TOKYO TECH
Pursuing Excellence

Back-Propagation Algorithm for Deep Neural Networks and Contradictive Diverse Learning for Restricted Boltzmann Machine

Masayuki Tanaka

Aug. 17, 2015

TOKYO INSTITUTE OF TECHNOLOGY

Outline

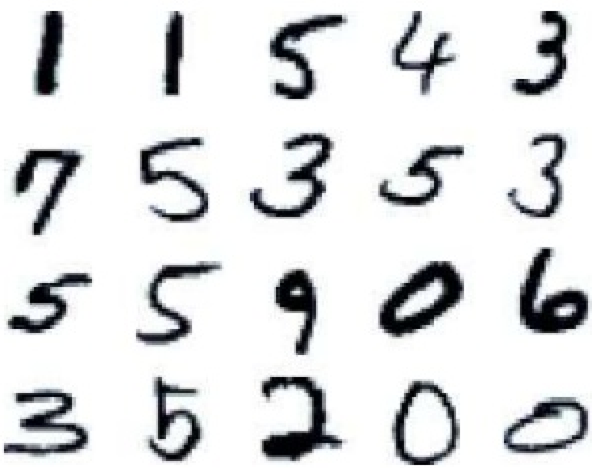
http://bit.ly/dnnicpr2014

1. Examples of Deep Learning
2. RBM to Deep NN
3. Deep Neural Network (Deep NN)
   – Back-Propagation (Supervised Learning)
4. Restricted Boltzmann Machine (RBM)
   – Mathematics, Probabilistic Model and Inference Model
   – Pre-training by Contradictive Diverse Learning (Unsupervised Learning)
5. Inference Model with Distribution

Deep learning

- Top performance in character recognition
  – MNIST (handwritten digits benchmark)

MNIST



Result	Method	Venue	Details
0.21%	Regularization of Neural Networks using DropConnect	ICML 2013	
0.23%	Multi-column Deep Neural Networks for Image Classification	CVPR 2012	
0.35%	Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition	Neural Computation 2010	Details
0.39%	Efficient Learning of Sparse Representations with an Energy-Based Model	NIPS 2006	Details
0.39%	Convolutional Kernel Networks	arXiv 2014	Details
0.39%	Deeply-Supervised Nets	arXiv 2014	
0.4%	Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis	Document Analysis and Recognition 2003	
0.45%	Maxout Networks	ICML 2013	Details
0.47%	Network in Network	ICLR 2014	Details
0.52 %	Trainable COSFIRE filters for keypoint detection and pattern recognition	PAMI 2013	Details
0.53%	What is the Best Multi-Stage Architecture for Object Recognition?	ICCV 2009	Details
0.54%	A trainable feature extractor for handwritten digit recognition	Journal Pattern Recognition	Details

Deep learning

- Top performance in image classification
  - – CIFAR (image classification benchmark)

CIFAR10

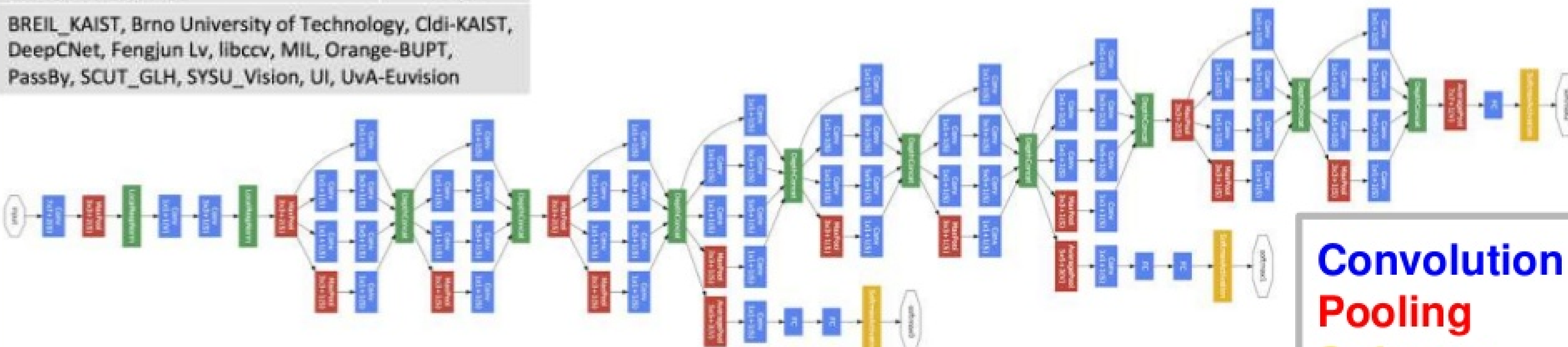
Result	Method	Venue	Details
94%	Lessons learned from manually classifying CIFAR-10	unpublished 2011	Details
91.78%	Deeply-Supervised Nets	arXiv 2014	Details
91.2%	Network In Network	ICLR 2014	Details
90.68%	Regularization of Neural Networks using DropConnect	ICML 2013	
90.65%	Maxout Networks	ICML 2013	Details
90.61%	Improving Deep Neural Networks with Probabilistic Maxout Units	ICLR 2014	Details
90.5%	Practical Bayesian Optimization of Machine Learning Algorithms	NIPS 2012	Details
89%	ImageNet Classification with Deep Convolutional Neural Networks	NIPS 2012	Details
88.79%	Multi-Column Deep Neural Networks for Image Classification	CVPR 2012	Details
84.87%	Stochastic Pooling for Regularization of Deep Convolutional Neural Networks	arXiv 2013	
84.4%	Improving neural networks by preventing co-adaptation of feature detectors	arXiv 2012	Details
83.96%	Discriminative Learning of Sum-Product Networks	NIPS 2012	
82.9%	Stable and Efficient Representation Learning with Nonnegativity Constraints	ICML 2014	Details

Deep learning

- Top performance in visual recoginition

Team Name	Error (%)
GoogLeNet	6.7
VGG	7.3
MSRA Visual computing	8.1
Andrew Howard	8.1
DeeperVision	9.5
NUS-BST	9.8
TTIC_ECP – Epitomic Vision	10.2
XYZ	11.2
BDC-I2R-UPMC	11.3
BREIL_KAIST, Brno University of Technology, Cldi-KAIST, DeepCNet, Fengjun Lv, libccv, MIL, Orange-BUPT, PassBy, SCUT_GLH, SYSU_Vision, UI, UvA-Euvision	

Image Large Scale Visual Recognition Challenge (ILSVRC)

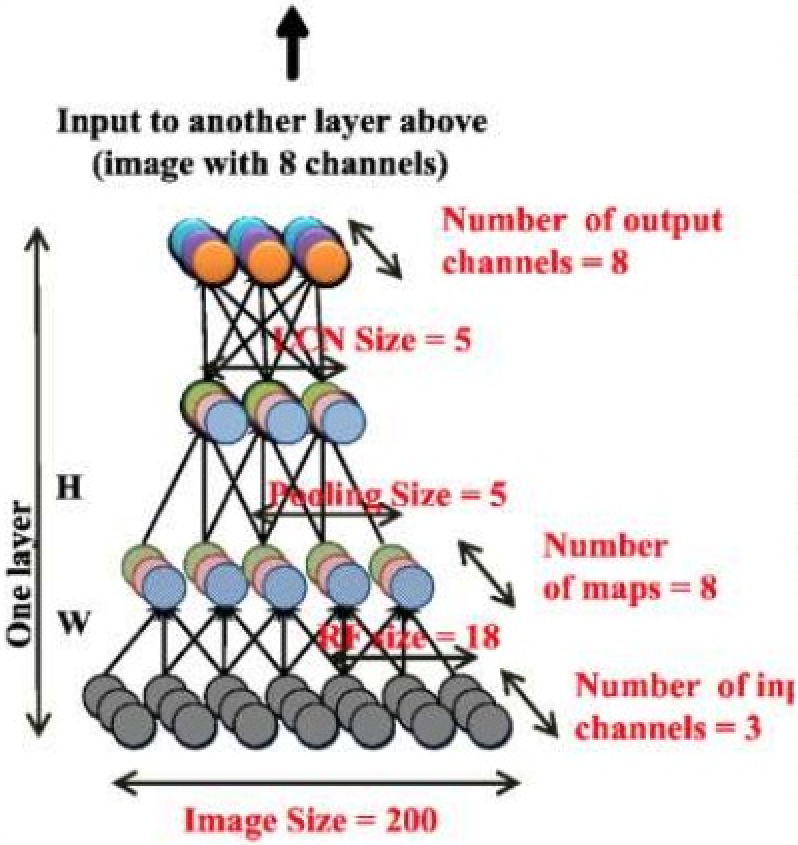


Convolution
Pooling
Softmax
Other

GoogLeNet, ILSVRC2014

Deep learning

➢ "Cat neuron"



Automatic learning with youtube videos,
neuron for human's face
neuron for cat

10,000,000:
training samples

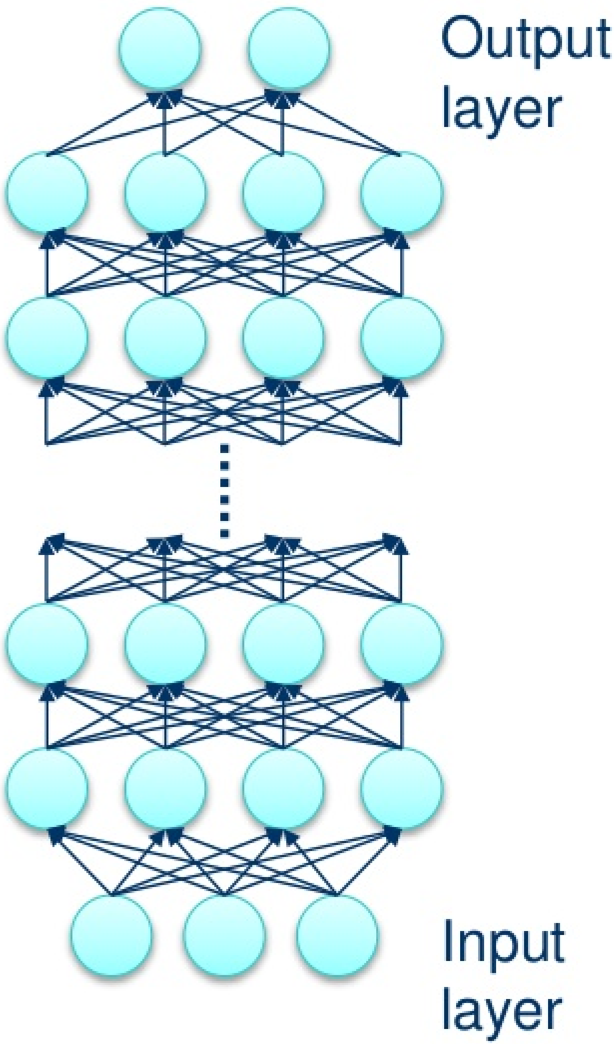
Three days learning with
1,000 computers

http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en//archive/unsupervised_icml2012.pdf

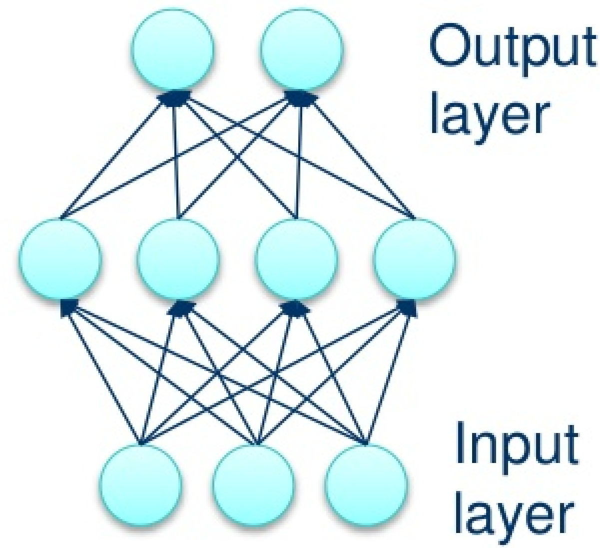
Deep??

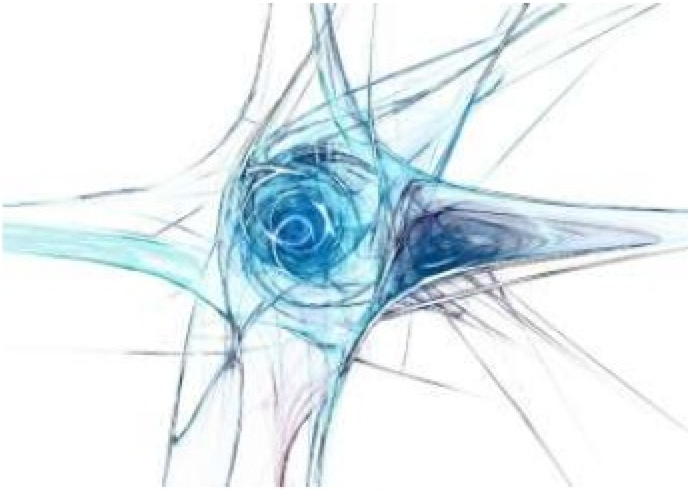
NN: Neural network

Deep NN



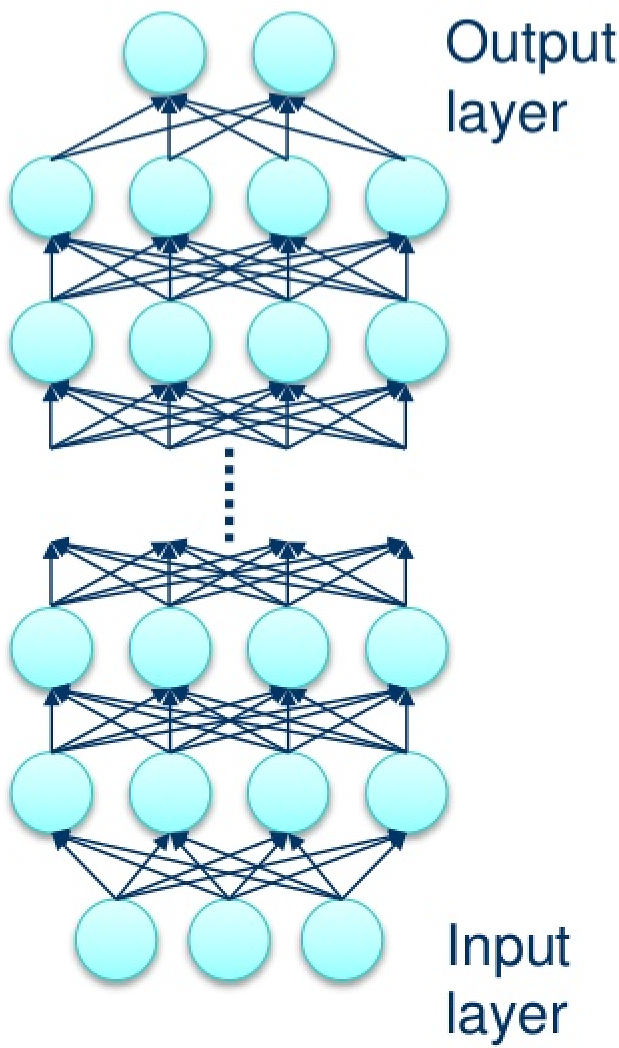
(Shallow) NN





Pros and Cons of Deep NN

Deep NN



Until a few years ago…

1. Tend to be overfitting
2. Learning information does not reach to the lower layer

・Pre-training with RBM
・Big data

Image net
More than 1,5 M: Labeled images
http://www.image-net.org/

Labeled Faces in the Wild
More than 10,000: Face images
http://vis-www.cs.umass.edu/lfw/

High-performance network

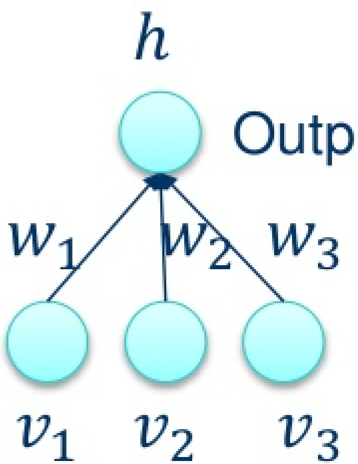
Outline

http://bit.ly/dnnicpr2014

1. Examples of Deep NNs
2. RBM to Deep NN
3. Deep Neural Network (Deep NN)
   – Back-Propagation (Supervised Learning)
4. Restricted Boltzmann Machine (RBM)
   – Mathematics, Probabilistic Model and Inference Model
   – Pre-training by Contradictive Diverse Learning (Unsupervised Learning)
5. Inference Model with Distribution

Single Layer Neural Network

- Single node output



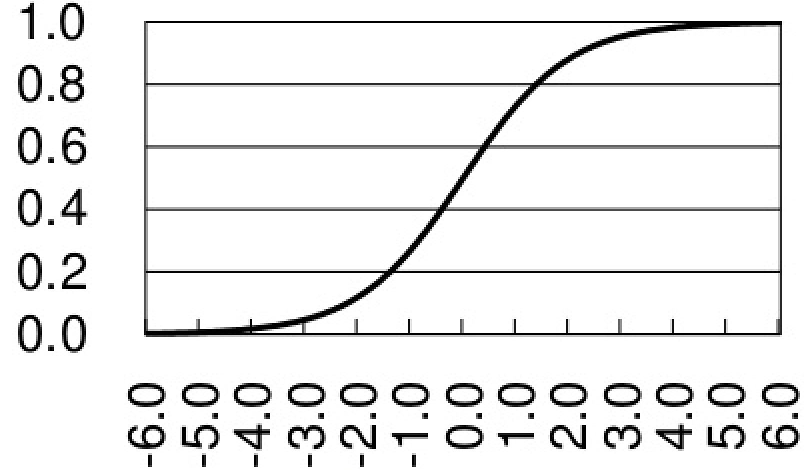
Output

Input layer

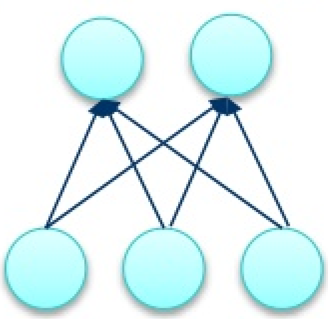
$$h = \sigma\left(\sum_i w_i v_i + b\right)$$

Sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



- Multiple nodes output (Single Layer NN)



Output layer $\boldsymbol{h}$

Input Layer $\boldsymbol{v}$

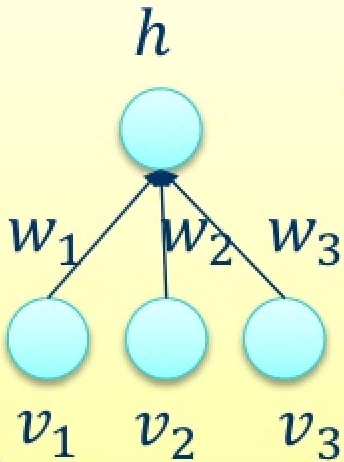
$$h_j = \sigma\left(\sum_i w_{ij} v_i + b_j\right)$$

Vector representation of Single layer NN

$$\boldsymbol{h} = \sigma(\boldsymbol{W}^T \boldsymbol{v} + \boldsymbol{b})$$

It is equivalent to the inference model of the RBM

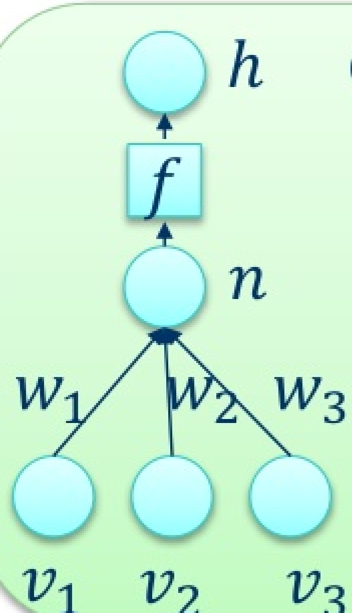
Weighted sum and Activation functions



Output layer

$$h = f\left(\sum_i w_i v_i + b\right)$$

Input layer



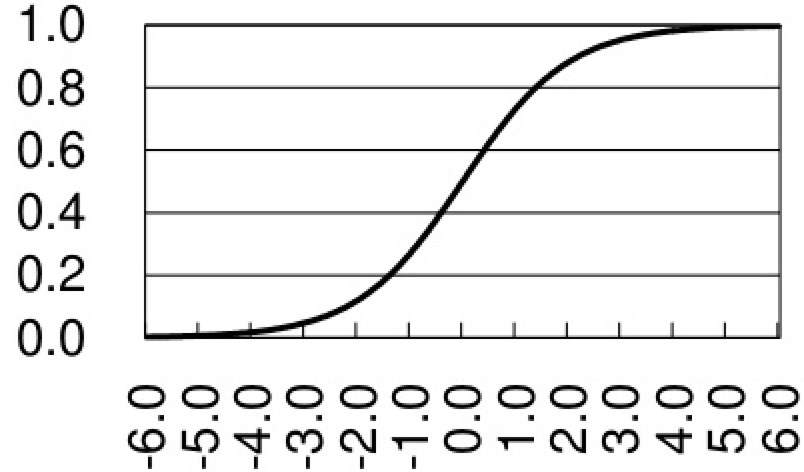
Output layer

$$h = f(n)$$

$$n = \sum_i w_i v_i + b$$

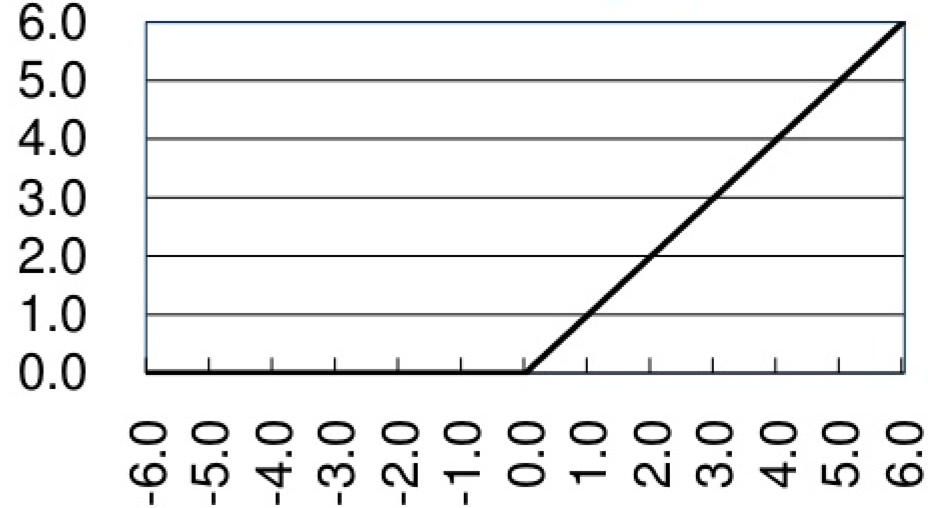
Input layer

Sigmoid function $f(x) = \sigma(x) = \dfrac{1}{1 + e^{-x}}$



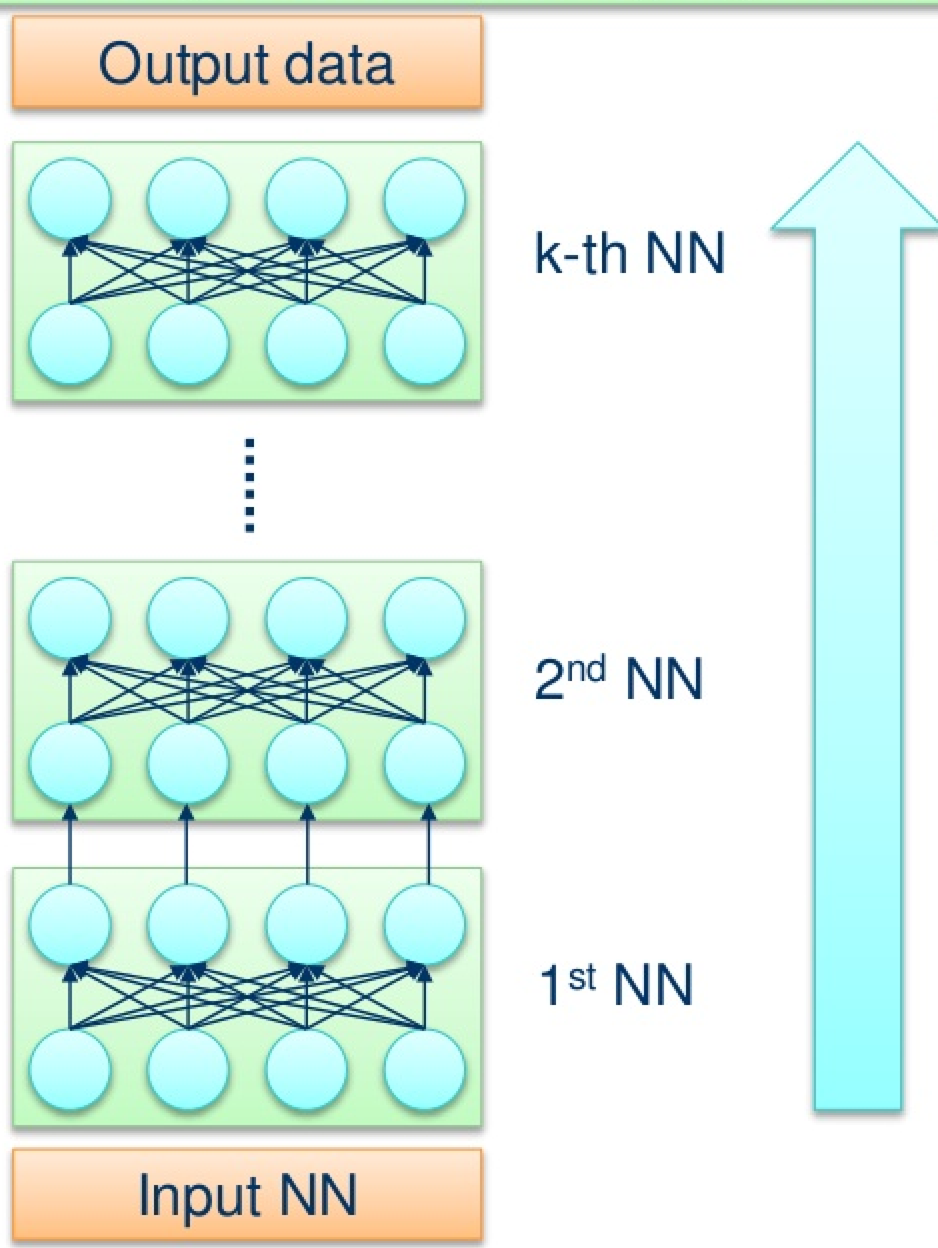
Rectified linear unit

$$f(x) = \mathrm{ReLU}(x) = \begin{cases} 0 & (x < 0) \\ x & (x \geq 0) \end{cases}$$



Single layer NN to Deep NN

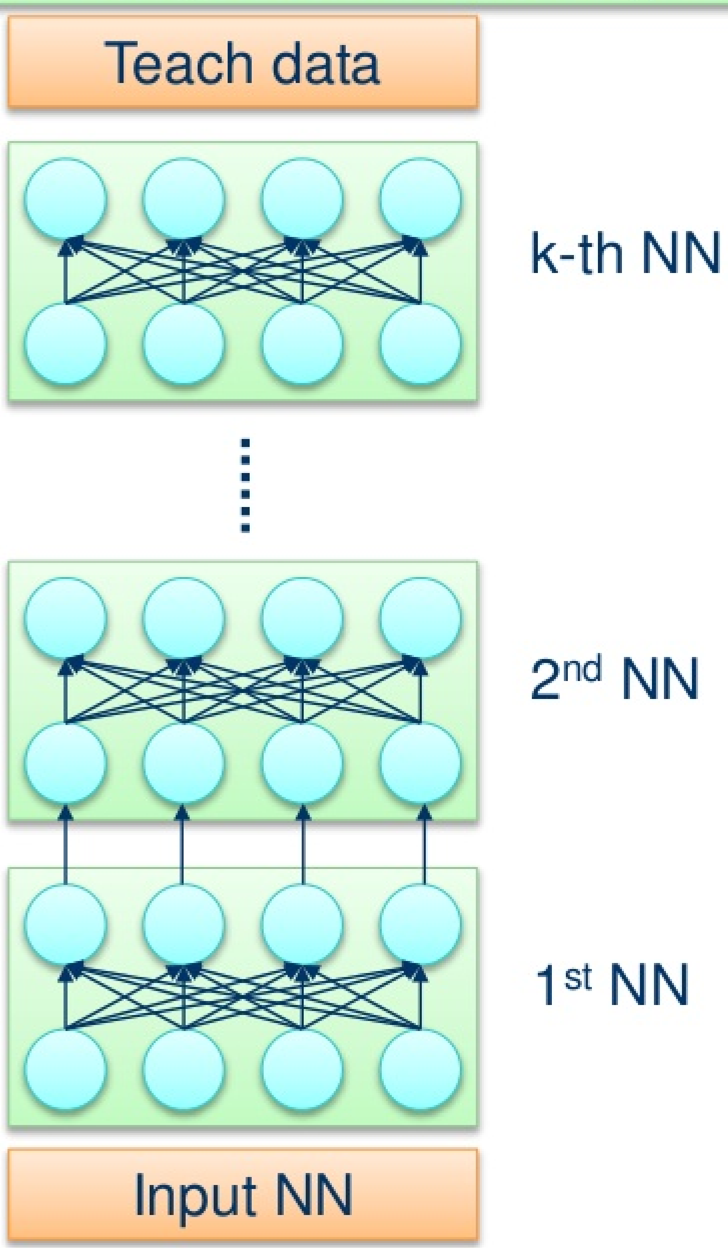
- The deep NN is build up by stacking single layer NNs.



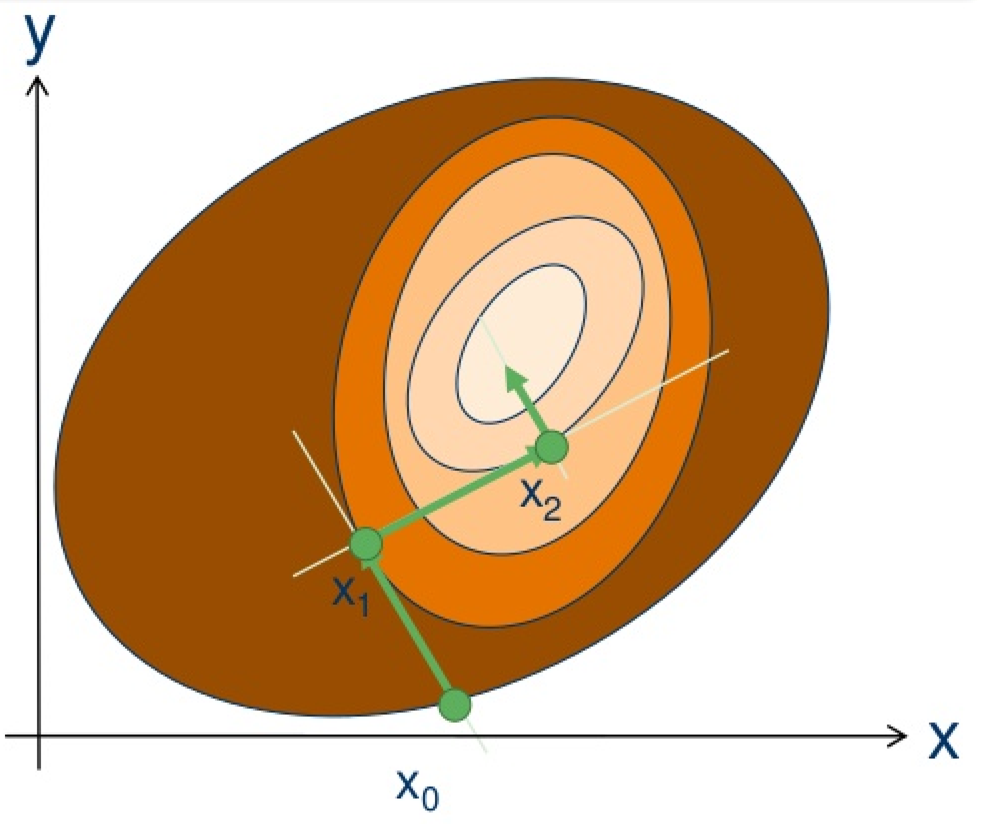
The output of the single layer NN will be the input of the next single layer NN.
The output data of the deep NN is inferred by iterating the process.

Parameters estimation for deep NN

The deep NN is build up by stacking single layer NNs.

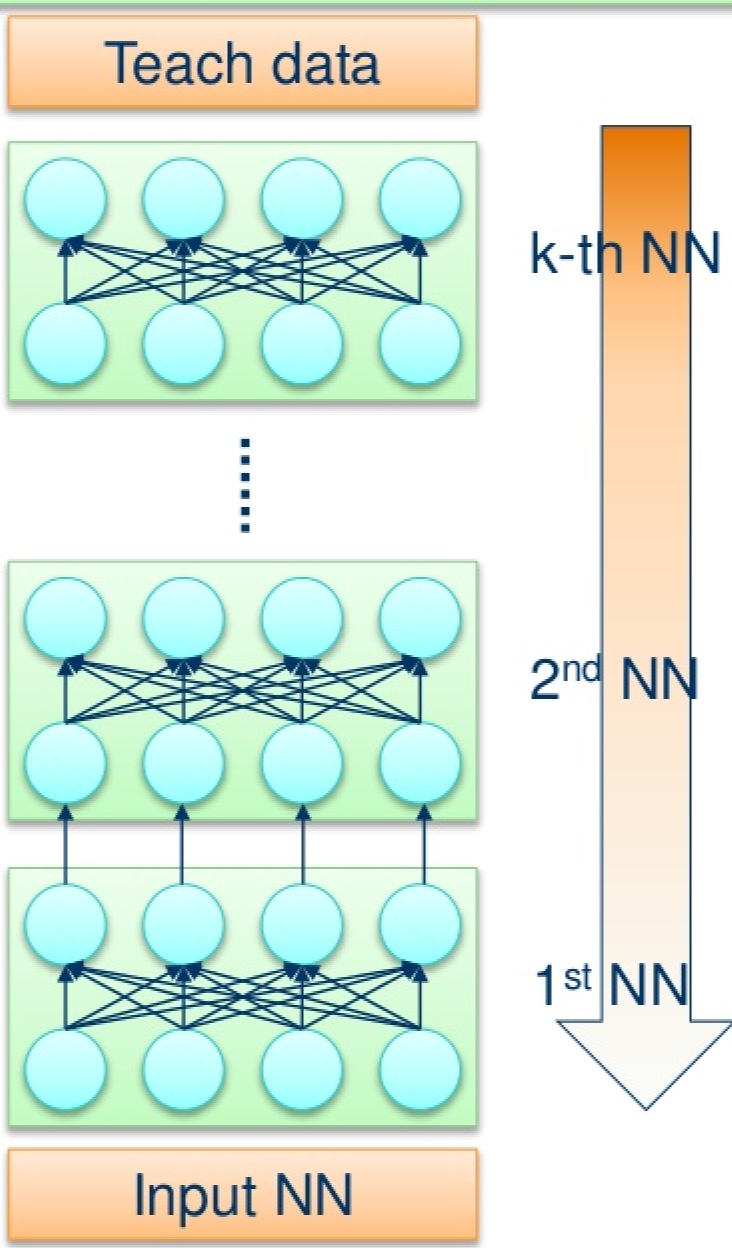


Parameters are estimated by gradient descent algorithm which minimizes the difference between the output data and teach data.



Parameters estimation for deep NN

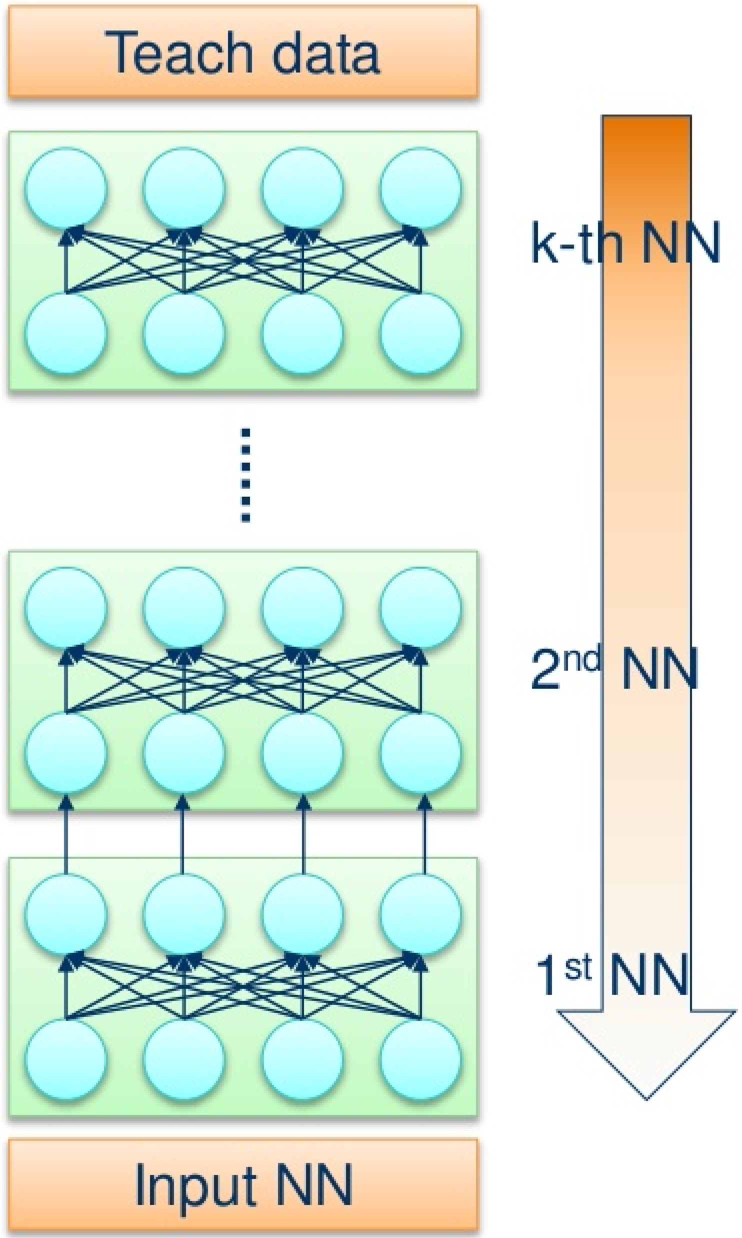
The deep NN is build up by stacking single layer NNs.



Parameters are estimated by gradient descent algorithm which minimizes the difference between the output data and teach data.

Back-propagation:
The gradients can be calculated as propagating the information backward.

Why the pre-training is necessary?

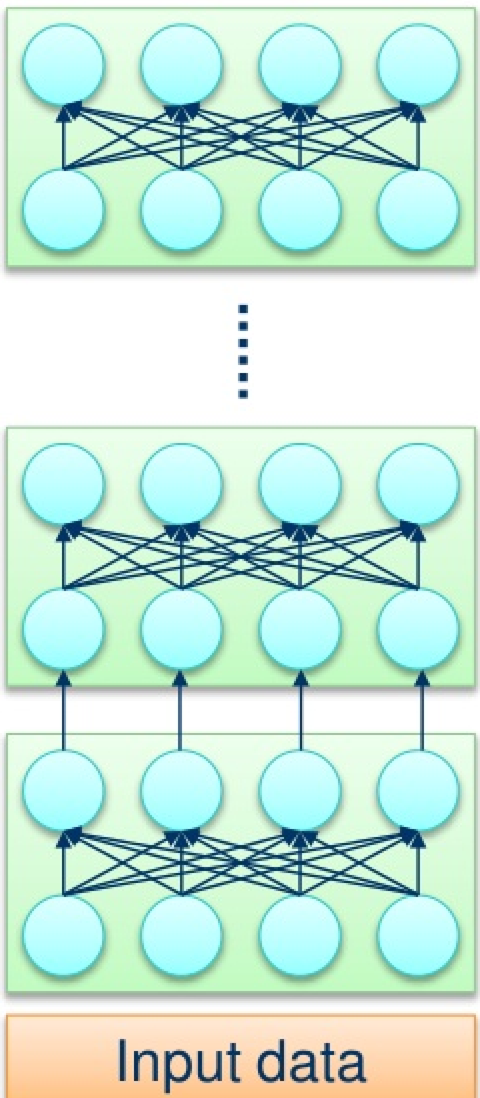


The back-propagation calculates the gradient from the output layer to the input layer.
The information of the back-propagation can not reach the deep layers.

Deep layers (1st layer, 2nd layer, …) are better to be learned by the unsupervised learning.

Pre-training with the RBMs.

Pre-training with RBMs

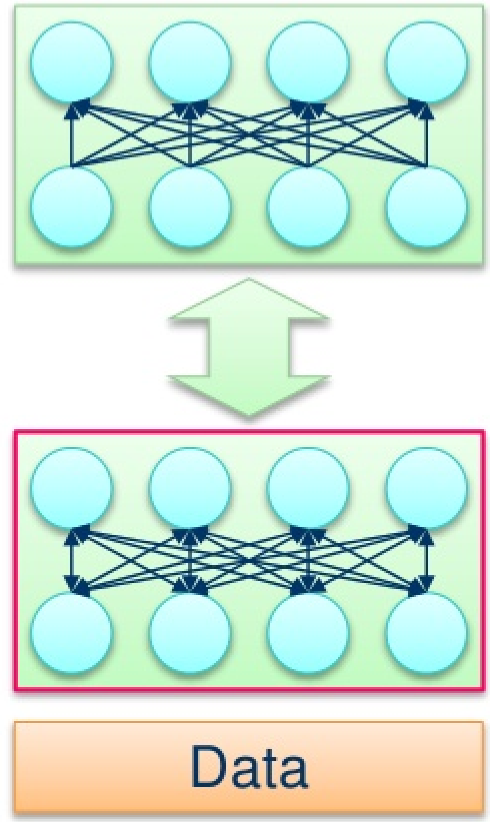


k-th NN

2nd NN

1st NN

The inference of the single layer NN is mathematically equivalent to the inference of the RBM.

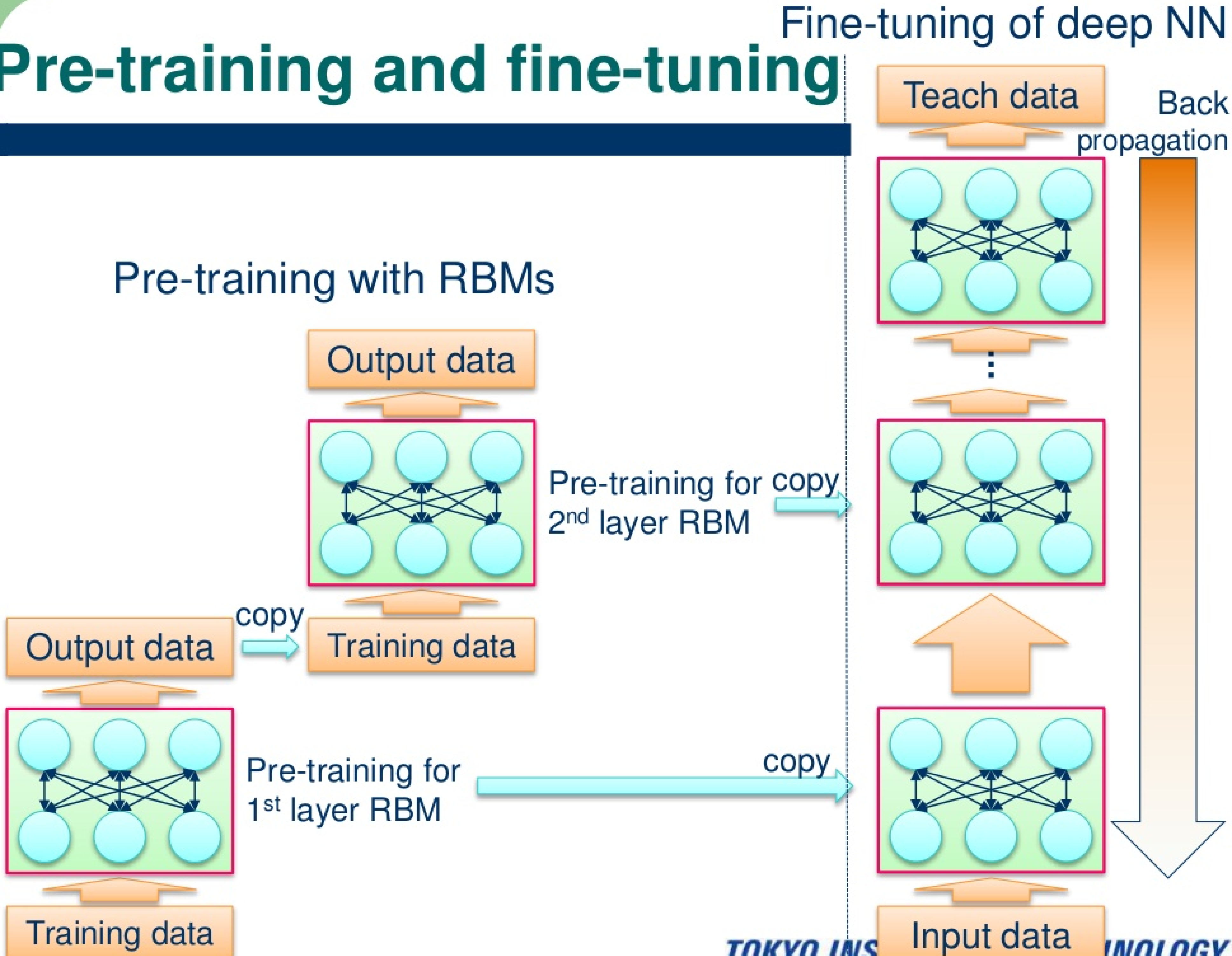


Single layer NN

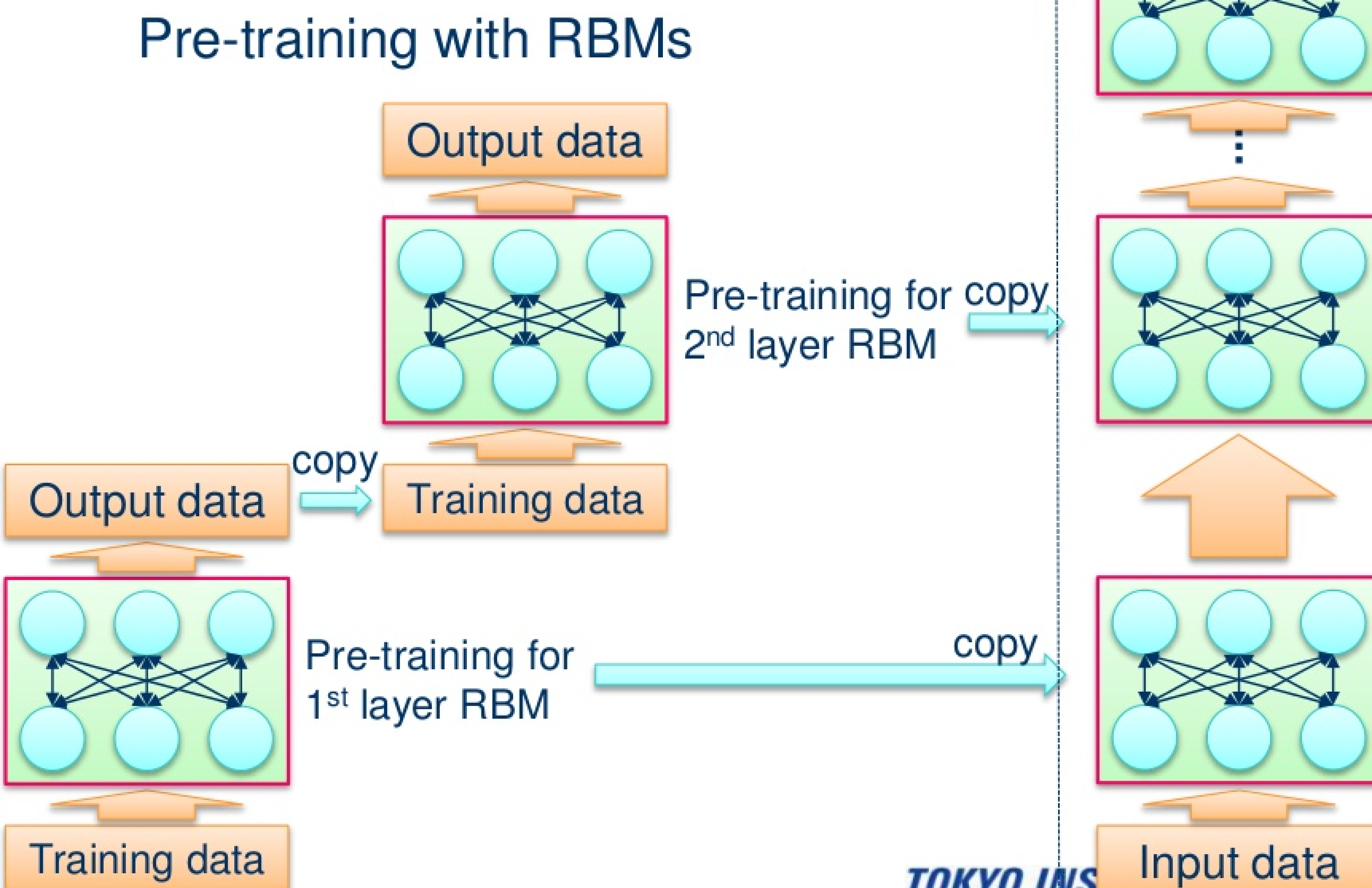
RBM

The RBM parameters are estimated by maximum likelihood algorithm with given training data.

Pre-training and fine-tuning



Feature vector extraction

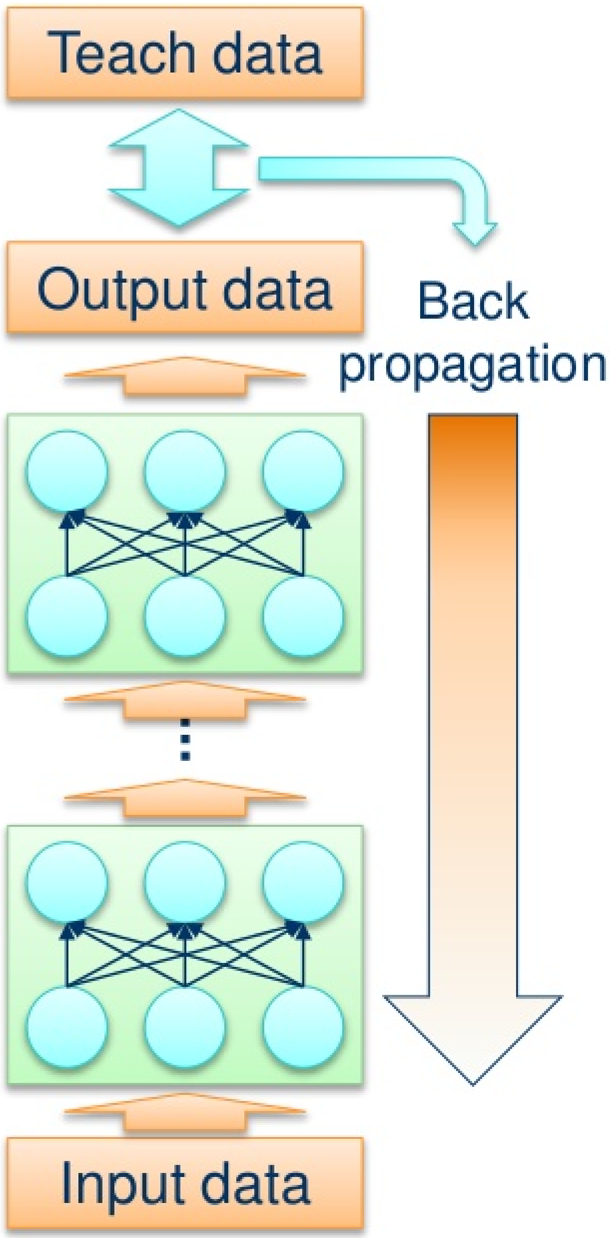


Outline

http://bit.ly/dnnicpr2014

1. Examples of Deep NNs
2. RBM to Deep NN
3. Deep Neural Network (Deep NN)
   – Back-Propagation (Supervised Learning)
4. Restricted Boltzmann Machine (RBM)
   – Mathematics, Probabilistic Model and Inference Model
   – Pre-training by Contradictive Diverse Learning (Unsupervised Learning)
5. Inference Model with Distribution

Back-Propagation Algorithm



Vector representation of the single layer NN $\boldsymbol{h} = \sigma(\boldsymbol{W}^T \boldsymbol{v} + \boldsymbol{b})$

The goal of learning:
Weights W and bias b of the each layer are estimated, so that the differences between the output data and the teach data are minimized.

Objective function $I = \frac{1}{2}\sum_k \left(h_k^{(L)} - t_k\right)^2$

Efficient calculation of the gradient $\frac{\partial I}{\partial \boldsymbol{W}^{(\ell)}}$ is important.

Back-propagation algorithm is an efficient algorithm to calculate the gradients.