Graph Neural Networks

Juan Elenter, Ignacio Hounie,
Charilaos Kanatsoulis, and Alejandro Ribeiro*

March 15, 2024

1 Graph Neural Networks

Graph Neural Networks (GNNs) are information processing architectures made up of a composition of layers, each of which is itself the composition of a linear graph filter with a pointwise nonlinearity.

For a network with a given number of layers L we define the input output relationship through the recursion

$$\mathbf{X}_\ell = \sigma\Big(\mathbf{Z}_\ell\Big) = \sigma\left(\sum_{k=0}^{K_\ell} \mathbf{S}^k \mathbf{X}_{l-1} \mathbf{H}_{\ell k}\right), \tag{1}$$

In this recursion the output of Layer $\ell - 1$ is $\mathbf{X}_{l-1}$ and it is recast as an input to Layer ℓ. In this layer, the input $\mathbf{X}_{l-1}$ is processed with a graph filter to produce the intermediate output $\mathbf{Z}_\ell$. The coefficients of this graph filter are the matrices $\mathbf{H}_{\ell k}$. This intermediate output is processed with a pointwise nonlinearity σ to produce the output $\mathbf{X}_\ell$ of Layer ℓ. That the nonlinear operation is pointwise means that it is acting separately on each entry of $\mathbf{Z}_l$.

To complete the recursion we redefine the input $\mathbf{X}$ as the output of Layer 0, $\mathbf{X}_0 = \mathbf{X}$. The output of the neural network is the output of layer L,

*In alphabetical order.

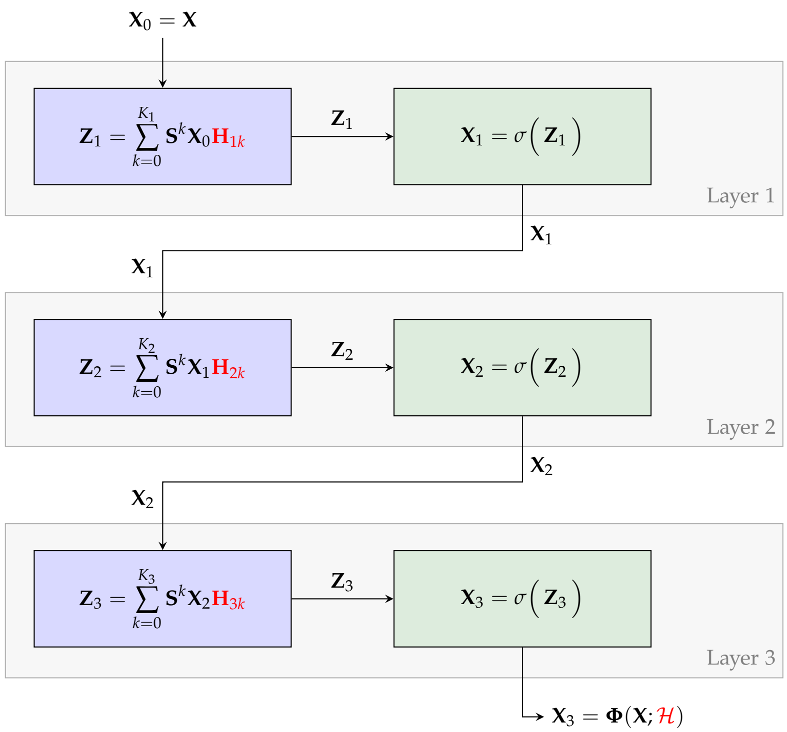

Figure 1. A Graph Neural Network (GNN) with three layers. A GNN is a composition of layers, each of which is itself the composition of a linear graph filter with a pointwise nonlinearity. [cf. (1)].

$\mathbf{X}_L = \mathbf{\Phi}(\mathbf{X}; \mathcal{H})$. In this notation $\mathcal{H}$ is the tensor $\mathcal{H} := [\mathbf{H}_{11}, \ldots, \mathbf{H}_{LK_\ell}]$ that groups all of the filters that are used at each of the L layers.

A graph neural network with three layers is depicted in Figure 1.

1.1 Graph Neural Network Specification

To specify a GNN we need to specify the number of layers L and the characteristics of the filters that are used at each layer. The latter are the number of filter taps K_ℓ and the number of features F_ℓ at the output of the layer. The number of features F_0 must match the number of features at the input and the number of features F_L must match the number of features at the output. Observe that the number of features at the output of Layer $(\ell - 1)$ determines the number of features at the input of Layer ℓ. Then, the filter coefficients at Layer ℓ are of dimension $F_{\ell-1} \times F_\ell$.

Task 1 Program a class that implements a GNN with L layers. This class receives as initialization parameters a GNN specification consisting of the number of layers L and vectors $[K_1, \ldots, K_L]$ and $[F_0, F_1, \ldots, F_L]$ containing the number of taps and the number of features of each layer.

Endow the class with a method that takes an input feature $\mathbf{X}$ and produces the corresponding output feature $\mathbf{\Phi}(\mathbf{X}; \mathcal{H})$. ■

Task 2 Train a GNN to predict movie ratings. Plot the evolution of the training loss and evaluate the loss in the test dataset. To obtain a good loss we need to experiment with the number of layers and the number of filter taps per layer.

2 Report

Do not take much time to prepare a lab report. We do not want you to report your code and we don't want you to report your work. Just give us answers to questions we ask. Specifically give us the following:

Question	**Report deliverable**
Task 1	Do not report
Task 2	Report training loss
Task 2	Report test loss
Task 2	Report GNN specification

We will check that your answers are correct. If they are not, we will get back to you and ask you to correct them. As long as you submit responses, you get an A for the assignment. It counts for 10% of your lab grade.