

1 Programming Assignment – Notes

Notes from Programming Assignment

Tool	Comment
Programming Assignment 1	<p>Forces familiarity with:</p> <ol style="list-style-type: none"> 1. The page rank algorithm through the implementation of the function from scratch. 2. Familiarity with NetworkX through the use of the functions within this library.
Tensors	<p>Specialized data structures used in PyTorch that are very similar to matrices and arrays. It is used to encode inputs and outputs of a model, as well as a model's parameters.</p> <p>Tensors can run on GPUs and other specialized hardware to accelerate computing.</p> <p>Source: https://pytorch.org/tutorials/beginner/blitz/tensor_tutorial.html</p>
Programming Assignment 2	<p>Forces understanding of graph convolution networks (GCNs).</p> <ul style="list-style-type: none"> - Convolutional layer: performs a convolution, which is a multiplication of a set of weights with the input (akin to traditional neural network). The 2d set of weights used to multiply the inputs with is called a kernel or filter. Of note: the filter is smaller than the input data. The type of multiplication performed is a dot product. The filter being smaller than the input allows for it to be systematically applied to the input data – if the filter is a feature detector, this is akin to allowing the filter the opportunity to detect the feature across the entire image. This is a technique that is common in computer vision. The innovation behind using this technique in machine learning is that the weights or values of the filter are learned during the training process. <p>[cite Machine Learning Mastery – Convolution in Convolutional Neural Networks]</p> <p>Challenge:</p> <ul style="list-style-type: none"> - Having the right physical environment e.g. this assignment could not be completed on Google Colab without incurring extra cost because it required greater computing power in the form of graphics processing units (GPUs). - Hence, required implementation on non-cloud resources which also necessitate software and hardware configurations which is a non trivial undertaking. <p>Great discovery from this assignment is:</p> <ul style="list-style-type: none"> - iPywidget: interactive HTML widgets for Jupyter Notebook and the IPython kernel. Can build interactive HTML widgets that will display in the Jupyter Notebook. Great for interactive, immersive learning. From a research perspective, it allows for understanding how parameters affect outputs - a good video demonstrating this can be found here: https://www.youtube.com/watch?v=wb6k_T4rKBQ.
Programming Assignment 3	<p>Assignment 2: created a GNN model using the GCNConv layer which is built into PyTorch Geometric (PyG). This assignment looks at another type of layer: GraphSAGE [citation]. Specifically, this assignment performed node-level classification on the CORA¹ dataset.</p> <p>Required need to understand:</p>

¹ The CORA dataset is a citation dataset comprised of 2708 scientific publications classed into one of 7 prediction classes, 5429 links and 1433 features per node. Source: Colab4.

Tool	Comment
	<ul style="list-style-type: none"> - Message passing. 3 crucial functions required: <ul style="list-style-type: none"> o Forward o Message o Aggregate - Message passing: 3 steps applied to every node in a graph: <ul style="list-style-type: none"> o Given a central node, each neighbouring node passes its current message to the central node. o The central node aggregates the messages from its neighbour (e.g. through a sum or mean) o The aggregated information is transformed by applying linear or non-linear transformations. - Message passing layer: role is to update current feature representation/embedding of each node in a graph. This is what the forward function does. <p>What I learnt:</p> <ul style="list-style-type: none"> - The difference between GraphSAGE and GCN: <ul style="list-style-type: none"> o GCN aggregates messages using the mean. GraphSAGE generalizes the aggregation function. [explain what that means better. Useful website: https://dsgittr.com/blogs/graphsage/]
Programming Assignment 4	<p>This assignment introduces another type of layer – Graph Attention Networks (GATs) [citation], again on the CORA dataset. Of note:</p> <ul style="list-style-type: none"> - The attention mechanism which has become state of the art in many sequence based tasks such as machine translation and learning sentence representations. - Where GATs vary from GraphSAGE is in: <ul style="list-style-type: none"> o The aggregation function. GATs use a graph attention layer which outputs weights corresponding to each input which can be interpreted as the importance of that node, rather than a mean or sum of the inputs such as is the case with GraphSAGE. - Rather than a standard attention mechanism which applies to all nodes (which effectively ignores the graph structure), only attention coefficients corresponding to nodes within some neighbourhood of the central node is computed. - Why do we normalized coefficients? to allow the coefficients to be compared across different nodes (by standardizing their values to sit between 0 and 1). - Multi head attention mechanism: <ul style="list-style-type: none"> o what is it? Where the self-attention mechanism is performed multiple times. o o Why? To stabilize the process of self-attention (huh?) o
Programming Assignment 5	<p>Learnt how to use DeepSNAP:</p> <ul style="list-style-type: none"> - What is DeepSNAP? Python library for deep learning on graphs. - How does it differ from other libraries e.g. PyTorch Geometric, DGL and GraphNets? <ul style="list-style-type: none"> o More flexible graph manipulation o Support for heterogeneous graphs

