The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Constructing Hierarchical Bayesian Networks with Pooling

Kaneharu Nishino, Mary Inaba
Graduate School of Information Science and Technology,
The University of Tokyo, Tokyo, Japan
{nishino.kaneharu, mary}@ci.i.u-tokyo.ac.jp

Abstract

Inspired by the Bayesian brain hypothesis and deep learning, we develop a Bayesian autoencoder, a method of constructing recognition systems using a Bayesian network. We construct hierarchical Bayesian networks based on feature extraction and implement pooling to achieve invariance within a Bayesian network framework. The constructed networks propagate information bidirectionally between layers. We expect they will be able to achieve brain-like recognition using local features and global information such as their environments.

Introduction

Deep learning (Bengio et al. 2007) extracts multiple levels of abstract features from input data. Local concrete features appear in the lower layers, while global abstract features, composed of lower-level features, appear in the higher layers. These feature hierarchies are similar to the visual areas in the cerebral neocortex (Lee, Ekanadham, and Ng 2008).

That said, in contrast to most neural networks, brains have paths not only from lower to higher areas (bottom-up), but from higher to lower areas (top-down). The top-down paths integrate global information for recognition (Bullier 2001). The Bayesian brain hypothesis (Doya et al. 2007) proposed for handling top-down paths uses a Bayesian network (Bayes net) as a neural computation model and integrates top-down information such as prediction and bottom-up sensory perception in a manner of probability theory.

Based on this work, we propose the Bayesian autoencoder (BAE) as a method of constructing networks that can extract and recognize features with in a Bayes net framework.

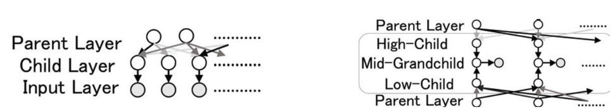
Bayesian Autoencoder

BAE Network

BAE extracts features from the input and constructs a hierarchical Bayes net called a *BAE net*. Each hidden variable represents a feature and is binary in nature: it can be true (T: the feature exists) or false (F: the feature does not). First, we describe a BAE net with one parent-layer, as shown in Fig. 1a. The lowest layer is the input-layer, and the observable nodes represent continuous variables. The child-layer sits on top of this layer, and each of the input nodes is a child of one of the child nodes. The child variables are ternary, allowing states of T, F, and X, where T and F are as before and X is required for consistency of the 1pT assumption that will be described later (The X state is not used in recognition and learning). After that, the parent-layer sits on top of the child-layer, and the links between them are tuned through learning.

BAE nets use two parameter types: conditional probabilities (CPs) and link intensities (LIs). To reduce the number of CPs, we employ the *one-parent-T (1pT)* assumption, namely that features cannot coexist in the same region. Under this assumption, the CPs that a child is in state T or F given that two or more of its parents are in state T are both 0. The 1pT assumption reduces the number of CPs to $O(n)$ from $O(2^n)$ for n parents. Moreover, the potential inferences that multiple parents have state T are eliminated. Features are recognized exclusively within the same region and their sparsity is obtained. The LIs take continuous values between 0 (completely cut) and 1(fully connected). The 1pT assumption strongly limits the conditional probabilities, to the point where they cannot express the situation where variables are independent. The LIs are used to extend the conditional probabilities so that they can handle the case.

The parameters are tuned in different ways. The CPs are tuned to minimize the difference between top-down and bottom-up inference for the child, while the LIs are tuned to maximize the mutual information between the parents and the input data.



(a) One parent-layer BAE net

(b) BAE net pooling layer

Figure 1: Architectures of BAE nets

BAE Net Pooling Layer

The parent variables are inferred by the product of messages from its children. Therefore if we construct BAE nets only with parent-layers, its features will have no invariance. To avoid this, we implement pooling layers to BAE nets.

The pooling layer is shown in Fig. 1b and is constructed as follows. Low-child variables are connected to the lower parents, mid-grandchild variables are connected to each low-child, and high-child variables are connected to each mid-grandchild. The observable variables are children of the mid-grandchild, and observed as fixed state regardless of input.

The CPs of the mid-grandchild are fixed to the particular values so that messages from it to the high-child has same value as messages from the low-child to it. Messages from a low-child is given by the weighted sum of messages from the lower parents to low-child. This layer therefore behaves as a pooling layer.

Advantages of BAE Nets

BAE nets inherit the characteristic inference features of Bayes nets. With this property, BAE nets can propagate information both from lower to higher layers and from higher to lower layers. Therefore when a BAE net has a deep architecture and hierarchical features, it can use not only bottom-up local information but top-down global information, such as context and environment, to recognize each object.

In addition, the 1pT assumption results in the sparse features due to explaining away. Parents inhibit each other via the shared child and the features are recognized exclusively.

Moreover, BAE nets tune the links between the pooling and parent layers. In contrast with built-in invariance, such as in a convolutional neural network (Krizhevsky, Sutskever, and Hinton 2012), BAE nets learn which features should be pooled. We consider exclusive features should be pooled and BAE connects them to a child of pooling layers dynamically.

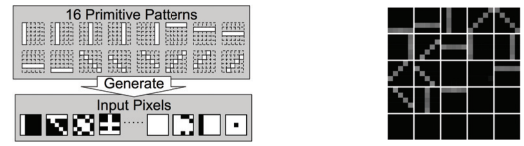
Experiments for BAE

In this section, we present the results of experiments to confirm the performance of BAE. First, we conducted a 5x5 experiment to confirm BAE feature extraction, as shown in Fig. 2. We generated 5x5 pixel images as input data using 16 primitive patterns, and trained a BAE net with one parent layer using the input data. The results are shown in Fig. 2b, 16 of the 25 parents correctly extracted features that were the same as all original primitive patterns.

The pooling experiment involved training a pooling layer (Fig. 3). Input data consisted of randomly-generated 1x10 belt-like input images in which no two successive pixels were white. In the initial network structure, each child had one parent (as shown in Fig. 3a). The links between parents and low-child variables were trained. Since no two adjacent pixels were white, their activities could be considered as being exclusively each other. We therefore expected parents corresponding to adjacent positions to become connected to the same low-child. As shown in Fig. 3b, adjacent parents have links to the same low-child as we expected. BAE nets selected exclusive parents and pooled them successfully.

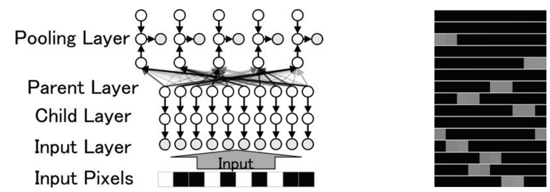
Concluding Remarks

Using hierarchical Bayes nets, we expect to use high-level concepts to recognize low-level objects to improve recognition performance. So far, we can construct BAE nets by feature extraction from small images, and obtain a pooling layer over parents. In future work, we plan to construct multiple layers to extract feature hierarchies.



(a) Generation of 5x5 input pixels (b) LIs of parents

Figure 2: 5x5 Experiment. (a) shows how to generate 5x5 input pixels. Each primitives arose independently in 5% and set values of their pixels at 0.995. The values of pixels where no primitives arose were left at 0.005. (b) The resulting LIs of the parents. The boxes correspond to parents and panels show the LI values of the corresponding links. Black represents 0 and the lighter colors represents larger values.



(a) Structure of the trained network (b) LIs of parents

Figure 3: Pooling Experiment. (a) Example showing the network structure and a belt-like input pixel array. In the input arrays, no two successive pixels were white. The links between the pooling layer and parent layer were trained. (b) The LIs of trained low-child in the pooling later. Each belt corresponds to a low-child and the panels represent the LIs of the links to parents corresponding to each input position.

Acknowledgments

This work was supported by JSPS Grant-in-Aid for JSPS Fellows, JP17J09110.

References

Bengio, Y.; Lamblin, P.; Popovici, D.; and Larochelle, H. 2007. Greedy layer-wise training of deep networks. In Bernhard Schölkopf, J. P., and Hoffman, T., eds., *Advances in Neural Information Processing Systems 19 (NIPS' 06)*, 153 – 160. MIT Press.

Bullier, J. 2001. Integrated model of visual processing. *Brain Research Reviews* 36(2):96 – 107. The Brain in Health and Disease - from Molecules to Man. Swiss National Foundation Symposium NRP 38.

Doya, K.; Ishii, S.; Pouget, A.; and Rao, R. P. 2007. *Bayesian brain : probabilistic approaches to neural coding*. Computational neuroscience. Cambridge, Mass. MIT Press.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1097–1105.

Lee, H.; Ekanadham, C.; and Ng, A. Y. 2008. Sparse deep belief net model for visual area v2. In Platt, J.; Koller, D.; Singer, Y.; and Roweis, S., eds., *Advances in Neural Information Processing Systems 20*. Curran Associates, Inc. 873–880.