THEORY OF MACHINE LEARNING

LECTURE 17

NEURAL NETWORKS - INTRODUCTION

REVIEW OF OPTIMIZATION

- Convex optimization (minimizing convex function over convex domain)
- Local min = global min (false for non-convex – only local min "tractable")
- Gradient descent
  - Any Lipschitz function – $\frac{1}{\sqrt{T}}$ error after T iterations
  - Improved bounds for smooth functions (1/T) and strongly convex exp(-T) (extends to Polyak-Lojasiewicz)
  - Generic analysis technique – maintain a potential function $||x_t - x^*||^2$ or Fn value

IMPROVEMENTS, GENERALIZATIONS

- Nesterov's method for smooth functions (gets $\frac{1}{T^2}$ convergence)
- Polyak's "heavy ball" method (momentum)
  - Originally designed for strongly convex functions – achieves $\sqrt{\kappa}$ in exponent
- Second order methods, first order "proxies" (AdaGrad)
- <u>Theme:</u> avoid "slow" convergence – take large steps when possible
  - Non-convex functions – "slip out" of local minima
  - Perturbed gradient descent -- if you're not moving much via gradient descent, just make a "random jump" to a point in a neighborhood
- <u>Last lecture:</u> regularization, "stability" and generalization

NEURAL NETWORKS

BASICS

- Recall linear threshold functions (hyperplanes)
- Earliest neural net – perceptron
- “Activation function” -- biologically inspired
- Natural view as a (logic) circuit

BASICS

- Can view output as detecting some "basic feature" in data
- What if we want to use a "composition" of features?
  - E.g., we have linear classifiers for basic shapes; complex shapes expressible as different combinations of basic ones
- (Also biologically inspired)

BASICS (“ARTIFICIAL”/DEEP NEURAL NETWORK)

- **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
  - …

BASIC GOAL

- Neural networks are basically a (fairly complex) hypothesis class – takes input x, produces y
- **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 fit classifier
- Non-convex optimization problem, NP-hard even in very simple cases
- Works surprisingly well in practice!

THEORY OF DEEP LEARNING

- Expressibility (inductive bias, etc.)
- Training complexity & training dynamics for GD and variants
- Generalization

Key: worst case answers are easy; challenging to answer questions about "realistic" settings