MATH 462

project examples.

Image Segmentation Losses

Boundaries & Geometry.

RL Losses (time permitting)

Lecture 16

29.10.2021

project examples.

Loss design for Image segmentation.

$\mathcal{X} = \mathbb{R}^d$ $\quad d = d_1 \times d_1$ $\quad$ ~~color~~ image.

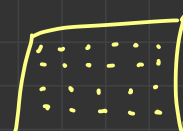
$\mathcal{Y} = \mathcal{Y}_K^d$ $\quad$ each pixel has $y_j \in \{1, \dots, K\} = \mathcal{Y}_K$

$x_i \quad y_i \in \mathcal{Y}_K^d$

Goal learn $h(x) \in \mathcal{Y}$

Our Goal understand losses for this type of $\mathcal{Y}$

Step 1 Ignore structure of image



adjacent pixels should usually have same class.

Convolutional Neural Networks.
CNN

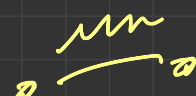
1. NN activations
2. DNN
3. CNN

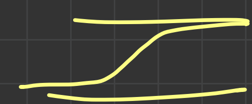
0. Linear models.

History Neural Networks X

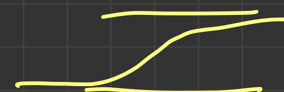
X ML x NN x CNN.

neurons





$$H = \left\{ x \to \begin{pmatrix} w_1\cdot x \\ \vdots \\ w_n x \end{pmatrix} \to \begin{pmatrix} \sigma(w_1\cdot x) \\ \vdots \\ \sigma(w_n\cdot x) \end{pmatrix} \right\}$$



$\sigma(t) \in (0,1)$ $\quad$ $\sigma(t) = \frac{1}{1+e^{-t}}$

$t \in \mathbb{R}$

one layer neural network

nonlineartity.

good for a while.

XOR

input $(\pm 1, \pm 1)$

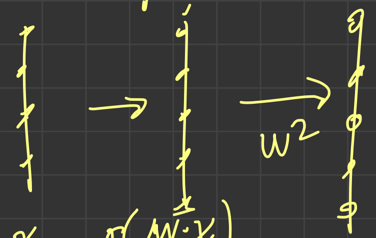
OR $\max(b_1, b_2)$

and $\min(b_1, b_1)$

OR − and

	-1	$+1$
-1	0	1
$+1$	1	0

Add hidden layer



x_1 $\quad \underbrace{\sigma(W\cdot x)}_{=f_1}$ $\quad y_1 = \sigma(W^2 \cdot f_1)$

$f_1(x) = \sigma(W\cdot x)$

2-Layer network

<u>Thm</u> universal approximation.

"Can fit any fn" with 2-layer n-n.

missing

→ rate

→ method

→ generalization.

<u>ML</u> later said more.

fit with algorithm

min $L(W)$

DNN Deep neural network with L layers

ResNet 56

$$x \to f_1 = \sigma(W_1 x)$$

$$f_{k+1} = \sigma(W_k f_k(x))$$

56 layers

size big. $W : n_k \to n_{k+1}$ full matrix.

Apps RL NLP (early)

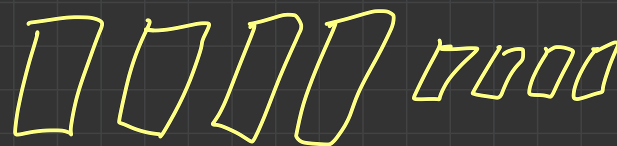
shallow n.n.

Computer Vision U of T 2007 ImageNet CNN

ResNet

$$f_k(x) \in \mathbb{R}^{n_k} \longrightarrow f_{k+1} = f_k(x) + \sigma(W_k f_k(x)) \in \mathbb{R}^{n_k}$$

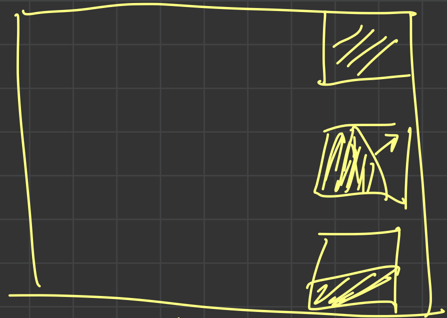
or $\sigma((I + W_k) f_k(x))$



No free lunch: one learning alg. can't work for everything.

Ans: model should incorporate some domain knowledge

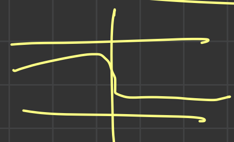
CNN



$d = 10^6$

$W : \mathbb{R}^{10^6} \to \mathbb{R}^{10^6}$

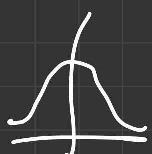
Gabor Filter



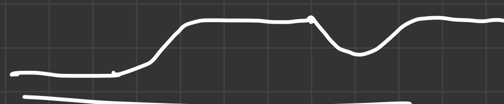
$|f'(x)|$

Discrete $\left|\frac{f(x+h) - f(x)}{h}\right|$ $\frac{1}{h}(+1,-1)\cdot(f_i, f_{i+1}, \ldots)$

Convolution



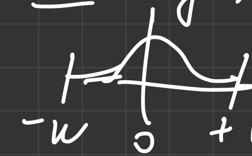
$\int g(x) = 1$



Given f: functn

$f : \mathbb{R} \to \mathbb{R}$

Given $g(x) : \mathbb{R} \to \mathbb{R}$



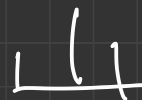
$g(y) = 0$

$|y| > w$

$f * w(x)$

$$= \int_{-w}^{w} f(x+y)\, g(y)\, dy$$

Ex  $f_i(x)$

$G(x) =$ 

$g_{-1} = 0.25$

$g_0 = 0.5$

$g_1 = 0.25$

$$(f * g)_i = \sum_{j=-1}^{1} f_{i+j}\, g_j$$

Check $f = (1, 1, 1, -1, 1, 1, 0, 1, 1)$ 

1

BC $f_{-1} = f_{left}$

0.25 .5 .25

$-\frac{1}{2} + \frac{1}{2}$

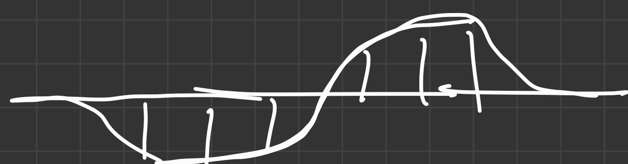
$f * g = (1, 1, 0.5, 0, 0.5, 0.75, 0.5, .75, 1)$ 

Diff conv $g = (+1, -1)\frac{1}{2}$ $f * g \cong f'$

Approximate $f'(x)$ several ways.

$$\frac{f(x+h)-f(x)}{h} \qquad \frac{f(x)-f(x-h)}{h} \quad \frac{(-1,\ +1)}{h}$$

$$\frac{f(x+h)-f(x-h)}{2h} \quad \left(\frac{-1\ 0\ +1}{2h}\right)$$



2D edge detectors Gabor filters

CNN trained see 1^{st} layer
is close Gabor filters

1d convolution Matrix

$g = (g_1, g_2, g_3, g_4, g_5)$

$$M = \begin{bmatrix} g & 0 & & & & \\ 0 & g & 0 & & & \\ 0 & 0 & g & 0 & & \\ 0 & 0 & 0 & g & 0 & \end{bmatrix} \quad (**)$$

$$M \cdot f = f * g$$

$(Mf)_i$

$= \sum m_{ij} f_j$

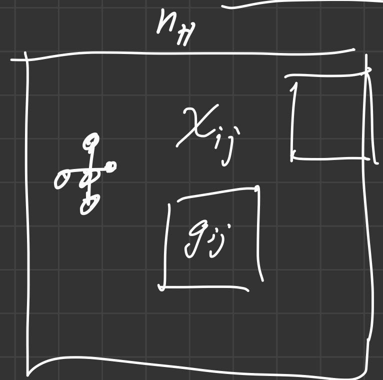
$$(f * g)_i = \sum_{j=-1}^{2} f_{i+j}\, g_j \quad (*)$$

Exercise show can represent with a matrix of form $(**)$

EX $f'(x)$ as $f * g$ $\quad g = (-1, 0, 1)$

$$M = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ & & & \cdots & & & \end{bmatrix}$$

2d conv Matrix



$\mathrm{Ind}(i,j) = ni + j$

1	6	11
2	7	12
3	8	13
4	9	14
5	10	15

2d $\quad f * g = \int_{-w}^{w}\int_{-w}^{w} f(\vec{x}+\vec{y})\,g(\vec{y})\,dy$

2d convolute matrix.

Ex $\quad \dfrac{\partial f}{\partial x} \qquad \dfrac{\partial f}{\partial y}$ $\qquad$ convolution $\qquad \Delta f = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2}$

$$\begin{bmatrix} -1 & 0 & +1 & & & \\ & -1 & 0 & +1 & & \\ & & -1 & 0 & +1 & \\ & & & & & \ddots \end{bmatrix}$$

$$\begin{array}{c} \overbrace{\qquad\qquad\qquad\qquad}^{n_H} \\ \begin{bmatrix} -1 & 0\,0\,0\,0\;0\,0\,0 & +1 & \\ & -1 & 0\,0\,0\,0\,\cdot & +1 \end{bmatrix} \end{array}$$

similar sparsity.

Conv Matrix implements Mf efficient way layer 1

$f_1 = \sigma(W_1 x)$

Wx
implement as
a small convolution
$[3 \times 3]$

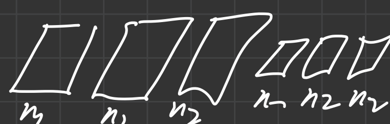
layers k

$f_{k+1} = \sigma(W_{k+1} f_k)$

still convolution
for hardware purposes.

Conv. NN

hyperparam. Architecture



$W \in n \times M$

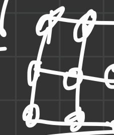
W full
W conv. shape

1d convolution
$[-w, 0, +w]$

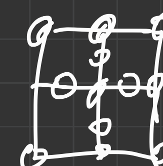
OR

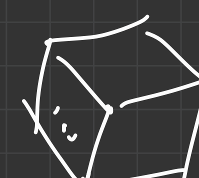
$-4\ -2\ 0\ 2\ 4$
skip

2d



adjacent





Convolutional

f_i represent $f_1, \ldots, f_n$

f_{ij}

f_{ijk}

rectangle