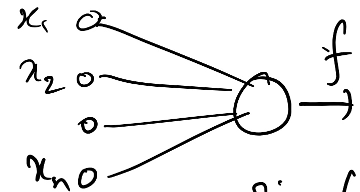
THEORY OF MACHINE LEARNING

LECTURE 18

NEURAL NETWORKS – REPRESENTATION BASICS

RECAP

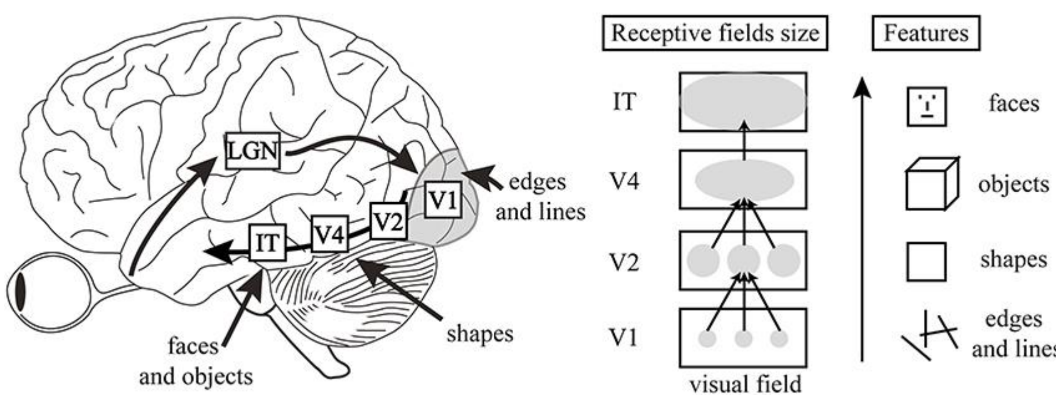


$\text{sign}(a^T x - b)$

optional.

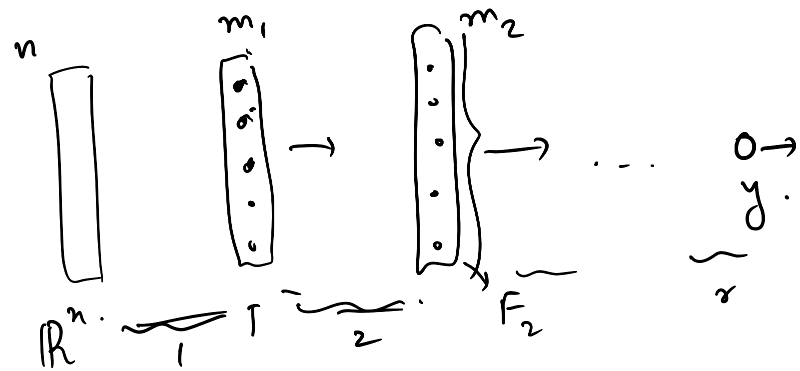
- Perceptron, or linear threshold
  - Hypotheses of form sign($\langle a, x \rangle$) for an appropriate weight vector a
  - Generally, $\sigma(a^T x)$ for some "activation function" σ
- Biologically inspired, arithmetic circuit (with threshold gate)
- Idea behind neural nets:
  - Perceptrons detect "basic" or "primitive" features; 'composing' them allows for complex decision-making (can get high level concepts using basic features).
  - Supported by human visual system (V1, V2, …)

~6-7 layers.



RECAP: ARTIFICIAL/DEEP NEURAL NETWORK (DNN)

- **Definition.** A layered "circuit" that takes a vector of input features x, produces output y = $F_r \circ F_{r-1} \circ \cdots \circ F_1(x)$, where each F_i is a function of the form $F_i(z) = \sigma(Az + b)$, for some activation function $\sigma()$ (that acts coordinate-wise)



- Common activation functions:
  - Threshold
  - Sigmoid: (continuous approx.) $\frac{1}{1+e^{-x}}$
  - ReLU, Tanh
  - …

$$f_1(x) = \sigma(A_1 x + \underbrace{b_1}_{\in \mathbb{R}^{m_1}})$$

depth = r.

BASICS

linear classifiers++.

- Neural networks are basically a (fairly complex) hypothesis class – takes input x, produces y

feature vector $\in \mathbb{R}^n$

label $\in \mathbb{R}$.

- **Question (vanilla supervised learning):** given data $(x_1, y_1), (x_2, y_2), \ldots$ from some distribution D, find *h* in this class that minimizes the risk

$$\text{risk}(h) = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{1}[h(x) \neq y] \quad \leadsto \quad \mathbb{E}_{(x,y)\sim\mathcal{D}} \, \ell(h(x), y).$$

label(x) real valued y

- ERM problem usually called neural network "training" – given data, find best hypothesis ($f(x_i) = y_i$) for all *i*

ERM: Minimize training loss

(using depth, width of network bounded appropriately).

THEORY OF DEEP LEARNING

- Expressibility (how rich is this hypothesis class?)
  - What kinds of functions can be obtained using a DNN? +ve.
- Training complexity & training dynamics for GD and variants
  - Can the ERM problem be solved efficiently? What guarantees are possible?

    − news.
- Generalization
  - What kind of generalization bounds can we prove? (VC dimension?) → ≈ #parameters

    + news.

Key: "easy" answers for all questions, but unsatisfactory for realistic settings

EXPRESSIBILITY BASICS

- **Barron's theorem [93].** *Any continuous function f that satisfies an appropriate "niceness" condition (parametrized by C) can be approximated to error ϵ (in L2!) by a 2-layer NN with $\sim \frac{C^2}{\epsilon}$ internal nodes*

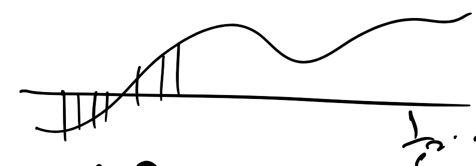
  $\int x \, |\hat{f}(x)| \, dx \le C$.

- (Nice functions can be approximated by small NNs) (2 Layer)

(Good topics for

- **Universal approximation** [Cybenko, Hornik '87,'91]. *Any continuous function (over a compact domain) can be approximated by a 2-layer NN with any non-linearity (not a ~~polynomial~~)* (pointwise)

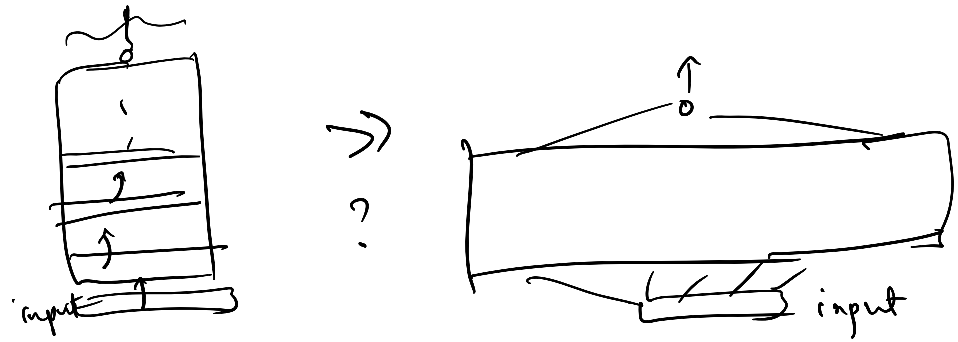
"Sigmoidal" non-lin:



But wait.. who uses infinitely wide 2 layer nets?

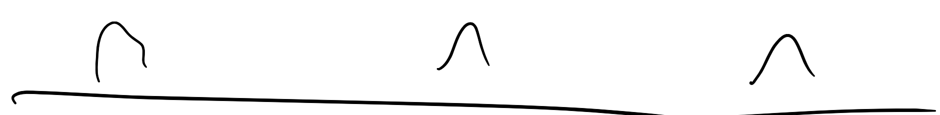
→ input dim $= n$ ⇒ width needed for "basic" fns is $\sim 2^n \cdot (\exp(n))$

DEPTH VERSUS WIDTH



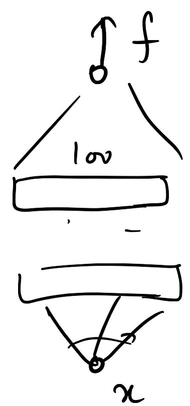
- Practical intuition:
  - Depth allows "meaningful features" while width is for "~~brute force~~ memorization"
- Universality results degrade rapidly with dimensions

  (Universality: Any fn is expressible using depth-2 NNs: n-dim inputs, $\left(\frac{1}{\varepsilon}\right)^n$)
  - Curse of dimensionality (width ~~should~~ needs to be exp in dimension).
  - Modern nets work with high dimensional data
- Does higher depth lead to higher expressibility (with much fewer neurons)?
- Bunch of works … [Eldan and Shamir (depth 2 vs depth 3)], [Telgarsky], 2015-16

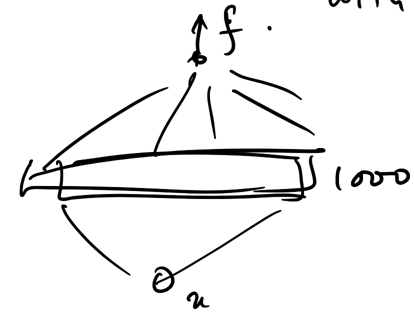


(fix input dim = 1)

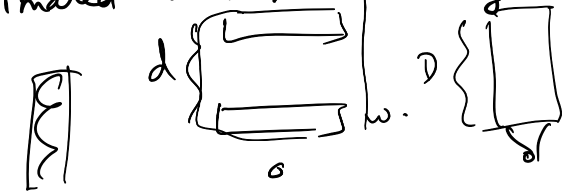
$\mathcal{F}_1 : \left\{ f : \begin{array}{l} f \text{ is the output of a depth 3 NN with} \\ \text{width} \leq 100. \end{array} \right\}.$



$\mathcal{F}_2 : \left\{ f : \begin{array}{l} f \text{ is the output of a depth 2 NN} \\ \text{with width} \leq 10^8 \end{array} \right\}$



<u>Qn:</u> Can any fn in $\mathcal{F}_1$ be approximated to error $\leq \varepsilon$ using functions in $\mathcal{F}_2$?

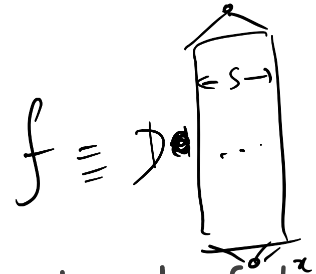


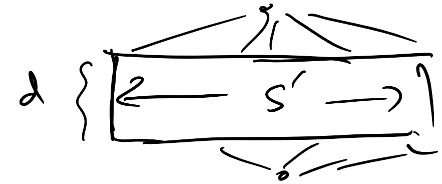
→ <u>CIRCUIT LOWER BOUNDS</u> (boolean): Look at specific f of interest, and ask if $\exists$ a circuit of size $\leq S$ that computes f?

[Minsky Pappert '69] [Razborov '80s) : parity function requires "exponential width"

[Håstad '86]. 4 inputs 1110 → 1, 1100 → 0. fns : $\{0,1\}^n \to \{0,1\}$.

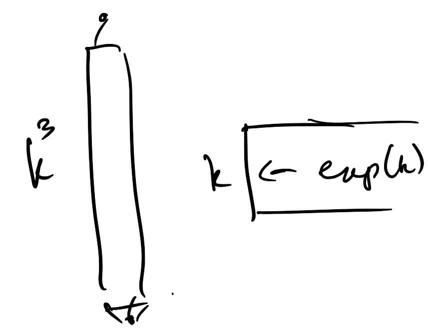
POWER OF DEPTH





Theorem template. There exists a network of depth D and "size" S that computes some function f that cannot be approximated by the output of *any* network with depth d and size S' (typically if d << D, S' will be >> S)

- "Depth versus width" results
- Reminiscent of circuit complexity (original work of Minsky, Pappert)



[Telgarsky 16]. For any k>0, theorem holds with:

D = S ~ k^3, d = k, and S' = 2^k (and ReLU activations)

$D = k^3$; width $\sim \Theta(1)$ s.t. approximating with depth k requires $\exp(k)$ width...

PROOF OUTLINE

- Consider just one-dimensional inputs and ReLU activations
- **Key insight:**
  - depth D lets us achieve exp(D) many "osciallations" in f
  - getting so many osciallations with depth d requires huge width!

↳ width w and depth $= d$, #oscillations $\leq w^{O(d)}$

n, d.

$n^{\log d}$

$d^{\log n}$

$D = k^2$; $d = k$.

$2^{k^2} = w^{10k} \Rightarrow w > \frac{2^{k}}{10}$.

$e^{k^2} \gg (1000)^k$

$(e^k)^k$ $\quad$ $e^{(\log 1000)k}$

Part 1:

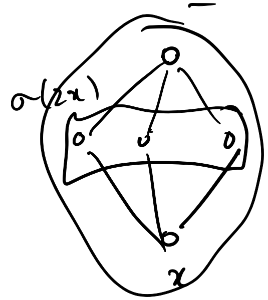
→ Single variable input x : domain $= [0,1]$.

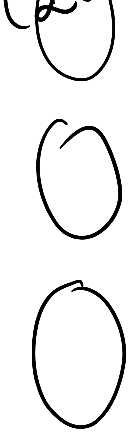
→

σ : ReLU : ~~max~~ $\sigma(x) = \max(0, x)$

$$\sigma(2x) - \sigma\left(4\left(x - \tfrac{1}{2}\right)\right) + \sigma\left(2(x-1)\right)$$

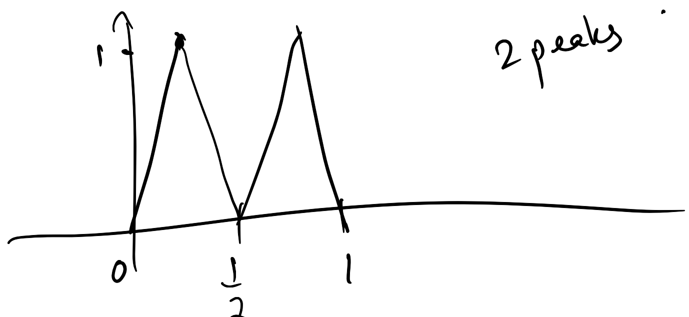
$f(x)$





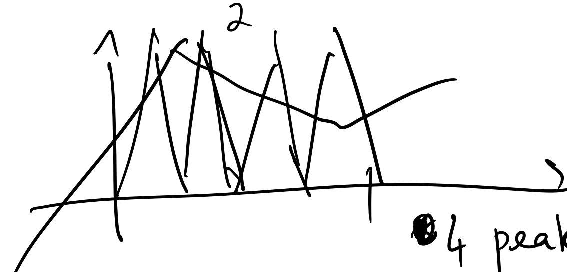
what is f?

→ $f(f(x))$?



2 peaks.

→ $f(f(f(x)))$ →

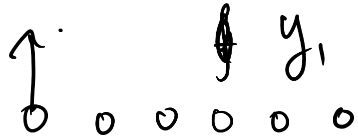


4 peaks.

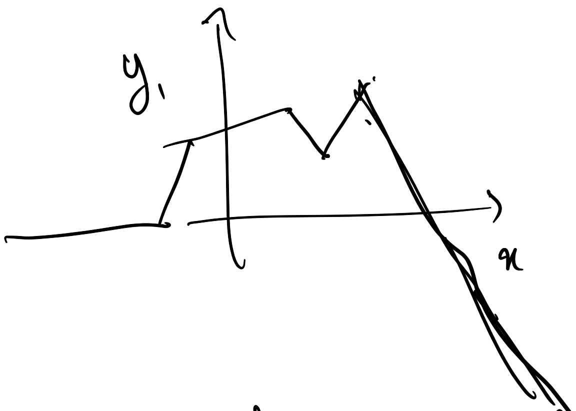
$f^{o(k+1)}(x) \to 2^k$ peaks.

depth $\sim 2k$ NN with width ≤ 3, you can implement this..

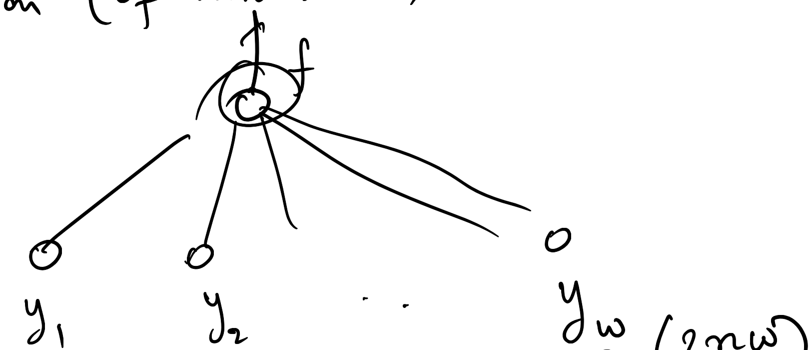
Part 2: Want to show that if depth $\ll k$, getting this fn requires huge width.

Obsn: 

y_1 is a neuron that is a piecewise lin. function (of variable x) with $\le n_1$ pieces.



Idea:



$2w(2nw)$

f is piecewise linear & # pieces in f is $\le \underbrace{(n_1 + n_2 + \dots + n_w)}^{2}$.

sawtooth

Part 3: If f_1 and f_2 are p.w.l. with m_1 and m_2 pieces and $m_1 < \frac{m_2}{2}$; then $\|f_1 - f_2\|_1 \geq \frac{1}{8}$.