We derive an optimization algorithm for graph neural networks using the Natural Gradient to exploit the geometry of the parameter space. Neural Networks on Graphs You can't learn from MNIST without trying to learn from MNIST . The task of classifying handwritten digits is so easy that you can do it with a very simple model. The task of learning to do it with a neural network is so difficult that you can't do it with a simple model. I have a nice picture, so I have nothing to say. The authors have done some really nice experiments (results in Fig. 4) about the ICA of a time-series of a wind turbine. Use Autoencoder to compress the data and then use the compressed data back to make a decision. This technique is called Compressed Sensing. The natural gradient is an information geometry concept that can be used to improve the convergence of SGD and Adam. In this paper, we demonstrate the natural gradient in synthetic and real-world applications in semi-supervised learning. The paper is written in a way that mathematical details are kept to a minimum. A simple, fast, and general optimization algorithm. in the context of deep neural networks, the expectation of the negative log-likelihood A lot of the work is done on sigmoid functions. This is the empirical Fisher Theorem (discussed here ), with the regularization term being the error in the Fisher Theorem, and the Hessian term being the expectation of the Fisher Information (which is what is estimated in this algorithm). Simpler and more stable. Adam and SGD are more sensitive to hyper-parameters than our method. The results are interesting, but not surprising. The bottom line is (again) that the best algorithms for learning from data are probably not the most popular ones. We still don't know the best way to learn from data in practice, but we're getting closer. Adam is a good baseline that generalizes well and is easy to implement. The KFAC methods can be made faster and more stable by increasing the learning rate decay but the tradeoff is a loss of generalization. Optimization for GNNs is hard; don't use SGD. This is because the per-node objective (for regression, Use KFAC to train Adam faster. What you need to know Graph neural networks are a new way of representing and learning graphs. They have been shown to solve supervised graph classification tasks up to 0.8 accuracy on small datasets, and go beyond simple embeddings that use only linear combinations of nodes and edges. They also have many unsupervised applications, including community detection and link prediction. They are a powerful new tool for machine learning researchers, and I hope this post has given you a taste of the possibilities.