Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering

Authors: Michael Defferrard, Xavier Bresson, Pierre Vandergheynst

Introduction

Prior to this work, convolutional neural networks (CNN) have been used extensively to low-dimensional regular grids such as images, video and speech. In this work, the authors propose to extend the use of CNNs to high-dimensional data represented by graphs. The convolution and pooling operators are defined only for regular grids, and extending them to graphs is not straightforward. The authors use tools in graph signal processing to build a spectral graph formulation of CNNs. Furthermore, the authors establish a low computational linear complexity, owing to an efficient pooling strategy on graphs. The authors test the performance of the proposed mechanism on the MNIST dataset and the 20NEWS dataset.

Proposed Technique

The convolutional filters can either be built by a spectral approach or a spatial approach. In the graph convolution technique, one major challenge is matching the local neighborhoods. A filter defined in the spectral domain is not naturally localized. To enable this localization, a a translation operator is proposed which in the vertex domain. However, there are limitations to this approach, in that, they are localized in space, and have a learning complexity of $O(n)$. To further overcome these limitations, a polynomial filter is proposed where te kernel is localized via convolution. This limits the complexity to $O(k)$, where, 'K' is the support size of the filter, and thus has the same complexity as CNNs.

The calculations involved in learning filters boil down to K sparse matrix-vector multiplications, thus leading to efficient computation over parallel architectures and faster training times. Meaningful neighborhoods on graphs is provided by the pooling operation in the form of coarsening in the multilevel clustering algorithms. This algorithm uses a greedy approach to compute successive coarser versions in a given graph, thus, ensuring maximal neighborhood estimations. Considering the high frequency of the pooling operations, a fast pooling algorithm is also proposed. In this, after coarsening, the vertices of the input graph and it's coarsened versions are not arranged in any meaningful way. This leads to the formation of disconnected nodes, which ensures that filtering does not impact the initial neutral values at these nodes. This technique makes the algorithm very efficient, alongside satisfying parallel architectures.

Experimentation

In this technique, the authors use the Graculus coarsening algorithm which computes the successive coarser versions of a given graph greedily. In the network architecture, all fully connected layers,

convolutional layers and graph convolutional layers are followed by ReLU activation units, alongwith a softmax layer in the final layer. For the MNIST classification problem, the authors construct an 8-NN graph of the 2D grid which produces 976 nodes and 3198 edges. In Table 1, the difference in accuracies between the classical CNN and the proposed CNN on the MNIST dataset is attributed to the isotropic nature of the spectral filters. Performance of a LeNet-5-like network architecture with default hyperparameters from the TensorFlow MNIST tutorial is also recorded.

The technique is also applied to the unsupervised text categorization problem on the 20NEWS dataset associated with 20 classes. The model is tested with a 16-NN graph with word embeddings for each of the words, thus producing a graph with 10,000 nodes and 132,834 edges. The architecture GC32 was used by the authors, the performance of which is recorded in Table 2. It can be seen that the model does not outperform the multinomial Naive Bayes classifier, but performs better than the fully connected networks.

The quality of the learned filters and the classification performance depends on the quality of the graph. For data lying on Euclidean space, a simple kNN would sufficient to exact the performance of standard CNNs, while for images, a grid graph needs to be built; a feature graph needs to be built for text documents. In Table 6, the authors report the clasification accuracies of GC32 architecture with different graph constructions of the 20NEWS dataset, with a learned word2vec graph construction performing the best.

Strengths & Weaknesses

One of the major strengths of this paper is that the mechanism provides control over the local support of filters required to generalize CNNs to graphs. Another strength of this work is that it addresses the major computational deficiencies in the previous works related to graph spectral filtering. One of the weaknesses with the technique proposed in this work is that it can not be scaled to large graphs, and are not designed for whole-graph classification. [1]. One other major weakness as pointed out by Velickovic et.al [2] is that this technique is dependent on the Laplacian eigenbases, which depends on the graph structure, thus proving that the model is trained on a specific structure of the graph, and cannot be applied to those with different structures.

Further Work

As a further improvement on the current technique, Hamilton et.al propose the inductive representation learning on large graphs, which generalizes to completely unseen graphs using a multi-graph dataset. Furthermore, Monti et al [3] propose geometric deep learning which generalizes the deep learning methods to non-Euclidean structured data, and outperforms all other graph processing techniques.

References

[1] Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." Advances in Neural Information Processing Systems. 2017.

[2] Veličković, Petar, et al. "Graph attention networks." arXiv preprint arXiv:1710.10903 (2017).

[3] Bronstein, Michael M., et al. "Geometric deep learning: going beyond euclidean data." IEEE Signal Processing Magazine 34.4 (2017): 18-42.