Semi-Supervised Learning with Ladder Networks
COMP3801 Project Proposal
BM

Problem Statement and Background

State of the art models for image classification require vast amounts of supervised, labeled data. Acquiring this data requires human effort – meaning data is scarce and high-cost. Further motivation for pursuing an alternative paradigm of learning can be taken from the fact that humans learn largely in an unsupervised manner, with a limited amount of supervised signals that are available to guide us.

Deep neural networks are often pre-trained with unsupervised models such as deep belief networks (Hinton, Osindero, and Teh 2006) and stacked denoising autoencoders (Vincent, Larochelle, Bengio, and Manzagol 2008). These methods result in the networks learning useful representations prior to training, after which the parameters of the model can be fine-tuned by supervised data. However, after this unsupervised initialization of the network, no more unsupervised learning occurs – and further training depends on the use of labeled data. Additionally, representations in the model that are learned in the pre-training phase may not be relevant for the task at hand. Knowing which representations are worth learning and then building useful higher-level abstractions upon them requires targets that can then be used as contexts for learning.

Autoencoders are an unsupervised class of neural networks that when given an input, aim to represent the input in a lower dimensional space before reconstructing them back to their original form. They consist of an encoder network that encodes the input into the low dimensional space, followed by a decoder network that attempts to reconstruct the original input from the low-dimensional representation. These models can be slow to train since deep layers in the encoder are required to retain information about the low level features that exist in the data, meaning high-level abstractions are hard to learn. This is contrasted with supervised networks for classification, where the low level representations in the early layers can be discarded by later layers that learn higher level representations. Autoencoders also suffer from the problem of learning useful representations, which may vary depending on the domain of the data. Finally, denoising autoencoders add noise to the input, and minimize the error between the reconstructed, noisy input and the original, denoised input.

Valpola (2015) presents the Ladder Network model which enables deep autoencoders to be efficiently trained by removing the requirement that deeper layers retain low level representations. This is done by having two encoder networks that are identical to each other, with the exception that one has noise added to the activations in each layer. A denoising decoder attempts to reconstruct the clean input from the noisy encoder, just as a denoising autoencoder would do. Lateral connections between the corresponding layers in the noisy network and decoder network allow the decoder to have access to the low level representations, removing the need for deeper encoder layers to retain that information. Additionally, each layer in the networks are trained separately from one another. The layers in the clean encoder network act as the targets for the layers in the decoder network, and vice versa. Finally, the target output of the clean encoder is used as the target output for the noisy encoder.

(Rasmus, Valpola, Honkala, Berglund, and Raiko 2015) build on the Ladder Network proposed by Valpola, and show that it can be used for semi-supervised learning. Their network is able to learn in both supervised and unsupervised settings simultaneously – which allowed supervised learning to provide the context for the "right" representations that unsupervised learning should learn. They tested their network on the MNIST, permutation invariant MNIST, and CIFAR10 datasets. An impressive result was achieving 1.06% error rate on the MNIST dataset while only using 100 labeled examples for training.

References

- Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18, 7 (July 2006), 1527-1554.
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (ICML '08). ACM, New York, NY, USA, 1096-1103.
- Harri Valpola. From neural PCA to deep unsupervised learning. In Adv. in Independent Component Analysis and Learning Machines, pages 143–171. Elsevier, 2015. arXiv:1411.7783.
- Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. 2015. Semi-supervised learning with Ladder networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15), C. Cortes, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). MIT Press, Cambridge, MA, USA, 3546-3554.

Objectives

Rasmus et al. trained their semi-supervised network with data belonging to the same domains. For example the MNIST dataset consists of 50,000 labeled examples. N = 100, 1000, or 50,000 labeled examples were used to evaluate the network, along with the entire dataset being used as unsupervised examples. The objective of this project will be to understand and analyze the Ladder Network Model, but also apply it to using unsupervised data belonging to a different domain to aid in classification in a target domain. Further research into appropriate benchmarks to base any experiments on will need to be done, though testing data will be acquired by running the network in an image classification task.

Theoretical Research and Analysis Required

Ladder networks are analogous to what are known as hierarchical latent variable models. These models are trained using the expectation maximization (EM) algorithm. Understanding the deeper theory behind them is required to analyze Ladder Networks and their performance on different semi-supervised classification tasks.

Timeline

<u>October 10</u>: Research latent variable models and their link to the Ladder Network model.

<u>October 24</u>: Begin implementing the Ladder Network model, and further researching the theoretical basis of Ladder Networks.

<u>November 7</u>: Conduct experiments after acquiring an appropriate methodology for testing the network in the semi-supervised domain. Begin writing final project report.

<u>November 14</u>: Continue working on final project report and finishing any loose ends in terms of the theoretical understanding of the Ladder Network.