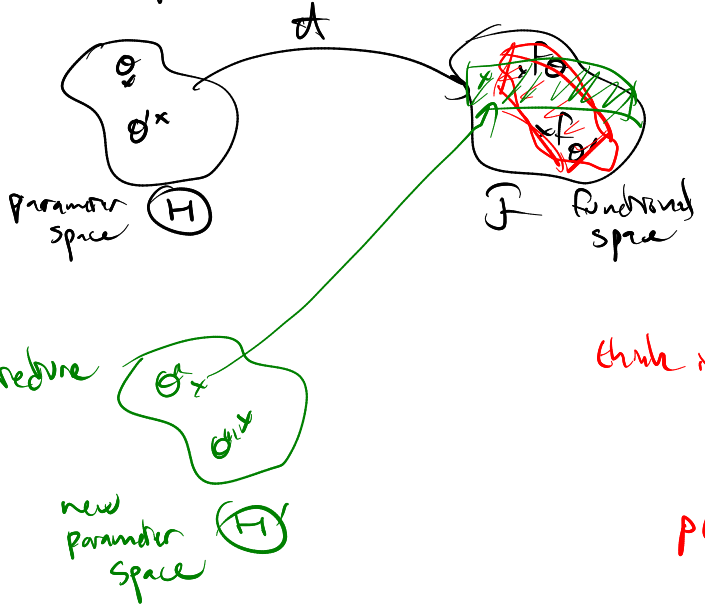
Chapter 3 : Architectures

I Architectures are priors on function space

Change of paradigm :

- classical ML : design features by hand
- deep ML : meta-design of features → design the architecture

An architecture = parametrized space of functions



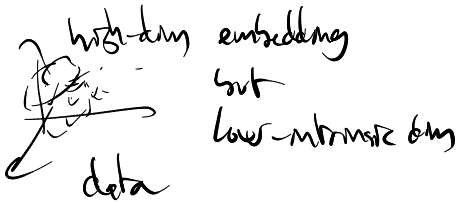
think in terms of probability

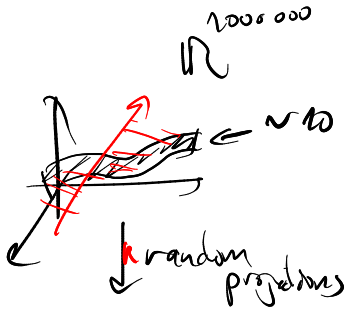
$f(M(\alpha,\mu))$

$p(f_\theta = f^* \mid \text{random } \theta \sim M(\alpha,\mu))$

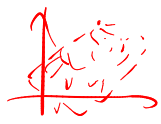
Architecture bias

→ random projections :



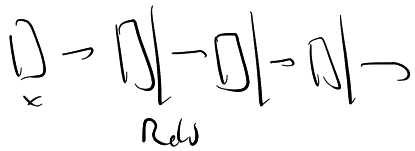


all information is there



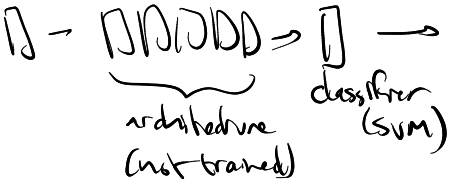
lower dim

→ with NN : random non-linear "projections"



In some cases : most of the performance is due to the architecture (not to the training)

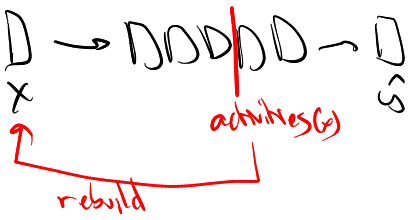
"extreme ML"



~ same perf

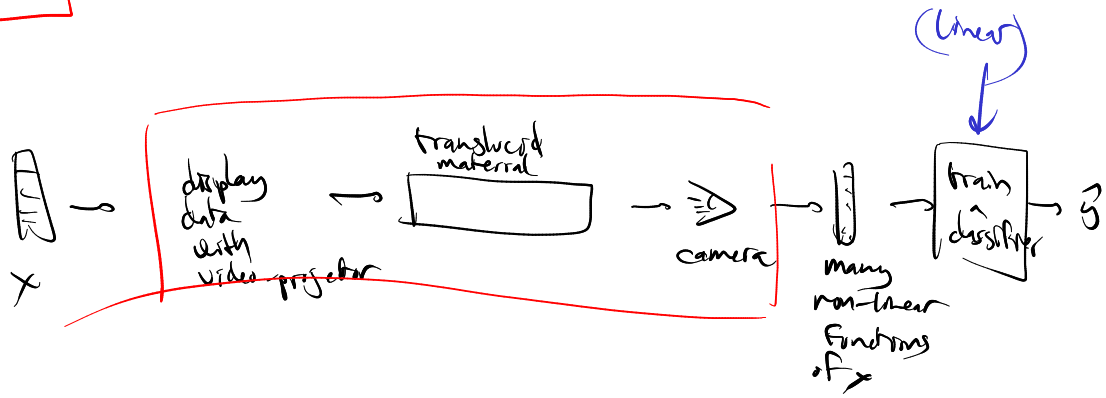
~~VGG~~ AlexNet

deeper model



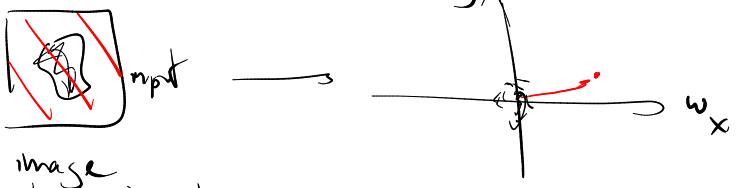
If random weights: good reconstruct of x
If trained: not as good (keep only inform° relevant for the task)

Light On:

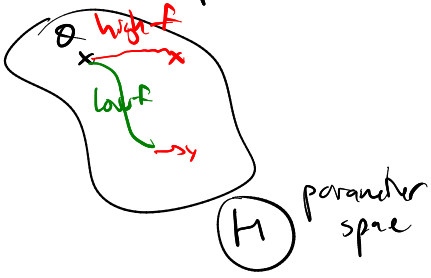


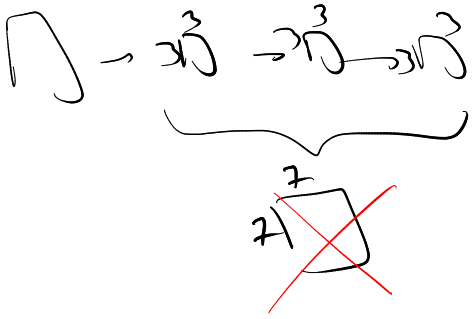
Architecture bias:

CNN



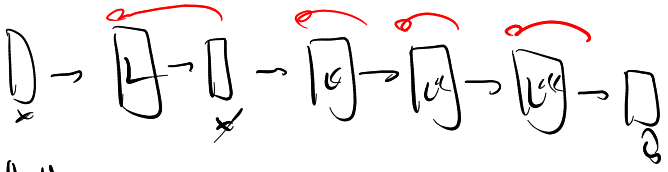
from CNN: focus on low Frequencies first
↓
main objects
before
pixel noise





Random initialization: with which law?

$\|x\| = \|x\|_{L^2} = \sqrt{\sum_i x_i^2}$



$\|x\| \iff \|x'\|$

$\|L(x)\| \simeq \gamma \|x\|$

$\gamma = ?$

Recurrent networks

$\|\hat{y}\| = \gamma^L \|x\|$

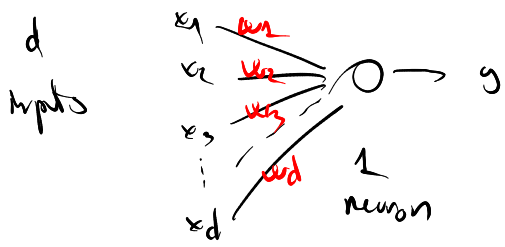
⇒ $\gamma > 1$ → exploding

$\gamma < 1$ → vanishing

$f(x)$

∇f

Xavier Glorot's initialization



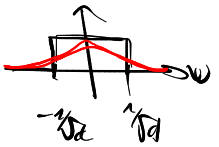
$$y = \sum_i w_i x_i \quad (+b)$$

Hp: inputs $x_i \sim \mathcal{N}(0,1)$ or mean $=0$, $\sigma = 1$

pick $w_i \sim \mathcal{L}$

goal: find $\mathcal{L}$ s.t. $\text{var}(y) = 1$

solution: $w \sim \mathcal{U}\left[-\frac{1}{\sqrt{d}}, +\frac{1}{\sqrt{d}}\right]$



or $w \sim \mathcal{N}(0, \sigma^2 = 1/d)$

Proof: $y = \sum_i w_i x_i$

mean: $\mathbb{E}_{x,w}[y] = \mathbb{E}_w\left[\sum_i w_i \mathbb{E}_x[x_i]\right] = 0$

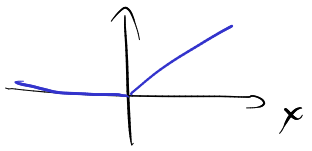
variance: $\mathbb{E}_{x,w}[y^2] = \mathbb{E}_{x,w}\left[\sum_i w_i^2 x_i^2 + \sum_{i \neq j} w_i w_j x_i x_j\right] = \mathbb{E}_w\left[\sum_i w_i^2 \mathbb{E}_x[x_i^2]\right]$

$= d\,\mathbb{E}[w^2] \cdot 1$

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

With activation:

$y = \text{ReLU}(x)$ =



$\text{var}(\text{ReLU}(x)) = \frac{1}{2}\text{var}(x)$

↳ standard deviation: $\sqrt{2}$

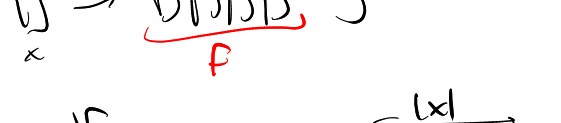
↳ correcting factor: $\times \frac{\sqrt{2}}{\sqrt{d}}$

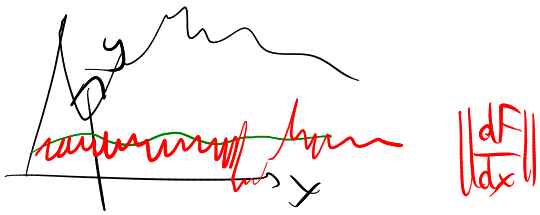
Bayes: not to 0

Other approaches: Kaiming He (& al)

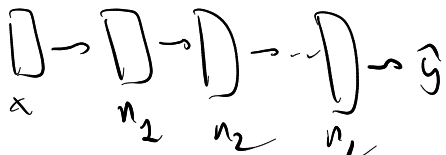
$$\frac{1}{\sqrt{\sqrt{d} \cdot \sqrt{d}}}$$

Further: (Boris Hanin, 2018): Jacobian properties





$J = \frac{dF}{dx}$



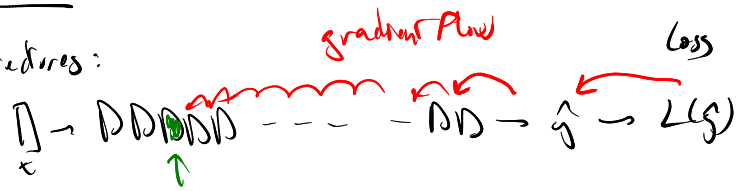
$\text{var}\left(\frac{dF}{dx}\right) = 1/n_0$

$\alpha\, e^{\sum_l 1/n_l} \leq \text{var}\left(\frac{dF^2}{dx}\right) \leq \beta\, e^{\sum_l 1/n_l}$

Neural Tangent Kernel (NTK) : $\nabla_\theta f \times \nabla_\theta f^T$ limit of case $n_\ell \to +\infty$

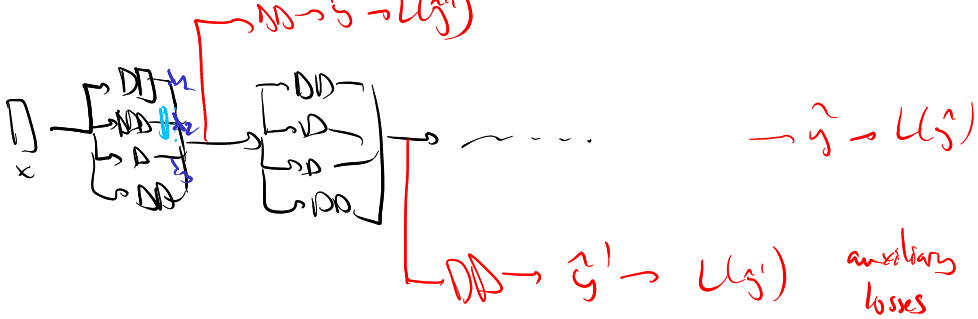
Design easy-to-train arch.

- deeper architectures :



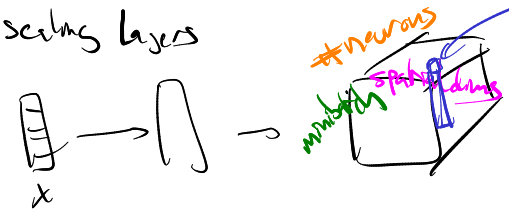
ResNet

Inception :

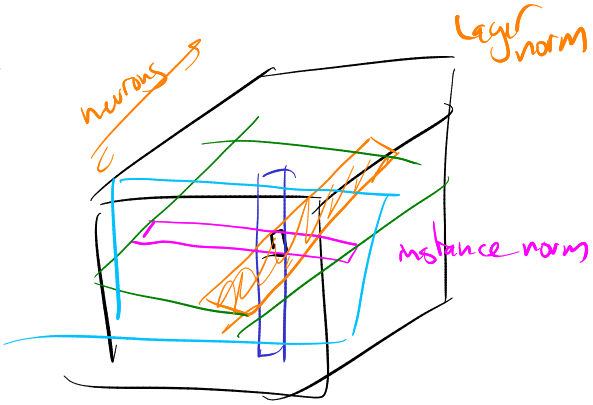


Normal :

- SELU activ° : keeps the variance
- add scaling layers



- 4 dims



$$\lambda \frac{A_{ij} - \mu}{\sigma} + b$$

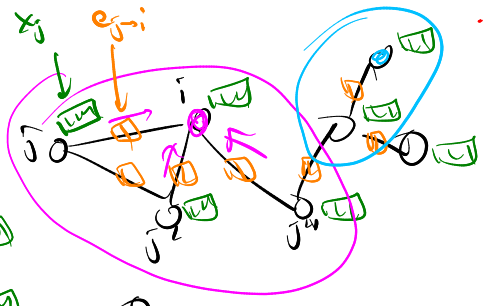
parameters

average or sum

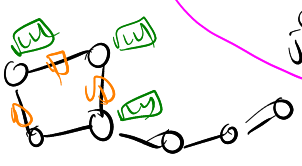
$$\cdot \left(\frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} w\, x_j \right) + w' x_i$$

Graph-NN

1 input = 1 graph



eg :
- molecule
- mesh



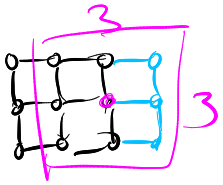
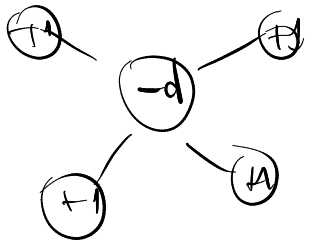


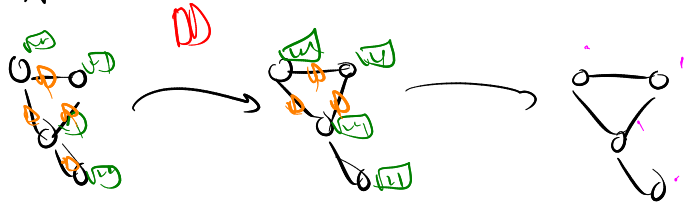
image
regular grid

- social graph



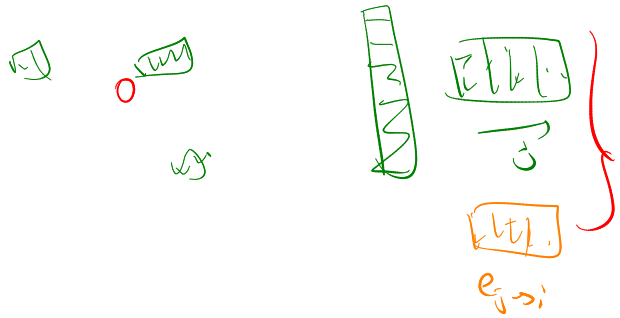
$w_j = g(e_{j\to i})$ $\quad g(e_{j\to i}, x_j, x_i)$

graph attention networks GAT

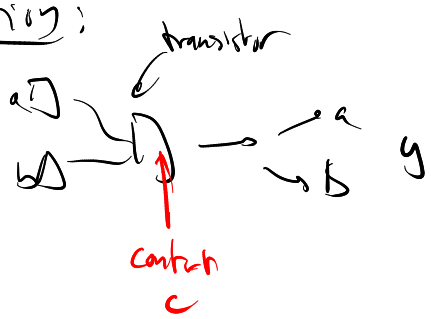


$e'_{j\to i} = h(e_{j\to i}, x_i, x_j)$

average over graph → 1 output for full graph



Attention:

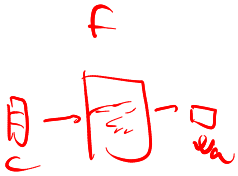


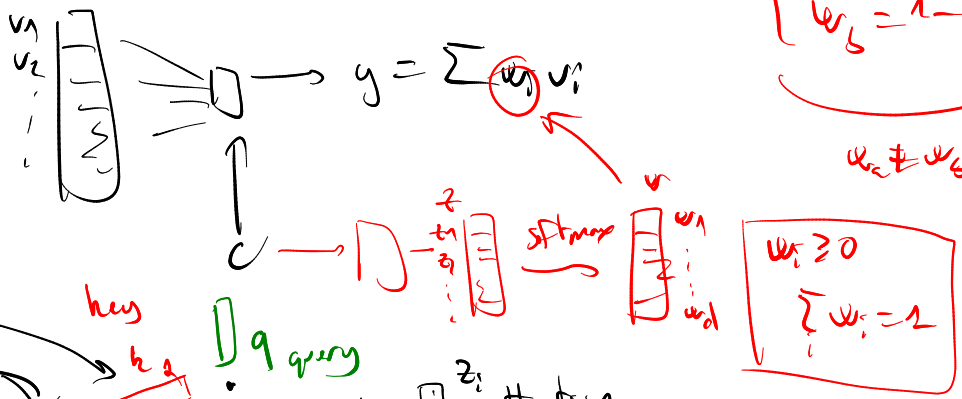
$y = w_a a + w_b b$

$= f(c)$

$\begin{cases} w_a = f(c) \\ w_b = 1 - w_a \end{cases}$

$w_a + w_b = 1$

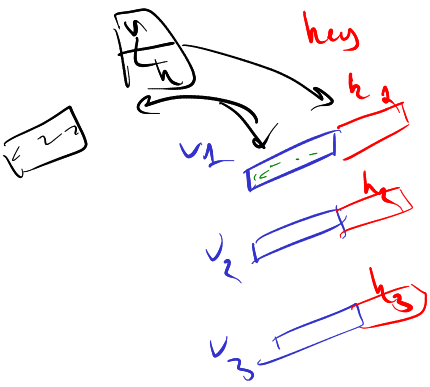




$y = \sum w_i v_i$

$w_i \geq 0$

$\sum_i w_i = 1$

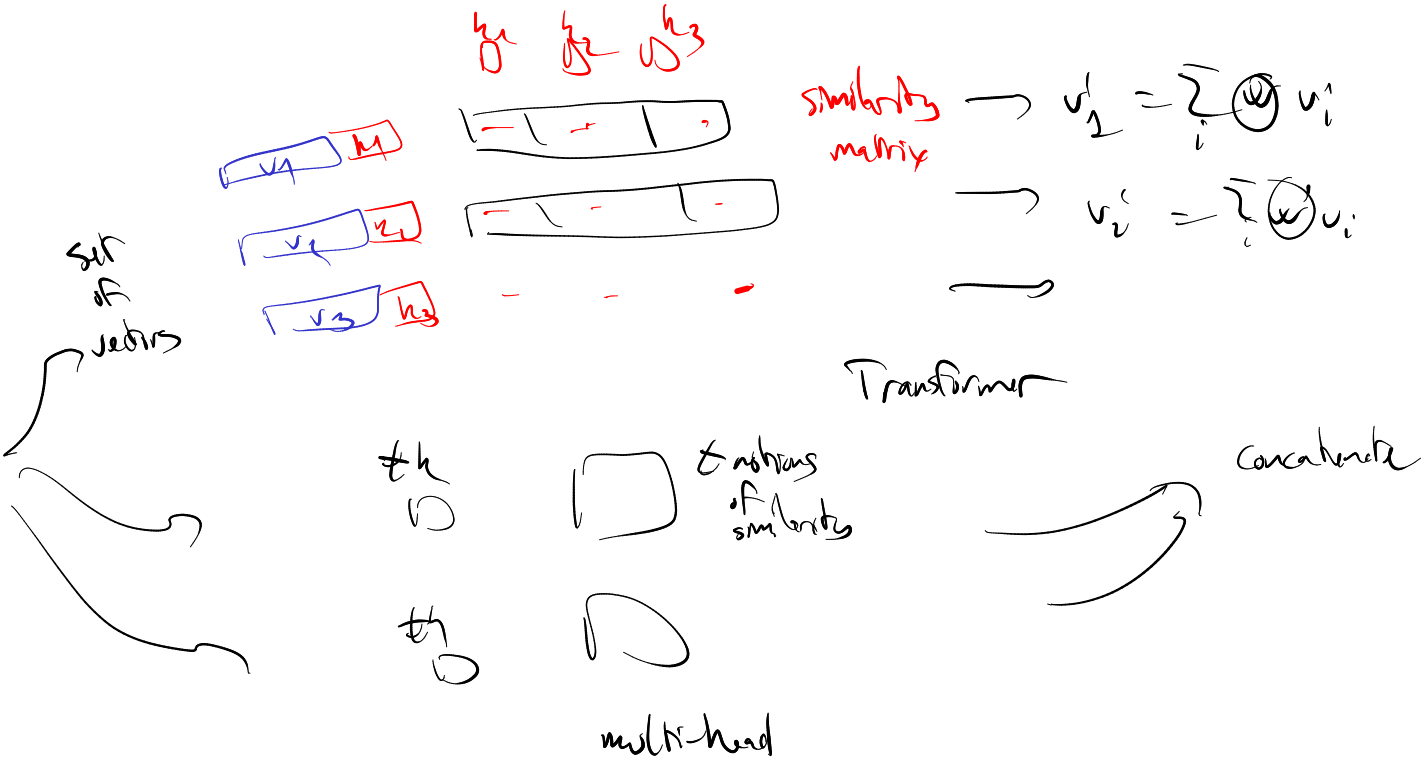


keys

query

$q \cdot k_i$ attention

$\rightarrow y = \sum_i w_i v_i$


similarity matrix
Set of vectors
Transformer
heads
Concatenate
multi-head