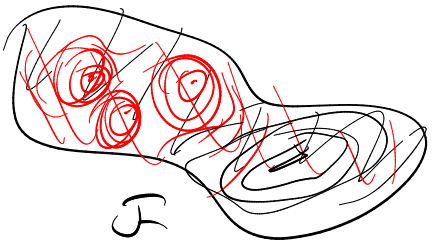
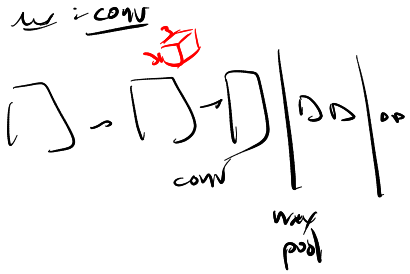
Architectures

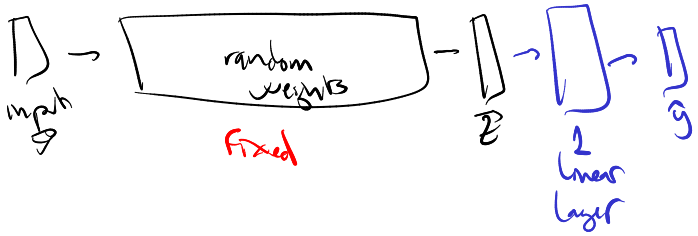
I) Architectures as priors on function space

not just about expressivity, but about probability if random weights





"Extreme ML":



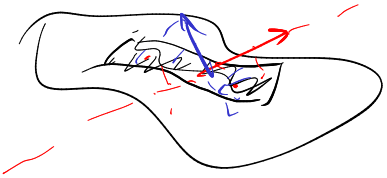
better accuracy (or ≃) than with training

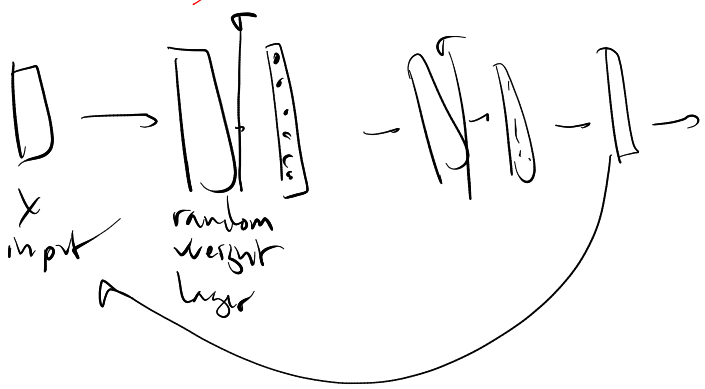
architecture bias

Random projections

high-dim space (embedding)

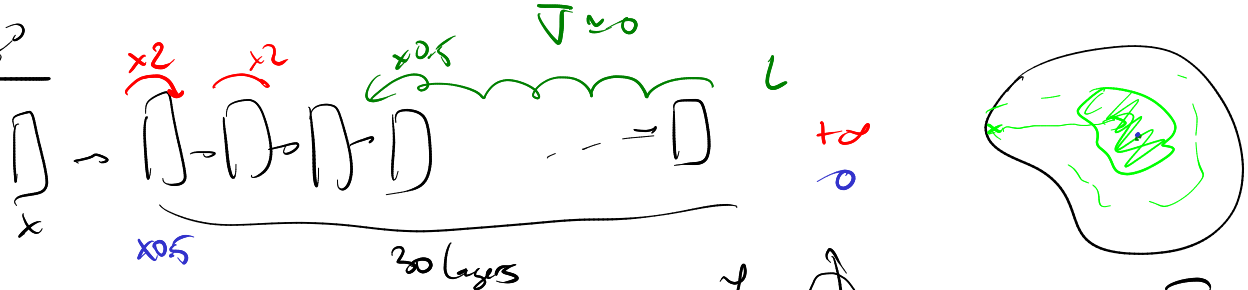
but intrinsic dim : not that much





eg: keep all information if #projections ≥ rank(manifold)

Initialization

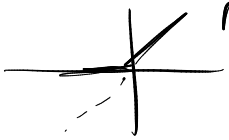


$\times 2^{30}$

$\times 2^{-30}$

$w_i \sim$

eg: Xavier Glorot init° Kaiming He



ReLU ⇒ /2 variance ⇒ $\times\sqrt{2}$

Linear layer: weights w_i : $\bar{w}=0$, $\sigma_w = \frac{1}{\sqrt{d}}$ ← input dim

inputs : x_i

$\vec{w}\cdot\vec{x} = \sum_i w_i x_i$

$\mathbb{E}_w[\dots] \Rightarrow$

var: $\mathbb{E}_x \mathbb{E}_w \left(\sum_i w_i x_i\right)^2 = \sum_i \left[\mathbb{E}_x x_i^2\right]\left[\mathbb{E}_w w_i^2\right]$

$= d \times 1 \times \frac{1}{d}$

$= 1$

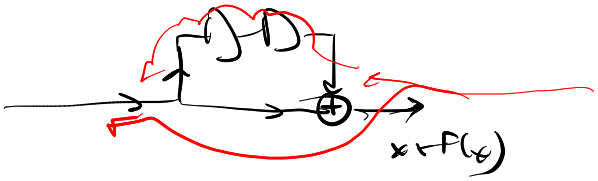
Pick architectures easy to train

VGG

vs ResNets

prior: $F \sim 0$

$g \sim Id + 0$



output = $\sum_{\text{blocks}} F_{\text{block}}(x)$

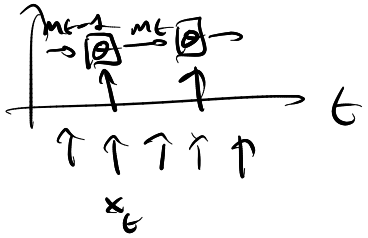
$\frac{\partial L}{\partial \text{output}}$

Zoo

- Fully-connected layers ("MLP") multi-layer perceptron
- conv nets : 1D, 2D, 3D ... geometry : structured data ⟹ 2D
- recurrent networks : LSTM, GRU

  $\vec{x}$ = temporal signal

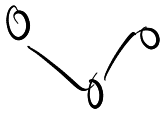
  1 input



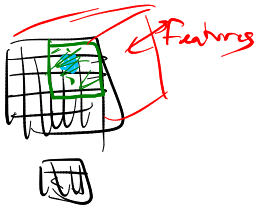
- attention block
- graph-NN

Graph-NN

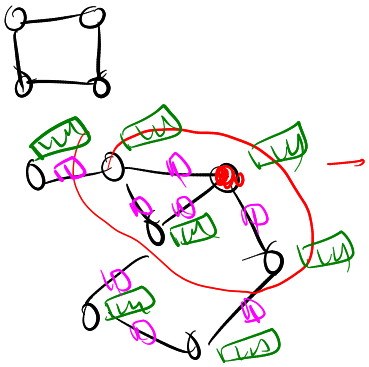
1 input = 1 graph



Conv:

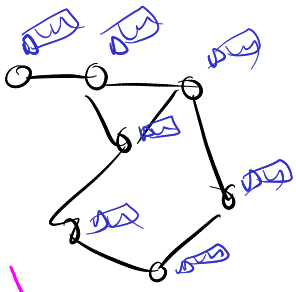


$\sum_{p \in P_0} w_p x_p$

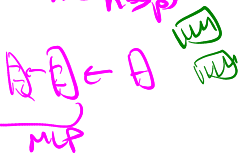


average $\propto \frac{1}{N}$

$w_c x_p + \sum_{n \in \text{neighbors}} w_n x_n$



$w_n(e_{n \to p})$



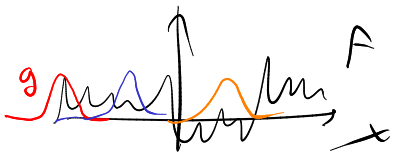
GAT

Graph Attention Network

↳ attention within neighborhood

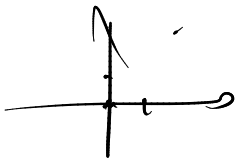
Conv:

$$f * g\,(x) = \int_{x'} f(x')\, g(x - x')\, dx'$$



Fourier ↓

$$\widehat{f * g} = \hat{f} \cdot \hat{g}$$



image

$\cdot \sin(\omega x)$

$\Delta \rightarrow$

- Graph Laplacian

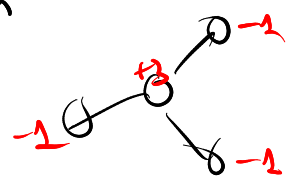
↓

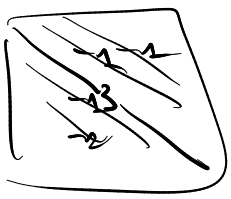
eigenmodes

↓

coordinate system

each node $\hookrightarrow (i) \uparrow h$ (of h eigenmodes)

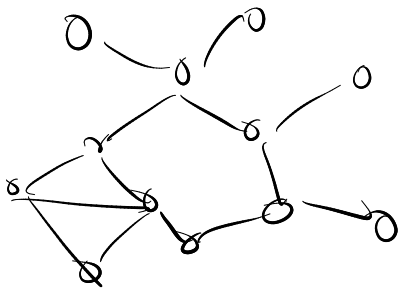




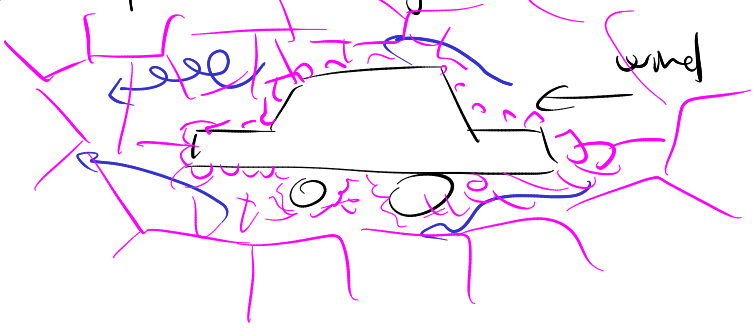
good to describe shapes (clouds of points)

3D laser points on a surface

Other graphs: . social networks:



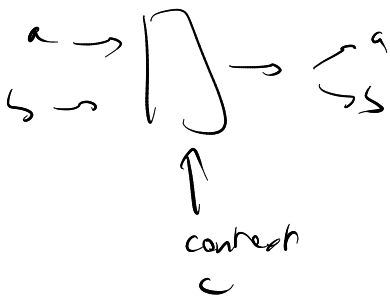
- mesh: computational fluid dynamics



⚠ dealing with resolution: graph pooling / unpooling
graph U-net
etc
ResNet structure ~

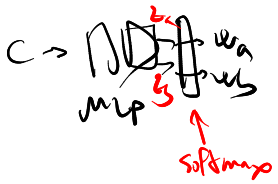
Transformers

attention mechanism:



$w_a a + w_b b$

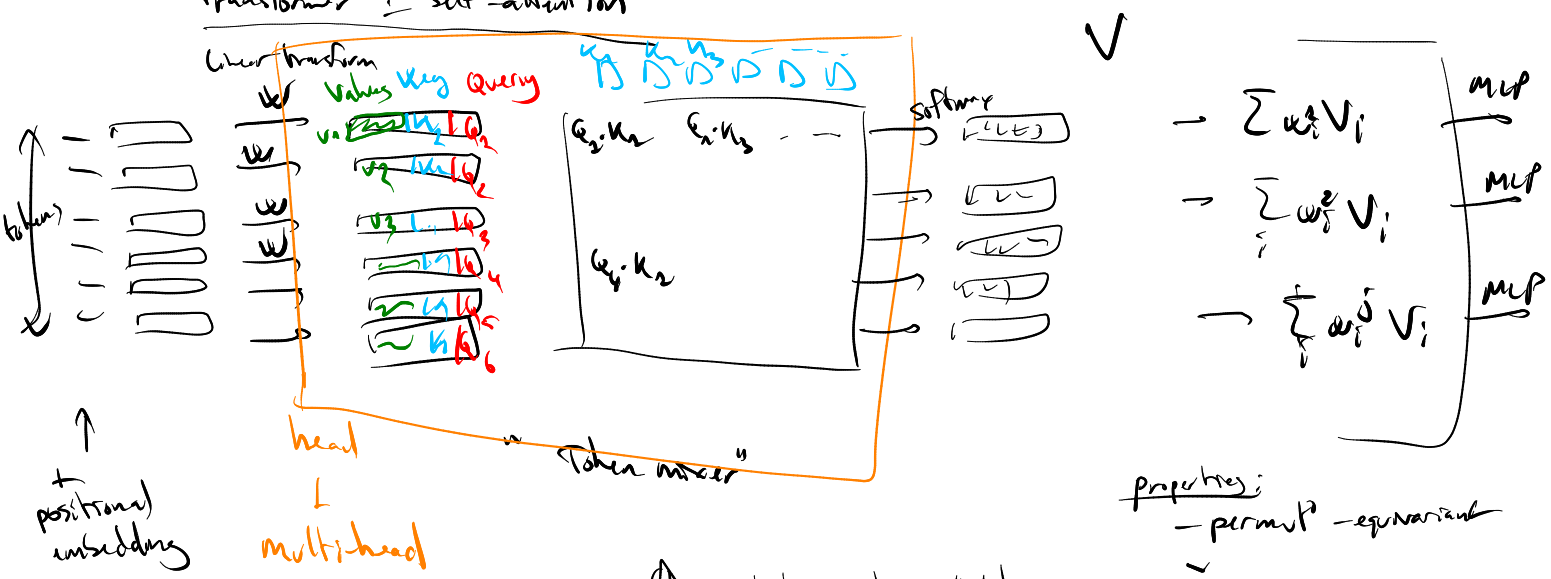
1 0
0 1



$w_a(c)\, a + w_b(c)\, b$

≥ 0 $\quad \Sigma = 1$

"Transformer": self-attention



+ positional embedding

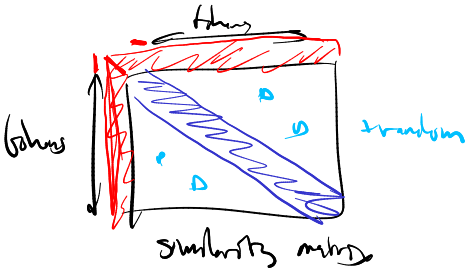
multi-head

properties:
- permut - equivariant

⚠ quadratic cost in # tokens

Variations

- Big Bird



2020

- kernelize: $\text{softmax}(Q^T K)V$

$\simeq \text{sim}$

$k(\text{token}, \text{token}') = \phi(\text{token}) \cdot \phi(\text{token}')$ kernel trick

$\Rightarrow$ linear !

- reformer : hashes
  - discretize $\mathbb{R}^d$ into bins
    - compute similarity only within the bin
- Linformer : low rank approx
- set transformer :

S4 models → Mamba

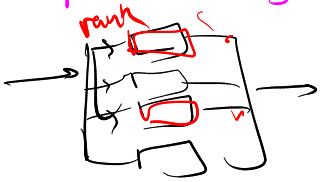
Neural Scaling Laws

- dataset size
- network size
- compute power

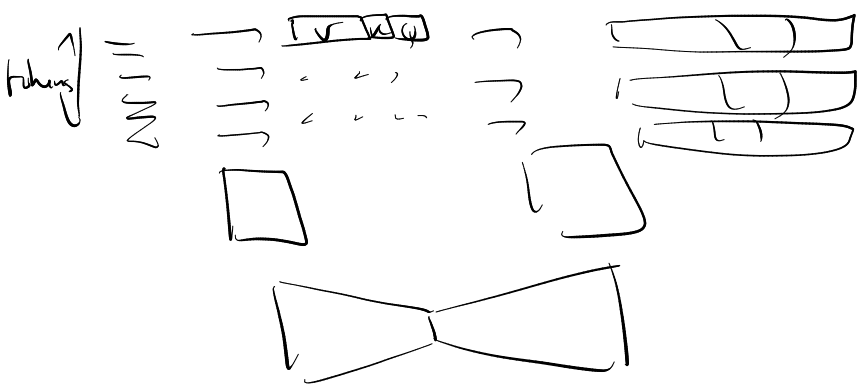
HPC High Performing Computing

- Mixtral & DeepSeek

"Mixture of Experts" ≡ many MLPs in //



- DeepSeek :



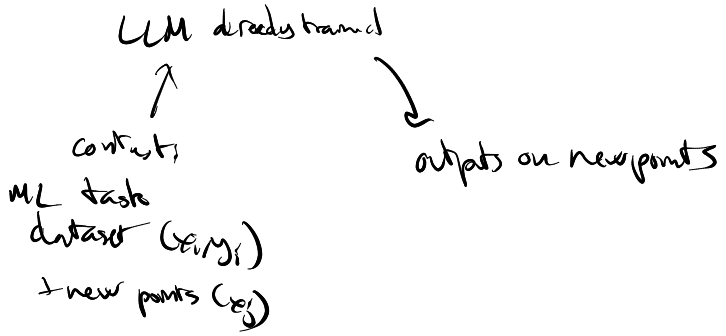
Low precision

FP8

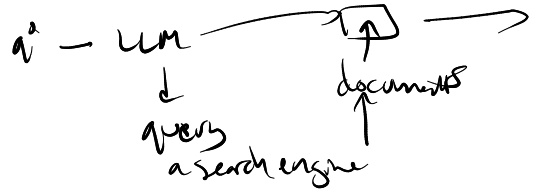
FP32

On top of LLMs

- prompting : alters significantly the output
- "In-Context Learning":



- RAG Retrieval-Augmented Gen.



- Agentic AI —

Computer Vision Transformers ViT

context

ViT : +10 %

ResNets ⇝ ViT

recipe to train ≠

- +1%
- +2%
- -0.1%
- +3%

+ MY.