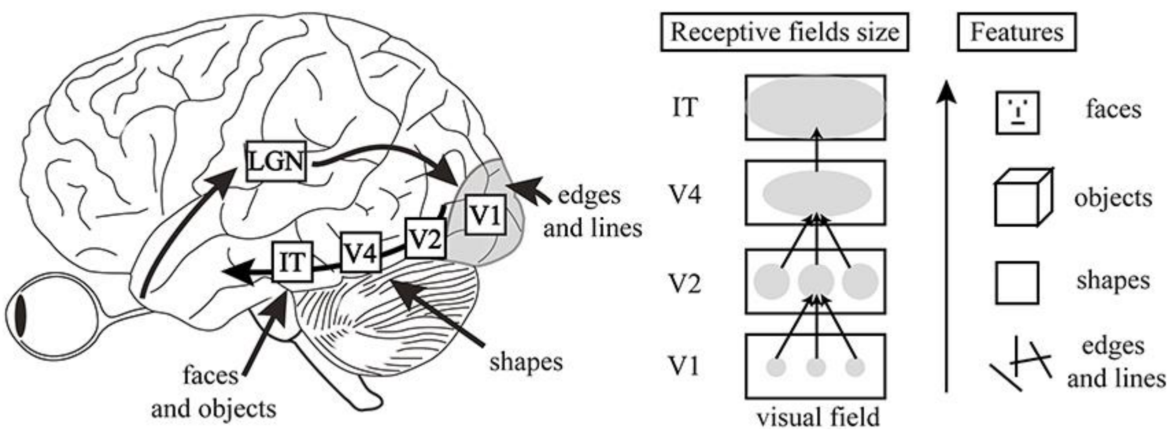
THEORY OF MACHINE LEARNING

LECTURE 19

NEURAL NETWORKS – REPRESENTATION, OPTIMIZATION

RECAP

- Idea behind neural nets:
  - Perceptrons detect "basic" or "primitive" features; 'composing' them allows for complex decision-making
  - Supported by human visual system (V1, V2, …)



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
  - …

LEARNING NEURAL NETWORKS

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

THEORY OF DEEP LEARNING – THREE BROAD DIRECTIONS

- Expressibility
  - What kinds of functions can be obtained using a DNN?
- Training complexity & training dynamics for GD and variants
  - Can the ERM problem be solved efficiently? What guarantees are possible?
- Generalization
  - What kind of generalization bounds can we prove? (VC dimension?)

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*
- (Nice functions can be approximated by small NNs)
- **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)*

Curse of dimensionality for Cybenko **(not Barron)**

WHY "DEEP" NETWORKS?

- Practical intuition:
  - Depth allows "meaningful features" while width is for "brute force memorization"
- Universality results degrade rapidly with dimensions
  - Curse of dimensionality
  - Modern nets work with high dimensional data
- Does higher depth lead to higher expressibility (with much fewer neurons)?
- Yes! [Eldan and Shamir, Telgarsky]

POWER OF DEPTH

Theorem [Telgarsky 16]. There exists a network of depth k^2 and $O(1)$ width that computes function f, with the property that *any* network of depth k that approximates f requires width > 2^k

(For more general piecewise poly functions, first bound changes to k^3)

MORALS

- Depth allows capturing “complex patterns”
- Width allows capturing “different regions of space”

- What is the right network for an application?
  - Very hard question (Neural Architecture Search)
  - Example of Vision + NLP problems
  - Hebbian principle
  - Needs exploiting domain knowledge (physics informed ML)

NEURAL NETWORK TRAINING

- **Supervised learning of NN:** *given* data $(x_1, y_1), (x_2, y_2), \ldots$ from some distribution D, find *h* that minimizes the empirical risk
  - Standard metrics: squared loss, cross entropy

- ERM problem for neural nets
- NP hard to learn weights, even if classification error is 0 and we just have 3 internal nodes

COMMON ALGORITHM – GRADIENT DESCENT