CNN to Graph and Graph to CNN Interconnects

Amit Lohan
Department of Electrical Engineering
Indian Institute of Technology, Bombay
amitlohan@iitb.ac.in

Amit Sethi
Department of Electrical Engineering
Indian Institute of Technology, Bombay
asethi@iitb.ac.in

June 10, 2021

Abstract

A Graph Neural Network (GNN) requires its inputs to be in the form of a graph. To use a GNN for inference on data which is not inherently structured as graph, a transformation of data structure may be required. More specifically, to process data such as images using a GNN, it may be required to transform the image or image feature maps (After some processing with a CNN) to a graph, to make it a valid GNN input. Similarly, to process the graph structured data using a CNN, a graph to a meaningful 3D tensor transformation may be required while preserving the possibility of gradient flow for backpropagation. In this work, two such interconnects i.e, a GNN to CNN and a CNN to GNN interconnects are presented.

1 Introduction

In addition to the node features, graphs also have the connections information, whereas the CNN feature maps are represented as 3 dimensional tensors with the inherent spatial arrangement. An interconnect between a CNN and a GNN has to take care of transformation of the spatial arrangement to edge weights and vice versa in addition to assignment of features from nodes to spatial locations and vice versa.

In case of an interconnect where the entire 3D feature maps are required to be converted to a graph, every feature vector at a location (i, j) can become the feature vector of a node in the graph, i.e a tensor of size $N \times M \times L$ will produce a graph with $N \times M$ nodes with each node having a feature vector of length L. The inverse of distance between each pair of location (i, j) in the feature map or any other measure of connectivity can become the edge weight of connection between corresponding nodes. This way, for feature maps of size $N \times M \times L$, size of adjacency matrix would be $(NM) \times (NM)$. Similarly, an inverse transformation can be applied if location of each node as in the original 3D tensor of feature maps is tracked by rearranging the nodes features in the form of a 3D tensor using a suitable way for spatial assignment.

In case of an interconnect where only a handful of locations from the 3D tensor of feature maps are required for constructing a graph, for example in case processing features only from key-points in images, a slightly different approach than discussed in above paragraph is required. Two such interconnects for key-points based processing using a hybrid model of GNNs and CNNs are discussed in Sections 2 and 3.

2 CNN to Graph Interconnect

To transform the output feature maps of a convolutional layer, at any stage in CNN, into a graph, two pieces of information are required:

- Node features (The features from CNN output, which will become the features of Nodes of the graph).
- Connections information between the nodes.

In case of a key-points processing based CNN-to-graph interconnect, as shown in Fig. 1, the feature vectors belonging to the key-point locations become the features of nodes and the inverse of distances between every pair of these key-points

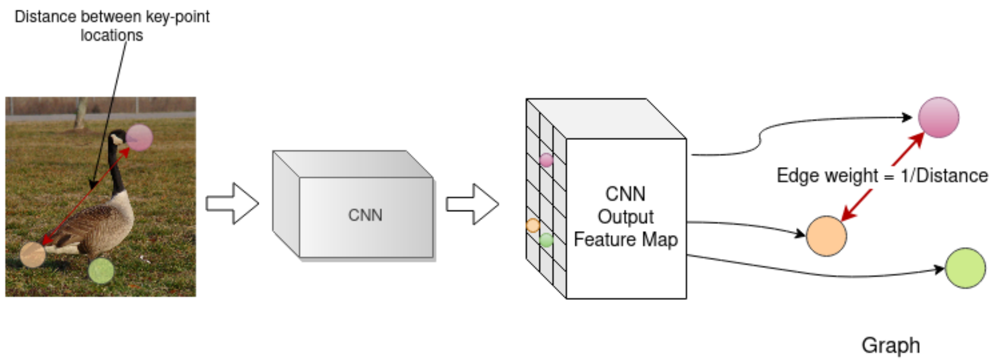

Figure 1: Sample figure caption.

becomes the strength of interconnection between the corresponding pair of nodes in the graph. This way if there are N key-points then the adjacency matrix will have dimensions of $N \times N$.

The implementation of this type of interconnect is available at:
`https://github.com/amitlohan/nn_interconnects`

3 Graph to CNN Interconnect

For transforming a graph into a tensor of CNN feature maps, we require node feature vectors and a way to arrange them spatially in a grid. To arrange the features correctly into a 2D grid, the original key-point locations are required. As shown in Fig. 2, first of all we take a 3D tensor of zeros with dimensions $N \times H \times W$, where N is the size of feature vectors of the nodes of graph, and W and H are respectively the width and height of the CNN feature maps to be obtained, then the features are assigned to the 2D locations as given by original locations of key-points which was used for conversion of feature maps to graph. This way a valid CNN features map is obtained.

The implementation of this type of interconnect is available at: `https://github.com/amitlohan/nn_interconnects`

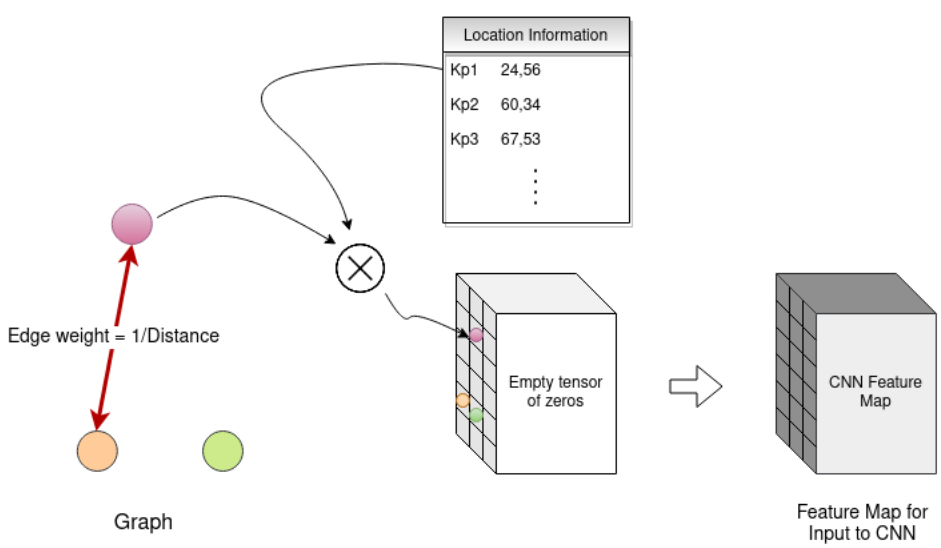

Figure 2: Sample figure caption.

References

[1] Mateusz Bednarski, Understanding indexing with pytorch gather, https://medium.com/analytics-vidhya/understanding-indexing-with-pytorch-gather-33717a84ebc4