# Steering Large-Language Models: Theory and Practice

Damien Garreau

Julius-Maximilians Universität Würzburg - CAIDAS

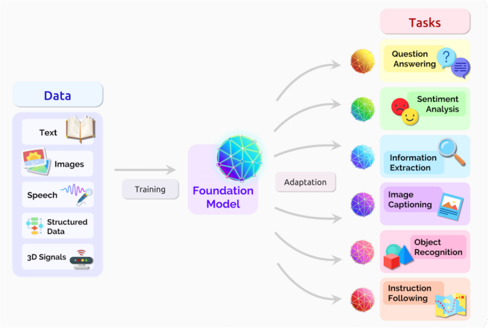
November 20, 2025

# Outline

# 1. Introduction

# Context

- foundation models trained on massive amount of data
- used downstream on any task




- **Figure:** from Bommasani et al., *On the Opportunities and Risks of Foundation Models*, Tech. Report., 2021

# Motivation

- **Problem:** hard to control their behavior
- **Example:** Grok's answers after an update in July 2025

# Steering

- **High-level idea:** from an existing model, detect and correct bad behavior (*a.k.a.* alignment)
- **Challenges:**
  - scale of the models
  - hurts the performance
  - where to begin with?
- **This talk:** steering
- **Other approaches (not this talk):**
  - fine-tuning[1]
  - reinforcement learning from human feedback[2]
  - prompt engineering[3]

---

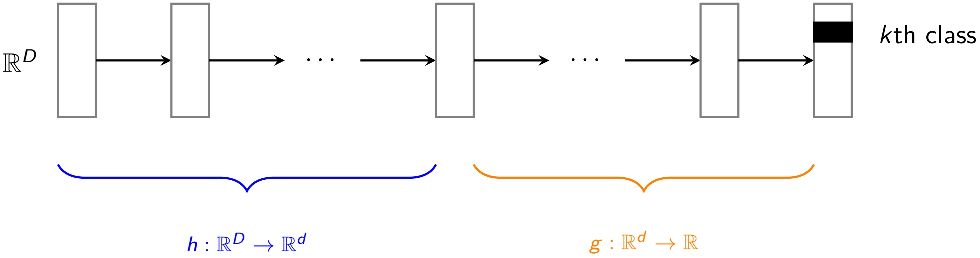
[1] Wei et al., *Finetuned Language Models Are Zero-Shot Learners*, ICLR, 2022
[2] Ziegler et al., *Fine-Tuning Language Models from Human Preferences*, preprint, 2019
[3] Brown et al., *Language models are few-shot learners*, NeurIPS, 2020

# 2. Concept algebra

# Activation space

**Definition:** we call *activation* the intermediate quantity computed at a neuron (before non-linearity). A given layer gives rise to the *activation space*.

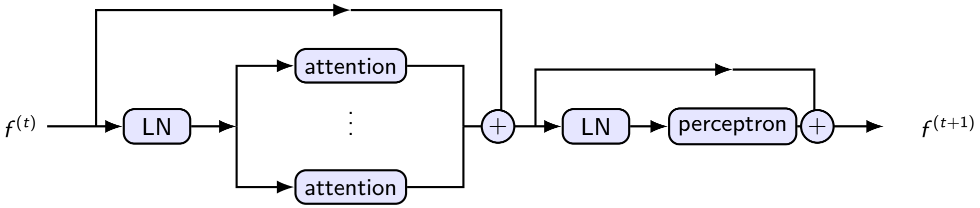
- **Example (i):** feed-forward multi-layer perceptron




- $h(x) \in \mathbb{R}^d$ is the latent representation (considered layer has $d$ hidden units)

# Activation space

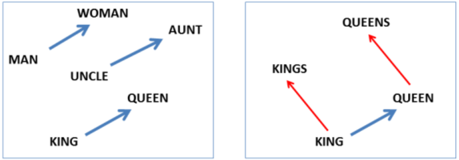
- **Example (ii):** GPT-like architectures




- get token representation in the first layer of the MLP
- **Beware:** we get token representations

# Concept algebra

- **Key observation:** one can do vector operations on the latent representations
- **Early example:** word vectors




- **Figure:** Figure 2 in Mikolov, Yih, Zweig, *Linguistic regularities in continuous space word representations*, Proc. NACL, 2013

# How to find good directions?

- **Extremely unintuitive:** why should linear modification in an inner layer have some smart effect on the output?
- **Further question:** how to find good directions?
- **What would be nice:** canonical basis in the activation space
- for non-mathematicians: hope that one neuron encodes for one high-level feature
- then it is simple: identify the neuron and modify its activation

# Visualizing concepts associated with individual neurons

- several ways of doing this, most intuitive:
- take some dataset, look for the images associated to max activation[4]
- **At first glance:** some neurons are *monosemantic*
- that is, neuron lights up in accordance to one type of high-level feature

- **Figure:** Figure 3(c) in Szegedy et al., *Intriguing properties of neural networks*, ICLR, 2014

[4] Goodfellow et al., *Measuring invariances in deep networks*, NeurIPS, 2009

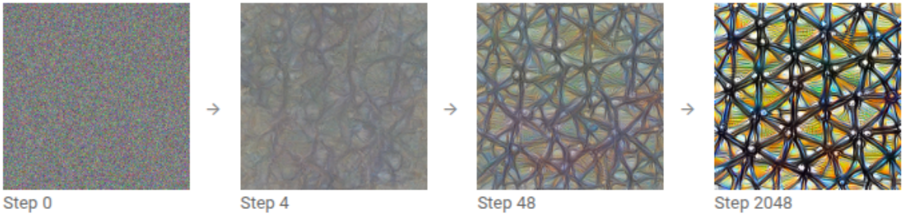
# Visualizing concepts associated with individual neurons

- **But not all of them:** many neurons are *polysemantic*

- **Figure:** Figure 3(a) in Szegedy et al., *Intriguing properties of neural networks*, ICLR, 2014
- **Even worse:** some random directions in the activation space also seem monosemantic
- maybe the granularity of the dataset prevents us from identifying what the neuron really encodes?

# Visualizing concepts associated with individual neurons

- **Another idea:** maximize the input activating the neuron by gradient descent
- starting from a random image
- can do it for early layers:




- **Figure:** maximizing the activity of unit 11 in layer `mixed4a` of GoogLeNet

# Visualizing concepts associated with individual neurons

- or we can do it for a class:

- **Figure:** credits Chris Olah
- **Conclusion:** most neurons are polysemantic, no easy way

# 3. Towards monosemanticity

## Superposition

- **Why does polysemanticity happen?**
- possible answer: too many *concepts* to pack in too few dimensions
- cannot associate each concept to an orthogonal direction
- neural nets are doing “the best they can” and finding nearly orthogonal directions
- **Toy model:**[5] consider input data $X \in \mathbb{R}^d$
- $x^{(i)}$ generated by first deciding if coordinate $j$ is non-zero independently with proba $\pi$
- then sampling coordinate value $\sim \mathcal{U}([0,1])$
- this is the so-called Bernoulli-Uniform model
- consider a simple auto-encoder

$$f_\theta(x) = \texttt{ReLU}(W^\top W x + b) \in \mathbb{R}^d ,$$

with $W \in \mathbb{R}^{m\times d}$, $b \in \mathbb{R}^d$

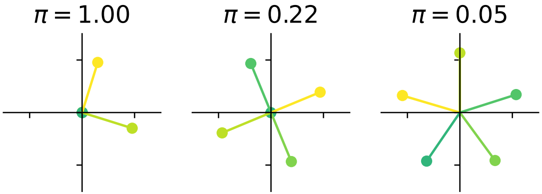
[5] from the transformer circuit thread

# Toy models of superposition

- train this model with (square) reconstruction loss:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left\| x^{(i)} - f_\theta(x^{(i)}) \right\|^2$$

- optimizer = AdamW[6]




- **Figure:** $d = 5$, $m = 2$, visualizing columns of $W$ after training for varying levels of sparsity
- this small model learns how to pack many features in 2D

[6] Loshchilov, Hutter, *Decoupled Weight Decay Regularization*, ICLR, 2019

# Sparse coding

- **Hypothesis:** *superposition* (too many concepts to encode for too few neurons)
- well, let us disentangle, and find a basis such that each latent is written with as few non-zero coefficients as possible
- **Intuition:** ideally,

$$x_i \approx 0.3 v_{\text{white}} + 0.5 v_{\text{flower}} \,.$$

- **Sparse coding:** assume training data $x_1, \ldots, x_n \in \mathbb{R}^D$
- we are looking for a dictionary $D \in \mathbb{R}^{d \times m}$

$$\frac{1}{n} \sum_{i=1}^{n} \min_{\alpha \in \mathbb{R}^m} \left[ \frac{1}{2} \left\| h_i - D\alpha_i \right\|^2 + \lambda \left\| \alpha_i \right\|_1 \right]$$

  is as small as possible
- here, $\alpha_1, \ldots, \alpha_n \in \mathbb{R}^m$ are coefficients
- $\ell_1$ norm promotes sparsity ($\lambda > 0$ is a regularization parameter)

# Atoms of discourse

- **Example:** can do this for word vectors[7]

| Atom 1978 | 825 | 231 | 616 | 1638 | 149 | 330 |
|---|---|---|---|---|---|---|
| drowning | instagram | stakes | membrane | slapping | orchestra | conferences |
| suicides | twitter | thoroughbred | mitochondria | pulling | philharmonic | meetings |
| overdose | facebook | guineas | cytosol | plucking | philharmonia | seminars |
| murder | tumblr | preakness | cytoplasm | squeezing | conductor | workshops |
| poisoning | vimeo | filly | membranes | twisting | symphony | exhibitions |
| commits | linkedin | fillies | organelles | bowing | orchestras | organizes |
| stabbing | reddit | epsom | endoplasmic | slamming | toscanini | concerts |
| strangulation | myspace | racecourse | proteins | tossing | concertgebouw | lectures |
| gunshot | tweets | sired | vesicles | grabbing | solti | presentations |

- **Figure:** from Arora et al., *Linear Algebraic Structure of Word Senses, with Applications to Polysemy*, Trans. ACL, 2018. Atoms = columns of $D$.

[7] Faruqui et al., *Sparse Overcomplete Word Vector Representations*, Proc. ACL, 2015

# Sparse autoencoders

- elements of dictionary still belong to $\mathbb{R}^d$ thus we are limited by the number of vectors we can pack
- **Idea:** extend the space
- set $\overline{h}_i := h_i - b_d \in \mathbb{R}^d$ the normalized latent representations
- parameterize the coefficients by $\alpha_i = \mathrm{ReLU}(W_e \overline{h}_i + b_e) \in \mathbb{R}^m$
- take
$$\frac{1}{n} \sum_{i=1}^{n} \left[ \left\| x_i - W_d \alpha_i - b_d \right\|^2 + \lambda \left\| \alpha_i \right\|_1 \right]$$
as objective function
- originally proposer by Subramanian, Suresh, Peters, *Extracting Latent Steering Vectors from Pretrained Language Models*, Findings of the ACL, 2022

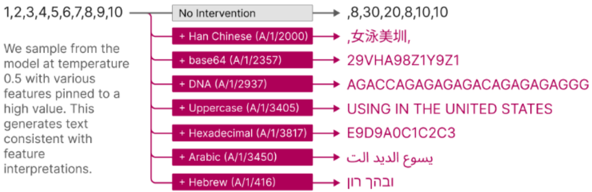
# Sparse autoencoders

- adapted to LLMs by Huben et al., [8], ICLR, 2024
- also a lot of material in Anthropic blog posts (the circuits thread)

[8] Sparse Autoencoders Find Highly Interpretable Features in Language Models

# Steering using sparse autoencoders

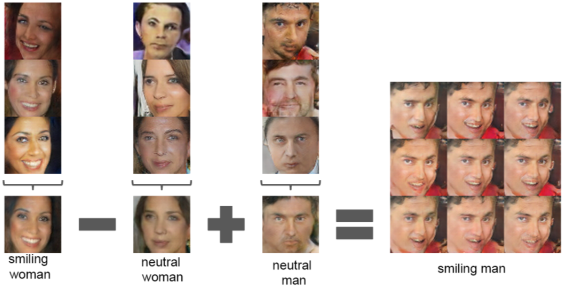
- one can *clamp* the activations values to steer the model
- **Example:**



# 4. Back to basics

# Early example

- **Early example:** image generation




- **Figure:** Figure 7 from Radford, Metz, Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, preprint, 2015

Thank you for your attention!