THEORY OF MACHINE LEARNING

LECTURE 20

NEURAL NETWORKS -- OPTIMIZATION

NEURAL NETWORKS (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

- **Question (supervised learning):** *given* data $(x_1, y_1), (x_2, y_2), \ldots$ from some distribution D, find *h* (with *given* "architecture") that minimizes the risk
- ERM problem usually called neural network **training**

- Neural networks can represent/approximate *any* function (Barron, Cybenko)
- Depth vs width trade-offs
- Choosing network architecture is key (inductive bias)
  - No general rules (heuristics like CNN, transformers, Hebbian learning, …)

LEARNING NEURAL NETWORKS

Theorem. (see textbook) Given an architecture, it is NP-hard to learn weights, even if classification error is 0 and we just have 3 internal nodes

- Worst case result – clearly not reflective of practice
- Can we obtain more "positive" results?

- <u>Common algorithm:</u> gradient descent – not too hard to compute gradients (exercise in chain rule)
  - Linear time implementation via "back propagation" (Rumelhart, Hinton, Williams)

IS GRADIENT DESCENT (GD) GOOD?

- **Running time?**

- **Question:** *given* data $(x_1, y_1), (x_2, y_2), \ldots,$ does running GD for N iterations result in training error <= OPT + f(N) [for some decreasing function?]
- Assuming the network architecture allows for zero error, does GD converge to zero error?

- Alternatives to GD – method of moments (shallow nets), …
  - [Chen, Klivans, Meka 2020]: in time exp(# internal nodes, depth, other params), can learn what GD can't ☺

OVERPARAMETRIZATION

- Question out of desperation: can we show that GD is good in *any* reasonable generality?

Theorem. [Jacot, Gabriel, Hongler 18] [Arora, et al. 2019] A width ~ n^3 network with *any* number of layers trained via GD from random initialization achieves zero training error.

(key idea: parameters don't change much during training if width is so large...)

FEATURE LEARNING

- Problems with the infinite width regime
- What about learning "features" from data?
  - What does this even mean?

- Traditional approaches: sparse coding