Generalization

I. Generalization

I.A) Ensemble methods
- aggregate different predictors
  - average all predictions
  - or more advanced ways (use the validation set)
  - in simple case: \( x \sum_{k} f_k(x) / \#k \) (predicted)

[Distilling the knowledge in a Neural Network] "Teacher-student" / "Distillation"
- Hard test
- "Small" network: training \( \Rightarrow \) poor performances
- "Small" network: teacher
- Set of predictions
- Ensemble method (average) \( \Rightarrow \) good performance
- Issue: many networks
  - use lots of space
  - use lots of computational resources
- train a "small" network \( g \)
  - to mimic the ensemble model
  - \( g \) is the training weights

Other ensemble techniques:
- Seeding: combining weak classifiers into a strong one

\( F_e \) \( \rightarrow \) \( F_{e+1} \) : different weights to samples
  - \( g \) weight for sample classified
  - train a weak classifier with these weights
  - agglomerate: \( F_{e+1} = F_e + \omega_g \)

I.B Generalization without regularization
- "double gradient descent"
- \( \text{log}: \sum \| \hat{s}_{ij} - y_i \|^2 \) or cross-entropy
- Search architecture
  - \( n \) neurons/layer
  - Neural Target Kernel
  - CNN model within layers
II. Learning from noisy data

Demisting auto-encoder [ICML 2009]
- dealing with noisy data vs. noise modeling
- originally: \( x \rightarrow \mathbb{R} \rightarrow \mathbb{R} \) auto-encoder
  - usually: narrower middle layer to concentrate information
- set-up:
  - input: noisy image \( x \)
  - \( \text{noise vars} \) if \( x \)
  - \( x^{\text{denoised}} \rightarrow \mathbb{R} \circ \mathbb{R} \rightarrow \mathbb{R} \)
  - goal: reconstruct \( \text{i.e., get rid of noise} \)
  - \( \Rightarrow \) learn more robust features to demisne

Classification with noisy labels
- True distribution \( (x, y) \)
- Given noisy \( (x, \hat{y}) \)
  - sometimes \( y \)
  - sometimes random labels

- If some samples are mislabeled (in the training set) \( \Rightarrow \) not much change of accuracy
- \( \text{significant part} \) of \( \text{train set} \)
  - \( 90\% \) \( \Rightarrow \) still possible to get reasonable results
  - provided that data is available in large quantities
  - what matters is the number of correct labels
    - \( 90\% \) \( \Rightarrow \) \( x \times 10 \) data
    - \( 85\% \) \( \Rightarrow \) \( x \times 100 \) data

Regression with noisy labels
- dataset \( (x, y + \epsilon) \)
- choose iid centered noise \( \epsilon \)

- Simplistic case:
  - always the same update
  - \( \epsilon \) is average over all samples

Loss
- \( \mathbb{E}_i \) \( \ell \) loss
- \( \ell = (y - \hat{y})^2 \)
- \( \text{noisy target} \)
- \( \hat{y} = g(x) + \epsilon \)
- \( \sum_i \left\| \hat{y}_i - y_i \right\|^2 \)
- \( \text{averaging} \) \( \hat{y} \) instead of \( y \)
Classical statistics

\[ N \text{ samples } \Rightarrow [\mu - \sigma] \sim \frac{1}{\sqrt{N}} \]

\[ \Rightarrow \text{W.N. will perform an "average" over similar points} \]

\[ D(x) \]

Let's define "similarity" according to the network:

based on the ability of the W.N. to distinguish samples

\[ \text{Similarity between samples } x \text{ and } x': \frac{\| \hat{f}(x) - \hat{f}(x') \|}{\| \|} \in [0, 1] \Rightarrow \text{within a range} \]

Let's see how does label noise in training set affect prediction at a given test point?

Let's compute a classifying factor

\[ \text{noise } \xrightarrow{\text{predictor noise}} \frac{1}{\text{Factor}} \]

\[ \text{depends on } x \text{ test sample} \]

Understanding deep learning predictions via influence functions, ICML 2017

\[ \text{based on classical statistics} \]

\[ \Rightarrow \text{noise on } x \]

\[ \Rightarrow \text{noise on the sampling distribution: probability to pick } x \]

\[ \text{some level of formulae} \]

\[ \Rightarrow \text{loss instead of } \hat{f}^2 \]

\[ \Rightarrow \text{inversely biased} \]

[Classification:

Learning with noisy labels]

Let's hypothesize the amount of noise

\[ \Rightarrow \text{20\% of labels are inverted (given classifier)} \]

\[ \Rightarrow \text{build a loss to balance} \]

\[ \Rightarrow \text{an average correctness} \]

\[ \Rightarrow \text{as if with true labels} \]

\[ \Rightarrow \text{reduces bias} \text{and improves bound!} \]

\[ \Rightarrow \text{given feat, size of dataset and complexity} \]
Formal proofs

- To prove (formally) that the NN will never mistake

- Express a property → check it

\[ \forall x \in \mathcal{X}, |\hat{y}(x) - y(x)| < \varepsilon \]

- ACAS Xu:

  \[ \text{given: locations, speeds (no perception)} \]

  \[ \text{prove: no collision} \]

- Local properties: locally robust to adversarial attacks

  \[ \text{no adversarial input in this ball} \]

- Keywords:
  - Obfuscation
  - ReLU

  N.N. with ReLU

  \( \Rightarrow \) piecewise-affine F

  \( \Rightarrow \) more replication of traditional powers \( \Rightarrow \) complexity \( \Rightarrow \) neurons

  \[ \text{small network} \]

  3 inputs \( \rightarrow \) few outputs

  1000 neurons