

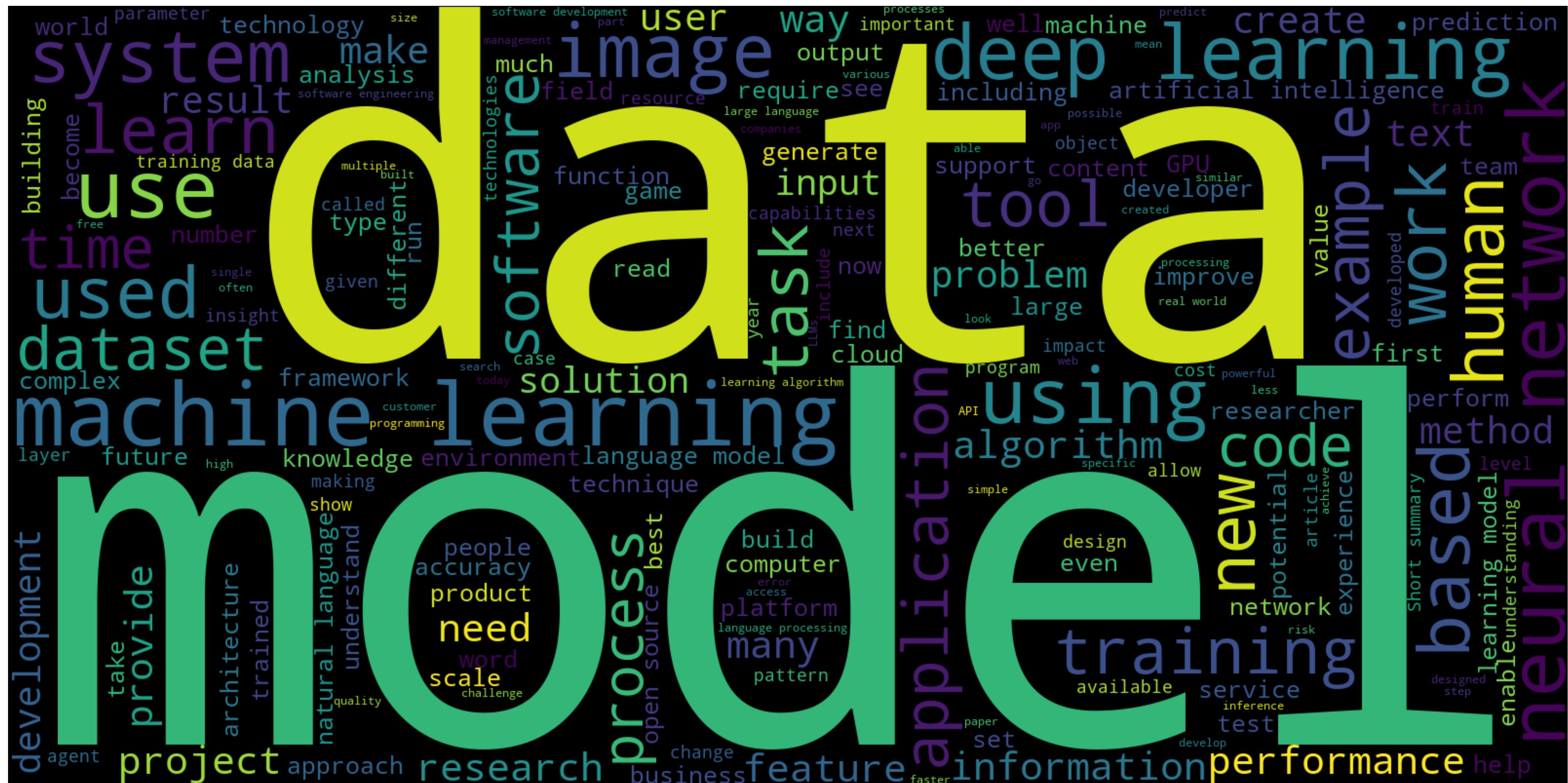
Deep Learning

software 2.0, artificial intelligence

Didier Guillevic
didier.guillevic.net



Deep Learning - Software 2.0 - Artificial Intelligence



- 0. Software 2.0
- 1. Deep Neural Networks
- 2. Concepts: transfer (meta) learning, tabula rasa, self-supervised
- 3. Generative AI

Generative Adversarial Networks (GANs)

Stable Diffusion Model

- 4. Natural Language Processing

Embeddings, neural language model, RNN, LSTM

Attention, transformers, large language model

Chat LLM: prompt engineering, updates, private documents

Deep Learning - Software 2.0

Source: [Andrej Karpathy \(2017-11\)](#)

Software 1.0



Program space

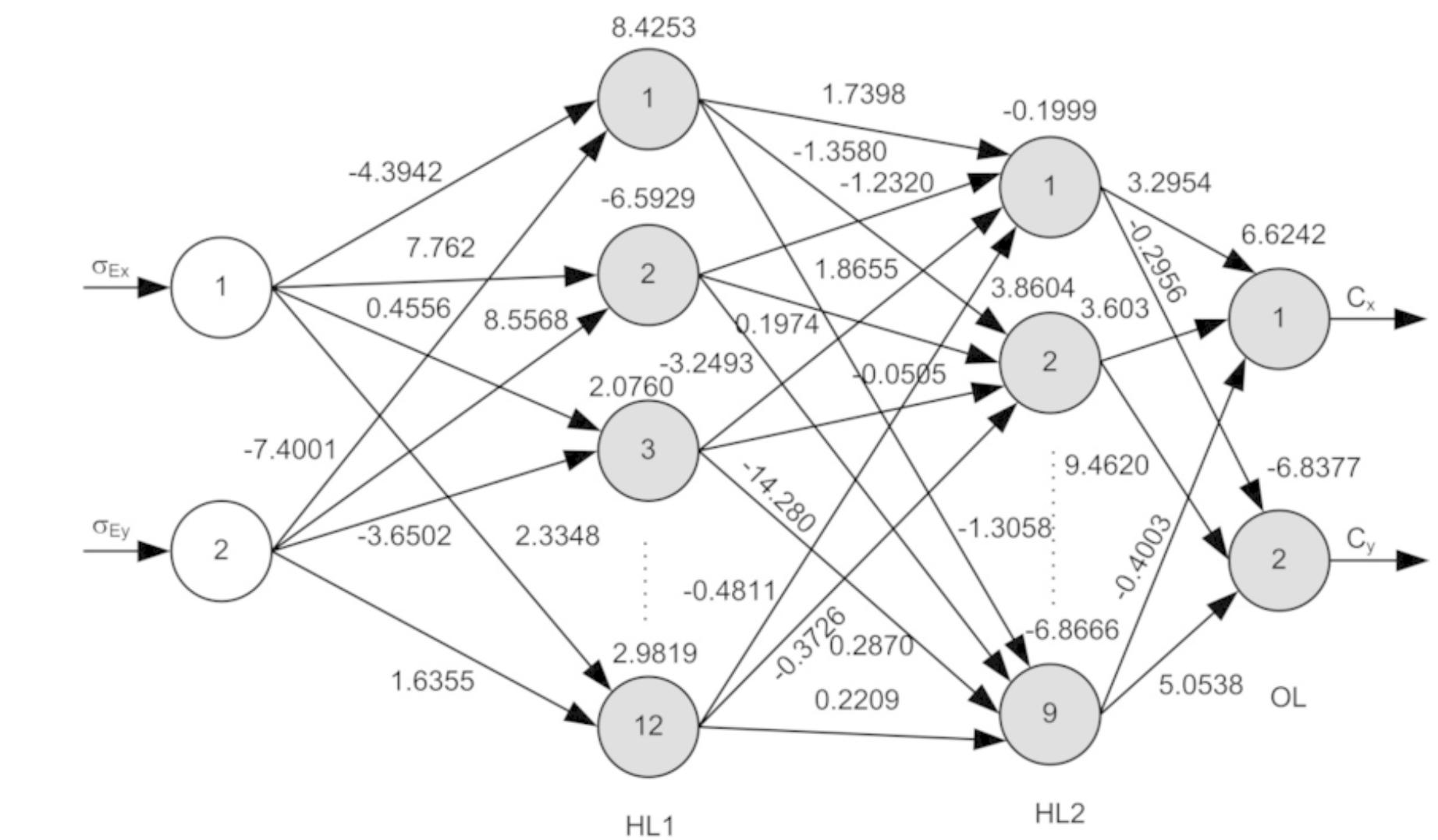
Software 1.0

Program complexity

Software 2.0

(optimization)

Software 2.0



Software 2.0 - Applications

Andrej Karpathy (2017-11)

Domain	Software 1.0	Software 2.0
Visual Recognition	Engineered features with a bit of machine learning	Searching the space of Convolutional Neural Network architectures
Speech Recognition	Preprocessing, gaussian mixture models, hidden markov models	Almost entirely neural networks
Speech Synthesis	Various sticking mechanisms	Large Convolutional Neural Networks (e.g. WaveNet)
Machine Translation	Phrase-based statistical techniques	Neural networks (supervised and unsupervised)
Games	Hand-coded Go playing programs	AlphaGo Zero: only uses the game's set of rules and learns strategies by itself
Databases	Indices using B-trees or Hash indexes	Learned Index Structures (using deep learning models) arxiv: 1712.01208, 2006.12804

Software 2.0 - Benefits

Andrej Karpathy (2017-11)

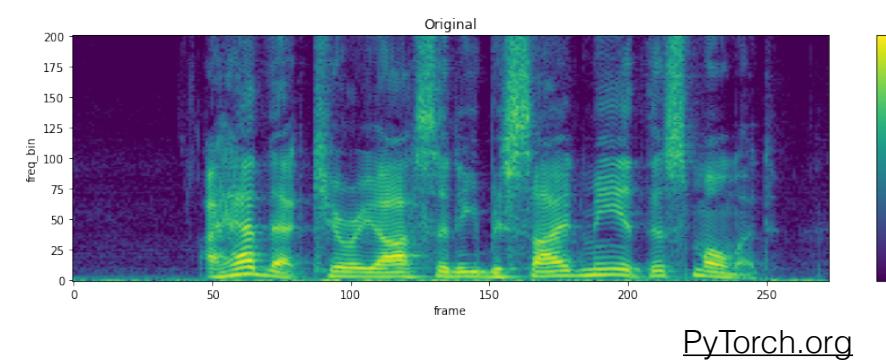
Software 1.0: production-level C++ code base

Software 2.0: Convolutional Neural Network / Transformers

Homogeneous computation	Software 1.0: instruction set of classical software (heterogenous and complex) Software 2.0: matrix multiplication (and thresholding at zero)
Simple to bake in silicon	Easier to make custom ASICs Small inexpensive chips could come with a pre-trained ConvNet
Constant running time	C++ code could have unintended infinite loop Forward pass of a neural network takes exactly the same amount of FLOPS Zero variability
Constant memory use	No dynamically allocated memory as in C++ Little possibility of memory leaks
Highly portable	Sequence of matrix multiplies is easy to run-on arbitrary computational configurations (not the same for binaries)
Very agile	C++: non-trivial to make a system run twice as fast Software 2.0: remove half of the nodes and re-train to get twice the speed OR make program work better by adding more channels (or averaging models)
Melding modules together	Software 1.0: communicates through public functions, APIs, ... Software 2.0: two modules trained separately can be connected and the weights adjusted by back-propagation

Deep Learning: State Of The Art (SOTA)

Speech



[PyTorch.org](https://pytorch.org)

 **OpenAI** [Whisper](#) (2022-09): 680,000 hours of multi-lingual speech

Transcription in multiple languages, any-to-English transcription,
Language identification, phrase-level timestamps

 **Meta AI** [Massively Multilingual Speech \(MMS\)](#) (2023-05)

Speech-to-text, text-to-speech in 1,100+ languages

Image / Video



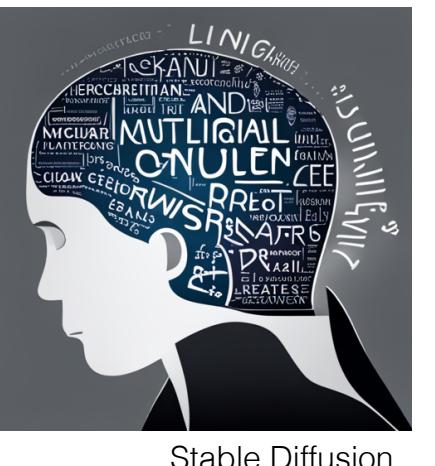
Stable Diffusion

 **Meta AI**

[Segment Anything](#) (2022-09)



Natural Language Processing



Stable Diffusion

 **Meta AI** [No Language Left Behind \(NLLB\)](#) (2022): translation to/from 200 languages

 **OpenAI** [ChatGPT](#) (2022-11)

Large Language Models: Bloom, Llama, GPT, PaLM, ...

Medical

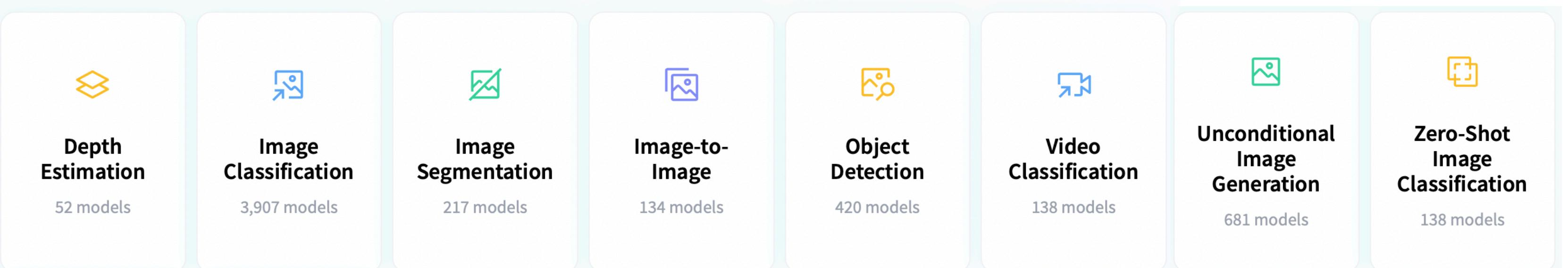


Google DeepMind

[AlphaFold](#) (2020)



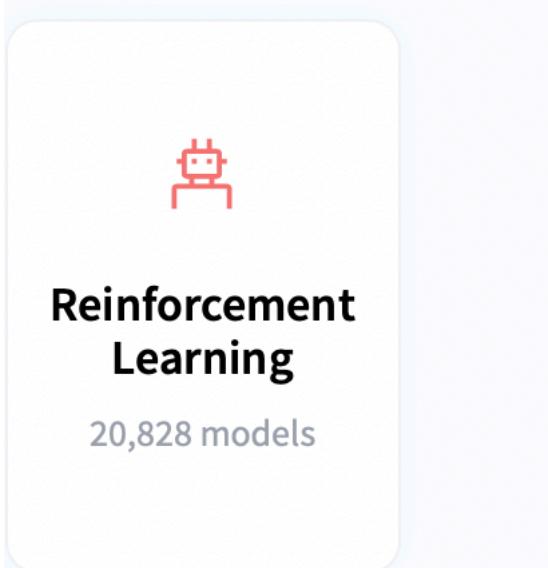
Computer Vision



A grid of eight cards representing different Computer Vision tasks:

 Depth Estimation 52 models	 Image Classification 3,907 models	 Image Segmentation 217 models	 Image-to-Image 134 models	 Object Detection 420 models	 Video Classification 138 models	 Unconditional Image Generation 681 models	 Zero-Shot Image Classification 138 models
---	--	--	--	--	--	--	--

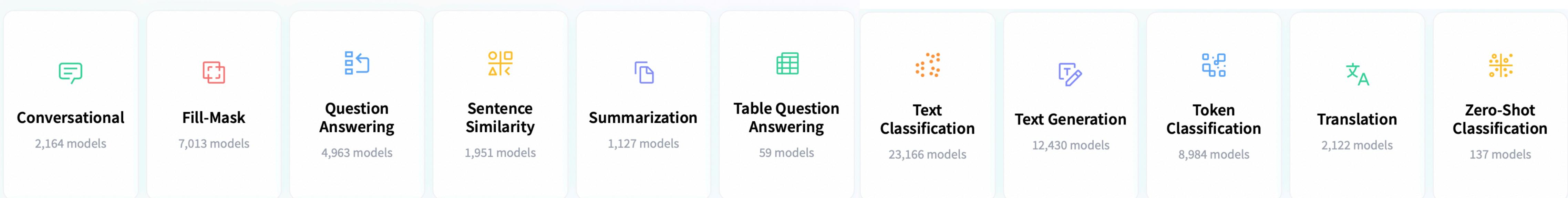
Reinforcement Learning



A single card representing Reinforcement Learning:

 Reinforcement Learning 20,828 models

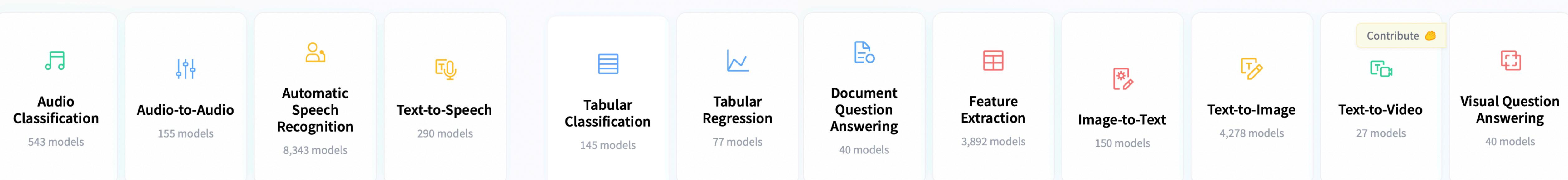
Natural Language Processing



A grid of twelve cards representing different NLP tasks:

 Conversational 2,164 models	 Fill-Mask 7,013 models	 Question Answering 4,963 models	 Sentence Similarity 1,951 models	 Summarization 1,127 models	 Table Question Answering 59 models	 Text Classification 23,166 models	 Text Generation 12,430 models	 Token Classification 8,984 models	 Translation 2,122 models	 Zero-Shot Classification 137 models
--	---	--	---	---	---	--	--	--	---	--

Audio



A grid of four cards representing different Audio tasks:

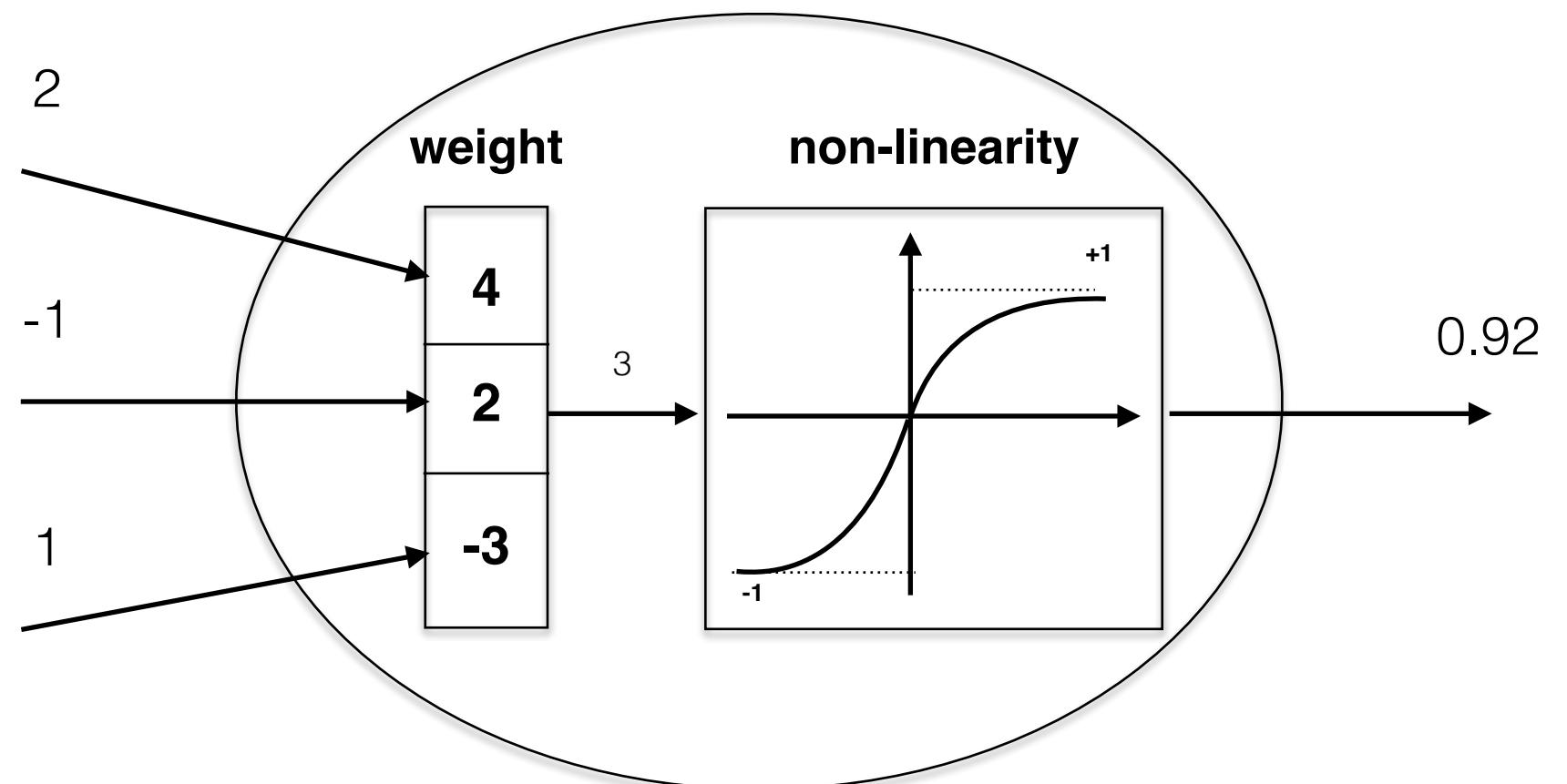
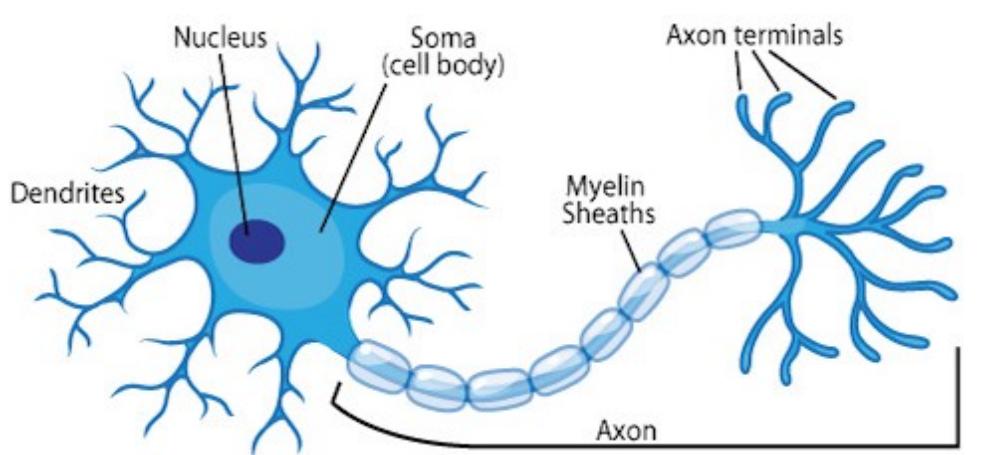
 Audio Classification 543 models	 Audio-to-Audio 155 models	 Automatic Speech Recognition 8,343 models	 Text-to-Speech 290 models
--	--	--	--

Below the audio section are three more categories:

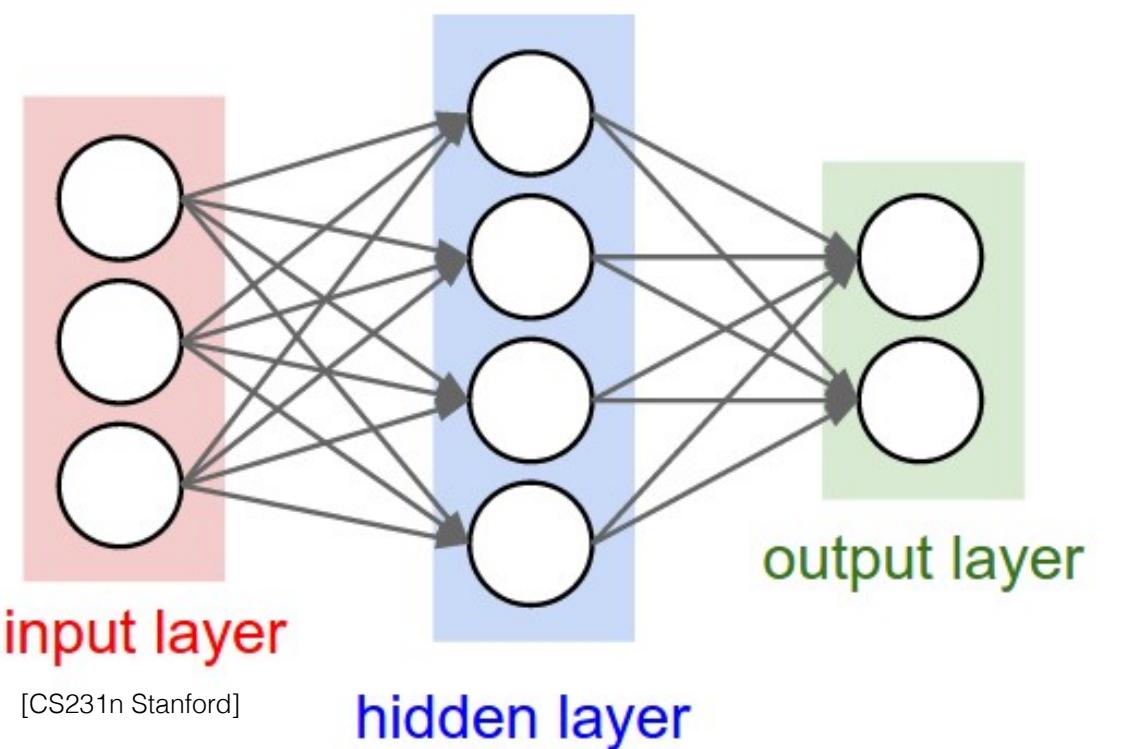
- Tabular**: Contains two cards: Tabular Classification (145 models) and Tabular Regression (77 models).
- Multimodal**: Contains five cards: Document Question Answering (40 models), Feature Extraction (3,892 models), Image-to-Text (150 models), Text-to-Image (4,278 models), and Text-to-Video (27 models).
- Visual Question Answering**: A single card with a "Contribute" button.

Deep Neural Networks

Neuron

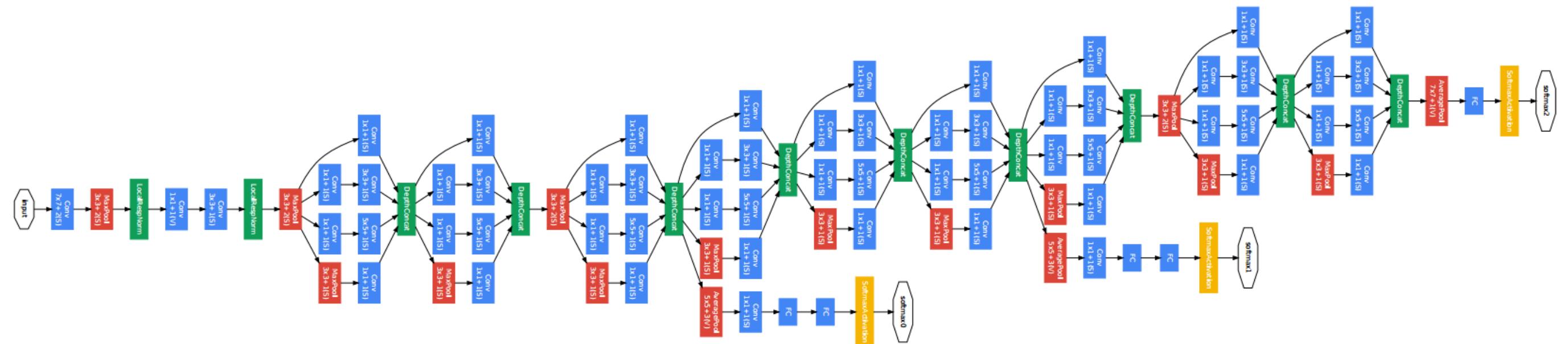


Neural Networks (2 - 5 layers) (until 2006)



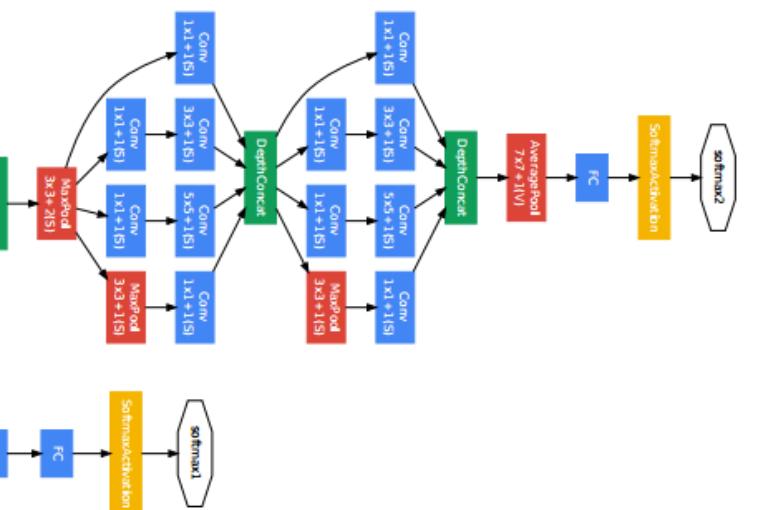
Deep Neural Networks (10 - 150 layers)

Convolutional Neural Networks: deep and narrow

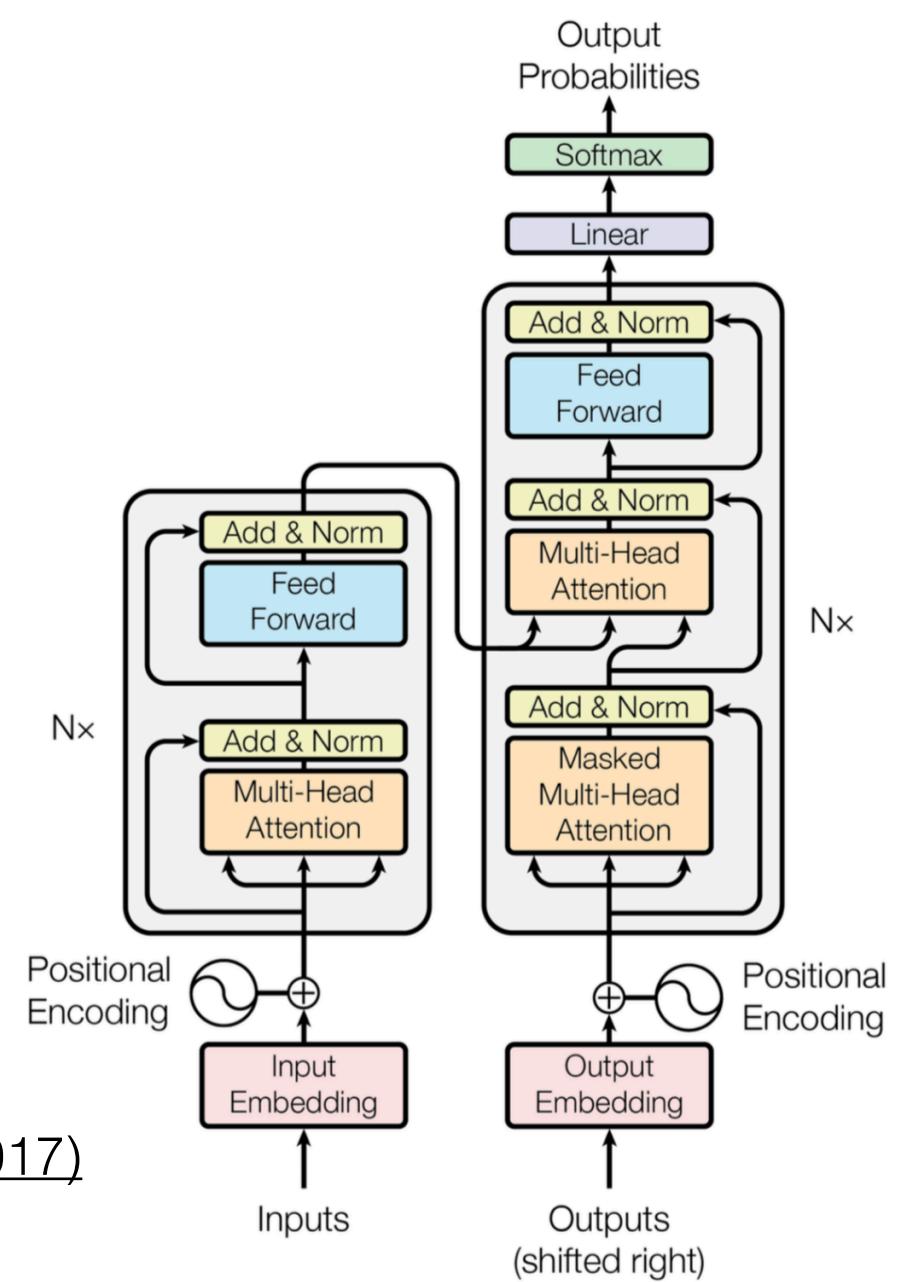


GoogLeNet (2014)

Transformers: “shallow” and wide



Attention Is All You Need (2017)

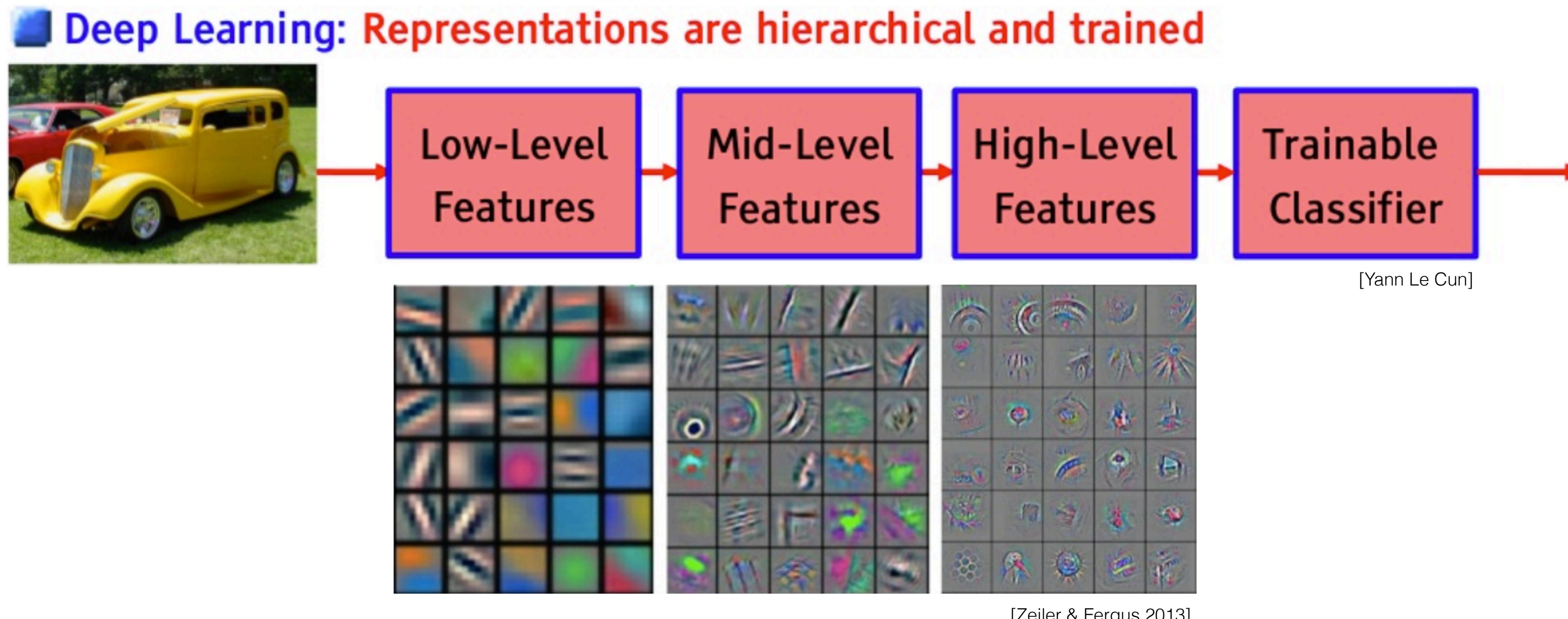


Deep Learning: Change in paradigm

Before: Handcrafting features: domain experts: 10s of years



Now: Learned features: **End-to-end** Learning



Why only recently (circa 2012)? Why not sooner?

1. A lot more (labeled) data
2. A lot more computing power
3. Knowledge on how to train deep networks

1. A lot more (labeled) data

Data



14 millions images
21000 labels (WordNet)



WIKIPEDIA
The Free Encyclopedia

6 millions articles in the English Wikipedia

Common Crawl

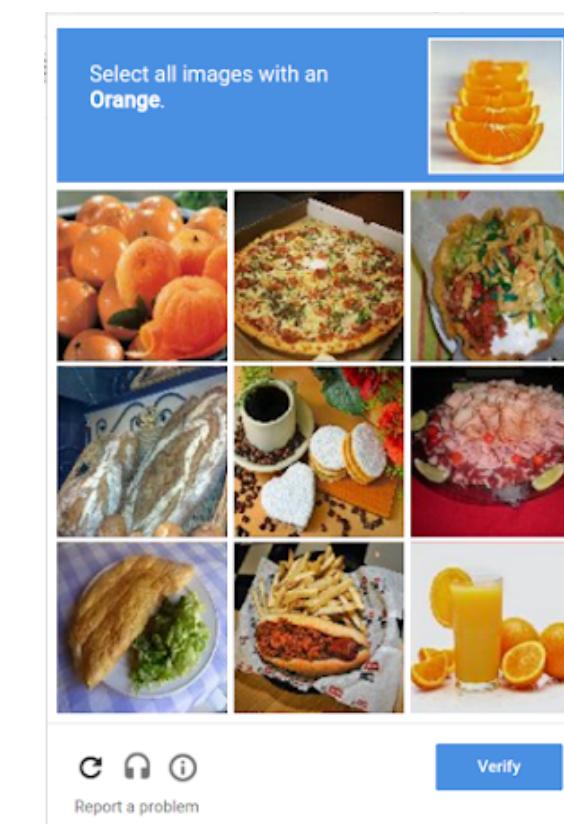


3 billion web pages

Labels



Low-cost global, 24x7 workforce



scale.com



reCAPTCHA

Free global, 24x7 workforce

2. A lot more computing power



[Apple M2]

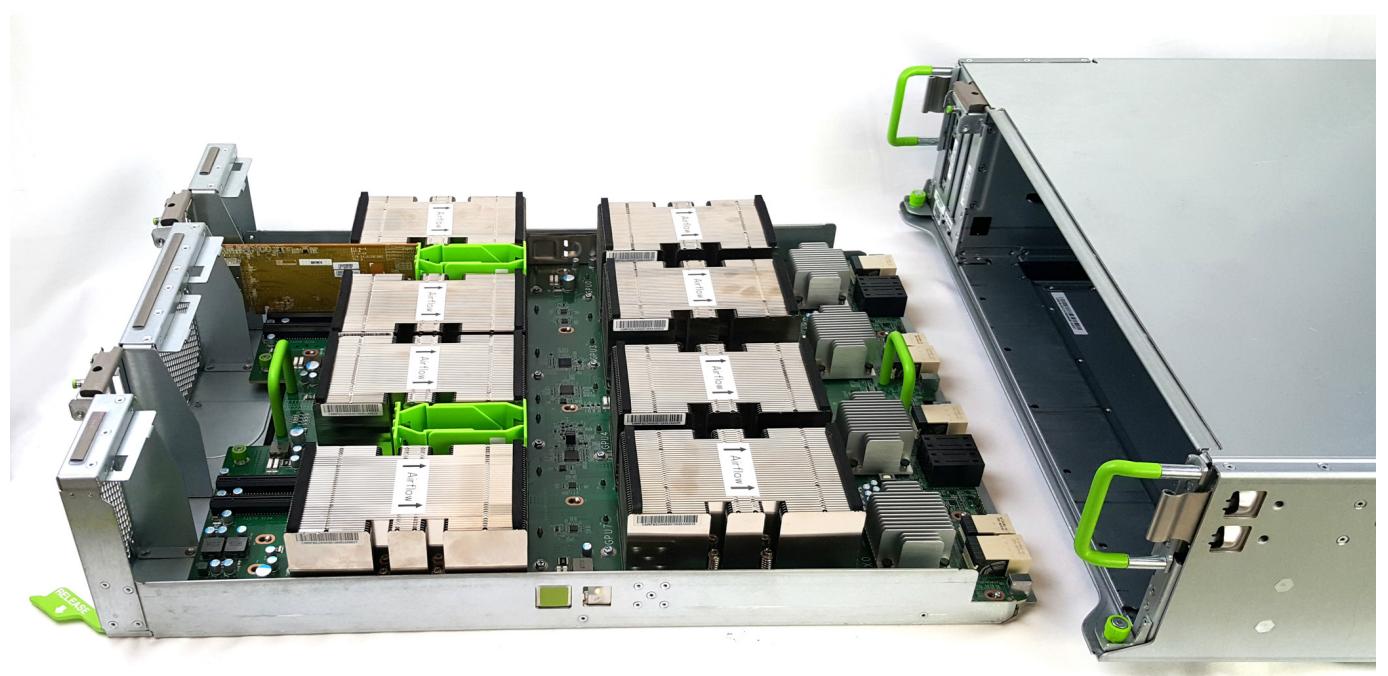


CPUs: 20 cores



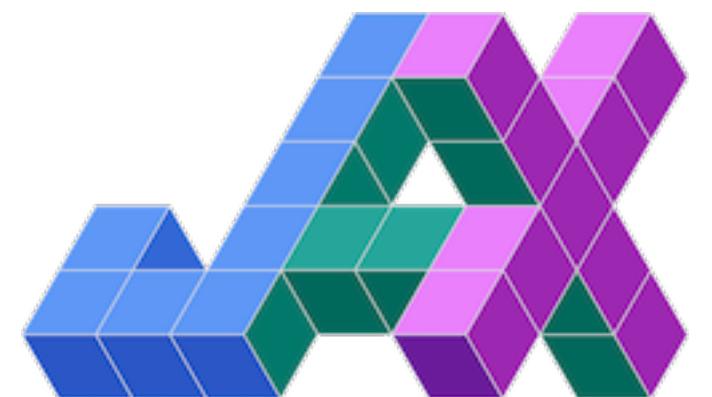
[NVIDIA A100]

GPUs: 6000 cores



Facebook Big Basin (2017)

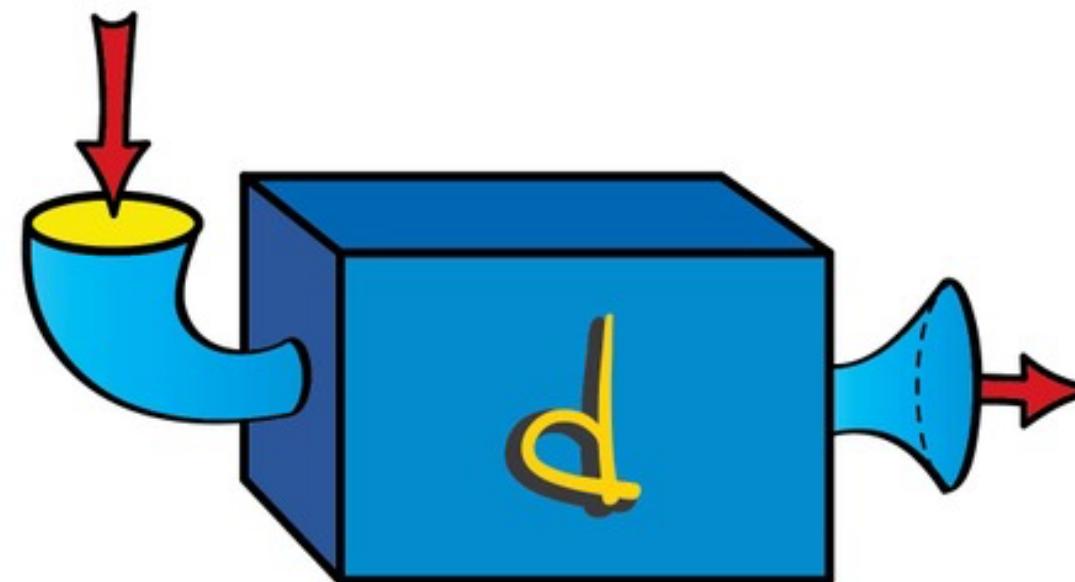
Platforms



Distributed Computing

3. Knowledge on how to train deep networks

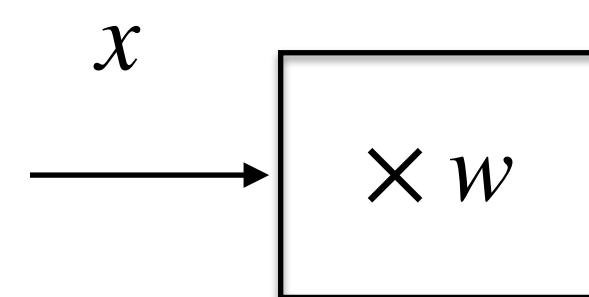
Learning using (Stochastic) Gradient Descent ("going downhill")



[Robert Ghrist - Calculus - U. of Penn]

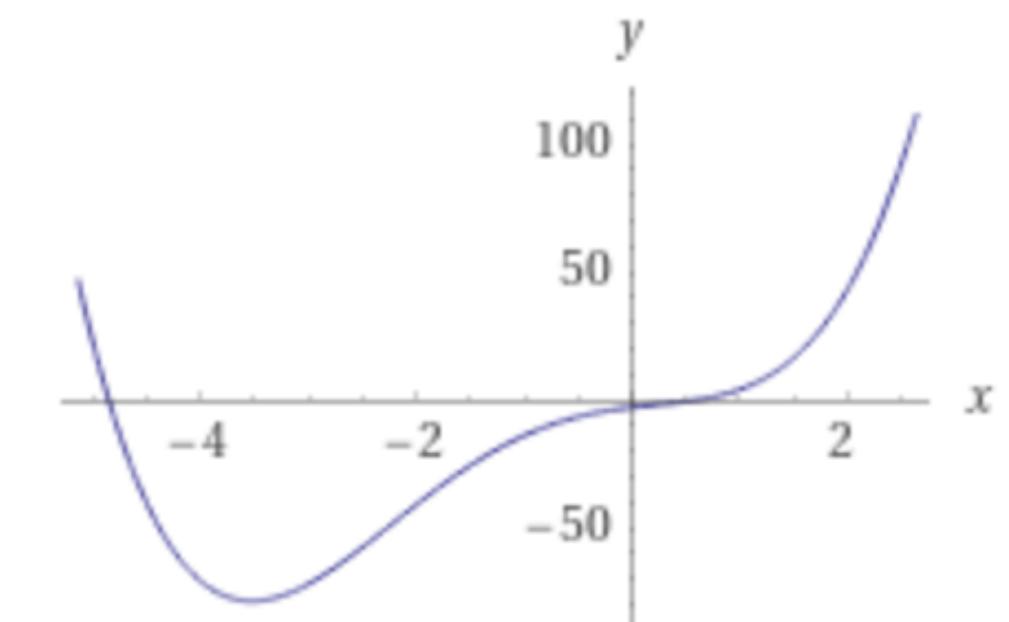
$$\frac{\text{change in output}}{\text{change in input}} = \frac{dy}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{(x+h) - x} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

$$f(x + h) = f(x) + \mathbf{d}h$$



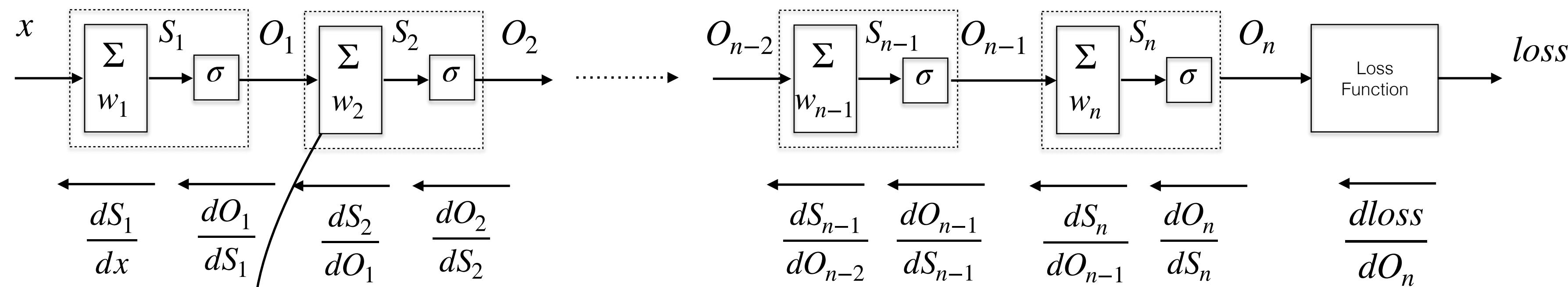
$$(x + \underline{h})w = xw + \underline{h}w \quad \rightarrow \quad \frac{dy}{dx} = w$$
$$x(w + \underline{h}) = xw + x\underline{h} \quad \rightarrow \quad \frac{dy}{dw} = x$$

$$w_{new} = w_{old} - learning_rate * \frac{dy}{dw}$$



3. Learning: chain rule of gradients

Chain rule of gradients (multiplicative)



$$w_{2new} = w_{2old} - lr * \frac{dloss}{dw2}$$

$$w_{2new} = w_{2old} - lr * \frac{dloss}{dO_n} * \frac{dO_n}{dS_n} * \frac{dS_n}{dO_{n-1}} * \frac{dO_{n-1}}{dS_{n-1}} * \frac{dS_{n-1}}{dO_{n-2}} * \dots * \frac{dO_2}{dS_2} * O_1$$

$$w_{2new} = w_{2old} - lr * 10 * 10 * 10 * 10 * 10 * \dots * 10 * 10 \approx w_{2old} - lr * \infty$$

$$w_{2new} = w_{2old} - lr * 0.10 * 0.10 * 0.10 * \dots * 0.10 * 0.10 \approx w_{2old} - lr * 0.0$$

Exploding gradient

Vanishing gradient

⇒ We want all terms to have a norm centred around 1.0 (normally distributed)

3. Tricks: weights initialization, batch norm, ReLU

⇒ We want the norm of all terms to follow a normal distribution (mean 0, standard deviation 1)

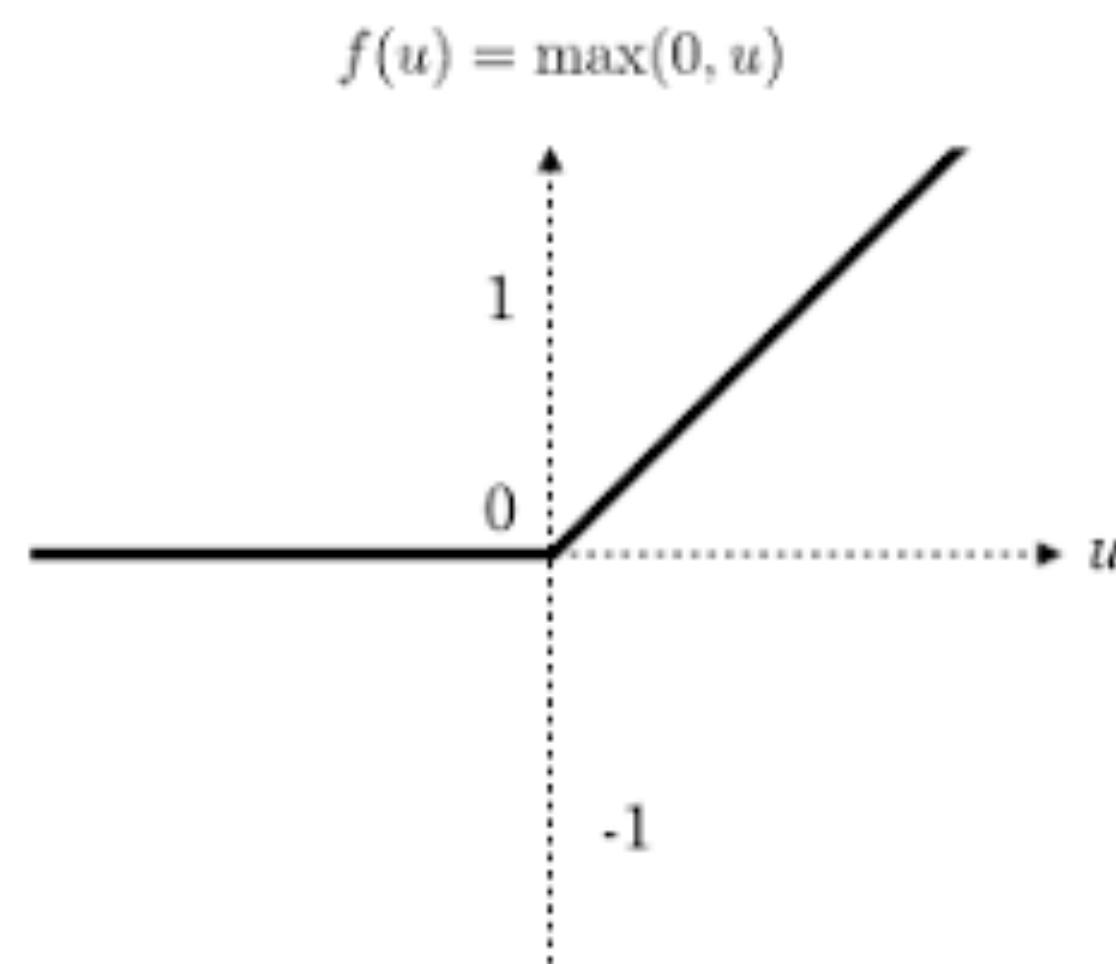
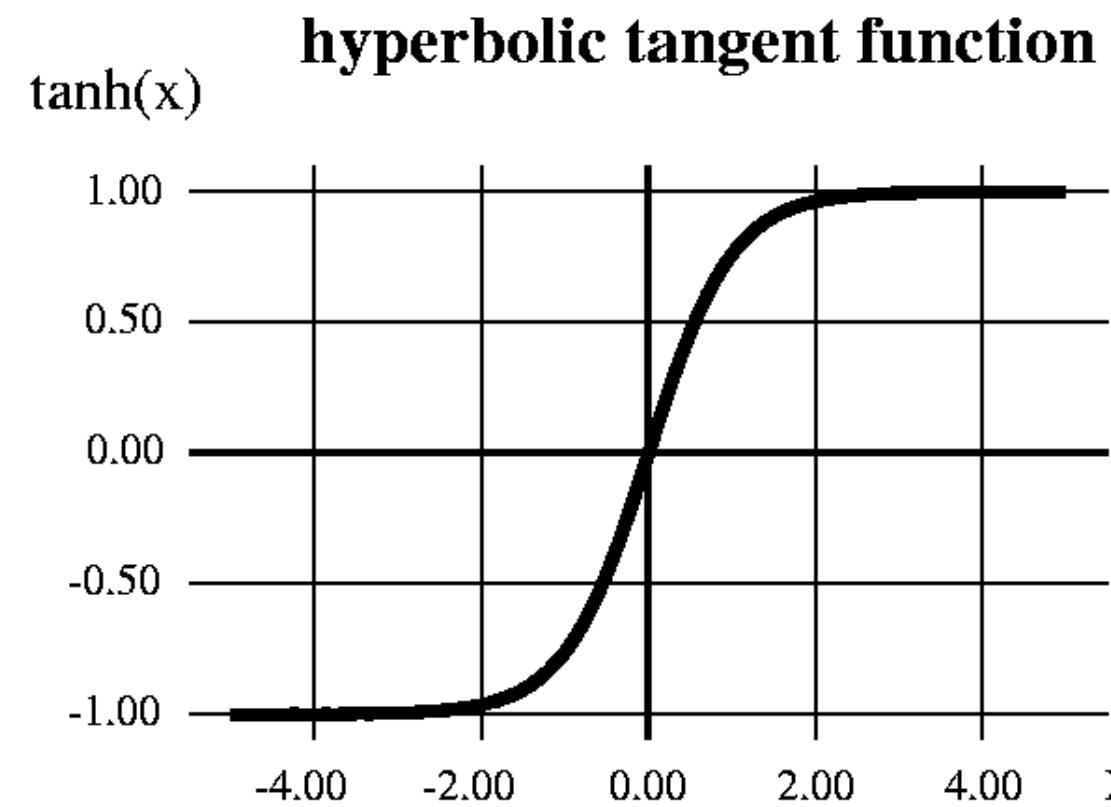
Weight initialization: inversely proportional to number of inputs (e.g. Xavier's initialization scheme)

$$S = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + \dots + w_n * x_n$$

BatchNorm: scales inputs at each layer to be centred around 0.0 with variance 1.0 (unit Gaussian)

LayerNorm

Non-linearity function: sigmoid have zero gradients → use ReLU



3. Tricks: Dropout, update rules, residual (skip) networks, ...

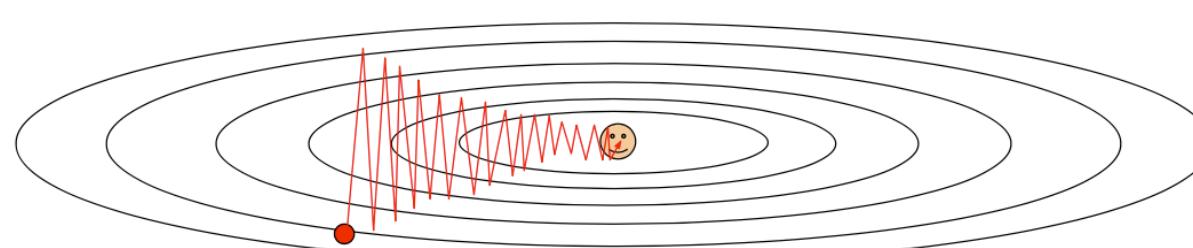
DropOut: randomly ignore some neurons during training

Update rule:

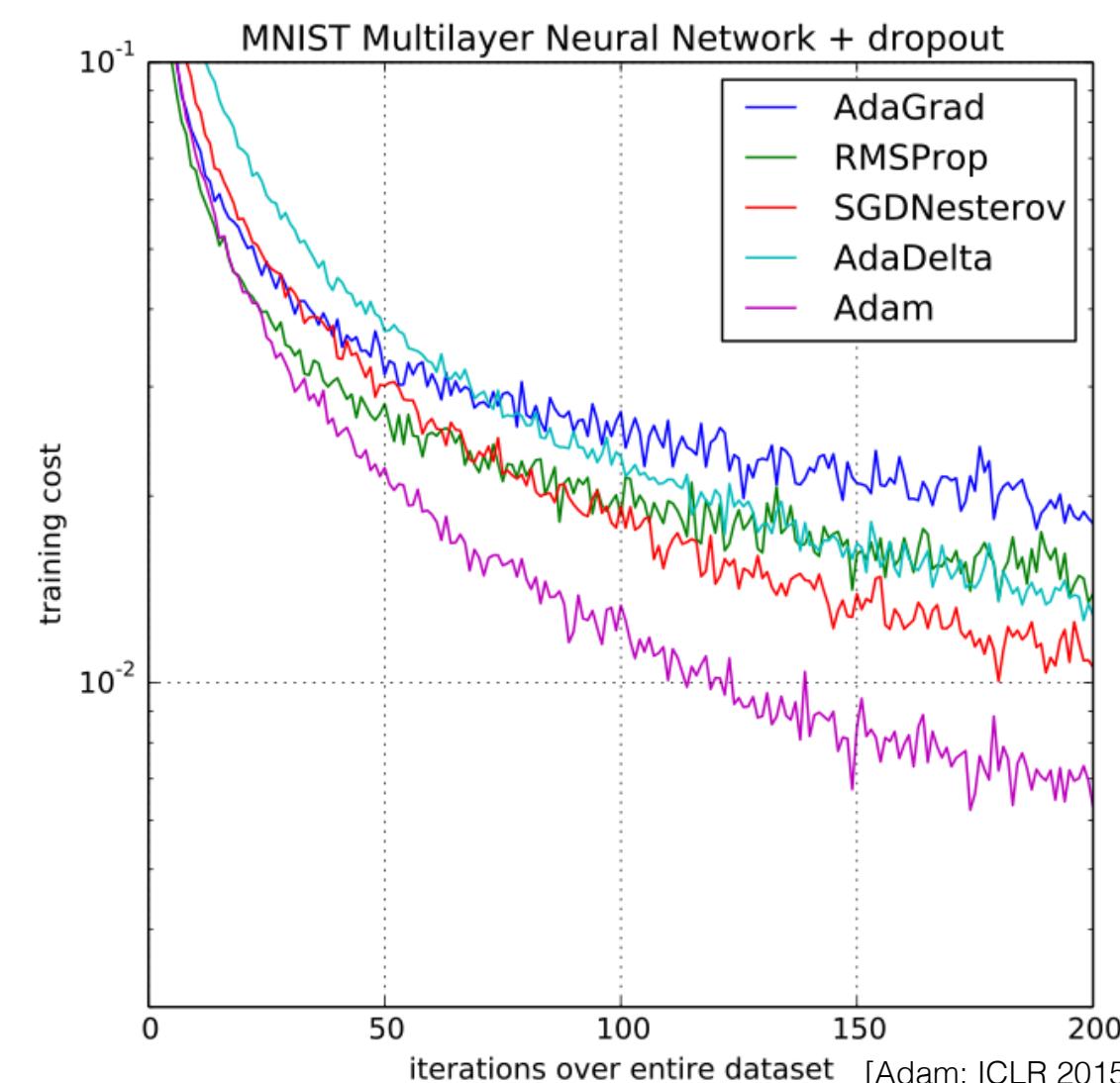
Vanilla update: $w = w - \text{learning_rate} * dw$

Adam update: $w = w - \text{learning_rate} * mb / (\sqrt{vb} + \text{eps})$

(Per parameter adaptive learning rate methods with momentum)

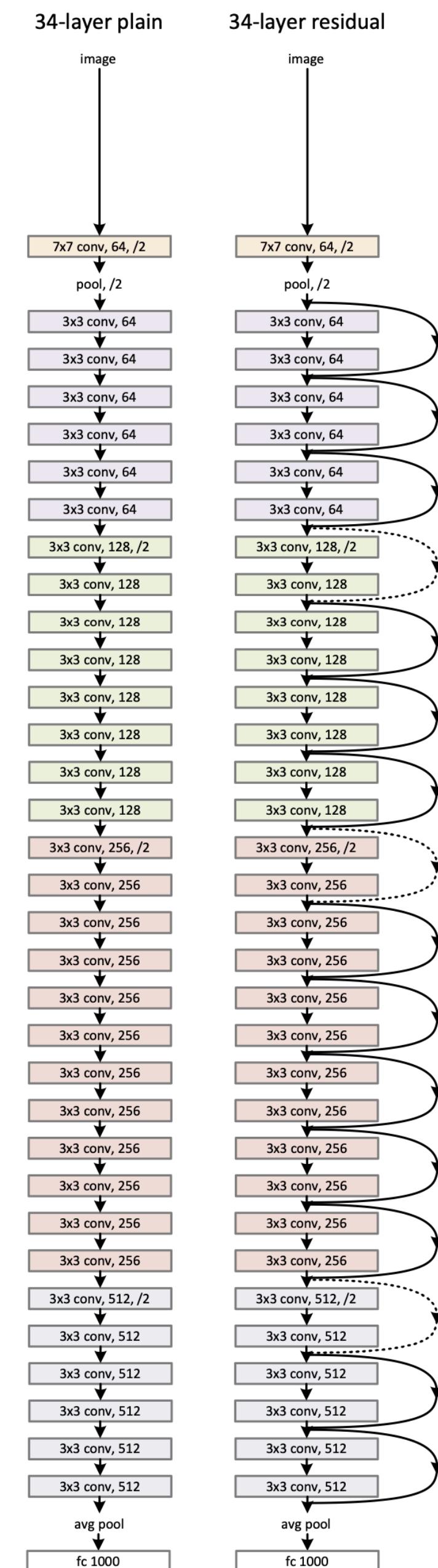
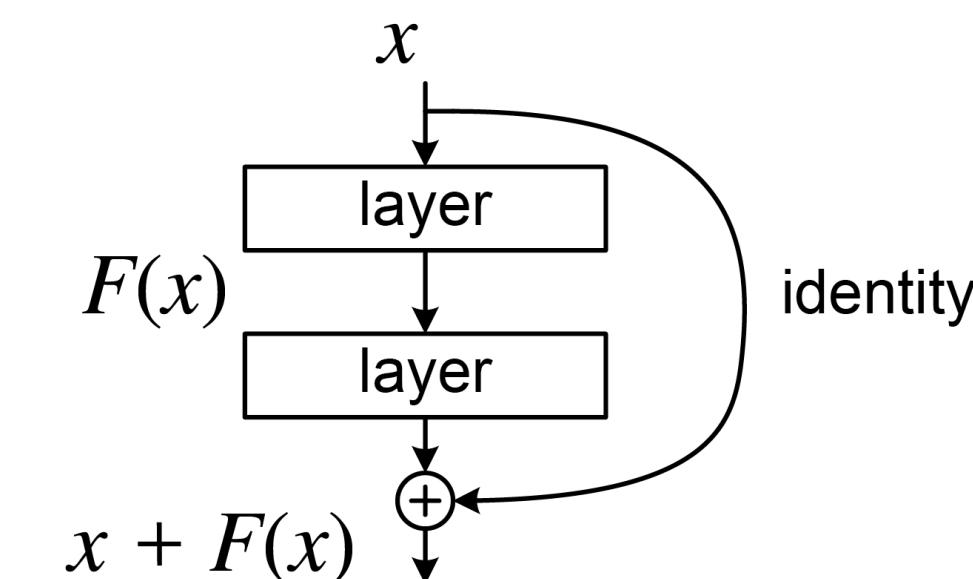


[CS231n - Stanford]



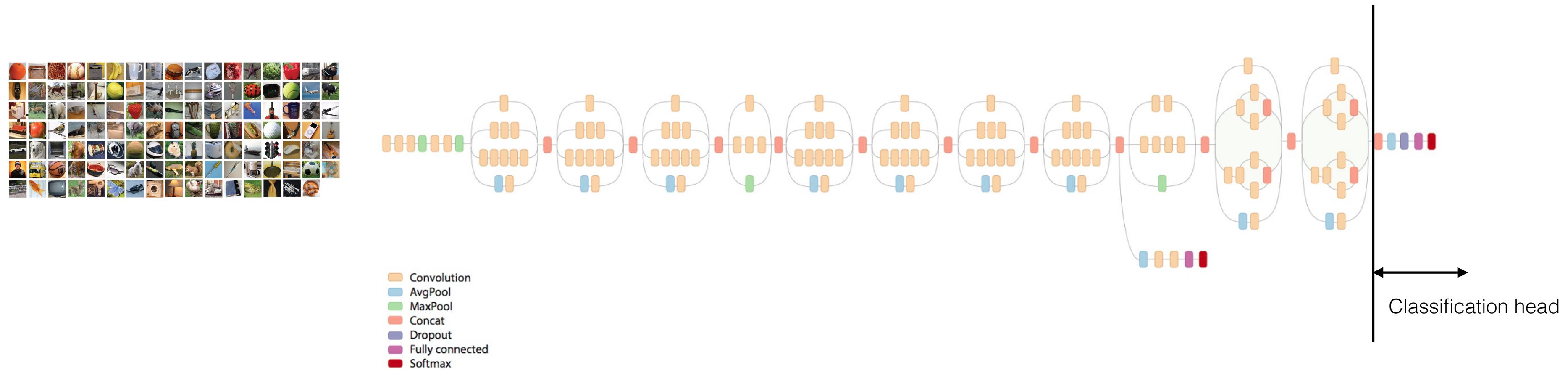
Residual (skip) networks

[Deep Residual Learning for Image Recognition - Microsoft 2016]

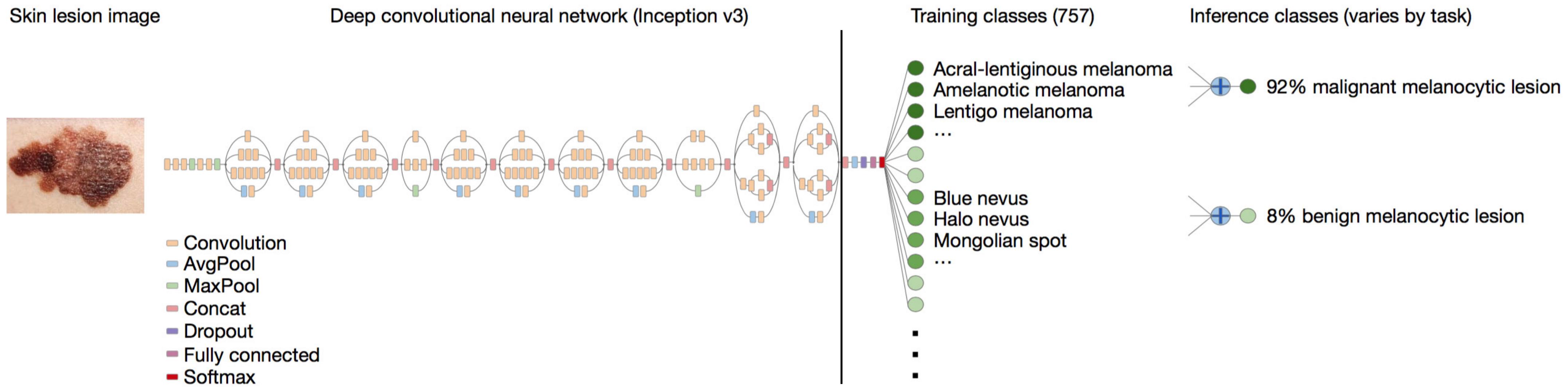


Meta Learning - Transfer Learning

1. Take a model pre-trained on ImageNet (e.g. available on [HuggingFace.co](#)); i.e. Inception V3 model

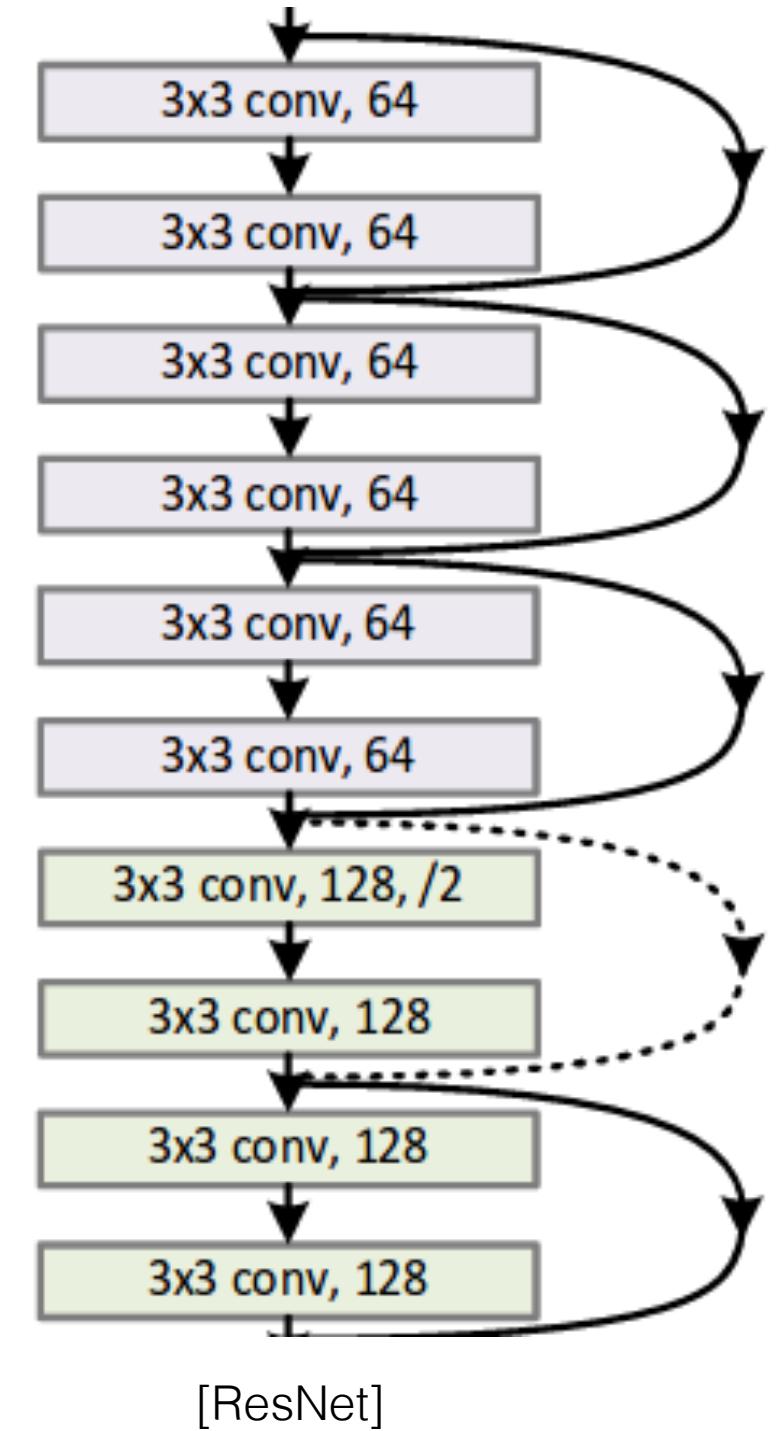
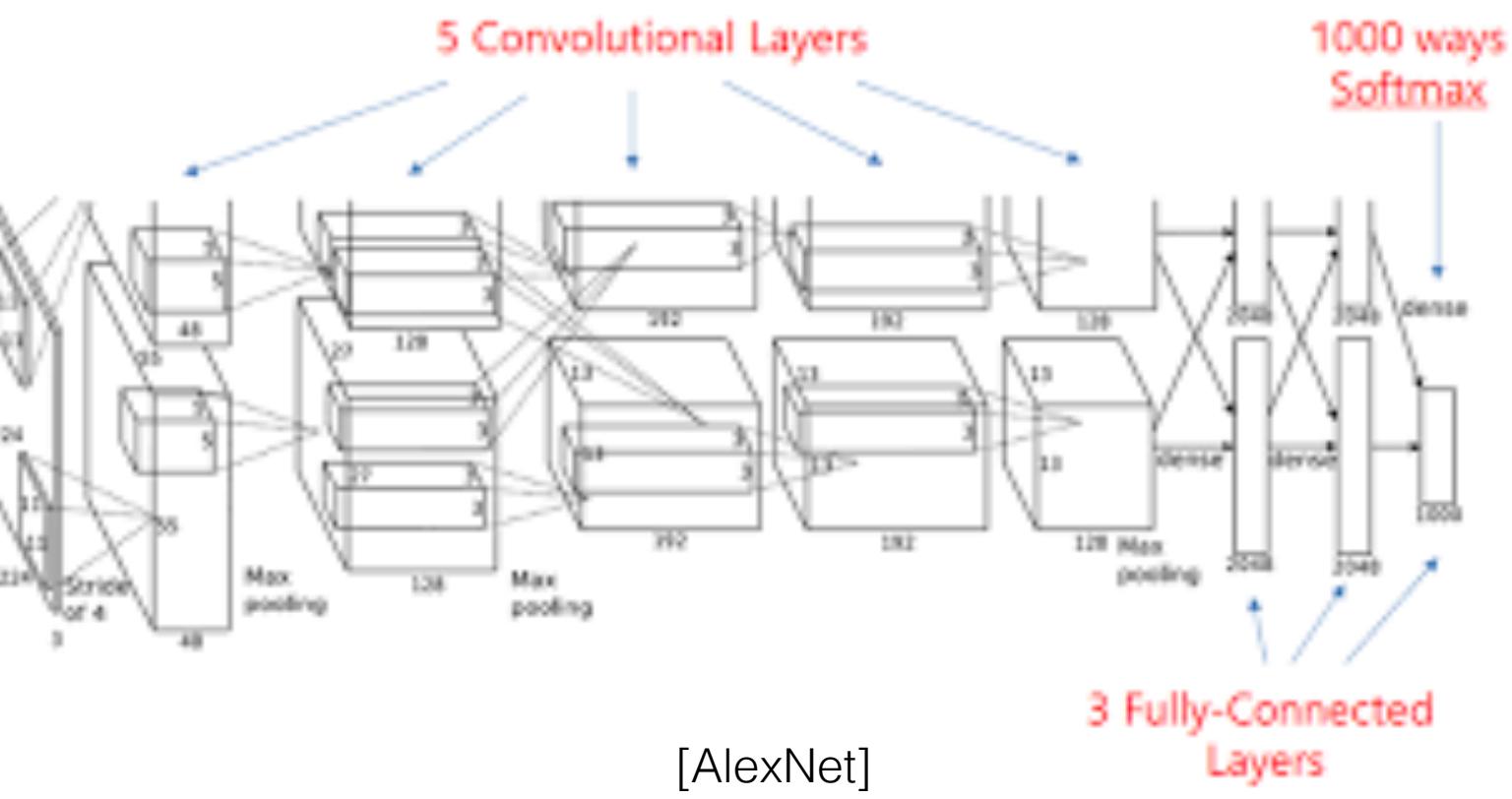
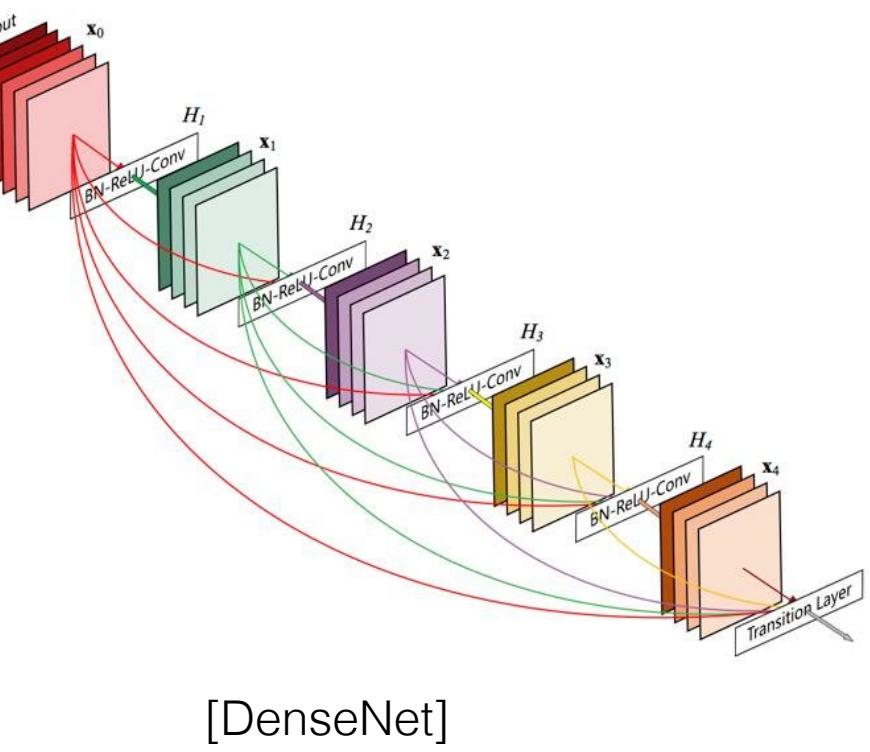
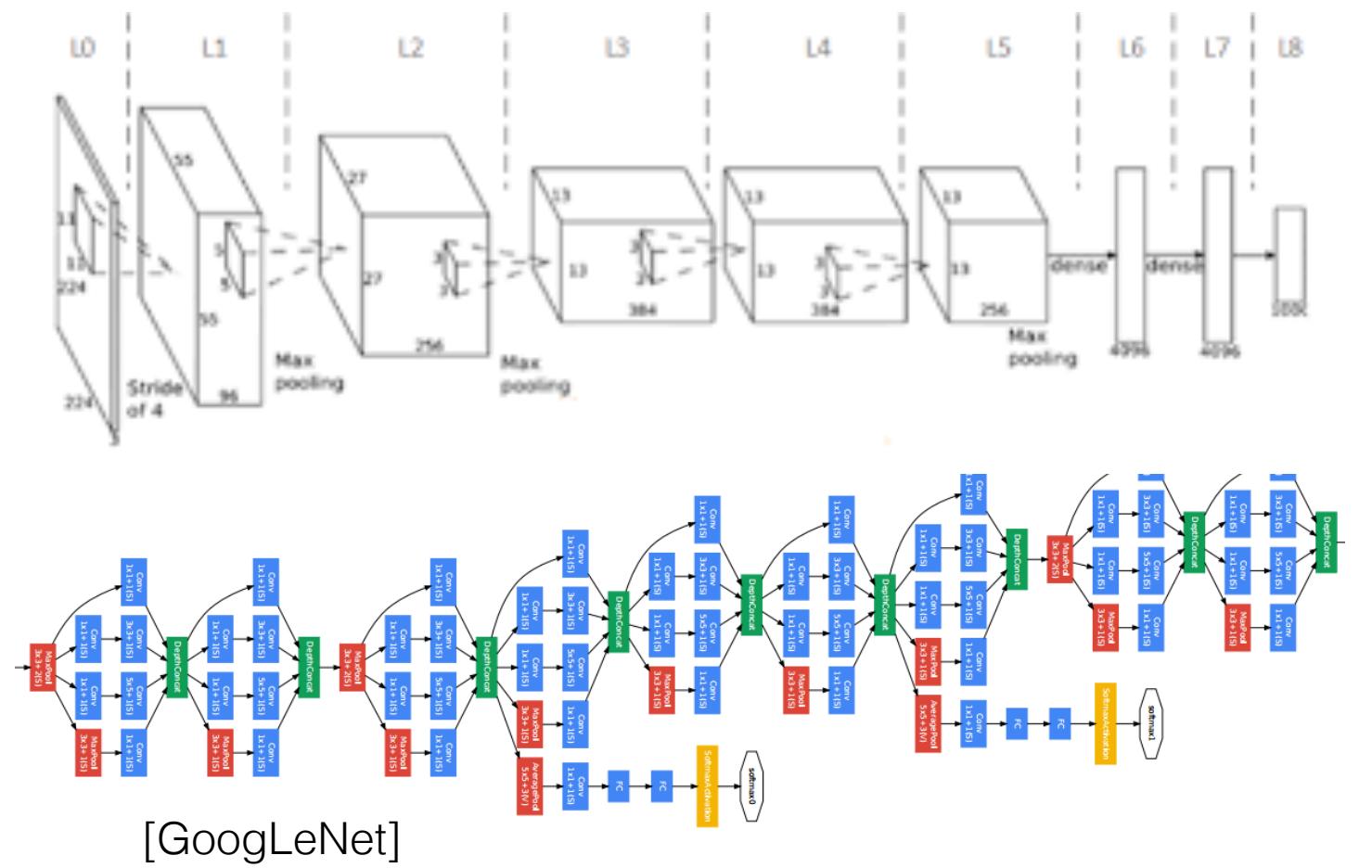


2. Replace classification head and re-train on new smaller data set



[Andre Esteva - Stanford - Skin Cancer Classification with Deep Learning - 2017]

Network architecture search



Manual task: search for optimal model architecture for given dataset

Time consuming - performed by skilled professionals

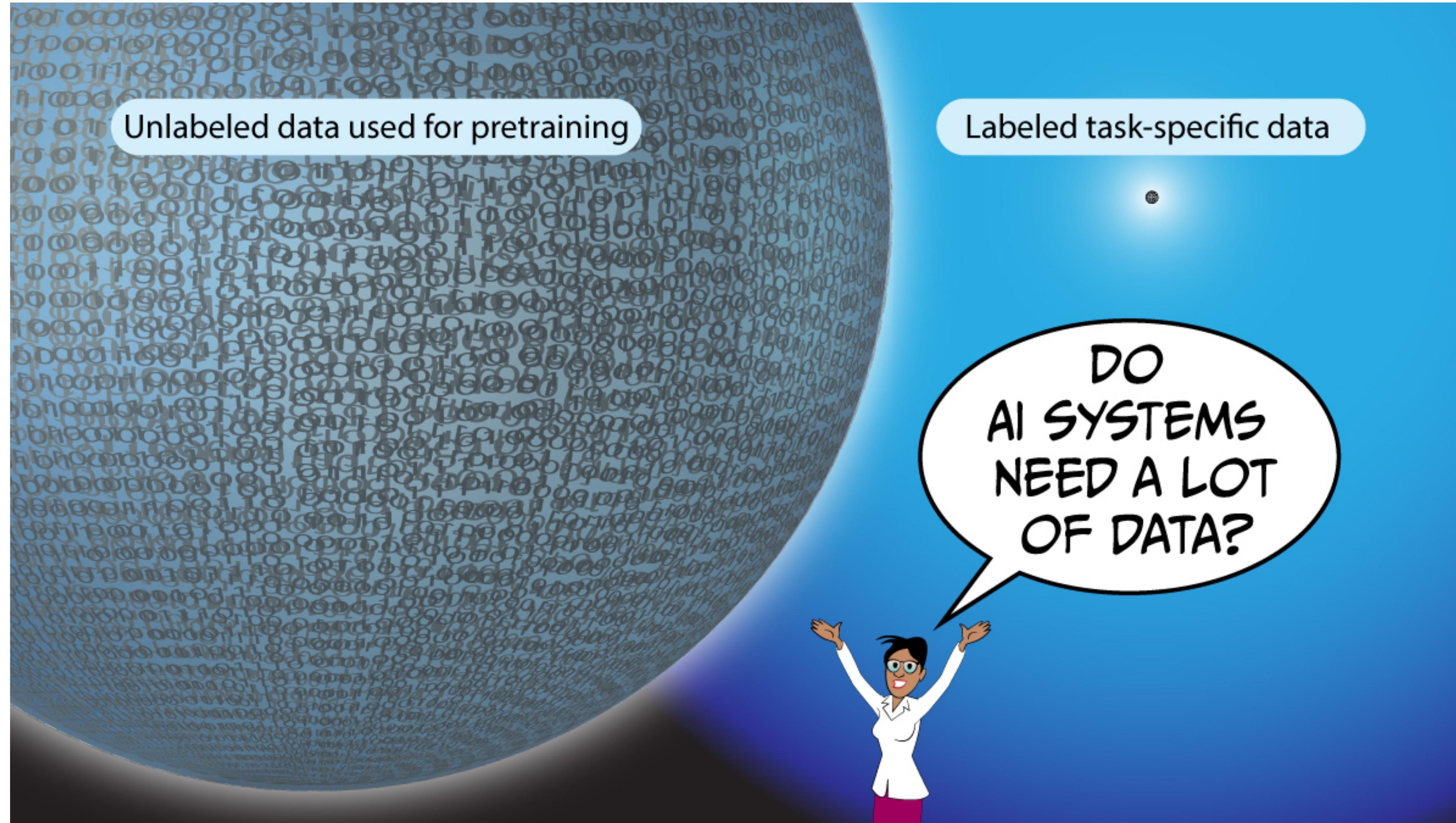
Efficient Neural Architecture Search (NAS - ENAS) (2018)

Google AutoML: cloud.google.com/automl

- Transfer learning + Neural Architecture Search
- Network built layer-wise in search for optimal performance

Efficient Neural Architecture Search via Parameter Sharing

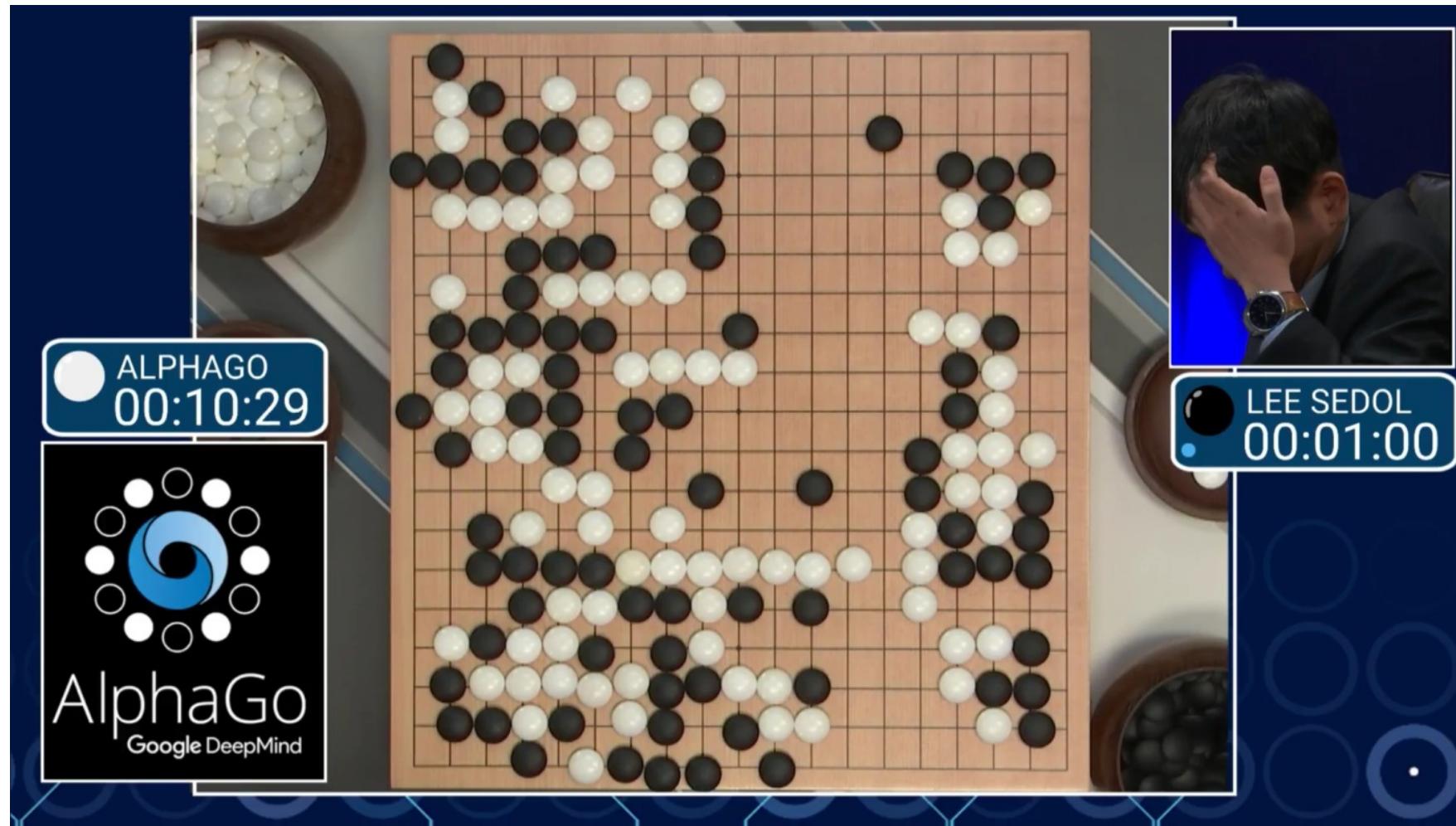
Self-supervised (pre-)training



[The Batch 2023-05]

Tabula Rasa: Learning from scratch

Towards **unsupervised** (self-supervised) learning



AlphaGo (2015)

- Data: Use thousands of games
- Handcrafted features
- Networks: 2 (policy + evaluation)
- Rollouts: fast random games to predict moves
- Big Data, Big Processing Power

AlphaGo Zero (2017)

- Data: **None**
- Handcrafted features: **None**
- Networks: 1
- Rollouts: **None** (computed by network)
- No Data, Less Processing Power
- Better algorithms better than power and data
- Not constrained by limits of human knowledge

deepmind.com/blog/alphago-zero-learning-scratch/

Generative Adversarial Networks (GANs) - 2014

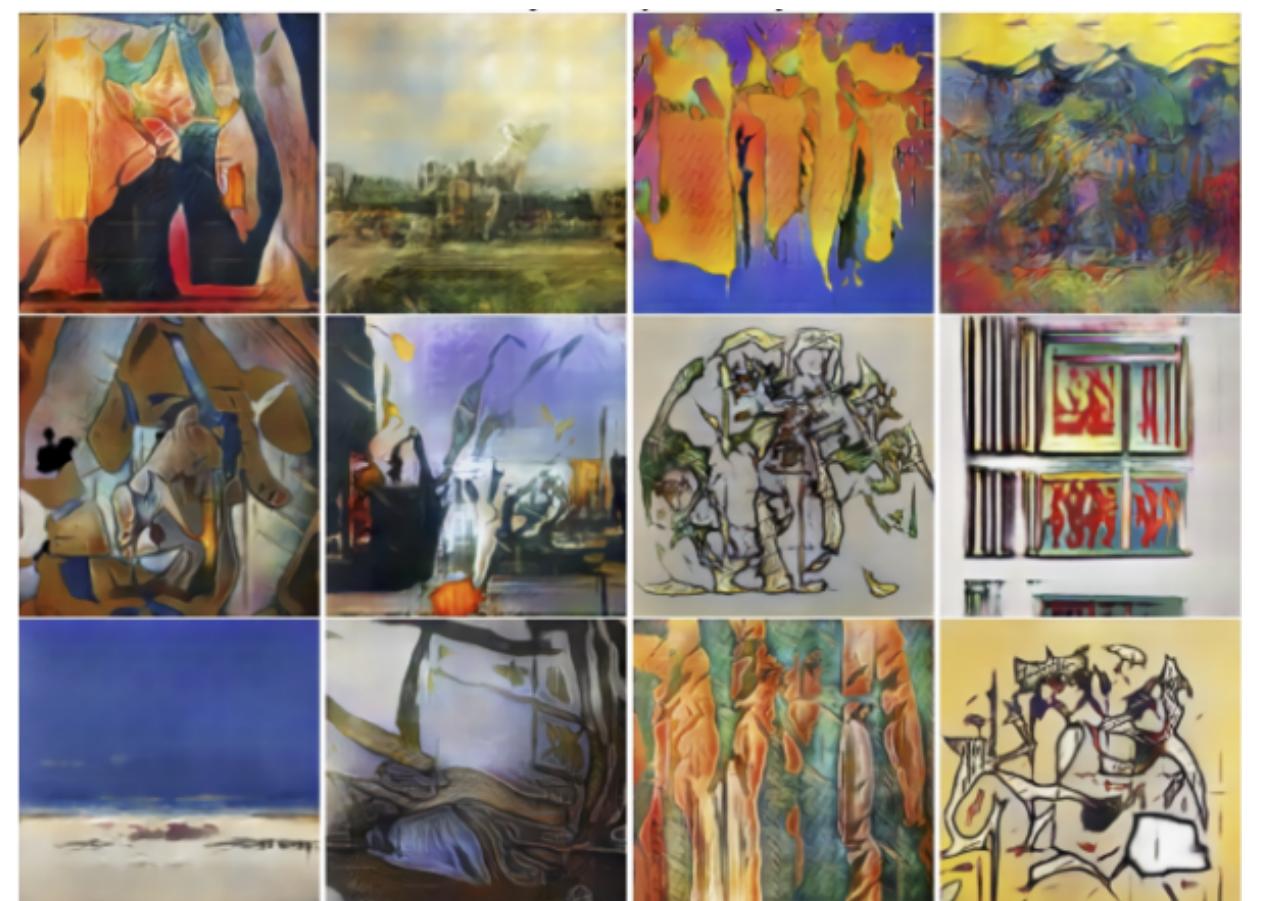
<https://arxiv.org/abs/1406.2661>

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie*, Mehdi Mirza, Bing Xu, David Warde-Farley,
Sherjil Ozair,[†] Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7



Karras et al. (NVIDIA) - ICLR 2018

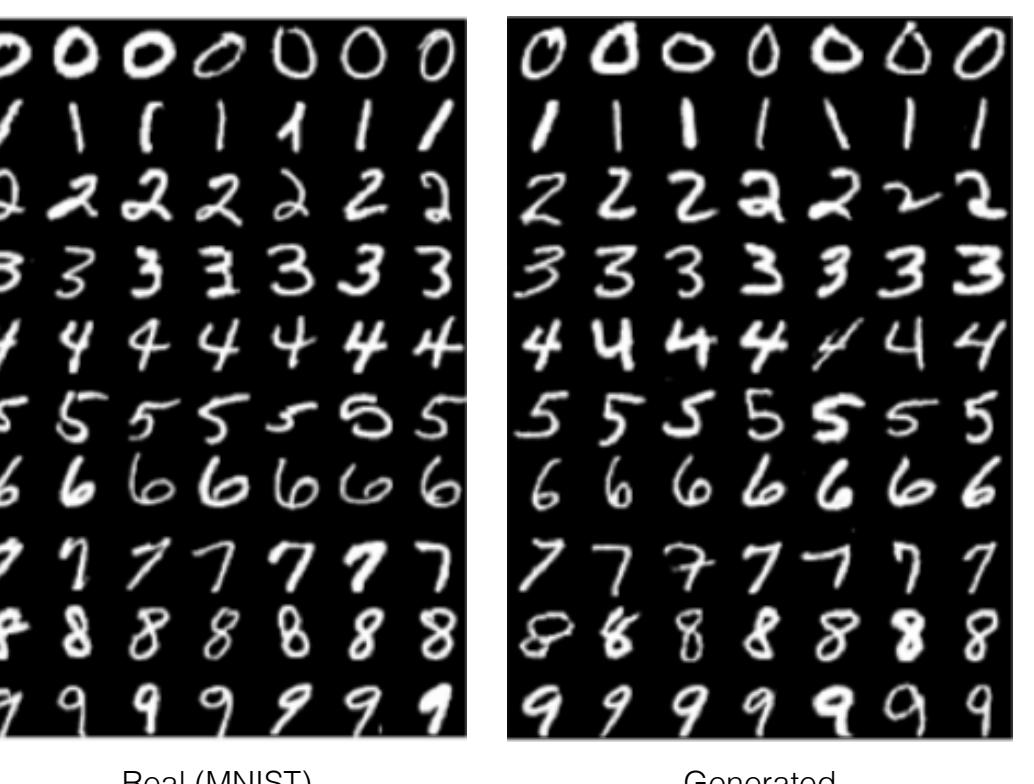


Elgammal et al. ICCC 2017

- Fashion industry: Alibaba, Amazon, ...



Yanghua Jin et al. - 2017



Alec Radford et al. - ICLR 2016

- Music generation,



Only two real images (Spoiler: Row 1, C and Row 2, D) Vue.ai



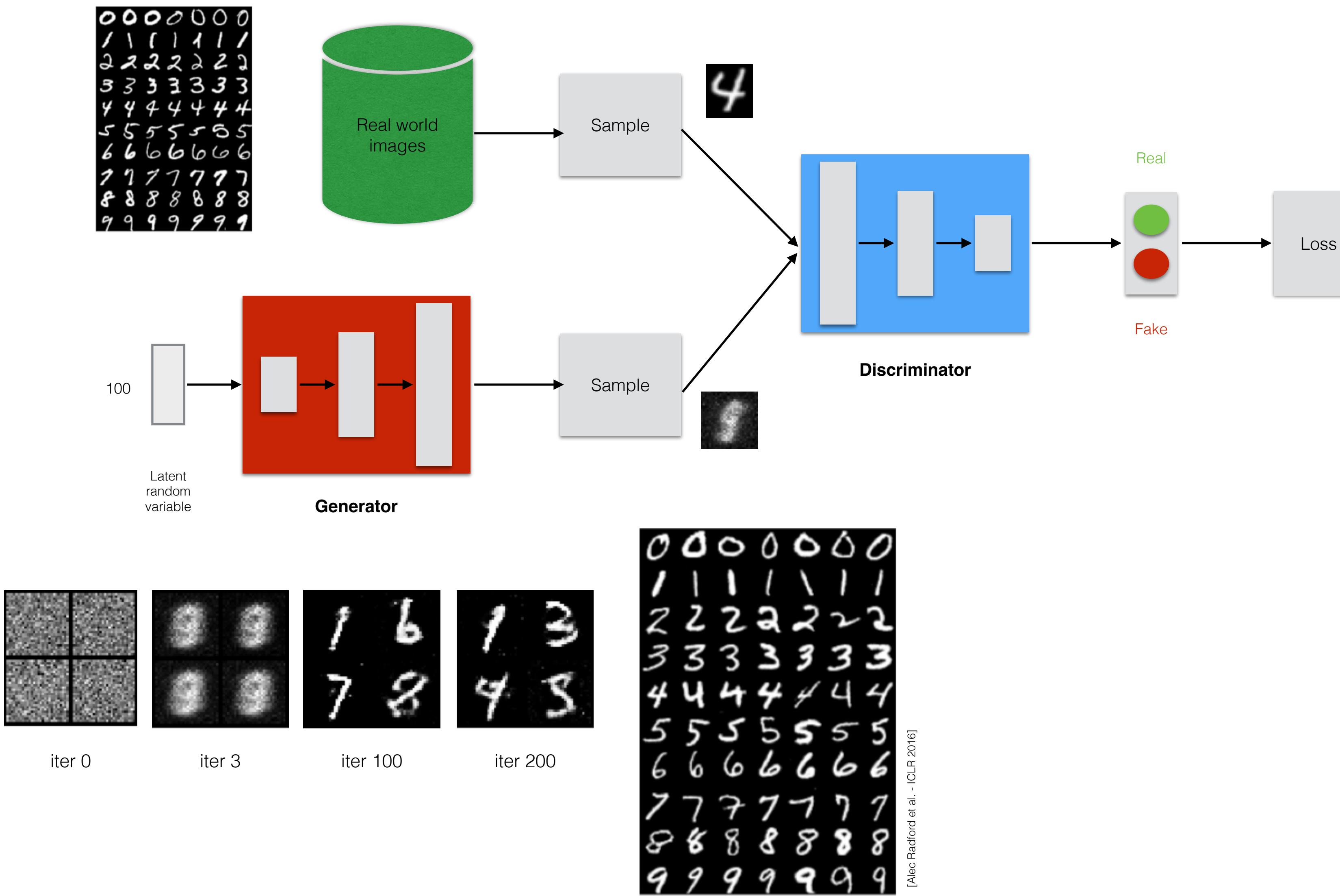
(b) Handbag images (input) & Generated shoe images (output)

DiscoGAN

Generative Adversarial Networks (GANs)

“What I cannot create, I do not understand.” — Richard P. Feynman

GANs from Scratch 1: A deep introduction - Diego Gomez Mosquera (2018)

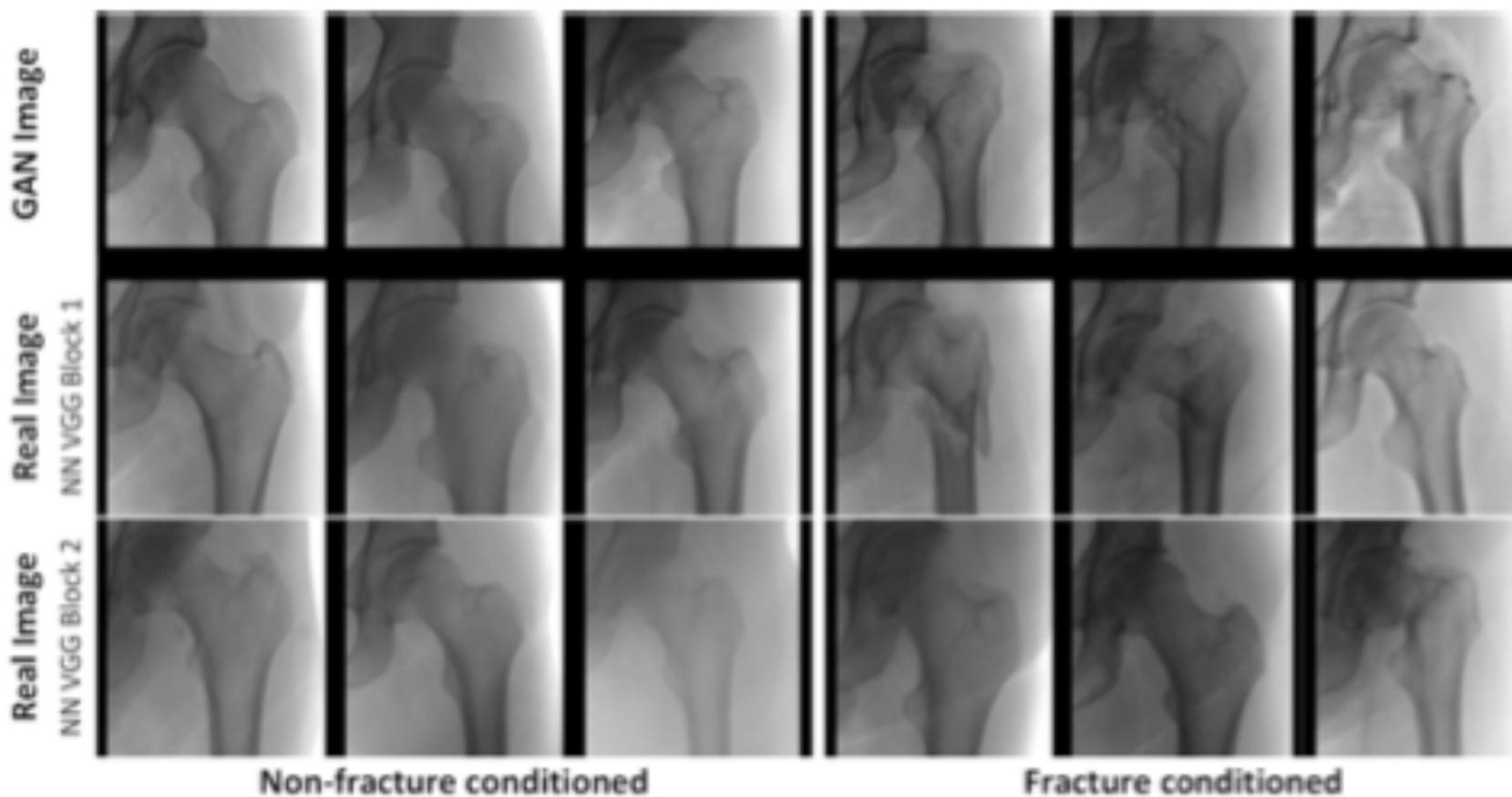


GANs applications

[Submitted on 4 Dec 2018]

Towards generative adversarial networks as a new paradigm for radiology education

Samuel G. Finlayson, Hyunkwang Lee, Isaac S. Kohane, Luke Oakden-Rayner



<https://arxiv.org/abs/1812.01547> (NeurIPS 2018)

[Submitted on 1 Sep 2017 (v1), last revised 14 Feb 2019 (this version, v3)]

PassGAN: A Deep Learning Approach for Password Guessing

Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, Fernando Perez-Cruz

<https://arxiv.org/abs/1709.00440> (code)

<https://github.com/hindupuravinash/the-gan-zoo> (latest update 5 years ago)

- 3D-ED-GAN - [Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks](#)
- 3D-GAN - [Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling](#) ([github](#))
- 3D-IWGAN - [Improved Adversarial Systems for 3D Object Generation and Reconstruction](#) ([github](#))
- 3D-PhysNet - [3D-PhysNet: Learning the Intuitive Physics of Non-Rigid Object Deformations](#)
- 3D-RecGAN - [3D Object Reconstruction from a Single Depth View with Adversarial Learning](#) ([github](#))
- ABC-GAN - [ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks](#) ([github](#))
- ABC-GAN - [GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference](#)
- AC-GAN - [Conditional Image Synthesis With Auxiliary Classifier GANs](#)
- acGAN - [Face Aging With Conditional Generative Adversarial Networks](#)
- ACGAN - [Coverless Information Hiding Based on Generative adversarial networks](#)
- acGAN - [On-line Adaptative Curriculum Learning for GANs](#)
- ACTuAL - [ACTuAL: Actor-Critic Under Adversarial Learning](#)
- AdaGAN - [AdaGAN: Boosting Generative Models](#)
- Adaptive GAN - [Customizing an Adversarial Example Generator with Class-Conditional GANs](#)
- AdvEntuRe - [AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples](#)
- AdvGAN - [Generating adversarial examples with adversarial networks](#)
- AE-GAN - [AE-GAN: adversarial eliminating with GAN](#)
- AE-OT - [Latent Space Optimal Transport for Generative Models](#)
-
- Text2Shape - [Text2Shape: Generating Shapes from Natural Language by Learning Joint Embeddings](#)
- textGAN - [Generating Text via Adversarial Training](#)
- TextureGAN - [TextureGAN: Controlling Deep Image Synthesis with Texture Patches](#)
- TGAN - [Temporal Generative Adversarial Nets](#)
- TGAN - [Tensorizing Generative Adversarial Nets](#)
- TGAN - [Tensor-Generative Adversarial Network with Two-dimensional Sparse Coding: Application to Real-time Indoor Localization](#)
- TGANs-C - [To Create What You Tell: Generating Videos from Captions](#)
- tiny-GAN - [Analysis of Nonautonomous Adversarial Systems](#)
- TP-GAN - [Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis](#)
- TreeGAN - [TreeGAN: Syntax-Aware Sequence Generation with Generative Adversarial Networks](#)
- Triple-GAN - [Triple Generative Adversarial Nets](#)
-

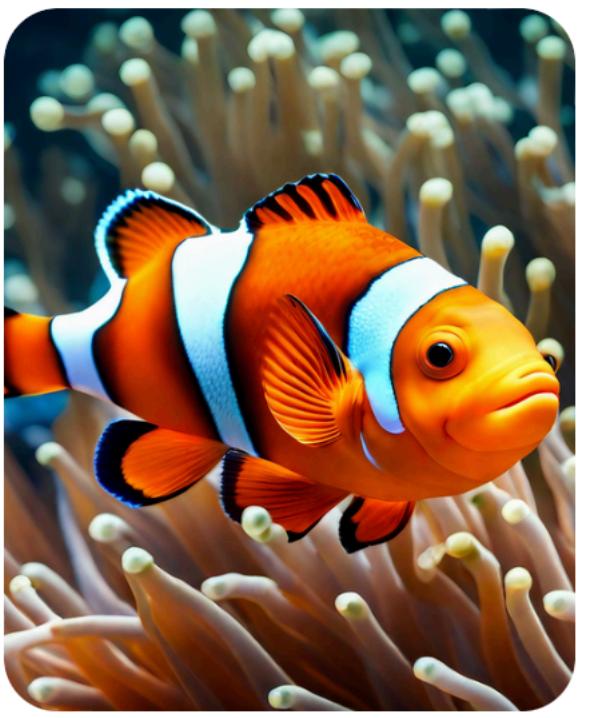
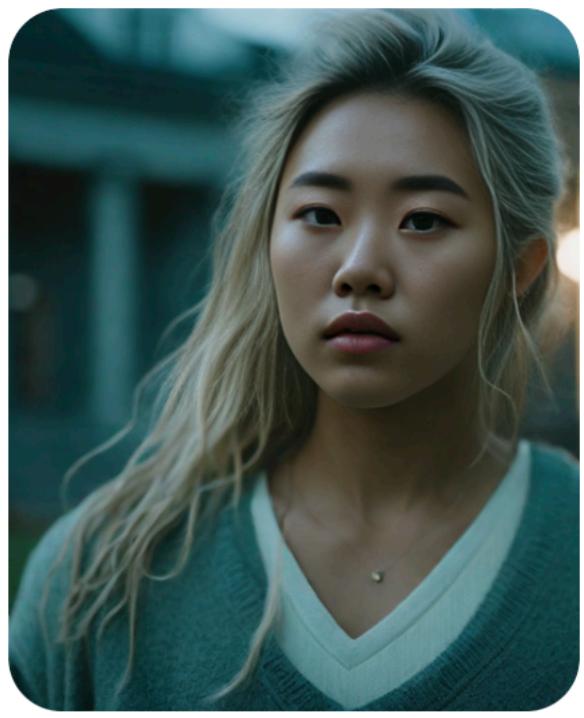
stability.ai

Stable Diffusion

OpenAI Dall-E



Dynamiteffusers



[<https://stability.ai/stable-diffusion>]

Stable Diffusion



Generated with Stable Diffusion

From Deep Learning Foundations to Stable Diffusion

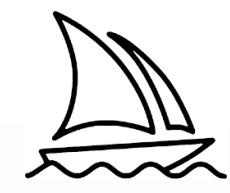
We've released our new course with over 30 hours of video content.

COURSES

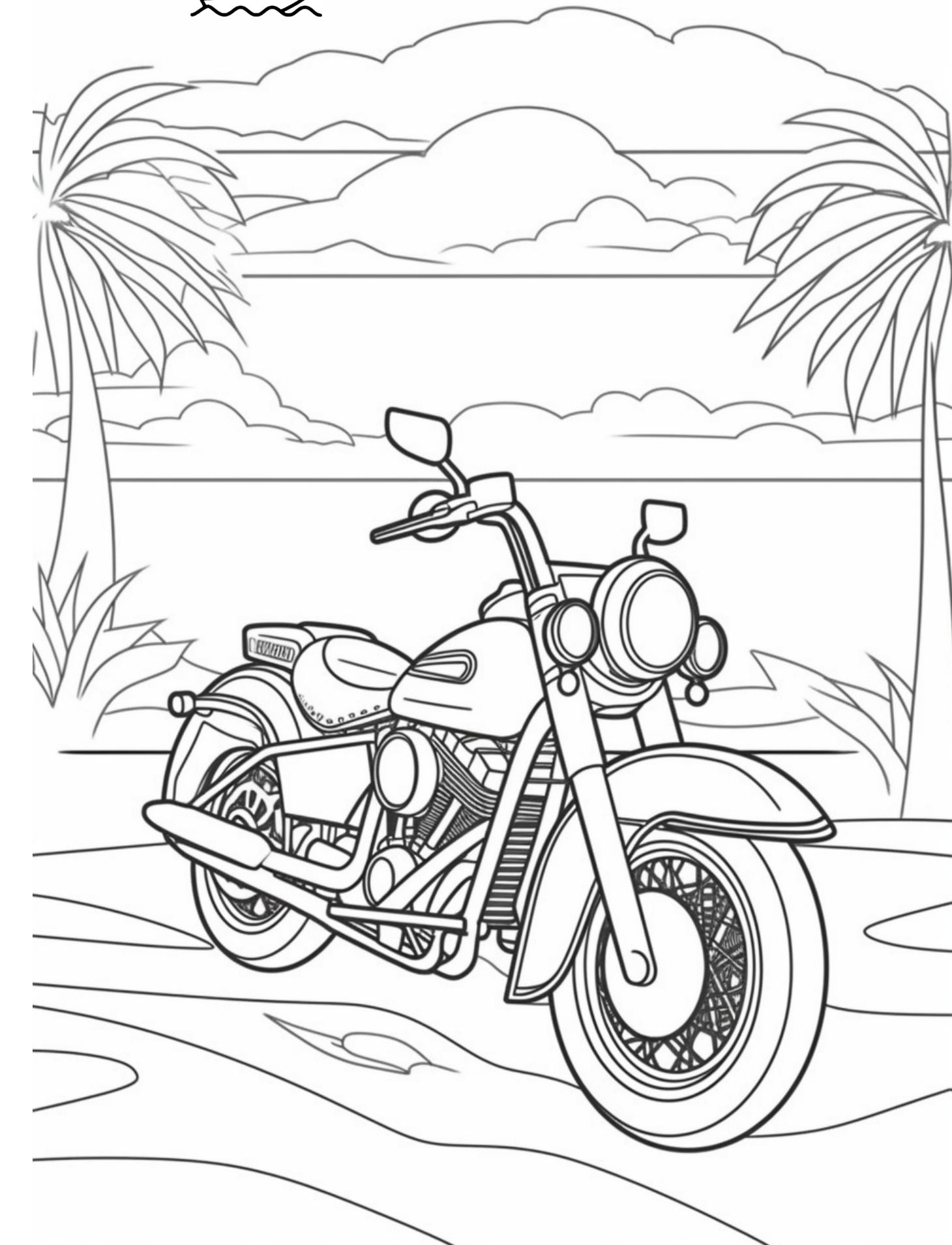
AUTHOR
Jeremy Howard

PUBLISHED
April 4, 2023

<https://course.fast.ai/Lessons/part2.html>



midjourney.com



Midjourney Bot ✅ BOT Today at 9:45 PM
A simple coloring book page for kids of a west coast style chopper motorcycle on the beach, all black and white, line art --ar 8:11 --v 5 --s 750 --c 50 - Image #4 @gooner925

Word embeddings

Recurrent Neural Networks

Transformers

Large Language Models

Latest “programming” language: natural language e.g. French (prompt engineering)

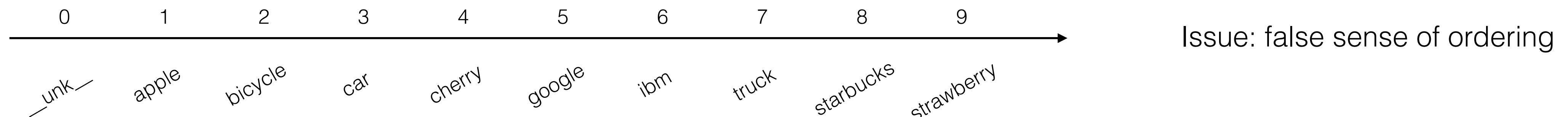
NLP: word representations

Vocabulary: [unk, apple, bicycle, car, cherry, google, ibm, truck, starbucks, strawberry]

Word IDs: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Encoding schemes

Option 0: Using the word IDs



Option 1: one-hot encoding

<u>unk</u>	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	google	(0, 0, 0, 0, 0, 1, 0, 0, 0, 0)
apple	(0, 1, 0, 0, 0, 0, 0, 0, 0, 0)	ibm	(0, 0, 0, 0, 0, 0, 1, 0, 0, 0)
bicycle	(0, 0, 1, 0, 0, 0, 0, 0, 0, 0)	truck	(0, 0, 0, 0, 0, 0, 0, 1, 0, 0)
car	(0, 0, 0, 1, 0, 0, 0, 0, 0, 0)	starbucks	(0, 0, 0, 0, 0, 0, 0, 0, 1, 0)
cherry	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0)	strawberry	(0, 0, 0, 0, 0, 0, 0, 0, 0, 1)

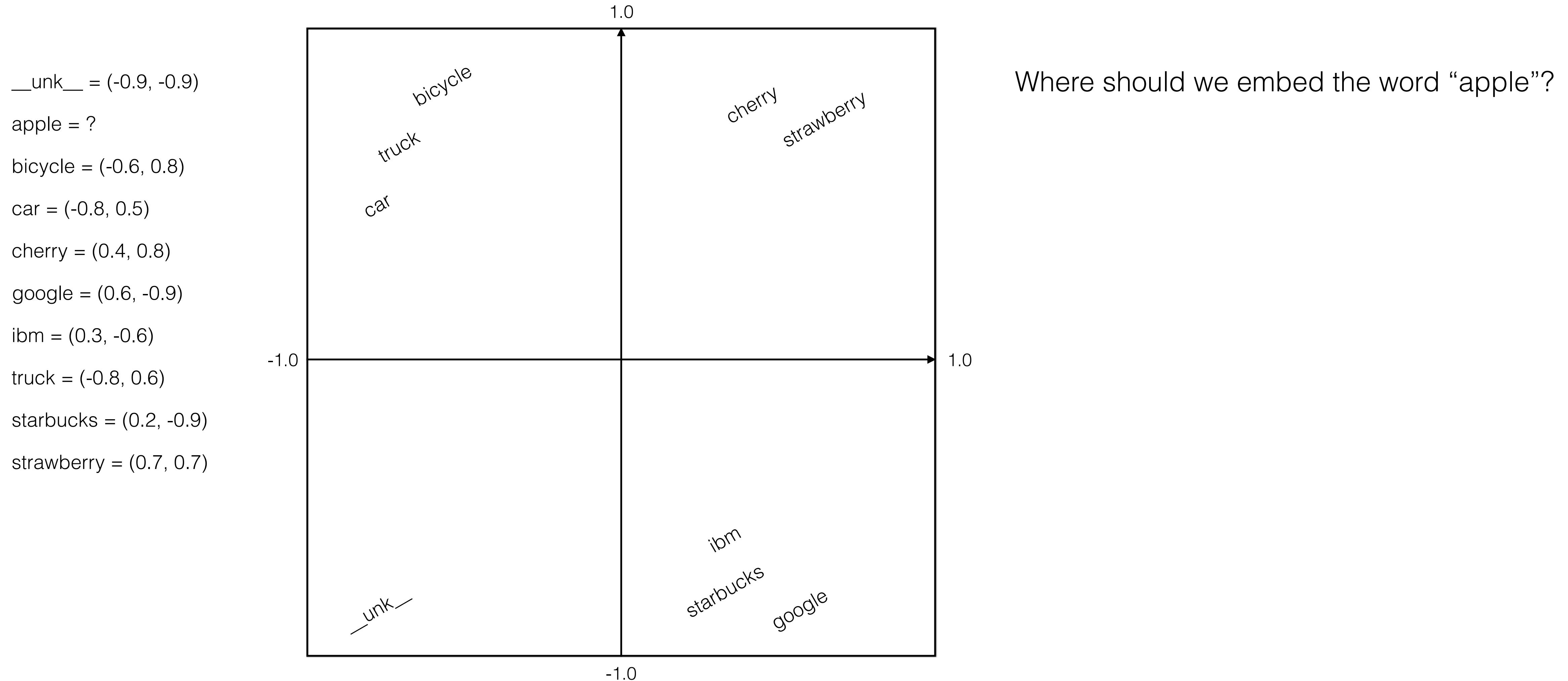
Issues:

- Does not scale (vocab. size 10^6)
- No notion of word semantic similarity
e.g. (car, truck, strawberry)

NLP: word embeddings

“word embedding” refers to a technique used to represent words or phrases as **dense** numerical vectors in a high-dimensional space (ChatGPT)

Vocabulary: [__unk__, apple, bicycle, car, cherry, google, ibm, truck, starbucks, strawberry]



NLP: word2vec, GloVe, fastText, ...

Efficient Estimation of Word Representations in Vector Space

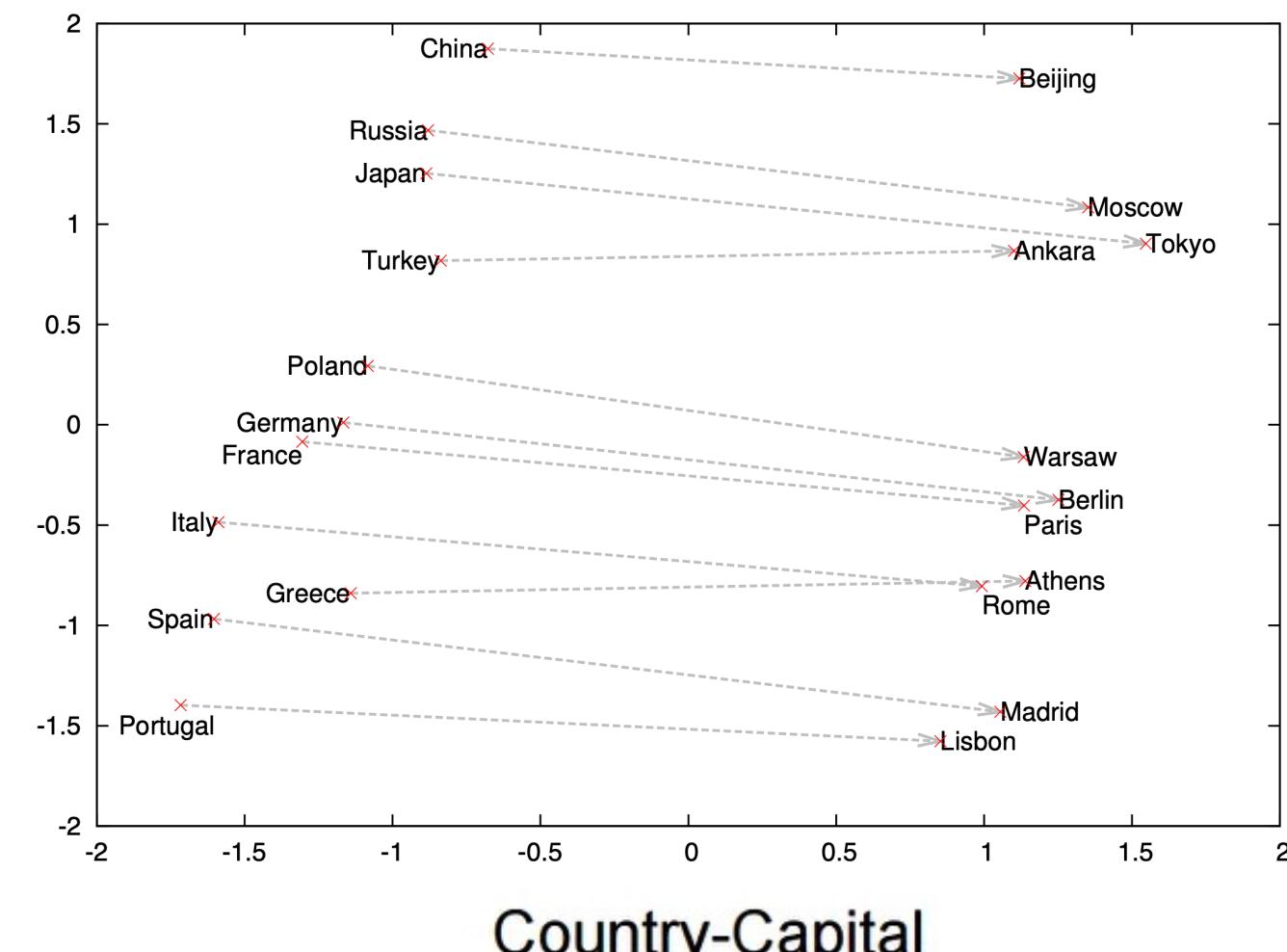
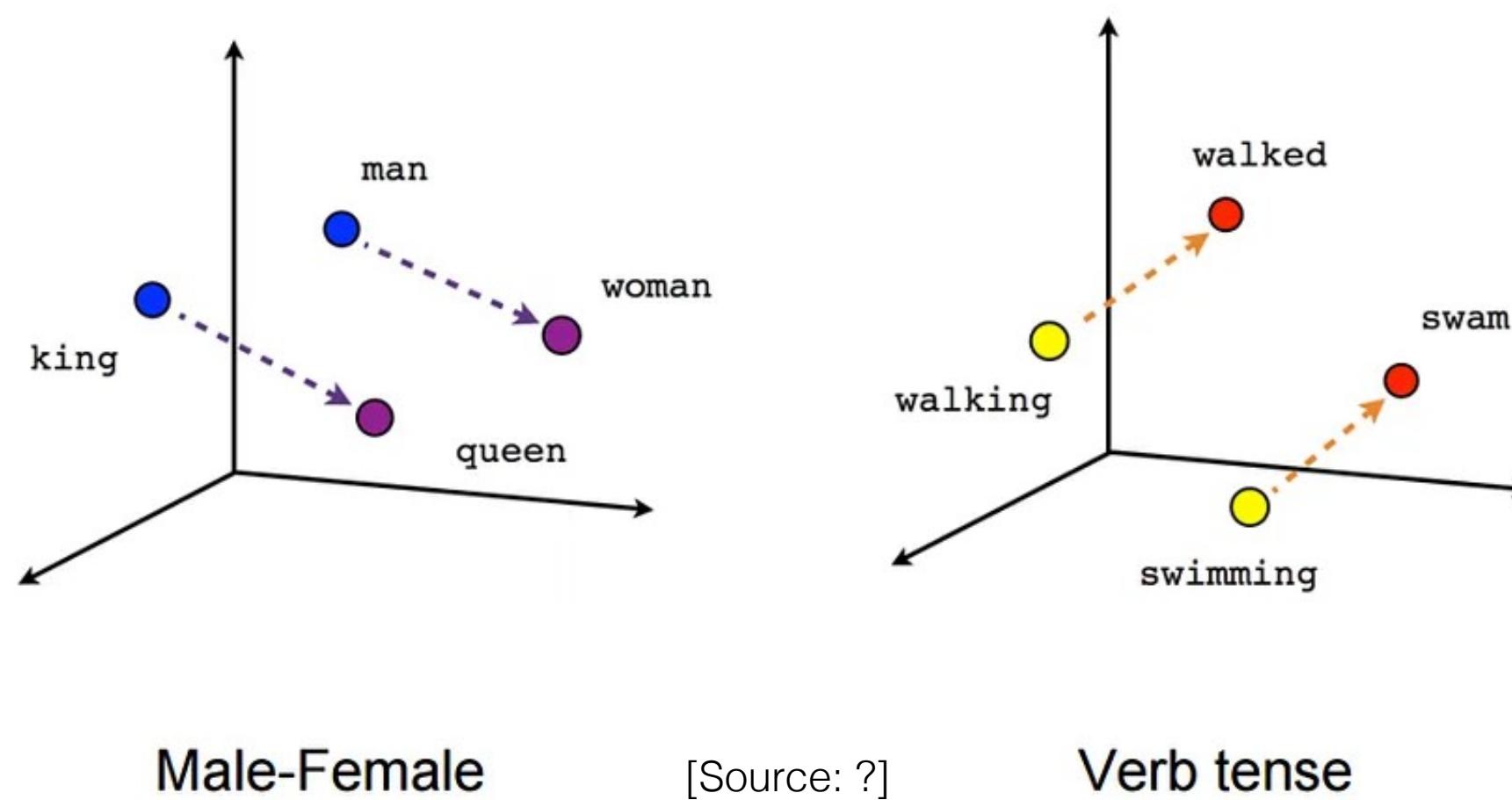
Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

word2vec (2013)



[word2vec (2013)]

- ⊕ Revolutionary, simple, self-supervised, only 2 matrices of size (vocab size, embedding dimension)
- ⊖ All unseen words are still matched to the same embedding vector

Enriching Word Vectors with Subword Information

Piotr Bojanowski* and **Edouard Grave*** and **Armand Joulin** and **Tomas Mikolov**

Facebook AI Research

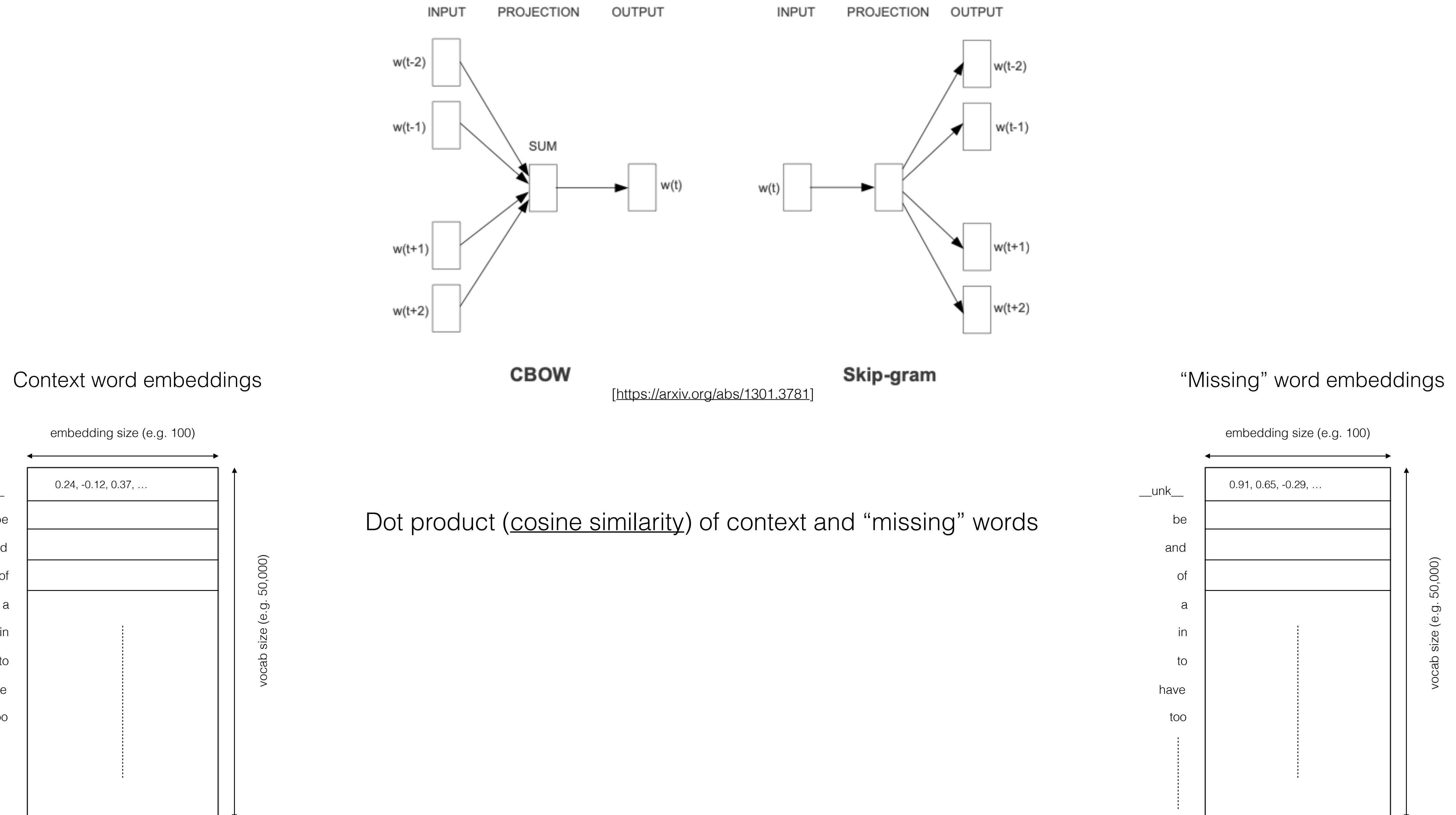
{bojanowski,egrave,ajoulin,tmikolov}@fb.com

Enriching Word Vectors with Subword Information (2017)

unadvised = {<unadvised> + <un + una + nad + adv + dvi + vis + ise + sed + ed>}

- ⊕ Getting embeddings even for “unseen” words. No more _unk_ symbol.
- ⊖ Ambiguous words (e.g. “apple”, “run”) get a single embedding vector

Word2vec algorithm

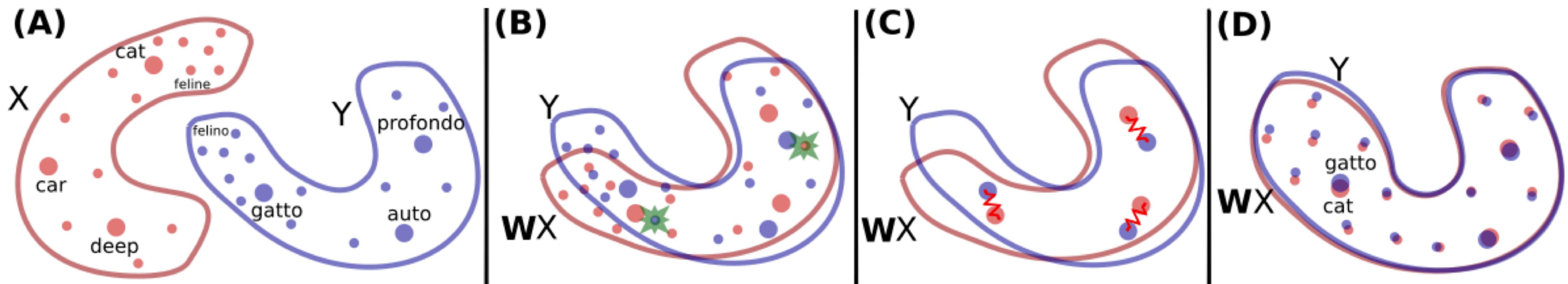


NLP: word embeddings: multilingual alignment

Published as a conference paper at ICLR 2018

WORD TRANSLATION WITHOUT PARALLEL DATA

Alexis Conneau^{*†‡}, Guillaume Lample^{*†§},
Marc'Aurelio Ranzato[†], Ludovic Denoyer[§], Hervé Jégou[†]
`{aconneau, glample, ranzato, rvj}@fb.com`
`ludovic.denoyer@upmc.fr`



The Procrustes problem

(Neural) Language Models

Journal of Machine Learning Research 3 (2003) 1137–1155

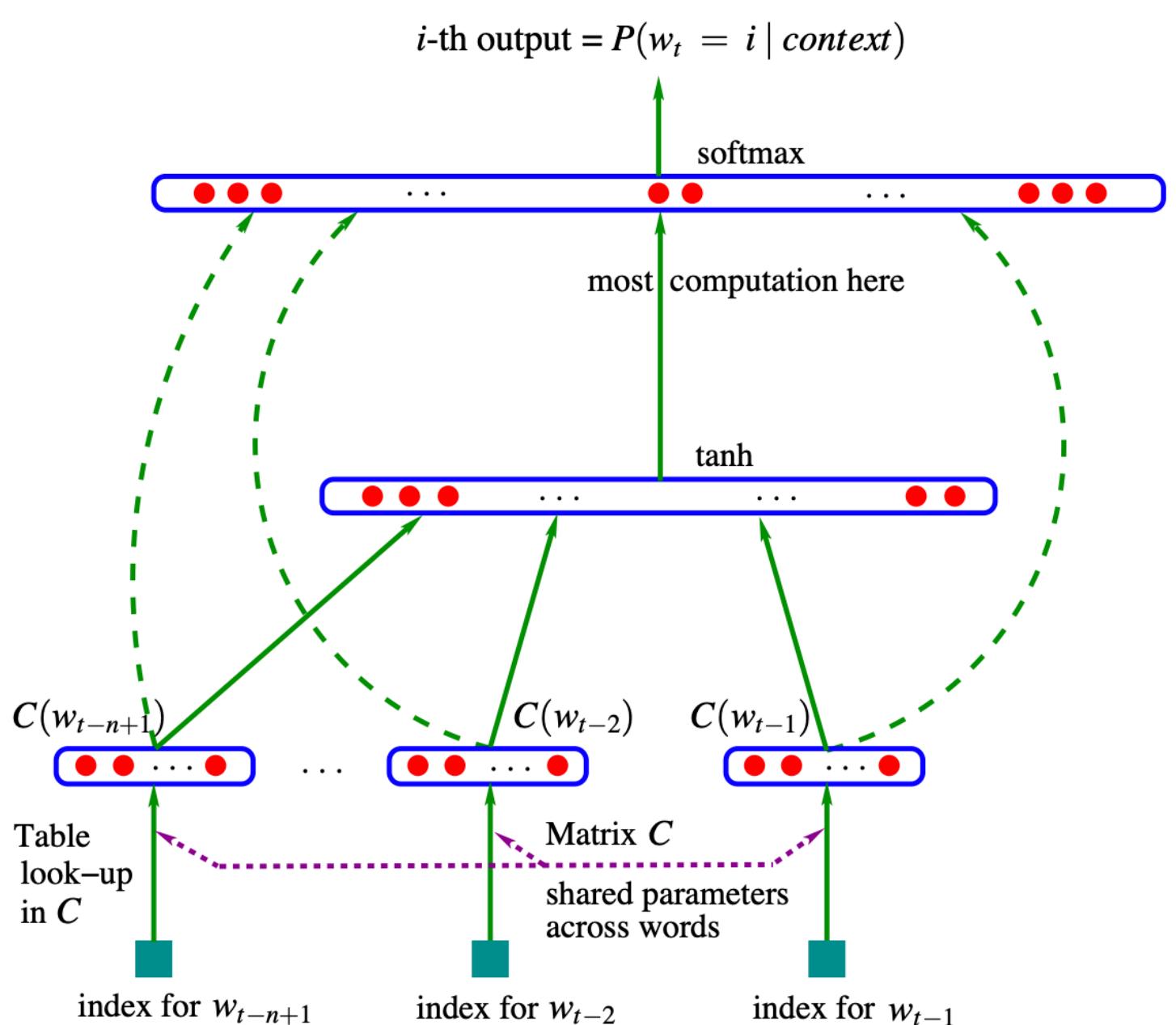
Submitted 4/02; Published 2/03

A Neural Probabilistic Language Model

Yoshua Bengio
Réjean Ducharme
Pascal Vincent
Christian Jauvin

Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal, Montréal, Québec, Canada

BENGIOY@IRO.UMONTREAL.CA
DUCHARME@IRO.UMONTREAL.CA
VINCENTP@IRO.UMONTREAL.CA
JAUVINC@IRO.UMONTREAL.CA



Neural Machine Translation: RNN, LSTM (2014)

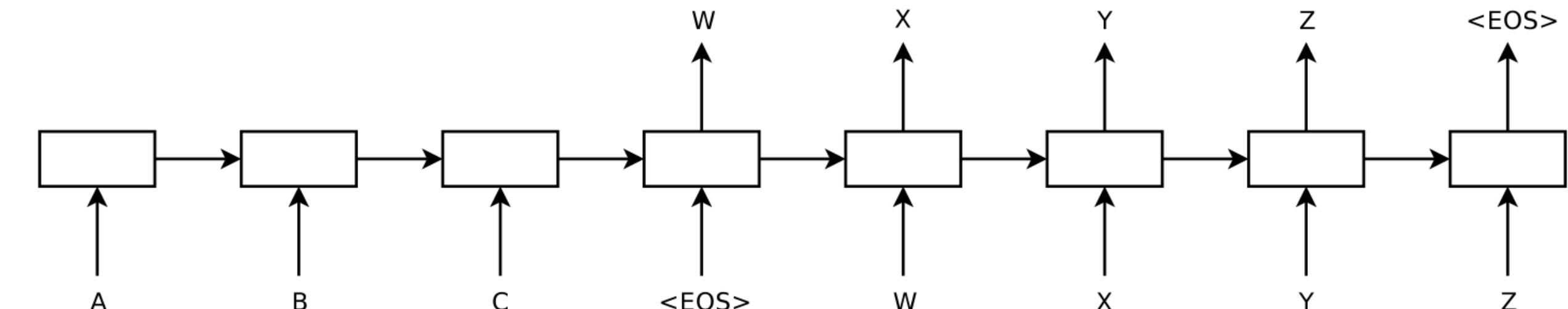
Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

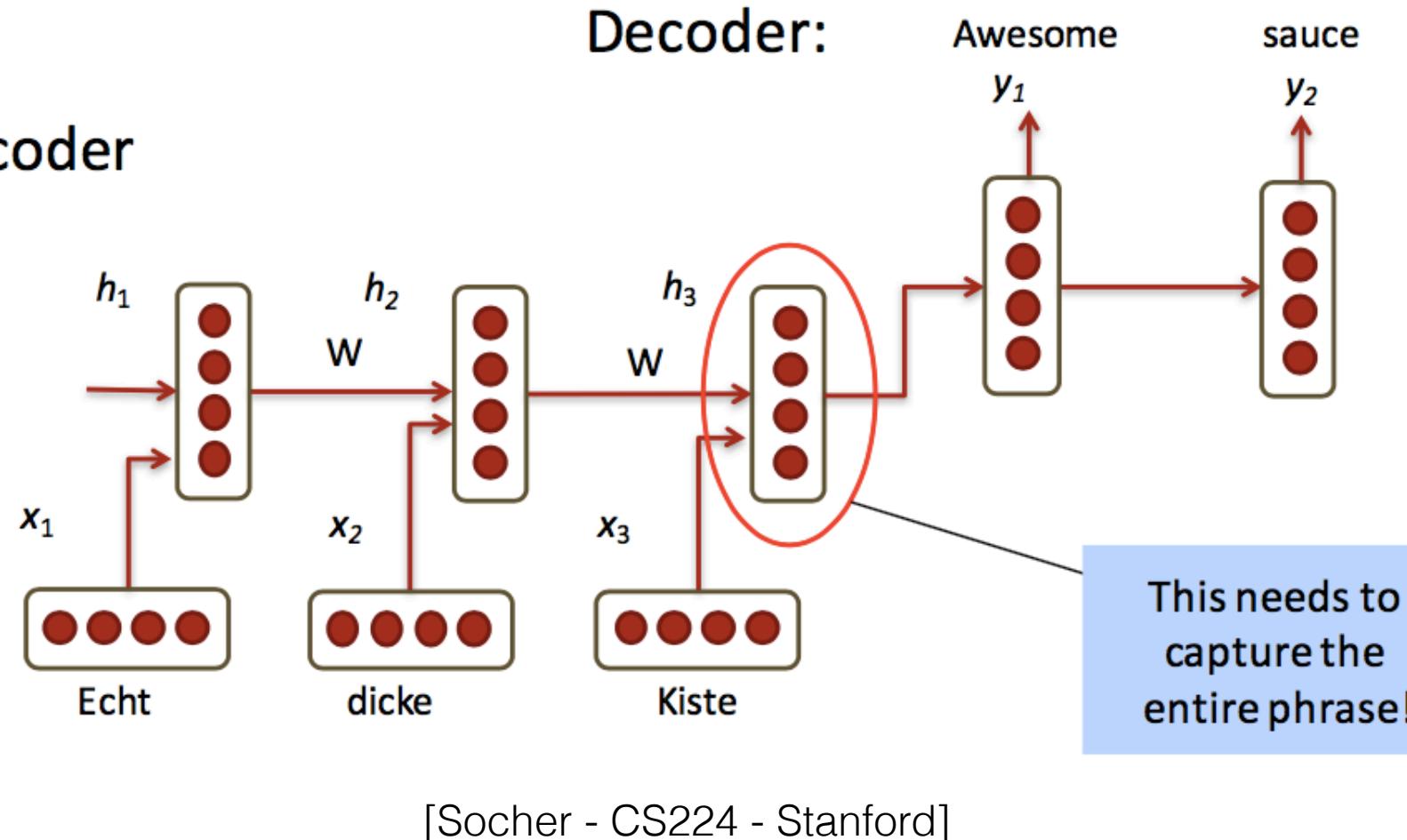
Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

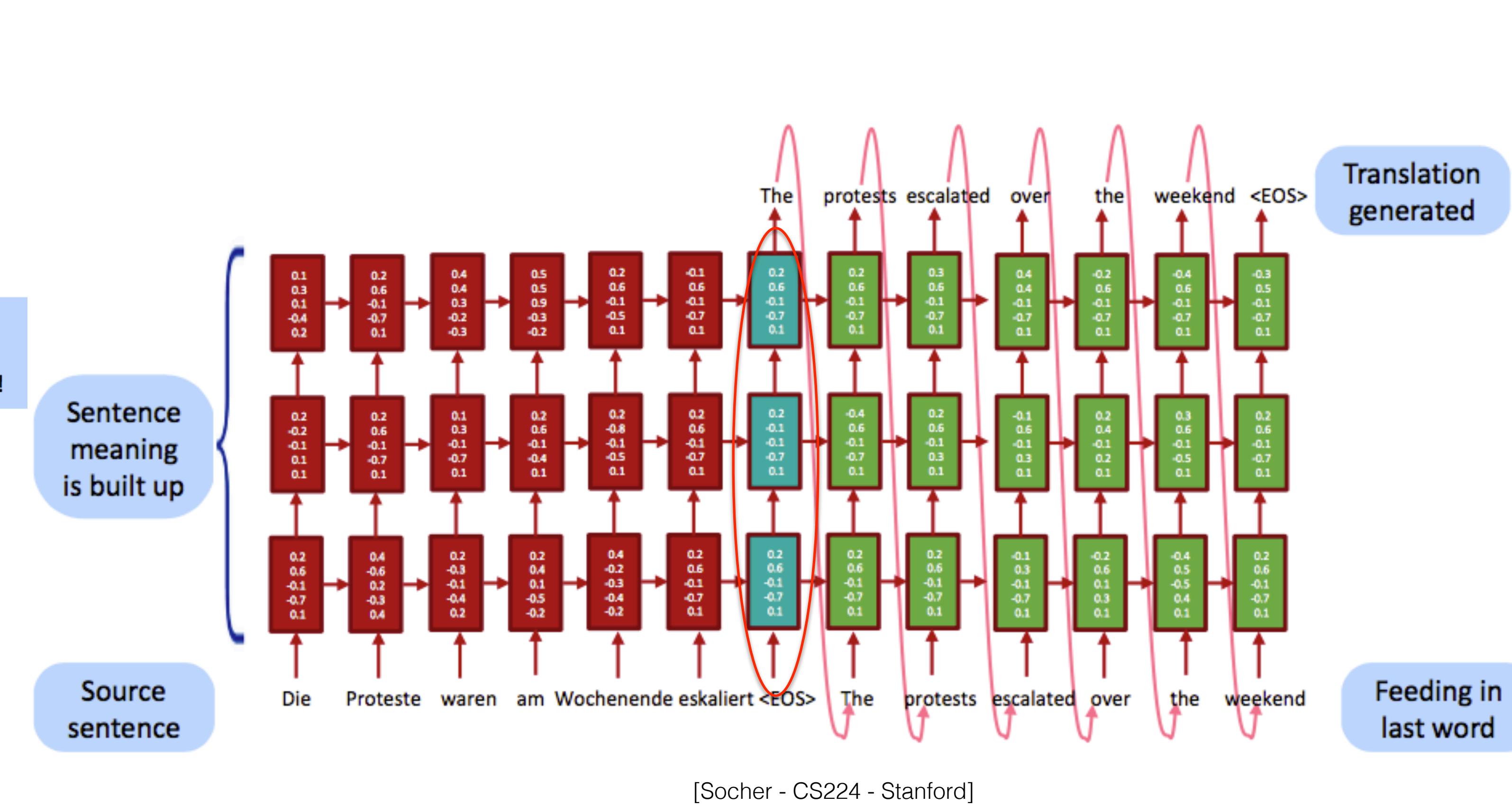
2014



Encoder



[Socher - CS224 - Stanford]

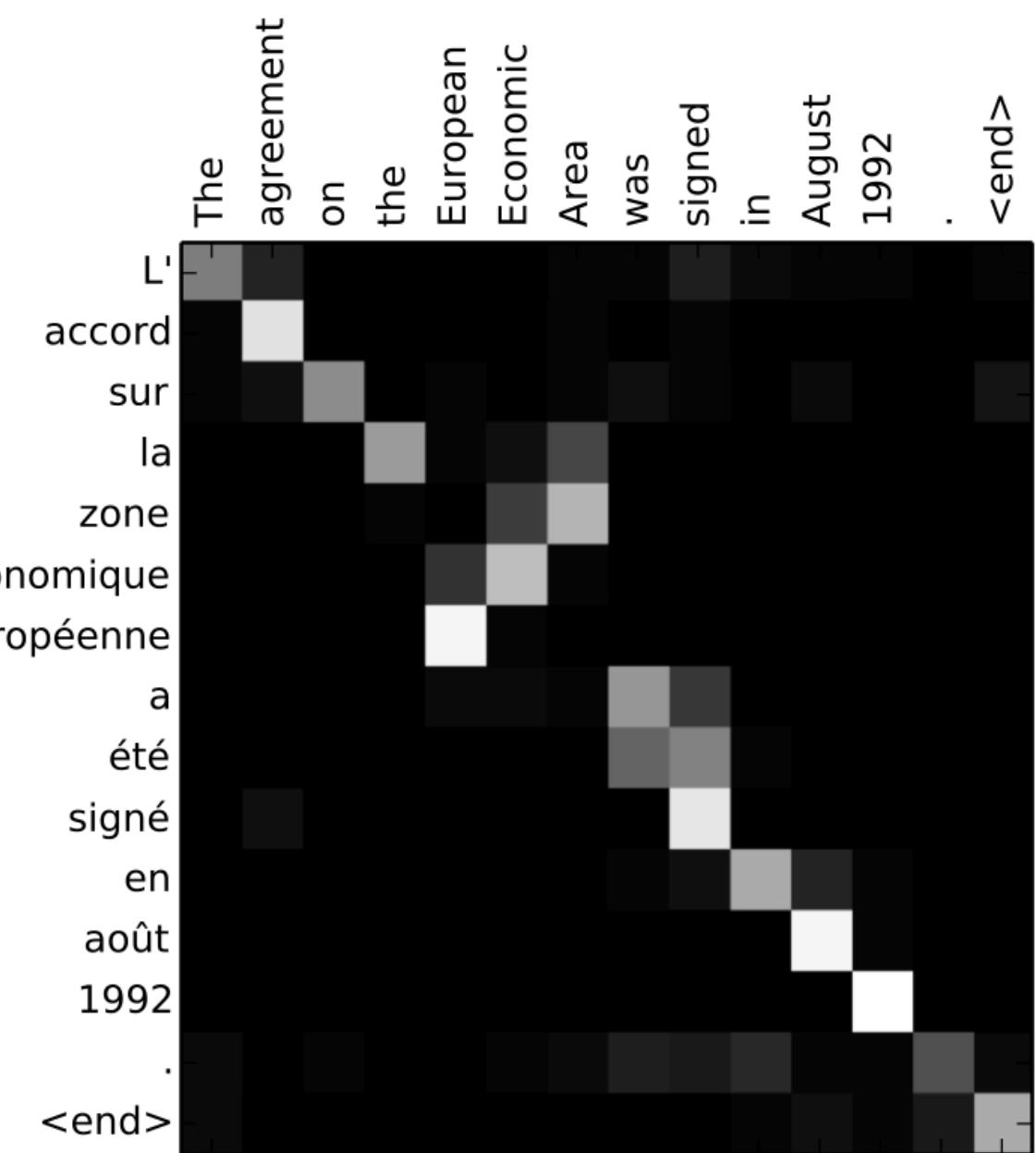


[Socher - CS224 - Stanford]

(Neural) Machine Translation: Attention (2015)

Published as a conference paper at ICLR 2015

Source sentence: English



NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau

Jacobs University Bremen, Germany

KyungHyun Cho Yoshua Bengio*

Université de Montréal

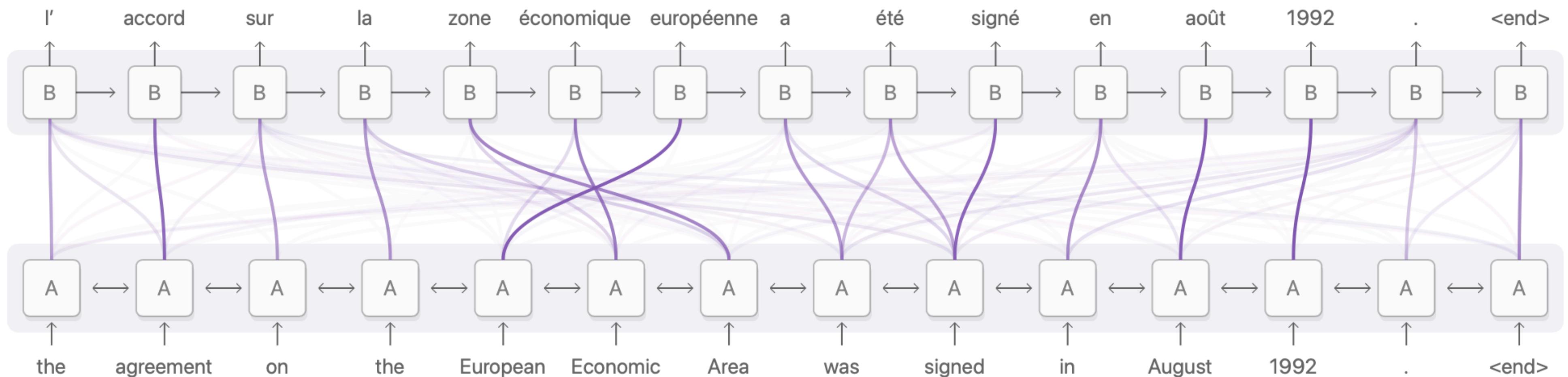


Diagram derived from Fig. 3 of [Bahdanau, et al. 2014](#)

[Chris Olah & Shan Crater 2016 <https://distill.pub/2016/augmented-rnns/>]

Attention is all you need - Transformers (2017)

<https://arxiv.org/abs/1706.03762>

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

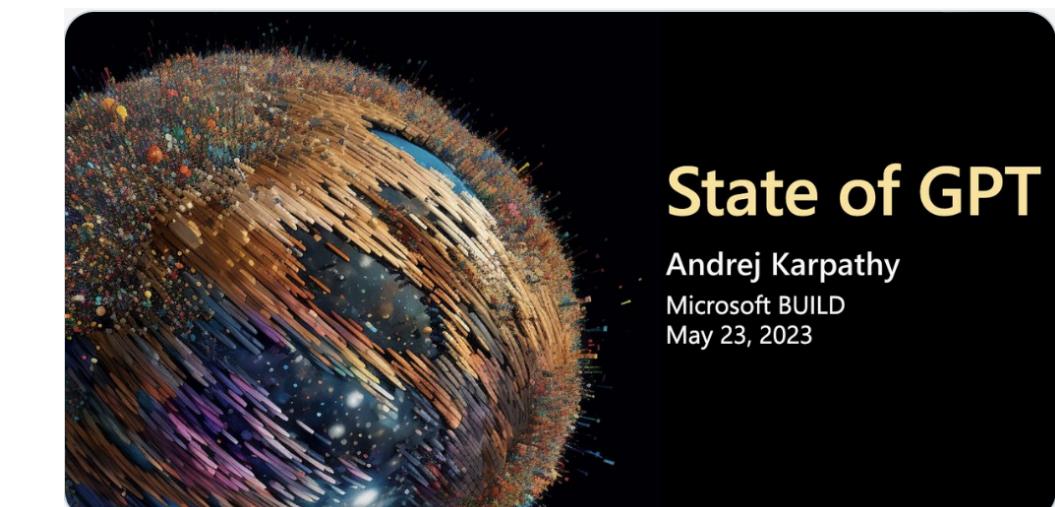
Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Great resources to understand the transformer architecture:



Let's build GPT: from scratch,
in code, spelled out.

Andrej Karpathy
2.4M views • 4 months ago



First application: machine translation

Multi-Head Attention

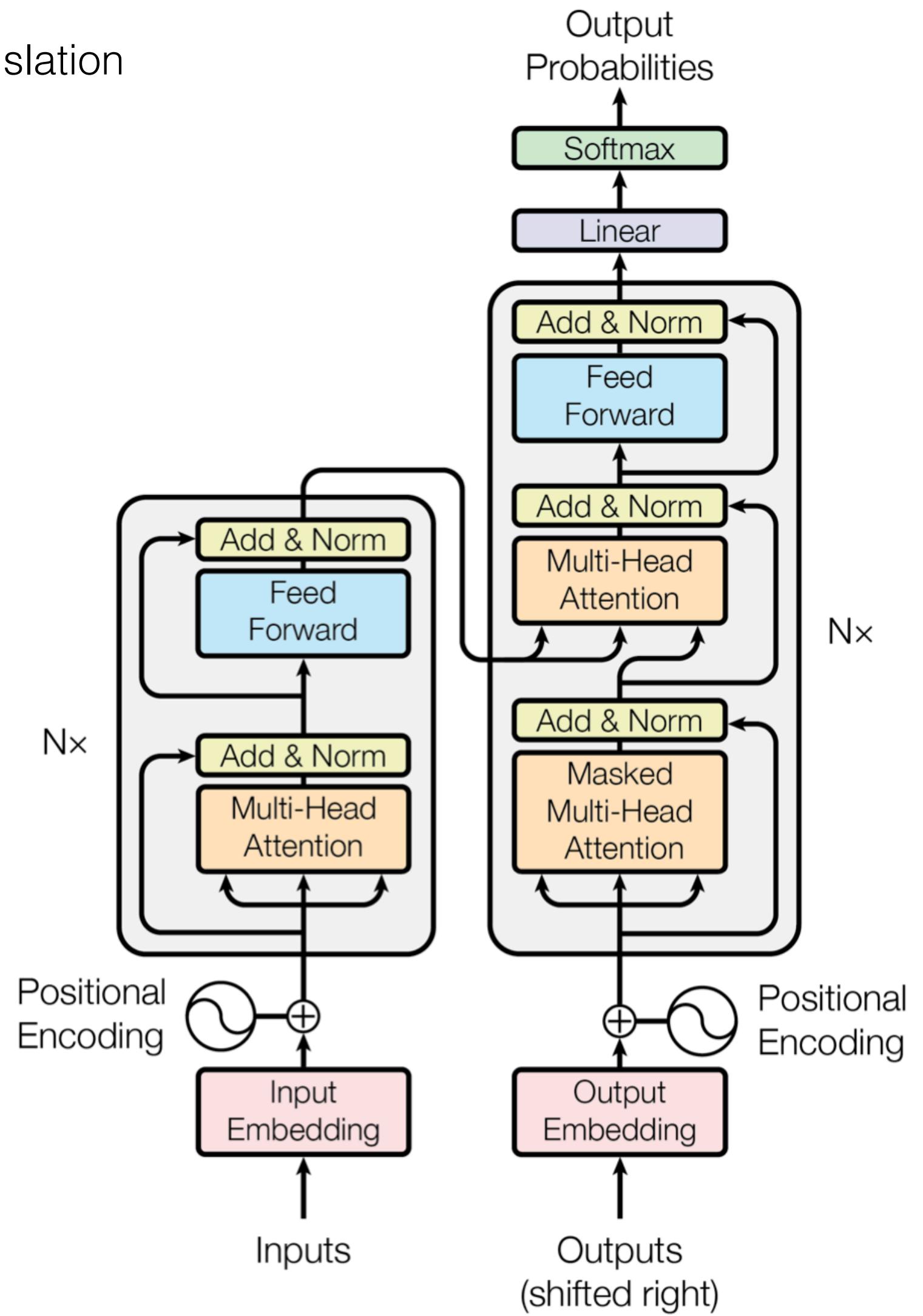
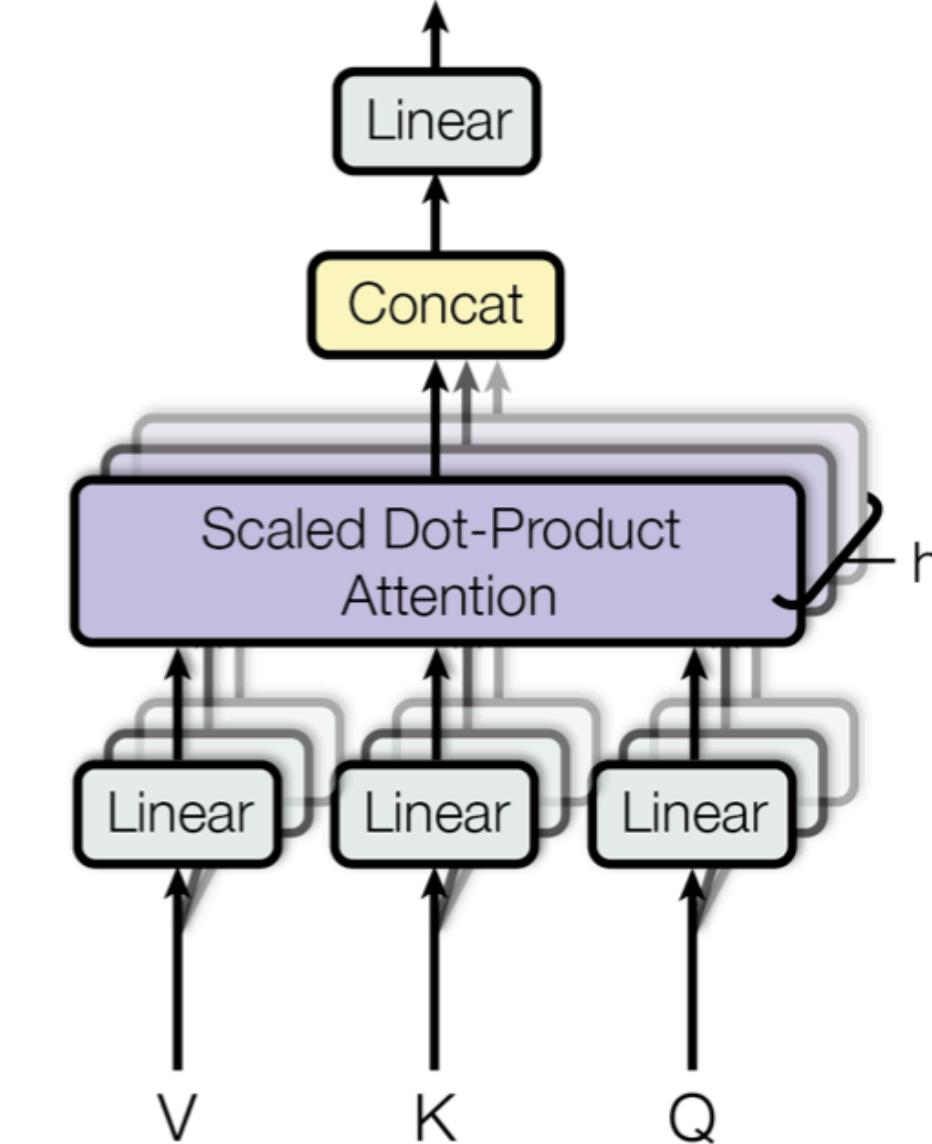
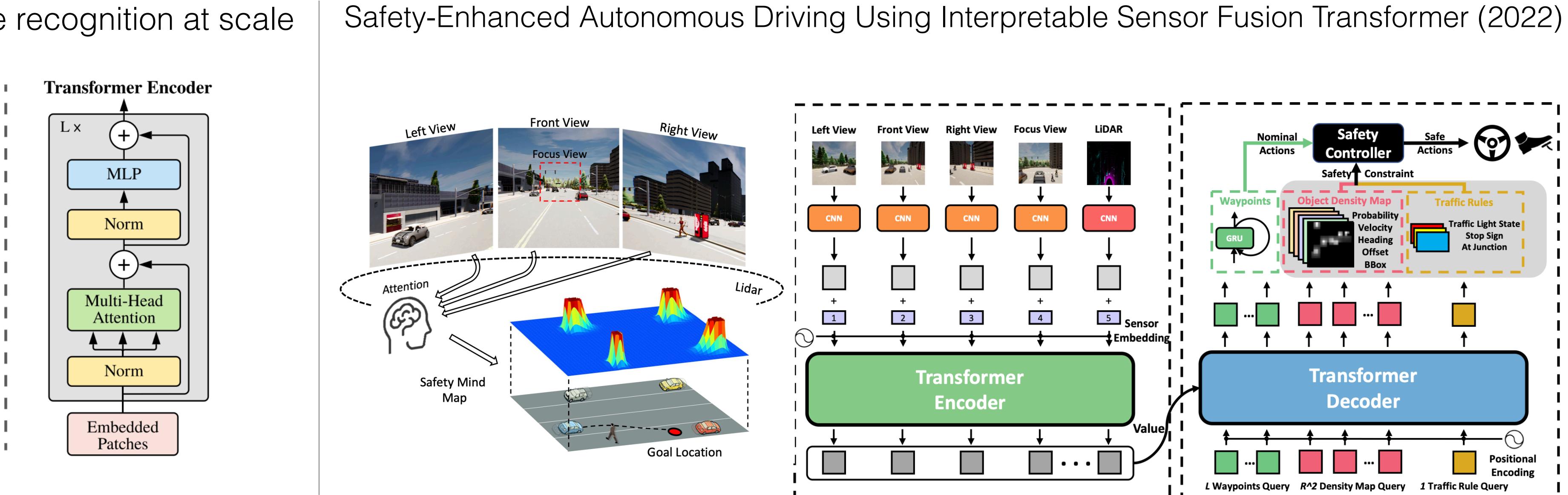
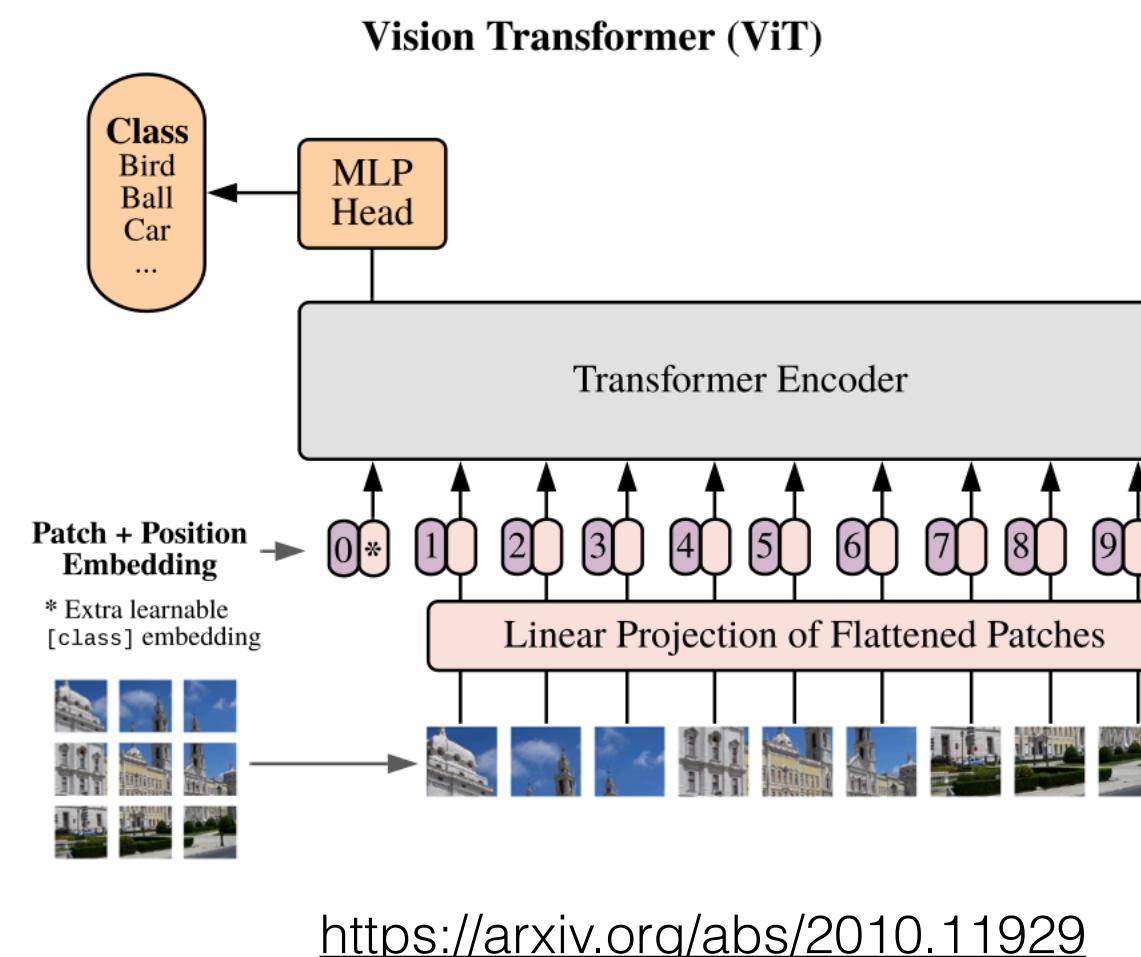


Figure 1: The Transformer - model architecture.

Transformer architecture: 2017—2023—

@karpathy: “The Transformer is a ... general-purpose differentiable computer” (2022)

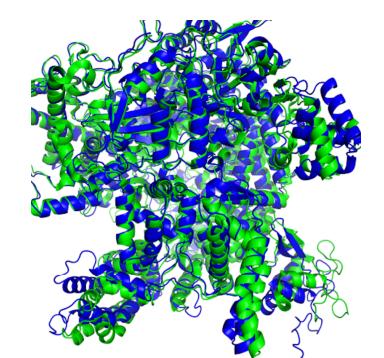
Vision: (2020) Transformer for image recognition at scale



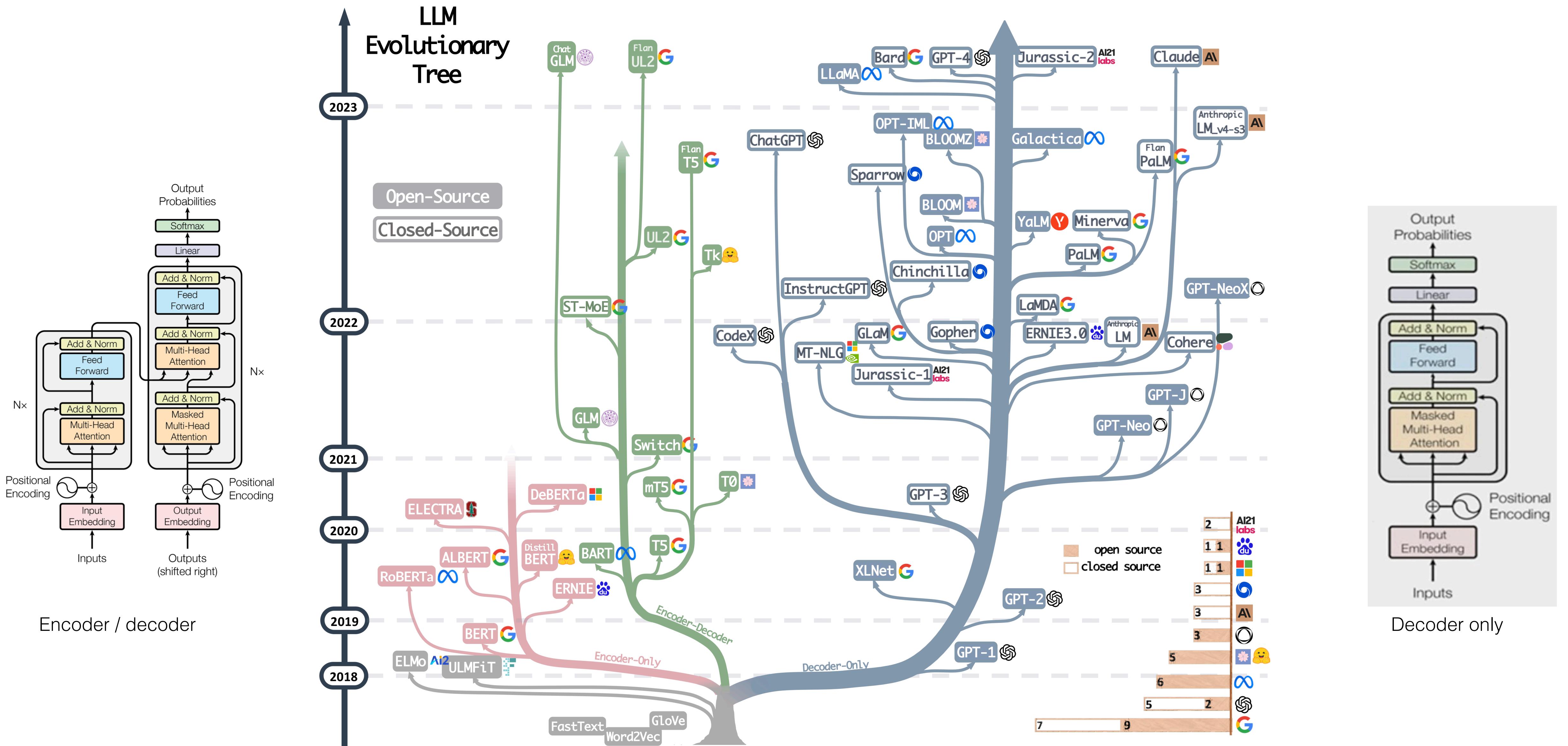
Speech: (2020) Convolution-augmented Transformer for Speech Recognition

Speech: (2022) Whisper

Protein folding: (2020) AlphaFold



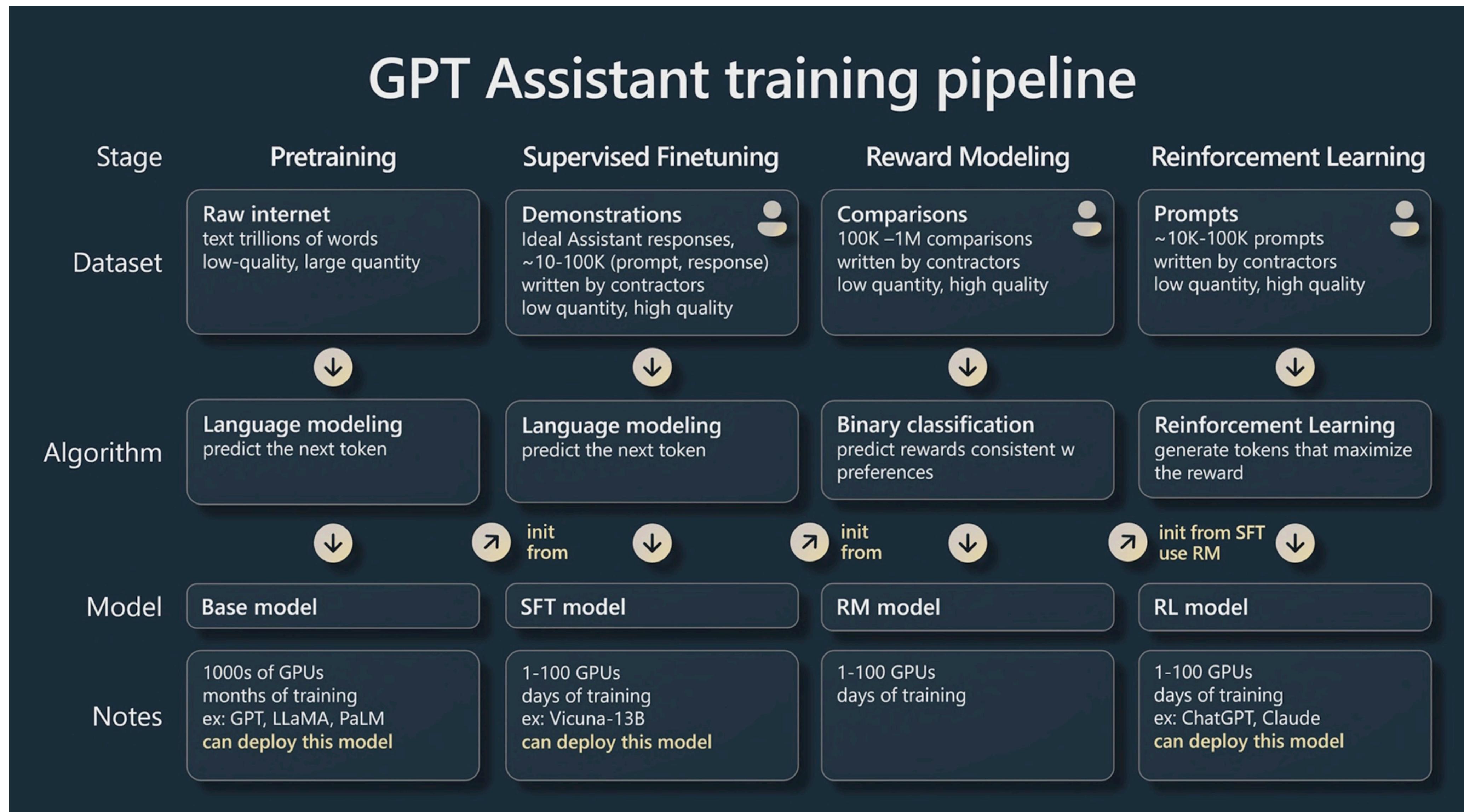
Large Language Models



[<https://notes.kateva.org/2023/04/large-language-models-evolutionary-tree.html>]

Large Language Models (LLMs)

GPT Assistant training pipeline



[Andrej Karpathy - State of GPT - 2023-05]

1. Large Language Model (pre-training)



GPT-3 Language Models are Few-Shot Learners ([paper](#)) - 2020

175B parameters, 96 layers, 96 heads, trained on 300B tokens, 2,408 tokens context length

1 month training on thousands of GPUs, token vocabulary size approx. 50,000



BLOOM (BigScience Large Open-science Open-access Multilingual) language model ([blog](#), [model](#)) - 2022

176B parameters, 46 natural languages, 13 programming languages



LLaMA (Large Language Model Meta AI) ([paper](#)) - 2023

7B to 65B parameters, 64 heads, 80 layers, trained on 1.4T tokens, 2,048 tokens context length

21 days training on 2,048 of A100 GPUs, token vocabulary size approx. 30,000



Red Pajama ([together.xyz](#) - UdeM, ETH, Stanford, ...)

7B parameters, trained on 1.2T tokens

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

LLM tokenization

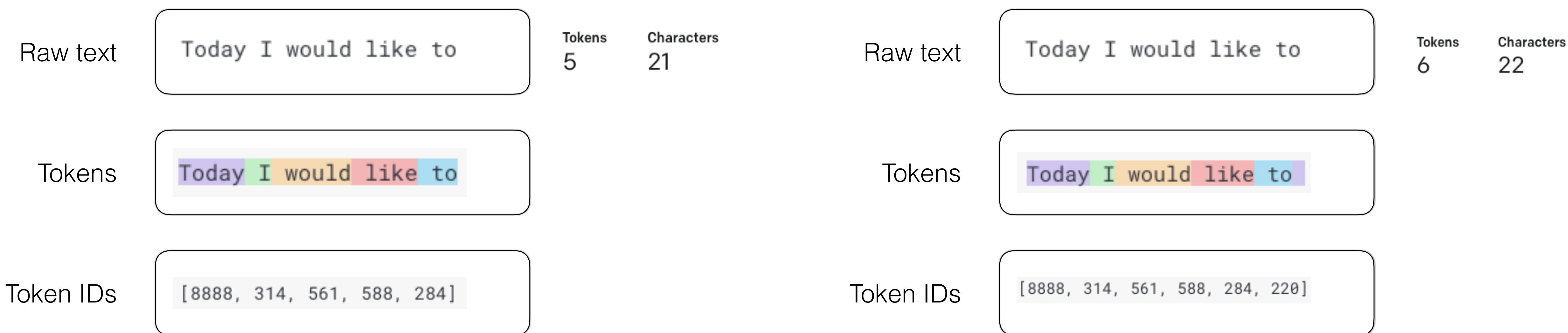
Tokenization: transforms some text into a long list of integers...

Token dictionary: A LLM uses a fixed token dictionary representing most common character sequences in training corpora.

Most common tokenizers: Byte-Pair-Encoding, WordPiece (see [HuggingFace's tokenizers](#))

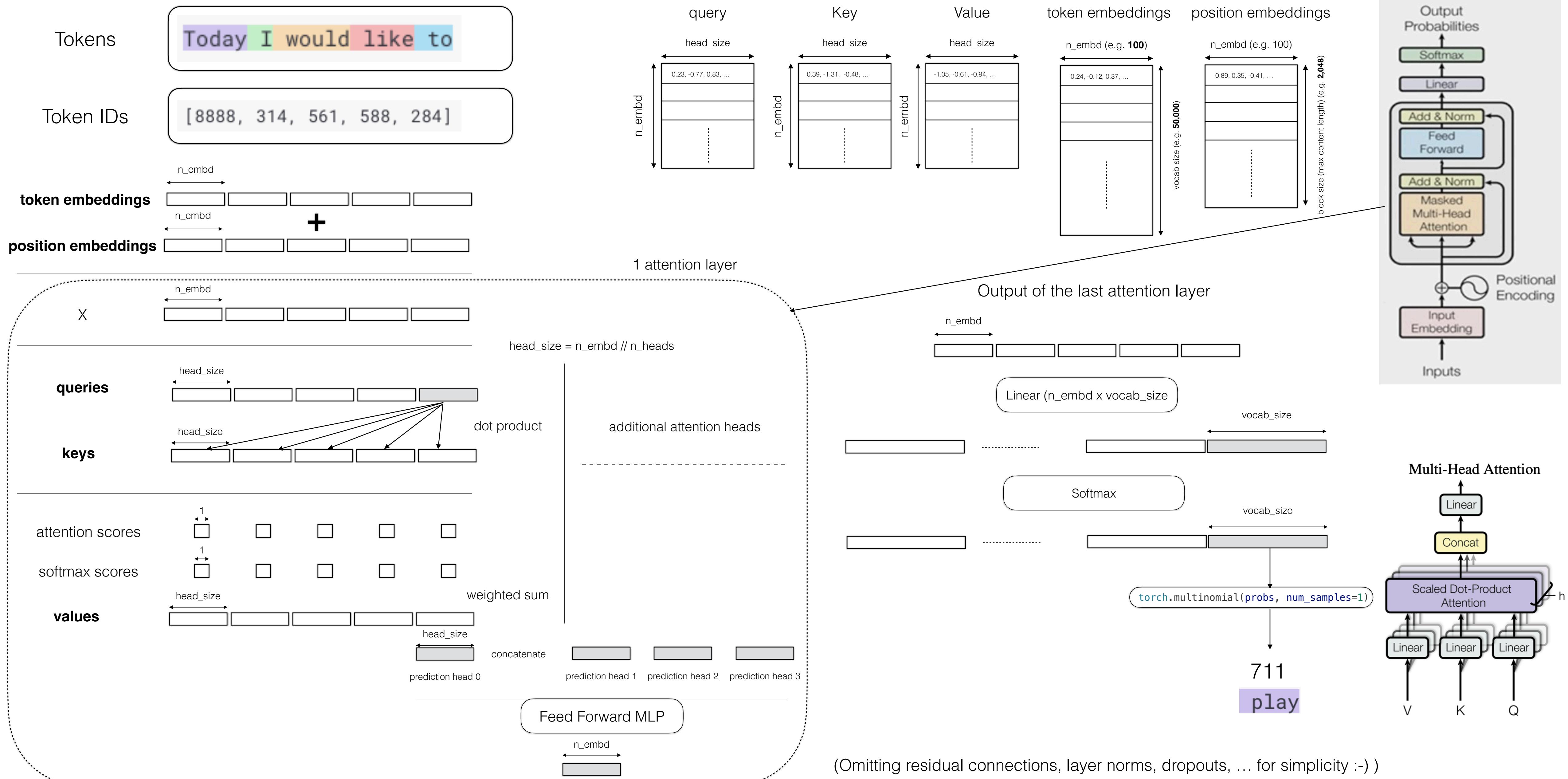
Typical token vocabulary size: 10 to 100K

100 tokens equals approx. 75 words



platform.openai.com/tokenizer

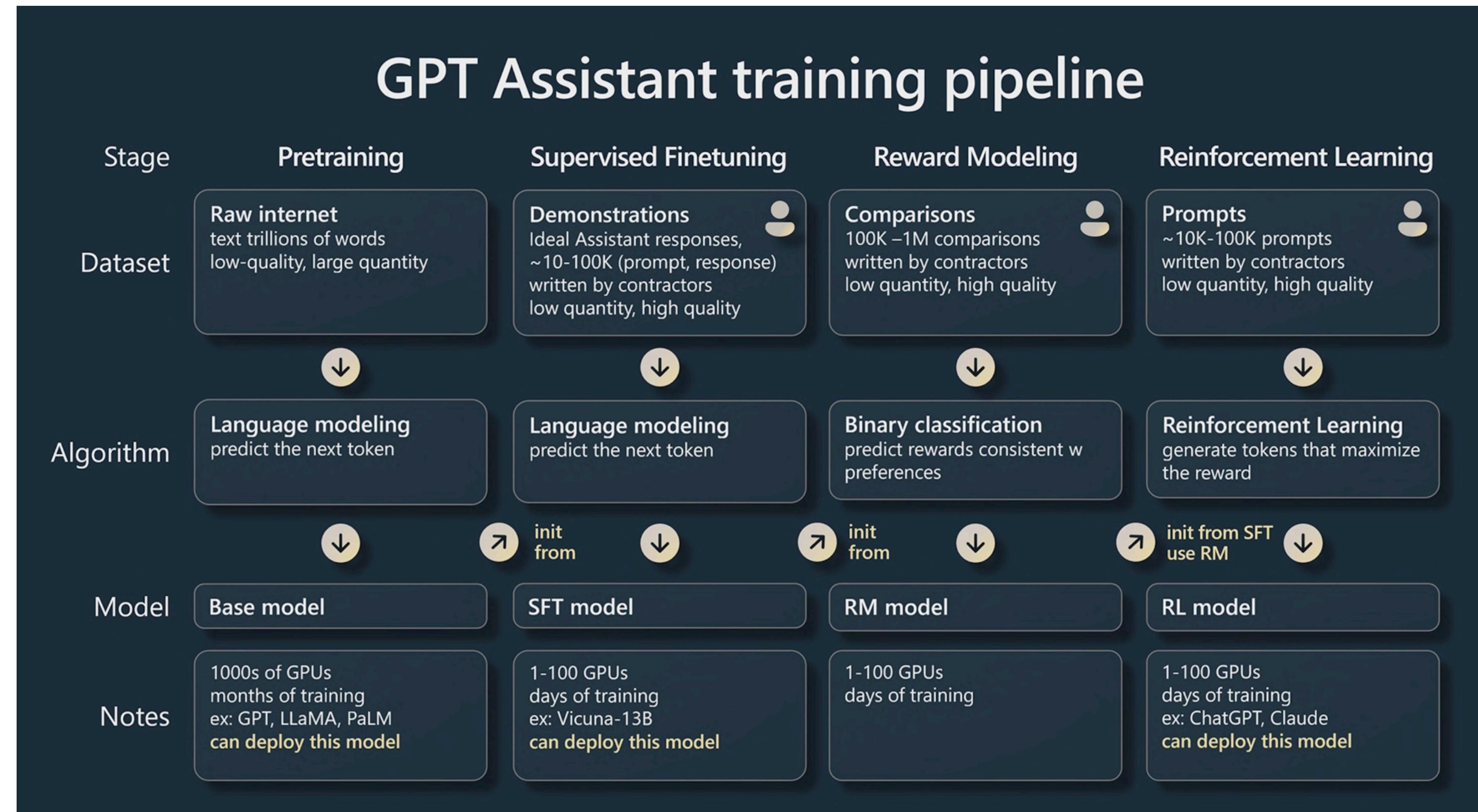
LLM transformer



Pre-trained LLM -> supervised fine-tuning -> reward modelling -> reinforcement learning

Pre-trained LLMs (e.g. GPT-3, BLOOM, LIAMA, ...) are not (yet) (chat) assistants

Self-supervised pre-trained LLMs “only” auto-complete text... (predicting next most likely words)



[Andrej Karpathy - State of GPT - 2023-05]

ChatGPT (OpenAI)

November 30, 2022

chat.openai.com

ChatGPT

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

Send a message...

Deep Learning

Other chat interfaces to LLMs

 Open Assistant

open-assistant.io/dashboard



gpt4all.io

 Perplexity

perplexity.ai



you.com

chat.lmsys.org

Prompt Engineering



Andrej Karpathy ✅ @karpathy · Jan 24
The hottest new programming language is English
In fact, any language: French, Japanese, ...

Assume you are writing to a subcontractor who cannot call you back for more info...

Use delimiters to clearly indicate distinct parts of the input; e.g. ```, "", <>, <tag>, </tag>, :



Your task is to perform the following actions:

- 1 - Summarize the following text delimited by <> with 1 sentence.
- 2 - Translate the summary into French.
- 3 - List each name in the French summary.
- 4 - Output a JSON object that contains the following keys: french_summary, num_names.

Use the following format:

Text: <text to summarize>

Summary: <summary>

Translation: <summary translation>

Names: <list of names in the French summary>

Output JSON: <json with summary and num_names>

Text:

<

In his closing statement to a Moscow court, just before he was sentenced to 25 years in prison for his outspoken opposition to the invasion of Ukraine, Vladimir Kara-Murza reminded those in attendance that Russia had been down this repressive direction before.

"I've been surprised by the extent to which my trial, in its secrecy and its contempt for legal norms, has surpassed even the trials of Soviet dissidents in the 1960s and '70s," Mr. Kara-Murza told the court on April 17. "We've gone beyond the 1970s – all the way back to the 1930s."

The 1930s in Russia were the time of Joseph Stalin's purges, show trials and extrajudicial executions. For the 23 years that Russian President Vladimir Putin has been in power, he has faced accusations that he seeks to rehabilitate Stalinism and restore the Soviet empire. Until a year ago, the Kremlin could comfortably swat such talk away as hyperbole.

>

1. Write clear and specific instructions

ChatGPT
Prompt Engineering
for Developers



Isa Fulford



Andrew Ng

learn.deeplearning.ai/chatgpt-prompt-eng



Summary: Vladimir Kara-Murza compares his trial for opposing the invasion of Ukraine to the repressive trials of Soviet dissidents in the 1960s and '70s, stating that Russia has regressed back to the era of Stalin's purges in the 1930s.

Translation: Vladimir Kara-Murza compare son procès pour s'opposer à l'invasion de l'Ukraine aux procès répressifs des dissidents soviétiques des années 1960 et 1970, déclarant que la Russie est retournée à l'époque des purges de Staline dans les années 1930.

Names: Vladimir Kara-Murza, Joseph Stalin, Vladimir Putin

Output JSON:

```
{  
  "french_summary": "Vladimir Kara-Murza compare son procès pour s'opposer à l'invasion de l'Ukraine aux procès répressifs des dissidents soviétiques des années 1960 et 1970, déclarant que la Russie est retournée à l'époque des purges de Staline dans les années 1930.",  
  "num_names": 3  
}
```

Prompt engineering: extracting information / infer topics

```
lamp_review = """  
Needed a nice lamp for my bedroom, and this one had \  
additional storage and not too high of a price point. \  
Got it fast. The string to our lamp broke during the \  
transit and the company happily sent over a new one. \  
Came within a few days as well. It was easy to put \  
together. I had a missing part, so I contacted their \  
support and they very quickly got me the missing piece! \  
Lumina seems to me to be a great company that cares \  
about their customers and products!!  
"""
```

```
prompt = f"""\nIdentify the following items from the review text:
```

- Sentiment (positive or negative)
- Is the reviewer expressing anger? (true or false)
- Item purchased by reviewer
- Company that made the item

```
The review is delimited with triple backticks. \  
Format your response as a JSON object with \  
"Sentiment", "Anger", "Item" and "Brand" as the keys. \  
If the information isn't present, use "unknown" \  
as the value.
```

Make your response as short as possible.

Format the Anger value as a boolean.

```
Review text: '''{lamp_review}'''  
"""
```

```
{  
    "Sentiment": "positive",  
    "Anger": false,  
    "Item": "lamp with additional storage",  
    "Brand": "Lumina"  
}
```

```
story = """  
In a recent survey conducted by the government, \  
public sector employees were asked to rate their level \  
of satisfaction with the department they work at. \  
The results revealed that NASA was the most popular \  
department with a satisfaction rating of 95%.
```

```
One NASA employee, John Smith, commented on the findings, \  
stating, "I'm not surprised that NASA came out on top. \  
It's a great place to work with amazing people and \  
incredible opportunities. I'm proud to be a part of \  
such an innovative organization."
```

```
The results were also welcomed by NASA's management team, \  
with Director Tom Johnson stating, "We are thrilled to \  
hear that our employees are satisfied with their work at NASA. \  
We have a talented and dedicated team who work tirelessly \  
to achieve our goals, and it's fantastic to see that their \  
hard work is paying off."
```

```
The survey also revealed that the \  
Social Security Administration had the lowest satisfaction \  
rating, with only 45% of employees indicating they were \  
satisfied with their job. The government has pledged to \  
address the concerns raised by employees in the survey and \  
work towards improving job satisfaction across all departments.  
"""
```

```
prompt = f"""\nDetermine five topics that are being discussed in the \  
following text, which is delimited by triple backticks.
```

Make each item one or two words long.

Format your response as a list of items separated by commas.

```
Text sample: '''{story}'''  
"""
```

government survey, job satisfaction, NASA, Social Security Administration, employee concerns

Prompt engineering: transforming



Translate the following from slang to a business letter:
'Dude, do you have a job for a software developer?'



Dear Sir/Madam,

I hope this letter finds you in good health and high spirits. I am writing to inquire about any potential job opportunities within your esteemed organization. Specifically, I am interested in a position as a software developer.

Allow me to introduce myself; my name is [Your Name]. I have a strong background in software development and possess a wide range of skills and expertise in this field. I am confident that my qualifications align well with the requirements of a software developer role.

I kindly request you to consider my application and notify me if there are any open positions or if you anticipate any future opportunities within your organization. I would be more than happy to provide you with my detailed resume and any additional information you may require.

Thank you for considering my inquiry. I eagerly look forward to the possibility of contributing to the success of your organization as a software developer. Should you require any further information or wish to schedule an interview, please do not hesitate to contact me.

Yours faithfully,

[Your Name]



Proofread and correct this review. Make it more compelling.
Ensure it follows APA style guide and targets an advanced reader.
Output in markdown format.

Text:

''''''

Got this for my daughter for her birthday cuz she keeps taking \ mine from my room. Yes, adults also like pandas too. She takes \ it everywhere with her, and it's super soft and cute. One of the \ ears is a bit lower than the other, and I don't think that was \ designed to be asymmetrical. It's a bit small for what I paid for it \ though. I think there might be other options that are bigger for \ the same price. It arrived a day earlier than expected, so I got \ to play with it myself before I gave it to my daughter.

''''''



Review: An Adorable Gift for All Ages

Introduction

I recently purchased a charming panda plush toy as a birthday gift for my daughter, as she had developed a penchant for borrowing mine from my room. Contrary to popular belief, pandas hold an undeniable appeal for adults as well. This review aims to provide a comprehensive assessment of the product, catering to advanced readers who value attention to detail and elegance.

The Perfect Companion

My daughter's delight knew no bounds when she received the panda plush. Its irresistibly soft texture and adorable appearance make it an instant favorite. Wherever she goes, this delightful companion faithfully accompanies her, bringing a sense of joy and comfort to her every adventure.

Exceptional Quality, Minor Flaw

While the overall quality of the plush toy is exceptional, I noticed a slight asymmetry in its design. One of the ears appears to be positioned slightly lower than the other, which, to my discerning eye, does not seem intentional. This minor imperfection, although noticeable upon closer inspection, does not diminish the overall appeal of the product.

Size Considerations

It is worth mentioning that the size of the panda plush was slightly smaller than my initial expectations, considering the price I paid for it. However, I believe it is important to note that there might be alternative options available at a similar price point that offer a larger size. Potential buyers seeking a more substantial plush may want to explore such alternatives before making their final decision.

Expedited Delivery, Personal Delight

To my pleasant surprise, the package arrived a day earlier than the anticipated delivery date. This unexpected advantage allowed me to indulge in the joy of playing with the panda plush myself before presenting it to my daughter. The extra time spent with this delightful toy reinforced my confidence in the quality and appeal of the product.

Conclusion

In summary, the panda plush toy I acquired proved to be an enchanting gift suitable for all ages. Its exceptional softness, endearing appearance, and unwavering companionship have made it an instant favorite of my daughter. Although I observed a minor design flaw and found its size to be slightly smaller than expected, the overall quality and charm of the product remain undeniable. For those seeking an elegant and delightful plush companion, this panda plush stands as a compelling option to consider.

Prompt engineering: designing a chatbot

The screenshot shows a conversational AI interface with two columns for a user and a chatbot.

User Side (Left):

- Role definition: A system role message: '{'role':'system','content':'You are friendly chatbot.'}, { 'role': 'user', 'content': 'Hi, my name is Isa'}]
- Initial greeting from the user: "Hello Isa! It's nice to meet you. How can I assist you today?"
- OrderBot's initial response (scripted): You are OrderBot, an automated service to collect orders for a pizza restaurant. You first greet the customer, then collects the order, and then asks if it's a pickup or delivery. You wait to collect the entire order, then summarize it and check for a final time if the customer wants to add anything else. If it's a delivery, you ask for an address. Finally you collect the payment. Make sure to clarify all options, extras and sizes to uniquely identify the item from the menu. You respond in a short, very conversational friendly style. The menu includes pepperoni pizza 12.95, 10.00, 7.00 cheese pizza 10.95, 9.25, 6.50 eggplant pizza 11.95, 9.75, 6.75 fries 4.50, 3.50 greek salad 7.25 Toppings: extra cheese 2.00, mushrooms 1.50 sausage 3.00 canadian bacon 3.50 AI sauce 1.50 peppers 1.00 Drinks: coke 3.00, 2.00, 1.00 sprite 3.00, 2.00, 1.00 bottled water 5.00 """}]
- Final welcome message: "Hi there! Welcome to OrderBot, your friendly automated ordering service for our pizza restaurant. How can I assist you today?"

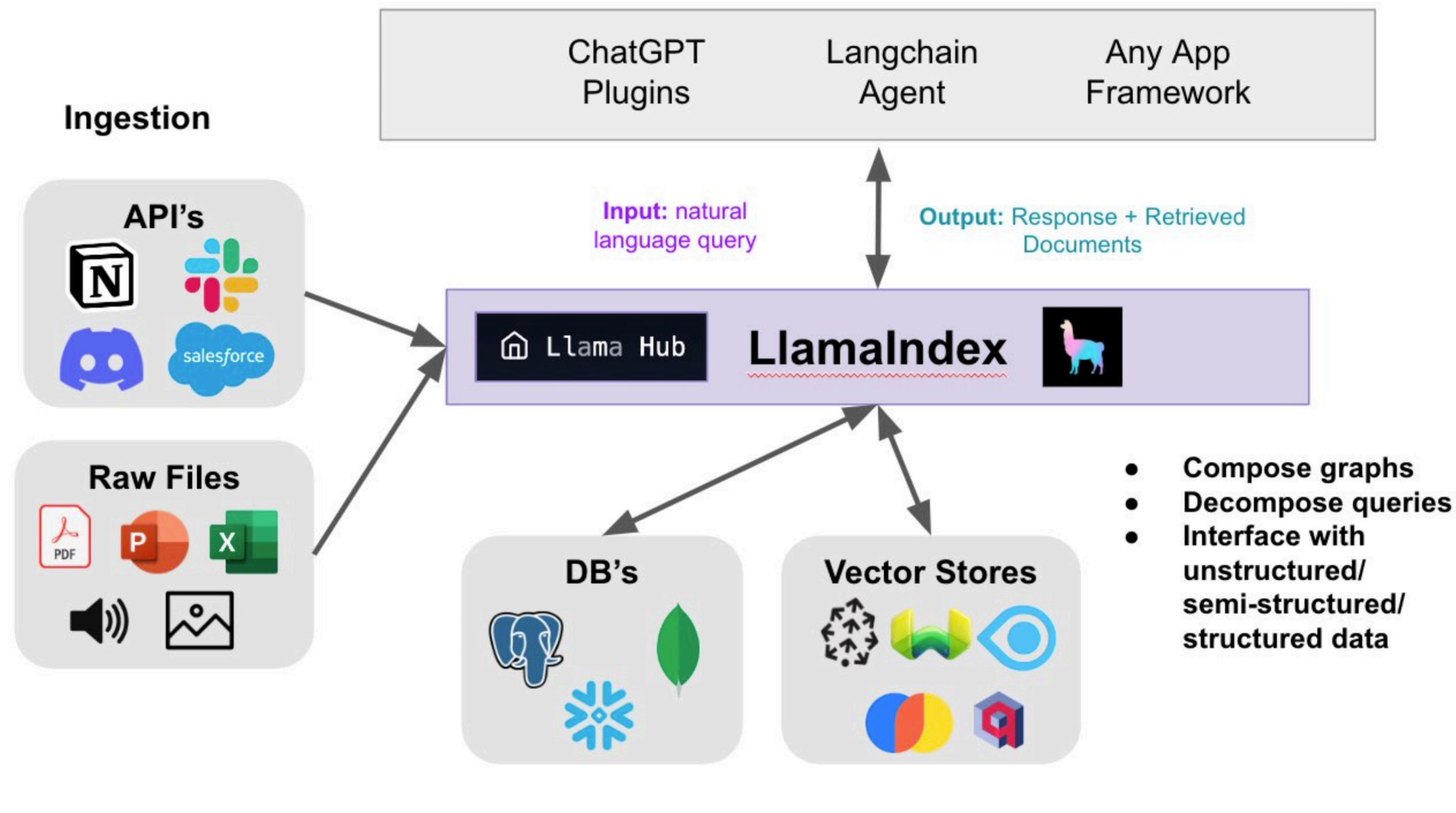
Chatbot Side (Right):

- User message: "I would like to order a pizza." Response: Great! I'm here to help you with that. Could you please let me know which type of pizza you'd like to order? We have pepperoni pizza, cheese pizza, and eggplant pizza. Just tell me the name of the pizza you want, and I'll provide you with the available sizes and prices.
- User message: "pepperoni is fine." Response: Excellent choice! Now, let's talk about the size. We have three options for our pepperoni pizza: small, medium, and large. The prices are as follows: small is \$7.00, medium is \$10.00, and large is \$12.95. Please let me know which size you prefer for your pepperoni pizza.
- User message: "I would like a medium size." Response: Perfect! You've chosen a medium-sized pepperoni pizza. Is there anything else you'd like to add to your order, such as toppings, side dishes, or drinks? Let me know, and I'll make sure to include them for you.
- User message: "what kind of drinks to you have?" Response: We offer a selection of refreshing drinks. Here are your options:
 1. Coke: We have three sizes available - small for \$1.00, medium for \$2.00, and large for \$3.00.
 2. Sprite: Similar to Coke, we have small for \$1.00, medium for \$2.00, and large for \$3.00.
 3. Bottled Water: We have bottled water available for \$5.00.Please let me know if you'd like to add any drinks to your order, and if so, the type and size you prefer.
- User message: "I will take a medium coke please." Response: Great! You've added a medium Coke to your order. Just to confirm, you're ordering a medium-sized pepperoni pizza and a medium Coke. Is that correct?
- User message: "yes" Response: Got it! You've ordered a medium-sized pepperoni pizza and a medium Coke. Now, I need to ask if you'll be picking up your order or if you'd like it delivered to your location. Could you please let me know your preference?
- User message: "I will pick it up." Response: Perfect! You'll be picking up your order. Thank you for letting us know. Now, to complete your order, I just need to confirm your payment method. We accept cash and credit/debit cards. Please let me know your preferred payment method, and we'll get everything ready for you to pick up.
- User message: "credit" Response: Great! We accept credit cards as a payment method. When you come to pick up your order, please make sure to have your credit card ready for payment.
- User message: "No. Thank you for your help." Response: Just to summarize, your order includes a medium-sized pepperoni pizza, a medium Coke, and you'll be picking it up. Is there anything else I can assist you with or add to your order?
- User message: "learn.deeplearning.ai/chatgpt-prompt-eng" Response: You're very welcome! I'm glad I could assist you with your order. Your medium-sized pepperoni pizza and medium Coke will be ready for pickup. If you have any further questions or need any assistance in the future, feel free to reach out. Enjoy your meal!

LLM: in-context learning / adding information

LLMs can only extract information from their context window size: 2,048 tokens (approx. 1,000 words) for GPT-3

LLMs have information from the web up to a given date: GPT 3.5 (ChatGPT) cut-off date is September 2021



[Jerry Liu - twitter.com/jerryliu0/status/1643998635984818178](https://twitter.com/jerryliu0/status/1643998635984818178)

LLM: give them a second chance

They cannot backtrack... But they can review in a subsequent response

LLMs: making them smaller / faster

Several sizes trained: reducing the number of heads and layers

Bit quantization: trained with 16 bits floating values (bfloat16 “brain float point 16)... reduced to 4 bits.

LLMs running on iPhone



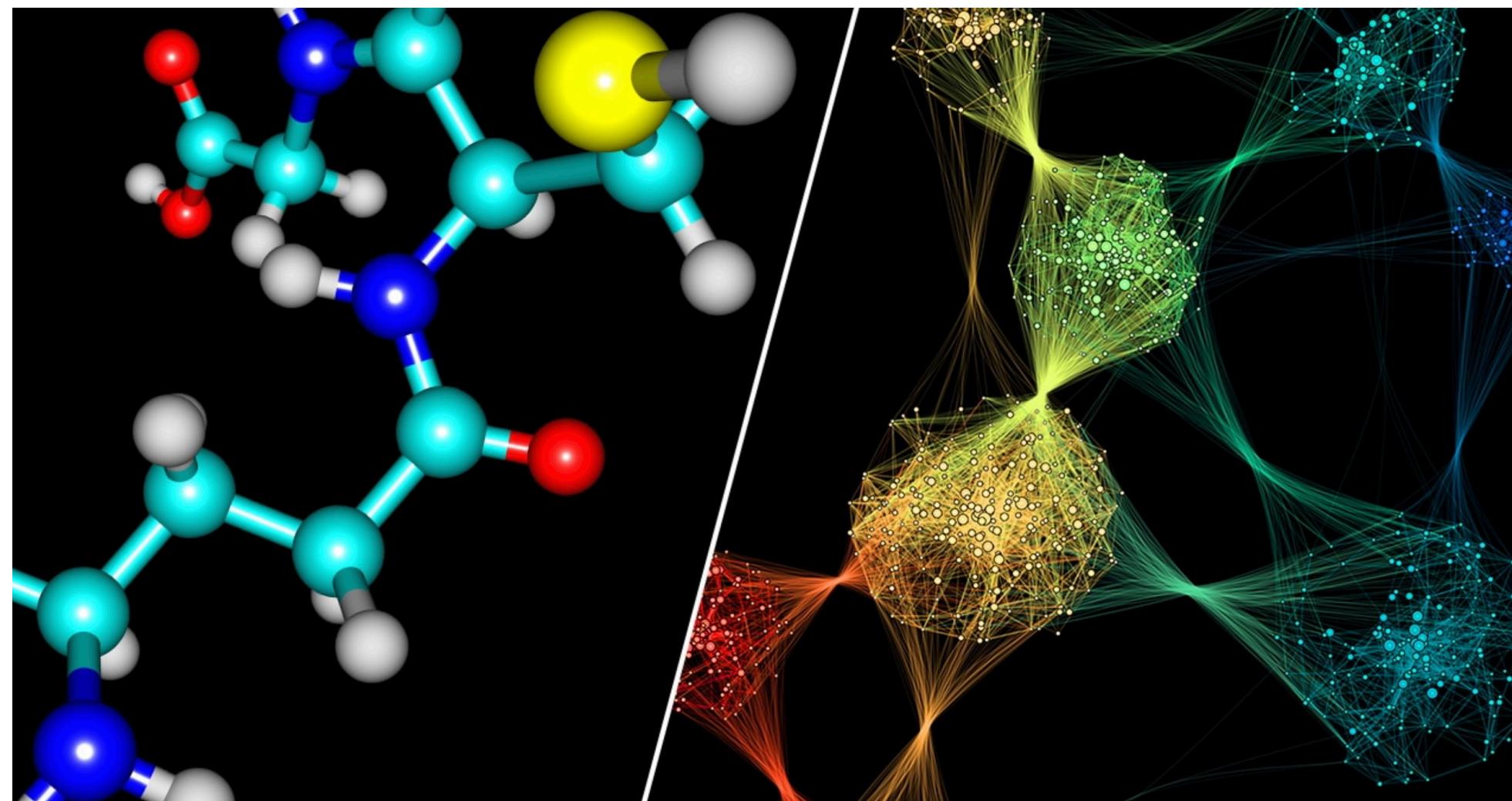
MACHINE LEARNING
COMPIRATION

mlc.ai/mlc-llm/

(Semi) Deep Learning (like) on Graphs



PyG (PyTorch Geometric)



blogs.nvidia.com/blog/2022/10/24/what-are-graph-neural-networks/

Do Transformers Really Perform Bad for Graph Representation?

Chengxuan Ying^{1*}, Tianle Cai², Shengjie Luo^{3*},
Shuxin Zheng^{4†}, Guolin Ke⁴, Di He^{4†}, Yanming Shen¹, Tie-Yan Liu⁴
¹Dalian University of Technology ²Princeton University
³Peking University ⁴Microsoft Research Asia
yingchengsyuan@gmail.com, tianle.cai@princeton.edu, luosj@stu.pku.edu.cn
{shuz, guoke, dihe, tyliu}@microsoft.com, shen@dlut.edu.cn

CS224W: Machine Learning with Graphs

Stanford / Winter 2023

Resources

Courses: cs231n.stanford.edu, cs224n.stanford.edu, fast.ai, Andrej Karpathy (Youtube)

News: deeplearning.ai/thebatch, [@karpathy](https://twitter.com/karpathy) (Twitter)

Blogs: karpathy.github.io

Podcast: [Lex Fridman](https://lexfridman.com/podcast/), [The Robot Brains](https://www.pieterabdeel.com/podcasts/) (Pieter Abdeel)

Tools: pytorch.org, [Huggingface.co](https://huggingface.co) 

Free GPU: [Google Colab](https://colab.research.google.com)

Papers: arxiv.org 