Learing Associative Fields from Natural Images in a Boltzmann Machine

Wen-Hao Zhang
Carnegie Mellon University

October 6, 2016

1 Model Structure

In order to learn the associative fields of V1 neurons from natural images, we consider a cascade model consisting of a set of Gabor filters followed by a Boltzmann machine (BM). The cascade processing in the model can be regard to the processing from retina to V1 simple cells, and then to V1 complex cells where the associative fields were found [5]. The processing from retina to V1 simple cell is modelled as a set of Gabor filters followed by a nonlinearity to generate neurons' firing rate. And then the firing activities are used as inputs of a Boltzmann machine, which is supposed to learn the associative fields and thus capture the regularities hidden in the inputs. In the following, we will introduce the details of model structure along the direction of information pathway.

1.1 Gabor filters: modeling the process from retina to V1 simple cells

It was found that Gabor filters can well describe the receptive fields of V1 simple cells [4]. In our model, we use a family of Gabor filters,

$$G(x,y,\omega,\theta) \propto \exp\left[\frac{\omega^2}{8\kappa^2}(4x'^2+y'^2)\right]\cos(\omega_0 x') \tag{1}$$

where

$$\begin{aligned} x' &= x\cos\theta + y\sin\theta, \\ y' &= -x\sin\theta + y\cos\theta. \end{aligned}$$

x and y denote the spatial position, and ω and θ specifies the spatial frequency and orientation respectively. κ is a constant controlling the frequency bandwith. Fig. 1 plots the family of Gabor filters used in our model (see the detailed parameters in figure caption). The gabor filters $G(x,y,\omega,\theta)$ are normalized as zero mean and unit variance.

Denotes the image patches in database as $\mathrm{Img}(x,y)$, the family of Gabor filters will convolve with image patch to produce their outputs $I_G(x,y,\omega,\theta)$,

$$I_G(x,y,\omega,\theta) = G(x,y,\omega,\theta) * \mathrm{Img}(x,y), \tag{2}$$

where $*$ denotes a 2D convolution.

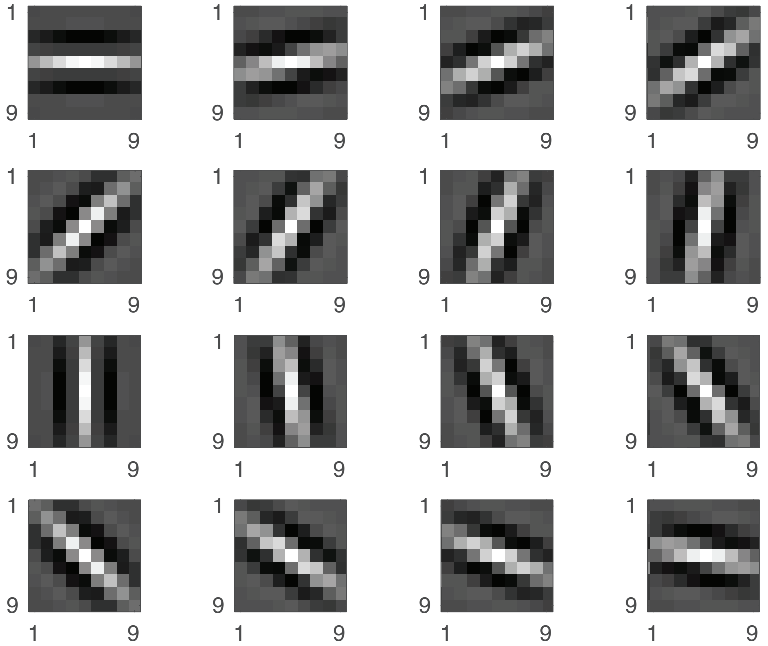

Figure 1: A family of Gabor filters used in this model. Parameters: size [9, 9], spatial frequency $\omega = 4$, frequency bandwidth $\kappa = 2.5$, orientation $\theta = 0 : 11.25 : 180$.

1.2 Nonlinearity: histogram equalization

The outputs of each Gabor filter is then applied into an individual nonlinearity, which is set as the cumulative density function of the outputs of the Gabor filter and then can transform Gabor outputs into an uniform distribution. The output of nonlinearity is then fed into a Poisson spike generator to generate spike trains (binary sequence), which are served as the inputs of a Boltzmann machine. Mathematically, the process of nonlinearity and spike generator is described as

$$\begin{aligned}\mathbf{r}_1(x,y,\omega,\theta) &= F_{cdf}\left[I_G(x,y,\omega,\theta)\right], &(3)\\ \mathbf{x}_1(x,y,\omega,\theta) &= Poiss\left[\mathbf{r}_1(x,y,\omega,\theta)\right], &(4)\end{aligned}$$

where $\mathbf{r}_1(x,y,\omega,\theta)$ and $\mathbf{x}_1(x,y,\omega,\theta)$ can be respectively regarded as the mean firing rate and spike train of V1 simple cell. And $Poiss[\cdot]$ denotes a Poisson spike generator.

The benefit of including a nonlinearity is three-fold: 1) mapping Gabor outputs into a desirable range of firing rate; 2) maximizing the entropy of Gabor outputs; 3) mimicking the nonlinear process of spike generation [7].

1.3 Boltzmann machine: capturing the associative field of V1 complex cell

To capture the correlation structure between spike train of V1 simple cells ($\mathbf{x}_1$), we train a Boltzmann machine to model $\mathbf{x}_1$. The neurons in the Boltzmann machine is composed of two layers: the neurons at 1st layer representing the activities of V1 simple cell $\mathbf{x}_1$, and those in the 2nd layer $\mathbf{x}_2$ are supposed to represent the activities of V1 complex cells. Note that $\mathbf{x}_1$ and $\mathbf{x}_2$ are both spike trains, i.e., binary sequences with values are either 0 or 1. For concise of notation, we omit the dependence of (x,y,ω,θ) in $\mathbf{x}_1$ and $\mathbf{x}_2$ and reshape them as column vectors.

Mathematically, the neuronal activities of the BM satisfying following equation in their equilibrium state,

$$P(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{Z} \exp\left(\mathbf{b}_1^\top \mathbf{x}_1 + \mathbf{b}_2^\top \mathbf{x}_2 + \mathbf{x}_1^\top \mathbf{W}_{12} \mathbf{x}_2 + \mathbf{x}_2^\top \mathbf{W}_{22} \mathbf{x}_2\right). \tag{5}$$

Note that there are no self-connections of a neuron, i.e., the diagonal elements of $\mathbf{W}_{22}$ are all zero. In contrast, compared with common BM [2], the BM we used in this study have some simplifications, in order to simplify the training process and emphasize the rolf of lateral connections in 2nd layer. First, we assume there are no interactions among neurons at 1st layer, i.e., $\mathbf{W}_{11} = 0$. This can be supported by the fact that there are few lateral connections among V1 simple cells in layer 4 of cortex [3]. And the influence of connections are already incorporated in the receptive field depicted as Gabor filters (Eq. 1). Second, we assume the number of neurons at 1st layer is exactly the same as the one in 2nd layer, making the inter-layer connections $\mathbf{W}_{12}$ as a square matrix. Furthermore, to simplify the training process, we further simplify that the neurons in two layers are connected in a one-to-one manner with the same synaptic weight, i.e., $\mathbf{W}_{12} = w_{12}\mathbf{I}$ is a diagonal matrix, where $\mathbf{I}$ is an identity matrix. And $\mathbf{W}_{12}$ is fixed without learning.

1.4 Contrastive divergence mean field learning of a Boltzmann machine

To avoid slow sampling process to reach equilibrium state of a Boltzmann machine, we use a mean field dynamics of the BM and contrastive divergence process to learn connections in a BM [8]. Denote the mean firing rate of $\mathbf{x}_1$ and $\mathbf{x}_2$ by $\mathbf{r}_1$ and $\mathbf{r}_2$ respectively. In mean field learning, the connections are updated by using the mean firing rate of neurons ($\mathbf{r}_1$ and $\mathbf{r}_2$), instead of spike trains ($\mathbf{x}_1$ and $\mathbf{x}_2$). In contrastive divergence process, the mean firing rate of neurons in 1st layer, $\mathbf{r}_1$ is initially set as the inputs, and then the activities of neurons in 1st and 2nd layer are iteratively solved by using the mean field dynamics of BM,

$$\begin{aligned} \mathbf{r}_{2,m} &= \sigma\left(\mathbf{W}_{21}\mathbf{r}_{1,m} + \mathbf{W}_{22}^\top \mathbf{r}_{2,m} + \mathbf{b}_2\right), && (6) \\ \mathbf{r}_{1,m+1} &= \sigma\left(\mathbf{W}_{12}\mathbf{r}_{2,m} + \mathbf{b}_1\right), && (7) \\ \mathbf{r}_{2,m+1} &= \sigma\left(\mathbf{W}_{21}\mathbf{r}_{1,m} + \mathbf{W}_{22}^\top \mathbf{r}_{2,m+1} + \mathbf{b}_2\right), && (8) \end{aligned}$$

where index m denotes the step of contrastive divergence process. And $\sigma(x) = 1/(1+e^{-x})$ is a standard sigmoid function.

Note that in above three equations, the terms in left-hand-side also appear inside the sigmoid function in the right-hand-side, and thus no closed form solutions can be obtained. To solve above equations, we can design a dynamical equation with its stataionary state is the same as above equations and then use Euler method to solve it. For example, to solve $\mathbf{r}_{2,t}$, we use the dynamics as

$$\tau \frac{d\mathbf{r}_{2,m}}{dt} = -\mathbf{r}_{2,m} + \sigma\left(\mathbf{W}_{21}\mathbf{r}_{1,m} + \mathbf{W}_{22}^\top \mathbf{r}_{2,m} + \mathbf{b}_2\right). \tag{9}$$

And the iterative equation of above dynamics by using Euler method is

$$\mathbf{r}_{2,m}(n+1) = \alpha \mathbf{r}_{2,m}(n) + (1-\alpha)\sigma\left(\mathbf{W}_{21}\mathbf{r}_{1,m} + \mathbf{W}_{22}^\top \mathbf{r}_{2,m}(n) + \mathbf{b}_2\right), \tag{10}$$

where $\alpha = 1 - dt/\tau$ is a coefficient controlling the converging speed of finding the solution, and satisfying that $\alpha \in [0,1]$. The solution of $\mathbf{r}_{1,t}$ can be solved in a similar approach.

The connections are updated by using following equations

$$\Delta \mathbf{W}_{22} = \langle \mathbf{r}_{2,m} \mathbf{r}_{2,m}^\top \rangle - \langle \mathbf{r}_{2,0} \mathbf{r}_{2,0}^\top \rangle, \tag{11}$$

$$\Delta \mathbf{b}_i = \langle \mathbf{r}_{i,m} \rangle - \langle \mathbf{r}_{i,0} \rangle, \quad i = 1, 2 \tag{12}$$

where $\langle \cdot \rangle$ denotes an average over traning examples.

2 Model Parameters and Learning Details

2.1 Training dataset

The training images are from Van Hateren's natural image database [1], which contains 4167 images with each of size $[1536, 1024]$. The image patches in training dataset are uniformly sampled from Van Hateren's database. For each image in Van Hateren's database, we randomly sample 40 image pathces with the size of $[49, 49]$. This makes we have more than 160k image patches in our training dataset.

2.2 Gabor filters

The model uses 16 Gabor filters with the same spatial frequency $\omega = 4$ and frequency bandwidth $\kappa = 2.5$, but with different orientations. The orientations of Gabor filters are evenly distributed in the range $[0°, 180°]$. The Gabor filters have the same size of 9×9 pixels, and those from neighbor hypercolumns are half-overlapping in space, i.e., their spatial displacement is $[5, 5]$.

2.3 Boltzmann machine

The Boltzmann machine contains 9×9 hypercolumns aligning on a 2D grid sheet, with each hypercolumn containing 16 neurons. The feedforward weights of the 16 neurons within a hypercolumn are Gabor functions, which are shown in Fig. 1. There are $16 \times 9 \times 9 = 1296$ neurons at 2nd (hidden) layer in the Boltzmann machine.

To speed up learning process, only $\mathbf{W}_{22}$, $\mathbf{b}_1$ and $\mathbf{b}_2$ are learnt from natural images, while the inter-layer connections $\mathbf{W}_{12}$ are fixed.

Initialization of parameters

- $\mathbf{W}_{22}$ is initialized as a Gaussian random variable with zero mean and standard deviation 0.01. The reason to choose this deviation is to make the inputs received by every neurons are located in the lienar range of sigmoid activation function, which is beneficial for fast learning.

- $\mathbf{W}_{12}$ is tailored to make the magnitude of feedforward inputs and recurrent inputs are basically in the same order. I found when when this condition is violated, the learning becomes very slow or even not effective. See Discussion for the explanation of the detailed reason. In order to satisfy this condition, the magnitude of $\mathbf{W}_{12}$ should be scled with the number of neurons in 2nd (hidden) layer of a BM.

- $\mathbf{b}_1$ is initialized as $\ln(\mathbf{x}_1/(1 - \mathbf{x}_1))$ to capture the mean firing rate of inputs. Otherwise the early phase of learning is mainly adjusting the value of $\mathbf{b}_1$.

- $\mathbf{b}_2$ is initialized as a all zero vector.

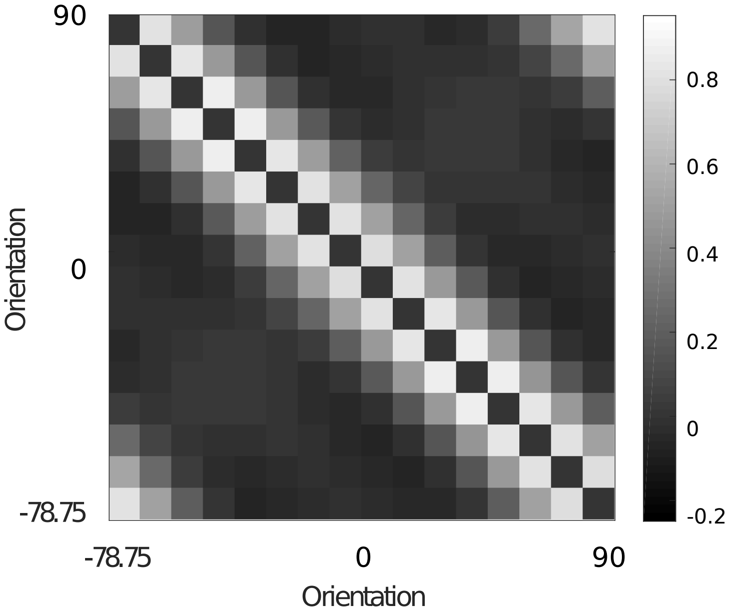

Figure 2: Connections among neurons within an example hypercolumn. Other hypersolumns have similar connection profiles.

Learning parameters

- The learning rate is adjusted to make sure the update of parameters is at the order 10^{-3} times smaller than current parameters. I set learning rate as 0.1 in my model.
- To speed up learning, I used gradient ascent with a momentum with strength of 0.8.
- A slight weight decay (L2 regularization) is used on $\mathbf{W}_{22}$ with strength of 1×10^{-5}.
- Only 1-step contrastive divergence process is used. The coefficient α is set to 0.2 in the solving the deterministic dynamics of BM (Eq. 10).
- Mini-batch learning with each mini-batch contains 100 traning examples. The model is trained by taking 1000 epoches. No validating dataset is used to monitor the learning process, because it is not easy to estimate the likelihood of model on validating dataset. Instead, I monitor reconstruction error of model during learning.

3 Results

Fig. 2 shows the connections among neurons within an example hypercolumn, which is located at the center of all hypercolumns. Overall, the connections within all hypercolumns are very similar with the one shown in Fig. 2. We see the neurons are more tightly connected if their preferred orientations are more similar. Besides, the connection pattern approximately displayes translation-invarient property, i.e., the connection strength between two neurons are only dependent on their difference of preferred orientation.

The learnt connections between an example neuron with neurons from different hypercolumns are also shown in Fig. 3, which is the so called associative field of the example neuron. Fig. 3 plots the associative field of the neuron preferring 0° orientation at the center hypercolumn. Fig. 3A and B are the connections between the example neuron with all other neurons

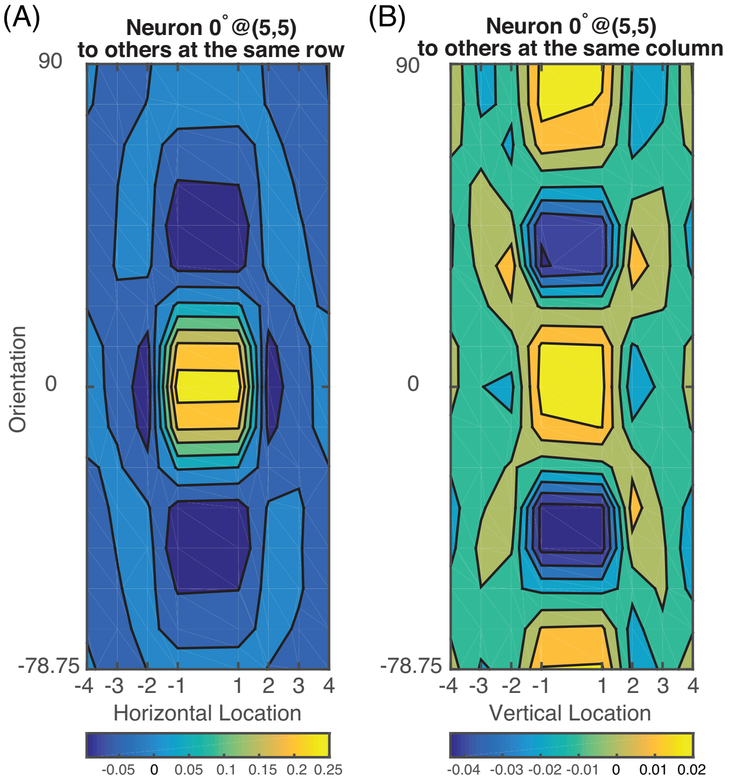

Figure 3: The connections from neurons prefering 0° at central hypercolumn to other neurons from other hypercolumns. Note that the connections between neurons at the same hypercolumn with the example neuron are not shown.

from hypercolumns at the same vertical and horizontal location with the example neuron, respectively. For the connections between neurons at the same horizontal location (Fig. 3A), when two neurons have similar orientation preference and their locatioins are close with each other, they have positive connections, and otherwise negative connections. The connections between the example neurons with neurons at the same vertical locations (Fig. 3B) are smaller than those with neurons at the same horizontal locations (Fig. 3A) by a order. Overall, this result doesn't fully satisfy the associative field used in modelling study, I may consider to refine my model in the future.

4 Discussions and Future Works

4.1 Training of a Boltzmann machine

I found there is a fundamental problem in training a Boltzmann machine and a general recurrent nonlinear network, which was not discussed at all in [8]. The mean field dynamics of a Boltzmann machine is a deterministic nonlinear dynamics (Eq. 9). In theory, when $\mathbf{W}_{22}$ is large enough and with a certain bias $\mathbf{b}_2$, such nonlinear dynamics will emerge hysteresis, i.e., it has two different sets of stationary network activities (attractors).

Fig. 4 explains how the hysteresis comes out in the mean field dynamics. The blue and red lines correspond to the 1st and the 2nd terms in the right hand side of Eq. 9 and their intersections are fixed points of the mean field dynamics. Of all fixed points, the middle one (open circle) is an unstable fixed point, while the others (filled circles) are stable fixed points (attractor). The final attractor the network state will go into is actually depends on the *intensity* of input. For low input intensity, the network will evolve to the attractor with small activity; while the network will go to the higher activity when input intensity is high. For the attractor with small activity, the network is dominated by external input, becaue the network activities will decay to zero once the input is switched off. In contrast, for the attractor with high activity, the network is dominated by its recurrent inputs and it can maintain non-zero activities after switch off input. In this case, the network activities under different inputs are nearly the same, which, in Hinton's view, should be avoided as much as possible.

The learning of connection weight in BM is determined by correlation structure between network activities (Eq. 11). Different stationary network activities have different correlation structures, which will in turn lead to different gradients of connection weight, and then finally lead to different learnt weight in BM. Given an input image, we don't know and cannot control which attractor the network activity will evolve to, and thus it remains a problem of how the weight is determined. It seems that current methods are just set the initial value of recurrent connection weight as small enough compared with feedforward weights, in order to make the network is dominated by feedforward inputs.

Although Hinton thinks the attractor of higher activities should be avoided, I think it may be a good thing, as long as we can actively control which attractors the network will go to. First, two sets of attractors in recurrent network increase the capacity of network's representation. Second, it is potentially interesting to study how to adjust the recurrent weight when network is in its attractor of high activity, because working memory brain areas, e.g., LIP, FEF both face a problem of how to learn their connection weight.

To my best knowledge, there is only one paper investigating this issue [6], but it seems that their main idea is to use a large gradient to escape from the recurrent input dominant attractor (high activity) and use the external input dominant attracotor (low activity). In contrast, I am more interested in introducing another variable which can coordinate two sets of attractors in the network and use them together.

4.2 Likelihood

I haven't estimated the likelihood of trained Boltzmann machine on dataset, because it is in general very difficult to calculate the partition function. In the future, I may try to use some approximated methods to estimate the partition function.

Some alternative way may be used to evaluate the performance of trained model. For example, mimicking the contour integration experiment [5], we can present some small line segments which are located in a colinear line but embeded in a noisy background to the Boltzmann machine, and then to see whether the activities of 2nd (hidden) layer in BM appear the shape of the colinear line.

4.3 Future works

In this study, some hand-designed Gabor filters are used to transform images to the inputs of Boltzmann machine. In the future, we may consider to learn the feedforward connections directly from statistics of images, e.g., by using sparce coding. Futhermore, we may consider a biologically plausible neural network model consisting of excitatory and inhibitory neurons to replace Boltzmann machine in the future.

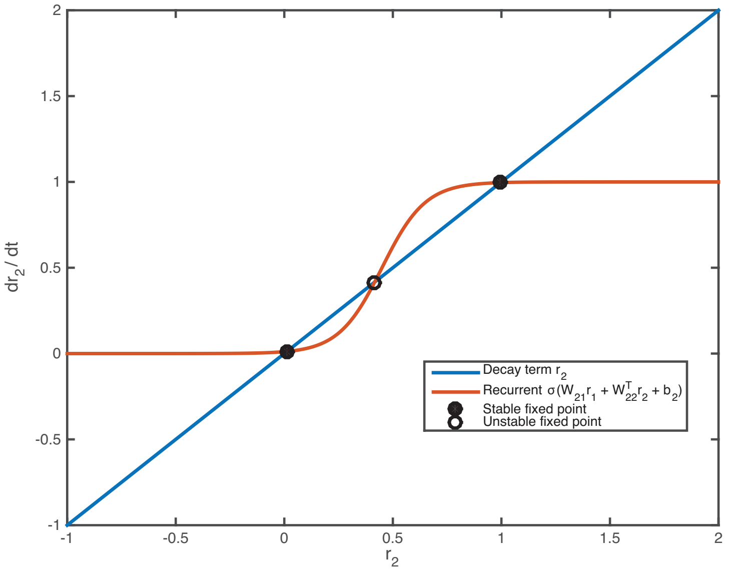

Figure 4: Hysteresis emerges in a Boltzmann machine under some connection weight. Blue and red lines represent the decay and recurrent inputs in Eq. (9). Three circles incidate the fixed points of mean field dynamics of Boltzmann machine, while the open circle represents an unstable fixed point and filled circle is a stable fixed point.

References

[1] J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings: Biological Sciences*, 265(1394):359–366, Mar 1998.

[2] Geoffrey E Hinton and Terrence J Sejnowski. Learning and releaming in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:282–317, 1986.

[3] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven A Siegelbaum, and AJ Hudspeth. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

[4] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on pattern analysis and machine intelligence*, 18(10):959–971, 1996.

[5] Justin NJ McManus, Wu Li, and Charles D Gilbert. Adaptive shape processing in primary visual cortex. *Proceedings of the National Academy of Sciences*, 108(24):9739–9746, 2011.

[6] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318, 2013.

[7] Jonathan W Pillow, Liam Paninski, Valerie J Uzzell, Eero P Simoncelli, and EJ Chichilnisky. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *The Journal of neuroscience*, 25(47):11003–11013, 2005.

[8] Max Welling and Geoffrey E Hinton. A new learning algorithm for mean field boltzmann machines. In *International Conference on Artificial Neural Networks*, pages 351–357. Springer, 2002.