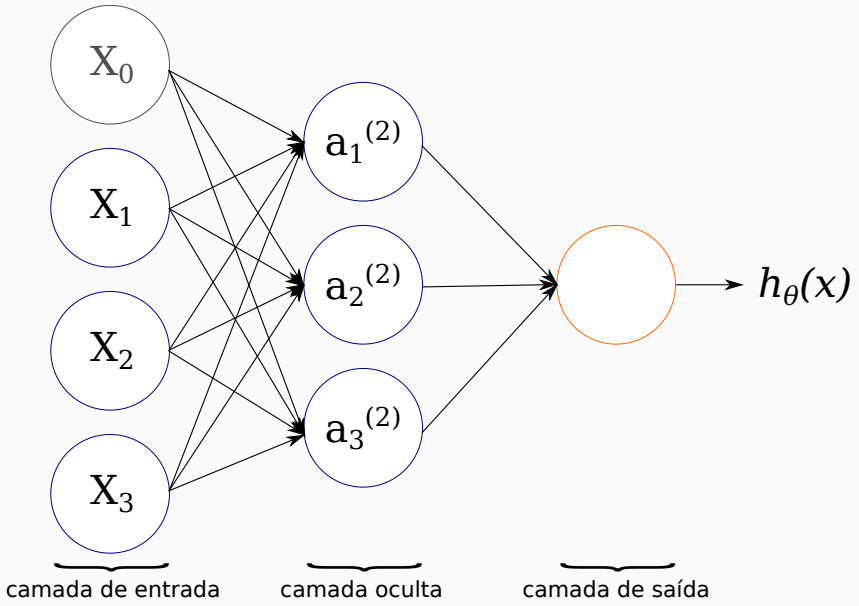


Imagem criada pelo t-SNE
Fonte: Turian, Ratinov e Bengio (2010)

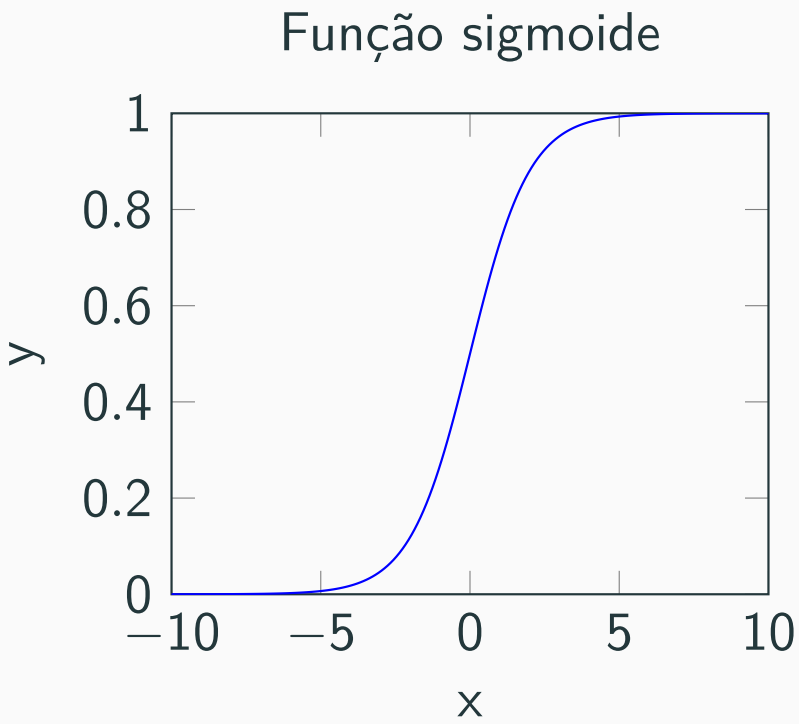
Redes neurais

- Simulação do cérebro humano
- Unidades de ativação: $a_i^{(j)}$
- Pesos: $\theta^{(j)}$
- Função de ativação: $g(\cdot)$
- $a^{(j+1)} = g(\theta^{(j)} a^{(j)})$
- *Forward propagation* e *Backpropagation*



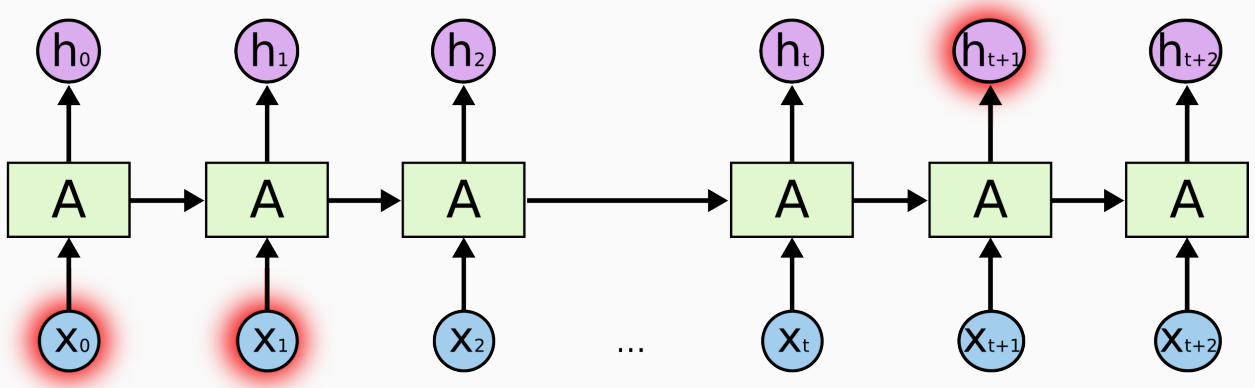
Redes neurais

- Simulação do cérebro humano
- Unidades de ativação: $a_i^{(j)}$
- Pesos: $\theta^{(j)}$
- Função de ativação: $g(\cdot)$
- $a^{(j+1)} = g(\theta^{(j)} a^{(j)})$
- *Forward propagation* e *Backpropagation*



Aprendizagem profunda

- Muitas tranformações não lineares
- Extração automática de *features*
- Redes neurais profundas
  - Redes neurais convolucionais
  - Redes neurais recursivas
  - Redes neurais recorrentes
    - *Long Short Term Memory* (LSTM)
    - *Gated Recurrent Unit* (GRU)
    - Bidirecional



Exemplo de rede neural recorrente com longa dependência
Fonte: Olah (2015)

Roteiro

- Introdução
- Fundamentação
- Trabalhos relacionados
- Modelo neural recursivo
- Modelo neural recorrente bidirecional
- Testes e resultados
- Considerações finais

Trabalhos relacionados

- Escopo do português brasileiro

Autores	**Modelo**	**Rep. palavras**	**Córpus**
Kepler e Finger (2010)	VLMC	Seq. de carac.	Tycho Brahe
Santos e Zadrozny (2014)	RNs	Vet. de pal. e carac.	Tycho Brahe; Mac-Morpho v1, v2
Fonseca, Rosa e Aluísio (2015)	RNs	Vet. de palavras	Tycho Brahe; Mac-Morpho v1, v2, v3
Este trabalho	RNs recur. e recor.	Vet. de palavras	Tycho Brahe; Mac-Morpho v1, v3

- Estado da arte

Córpus	**Autores**	**Acurácia**
Mac-Morpho v1 (original)	Fonseca, Rosa e Aluísio (2015)	97.57%
Mac-Morpho v3 (revisado)	Fonseca, Rosa e Aluísio (2015)	97.33%
Tycho Brahe	Santos e Zadrozny (2014)	97.17%

Roteiro

- Introdução
- Fundamentação
- Trabalhos relacionados
- Modelo neural recursivo
  - Pré-processamento
  - Arquitetura
  - Treinamento
  - Predição
- Modelo neural recorrente bidirecional
- Testes e resultados
- Considerações finais

Pré-processamento

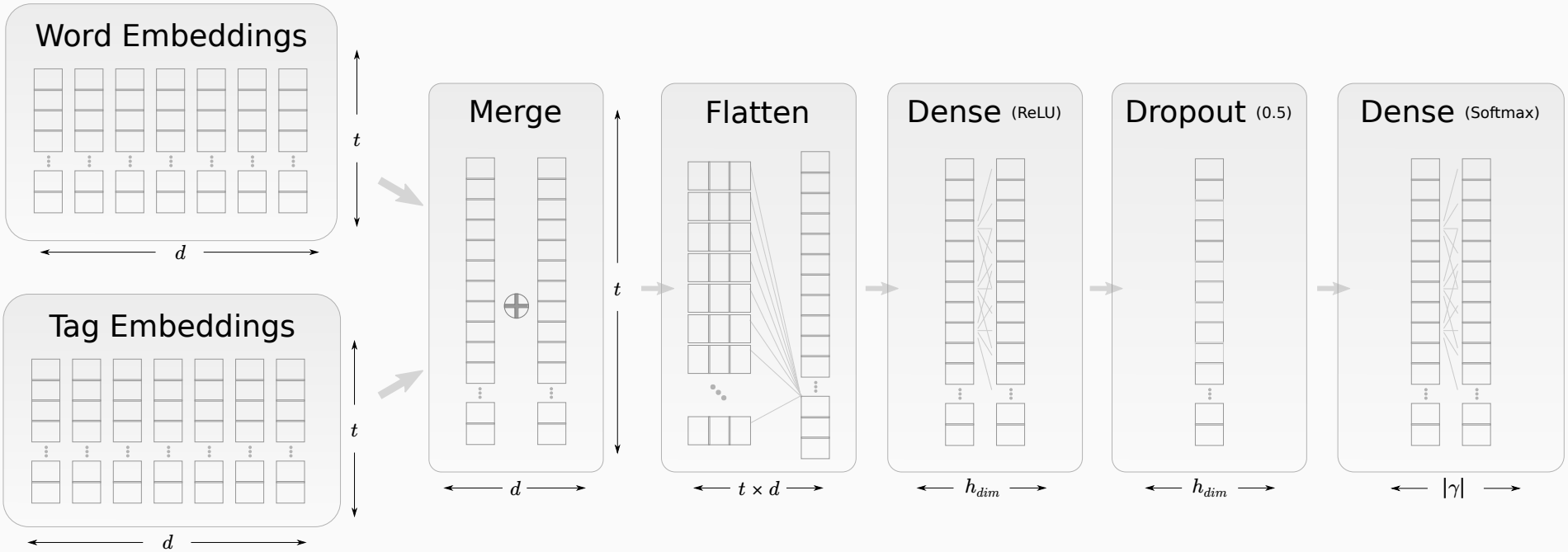
- Janela de palavras com tamanho t:

$$V_n = \left\{ \imath(w_{n-\lfloor t/2 \rfloor}), ..., \imath(w_n), ..., \imath(w_{n+\lfloor t/2 \rfloor}) \right\}$$

- Janela de etiquetas com o mesmo tamanho que a janela de palavras

- `<mask>`: Para as extremidades das janelas
- `<unknown>`: Para palavras raras ou desconhecidas
- `<padding_prefix>`: Para completar o vetor de prefixos
- `<padding_suffix>`: Para completar o vetor de sufixos

Arquitetura



$$ReLU(x) = max(0, x)$$

$$softmax(x) = \frac{e^{x_j}}{\sum_{k=1}^{n} e^{x_k}}, \quad \text{para } j = 1, ..., n$$

Treinamento

- Guiado por palavras mais fáceis (SHEN; SATTA; JOSHI, 2007)

Palavra/Etiqueta	substantivo	adjetivo	verbo
Computação	0.6	0.2	0.2
é	0.7	0.2	0.1
um	0.1	0.6	0.3
curso	**0.8**	0.1	0.1
legal	0.4	0.4	0.2
!	0.5	0.4	0.1

$$mp(M) = \arg\max_{i \notin Q} \left(\max_{0 \leq j < |\gamma|} (M_{i,j}) \right)$$

$$J_c[mp(M) - \lfloor t/2 \rfloor : mp(M) + \lfloor t/2 \rfloor][I_t] = c_{mp(M)}$$

Predição

- Semelhante ao algoritmo treinamento
- *Beam search* de tamanho B
- B sequências de etiquetas mais prováveis para a sentença
- Dá para resumir nas seguintes etapas:
  1. Inicialização (pré-processamento)
  2. Atualização da janela de etiquetas
  3. Predição na rede neural
  4. Atualização das B sentenças de acordo com a palavra mais provável
     - Levando em consideração a probabilidade da sentença e a probabilidade de emissão da etiqueta

Roteiro

- Introdução
- Fundamentação
- Trabalhos relacionados
- Modelo neural recursivo
- Modelo neural recorrente bidirecional
  - Pré-processamento
  - Arquitetura
  - Treinamento e Predição
- Testes e resultados
- Considerações finais

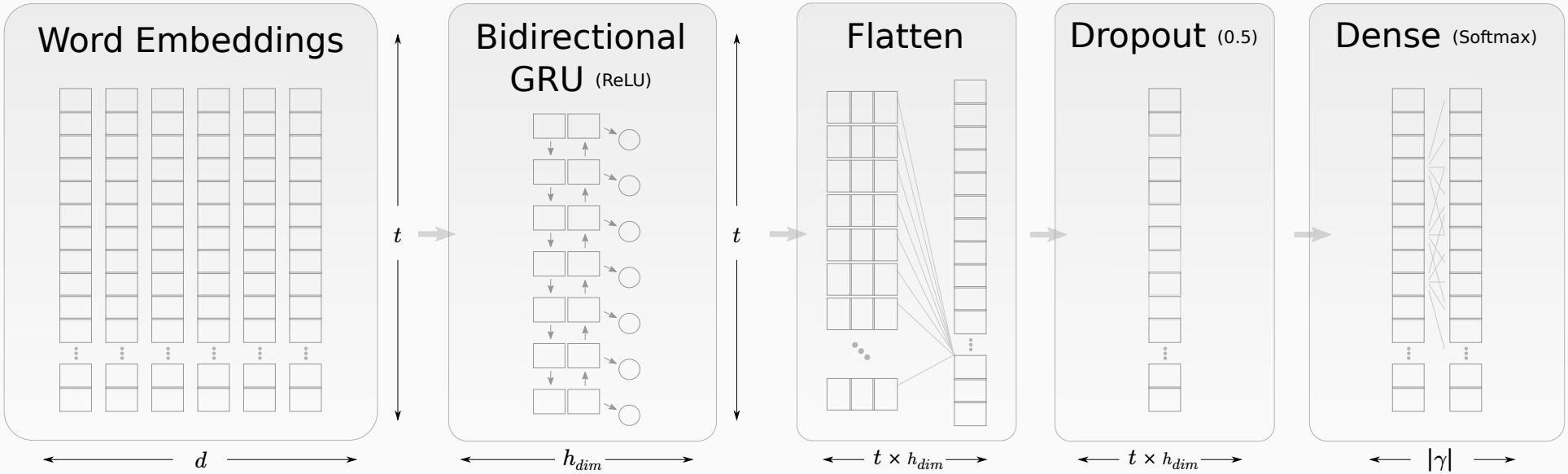
Pré-processamento

- Janela de palavras com tamanho t:

$$V_n = \left\{ \imath(w_{n-\lfloor t/2 \rfloor}), ..., \imath(w_n), ..., \imath(w_{n+\lfloor t/2 \rfloor}) \right\}$$

- `<mask>`: Para as extremidades das janelas
- `<unknown>`: Para palavras raras ou desconhecidas
- `<padding_prefix>`: Para completar o vetor de prefixos
- `<padding_suffix>`: Para completar o vetor de sufixos

Arquitetura



- GRU: Camada recorrente com memória
  - Cada palavra da janela vai possuir uma representação vetorial com tamanho h_{dim}

Treinamento e Predição

- **Treinamento:**
  - Minimização da função de custo
  - *Categorical Cross-entropy*:
    $J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)\right]$
  - Otimizador: Adadelta
- **Predição:**
  - Saída da rede neural: $c_{x_i} = \hat{y}_i$

Roteiro

- Introdução
- Fundamentação
- Trabalhos relacionados
- Modelo neural recursivo
- Modelo neural recorrente bidirecional
- Testes e resultados
  - Ambiente de teste
  - Pré-processamento
  - Hiperparâmetros
  - Resultados
  - Comparação com trabalhos relacionados
- Considerações finais

Ambiente de teste

- **Bibliotecas:**
  - Theano: Bib. que otimiza e avalia expressões matemáticas
  - Keras: Bib. para aprendizagem profunda
  - Numpy: Bib. para operações com matrizes
  - gensim: Bib. que possui um *parser* dos *dumps* da Wikipédia

- **Máquina:**
  - Sistema operacional: Ubuntu 14.04 LTS
  - Processador: Intel Xeon X5690 CPU @ 3.47GHz $\times$ 24
  - Memória: 64GB 1333 MHz DDR3
  - Python 3.4.3

Pré-processamento

- Transformação de palavras raras em `<unknown>`
- Transformação de todos os dígitos em “9”
- Utilização de *features* de prefixos, sufixos e capitalização
- Utilização de vetores distribuídos de palavras:
  - Treinamento não-supervisionado com um *dump* de artigos da Wikipédia:
    - 44 milhões de *tokens*
    - 618966 vetores
- Utilização de um vetor aleatório compartilhado para palavras fora do vocabulário

Hiperparâmetros

- A maioria foi escolhida de modo empírico
- Os hiperparâmetros não foram micro-ajustados

Hiperparâmetro	**Modelo recursivo**	**Modelo recorrente bidirecional**
Tamanho da janela de palavras	5	11
Tamanho da janela de etiquetas	5	-
Tamanho dos vetores de palavras	50	50
Tamanho dos vetores de capitalização	7	7
Tamanho dos vetores de prefixos	5	5
Tamanho dos vetores de sufixos	5	5
Número de unidades ocultas	250	250
Taxa de *Dropout*	0.5	0.5
Taxa de aprendizagem	1.0	1.0
Épocas de treinamento	3	20
Tamanho do *beam*	3	-
Taxa de raridade	5	5

Resultados

- Utilizamos três *córpus* nos experimentos:
  - Mac-Morpho original (versão 1)
  - Mac-Morpho revisado (versão 3)
  - Tycho Brahe

- Dividimos o conjunto de dados em:
  - 80% para treinamento
  - 20% para validação

- Utilizamos três tipos de vetores distribuídos:
  - Word2Vec
  - Wang2Vec
  - Fonseca (Treinados por Fonseca, Rosa e Aluísio (2015))

Resultados - Acurácia

$$\frac{1}{m}\sum_{i=1}^{m} equal(\hat{y}_i, y_i)$$

- Para palavras conhecidas
- Para palavras desconhecidas
- Para palavras ambíguas
- Para sentenças
  - Sentença é dita correta se todas as palavras na sentença foram classificadas corretamente

Resultados - Modelo neural recursivo

Acurácia sobre o Mac-Morpho original

Representação	Conhecidas	Desconhecidas	Ambíguas	Total	Sentenças
Word2Vec	94.80%	82.54%	94.24%	94.28%	46.39%
Wang2Vec	94.44%	82.12%	94.53%	93.91%	46.28%
Fonseca	**95.92%**	**86.77%**	**95.26%**	**95.53%**	**47.39%**

Acurácia sobre o Mac-Morpho revisado

Representação	Conhecidas	Desconhecidas	Ambíguas	Total	Sentenças
Word2Vec	94.13%	80.99%	93.02%	93.78%	42.14%
Wang2Vec	95.22%	81.57%	94.17%	94.56%	42.35%
Fonseca	**96.12%**	**88.32%**	**96.44%**	**95.79%**	**47.28%**

Acurácia sobre o Tycho Brahe

Representação	Conhecidas	Desconhecidas	Ambíguas	Total	Sentenças
Word2Vec	90.27%	48.12%	90.81%	88.82%	22.93%
Wang2Vec	**91.23%**	**48.77%**	**90.86%**	**89.78%**	**23.47%**
Fonseca	89.74%	47.75%	89.24%	88.31%	19.54%

Resultados - Modelo neural recorrente bidirecional

Acurácia sobre o Mac-Morpho original

Representação	Conhecidas	Desconhecidas	Ambíguas	Total	Sentenças
Word2Vec	97.01%	88.79%	97.05%	97.01%	59.43%
Wang2Vec	97.03%	87.60%	96.70%	96.63%	56.63%
Fonseca	**97.60%**	**92.63%**	**97.25%**	**97.37%**	**66.38%**

Acurácia sobre o Mac-Morpho revisado

Representação	Conhecidas	Desconhecidas	Ambíguas	Total	Sentenças
Word2Vec	96.92%	87.28%	96.30%	96.45%	**57.03%**
Wang2Vec	97.33%	90.03%	96.50%	96.99%	56.04%
Fonseca	**97.33%**	**92.18%**	**96.50%**	**97.08%**	56.77%

Acurácia sobre o Tycho Brahe

Representação	Conhecidas	Desconhecidas	Ambíguas	Total	Sentenças
Word2Vec	96.42%	65.45%	95.85%	95.53%	57.54%
Wang2Vec	96.33%	68.07%	95.37%	95.36%	58.91%
Fonseca	**96.80%**	**73.39%**	**96.09%**	**96.00%**	**64.80%**

Resultados - Análise

- Taxa de vetores não encontrados:

$$\frac{\texttt{número de vetores não encontrados}}{\texttt{número de palavras no vocabulário}}$$

Representação	Mac-Morpho original (%)	Mac-Morpho revisado (%)	Tycho Brahe (%)
Word2Vec	33.92	27.74	78.52
Wang2Vec	33.92	27.74	78.52
Fonseca	39.33	32.12	93.98

- Taxa de ocorrência de vetores não encontrados:

$$\frac{\texttt{ocorrência de vetores não encontrados}}{\texttt{número de palavras no conjunto de (treinamento} \cup \texttt{validação)}}$$

Representação	Mac-Morpho original (%)	Mac-Morpho revisado (%)	Tycho Brahe (%)
Word2Vec	21.01	17.11	18.42
Wang2Vec	21.01	17.11	18.42
Fonseca	2.14	1.72	6.22

Comparação com trabalhos relacionados

- Melhores resultados em cada *córpus*:

Córpus	**Mac-Morpho original**		**Mac-Morpho revisado**		**Tycho Brahe**	
	Todas(%)	**FDV(%)**	**Todas(%)**	**FDV(%)**	**Todas(%)**	**FDV(%)**
Kepler e Finger (2010)	-	-	-	-	96,29	71,60
Santos e Zadrozny (2014)	97.47	92.49	-	-	**97.17**	**86.58**
Fonseca, Rosa e Aluísio (2015)	**97.57**	**93.38**	**97.33**	**93.66**	96.93	84.14
Este trabalho	97.37	92.63	97.08	92.18	96.00	73.39

- *dump* da Wikipédia

- Modelo de caracteres

Roteiro

- Introdução
- Fundamentação
- Trabalhos relacionados
- Modelo neural recursivo
- Modelo neural recorrente bidirecional
- Testes e resultados
- Considerações finais
  - Conclusão
  - Trabalhos futuros

Conclusão

- Dois modelos para POS Tagging
  - Modelo neural recursivo
  - Modelo neural recorrente bidirecional

- Análise de acurácia
  - Para três *córpus* diferentes
  - Para três tipos de vetores distribuídos como representação de palavras

- Modelo neural recorrente bidirecional mostrou-se mais eficiente que o recursivo

- Segundo melhor resultado para palavras fora do vocabulário no Mac-Morpho original

- Criação de uma ferramenta chamada DeepTagger: <https://bitbucket.org/fabiokepler/deeptagger>

Trabalhos futuros

- Realizar mais testes
- Revisar o *dump* utilizado: aumentar caso necessário
- Melhorar o DeepTagger com novas técnicas:
  - Modelo de caracteres
  - Mecanismos de atenção
  - Suporte a outras representações distribuídas de vetores
- Paralelizar o código de treinamento e predição do modelo neural recursivo

Trabalhos futuros

- Com a inclusão de um modelo de caracteres e trocando GRU por LSTM:

Córpus	Mac-Morpho original		Mac-Morpho revisado		Tycho Brahe	
	Todas(%)	**FDV(%)**	**Todas(%)**	**FDV(%)**	**Todas(%)**	**FDV(%)**
Kepler e Finger (2010)	-	-	-	-	96,29	71,60
Santos e Zadrozny (2014)	97.47	92.49	-	-	97.17	**86.58**
Fonseca, Rosa e Aluísio (2015)	**97.57**	**93.38**	**97.33**	**93.66**	96.93	84.14
Antes	97.37	92.63	97.08	92.18	96.00	73.39
Depois	97.56	93.08	97.29	92.84	**97.27**	84.06

- Usando os vetores treinados por Fonseca, Rosa e Aluísio (2015)

Referências I

BENGIO, Y.; GOODFELLOW, I. J.; COURVILLE, A. Deep learning. Book in preparation for MIT Press. 2015. Disponível em: <http://www.iro.umontreal.ca/~bengioy/dlbook>.

FONSECA, E. R.; ROSA, J. L. G.; ALUÍSIO, S. M. Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. *Journal of the Brazilian Computer Society*, Springer, v. 21, n. 1, p. 1–14, 2015.

KEPLER, F. N.; FINGER, M. Variable-length markov models and ambiguous words in portuguese. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*. [S.l.], 2010. p. 15–23.

OLAH, C. Understanding lstm networks. 2015. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

Referências II

SANTOS, C. N. dos; ZADROZNY, B. Training state-of-the-art portuguese pos taggers without handcrafted features. In: *Computational Processing of the Portuguese Language*. [S.I.]: Springer, 2014. p. 82–93.

SHEN, L.; SATTA, G.; JOSHI, A. Guided learning for bidirectional sequence classification. In: CITESEER. *ACL*. [S.I.], 2007. v. 7, p. 760–767.

TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: a simple and general method for semi-supervised learning. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 48th annual meeting of the association for computational linguistics*. [S.I.], 2010. p. 384–394.

Dois modelos de aprendizagem profunda para análise morfossintática

Marcos Vinícius Treviso

marcosvtreviso@gmail.com

Orientador: Fabio Natanael Kepler

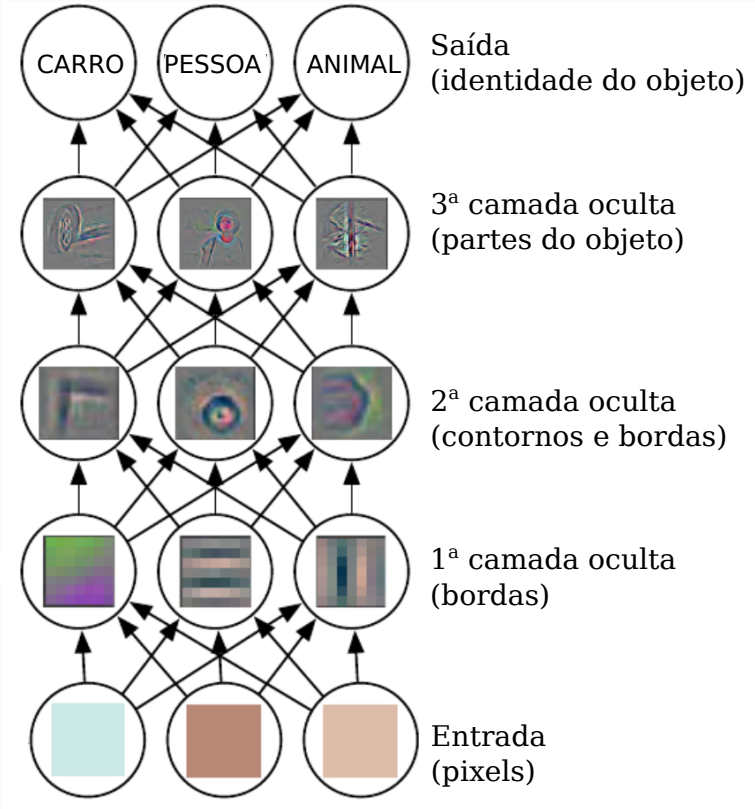
Trabalho de Conclusão de Curso II

3 de dezembro de 2015

Universidade Federal do Pampa

Aprendizagem profunda

- Muitas tranformações não lineares
- Objetivo de aprender automaticamente boas *features*
- Crescimento do desempenho computacional e criação de novos algoritmos
- Redes neurais com múltiplas camadas



Adaptado de: Bengio, Goodfellow e Courville (2015)

Redes neurais recursivas

- Grafo computacional parece como uma árvore
- Aplica-se transformações recursivamente
- Composição da saída com entrada