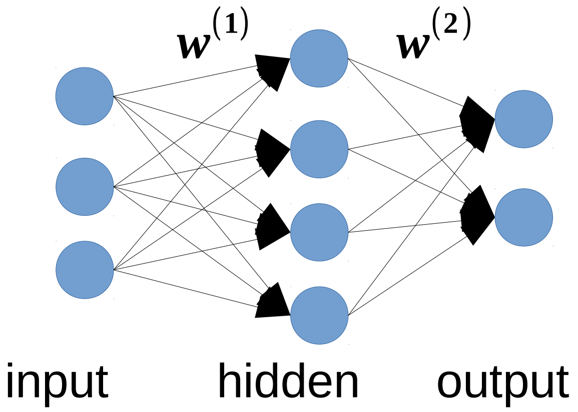
Layer-wise Training in Deep Neural Network

Do Quoc Truong
Nara Institute of Science and Technology (NAIST)
1/6/15

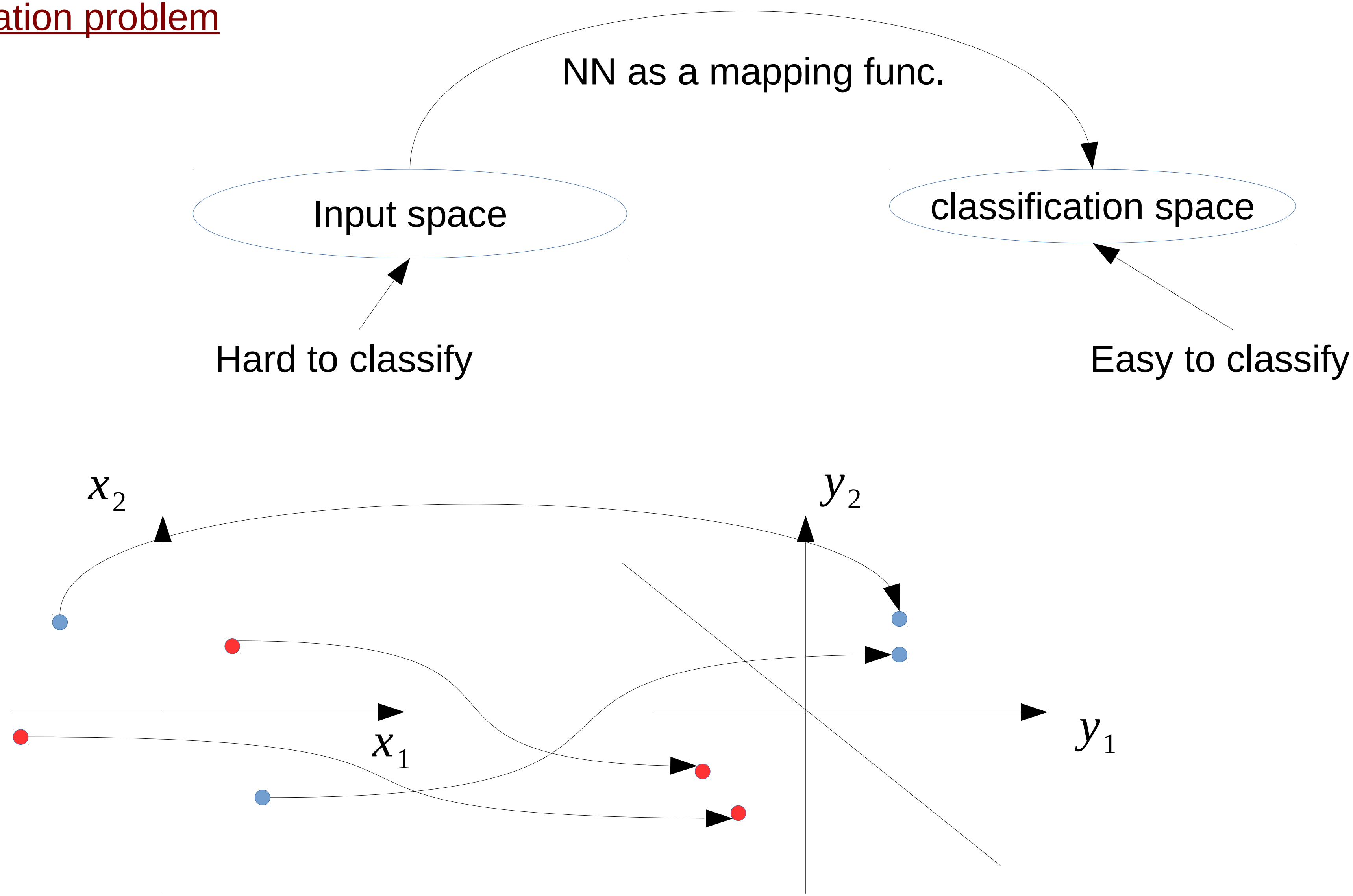
Neural Networks
(Quick overview)

Neural Networks

- A network consists of perceptions and their connection, divided into layers.
- Solve either classification or regression problem

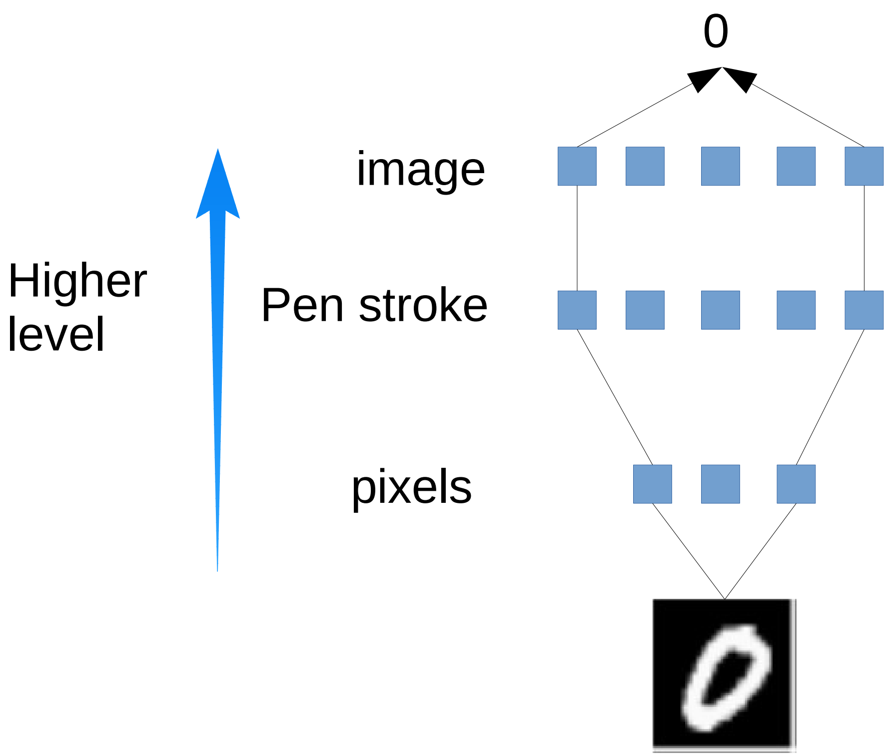


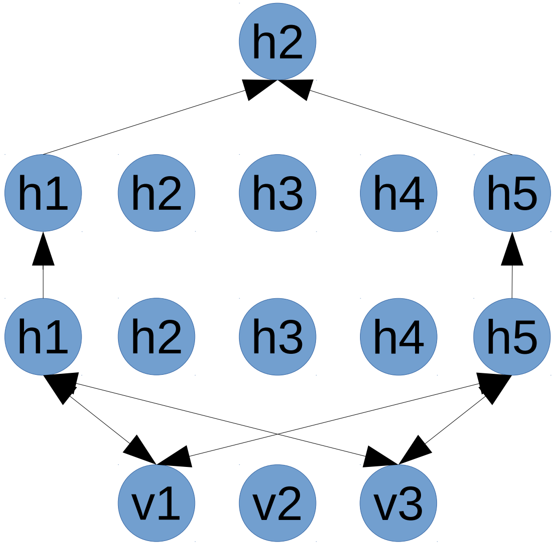
Classification problem



Deep Neural Network

- Deep neural network involve more hidden layers
  - Model non-linear function
  - Learn higher level of representation of data

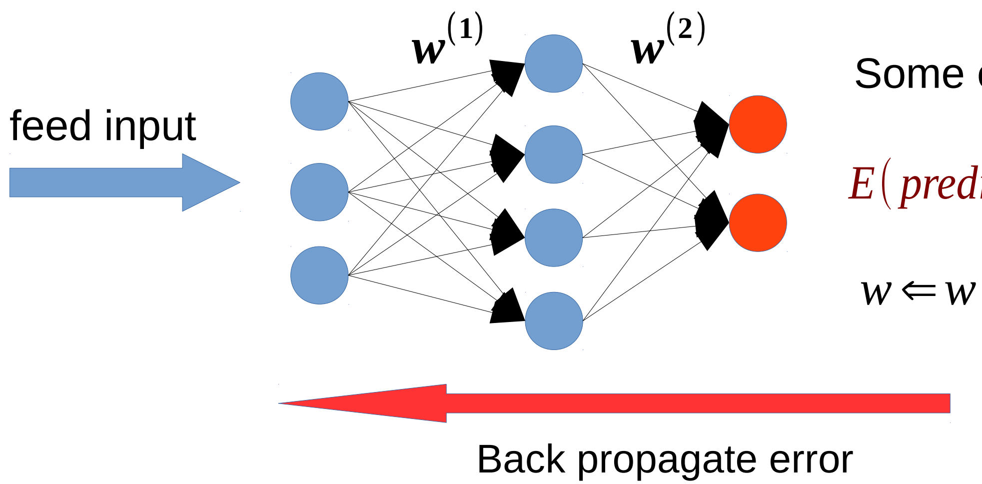




Deep Neural Networks

Neural Networks

- Training:
  - stochastic gradient descent + back propagation [Yann LeCun et al. 1989]



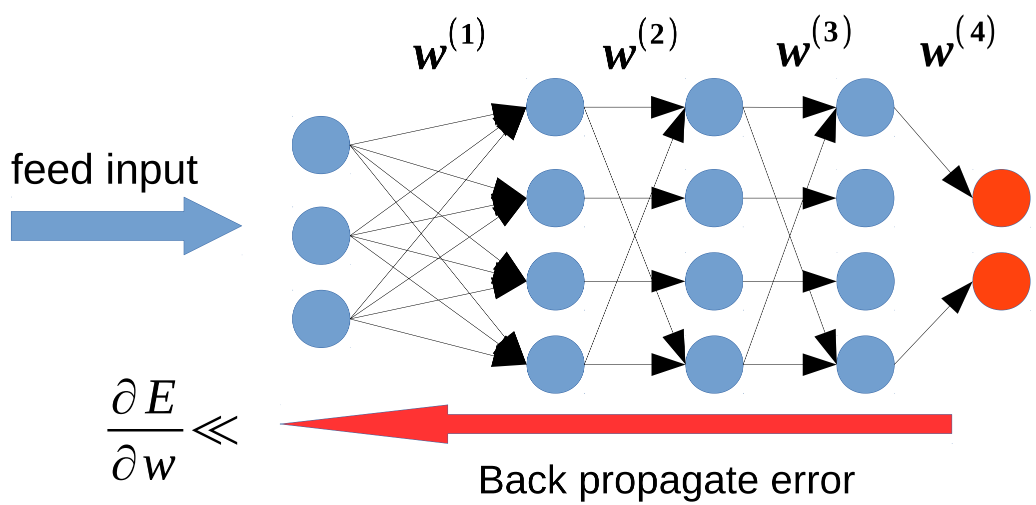
Some error:

$E(predict, label)$

$$w \Leftarrow w + \alpha \left(\frac{\partial E}{\partial w} \right)$$

Deep NN

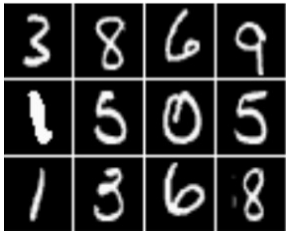
- Trap in local optimal
- Gradient vanishing



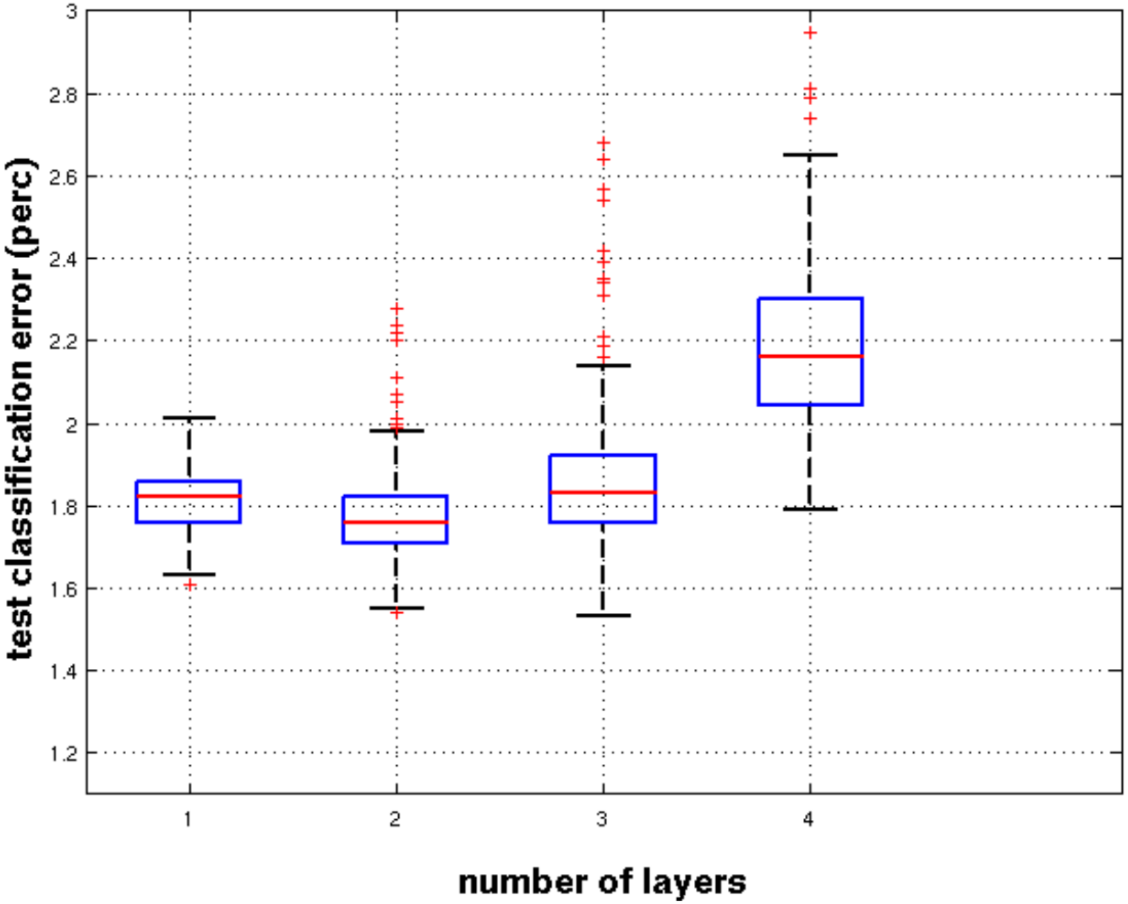
$$\frac{\partial E}{\partial w} \ll$$

Deep Neural Networks

- Difficulty in training DNN
  - MNIST digit classification task

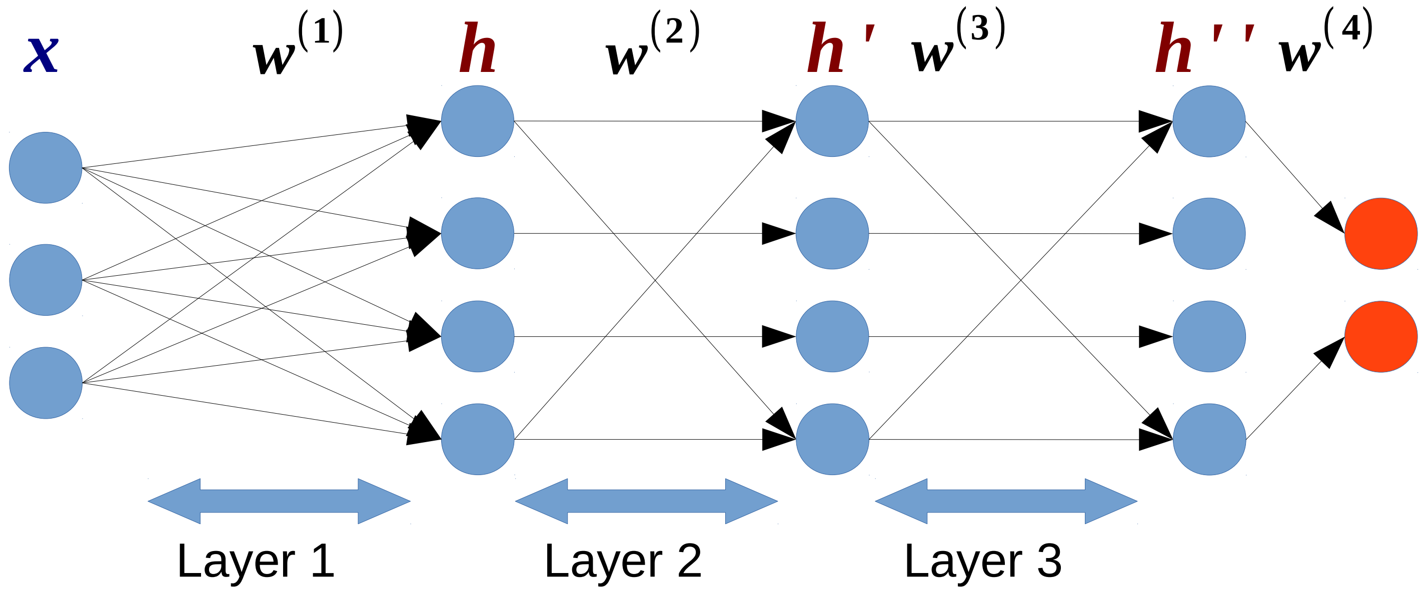


[Erhan et al., 2009]



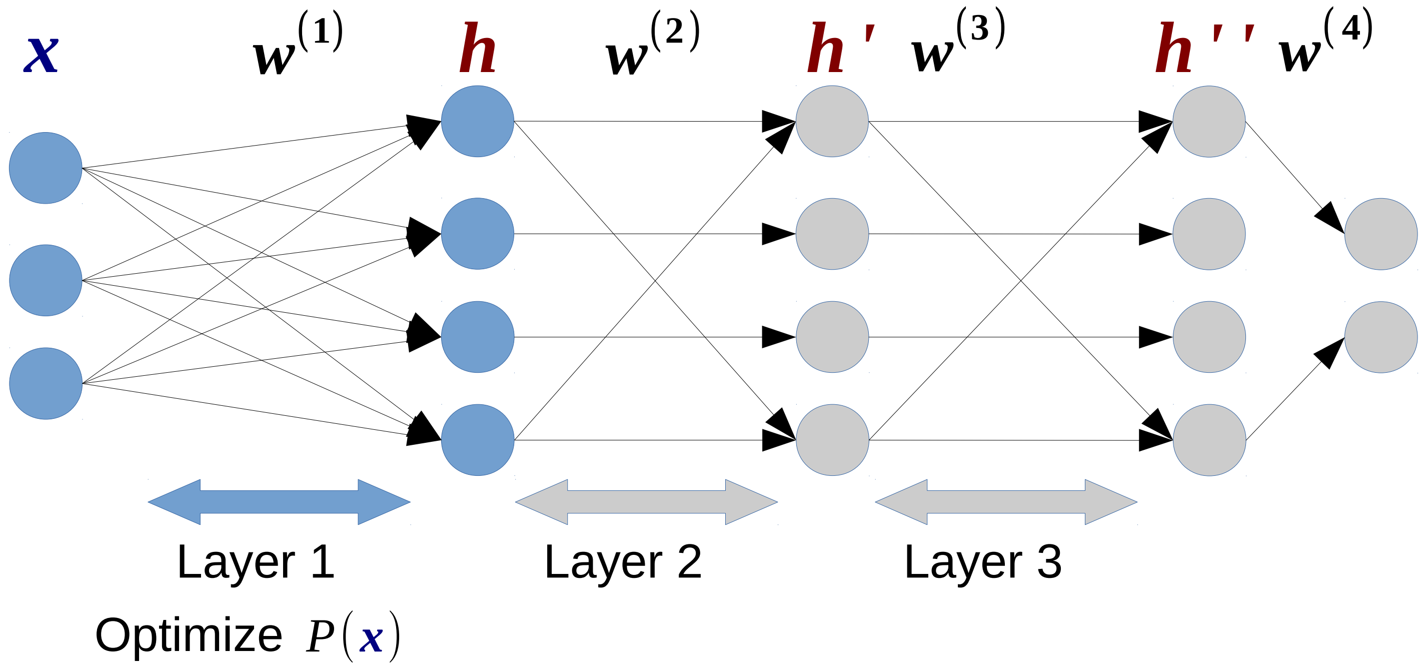
Layer-wise training

Train **one layer** at a time



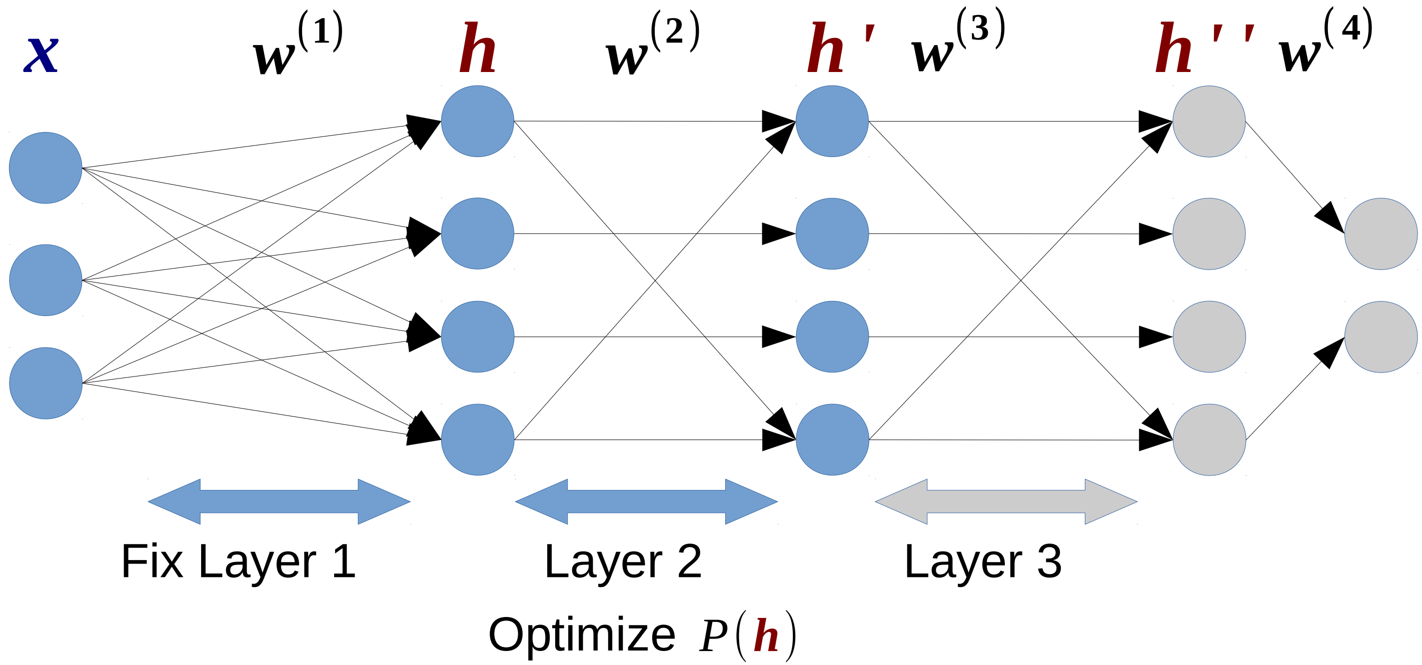
Layer-wise training

Train **one layer** at a time



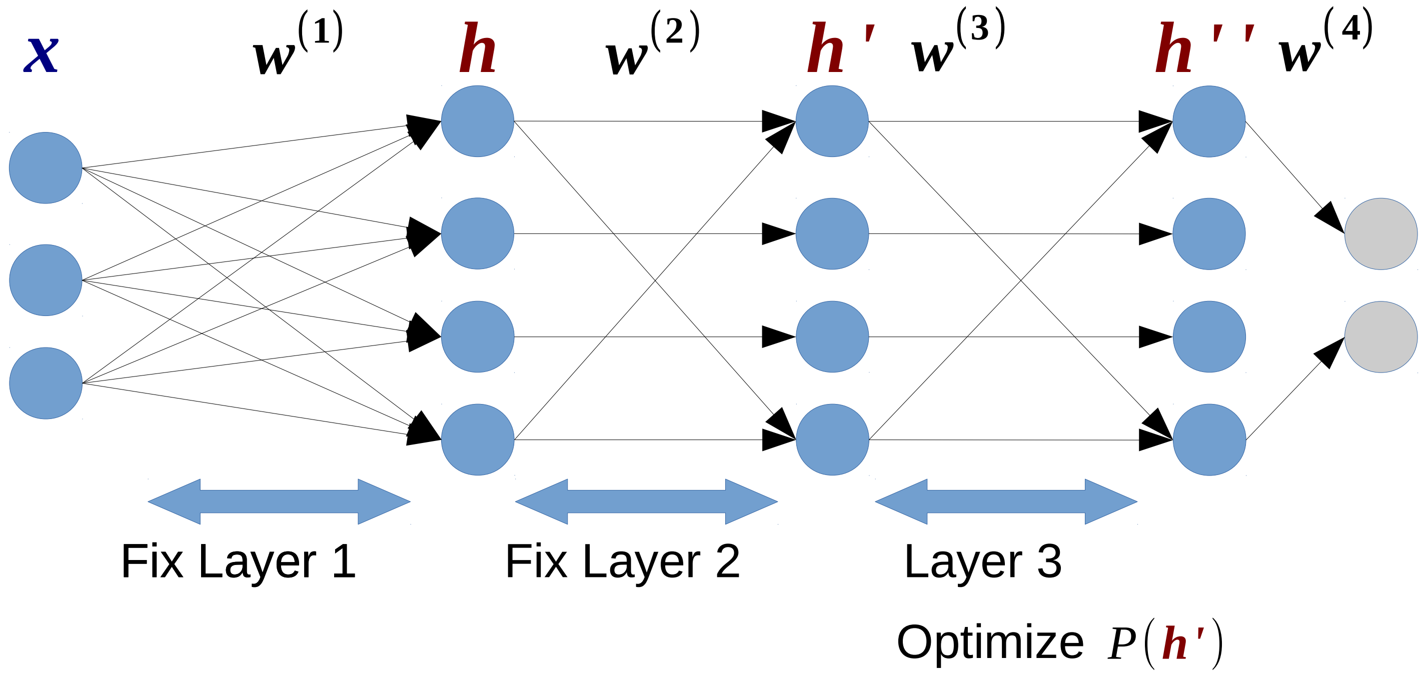
Layer-wise training

Train **one layer** at a time



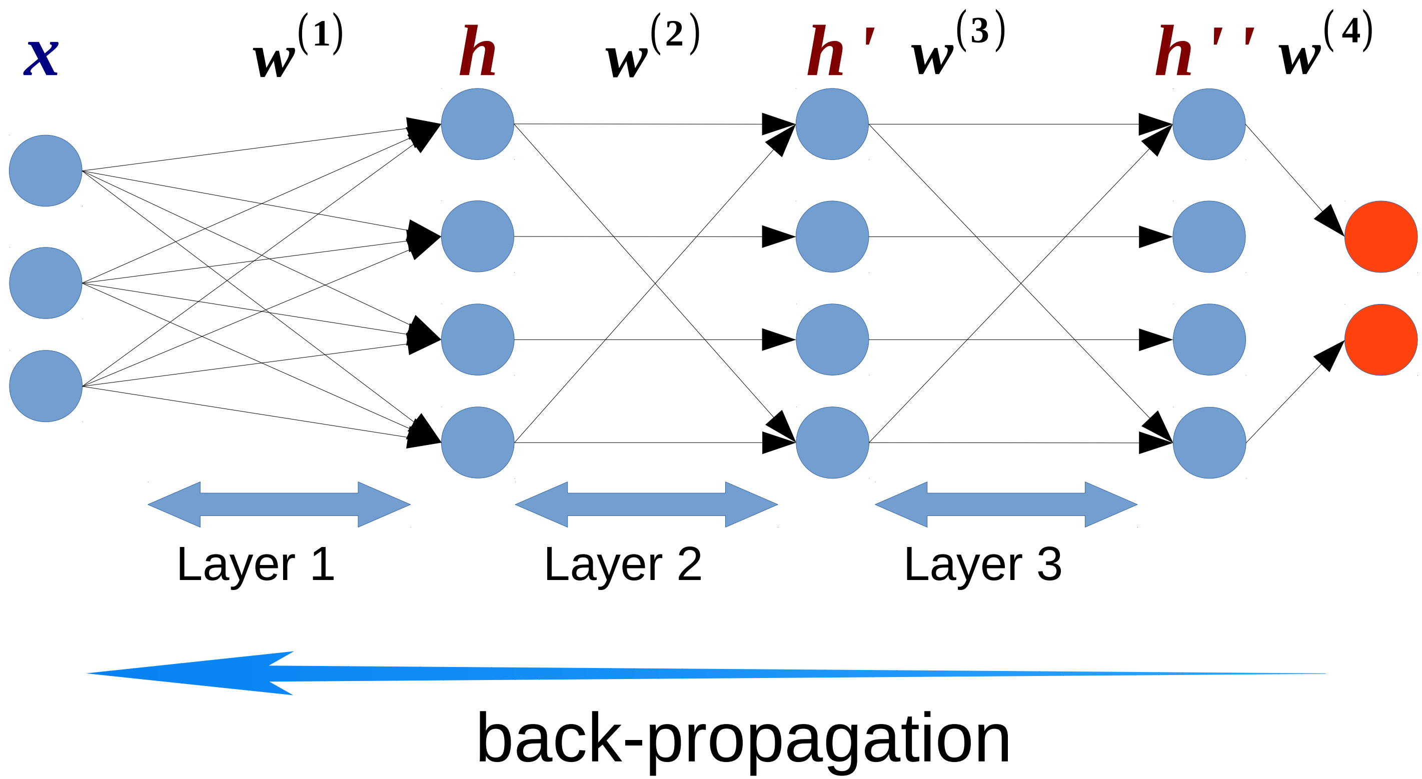
Layer-wise training

Train **one layer** at a time



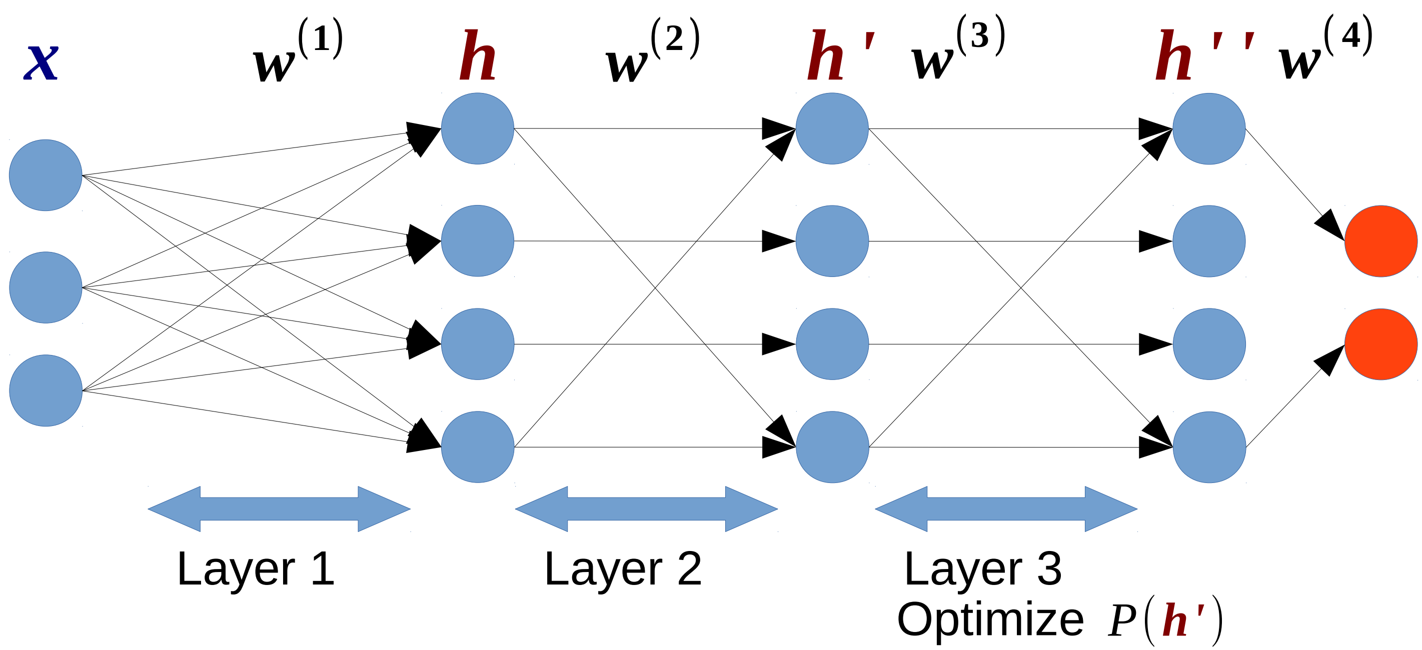
Layer-wise training

Finally, optimize P(label | input) with back-propagation



Layer-wise training

Finally, optimize P(label | input) with back-propagation



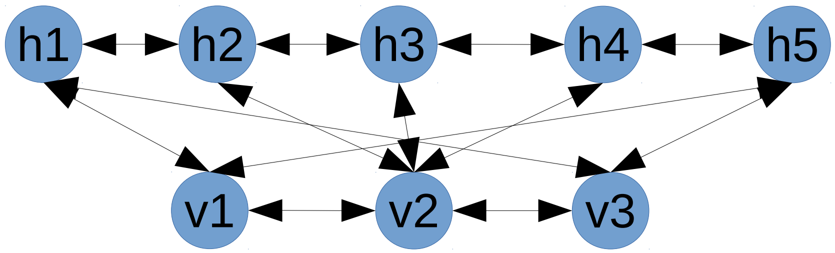
We need to understand our data first.
In other words, model P(input) before model P(label | input)

Layer-wise training

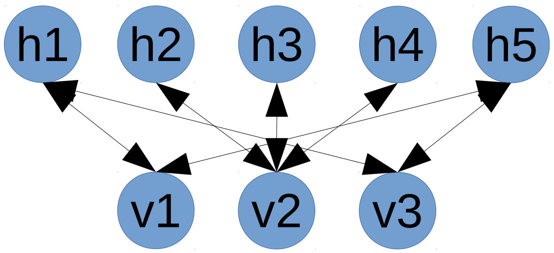
- Restricted Boltzmann Machine (RBM)
  - How to model P(input) ?
  - How it can help DNN training ?

Restricted Boltzmann Machine (RBM)

- Boltzmann Machine ?
  - a network that connect binary neurons using symmetric connection
- Restricted Boltzmann Machine ?
  - 2 layers: one hidden, one input
  - No connection between hidden nodes (Restricted)



Boltzmann Machine

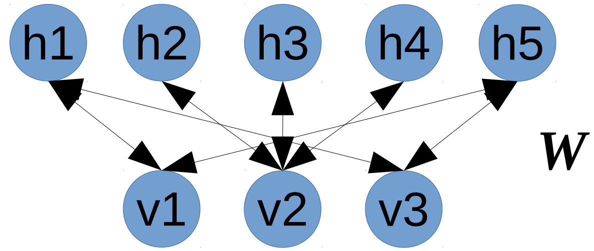


Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM)

Weights → **Energy** → **Probabilities**

W $\quad E(x,h) \quad P(x,h)$



Restricted Boltzmann Machine

Each possible join configuration has an energy:

$$-E(x,h)=x^T W h+b^T x+d^T h=\sum_{i,j} x_i W_{ij} h_j+\sum_i b_i x_i+\sum_j d_j h_j$$

Probability of a join configuration:

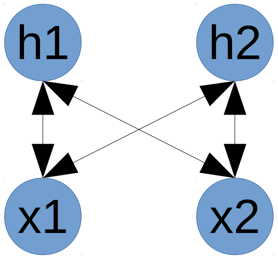
$$P(x,h)=\frac{1}{Z}e^{-E(x,h)} \quad \Rightarrow \quad P(x)=\sum_h P(x,h)$$

Where:

- Z is partition function (normalizer) $\quad Z=-\sum_{x,h} e^{-E(x,h)}$
- b, and d are bias terms

Restricted Boltzmann Machine (RBM)

Examples



RBM with 2 visible, 2 hidden units

Observe: $x_1 = 1, x_2 = 1$

$$P(v_1=1, v_2=1, h_1=1, h_2=1)$$
$$P(v_1=1, v_2=1, h_1=1, h_2=0)$$
$$P(v_1=1, v_2=1, h_1=0, h_2=1)$$
$$P(v_1=1, v_2=1, h_1=0, h_2=0)$$

never observe: $x_1 = 0, x_2 = 0$

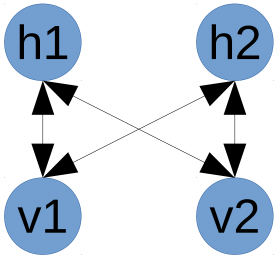
$$P(v_1=0, v_2=0, h_1=1, h_2=0)$$
$$P(v_1=0, v_2=0, h_1=0, h_2=1)$$

...

Restricted Boltzmann Machine

- Inference

  - Hidden nodes are conditional independent given observation x



$$\Rightarrow P(\mathbf{h}|\mathbf{x}_m)=\frac{P(\mathbf{x}_m,\mathbf{h})}{p(\mathbf{x}_m)}=\frac{e^{-E(\mathbf{x}_m,\mathbf{h})}/Z}{\sum_{\mathbf{h}'} e^{-E(\mathbf{x}_m,\mathbf{h}')}/Z}=\frac{e^{(\mathbf{x}_m^T W\mathbf{h}+\mathbf{x}_m\mathbf{b}^T+\mathbf{h}\mathbf{d}^T)}}{\sum_{\mathbf{h}'} e^{(\mathbf{x}_m^T W\mathbf{h}'+\mathbf{x}_m\mathbf{b}^T+\mathbf{h}'\mathbf{d}^T)}}$$

$$=\frac{e^{(\mathbf{x}_m^T W\mathbf{h}+\mathbf{h}\mathbf{d}^T)}}{\sum_{\mathbf{h}'} e^{(\mathbf{x}_m^T W\mathbf{h}'+\mathbf{h}'\mathbf{d}^T)}}=\frac{e^{(\sum_j \mathbf{x}_m^T W_j h_j+h_j d_j)}}{\sum_{\mathbf{h}'} e^{(\sum_j \mathbf{x}_m^T W_j h'_j+h'_j d_j)}}$$

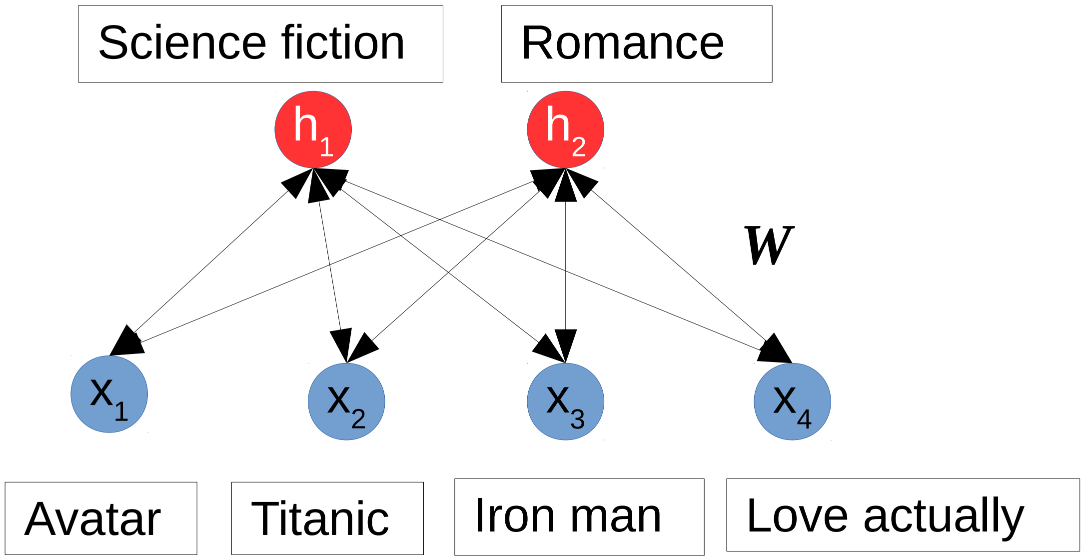
$$=\frac{\prod_j e^{\mathbf{x}_m^T W_j h_j+h_j d_j}}{\prod_j \sum_{h_j'} e^{\mathbf{x}_m^T W_j h'_j+h'_j d_j}}=\prod_j \frac{e^{\mathbf{x}_m^T W_j h_j+h_j d_j}}{\sum_{h_j'} e^{\mathbf{x}_m^T W_j h'_j+h'_j d_j}}=\prod_j P(h_j|\mathbf{x}_m)$$

$$\Rightarrow P(\mathbf{x}_m|\mathbf{h})=\prod_j P(x_m^{(j)}|\mathbf{h})$$

$$\Rightarrow P(h_j=1|\mathbf{x}_m)=\frac{e^{x_m^T W_j+d_j}}{1+e^{x_m^T W_j+d_j}}=\frac{1}{1+e^{-(x_m^T W_j+d_j)}}=sigmoid(\mathbf{x}_m^T \mathbf{W}_j+d_j)$$

Restricted Boltzmann Machine

- Examples



➡ Observations 1 0 1 0

$P(h_1=1|\boldsymbol{x_m})=sigmoid(\boldsymbol{x_m^T W_j}+d_j)=0.8$ ➡ This person likes SF with p = 0.8

$P(h_1=0,\boldsymbol{x_m})=1-0.8=0.2$

In practice: $h_1=1 \, if \; P(h_1|\boldsymbol{x_m})>U(0,1)$

➡ Generate data

Given a person likes SF: h_1 = 1, h_2 = 0

Generate which movies they should watch with $P(\boldsymbol{x}|h_1=1,h_2=0)$

Training process:

Given training sample $\boldsymbol{x_m}$, we want to maximize the likelihood

$$P(X=x_m)=\sum_h P(x_m,y)$$

$$\hat{W}=\underset{W}{argmax}\log(P(X=x_m))=\underset{W}{argmax}\log\left(\sum_h P(x_m,y)\right)$$

Take derivative of Log-Likelihood w.r.t w_{ij}

$$\frac{\partial\log(P(X=x_m))}{\partial w_{ij}}=\sum_{\boldsymbol{x},\boldsymbol{h}} P(\boldsymbol{x},\boldsymbol{h})\frac{\partial(E(\boldsymbol{x},\boldsymbol{h}))}{\partial w_{ij}}-\sum_{\boldsymbol{h}} P(\boldsymbol{h}|\boldsymbol{x_m})\frac{\partial E(\boldsymbol{x_m},\boldsymbol{h})}{\partial w_{ij}}$$

$$W\Leftarrow W-\alpha\left(\frac{\partial\log(P(X=x_m))}{\partial w_{ij}}\right)=W+\alpha\left(\underbrace{\sum_{\boldsymbol{h}} P(\boldsymbol{h}|\boldsymbol{x_m})\frac{\partial E(\boldsymbol{x_m},\boldsymbol{h})}{\partial w_{ij}}}_{\text{Positive terms}}-\underbrace{\sum_{\boldsymbol{x},\boldsymbol{h}} P(\boldsymbol{x},\boldsymbol{h})\frac{\partial(E(\boldsymbol{x},\boldsymbol{h}))}{\partial w_{ij}}}_{\text{Negative terms}}\right)$$

Restricted Boltzmann Machine

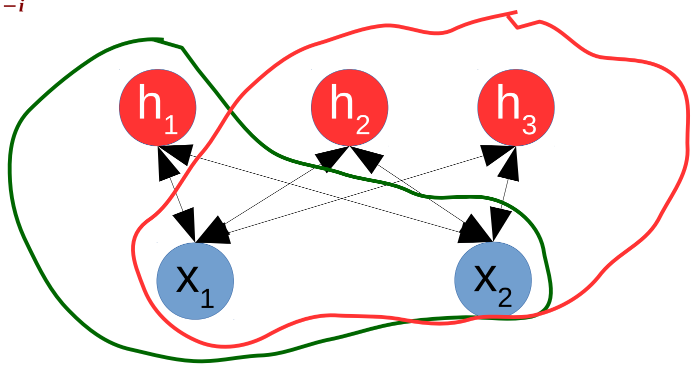
Training process:

The positive **term** is easy to compute:

$$\sum_{\boldsymbol{h}} P(\boldsymbol{h}|\boldsymbol{x_m}) \frac{\partial E(\boldsymbol{x_m},\boldsymbol{h})}{\partial w_{ij}} = \sum_{\boldsymbol{h}} P(\boldsymbol{h}|\boldsymbol{x_m}) h_i x_j = \sum_{h_i} \sum_{\boldsymbol{h_{-i}}} P(h_i|\boldsymbol{x_m}) P(\boldsymbol{h_{-i}}|\boldsymbol{x_m}) h_i x_j$$

$$= \sum_{h_i} P(h_i|\boldsymbol{x_m}) h_i x_j \underbrace{\sum_{\boldsymbol{h_{-i}}} P(\boldsymbol{h_{-i}}|\boldsymbol{x_m})}_{=1}$$

$$= P(h_i=1,\boldsymbol{x_m}) x_j = \sigma(\sum_{j=1} w_{ji} x_j + d_i) x_j$$



The **negative term** is hard when big hidden units

$$\sum_{\boldsymbol{x},\boldsymbol{h}} P(\boldsymbol{x},\boldsymbol{h}) \frac{\partial (E(\boldsymbol{x},\boldsymbol{h}))}{\partial w_{ij}}$$

2 visible, 3 hidden ➡ $2^2 . 2^3 = 32$

# of configurations **x**, **h** is exponential $2^{num\ of\ units}$

➡ Can not compute $\frac{\partial \log(P(X=x_m))}{\partial w_{ij}}$ directly

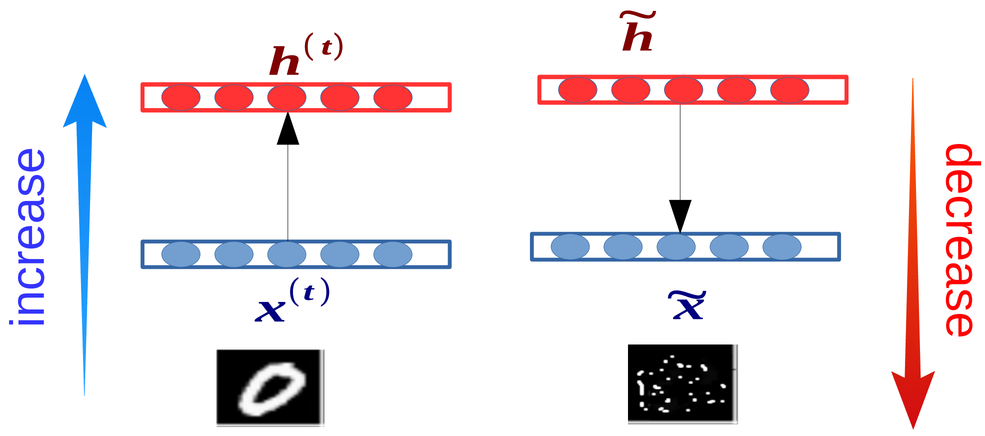
CONSTRASTIVE DIVERGENCE [HINTON et al. 2002]

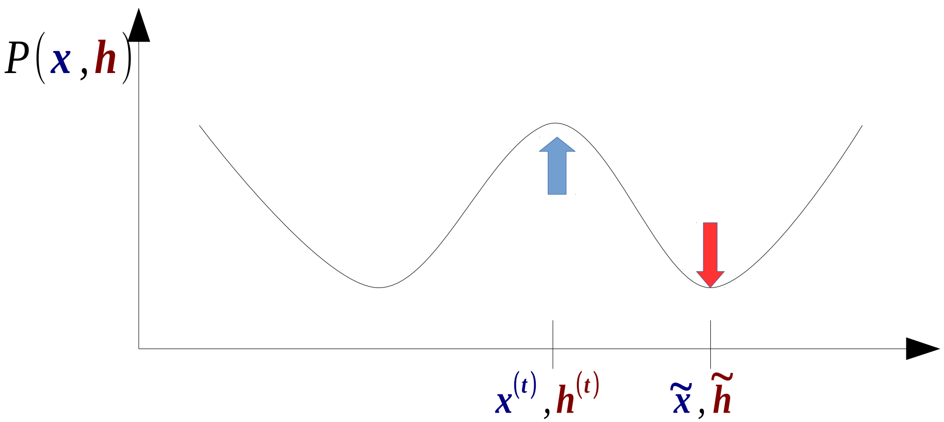
CONSTRASTIVE DIVERGENCE [HINTON et al. 2002]

ML
$$\underbrace{\sum_{\boldsymbol{h}} P(\boldsymbol{h}|\boldsymbol{x}^{(t)})\frac{\partial E(\boldsymbol{x}^{(t)},\boldsymbol{h})}{\partial w_{ij}}}_{\wr\wr} - \underbrace{\sum_{\boldsymbol{x},\boldsymbol{h}} P(\boldsymbol{x},\boldsymbol{h})\frac{\partial(E(\boldsymbol{x},\boldsymbol{h}))}{\partial w_{ij}}}_{\wr\wr}$$

CD
$$\frac{\partial E(\boldsymbol{x}^{(t)},\boldsymbol{h}^{(t)})}{\partial(w_{ij})} - \frac{\partial(E(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{h}}))}{\partial w_{ij}} = \boldsymbol{x}^{(t)T}\boldsymbol{h}^{(t)} - \tilde{\boldsymbol{x}}^T\tilde{\boldsymbol{h}}$$

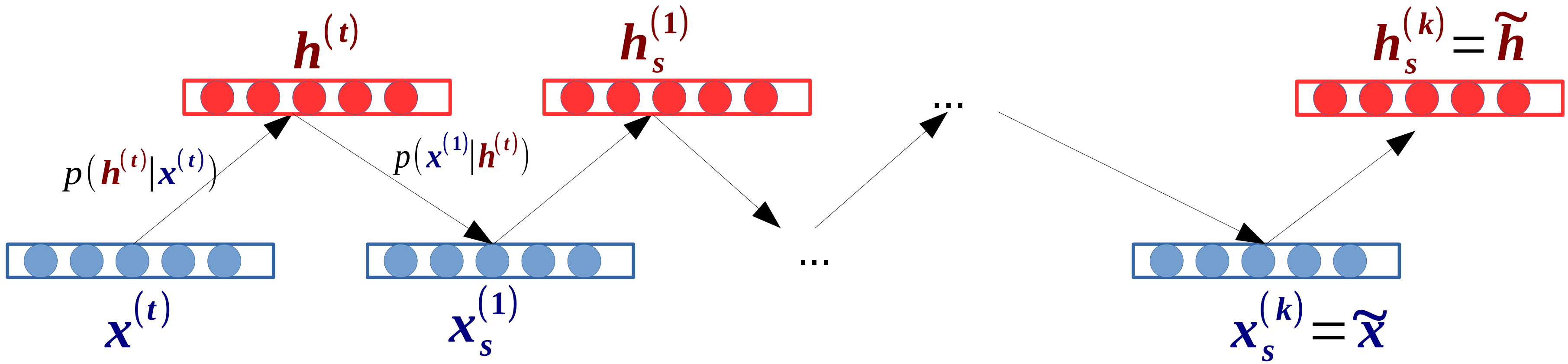
$$\Rightarrow \quad W \Leftarrow W + \alpha(\boldsymbol{x}^{(t)T}\boldsymbol{h}^{(t)} - \tilde{\boldsymbol{x}}^T\tilde{\boldsymbol{h}})$$





CONSTRASTIVE DIVERGENCE [HINTON et al. 2002]

Gibbs sampling



For each training sample $x^{(t)}$

1. Generate $\tilde{x}$ using k-step Gibbs sampling from x^(t)
2. Update parameters:

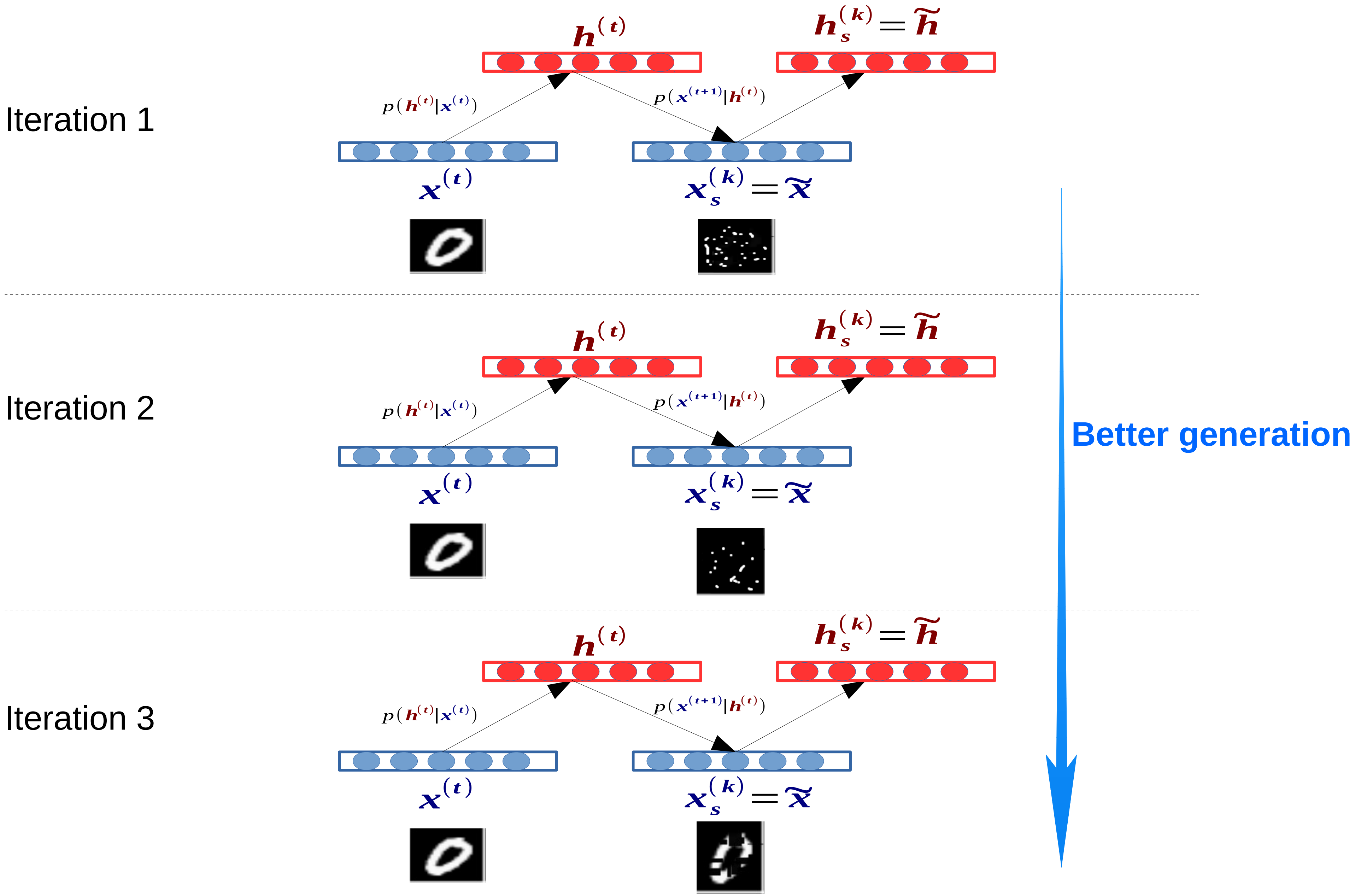
$$W \Leftarrow W + \alpha \left(x^{(t)T} h^{(t)} - \tilde{x}^T \tilde{h} \right)$$

$$b \Leftarrow b + \alpha \left(x^{(t)} - \tilde{x} \right)$$

$$d \Leftarrow d + \alpha \left(h^{(t)} - \tilde{h} \right)$$

3. Repeat until stopping criteria

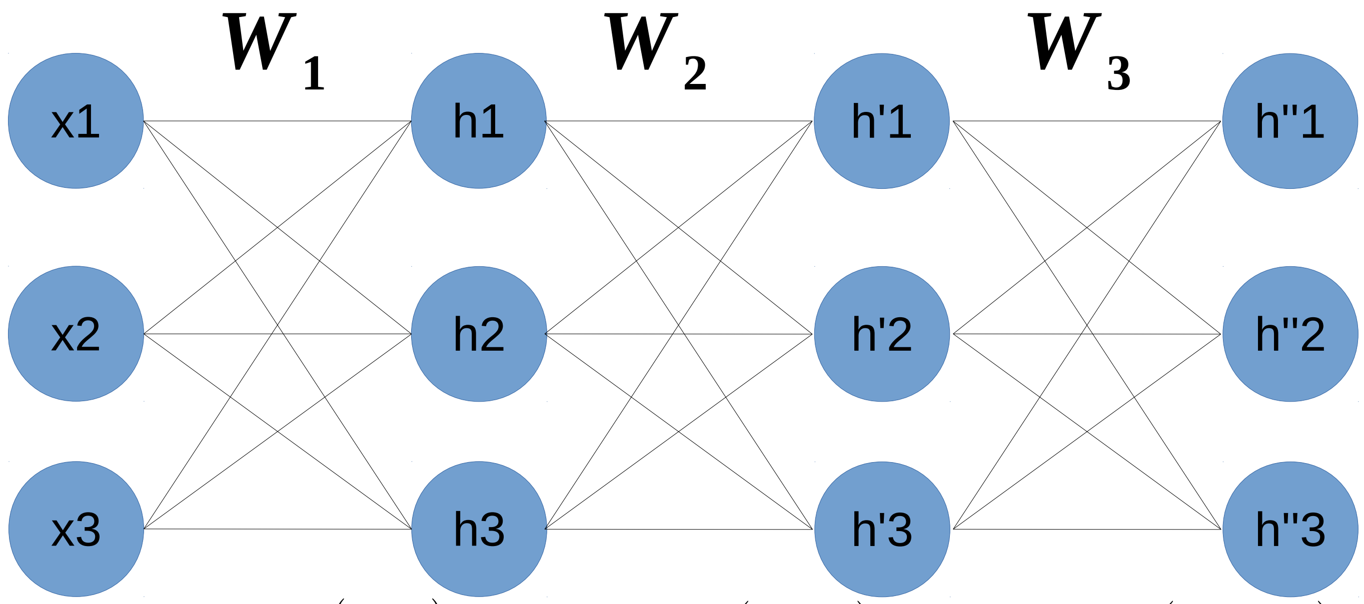
CONSTRASTIVE DIVERGENCE [HINTON et al. 2002]



CONSTRASTIVE DIVERGENCE [HINTON et al. 2002]

- CD-k: k iterations of Gibbs sampling
- The bigger k is, the better the estimate of gradient will be (Law of large number)
- In practice, k=1 is good for pre-training:
  - Optimize P(x) is not the goal
  - Later fine-tuning with BP: P(label | x)

Layer-wise Pre-training RBM

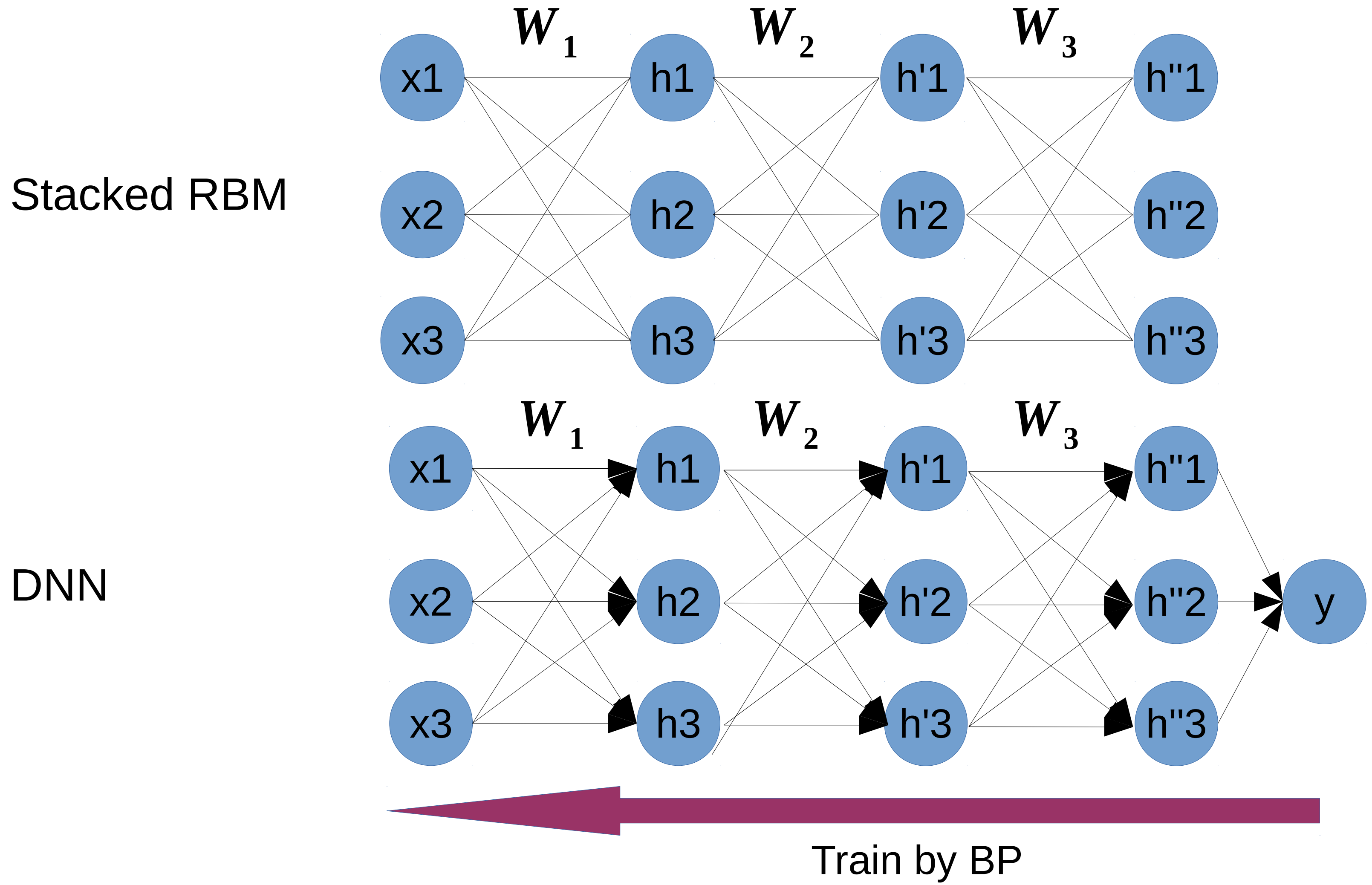


$maximize\ P(\boldsymbol{x}, \boldsymbol{h})$ $maximize\ P(\boldsymbol{h}, \boldsymbol{h}')$ $maximize\ P(\boldsymbol{h}', \boldsymbol{h}'')$

Layer-wise Pre-training RBM

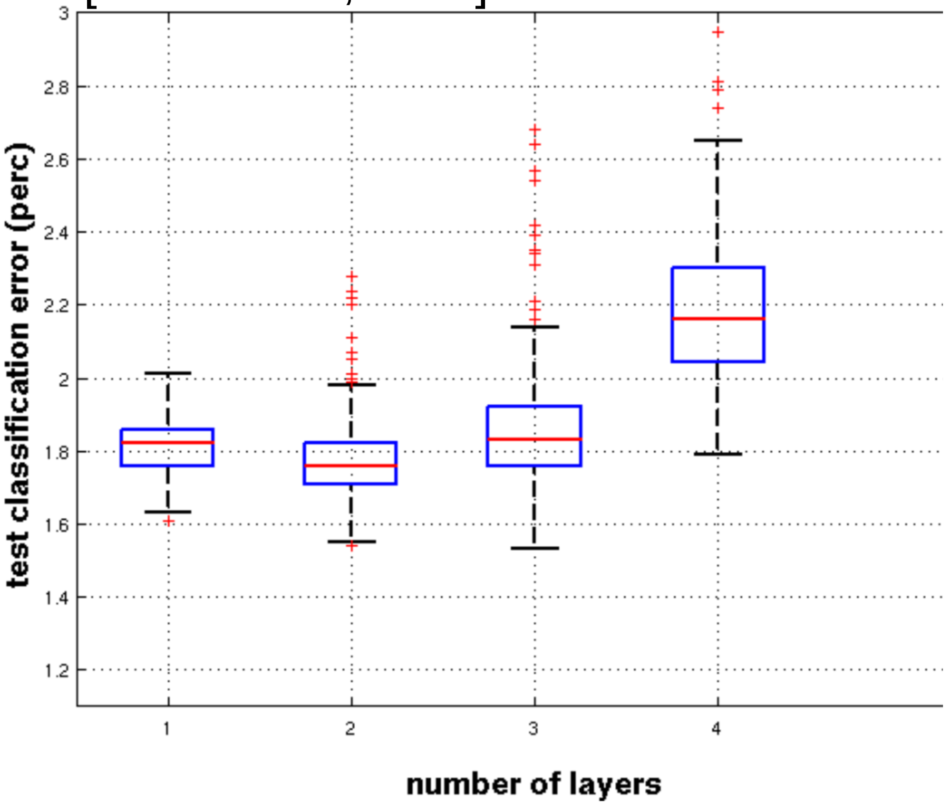
Application:

- Stacked RBM – Deep Belief Network: Generative model
- Use as initial parameter for DNN

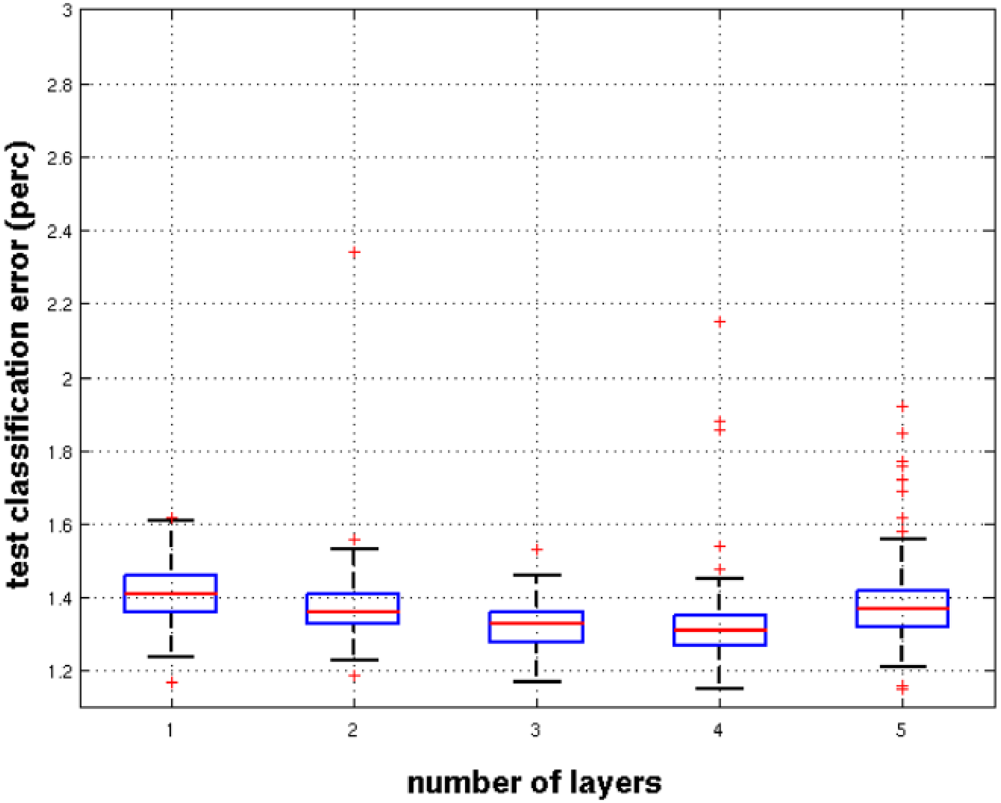


- Pre-train help DNN works better

[Erhan et al., 2009]

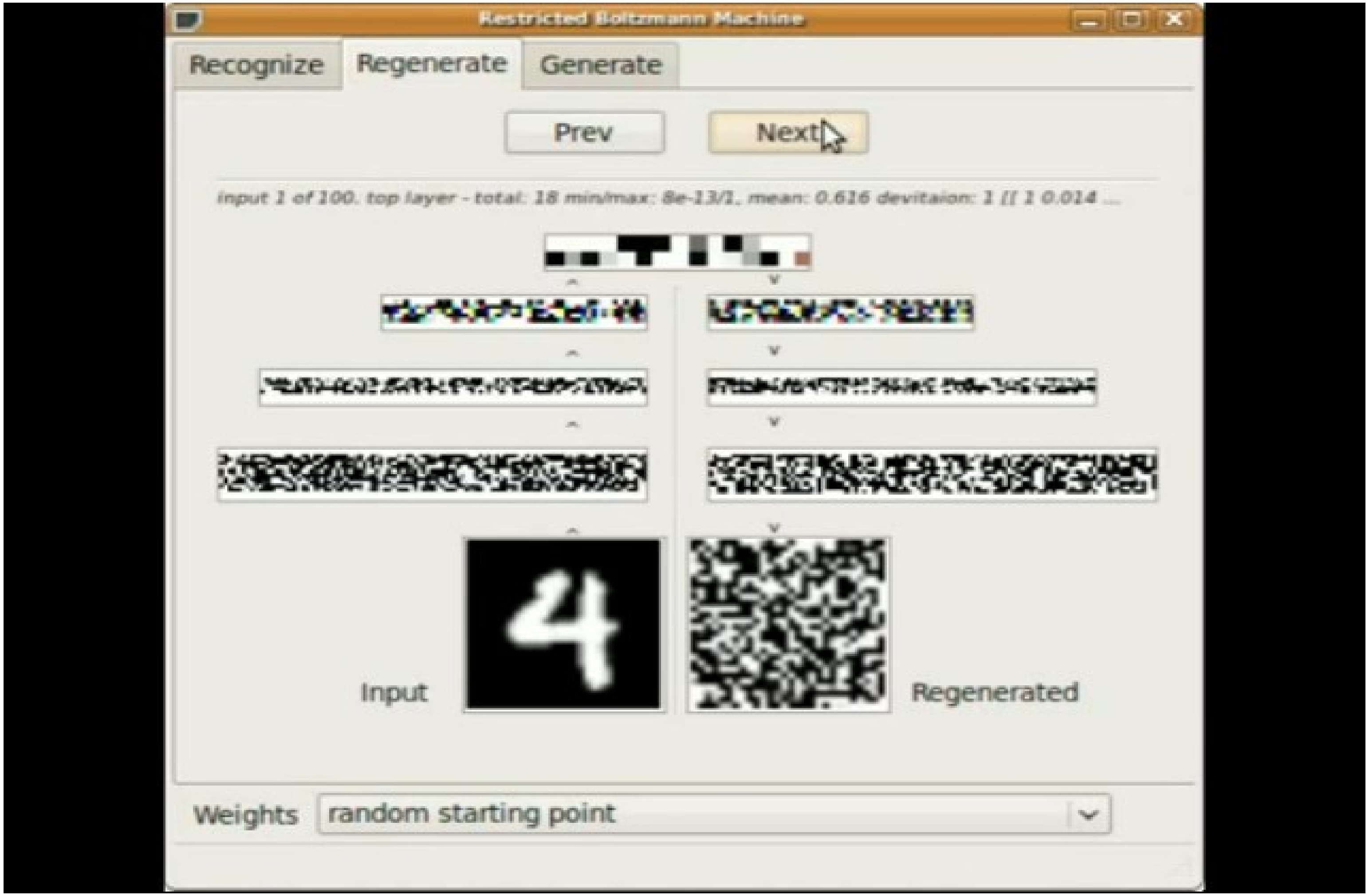


DNN w/o pre-training



DNN w/ pre-training

RBM examples



https://www.youtube.com/watch?v=0LTG64s6Xuc