Lecture #: Topic

Instructor: Aditya Bhaskara *Scribe: Nripesh Pokala*

CS 5966/6966: Theory of Machine Learning

May 4^{th}, 2022

Abstract

In this class we discussed NTK (Neural Tangent Kernel), some NTK experiments and its properties. We also touched up on the general idea of Representational learning, what makes a good representation and supervised vs unsupervised representation.

1 NTK

Theorem :

A network width of approximately n^3 trained by a GD from random initialization achieves zero training error.

Arriving at the final solution, this is very similar to solving a "Kernel Regression" problem with a specific kernel.

Before we discuss the theory of NTK, let us understand Kernel Regression first,

which can be elaborated as finding some f when we have similarity function K where K(x,y) gives us the similarity between x and y and a training sample t, then the hypothesis is given as,

$$f' = \sum_{i \in t} \alpha_i * K(x, x_i) \tag{1}$$

Any training model can be viewed as Kernel Regression with time varying kernel.

The kernel does not change much with wide DNNs, this further implies that if width drastically higher than the size of the training data set, then NN training via gradient descent converges to kernel regression with NTK.

The afore mentioned result holds true if and only if we train in with GD with very small step size.

2 NTK Experiments

some experiments were conducted with regards to certain parameters effecting the solution like what happens if we forget bout NNs, compute close form NTK, which is only determined by number of layers , types of connections and activation function, i.e, thinking about it more like a similarity function than something which came from an neural network.

Here are the behaviours expressed at varying depths, we can see that we get reasonably good performances even with a vanilla kernel were where simply calculating some alphas.

This imposed advantage was further leveraged and improved upon by incorporating some hacks to it.

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%

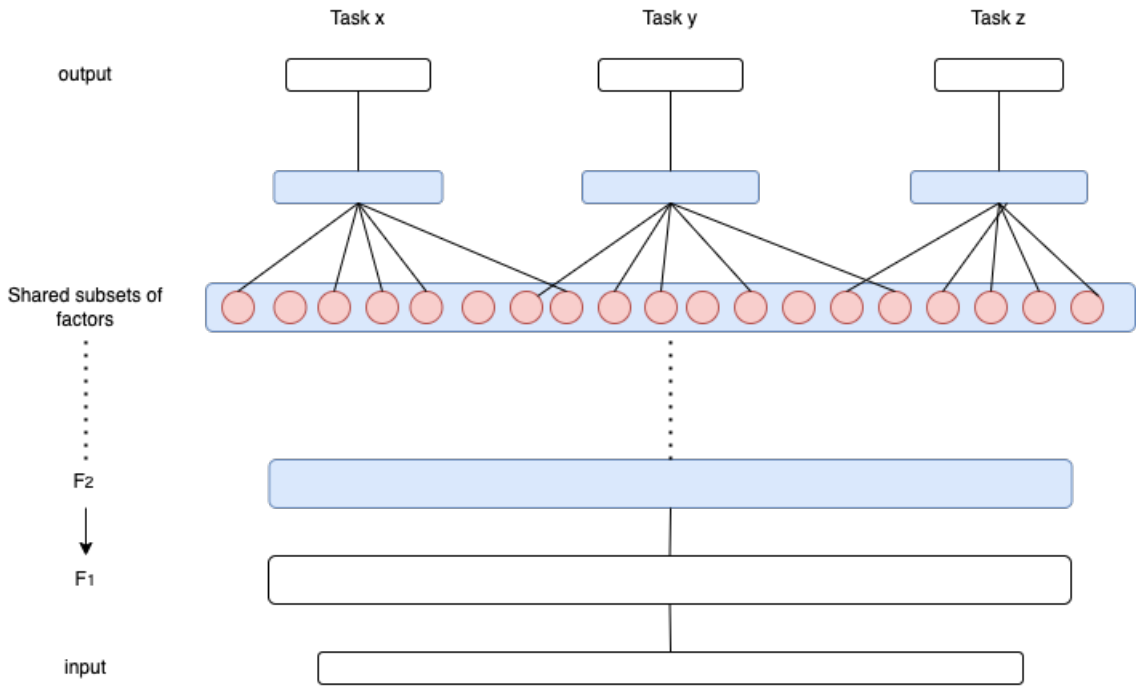
Here CNTK-V is the vanilla kernel.

3 Represntational Learning

3.1 definition. "Representation learning is a class of machine learning approaches that allow a system to discover the representations required for feature detection or classification from raw data. The requirement for manual feature engineering is reduced by allowing a machine to learn the features and apply them to a given activity."

The general idea here is that neural networks are "hierarchical feature extractors". Layered neural networks build layered iterative representations of data.

Here is an illustration of representation-learning, depicting explanatory factors (middle hidden layer in mild red) some explaining the input (semi supervised setting) with some explaining target for each task.



This shows that the subsets overlap, sharing of statistical strength helps generalization.

Properties that make a good representation are as follows,

- If we want to use this representation for classification or in the feature space where inputs get clustered based on class, the representation should have a great contrastive property.

  There is ofcourse a challenge here given what is contrastive for task may not always be contrastive for another class.

- Each individual portion of the representation are supposed to be orthog-

onal and disentangled to one another thus avoiding redundancy in the features.

- Sparse explanations for the tasks supported by the representation.

- Hierarchical organization.

- Robustness. This ofcourse improves the versatility but then hinders the parity of the model.

4 Supervised and Unsupervised Representational Learning

4.1 *Supervised*

Here the Representational Learning happens in a supervised or guided environment, where we know which data corresponds to which data, may it be speech, images or text.

If we consider a coarse grained example of classification, supervised learning may not offer much flexibility or versatility among a set of tasks.

Examples where this excels are,

- Word Embedding in NLP.
- Graph Vertex Embeddings.

4.2 *Unsupervised*

Here the Representational Learning happens in a unsupervised or unguided environment. Essentially we are to find some representation for unlabelled tasks.

This way of learning excels when we expect some extent pf generalization over a set of tasks.

It is also crucial in the sense that it can supports multiple tasks with same representation.

This mode of learning has recently become very active interms of research and development.

The meta question here is that how we could understand some data without any label nor a task, then the two aspects of unsupervised learning of representation are,

- Sparse coding / auto-encoders : looking for a ”basis” with in the data over which all the data points are approximately sparse.

  consider and example where we have the inputs $x_1, x_2, ..., x_n \in R^d$,
  if we could find a basis for R^d which can be represented as,
  $v_1, v_2, .., v_m \in R^d$,
  such that for every $x_i \approx \Sigma_j \alpha_j^{(i)} v_j$, fopr some ”sparse” $\alpha^{(i)}$.

  Here, the goal is to find a basis under which every input input can be deduced as a sparse combination.

- Self-supervised : self-supervision can be declared as the present and the future of the representational learning. also refers to a notion of iterative layer wise learning whoich was recently termed as ”layer-wise unsupervised pre-training”.
- formalizing : being able to find common patterns with in the data and compressing the same, we can call it a succinct representation of the data.