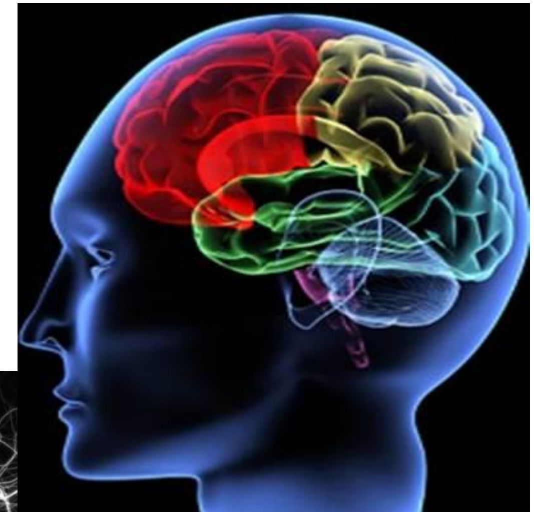
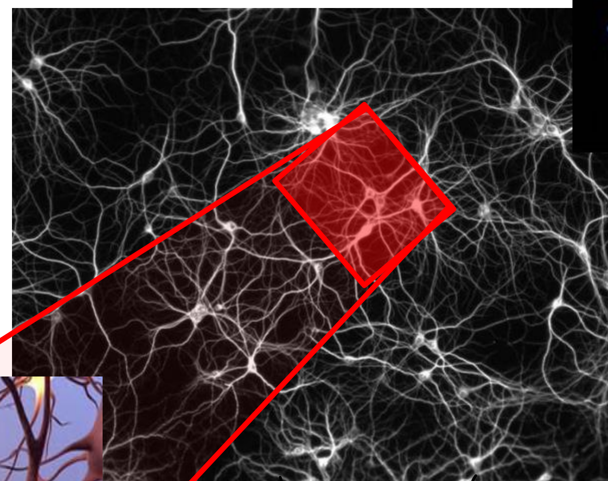
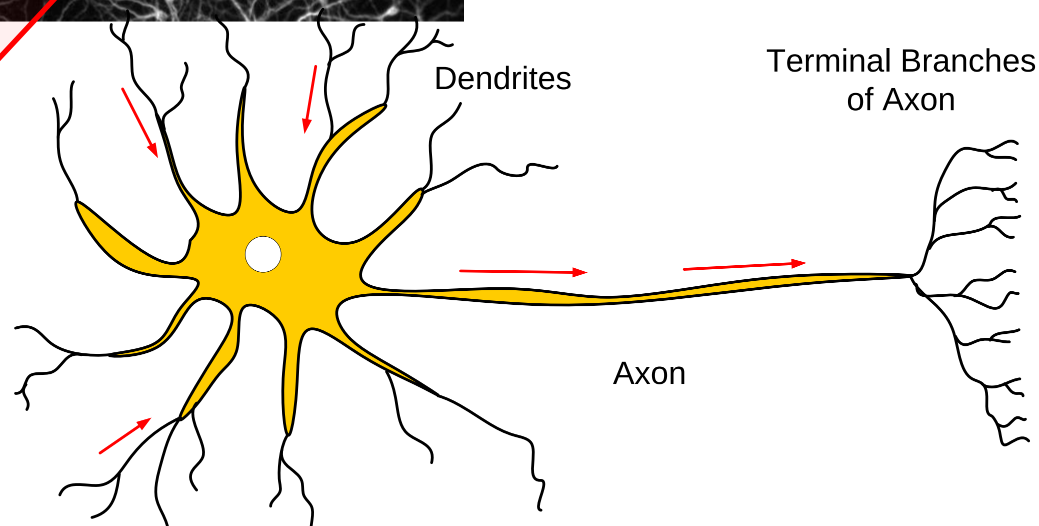
Deep Learning and Application in Neural Networks

Hugo Larochelle
Yoshua Bengio
Jerome Louradour
Pascal Lamblin
Geoffrey Hinton
Andrew Ng.
Andrew L. Nelson
R. Salskhutdinov

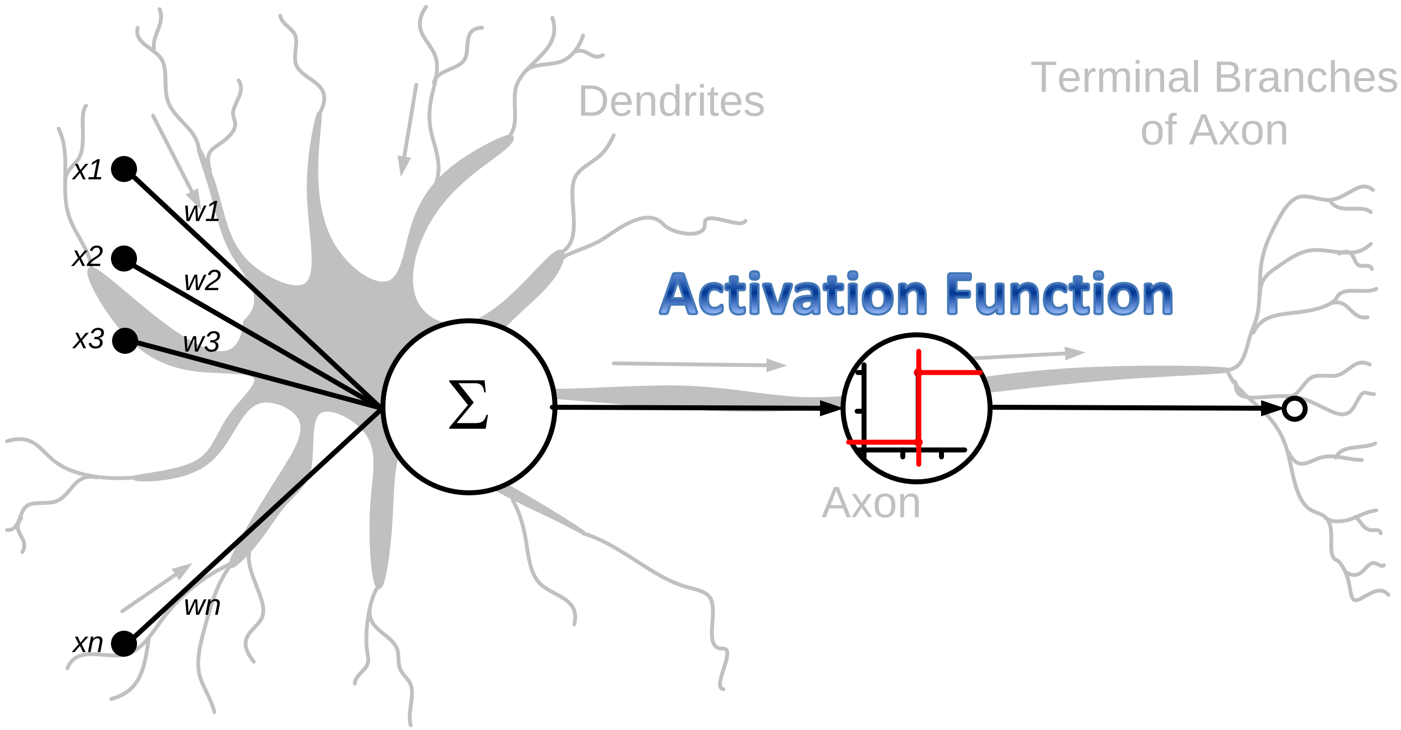
Biological Neurons





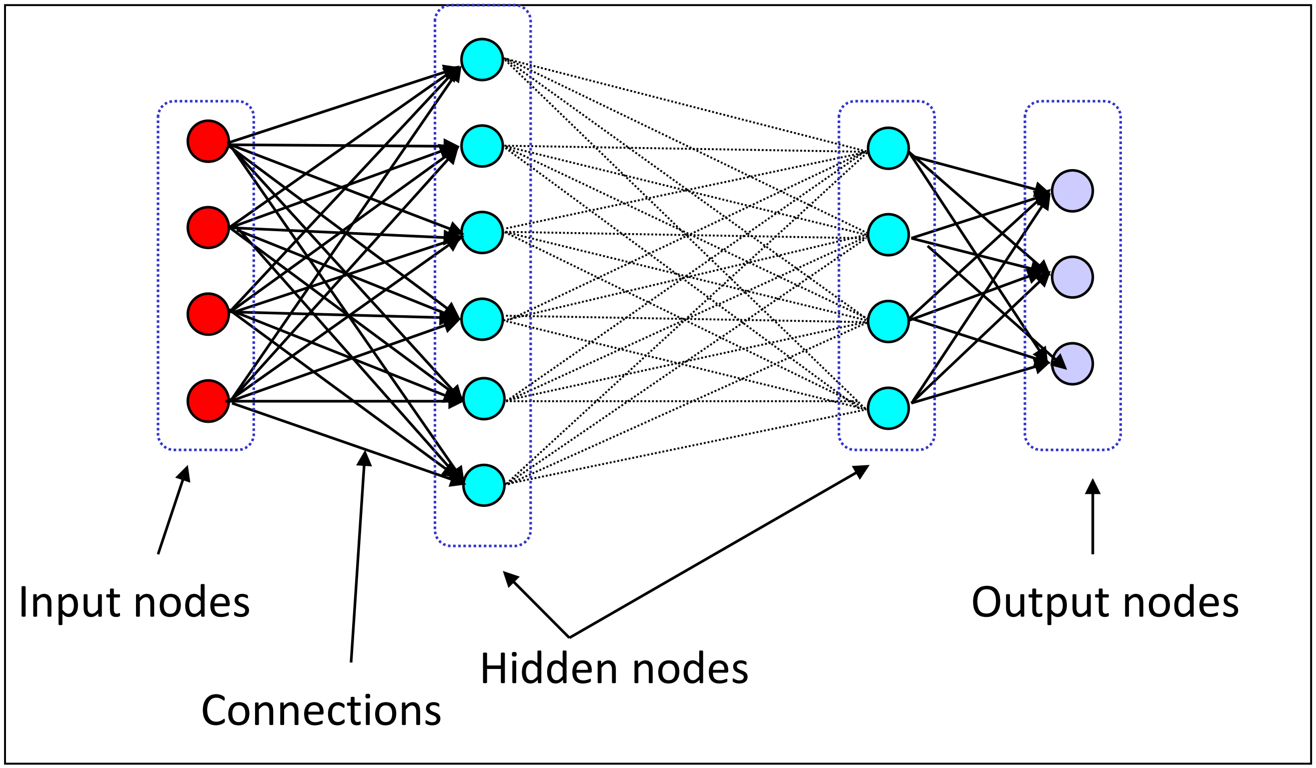


Artificial Neural Networks (ANN)



Slide credit : Andrew L. Nelson

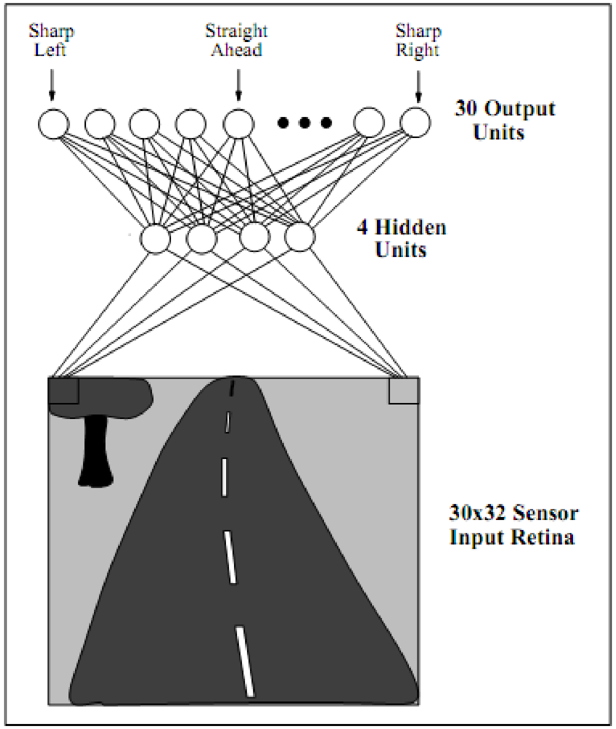
Layered Networks



$$Output: \quad y_i = f(w_i^1 x_1 + w_i^2 x_2 + w_i^3 x_3 + \cdots + w_i^m x_m)$$
$$= f(\sum_j w_i^j x_j)$$

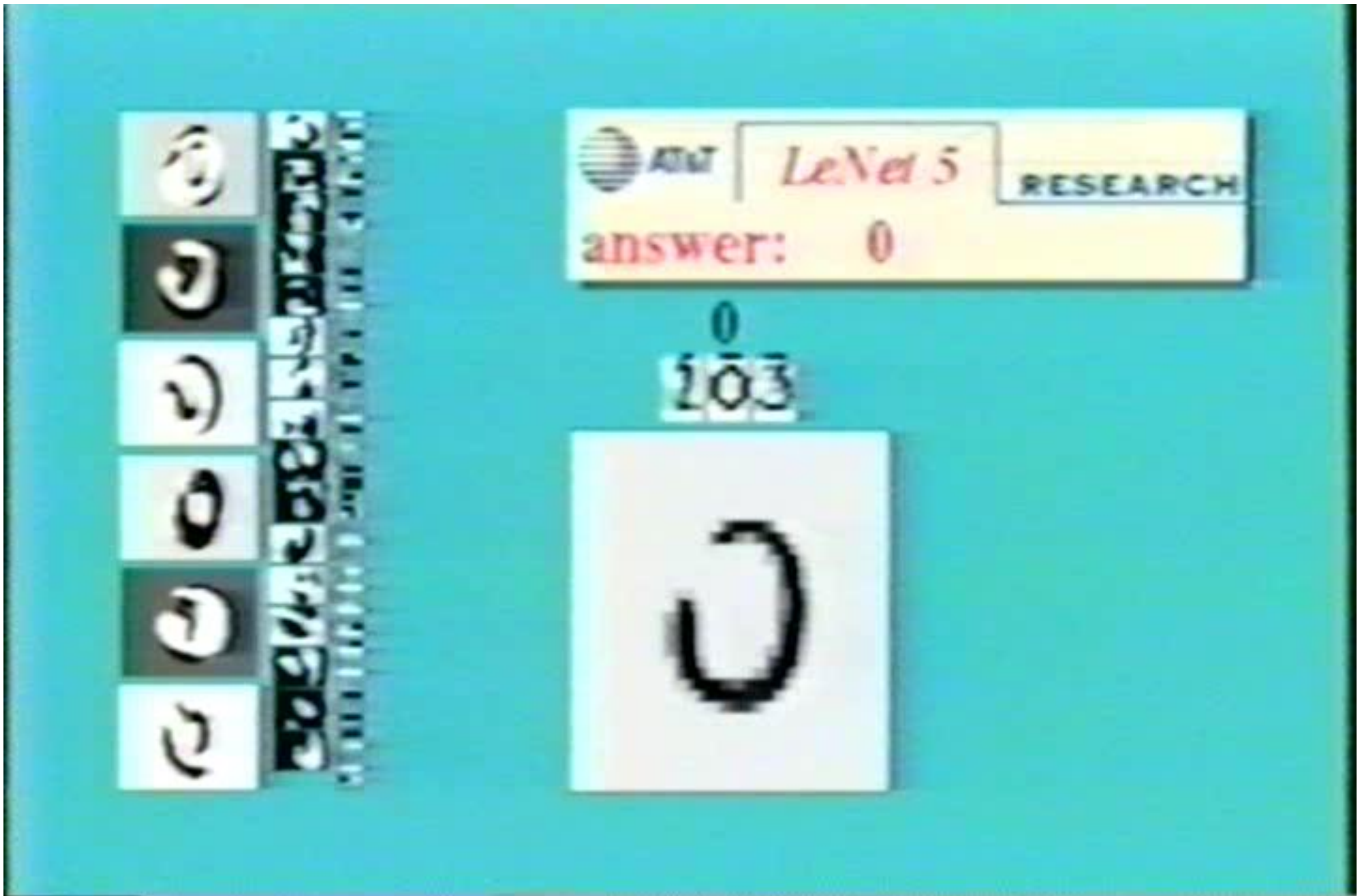
Neural network application

- ALVINN drives 70 mph on highways



Slide credit : T.Michell

Neural network application

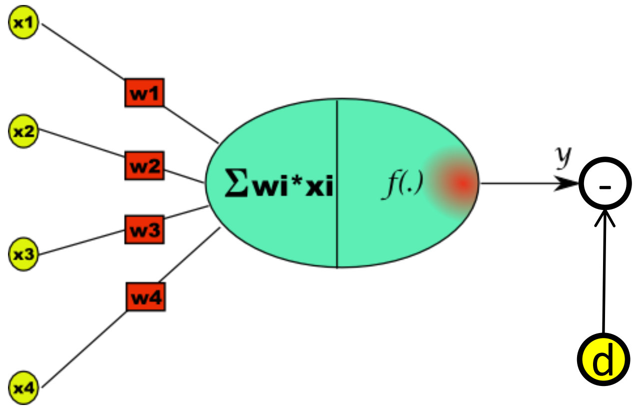


Slide credit : Andraw Ng.

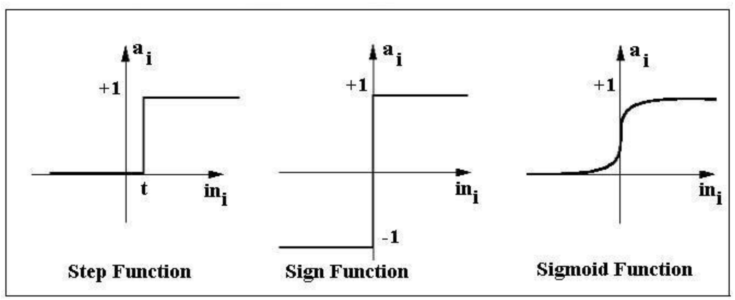
The simplest model- the Perceptron

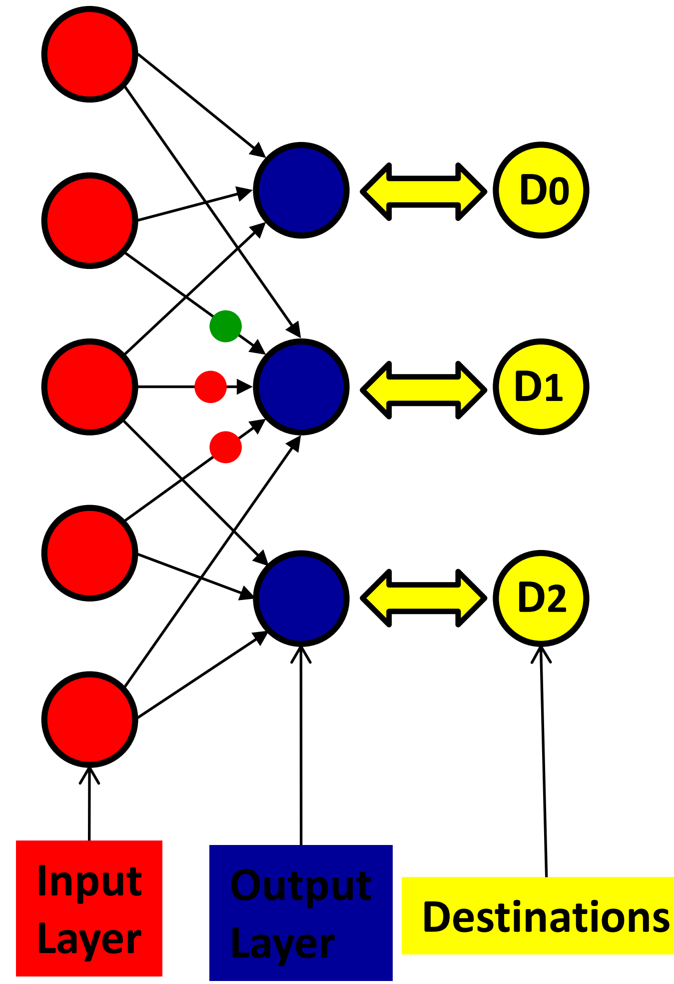
- The Perceptron was introduced in 1957 by Frank Rosenblatt.

Perceptron:



Activation functions:





Learning:

$$y^{(t)} = f\left\{\sum_i w_i^{(t)} x_i^{(t)}\right\}$$

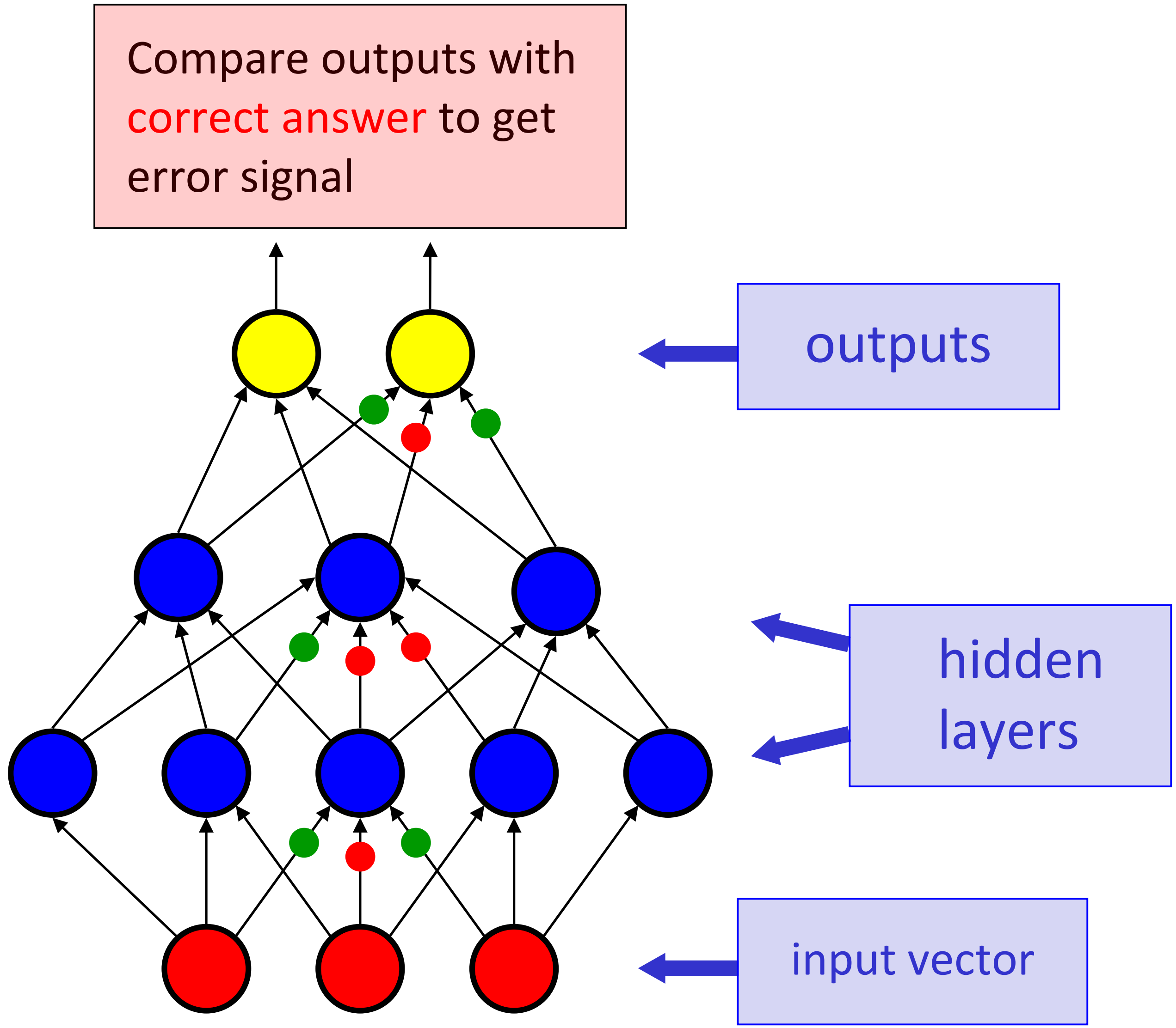
Update

$$\begin{cases} \Delta w_i^{(t)} = \varepsilon\left(d^{(t)} - y^{(t)}\right) x_i^{(t)} \\ w_i^{(t+1)} = w_i^{(t)} + \Delta w_i^{(t)} \end{cases}$$

Slide credit : Geoffrey Hinton

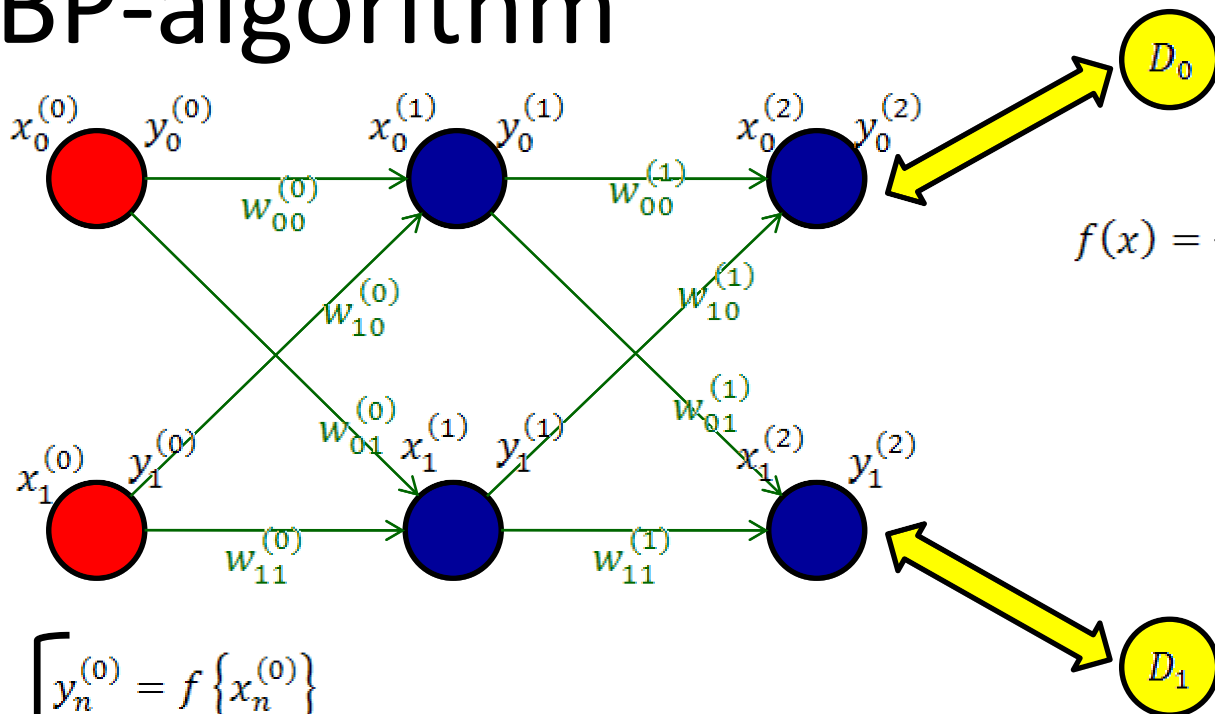
Second generation neural networks (~1985)

Back-propagate error signal to get derivatives for learning

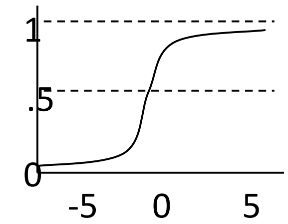


Slide credit : Geoffrey Hinton

BP-algorithm



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



Activations

$$y_n^{(0)} = f\left\{x_n^{(0)}\right\}$$

$$y_n^{(1)} = f\left\{\sum_{i=0}^{1} y_i^{(0)} w_{in}^{(0)}\right\}$$

$$y_n^{(2)} = f\left\{\sum_{i=0}^{1} y_i^{(1)} w_{in}^{(1)}\right\}$$

errors

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = -2\left(D_j - y_j^{(2)}\right) f\left\{x_j^{(2)}\right\}\left(1 - f\left\{x_j^{(2)}\right\}\right) y_i^{(1)} \stackrel{\text{def}}{=} \delta_j^{(1)} y_i^{(1)}$$

$$\frac{\partial E}{\partial w_{ij}^{(0)}} = f\left\{x_j^{(1)}\right\}\left(1 - f\left\{x_j^{(1)}\right\}\right) \sum_{n=0}^{1} \delta_n^{(1)} w_{jn}^{(1)} x_i^{(0)} \stackrel{\text{def}}{=} \delta_j^{(0)} x_i^{(0)}$$

The error: $E = \sum_{n=0}^{1} \left(D_n - y_n^{(2)}\right)^2$

Update Weights: $w_{ij}^{(k)} = w_{ij}^{(k)} + \varepsilon \frac{\partial E}{\partial w_{ij}^{(k)}}$

Update

$$w_{ij}^{(1)} = w_{ij}^{(1)} + \varepsilon \delta_j^{(1)} y_i^{(1)}$$

$$w_{ij}^{(0)} = w_{ij}^{(0)} + \varepsilon \delta_j^{(0)} x_i^{(0)}$$

Slide credit : Geoffrey Hinton

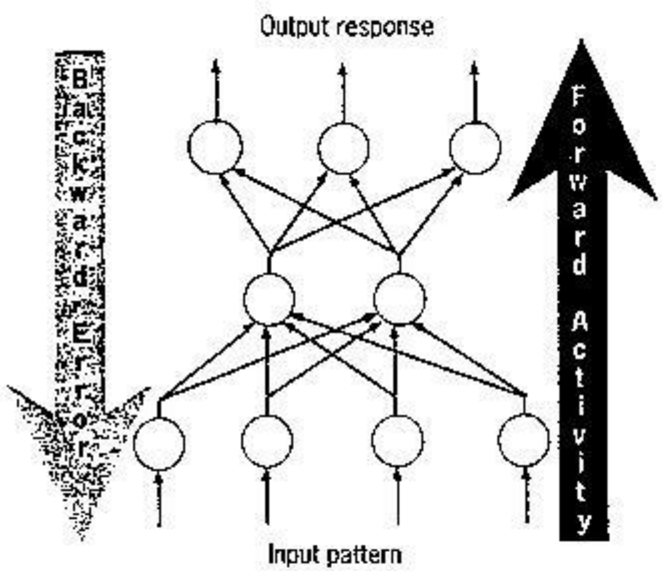
Back Propagation

Advantages

• Multi layer Perceptron network can be trained by the back propagation algorithm to perform any mapping between the input and the output.

What is wrong with back-propagation?

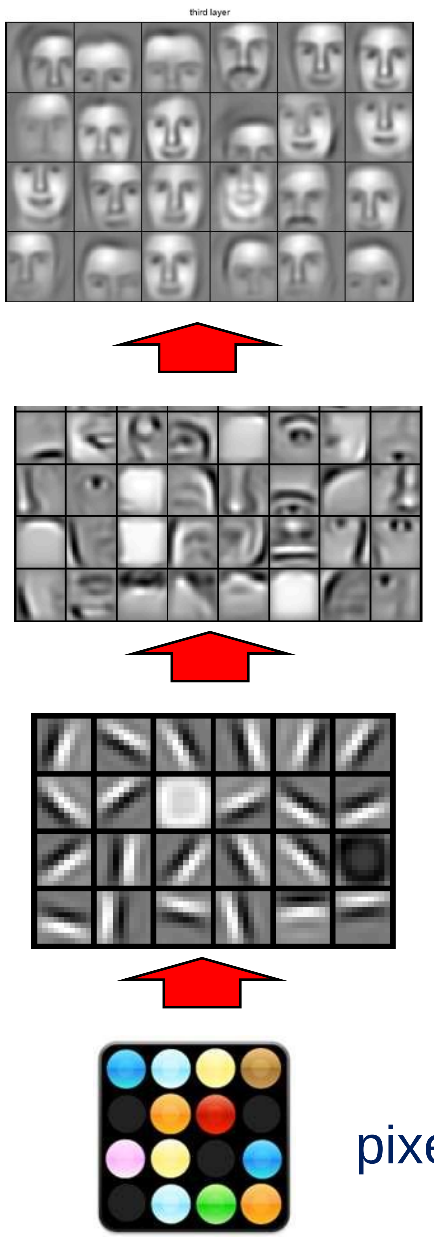
- It requires labeled training data.
  - Almost all data is unlabeled.
- The learning time does not scale well
  - It is very slow in networks with multiple hidden layers.
- It can get stuck in poor local optima.



A backpropagation network trains with a two-step procedure. The activity from the input pattern flows forward through the network, and the error signal flows backward to adjust the weights.

Slide credit : Geoffrey Hinton

Why Deep multi-layered neural network

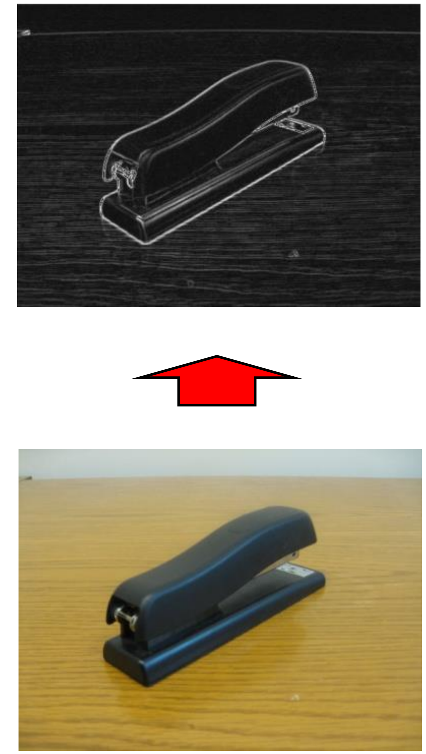


object models

object parts
(combination
of edges)

edges

pixels



Before 2006

Failing to train deep architectures

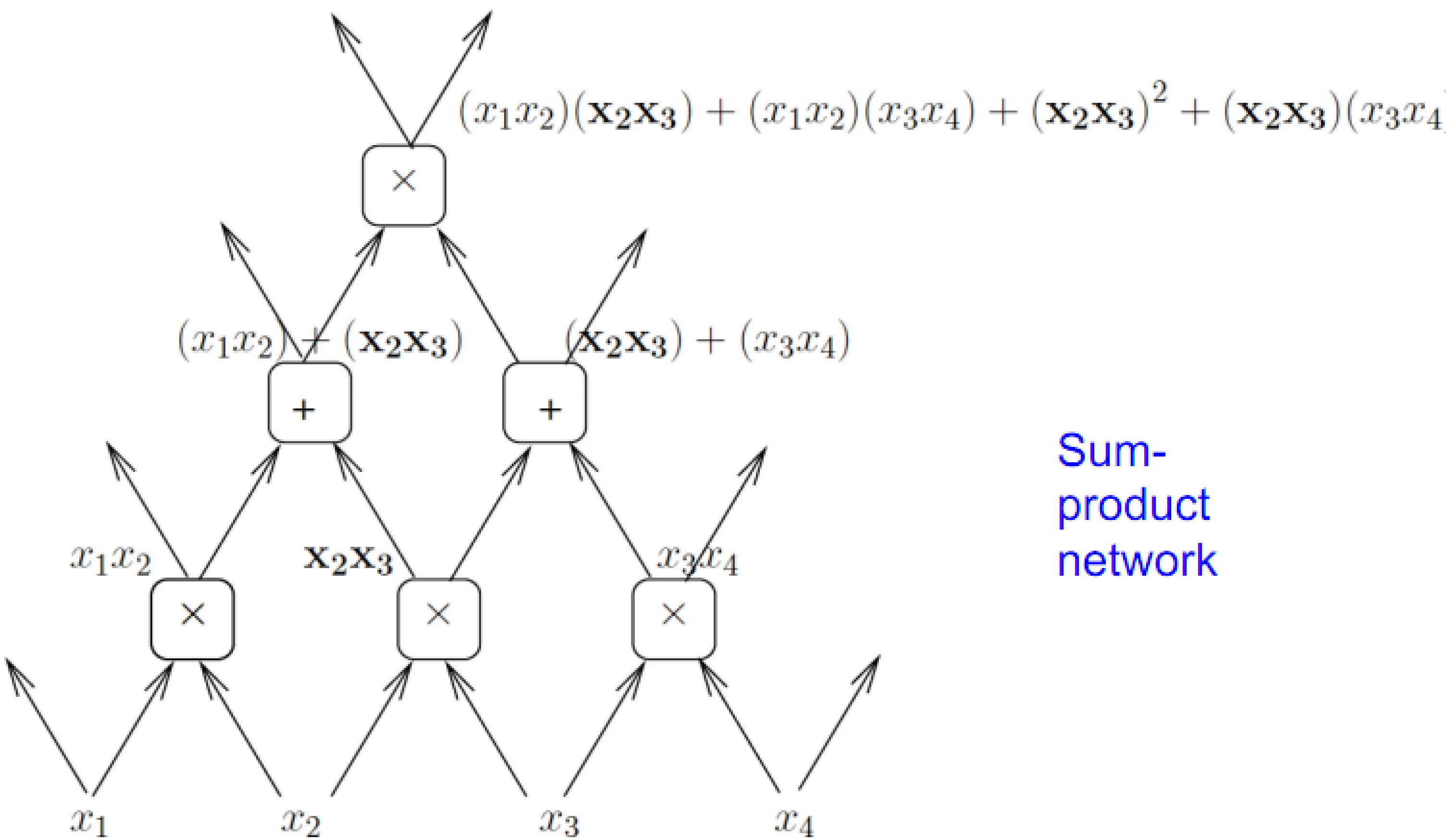
Deep Neural Networks

- Standard learning strategy
  - Randomly initializing the weights of the network
  - Applying gradient descent using backpropagation
- But, backpropagation does not work well (if randomly initialized)
  - Deep networks trained with back-propagation (without unsupervised pre-train) perform worse than shallow networks
  - ANN have limited to one or two layers

Sharing Components in a Deep Architecture

Polynomial expressed with shared components:

advantage of depth may grow exponentially



Sum-product network

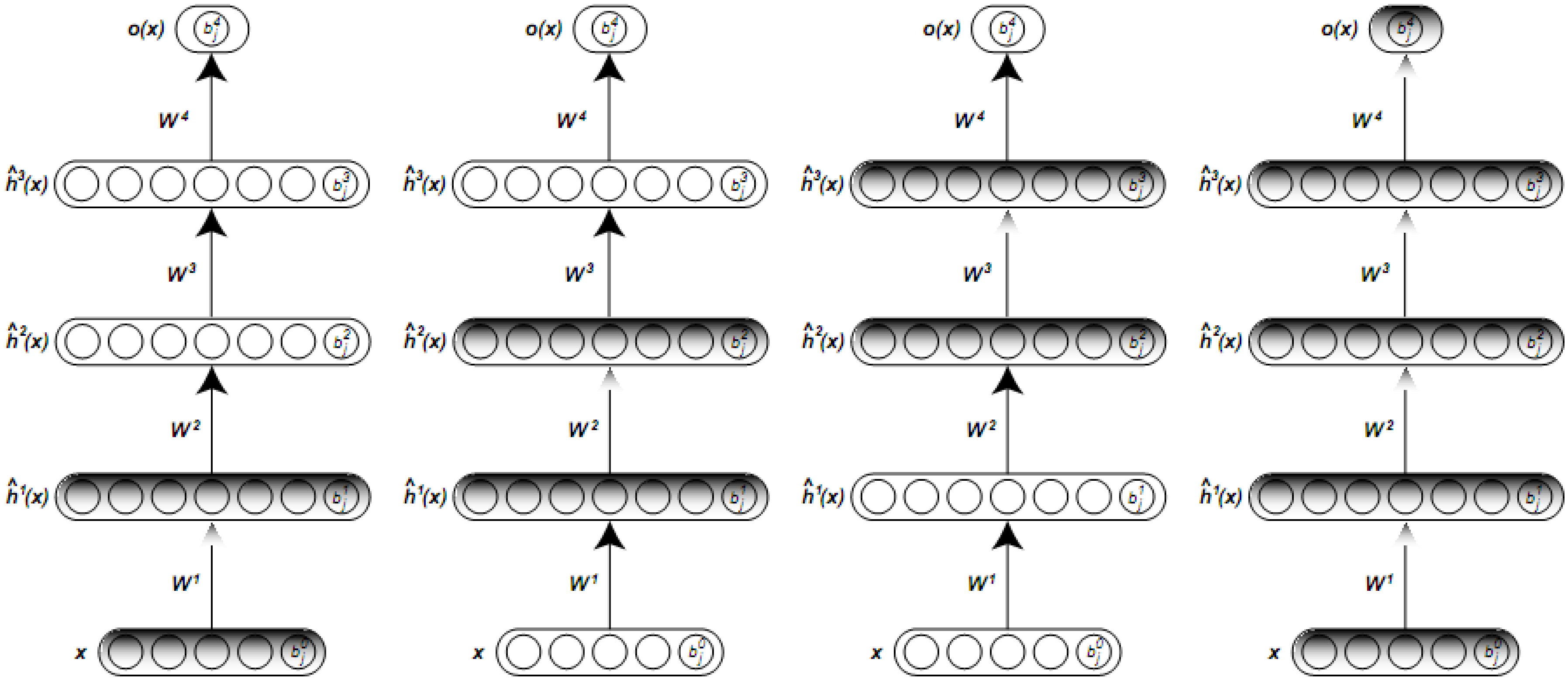
2006

Breakthrough!

2006: The Deep Breakthrough

- Hinton, Osindero & Teh « A Fast Learning Algorithm for Deep Belief Nets », *Neural Computation*, 2006
- Bengio, Lamblin, Popovici, Larochelle « Greedy Layer-Wise Training of Deep Networks », *NIPS'2006*
- Ranzato, Poultney, Chopra, LeCun « Efficient Learning of Sparse Representations with an Energy-Based Model », *NIPS'2006*

Unsupervised greedy layer-wise training procedure.



(a) First hidden layer pre-training

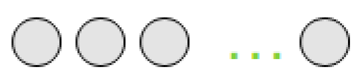
(b) Second hidden layer pre-training

(c) Third hidden layer pre-training

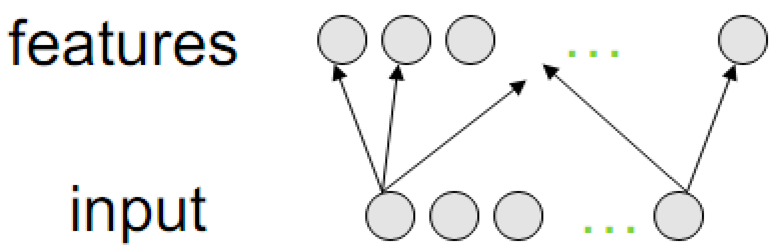
(d) Fine-tuning of whole network

Deep training

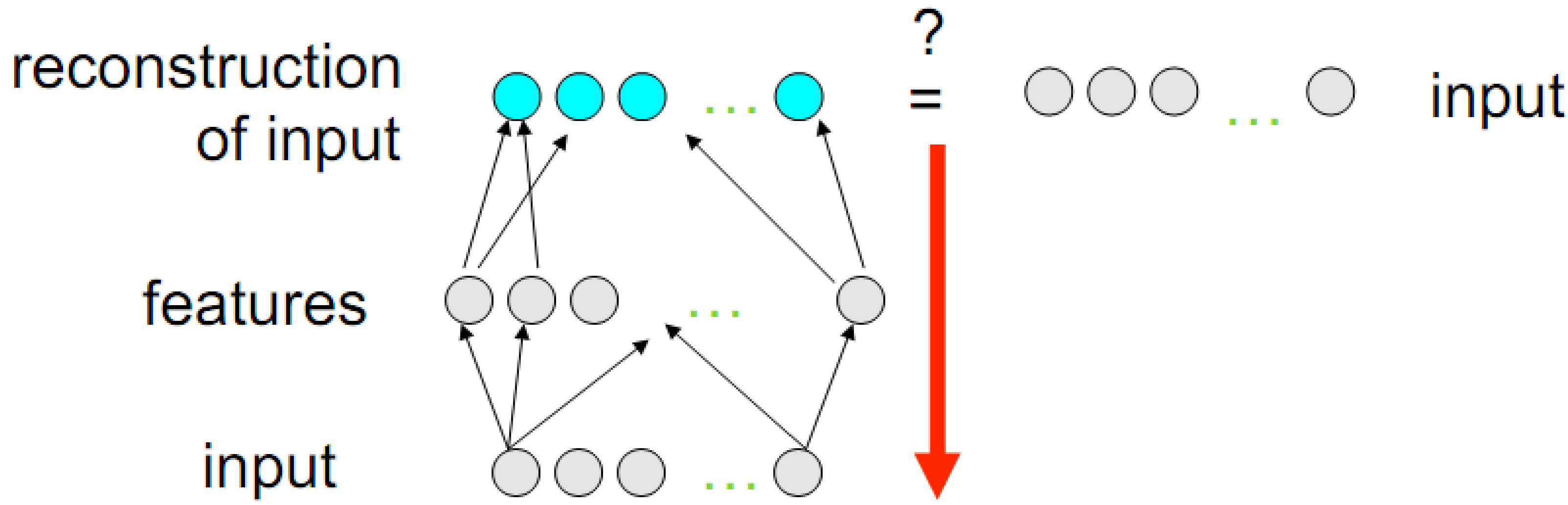
input



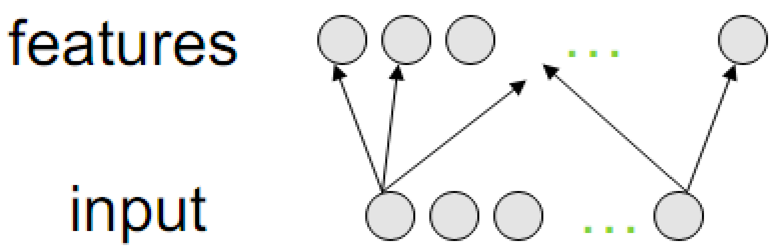
Layer-Wise Unsupervised Pre-training



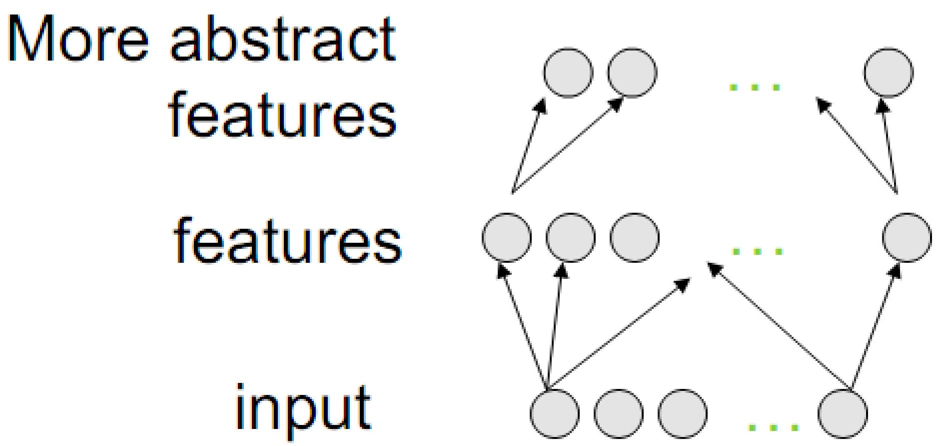
Layer-Wise Unsupervised Pre-training



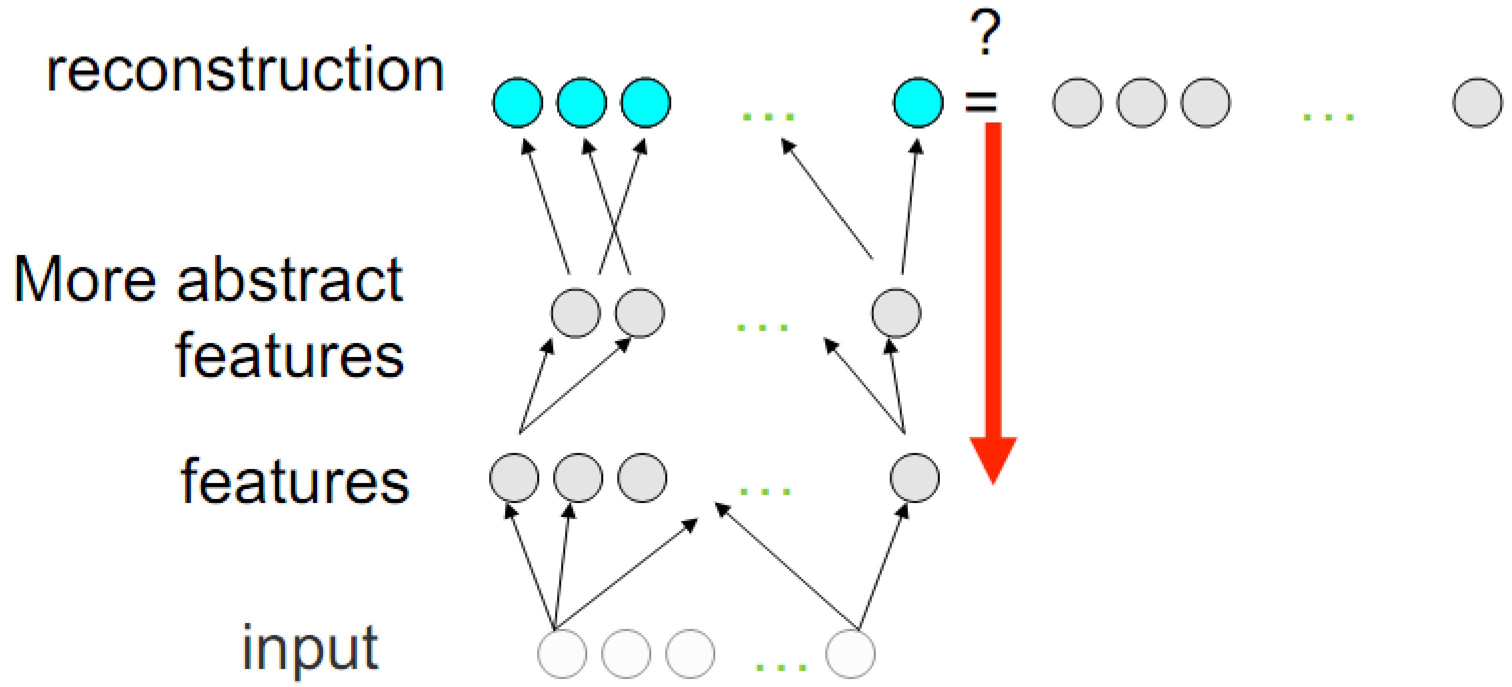
Layer-Wise Unsupervised Pre-training



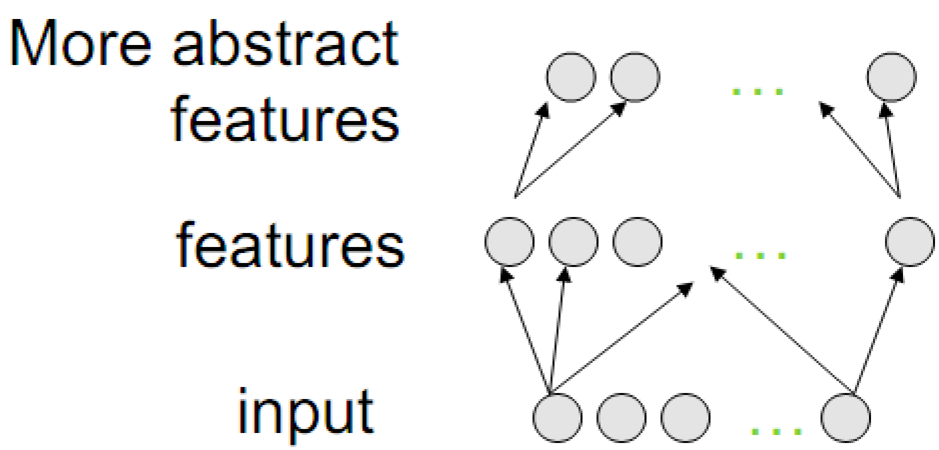
Layer-Wise Unsupervised Pre-training



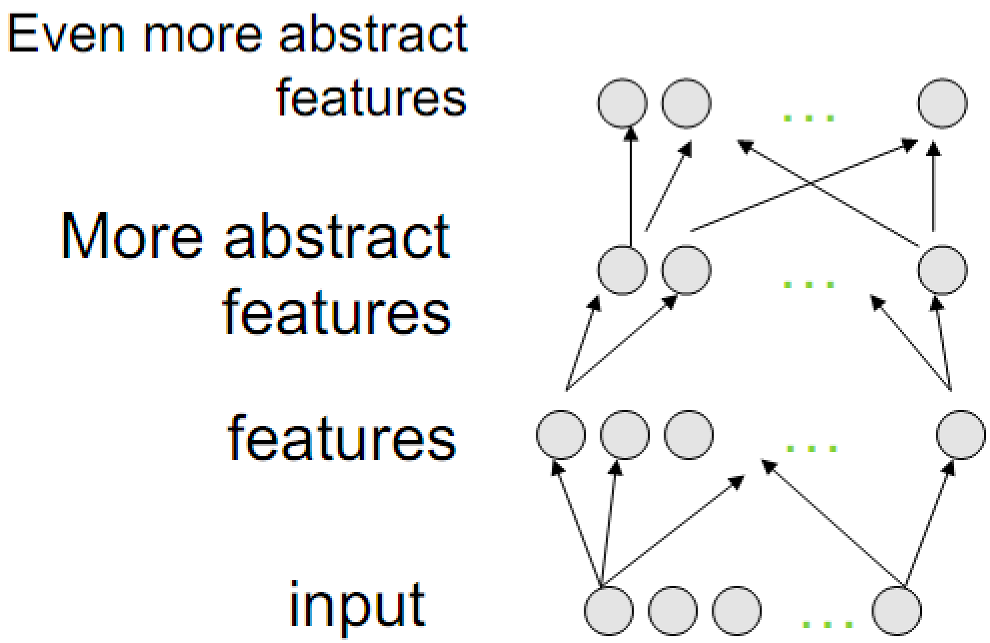
Layer-Wise Unsupervised Pre-training



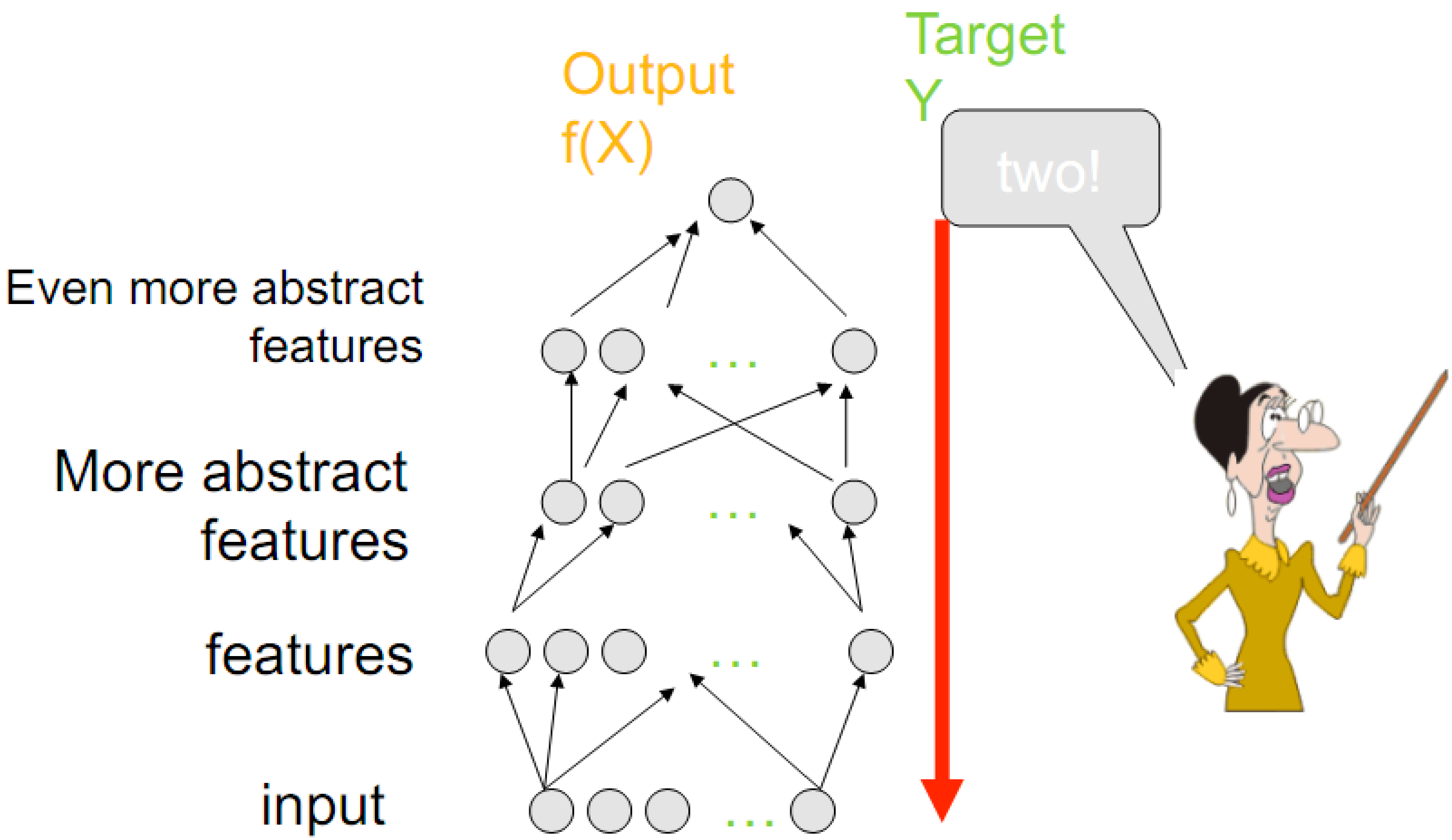
Layer-Wise Unsupervised Pre-training



Layer-Wise Unsupervised Pre-training



Supervised Fine-Tuning



Layer-Local Unsupervised Learning

- Restricted Boltzmann Machine (SRBM) *(Hinton et al, NC'2006)*
- Auto-encoders *(Bengio et al, NIPS'2006)*
- Sparse auto-encoders *(Ranzato et al, NIPS'2006)*
- Kernel PCA *(Erhan 2008)*
- Denoising auto-encoders *(Vincent et al, ICML'2008)*
- Unsupervised embedding *(Weston et al, ICML'2008)*
- Slow features *(Mohabi et al, ICML'2009, Bergstra & Bengio NIPS'2009)*

Experiments

- MNIST data set
  - A benchmark for handwritten digit recognition
  - The number of classes is 10 (corresponding to the digits from 0 to 9)
  - The inputs were scaled between 0 and 1

Exploring Strategies for Training Deep Neural Networks.
Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, Pascal Lamblin; 10(Jan):1--40, 2009.

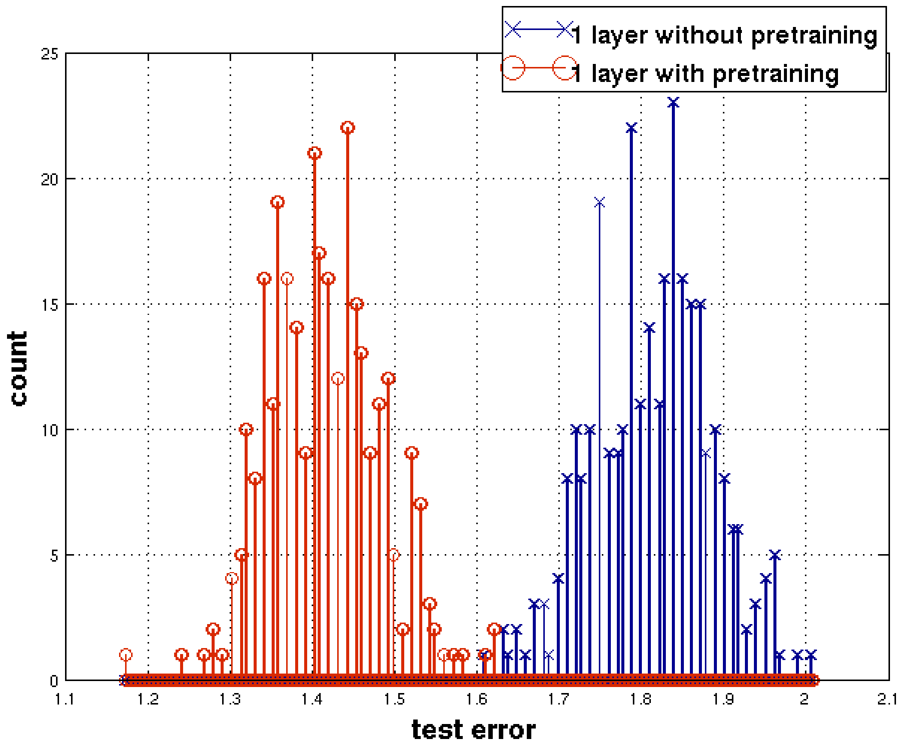
Classification error on MNIST training

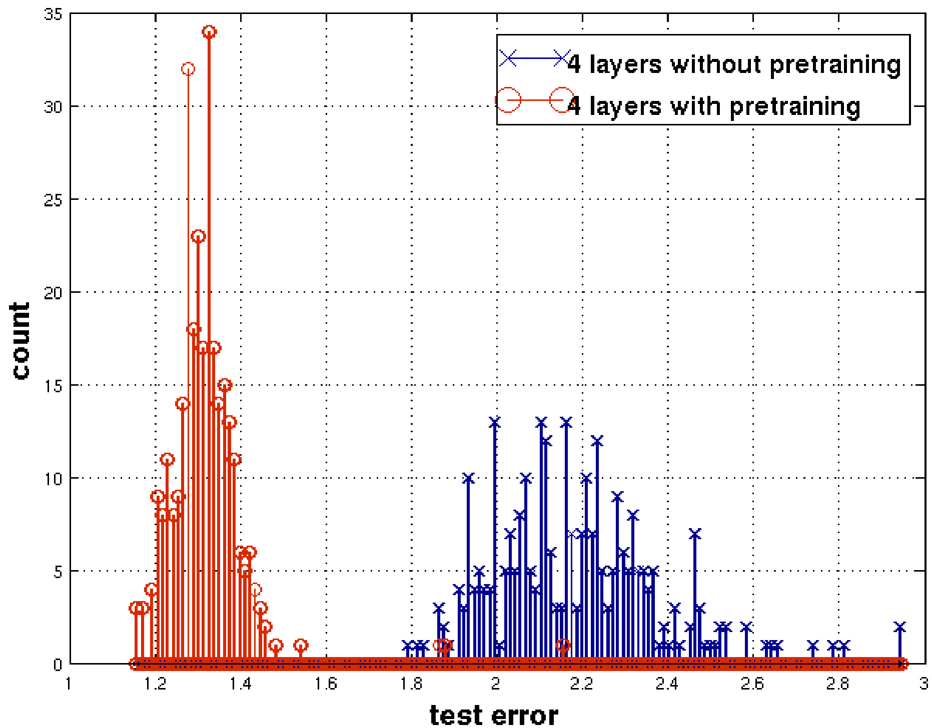
Models	Train.	Valid.	Test
SRBM (stacked restricted Boltzmann machines) network	0%	1.20%	1.20%
SAA (stacked autoassociators) network	0%	1.31%	1.41%
Stacked logistic autoregressions network	0%	1.65%	1.85%
Deep network with supervised pre-training	0%	1.74%	2.04%
Deep network, no pre-training	0.004%	2.07%	2.40%
Shallow network, no pre-training	0%	1.91%	1.93%

Table 1: Classification error on MNIST training, validation, and test sets, with the best hyperparameters according to validation error.

Experiments

- Histograms presenting the test errors obtained on MNIST using models trained with or without pre-training.

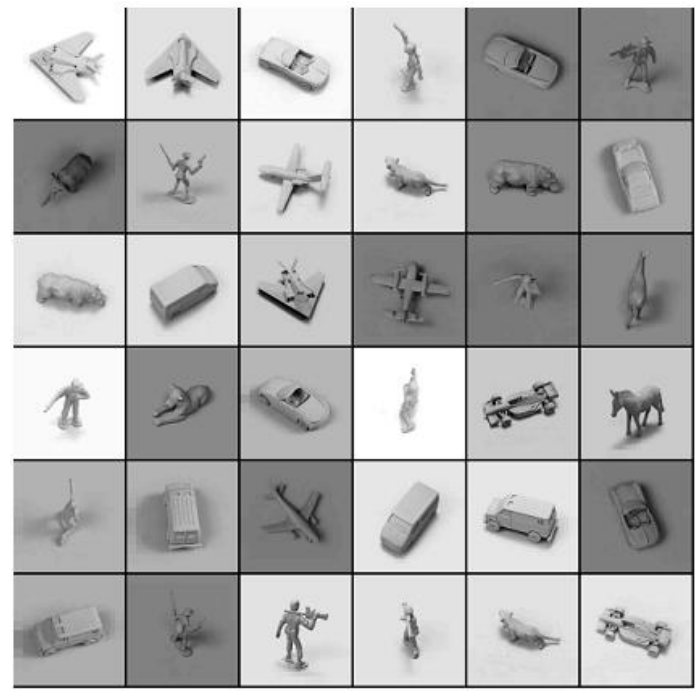




The difficulty of training deep architectures and the effect of unsupervised pre-training. Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent; pages 153-160, 2009.

Experiments

- NORB data set
  - 5 object categories, 5 difference objects within each category.
- Classification error rate
  - DBM : 10.8 %
  - SVM : 11.6 %
  - Logistic Regression : 22.5 %
  - KNN : 18.4 %



Efficient Learning of Deep Boltzmann Machines.
Ruslan Salakhutdinov, Hugo Larochelle ;
JMLR W&CP 9:693-700, 2010.

Conclusion

- Deep Learning : powerful arguments & generalization priciples
- Unsupervised Feature Learning is crucial many new algorithms and applications in recent years
- Deep Learning suited for multi-task learning, domain adaptation and semi-learning with few labels

Application

- Classification *(Bengio et al., 2007; Ranzato et al., 2007b; Larochelle et al., 2007; Ranzato et al.,2008)*
- Regression *(Salakhutdinov and Hinton, 2008)*
- Dimensionality Reduction *(Hinton Salakhutdinov, 2006; Salakhutdinov and Hinton, 2007)*
- Modeling Textures *(Osindero and Hinton, 2008)*
- Information Retrieval *(Salakhutdinov and Hinton, 2007)*
- Robotics *(Hadsell et al., 2008)*
- Natural Language Processing *(Collobert and Weston, 2008; Weston et al., 2008)*
- Collaborative Filtering *(Salakhutdinov et al., 2007)*

Recent Deep Learning Highlights

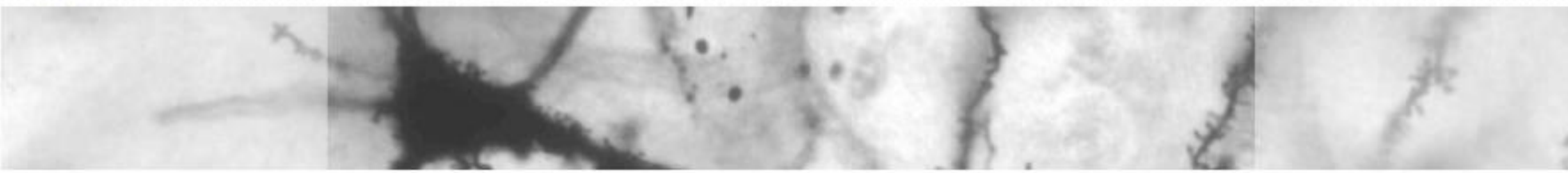
http://deeplearning.net

HOME | TUTORIALS | ABOUT | READING LIST | SOFTWARE LINKS | BLOG | DEMOS | DATASETS | EVENTS | BIBLIOGRAPHY

Deep Learning

... moving beyond shallow machine learning since 2006!

Posts Comments



Recent Posts

- LISA Lab Wins the Final Phase of UTLC Challenge
- New Challenge Announced
- Deep Learning Workshop at NIPS 2010
- New Events Page
- Deep Learning papers at ICML 2010

Tags

Meta

- Log in
- Entries RSS
- Comments RSS
- WordPress.org

Welcome to Deep Learning

Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence.

This website is intended to host a variety of resources and pointers to information about Deep Learning. In these pages you will find

- a **reading list**,
- links to **software**,
- **datasets**,
- a **discussion forum**,
- as well as **tutorials** and cool **demos**.

For the latest additions, including papers and software announcement, **be sure to visit the Blog section** of the website. **Contact us** if you have any comments or suggestions!

Last modified on April 26, 2010, at 9:45 am by ranzato

Pages

- About
- Bibliography
- Blog
- Datasets
- Demos
- Events
- Reading List
- Software links
- Tutorials

Links

- Discussion forum

Recent Deep Learning Highlights

- Google Goggles uses ***Stacked Sparse Auto Encoders*** (Hartmut Neven @ ICML 2011)
- The monograph or review paper ***Learning Deep Architectures for AI*** (Foundations & Trends in Machine Learning, 2009).
- ***Exploring Strategies for Training Deep Neural Networks***, Hugo Larochelle, Yoshua Bengio, Jerome Louradour and Pascal Lamblin in: The Journal of Machine Learning Research, pages 1-40, 2009.
- The LISA publications database contains ***a deep architectures category***. http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/ReadingOnDeepNetworks
- Deep Machine Learning – ***A New Frontier in Artificial Intelligence Research*** – a survey paper by Itamar Arel, Derek C. Rose, and Thomas P. Karnowski.

Thank You