Lecture # 23: Representation Learning

Instructor: Aditya Bhaskara *Scribe: Carson Storm*

CS 5966/6966: Theory of Machine Learning

April 7th, 2022

Abstract

1 NTK Recap

For certain classes of functions, gradient descent can be reasoned about in terms of Kernel regression with a time-varying kernel, and for a wide network, the kernel remains roughly "fixed".

2 Representation Theory Recap

2.1 *Good Representations*

What makes for a good representation? There is not a standard metric for a good representation, but some useful properties for representations are:

Early worked sought representations that were "Auto-encoders", i.e., the representation encoded enough information to "approximately recover the input."

- **Contrastive:** For classification, inputs are clustered in "feature space" based on class.
- **Orthogonal:** Features are mostly disjoint from each other or "orthogonal."
- **Sparse:** Features are spares, *i.e.* there are not many important features.
- **Hierarchical organization:** Features identified at different layers are organized in a hierarchy.
- **Domain Knowledge:** Representation captures domain knowledge, *i.e.*, model should have learned domain knowledge. For example a fluid simulation model should learn about gravity.

2.2 *Supervised vs Unsupervised Representation Learning*

Unsupervised learning is crucial if the goal is to multiple tasks with the same learned representation. Features give important insights into what an unsupervised model is learning.

3 Unsupervised Representation Learning

The classical approach to unsupervised representation learning is manual feature encoding, auto-encoders and sparse coding. The goal of Sparse Coding/Auto-encoders are to find a more succinct representation of inputs by finding common patterns in data. In other words is there a near "basis" for data in

which all of the data points are roughly sparse. Formally is there a "basis" $v_1, \ldots, v_m \in \mathbb{R}^d$ such that for each input $x_1, \ldots, x_n \in \mathbb{R}^d$

$$x_i \approx \sum_j \alpha_j^{(i)} \cdot v_j$$

for some "sparse" $\alpha^{(i)}$ (at most k coordinates are non-zero). Want $kn + md \ll dn$.

The more modern approach is to use self-supervised learning, invariants and data augmentation. Self-supervised learning generates "pseudo-labels" from training data, then learns these labels. Then features are extracted from the network.

4 Representation Learning in NLP

Most approaches are based on Firth's hypothesis: the meaning of a word is defined by "the company it keeps." The idea is to look at n-grams sequences of words and ask for each n-gram word w_i, how frequently does it occur near some other word w_j? We can then construct a matrix (α_{ij}), where α_{ij} represents the frequency for which w_i occurs near w_j. If we then perform SVD on (α_{ij}), to get

$$(\alpha_{ij}) = \underbrace{U}_{N \times k} \cdot \underbrace{V}_{K \times N}$$

then extract the embedding of a word from the factors to get $\alpha_{ij} \approx \langle u_i, v_j \rangle$.

5 Representation Learning in Graphs

Likewise for a graph, we can represent the graph as an adjacency matrix and the use SVD to factor the adjacency matrix. This can be useful when a graph represents relationships or similarities between nodes.