Neural Net → Supervised learning

↳ Kernel

→ MLP → CNN, ResNet → Transf.

Optim°: SGD → Backprop

Theory MLP

Not°: Data $(x_i, y_i)_{i=1}^{n}$ n #pts

x_i: Features $\in \mathbb{R}^d$

y_i: Label

Parameters: θ

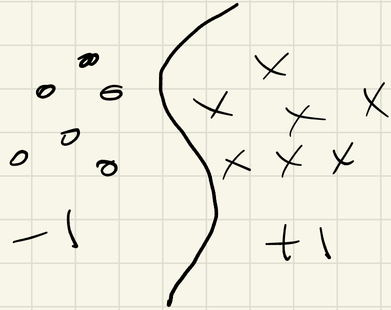
Supervised learning (Regress°)

$$y_i \approx f_\theta(x_i)$$

??

Classif° : $y_i \in \{-1, +1\}$.

$y_i \approx \underline{\text{sign}}\big(\underbrace{f_\theta(x_i)}_{\text{Score}}\big)$



ERM : $\min_{\theta} \underbrace{\frac{1}{n} \sum_i \ell\big(f_\theta(x_i), y_i\big)}_{\mathcal{E}(\theta)}$

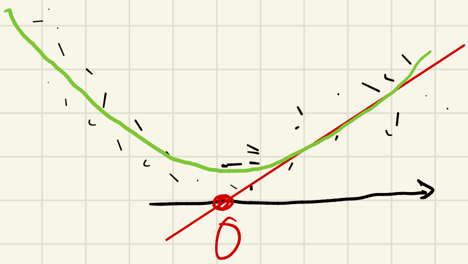
Unregularized

$$\begin{cases} \ell(y, y') = (y - y')^2 & \text{leas square} \\ \ell(y, \underbrace{y'}_{\in\{-1,+1\}}) = \log\left(1 + e^{-yy'}\right) & \left(\text{logistic loss}\right) \end{cases}$$

Kernel method : $\underset{\in \mathbb{R}^d}{x_i} \longrightarrow \varphi(x_i) \in \mathbb{R}^D$

Linear method: $y_i \approx f_\theta(x_i) = \langle \theta, x_i \rangle$

$$\langle x, \theta \rangle = \sum_k x[k]\, \theta[k]$$



$$\varphi(x) = [1, x, x^2]$$

$d = 1$

$$\langle \theta, \varphi(x) \rangle = \theta[1] \times 1 + \theta[2] \times x + \theta[3] \times x^2$$

Kernel Trick

$$\mathcal{E}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\langle \varphi(x_i), \theta \rangle - y_i \right)^2 + \lambda \|\theta\|^2$$

$$\Phi = \begin{bmatrix} \varphi(x_1) \\ \varphi(x_2) \\ \vdots \\ \varphi(x_n) \end{bmatrix} \in \mathbb{R}^{n \times D}$$

$$\mathcal{E}(\theta) = \frac{1}{n} \left\| \Phi \times \theta - y \right\|^2 + \lambda \|\theta\|^2$$

$$\min_{\Theta} \mathcal{E}(\Theta) \iff \nabla \mathcal{E}(\Theta) = 0$$

$$\nabla \mathcal{E}(\Theta) = \frac{1}{m}\Phi^T(\Phi\Theta - \underline{\underline{y}}) + 2\lambda\Theta = 0$$

$$\left(\frac{\Phi^T\Phi}{m} + \lambda \mathrm{Id}\right)\Theta = \frac{\Phi^T y}{m}$$

$$\Theta_{LS} = \left(\frac{\Phi^T\Phi}{m} + \lambda \mathrm{Id}_{D\times D}\right)^{-1} \Phi^T y$$

Kernel Trick / Woodbury formula

<u>Thm</u> $\Theta_{LS} = \Phi^T \left(\frac{\boxed{\Phi\Phi^T}}{m} + \lambda \mathrm{Id}_{m\times m}\right)^{-1} y$

kernel meth

$$\hat{K} = \Big(\underbrace{\langle \varphi(x_i), \varphi(x_j)\rangle}_{k(x_i, x_j)}\Big)_{i,j=1}^{m}$$

Works if k is "valid" kernel

<u>Def</u>: k is an SDP kernel iff

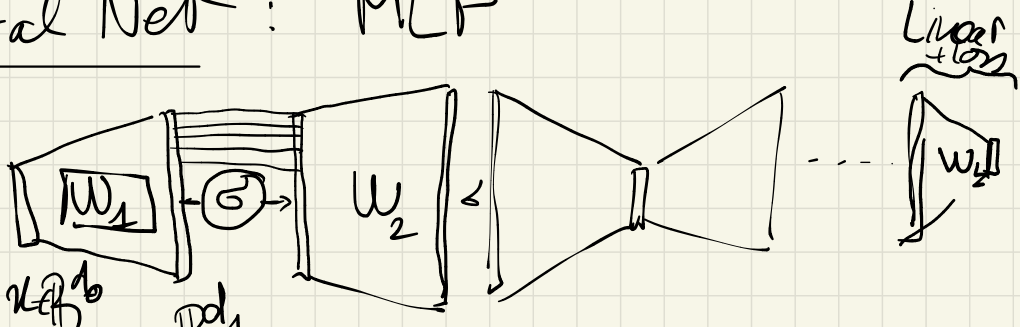
$$\left(k(x_i, x_j)\right)_{ij=1}^{M} \text{ has positive eigen}$$

$$\underset{\uparrow}{k}(x, x') = \exp\left(-\frac{\|x-x'\|^{p}}{\sigma}\right) \qquad p = 1$$

$p \geq 1$

x is a graph $\qquad k(x, x)$

<u>Neural Net</u>: MLP

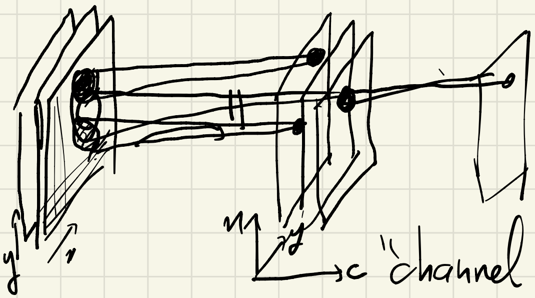


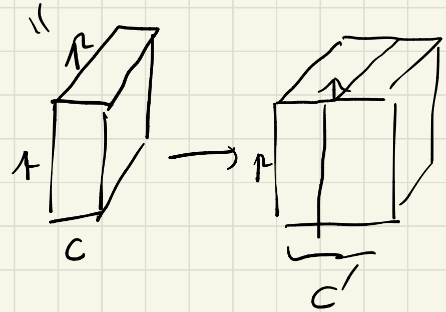
$$\langle \varphi_\theta(x), W_L \rangle$$
$$\theta = (W_1, W_2, \dots, W_{L-1})$$

$x_0 = x$ $\qquad$ $x_{k+1} = \sigma(W_k x_k)$.

$$f_\theta(x_0) = x_L$$

CNN: "Weight sharing".





Convol^n: ψ "kernel" $\qquad a*\psi = \psi * a$

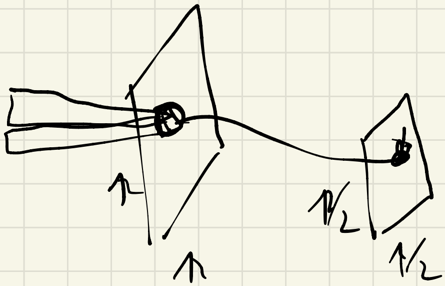
$$(a*\psi)[r] = \sum_{\delta \in \{-1,0,1\}\times\{-1,0,+1\}} a[r-\delta]\,\psi[\delta]$$

Each CNN layer: $c \to c'$

$$\{\psi_{st}\}_{\substack{s=1\dots c\\ t=1\dots c'}}$$

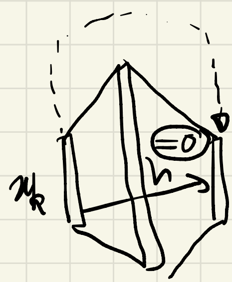
$$x_{k+1}[\cdot,\cdot,t] = \sigma\left[\sum_{s=1}^{c} x_{k+1}[\cdot,\cdot,s] * \psi_{st}\right]$$

$$W_k \longrightarrow \{C_{st}\}_{s,t}$$



CNN = conv + sub

ResNet: Block



$$x_{k+1} = \underline{\underline{x_k}} + \underbrace{h(x_k)}_{=0}$$