THEORY OF MACHINE LEARNING

LECTURE 22

NTK SUMMARY, REPRESENTATION LEARNING

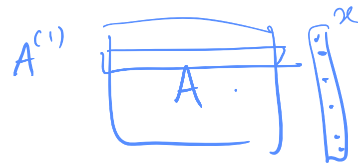
ANNOUNCEMENTS

- **Homework 3 due on Monday April 11**
- **<u>Project presentations:</u>** starting in two weeks! (~18 projects)
  - <u>Dates</u>: April 19, 22, 26 (5 projects /class), couple online
- This week and next: representation learning, robustness

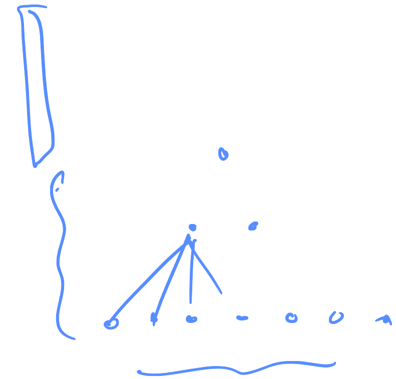
LAST WEEK

$f(A^{(1)}) = b_1$ $\langle x, \cdot \rangle$

$f(A^{(2)}) = b_2$ . . .



x b .



- Learning NNs is hard, often done via gradient descent
  - Topics skipped – "strongest" hardness results based on crypto [Klivans, Sherstov]

    (hardness of improper learning).

intersection of n half-spaces.

- Analyzing gradient descent:
  - Can one analyze dynamics of gradient descent? (for NNs) [can view as Kernel regression for a time-varying kernel]

    re-phrasing.
  - Are there cases where we can reason about resulting solution? [for infinitely wide nets, kernel remains "fixed" – neural tangent kernel]

Law of Large Numbers (Concentration bounds).

NTK REVIEW

Theorem. [Jacot, Gabriel, Hongler 18] [Arora, et al. 2019] A width ~ n^3 network (any number of layers) trained via GD from random initialization achieves zero training error. Moreover, the final solution is equivalent to solving a "Kernel regression" problem with a specific kernel.

f. $K(x,y)$: similarity between x, y.

hypotheses of the form : $\hat{f}(x) = \sum_{i \in \text{training samples}} \alpha_i K(x, x_i)$ — x_i: ith training sample.

- Kernel regression
- Any model training can be viewed as Kernel regression with time varying kernel
- With wide DNNs, kernel doesn't change much!

☆ (GD with tiny step size).

(Can be used to show that if width >> |training data|, then NN training via GD converges to Kernel regression with NTK.)

NTK EXPERIMENTS

$K(x,y) = e^{-\|x-y\|^2}$

x.

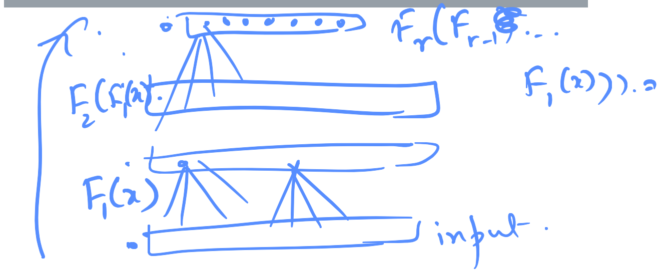
[Arora, et al. 2019] What happens if we forget about NNs, compute closed form for NTK (determined only by number of layers, types of connections, activation function), perform kernel regression?

"vanilla" nn.

Depth	CNN-V	CNTK-V	CNN-GAP	CNTK-GAP
3	61.97%	64.67%	57.96%	70.47%
4	62.12%	65.52%	80.58%	75.93%
6	64.03%	66.03%	80.97%	76.73%
11	70.97%	65.90%	75.45%	77.43%
21	80.56%	64.09%	81.23%	77.08%

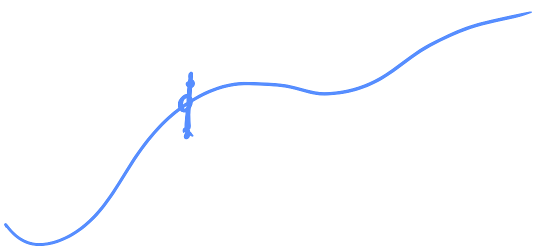
CIFAR-10.

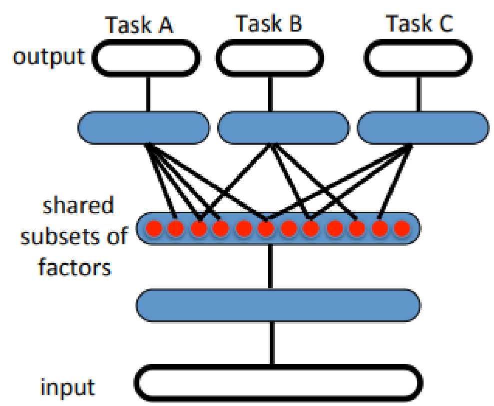
REPRESENTATION LEARNING



- General idea – neural networks are "hierarchical feature extractors"
- Circa 2000s – manual feature extraction (HOG, SIFT)
- NNs embed inputs -> "feature space" (alternative 'representation')

layered NN → builds iterative representations of data.





→ "related" tasks

→ Caption generation.

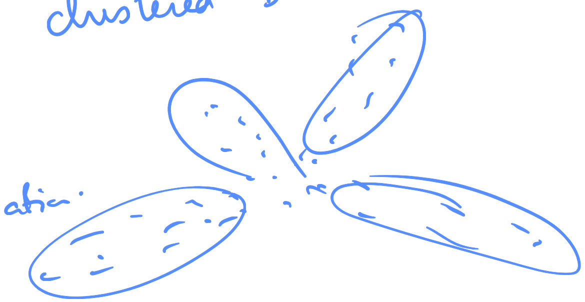
→ [Bengio, etal. 2012]

Fig. 1. Illustration of representation-learning discovering explanatory factors (middle hidden layer, in red), some explaining the input (semi-supervised setting), and some explaining target for each task. Because these subsets overlap, sharing of statistical strength helps generalization.

REPRESENTATION LEARNING

features ⟷ representation.
~~space~~ embedding

- What makes a good representation?
  - Contrastive (for classification) ⤳ in this "feature space", inputs get clustered based on class.
  - "Disentangled" or orthogonal
  - Sparse "explanations" for phenomena classification.
  - Hierarchically organized, explanatory



robustness ...

basis in which inputs have a "sparse" representation.

- Supervised vs Unsupervised → no ta

Crucial if we want to do multiple tasks with same representation.

animal household object boat.

Word embeddings in NLP.

Graph vertex embeddings.

REPRESENTATION LEARNING

given data; no task;
no labels.

- Unsupervised learning of representation
  - Sparse coding / autoencoders (past)
  - Self-supervision (present/ future)

"Meta qn": ~~Under~~ Want to "understand" data...

Formalizing: find common patterns / compression.
succinct representation.

Sparse Coding: Is there a "basis" for data in which all the data points are ~ sparse?
(in the lin. alg. sense).

Inputs: $x_1, x_2, \ldots, x_N \in \mathbb{R}^d$.

Can you find a "basis", ie, $v_1, v_2, \ldots, v_m \in \mathbb{R}^d$ such that every $x_i \approx \sum_j \alpha_j^{(i)} v_j$, for some "sparse" $\alpha^{(i)}$?
(at most k are non-zero).

\# parameters in "input rep": dN

\# parameters in "new" rep: ~~[illegible]~~ $kN + md$

coeffs.

Want $kN + md \ll dN$.

JPEG: based on ~ ideas..

→ Layerwise unsupervised pre-training (2013..).