Embedding Spatial Vector Data using LBANN

Anmol Paudel 1, Brian Van Essen 2

1Marquette University, Milwaukee, WI, USA, 2Lawrence Livermore National Laboratory, CA, USA

INTRODUCTION

Spatial data stores information about location, position, space. In Geographic Information Systems (GIS), geo-spatial data stores information about the position and trace of elements like rivers, buildings, roads, counties, lakes etc. These data are primarily available in 2 forms: Raster and Vector. Raster data are basically like pixeled images so they are of fixed size. Vector data, however, are a collection of points/polygons/shapes. So, the data referring to a shape of a river can have a lot of points to trace it out. Vector data are extremely useful because it allows us to do more refined and minute computations.

PROBLEM

Neural networks usually have a fixed size input layer that takes a certain fixed dimension of data. But a polygon representing a spatial element could look like the following

POLYGON ((-87.866893 32.825274, -87.863009 32.815742, -87.863257 32.815652, -87.867142 32.82521, -87.866893 32.825274))

and another polygon could look like the following or bigger

POLYGON ((-87.906508 32.896858, -87.906483 32.896926, -87.906396 32.897053, -87.906245 32.897233, -87.90618 32.897281, -87.906101 32.897303, -87.90594 32.897277, -87.905873 32.897224, -87.905743 32.897125, -87.905635 32.897063, -87.905471 32.896967, -87.905454 32.896497, -87.905432 32.896858))

CURRENT APPROACH

Of the different approach to deal with variable length input sizes the simplest approach would be to use zero padding to make the sizes of all the inputs the same. However, if there is very high variability in the sizes or some outliers with extremely large sizes, this would unnecessarily increase space and computation. To avoid this, another approach would be to encode the data into fixed sized vectors. In this way, we could use fixed encoding size to keep the space low yet large enough to capture all the necessary feature of the data. For our spatial data, we are using a graph based approach called node2vec[1] to map our data to a low-dimensional space of features.

REFERENCES


TRANSFERABILITY/APPLICATION

Our work on encoding/embedding variable length data into neural networks can be extended to use with various other types of data in other domains. For example, in the DSSI challenge we had process SMILES (Simplified molecular-input line-entry system) which are used to represent the structures of a molecule. Depending on the structure and size of the molecule, SMILES vary in length too. But being able to encode and vectorize the SMILES in our dataset, the SMILES could then be easily used by a neural network for further analysis.

SUMMARY/FUTURE

In our work, we are trying to encode variable length data into the neural network so that it can be used for further processing. In doing so, we would like to create a x2vec framework for LBANN, where x:(word,string,spatial data, graphs, documents, SMILES or any variable length sized input).