Project Proposal: Neural Nearest Neighbours

Sheng Zhong

February 12, 2019

1 Paper description

Neural Nearest Neighbors Networks by Tobias Plotz and Stefan Roth is the paper of interest. Its contribution is formulating a differentiable way of approximating k-nearest neighbour (KNN) selection (for each data point, find the k-nearest other data points under L2 norm). KNN is useful at the start of machine learning pipelines to collect global information, or at the end of some architectures as a non-parametric classifier. Its relegation to the start and end of pipelines is because gradients can't be passed through it for gradient descent. KNN as an intermediate operation is theorized to be useful because it collects global information while most vision operations are local (max pool, convolution). They demonstrated that in certain domains such as images, data self-similarity such as a part of the image looking similar to another part could be exploited for better performance on denoising applications.

2 Replication/Extension Interest

The authors are in a lab focused on computer vision. I am interested if the same algorithm could be adapted to other domains that also has high degrees of self-similarity. Specifically, I want to consider the problem of finding an embedding for books to enable book recommendations. Embedding is the reduction of some high dimensional data down to a smaller space (for example a vector of 50 real numbers) such that close elements in this embedding (according to some distance) are also similar in the full space. For books, this could mean similar writing style, content, or author. My hypothesis is that this method could lead to significant improvements without too much parameter tuning (evaluation metric proposed in the next section). A tutorial on this problem is covered here: neural-network-embeddings-explained. The general problem of generating an embedding is crucial to most recommendation systems, an area of key commercial interest.

3 Replication/Extension Plan

PyTorch code for their algorithm is available on github, which I've already downloaded and started testing with. I plan to work entirely in python using PyTorch 1.0+. Their non-vision code seems poorly tested because it failed on a sanity check data set I designed, so using

their code out of the box is suspect. I contacted the author for resolution on the issue, but lacking a reply I will reimplement their code from scratch.

Following that I will apply KNN somewhere along the pipeline for book recommendation through embedding. The tutorial and data for the book recommendation project is also on github. The tutorial proposes no quantitative way of evaluating the generated embeddings, and relies on visualization of the dimension-reduced (marginalized) embedding space to judge clustering effectivness. I propose to use externally labelled discrete properties (author, language, genre). Looking at a single property at a time, we evaluate the average embedding space distance of books with the same property value against those with a different property value. We wish to minimize this ratio across all properties - we want dense clusters that are far apart from each other.

With this quantitative evaluation of cluster-ness, I will compare the performance of inserting a KNN inside the pipeline against not. Additionally, I will produce embedding visualizations such as Figure 3.