FINAL REPORT: ACTIVATION FUNCTIONS IN GRAPH NEURAL NETWORKS

ACS 4511: CORE APPLICATIONS OF ARTIFICIAL INTELLIGENCE

Gobind Puniani

Wednesday 15th December, 2021

Abstract

Activation functions are often overlooked in the tuning of hyperparameters in neural networks. Most deep learning researchers and engineers simply use the most popular activation functions, such as ReLU, without much consideration. In my senior capstone project, I investigated the performance of five different activation functions (ReLU, Swish, Mish, TAct, and mTAct) in many different neural network architectures in the context of image classification. In this project, I continue the investigation into the performance of activation functions – but now in the context of graph neural network architectures. The task is a relatively simple binary classification using a molecular odor dataset. After running three trials for each activation function, it seems that all of the five activation functions performed roughly the same in test accuracy in this task.

1 Introduction

A Graph Neural Network (GNN) is an artificial neural network that takes in graph data as input and performs operations on the graph attributes while preserving the connections. This means that the output of GNNs sometimes appears the same as the input data. In these cases, the input graph and the output graph possess the same nodes and connections, but other graph data are changed. For example, node data (values or vectors associated with the nodes) or edge weights may change after passing through a GNN.

Although images and text *can* be represented as graphs, they are not ideal for GNNs because they have intrinsic regularity in their structures. Images can be represented as arrays of pixels, and text can be represented as matrices (such as Bag-of-Words or via other techniques). Graphs are best suited for data with irregular structures.

- Social networks are prime examples of graphs in action, as each node represents a user and each edge represents a friendship/connection between users. Similarly, citation networks can model the citation of scientific papers in other papers, so each node represents a paper and a directed edge represents a citation of that paper.
- Chemical molecules are ideal for graphical representations. In fact, common depictions of molecules often resemble graphs. Molecules have highly varied spatial arrangements of their constituent atoms, so each node can represent an atom and each edge can represent a chemical bond.
- In computer vision, keypoint detection lends itself to graphical representation, as fixed points (represented as nodes) are connected with lines (represented as edges) to form stick figure-like diagrams.
- Dataflow graphs are mathematical or programming algorithms encoded as a data structure. Each variable is represented as a node, and each operation between nodes is represented as an edge.

There are many applications of GNNs, such as recommendation systems, traffic prediction, anti-bacterial discovery, fake news detection, and physics simulations.

2 Background

At my previous university, my senior capstone research project was "Exploring the Role of Activation Functions in Deep Learning". The goal of this project was to determine whether the choice of activation function significantly affects the performance of that neural network. This was important to investigate because most developers of neural network architectures don't give too much consideration to the activation function used. Generally, such developers opt for the most popular activation functions, such as ReLU. This is yet another example in deep learning of a hyperparameter tuned arbitrarily rather than with precision and intention. This capstone project was designed to shed some light on which activation function to use for a given task and model. We tested five different activation functions: ReLU, Swish, Mish, TAct, and mTAct. Mish is a modified version of Swish. TAct and mTAct (modified TAct) are two custom activation functions designed by my research advisor to interpolate between three popular activation functions during training: ReLU, Swish, and Tanh. Despite testing these activation functions on a multitude of neural network architectures on the CIFAR-10 and CIFAR-100 datasets for the task of image classification, the results were inconclusive. It appeared that there was no clear pattern between activation function and architectural feature or other hyperparameter, since no activation function could consistently outperform the others on these dimensions.

For this project, I wanted to apply the same goal of comparing the performance of activation functions, but now in the context of graph neural networks. Thus, this project is an expansion of my senior capstone research project.

3 Task

The goal of this project is to determine whether the choice of activation function significantly affects the performance of a GNN.

We begin with a task of fairly small scope first so that we can add complexity later after initial trials. This project was inspired by a Medium post [1] about an implementation of a solution to a challenge from AIcrowd [2]. The challenge was to predict the odors associated with a given molecule using a variation of the Leffingwell Odor Dataset. The input data consisted of SMILES (Simplified Molecular-Input Line-Entry System) data, which uses ASCII-string notation to describe molecular structure, and the output data consisted of strings of odor names. This challenge originally involved multiple classification, in which the model would need to identify all odors associated with a particular molecule. To simplify the scope of this project, the task was changed to binary classification, so the model's assignment was to predict whether a fruity odor was associated with a molecule or not.

First, we modify the labels to reflect the task change to binary classification. Since the labels are binary, we can encode the labels with 0 and 1: 0 for the absence of a fruity odor, and 1 for the presence of a fruity odor. Next, we convert the SMILES objects into DGL (Deep Graph Library) graph objects, extract the feature vectors (if possible), and then apply the new labels to each graph object. This then populates our custom DGL dataset. Our dataset is formatted as a DGL dataset to make data loading and splitting easier. Finally, we create five different versions of our GNN model: one version for each activation function. We then run each model three times each for 20 epochs to arrive at an average test accuracy for each activation function.

4 Results

Table 1: Results for all Activation Functions

Activation Function	Test accuracy 1	Test accuracy 2	Test accuracy 3	Average test accuracy
ReLU	0.7825581395348837	0.7802325581395348	0.7802325581395348	0.7810077519379846
Swish	0.7790697674418605	0.7825581395348837	0.786046511627907	**0.7825581395348836**
Mish	0.7790697674418605	0.7779069767441861	0.7825581395348837	0.7798449612403102
TAct	0.7802325581395348	0.7802325581395348	0.7825581395348837	0.7810077519379846
mTAct	0.7825581395348837	0.7790697674418605	0.7825581395348837	0.7813953488372093

In the results table above, we see that all activation functions perform roughly equivalently to each other over three distinct runs. Swish appears to have a slight, negligible lead over the others in average test accuracy. It seems that no activation function was able to surpass 0.783 in test accuracy for any run.

5 Conclusion

Just as with my original senior capstone project, this project did not yield a clear winner among the activation functions tested. From these results, it appears that the choice of activation function does not make a significant difference in the performance of our simple GNN in the task of binary classification for molecular odors using a variation of the Leffingwell Odor Dataset.

However, this project so far is simply the first foray into the investigation of the role of activation functions in GNN architectures. Now is the time to expand the scope of this project, with some ideas for this discussed in the next section.

6 Future Work

This project only examined a handful of activation functions in the context of one small task with a relatively simple GNN structure. Thus, there are many possibilities to explore if this project were to be expanded further. Just within DGL, there are so many other GNN architectures that can tackle many other tasks. The odor classification task used in this project was chosen somewhat arbitrarily; it was used here because its architecture was relatively simple and employed an activation function, plus it was already paired with a nice dataset from the AIcrowd Learning to Smell Challenge. Furthermore, the scope of this project was reduced even further by reframing the task as simple binary classification (fruity or not fruity) instead of the original multiple classification task of identifying all associated odors for a given molecule.

Additionally, the rigor of comparison between the different performances is another area of potential improvement. Comparing the performances between activation functions over the entire training period instead of only after might reveal some more information, especially if visualized. Since this was a classification task, it might also be helpful to have more statistical data besides accuracy, such as precision, recall, etc.

References

[1] Learn to smell (molecules) with graph convolutional neural networks: An end to end project combining chemistry and deep learning on graphs. https://towardsdatascience.com/learn-to-smell-molecules-with-graph-convolutional-neural-networks-62fa5a826af5. Published: 2021-08-20.

[2] Learning to smell challenge: Predicting smell of molecular compounds. https://www.aicrowd.com/challenges/learning-to-smell. Accessed: 2021-11-28.