

Hacking a Large Language Model: an attempt of Explainable AI (XAI)

L. Gianfagna

E. Zimuel



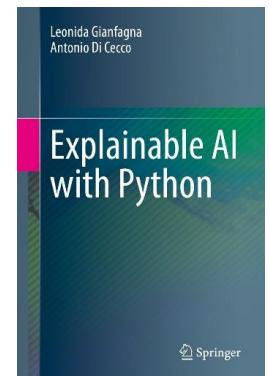
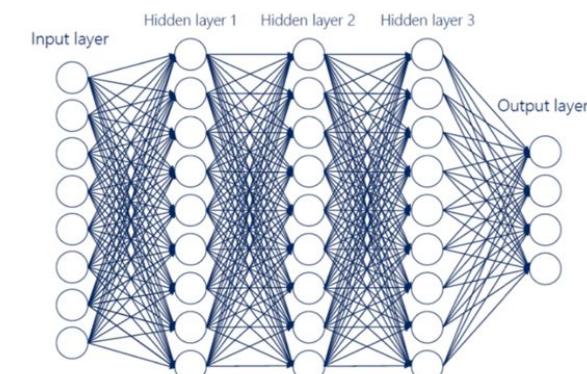
eXplainable AI (XAI)

GO champion Fan Hui commenting the famous 37th move of AlphaGo, the software developed by Google to play GO, that defeated in March 2016 the Korean champion Lee Sedol with an historical result: "It's not a human move, I've never seen a man playing such a move".

GO is known as a “computationally complex” game, more complex than Chess and before this result the common understanding was that it was not a game suitable for a machine to play successfully.

The GO champion could not make sense of the move even after having looked at all the match, he recognized it as brilliant, but he had no way to provide an explanation

A (non-mathematical) definition by Miller (2017) is:
Explainability is the degree to which a human can understand the cause of a decision



XAI

- After the training, **the algorithm learned to distinguish the classes with remarkable accuracy**: only a misclassification over 100 images! But if we use an Explainable Ai method asking the model **“Why have you predicted wolf?”** The answer will be with a little of surprise **“because there is snow!”**
- This is an **experiment** conducted to fool the **Deep Neural Network (DNN)**: the engineers maintained in the second and fourth images only the elements that the system used to recognize a guitar and a penguin and changed all the rest so that the system still “see” them like a guitar and a penguin.
- The work from Goodfellow et al. (2014) opened the door to further evolutions starting from universal perturbations (Moosavi-Dezfooli et al. 2017) to the recent one-pixel attacks that showed how to fool a neural network by just changing one pixel in the input image.

[Notebook Here](#)

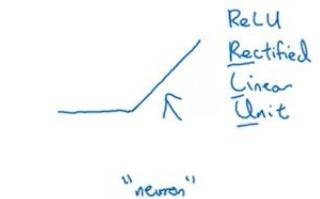
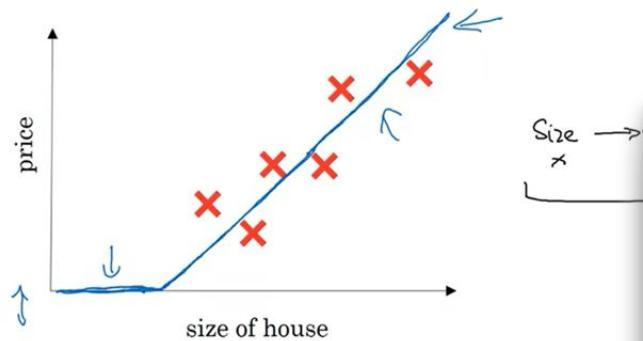
[One Pixel attack orginal paper](#)



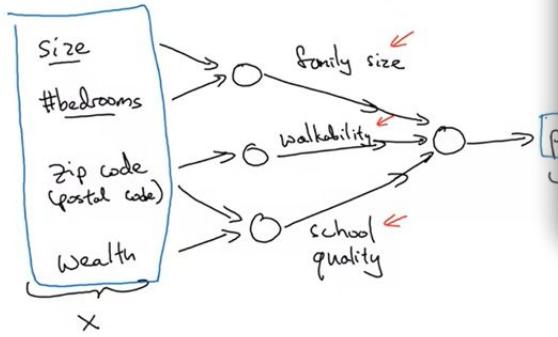
Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

To give you a feeling of what ML really is...

Housing Price Prediction

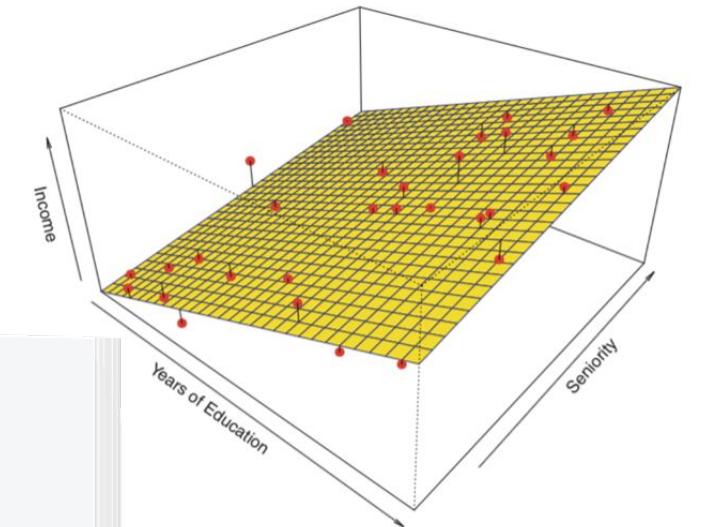


Housing Price Prediction



Statistics
Legendre
1805

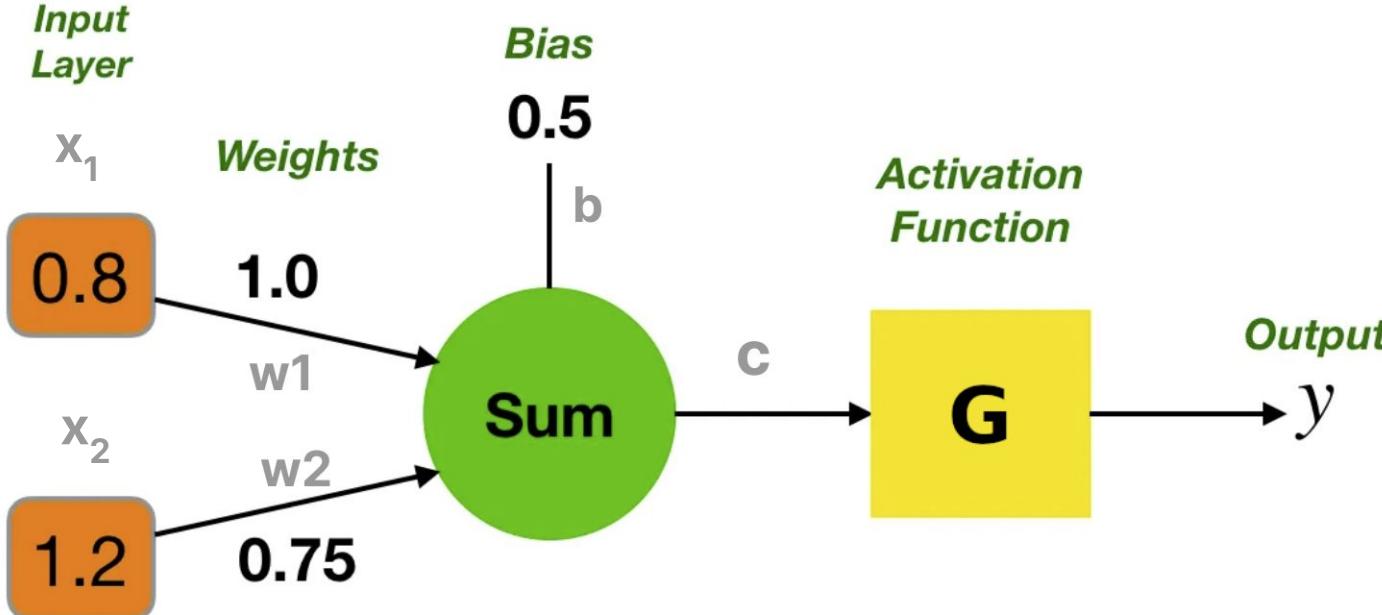
y = βX + ε
MACHINE LEARNING
ML 2024



Representation

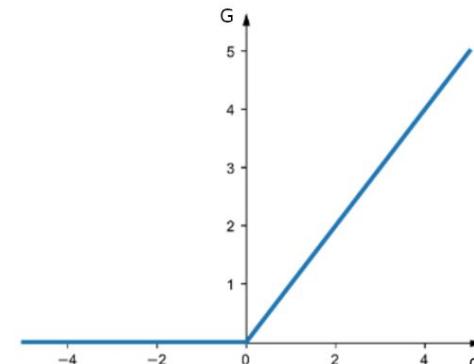
$$\begin{aligned} z^{[1]}_1 &= w_1^{[1]T} x + b_1^{[1]}, & a_1^{[1]} &= \sigma(z_1^{[1]}) \\ z^{[1]}_2 &= w_2^{[1]T} x + b_2^{[1]}, & a_2^{[1]} &= \sigma(z_2^{[1]}) \\ z^{[1]}_3 &= w_3^{[1]T} x + b_3^{[1]}, & a_3^{[1]} &= \sigma(z_3^{[1]}) \\ z^{[1]}_4 &= w_4^{[1]T} x + b_4^{[1]}, & a_4^{[1]} &= \sigma(z_4^{[1]}) \end{aligned}$$
$$\hat{y} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \begin{bmatrix} w_1^{[1]T} x + b_1^{[1]} \\ w_2^{[1]T} x + b_2^{[1]} \\ w_3^{[1]T} x + b_3^{[1]} \\ w_4^{[1]T} x + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix}$$

Single layer neuron



Example: ReLU activation

$$G(c) = \begin{cases} c & \text{if } c \geq 0 \\ 0 & \text{if } c < 0 \end{cases}$$

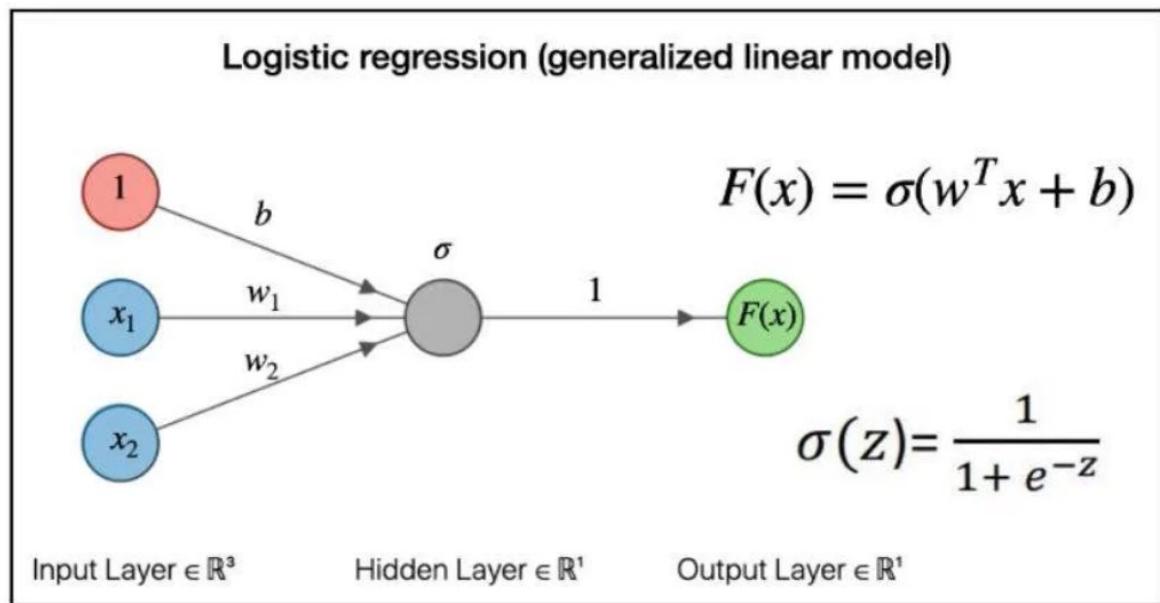


$$\begin{aligned}c &= x_1 * w_1 + x_2 * w_2 + b \\&= 0.8 * 1.0 + 1.2 * 0.75 + 0.5 \\&= 2.2\end{aligned}$$

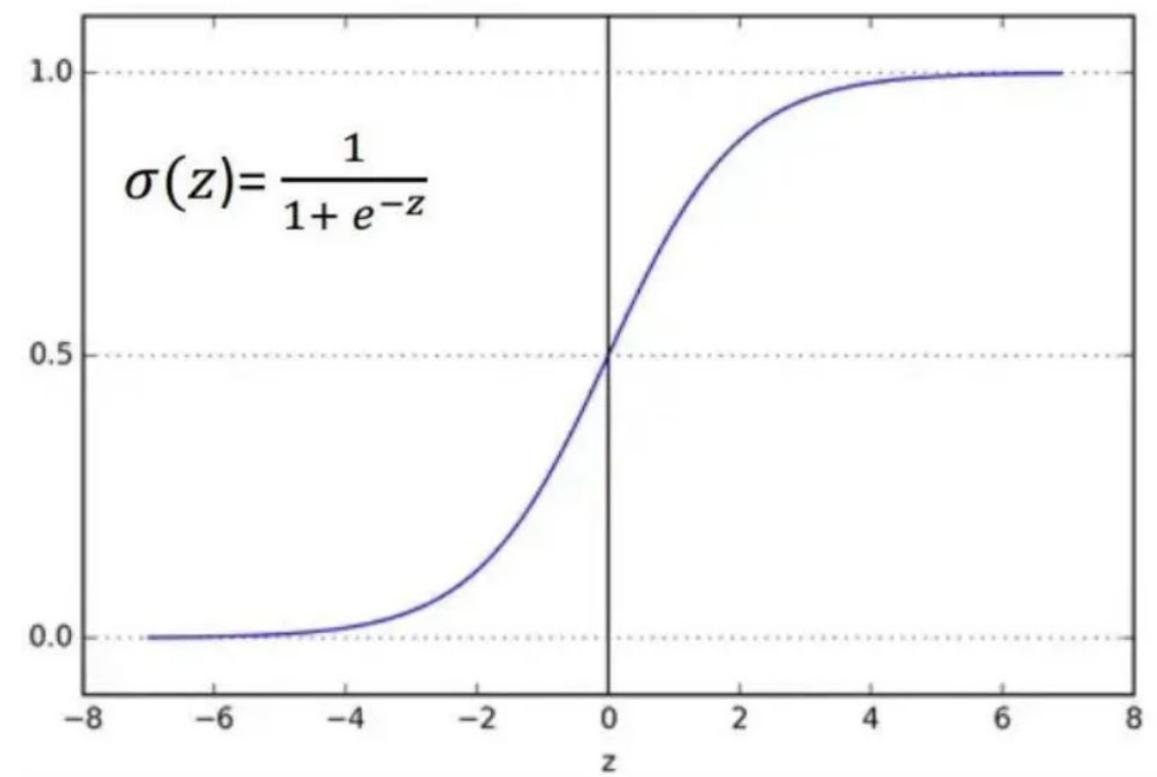
$$y = G(c) = G(2.2) = 2.2$$

Logistic regression

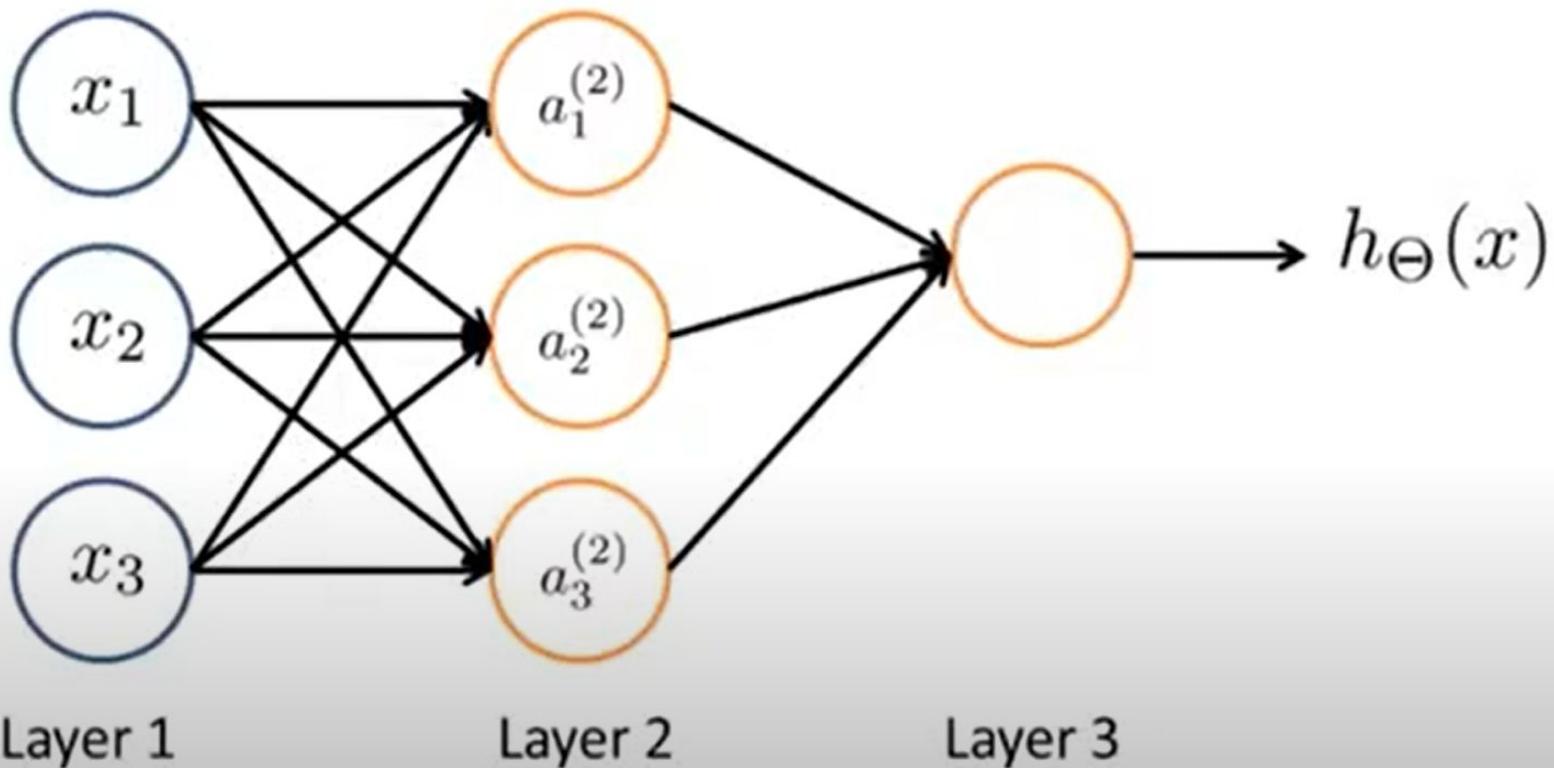
Example: Sigmoid activation



Logistic Regression (Photo Credit: [Joshua Goings](#))

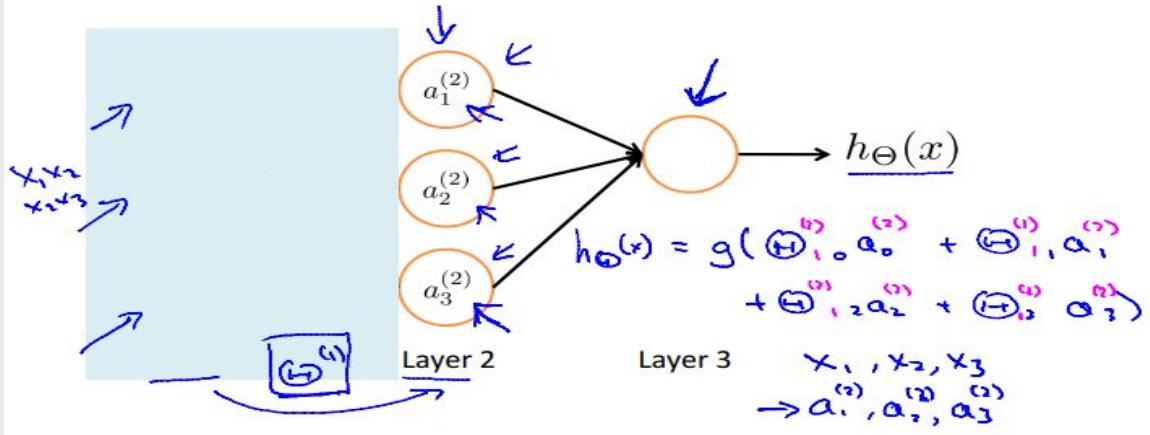


FROM LINEAR AND NON LINEAR MODELS FROM A XAI PERSPECTIVE

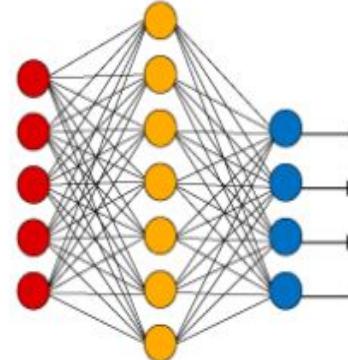


FROM LINEAR AND NON LINEAR MODELS FROM A XAI PERSPECTIVE

Neural Network learning its own features

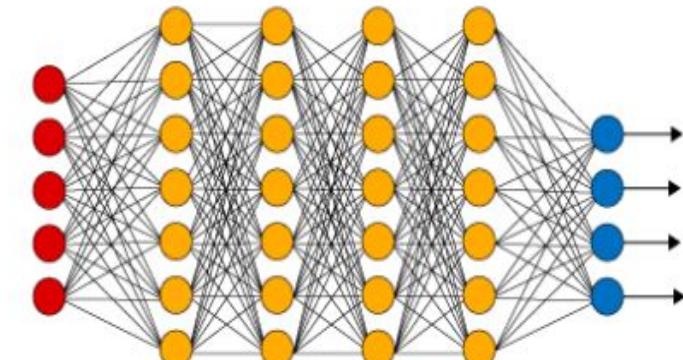


Simple Neural Network



Input Layer

Deep Learning Neural Network



Hidden Layer

Output Layer

What this neural network is doing is just like logistic regression, except that rather than using the original features x_1, x_2, x_3 , is using these new features a_1, a_2, a_3 . This algorithm has the flexibility to try to learn whatever features at once, using these a_1, a_2, a_3 in order to feed into this last unit that's essentially a logistic regression here.

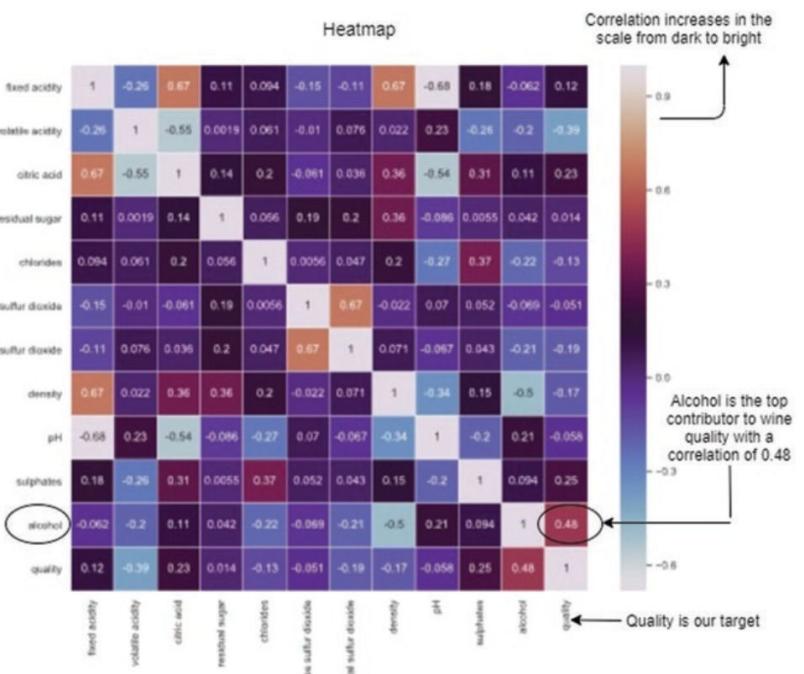
INTRINSIC EXPLAINABLE MODELS: LINEAR REGRESSION

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Top 5 rows of Wine Quality dataset

$$Y = m_0 + m_1x_1 + m_2x_2 + \cdots + m_kx_k$$

Correlations with target	
Alcohol	0.476166
Volatile acidity	-0.390558
Sulfates	0.251397
Citric acid	0.226373
Total sulfur dioxide	-0.185112
Density	-0.174919
Chlorides	-0.128907
Fixed acidity	0.124052
pH	-0.057731
Free sulfur dioxide	-0.050554
Residual sugar	0.013732



- Correlation is a measure of the degree of the linear relation between two variables
- It can vary from -1 (full negative correlation, one variable's increase makes the other to decrease) to 1 (positive correlation, the two variables increase together).
- Every variable has obviously correlation = 1 with itself

AGNOSTIC METHODS: PERMUTATION IMPORTANCE

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24



- **Permutation importance is calculated after a model has been fitted.** So we won't change the model or change what predictions we'd get for a given value of height, sock-count, etc.
- Instead we will ask the following question: If I randomly shuffle a single column of the data, leaving the target and all other columns in place, how would that affect the accuracy of predictions in that now-shuffled data?

AGNOSTIC METHODS: PERMUTATION IMPORTANCE

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

data = pd.read_csv('../input/fifa-2018-match-statistics/FIFA 2018 Statistics.csv')
y = (data['Man of the Match'] == "Yes") # Convert from string "Yes"/"No" to binary
feature_names = [i for i in data.columns if data[i].dtype in [np.int64]]
X = data[feature_names]
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state=1)
my_model = RandomForestClassifier(n_estimators=100,
                                  random_state=0).fit(train_X, train_y)
```

```
import eli5
from eli5.sklearn import PermutationImportance

perm = PermutationImportance(my_model, random_state=1).fit(val_X, val_y)
eli5.show_weights(perm, feature_names = val_X.columns.tolist())
```

Weight	Feature
0.1750 ± 0.0848	Goal Scored
0.0500 ± 0.0637	Distance Covered (Kms)
0.0437 ± 0.0637	Yellow Card
0.0187 ± 0.0500	Off-Target
0.0187 ± 0.0637	Free Kicks
0.0187 ± 0.0637	Fouls Committed
0.0125 ± 0.0637	Pass Accuracy %
0.0125 ± 0.0306	Blocked
0.0063 ± 0.0612	Saves
0.0063 ± 0.0250	Ball Possession %
0 ± 0.0000	Red
0 ± 0.0000	Yellow & Red
0.0000 ± 0.0559	On-Target
-0.0063 ± 0.0729	Offsides
-0.0063 ± 0.0919	Corners
-0.0063 ± 0.0250	Goals in PSO
-0.0187 ± 0.0306	Attempts
-0.0500 ± 0.0637	Passes

- Our example will use a model that predicts whether a soccer/football team will have the "Man of the Game" winner based on the team's statistics. The "Man of the Game" award is given to the best player in the game.

Partial dependence plot (PDP)

- The main strength of this permutation importance method is to provide a simple and direct answer about the most important feature.
- But it doesn't help no answering the "How": we may be interested or asked to answer how goal scored may change the predictions
- PDP sketches the functional form of the relationship between an input feature and the target
- What is performed under the covers by PDP method is to evaluate the effect of changes in a feature over multiple rows to get an average behavior and provide the related functional relationship.

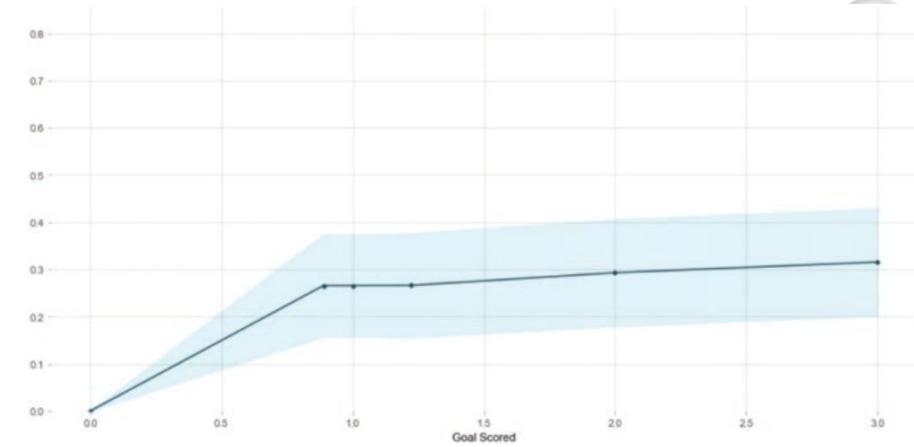


Fig. 4.4 Partial Dependence Plot diagram that shows how “Goal Scored” influences the prediction (Becker 2020)

PDP for feature “Distance Covered (Kms)”
Number of unique grid points: 10

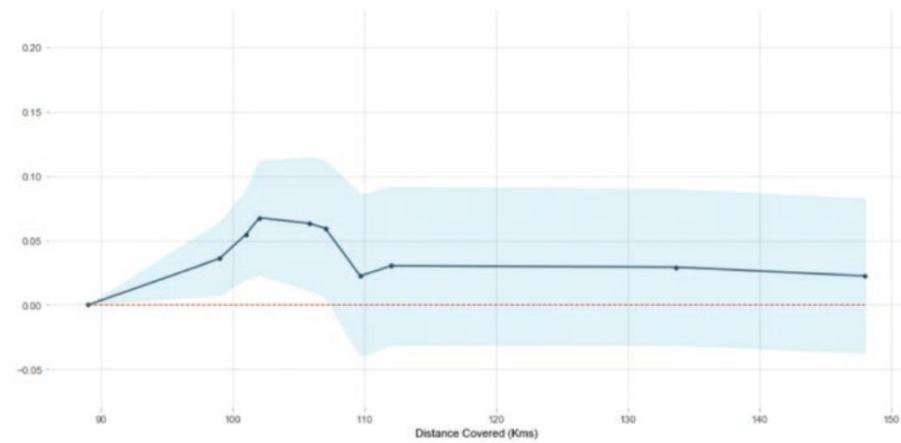


Fig. 4.5 Partial Dependence Plot diagram that shows how “Distance Covered” influences the prediction

Partial dependence plot (PDP)

- Looking at the single diagram of goal scored, it seems there is just a slight variation above one goal
- The maximum effect from distance covered is achieved around 100 km, but with more goals also longer distances produce the same overall effect.

PDP interact for "Goal Scored" and "Distance Covered (Kms)"
Number of unique grid points: (Goal Scored: 6, Distance Covered (Kms): 10)

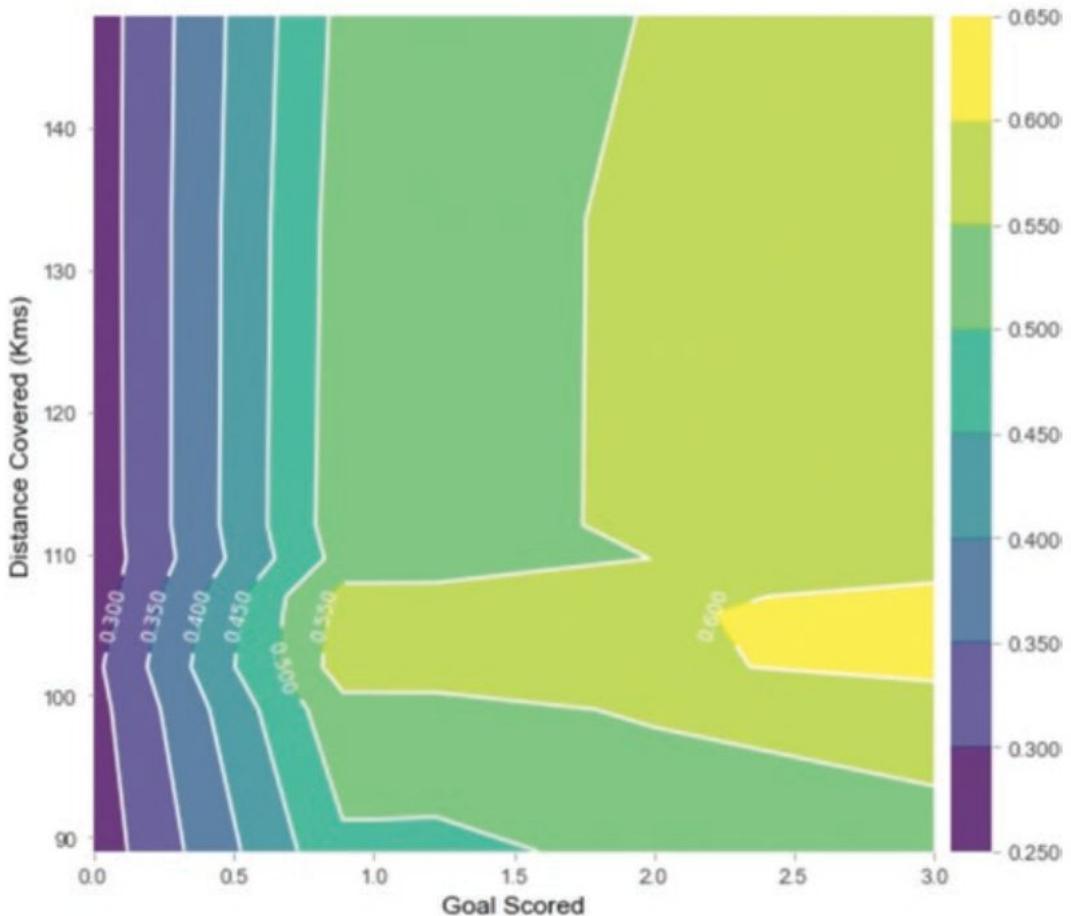


Fig. 4.6 PDP diagram that shows the interaction of the two main features and their impact on the prediction

SHAP (SHapley Additive exPlanations): A Game Theoretical Approach

- If we move to our working scenario of “Player of the Match” prize, so far we provided explanations about the most important features and the functional relationship of these features with the prediction, but we are not able to answer the direct question: considering the features in the figure, **how much the specific prediction for his match has been driven by the number of goals scored by Uruguay?**
- SHAP method relies on Shapley value, named by Lord Shapley in 1951 who introduced this concept to find solutions in cooperative games. To set the stage, game theory is a theoretical framework to deal with situations in which we have several individual players and we search for the optimal decisions that depend from the strategy adopted by the other players.
- You see on the left the list of features and on x axis the SHAP value. **The color of each dot represents if that feature is high or low for that specific row of data. The relative position of the dot on x axis shows if that feature contributed positively or negatively to the prediction. In this way, you may quickly assess if, for each prediction, the feature is almost flat or impacting a lot some rows and nothing to the others.**

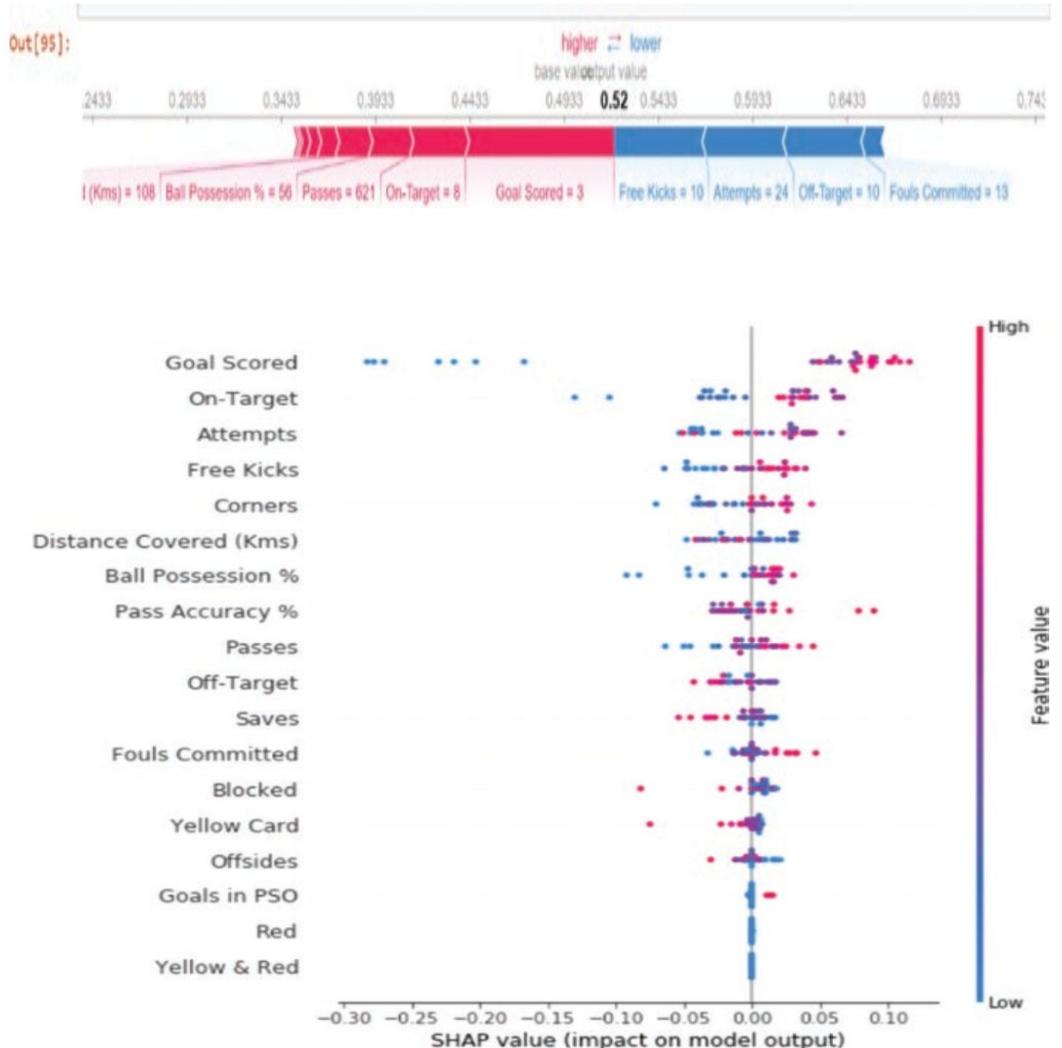


Fig. 4.8 SHAP diagram that shows the features’ ranking and the related impact on the match prediction (Becker 2020)

Occlusions as agnostic method

- Using the occlusion in the training phase, we force the black box not to learn by looking at the finer details.
- We can take a pre-trained black box and question it on image content using occlusions as a XAI method. The model has already been trained and fixed, so we don't care in this phase of training it in a robust way
- We want to understand which details of the image are most significant for the class' attribution or to evaluate the importance of some group of pixels

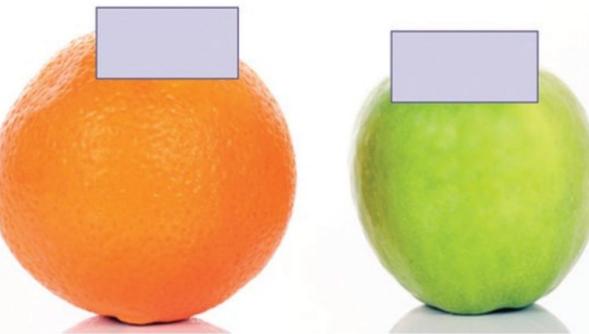


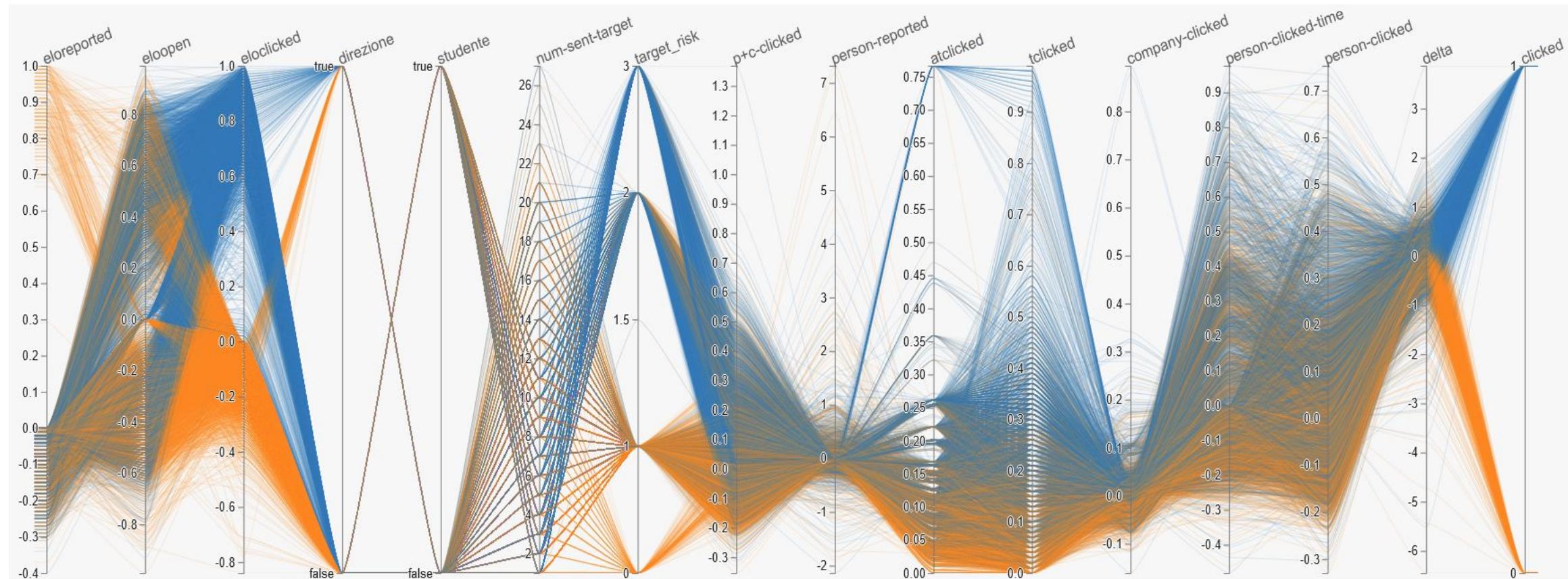
Fig. 5.4 The occlusion idea as an augmentation technique: random gray rectangles force the model to rely more on robust features such as skin's texture

Fig. 5.5 Original image



Fig. 5.6 Occlusions used to highlight relevant features respectively for the tabby cat and the dog classes

Features Exploration @Cyber Guru Phishing

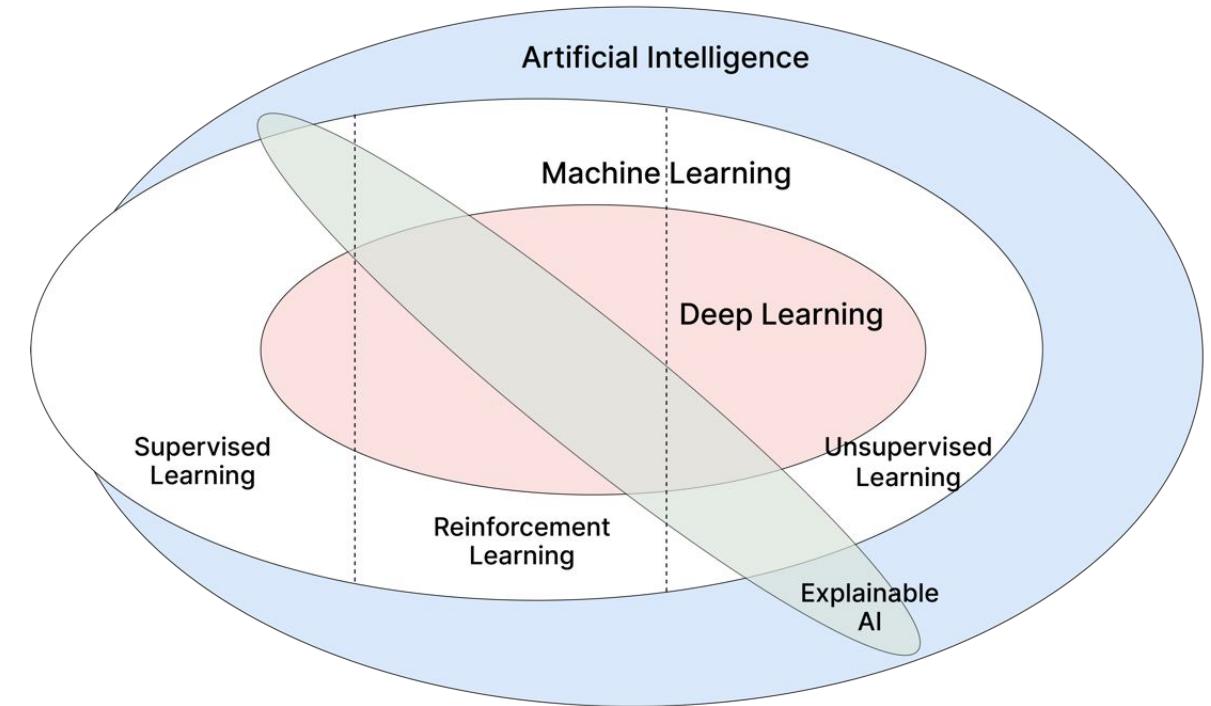
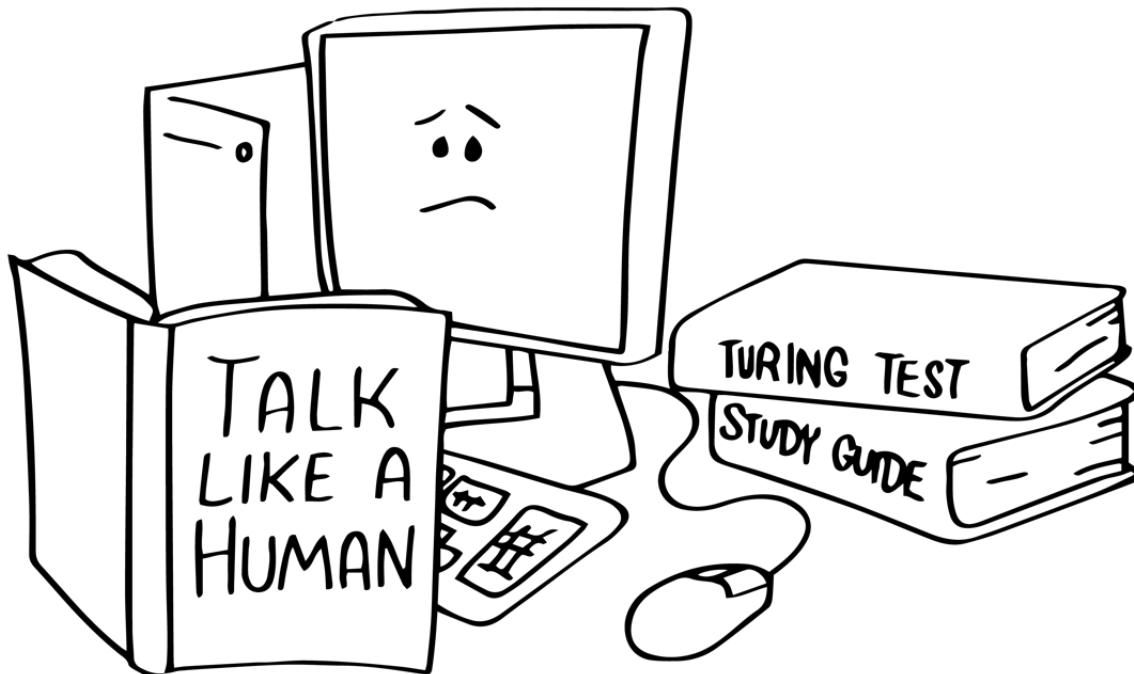


Parallel Plot, features importance

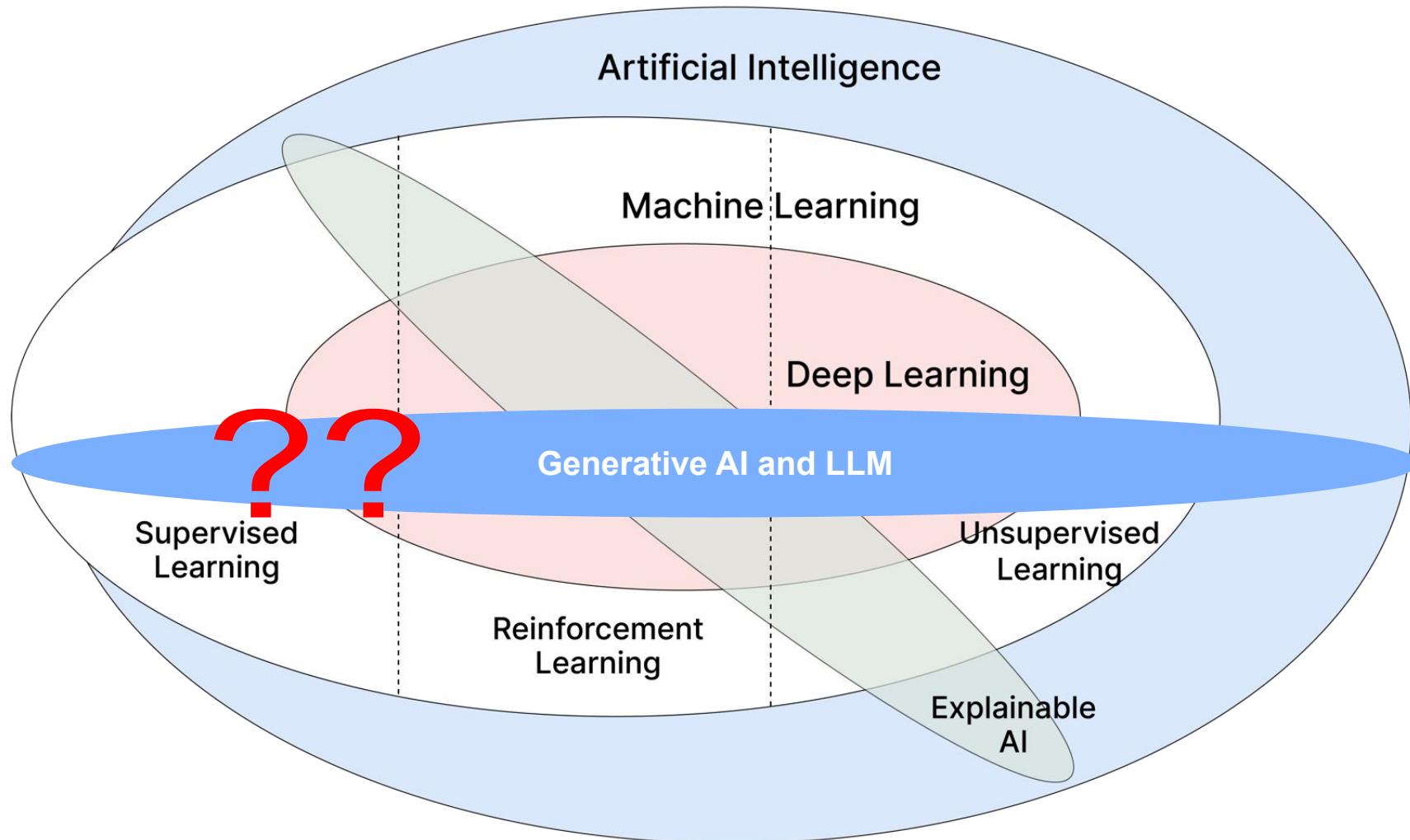
AI Taxonomy

“Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed”

(A. Samuel 1959)



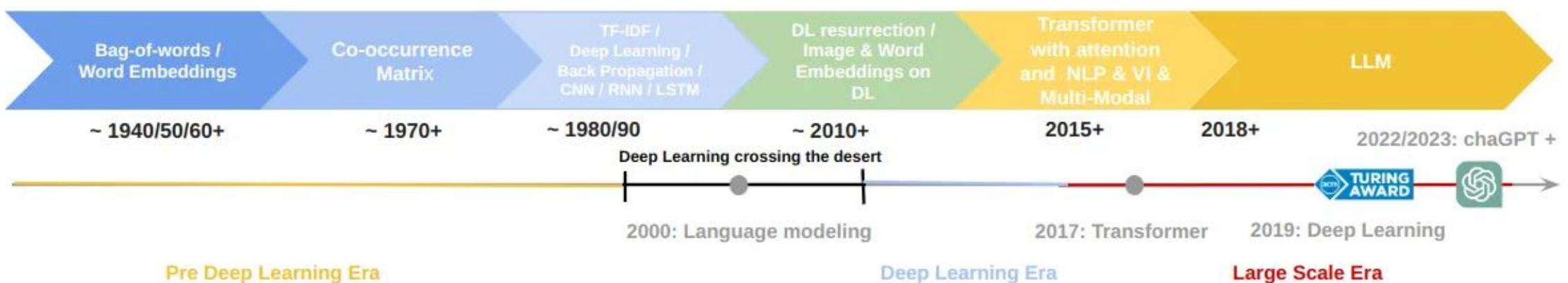
Updated AI Taxonomy



Generative AI and LLM

Generative AI is a type of artificial intelligence that is able to create original content, such as text, images, music, and video. These systems are trained on large datasets of human-generated content and are able to generate new pieces of content that are original and impressive.

There are various techniques used in generative AI, such as generative adversarial networks (GANs) and transformers. **GANs** consist of two neural networks, a generator and a discriminator, which work together to produce new content that resembles the source data. **Transformers**, such as GPT-3, are trained to understand language or images and generate text or images from large datasets. Variational auto-encoders (VAEs) are another technique used in generative AI, which involves learning the underlying structure of a dataset and generating new data that fits this structure.



XAI @ LLMs

- In contrast to traditional deep learning models, the scale of LLMs in terms of parameters and training data introduces both complex challenges and exciting opportunities for explainability research.
- On the one hand, traditionally practical feature attribution techniques, such as gradient-based methods and SHAP values, could demand substantial computational power to explain LLMs with billions of parameters.
- This makes these explanation techniques less practical for real-world applications that end-users can utilize

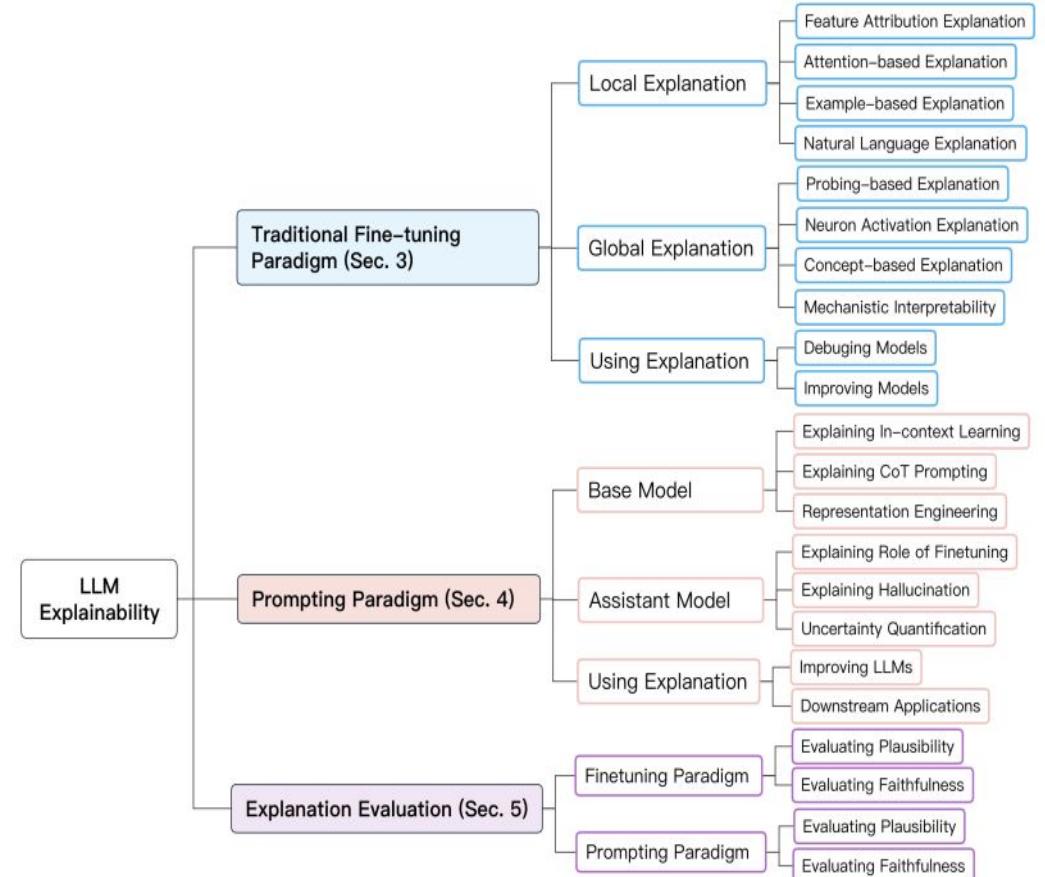


Figure 1: We categorize LLM explainability into two major paradigms. Based on this categorization, we summarize different kinds of explainability techniques associated with LLMs belonging to these two paradigms. We also discuss evaluations for the generated explanations under the two paradigms.

Large Language Model (LLM)

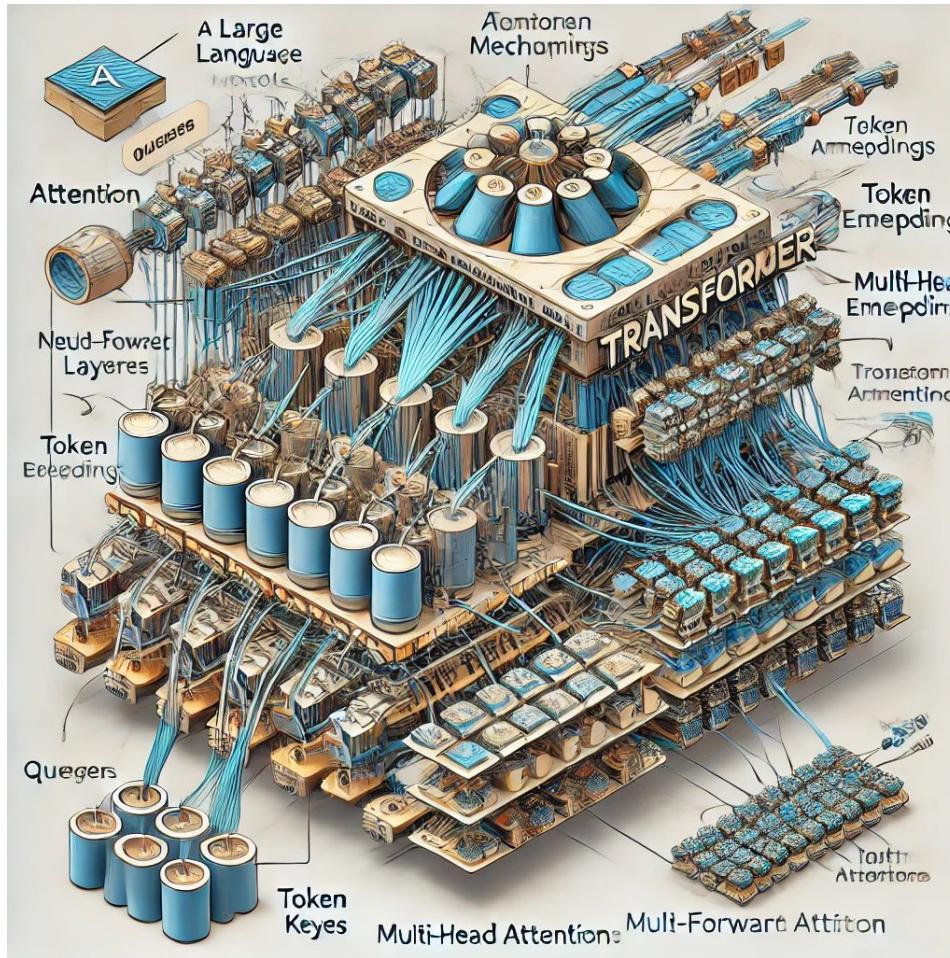


Image generated with GPT-4o

LLM

- **Large Language Model** (LLM) are giant neural networks with billions of parameters (weights), trained on large quantities of unlabelled text using self-supervised learning
- A message is splitted in **tokens** (sub-word)
- Each token is translated in a number using an operation called **embeddings**
- LLM works by taking an input text and **repeatedly predicting** the next token or word

Size of GPT-4

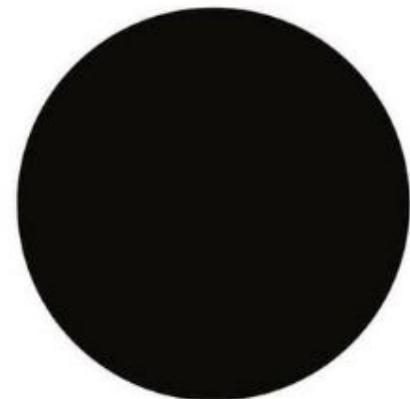
- Around **1.76 trillion** parameters
- Neural network with **120** layers
- Process up to **25,000** words at once
- Estimated training cost is \$200M using 10,000 Nvidia A100 GPU for 11 months

GPT-3



175 billion

GPT-4

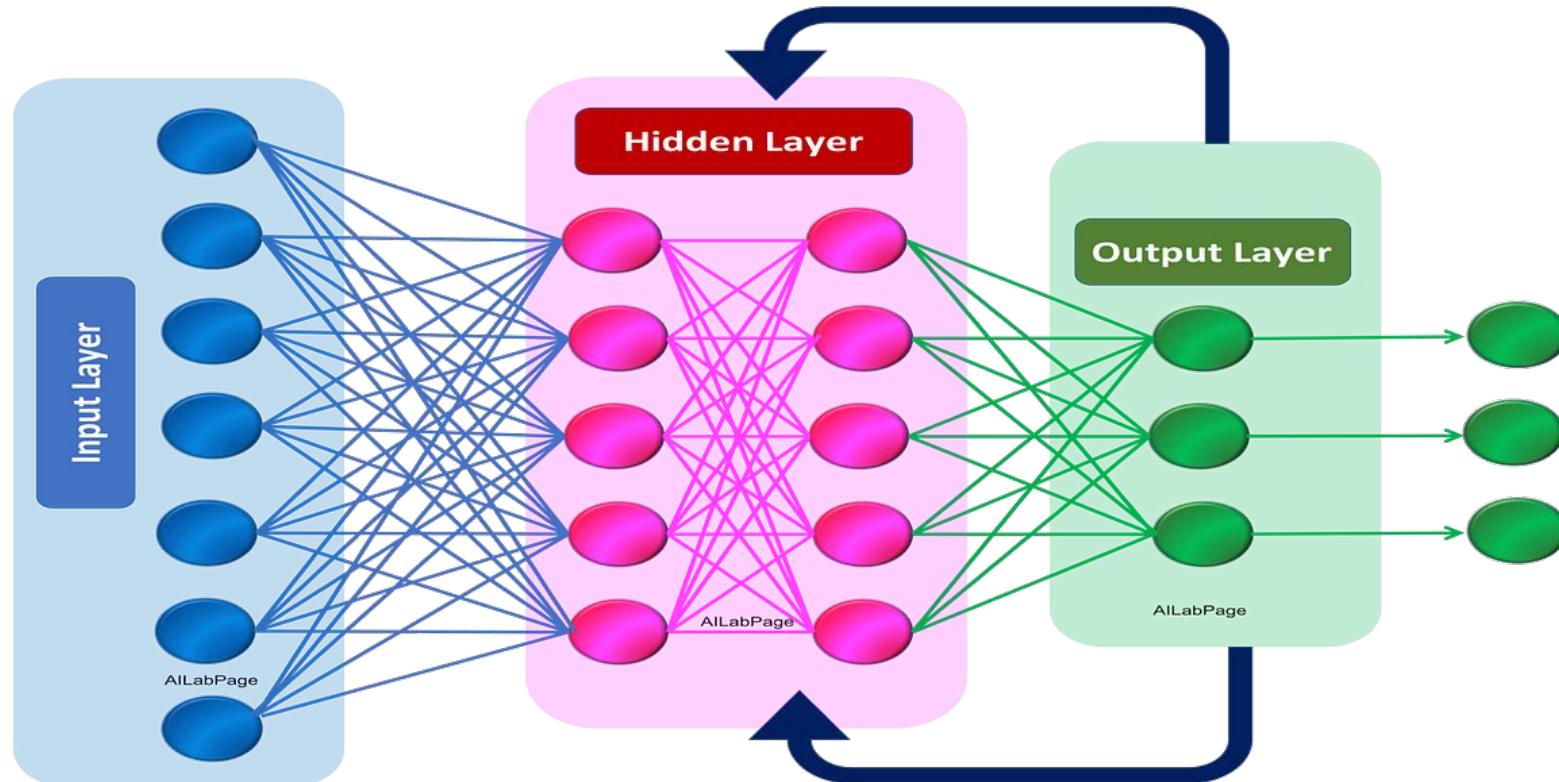


1.76 trillion

RNN, before LLM

- **Recurrent Neural Networks (RNN)**
- Prediction of the next words based on the previous words
- RNN does not scale
- To complete a sentence the model needs to understand the structure of the entire sentence
- Eg. “The teacher taught the students with the book”
 - Did the teacher teach using the book?
 - Did the student have the book?
 - Or was it both?

RNN (2)



Attention Is All You Need

- Google and University of Toronto published a paper in 2017 “[Attention is All You Need](#)”
- In this paper they introduced the **Transformers architecture**
- This novel approach unlocked the progress in generative AI that we see today
- Scale efficiently, parallel process, attention to input meaning

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

Lukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

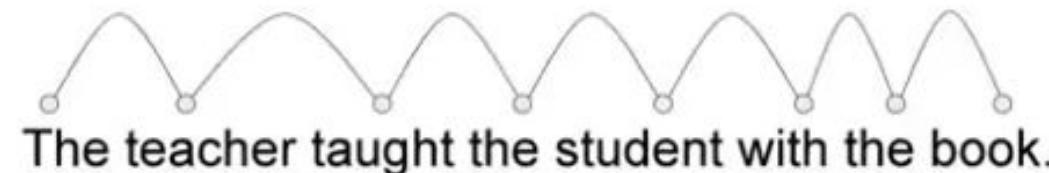
*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

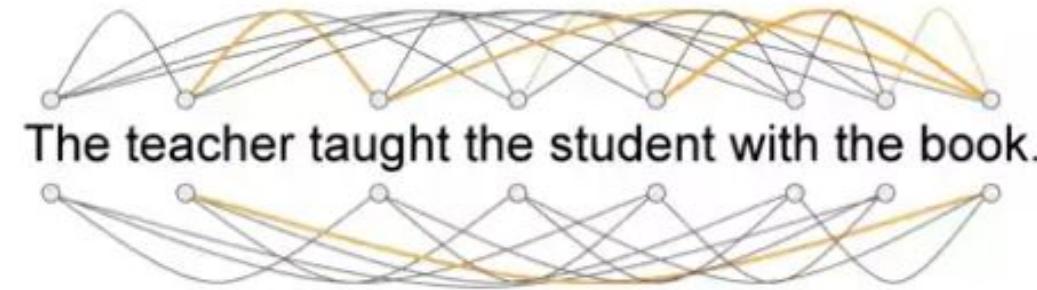
‡Work performed while at Google Research.

RNN vs Transformers

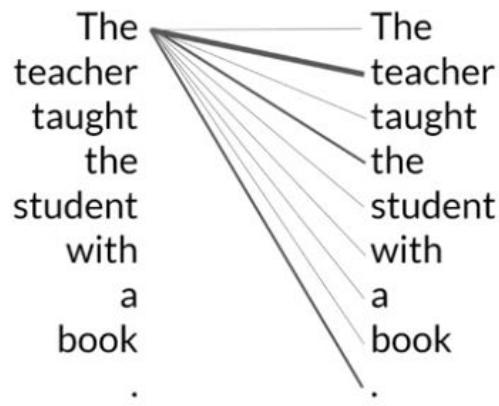
RNN



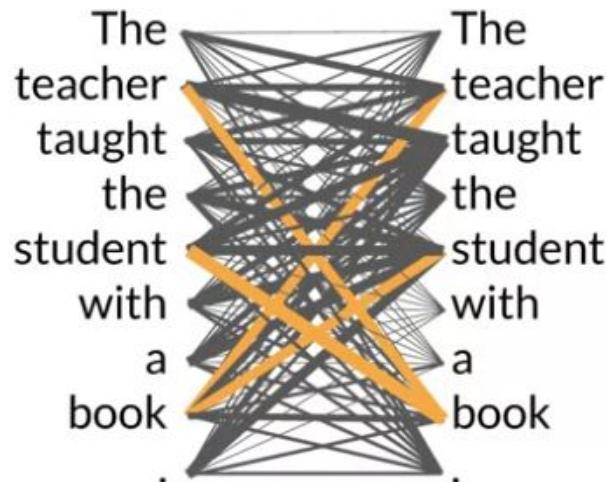
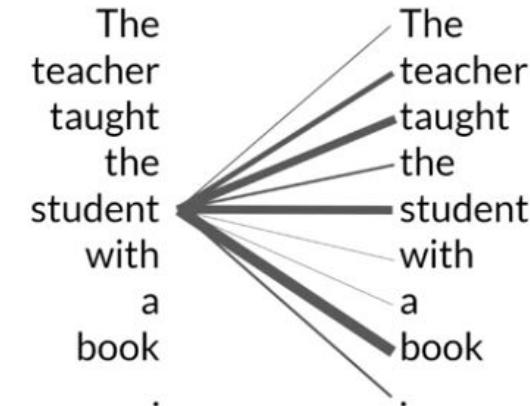
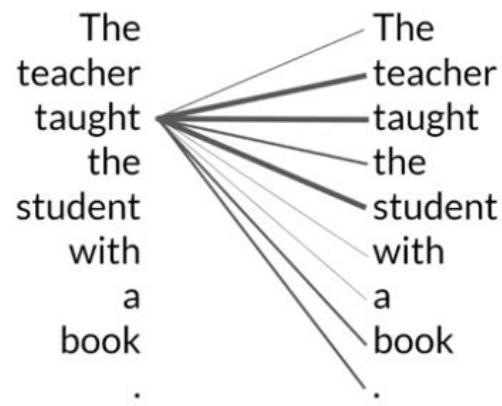
Transformers



Attention map

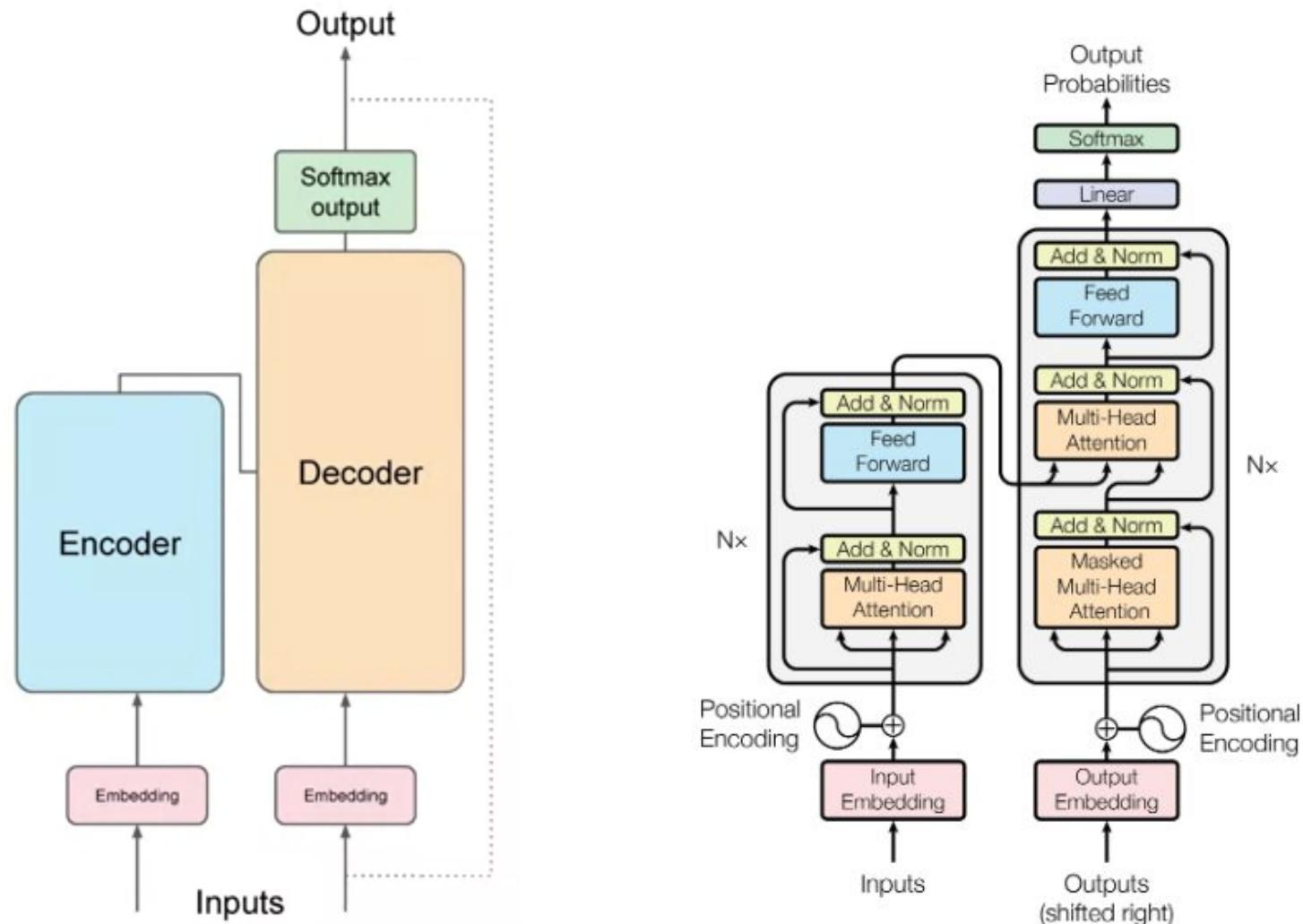


eg. **book** is
strongly
connected
with **teacher**
and **student**



self-attention

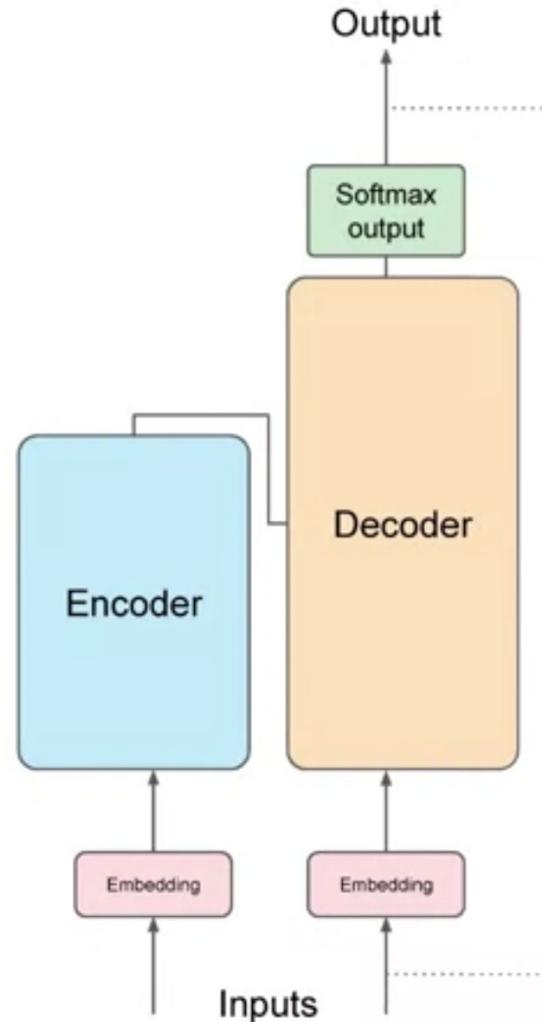
Transformers architecture



Encoder and Decoder

Encoder

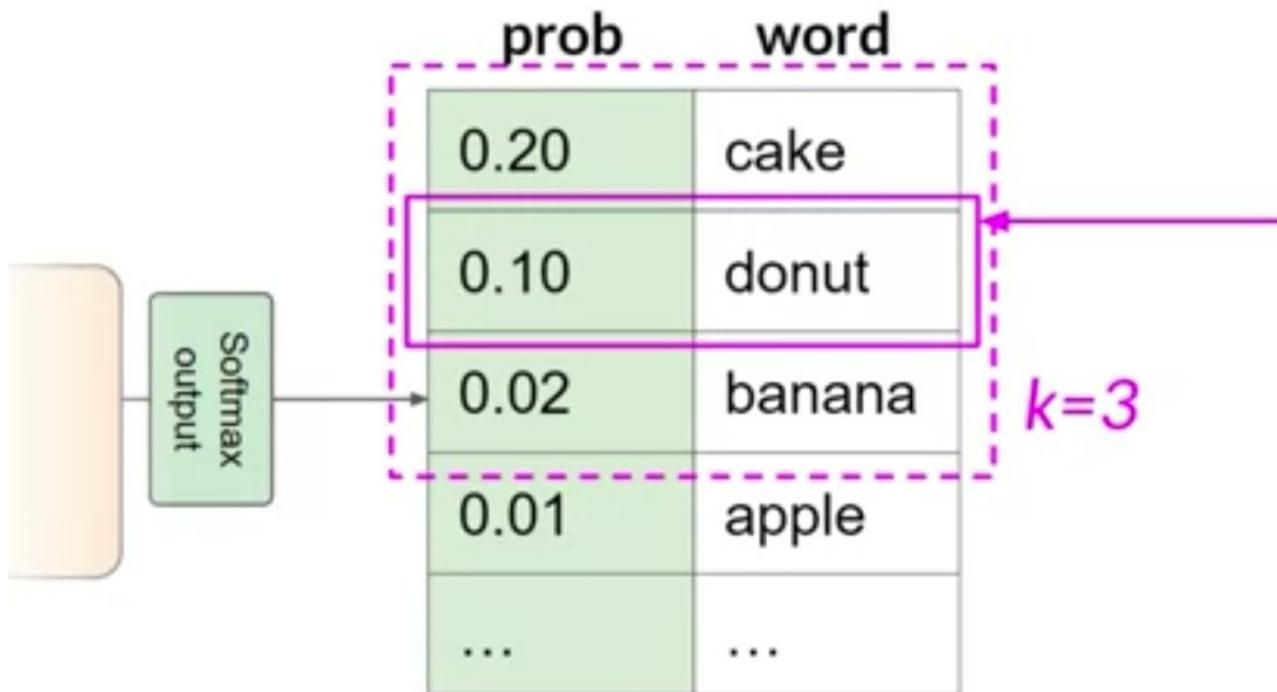
Encodes inputs (“prompts”) with contextual understanding and produces one vector per input token



Decoder

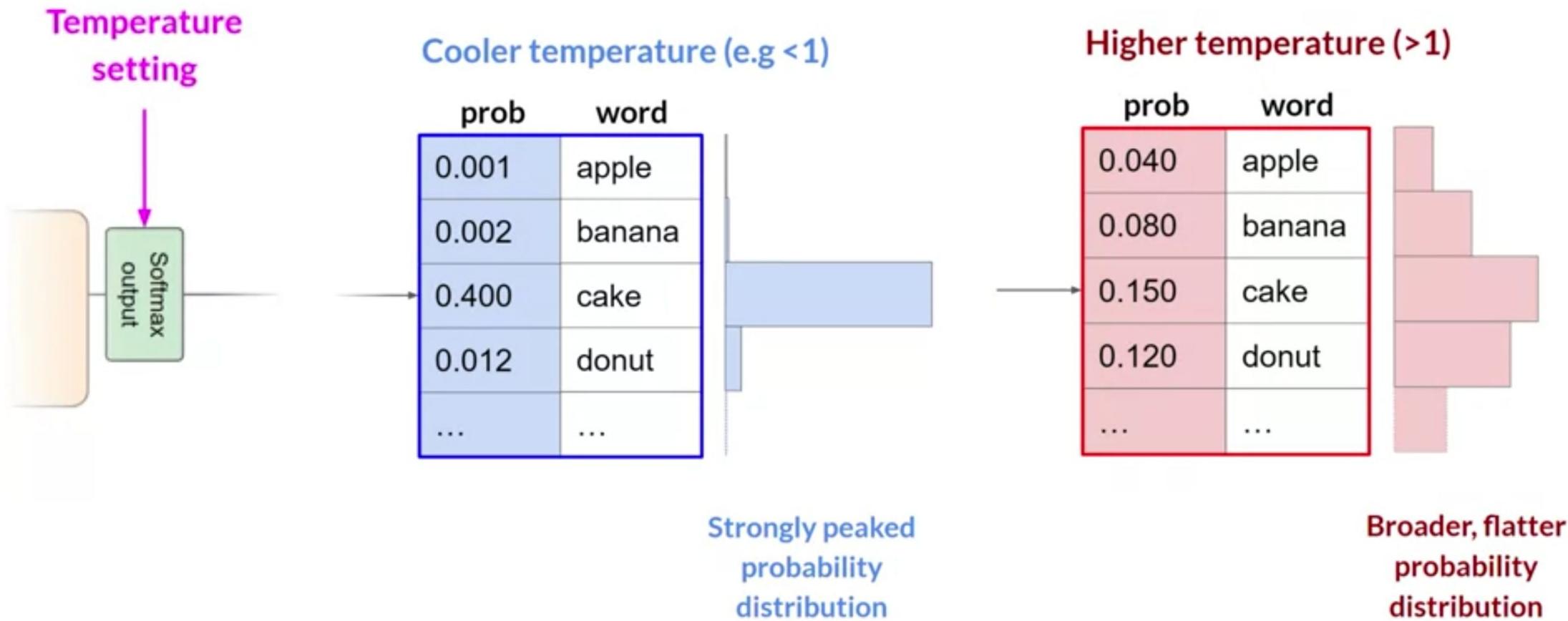
Accept input tokens and generate new tokens

Top-k



top-k: select an output from the top-k results after applying random-weighted strategy using the probabilities

Temperature



LLM visualization

<https://bbycroft.net/llm>

LLM Visualization

Chapter: Overview

GPT-2 (small) nano-gpt GPT-2 (XL) GPT-3

Table of Contents

- Intro
- Introduction**
- Preliminaries
- Components
- Embedding
- Layer Norm
- Self Attention
- Projection
- MLP
- Transformer
- Softmax
- Output

How to predict

text tokens words

2437 284 4331 16326 2420 2456

LLM

tok embed

pos embed

transformer i

layer norm

multi-head, causal self-attention

layer norm

feed forward

layer norm

linear

softmax

Welcome to the walkthrough of the GPT large language model! Here we'll explore the model *nano-gpt*, with a mere 85,000 parameters.

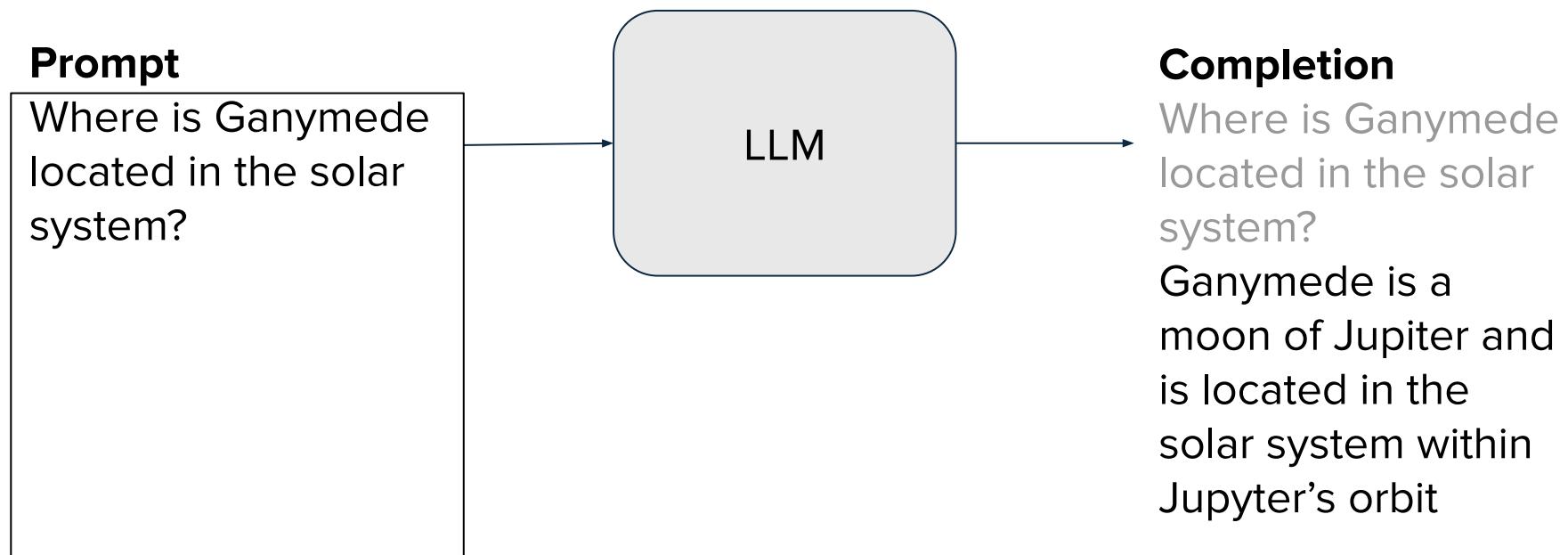
Its goal is a simple one: take a sequence of six letters:

C B A B B C

and sort them in alphabetical order, i.e. to "ABBBCC".

nano-gpt
n_params = 85,584

Prompt



Context window: few thousand words

Prompt engineering

- You can encounter situations where the model doesn't produce the outcome that you want on the first try
- You may have to revisit the language several times to get a good answer
- The development and improvement of the prompt is known as **prompt engineering**
- One powerful strategy is to include examples of the task that you want the model to carry out inside the prompt
- This is called **In-Context Learning (ICL)**

ICL - zero shot inference

Prompt

Classify this review:

I loved this movie!

Sentiment:



Completion

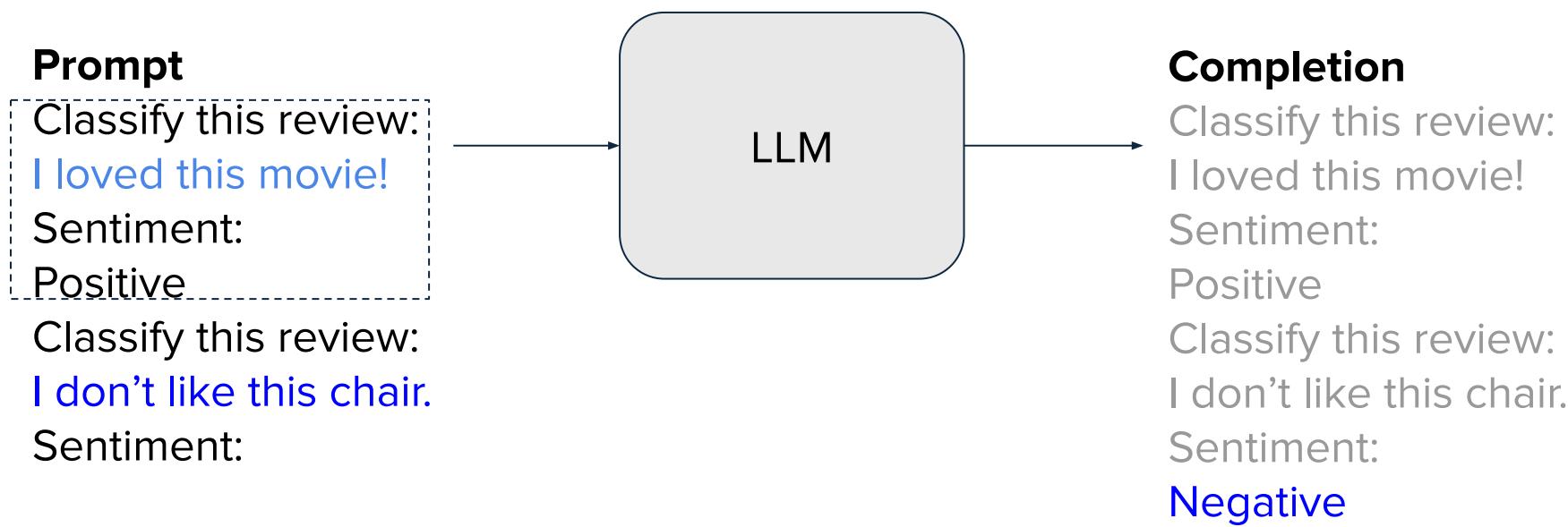
Classify this review:

I loved this movie!

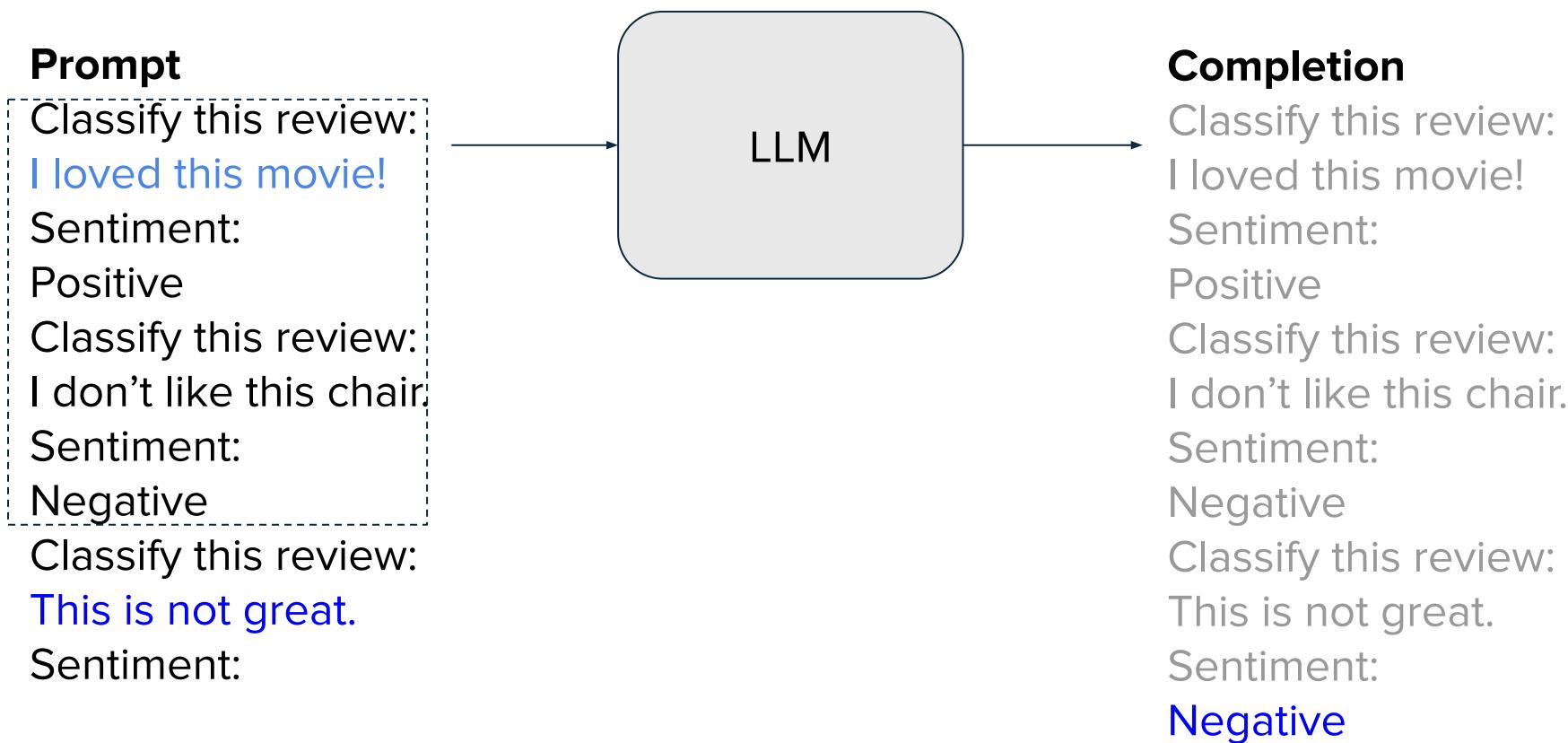
Sentiment:

Positive

ICL - one shot inference

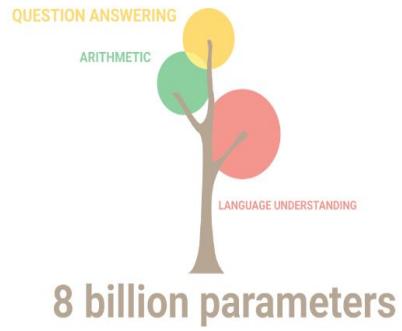


ICL - few shot inference



LLM emerging properties

“Emergence is when quantitative changes in a system result in qualitative changes in behavior.” (P. Anderson, 1972)



“..Only twenty years ago we expected to have to solve two tasks separately, modeling language and the world, and then combine them. Things turned out differently, and I wonder if the distinction between understanding the world and understanding language isn't arbitrary, and if another kind of mind might not draw very different boundaries..” N. Cristianini

“These things are totally different from us,” he says. “Sometimes I think it’s as if aliens had landed and people haven’t realized because they speak very good English.” G. Hinton

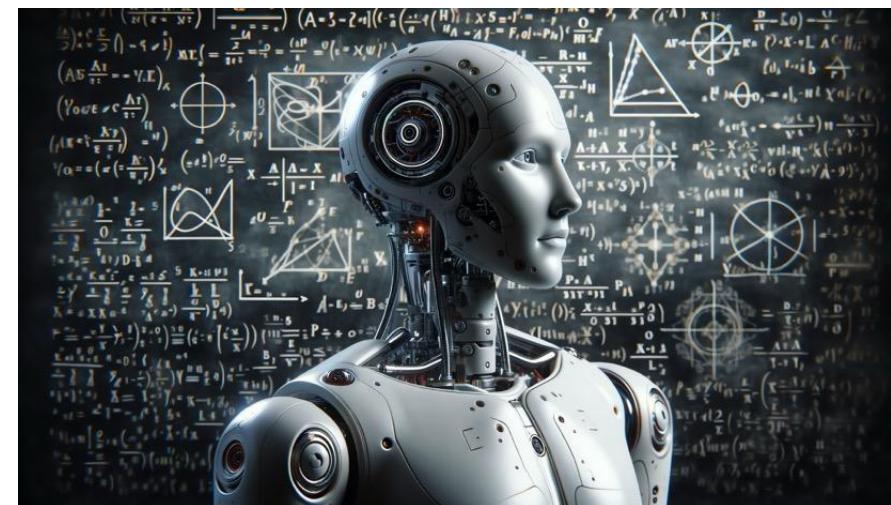
Artificial General Intelligence

The Chinese room argument holds that a digital computer executing a **program cannot have a "mind", "understanding", or "consciousness"**, regardless of how intelligently or human-like the program may make the computer behave.

The argument was presented by philosopher John Searle in his paper **"Minds, Brains, and Programs"**, published in **Behavioral and Brain Sciences** in 1980.

“...Because of this, I have mixed feelings about attempts to come up with new definitions of artificial general intelligence (AGI). I believe that most people, including me, currently think of AGI as AI that can carry out any intellectual task that a human can. With this definition, I think we’re still at least decades away from AGI. This creates a temptation to define it using a lower bar, which would make it easier to declare success: the easiest way to achieve AGI might be to redefine what the term means!”

Andrew NG





September 12, 2024

Introducing OpenAI o1-preview

A new series of reasoning models for solving
hard problems. Available starting 9.12

We've developed a new series of AI models designed to spend more time thinking before they respond. They can reason through complex tasks and solve harder problems than previous models in science, coding, and math.

Today, we are releasing the first of this series in ChatGPT and our API. This is a preview and we expect regular updates and improvements. Alongside this release, we're also including evaluations for the next update, currently in development.

➤ **Many of these New & Exciting ideas involve introducing increasingly opaque abstraction layers.** They promise to push us towards The Future, yet only bring us further from understanding our own abilities and needs. It's easy to sell ideas like these. What isn't easy, is creating something both practical and sustainable. If we want to make the world more sustainable, we need to understand the inputs, outputs, dependencies, constraints, and implementation details of the systems we rely on. Whenever we make it more difficult to know something, we inch closer to an information dark age.

➤ **Fortunately, there are still hackers.** For every smokescreen that clouds our vision, hackers help to clear the air. For every new garden wall erected, hackers forge a path around it. For every lock placed on our own ideas and cultural artifacts, hackers craft durable picks to unshackle them. Hackers try to understand what lies beyond their perspective. Hackers focus on what is real, and what is here.

==Phrack Inc.==
Volume 0x10, Issue 0x47, Phile #0x01 of 0x11
=[Introduction]=
-[Phrack Staff]=
-[staff@phrack.org]=
-[August 19, 2024]=
--[Breaking The Spell

It can feel like the world is in a dreamlike state; a hype-driven delirium, fueled by venture capital and the promises of untold riches and influence. Everyone seems to be rushing to implement the latest thing, hoping to find a magic bullet to solve problems they may not have, or even understand.

While hype has always been a thing, in the past few years (2020-2024), we have witnessed several large pushes to integrate untested, underdeveloped, and unsustainable technology into systems that were already Going Through It. Once the charm wears off, and all the problems did not just magically disappear, they drop these ideas and move on to the next, at the cost of everyone else.

Many of these New & Exciting ideas involve introducing increasingly opaque abstraction layers. They promise to push us towards The Future, yet only bring us further from understanding our own abilities and needs. It's easy to sell ideas like these. What isn't easy, is creating something both practical and sustainable. If we want to make the world more sustainable, we need to understand the inputs, outputs, dependencies, constraints, and implementation details of the systems we rely on. Whenever we make it more difficult to know something, we inch closer to an information dark age.

After the past several decades of humanity putting all of its collective knowledge online, we are seeing more ways to prevent us from accessing it. Not only is good information harder to find, bad information is drowning it out. There are increasing incentives to gatekeep and collect rent on important resources, and to disseminate junk that is useless at best, and harmful at worst. In all of this chaos, the real threat is the loss of useful, verified, and trusted information, for the sake of monetizing the opposite.

Fortunately, there are still hackers. For every smokescreen that clouds our vision, hackers help to clear the air. For every new garden wall erected, hackers forge a path around it. For every lock placed on our own ideas and cultural artifacts, hackers craft durable picks to unshackle them. Hackers try to understand what lies beyond their perspective. Hackers focus on what is real, and what is here.

We can move forward through this bullshit. We can work together to maintain good information, and amplify the voices of those who are creating and curating it. We can learn how things actually work, share the details, and use these mechanisms to do some good. We can devise new methods of communication and collaboration, and work both within and between our communities to jam the trash compactor currently trying to crush us to death.

Hacking is both a coping mechanism and a survival skill. It represents the pinnacle of our abilities as humans to figure out how to use whatever tools we may have, in whatever way we can, to do what we need to do. Hacking is a great equalizer, a common dialect, a spirit that exists within all of us. It has the power to shape the world into one we want to live in.
The hacker spirit breaks any spell.

References

- Ashish Vaswan et al., [Attention Is All You Need](#), Proceedings of 31st Conference on Neural Information Processing Systems (NIPS 2017)
- Nello Cristianini, [Machina sapiens. L'algoritmo che ci ha rubato il segreto della conoscenza](#), Il Mulino (2024)
- Leonida Gianfagna, Antonio Di Cecco, [Explainable AI with Python](#), Springer 2021
- Rylan Schaeffer, Brando Miranda, Sanmi Koyejo, [Are Emergent Abilities of Large Language Models a Mirage?](#), <https://arxiv.org/abs/2304.15004>, 2023
- Microsoft Research, [Sparks of Artificial General Intelligence:Early experiments with GPT-4](#), <https://arxiv.org/abs/2303.12712>, 2023
- Jiawei Su*, Danilo Vasconcellos Vargas and Kouichi Sakurai, [One Pixel Attack for Fooling Deep Neural Networks](#), <https://arxiv.org/pdf/1710.08864>, 2019
- OpenAI Research, [Improving Language Understanding by Generative Pre-Training](#), 2018
- Haiyan Zhao et al., [Explainability for Large Language Models: A Survey](#), 2023
- Andrej Karpathy, [Let's build GPT: from scratch, in code, spelled out](#), Youtube video, 2023

Thank you!

Contacts:

enrico (at) zimuel.it
leonida.gianfagna (at) gmail.com

