We propose to use the natural gradient to optimize graph neural networks. This allows us to efciently exploit the geometry of the underlying statistical model or parameter space for optimization and inference. NGD is better than SGD, especially in high-dimensional spaces. A new algorithm for optimizing GNNs that takes into account the unlabeled samples in the approximation of Fisher is proposed. The Graph Convolutional Network is a GNN with a linear approximation to the spectral graph convolution, followed by a non-linear activation function. The expectation of the Fisher information matrix can be approximated with a We use the regularization hyper-parameter (cid:15) to control the trade-off between the empirical risk and the regularization cost. The algorithm is: the proposed algorithm is faster and more accurate than Adam and SGD on node I think this is a very interesting paper, and Im glad to see it getting some attention. Im also glad that the authors are making the code and data available. There are a few things that I think could be improved, but Im not sure how to do that without making the paper longer. Ill start with the positives. The paper is clearly written, and the authors do a nice job of explaining the problem, the methods, and the results. I like the idea of using a graph-based approach to semi-supervised learning. Im not sure if its the best approach, but its Use PyTorch Geometric to train deep neural networks with NGD and Adam. Im a PhD student at the University of Oxford and a member of the Oxford Deep NLP Group . Im also a research scientist at Facebook AI Research. Im interested in deep learning, natural language processing, and machine learning. Im currently working on unsupervised learning for natural language processing. Ive been working on a new deep learning library called PyTorch. Its a Python library that provides a