Individual Activity 4

IA5008 Sistemas neuronales

Author: Nisim Hurst

Registry ID: A01012491

Due Date: 7th of February 2018

Abstract

Infer conditional distributions on hidden layers of densely connected networks is intractable. Approximations fail to acknowledge covariance in the deepest hidden layers and parameter learning doesn't scale well.

The letter on [1] sent to Yann Le Cun by Geoffrey E. Hinton proposes a new network architecture in which the 2 first layers form *undirected associative memory* and the remaining hidden layers form a DAG that decodes the memory into observable variables. Then, the article proposed an unsupervised fast greedy algorithm regardless of hidden layer size based on *complementary priors* and RBM's. Finally, tested using the MNIST digit database this generative model outperforms discriminative methods.

Complementary Priors

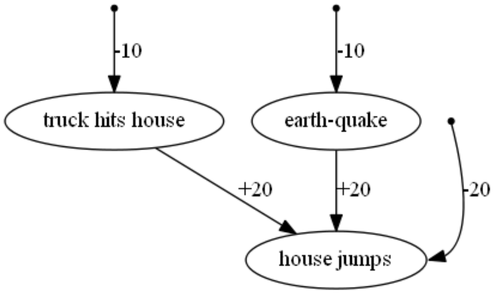

Figure 1: One variable "explains away" the other

Figure 1 show the problem of one node "explaining away" the other. Bias of both input nodes are 10, making them e^{10} times more likely to be off than on. When one of the inputs is on a the other of the output node receives an input of 0 which translate on an even chance of the house jumping (assuming *tanh* activation function) which is better than the odds of the house jumping when no node is active e^{-20} (bias). Because the probability of both happening is $e^{-10} \times e^{-10} = e^{-20}$ that is equally unlikely, the evidence in one node explains away the other.

This phenomenon makes inference in directed acyclic bayesian nets very difficult. The Vitterbi algorithm can be used in HMM but it is time consuming.

The inner workings of logistic belief nets suggest that extra hidden layers can help to countermand the *explaining away* phenomenon by fabricating **complementary priors** based on tying weights of previous layers in directed graph models. By multipliying the likelihood term by the prior we seek to have a factorial prior distribution ($O(n^2)$) over the hidden variables of undirected (cyclic) models to be tractable. Sampling can be made using the Gibbs algorithm.

Restricted Boltzmann Machines and Contrastive Divergence

An (RBM) is is a single layer autoencoder. The units are updated in parallel based in a cyclic process considering the output of the other layer (input or hidden). It is called restricted because the nodes on the same layer are not connected to each other but each input node is connected to all the other nodes on the hidden layer. Gibbs sampling is used to reduce complexity. This process is equivalent to generating data in a belief net with tight weights.

A Greedy Learning Algorithm for Transforming Representations

A new architecture based on boosting is proposed in which stacked data representations of RBS form 3 kind of layers:

1. Undirected connection layers that form associative memory. This is equivalent to have many directed hidden layers with tied weights.
2. Directed connection layers to map the state of the memory.
3. Directed connection layers to learn factorial representations from the layer below.

A simplifying assumption is that each layer has the same number of units.

Due to the impossibility of the RBM's to model input data perfectly we need a greedy algorithm that consists in the following:

1. Learn W_0 (the *generative* weights of the first layer) assuming all weights are tied.
2. Freeze W_0 for using W_0^T (transpose of the *generative* weights) to infer factorial approximate posterior distributions over the states of the variables of the first hidden layer.
3. Learn an RBM model of the higher-level "data" that was produced using W_0.

Finally we can untie the weights of the bottom layer with the guarantee that we will never decrease the data likelihood applying it recursively.

Back-Fitting with the Up-Down Algorithm

Underfitting caused by learning just one layer at a time can be ameliorated using back-fitting. This is done by revising the weights at each new learned layer. There are two kinds of weights: *recognition* used for inference and *generative* that define the model. The *recognition* weights are used bottom-up and then the *generative* weights are maximized. Then the *generative* weights are used top-down to activate the lower layers and update the *recognition* weights.

Performance on the MNIST Database

Without any geometrical preprocessing 1.25% errors was achieved on the official test set (using handcrafted connectivity for the task) after six weeks. 10-example mini-batches were used for 30 epochs. Pixel intensities were normalized in each RBM. RBM greedy training testing on MATLAB gave 2.5% errors on the test set. Different loss functions or weight decay can reduce this error rate. One-hot softmax output layer was used. Stochastic Gibbs sampling was also tested, with more iterations less error on the validation set. Using enhancement of the training set with geometry knowledge the error can be lowered to 0.4%.

Looking into the Mind of a Neural Network

Gibbs sampling was used to generate samples from the associative memory. An image shown in the article shows the hypothetical learned world.

Conclusion

By tying up weights of higher layers it is possible to estimate posterior probabilities in a densely connected belief network with factorial complexity. After each layer has been learned we can untie the *recognition* weights from the weights of higher layers. The greedy learning algorithm can be used to initialize the weights for a much slower generative model. However, deeper or larger training networks can also take advantage of this fast greedy algorithm.

The most important limitations of the generative model of natural images are:

1. It does not have a systemic way of dealing with perceptual invariances.
2. It does not learn to attend the most informative parts of an object (image).

Benefits over discriminative methods are:

1. Learn low level features without labels and don't overfit.
2. We can visually assess and interpret what the model has learned just by seeing the generated samples.
3. Moore's law is allowing to treat many more domains and thus outperform the discriminative methods also on those domains.

Appendix A talks about the math behind *Complementary Priors*. Appendix B. gives pseudocode for the Up-Down algorithm.

References

[1] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets." *Neural computation*, vol. 18, no. 7, pp. 1527–54, 2006.