Introduction

ATML track 1: Optimization and Neural Networks

Rik Sarkar

Overview of the track

- ML Theory Basics
  - The elements of a general ML system – Hypothesis classes, Domains, generalization
- Linear classifiers and Convex optimization
  - Convexity and convex optimization
  - GD and SGD, convergence rates
- Neural nets
  - ReLU vs other activations — what makes ReLU successful?
  - Why are deep networks better than shallow networks?
  - Cross entropy loss and loss landscapes – why neural networks overfit
- Why does SGD work?
  - Randomness
  - Sharp and flat minima
  - Fractal Dimensions and generalization

Overview of the track

- What we know about neural networks
  - Neural collapse
  - Overparameterization and Pruning
  - Double descent
- Additional topics:
  - Fairness – definitions, impossibility, why fairness is hard
  - Explainability – what does it mean to explain model behaviour?

Today

- Elements of machine learning: definitions and notations
  - Data, algorithms, sampling
- Empirical risk minimisation
- Generalization

Data domain $\mathcal{X}$

- We assume that all the input data comes from some known set $\mathcal{X}$
- Examples
- If input is a single number: distance
  - Then the domain can be real numbers: $\mathbb{R}$
  - Or more restrictive, positive real numbers: $\mathbb{R}^+$
- If input is two real numbers (e.g. age, income)
  - Then $\mathcal{X} = \mathbb{R} \times \mathbb{R} = \mathbb{R}^2$ is two dimensional
- What is the domain if the inputs are images?

Labels $\mathcal{Y}$

- The labels are outputs of the model from a set $\mathcal{Y}$
- Examples

- For classification, $\mathcal{Y} = \{0,1\}$ or sometimes $\mathcal{Y} = \{-1,1\}$
- Question, what are label sets for:
  - Multiclass classifiers
  - Regression
  - Image generative models

Model or hypothesis h

- A model or hypothesis is:
    - A map $h: \mathcal{X} \to \mathcal{Y}$

Hypothesis class $\mathcal{H}$

- A set of hypothesis or models under consideration
  - We will try to find the best model from this set
- Example:
- A type of model – SVM, decision tree..
- A neural network with a particular architecture
  - Then $\mathcal{H}$ is all possible assignments of values to parameters (edge weights)

Question: Why do we need an $\mathcal{H}$?

- What restricts our choice of models? Can we apply any neural network to any problem?
  - We are restricted at least by $\mathcal{X}$ and $\mathcal{Y}$
  - The model has to be compatible with input and output
- Can we search for the best over all neural network architectures?
  - No, because we don’t even know what all possible architectures are

Data generating distribution $\mathcal{D}$

- How will we get our training data?
- We usually can't get all of $\mathcal{X}$
- We can get a small sample from it

- There is some process that we use to get data
  - Question: what are examples of such processes?
- The abstract mathematical way of saying this is that data comes from a probability distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$
  - And we have some way of getting a sample data $s \sim \mathcal{D}$ (s is drawn from $\mathcal{D}$)

Training sample set S

- The training sample set is m samples from $\mathcal{D}$
- Written as $S \sim \mathcal{D}^m$
  - (Question: why is it reasonable to write it like this?)

- The m samples are assumed to be i.i.d
  - Independent: Sampling of one data point has no causal impact on the sampling of another
  - Identically distributed: they are all drawn according to the same distribution ($\mathcal{D}$)

Algorithm A

- Using S, finds and returns an $h \in \mathcal{H}$
  - $h = A(S)$

Loss or risk

- Consider a classifier
- Supposer for $x \in \mathcal{X}$,
  - The true label is y
  - And the computed label is $\hat{y} = h(x)$

- Let us define the error or risk or loss as
  - $\ell(\hat{y}, y) = \begin{cases} 1 \text{ if } \hat{y} \neq y \\ 0 \text{ if } \hat{y} = y \end{cases}$

Empirical risk minimization

- The empirical risk or loss of the model h over the whole set S is
  - $L_S(h) = \frac{1}{m}\sum_i^m \ell\,(h(x_i), y_i)$
- The best possible model within $\mathcal{H}$ is one that achieves the smallest empirical loss:
- $h^* = \operatorname*{argmin}_{h\in\mathcal{H}} L_S(h)$

Discussion

- Does h^* have zero loss?
  - No, because h^* is restricted to $\mathcal{H}$
  - May be no model in $\mathcal{H}$ achieves perfect classification

- Can we always find $h^* \in \mathcal{H}$?
  - Often not. The useful classes are frequently too large (e.g. real valued parameters, infinite number of models)

- What is a case for a finite $\mathcal{H}$?
  - Can you think of a situation where we are trying to choose from a few (e.g. 5) possible models?