Neural Networks for Data Science Applications

Master's Degree in Data Science

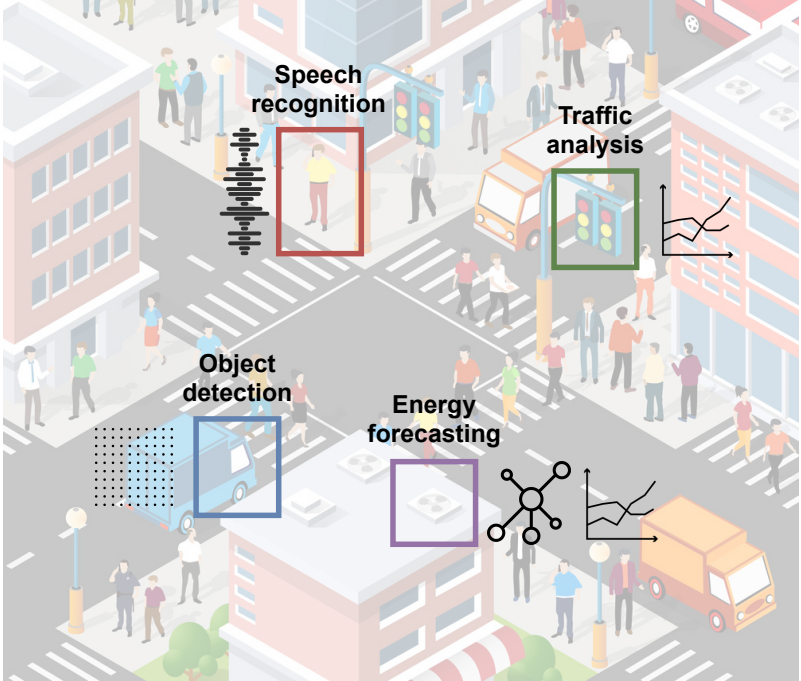
Lecture 1: Introduction

Lecturer: S. Scardapane

SAPIENZA
UNIVERSITÀ DI ROMA

Introduction

Neural networks are everywhere


Speech recognition
Traffic analysis
Object detection
Energy forecasting

In just a few years (roughly from 2012) **neural networks** have become an everyday technology:

- **Large language models** (LLMs) – e.g., ChatGPT, Gemini, Claude, ...
- **Speech transcription systems** – e.g., Whisper.
- **Reinforcement learning** – e.g., AlphaGo, robot foundation models, ...
- **Scientific discovery** – e.g., antibiotic discovery, theorem proving, ...

A model like ChatGPT generates a distribution over the **next piece of text** (token), hence we call it a **language model**. By using it repeatedly we can generate very long texts (**autoregressive generation**).

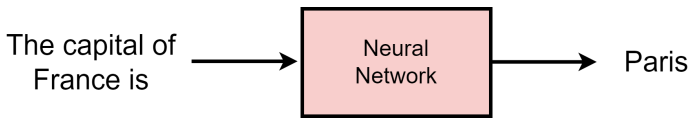
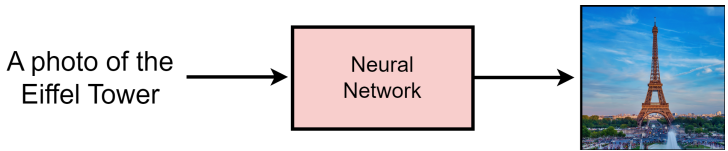
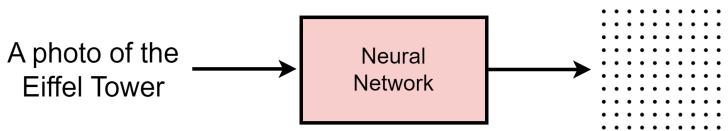


Figure 1: MCU Characters as 80s Wrestlers [Reddit]



The network needs to predict the RGB colours for *each* pixel in the image, maintaining spatial and semantic consistency.



Despite their differences, both examples share some characteristics:

- The data is **high-dimensional** (e.g., an image corresponds to millions of points), and with potentially infinite variety.
- Manually coding the procedure is impossible.
- It is relatively easy to collect **examples** of the desired behaviour (e.g., paired image-text pairs).
- They are both implemented using neural networks.

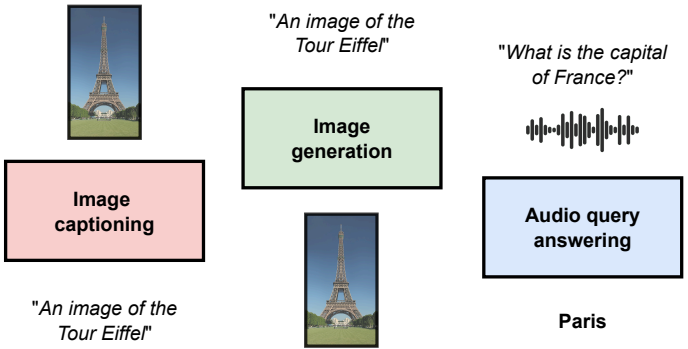


Figure 2: Most tasks can be categorized based on the desired input - output we need: image generation wants an image from a text, while the inverse (image captioning) is the problem of generating a caption from an image. Fascinatingly, the design of the models follow similar specifications in all cases.

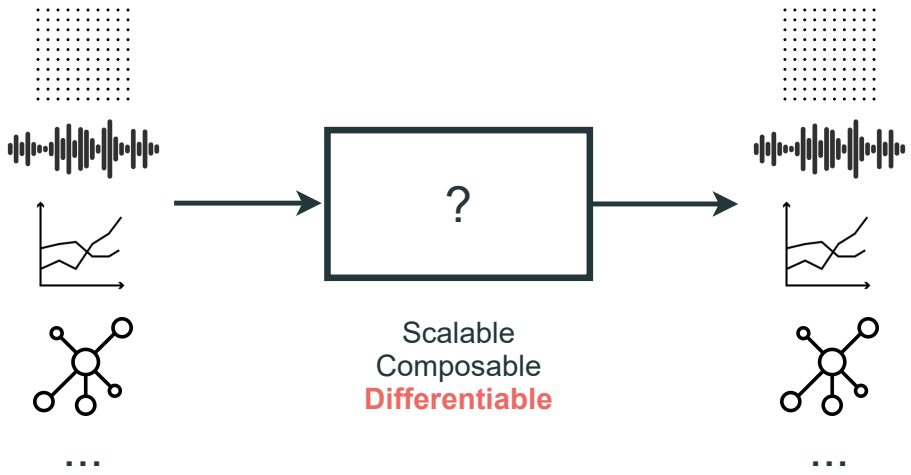
Listing all notable applications of neural networks is almost impossible: think of a complex problem, and someone has probably developed a state-of-the-art model for it, ranging from **neural translation** to **protein folding**, **videogame playing**, **neural rendering**, **physics simulations**, ...

Amazingly, all this is powered by a very small set of layers and organizing principles (e.g., differentiability, invariances and equivariances, sparsity, locality). **Data**, **computing**, and **software** are keys.

Hint: browse `https://paperswithcode.com/sota` for a few examples.

Introduction

Some definitions


?
Scalable
Composable
Differentiable

(Deep) neural networks are **composable**, **differentiable** functions that can be **optimized end-to-end** numerically.

- All these inputs/outputs can be represented as **tensors**, i.e., large n-dimensional arrays of numbers.
- Neural networks are composed of multiple blocks (**layers**), each of which performs a simple manipulation on these tensors.
- The operation of a layer may involve another tensor, whose values can be chosen freely (e.g., a matrix multiplication). These are called **parameters** of the layer.
- All parameters can be **optimized** numerically (**training**) by maximizing the performance of the network on a set of examples (**dataset**).

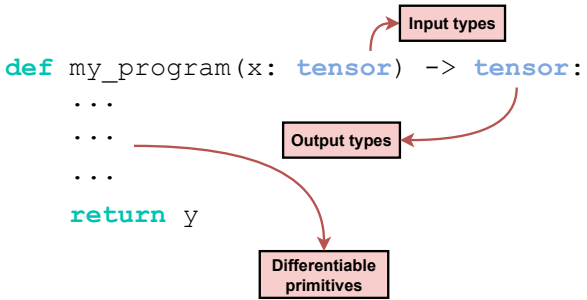


Figure 3: Neural networks are sequences of differentiable primitives which operate on structured arrays (tensors): each primitive can be categorized based on its input/output signature, which in turn defines the rules for composing them.

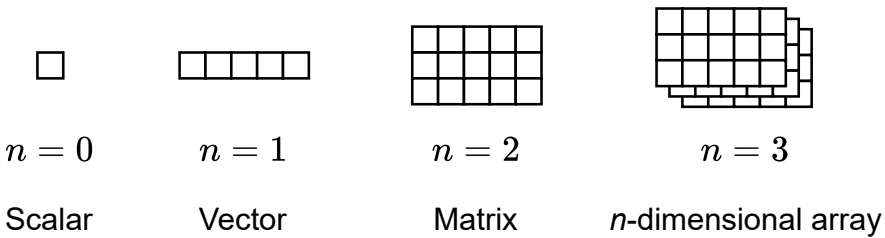
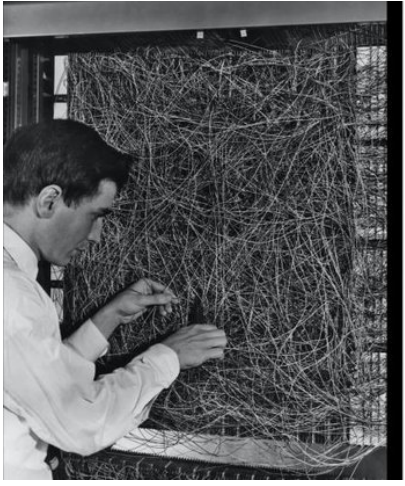


Figure 4: Fundamental data types: scalars, vectors, matrices, and generic n-dimensional arrays. We use the name tensors to refer to them. *n* is called the rank of the tensor.

Introduction

A bit of history



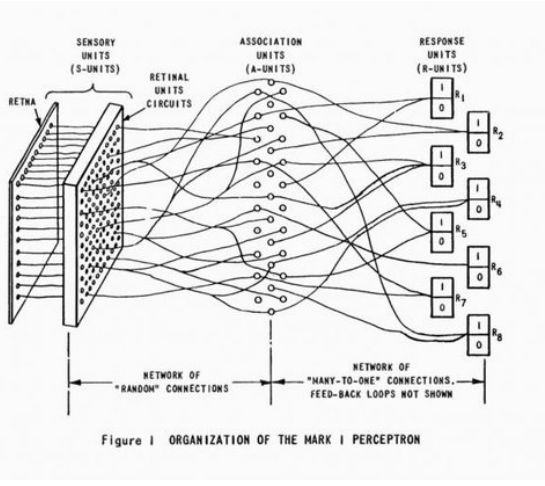
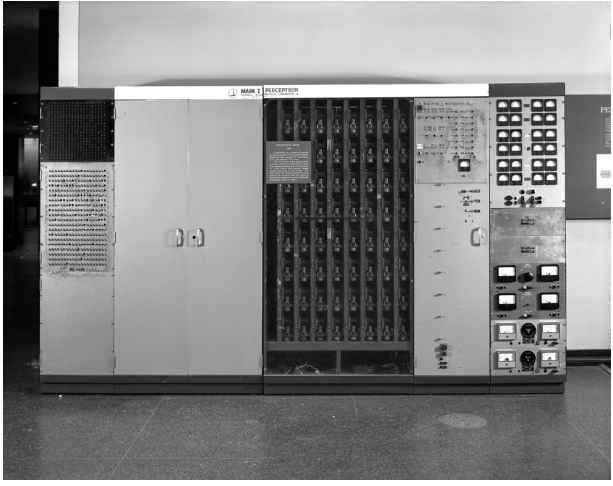


Figure 1 ORGANIZATION OF THE MARK I PERCEPTRON

Perceptron (1950s)



The New York Times

NEW NAVY DEVICE LEARNS BY DOING; Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

July 8, 1958

Frank Rosenblatt and the perceptron were poster childs of a scientific movement called **cybernetics**, sperheaded by the eclectic mathematician Norbert Wiener, a discipline that studied control problems with a strong focus on negative feedback, self-organization, and reinforcement.

Ironically, the term *artificial intelligence* was coined in the Dartmouth Workshop of 1956 in opposition to this trend, with a focus on symbolic systems, reasoning, deductivity, and eventually **expert systems**.

In modern terminology, a perceptron is more or less equivalent to a neural network with a single layer. As such, it was too limited to handle what it was advertised for, as were the current computing power and data availability.

Attacks from the AI field (e.g., the Perceptron book), the end of funding, and multiple personal conflicts (including Wiener's itself) led to a quasi-disappearence of neural network's research for many years.

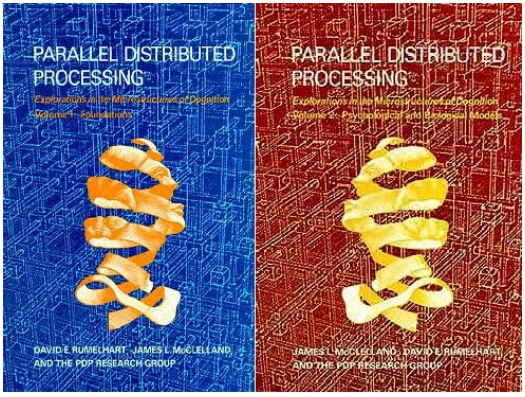


Figure 5: The PDP group was instrumental in revitalizing neural networks in the 80s, including the popularization of backpropagation, a principled way to train NNs with many layers. The group's interests were much larger and spanned psychology, development processes, and neurology.

Though the appeal of PDP models is definitely enhanced by their physiological plausibility and neural inspiration, these are not the primary bases for their appeal to us. We are, after all, cognitive scientists and PDP models appeal to us for psychological and computational reasons.

— McClelland, Rumelhart, Hinton (1986)

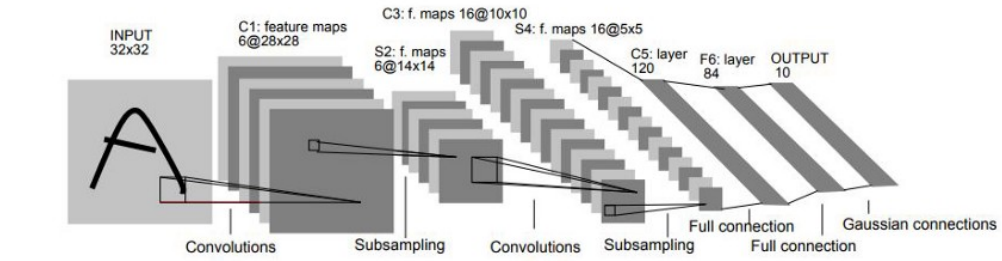


Figure 6: In 1998, the team of Y. LeCun at Bell Labs already have a working neural network for optical character recognition (5-7 layers), termed **LeNet-5**, fundamentally identical to a modern NN in its design and training. However, data and computing power were still not enough, and a new winter came.

LeCun, Y., et al., 1998. **Gradient-based learning applied to document recognition**. *Proceedings of the IEEE*, 86(11), pp.2278-2324.

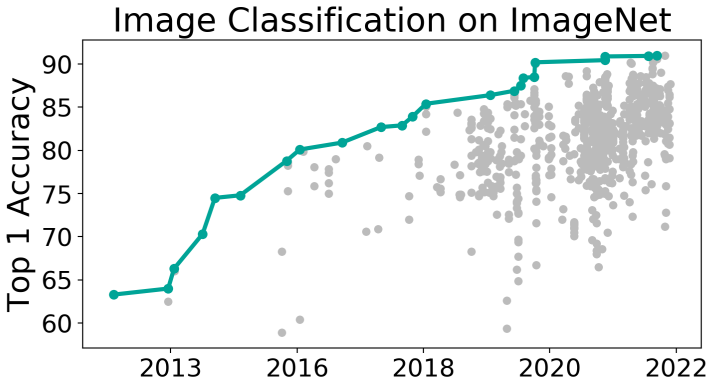


Figure 7: Evolution of accuracy on the **ImageNet Large Scale Visual Recognition Challenge** (ILSVRC). The 2012 victory by AlexNet (fundamentally, a slightly larger LeNet) was a key element in restarting again a major interest in NNs.

From 2012, the size of the datasets and the size of the neural networks themselves have kept increasing at an exponential rate, and NNs have slowly taken over multiple fields, from audio processing to natural language processing, graph data, and computer vision.

Remarkably, outside of scale the underlying principles have stayed consistent, and todays' ChatGPT is much closer to LeNet than you would imagine.

Scaling laws have been developed to predict the evolution of accuracy based on scale: for most tasks, increasing data, compute, and parameters leads to a predictable improvement in performance.

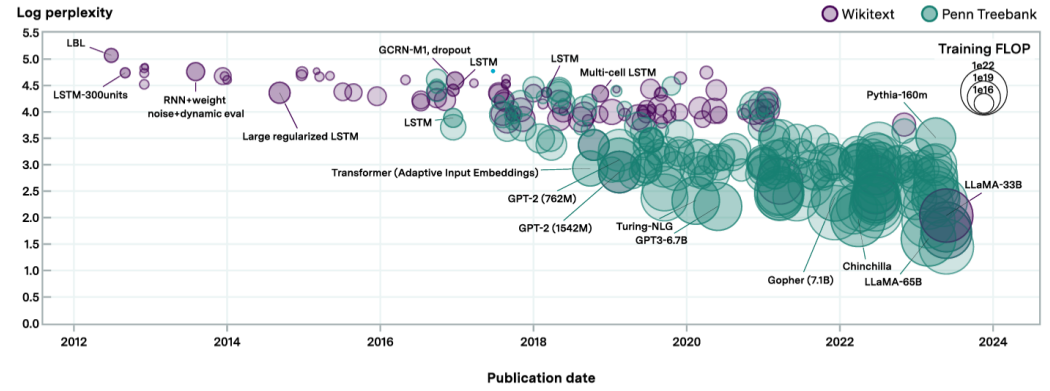


Figure 8: Performance in language modeling has steadily improved, while the size of the models has constantly increased. The increase in performance is also matched by equivalent data scaling, with variations in modelling becoming asymptotically less significant.

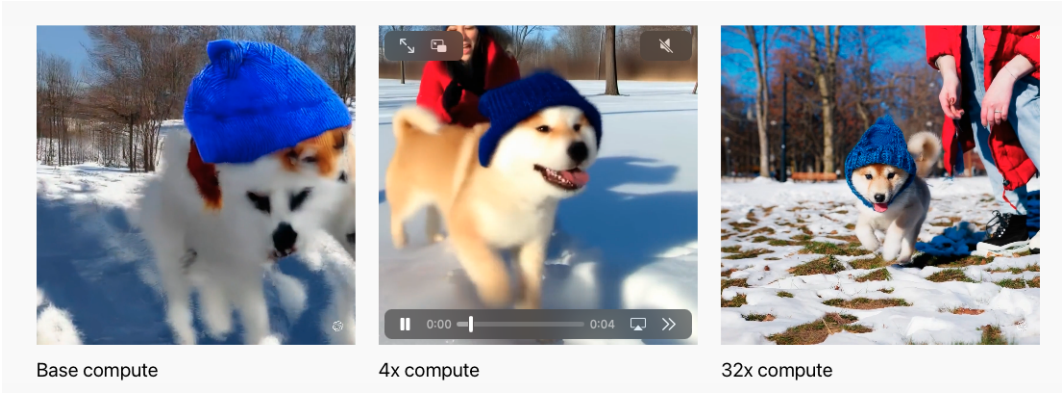


Figure 9: https://openai.com/index/video-generation-models-as-world-simulators/

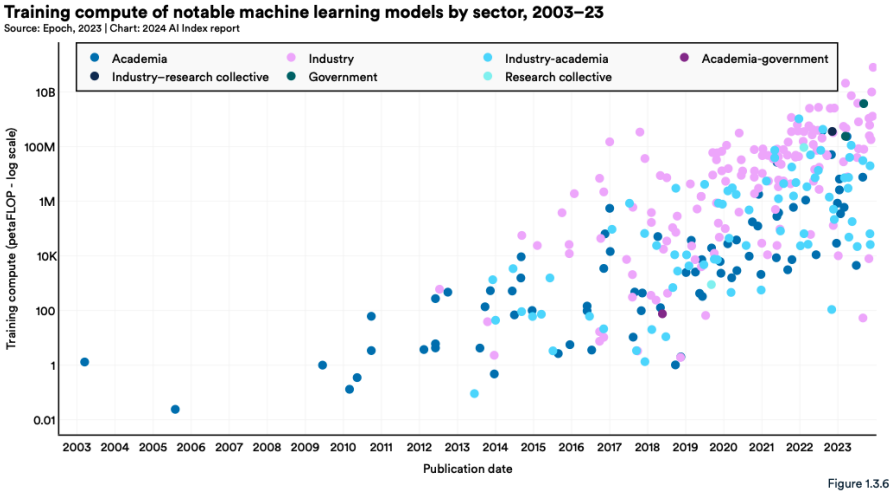


Figure 10: Reproduced from the AI Index Report 2024 (Stanford).

Scaling has led to emergence of extremely large, *generalist* models that can be used for a wide variety of tasks (**foundation models**). Due to their cost, these models are becoming the domain of only a handful of players in the world.

An entire field has grown up around LLMs and similar models: prompting, efficient inference, agents, ...Many of these emerging disciplines take a pre-trained model as a complete black-box, whose internal components are of little interest.

This course is about the key components and blocks of which most neural networks are composed – we will only touch briefly upon scaling to the sizes known today. We will also see how their design informs the design of existing software libraries, notably JAX.

Still, knowing what's inside LLMs is fundamental for a huge number of tasks, from **instruction tuning** to **merging**, **quantization**, or for developing new models and techniques (e.g., for science).

- Read **Chapter 1** from the book and **Appendix A** for probability theory.