Exploring Helmholtz Machines with Sparse Representations

Justin Theiss, Steven Shepard | VS265 Fall 2018

This semester in VS265, we have considered the human perceptual system as an inference engine that actively solves the intractable problem to infer probable causes of raw sensory input. To support this inference endeavor, the course has explored methods of neural computation which provide insight into the structure of data. Among these methods, unsupervised learning of parameters in a multilayer neural network has demonstrated capabilities in revealing structure in data such as natural images. Moreover, unsupervised learning avoids two costly components needed for supervised learning: teacher signals, and a method for communicating error information to all connections. Still, unsupervised methods are non-trivial to train. We were thus motivated to explore how a particular unsupervised model, the Helmholtz Machine, learns from cycles of sensing and dreaming to eventually learn a concise internal model of the world without labels. The Helmholtz Machine was first proposed by (Dayan et al., 1995) as a multilayer neural network that learns with a wake-sleep algorithm to understand binary input data. The name of the model was inspired by the influential neurophysiologist Hermann Von Helmholtz (1821-1894), who was an early proponent of the idea that human perception is an active inference process. Here, we have worked on re-implementing the Helmholtz Machine on toy datasets and incorporating sparsity constraints. The Helmholtz Machine is an especially intriguing model because it supports the theory of free-energy based learning in the brain mediated from bottom-up and top-down connections in the brain (Friston, 2009).

The Helmholtz Machine is considered an “analysis by synthesis” explicit density estimation model. Top-down connections synthesize an example input vector by passing a sampled vector down through the network, meanwhile bottom-up connections build the recognition/analysis model. This dual model is trained through an alternating, two-step learning process called the wake-sleep algorithm. The aim of the wake-sleep algorithm is to learn representations that are economical to describe the data while also allowing the sensory input data to be reconstructed accurately. In the wake phase, unit activations are driven by recognition

weights and real world data. At each layer, the probability of a unit being active can be computed as the linear sum of the previous layer's stochastic binary outputs multiplied by the recognition weights and passed through a sigmoid activation function (equation 1 below):

(1) $$q_j^{\ell}(\phi, \mathbf{s}^{\ell-1}) = \sigma(\sum_i s_i^{\ell-1} \phi_{i\ ,j}^{\ell-1,\ell})$$

(2) $$p_j^{\ell}(\theta, \mathbf{s}^{\ell+1}) = \sigma(\sum_i s_k^{\ell+1} \theta_{k\ ,j}^{\ell+1,\ell})$$

These probabilities are then used to stochastically sample the binary outputs for each layer. However, during the wake phase the model only makes adjustments to generative model weights with a purely local delta rule. In the sleep phase, unit activations are driven by generative weights by stochastically sampled outputs of the layer above (equation 2 above). Rather than using the training data, the sleep phase begins from the top layer by sampling from the generative biases passed through the sigmoid activation function. For learning in the sleep phase, the model now makes adjustments to the recognition model weights with an objective to decrease an approximate to the variational free energy. As the name suggests, variational free energy is an augmentation of free energy (also known as Helmholtz free energy). The Helmholtz free energy, given fixed real world data and a generative model distribution, is defined as the surprise elicited by a pattern generated from the real world probability distribution. Mathematically, the surprise is the negative logarithm of the probability distribution. The variational free energy is a reparametrization of the same quantity with a different probability distribution, in the case of the Helmholtz Machine, defined by the recognition weights instead of the generative weights.

The wake-sleep algorithm's pursuit to discover an economical representation of sensory input data while being able to parsimoniously reconstruct the inputs from the latent factors is not unlike the pursuit of sparse coding. Another way to characterize the Helmholtz Machine is as a hierarchical self-supervised compression network where the goal of learning is to minimize the description length, total number of required bits, needed for a message to be sent from a transmitter/sender to a receiver. The message sent in the case of a Helmholtz Machine is a

hidden representation of input data and the difference between the original input data and the top-down reconstruction using the hidden representation. Interestingly this information theoretic perspective, minimizing the description length of a message, is congruent with the information encoding endeavor of sparse compressive sensing. Given these common goals we wanted to explore how the wake-sleep algorithm and sparsity principles may work together. Further, since the Helmholtz Machine is trained in an unsupervised manner, it will naturally be likely to expend resources on representing potentially non-informative regions in the input space. Adding a sparse constraint on the hidden activations could thus encourage the units to become more selective of the input regions they encode.

Experiments & Results

First, we demonstrated that the wake-sleep algorithm can be used to learn a true distribution of vertical and horizontal bars (66% vertical, 33% horizontal; 3x3 pixels). Using a two-layer model with 6 hidden units, the trained model generated samples from a distribution similar to the true distribution (67% vertical, 22% horizontal, and 11% other).

Next, we were interested in comparing the effects from training the Helmholtz Machine with different types of sparsity constraints. The three types used were an L1 norm constraint, overcomplete sparsity, and top-k sparsity. For the L1 norm sparsity constraint, the L1 norm of the recognition activities at each hidden layer was included in the cost from which the gradients for the recognition weights were computed. For the overcomplete sparsity, three layers were trained with the same number of hidden units (784). For top-k sparsity, only the k-highest activities were kept at each layer and lower activities were set to zero. This is similar to the winner-take-all algorithm in which the top activity is set to one and all others are set to zero.

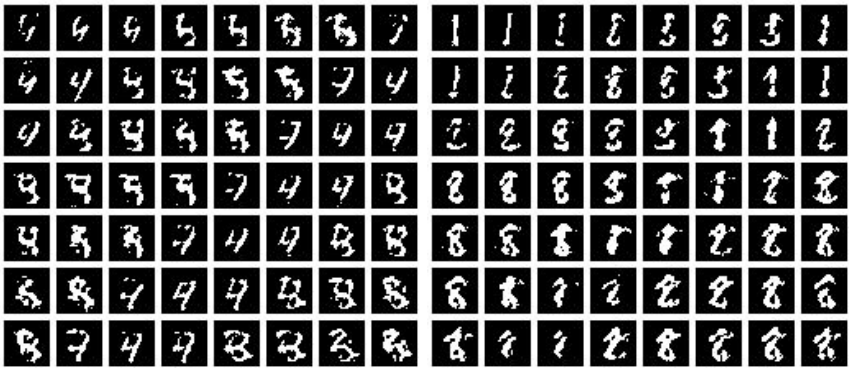
Each trained model contained three layers (including the visible layer) with 128 and 32 hidden units in the second and third layers, respectively. We trained each model using the wake-sleep algorithm on the MNIST handwritten digit data set (LeCun et al., 1998). In addition to speeding up training, sparsity generally led to more interpretable learned features. However, in order to assess the effect of each type of sparsity constraint on latent representations of the model, we visualized reconstructed images from the model while introducing noise to the

top-layer samples. We began by sampling the top-layer generative biases as done during the sleep phase of the wake-sleep algorithm. Then we iteratively flipped bits within this sampled representation and propagated the corrupted sample down through the generative model to the visible layer in order to obtain the reconstructed image.

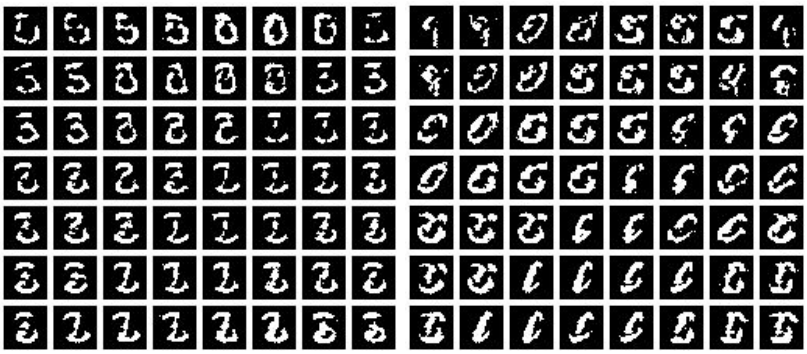
As seen in the visualizations below, sparsity does appear to help make the latent representations more robust to noise added to the top layer. Whereas the identity of the digit reconstructed from the Helmholtz Machine trained without sparsity changes as the top-layer representation is corrupted, the Helmholtz Machines trained with sparsity tend to maintain a single digit identity or have more gradual changes. Specifically, using the L1 norm sparsity constraint appears to provide a more stable latent representation with the addition of noise to the top-layer representation. However, the reconstructed images from this model also show some degradation. On the other hand, using top-k sparsity provides good fidelity in the reconstructed image but with the downside that not all reconstronstructed images might be classified as a digit. Finally, the overcomplete sparsity approach demonstrates its clear advantage where the latent representation appears to be very robust to the added noise. However, there seems to be a tradeoff in that the generated output has coarser resolution.

Although these experiments demonstrate the benefits of sparsity in robustness to noise, there do appear to be differences among the different types of sparsity constraints employed. Often it will be unclear what effects a specific constraint might have when other variables such as the number of hidden units or layers are changed. However, a relatively new regularization approach known as “dropout” seems to reduce overfitting and provide sparse representations in various types of models, including deep neural networks as well as graphical models (Srivastava et al., 2014). Regardless the method employed to encourage sparse representations, sparsity also improves the capacity for storing associative memories and simplifies the representations for hierarchical models to learn complex structure (Olshausen & Field, 2004). These properties are particularly important for studying how the visual cortex represents, stores, and acts upon information.

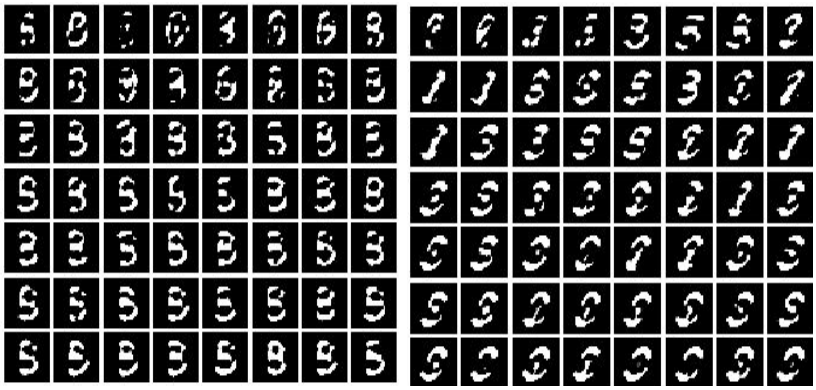
Standard linear MNIST (Layers = 784, 128, 32)



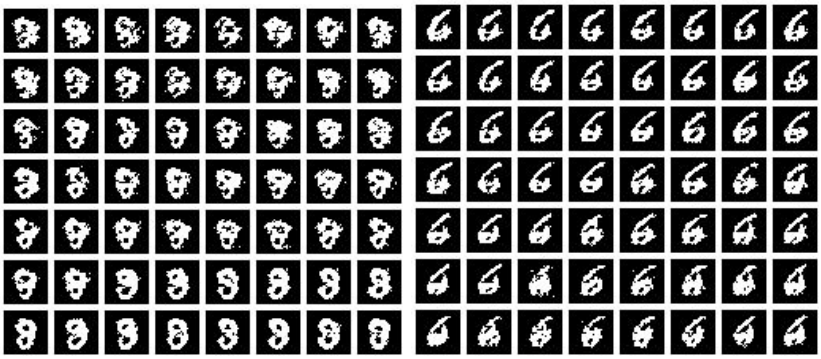
Standard linear MNIST w/ Sparse Constraint (L1 Norm) (Layers = 784, 128, 32)



Sparse linear MNIST w/ Top-k Sparsity (k = 25) (Layers = 784, 128, 32)



Overcomplete Sparse MNIST (L1 Norm) (Layers 784, 784, 784)



References

Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The helmholtz machine. *Neural computation*, *7*(5), 889-904.

Friston, K. (2009). The free-energy principle: a rough guide to the brain?. *Trends in cognitive sciences*, *13*(7), 293-301.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Olshausen, B. A., & Field, D. J. (2004). Sparse coding of sensory inputs. *Current opinion in neurobiology*, *14*(4), 481-487.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.