Theoretical Remarks on Deep Belief Networks

Nicolas Le Roux

August 18, 2006

CIAR Summer School 2006

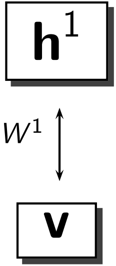
Disclaimer

All the theoretical results presented here go against Geoff's intuitions. As there is a strong prior on where the truth lies, this presentation should be considered as pure entertainment.

Motivation

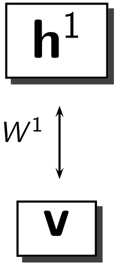
1. Justify the CD criterion with theoretical results
2. Take advantage of the knowledge of the final number of layers in the DBN

Why CD is a good thing?



1. $p(\mathbf{h}^1)$ is the marginal associated to the RBM.

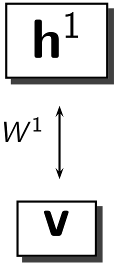
Why CD is a good thing?



1. $p(\mathbf{h}^1)$ is the marginal associated to the RBM.
2. the best weights are those who maximize

$$ML = \sum_{\mathbf{v}} p_0(\mathbf{v}) \log \left(\sum_{\mathbf{h}^1} p\left(\mathbf{v}, \mathbf{h}^1\right) \right)$$

Why CD is a good thing?

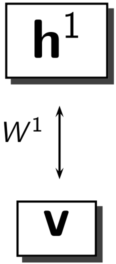


1. $p(\mathbf{h}^1)$ is the marginal associated to the RBM.
2. the best weights are those who maximize

$$ML = \sum_{\mathbf{v}} p_0(\mathbf{v}) \log \left(\sum_{\mathbf{h}^1} p\left(\mathbf{v}, \mathbf{h}^1\right) \right)$$

3. maximizing ML leads to good features

Why CD is a good thing?

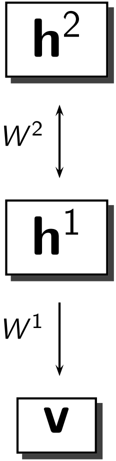


1. $p(\mathbf{h}^1)$ is the marginal associated to the RBM.
2. the best weights are those who maximize

$$ML = \sum_{\mathbf{v}} p_0(\mathbf{v}) \log \left(\sum_{\mathbf{h}^1} p\left(\mathbf{v}, \mathbf{h}^1\right) \right)$$

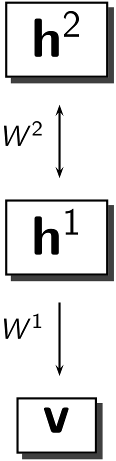
3. maximizing ML leads to good features
4. CD is faster than ML

What are the “problems” of CD?



1. $p(\mathbf{h}^1)$ is NOT the marginal associated to the RBM

What are the "problems" of CD?



1. $p(\mathbf{h}^1)$ is NOT the marginal associated to the RBM
2. CD is a "trick" to speed up training and reduce variance

Why the greedy procedure?

1. There are strong dependencies between W^1, W^2, ...

Why the greedy procedure?

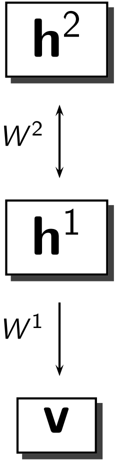
1. There are strong dependencies between W^1, W^2, ...
2. The obtained solution leads to good features

Why the greedy procedure?

1. There are strong dependencies between W^1, W^2, ...
2. The obtained solution leads to good features

- Could we remove the dependencies between the W's?

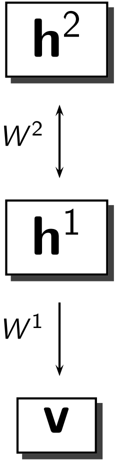
The variational bound



- Variational bound:

$$
\begin{aligned}
p(\mathbf{v}) &\geq C(W^1) + \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1|\mathbf{v}_0) \log p(\mathbf{h}^1) \\
p^*(\mathbf{h}^1) &= \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1|\mathbf{v}_0)
\end{aligned}
$$

The variational bound



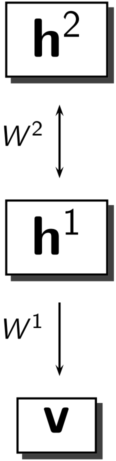
- Variational bound:

$$p(\mathbf{v}) \geq C(W^1) + \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1 | \mathbf{v}_0) \log p(\mathbf{h}^1)$$

$$p^*(\mathbf{h}^1) = \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1 | \mathbf{v}_0)$$

- Given $p(\mathbf{h}^1)$, W^1 is independent from the other W

The variational bound



- Variational bound:

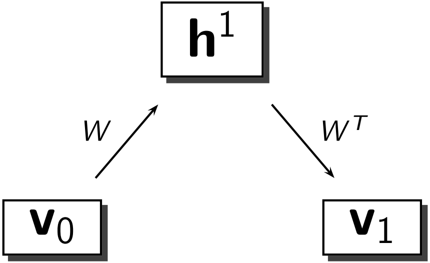
$$
\begin{aligned}
p(\mathbf{v}) &\geq C(W^1) + \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1 | \mathbf{v}_0) \log p(\mathbf{h}^1) \\
p^*(\mathbf{h}^1) &= \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1 | \mathbf{v}_0)
\end{aligned}
$$

- Given $p(\mathbf{h}^1)$, W^1 is independent from the other W
- What is the best W^1?

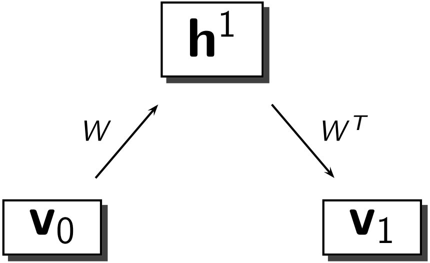
Optimal model distribution

$$\begin{aligned}
p^*(\mathbf{h}^1) &= \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1|\mathbf{v}_0) \\
p^*(\mathbf{v}) &= \sum_{\mathbf{h}^1} p^*(\mathbf{h}^1) P(\mathbf{v}|\mathbf{h}^1) \\
p^*(\mathbf{v}) &= \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1|\mathbf{v}_0) P(\mathbf{v}|\mathbf{h}^1)
\end{aligned}$$

One-step RBM

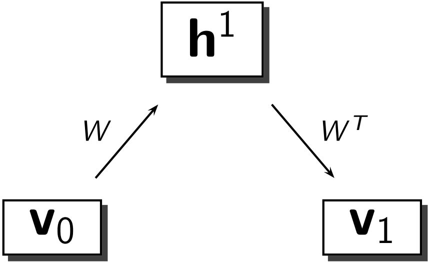


One-step RBM



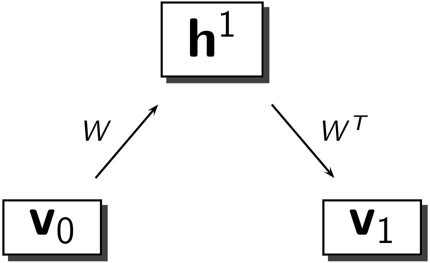
1. $p^*(\mathbf{v}_1) = \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1 | \mathbf{v}_0) P(\mathbf{v}_1 | \mathbf{h}^1)$

One-step RBM



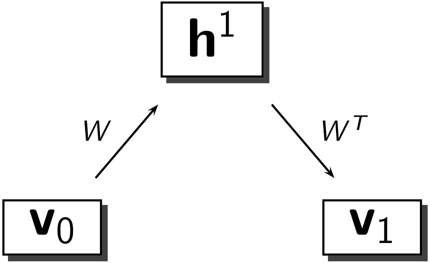
1. $p^*(\mathbf{v}_1) = \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1|\mathbf{v}_0) P(\mathbf{v}_1|\mathbf{h}^1)$
2. $p^*(\mathbf{v}_1) = p_1(\mathbf{v}_1)$

One-step RBM



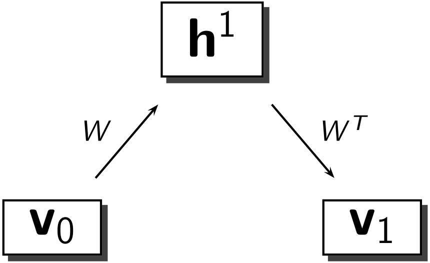
1. $p^*(\mathbf{v}_1) = \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1|\mathbf{v}_0) P(\mathbf{v}_1|\mathbf{h}^1)$
2. $p^*(\mathbf{v}_1) = p_1(\mathbf{v}_1)$
3. Minimizing the likelihood is minimizing $KL(p_0||p_1)$

One-step RBM



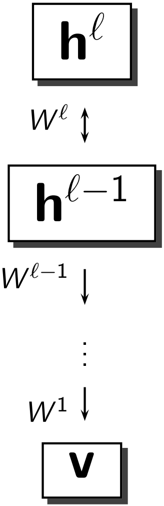
1. $p^*(\mathbf{v}_1) = \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1|\mathbf{v}_0) P(\mathbf{v}_1|\mathbf{h}^1)$
2. $p^*(\mathbf{v}_1) = p_1(\mathbf{v}_1)$
3. Minimizing the likelihood is minimizing $KL(p_0||p_1)$
4. $KL(p_0||p_1) \approx KL(p_0||p_\infty) - KL(p_1||p_\infty)$

One-step RBM



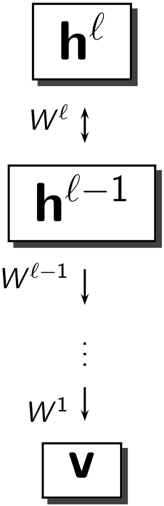
1. $p^*(\mathbf{v}_1) = \sum_{\mathbf{h}^1} \sum_{\mathbf{v}_0} p_0(\mathbf{v}_0) Q(\mathbf{h}^1|\mathbf{v}_0) P(\mathbf{v}_1|\mathbf{h}^1)$
2. $p^*(\mathbf{v}_1) = p_1(\mathbf{v}_1)$
3. Minimizing the likelihood is minimizing $KL(p_0||p_1)$
4. $KL(p_0||p_1) \approx KL(p_0||p_\infty) - KL(p_1||p_\infty)$
5. Exact gradient does not depend on $p_\infty \implies$ fast !

And then?



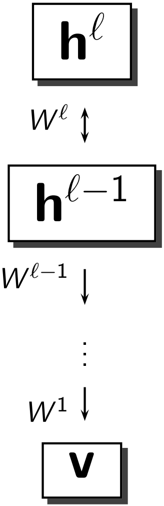
1. If we had the best marginal $p^*(\mathbf{h}^1)$, we'd have the perfect W^1

And then?



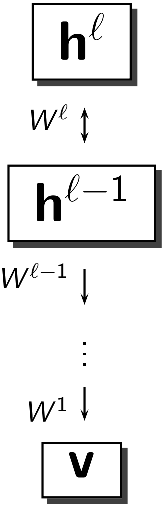
1. If we had the best marginal $p^*(\mathbf{h}^1)$, we'd have the perfect W^1
2. Let's try to be as close as $p^*(\mathbf{h}^1)$ as possible

And then?



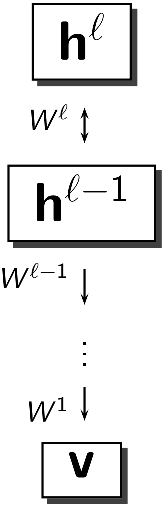
1. If we had the best marginal $p^*(\mathbf{h}^1)$, we'd have the perfect W^1
2. Let's try to be as close as $p^*(\mathbf{h}^1)$ as possible
3. $p^*(\mathbf{h}^1)$ is exactly what we obtain if we clamp the empirical distribution and go through W^1

And then?



1. If we had the best marginal $p^*(\mathbf{h}^1)$, we'd have the perfect W^1
2. Let's try to be as close as $p^*(\mathbf{h}^1)$ as possible
3. $p^*(\mathbf{h}^1)$ is exactly what we obtain if we clamp the empirical distribution and go through W^1
4. Continue until the last layer

And then?



1. If we had the best marginal $p^*(\mathbf{h}^1)$, we'd have the perfect W^1
2. Let's try to be as close as $p^*(\mathbf{h}^1)$ as possible
3. $p^*(\mathbf{h}^1)$ is exactly what we obtain if we clamp the empirical distribution and go through W^1
4. Continue until the last layer
5. Train it using CD (it is a usual RBM)

A few problems

1. $p^*\left(\mathbf{h}^1\right)$ is obtained using the variational bound and not the true likelihood

A few problems

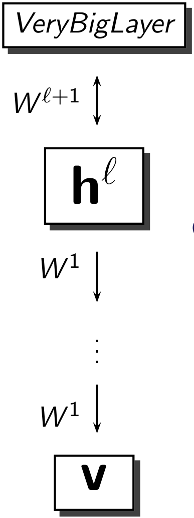
1. $p^* \left(\mathbf{h}^1\right)$ is obtained using the variational bound and not the true likelihood
2. Minimizing $KL\left(p^* \left(\mathbf{h}^1\right) || p\left(\mathbf{h}^1\right)\right)$ does not guarantee to improve the variational bound

A few problems

1. $p^*\left(\mathbf{h}^1\right)$ is obtained using the variational bound and not the true likelihood
2. Minimizing $KL\left(p^*\left(\mathbf{h}^1\right)||p\left(\mathbf{h}^1\right)\right)$ does not guarantee to improve the variational bound

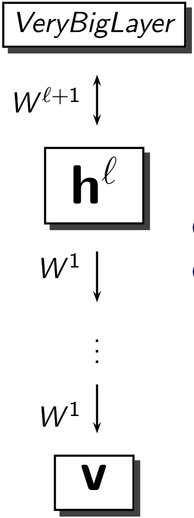
- Could we adapt that framework such that the guarantee remains?

Using a very big top hidden layer



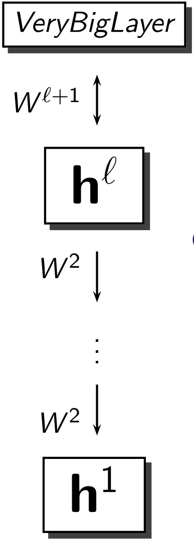
1. Any marginal on $\mathbf{h}^\ell$

Using a very big top hidden layer



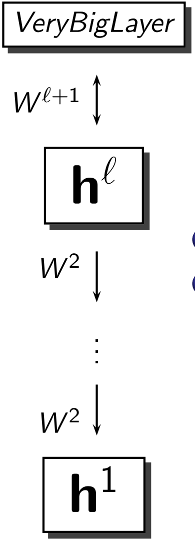
1. Any marginal on $\mathbf{h}^{\ell}$
2. Maximizing the likelihood of the data needs minimizing $KL\left(p_0^0 || p_\ell^0\right) \longrightarrow W^1$

Using a very big top hidden layer (2)



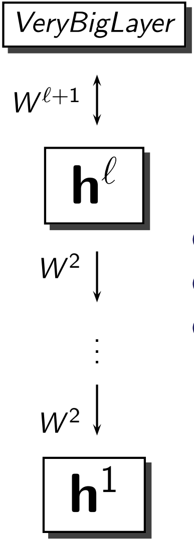
1. Compute $p_0^1(\mathbf{h}^1)$ from $p_0^0(\mathbf{v})$ and W^1

Using a very big top hidden layer (2)



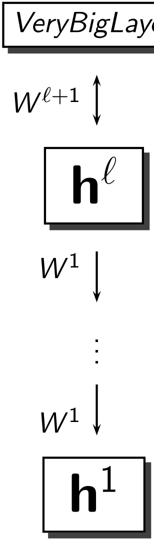
1. Compute $p_0^1(\mathbf{h}^1)$ from $p_0^0(\mathbf{v})$ and W^1
2. Minimize $KL\left(p_0^1(\mathbf{h}^1)||p_{\ell-1}^1(\mathbf{h}^1)\right)$

Using a very big top hidden layer (2)



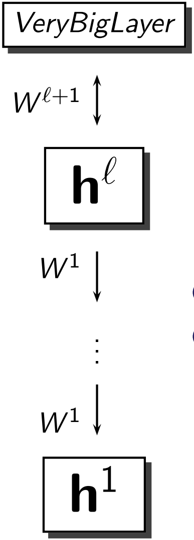
1. Compute $p_0^1(\mathbf{h}^1)$ from $p_0^0(\mathbf{v})$ and W^1
2. Minimize $KL\left(p_0^1(\mathbf{h}^1)||p_{\ell-1}^1(\mathbf{h}^1)\right)$
3. Iterate

Layers are regularizers



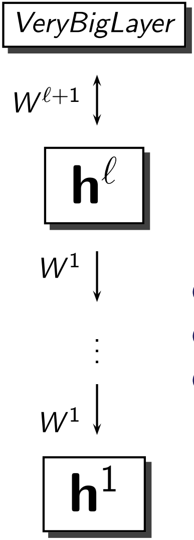
1. We have any marginal on $\mathbf{h}^\ell$

Layers are regularizers



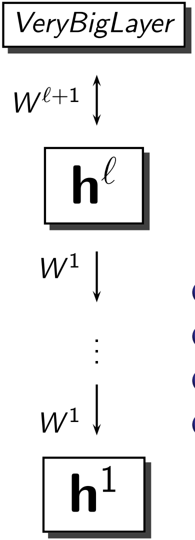
1. We have any marginal on $\mathbf{h}^\ell$
2. This architecture suggests memorizing high-level features

Layers are regularizers



1. We have any marginal on $\mathbf{h}^{\ell}$
2. This architecture suggests memorizing high-level features
3. To sample, you just need to do a forward-backward pass

Layers are regularizers



1. We have any marginal on $\mathbf{h}^{\ell}$
2. This architecture suggests memorizing high-level features
3. To sample, you just need to do a forward-backward pass
4. Going through the layers adds noise and regularizes

Summary and Conclusion

- We now have

Summary and Conclusion

- We now have

1. A new ungreedy training procedure for the DBN

Summary and Conclusion

- We now have
  1. A new ungreedy training procedure for the DBN
  2. A justification of that procedure

Summary and Conclusion

- We now have

1. A new ungreedy training procedure for the DBN
2. A justification of that procedure
3. An extension to take into account the final number of layers

Summary and Conclusion

- We now have
  1. A new ungreedy training procedure for the DBN
  2. A justification of that procedure
  3. An extension to take into account the final number of layers
- But there are a few open questions

Summary and Conclusion

- We now have
  1. A new ungreedy training procedure for the DBN
  2. A justification of that procedure
  3. An extension to take into account the final number of layers
- But there are a few open questions
  1. Can we use the ML instead of the variational bound to find $p^*\left(\mathbf{h}^i\right)$?

Summary and Conclusion

- We now have
  1. A new ungreedy training procedure for the DBN
  2. A justification of that procedure
  3. An extension to take into account the final number of layers
- But there are a few open questions
  1. Can we use the ML instead of the variational bound to find $p^*\left(\mathbf{h}^i\right)$?
  2. Do we have the same guarantee as with ML?

Summary and Conclusion

- We now have
  1. A new ungreedy training procedure for the DBN
  2. A justification of that procedure
  3. An extension to take into account the final number of layers
- But there are a few open questions
  1. Can we use the ML instead of the variational bound to find $p^*\left(\mathbf{h}^i\right)$?
  2. Do we have the same guarantee as with ML?
  3. What is the set of distributions we can model with a ℓ-layer DBN?

Summary and Conclusion

- We now have
  1. A new ungreedy training procedure for the DBN
  2. A justification of that procedure
  3. An extension to take into account the final number of layers
- But there are a few open questions
  1. Can we use the ML instead of the variational bound to find $p^*\left(\mathbf{h}^i\right)$?
  2. Do we have the same guarantee as with ML?
  3. What is the set of distributions we can model with a ℓ-layer DBN?
  4. Why do I keep getting results opposed to what Geoff finds?