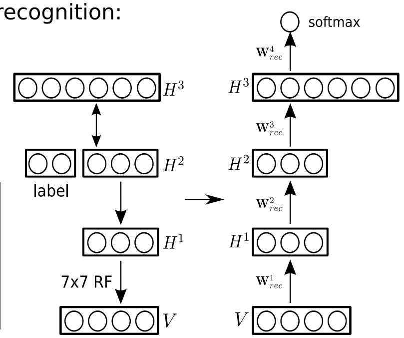
Deep Networks for Robust Visual Recognition

Charlie Tang, Chris Eliasmith
Centre for Theoretical Neuroscience
University of Waterloo
Computational Neuroscience Research Group

Introduction

- Deep Belief Nets are excellent models of high dimensional visual data
- However, they are not robust to simple noise not in the training set
- We introduce two ways to boost recognition:
  1. Sparsely connected first layer
  2. Probabilistic denoising on H^1

MNIST test error for standard DBN [784 500 500 2000 10]



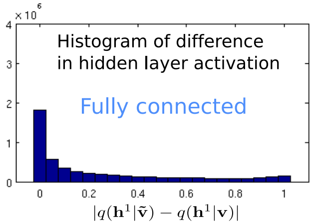
Sparse RBM (Locally Connected)

$$E(\mathbf{v},\mathbf{h};\theta) = -\mathbf{b}^\mathsf{T}\mathbf{v} - \mathbf{c}^\mathsf{T}\mathbf{h} - \mathbf{v}^\mathsf{T}\mathbf{W}\mathbf{h}$$

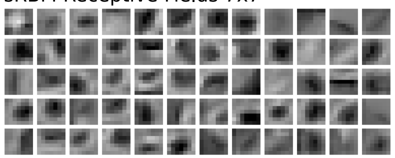
$$p(\mathbf{v},\mathbf{h}) = \frac{p^*(\mathbf{v},\mathbf{h})}{Z(\theta)} = \frac{\exp^{-E(\mathbf{v},\mathbf{h})}}{Z(\theta)}$$

$$\Delta W_{ij} \propto (\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{recons})\widetilde{W}_{ij}$$

$$\widetilde{W}_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is in } h_j\text{'s RF} \\ 0 & \text{otherwise} \end{cases}$$



sRBM Receptive Fields 7x7



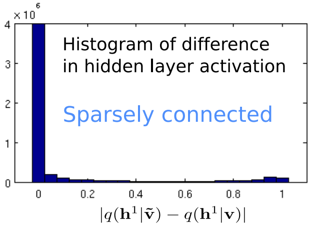
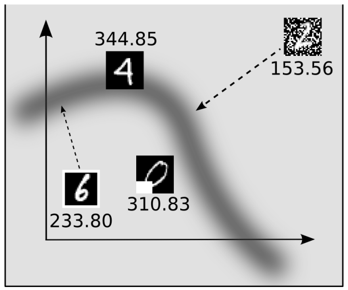


Table 1. Sparse RBM and sparse DBN evaluations

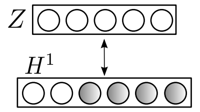
RF size	# hidden	log probability	sDBN error
7x7	500	-94.62	1.19%
	1000	-92.50	1.20%
	1500	-91.77	1.60%
10x10	500	-91.53	1.17%
	1000	-90.16	1.24%
	1500	-89.78	1.55%
12x12	500	-90.30	1.18%
	1000	-89.72	1.16%
	1500	-89.56	1.63%

- *Annealed Importance Sampling to estimate the partition function*
- *Log probability slightly worse than fully connected RBM*
- *Error rate is on clean MNIST after 30 epochs of cross-entropy fine-tuning*

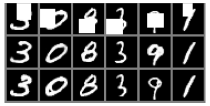
Denoising



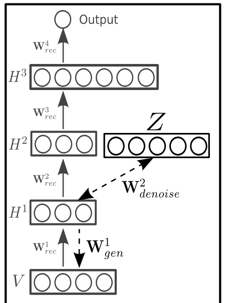
Hypothetical state space with the noisy images and their $\log p^*(\mathbf{h}^1)$



Denoising using Gibbs sampling



Denoising with the knowledge of the occluder



sDBN with denoising layer Z added

Denoising Details

$\mathbf{h}^1_0 = q(\mathbf{h}^1|\tilde{\mathbf{v}})$ ← *bottom up input, computed using recognition weights*

$\nabla_{\psi_j} \log p(\mathbf{h}^1_t) = \log p^*(\widetilde{\mathbf{h}}^1_{\setminus j,t}) - \log p^*(\mathbf{h}^1_t)$ ← *gradient ascent on unnormalized log pr*

$\mathbf{h}^1_{\setminus j}$ is the set of all nodes in H^1 except h^1_j $\quad$ $\widetilde{\mathbf{h}}^1_{\setminus j,t}$ is $\mathbf{h}^1_t$ with the j-th node replaced by $p(h^1_j = 1|\mathbf{h}^1_{\setminus j})$

$$p(h^1_j = 1|\mathbf{h}^1_{\setminus j}) = \frac{\exp(d_j)\prod_k^{N_z}(1+\exp(\phi_k + W^2_{jk}))}{\exp(d_j)\prod_k^{N_z}(1+\exp(\phi_k + W^2_{jk})) + \prod_k^{N_z}(1+\exp(\phi_k))}$$
← *where* $\phi = (\mathbf{W}^2_{\setminus j})^\mathsf{T}\mathbf{h}^1_{\setminus j} + \mathbf{e}$

$$\psi_{j,t} = \begin{cases} 1 & \text{if } \nabla_{\psi_j} \log p(\mathbf{h}^1_t) > \eta(t) \\ 0 & \text{otherwise} \end{cases}$$
← *determine which hidden nodes to unclamp*

$$p(z_k|\mathbf{h}^1) = \sigma\left(\textstyle\sum_j W^2_{jk} h^1_j + e_k\right)$$
$$p(h_j|\mathbf{z}) = \sigma\left(\textstyle\sum_k W^2_{jk} z_k + d_j\right)$$
← *block Gibbs sampling*

$\mathbf{h}^1_{t+1} \longleftarrow \mathbf{g}_t$ ← *update activation for next timestep*

Combining With Bottom Up

- *Attenuate noisy parts of V, with attention-like gating*
- *Neurophysiological evidence for attentional modulation early in visual processing pathway*

$$q(\mathbf{h}^1|\mathbf{v};\mathbf{u}) = \sigma\left((\mathbf{W}^1_{rec})^\mathsf{T}(\mathbf{v}\odot\mathbf{u}) + \mathbf{c}\right)$$
attenuated bottom up influence

$$\mathbf{u} = 1 - |\mathbf{v} - p(\mathbf{v}|\mathbf{g};\mathbf{W}^1_{gen})|$$
feedback gating signal

$$\mathbf{g}_{combined} = q(\mathbf{h}^1|\mathbf{v};\mathbf{u}) \odot \frac{\mathbf{u}^\mathsf{T}\widetilde{\mathbf{W}}}{\gamma^2} + \mathbf{g}\odot\left(1 - \frac{\mathbf{u}^\mathsf{T}\widetilde{\mathbf{W}}}{\gamma^2}\right)$$
combine bottom up and top down

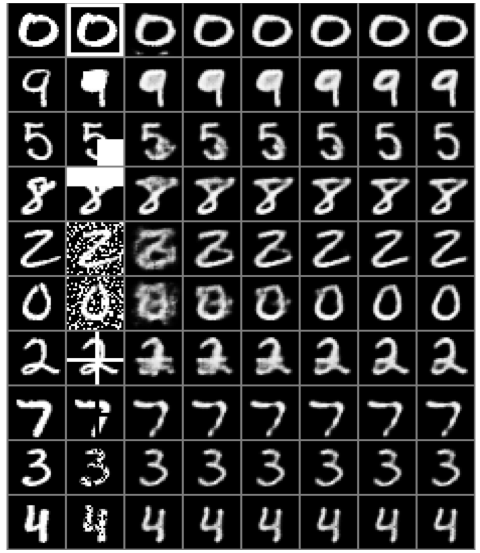
$$\mathbf{h}^1_{t+1} \longleftarrow \mathbf{g}_{combined,t}$$

Algorithm 1 Sparse DBN Training and Inference

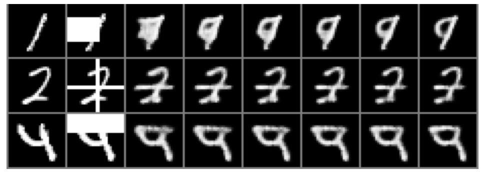
Learning:
1: Learn sRBM using eq. 4.
2: Greedy pretraining of higher layer RBMs and stack to form a sDBN.
3: Fine tune using the up-down algorithm.
4: Convert the sDBN into a discriminative classifier and minimize cross-entropy error.
5: Learn $\mathbf{W}^2_{denoise}$ using $q(\mathbf{h}^1|\mathbf{v})$ as input.
6: Learn $\mathbf{W}^1_{gen}$ by minimizing cross-entropy between the data and $p(\mathbf{v}|q(\mathbf{h}^1|\mathbf{v}))$.

Recognition:
1: For noisy input $\tilde{\mathbf{v}}$, compute $\mathbf{h}^1_0 = q(\mathbf{h}^1|\tilde{\mathbf{v}})$.
for $t = 1$ to n **do**
2: Estimate $\boldsymbol{\psi}_t$ using eq. 12
3: Gibbs sampling to obtain $\mathbf{g}_t$ using eq. 8
4: Combine with bottom up input to obtain $\mathbf{g}_{combined,t}$ using Eq. 16
5: $\mathbf{h}^1_{t+1} \longleftarrow \mathbf{g}_{combined,t}$
end for
6: Compute $q(\mathbf{h}^2|\mathbf{h}^1_{n+1})$, then feedforward to output.

Results



Successful



Failed

Table 2. Summary of recognition results

Network	clean	border	block	random
28x28 DBN	1.03%	66.14%	33.78%	79.83%
7x7 sDBN	1.19%	2.46%	21.84%	65.50%
7x7+denoised	**1.24%**	**1.29%**	**19.09%**	**3.83%**
28x28+noise	1.68%	1.95%	8.72%	8.01%
7x7+noise	1.61%	1.77%	8.39%	6.64%

Conclusions

- Sparse connections and denoising improves recognition
- Knowledge of the occluder greatly facilitates denoising
- In many cases, less error with denoising than after supervised training including noisy images
- Tradeoff between speed and accuracy (feedforward only or with feedback)