Knowledge Extraction From Deep Belief Networks

Son N. Tran
Supervisors: Artur d'Avila Garcez
Jason Dykes

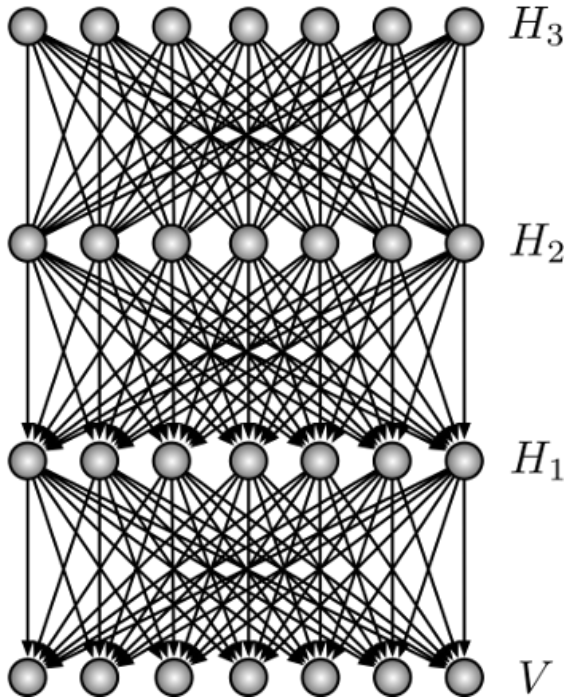
[1]Department of Computing
City University London

17-Dec: MPhil/PhD Transfer Presentation

Contents

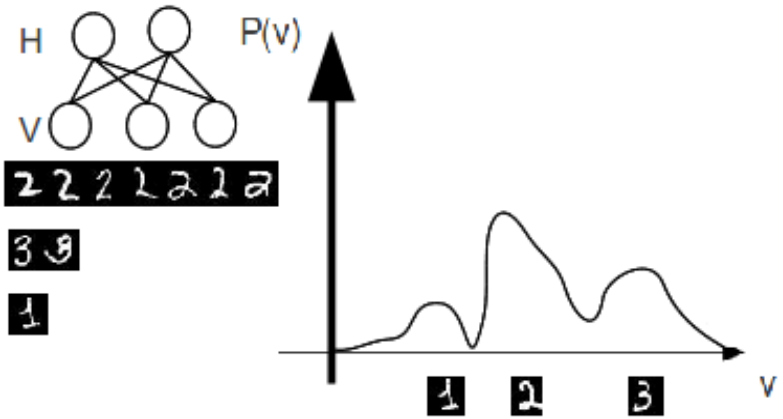
Logic and Deep Architectures

- Deep architectures shows good performance in Vision and Sound
  - Deep Belief Networks (DBN)
  - Stacked Autoassociator
  - Convolutional Neural Networks
- We want to study the role of logic in DBN to find out:
  - Why being "deep" is good?
  - What knowledge is captured in deep architectures
  - How to transfer this knowledge to another domain

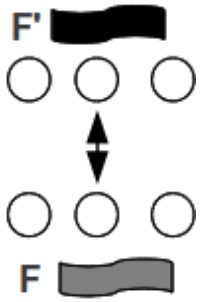


Restricted Boltzmann Machine (RBM)

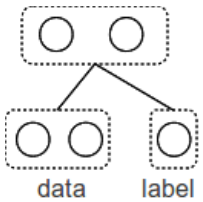
- Main component of DBN which represent the data distribution (a)
- Applications of RBM
  - Feature extraction (b)
  - Classification (c)
  - Construct Deep Belief Networks (d)



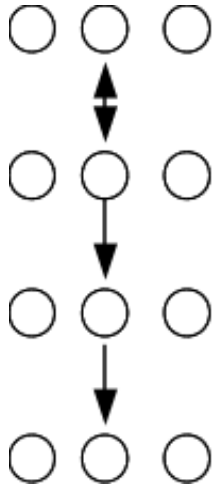
(a)



(b)



(c)

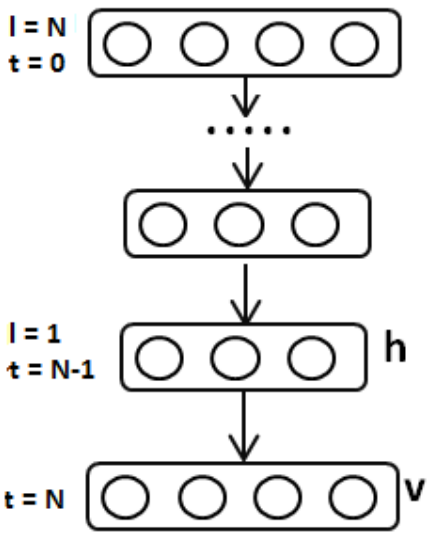


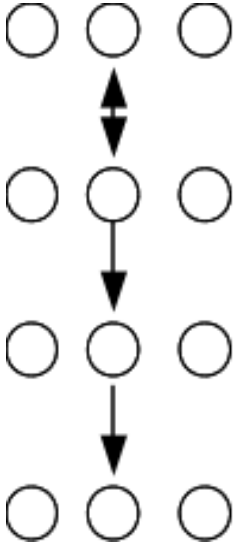
(d)

Deep Belief Networks

Deep Belief Networks [Hinton JNC2006]

- Directed graphical model (Bayesian Nets)
- Very large (infinite) number of layer
- The higher layers can be replaced by undirected model: RBM





Logic Extraction From RBM: Related works

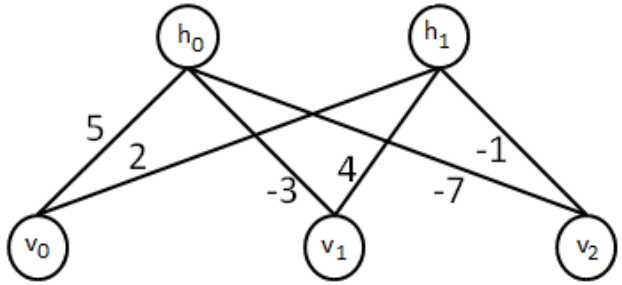
- Study the extraction in RBM and extend it to DBN
- Rules extraction using penalty logic theory [Pinkas JAI1995]
- Temporal rule extraction with sampling [Leo et al. IJCAI2011]
- Rule extraction using confidence-value [Son & Artur ICML_WS2012]

Partial-model

- Extracted rule $c_j : h_j \leftrightarrow \bigwedge_{w_{tj}>0} v_t \wedge \bigwedge_{w_{kj}<0} \neg v_k$
- $c_j = f(\sum_{w_{ij}>0} w_{ij} - \sum_{w_{tj}<0} w_{tj})$
- f is monotonically-increasing function (usually logistic function is used)
- Also apply to visible units v_i
- Example

$15 : h_0 \leftrightarrow v_1 \wedge \neg v_2 \wedge \neg v_3$

$7 : h_1 \leftrightarrow v_1 \wedge v_2 \wedge \neg v_3$



- These rules are named as *partial-model* because they partially capture the achitecture and behavior of the system

Partial-model with ground-truth (1)

Suppose that we want to learn the relationship between:

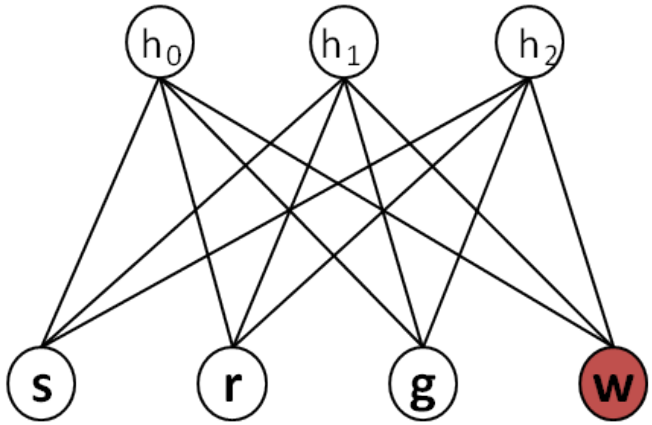
w : A dog is wet $(yes = 1, no = 0)$

g : It was playing in the garden $(yes = 1, no = 0)$

r : It was raining $(yes = 1, no = 0)$

s : The sprinkler was on $(yes = 1, no = 0)$

Partial-model with ground-truth (2)



Encode these variables into RBM and training, we may get some rules as following:

$9999 : h_0 \leftrightarrow w \wedge g \wedge \neg s \wedge r$

$9999 : h_1 \leftrightarrow w \wedge g \wedge s \wedge \neg r$

$9999 : h_2 \leftrightarrow w \wedge \neg g \wedge s \wedge r$

But we want the relationship between variables in the domain!!!

Partial-model with ground-truth (3)

Assumptions

Assumption 1: When the confidence value is big enough the rule is **believed** to be true.

Assumption 2: If there is a ground-truth in the conjunction and the head is positive literal we can devide the rule into two implication rules

$$9999 : h_1 \leftrightarrow w \wedge g \wedge \neg s \wedge r$$

$$\overset{\propto}{\Rightarrow} \top \leftrightarrow w \wedge g \wedge \neg s \wedge r$$

$$\overset{\propto}{\Rightarrow} \{w \leftarrow g \wedge \neg s \wedge r \; ; \; w \rightarrow g \wedge \neg s \wedge r\}$$

Combining rules-1

Combining rules 1: Two rules have a single different conjunct can be combined by removing this conjunct.

$$w \leftarrow g \wedge \neg s \wedge r$$

$$w \leftarrow g \wedge s \wedge r$$

$$w \leftarrow g \wedge r$$

Combining rules 2: If two rules have same ground-truth then it can be combined by disjunction

$$w \leftarrow g \wedge r$$

$$w \leftarrow g \wedge s$$

$$w \leftarrow (g \wedge r) \vee (g \wedge s)$$

Translation of combined rules

$$w \leftarrow (g \wedge r) \vee (g \wedge s)$$
$$\Leftrightarrow w \leftarrow g \wedge (r \vee s)$$

If a dog was playing in the garden when it was raining or the sprinkler was on then it is wet

$$w \rightarrow (g \wedge r) \vee (g \wedge s)$$
$$\Leftrightarrow w \rightarrow g \wedge (r \vee s)$$

If a dog is wet then it should have been playing in the garden when it was raining or the sprinkler was on

Inference

- Inference with *partial-models* is difficult!

$$15 : h_1 \leftrightarrow v_1 \wedge \neg v_2 \wedge \neg v_3$$

$$v_1, v_2, v_3$$

---

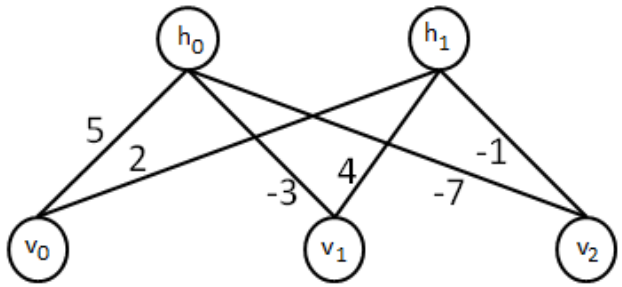
h_1 holds with confidence-value c =?

Complete-model

- Confidence-vector: $hv_j = \{|w_{1j}|, |w_{2j}|, ...\}$
- Complete-model: $c_j : h_j \overset{hv_j}{\leftrightarrow} \bigwedge_{i,w_{ij}>0} v_i \wedge \bigwedge_{t,w_{tj}<0} \neg v_t$

$15 : h_0 \overset{hv_0:[5,3,7]}{\leftrightarrow} v_1 \wedge \neg v_2 \wedge \neg v_3$

$07 : h_1 \overset{hv_1:[2,4,1]}{\leftrightarrow} v_1 \wedge v_2 \wedge \neg v_3$



Partial-model and Complete-model

Partial-models

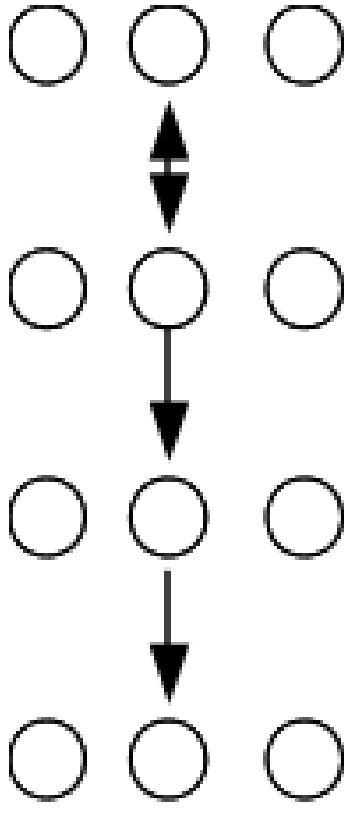
- Pros: Simple, approximate relations of units in systems
- Cons: Hard to use for inference

Complete-models

- Pros: Good for inference
- Cons: Too heavy and have redundant elements

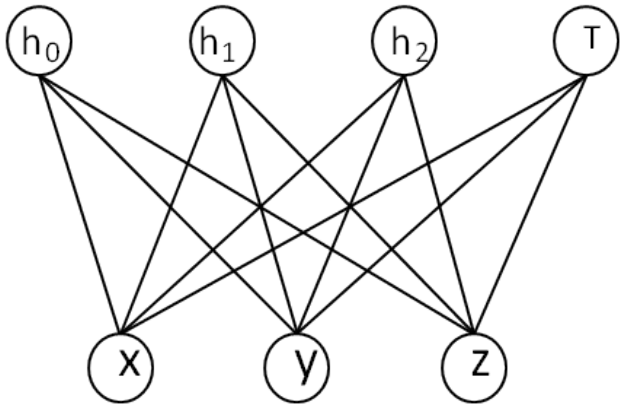
Logic Extraction From DBN

- Deep Belief Networks
- Rules combination
- Logic inference
- Herding [Welling ICML2009]



Outline

XOR's problem



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

$$\mathrm{W} = \begin{pmatrix} -10.0600 & 3.9304 & -9.8485 \\ 9.6408 & 9.5271 & -7.5398 \\ 5.0645 & -9.9315 & -9.8054 \end{pmatrix}$$

$$\mathrm{visB} = [4.5196 \ -4.3642 \ 4.5371]^{\top}$$

$$25 : h_0 \leftrightarrow \neg x \wedge y \wedge z$$
$$23 : h_1 \leftrightarrow x \wedge y \wedge \neg z$$
$$27 : h_2 \leftrightarrow \neg x \wedge \neg y \wedge \neg z$$
$$13 : \top \leftrightarrow x \wedge \neg y \wedge z$$

If z is ground-truth then the combined rule is:

$$z \leftarrow (x \wedge \neg y) \vee (\neg x \wedge y)$$

Knowledge in Handwritten digits (MNIST)-1

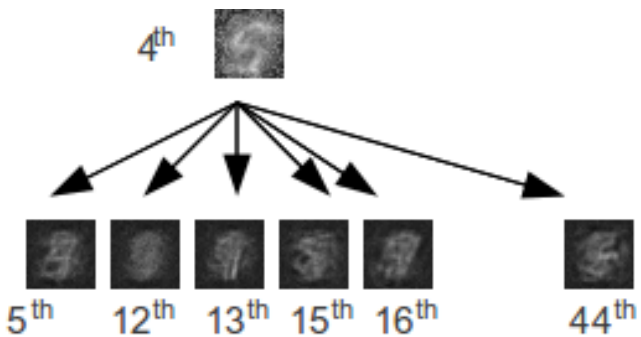
- Rule evaluation using confidence-value

(Demo...)

Knowledge in Handwritten digits (MNIST)-2

- Logical inference vs. Stochastic inference

- Explaining features relation

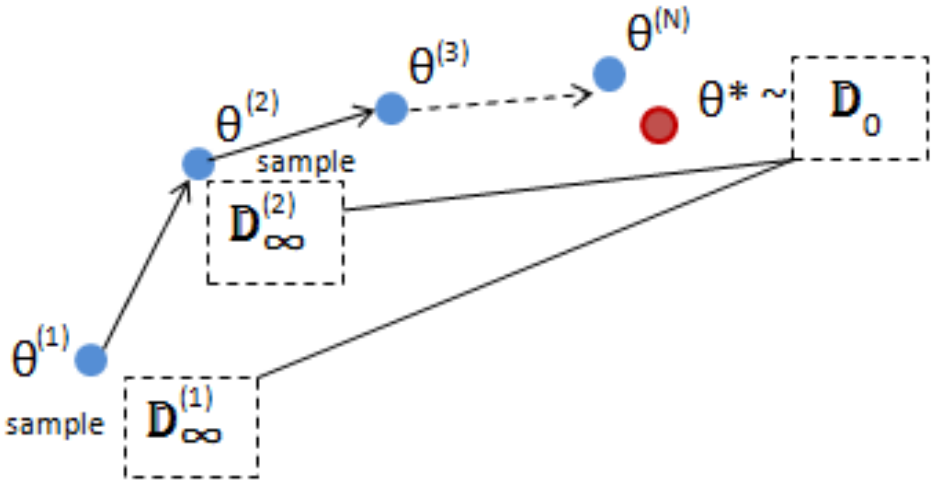


Outline

Contrastive Divergence

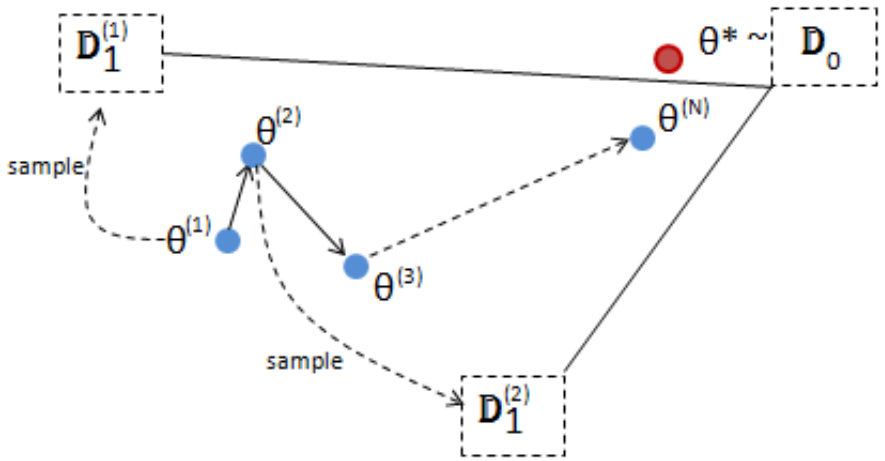
Learning RBM by MCMC

$\Delta w_{ij} = \langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_\infty$

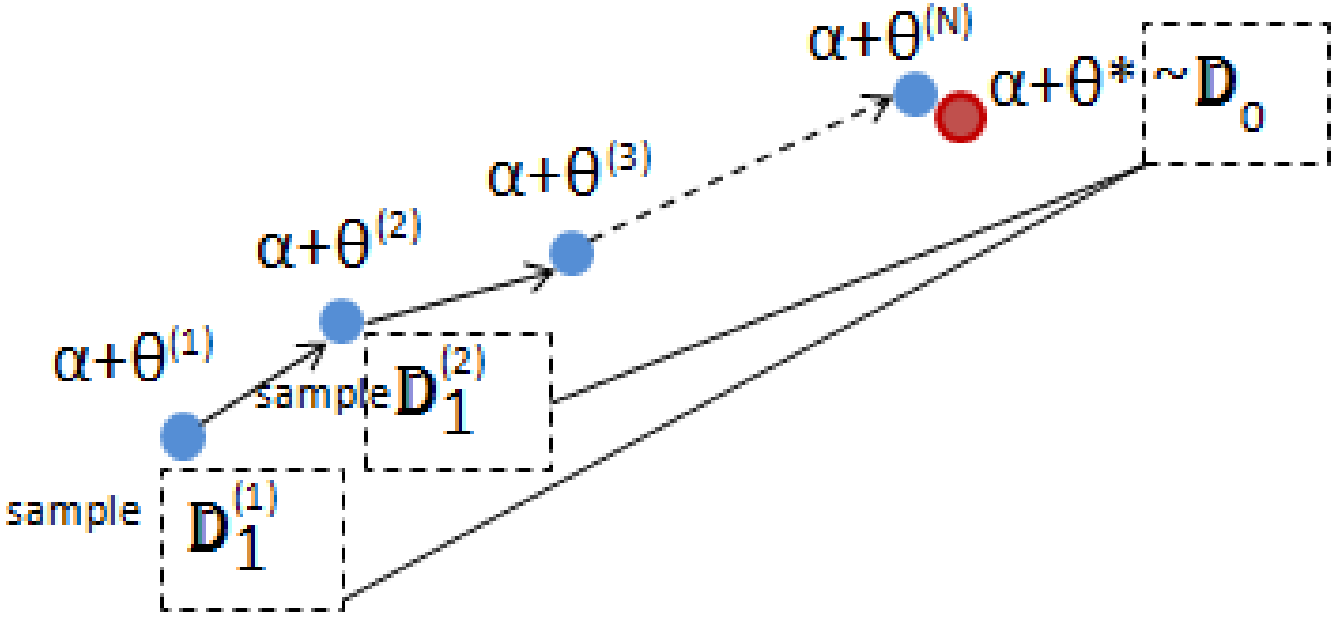


Learning RBM by CD

$\Delta w_{ij} = \langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_1$



Guiding CD Using Background-Knowledge



Background knowledge can be used to guide the learning

- Faster
- Converge to better approximation of desire model (possibly)

Experiment With XOR

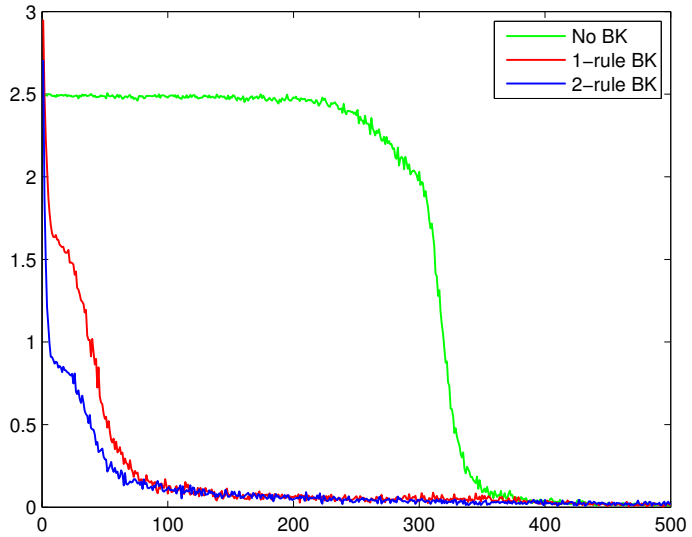
- Learning a RBM from XOR's truth table

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

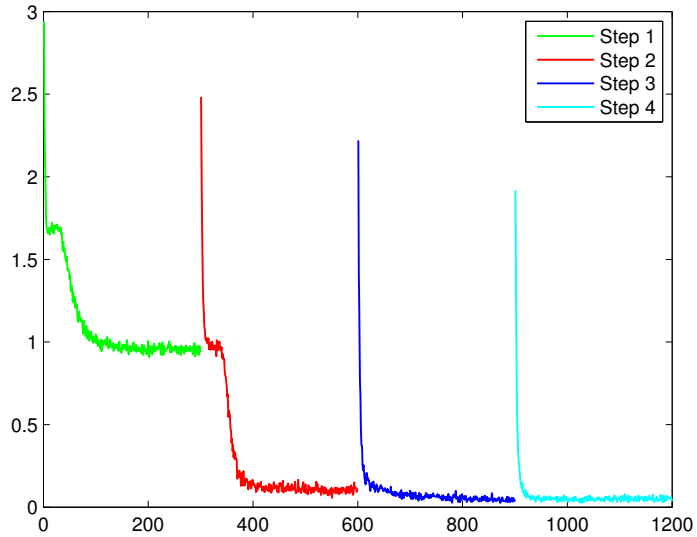
Table: XOR truth table

- What if training with some rules are given?

Experiment With XOR



(a) Learning XOR with different size of prior knowledge



(b) Learning XOR by incremental guided learning

Outline

Guiding Without Background-Knowledge

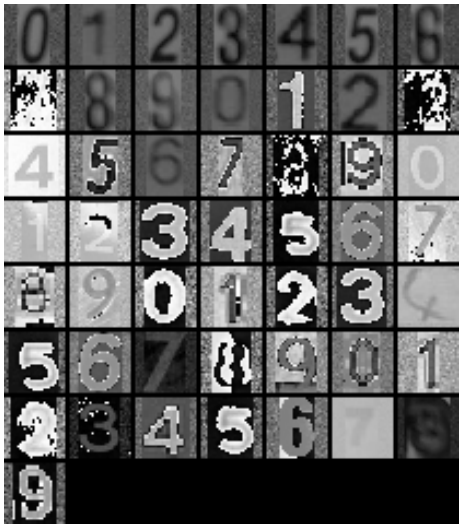
Problems in Machine Learning:

- Data in problem domain is limited
- Data in problem domain is difficult to label
- Prior knowledge in problem domain is hard to obtain

Solution: Learn the knowledge from unlabelled data from related domains which are largely available and transfer the knowledge to the problem domain.

Transfer Knowledge: From Handwritten Digit To Natural Digit Images

(a) MNIST dataset



(b) ICDAR dataset

Figure: Visualization of MNIST images (a) and ICDAR images (b)

Experiment-1

Raw features (pixel)	RBM	RBM with transfer
31.9588%	35.567%	**36.0821%**

Table: Classification accuracy using the different features

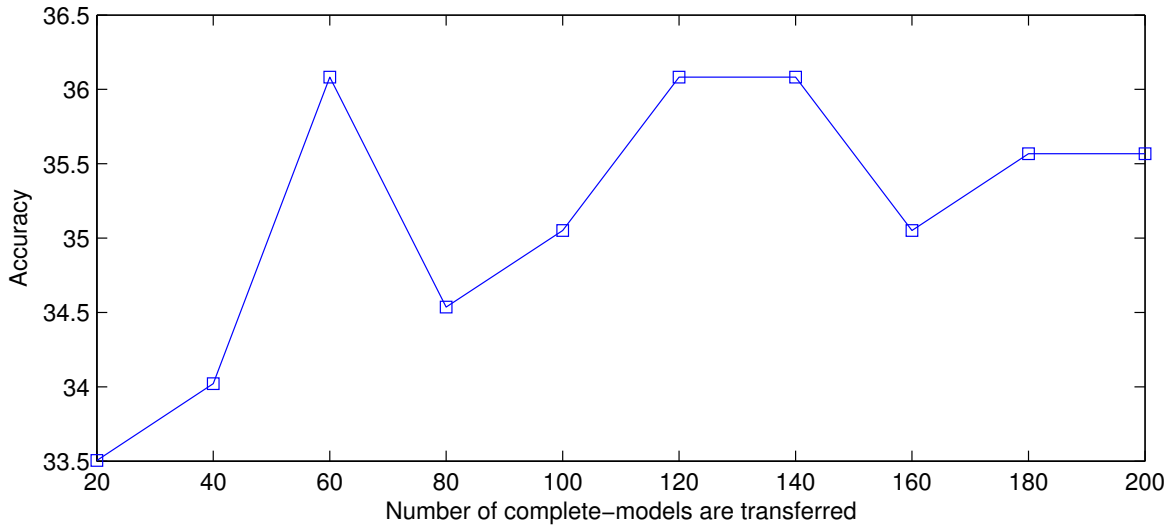
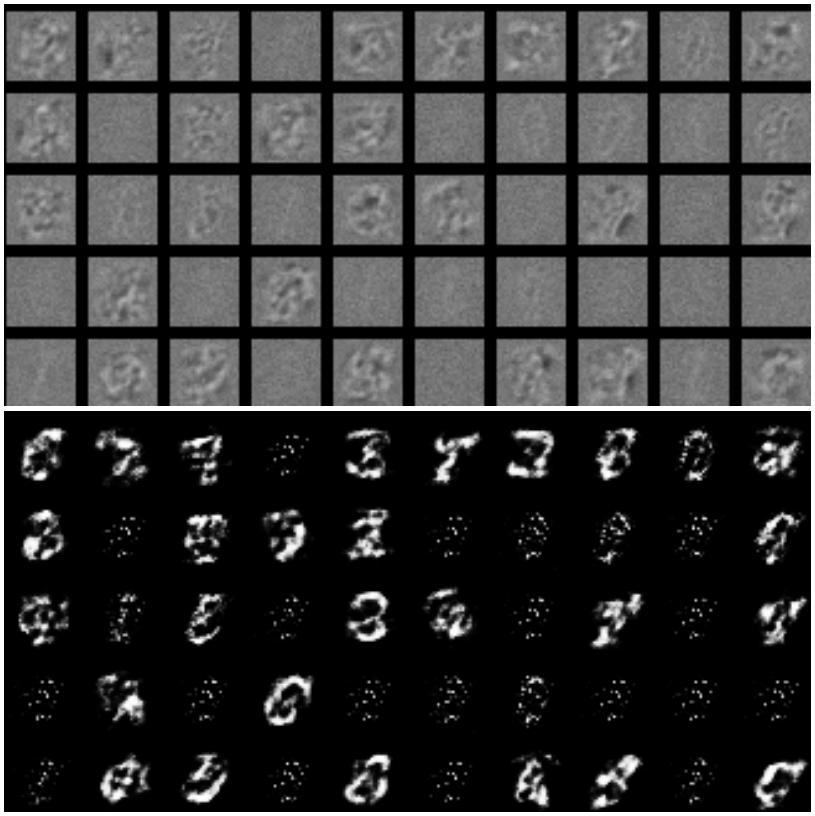


Figure: Performances of the model with different size of transferred knowledge

Outline

Pattern Groups-1



Application: Image reconstruction

- Given a noisy image, get the activated rules
- From the rules, find other rules which are related to them
- Reconstruct the images using the activated rules and the found rules

Pattern Groups-2

(..)

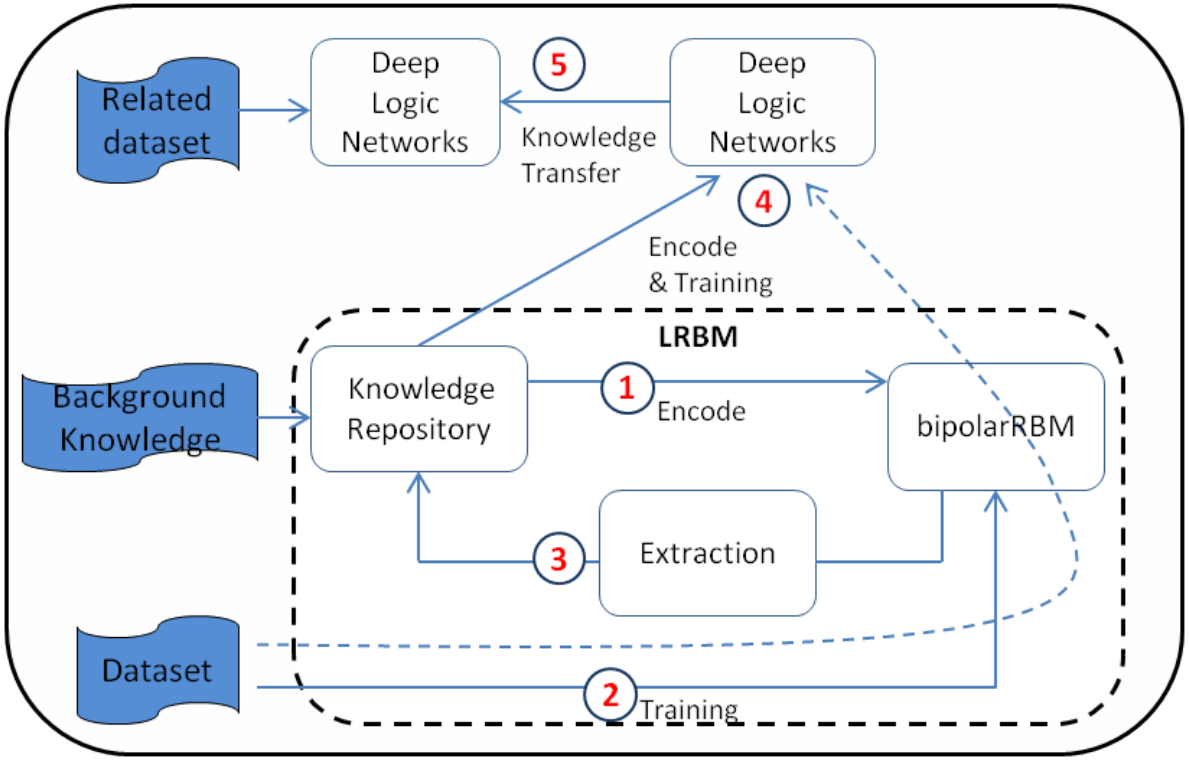
Demo

Work-in-progress

- Complete the extraction theory in DBN
- Experiment of Guiding Contrastive Divergence
- Experiment of Patten Discovery
- Experiment of Transfer Learning

Future Works

- Deep Logic Networks



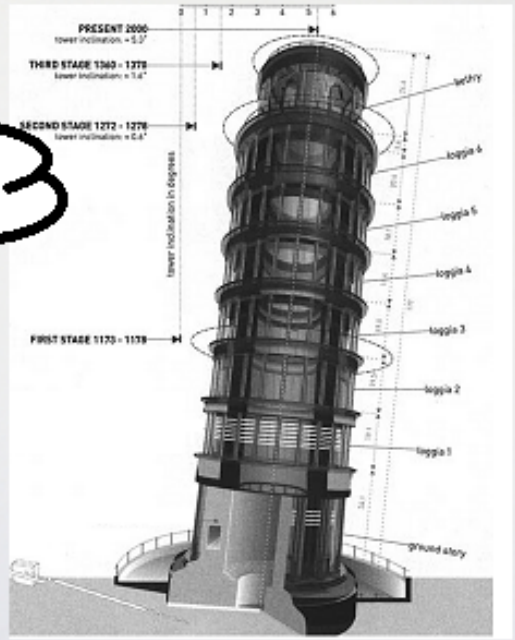
- Application in Transfer Learning for Abnormal Activity Detection

Conclusion

- Knolwedge from DBN can be obtained by extracting knowledge from RBMs in the stack
- The knowledge from higher is combination of knowledge from lower layer (deep logic??)
  $h_0^{(2)} \leftrightarrow (h_0^{(1)} \leftrightarrow v_0 \wedge v_1 \wedge \neg v_2) \wedge (h_1^{(1)} \leftrightarrow \neg v_0 \wedge \neg v_1 \wedge v_2)$
- The theory can be applied to different applications
  - Learning knowledge from a domain
  - Evaluate the system using the confidence-value
  - Develop new symbolic-connectionist system
  - Transfer knowledge to learn other domains

Thank you & Questions?

Deep Architecture?



Blank slides

This slide is kept empty

Criticism

reviewer's comment-*"The advantage of this procedure for visualization (over sampling) is not clear, nor is it clear that the extracted 'rules' are useful for doing anything that the original DBN can't deal with already"*
Answer: From the view of inference in non-relational domains, using extracted rules may not be "original" and better than sampling. However, the extracted rules can be useful for relational learning, knowledge transfer, and inference in relational domains.

- Guiding RBM to learn
- Connectionist-symbolic system: Deep Belief Logic Networks
- Relational Knowledge Transfer Learning
- Pattern Discovery

Relational RBM-2

- The circle of operations (1,2,3) is as in other connectionist-symbolic systems[Towell and Shavlik, 1994, Avila Garcez and Zaverucha, 1999, Richardson and Domingos, 2006]
- Convert the rules to the suitable form to encode into RBM
- The encode process will be different between generative RBM (for feature extraction) and discriminative RBM (for classification)
- RBM can be unipolar or bipolar (the latter is intuitively close to the extraction theory)
- Learning will be guided by the prior knowledge
- Extracted rules will be selectively added to the knowledge repository (as in incremental guided learning)

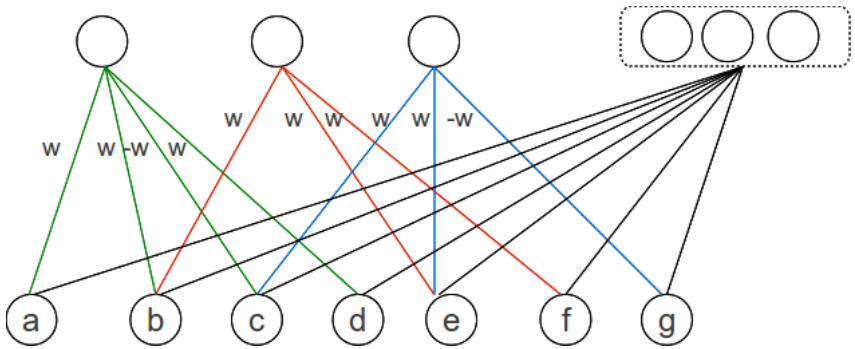
Relational RBM-3

$$A \wedge B \wedge \neg C \wedge D$$

$$B \wedge E \wedge F$$

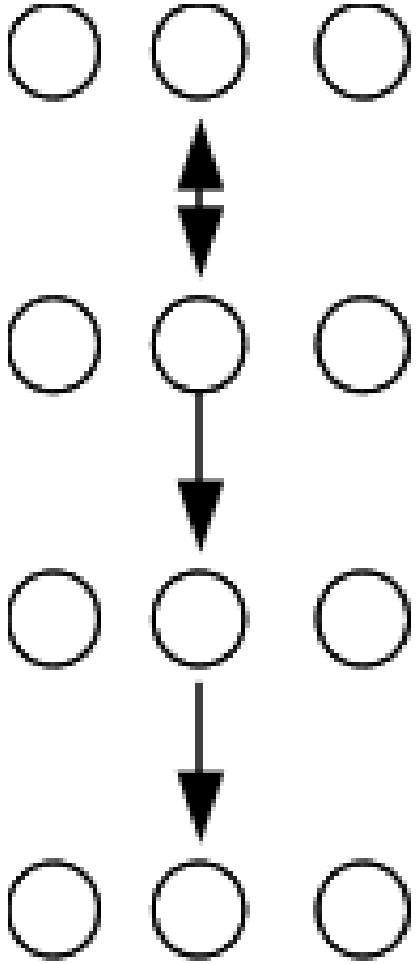
$$C \wedge E \wedge G$$

The rules are encoded into the RBM as:



Relational DBN

LOGIC



References I

Avila Garcez, A. S. and Zaverucha, G. (1999).
The connectionist inductive learning and logic programming system.
Applied Intelligence, 11(1):59–77.

Bonilla, E., Chai, K., and Williams, C. (2008).
Multi-task Gaussian process prediction.
In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*. MIT Press.

References II

Dai, W., Xue, G.-R., Yang, Q., and Yu, Y. (2007a).
Co-clustering based classification for out-of-domain documents.
In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, page 210–219, New York, NY, USA. ACM.

Dai, W., Yang, Q., Xue, G.-r., and Yu, Y. (2007b).
Boosting for transfer learning.
In *International Conference in Machine Learning*.

References III

Davis, J. and Domingos, P. (2009).
Deep transfer via second-order markov logic.
In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 217–224, New York, NY, USA. ACM.

Huang, J., Smola, A., Gretton, A., Borgwardt, K., and Schölkopf, B. (2007).
Correcting sample selection bias by unlabeled data.
In *Advances in Neural Information Processing Systems 19*, pages 601–608.

References IV

Lawrence, N. D. and Platt, J. C. (2004).
Learning to learn with the informative vector machine.
In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, page 65–, New York, NY, USA. ACM.

Mihalkova, L. and Mooney, R. J. (2009).
Transfer learning from minimal target data by mapping across relational domains.
In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, page 1163–1168, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

References V

Pan, S. J. and Yang, Q. (2010).
A survey on transfer learning.
IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359.

Penning, L. d., Garcez, A. S. d., Lamb, L. C., and Meyer, J.-J. C. (2011).
A neural-symbolic cognitive agent for online learning and reasoning.
In *IJCAI*, pages 1653–1658.

Pinkas, G. (1995).
Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge.
Artificial Intelligence, 77(2):203–247.

References VI

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. (2007).
Self-taught learning: transfer learning from unlabeled data.
In *Proceedings of the 24th international conference on Machine learning*, ICML '07, page 759–766, New York, NY, USA. ACM.

Richardson, M. and Domingos, P. (2006).
Markov logic networks.
Mach. Learn., 62(1-2):107–136.

Son Tran and Garcez, A. (2012).
ICML 2012 representation learning workshop.
In *ICML 2012 Representation Learning Workshop*, Edinburgh.

References VII

Taylor, M. E. and Stone, P. (2009).
Transfer learning for reinforcement learning domains: A survey.
J. Mach. Learn. Res., 10:1633–1685.

Towell, G. G. and Shavlik, J. W. (1994).
Knowledge-based artificial neural networks.
Artificial Intelligence, 70(1-2):119–165.