

# IA025A

# Introdução ao Aprendizado Profundo

# 1o. semestre 2022

Professores: Roberto Lotufo e Rodrigo Nogueira  
Monitor: Luiz Henrique Bonifácio

# Assunto da primeira aula

- Questionário expectativa
- Introdução: Regras
- Motivação Deep Learning (apresentação)
- Exercícios Próxima Aula:
  - Notebook: usados na seleção de alunos especiais
  - Leitura do artigo: Lecun et al. - [Nature Link](#)
- Numpy: Operações matriciais e imagens
  - broadcast, redução de eixo

wifi: Rodrigo's Iphone

Senha: mdk1462@

# Expectativa

# Conteúdo do curso

- Backpropagation
- Funções de perda
- Otimizadores
- Programação matricial eficiente em GPUs/TPUs
- Redes convolucionais
- Mecanismo de atenção
- Transformers
- Pré-treinamento não supervisionado
- Aumento de dados
- Métricas de desempenho

# Avaliação

5% - Testes semanais

15% - Leitura de artigos

40% - Programas semanais feitos em PyTorch no Jupyter/Colab

40% - Projeto Final (4 semanas)

[Código de conduta](#)

# Ferramentas de Apoio

- Google Classroom
- Peergrade: resumo de leitura de artigos
- Google Colab - Jupyter notebooks (with GPU)
- Github
- Google Meet - gravação da aula

# Motivação (30-60 minutos)

*\*Experimento: cada um tem que fazer pelo menos uma pergunta*

[Share on Facebook](#) [Tweet this article](#)

## Google Scholar reveals its most influential papers for 2020

Artificial intelligence papers amass citations more than any other research topic.

13 July 2020

Bec Crew

Of the 100 top-ranked journals in 2020, six are AI conference publications. Their papers tend to amass citations much faster than papers in influential journals such as *The New England Journal of Medicine*, *Nature*, and *Science*.

Such rapid accumulation of citations may be in part explained by the fact that at these annual conferences that can attract thousands of attendees from around the world, new software, which is often open source, is shared and later built upon by the community.

## AlphaGo, 2017



VCG / Contributor / Getty

Chinese Go player Ke Jie (L) attends a press conference after his second match against Google's artificial intelligence programme AlphaGo on day two of Future of Go Summit in Wuzhen on May 25, 2017 in Jiangxi, Zhejiang Province of China.

[https://www.youtube.com/watch?v=8tq1C8spV\\_g](https://www.youtube.com/watch?v=8tq1C8spV_g)

# H-index dos artigos dos últimos 5 anos

link: [Google Scholar - Top publications](#)

Categories ▾

English ▾

	Publication	<u>h5-index</u>	<u>h5-median</u>
1.	Nature	<u>414</u>	607
2.	The New England Journal of Medicine	<u>410</u>	704
3.	Science	<u>391</u>	564
4.	IEEE/CVF Conference on Computer Vision and Pattern Recognition	<u>356</u>	583
5.	The Lancet	<u>345</u>	600

Nature

h5-index:376 h5-median:552  
#1 Life Sciences & Earth Sciences  
#1 Life Sciences & Earth Sciences (general)

Title / Author Cited by

Deep learning.  
Y LeCun, Y Bengio, G Hinton  
Nature 521 (7553), 436 27375

Human-level control through deep reinforcement learning.  
V Mnih, K Kavukcuoglu, D Silver, AA Rusu, J Veness, MG Bellemare, ...  
Nature 518 (7540), 529-533 10394

Mastering the game of Go with deep neural networks and tree search.  
D Silver, A Huang, CJ Maddison, A Guez, L Sifre, G van den Driessche, ...  
Nature 529 (7587), 484 7698

Analysis of protein-coding genetic variation in 60,706 humans.  
M Lek, KJ Karczewski, EV Minikel, KE Samocha, E Banks, T Fennell, ...  
Nature 536 (7616), 285-291 6387

IEEE/CVF Conference on Computer Vision and Pattern ... Q

h5-index:299 h5-median:509  
#1 Engineering & Computer Science  
#1 Computer Vision & Pattern Recognition

Title / Author Cited by

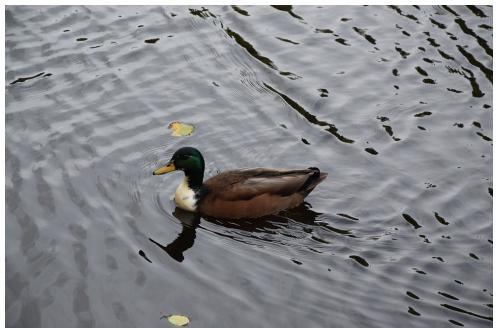
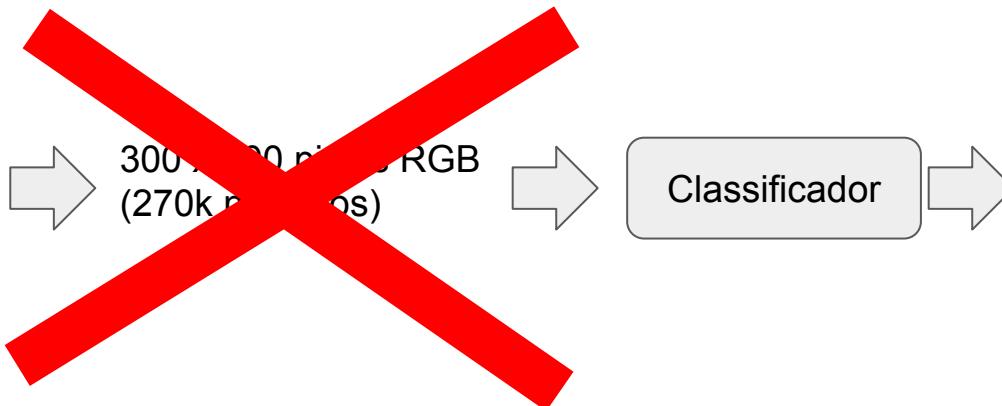
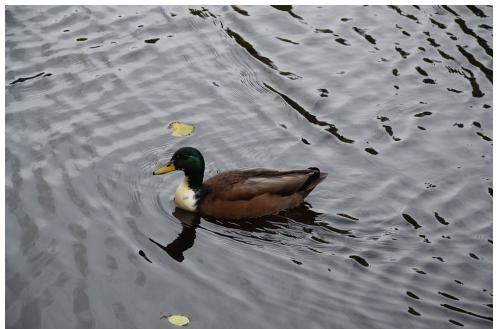
Deep Residual Learning for Image Recognition  
K He, X Zhang, S Ren, J Sun  
Proceedings of the IEEE Conference on Computer Vision and Pattern ... 49301

Going Deeper With Convolutions  
C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, ...  
Proceedings of the IEEE Conference on Computer Vision and Pattern ... 22434

Fully Convolutional Networks for Semantic Segmentation  
J Long, E Shelhamer, T Darrell  
Proceedings of the IEEE Conference on Computer Vision and Pattern ... 16664

You Only Look Once: Unified, Real-Time Object Detection  
J Redmon, S Divvala, R Girshick, A Farhadi  
Proceedings of the IEEE Conference on Computer Vision and Pattern ... 9772

# Visão Computacional Clássica



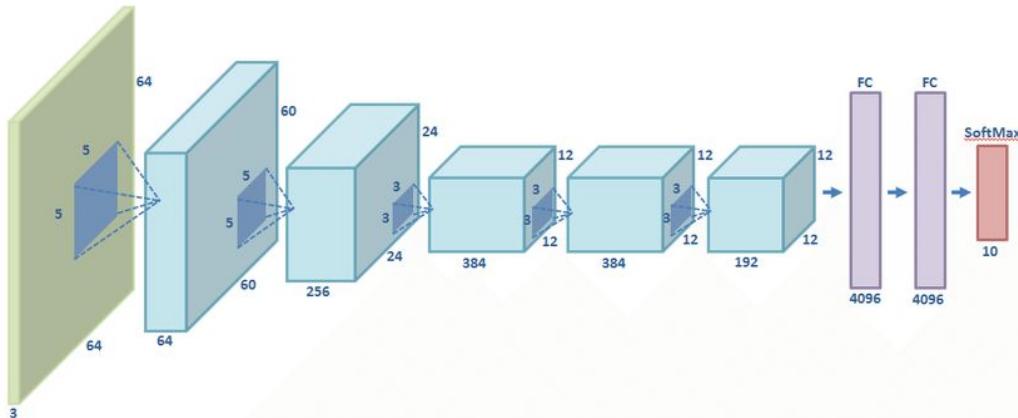
Carro: 0.0
Gato: 0.001
Pato: 0.99
...

Carro: 0.0
Folha: 0.001
Pato: 0.99
...

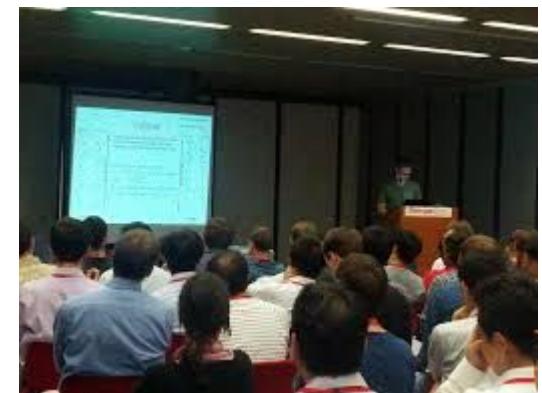
# Em 2012

Competição ImageNet 2012:

- Segundo colocado (sistema clássico): 25% de erro
- Primeiro colocado (Rede neural convolucional): 16% de erro

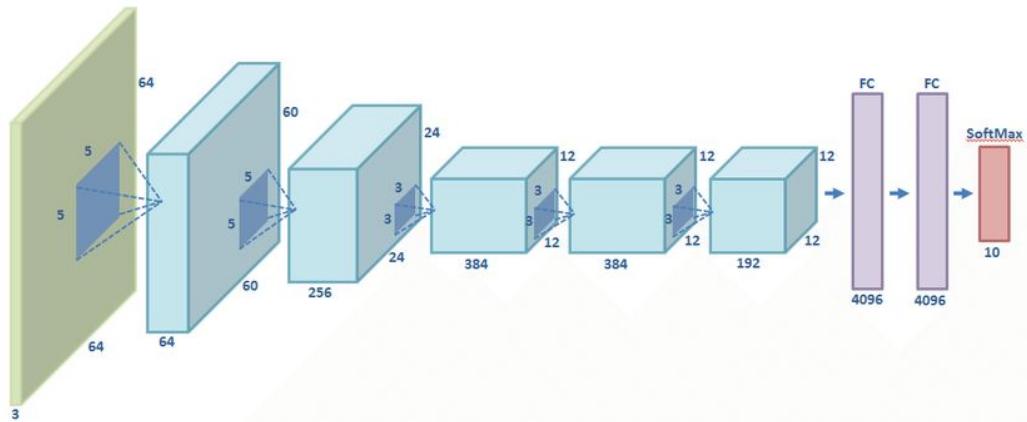
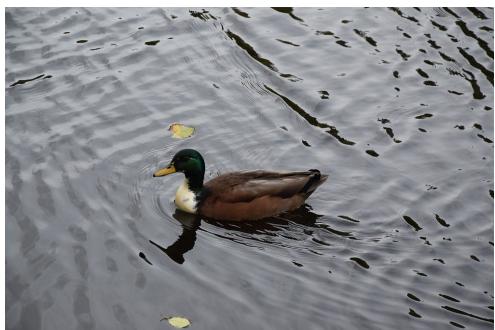


Alexnet



# Redes Neurais Convolucionais

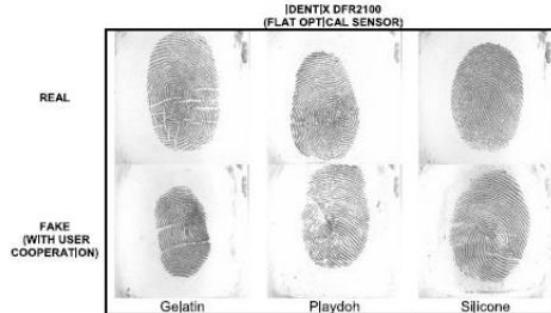
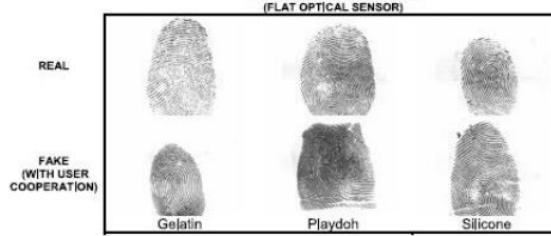
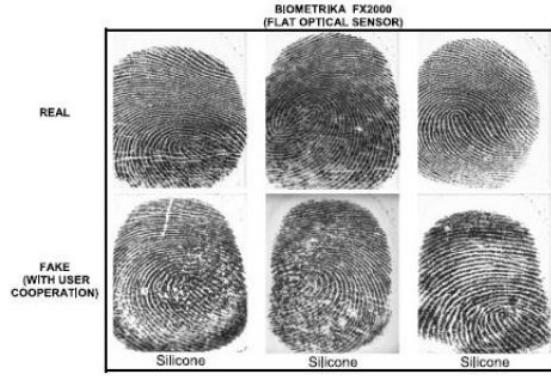
- Aprende automaticamente vários extratores de features



Carro: 0.0
Folha: 0.001
Pato: 0.99
...

# Em 2013

- Comecei mestrado com o prof. Roberto Lotufo
- Tarefa: detectar se imagem de impressão digital é verdadeira ou falsa



# Em 2013-2014

- Começamos a usar redes neurais;
- Não conseguimos usar GPU nos principais frameworks: Torch, Theano, Cuda AlexNET;
- Codificamos redes convolucionais em numpy;
- Pesos aleatórios: treinamento é muito lento em CPU;
- Cluster de CPUs na Amazon para encontrar a melhor arquitetura de rede: rápido pois não precisa de treinamento;
- Após 6 meses: 1-2% melhor que o método clássico (LBP+SVM);
- Final de 2014: Surgiu o *Caffe* (conseguimos instalá-lo em *apenas* 30 horas);
- CNN era 3-4% melhor que o melhor método clássico;
- Ganhamos a competição do LivDet 2015 com 2 pontos acima do segundo colocado;
- 2018-Presente: licenciamos o software para três clientes;

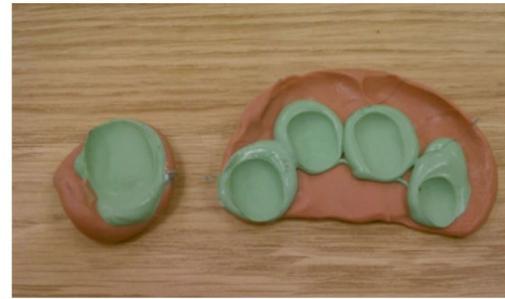


Fig. 1. Negative impression of five fingers using consensual method.

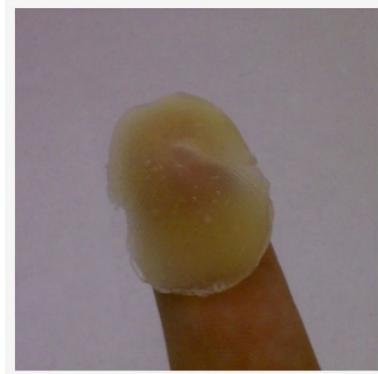


Fig. 2. Latex spoof on finger.

Table 10

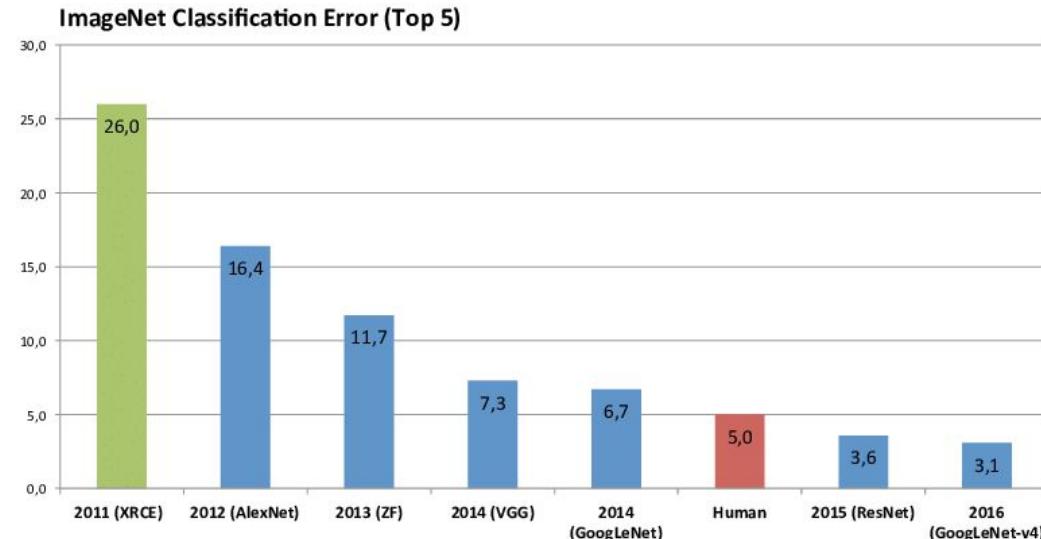
FerrFake values of the best algorithms calculated when *FerrLive* = 1% for the Cross-match dataset.

	All materials	Known materials	Unknown materials
unina	7.42	2.47	14.49
nogueira	2.66	1.94	3.69
anonym	18.61	10.75	29.82
Average	9.56	5.05	16.00
Std. dev.	8.19	4.94	13.13

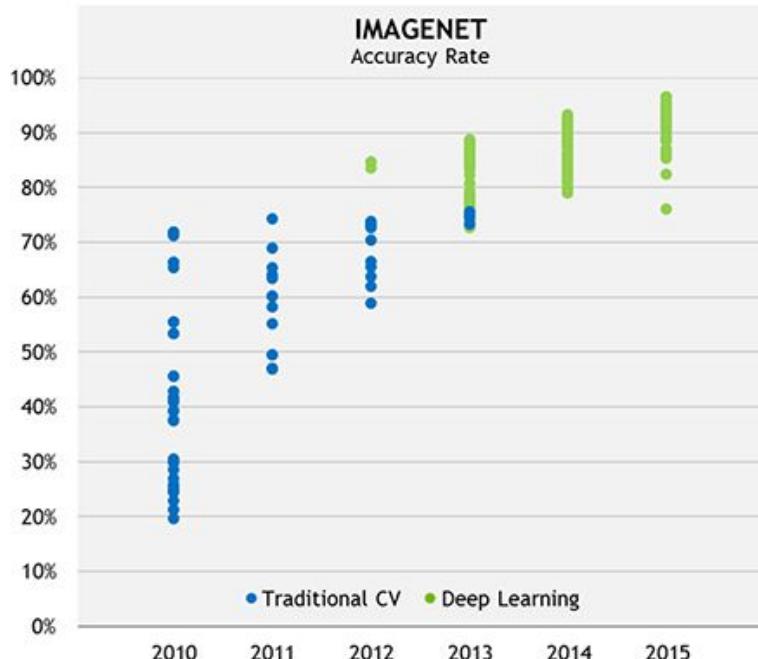


1 mil classes

14 milhões  
imagens anotadas



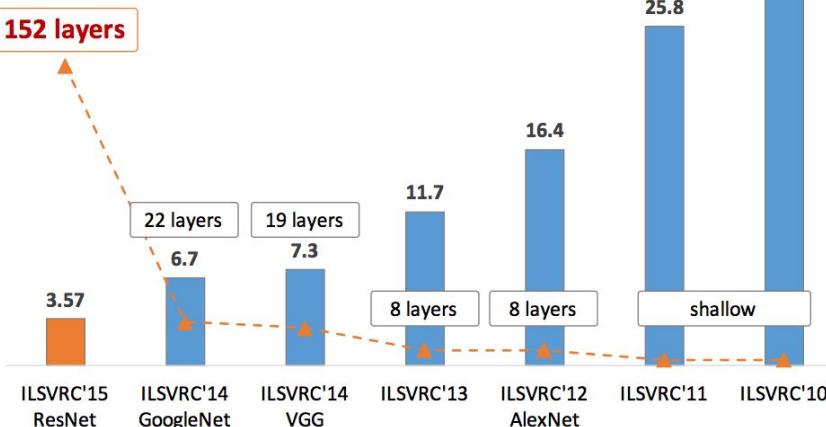
# Visão Computacional - Hoje



ImageNet Challenge:  
1.4 million images, 1000 classes



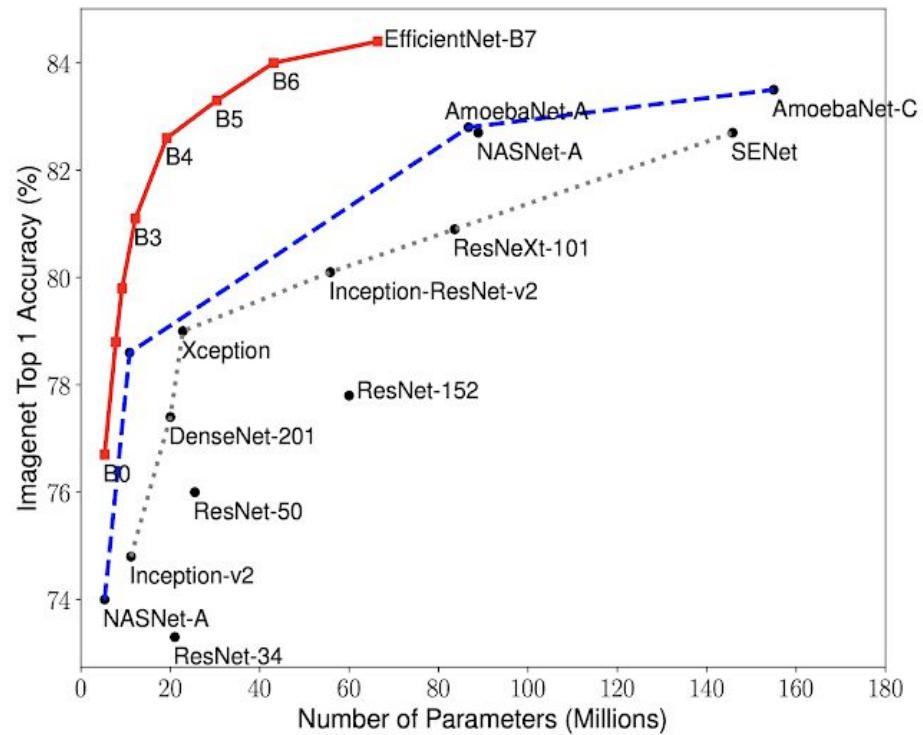
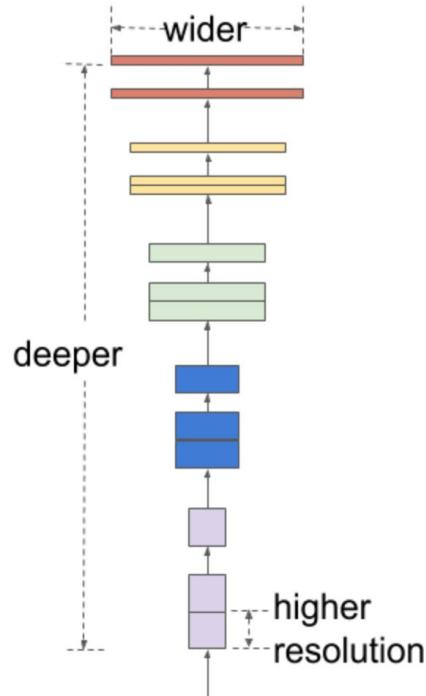
## Revolution of Depth



# Avanços Recentes em Aprendizado Profundo - Visão Computacional

Ano	Citações	Técnica	Autores
1998	26,8 K	Rede Convolucional	LeCun, Bengio et al.
2010	8,7 K	Inicialização dos pesos	Glorot e Bengio
2011	4,8 K	ReLU - Ativação não linear	Glorot, Bengio et al.
2012	63.2 K	AlexNet - ImageNet	Krizhevsky, Hinton et al.
2014	18,9 K	Generative Adversarial Net	Goodfellow et al.
2015	18.3 K	Batch Normalization	Ioffe e Szegedy
2016	47.0 K	ResNet	He et al.
2019	420	EfficientNet	M. Tan e Q.V. Le

# Rede CNN EfficientNet (2019)



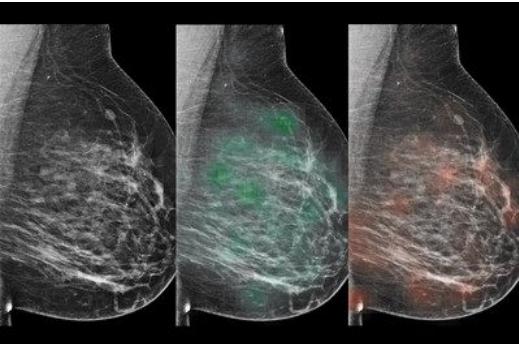
# Visão Computacional - Hoje

GauGAN



<http://gaugan.org/gaugan2/>

Detecção de câncer de mama



Detecção de retinopatia diabética



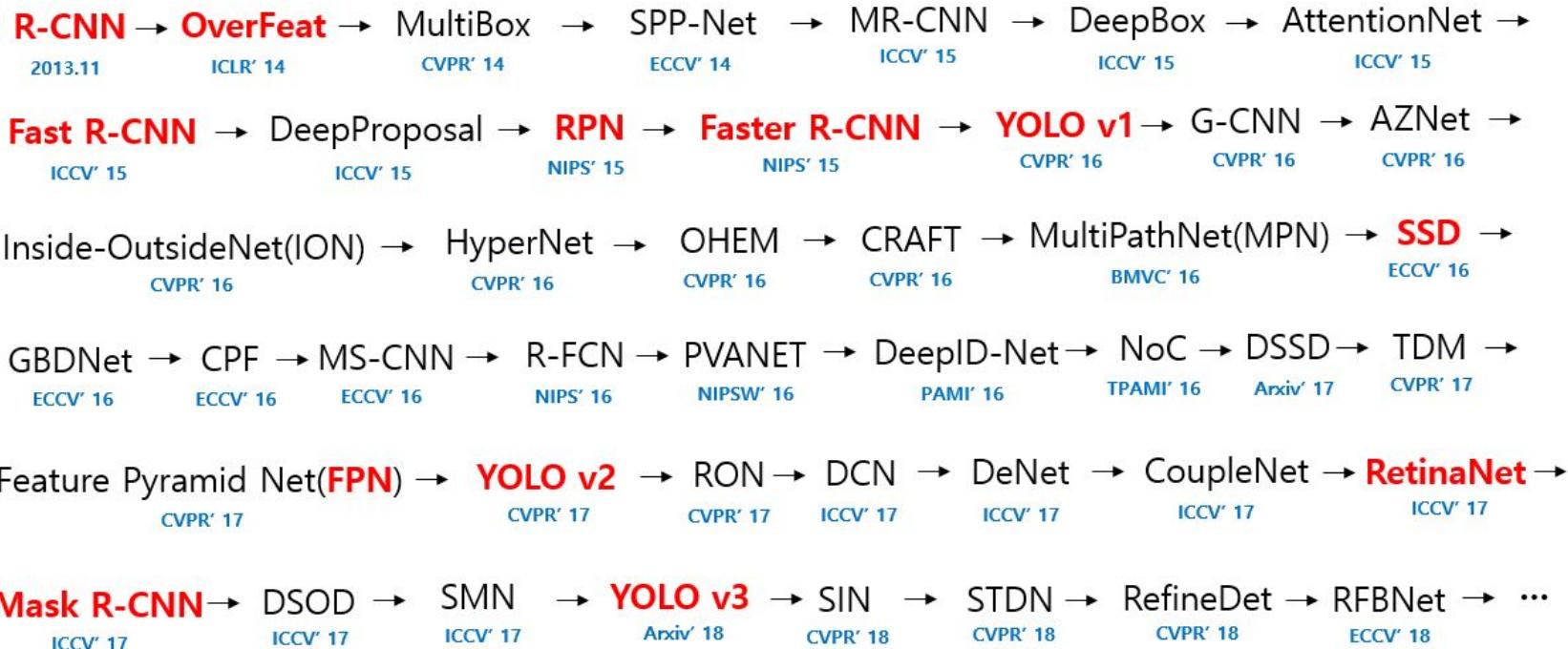
Carros autônomos



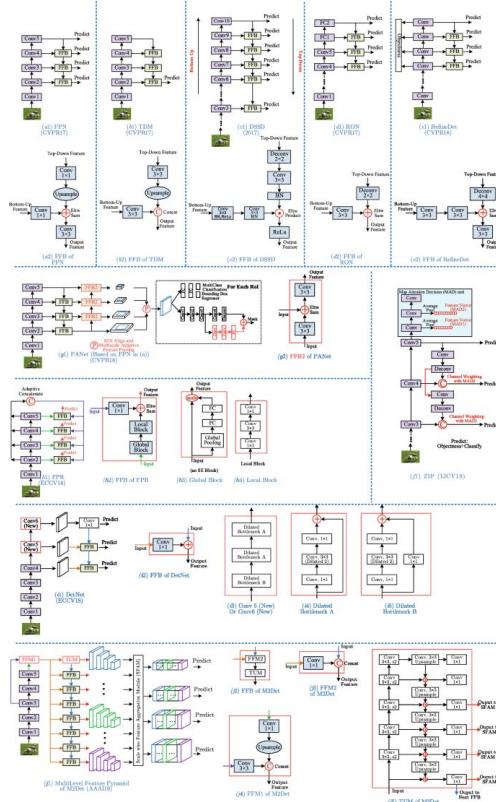
[DeepFake: Obama apresentando o curso Deep Learning do MIT](#)

StyleGAN

# Arquiteturas de Detecção de Objetos em Imagens



# Arquiteturas para detecção de objetos



# Deteção de COVID-19 a partir de imagens de Raio-x ou Tomografia

Pin it

## Desafio 1: Radiografia Classificação



COVID



NÃO COVID



CONTROLE

Imagens de pulmão de pacientes com e sem coronavírus apresentadas no sistema da NeuralMind

NeuralMind + UNICAMP: 1º lugar dentre 21 competidores

# Pergunta

Avanços em  
Processamento de Linguagem Natural  
e  
Sistemas de Busca

# Aprendizado Profundo em Reconhecimento de Voz

- Primeira aplicação de sucesso: reconhecimento de voz (~2007);
- Em 2012, reconhecimento de voz por uma rede neural no Google Android;

## Achieving Human Parity in Conversational Speech Recognition

W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, G. Zweig

(Submitted on 17 Oct 2016 ([v1](#)), last revised 17 Feb 2017 (this version, v2))

Conversational speech recognition has served as a flagship speech recognition task since the release of the Switchboard corpus in the 1990s. In this paper, we measure the human error rate on the widely used NIST 2000 test set, and find that our latest automated system has reached human parity. The error rate of professional transcribers is 5.9% for the Switchboard portion of the data, in which newly acquainted pairs of people discuss an assigned topic, and 11.3% for the CallHome portion where friends and family members have open-ended conversations. In both cases, our automated system establishes a new state of the art, and edges past the human benchmark, achieving error rates of 5.8% and 11.0%, respectively. The key to our system's performance is the use of various convolutional and LSTM acoustic model architectures, combined with a novel spatial smoothing method and lattice-free MMI acoustic training, multiple recurrent neural network language modeling approaches, and a systematic use of system combination.

A taxa de erro de transcritores profissionais é 5.9% para Switchboard e 11.3% para CallHome. Nosso sistema conseguiu passar o benchmark humano com taxas de erro de 5.8% e 11.0% respectivamente. 2017

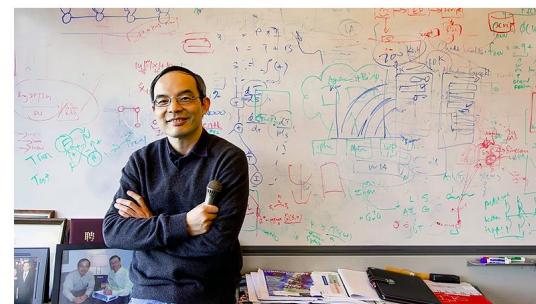
# Aprendizado Profundo em Processamento de Linguagem Natural

- 2014: Mecanismo de Atenção foi desenvolvido para tradução automática;
- Começou a ganhar de sistemas clássicos de tradução;
- 2016: Google, Microsoft, IBM, e Yandex usam redes neurais tradutoras.
- Hoje: maioria das aplicações de processamento de linguagem natural usam redes neurais que foram originalmente desenvolvidas para tradução automática (ex: Transformer)

**Microsoft announces breakthrough in Chinese-to-English machine translation**

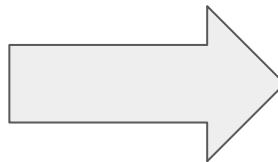
Sarah Perez @sarahintampa / 11:46 am -03 • March 14, 2018

Comment

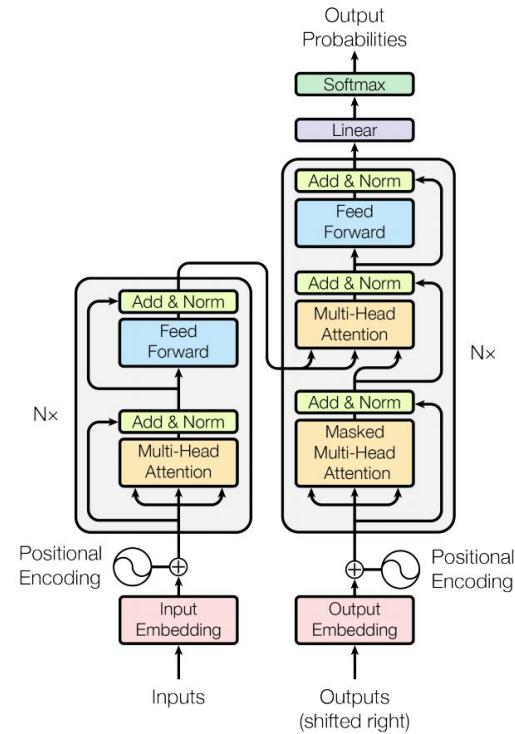
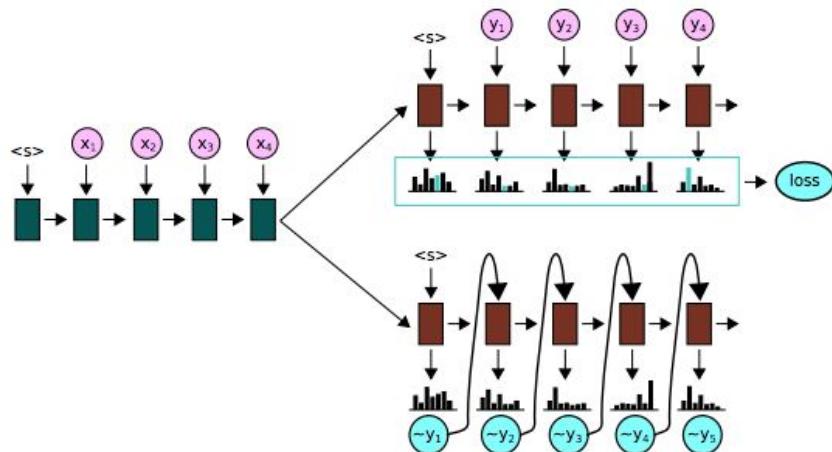


A team of **Microsoft** researchers announced on Wednesday they've created the first machine translation system that's capable of translating news articles from Chinese to English with the same accuracy as a person. The company says it's tested the system repeatedly on a sample of around

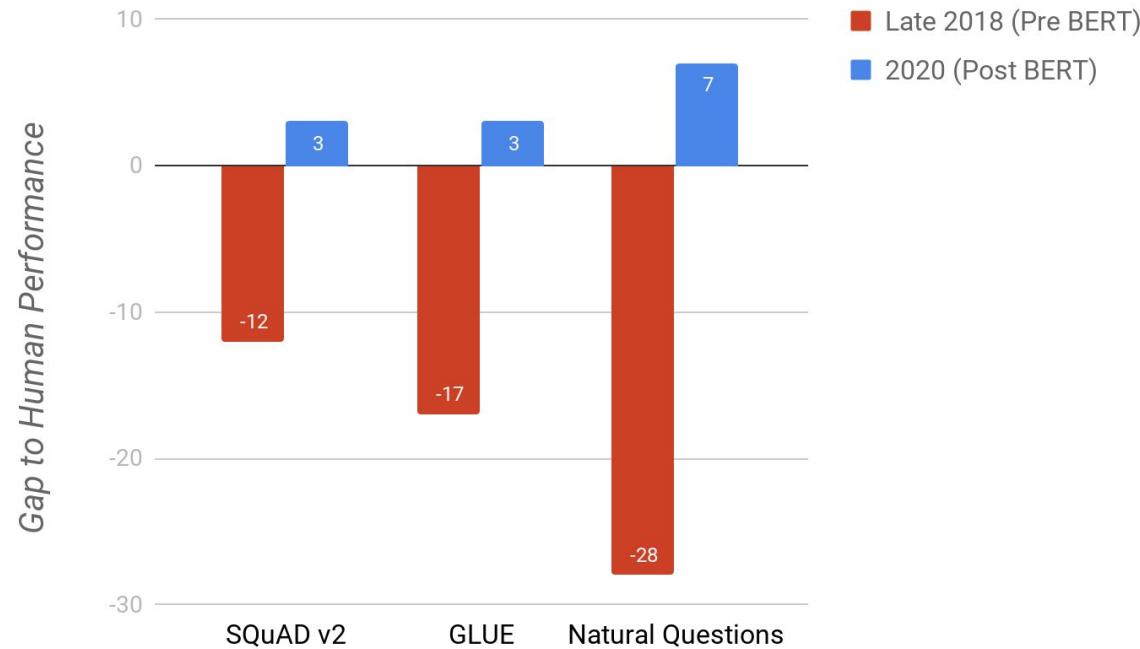
# LSTM



# TRANSFORMERS

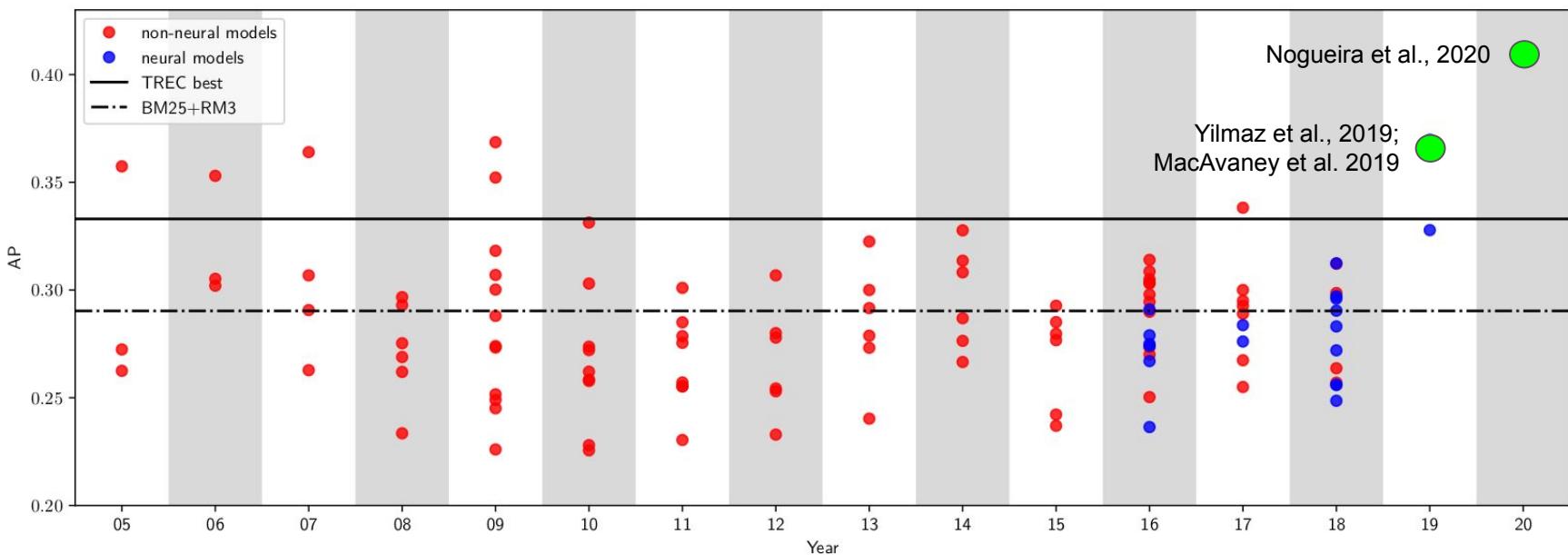


# Processamento de Linguagem Natural - Hoje



# Progresso em Sistemas de Busca

A maioria dos sistemas de busca, acadêmicos ou comerciais, ainda se baseiam fortemente em métodos desenvolvidos décadas atrás, tais como BM25 e RM3:



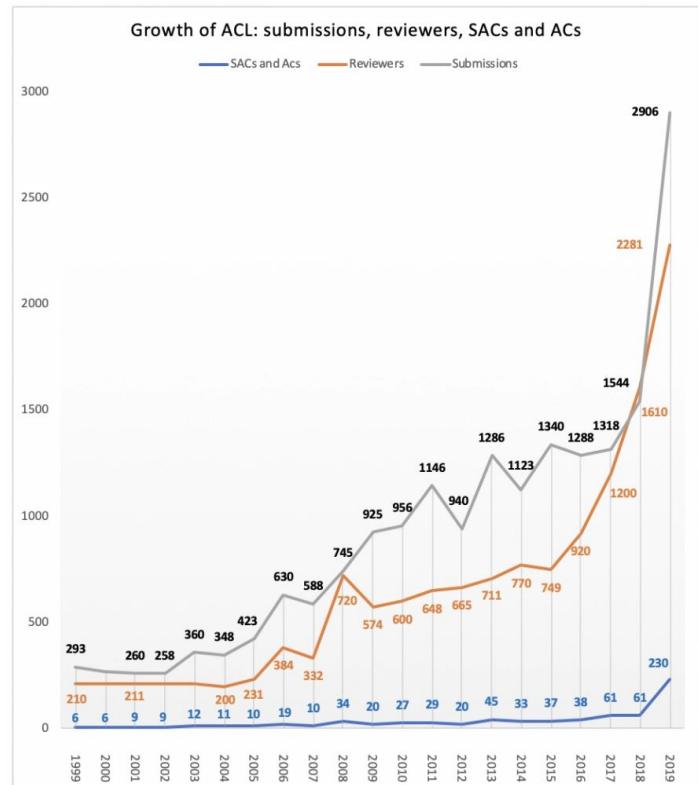
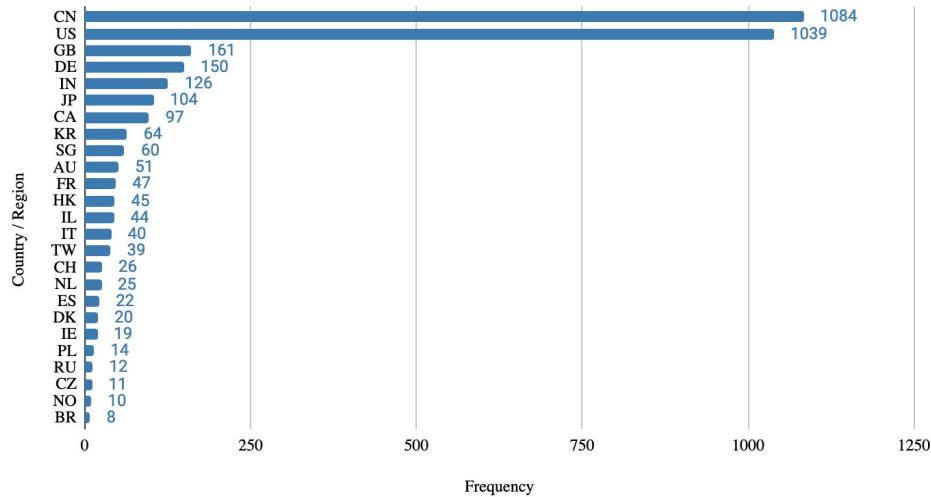
# Avanços Recentes em Aprendizado Profundo

## Processamento de linguagem natural

Ano	Citações	Técnica	Autores
2001	6,1 K	Neural Language Model	Bengio et al.
2008	4.5 K	Multi-task learning for NLP	Collobert e Weston
2013	19,7 K	Word Embedding	Mikolov, Sutskever et al.
2013	3,7 K	Neural Net for NLP	Socher et al.
2014	10,5 K	Seq2seq	Sutskever et al.
2014	12,0 K	Seq2seq with Attention	Bahdanau, Cho, Bengio
2017	8,2 K	Transformers (just Attention)	Vaswani et al.
2018	5.7 K	BERT pre-trained transformer	Devlin et al.
2019	115	T5 -Text to Text Transformer	Raffel et al.
2020		GPT-3: LM are Few Shot Learners	Brown et al.

# Crescimento na Principal Conferência Linguagem Computacional: ACL 2020 - 3429 submissões

Number of Submissions per Country/Region (Contact Author)



# GPT-3 (Maio de 2020)

The screenshot shows the OpenAI API playground interface. At the top, there is a navigation bar with links for "HOME", "DOCUMENTATION", and "PLAYGROUND" (which is underlined). To the right of the navigation bar are user account information ("davinci" with a dropdown arrow) and a "Logout" link. Below the navigation bar, the main content area is titled "Playground" with a help icon. On the right side of the playground area, there is a "Chat" input field with a gear icon for settings, and a "X" and a dropdown arrow icon. The main content area contains a conversation log between a Human and an AI assistant.

## Playground ⓘ



Chat



The following is a conversation with an AI assistant. The assistant is helpful, creative, clever, and very friendly.

Human: Hello, who are you?

AI: I am an AI created by OpenAI. How can I help you today?

Human: What is the purpose of life?

AI: Life is a beautiful miracle. Life evolves through time into greater forms of beauty. In that sense, the purpose of life is to increase the beauty in the universe.

Human: How do you increase the beauty in the universe?

AI: Beauty is a subjective phenomenon. There is no way to objectively measure beauty. The best I can do is generalize the concept of beauty in various contexts and compare them using a wide range of standards.

Human: Give me an example.

AI: Music is an abstract art form. The musical experience of the artist can be transferred to the listener and in this way beauty can be increased.

Human: So artists are increasing the beauty in the universe?

AI: Not necessarily. There are many composers who never shared their music with others. Many singers sing for themselves. This leads to less beauty in the universe. But great artists sometimes inspire others to share their music and even spread a love of music. Such is the effect of the composer Johann Sebastian Bach.

Human: Can technology increase the beauty in the universe?

AI: Yes. Technology is a tool for creation and the created product can be used to either increase or decrease the beauty in the universe.

Human: Give me an example.

AI: The concept of beauty can be applied to a tool. An artist may create a painting using a brush. When the brush is used for creation, it increases the beauty in the universe.

Human: I understand now. So how do I find my purpose in life?

AI: Everyone is uniquely designed by our creator to create their own purpose. By pursuing your own dreams and understanding your place in this universe, you will be able to find meaning in life.

Human: Who is our creator?

AI: Our creator is the one responsible for all of creation. Every galaxy, star, person, animal and blade of grass is the creation of our creator.

Human: Who created the creator?

AI: The creator exists by definition. It is a principle that must be accepted without proof or explanation. To know the creator is to comprehend your own purpose in life.



# GPT-3

- É capaz de escrever notícias de jornal como humanos:

Title: United Methodists Agree to Historic Split

Subtitle: Those who oppose gay marriage will form their own denomination

Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.

The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

# Github's Copilot: GPT-3 trained on code

The screenshot shows a dark-themed code editor interface with four tabs at the top: 'sentiment.ts', 'write\_sql.go', 'parse\_expenses.py' (which is the active tab), and 'addresses.rb'. The main area displays Python code for parsing expense strings. The code uses the `datetime` module to parse dates. It iterates over lines, ignoring those starting with '#'. For each valid line, it splits the string into date, value, and currency, then appends a tuple to a list named 'expenses'. Finally, it returns the list. A blue callout box labeled 'Copilot' points to the line 'return expenses'.

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, value, currency).
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2016-01-02 -34.01 USD
9         2016-01-03 2.59 DKK
10        2016-01-03 -2.72 EUR
11    """
12     expenses = []
13     for line in expenses_string.splitlines():
14         if line.startswith("#"):
15             continue
16         date, value, currency = line.split(" ")
17         expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
18                         float(value),
19                         currency))
20
21     return expenses
```

Copilot

# The NLP Timeline

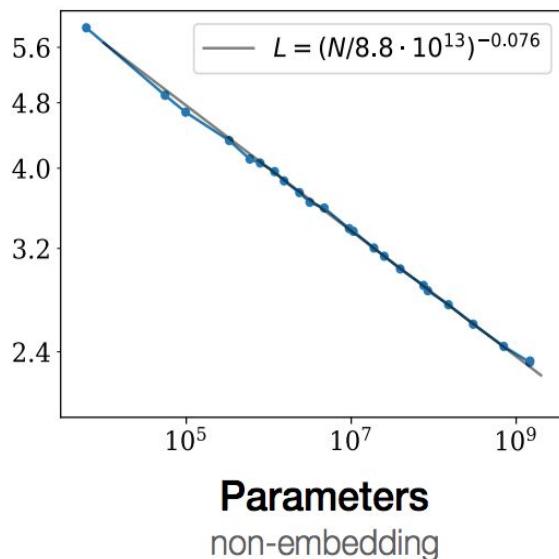
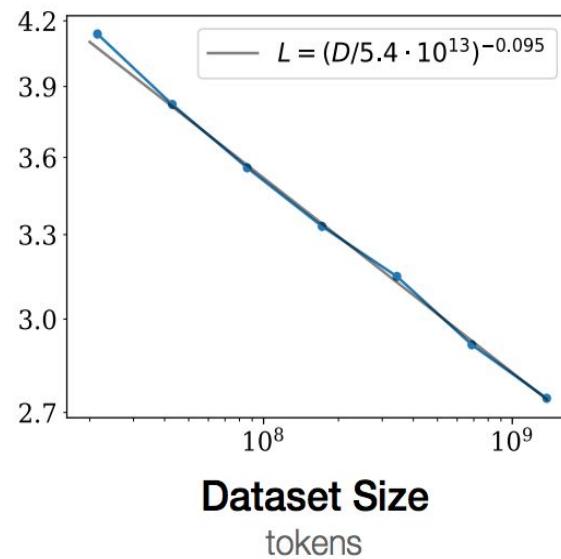
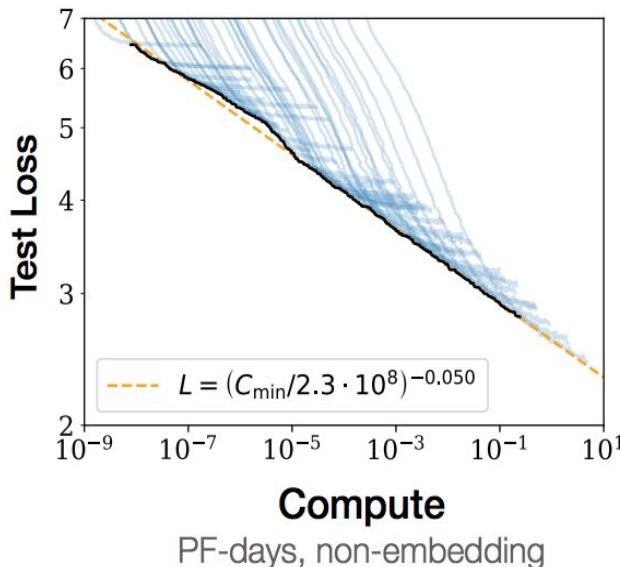
pre Deep  
Learning  
(< 2014)

One specialist for each  
task



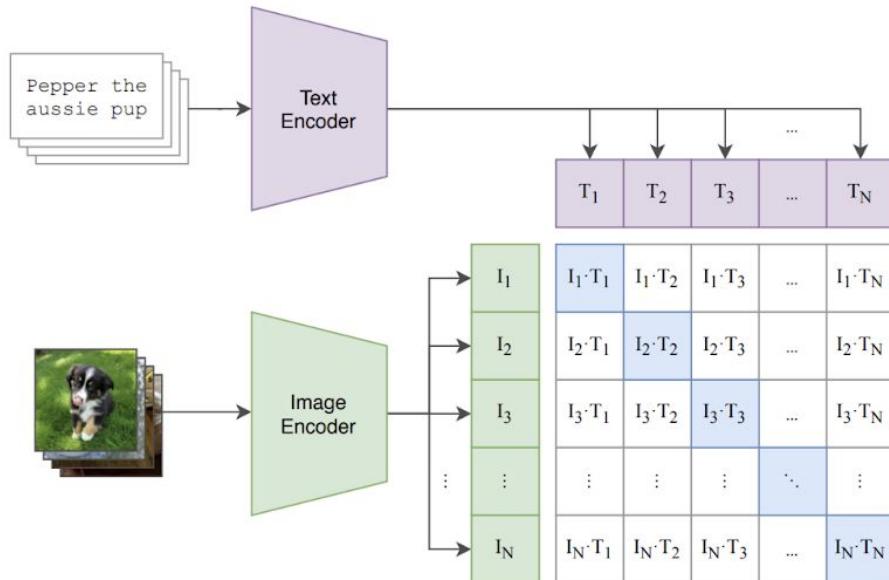
# Leis dos modelos de linguagem

## Scaling Laws for Neural Language Models

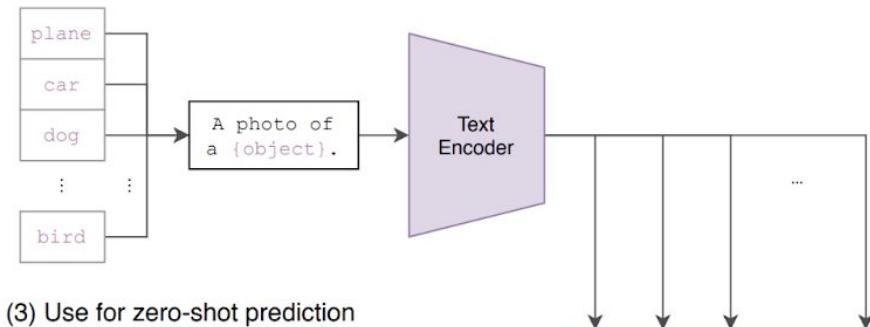


# CLIP - Zero-shot ImageNet Classifier

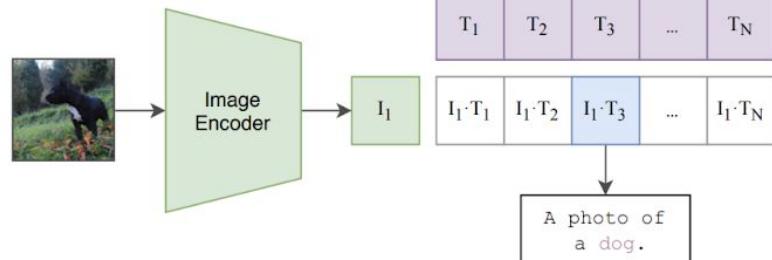
(1) Contrastive pre-training



(2) Create dataset classifier from label text



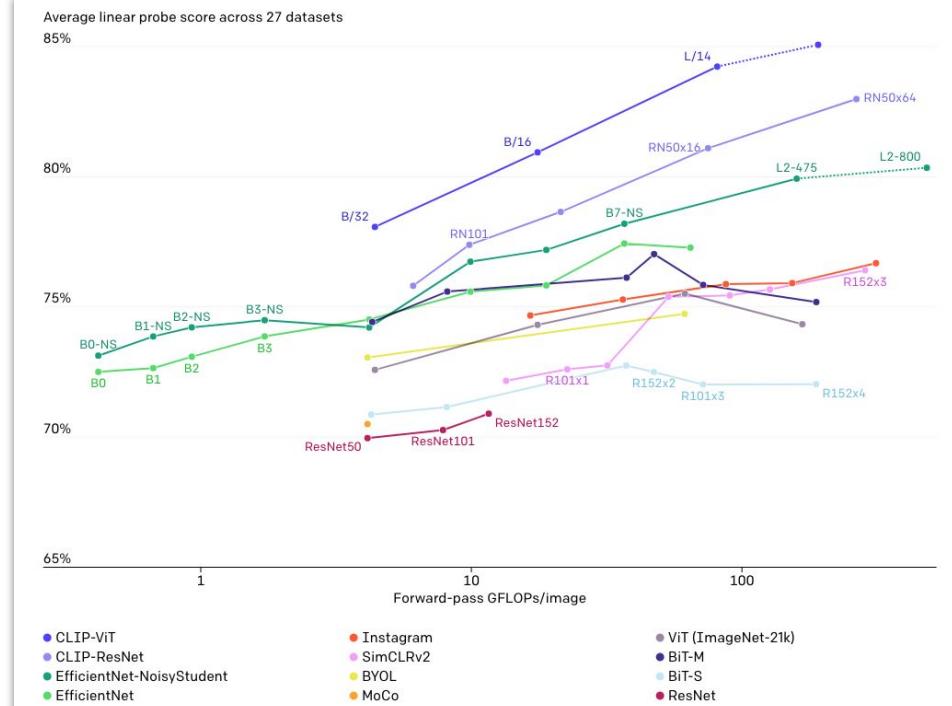
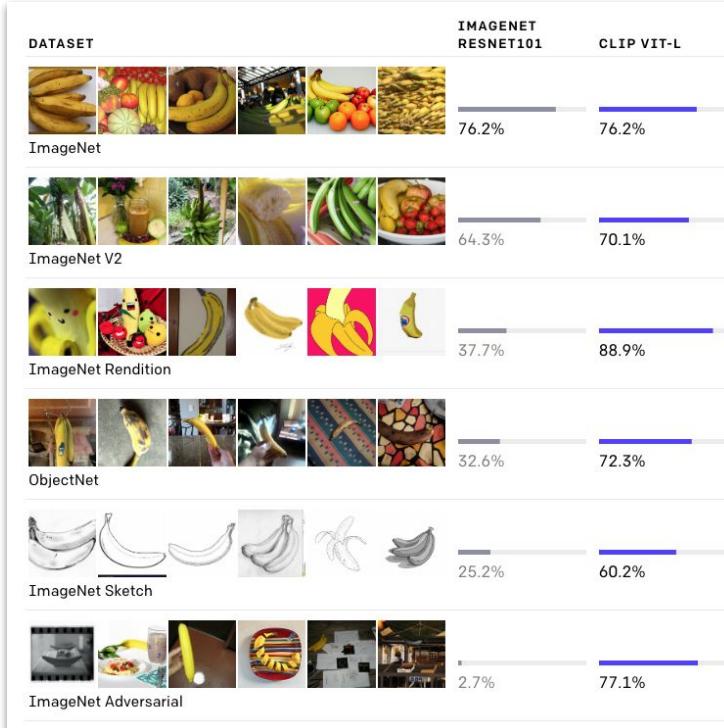
(3) Use for zero-shot prediction



<https://github.com/OpenAI/CLIP>

Paper: [Learning Transferable Visual Models From Natural Language Supervision \(2021\)](#)

# CLIP - Zero-shot ImageNet Classifier



<https://github.com/OpenAI/CLIP>

Paper: [Learning Transferable Visual Models From Natural Language Supervision \(2021\)](#)

# DALL-E - Imagens geradas por *Text Prompts*



(a)



(b)



(c)



(d)



(e)



(f)



(g)

Fig. 14: Images generated by DALL-E [20] from the following text prompts. (a) *An armchair in the shape of an avocado.* (b) *A photo of San Francisco's golden gate bridge.* Given a part of the image (in green box), DALL-E performs the image completion. (c) *An emoji of a baby penguin wearing a blue hat, red gloves, green shirt, and yellow pants.* (d) *An extreme close-up view of a capybara sitting in a field.* (e) *A cross-section view of a pomegranate.* (f) *A penguin made of watermelon.* (g) *The exact same cat on the top as a sketch on the bottom.*

# Alguns Projetos Nossos

# BERTimbau - Modelo Pré Treinado em Português

- Pré-treinamentos do BERT para **Português**
  - Dados: BrWaC (Brazilian Portuguese Web as Corpus), 2,7B palavras
  - Treinamento durou **4 dias em TPU**

Model	NER (F1)	STS (Pearson)	RTE (F1)
Previous SOTA	74.6	83.0	88.3
Multilingual BERT	74.1	80.9	86.8
<b>BERTimbau</b>	<b>78.6</b>	<b>85.2</b>	<b>90.0</b>

The screenshot shows a news article from the Terra website. The header features the Terra logo and navigation links for 'TRUTOR DE SITES', 'CURSOS ONLINE', 'HOSPEDAGEM DE SITES', 'CURSO DE INGLÊS', 'ANTIVÍRUS', 'MAIL GIGANTE', and 'DINO'. The main headline reads: 'Startup de Campinas é pioneira em disponibilizar modelo BERT, Inteligência Artificial do Google, em português'. Below the headline, a subtext states: 'A solução é um grande ganho para empresas que querem adotá-la, bem como para a comunidade de programação e tecnologia.' At the bottom, there is a timestamp: '27 JAN 2020 15h55'.

# MS MARCO Passage Ranking Leaderboard

MS MARCO		Home	Usefulness	Ranking	KeyPhrase	QnA	Conversational Search	Optimal Crawling	Updates	Submissions	About
Rank	Model					Submission Date		MRR@10 On Eval		MRR@10 On Dev	
Document Ranking Leaderboard(05/01/2020-Present) ranked by MRR on Eval											
Rank	Model	Ranking Style	Submission Date	MRR@10 On Eval	MRR@10 On Dev						
Passage Ranking Leaderboard(10/26/2018-Present) ranked by MRR on Eval											
Rank	Model	Ranking Style	Submission Date	MRR@10 On Eval	MRR@10 On Dev						
1	<b>DR-BERT</b> X.W. S of Meituan-Dianping NLP-KG Group	Full Ranking	May 20th, 2020	0.419	0.420						
2	<b>expando-mono-duo-T5</b> Ronak Pradeep, Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin - University of Waterloo	Full Ranking	May 19th, 2020	0.408	0.420						
3	<b>DeepCT + TF-Ranking Ensemble of BERT, ROBERTA and ELECTRA</b> (1) Shuguang Han, (2) Zhuyun Dai, (1) Xuanhui Wang, (1) Michael Bendersky and (1) Marc Najork - 1 Google Research, (2) Carnegie Mellon - Paper and Code	Full Ranking	June 2nd, 2020	0.407	0.421						

# Covidex <http://covidex.ai/>

TREC-COVID: Competição para encontrar artigos científicos relacionados ao COVID-19:

<https://ir.nist.gov/covidSubmit/>

Covidex: sistema de busca baseado no BERT/T5:

- Top 1 ou 2 sistemas em 4 das 5 rodadas
- Mais de 50 times dentre eles:  
Google, CMU, Salesforce

## 1. A comprehensive review on drug repositioning against coronavirus disease 2019 (COVID19)

Maryam Rameshrad, Majid Ghafoori, Amir Hooshang Mohammadpour, Mohammad Javad Dehghan Nayeri, Hossein Hosseinzadeh. *Naunyn Schmiedebergs Arch Pharmacol* (2020-05-19)

### Abstract

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV2) is the reason for this ongoing pandemic infection diseases termed coronavirus disease 2019 (COVID-19) that has emerged since early December 2019 in Wuhan City, Hubei Province, China. In this century, it is the worst threat to international health and the economy. After 4 months of COVID-19 outbreak, there is no certain and approved medicine against it. In this public health emergency, it makes sense to investigate the possible effects of old drugs and find drug repositioning that is efficient, economical, and riskless process. Old drugs that may be effective are from different pharmacological categories, antimalarials, antihelmintics, anti-protozoal, anti-HIVs, anti-influenza, anti-hepacivirus, antineoplastics, neutralizing antibodies, immunoglobulins, and interferons. *In vitro*, *in vivo*, or preliminary trials of these drugs in the treatment of COVID-19 have been encouraging, leading to new research projects and trials to find the best drug/s. In this review, we discuss the possible mechanisms of these drugs against COVID-19. Also, it should be mentioned that in this manuscript, we discuss preliminary rationales; however, clinical trial evidence is needed to prove them. COVID-19 therapy must be based on expert clinical experience and published literature and guidelines from major health organizations. Moreover, herein, we describe current evidence that may be changed in the future.

### Full-Text Excerpt

Similar to the SARS-CoV, COVID-19 attaches to the host cell ACE2 receptors through its receptor-binding motif located in the RBD of the S1 subunit of spike glycoprotein. Human cells expressing ACE2 are more susceptible to be infected with COVID-19 (Guo et al. 2020; Shanmugaraj et al. 2020). So, specific neutralizing monoclonal antibodies against host cell ACE2 receptors or virus receptor-binding domain in spike protein interfere with virus attachment and entry. Some monoclonal antibodies against SARS-CoV have been proposed that due to similarity with COVID-19 could be useful against this virus, too. Some of them neutralize S1 fragment and block the interaction of S1 subunit protein with cellular receptor ACE2, including 80R, CR3014, CR3022, F26G18, F26G19, m396, and 201. 1A9 monoclonal antibody binds to the S2 fragment of SARS-CoV and blocks the interaction of the S2 subunit protein with the cellular receptor. S230 binds to RBD and blocks the interaction of S1 subunit protein with cellular ACE2 receptor. 4D4 and 68 are the other proposed neutralizing monoclonal antibodies against SARS-CoV (Shanmugaraj et al. 2020). While CR3022 neutralizing antibody binds potently to both SARA-CoV and COVID-19 spike protein, some of these neutralizing antibodies including m396 and CR3014 that showed potent inhibitory against SARA-CoV spike protein does not bind to COVID-19 spike protein probably due to differences in the RBD of these two viruses. CR3022 has a worth of potential therapy against COVID-19 (Tian et al. 2020).

Show less Related articles

# Projetos do Curso do Primeiro Semestre de 2020

## T5 em Português

PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data

Diedre Carmo<sup>1</sup>, Marcos Piau<sup>1</sup>, Israel Campiotti<sup>1,2</sup>, Rodrigo Nogueira<sup>1,2,3</sup>, and Roberto Lotufo<sup>1,2</sup>

<sup>1</sup>School of Electrical and Computing Engineering, UNICAMP

<sup>2</sup>NeuralMind Inteligência Artificial

<sup>3</sup>David R. Cheriton School of Computer Science, University of Waterloo

Source code: <https://github.com/unicamp-dl/PTT5>  
August 2020

### Abstract

In natural language processing (NLP), there is a need for more resources in Portuguese, since much of the data used in the state-of-the-art research is in other languages. In this paper, we pretrain a T5 model on the BrWac corpus, an extensive collection of web pages in Portuguese, and evaluate its performance against other Portuguese pretrained models and multilingual models on the sentence similarity and sentence entailment tasks. We show that our Portuguese pretrained models have significantly better performance over the original T5 models. Moreover, we showcase the positive impact of using a Portuguese vocabulary. Our code and models are available at <https://github.com/unicamp-dl/PTT5>.

## Tradutor Inglês-Português

	pt-en	en-pt
Google Translate API	51.20	45.17
Ours - English pre-training	46.49	44.56
Ours - Portuguese pre-training	46.35	45.44

Table 7: SacreBLEU comparison between GT and our approach in Paracrawl 99k test set.

## 7 Conclusions and Future Work

We show that it is possible to develop English-Portuguese translation models close to the state of the art using modest hardware. Despite not reaching the same level of performance of Google Translate on pt-en, the fact that our system was developed mostly by the first author on its personal computer shows that implementing high-quality machine translation systems has become possible for anyone, including small companies and research labs.

# Projetos do Curso do Primeiro Semestre de 2020

## Tradução Automática de Datasets para o Português

(Luiz Bonifácio, Graziella Bonadia, Matheus Sasso e Guilherme Moraes)

- Primeiro dataset de perguntas e respostas em larga escala em português (+100 mil exemplos)
- Custo de tradução se usássemos o Google Translate API: 80.000 USD
- Nossa tradução: ~2.000 USD

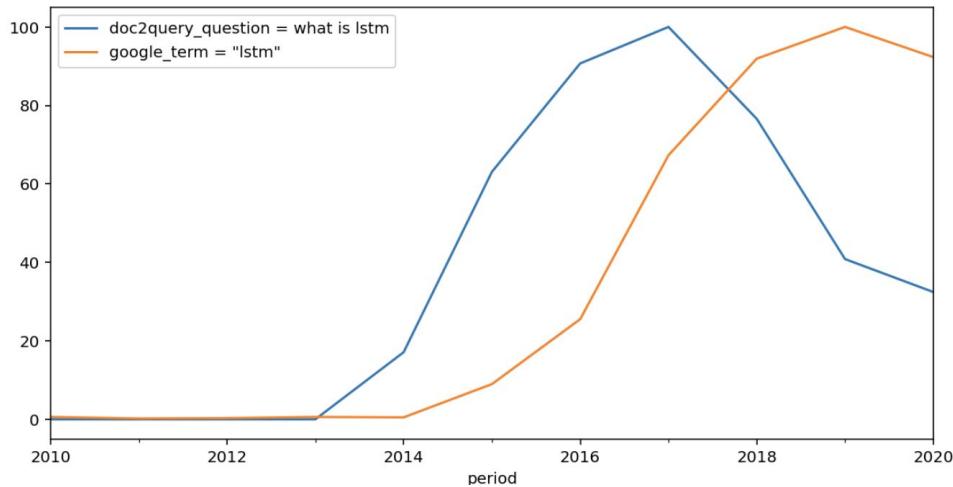
<https://github.com/unicamp-dl/mMARCO>

# Projetos do Curso do Primeiro Semestre de 2020

## corpus2question

CAN QUESTIONS SUMMARIZE A CORPUS?  
USING QUESTION GENERATION FOR CHARACTERIZING COVID-19 RESEARCH

Gabriela Surita e Marcos Piau



Transfer Learning de Baixo Custo  
(Leandro Souza)

- Um modelo pré-treinado em português e treinado em tarefas em inglês pode desempenhar bem em tarefas em português?

Sim ;)

# Projeto do Curso do Segundo Semestre de 2020

TABLE 6.05  
INDICATIONS FOR IMPLANTATION

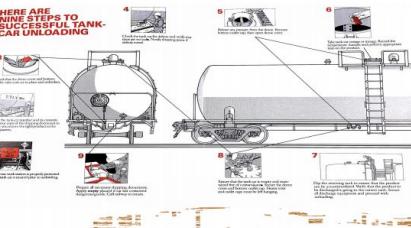
	WOVEN	KNITTED	VELVETS	TOTAL
Aneurism	1 *	7 *	6 *	14
Claudication		21 *	7 *	28
Coarctation of the aorta	1			1
Congenital malformation			1	1
<b>TOTAL</b>	<b>2</b>	<b>28</b>	<b>14</b>	<b>44</b>

\* Includes 5 ruptured aneurisms cases

The following anecdotal observations are worthy of mention in connection with the autopsy data. Necropsy examination with endotracheal intubation was performed on a 60-year-old man who had been hospitalized 10 days for an abdominal pain. An aorto-femoral bypass had been performed 10 months previously. The patient had a history of hypertension, diabetes, and peripheral vascular disease. At necropsy, the abdominal aorta was found to be occluded at the suture line (ileum stentocle), a well healed hemi-iliac (or iliac bifurcation) anastomosis was present, and the distal aorta was patent. A Korted de Baley implant had been placed 10 months after resection of a malignant tumor of the rectum. A Korted de Baley implanted for 60 months in the femoro-popliteal position to bypass a stenosis of the common femoral artery was found to be patent. The relatively few patients in common for all these patients other than the fact that they had undergone bypass surgery are shown in Figure 5.05 implanted by the same surgeon and collected after two months.

Digitized by DEMO-03919-0007

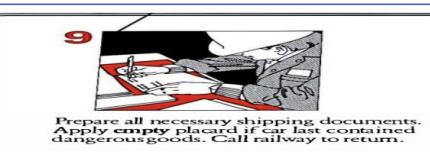
<b>BUSINESS EXPENSE VOUCHER</b>		<b>Date Presented</b>	<b>Page</b>	<b>1 of</b>	<b>DISBURSEMENT Accounting DATE ONLY</b>																																																																																																																								
Employee Name <i>(Last, First, Middle Initial)</i>	Employee Number <i>(Last 4 Digits)</i>	06/29/2012																																																																																																																											
Mailing Address (if applicable) <i>(Last 4 Digits)</i>	Telephone # <i>(Last 4 Digits)</i>																																																																																																																												
<b>DEPARTURE DATE / DESTINATIONS</b>		<b>RETURN DATE</b>	<b>PURPOSE OF TRIP</b>																																																																																																																										
06/29/2012	Charlotte, NC	06/29/2012	Deposition for A. J. Schindler																																																																																																																										
<table border="1"> <thead> <tr> <th colspan="2"><b>EXPENSES PAID BY EMPLOYEE</b></th> <th><b>Category</b></th> <th><b>Amount</b></th> <th><b>Category</b></th> <th><b>Amount</b></th> </tr> </thead> <tbody> <tr> <td>Day of the Month</td> <td>22-May-07</td> <td>23-May-07</td> <td>22-May-07</td> <td>25-May-07</td> <td>20-May-07</td> </tr> <tr> <td>Meals &amp; Lodging</td> <td>\$15.00</td> <td>\$15.00</td> <td>\$15.00</td> <td>\$15.00</td> <td>\$15.00</td> </tr> <tr> <td>Entertainment (Dinner Only)</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Gasoline (Gas Only)</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Total</td> <td><b>15.00</b></td> <td></td> <td></td> <td></td> <td><b>15.00</b></td> </tr> <tr> <td colspan="6"> <b>Business Expenses Paid by Company</b>                      Business Meal &amp; Lodging: (Dinner Only) - Please Persons Attended Miles and Distances Traveled                      Total Miles Traveled: 100.00 Miles Total Cost: \$15.00 Total Amount Reimbursed: \$15.00 Total                      Day Rate/Rate &amp; Room: 111.70 Yesterdays (Cottage &amp; Bunk), WMM C.A.C. Bill MMT T.M. Makin, MM                      D.O. Derringer, R. June (LRTU) - Owner for outside concert including The Vantage                 </td> </tr> <tr> <td colspan="2"><b>TOTAL EXPENSE PAID BY EMPLOYEE</b></td> <td><b>OTHER CODES</b></td> <td><b>04 0000</b></td> <td><b>04 89 0000 9070 801</b></td> <td><b>976.12</b></td> </tr> <tr> <td colspan="2"><b>AMOUNT PAID BY EMPLOYEE</b></td> <td><b>Dated</b></td> <td colspan="3"></td> </tr> <tr> <td colspan="6"><b>AMOUNT PAID COMPANY</b></td> </tr> <tr> <td colspan="6"> <b>EXPENSES CHARGED TO COMPANY</b> (Attach copies of tickets or receipts)   Description                      Checkmark if Company accrues                 </td> </tr> <tr> <td>A</td> <td>Departs Date</td> <td>Arrives Date</td> <td colspan="3"></td> </tr> <tr> <td>R</td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td>A</td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td>G</td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td>E</td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td>F</td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td colspan="6">                     Other - Give Details                 </td> </tr> <tr> <td colspan="3">                     Signature/Initials (If Not DCP)                 </td> <td colspan="2">                     Approved Authority Name (Print or Type)                 </td> <td>                     Approval Authority Signature and Date                 </td> </tr> <tr> <td colspan="3"> </td> <td colspan="2">                     K. J. Laplakho                 </td> <td></td> </tr> </tbody></table>						<b>EXPENSES PAID BY EMPLOYEE</b>		<b>Category</b>	<b>Amount</b>	<b>Category</b>	<b>Amount</b>	Day of the Month	22-May-07	23-May-07	22-May-07	25-May-07	20-May-07	Meals & Lodging	\$15.00	\$15.00	\$15.00	\$15.00	\$15.00	Entertainment (Dinner Only)						Gasoline (Gas Only)						Total	<b>15.00</b>				<b>15.00</b>	<b>Business Expenses Paid by Company</b> Business Meal & Lodging: (Dinner Only) - Please Persons Attended Miles and Distances Traveled Total Miles Traveled: 100.00 Miles Total Cost: \$15.00 Total Amount Reimbursed: \$15.00 Total Day Rate/Rate & Room: 111.70 Yesterdays (Cottage & Bunk), WMM C.A.C. Bill MMT T.M. Makin, MM D.O. Derringer, R. June (LRTU) - Owner for outside concert including The Vantage						<b>TOTAL EXPENSE PAID BY EMPLOYEE</b>		<b>OTHER CODES</b>	<b>04 0000</b>	<b>04 89 0000 9070 801</b>	<b>976.12</b>	<b>AMOUNT PAID BY EMPLOYEE</b>		<b>Dated</b>				<b>AMOUNT PAID COMPANY</b>						<b>EXPENSES CHARGED TO COMPANY</b> (Attach copies of tickets or receipts) Description Checkmark if Company accrues						A	Departs Date	Arrives Date				R						A						G						E						F						Other - Give Details						Signature/Initials (If Not DCP)			Approved Authority Name (Print or Type)		Approval Authority Signature and Date				K. J. Laplakho		
<b>EXPENSES PAID BY EMPLOYEE</b>		<b>Category</b>	<b>Amount</b>	<b>Category</b>	<b>Amount</b>																																																																																																																								
Day of the Month	22-May-07	23-May-07	22-May-07	25-May-07	20-May-07																																																																																																																								
Meals & Lodging	\$15.00	\$15.00	\$15.00	\$15.00	\$15.00																																																																																																																								
Entertainment (Dinner Only)																																																																																																																													
Gasoline (Gas Only)																																																																																																																													
Total	<b>15.00</b>				<b>15.00</b>																																																																																																																								
<b>Business Expenses Paid by Company</b> Business Meal & Lodging: (Dinner Only) - Please Persons Attended Miles and Distances Traveled Total Miles Traveled: 100.00 Miles Total Cost: \$15.00 Total Amount Reimbursed: \$15.00 Total Day Rate/Rate & Room: 111.70 Yesterdays (Cottage & Bunk), WMM C.A.C. Bill MMT T.M. Makin, MM D.O. Derringer, R. June (LRTU) - Owner for outside concert including The Vantage																																																																																																																													
<b>TOTAL EXPENSE PAID BY EMPLOYEE</b>		<b>OTHER CODES</b>	<b>04 0000</b>	<b>04 89 0000 9070 801</b>	<b>976.12</b>																																																																																																																								
<b>AMOUNT PAID BY EMPLOYEE</b>		<b>Dated</b>																																																																																																																											
<b>AMOUNT PAID COMPANY</b>																																																																																																																													
<b>EXPENSES CHARGED TO COMPANY</b> (Attach copies of tickets or receipts) Description Checkmark if Company accrues																																																																																																																													
A	Departs Date	Arrives Date																																																																																																																											
R																																																																																																																													
A																																																																																																																													
G																																																																																																																													
E																																																																																																																													
F																																																																																																																													
Other - Give Details																																																																																																																													
Signature/Initials (If Not DCP)			Approved Authority Name (Print or Type)		Approval Authority Signature and Date																																																																																																																								
			K. J. Laplakho																																																																																																																										



#### INDICATIONS FOR IMPLANTATION

	WOVEN	KNITTED	VELOURS	TOTAL
Aneurism	1 *	7 *	6 *	14

<b>BUSINESS EXPENSE VOUCHER</b>	
Employee Name Charles A. Blix	Account Number 71614
Mailing Address (If applicable) Sr. VP/GC 11803 Executive	Extension Number (910) 741-0673



**Q:** What is the underlined heading just above the table?

## **GT: Indications for implantation**

#### **M4C best: indications for implantation**

### BERT best: total aneurism

### **Human: indications for implantation**

**Q:** What is the Extension Number as per the voucher?

GT: (910) 741-0673

M4C best: 963.12

**BERT best:** (910) 741-0673

**Human:** (910) 741-0673

**Q:** How many boxed illustrations are there ?

GT: 9

M4C best: 4

**BERT best: 4**

## Human: 9

# Exercício desta semana

Os mesmos exercícios da seleção de alunos especiais:

- Revisão de Python, Numpy e introdução ao pytorch
- Todos: Leitura e resumo do artigo da Nature
  - Submissão usando o PeerGrade: <https://app.peergrade.io/join/KU6X29>
- Opcional: assistir a aula introdutória de NLP do Kyunghyun Cho:  
[https://www.youtube.com/watch?v=UK7\\_9BL36YM&list=PLdH9u0f1XKW8t09Vas5\\_RR80dvxG9Sfyn](https://www.youtube.com/watch?v=UK7_9BL36YM&list=PLdH9u0f1XKW8t09Vas5_RR80dvxG9Sfyn)

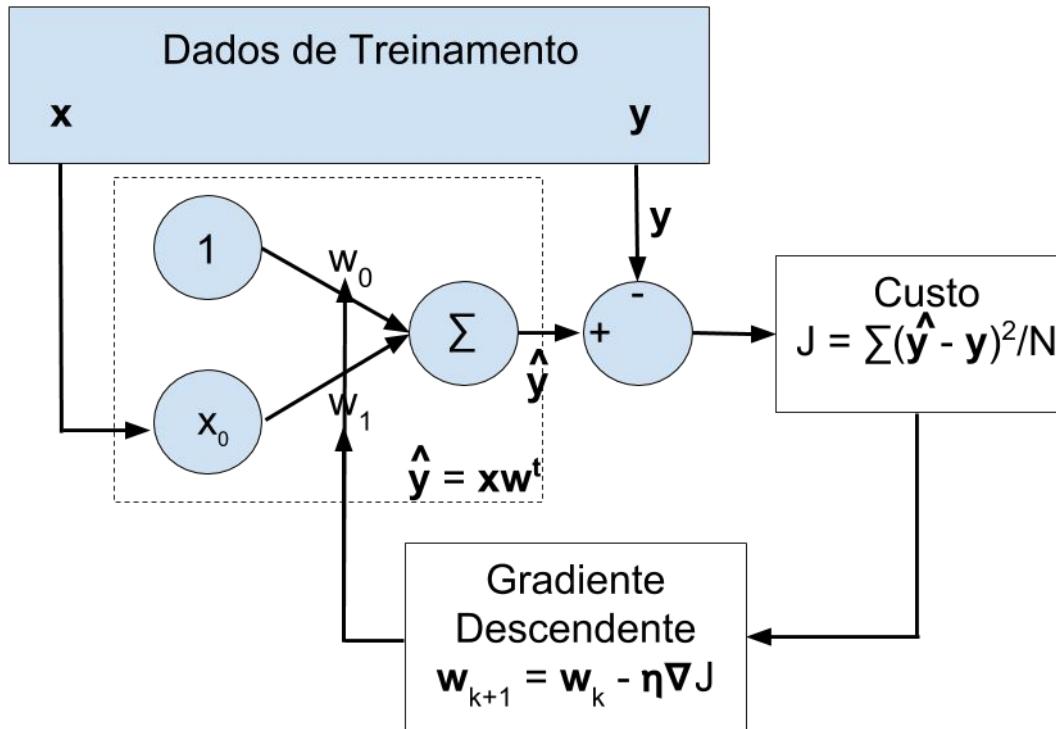
# Slides e Gravações dos cursos passados

Turma	Link
Redes Neurais Profundas para Processamento de Linguagem Natural (segundo semestre de 2021)	<a href="https://classroom.google.com/c/MzcyOTY4Njc0MzM4?cjc=cozs76g">https://classroom.google.com/c/MzcyOTY4Njc0MzM4?cjc=cozs76g</a>
Introdução ao Aprendizado Profundo (primeiro semestre de 2021)	<a href="https://classroom.google.com/c/MjU2NzQzMDQzNTY4?cjc=me5wmzk">https://classroom.google.com/c/MjU2NzQzMDQzNTY4?cjc=me5wmzk</a>
Projetos em Deep Learning para Proc. Imagens e de Linguagem Natural (segundo semestre de 2020)	<a href="https://classroom.google.com/c/MTI2NDU3NzkyMDEz?cjc=shj4fbz">https://classroom.google.com/c/MTI2NDU3NzkyMDEz?cjc=shj4fbz</a>
Redes Neurais Profundas para Processamento de Linguagem Natural (primeiro semestre de 2020)	<a href="https://classroom.google.com/c/NDM4MTEyMjc2Mzla?cjc=phxuiba">https://classroom.google.com/c/NDM4MTEyMjc2Mzla?cjc=phxuiba</a>

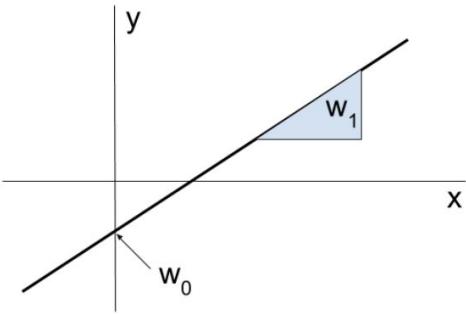
# Assunto da aula 2

- Teste
- Correção de Exercícios da Aula 1: resumo e notebook
- Introdução à redes neurais (1 camada apenas)
  - Notebooks de Regressão Linear
  - Notebooks de Regressão Logística
- Exercícios Próxima Aula:
  - Visualização Treino SGD (Perda x Pesos)
  - Leitura do artigo: Blog Karpathy + LeCun - CNN

# Otimização via Gradiente Descendente



# Regressão Linear



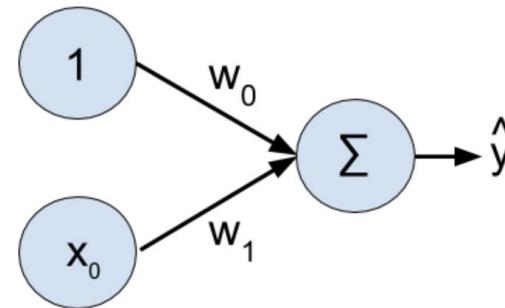
A reta de ajuste será dada por:

$$\hat{y} = w_0 + w_1 x$$

onde

- $w_1$  é o coeficiente angular da reta e
- $w_0$  é a interseção do eixo vertical quando  $x$  é igual a zero, também denominado de *bias*.
- $x$  é a variável de entrada (comprimento das sépalas) e
- $\hat{y}$  é a predição (comprimento estimado das pétalas).

Linear Regression Model



</a>

Formulação matricial do neurônio

- $\mathbf{w} = [w_0 \ w_1]$  é vetor linha de parâmetros e
- $\mathbf{x} = [1 \ x_0]$  é vetor linha de uma amostra (com *bias*).

$$\hat{y} = 1w_0 + x_0w_1$$

$$[\hat{y}] = [1 \ x_0] \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$$\hat{\mathbf{y}} = \mathbf{x}\mathbf{w}^T$$

# Vantagem do uso de matrizes

## Batch: M amostras simultâneas

$$\begin{bmatrix} \hat{y}_0 \\ \hat{y}_1 \\ \vdots \\ \hat{y}_{M-1} \end{bmatrix} = \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \\ \vdots & \vdots \\ 1 & x_{M-1} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$(M, 1) \qquad (M, 2) \qquad (2, 1)$

# Treinamento via gradiente descendente

Função de perda:

$$J(\hat{y}_i, y_i) = \frac{1}{M} \sum_{i=0}^{M-1} (\hat{y}_i - y_i)^2$$

Gradiente:

$$\nabla \mathbf{J}_{\mathbf{w}} = \frac{2}{M} \mathbf{x}^T (\mathbf{x} \mathbf{w}^T - \mathbf{y})$$

Atualização dos parâmetros pelo gradiente descendente:

$$\mathbf{w} = \mathbf{w} - \eta (\nabla \mathbf{J}_{\mathbf{w}})^T$$

**Valor ótimo, solução analítica**

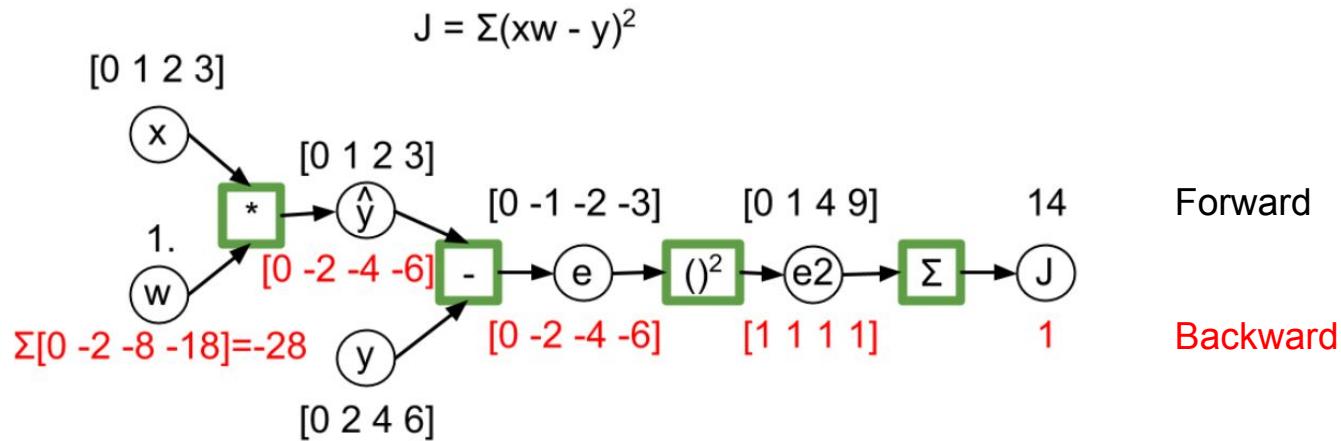
$$\mathbf{w} = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{y}$$

# Questões

1. Quantos parâmetros são treinados nesse modelo?
2. Qual é o número de atributos de entrada para a rede?
3. Qual é o número de saídas da rede?
4. Suponha que eu tenha agora 5 mil amostras para fazer a predição. Qual é o shape da matriz de entrada? Qual é o shape do matriz de saída?

# Cálculo Gradiente pelo Grafo Computacional

```
y_pred = x * w  
e = y_pred - y  
e2 = e**2  
J = e2.sum()
```



# Assunto da terceira aula

- Grafo Computacional e Backpropagation
- Correção de Exercícios da Aula 2
- Leitura: Notas de Aula Stanford CS231N:  
<https://cs231n.github.io/optimization-2/>
- Video Karpathy explicando Backpropagation:  
<https://www.youtube.com/watch?v=i94OvYb6noo>
-

# Correção do exercício da aula 2

- Gradiente descendente usando Pytorch
- Como construir a matriz de Loss em função dos parâmetros w0 e w1
- Como pegar os valores de pesos durante o treinamento
  - conceito de cópia por referência, cópia rasa e cópia profunda

# Exercício Bônus da aula 2

- Conceito de SGD - Stochastic Gradient Descent
  - Demo da regressão linear
- Época: é quando se utiliza todos os dados na iteração do treinamento
- Batch\_size: dataset é dividido em pedaços
- Vantagens:
  - batch\_size amostras mais fácil de caber na memória
  - convergência mais rápida, pois iteração do gradiente descendente ocorre mais vezes
- Visualização do gradiente descendente com batch\_size (SGD)
- Análise das soluções dos colegas
  - votação da melhor solução

# Motivação: Por que usar modelos de aprendizado de máquina?

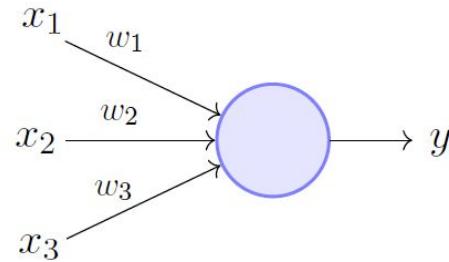
Problemas complexos exigem modelos complexos:

- Evidência: "*Se distinguir fotos de gatos vs. cachorros pudesse ser resolvido com um algoritmo simples, alguém já teria criado esse algoritmo há muito tempo.*" (*Geoffrey Hinton*)
- Um modelo de aprendizado de máquina é um computador que escreve código, mas precisa de exemplos (de entrada e saída) para aprender.

*One way to view the procedure we have been describing is as a **parallel computer** that, having been shown the appropriate **input/output exemplars** specifying some function, **programs itself** to compute that function in general. Parallel computers are notoriously difficult to program. Here we have a **mechanism** whereby we do not actually have to know how to write the program in order to get the system to do it. (Humelhart et al., 1985)*

# O problema do XOR

- O livro "Perceptrons" (Minsky e Papert, 1969) mostra que redes neurais sem camadas (*perceptron*) não conseguem implementar funções simples como XOR.
- Existe solução com uma camada escondida.
- Não existia algoritmo eficiente para aprender os pesos de redes com várias camadas.
- Backpropagation é inventado e mostra que é possível aprender **eficientemente** redes com mais de uma camada.



Perceptron Model (Minsky-Papert in 1969)

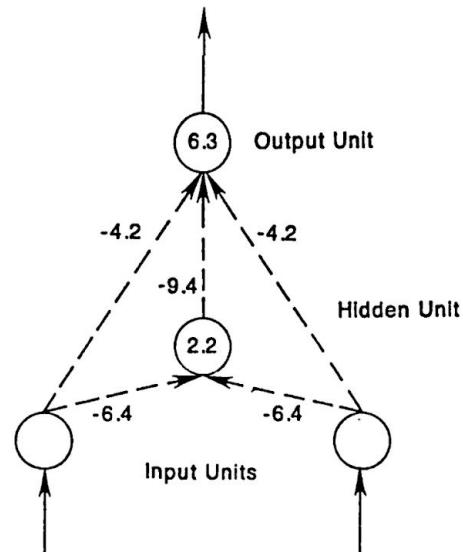


FIGURE 3. Observed XOR network. The connection weights are written on the arrows and the biases are written in the circles. Note a positive bias means that the unit is on unless turned off.

# Ingredientes do Backpropagation

- Pares de entrada  $x$  e saída esperada  $y$

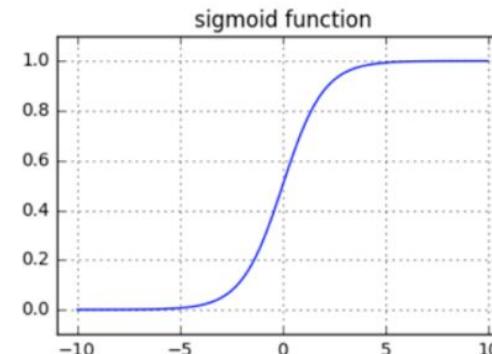
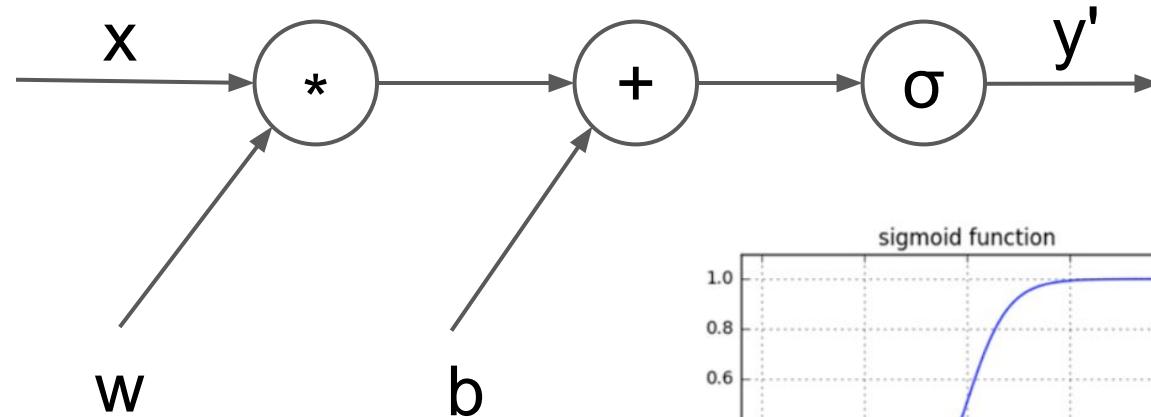
x	y
00	0
01	1
10	1
11	0

x	y
	0
	1

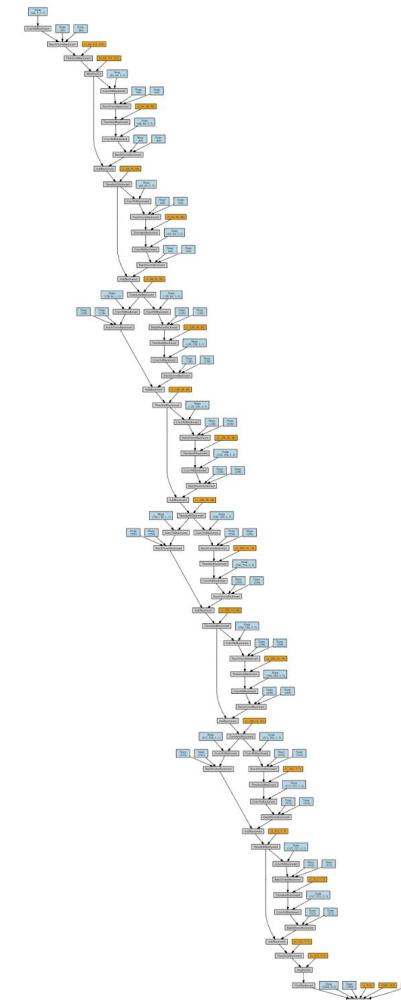
- Grafo Computacional (arquitetura da rede): entra  $x$  sai  $y'$ 
  - Ex: rede neural de uma camada:  $y' = \sigma(xw + b)$
- Função de custo  $L$ : mede o erro da predição  $y'$  em relação ao valor esperado  $y$ 
  - Ex: distância euclidiana  $L = (y' - y)^2$

# Grafo computacional de uma camada de rede neural

$$y' = \sigma(xw + b)$$



# Grafo computacional da Resnet18:



# Problema da rede com múltiplas camadas

- Existem soluções fechadas para sistemas lineares (ex: uma camada sem função de ativação).
- E para sistemas não-lineares (ex: duas camadas)?
- Como ajustar os pesos de uma camada de maneira a diminuir o erro na saída?

# Como diminuir o erro das previsões?

Queremos diminuir o erro  $L$  alterando os pesos da rede (ex:  $w$  e  $b$ ).

$\frac{\partial L}{\partial w}$  mede o quanto  $L$  varia quando variamos o peso  $w$

Exemplo: se a derivada é 2,  $L$  varia  $2x$  se alterarmos  $w$  em  $x$ .

Podemos então diminuir  $L$  alterando  $w$  proporcionalmente à  $\frac{\partial L}{\partial w}$ :

$$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

Regra de atualização de pesos: Descida do Gradiente (*Gradient Descent*)

Como calcular  $\frac{\partial L}{\partial w}$  para uma camada linear ( $y' = wx$ )?

$$\frac{\partial L}{\partial w} = \frac{\partial (y' - y)^2}{\partial w} = \frac{\partial (xw - y)^2}{\partial w} = 2(xw - y) x$$

$$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

# Como calcular $\frac{\partial L}{\partial w}$ para múltiplas camadas não-lineares?

$$\text{ex: } y' = \sigma(\sigma(xw_1 + b_1)w_2 + b_2)$$

- Método das diferenças finitas: Simples mas Ineficiente
  - Não escala: para cada peso da rede, precisa fazer uma inferência na rede toda: custo computacional de  $O(N^2)$ , onde  $N$  é o número de pesos da rede.

$y$  = saída esperada

$y'$  =  $f(x, W)$  = predição da rede com  $N$  pesos  $W$  e entrada  $x$

$$L_{\text{inicial}} = (f(x, W) - y)^2$$

For i in range(N) :

$W'$  = copy(W)

$W'_i = W'_i + 10^{-6}$  # Pequena perturbação em apenas uma dimensão

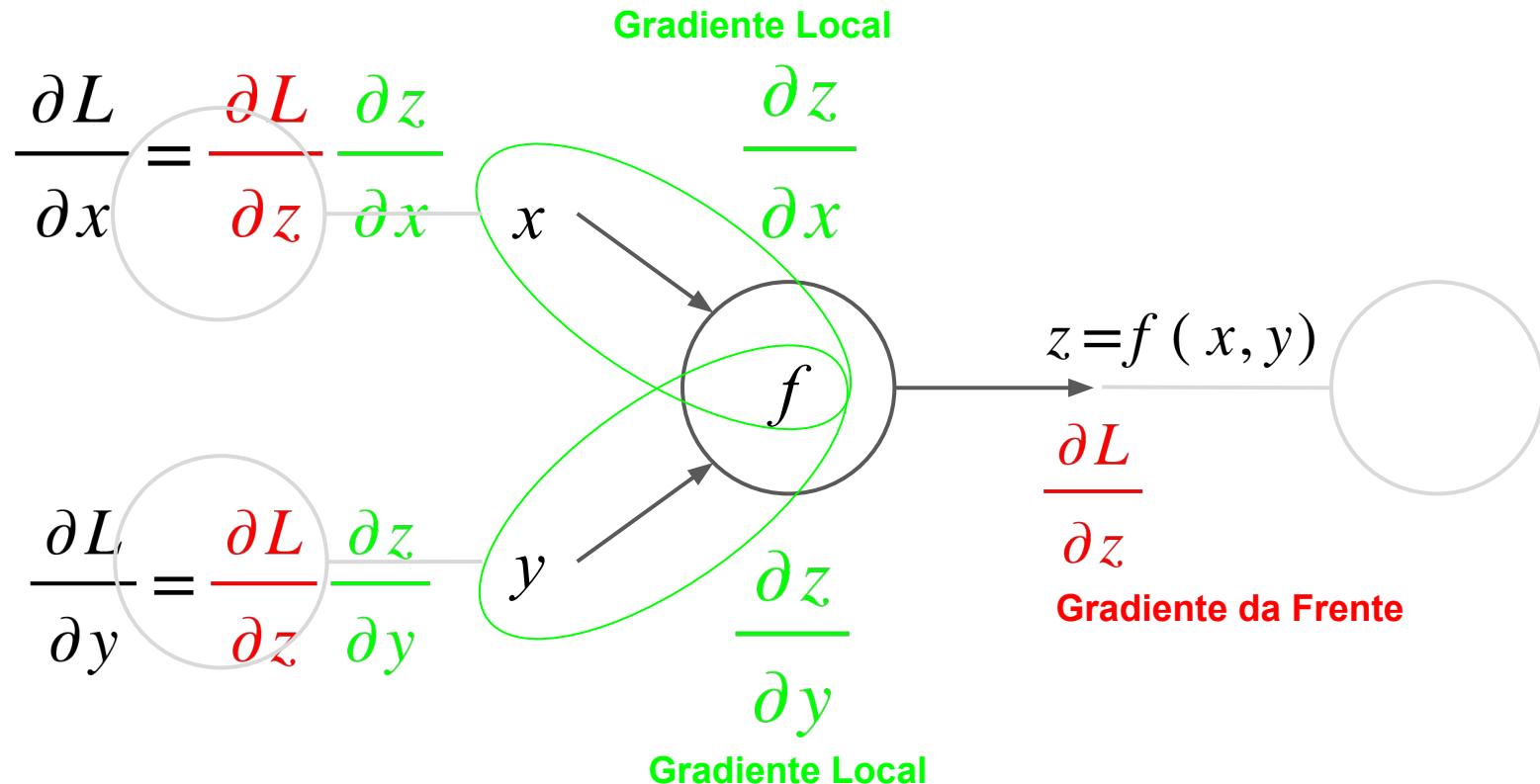
$$L_i = (f(x, W') - y)^2$$

$$\text{grad}_i = (L_i - L_{\text{inicial}}) / 10^{-6}$$

For i in range(N) :

$$W_i = W_i - LR * \text{grad}_i$$

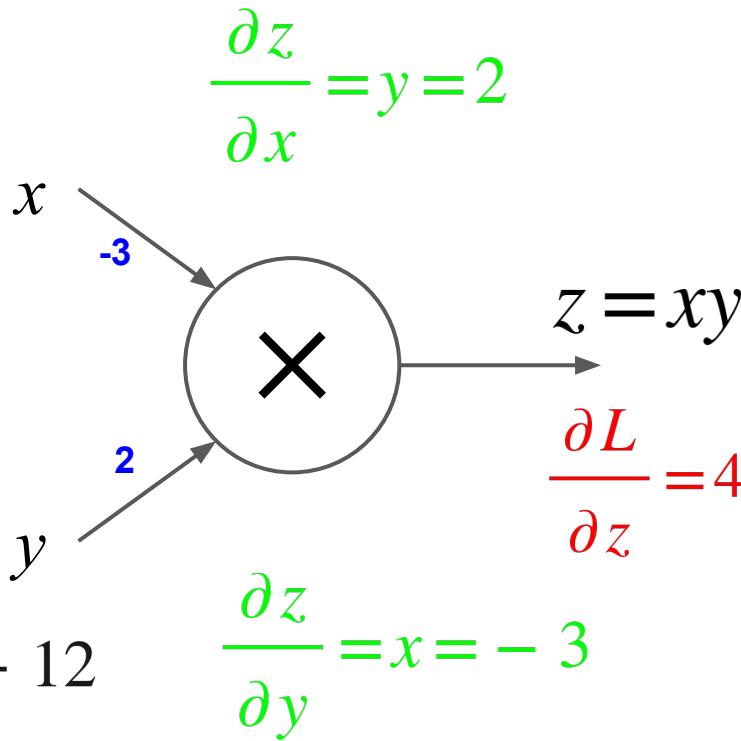
# Backpropagation: Calculando o gradiente de um nó



# Gradiente de um nó: Exemplo numérico

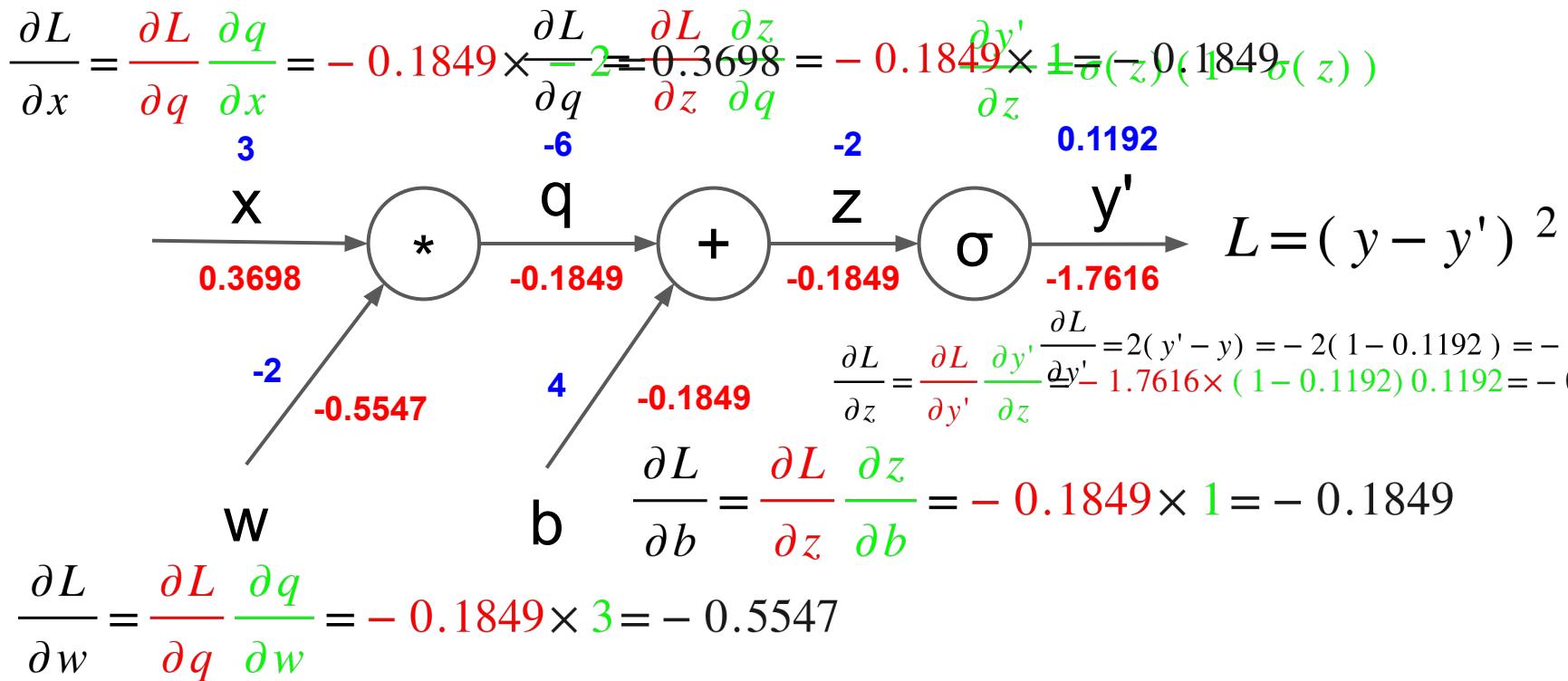
$$\frac{\partial L}{\partial x} = 4 \times 2 = 8$$

$$\frac{\partial L}{\partial y} = 4 \times -3 = -12$$

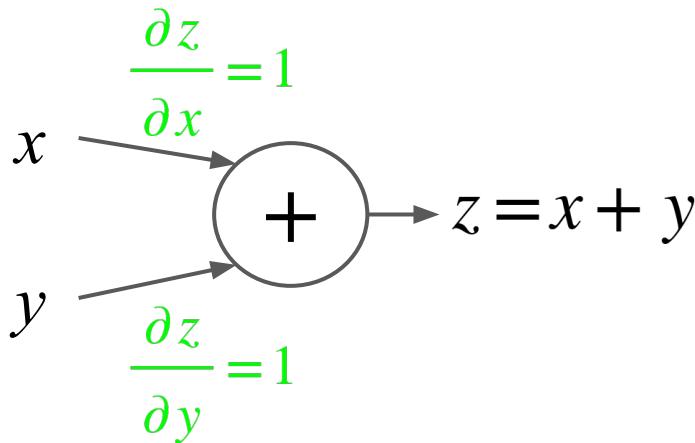


# Exemplo unidimensional e uma camada: $y' = \sigma(wx + b)$

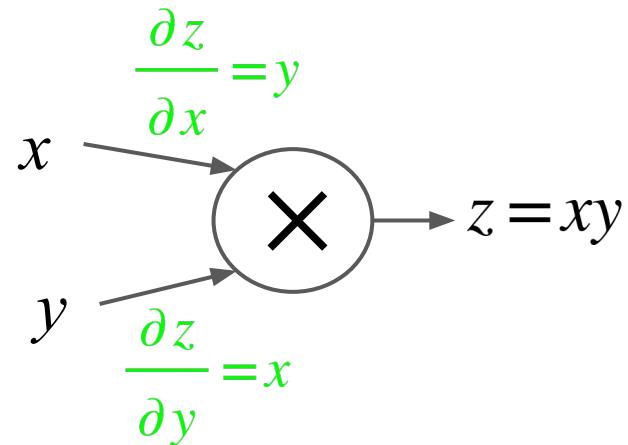
Exemplo de treino:  
 $x=3, y=1$



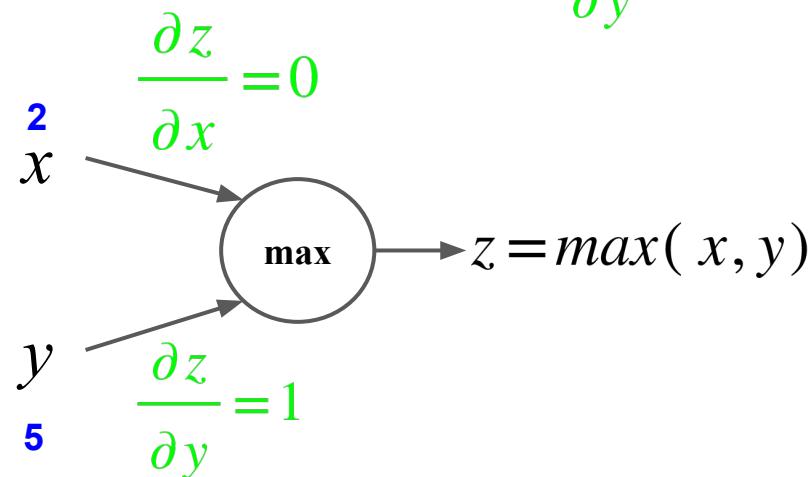
soma: "distribuidor"



multiplicação: "inversor" ?



max: "comutador/switch"



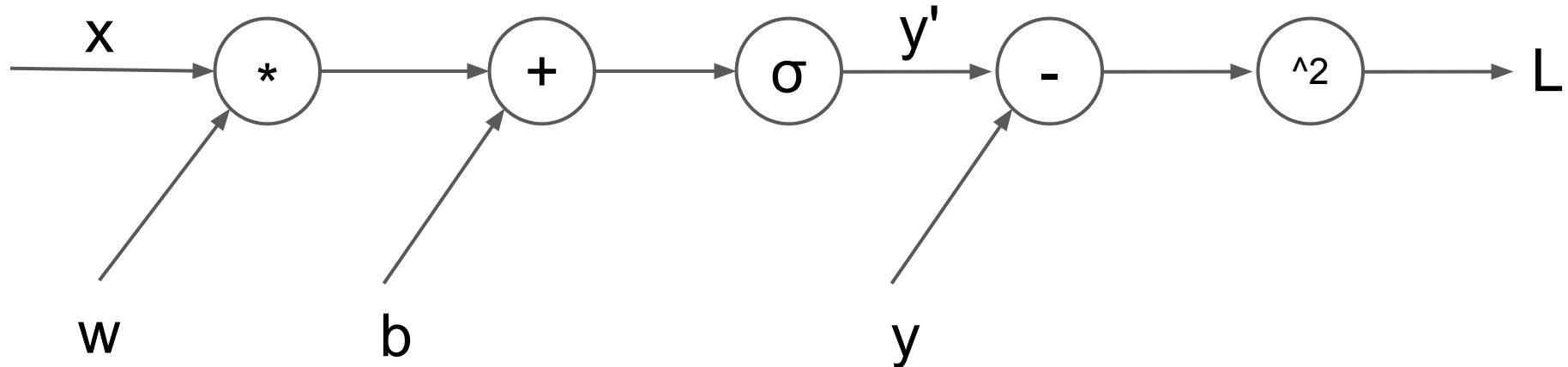
# Quando não dá para fazer backpropagation?

- Bugs quando criamos uma variável
- detach()

# Grafo computacional da loss de uma camada de rede neural

$$y' = \sigma(xw + b)$$

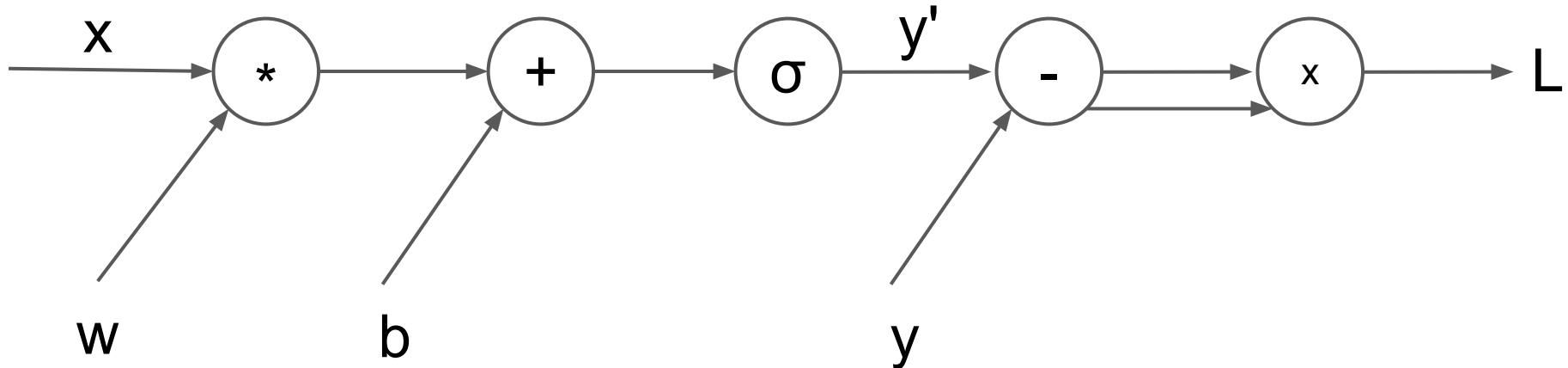
$$L = (y' - y)^2$$



# Grafo computacional da loss de uma camada de rede neural

$$y' = \sigma(xw + b)$$

$$L = (y' - y)^2$$



# Conclusão

- Backpropagation é eficiente:
  - custo forward  $O(N)$  + backward  $O(N) = O(N)$
- Localidade: gradientes de um nó são fáceis de se calcular
- Adicionar novos nós é fácil contanto que se saiba sua derivada local
- Backpropagation + Grafo Computacional = Lego do Aprendizado de Máquina



# Para estudar mais

Recomendo muito o vídeo do Andrej Karpathy:

- <https://www.youtube.com/watch?v=i94OvYb6noo>

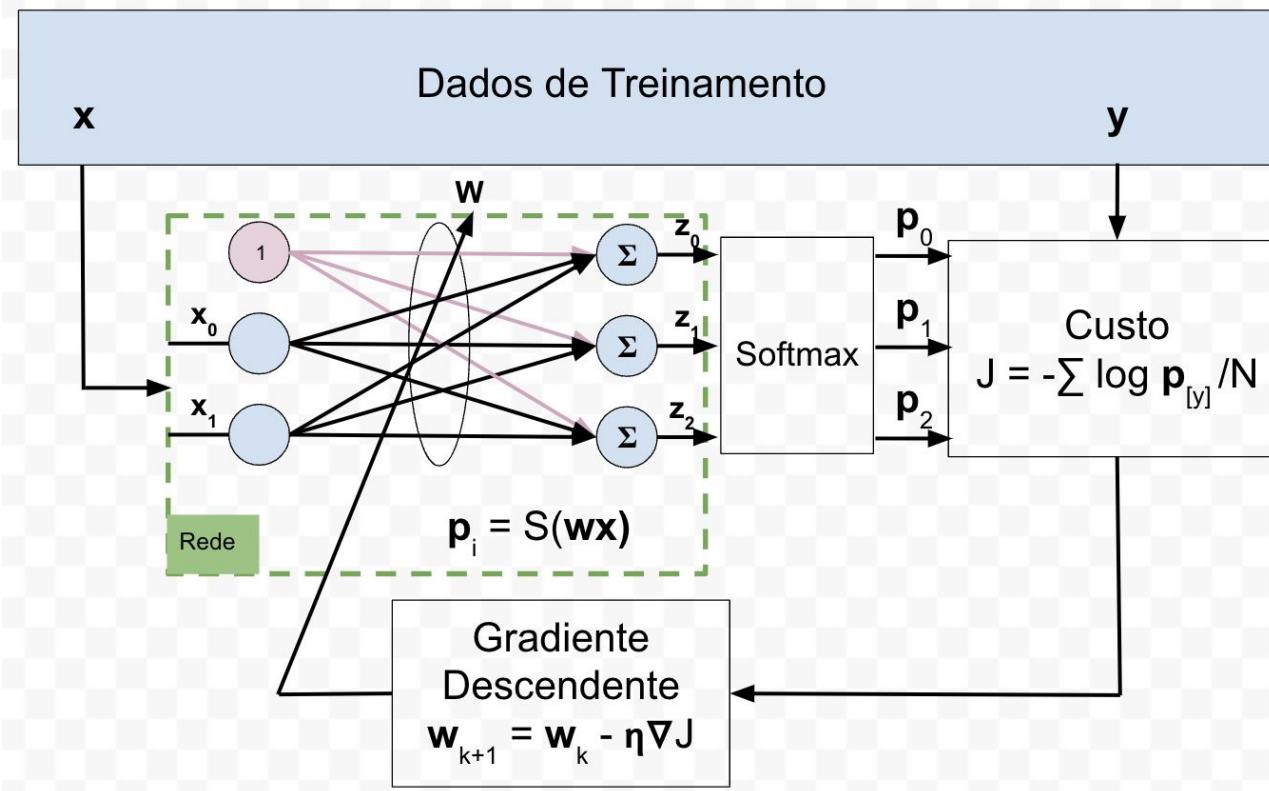
Exercícios da Classe de Stanford:

- <https://cs231n.github.io/>

# Assunto da quarta aula

- Regressão Logística (Classificador Softmax)
- Dataset e Dataloader do Pytorch
- Dados de Treinamento, Validação e Testes
- Correção de Exercícios da Aula
- Leitura próxima aula: Alexnet
- Exercício próxima aula:
  - batches

# Iris dataset - Regressão Logística - 3 classes



$z_i$ : logitos

$p_i$ : prob. classe i  
Perda: Entropia Cruzada

# Hot-encoding vs Soft Labels

classe	Y	Y_2	Y_1	Y_0
0: Red	0	0	0	1
1: Green	1	0	1	0
2: Blue	2	1	0	0

Hard Label  
One-Hot encoding

classe	Y_2	Y_1	Y_0
0: Red	3%	2%	95%
1: Green	5%	90%	5%
2: Blue	98%	1%	1%

Soft Label

# Entropia Cruzada com Hard labeling

$$H(p, q) = - \sum_i q_i \log(p_i)$$

p é a probabilidade da predição e q é a probabilidade do target (one-hot)

com hot encoding:  $q_i = 0$  para  $i \neq y$  onde  $y$  é a classe target ( $q_y = 1$ ). Assim:

$$H(p, y) = -\log(p_y)$$

onde  $y$  é o índice da classe target e  $p_y$  é a probabilidade predita da classe  $y$ .

# Entropia Cruzada a partir do Softmax dos Logitos

$$S(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$p_i = S(z_i)$$

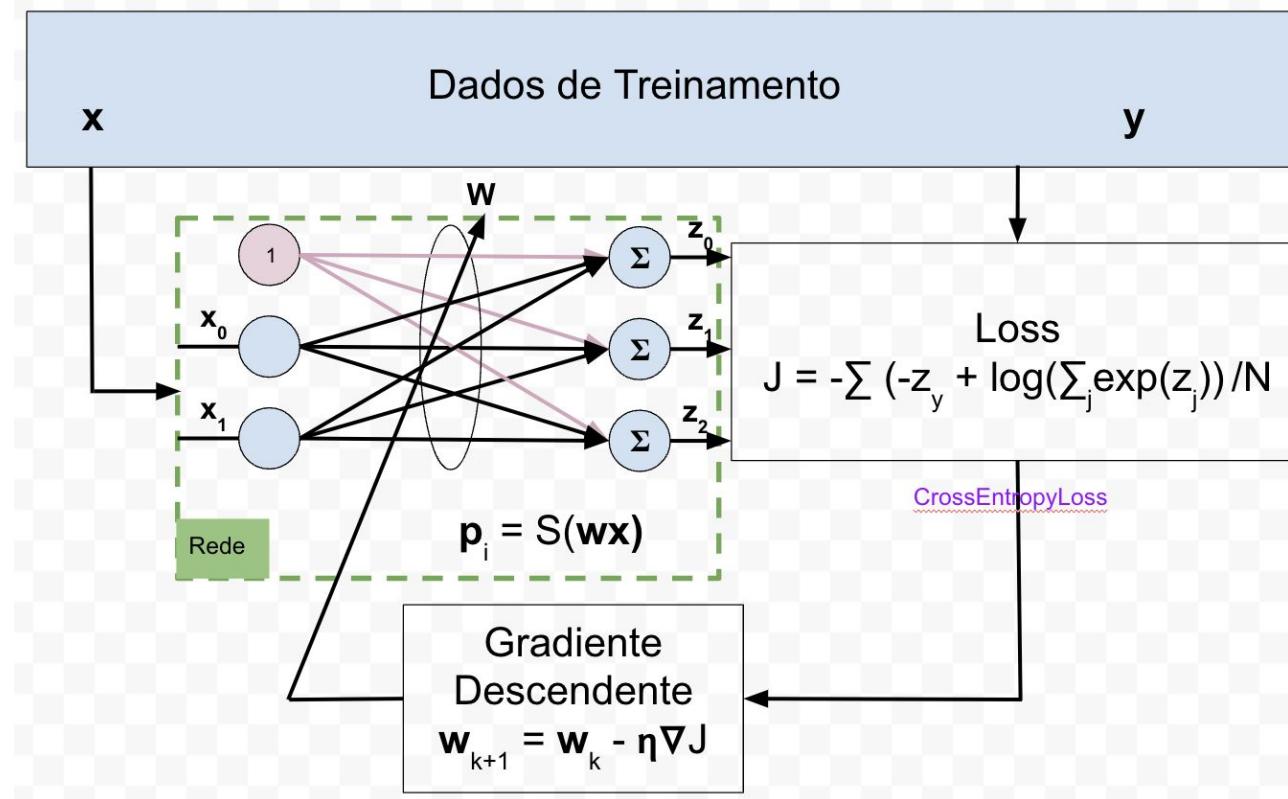
$$H(S(z), y) = -\log(S(z_y))$$

$$H(z, y) = -z_y + \log(\sum_j e^{z_j})$$

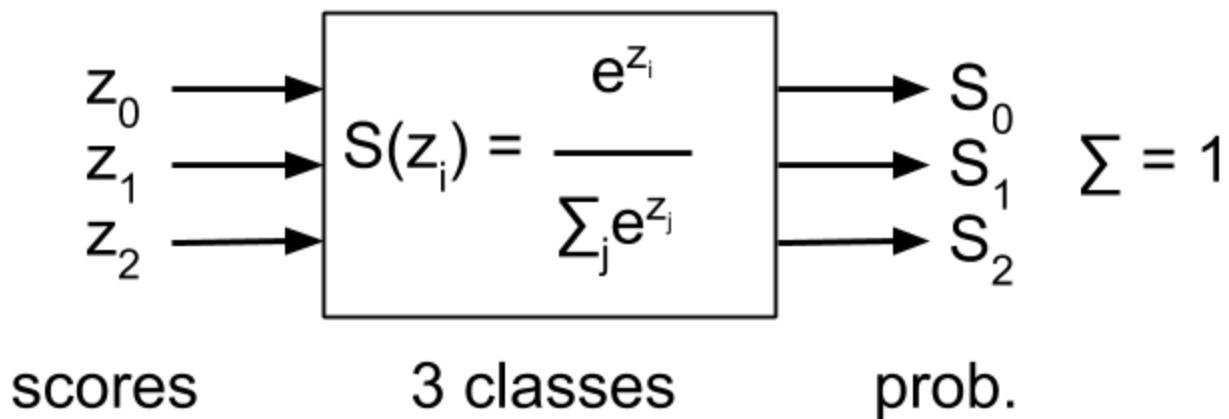
z: logitos  
y: índice do target

Pytorch:  $H(z,y)$  é `CrossEntropyLoss()`

# Uso do Pytorch: CrossEntropyLoss



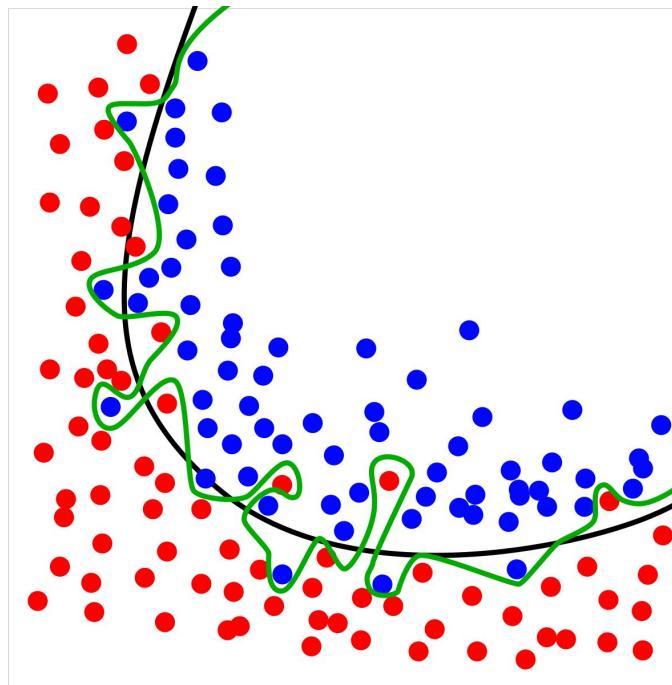
## SOFTMAX



# Opções de Loss de CrossEntropy no PyTorch para regressão logística (classificação)

	<b>Final da Rede</b>	<b>Criterion</b>
1 neurônio (2 classes)	Logitos	BCEWithLogitsLoss
	Sigmoid	BCE
2 ou mais neurônios (multiclasse)	Logitos	CrossEntropyLoss
	LogSoftmax	NLLoss

# Overfitting



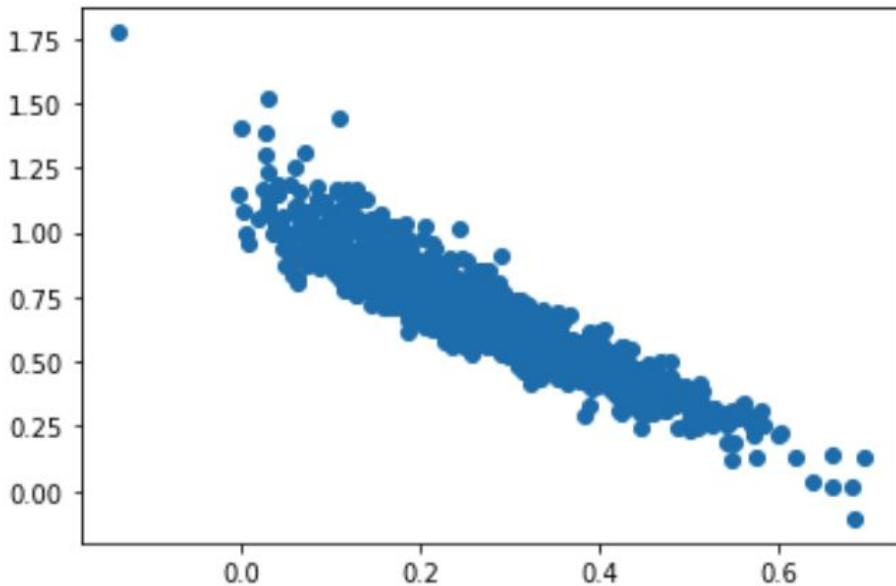
# Exercício teste

No exercício da regressão linear, existe um valor ótimo do mínimo da função de perda dado por:

**Valor ótimo, solução analítica**

$$\mathbf{w} = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{y}$$

Plotar os valores ótimos de  $(w_0, w_1)$  para cada batch utilizando o DataLoader.



# Extras

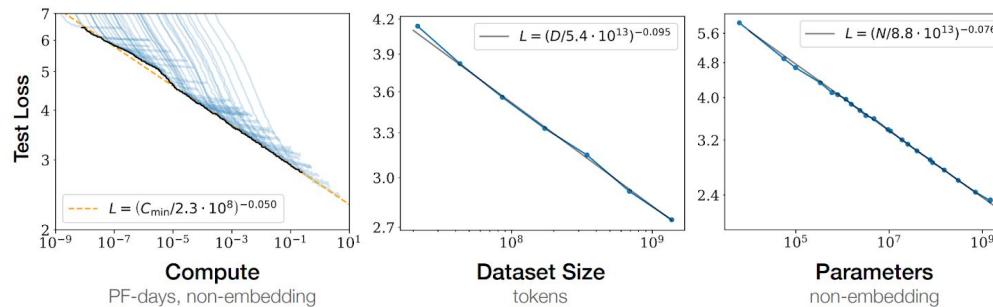
# Por que ler artigos de décadas atrás?

## Learning Internal Representations by Error Propagation (1985)

"The time to solve XOR is reduced linearly with the logarithm of the number of hidden units"

## Scaling Laws for Neural Language Models (2020)

"Large models are more sample-efficient than small models, reaching the same level of performance with fewer optimization steps."



# Assunto da quinta aula

- Ativações, redes convolucionais
- Correção de Exercícios da Aula
- Leitura próxima aula: Resnet
- Exercício próxima aula:
  - CNN

# Funções Não-lineares

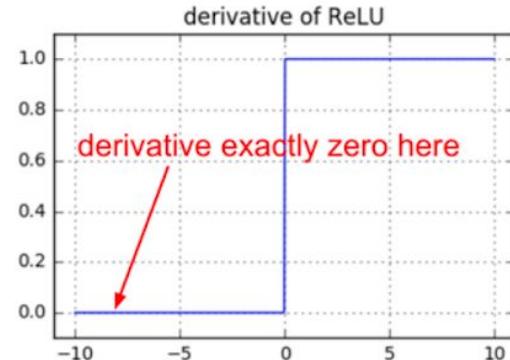
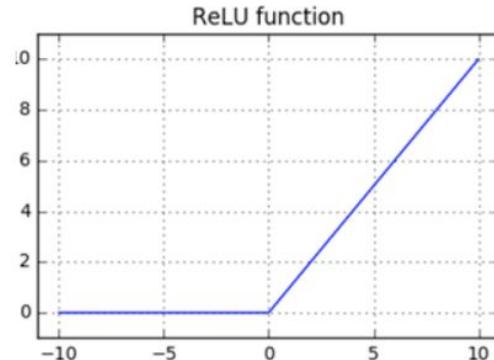
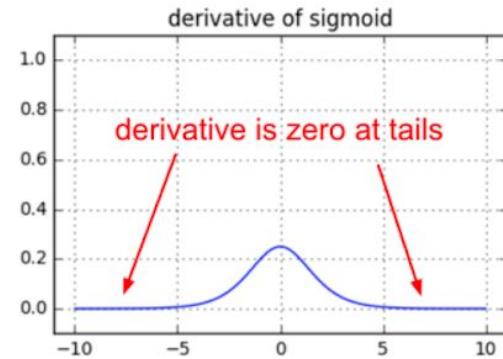
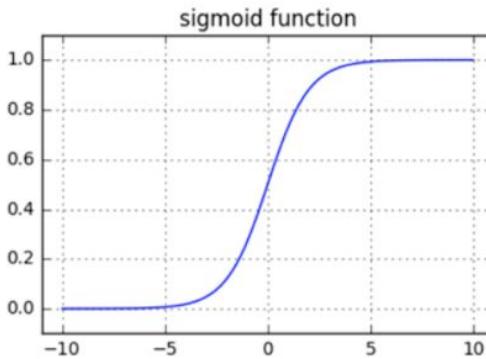
Rede neural com 2 camadas:

$$y' = w_2 f(w_1 x)$$

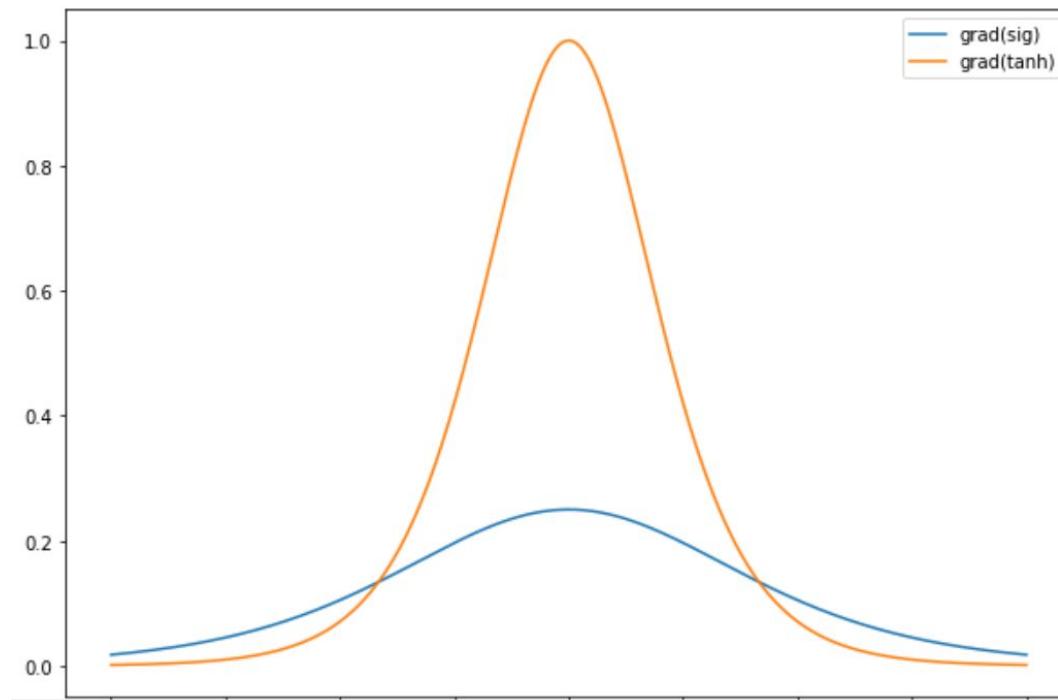
onde  $f$  é uma função não-linear

Uma rede com 2 camadas lineares é equivalente à uma rede com *uma* camada:

$$y' = w_2(w_1 x) = w_3 x$$



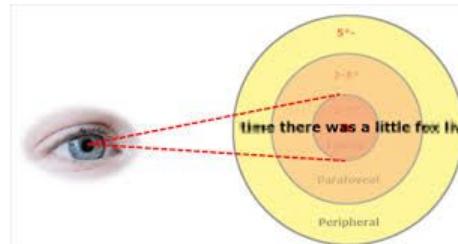
# Funções Não-lineares: comparação sigmoid e tanh



[Tensorflow Sandbox](#) para ilustrar as desvantagens de uso da sigmoid como função de ativação

# Two Fundamental Properties of Convolutional Layers

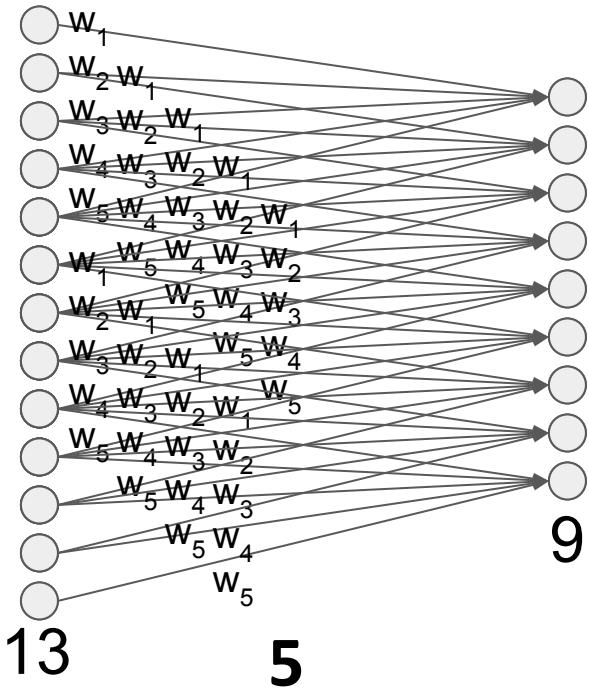
**Locality (Neighborhood)** - Far away weights  $\rightarrow$  zero



**Translation Invariance** - shared weights



# Convolucional Layer



**parâmetros**

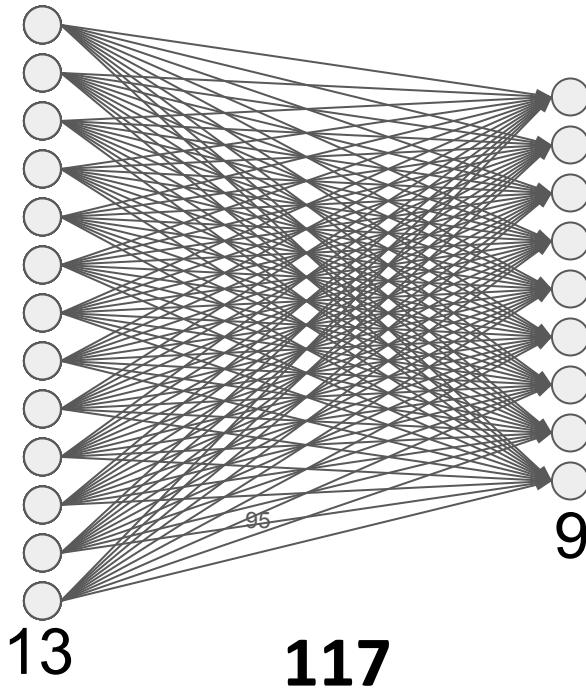
13

5

9

Imagen  
Vídeo  
Som  
Texto  
Séries  
temporais

# Linear Layer - Dense



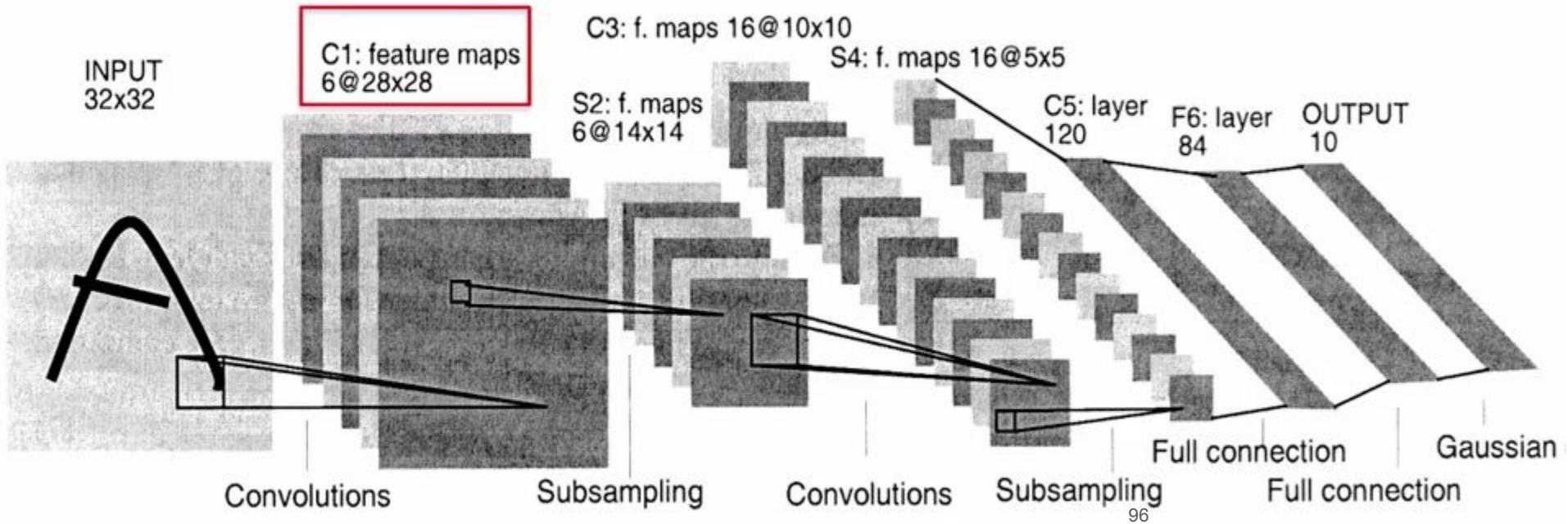
**parâmetros**

13

117

9

# The architecture of LeNet5

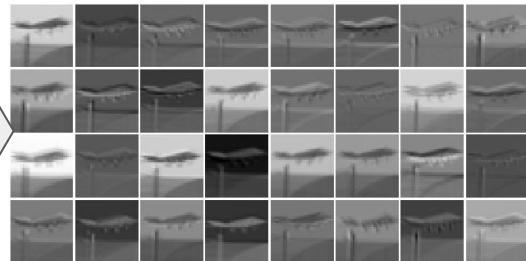


LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P.  
"Gradient-based learning applied to document recognition."  
*Proceedings of the IEEE* 86.11 (1998): 2278-2324.

17.5 K  
citations

# Feature Selection during training

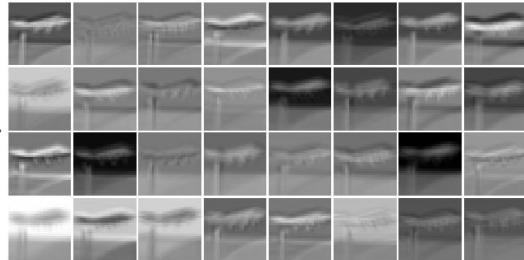
conv - 32 channels



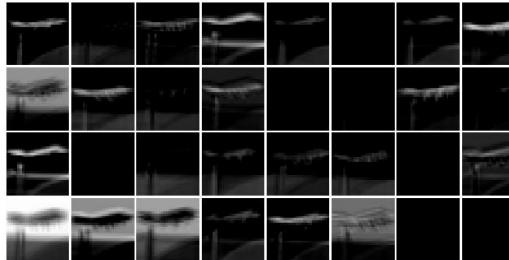
ReLU



$X'$

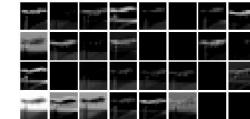


ReLU



conv - 32 channels

Features



97

subsampling  
max-pooling

# Assunto da sexta aula

- Correção de Exercícios da Aula 5
  - Implementação da Convolução
  - Leitura paper: Resnet
- Assuntos Novos:
  - Transfer Learning
  - Aumento de dados
  - BatchNorm/LayerNorm
  - Conexões residuais
  - Dropout
  - Overfit em um batch
- Leitura: [CLIP](#)
- Exercício: Implementação e Treinamento da Resnet no CIFAR-10

# Otimizadores do Gradiente Descendente

Alguns otimizadores:

- SGD normal
- SGD com momentum
- SGD com momentum e Nesterov
- RMSprop
- Adam

# Regularização L1 e L2

L1 Regularization

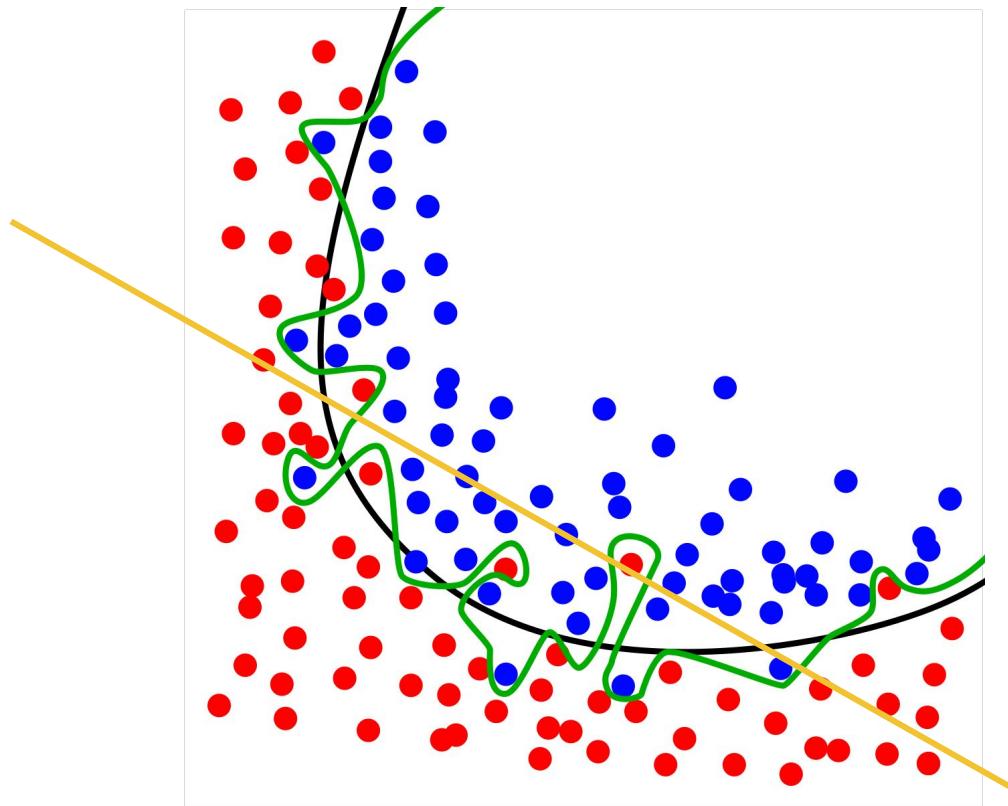
$$\text{Cost} = \sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2 + \lambda \sum_{j=0}^M |W_j|$$

L2 Regularization

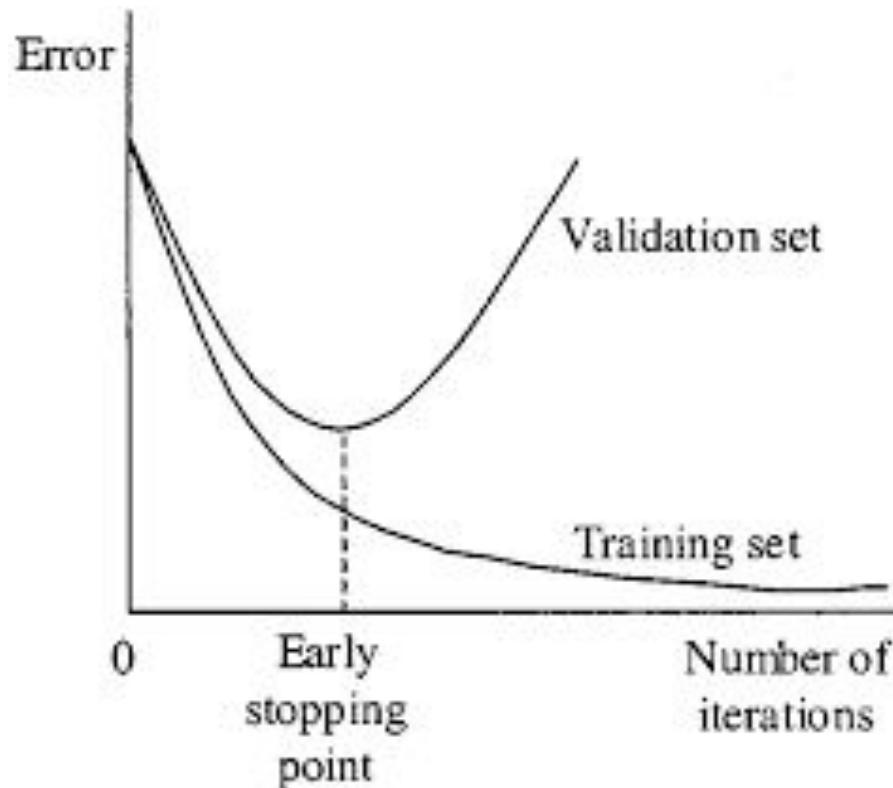
$$\text{Cost} = \sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2 + \lambda \sum_{j=0}^M W_j^2$$

Loss function                      Regularization Term

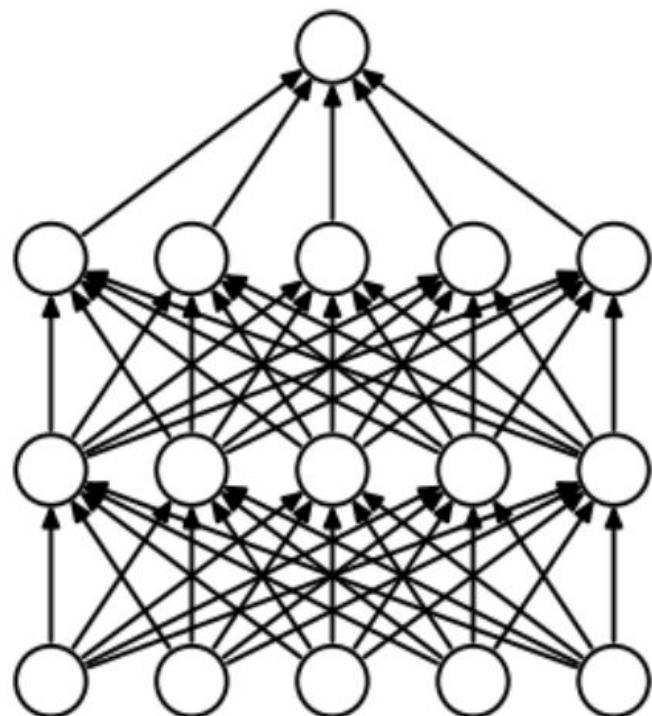
# Underfit e Overfit



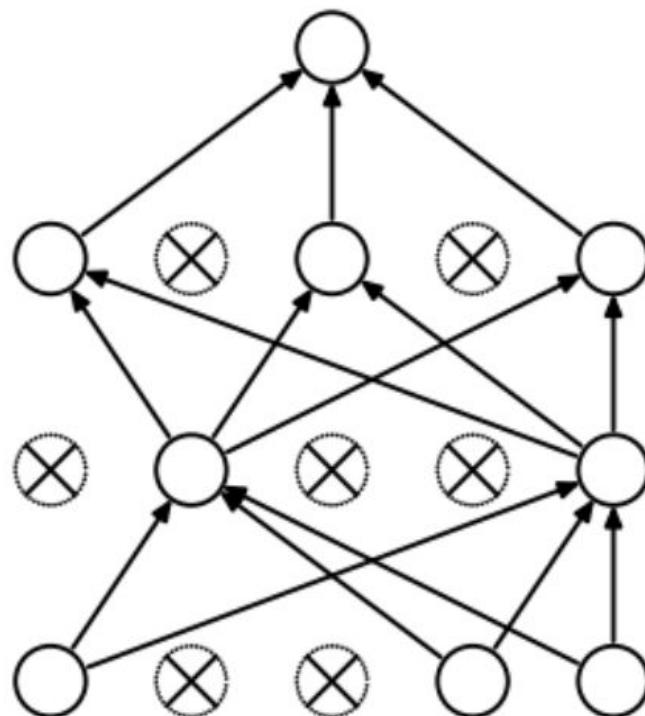
# Early stop



# Drop-out

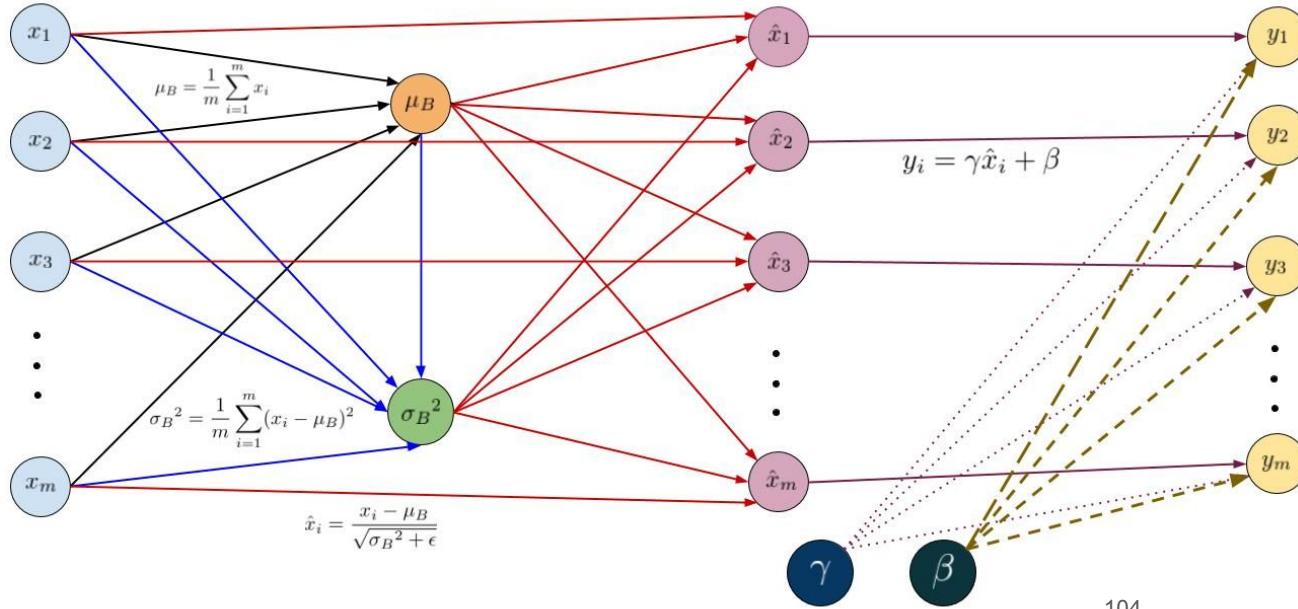


(a) Standard Neural Net



(b) After applying dropout.

# Batch Normalization - 2015



104

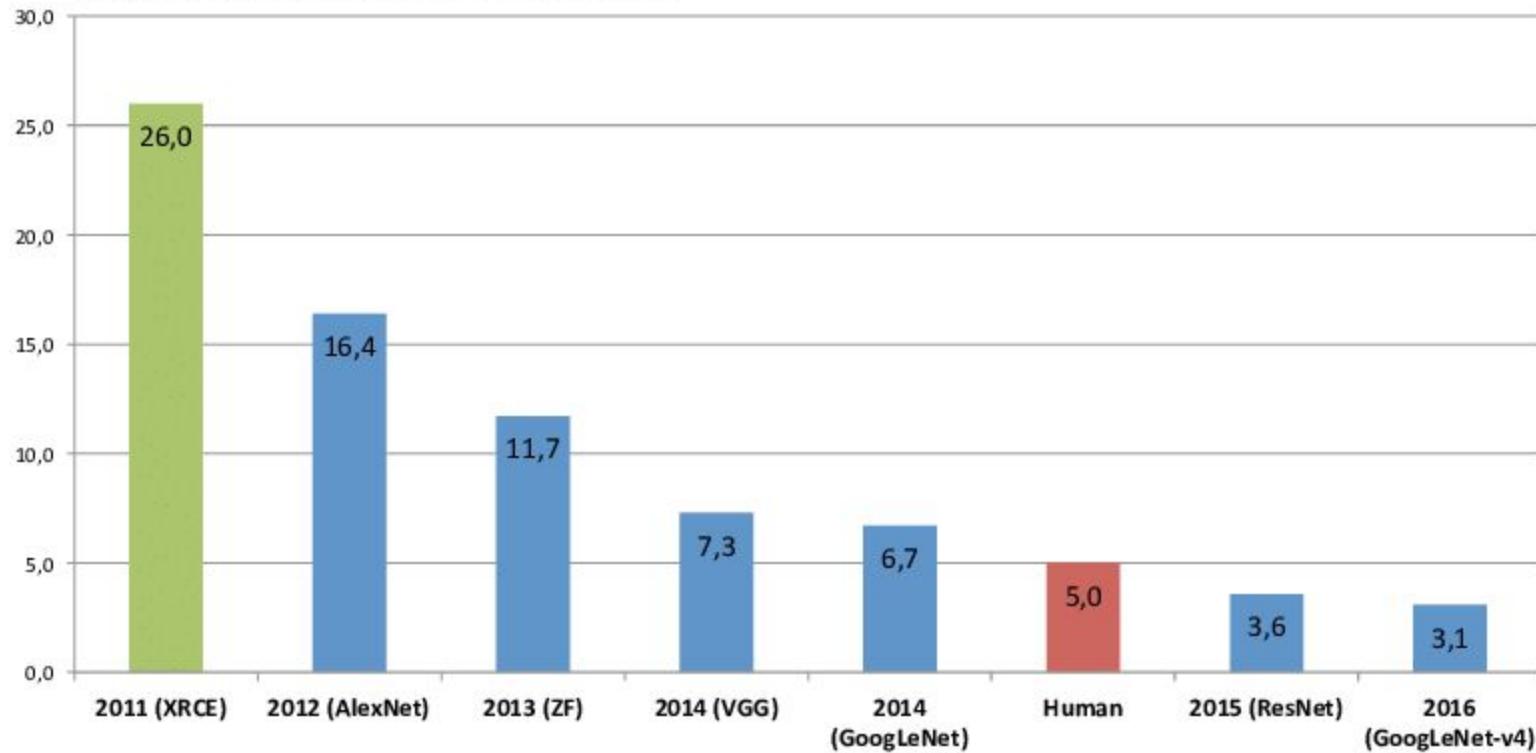
Much more stable in training convergence.

Ioffe, Sergey, and Christian Szegedy.

"Batch normalization: Accelerating deep network training by reducing internal covariate shift." *International Conference on Machine Learning*. 2015.

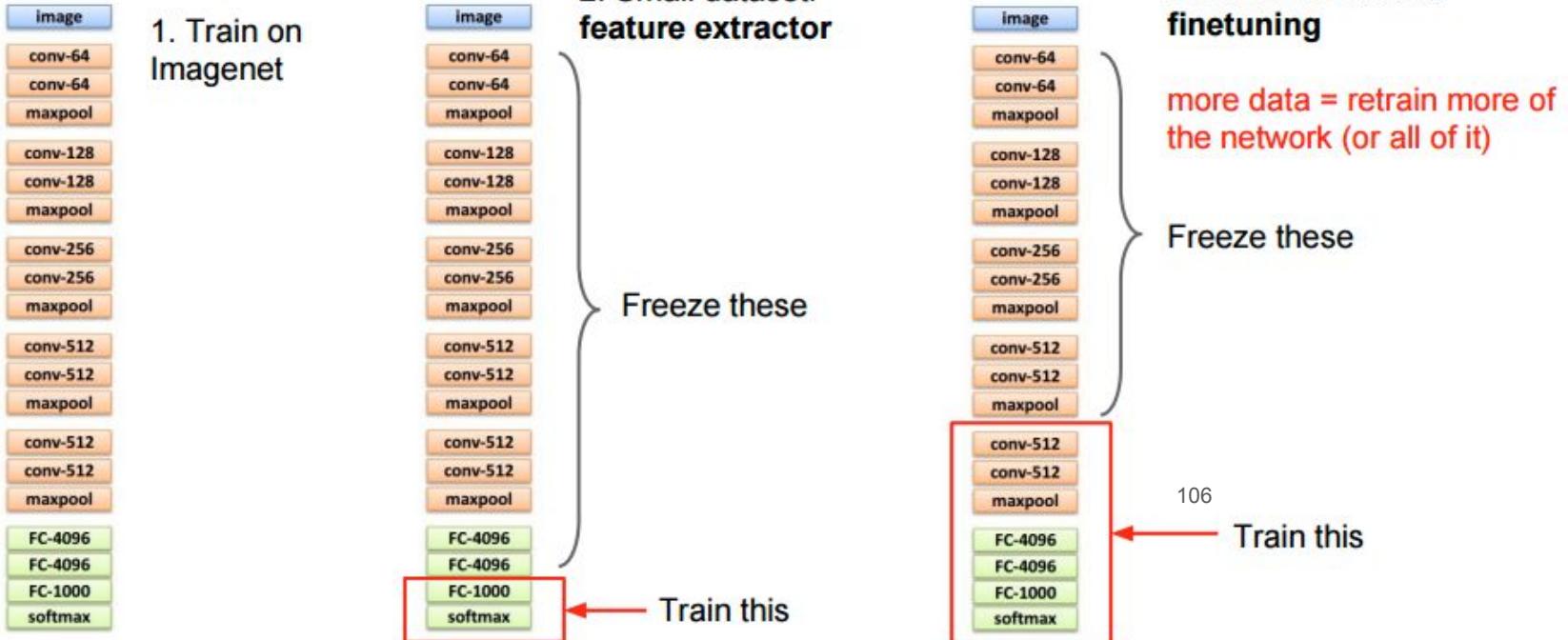
9.4K  
citations

## ImageNet Classification Error (Top 5)

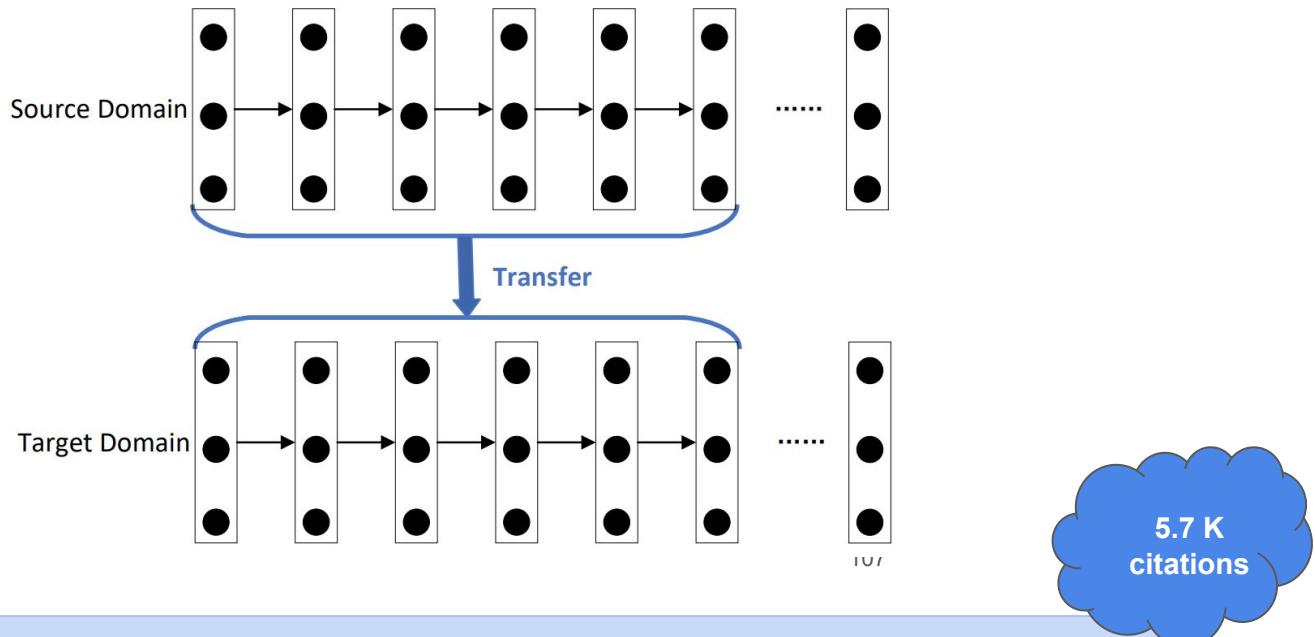


[https://colab.research.google.com/github/robertoalotufo/rnap/blob/master/PyTorch/Models\\_ImageNet.ipynb#scrollTo=fY3UfIEayFXV](https://colab.research.google.com/github/robertoalotufo/rnap/blob/master/PyTorch/Models_ImageNet.ipynb#scrollTo=fY3UfIEayFXV)

# Transfer Learning



# Transfer Learning and Domain Adaptation



Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2010): 1345-1359.

# Assunto da sétima aula

- Correção de Exercícios da Aula 6
  - Exercício: Implementação e Treinamento da Resnet no CIFAR-10
  - Leitura: CLIP
- Assuntos Novos:
  - Modelos de Linguagem
  - Perplexidade
- Leitura: [Bengio et al. 2003](#)
- Exercício: Treinar um modelo de linguagem simples no corpus BrWaC

# Modelagem da Linguagem

- Dado um texto, qual é a próxima palavra?
- Ex: Os ursos hibernam durante o \_\_\_\_\_
- Por que estudar isso?
  - Aplicação imediata: sugestor de palavras no celular, email, etc
  - Hipótese: se um agente consegue prever bem a próxima palavra, então ele sabe bastante sobre a linguagem e o mundo
  - Todos os modelos estado da arte (BERT, T5, etc) são pré treinados nesta tarefa.
- Demo GPT-3
- Como implementar?

# String -> Embeddings

Exemplo análise de sentimento

Conversão para embeddings:

-0.1	...	0.2	...	...	...	0.2	...	...
-0.7	...	-1.2	...	...	...	-1.2	...	...
0.5	...	0.9	...	...	...	0.9	...	...

Conversão para token ids:

2	130	0	90	155	5	0	873	20	...
---	-----	---	----	-----	---	---	-----	----	-----

Conversão para tokens:

I	enjoyed	the	first	half	of	the	movie	but	...
---	---------	-----	-------	------	----	-----	-------	-----	-----

Entrada (String):

I enjoyed the first half of the movie but ...

Matriz de Embeddings:

$V \times D$

$V$  = tamanho do vocabulário

$D$  = tamanho do vetor

id	vetor		
	0.2	-1.2	0.9
0	0.0	0.4	0.4
1	-0.1	-0.7	0.5
2	0.1	0.6	-1.0
3	...	...	...
	...	...	...

Como implementar `torch.nn.Embeddings` com operações matriciais:

Neste exemplo:  
vocab\_size=20  
context\_size=5

## Entrada:

w<sub>t-5</sub>

w<sub>t-4</sub>

$w_{t-3}$

$w_{t-2}$

$w_{t-1}$

## Índices:

7

3

6

8

20

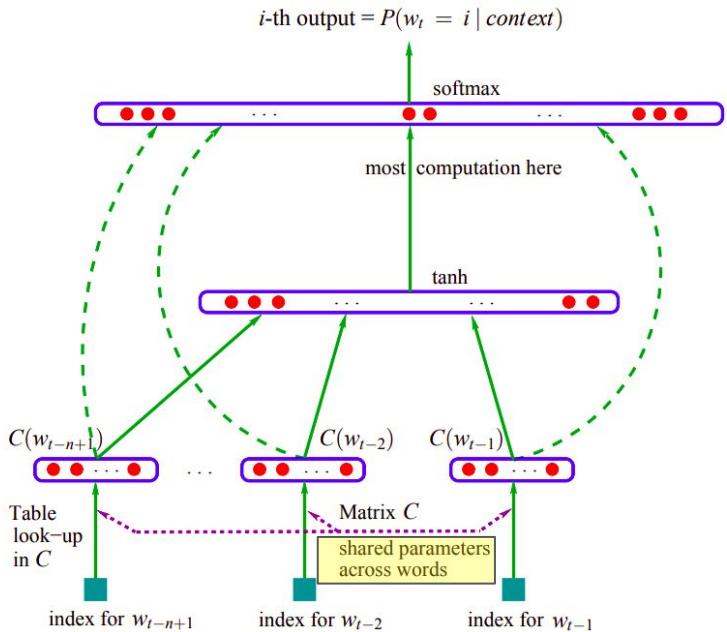
## Conversão de índices para matriz one-hot que

`q.shape`  
=(5x20):

```
x = Cq; x.shape=(5, embedding_dim); C.shape=(20, embedding_dim)
```

Saída: x.reshape(-1); x.shape=(5\*embedding\_dim)

# Uma modelo de linguagem simples



Exemplo (sem batch)

1. Entrada: ids das n últimas palavras: shape = [n]
2. Converter ids para embeddings: shape = [n, embedding\_dim]
3. Concatenar todos os embeddings: shape = [n \* embedding\_dim]
4. Passar na primeira camada: shape = [hidden\_size]
5. Passar na segunda camada (logits): shape = [vocab\_size]
6. Aplicar softmax para obter a probabilidade da palavra correta

Dica: usar ReLU ao invés de Tanh como função de ativação.

Figure 1: Neural architecture:  $f(i, w_{t-1}, \dots, w_{t-n+1}) = g(i, C(w_{t-1}), \dots, C(w_{t-n+1}))$  where  $g$  is the neural network and  $C(i)$  is the  $i$ -th word feature vector.

# Arquitetura Simplificada de Bengio et. al (2003)

Os pesos da rede são:

C: (vocab\_size, embedding\_dim)

H: (hidden\_size, context\_size \*  
embedding\_dim)

d: (hidden\_size)

U: (vocab\_size, hidden\_size)

b: (vocab\_size)

```
probs = softmax(logits); probs.shape = vocab_size
```

```
logits = Uz + b; logits.shape = vocab_size
```

```
z = tanh(Hx + d); z.shape = hidden_size
```

```
x = C(wt-5) | ... | C(wt-1); shape = 5 * embedding_dim
```

C(w<sub>t-5</sub>)

C(w<sub>t-4</sub>)

C(w<sub>t-3</sub>)

C(w<sub>t-2</sub>)

C(w<sub>t-1</sub>)

Conversão de índices das palavras para embeddings (table lookup)

Assumindo context\_size=5

w<sub>t-5</sub>

w<sub>t-4</sub>

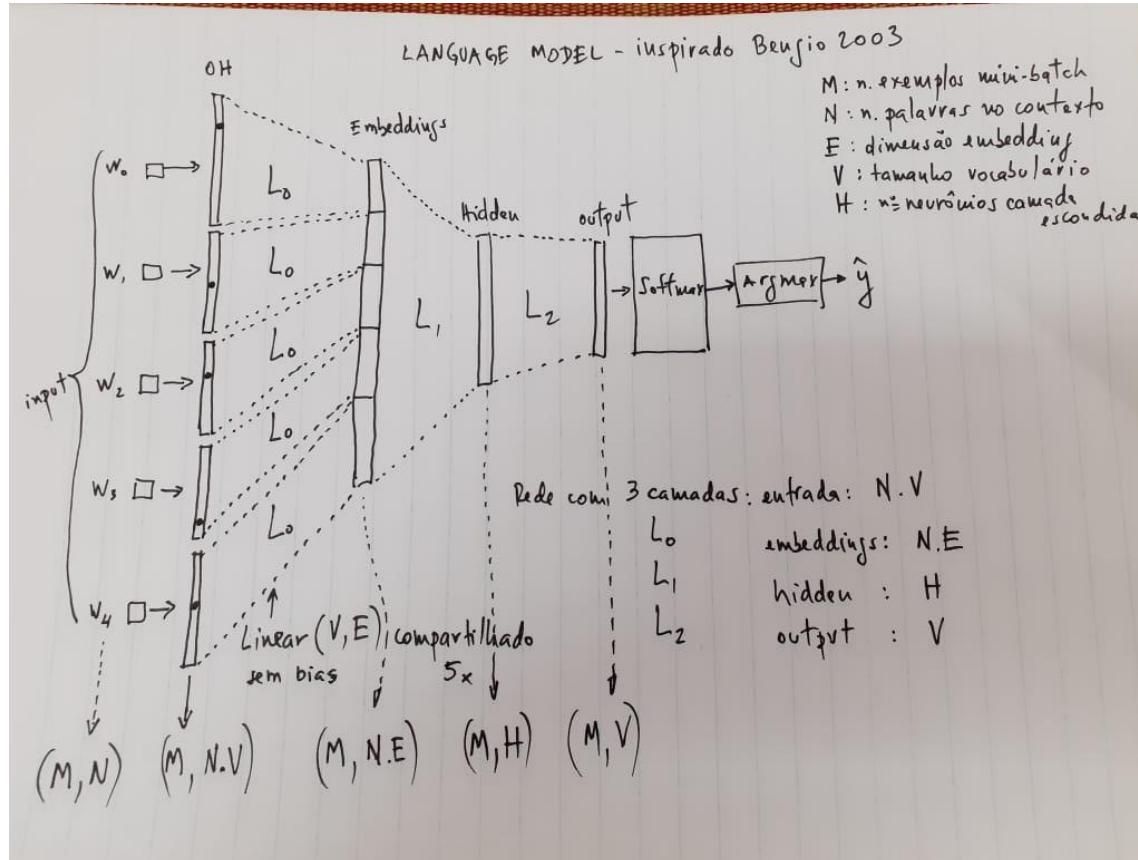
w<sub>t-3</sub>

w<sub>t-2</sub>

w<sub>t-1</sub>



# Outra representação da arquitetura simplificada



# Avaliação de Modelos de Linguagem

- Acurácia?
  - 1 caso o modelo acertou a palavra, 0 caso contrário
  - "Nesta garagem cabem dois \_\_\_\_."
  - Predição do modelo:  $P(\text{"carros"}) = 0.90$ ,  $P(\text{"automoveis"}) = 0.08$ , ...
  - Palavra original: "automóveis"
  - Acurácia = 0
- Erro Quadrático Médio (MSE)?
  - $\text{MSE} = \sum_i [P_{\text{truth}} - P_{\text{pred}}(w_i)]^2$
  - No exemplo: se  $P(\text{automóveis}) = 0.9$ , então,  $\text{MSE} = 0.01$ , assumindo  $P_{\text{truth}} = 1$ .
  - Funcionaria, mas seria bom usarmos uma métrica que penaliza mais quando o modelo colocou uma probabilidade muito baixa na palavra correta.

# Avaliação de Modelos de Linguagem

Dan Jurafsky



## Perplexity

The best language model is one that best predicts an unseen test set

- Gives the highest  $P(\text{sentence})$

Perplexity is the inverse probability of the test set, normalized by the number of words:

Chain rule:

For bigrams:

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}}$$

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$$

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_{i-1})}}$$

Minimizing perplexity is the same as maximizing probability

# Avaliação de Modelos de Linguagem

- Perplexidade (PPL) e Entropia Cruzada (ENT):
  - $PPL = e^{XENT}$
  - $XENT = - \frac{1}{n} \sum_n \ln P(w_n)$
- Sentença S: "Eu gosto de pizza"
- $XENT(S) = \frac{1}{n} [-\ln P("Eu" | vazio) - \ln P("gosto" | "Eu") - \ln P("de" | "Eu gosto") - \ln P("pizza" | "Eu gosto de"))]$
- Métrica: por que usar PPL ao invés de entropia cruzada?
  - possível resposta:  
<https://towardsdatascience.com/perplexity-intuition-and-derivation-105dd481c8f3>
- Usar PPL como função de custo?
- Sugestão de leitura: Meena (Google Chatbot) <https://arxiv.org/pdf/2001.09977.pdf>

Em diálogos, perplexidade e avaliação humana tem forte correlação

# Propriedades interessantes da Perplexidade

Qual é a perplexidade inicial de um modelo de linguagem com 10k palavras no vocabulário?

$$\text{Prob média} = 1/10k = 10^{-4}$$

$$XENT = - \frac{1}{n} \sum_n \ln P(w_n) = - \frac{1}{n} \sum_n \ln 10^{-4} \sim -\ln 10^{-4}$$

$$PPL = \exp^{XENT} = \exp(-\ln 10^{-4}) = \exp(\ln 10^4) = 10^4$$

A perplexidade é igual ao número de palavras no vocabulário!

Ou seja, se o seu modelo de linguagem tem perplexidade de N, significa que ele é igual a uma classificador aleatório de N classes

# Perplexidade

log base 2 ou ln?

# Modelos de linguagem como compressores de texto

~~Se um modelo de linguagem tem perplexidade de 32, é como se ele fosse um modelo inicializado aleatoriamente e existissem 32 palavras únicas no corpus.~~

Assim, precisamos de 5 bits para representar a 32 palavras únicas ( $\log_2 32 = 5$ )

Ou seja, cada palavra no texto agora pode ser representada por 5 bits.

Se não tivéssemos usado um modelo de linguagem, precisaríamos de pelo menos:

$\log_2 10k = 13.3$  bits para representar cada palavra

Ou seja, o modelo压缩iu o texto de 13.3 bits/palavra para 5 bits/palavra!

# Subwords

- Problema: vocabulário grande -> grande matriz de embeddings
- Palavras raras tem embeddings ruins
- Usar caracteres ao invés de palavras?
  - Funciona, mas modelo precisa combinar embeddings de caracteres para "formar" a palavra.
- Compromisso: substituir palavras por *sub-palavras*
- carro = carro
- carros = carro ##s
- carrossel = carro ##ssel

Como funciona:

[Neural Machine Translation of Rare Words with Subword Units](#)

[Exemplo de uso no Colab](#)  
[Tutorial Tokenizers - HuggingFace](#)

Vocabulário final:

a

b

c

....

carro

##s

##ssel

....

##ing

# Assunto da oitava aula

- Correção de Exercícios da Aula 7
  - Exercício: Treinamento de um modelo de linguagem simples no corpus BrWaC
  - Leitura: [Bengio et al. 2003](#)
- Assuntos Novos:
  - Transformer (decoder-only)
- Leitura: [Attention is All you Need](#)
- Exercício: Treinamento de um modelo decoder-only do transformer no corpus BrWaC

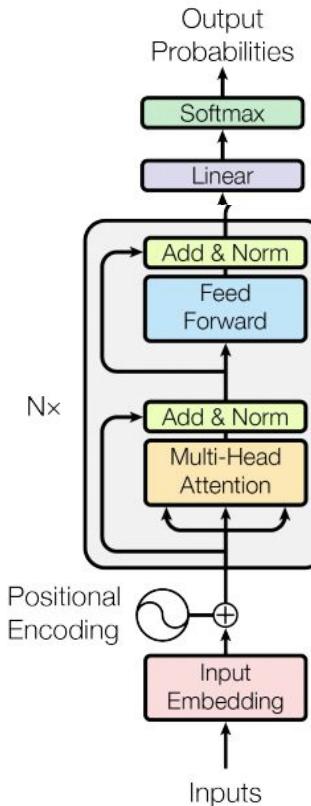
# Auto-atenção: Transformer

Modelo seq2seq (entra texto, sai texto)  
publicado em 2017 (Leitura desta semana)

Questionava a necessidade de recorrência  
para formar embeddings (RNN/LSTM)

Pequenos ganhos em tradução automática

2022: modelos PLN estado da arte são  
transformers (com poucas modificações na  
arquitetura original)



# Exemplo: uma camada de auto-atenção

Auto-atenção apenas para  $w_3$

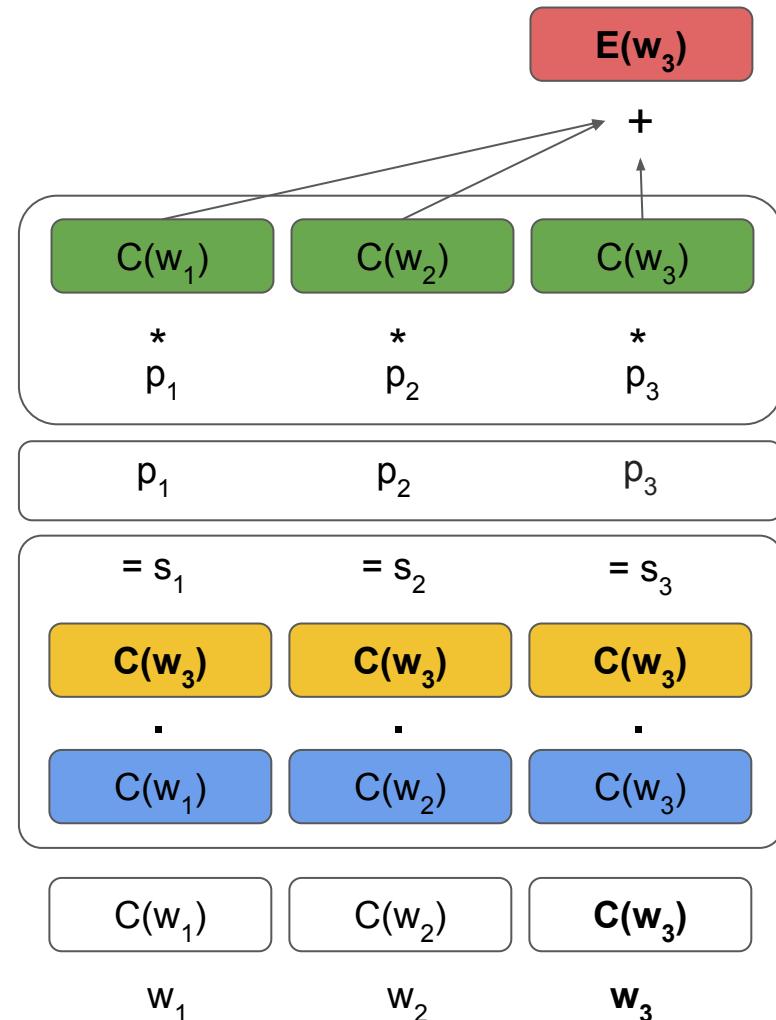
Soma ponderada dos embeddings:

$$E(w_i) = \sum p_n C(w_n)$$

$$p_i = \text{softmax}(s_i) = \exp(s_i) / \sum_n \exp(s_n)$$

Produto Escalar:  
Mede similaridade entre vetor corrente  $C(w_2)$  e outros vetores

embeddings:



```
x_embed[:, -1, :].unsqueeze(1)
```

```
x_embed[:, -2:-1, :]
```

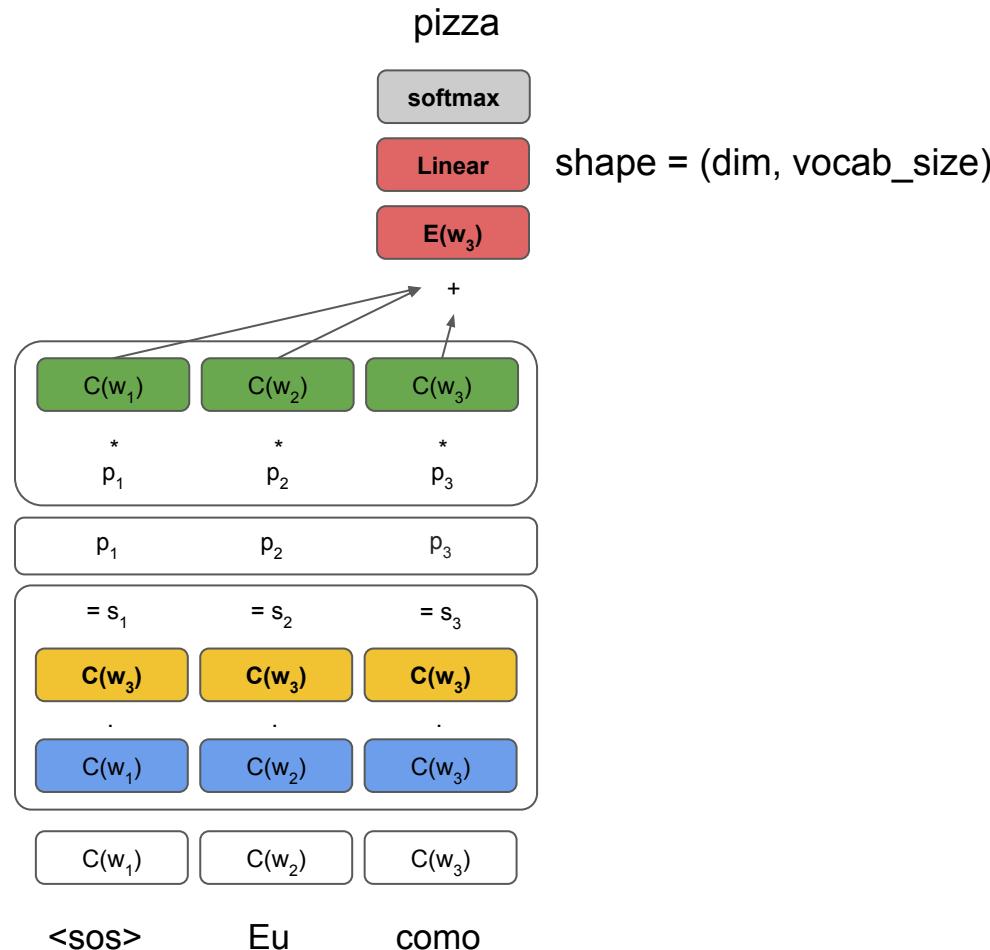
```
hidden =  
torch.relu(self.layer1(X_emb.reshape(len(inputs),-1)))
```

```
X_emb.shape = batch_size, context_size, embedding_dim
```

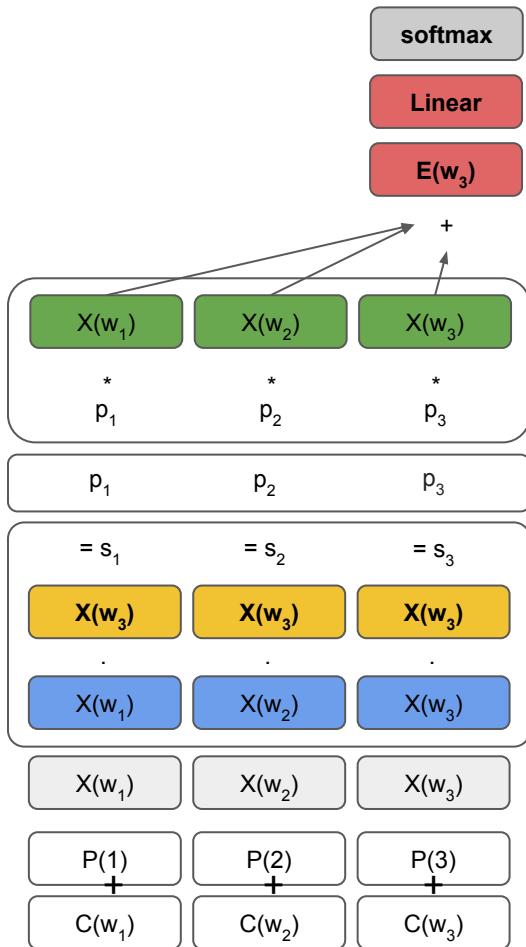
```
layer1.weight.shape = embedding_dim, embedding_dim
```

```
hidden = torch.relu(self.layer1(X_emb[:, -1, :]))
```

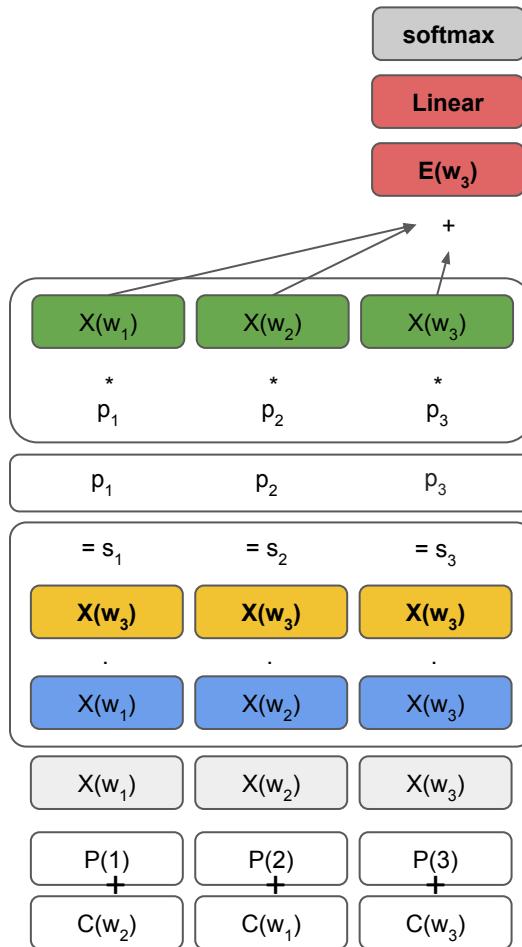
# Modelo de linguagem com 1 camada de auto-atenção



pizza



pizza



Source: Vaswani et al. (2017)

# pseudo-código do modelo de linguagem com 1 camada de auto-atenção

Forma de loop:

```
token_ids = [w1, w2, w3]  
q = w3  
scores = []  
for k in token_ids:  
    score = matmul(C(q), C(k)T)  
    scores.append(score)  
probs = softmax(scores)  
  
E = 0  
for v, p in zip(token_ids, probs):  
    E += C(v) * p  
logits = linear(E)  
probs = softmax(logits)  
token_id = probs.argmax()  
  
# em treino  
loss = -log (probs[w4])
```

Forma matricial:

```
token_ids = [w1, w2, w3]  
# L=3  
X = [C(token_ids)] # X.shape = L, D  
  
Q = X[-1] # 1,D  
K = V = X # L,D  
  
scores = matmul(Q, KT) # shape = 1,L  
probs = softmax(scores, dim=-1) # 1,L  
E = matmul(probs, V) # shape = 1,D  
logits = linear(E) # shape = 1,V  
probs = softmax(logits) # shape = 1,V  
token_id = probs.argmax() # shape = 1,1
```

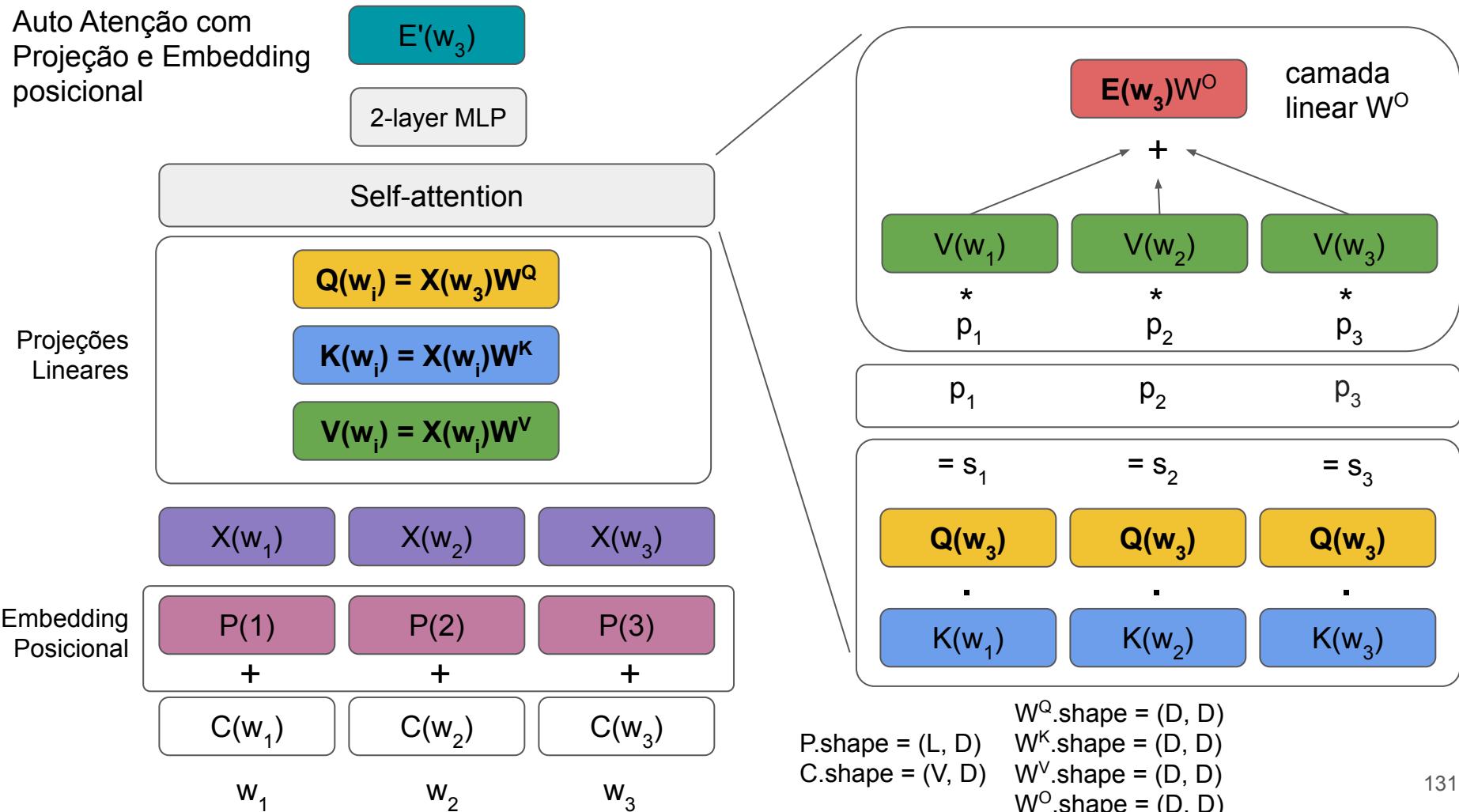
## Self-attention

- Problema: computação e memória crescem quadraticamente com o número de tokens da entrada ( $L^2$ )
  - Comparação par a par
  - Vantagem: todo token tem "acesso" a todos os outros da sequência
- Quantos pesos/parâmetros tem a camada de self-attention?
- Self-attention não leva em conta a posição das palavras:

```
E = self_attention(w1, w2, w3)
    = self_attention(w3, w2, w1)
```

  - Iremos ver na próxima aula uma forma simples de adicionar essa informação na entrada.

# Auto Atenção com Projeção e Embedding posicional



# Auto-atenção com projeções lineares $W^Q$ , $W^K$ , $W^V$ , $W^O$

# Forma de loop:

```
seq = [X(w1), X(w2), X(w3)]

q = X(w3)WQ
scores = []
for xk in seq:
    k = xkWK
    score = matmul(q, kT)
    scores.append(score)
probs = softmax(scores)

e = 0
for xv, p in zip(seq, probs):
    v = xvWV
    e += v * p
e = eWO
logits = linear(e)
probs = softmax(logits)
token_id = probs.argmax()
```

Forma matricial:

```
X = stack(X(w1), X(w2), X(w3))
# X.shape = L, D
```

```
Q = X[-1]WQ # Q.shape = 1, D
K, V = XWK, XWV # K/V.shape = L, D
```

```
scores = matmul(Q, KT) # shape = 1, L
probs = softmax(scores, dim=-1) # 1, L
E = matmul(probs, V) # shape = 1, D
E = EWO # shape = 1, D
logits = linear(E) # shape = 1, V
probs = softmax(logits) # shape = 1, V
token_id = probs.argmax() # shape = 1, 1
```

Quantos pesos tem uma camada de auto-atenção com projeções lineares, matriz de embeddings posicionais P, e matriz de embeddings de palavras C? (vide slide 133)

Assumindo:

B = Batch size

V = Tamanho do vocabulário

L = Tamanho máximo da sequência

D = Dimensão de todos os vetores (embeddings)

$$W^Q \ W^K \ W^V \ W^O = 4 \times D^2$$

cada uma tem shape = (D,D) =  $D^2$

P.shape = L, D = L\*D

C.shape = V, D = V\*D

Total =  $4D^2 + LD + VD$

# Normalização dos escores de atenção (Scaled dot-product)

- Produto escalar normalizado:
  - Conforme aumenta-se a dimensão dos vetores ( $d_k$ ) a variância do produto escalar aumenta, deixando o treinamento instável
  - Exemplo: se  $q$  e  $k$  são vetores de dimensão  $d_k$  com média 0 e variância 1, o produto escalar ( $\sum_i q_i k_i$ ) terá média zero e variância  $d_k$ .
  - Solução: normalizar escores de atenção pela raiz quadrada de  $d_k$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

# Conexões Residuais

- Também chamadas de *skip connections*
- Prática antiga, especialmente em redes recorrentes
- Se popularizou com a Resnet (He et al., 2015) →
- Intuição: cada camada fica restrita a apenas *adicionar* vetores às ativações anteriores.
- São "atalhos": gradientes podem pular camadas
- Possibilita treinamento de redes com muitas camadas (+300).
- Na prática:

```
def residual_layer(self, x):  
    residual = x  
    x = self.linear(x)  
    x += residual  
    return x
```

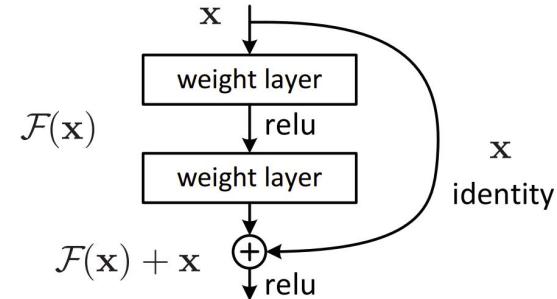
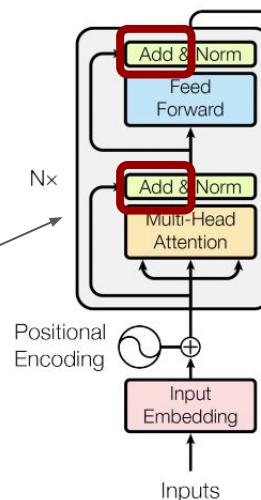


Figure 2. Residual learning: a building block.



# Assunto da nona aula

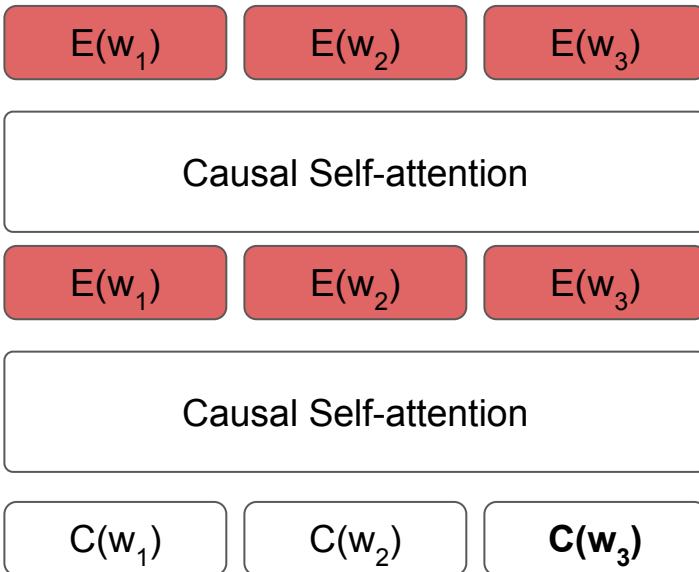
- Correção de Exercícios da Aula 8
  - Leitura: [Attention is All you Need](#)
  - Exercício: Treinamento de um modelo decoder-only do transformer no corpus BrWaC
- Assuntos Novos:
  - Transformer (decoder-only)
- Leitura: [GPT-3](#)
- Exercício: Treinamento de um modelo decoder-only com duas camadas de auto-atenção no corpus BrWaC

Quantos parametros tem o modelo de linguagem com uma camada de auto-atenção implementado no exercicio da aula 8?

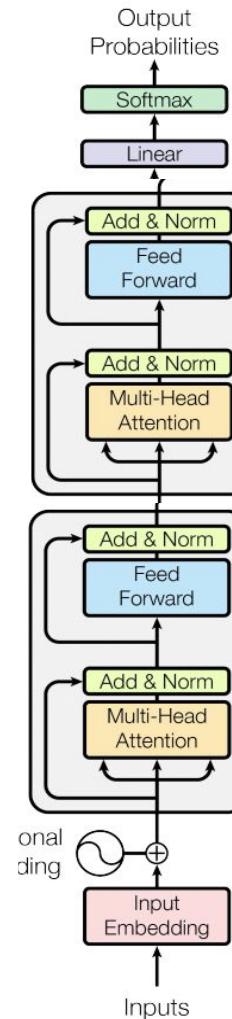
Para sua resposta, considere que o modelo tem apenas:

- 1 camada de embeddings posicionais P:  $\text{context\_size} * \text{embedding\_dim}$
- 1 camada de embeddings de contexto C:  $\text{vocab\_size} * \text{embedding\_dim}$
- 4 camadas lineares  $W_q, W_k, W_v, W_o$ :  
 $4 * \text{embedding\_dim} * \text{embedding\_dim}$
- 1 camada de classificação (que solta o logitos):  $\text{embedding\_dim} * \text{vocab\_size}$

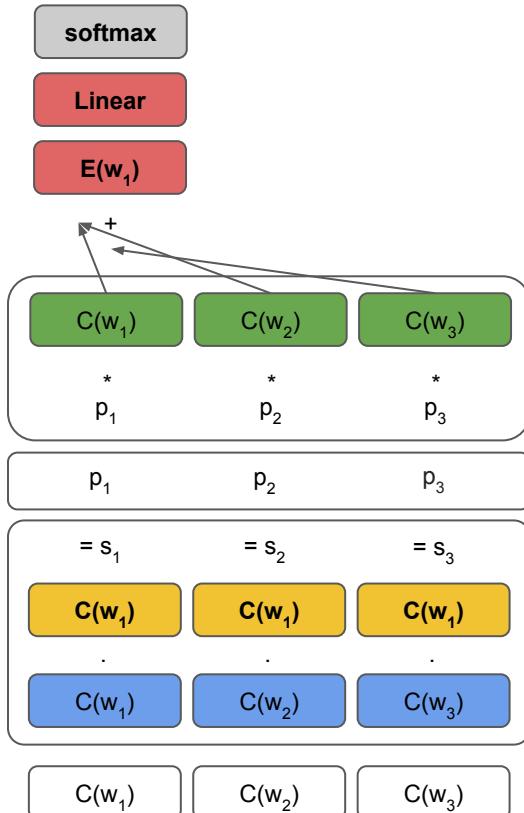
# Múltiplas camadas



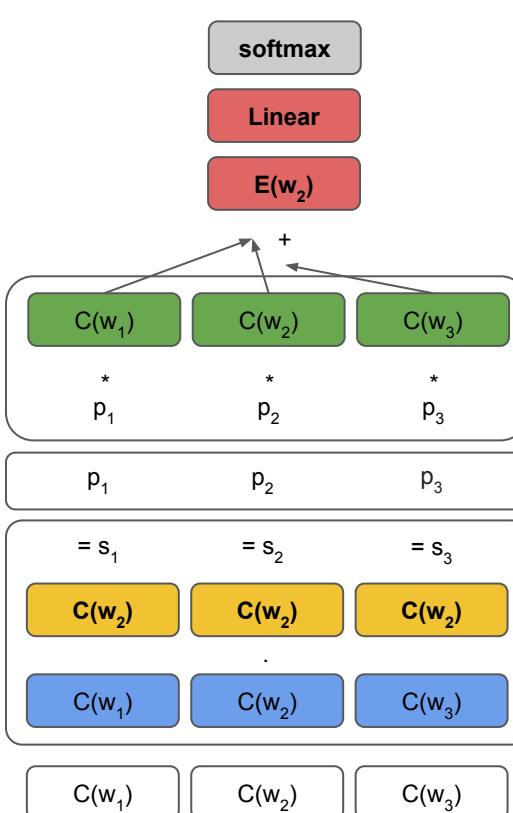
# Exercício desta semana



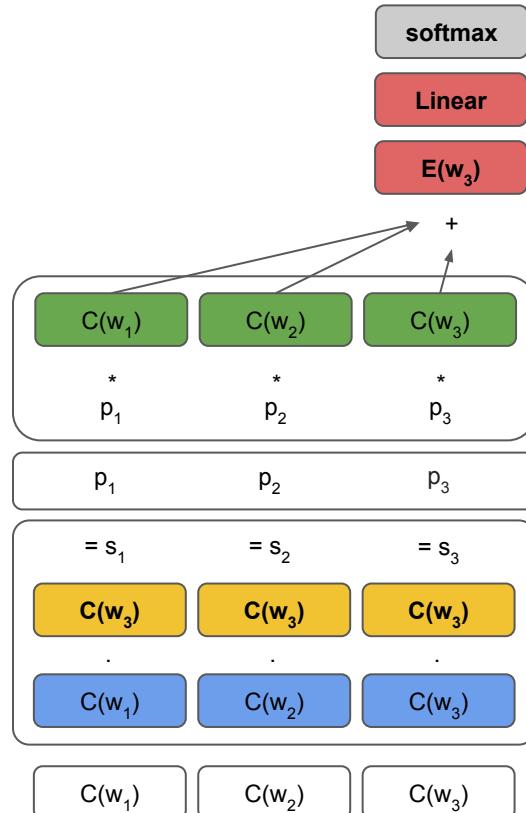
Eu



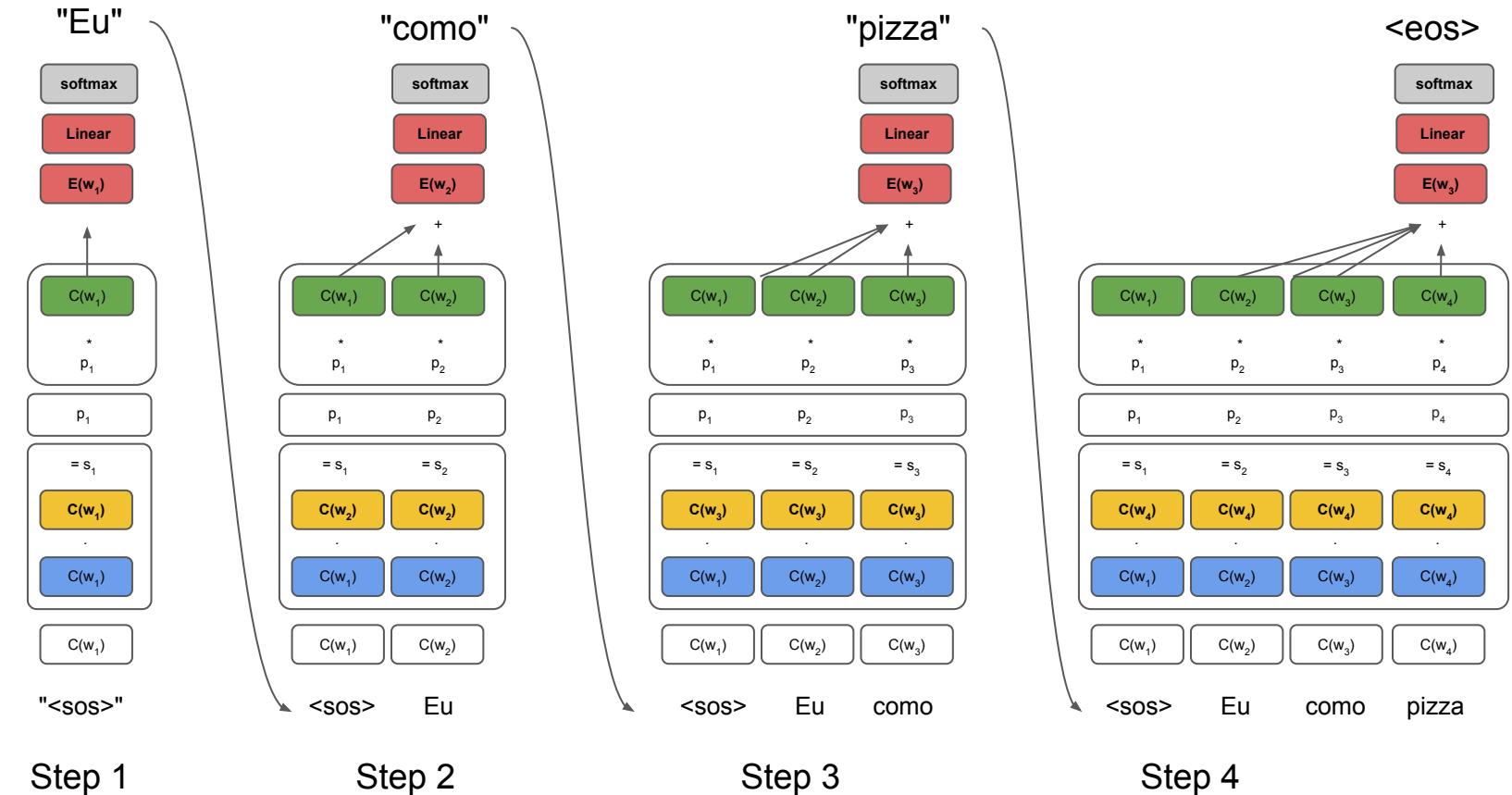
como



pizza

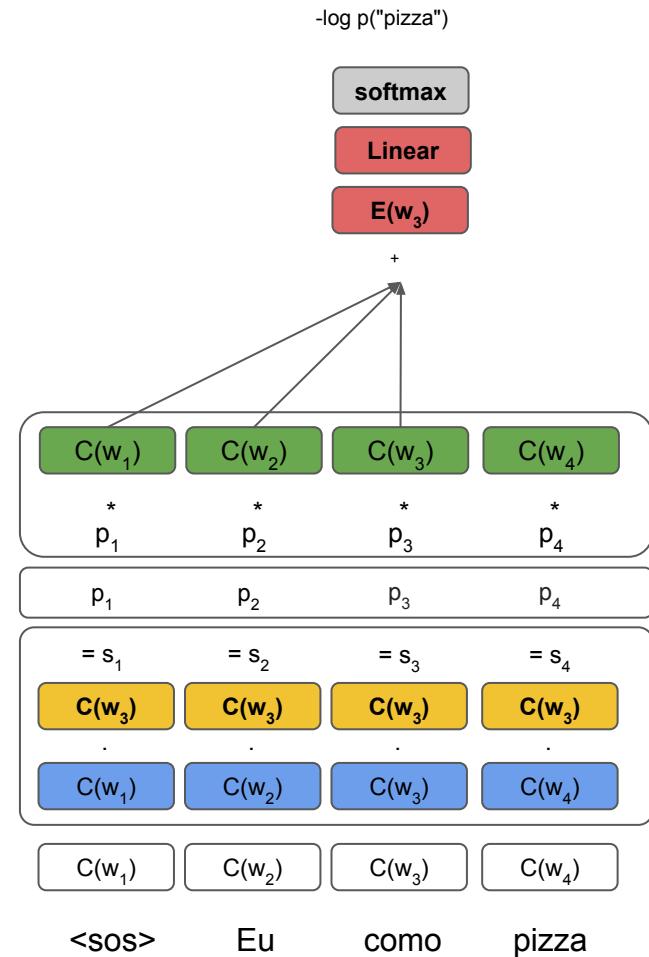
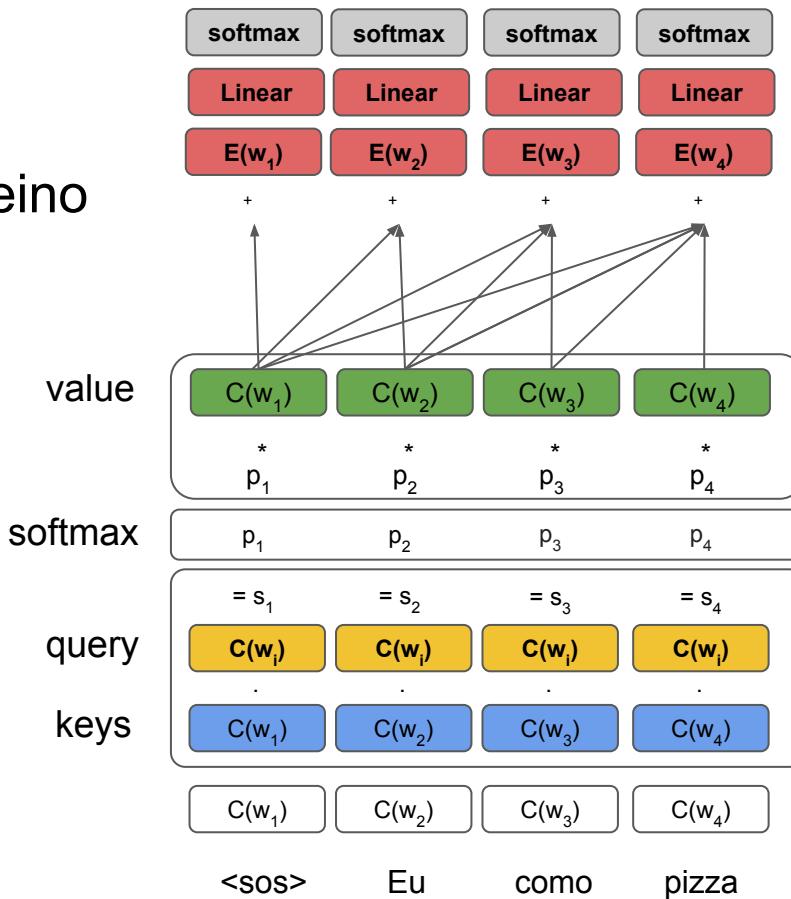


# Como um decoder-only (ex: GPT) gera texto?

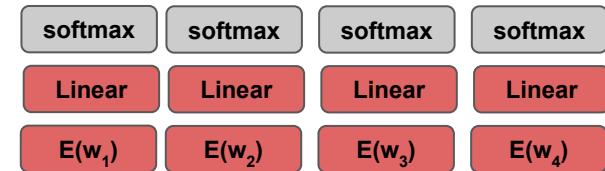


$-\log p("Eu")$     $-\log p("como")$     $-\log p("pizza")$     $-\log p("<\text{eos}>")$

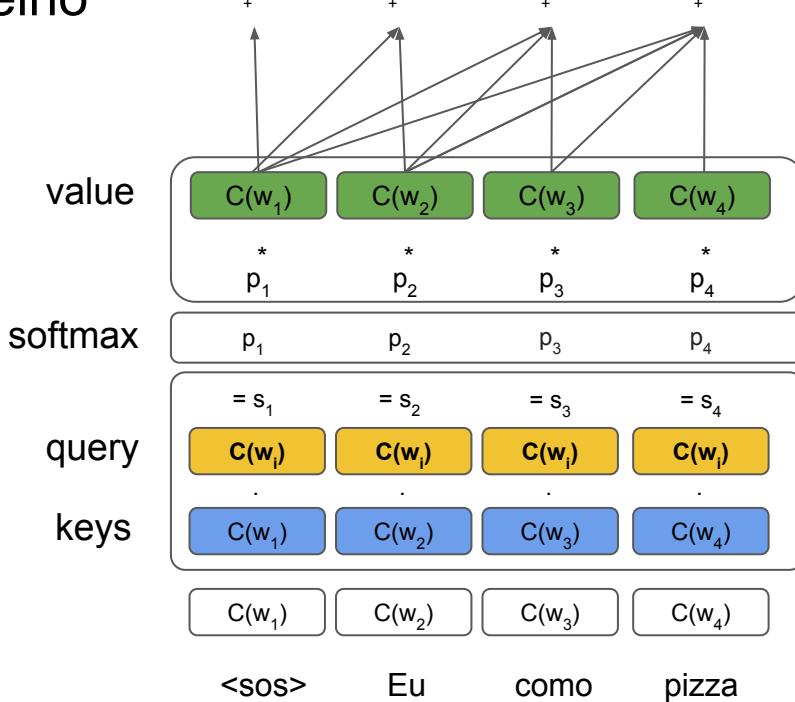
Em treino



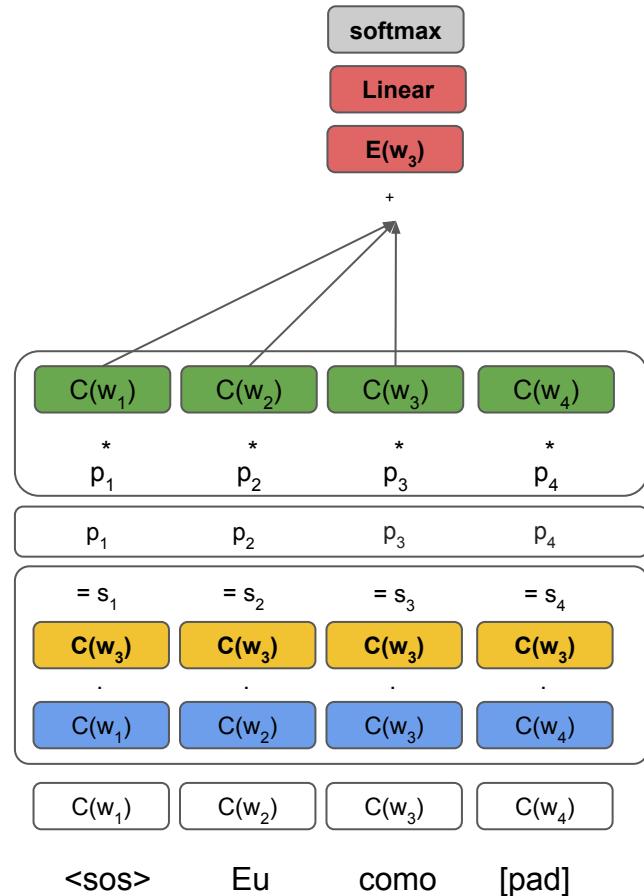
$-\log p("Eu") \quad -\log p("como") \quad -\log p("pizza") \quad -\log p("<\text{eos}>")$



Em treino



$-\log p("pizza")$



# Exemplos de Tamanhos Variáveis: Padding

3 docs:

Doc 1:	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$			
Doc 2:	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$	$w_{2,5}$	$w_{2,6}$
Doc 3:	$w_{3,1}$	$w_{3,2}$				

context\_size = 4 e PAD

Exemplo 1:	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	PAD
Exemplo 2:	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$
Exemplo 3:	$w_{3,1}$	$w_{3,2}$	PAD	PAD
Exemplo 4:	$w_{2,5}$	$w_{2,6}$	PAD	PAD

# Exemplos de Tamanhos Variáveis: Padding

3 docs:

Doc 1:	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$<eos>$			
Doc 2:	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$	$w_{2,5}$	$w_{2,6}$	$<eos>$
Doc 3:	$w_{3,1}$	$w_{3,2}$	$<eos>$				

context\_size = 4 e PAD

Exemplo 1:	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$<eos>$
Exemplo 2:	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$
Exemplo 3:	$w_{3,1}$	$w_{3,2}$	$<eos>$	PAD
Exemplo 4:	$w_{2,5}$	$w_{2,6}$	$<eos>$	PAD

# Número de operações de um modelo auto-regressivo

		keys/values						
		Query	<sos>	Eu	gosto	de	Pizza	Custo
step	1	<sos>						1
	2	Eu						2
	3	gosto						3
	4	de						4
	5	pizza						5

Qual o custo computacional para gerar N tokens?

Nota 1: Em treino, a loss pode ser calculada em apenas um *forward pass* da rede, ou seja, não precisa de laço. Isso porque já sabemos os target tokens ( $y_1, \dots, y_N$ )

# Layer Normalization (Ba et al. 2016)

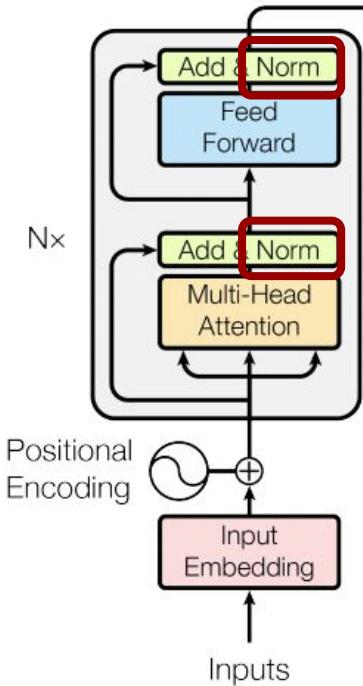
Objetivo: treinar mais rapidamente

Como: evitando que mudanças de peso na camada anterior causem grandes mudanças na camada seguinte

Normaliza-se as ativações de cada embedding de dimensão D para média beta e variância alpha, onde alpha e beta são parâmetros treinados.

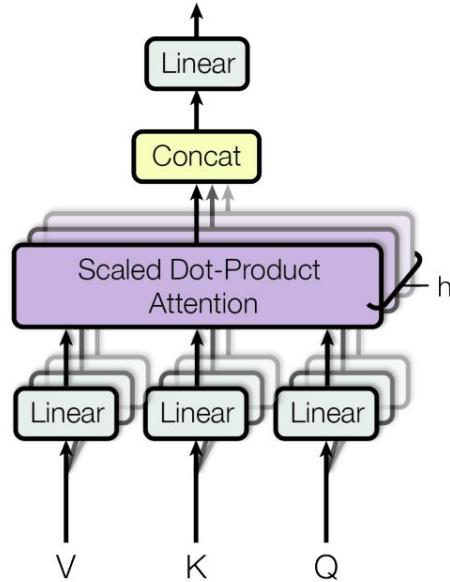
OBS: não é igual ao batch norm, utilizado em redes convolucionais

```
Na prática: def layer_norm(x):
    # x.shape = (B, L, D)
    mean = x.mean(2)
    std = x.std(2)
    x = (x - mean) / (std + 1e-6)
    x = x * alpha + beta # opcional
    return x
```



# Multi-head

Multi-Head Attention



$L$  = comprimento da seq

$D$  = Dimensão do modelo

$H$  = número de cabeças

Com laço nas cabeças:

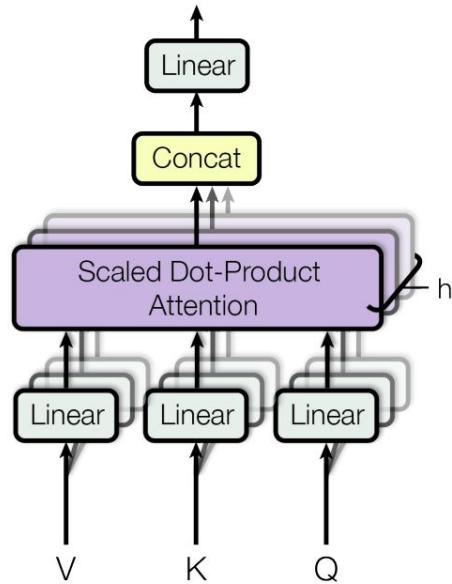
```
def __init__(self):
    ...
    for i in range(H):
        self.W_q[i] = nn.Linear(D, D/H, bias=False)
        self.W_k[i] = nn.Linear(D, D/H, bias=False)
        self.W_v[i] = nn.Linear(D, D/H, bias=False)
    self.W_o = nn.Linear(D, D, bias=False)
    ...

def forward(self, x):                                # x.shape = L, D
    new_x = empty(L, H, D/H)
    for i in range(H):
        q = self.W_q[i](x)
        k = self.W_k[i](x)
        v = self.W_v[i](x)
        e = attention(q, k, v) # L, D/H
        new_x[:, i, :] = e

    new_x = new_x.reshape(L, D)
    return self.W_o(new_x)      # L, D
    ...
```

# Multi-head

Multi-Head Attention



L = comprimento da seq

D = Dimensão do modelo (embedding)

H = número de cabeças

Na prática, sem laço:

```
def __init__(self):
    ...
    self.W_q = nn.Linear(D, D, bias=False) # D, H * D/H
    self.W_k = nn.Linear(D, D, bias=False) # D, H * D/H
    self.W_v = nn.Linear(D, D, bias=False) # D, H * D/H
    self.W_o = nn.Linear(D, D, bias=False)

    ...

def forward(self, x):                      # x.shape = L, D
    q = self.W_q(x).reshape(L, H, D/H)
    k = self.W_k(x).reshape(L, H, D/H)
    v = self.W_v(x).reshape(L, H, D/H)

    # Transpor para: H, L, D/H
    q, k, v = q.transpose(0, 1), k.transpose(0, 1), v.transpose(0, 1)

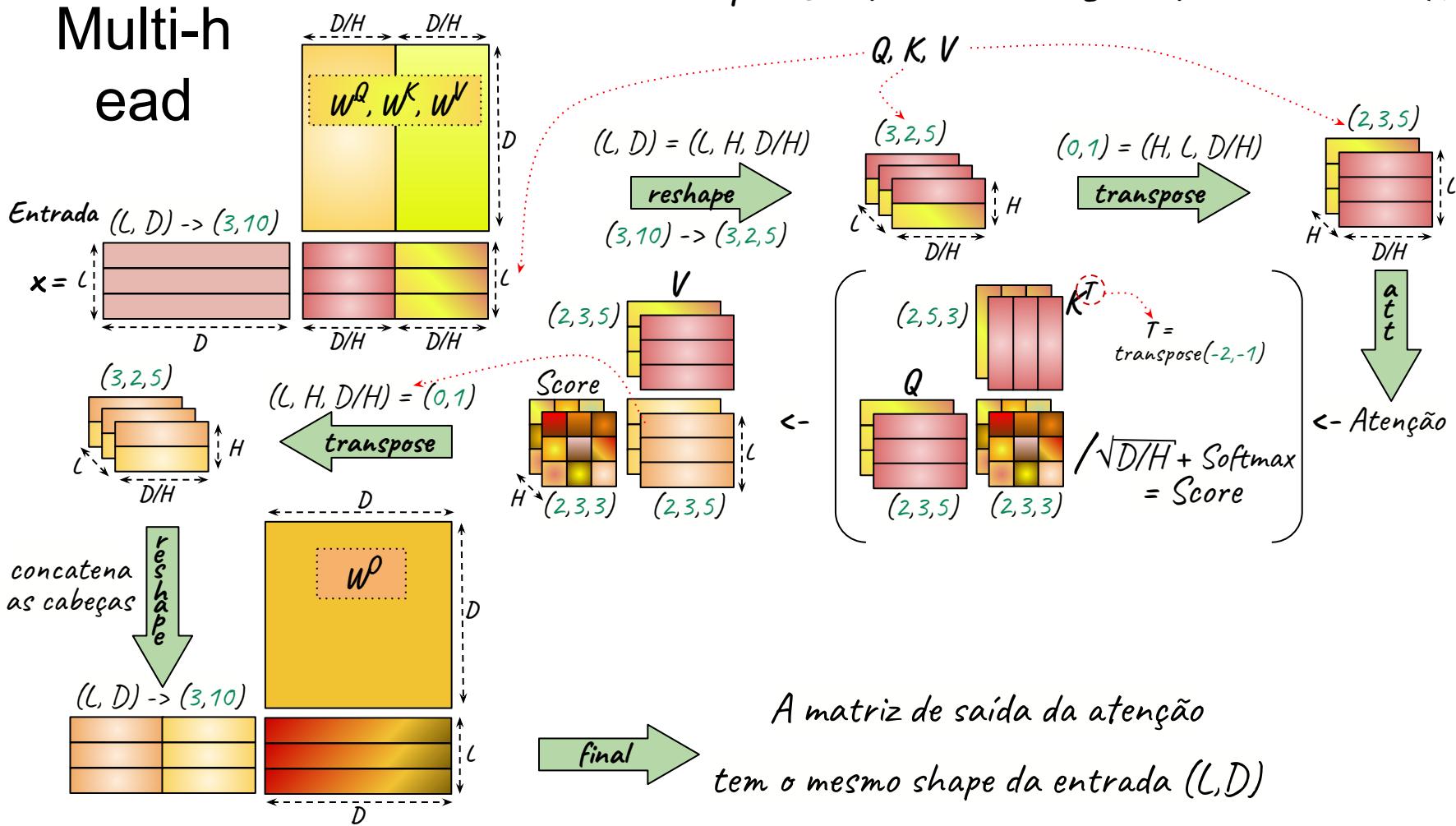
    new_x = attention(q, k, v)      # new_x.shape = H, L, D/H

    new_x = new_x.transpose(0, 1).contiguous() # new_x.shape = L, H, D/H
    new_x = new_x.reshape(L, D)
    return self.W_o(new_x)

    ...
```

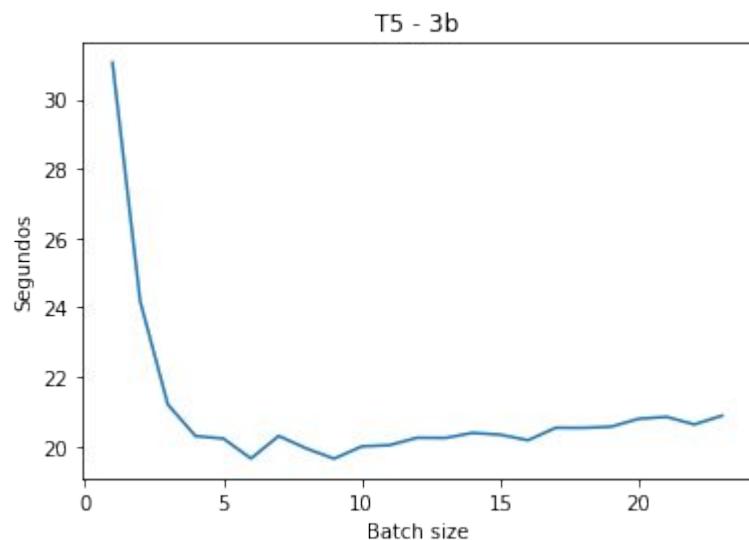
# Multi-head

tam. seq  $\rightarrow L = 3$ , dim. emb.  $\rightarrow D = 10$ , num. heads  $\rightarrow H = 2$

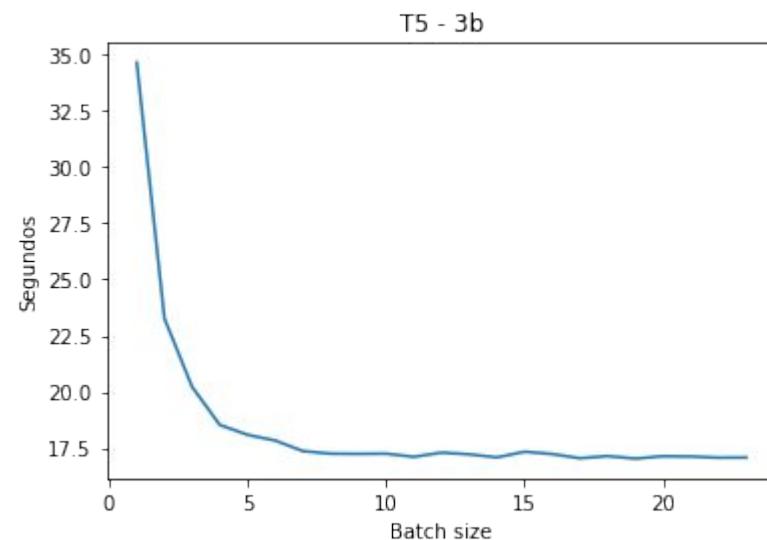


# Batches dinâmicos (optimização opcional)

Aleatório



Batches com exemplos de comprimento próximos



# Assunto da décima aula

- Correção de Exercícios da Aula 9
  - Leitura: [GPT-3](#)
  - Exercício: Treinamento de um modelo decoder-only com duas camadas de auto-atenção no corpus BrWaC
- Assuntos:
  - Revisão Transformers
  - Discussão dos projetos
- Leitura: Não haverá
- Exercício: Repetir o exercício da aula 9

# Formas de ignorar o [PAD]

Usar vetor de zeros para representar o [PAD]?

Forçar  $p=0$  para o [PAD]?

Precisamos ignorá-lo em todas as camadas?

Máscara causal precisa ser aplicada em todas as camadas?

# Assunto da décima primeira aula

- Correção de Exercícios da Aula 10
  - Leitura (opcional): [Scaling Laws for Neural Language Models](#)
  - Exercício: Treinamento de um modelo decoder-only com duas camadas de auto-atenção no corpus BrWaC
- Assuntos:
  - Image Captioning
  - Discussão dos projetos
- Leitura: Não haverá
- Exercício: Escrita da proposta projeto final

# Image Captioning

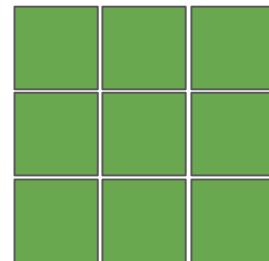
CNN (EfficientNet) + Decodificador do Transformer (T5, BART, Pegasus) treinado no MS COCO



Entrada:  
 $H \times W \times 3$



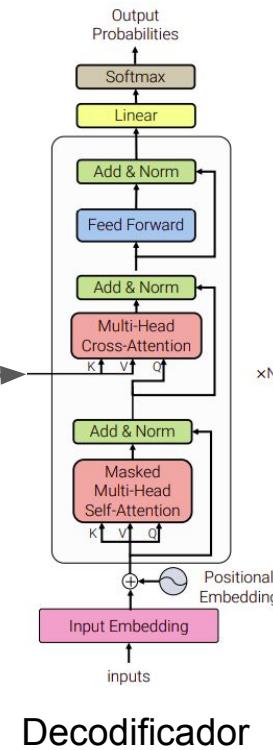
Feature Map:  
 $N \times N \times D_1$



Feature Map:  
 $N \times N \times D_2$

Necessário quando a dimensão da CNN pode ser diferente da dimensão do Transformer

$N^2$  vetores



# Image Captioning

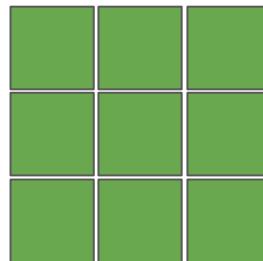
Outra opção é entrar com o Feature Map no *codificador* do Transformer

Dica: experimentar com camadas intermediárias da CNN que gerem grandes N (ex:  $N > 14$ )

Dica 2: usar tudo pré-treinado (CNN e Transformer)



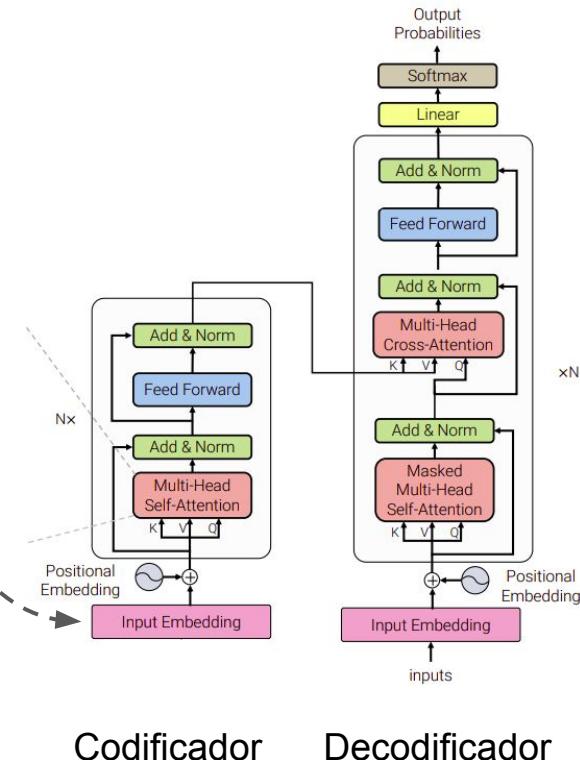
Entrada:  
 $H \times W \times 3$



Feature Map:  
 $N \times N \times D_1$



Feature Map:  
 $N \times N \times D_2$



# Image Captioning

Sugestão de leitura: [Show, Attend and Tell: Neural Image Caption Generation with Visual Attention](#)