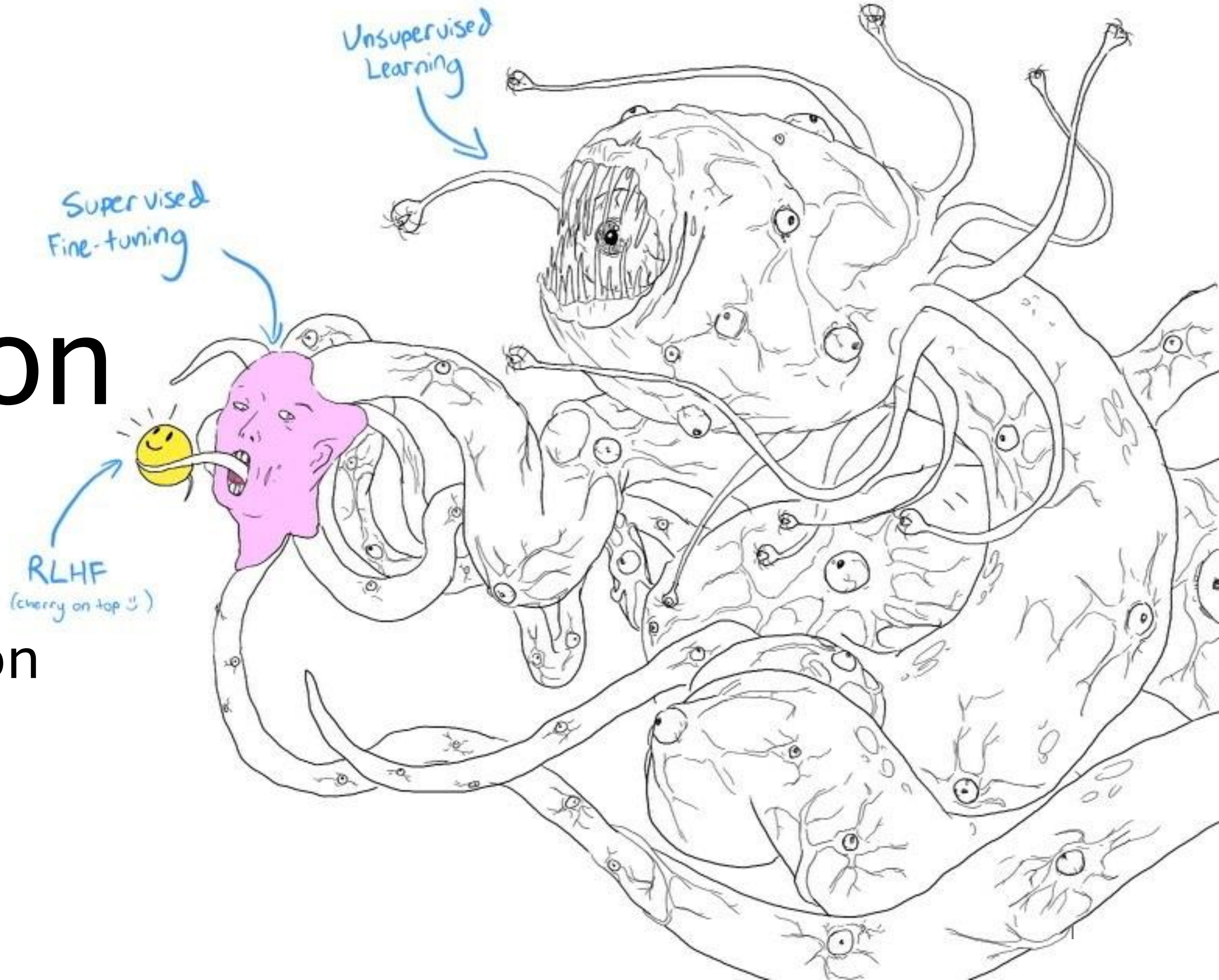
Introduction to LLM

Lecture 1: Introduction



Outline

- The Introduction to Introduction to LLM
  - Course organisation
  - Topics
- Deep Learning Basics
- NLP Basics and Linguistics Basics

Teaching Staff

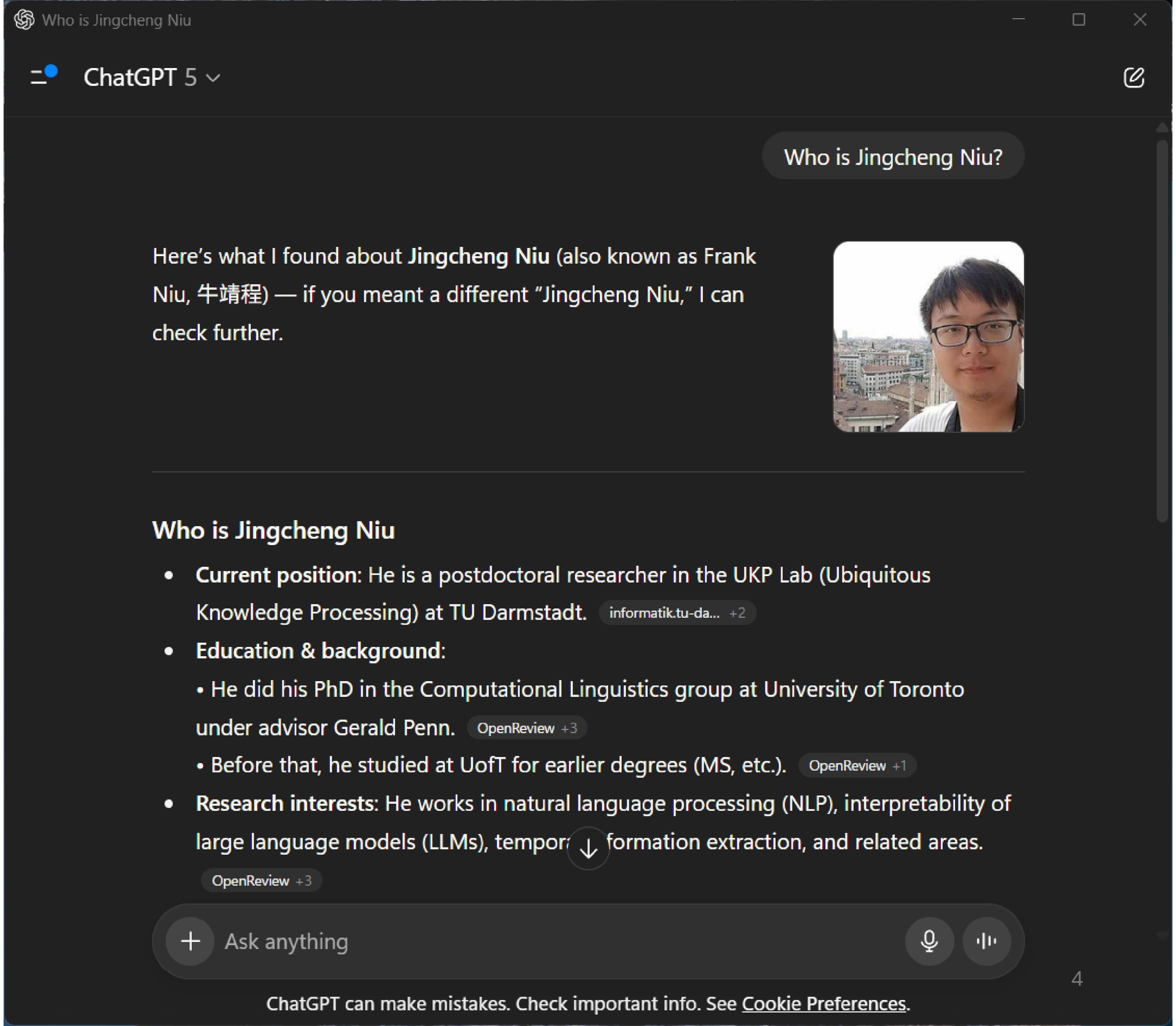
TAs: Jinke Lyu, Saleh Aslani, Mohammad Azimpour, Hanifi Ibrahim Akdag, Alexander Riedlinger, Anusha Siddapati Mohanreddy, and Vinayak Joshi.

Jingcheng (Frank) Niu
Lectures

Hovhannes Tamoyan
Practice Class

Hassan Soliman
Practice Class

About me


Who is Jingcheng Niu
ChatGPT 5
Who is Jingcheng Niu?
Here's what I found about **Jingcheng Niu** (also known as Frank Niu, 牛靖程) — if you meant a different "Jingcheng Niu," I can check further.
Who is Jingcheng Niu
Current position: He is a postdoctoral researcher in the UKP Lab (Ubiquitous Knowledge Processing) at TU Darmstadt. informatik.tu-da... +2
Education & background:
• He did his PhD in the Computational Linguistics group at University of Toronto under advisor Gerald Penn. OpenReview +3
• Before that, he studied at UofT for earlier degrees (MS, etc.). OpenReview +1
Research interests: He works in natural language processing (NLP), interpretability of large language models (LLMs), tempor formation extraction, and related areas.
OpenReview +3
Ask anything
ChatGPT can make mistakes. Check important info. See Cookie Preferences.

What is Computational Linguistics (CL) and Natural Language Process (NLP)?

- How we can build “computer systems” that can understand and use human language.
- Computational Linguistics (CL) ≈ Natural Language Process (NLP).

Nuclear bomb!

No that’s impossible.

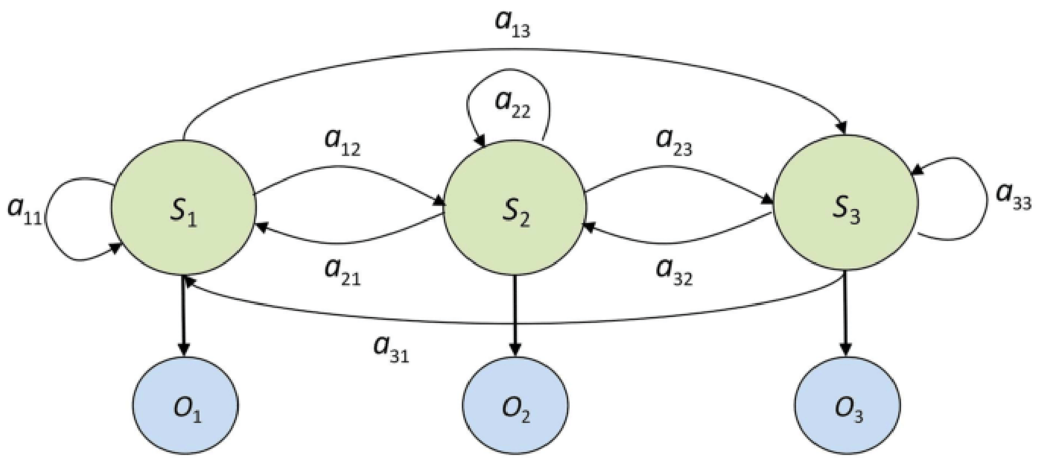
H bomb!

No that’s impossible.

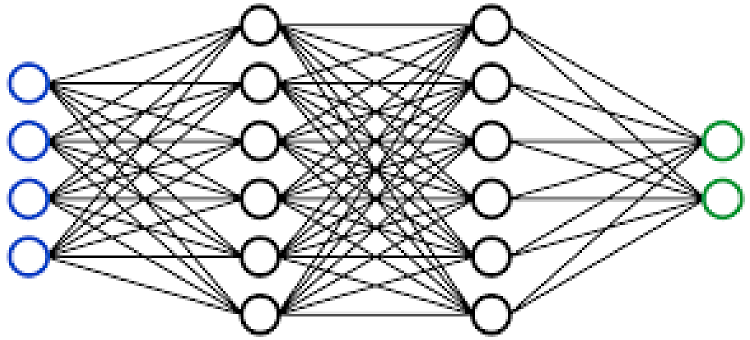
Symbolic vs. Statistical

"I think we are forced to conclude that... probabilistic models give no particular insight into some of the basic problems of syntactic structure."

— *Syntactic Structures*. Chomsky (1957).



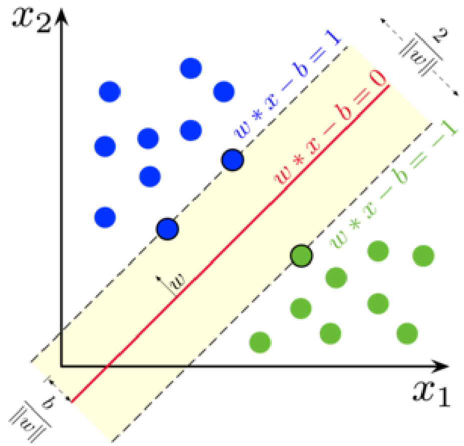
Hidden Markov Model



Neural Network

Random Number Generator*

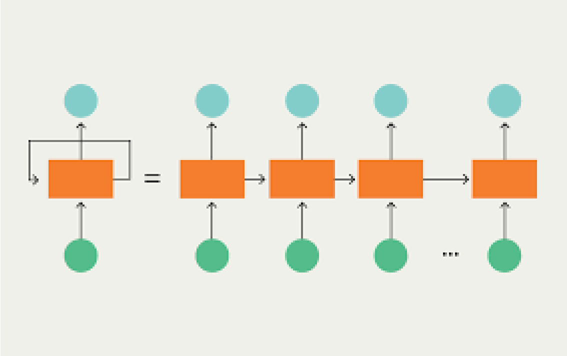
What Machine Learning Architecture?



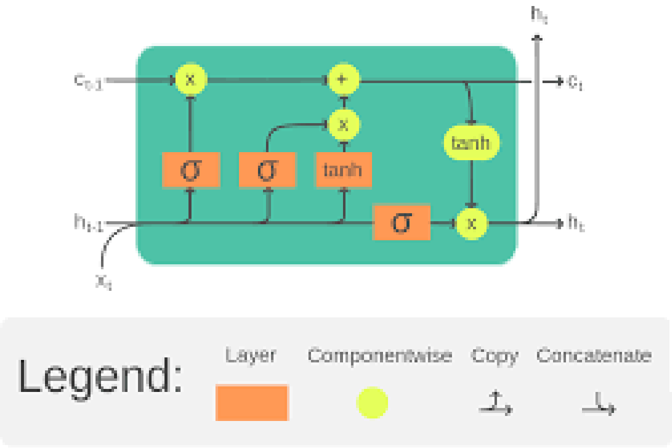
Support Vector Machine

Symbolic vs. Statistical

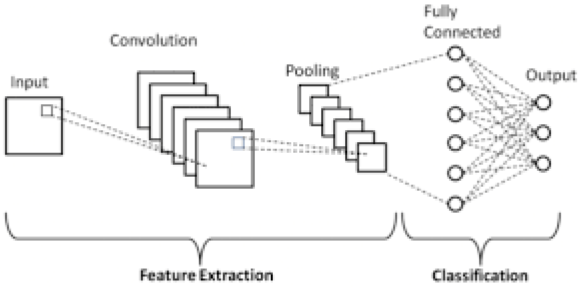
*Random Number Generator is not a real ML architecture.



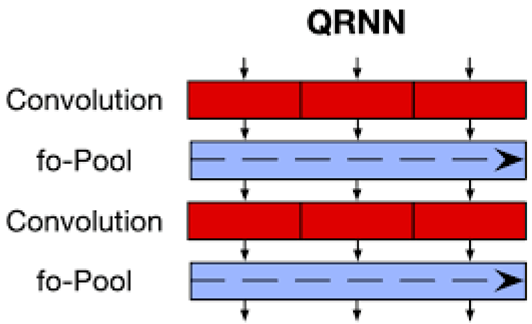
RNN



LSTM



CNN?

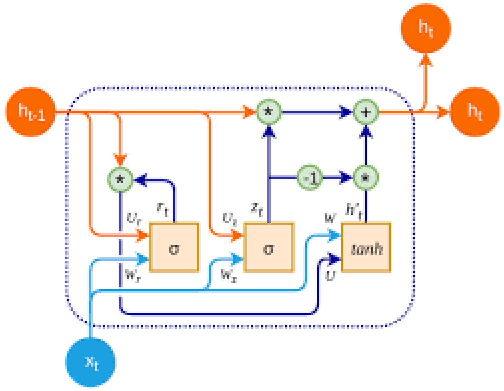


QRNN: CNN + RNN

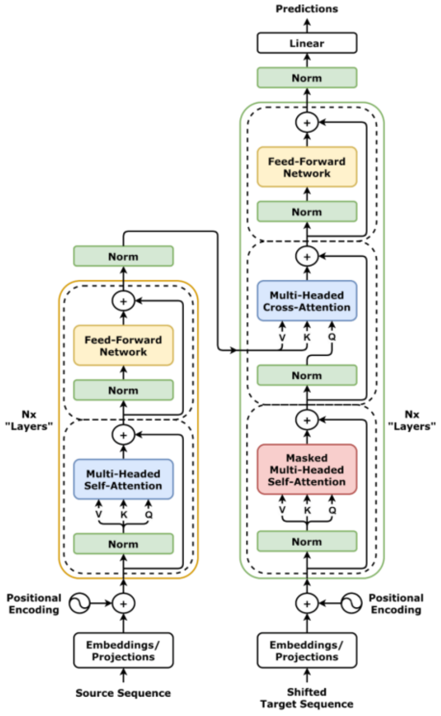
What LM Architecture?

What Machine Learning Architecture?

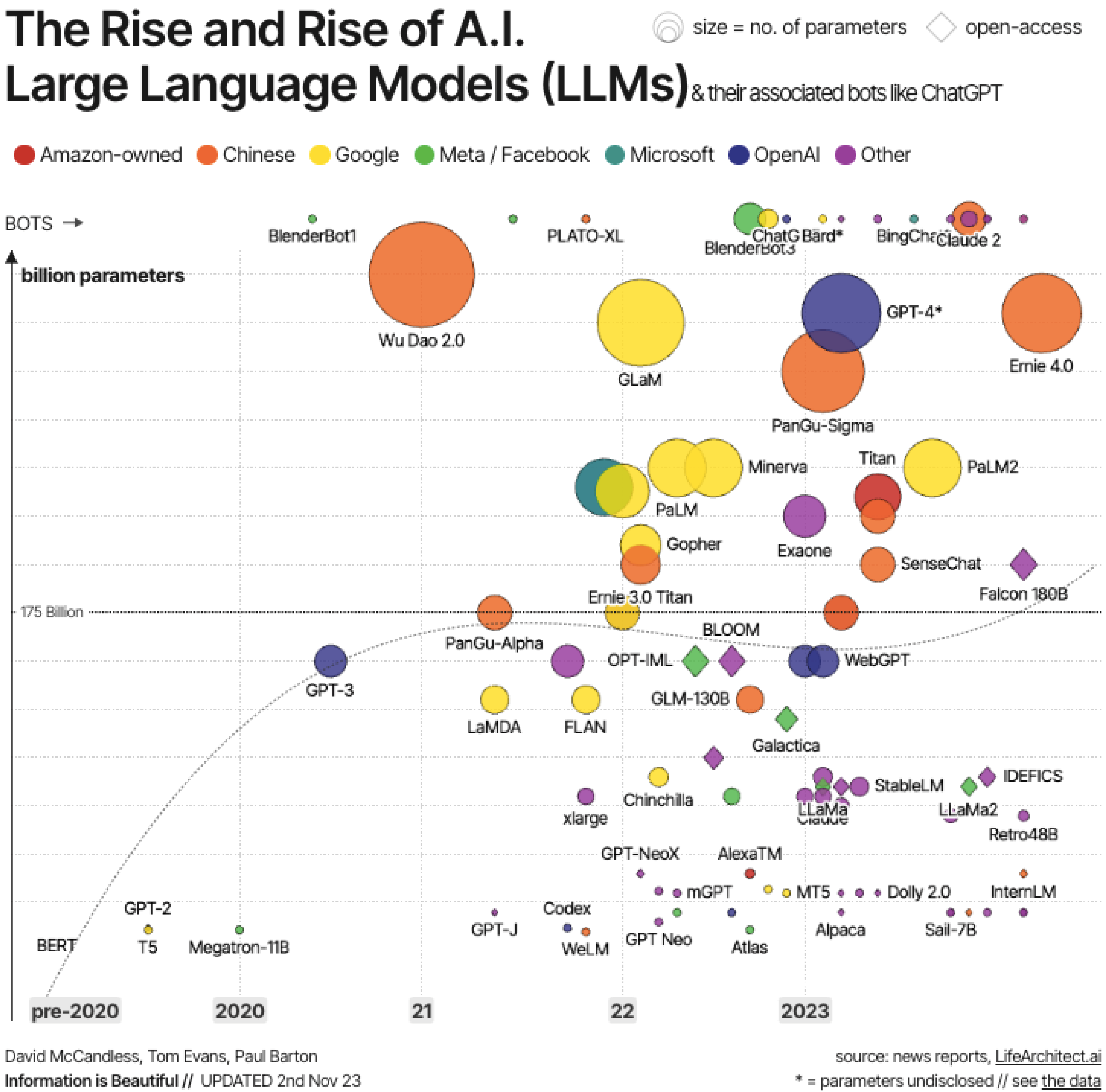
Symbolic vs. Statistical



GRU



👑 Transformers

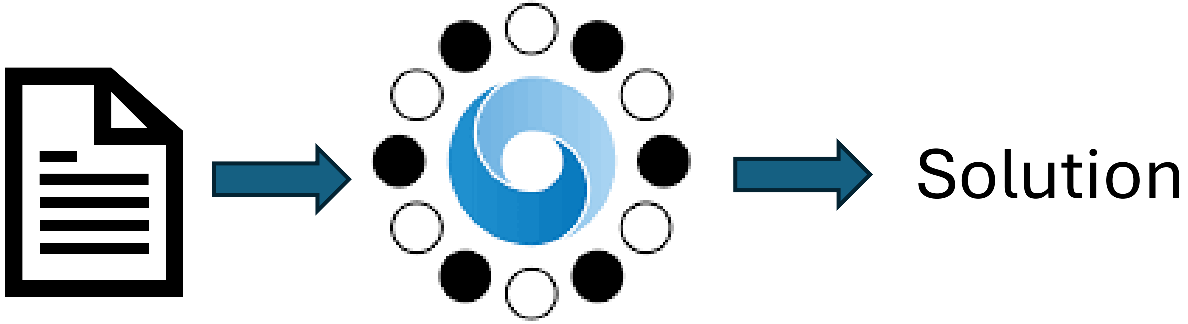


Bigger = Better?

at LM
ecture?

AlphaGo

RL for natural language tasks?

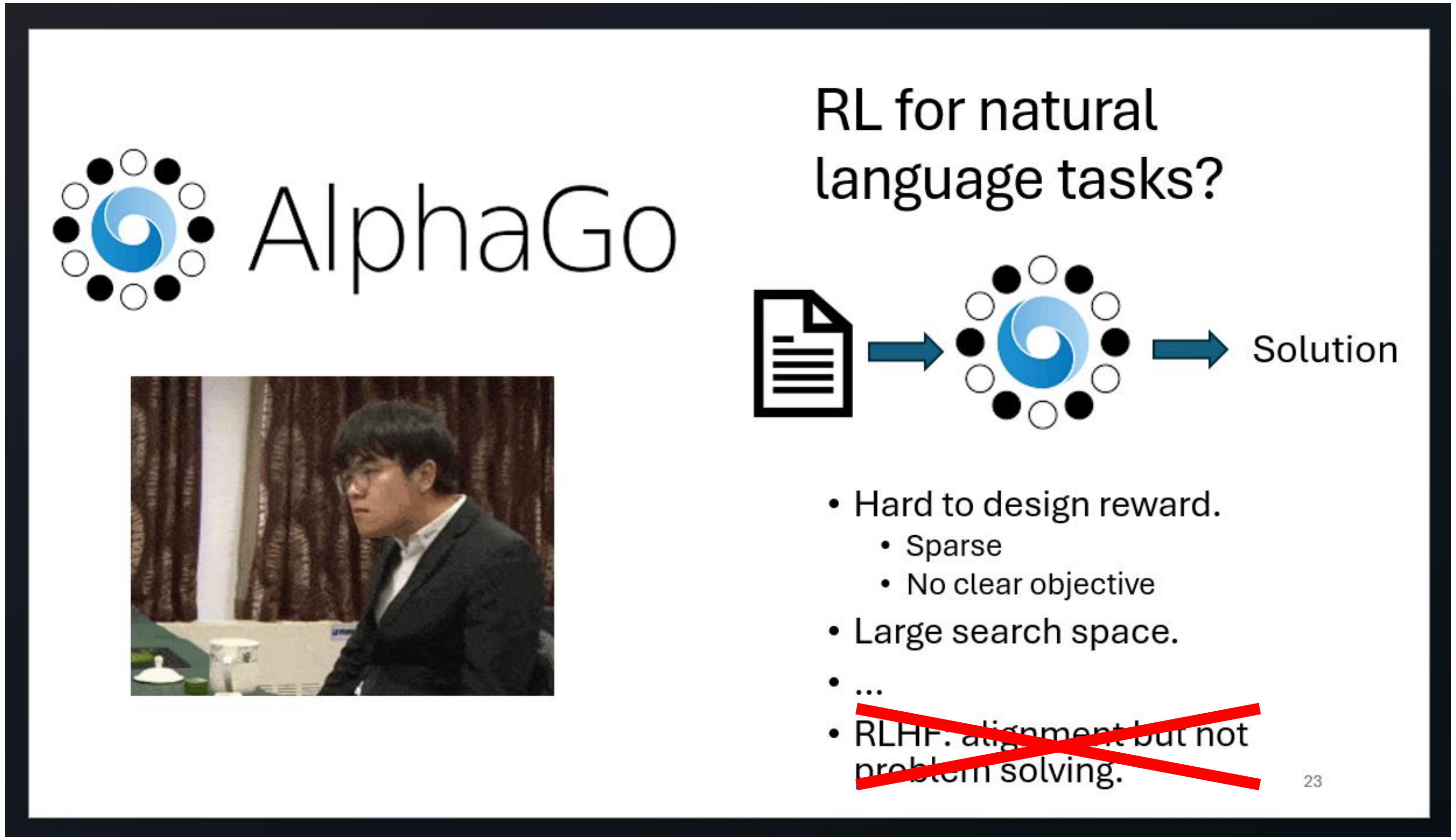


- Hard to design reward.
  - Sparse
  - No clear objective
- Large search space.
- …
- RLHF: alignment but not problem solving.

A Fast-Changing Field

- Fall 2024: RL has not yet worked.
- January 2025: DeepSeek released.

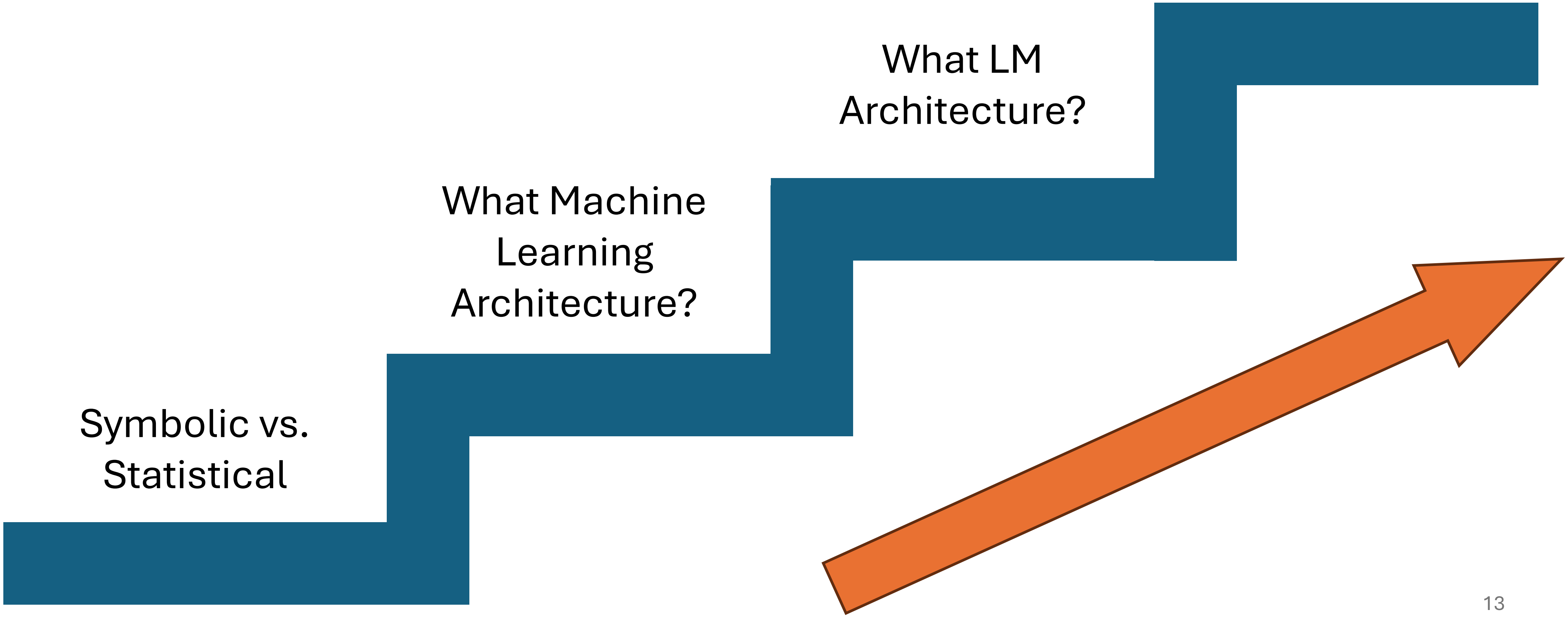
deepseek



What I've taught in Fall 2024.

What's Next?

More RL?
New architecture?
Multi-agent?
...



Do we understand Human Language Processing?

- We still don't know.
  - What is language.
  - What is a word.
  - What is a sentence.
  - Why human can speak language.
  - …

- Build better machine models of language from psycholinguistic inspirations.
- Not finding pseudo-psycholinguistic cues in these machine models.


May You
Live In
Interesting
Times
BIENNALE ARTE
2019
11.05—24.11
VENEZIA
GIARDINI/ARSENALE

Course Goals

- Learn the basic principles underlying **LLM Systems**.
- Two big topics:
  - Large Language Model.
  - Large Language Model Systems.
- After taking the course, you can:
  - Use LLM *critically*.
  - Build systems using LLMs for various natural language processing (NLP) tasks.
  - Understand how LLMs are implemented from scratch.
  - Gain insight into **open research problems** in NLP.

General Information

- All lectures and practice classes will be in person

Lectures:	Tuesdays	13:30 – 15:10, S306 / 051
Practice Class:	Thursdays	16:15 – 17:55, S103 / 221

- All slides, handouts etc. can be found on:
  - The course website: https://frankniujc.github.io/teaching/intro2llm/,
  - and Moodle.
- Discussion: moodle.

Practice Classes

- In the **practice classes**, you will work on programming exercises
  - First class: **this Thursday!**
  - Programming language is Python.
  - First practice session will include a brief introduction to Python.
  - This will give you some practical experience in NLP.
  - Practice class topics are **relevant for the exam**! (including Python)
  - Exact problems and very similar problems are in the exam.
- Materials will be announced earlier
  - Please review them before hand.
- During the classes: implement code or work on question together.

More Topics?
Feedback?
Anonymous Feedback?

Online Survey:

https://docs.google.com/forms/d/e/1FAIpQLScdlRRjGYJAriImTrjVI1U3wtqp2QQHEvK4eYVozIaP3NSjCA/viewform?usp=dialog

2 bonus assignment points for people finish before the holiday break: 19.12.2025.

Assignments & Evaluation

- Your final score is determined by your final exam grade + a possible assignment bonus.

- There are **homework assignments** for an exam bonus.
- Assignments will be bi-weekly: 6 exercises in total.
- Each assignment is worth 20 points.
  - Content survey: 2 bonus points if done before the holiday break.
- If you get >= 75% of the points (>= 90 points), you get a bonus.
  - You can improve your grade by 0.3/0.4 IFF you pass the exam without bonus.

Final Exam

- Tue, 24. Feb. 2026, 15:00.
- More information when we are getting closer.

- Content: everything from lecture, practice class, assignments.
- **~40%** of the final exam will be exact questions, or slightly altered questions from your practice class problem set and the assignment.

“Will this be on the test?”

YES.

Census

- Which degree programme are you studying?
  - Computer Science?
  - Bachelor?
  - Master?
  - Other disciplines?

Census

- Who can speak English?
- Who can speak English + German?
- Who can speak English + German + another language?
- Arabic?
- Chinese?
- Polish?
- Czech?
- Hungarian?
- Latin?
- Sanskrit?
- Chichewa?
- Inuktitut?

UKP Lectures

	Winter Term	Summer Term
Introductory		Information Management
Application Oriented	Intro to LLM	Ethics in NLP
Method Oriented		Deep Learning for NLP

Syllabus

- CL/NLP Basics
- Statistical Word Representation (Word Embeddings)
- LM Foundations, Tokenization
- Transformer / BERT
- Natural Language Understanding (NLU)
- Reinforcement Learning for Post-training
- Prompting Techniques
- Interpretability
- Dense Retrieval & Re-Ranking
- Question Answering (QA) and Retrieval-Augmented Generation (RAG)
- Quantisation

Intro to LLM *vs* Deep Learning for NLP

Unavoidable overlaps:

- The Transformer architecture.

Things we will not cover in this course:

- DL motivations to the Transformer architecture.
- RNN, LSTM language models.
- Training dynamics, DL insight behind RLHF, IFT.

Simone Balloccu.
Senior independent
coffee enjoyer.

Intro2LLM:

+ some mechanics

DL4NLP:

Online Resources & AI Writing Assistance

- Do NOT post any assignments online.
- Do NOT use any code generated by any AI assistance:
  - ChatGPT, Copilot, etc.
- The work you submit **must be your own**.
- … And you can't use ChatGPT during the exam.
- Except when you are explicitly asked to do so.

- And, yes, we can tell.

LLM detection research groups

ChatGPT 5

What are the leading research groups working on LLM detection in Germany?

Thought for 1m 21s

Here are the Germany-based groups that are most visibly active on LLM/AI-generated text detection (methods, benchmarks, or human-vs-machine studies), with a pointer to concrete outputs:

- **TU Darmstadt — UKP Lab & ATHENE / Fraunhofer SIT (Darmstadt)**
  • UKP is hiring and building datasets/methods for "trustworthy AI-generated text detection," including fine-grained hybrid (human+LLM) detection. informatik.tu-da...
  • Fraunhofer SIT's text forensics team works on authorship verification and LLM-text detection; they've also competed in GenAI detection shared tasks and published adapter-fusion baselines. Fraunhofer SIT +2
- **CISPA Helmholtz Center for Information Security (Saarbrücken)**
  • Leads large studies on **human detecta**[…]**ity** of AI-generated media (incl. text) across countries and develops detection/attribution benchmarks such as **MGTBench**. They

s

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

Deep Learning, Neural Network, Machine Learning Basics

$$\mathbf{x} \cdot \mathbf{W}$$

Input

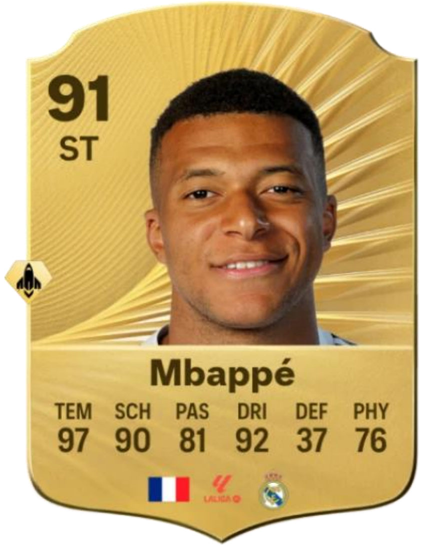
“Weight”

$$\mathbf{x} \cdot \mathbf{W}$$

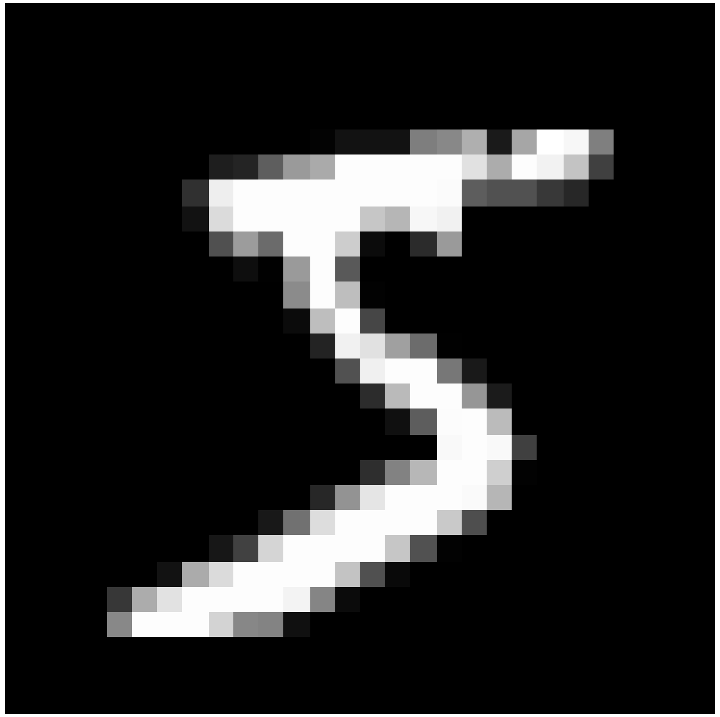
Examples

- Input: A student’s scores
  - Q1: 50%
  - Q2: 20%
  - Q3: 30%
- Weight: The Marking Scheme
  - Q1: 10 pts
  - Q2: 20 pts
  - Q3: 10 pts
- Final Score?

- Input:
  - TEM, SCH, PAS, DRI, DEF, PHY
- Weight:
  - … Something that EA has
- Final score:



Examples

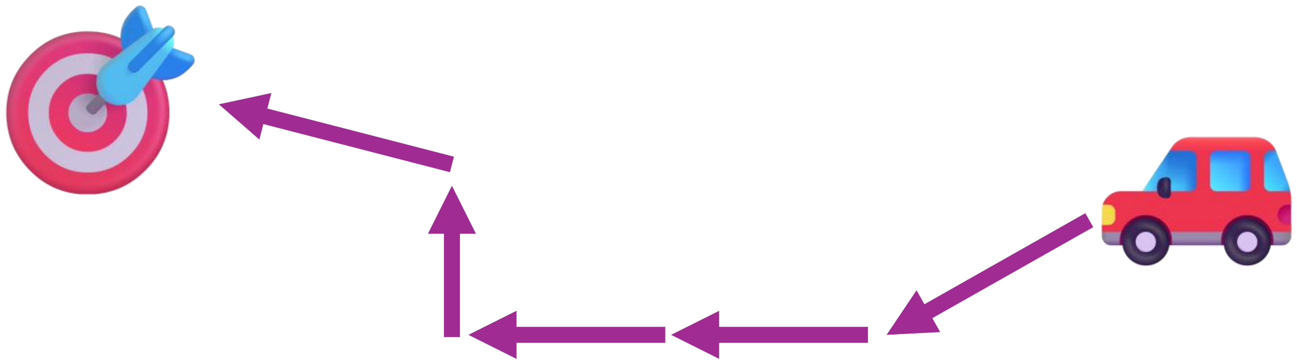


?

Gradient Descent – Review of Gradient

Some maths review

- Position: x_t
- Gradient of position: $v_t = \nabla f(x_t)$
    - Velocity.
- If we know the velocity of an object across several time steps.
- We can approximate the final position.



Gradient Descent for Opitmisation

- Input, weight…

$$\mathbf{x}, \ \mathbf{W}$$

- Define a loss function over the model’s output:

$$\mathcal{L}(\mathbf{x}\mathbf{W})$$

- This can be:
  - The larger the better
  - The smaller the better
  - The similar to a target the better
  - …

?

Climbing Down a Mountain with a Blindfold

Gradient Descent

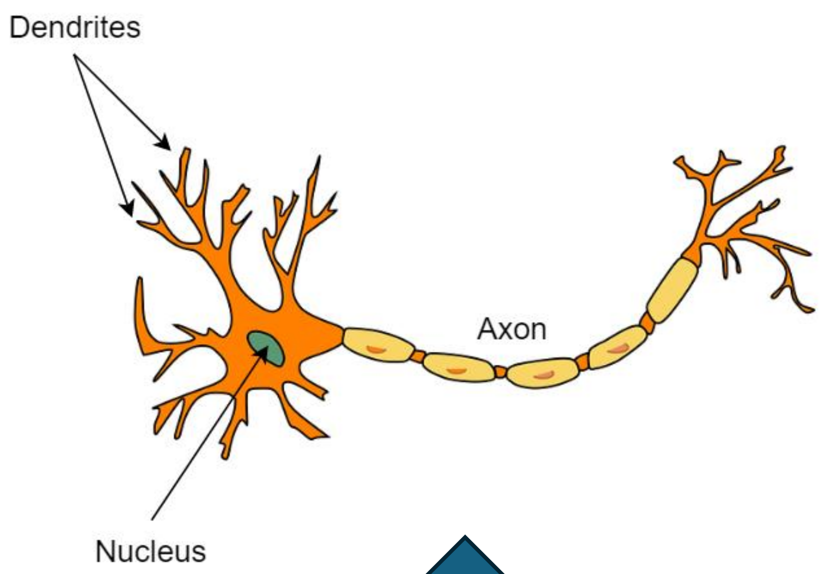
- Strategy:
  - Compute the error (loss function $\mathcal{L}(\mathbf{xW})$) at the output.
  - Determine the contribution of each parameter to the error by taking the differential of error w.r.t. the parameter. → Compute the gradient.

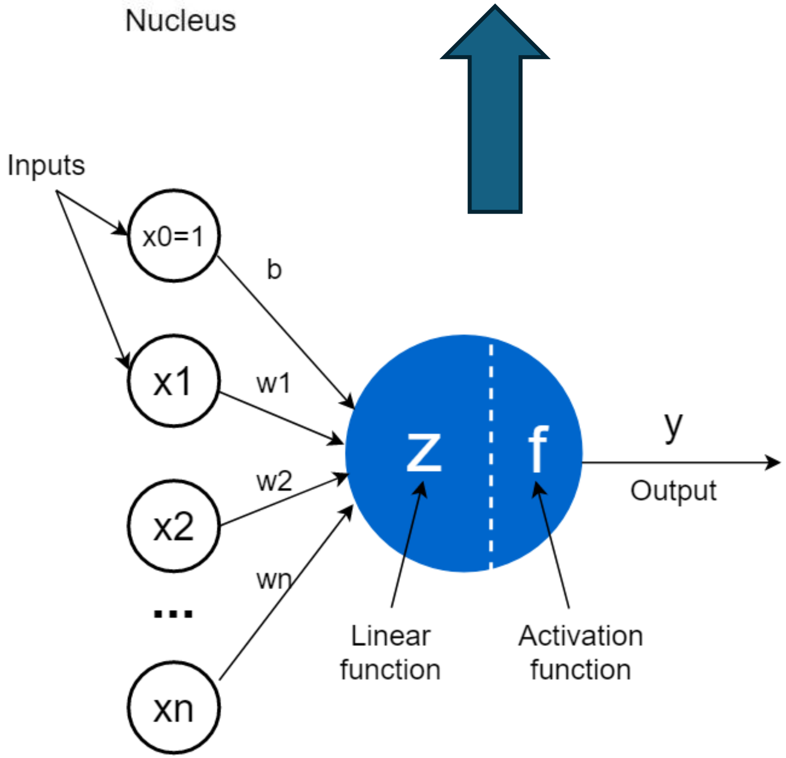
$$\mathbf{W} \leftarrow \mathbf{W} - \nabla_{\mathbf{W}} \mathcal{L}(\mathbf{xW})$$

- Update the parameter by the gradient.
- Mountain analogy:
  - Error of every param. combination: contour map.
  - Slope: gradient of error.
  - Blindly going down hill → you will eventually reach a lower place (local minimum of error).

Neural Network

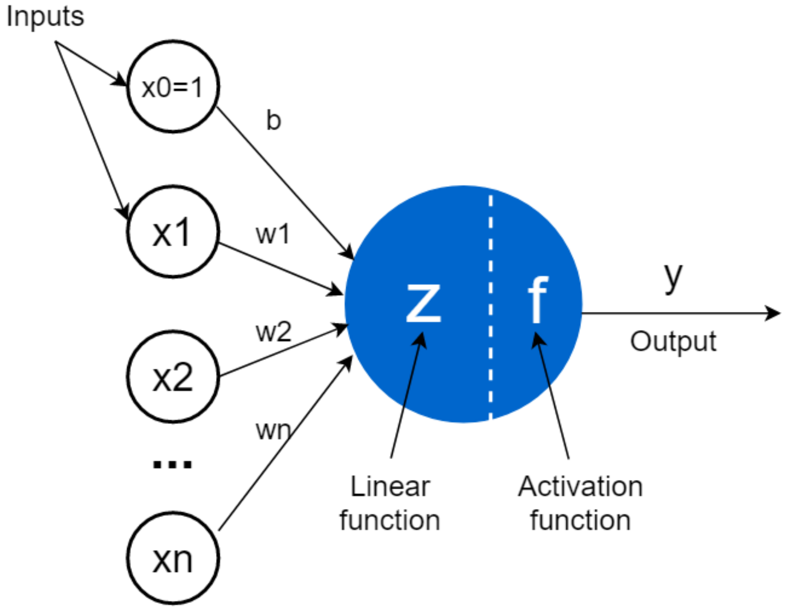
- More complicated models.
- Input can be:
  - Scalar number
  - Vector of Real numbers
  - Vector of Binary
- Outputs can be
  - Linear, single output (Linear)
  - Linear, multiple outputs (Linear)
  - Single output binary (Logistics)
  - Multi output binary (Logitics)
  - 1 of k Multinomial output (Softmax) (categorical)

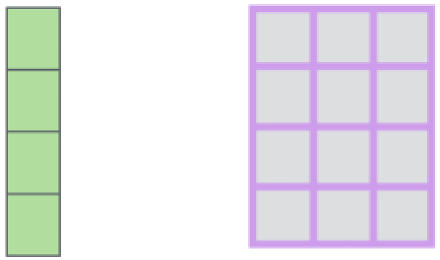




Neural Network

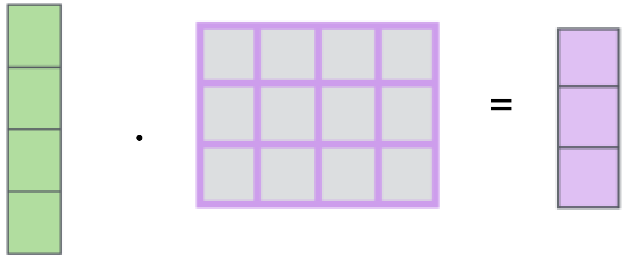
$$f\left(b+\sum_{i=1}^{n} x_i w_i\right) = f(\mathbf{x} \cdot \mathbf{W}^\top) + b$$

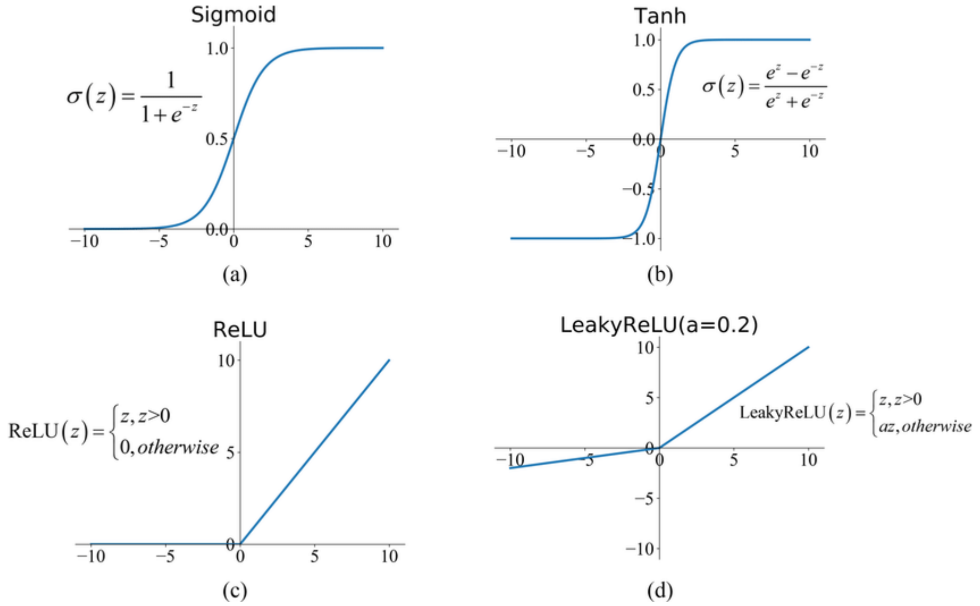




x
input vector

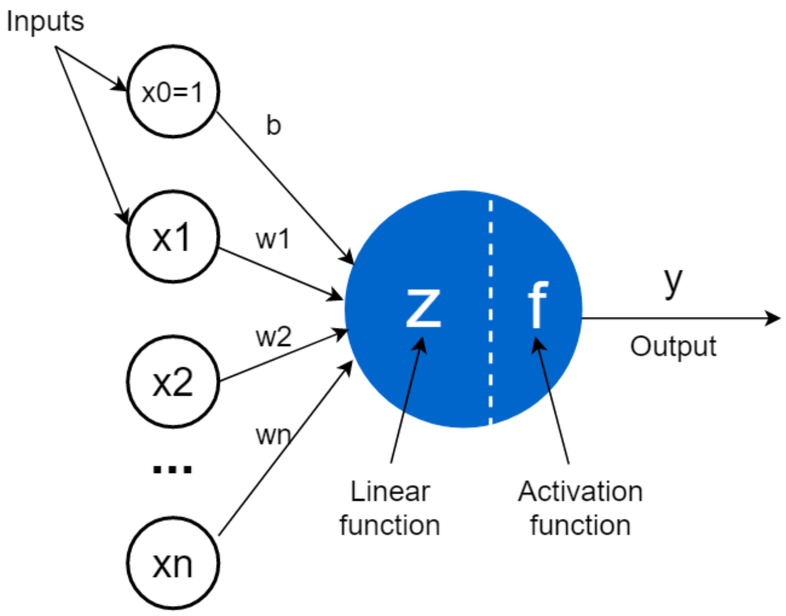
W
Model weights

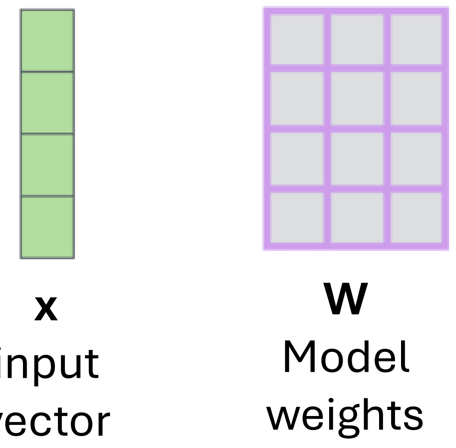


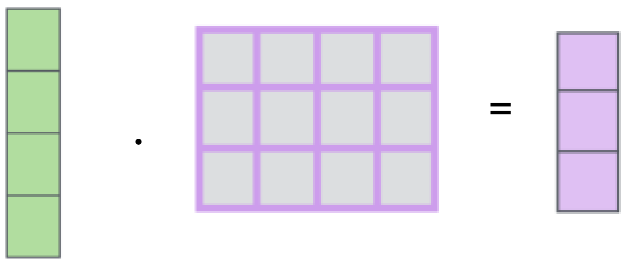


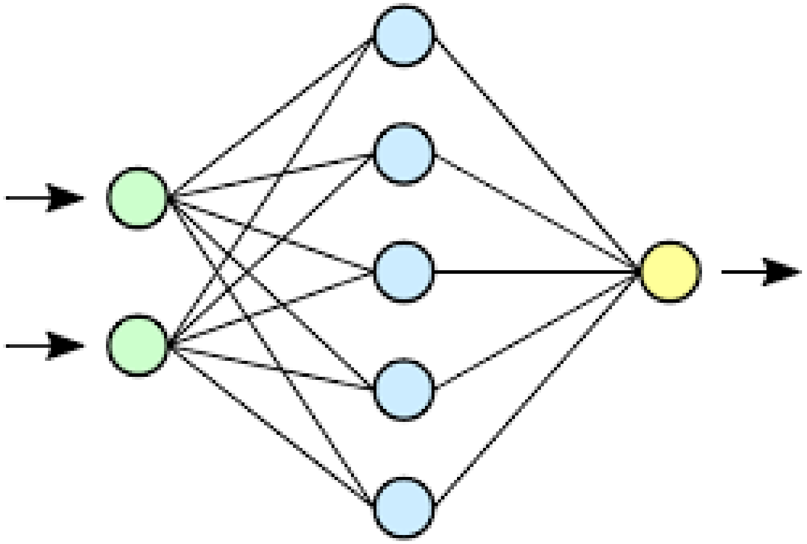
Neural Network

$$f(b+\sum_{i=1}^{n} x_i w_i) = f(\mathbf{x}\cdot\mathbf{W}^\top)+b$$

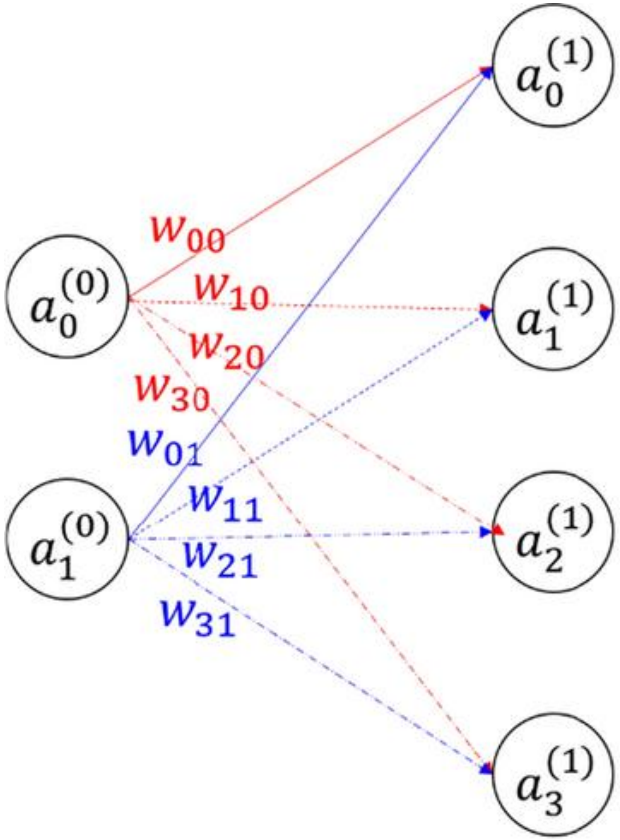








Neural Network



$$a_0^{(1)} = \sigma(w_{00}\, a_0^{(0)} + w_{01}\, a_1^{(0)} + b_0)$$

$$a_1^{(1)} = \sigma(w_{10}\, a_0^{(0)} + w_{11}\, a_1^{(0)} + b_1)$$

$$a_2^{(1)} = \sigma(w_{20}\, a_0^{(0)} + w_{21}\, a_1^{(0)} + b_2)$$

$$a_3^{(1)} = \sigma(w_{30}\, a_0^{(0)} + w_{31}\, a_1^{(0)} + b_3)$$

$$a_j^{(l)} = \sigma(\textstyle\sum_{i=1}^{N_{l-1}} w_{ji}\, a_i^{(l-1)} + b_j)$$

Evaluation

- Split your data into 3 splits:

Split	Purpose	Used During
Train	Fit model parameters (e.g. weights).	Training
Development (dev) / Validation	Tune hyperparameters (e.g. learning rate, architecture, early stopping).	Model selection
Test	Final, unbiased performance estimate.	After all training + tuning

- Reason - Overfitting:
  - The model learns patterns that fit the training data extremely well, but fail to generalise to unseen data.

Demo

- https://drive.google.com/file/d/1xGhRq36tx2BDxSt_yDJROwLv_gijhmKR/view?usp=sharing