Deep Learning

Conceptual Understanding & Application Variants

ben.hur@daumkakao.com

The aim of this slide is at conceptual understanding of deep learning, rather than implementing it for practical applications. That is, this is more for paper readers, less for system developers.
Note that, some variables are not uniquely defined nor consistently used for convenience.
Also, variable/constant and vector/matrix are not clearly distinct.
More importantly, some descriptions may be incorrect due to lack of my ability.

Deep Learning (DL)

is, in concise,

a **deep** and wide neural **network**.

That is half-true.

Deep Learning is difficult

because of

lack of understanding

about

— artificial neural network (ANN)

— unfamiliar applications.

DL

- ANN
  - Signal transmission between neurons
  - Graphical model (Belief network)
  - Linear regression / logistic regression
    - Weight, bias & activation
    - (Feed-forward) Multi-layer perceptron (MLP)
- Optimization
  - Back-propagation
  - (Stochastic) Gradient-decent

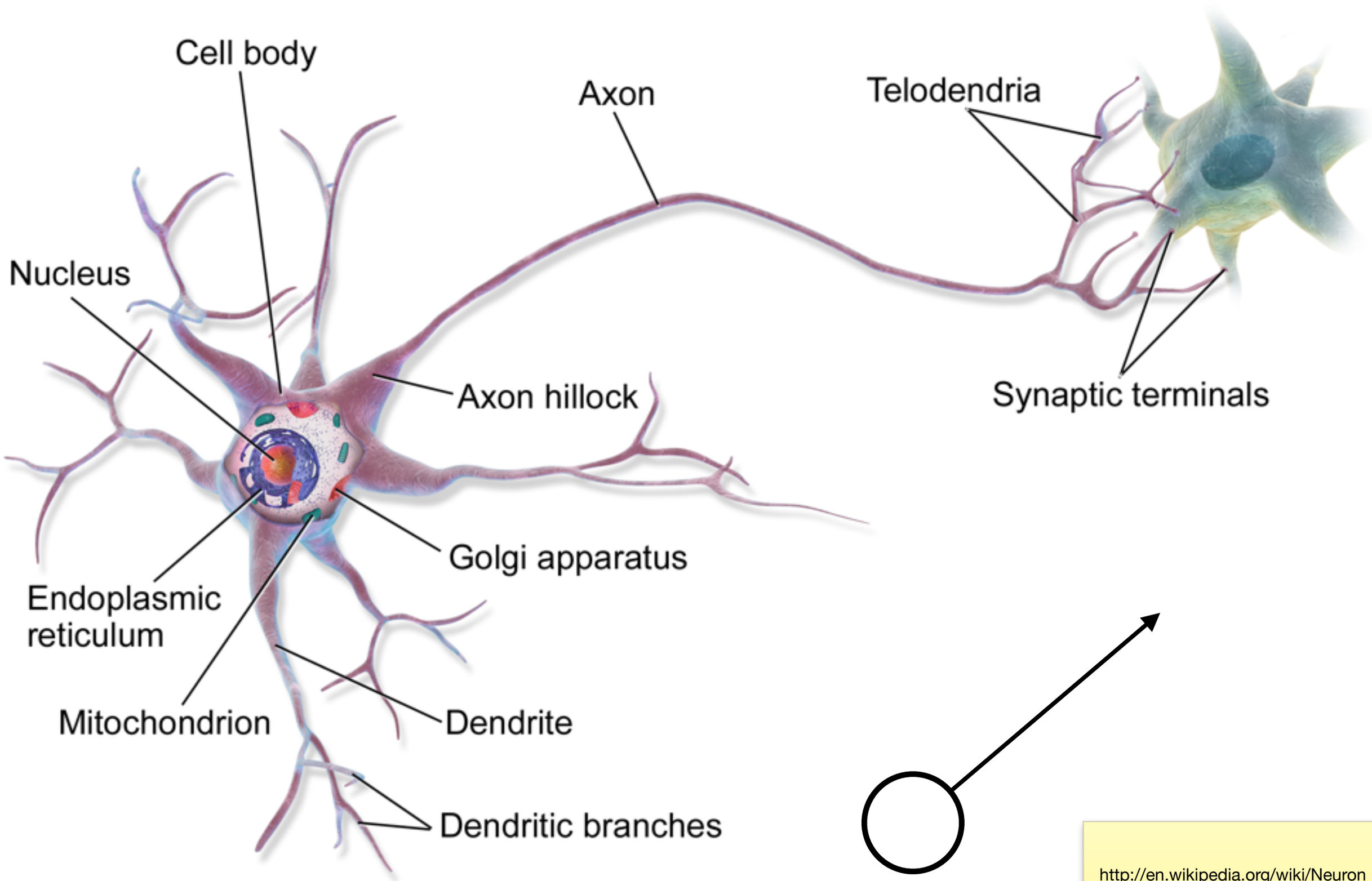
DL

- Applications
  - Speech recognition (Audio)
  - Natural language processing (Text)
  - Information retrieval (Text)
  - Image recognition (Vision)
- & yours (+)

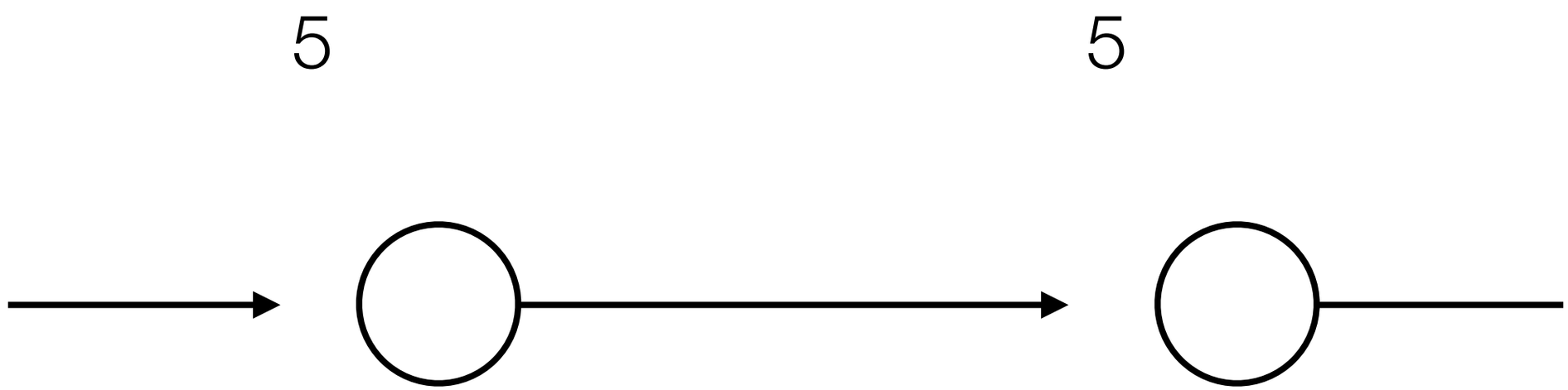
Artificial Neural Network (ANN)

ANN is a nested ensemble of [logistic] **regressions**.

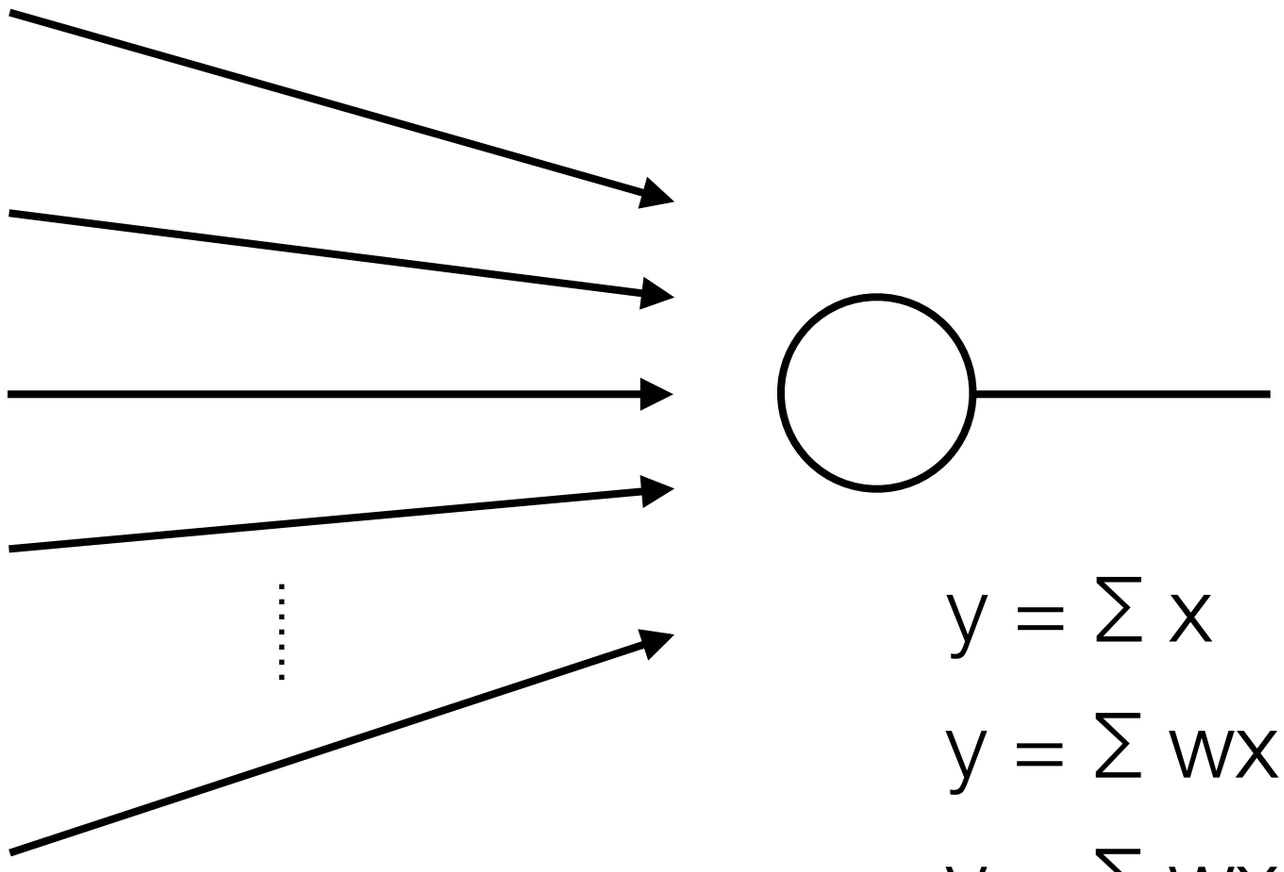
Perceptron & Message Passing



http://en.wikipedia.org/wiki/Neuron



Message passing

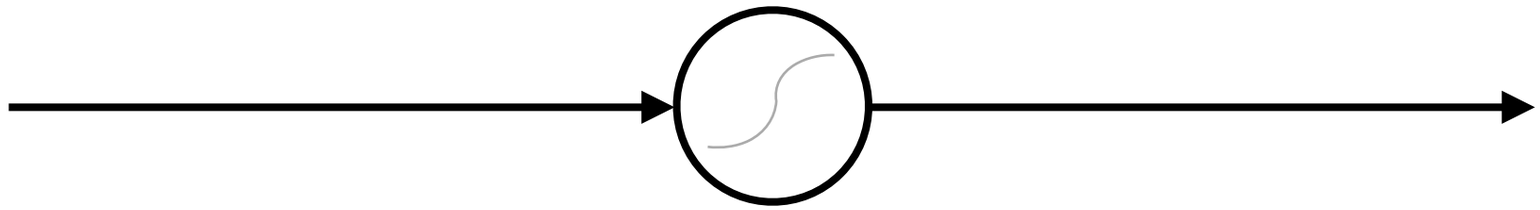


$$y = \Sigma\, x$$

$$y = \Sigma\, wx$$

$$y = \Sigma\, wx + b$$

Linear regression



Σwx

Σwx?

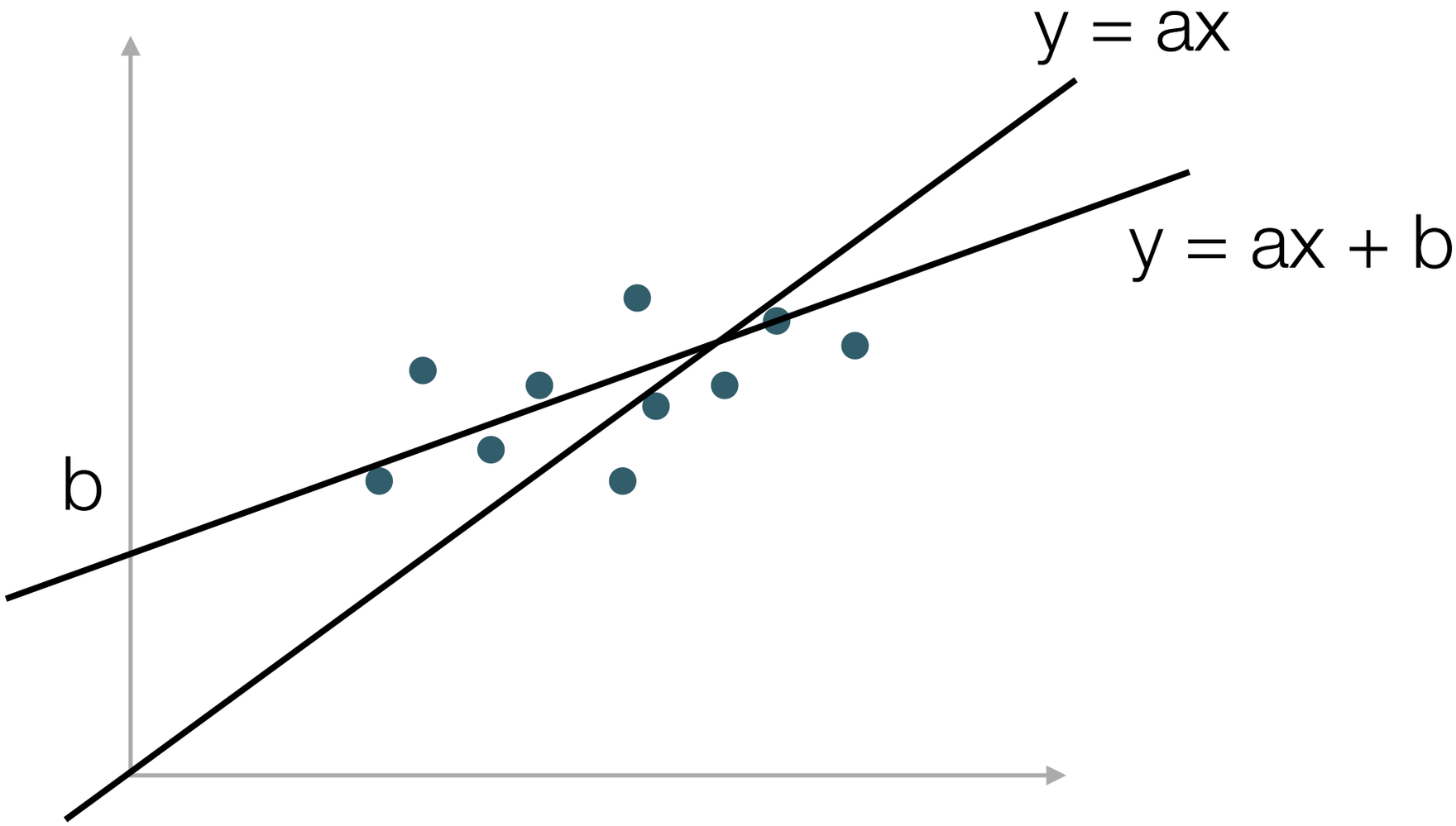
y = sigm(Σwx + b)

Logistic regression

Activation function

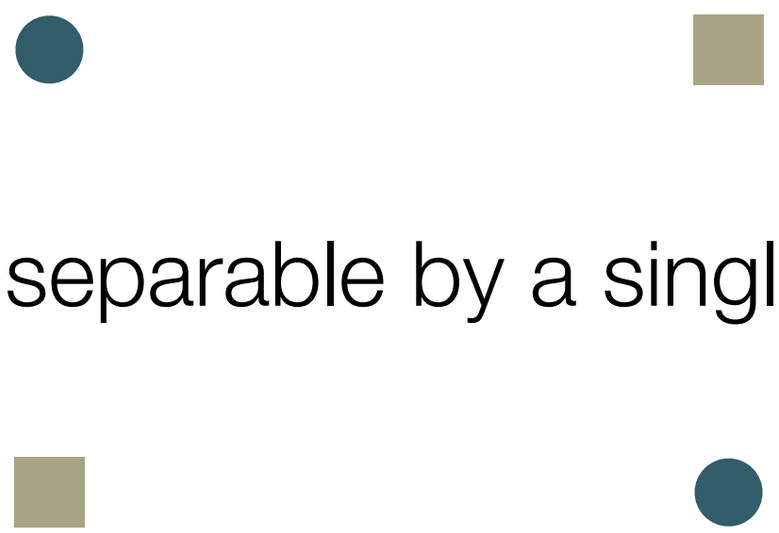
- Linear $g(a) = a$
- Sigmoid $g(a) = sigm(a) = 1 / (1 + exp(-a))$
- Tanh $g(a) = tanh(a) = (exp(a) - exp(-a)) / (exp(a) + exp(-a))$
- Rectified linear $g(a) = reclin(a) = max(0, a)$
- Step
- Gaussian
- Softmax, usually for the output layer

Bias controls activation.



Multi-Layer Perceptron (MLP)

Linearly non-separable by a single perceptron



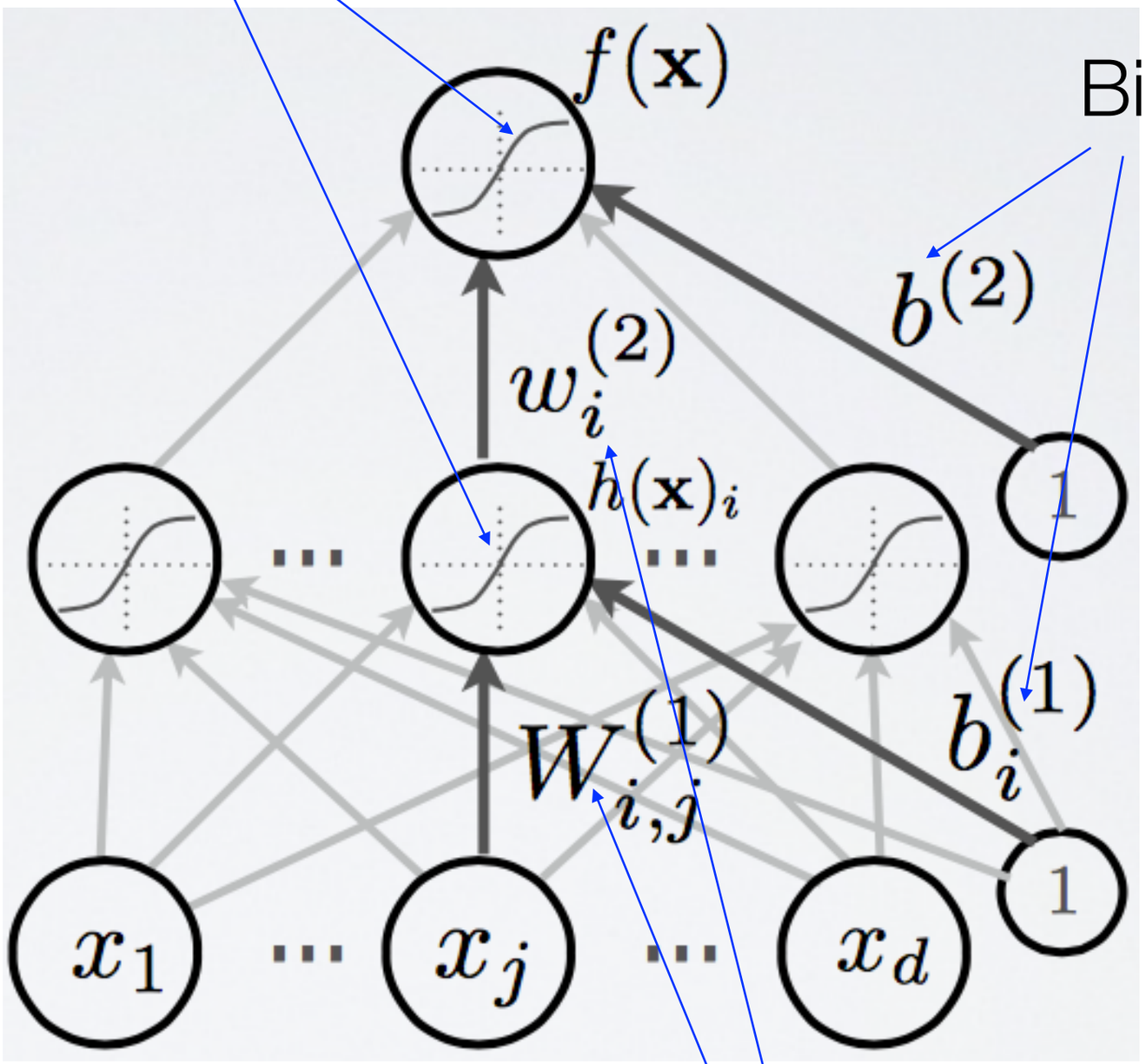
Activation function

Output layer

Bias

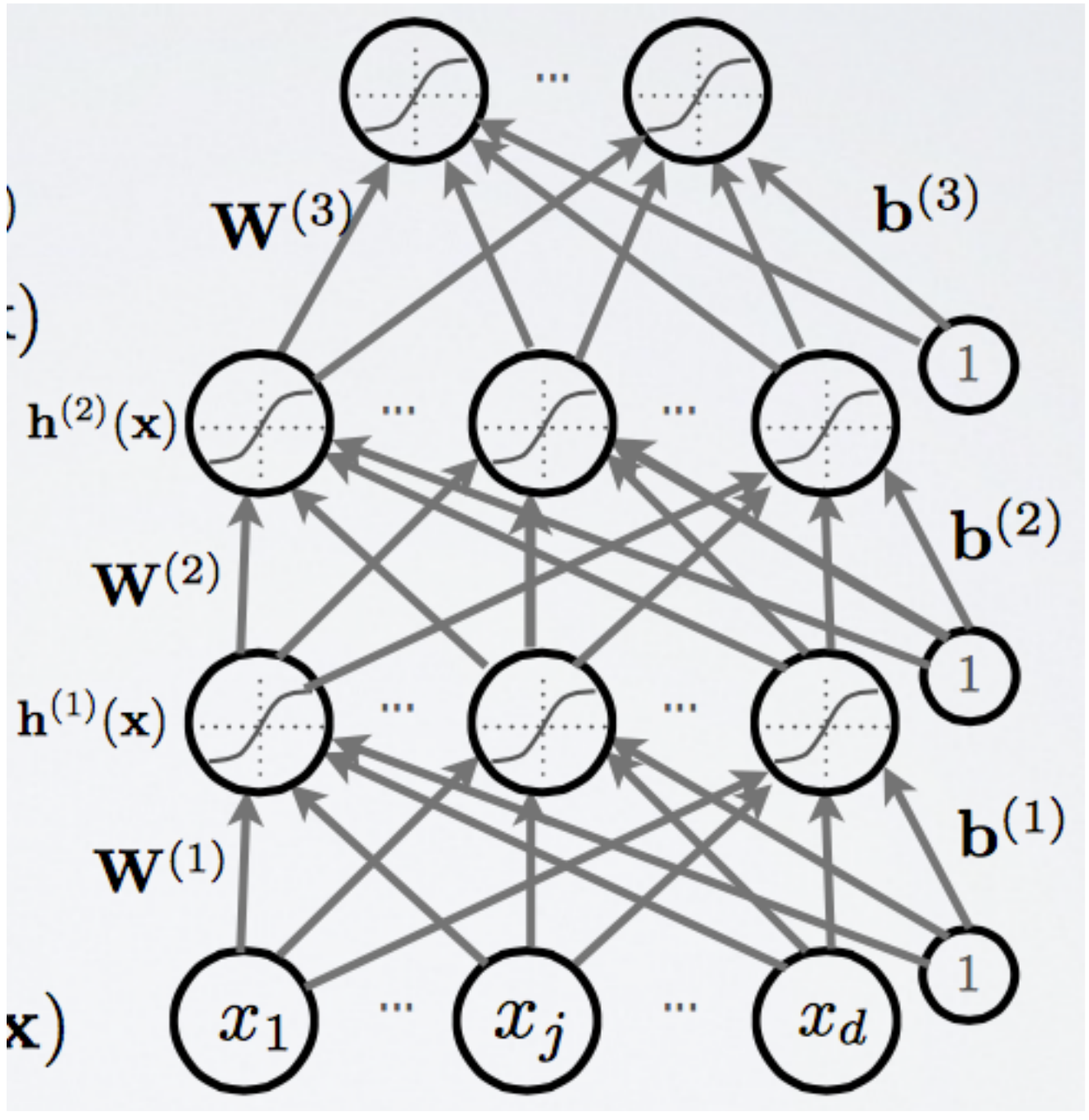
Hidden layer

Input layer



Interconnecting weight

http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html



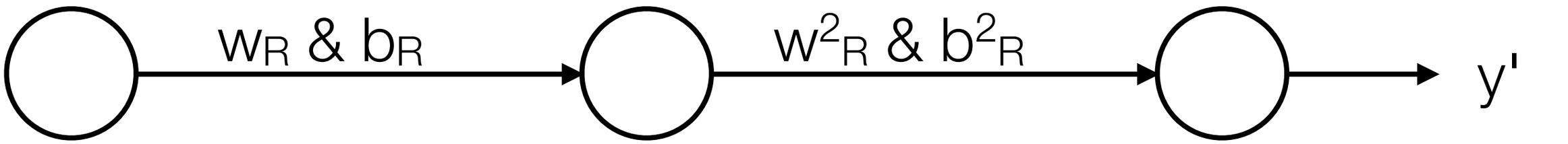
http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html

Back-Propagation & Optimization

{x, y}

$h=a(wx+b)$

$y'=a^2(w^2h+b^2)$



w & b

w^2 & b^2

$l = y - y'$

—> feed-forward
<— back-propagation

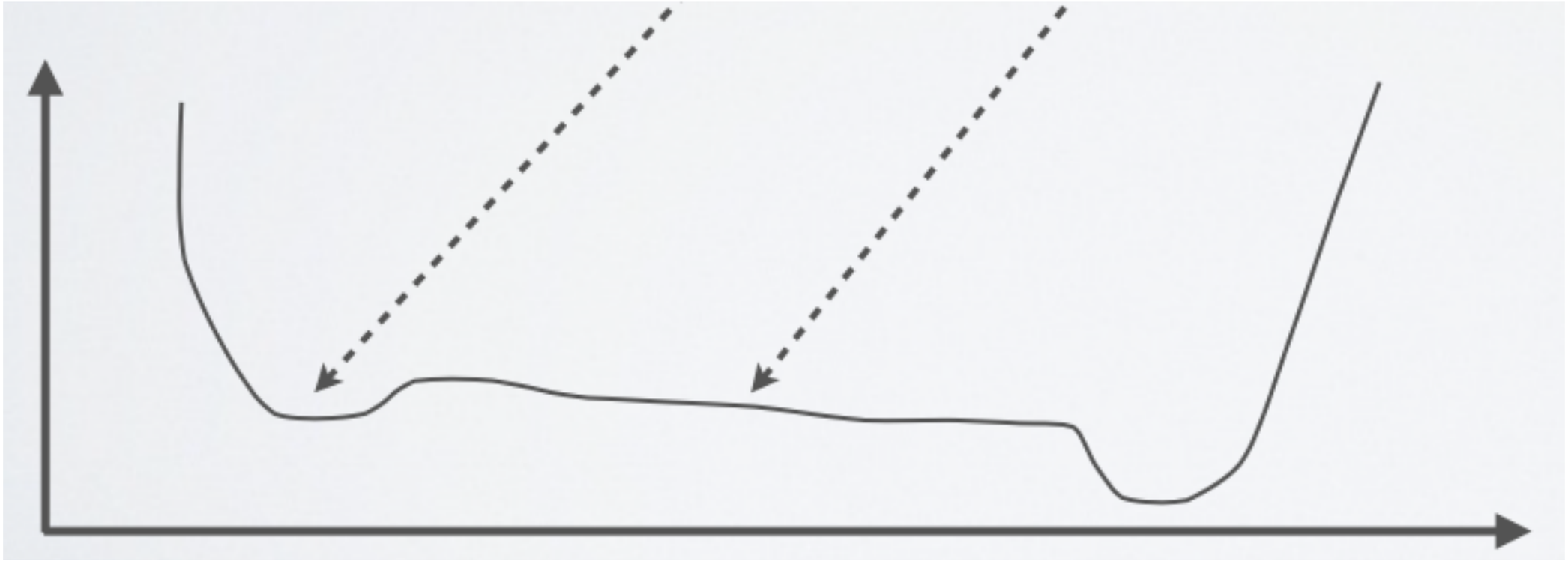
Back-propagation

Reinforced through epoch

ANN is a reinforcement learning.

Local minimum

Plateaus



http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html

Stochastic Gradient Descent

The slide excludes details.
See reference tutorial for that.

ANN is a good approximator of any functions,
if well-designed and trained.

Deep Learning

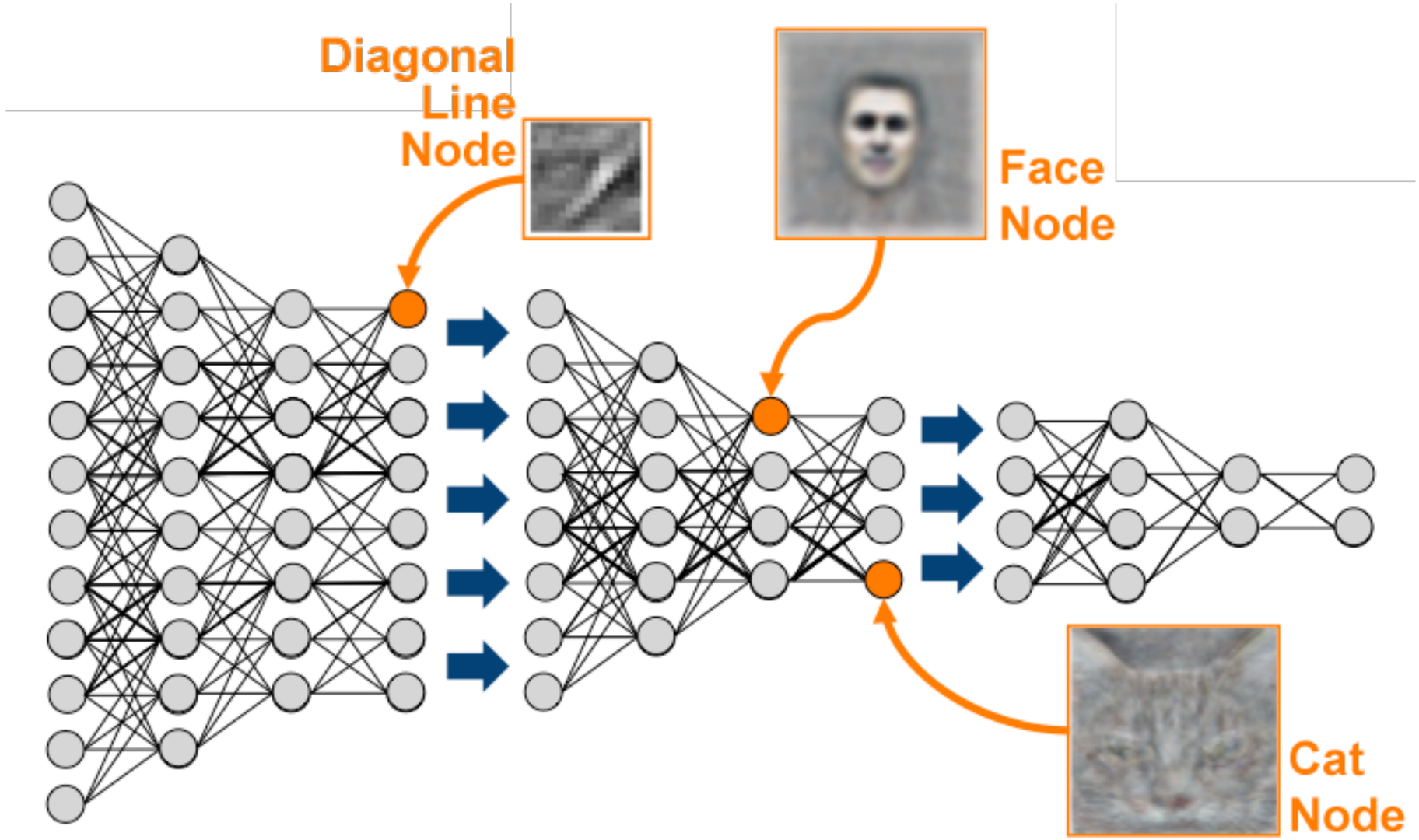
DL (DNN) is a **deep** and **wide** neural network.

many (2+) hidden layers

many input/hidden nodes

WIDE

Large matrix

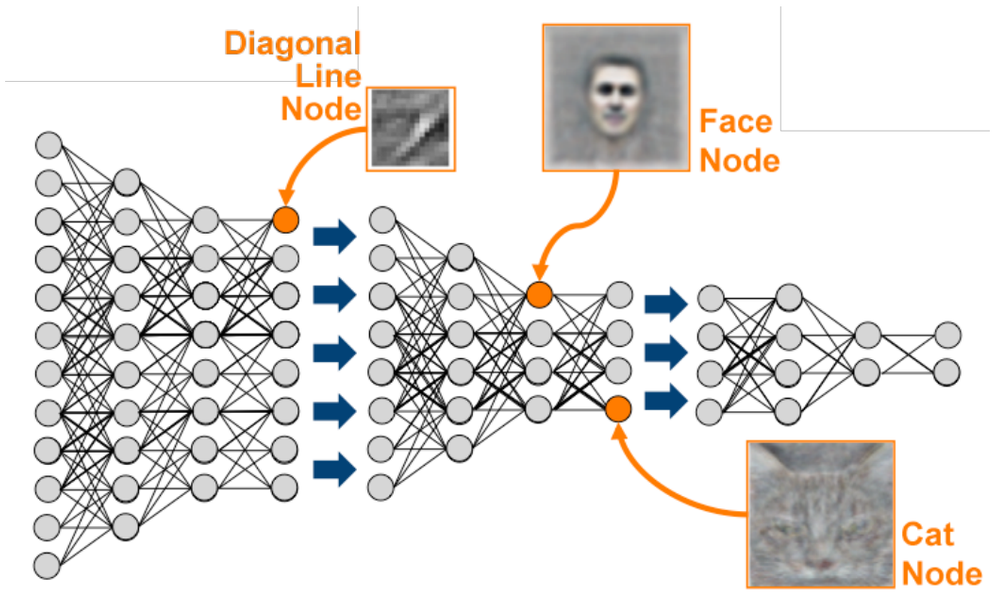


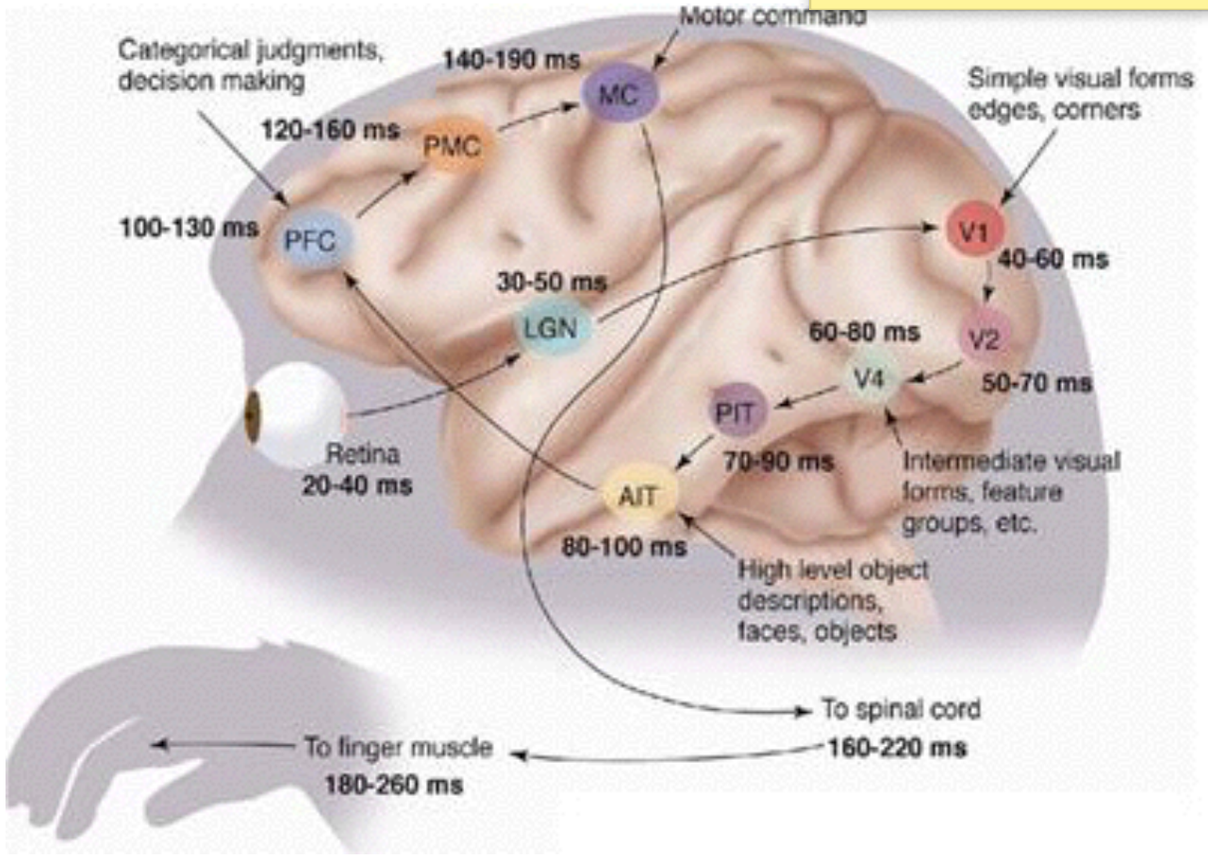
DEEP

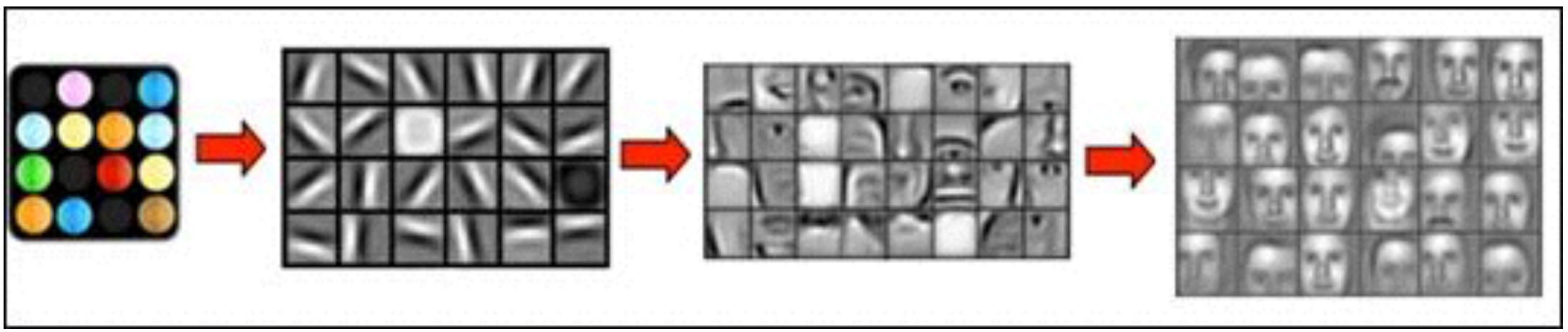
Many matrix

Image from googling

Brain image from Simon Thorpe
Other from googling







Many large connection matrices (W)

— Computational burden (GPU)

— Insufficient labeled data (Big Data)

— Under-fitting (Optimization)

— Over-fitting (Regularization)

Unsupervised makes DL success.

Unsupervised pre-training

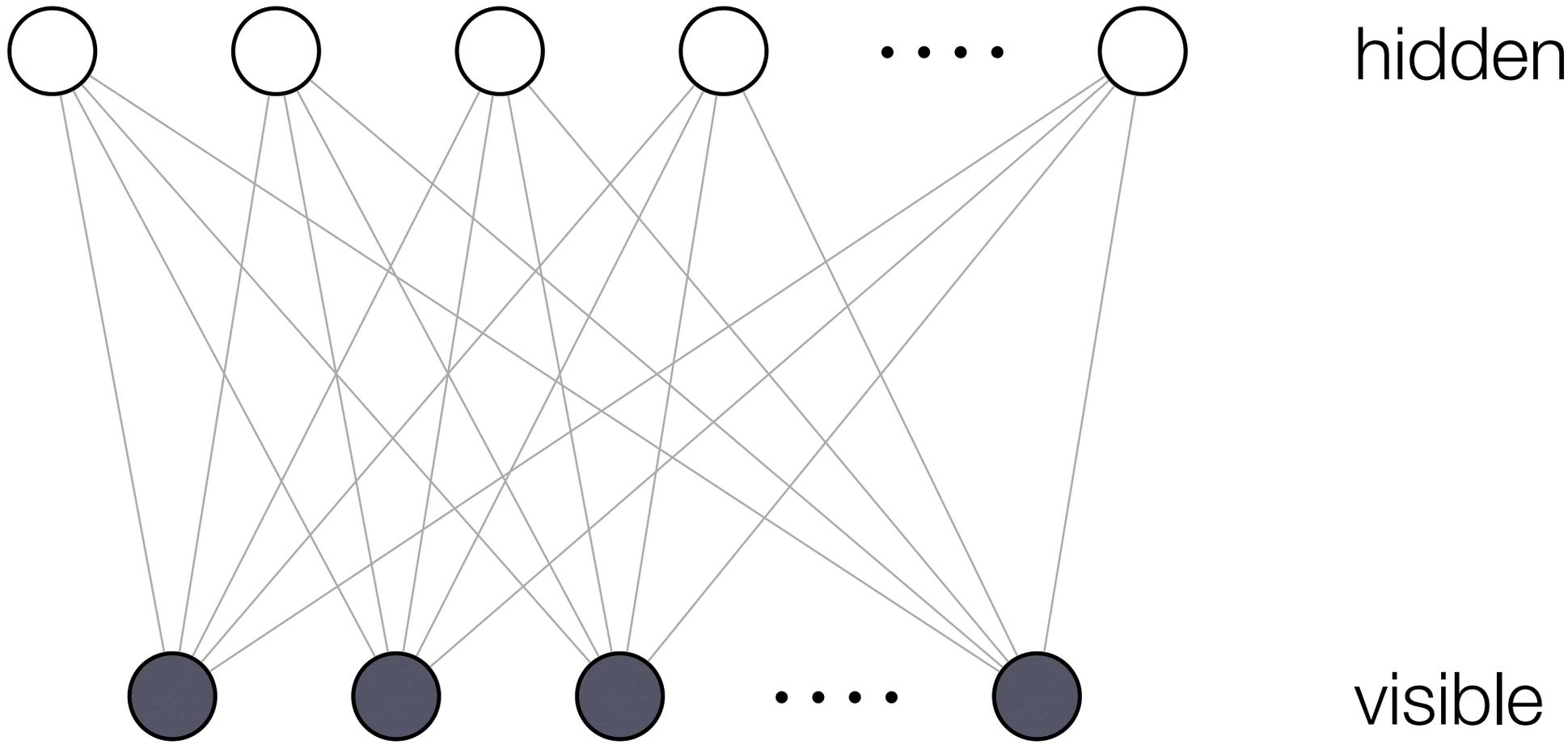
- Initialize W matrices in an unsupervised manner
  - Restricted Boltzmann Machine (RBM)
  - Auto-encoder
- Pre-train in a layer-wise (stacking) fashion
- Fine-tune in a supervised manner

Unsupervised representation learning

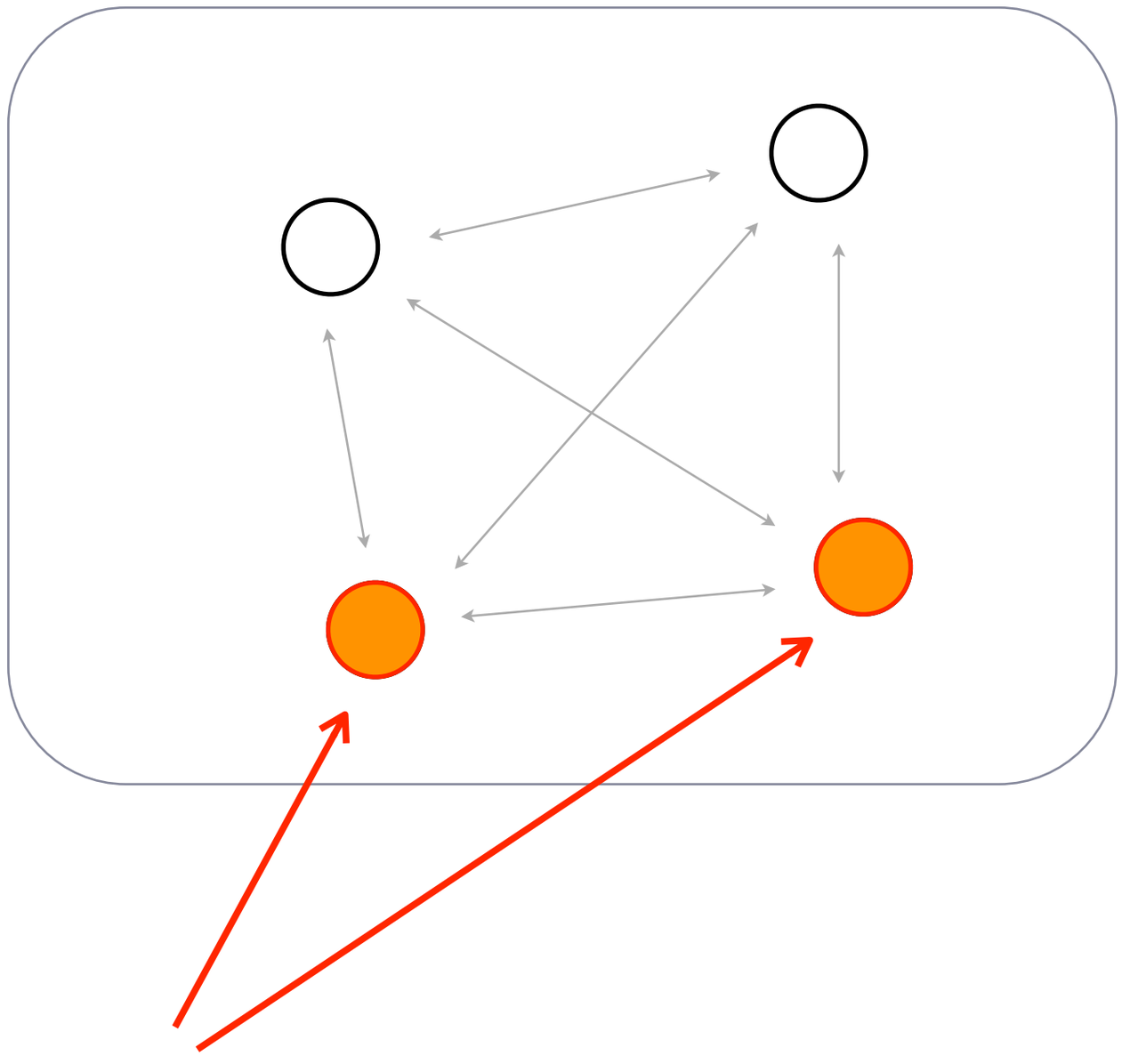
- Construct a feature space
  - DBM / DBN
  - Auto-encoder
  - Sparse coding
- Feed new features to a shallow ANN as input

cf, PCA + regression

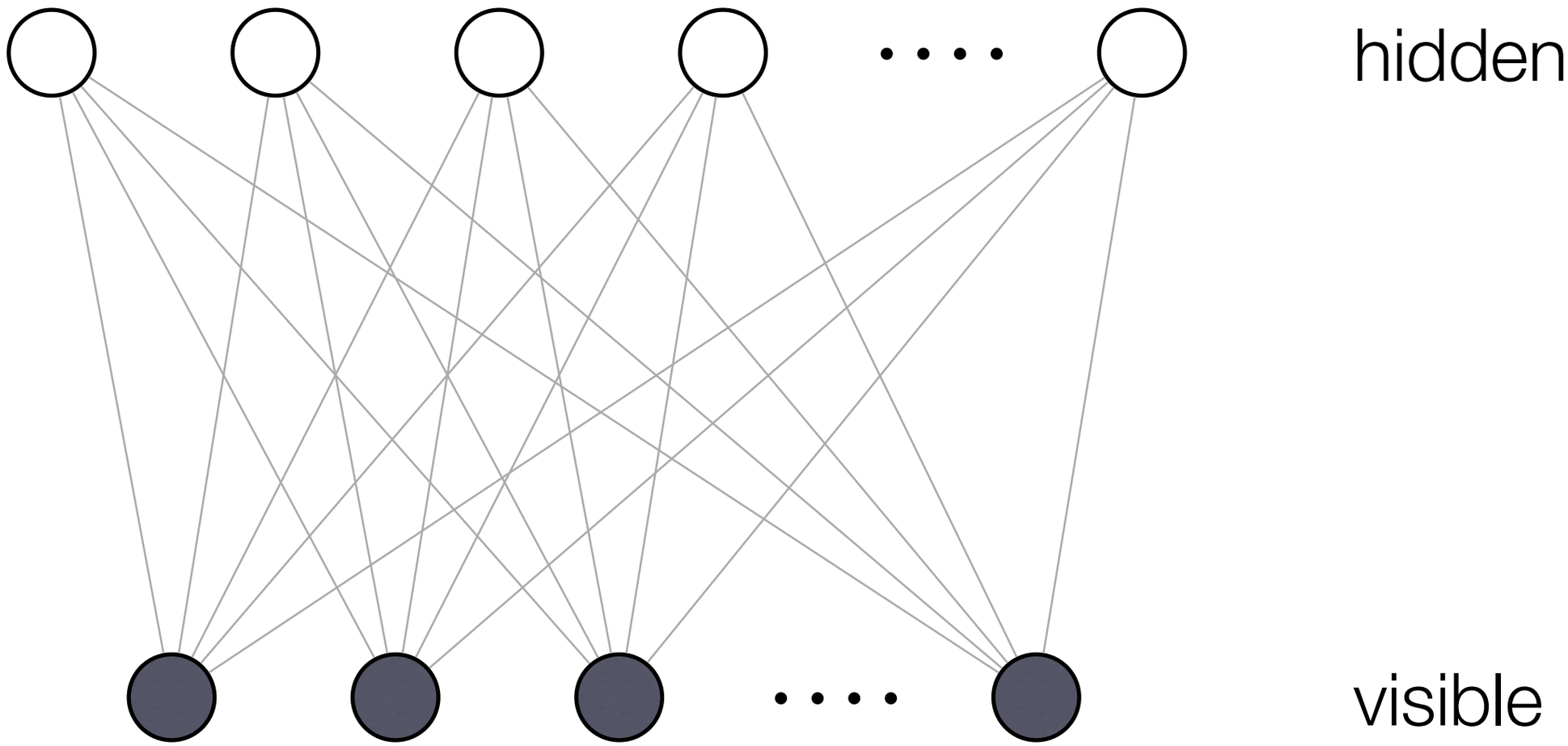
Network		MNIST-small	MNIST-rotation
Type	Depth	classif. test error	classif. test error
Neural network (random initialization, + fine-tuning)	1	**4.14** % ± 0.17	15.22 % ± 0.31
	2	**4.03** % ± 0.17	**10.63** % ± 0.27
	3	**4.24** % ± 0.18	11.98 % ± 0.28
	4	4.47 % ± 0.18	11.73 % ± 0.29
SAA network (autoassociator learning + fine-tuning)	1	3.87 % ± 0.17	11.43% ± 0.28
	2	**3.38** % ± 0.16	9.88 % ± 0.26
	3	**3.37** % ± 0.16	**9.22** % ± 0.25
	4	**3.39** % ± 0.16	**9.20** % ± 0.25
SRBM network (CD-1 learning + fine-tuning)	1	3.17 % ± 0.15	10.47 % ± 0.27
	2	**2.74** % ± 0.14	9.54 % ± 0.26
	3	**2.71** % ± 0.14	**8.80** % ± 0.25
	4	**2.72** % ± 0.14	**8.83** % ± 0.24

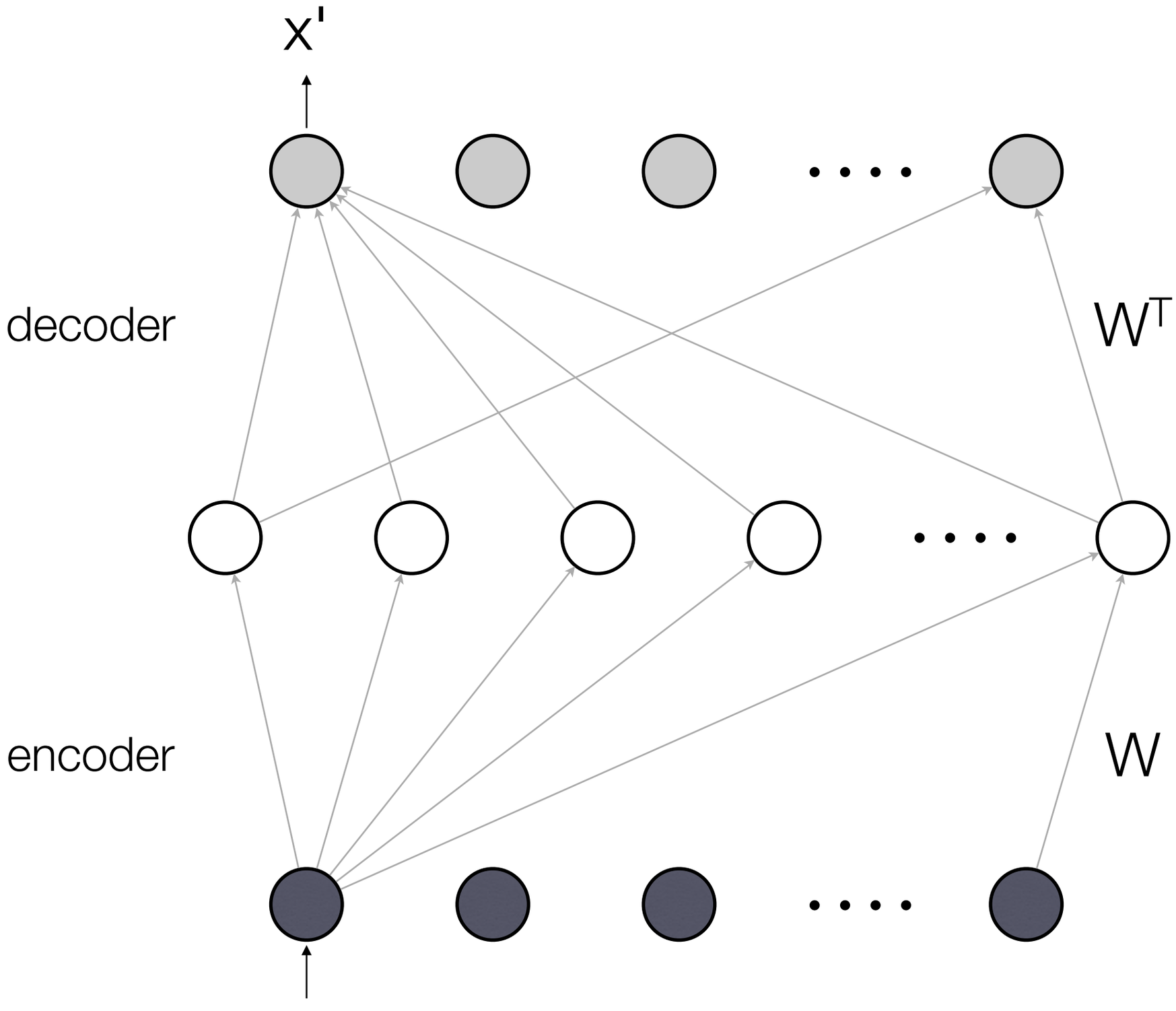


Restricted Boltzmann Machine (RBM)



Boltzmann Machine Example


hidden
visible



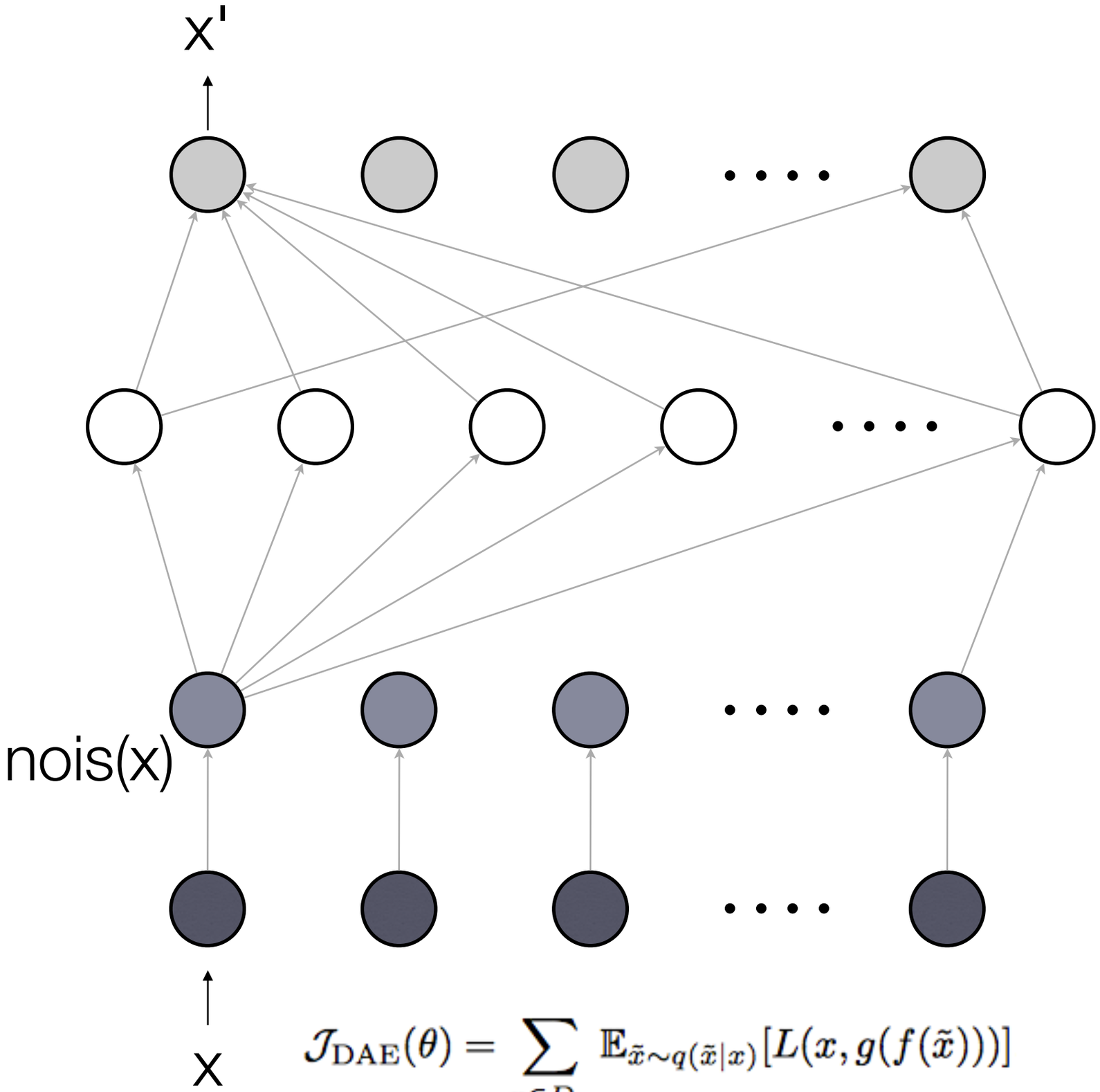
$$\mathcal{J}_{\mathrm{AE}}(\theta) = \sum_{x \in D_n} L(x, g(f(x)))$$

Auto-encoder

if k <= n (single & linear), then AE becomes PCA.

otherwise, AE can be arbitrary (over-complete).

Hidden & Output layers plays the role of semantic feature space in general
- compresses & latent feature (under-complete)
- expanded & primitive feature (over-complete)



$$\mathcal{J}_{\mathrm{DAE}}(\theta) = \sum_{x \in D_n} \mathbb{E}_{\tilde{x} \sim q(\tilde{x}|x)}[L(x, g(f(\tilde{x})))]$$

Denoising auto-encoder

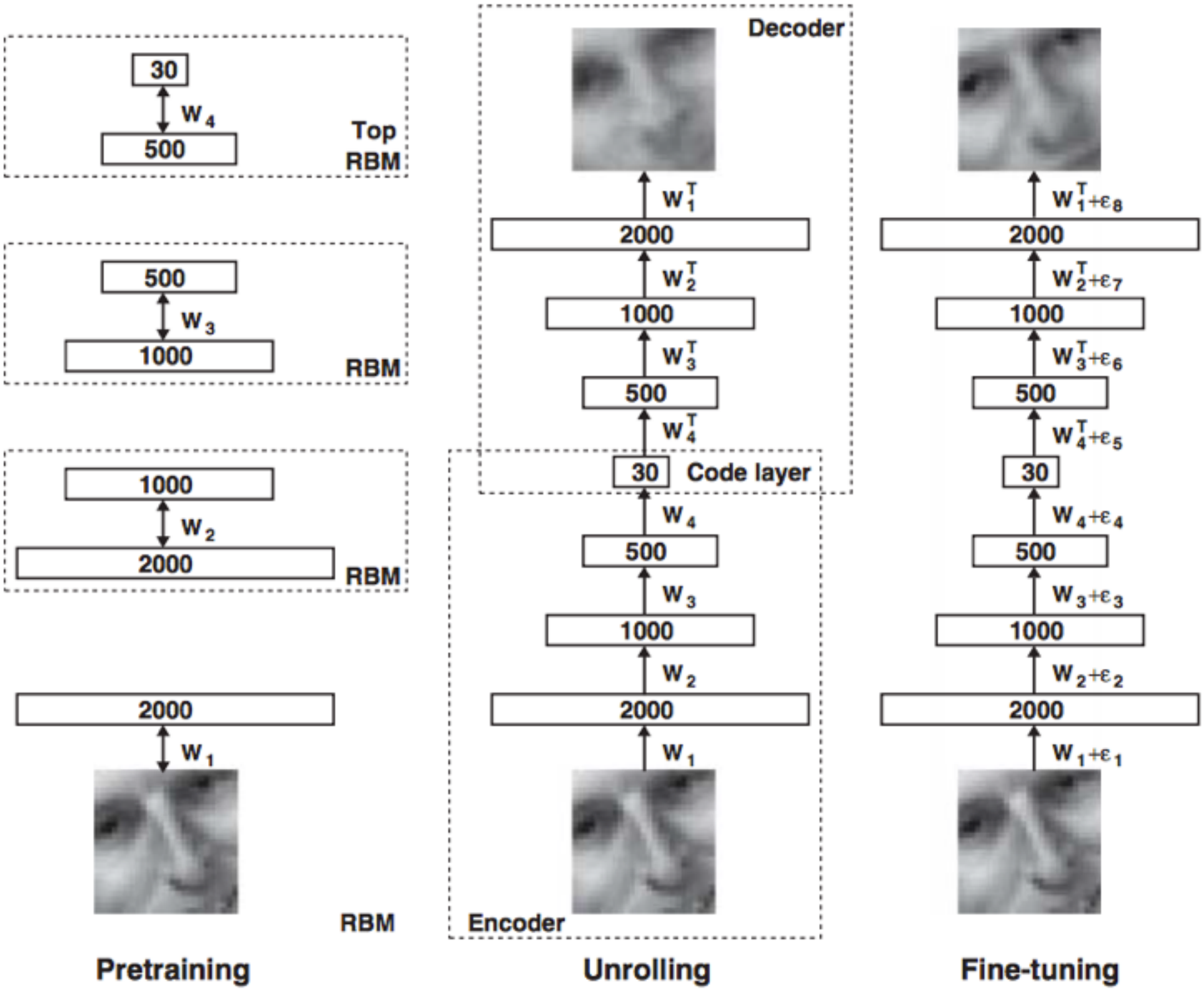
$$\mathcal{J}_{\text{AE+wd}}(\theta) = \left(\sum_{x \in D_n} L(x, g(f(x))) \right) + \lambda \sum_{ij} W_{ij}^2$$

Regularized auto-encoder

Same as Ridge (or similar to Lasso) Regression

$$\mathcal{J}_{\text{CAE}}(\theta) = \sum_{x \in D_n} \left(L(x, g(f(x))) + \lambda \| J_f(x) \|_F^2 \right)$$

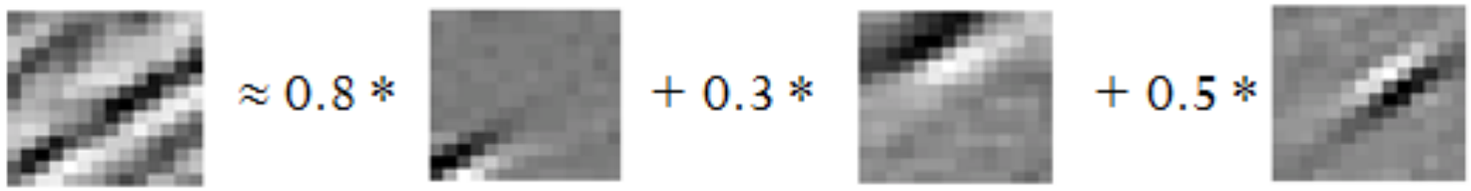
Contractive auto-encoder



Hinton and Salakhutdinov, Science, 2006

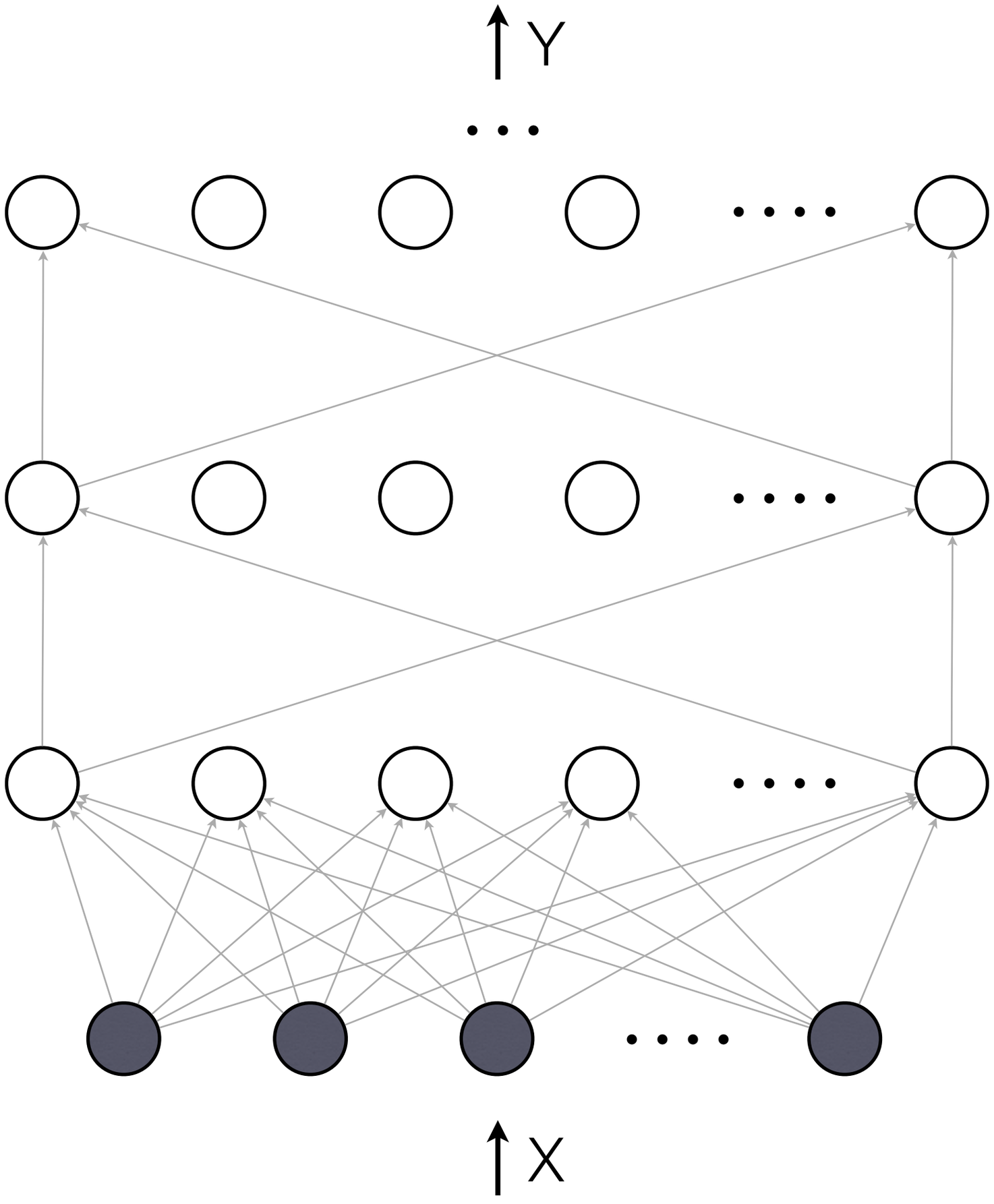
Deep auto-encoder

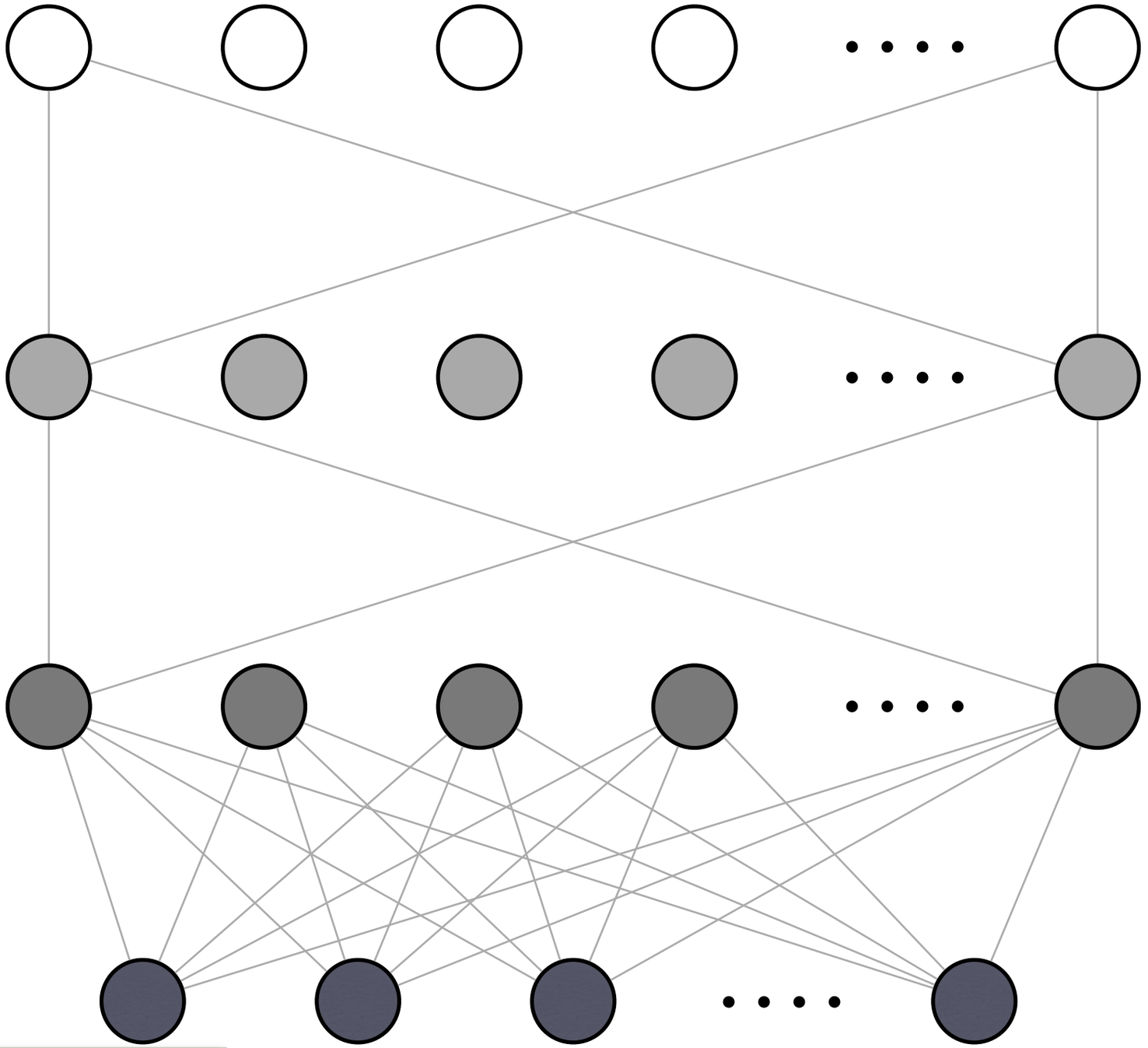
$$\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i$$ (k > n, over-complete)



Represent x_i as: $\mathbf{a}_i$ = [0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …]

Sparse coding


Y
X



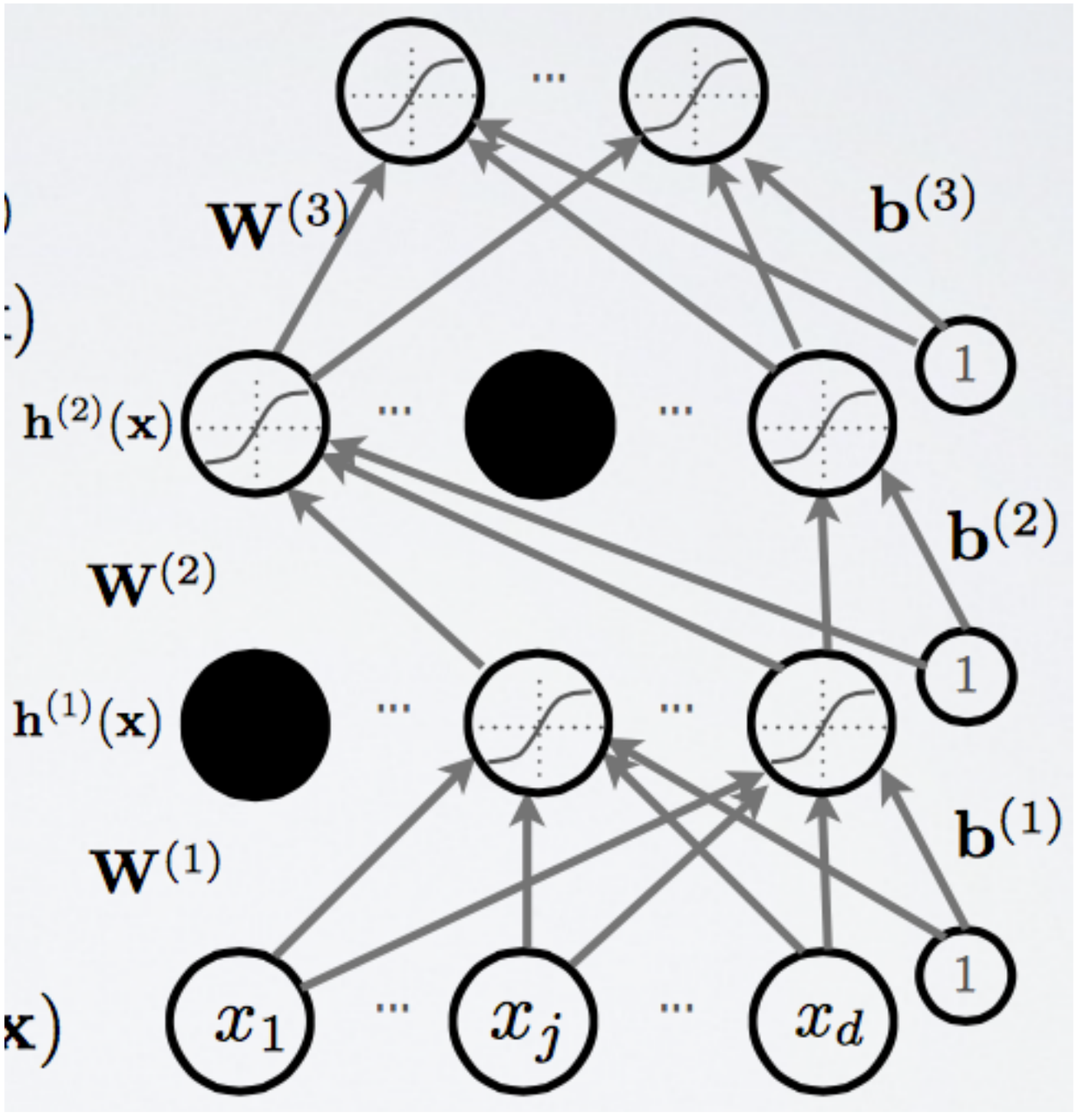
Stacking
Training in the order of RBM1, RBM2, RBM3, …

Representation Learning + Artificial Neural Network

- Sparse coding
- Auto-encoder
- DBN / DBM

Stochastic **Dropout** Training

for preventing over-fitting



http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html

Deep Architecture

Deep architecture is an ensemble of shallow networks.

Key Deep Architectures

— Deep Neural Network (DNN)

— Deep Belief Network (DBN)

— Deep Boltzmann Machine (DBM)

— Recurrent Neural Network (RNN)

— Convolution Neural Network (CNN)

— Multi-modal/multi-tasking

— Deep Stacking Network (DSN)

Applications

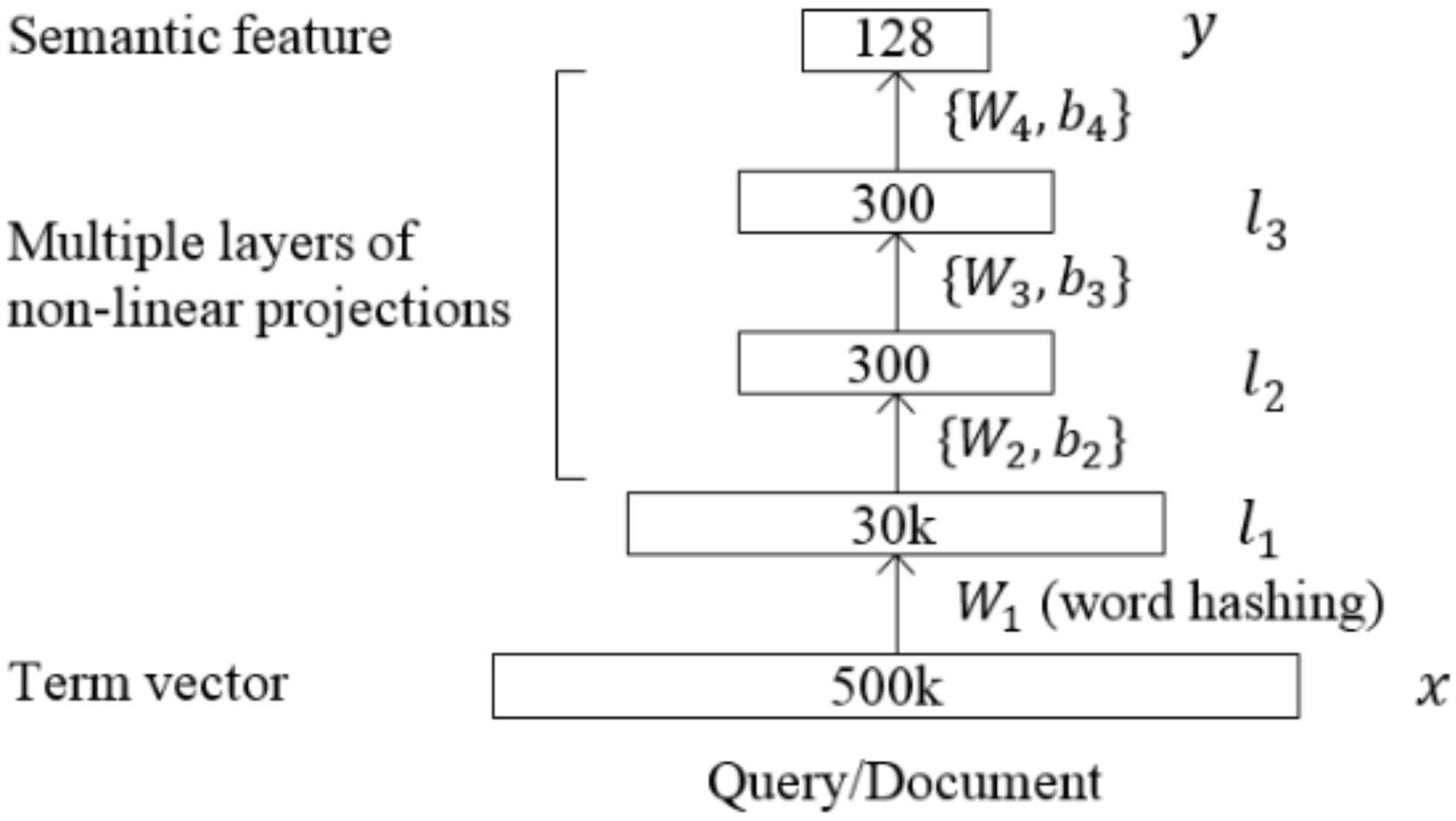
Applications

- Speech recognition
- Natural language processing
- Image recognition
- Information retrieval

Application Characteristic
- Non-numeric data
- Large dimension (curse of dimensionality)

Natural language processing

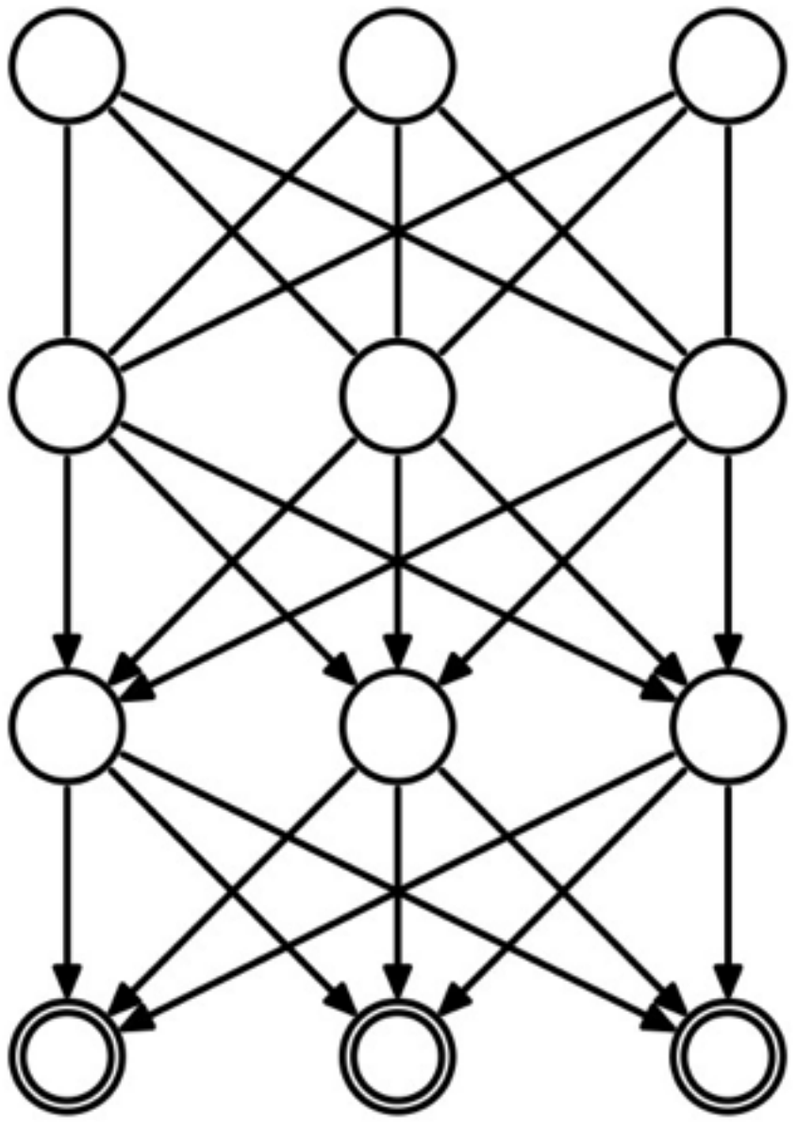
- Word embedding
- Language modeling
- Machine translation
- Part-of-speech (POS) tagging
- Named entity recognition
- Sentiment analysis
- Paraphrase detection
- ...



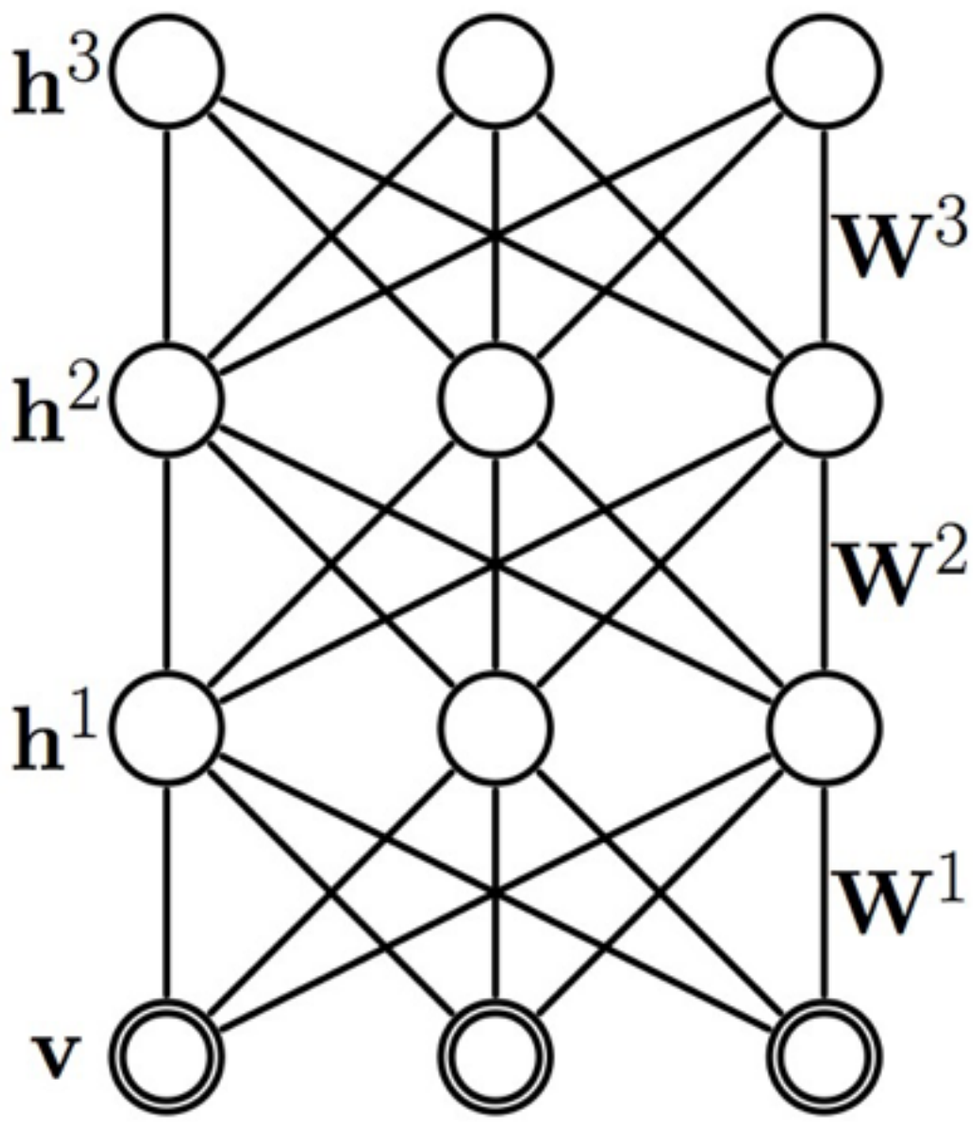
BM25(Q, D) vs Corr(sim(Q, D), CTR)

Deep-structured semantic modeling (DSSM)

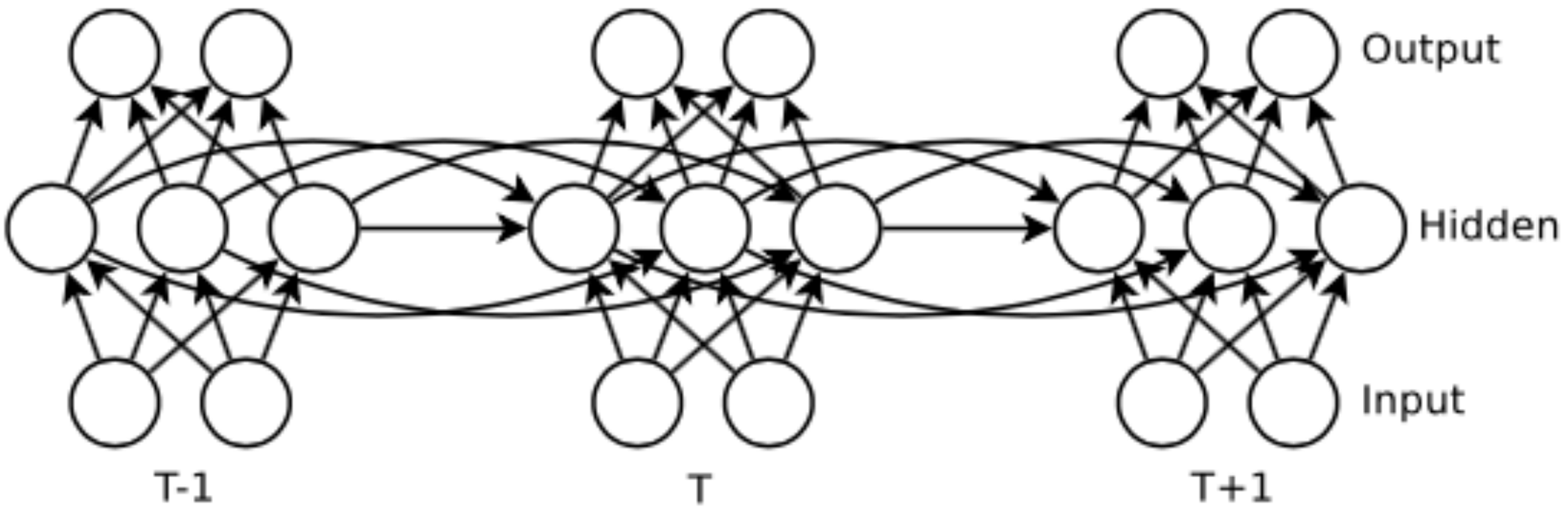
Deep Belief Network



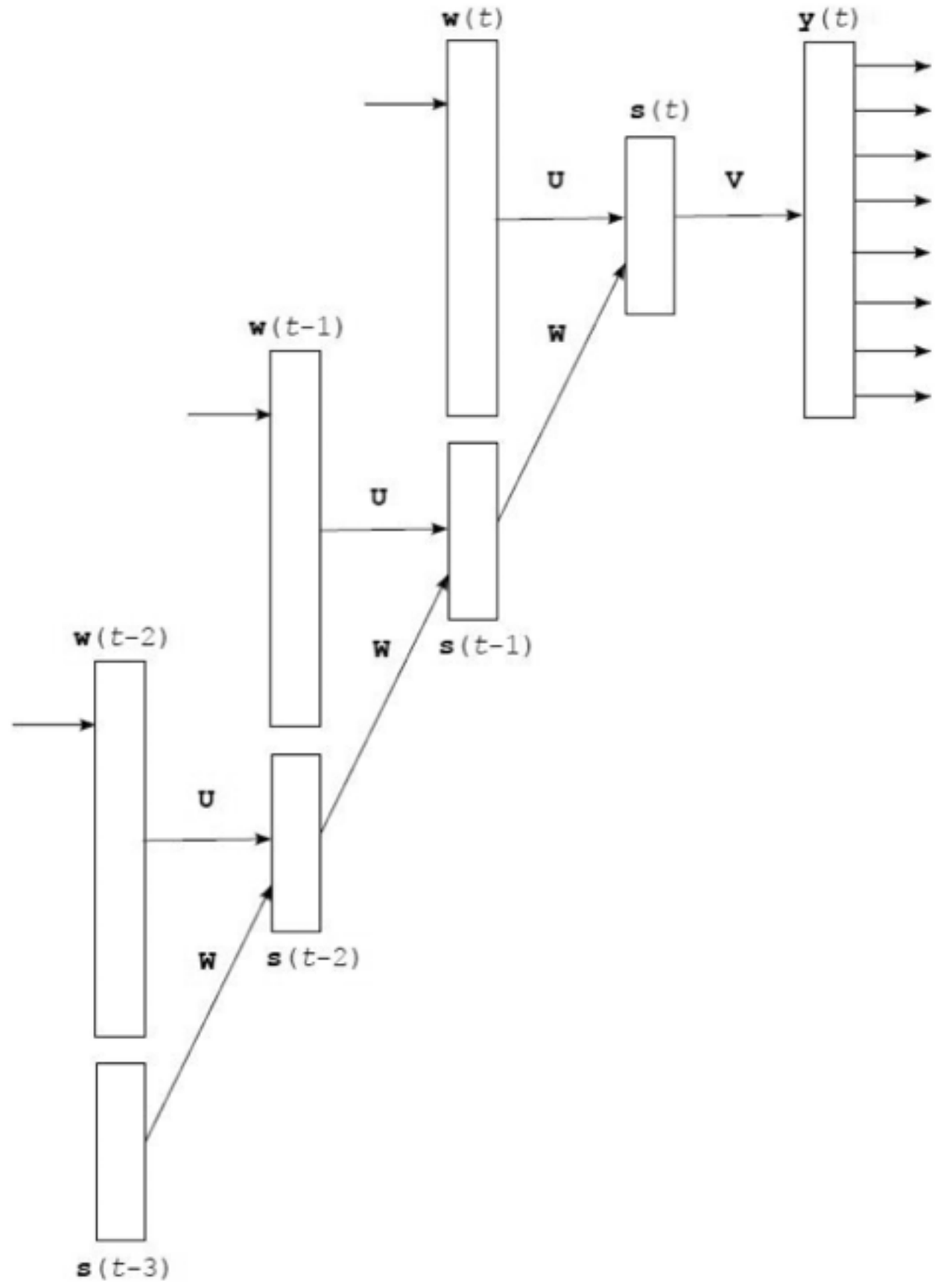
Deep Boltzmann Machine



DBN & DBM

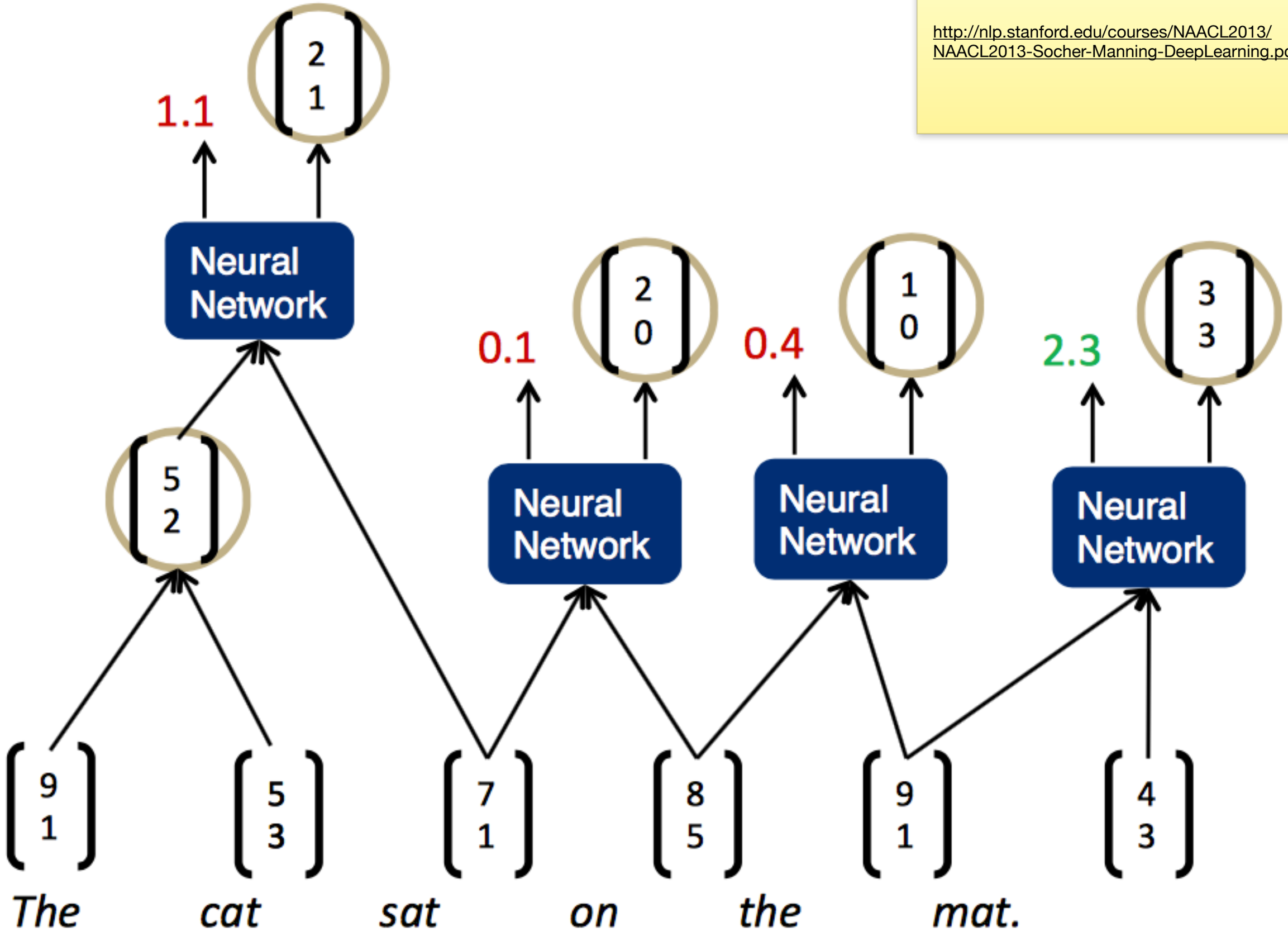


Recurrent Neural Network (RNN)

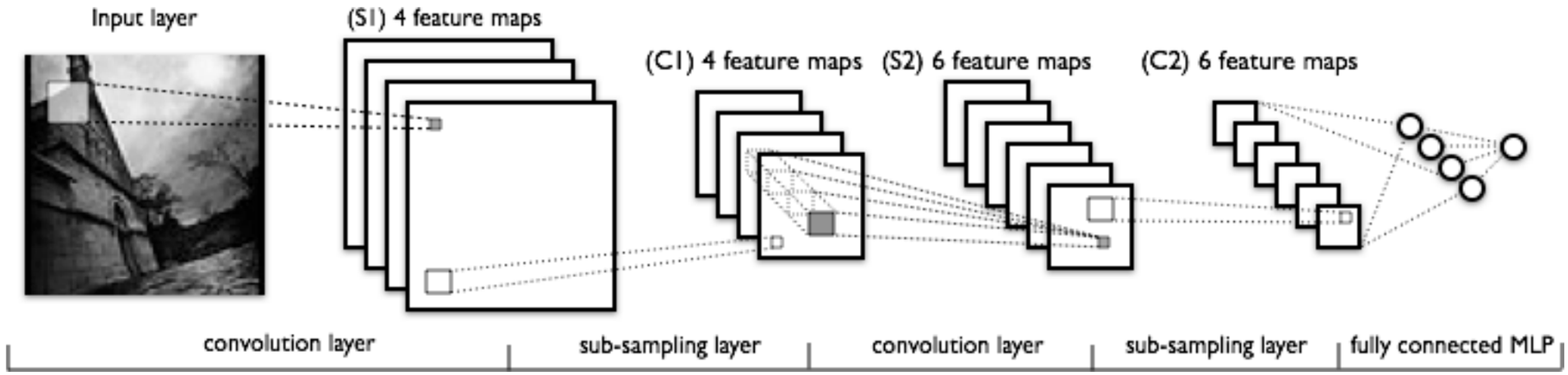


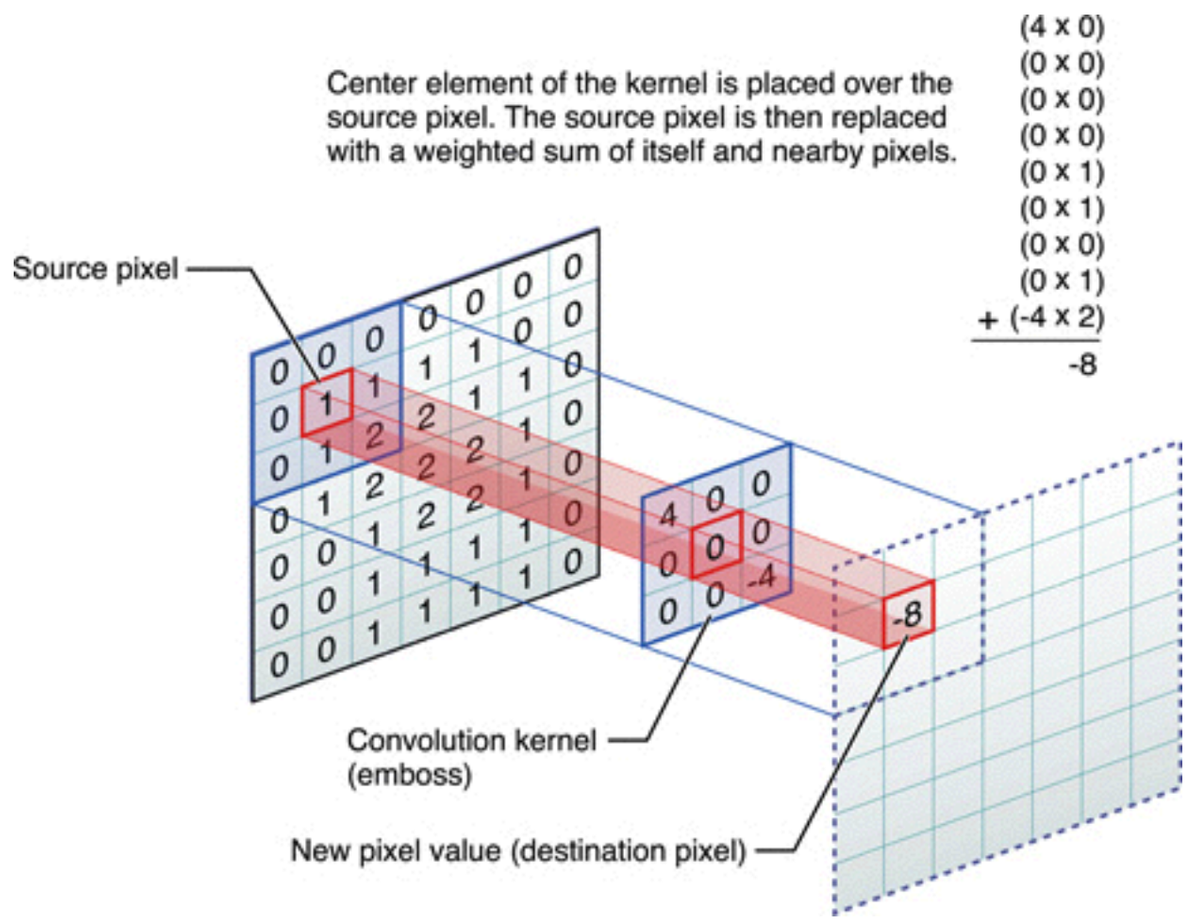
Recurrent Neural Network (RNN)

http://nlp.stanford.edu/courses/NAACL2013/NAACL2013-Socher-Manning-DeepLearning.pdf

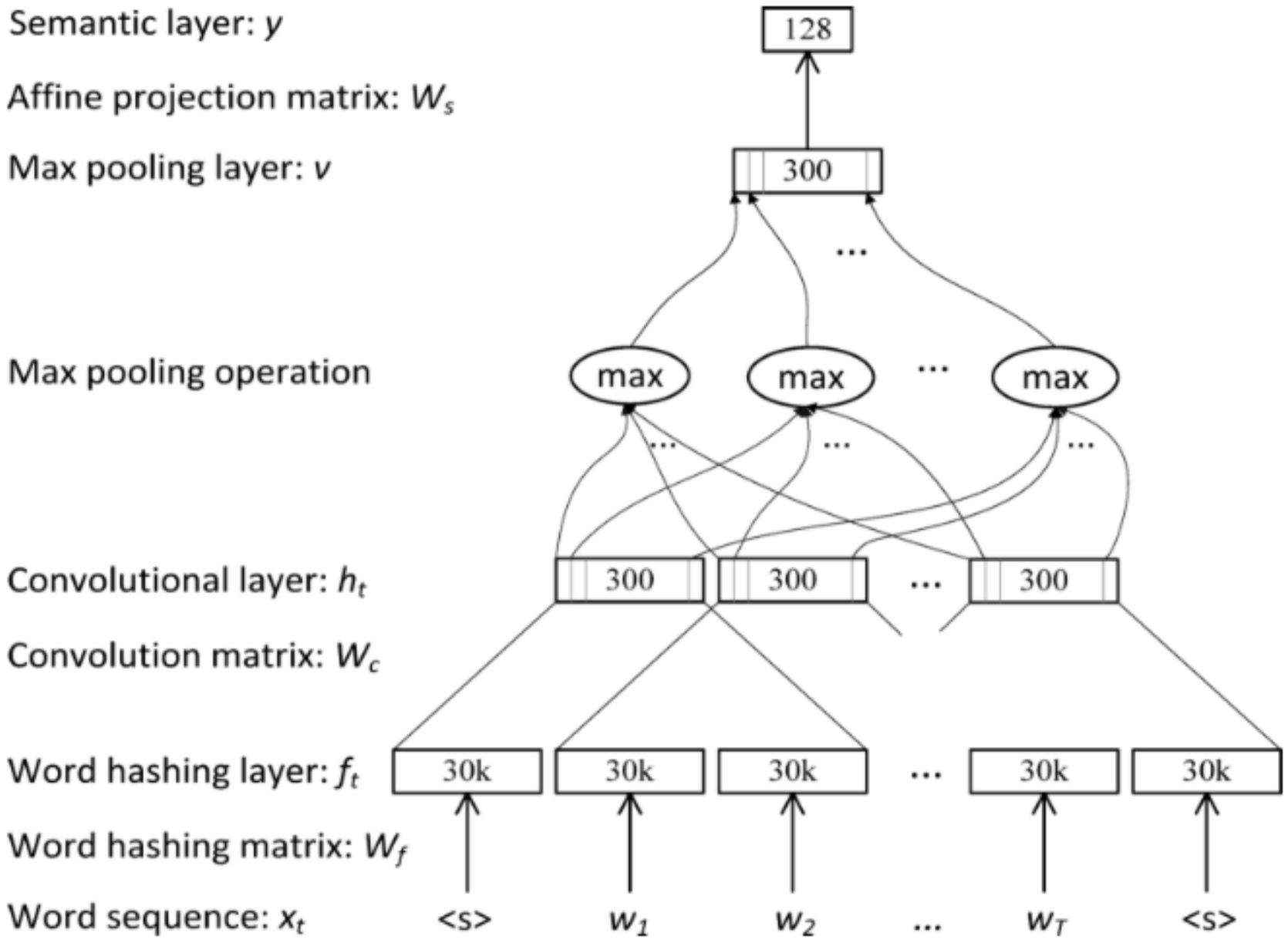


Recursive Neural Network

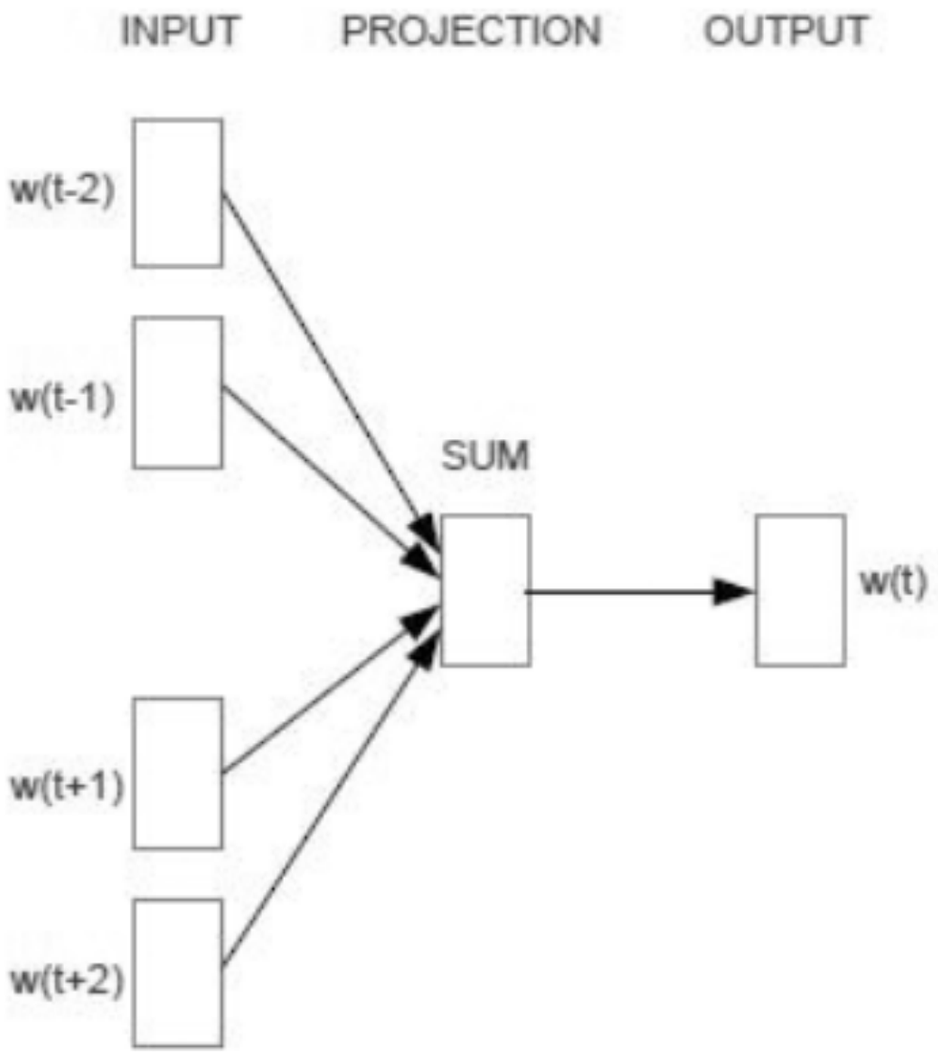


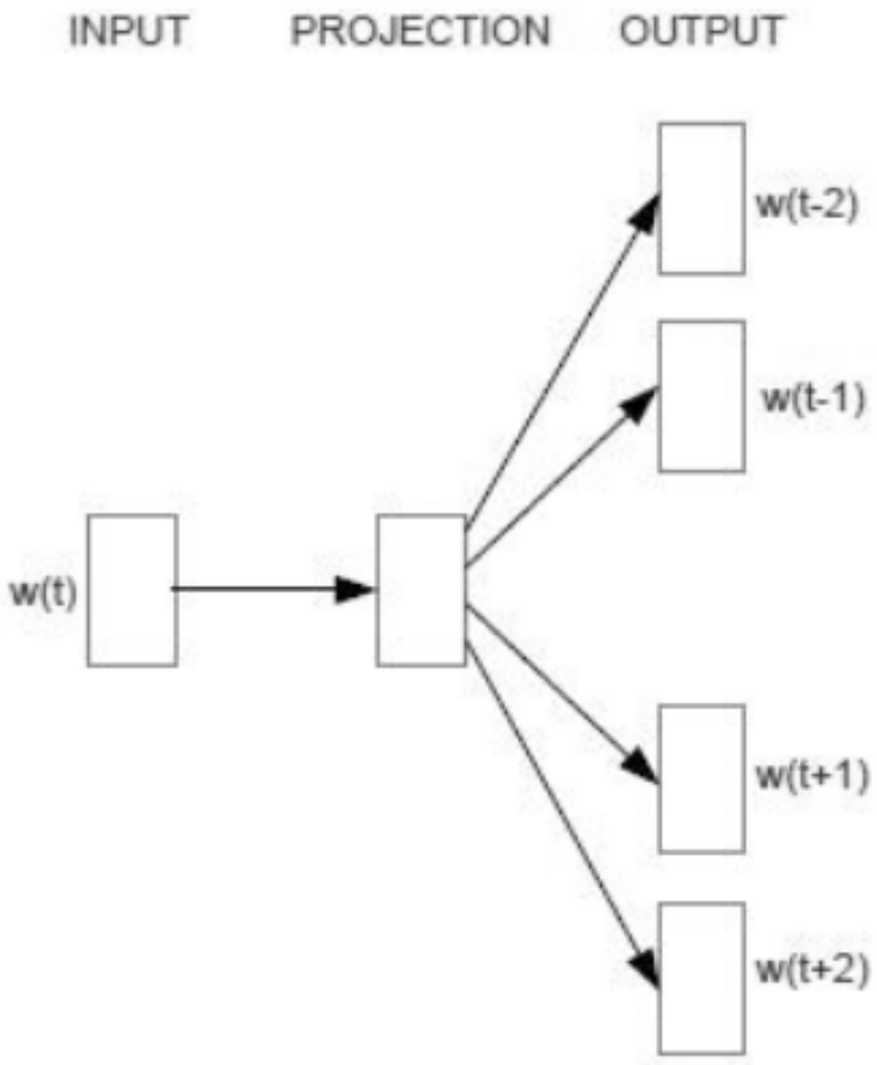


Convolution Neural Network (CNN)

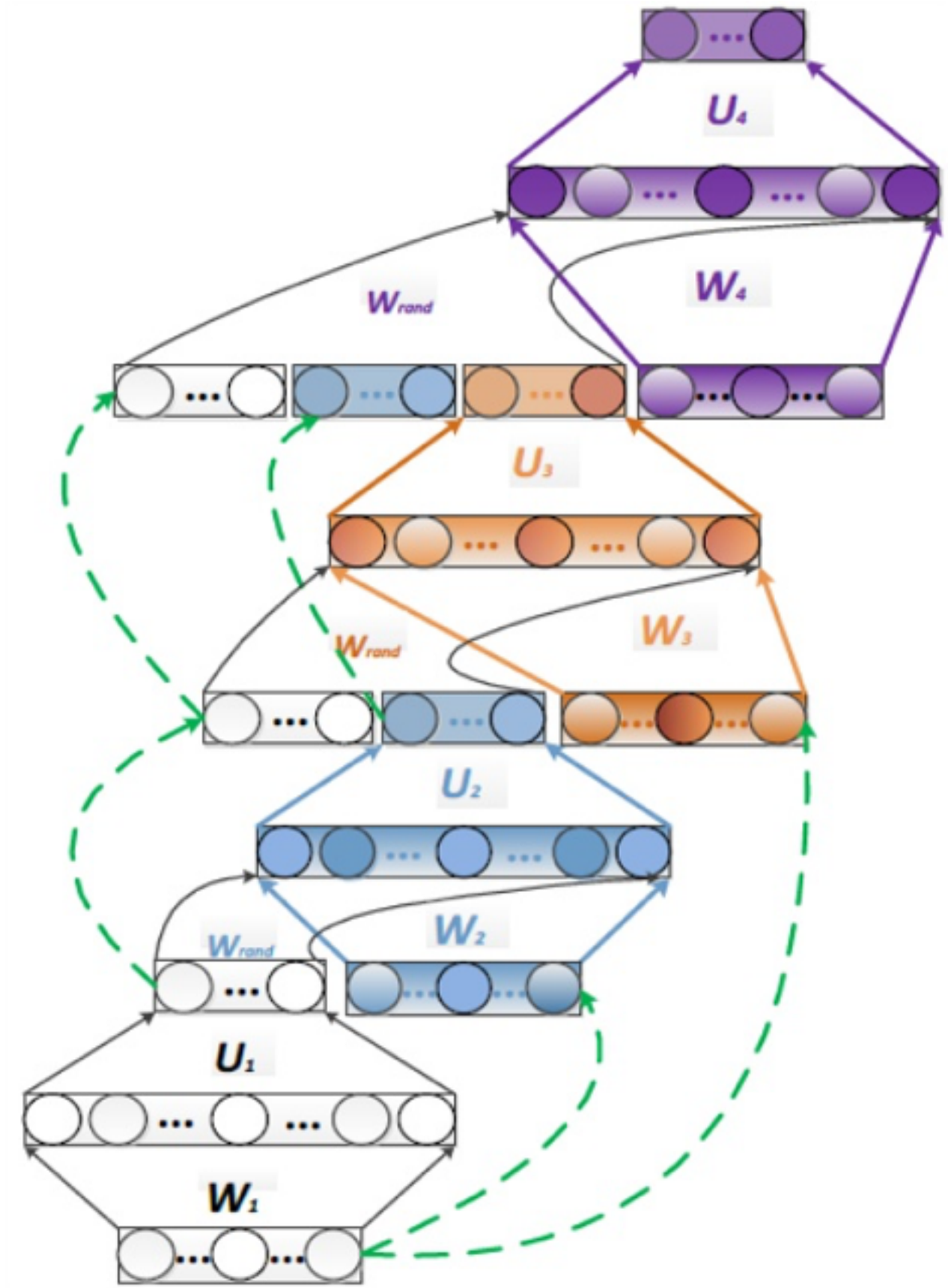

Semantic layer: y
Affine projection matrix: W_s
Max pooling layer: v
Max pooling operation
Convolutional layer: h_t
Convolution matrix: W_c
Word hashing layer: f_t
Word hashing matrix: W_f
Word sequence: x_t
128
300
max
max
max
300
300
300
30k
30k
30k
30k
30k
30k
<s>
w_1
w_2
w_T
<s>

C-DSSM

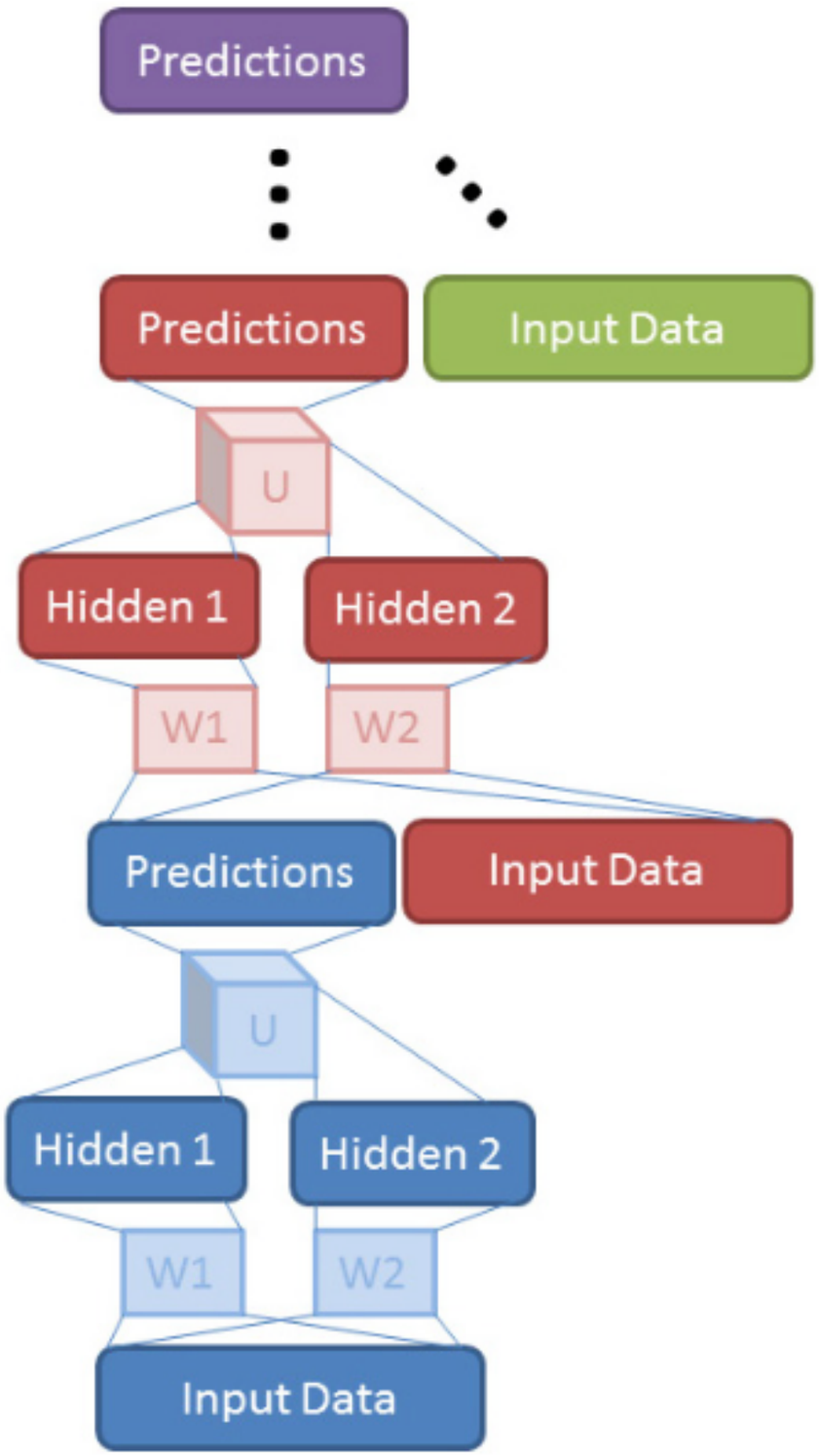




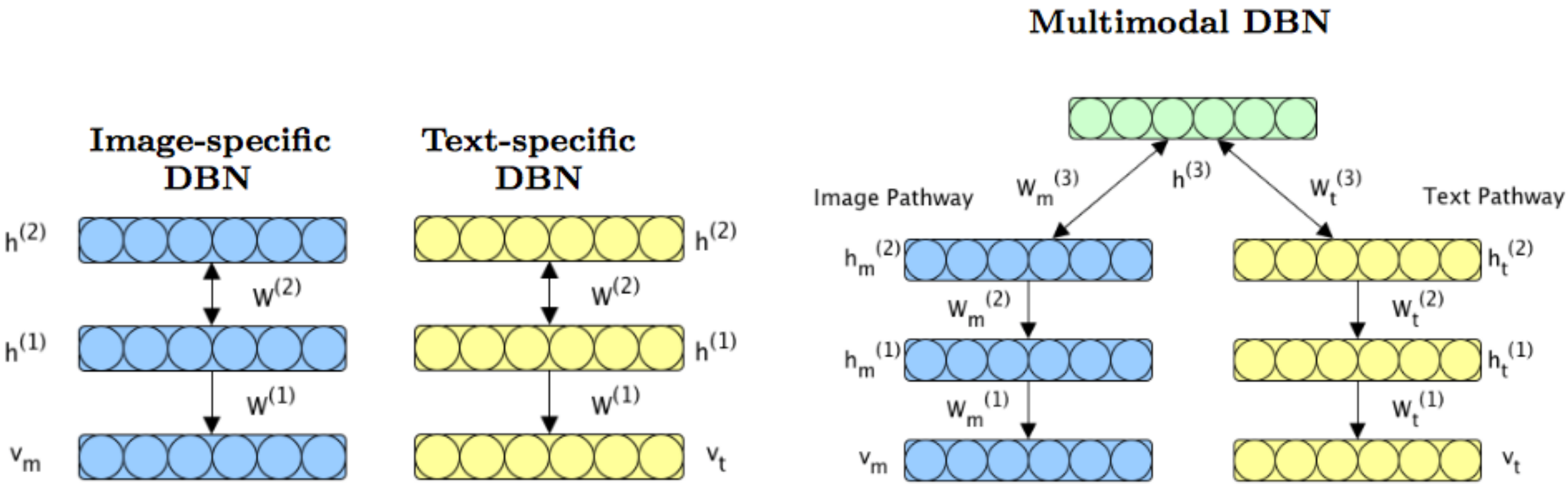
CBOW & Skip-gram (Word2Vec)



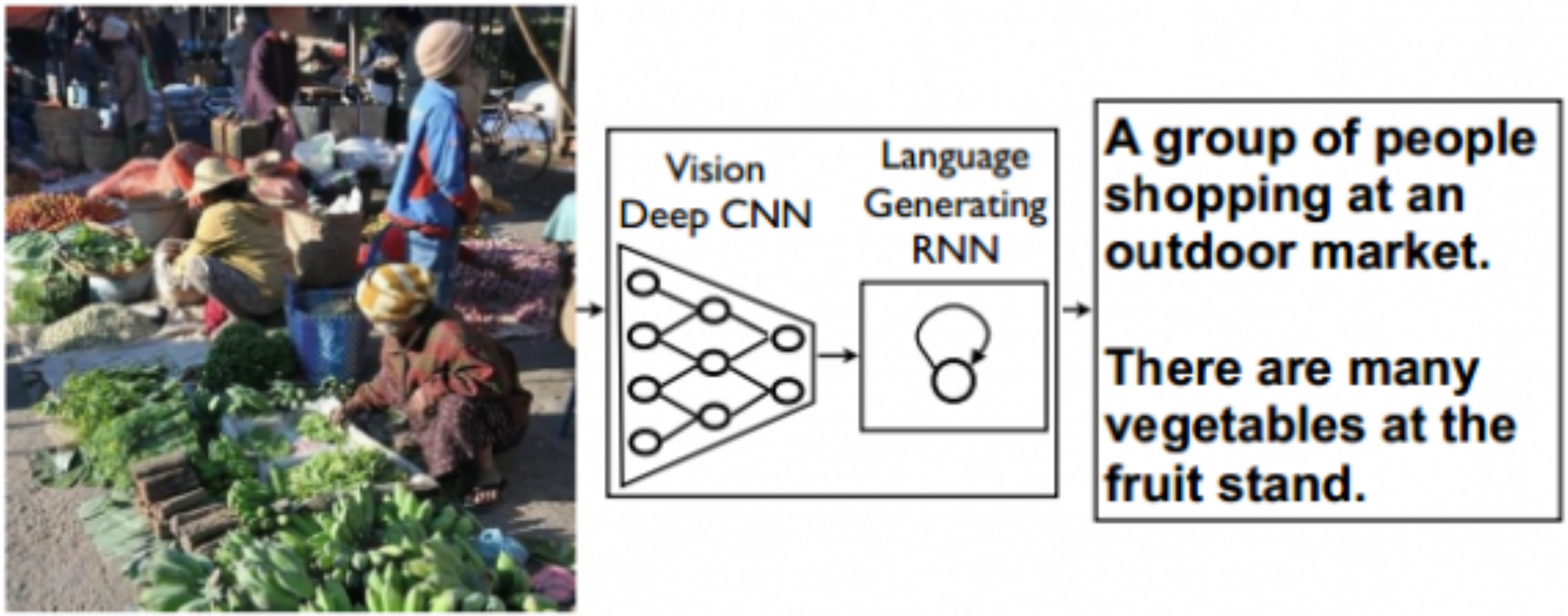
Deep Stacking Network (DSN)



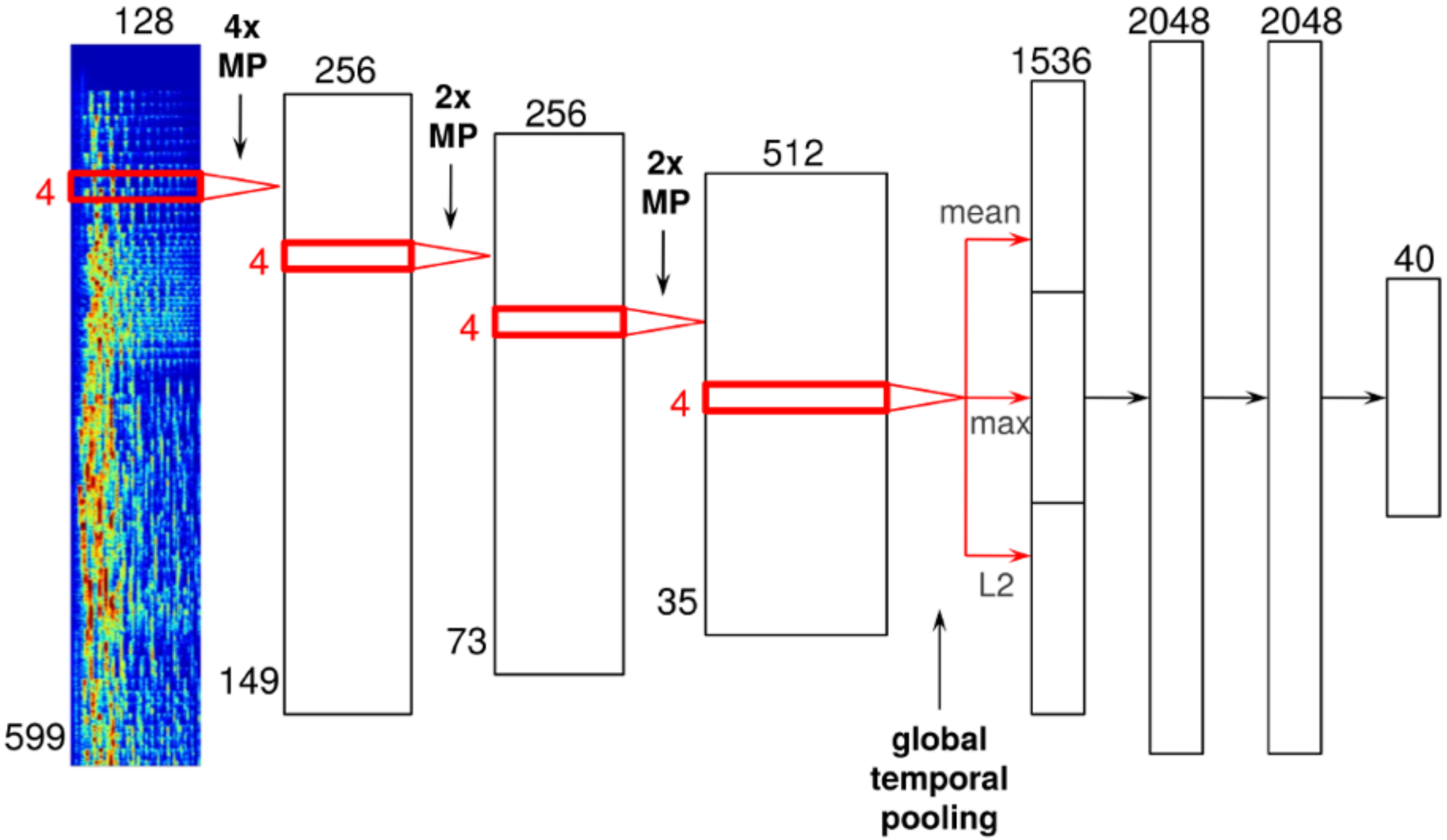
Tensor Deep Stacking Network (TDSN)



Multi-modal / Mulit-tasking



Google Picture to Text



Content-based Spotify Recommendation

In various applications,

Deep Learning provides either

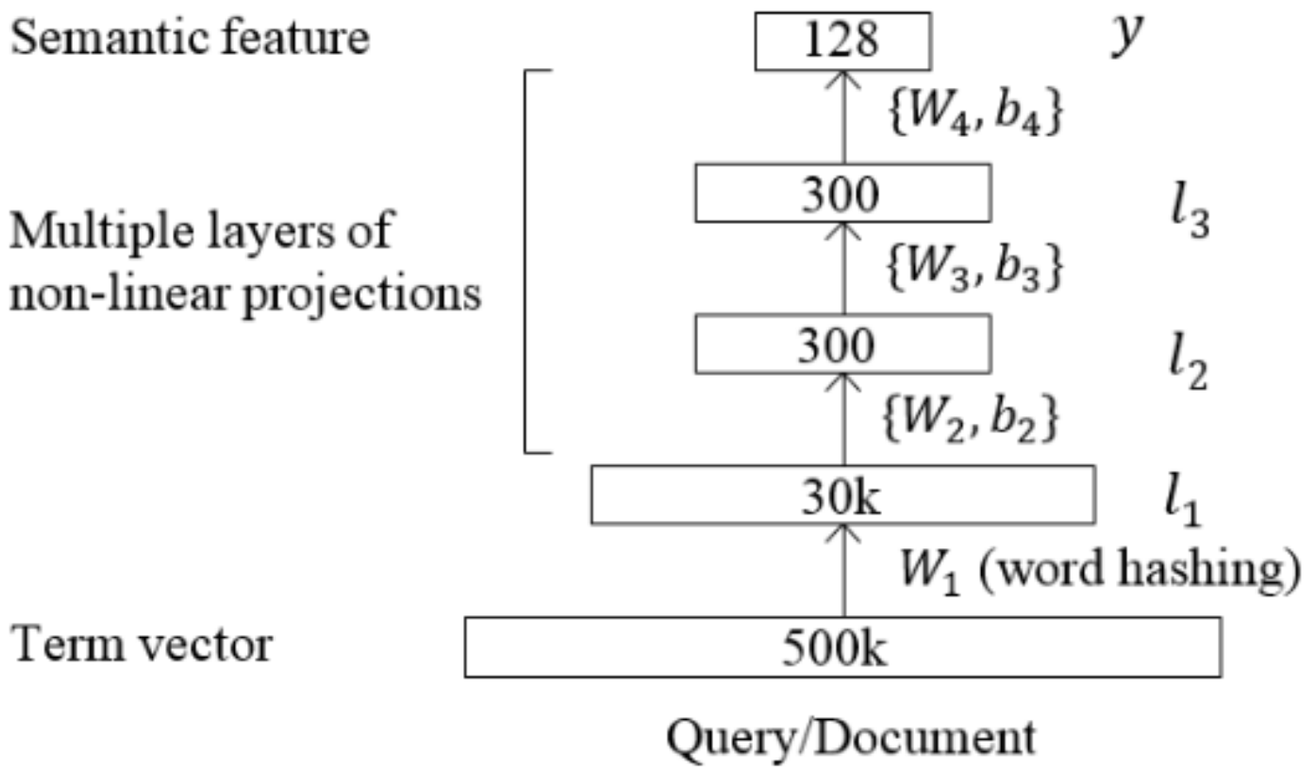
— better performance than state-of-the-art

— similar performance with less labor

Remarks

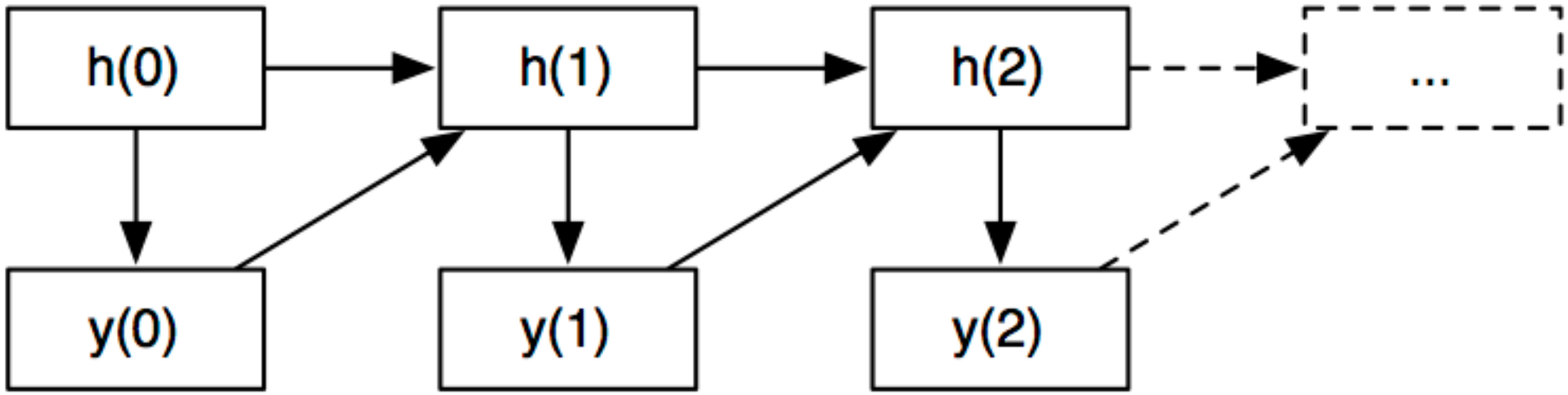
Deep Personalization

with Rich Profile in a Vector Representation



- Versatile / flexible input
- Dimension reduction
- Same feature space

==> Similarity computation



http://erikbern.com/?p=589

RNN for CF

Hard work

— Application-specific configuration

— Network structure

— Objective and loss function

— Parameter optimization

Network		MNIST-small classif. test error	MNIST-rotation classif. test error
Type	Depth		
Neural network (random initialization, + fine-tuning)	1	**4.14** % ± 0.17	15.22 % ± 0.31
	2	**4.03** % ± 0.17	**10.63** % ± 0.27
	3	**4.24** % ± 0.18	11.98 % ± 0.28
	4	4.47 % ± 0.18	11.73 % ± 0.29
SAA network (autoassociator learning + fine-tuning)	1	3.87 % ± 0.17	11.43% ± 0.28
	2	**3.38** % ± 0.16	9.88 % ± 0.26
	3	**3.37** % ± 0.16	**9.22** % ± 0.25
	4	**3.39** % ± 0.16	**9.20** % ± 0.25
SRBM network (CD-1 learning + fine-tuning)	1	3.17 % ± 0.15	10.47 % ± 0.27
	2	**2.74** % ± 0.14	9.54 % ± 0.26
	3	**2.71** % ± 0.14	**8.80** % ± 0.25
	4	**2.72** % ± 0.14	**8.83** % ± 0.24

Yann LeCun —> Facebook

Geoffrey Hinton —> Google

Andrew Ng —> Baidu

Joshua Benjio —> U of Montreal

References

Tutorial

— http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html

— http://deeplearning.stanford.edu/tutorial

Paper

— Deep learning: Method and Applications (Deng & Yu, 2013)

— Representation learning: A review and new perspective (Benjio *et al*., 2014)

— Learning deep architecture for AI (Benjio, 2009)

Slide & Video

— http://www.cs.toronto.edu/~fleet/courses/cifarSchool09/slidesBengio.pdf

— http://nlp.stanford.edu/courses/NAACL2013

Book

— Deep learning (Benjio *et al*. 2015) http://www.iro.umontreal.ca/~bengioy/dlbook/

Open Sources (from wikipedia)

— Torch (Lua) https://github.com/torch/torch7

— Theano (Python) https://github.com/Theano/Theano/

— Deeplearning4j (word2vec for Java) https://github.com/SkymindIO/deeplearning4j

— ND4J (Java) http://nd4j.org https://github.com/SkymindIO/nd4j

— DeepLearn Toolbox (matlab) https://github.com/rasmusbergpalm/DeepLearnToolbox/graphs/contributors

— convnetjs (javascript) https://github.com/karpathy/convnetjs

— Gensim (word2vec for Python) https://github.com/piskvorky/gensim

— Caffe (image) http://caffe.berkeleyvision.org

— More+++

FBI WARNING

Some reference citations are missed.

Many of them come from the first tutorial and the first paper, and others from googling.

Left to Blank