Measuring the Usefulness of Hidden Units in Boltzmann Machines with Mutual Information

Mathias Berglund, Tapani Raiko, and KyungHyun Cho

Department of Information and Computer Science
Aalto University School of Science, Finland
firstname.lastname@aalto.fi

Abstract. Restricted Boltzmann machines (RBMs) and deep Boltzmann machines (DBMs) are important models in deep learning, but it is often difficult to measure their performance in general, or measure the importance of individual hidden units in specific. We propose to use mutual information to measure the usefulness of individual hidden units in Boltzmann machines. The measure serves as an upper bound for the information the neuron can pass on, enabling detection of a particular kind of poor training results. We confirm experimentally, that the proposed measure is telling how much the performance of the model drops when some of the units of an RBM are pruned away. Our experiments on DBMs highlight differences among different pretraining options.

Keywords: Deep learning, restricted Boltzmann machine, deep Boltzmann machine, pruning, structural learning, mutual information

1 Introduction

Restricted Boltzmann machines (RBMs) and deep Boltzmann machines (DBMs) are important models in *deep learning*, helping to achieve state-of-the-art performance in many tasks. However, both models are also known to be difficult to train [1, 2].

If training of an RBM or DBM is not successful, it is often assumed that the hidden neurons do not learn to detect features that are useful for the task the Boltzmann machine (BM) is expected to perform. Whether training is successful is often measured by directly testing the performance of the trained model in this task at hand. The model is therefore often treated as a "black box" only evaluated based on final performance.

When training RBMs or DBMs, it would be beneficial to gain deeper insights into the details of a learned model beyond a mere final performance measure. One way to shed light on the underlying functionality of a particular model is to collect statistics of the individual neurons. E.g. in tasks where the visible neurons of the BM represent pixels in pictures, it is standard practice to visualize the learned weights.

Structural learning is an additional field where measuring the importance of individual neurons is crucial. One approach in structural learning is to add or prune neurons while training the model (see e.g. [3–5]). In general, the benefit of such an approach includes decreasing the number of hyperparameters that need to be defined a priori [4], better expected generalization, and faster performance [6]. However, it is unclear how training a BM is affected by adding or pruning neurons while training.

In this paper, we propose to use mutual information between the observation vector and a single hidden unit for evaluating its importance.

After reviewing RBMs and DBMs in Sect. 2, we propose the mutual information (MI) measure for studying the importance of individual hidden units of a BM in Sect. 3. Experimenting with RBMs in Sect. 4, we demonstrate the usefulness of the measure in pruning and adding neurons as well as visualizing the progress of learning. In Sect. 5, we compare pretraining choices of a DBM using the proposed measure.

2 Boltzmann Machines: Background

2.1 Restricted Boltzmann Machines

A restricted Boltzmann machine (RBM) [7] is a variant of a Boltzmann machine that has a bipartite structure such that each visible neuron is connected to all hidden neurons and each hidden neuron to all visible neurons, but there are no edges between the same type of neurons. An RBM defines an energy of each state $(\mathbf{x}, \mathbf{h})$ by

$$-E(\mathbf{x}, \mathbf{h} \mid \boldsymbol{\theta}) = \mathbf{b}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{h} + \mathbf{x}^\top \mathbf{W} \mathbf{h}, \tag{1}$$

and assigns the following probability to the state via Boltzmann distribution:
$p(\mathbf{x}, \mathbf{h} \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{-E\left(\mathbf{x}, \mathbf{h} \mid \boldsymbol{\theta}\right)\right\}$, where $\boldsymbol{\theta} = \{\mathbf{b}, \mathbf{c}, \mathbf{W}\}$ is a set of parameters consisting of visible and hidden biases as well as weights between visible and hidden neurons. $Z(\boldsymbol{\theta})$ is a normalization constant that makes the probabilities sum up to one.

2.2 Deep Boltzmann Machines

A deep Boltzmann machine (DBM) was proposed in [8] as a relaxed version of an RBM. A DBM simply stacks multiple additional layers of hidden units on the layer of hidden units of an RBM. As was the case with an RBM, consecutive layers are fully connected, while there is no edge among the units in one layer.

The energy function is defined as

$$-E(\mathbf{x}, \mathbf{h} \mid \boldsymbol{\theta}) = \mathbf{b}^\top \mathbf{x} + \mathbf{c}_{[1]}^\top \mathbf{h}_{[1]} + \mathbf{x}^\top \mathbf{W} \mathbf{h}_{[1]} + \sum_{l=2}^{L} \left(\mathbf{c}_{[l]}^\top \mathbf{h}_{[l]} + \mathbf{h}_{[l-1]}^\top \mathbf{U}_{[l-1]} \mathbf{h}_{[l]} \right), \tag{2}$$

where L is the number of hidden layers. The state and biases of the hidden units at the l-th hidden layer and the weight matrix between the l-th and $(l+1)$-th layers are respectively defined by $\mathbf{h}_{[l]} = \left[h_1^{[l]}, \ldots, h_{q_l}^{[l]}\right]^\top$, $\mathbf{c}_{[l]} = \left[c_1^{[l]}, \ldots, c_{q_l}^{[l]}\right]^\top$, $\mathbf{U}_{[l]} = \left[u_{ij}^{[l]}\right]$, where q_l is the number of the units in the l-th layer and $\mathbf{U}_{[l]} \in \mathbb{R}^{q_l \times q_{l+1}}$.

2.3 Why Interested in Boltzmann Machines?

RBM is an important basic building block of deep neural networks. In [9] it was shown that an MLP with many hidden layers can be trained well by greedily pretraining each pair of consecutive layers as an RBM. Furthermore, deep generative models such as

deep belief networks and DBMs were found to be easily trainable if the parameters were initialized by greedy layer-wise pretraining using an RBM [10, 8]. DBM was found to be effective at initializing the parameters of an MLP as well [8].

Furthermore, both RBM and DBM have been found to be useful on their own, as well. RBM and DBM were used to achieve high predictive performance on collaborative filtering [11], multimodal learning [12] and hierarchical feature extraction [13].

However, all these achievements by RBM and DBM require that these neural networks were trained *well*. Several recent research showed that training RBM and DBM is difficult, and that inappropriately trained ones may neither perform well on their own nor as a part of another model [1, 2]. Although computing log-likelihood by estimating the normalizing constant [14] has been oft-used to evaluate RBMs and DBMs, it is computationally expensive and does not tell much about how much contribution each hidden neuron makes. Hence, in this paper we try to explore one potential measure that can be used to evaluate the contribution of each hidden neuron in RBM and DBM.

3 Mutual Information Measure for Hidden Units

Neural networks such as multi-layer perceptron (MLP) networks are often criticized for being *black boxes*, that is, it is difficult to understand what the individual neurons are doing. One measure that can easily be studied is the variance of the neuron activation across samples (see e.g. [15]). The underlying rationale is that neurons with constant activation across the samples cannot convey any discriminative information about the samples. However, as Boltzmann machines are stochastic, measuring activation variance is clearly not appropriate, as even a neuron with constant activation probability could have a high activation variance.

We therefore propose to measure the *relevant* activity (or importance) of a single hidden neuron h_j in Boltzmann machines by measuring the mutual information (MI) $I(\mathbf{x}, h_j)$ between the observation vector (or the set of visible neurons) $\mathbf{x}$ and the hidden neuron h_j. Specifically, the MI-measure of the hidden unit MI_j is

$$\begin{aligned}
\mathrm{MI}_j = I(\mathbf{x}, h_j) &= \sum_{h_j=0}^{1} \sum_{\mathbf{x}} P(\mathbf{x}, h_j) \log_2 \left(\frac{P(h_j \mid \mathbf{x})}{P(h_j)} \right) \\
&= \sum_{h_j=0}^{1} \left[-P(h_j) \log_2 \left(P(h_j) \right) + \sum_{\mathbf{x}} P(\mathbf{x}) P(h_j \mid \mathbf{x}) \log_2 \left(P(h_j \mid \mathbf{x}) \right) \right] \\
&\approx \sum_{h_j=0}^{1} \left[-P(h_j) \log_2 \left(P(h_j) \right) + \sum_{t=1}^{T} \frac{1}{T} P(h_j \mid \mathbf{x}_t) \log_2 \left(P(h_j \mid \mathbf{x}_t) \right) \right].
\end{aligned}$$

We use the logarithm with base 2 in order to get the amount of information as bits. It is easy to show that the mutual information between the binary hidden neuron h_j and the visible neurons $\mathbf{x}$ ranges from 0 to 1 bit, and defines the upper bound of the average information the hidden neuron can convey about the state of the visible neurons $\mathbf{x}$. The measure is also independent of the particular task that the model learned by the Boltzmann machine is supposed to perform.

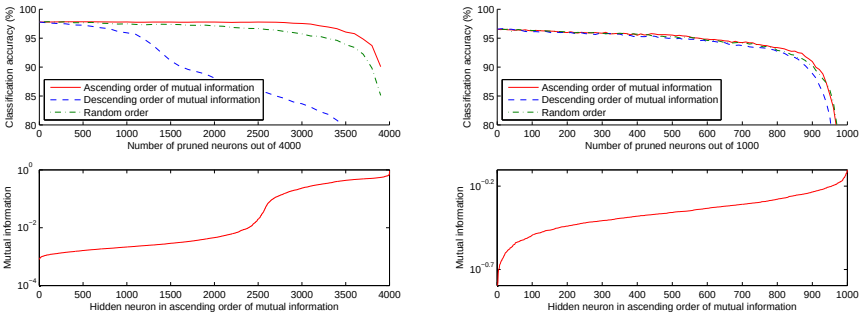

Fig. 1. Classification accuracy development when pruning neurons. *Left:* Original RBM was trained with 4000 hidden neurons for 1000 epochs using standard gradient. *Right:* Original network was trained with 1000 hidden neurons using enhanced gradient for 100 epochs.

Note that the MI-measure cannot measure how well the hidden unit works together with other hidden units. For instance, maximizing the MI-measure could not be used as the objective of training Boltzmann machines since training using such a criterion could lead to each hidden unit representing the same feature. However, the MI-measure serves as a useful measure of the upper bound for how useful the hidden neuron can be for any task the Boltzmann machine is trained for. Therefore, the hidden neurons with almost zero MI-measure will also be almost useless in any task. This is a particularly useful measure when training Boltzmann machines, as certain common training situations yield neurons with a very low MI-measure.

4 Experiments on Restricted Boltzmann Machines

This section studies the use of MI-measure as a measure of the usefulness of hidden neurons in the case of RBMs by (1) pruning, (2) adding new hidden neurons during training, and (3) visualizing the progress of training.

4.1 Pruning Neurons after Training

We tested the contribution of neurons with varying mutual information to a simple classification task. We trained an RBM with 4000 hidden neurons on the MNIST data set [16], and used the hidden neuron activations as inputs to a logistic classifier. We then pruned 100 hidden neurons at a time in order of the MI-measure. We did this both in ascending and descending order, in addition to randomly pruning 100 neurons at a time. The remaining neuron activations were then used as features in the classifier.

The results of the tests are shown in Fig. 1 (left). As predicted, the classification performance does not drop markedly when pruning neurons with very low MI-measure.

We also did a similar exercise for a 1000 hidden neuron RBM, where we used the enhanced gradient [1] in training. The results clearly differ from the previous model, in that the MI-measure is not clearly related to importance in the classification task. As

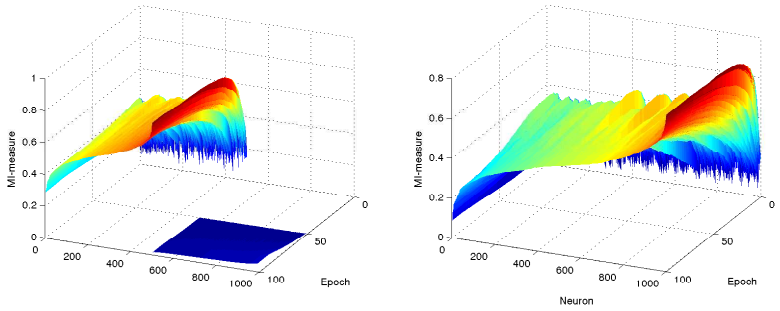

Fig. 2. MI-measure of RBM where 500 hidden neurons are added in the middle of the training (left) vs. a normal RBM with 1000 hidden neurons from the start of the training (right). Both models were trained for 100 epochs.

can be seen from Fig. 1 (right), this model differs from the previous in that all hidden neurons have a fairly high MI-measure. Therefore, no neurons have a low enough MI-measure as to not be able to convey enough information about the observations $\mathbf{x}$. This clearly reveals that high MI-measure is not always an accurate indicator of high significance for the classification task – only a sufficiently low MI-measure can confidentially predict that a neuron is not useful.

4.2 Adding Neurons During Training

One potential way to learn an optimal structure of an RBM would be to add neurons to the hidden layer. This has bee studied e.g. in [4, 3]. However, it is possible that adding neurons to a layer of hidden units where the previous hidden units have been trained for some time would not be beneficial, as the added neurons might not learn relevant structures in addition to the already co-adapted hidden neurons. In order to test that hypothesis, we trained an RBM of 500 hidden neurons for 50 epochs, after which we added another 500 hidden neurons to the model and trained it for another 50 epochs. The development of the MI-measure can be seen in Fig. 2.

We again used the hidden neuron activations as features for a logistic classifier and compared the performance of the two models. The table lists the median accuracy from three runs. The performance of the model with added features is clearly inferior to the model trained with 1000 hidden neurons for 100 epochs.

	500 hidden units	500 + 500 hidden units	1000 hidden units
Class. accuracy	95.22 %	95.45 %	96.49 %

When examining the MI-measure of the 500 neurons added after 50 epochs, it is clear that the measure stays considerably lower than for the 500 neurons added in the beginning. It is also worth noting that even during the 50 epochs, the mutual information stays much lower than the level the original 500 neurons reached already after only a few epochs of training. This would support the hypothesis that the neurons added later are not able to find very relevant features easily. We therefore recommend caution when considering naively adding neurons parallel to already trained neurons in an RBM.

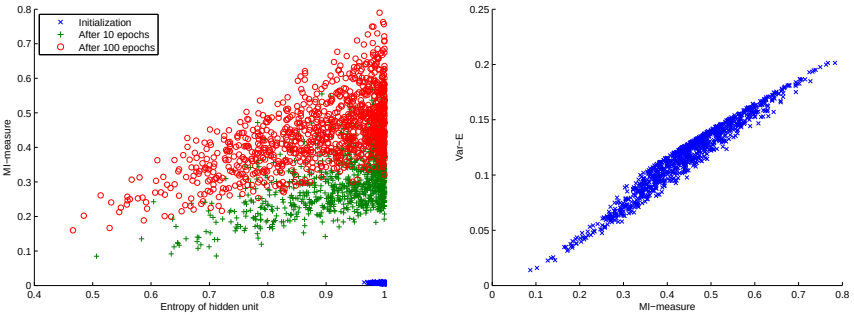

Fig. 3. *Left:* Mutual information of a hidden unit with the observation vector (MI-measure) plotted against the entropy of the hidden unit at different stages of learning. *Right:* Variance of the expected activation (Var-E) of each hidden neuron plotted against the MI-measure after 100 epochs of training for an RBM with 1000 hidden neurons.

4.3 Relations to Entropy and Variance

One use for the proposed measure is simply to visualize the progress of learning.

In Fig. 3 (left), we plot the entropy $H(h_j) = -\sum_{h_j=0}^{1} P(h_j) \log_2 \left(P(h_j)\right)$ of the hidden units against their MI-measure in the initialization phase, after 10 epochs, and after 100 epochs. We see that training increases their mutual information on average, but also decreases the entropy on some hidden neurons.

It can be seen in Fig. 3 (right) that the MI-measure has a close resemblance to the variance over samples of the expected activation $\text{Var-E}_j = \text{Var}_t \left(E_{P(h_j|\mathbf{x}_t)} \left[h_j\right]\right)$. Var-E highlights that the stochastic variation in $P(h_j \mid \mathbf{x}_t)$ does not count as relevant activity, whereas variation over the data index t does. Note that we used Var-E for studying neurons in Fig. 9 of [1].

5 Experiments on Deep Boltzmann Machines

Although Deep Boltzmann Machines [8] have been used with great success in several applications, they are generally considered difficult to train. One method that has been essential in alleviating that difficulty is greedy layer-wise pretraining.

In order to illustrate this difficulty, we trained two DBMs with two layers of 1000 hidden neurons for 100 epochs: the first without any pretraining, and the second one with two-stage pretraining [2]. The hidden neuron activations of both of the models were used as features for a logistic classifier. As can be seen from Fig. 4 the mutual information of especially the second layer was extremely low without pretraining. The mutual information is in fact so low, that the entire second layer only conveys on average a maximum of 1.5×10^{-10} bits of information about the observations. This illustration strengthens the hypothesis that the difficulty in training DBMs relates to the model not learning useful features of the data.

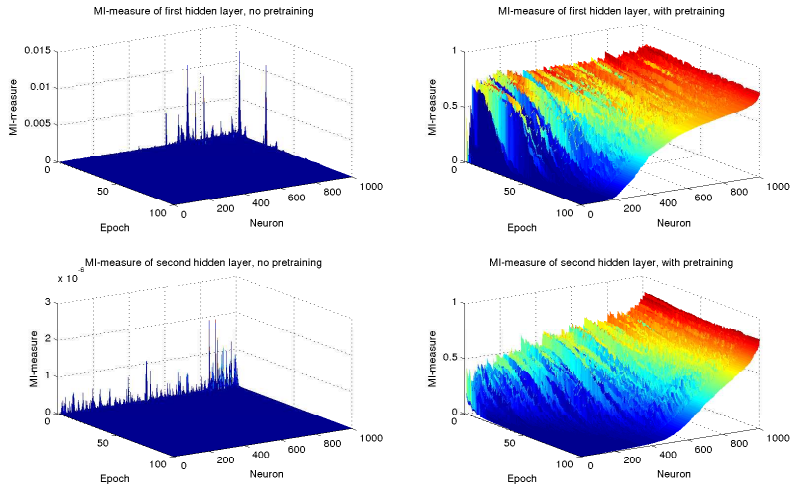

Fig. 4. MI-measure of a DBM with two layers of hidden units without pretraining (left) and with pretraining (right)

6 Discussion

We propose using mutual information between the observation vector and a single hidden unit (MI-measure) for evaluating the importance of individual hidden units of a Boltzmann machine. Following the progress of this measure during training would be useful at least for noticing situations where some of the units are not useful at all. We demonstrated several cases where it could happen. Firstly, training a large RBM with traditional gradient can include a lot of inactive units. Secondly, when an RBM has already learned a representation of the data, and new units are introduced in it, it is rather difficult to make them useful. Thirdly, when training deep Boltzmann machines without layer-wise pretraining, all the neurons in especially the upper layers might be useless.

We found that the MI-measure should only very cautiously be used as such to rank neuron importance among the active neurons, since it rather serves as an upper bound of importance. This might be due to at least two phenomena: Firstly, the MI-measure ignores the interaction among hidden units, and Boltzmann machines produce very distributed representations of data since each unit can only retain at most one bit of information. Secondly, it is well known that sparse representations perform well especially for classification tasks [17]. Sparse features have a lower entropy, and Fig. 3 shows that units with lower entropy tend to have a lower MI-measure, too. This would suggest that perhaps some combination of entropy and MI-measure could be used as a more accurate measure of usefulness in the future. Another direction in which to continue the work is to study the mutual information of the latent representation and class labels, assuming they are available. This has been proposed as a learning criterion by Peltonen and Kaski [18].

References

1. Cho, K., Raiko, T., Ilin, A.: Enhanced gradient for training restricted Boltzmann machines. Neural Computation **25**(3) (March 2013) 805–831
2. Cho, K., Raiko, T., Ilin, A., Karhunen, J.: A two-stage pretraining algorithm for deep Boltzmann machines. In: Proceedings of the 23rd International Conference on Artificial Neural Networks. (September 2013) to appear.
3. Zhou, G., Sohn, K., Lee, H.: Online incremental feature learning with denoising autoencoders. In: Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012). (2012) 1453–1461
4. Adams, R.P., Wallach, H.M., Ghahramani, Z.: Learning the structure of deep sparse graphical models. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010). (2010) 1–8
5. Engelbrecht, A.P.: A new pruning heuristic based on variance analysis of sensitivity information. Transactions on Neural Networks **12**(6) (November 2001) 1386–1399
6. Reed, R.: Pruning algorithms-a survey. Transactions on Neural Networks **4**(5) (September 1993) 740–747
7. Smolensky, P.: Information processing in dynamical systems: foundations of harmony theory. In: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations. MIT Press, Cambridge, MA, USA (1986) 194–281
8. Salakhutdinov, R., Hinton, G.: Deep Boltzmann machines. In: Proceedings of the Twelfth Internation Conference on Artificial Intelligence and Statistics (AISTATS 2009). (2009) 448–455
9. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science **313**(5786) (July 2006) 504–507
10. Hinton, G., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7) (July 2006) 1527–1554
11. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th international conference on Machine learning (ICML 2007), New York, NY, USA, ACM (2007) 791–798
12. Srivastava, N., Salakhutdinov, R.: Multimodal learning with deep boltzmann machines. In Bartlett, P., Pereira, F., Burges, C., Bottou, L., Weinberger, K., eds.: Advances in Neural Information Processing Systems 25. (2012) 2231–2239
13. Lee, H., Grosse, R., Ranganath, R., Ng, A.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009), ACM (2009) 609–616
14. Salakhutdinov, R.: Learning and evaluating Boltzmann machines. Technical Report UTML TR 2008-002, Department of Computer Science, University of Toronto (June 2008)
15. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010). (May 2010) 249–256
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. Volume 86. (1998) 2278–2324
17. Ranzato, M., Boureau, Y.L., LeCun, Y.: Sparse feature learning for deep belief networks. In Platt, J., Koller, D., Singer, Y., Roweis, S., eds.: Advances in Neural Information Processing Systems 20. MIT Press, Cambridge, MA (2008) 1185–1192
18. Peltonen, J., Kaski, S.: Discriminative components of data. Neural Networks, IEEE Transactions on **16**(1) (2005) 68–83