Machine Learning for Sensory Signals

Representation Learning in Deep Networks

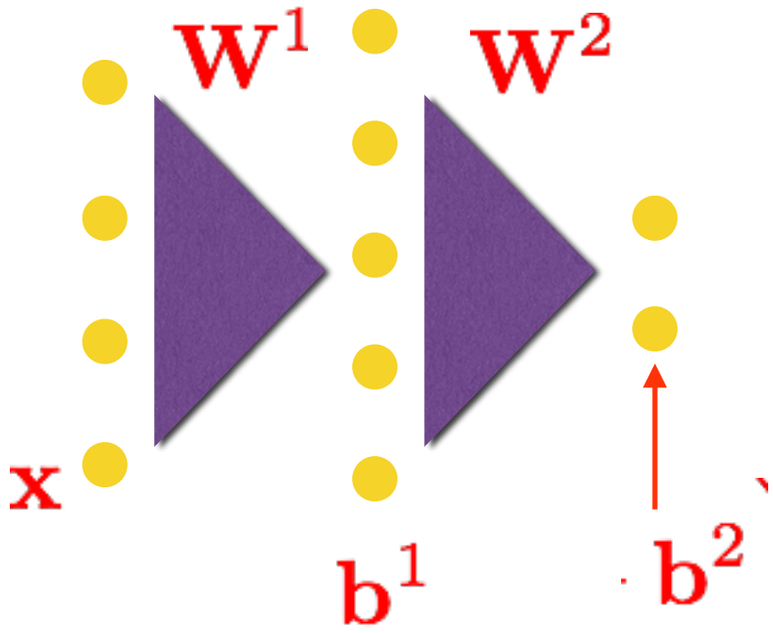
20-04-2017

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

Neural Networks

Multi-layer Perceptron [Hopfield, 1982]



$$\mathbf{v}^2 = \psi\left(\mathbf{W}^2\phi(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1) + \mathbf{b}^2\right)$$

ϕ non-linear function (*tanh*,*sigmoid*)

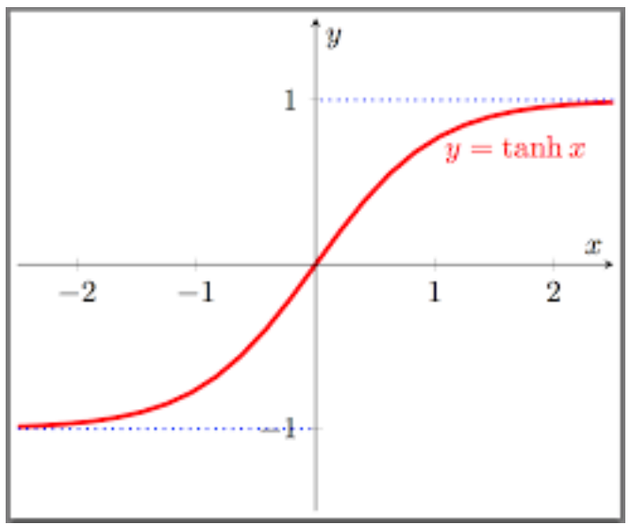
ψ thresholding function

- Useful for classifying non-linear data boundaries - non-linear class separation can be realized given enough data.

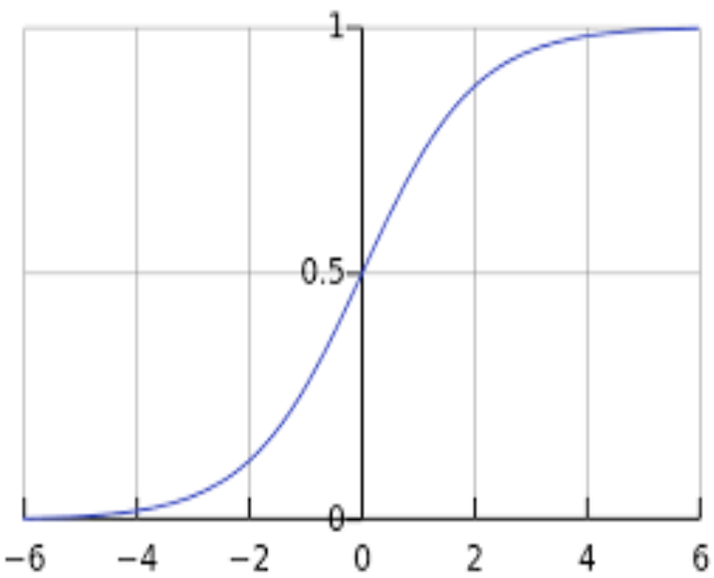
Neural Networks

Types of Non-linearities ϕ

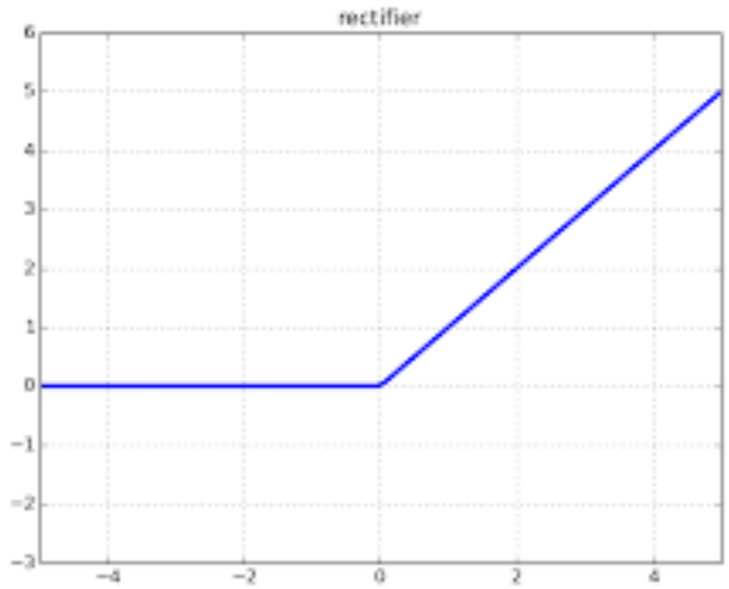
tanh



sigmoid



ReLu



Cost-Function

Mean Square Error

$$J_{MSE} = \sum_{i=1}^{M} ||\mathbf{v}_i - \mathbf{y}_i||^2$$

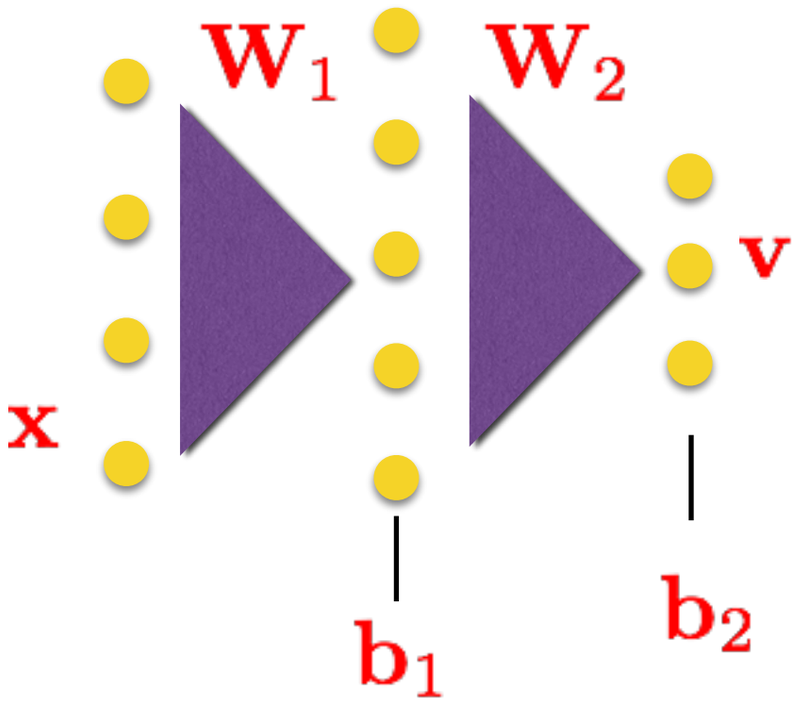
Cross Entropy

$$J_{CE} = -\sum_{i=1}^{M} \mathbf{y}_i^T \log(\mathbf{v}_i)$$

$\mathbf{y}_i$ are the desired outputs

Learning Posterior Probabilities with NNs

Neural networks predict posterior probabilities [Richard, 1991]



$$P(C_i|\mathbf{X}) = \frac{p(\mathbf{X}|C_i)p(C_i)}{p(\mathbf{X})}$$

When DNNs are trained with CE or MSE

$$\mathbf{v}(\mathbf{x}) = \mathcal{E}_{y|\mathbf{X}=\mathbf{x}}[\mathbf{y}]$$

Neural networks estimate conditional expectation of the desired targets given the input

When the targets are discrete classes $\mathbf{y} = [0\ 0\ ..0\ 1\ 0\ ..0]$ conditional expectation is the class posterior !

Richard, Michael D., and Richard P. Lippmann. "Neural network classifiers estimate Bayesian a posteriori probabilities." *Neural computation* 3.4 (1991): 461-483.

Learning Posterior Probabilities with NNs

Choice of target function ψ

- Linear for regression
- Softmax function for classification

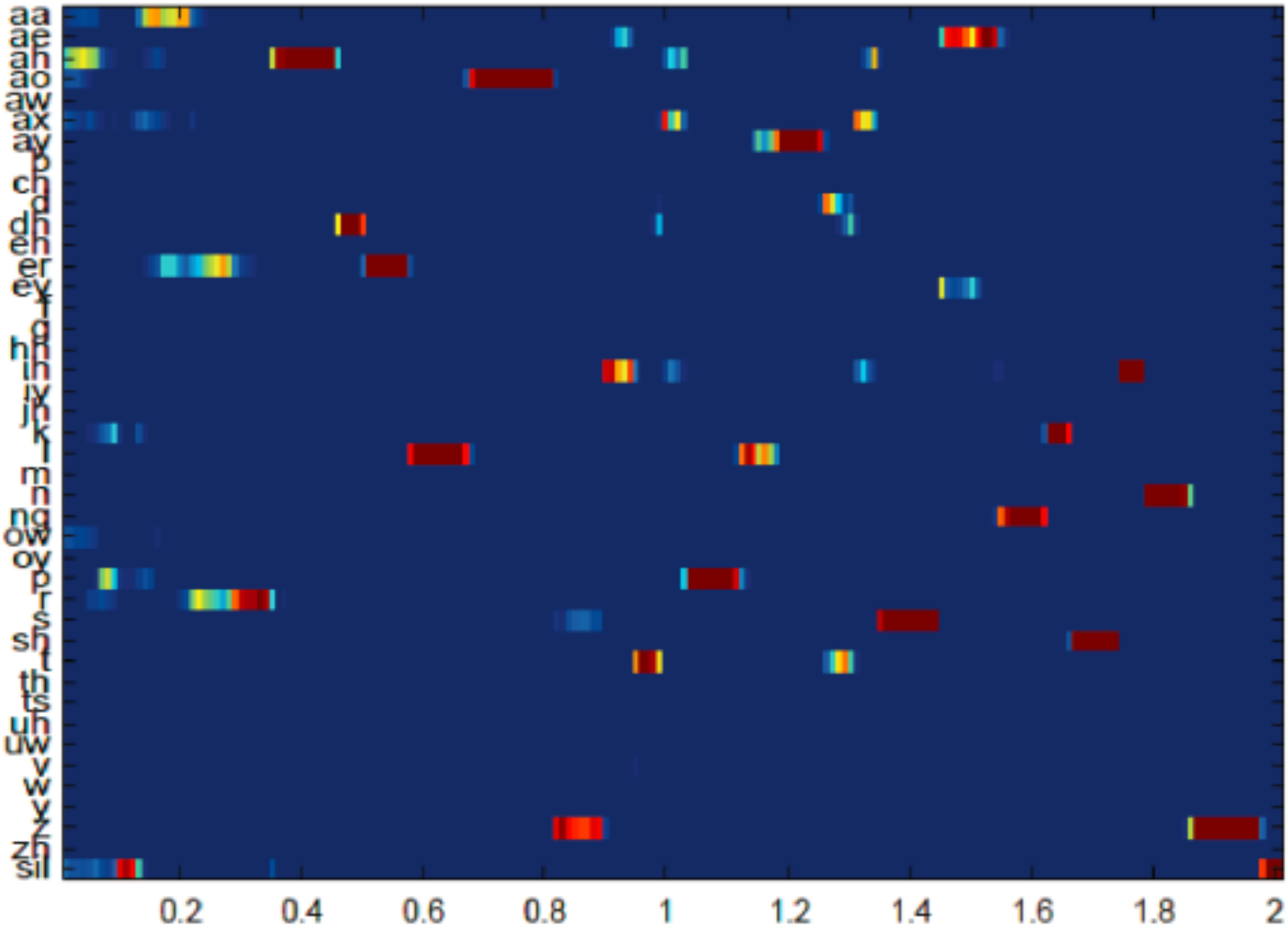
$$\psi(v_i) = \frac{e^{v_i}}{\sum_i e^{v_i}}$$

- Softmax produces positive values that sum to 1
- Allows the interpretation of outputs as posterior probabilities

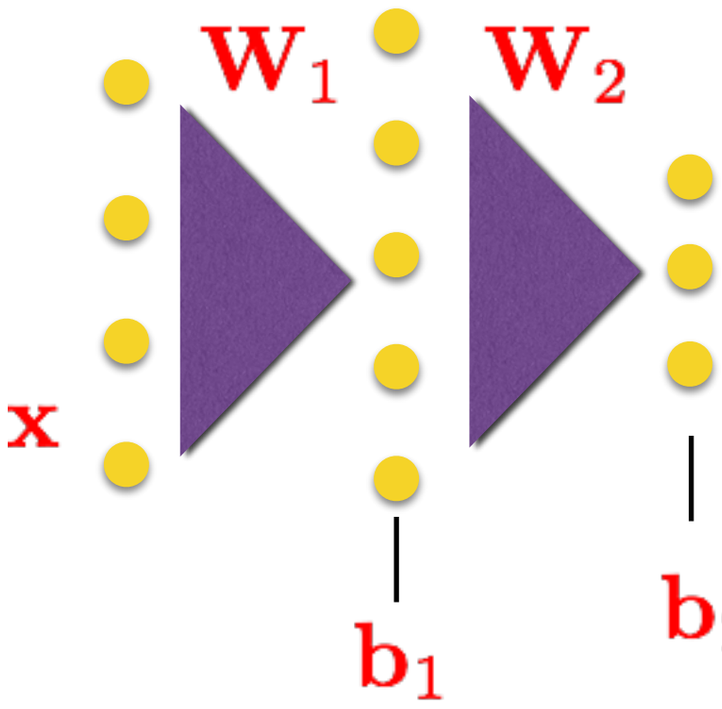
Example - Train a NN to approximate speech classes using sigmoidal non-linearity and softmax target function

Learning Posterior Probabilities with NNs

Example of a speech posteriogram



Parameter Learning

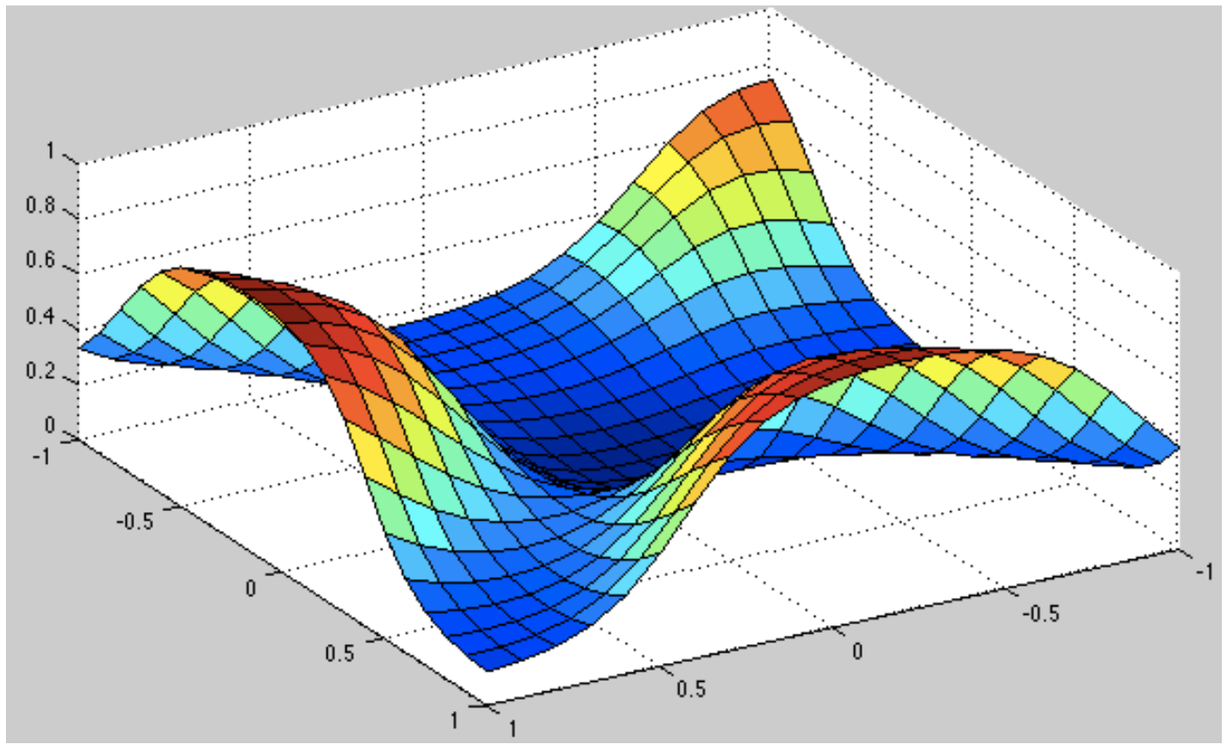


$$\mathbf{v}^2 = \psi\left(\mathbf{W}^2\phi(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1) + \mathbf{b}^2\right)$$

Error function for entire data

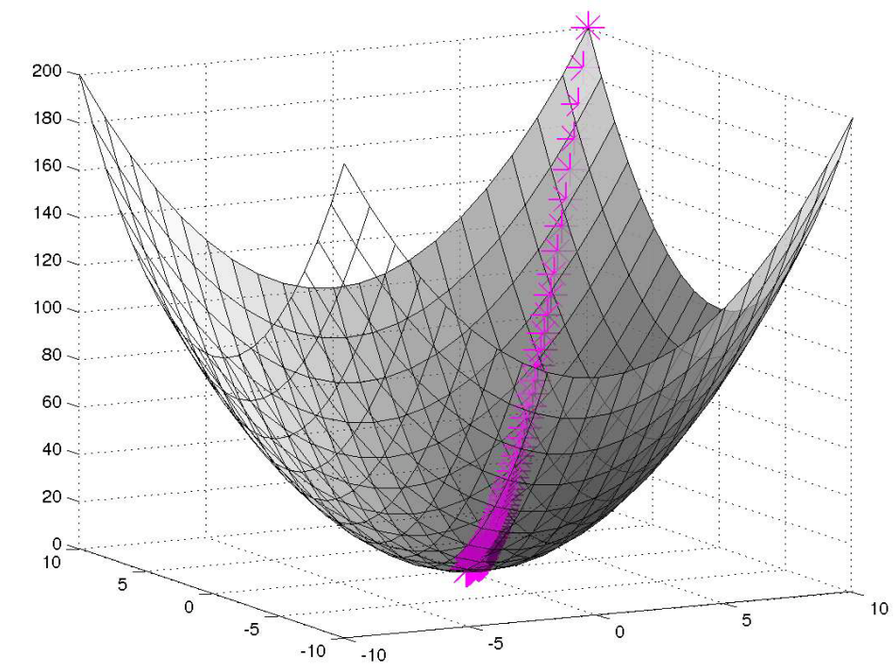
$$J_{MSE} = \sum_{i=1}^{M} ||\mathbf{v}_i - \mathbf{y}_i||^2$$

Typical Error Surface as a function of parameters (weights and biases)



Parameter Learning

Error surface close to a local optima



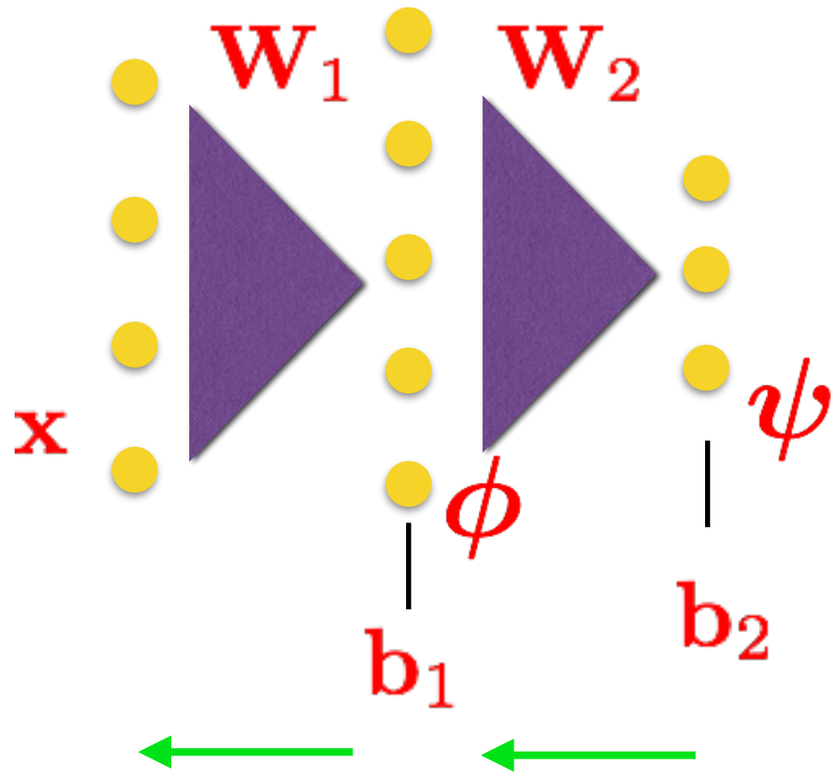
Non-linear nature of error function

- Move in the reverse direction of the gradient

$$\boldsymbol{W}_1^t = \boldsymbol{W}_1^{t-1} - \eta \frac{\partial J}{\partial \boldsymbol{W}_1}$$

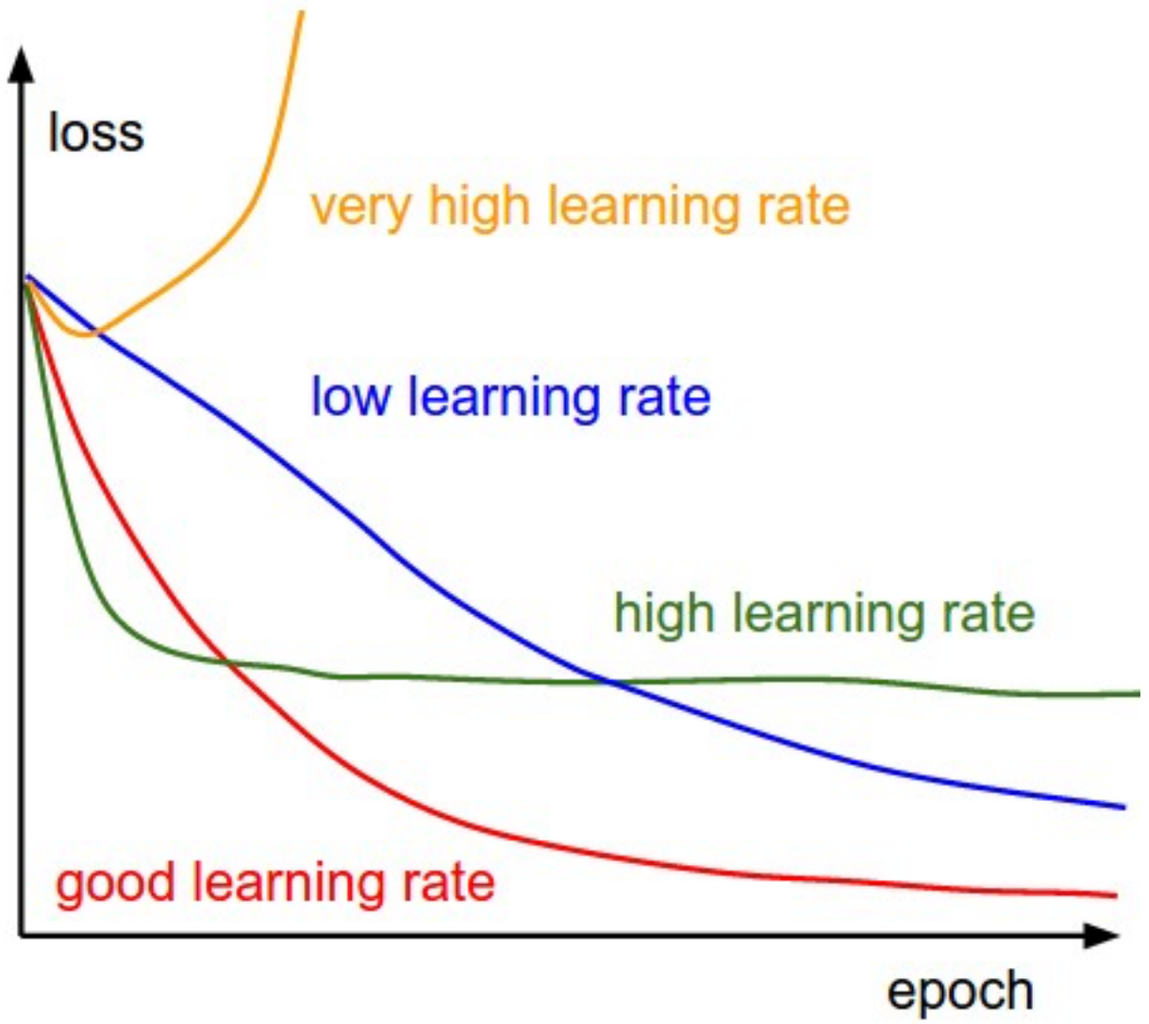
Error back propagation

$$\frac{\partial J}{\partial \mathbf{W}_1} = \frac{\partial J}{\partial \psi} \times \frac{\partial \psi}{\partial \phi} \times \frac{\partial \phi}{\partial \mathbf{W}_1}$$

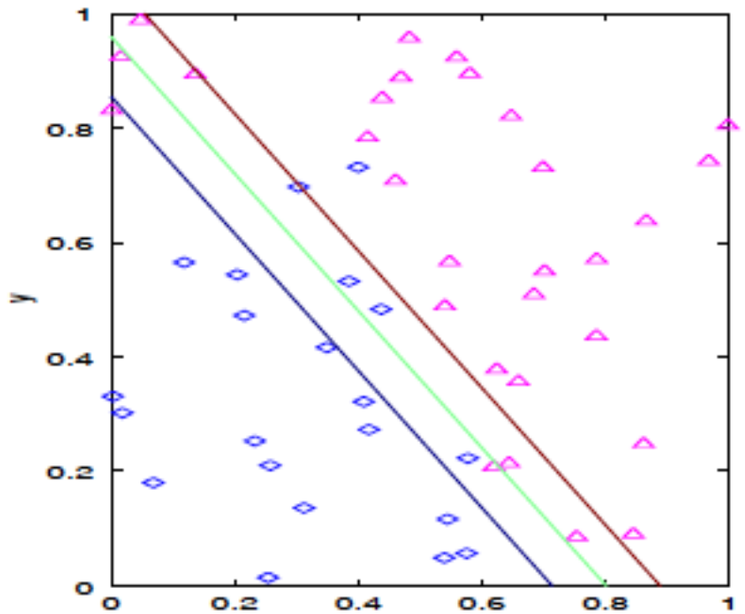


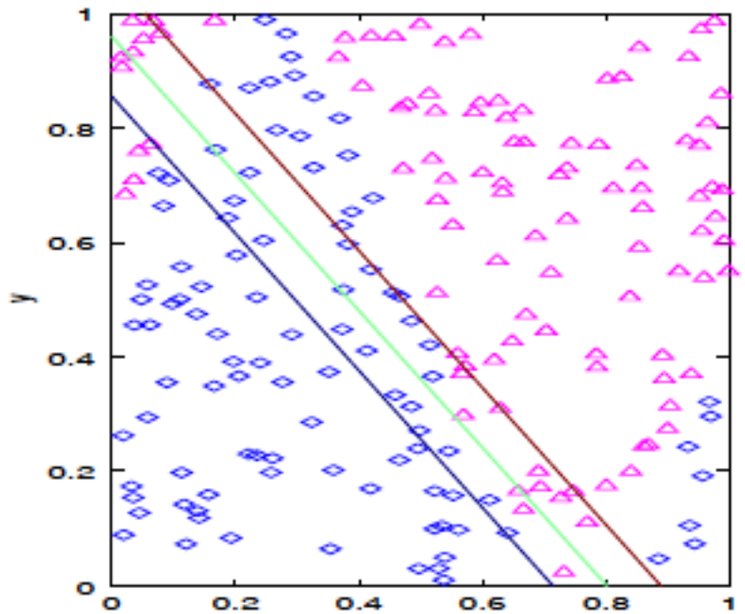
Parameter Learning

- Solving a non-convex optimization.
- Iterative solution.
- Depends on the initialization.
- Convergence to a local optima.
- Judicious choice of learning rate



Underfit

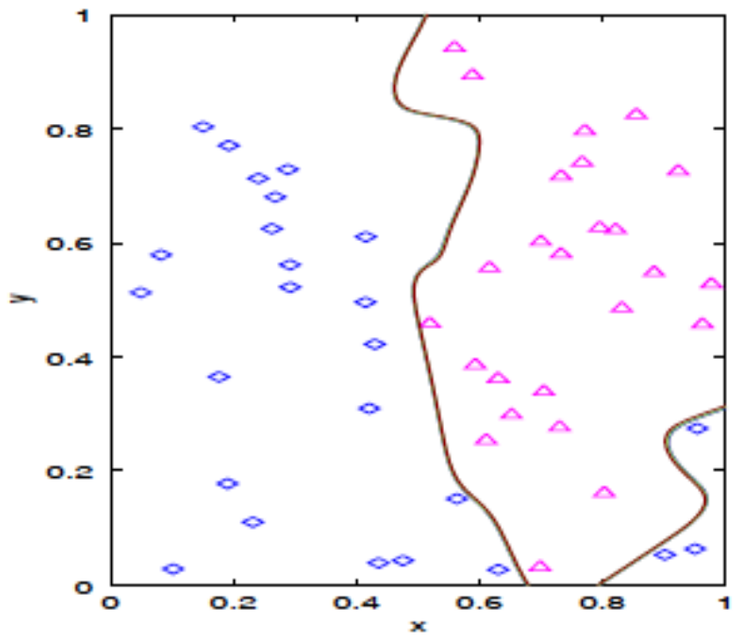


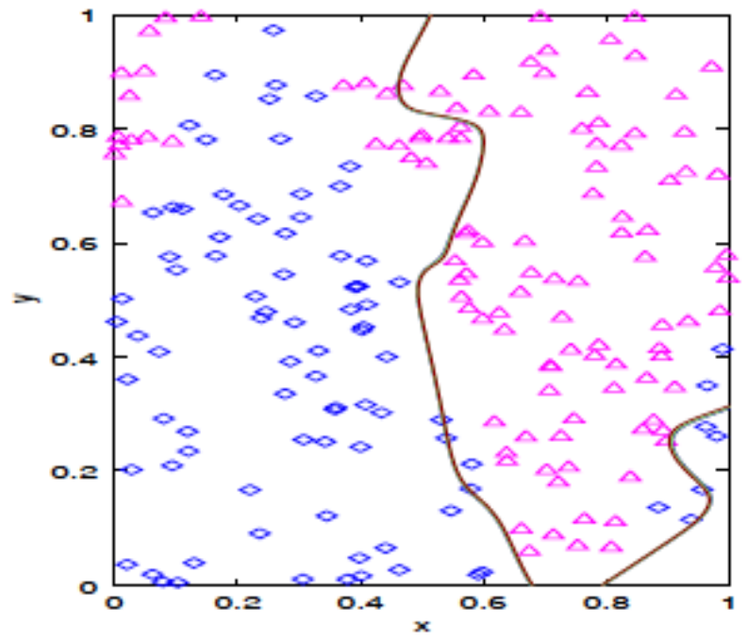


- The model is not able to capture the variability in the data (Linear Model)
- Both the training and testing error are high (15%,20%)
- Try to learn a more complex model – more features, more hidden neurons, decrease regularization
- More data would not help

(Courtesy - Dr D. Vijayasenan, NITK)

Overfit

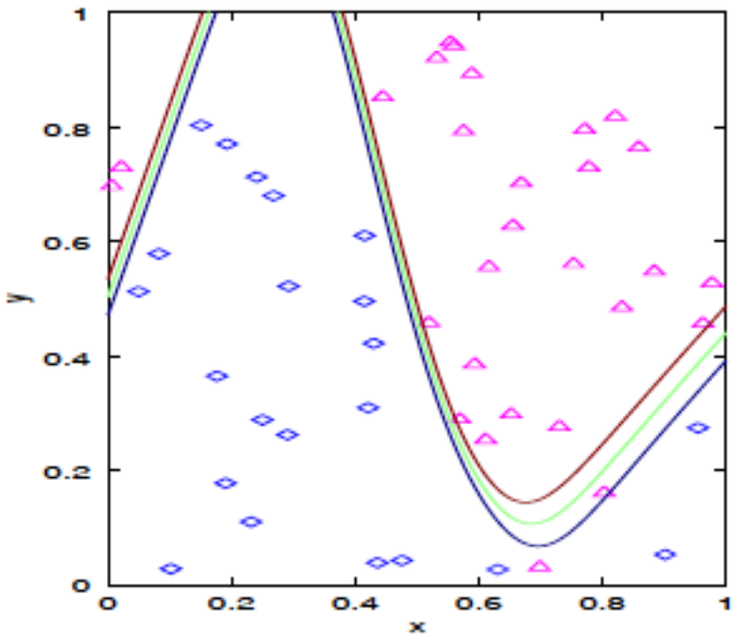


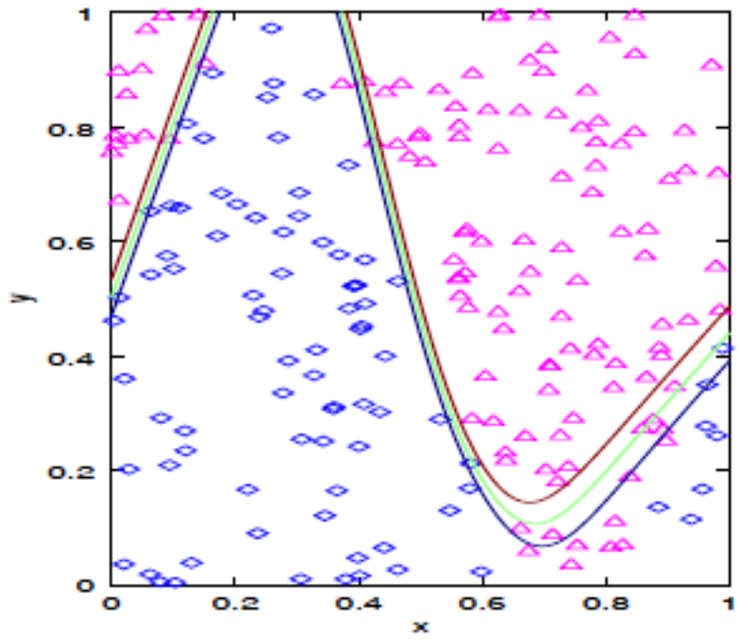


- The model is capturing data as well as accidental variations (100 hidden neurons)
- Training error is too low and testing error is too high (0%, and 16%)
- Try to learn a simpler model – less features, less hidden neurons, increase regularization
- More data would help

(Courtesy - Dr D. Vijayasenan, NITK)

Compromise





- Reasonable training and test errors – $(4\%, 8\%)$
- Appropriate model – capturing only the global characteristics not details

(Courtesy - Dr D. Vijayasenan, NITK)

Summary so far…

- Neural networks as discriminative classifiers
- Need for hidden layer
- Choice of non-linearities and target functions
- Estimating posterior probabilities with NNs
- Parameter learning with back propagation.

Roadmap

- Basics of Machine Learning
- Neural Networks
- Deep Networks
- Representation Learning in Deep Networks
- Applications in Speech Processing and Insights
- Future Research Directions

Need For Deep Networks

Modeling complex real world data like speech, image, text

- Single hidden layer networks are too restrictive.
- Needs large number of units in the hidden layer and trained with large amounts of data.
- Not generalizable enough.

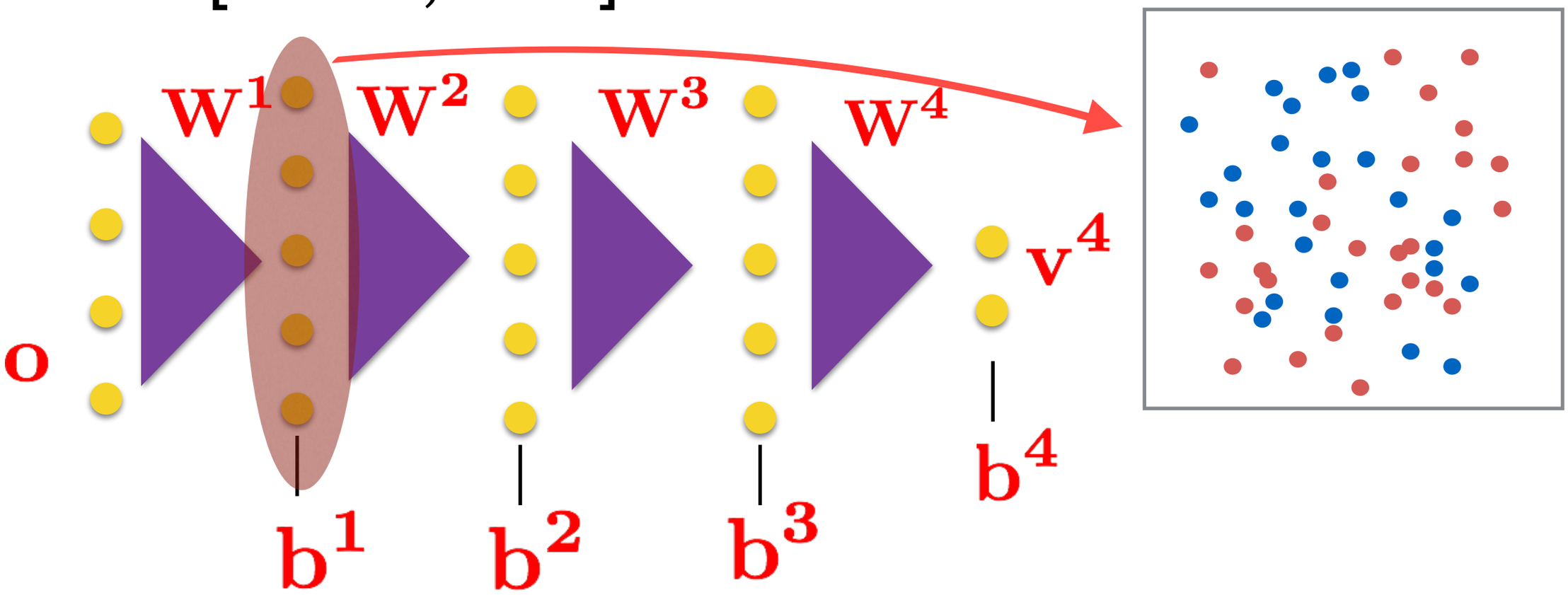
Networks with multiple hidden layers - deep networks

(Open questions till 2005)

- Are these networks trainable ?
- How can we initialize such networks ?
- Will these generalize well or over train ?

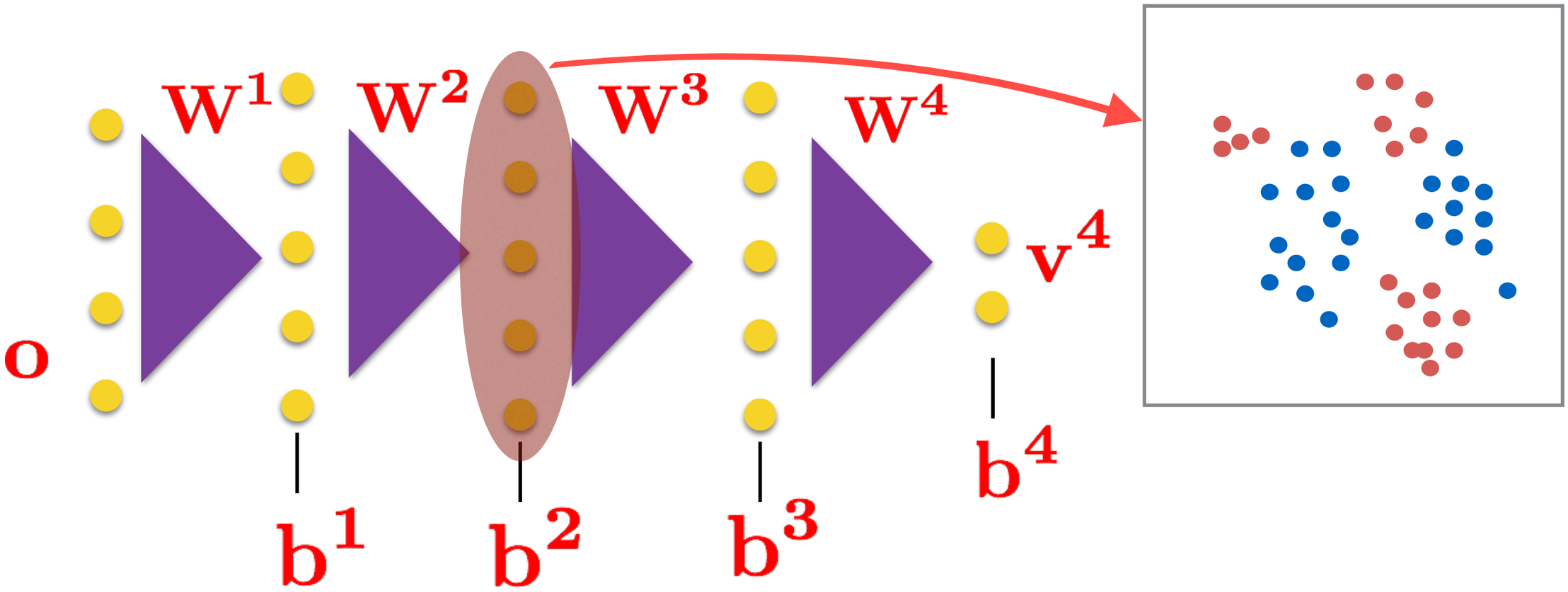
Deep Networks Intuition

Neural networks with multiple hidden layers - Deep networks [Hinton, 2006]



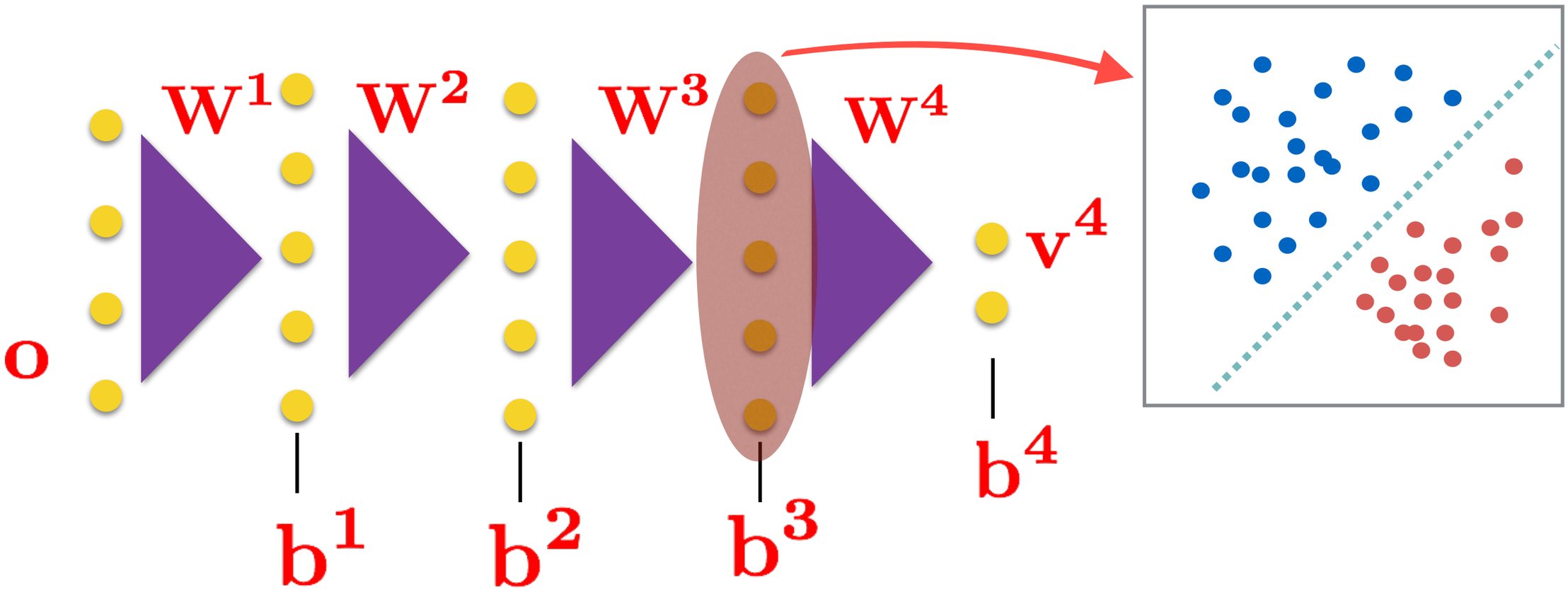
Deep Networks Intuition

Neural networks with multiple hidden layers - Deep networks



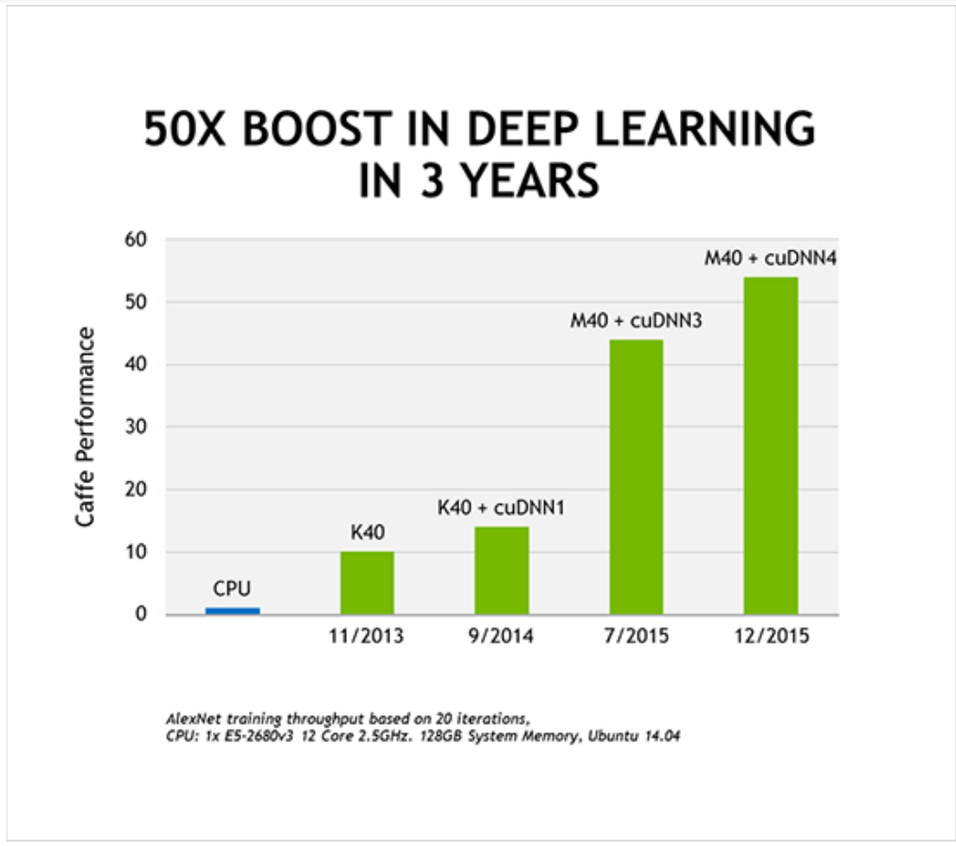
Deep Networks Intuition

Neural networks with multiple hidden layers - Deep networks



Deep networks perform hierarchical data abstractions which enable the non-linear separation of complex data samples.

Deep Networks

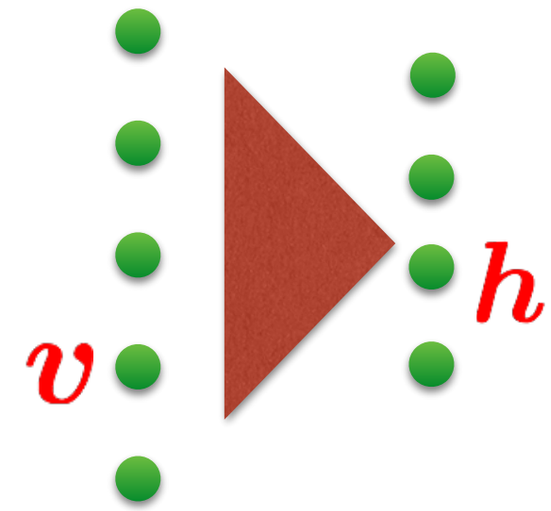


- Are these networks trainable ?
  - Advances in computation and processing
  - Graphical processing units (GPUs) performing multiple parallel multiply accumulate operations.
  - Large amounts of supervised data sets

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

Deep Networks Initialization



Initializing large networks with Restricted Boltzmann Machine (RBM)

- Gaussian Bernoulli RBM - Gaussian visible layer and Bernoulli hidden layer.
- Define an energy function

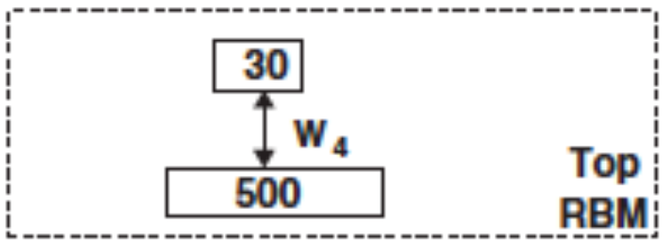
$$E(\mathbf{v}, \mathbf{h}) = 0.5(\mathbf{v} - \mathbf{a})^T(\mathbf{v} - \mathbf{a}) - \mathbf{b}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v}$$

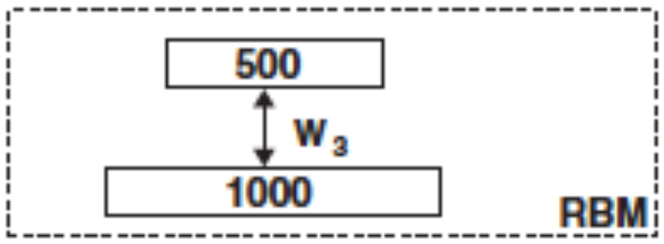
- Define joint probability density $P(\mathbf{v}, \mathbf{h}) = \dfrac{e^{-E(\mathbf{v},\mathbf{h})}}{Z}$
- Estimate the parameters $a,\ b,\ W$ by maximizing the joint density

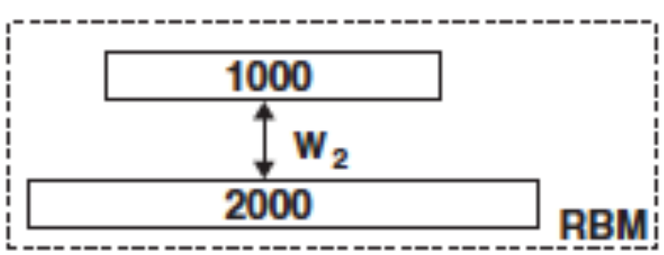
Established 1911

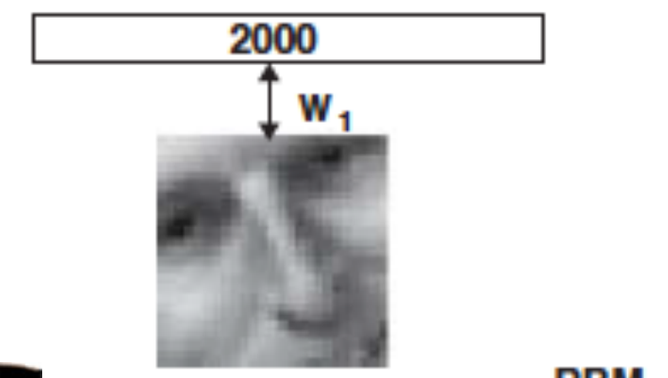
INDIAN INSTITUTE OF SCIENCE भारतीय विज्ञान संस्थान

Restricted Boltzmann Machine









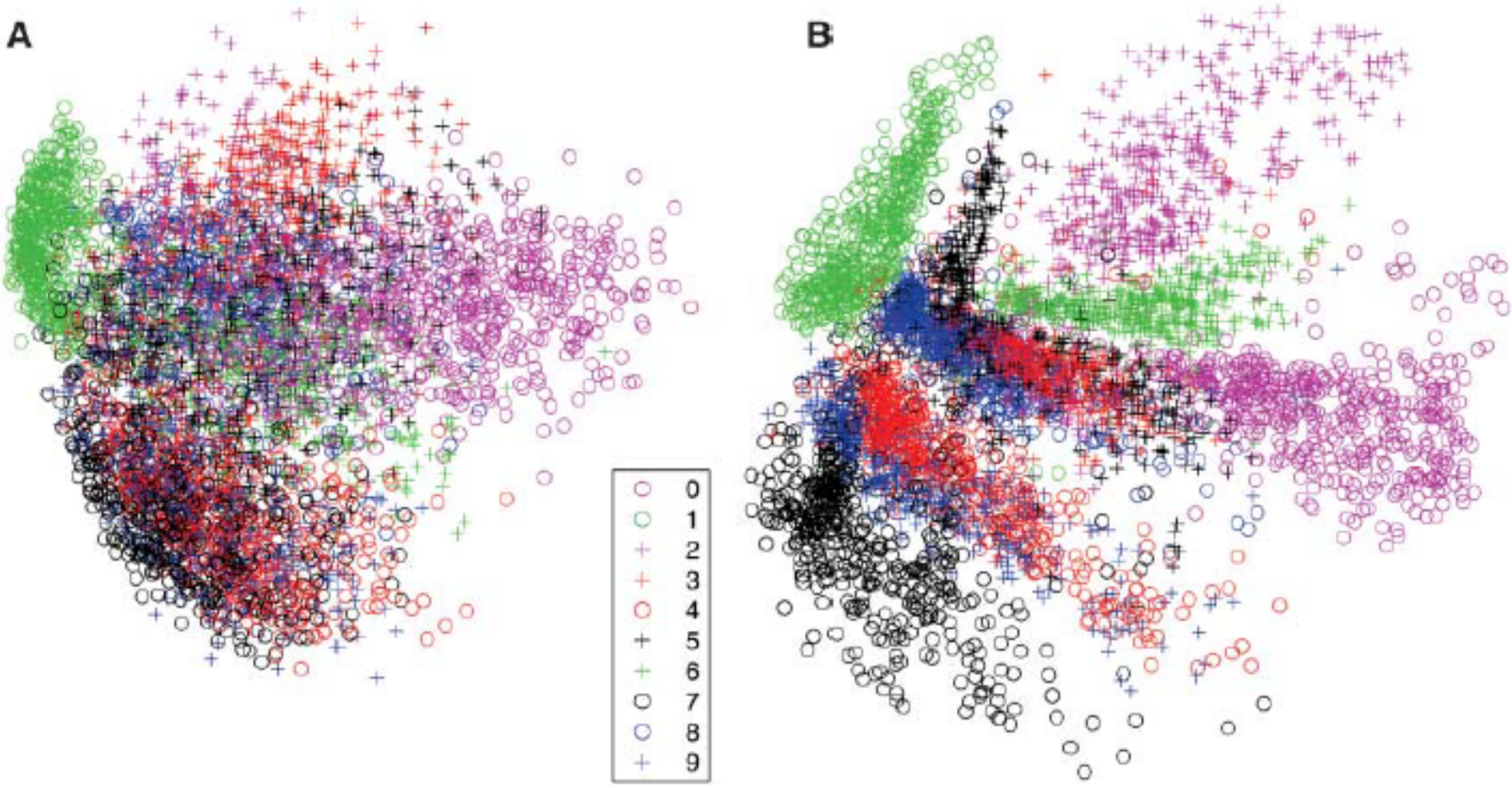
It can be shown that probability of hidden unit being is the sigmoid function

$$P(h_i = 1|\mathbf{v}) = \frac{1}{1 + e^{-(w_i^T \mathbf{v} + b_i)}}$$

RBM layers can be stacked one above the other in a hierarchical fashion.

Earliest application was in dimensionality reduction [Hinton, 2006].

Restricted Boltzmann Machine

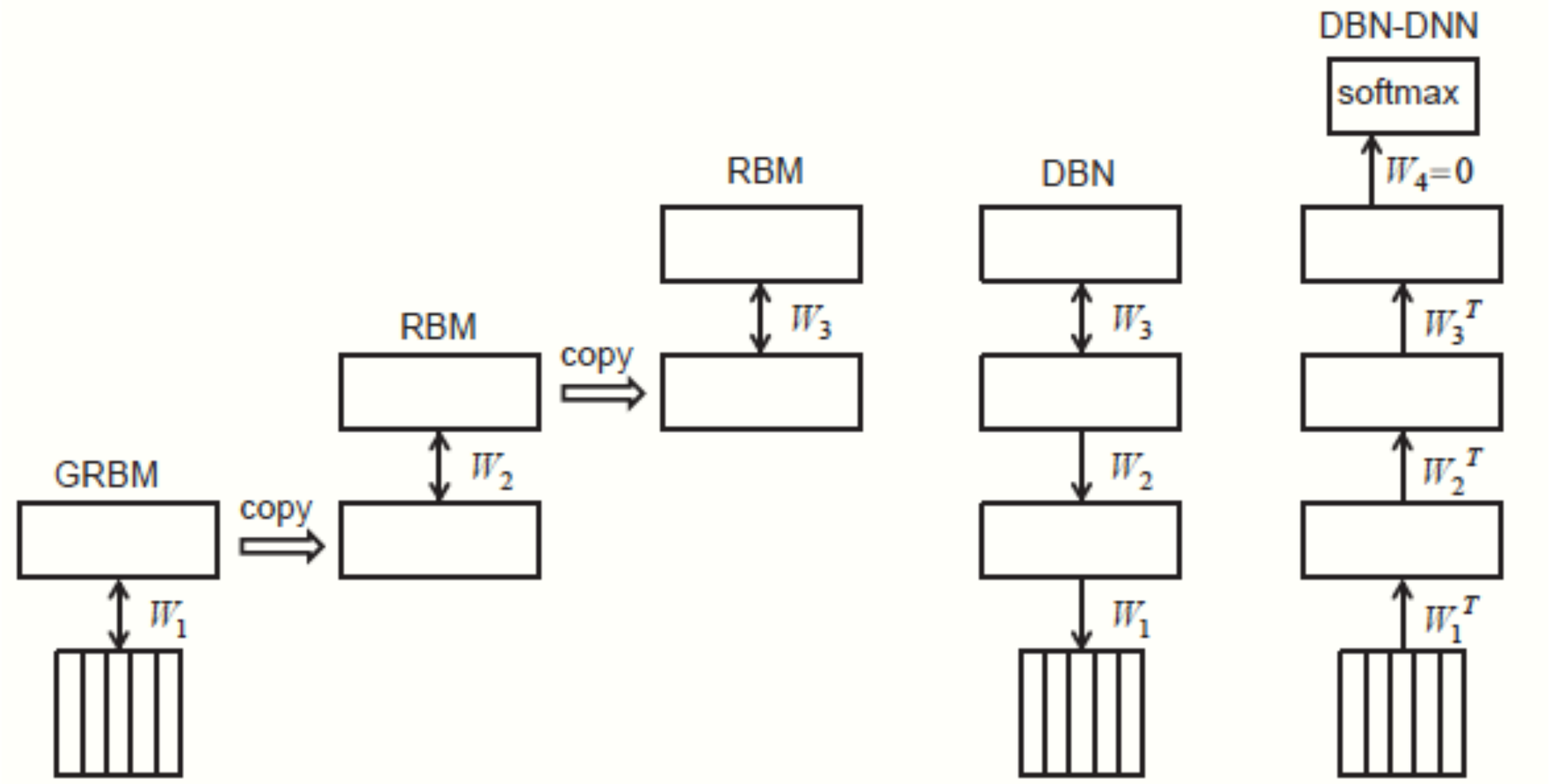


PCA **RBM**

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

DBN for DNN Initialization



[Hinton, 2013]

Summary so far…

- Deep Neural networks as extensions of NNs.
- Initiation behind multiple hidden layers
- Initialization with Restricted Boltzmann Machine
  - RBMs for dimensionality reduction

Roadmap

- Basics of Machine Learning
- Neural Networks
- Deep Networks
- Representation Learning in Deep Networks
- Applications in Speech Processing and Insights
- Future Research Directions

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

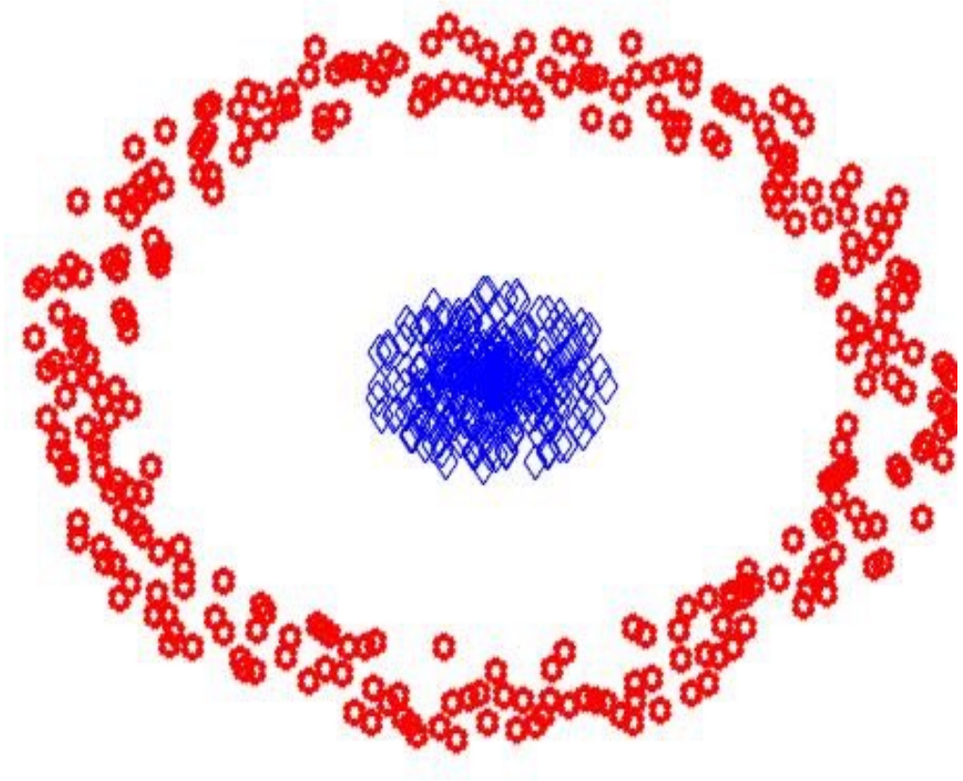
Deep Networks

- Will the networks generalize with deep networks
    - DNNs are quite data hungry and performance improves by increasing the data.
    - Generalization problem is tackled by providing training data from all possible conditions.
        - Many artificial data augmentation methods have been successfully deployed
- Providing the state-of-art performance in several real world applications.

Representation Learning in Deep Networks

- The input data representation is one of most important components of any machine learning system.

Cartesian Coordinates



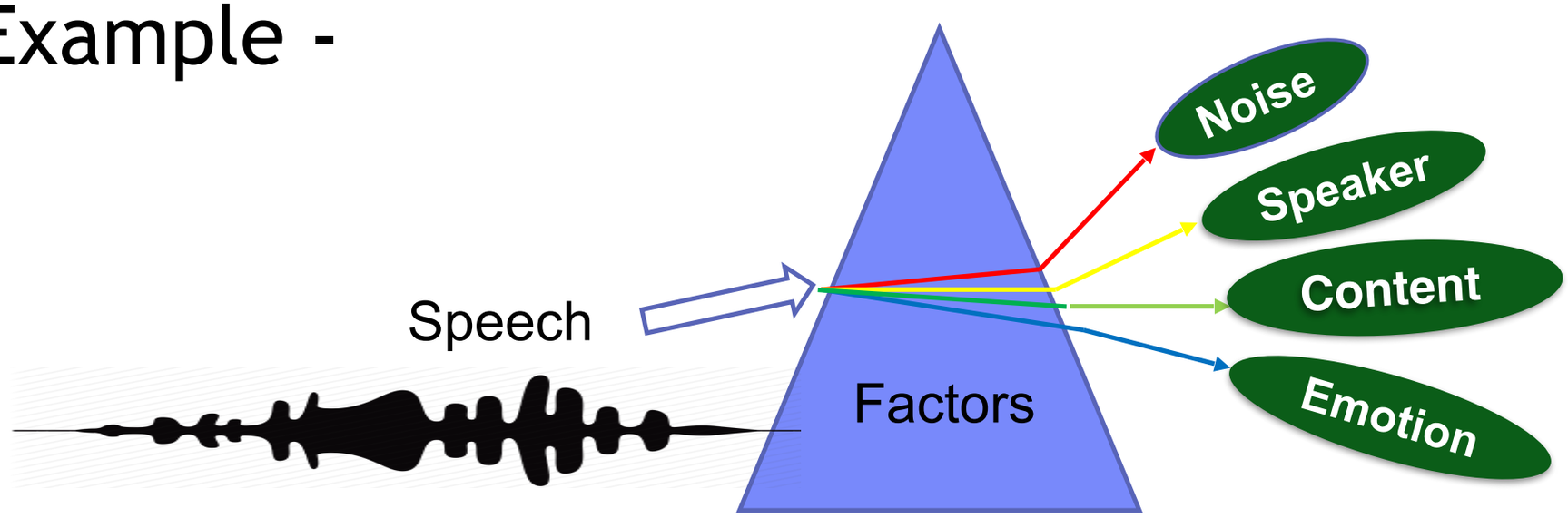
Polar Coordinates

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

Representation Learning in Deep Networks

- The input data representation is one of most important components of any machine learning system.
  - Extract features that enable classification while suppressing factors which are susceptible to noise.
- Finding the right representation for real world applications - substantially challenging.
  - Example -

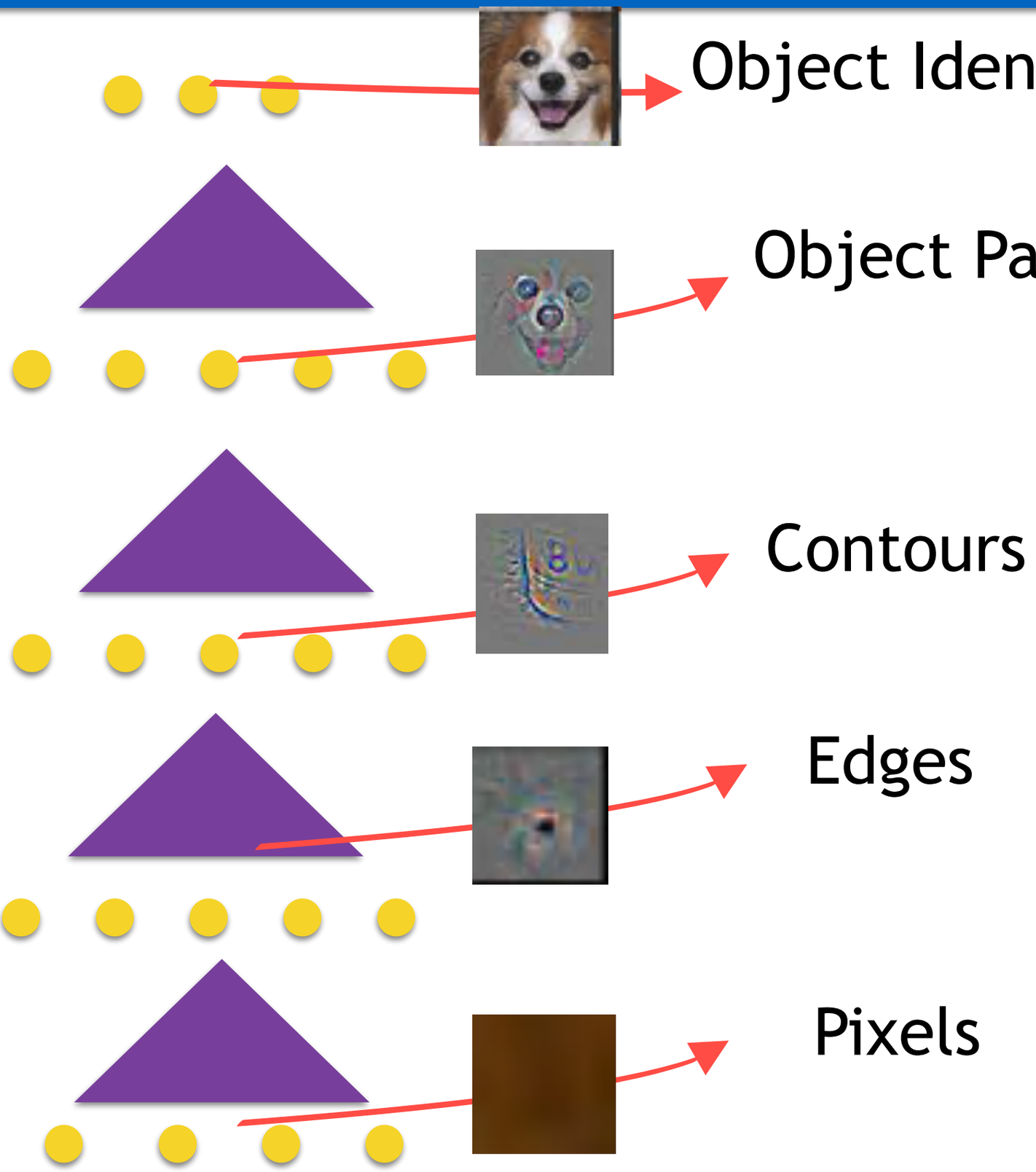


Representation Learning in Deep Networks

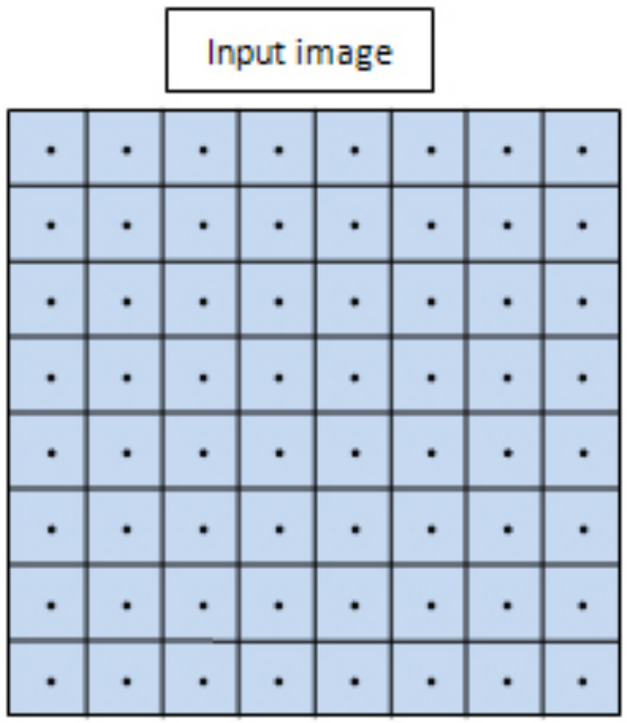
- The input data representation is one of most important components of any machine learning system.
  - Extract factors that enable classification while suppressing factors which are susceptible to noise.
- Finding the right representation for real world applications - substantially challenging.
  - Deep learning solution - build complex representations from simpler representations.
  - The dependencies between these hierarchical representations are refined by the target.

Representation Learning in Deep Networks

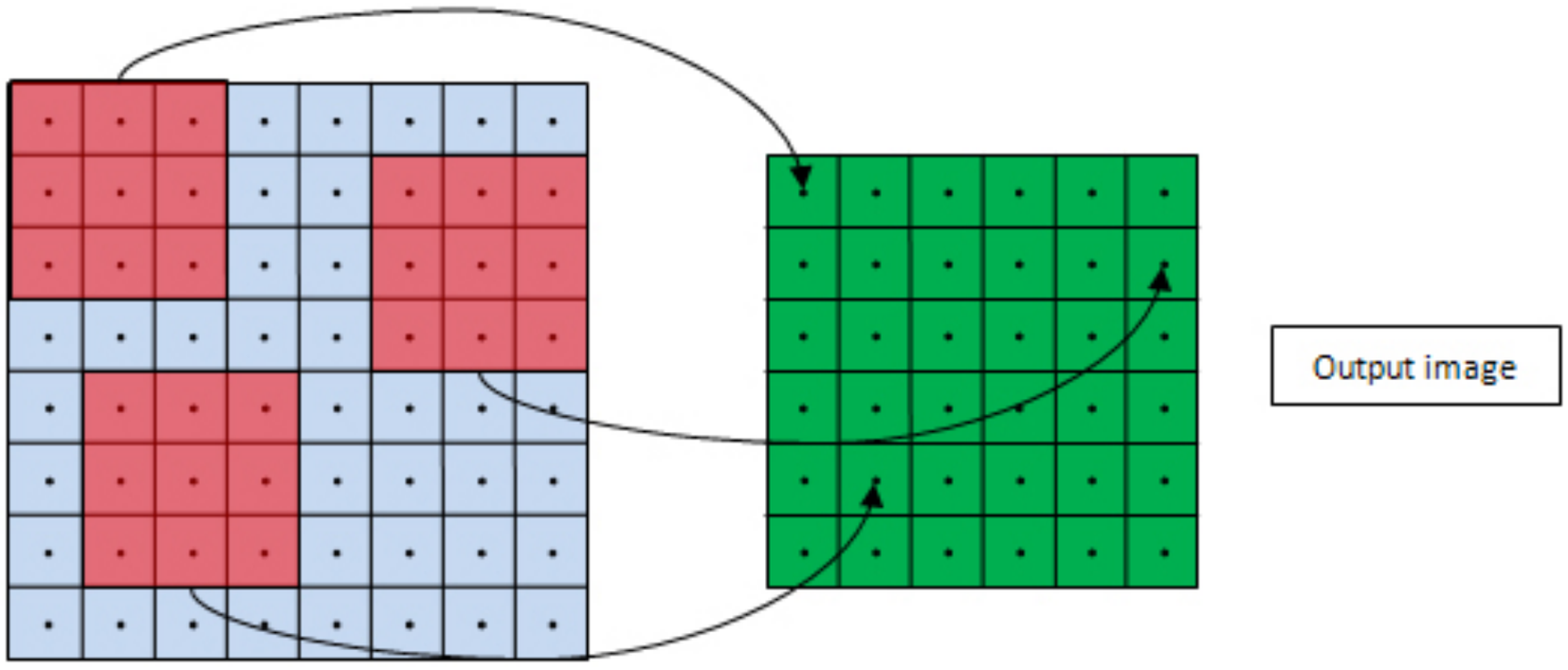
[Zeiler, 2014]



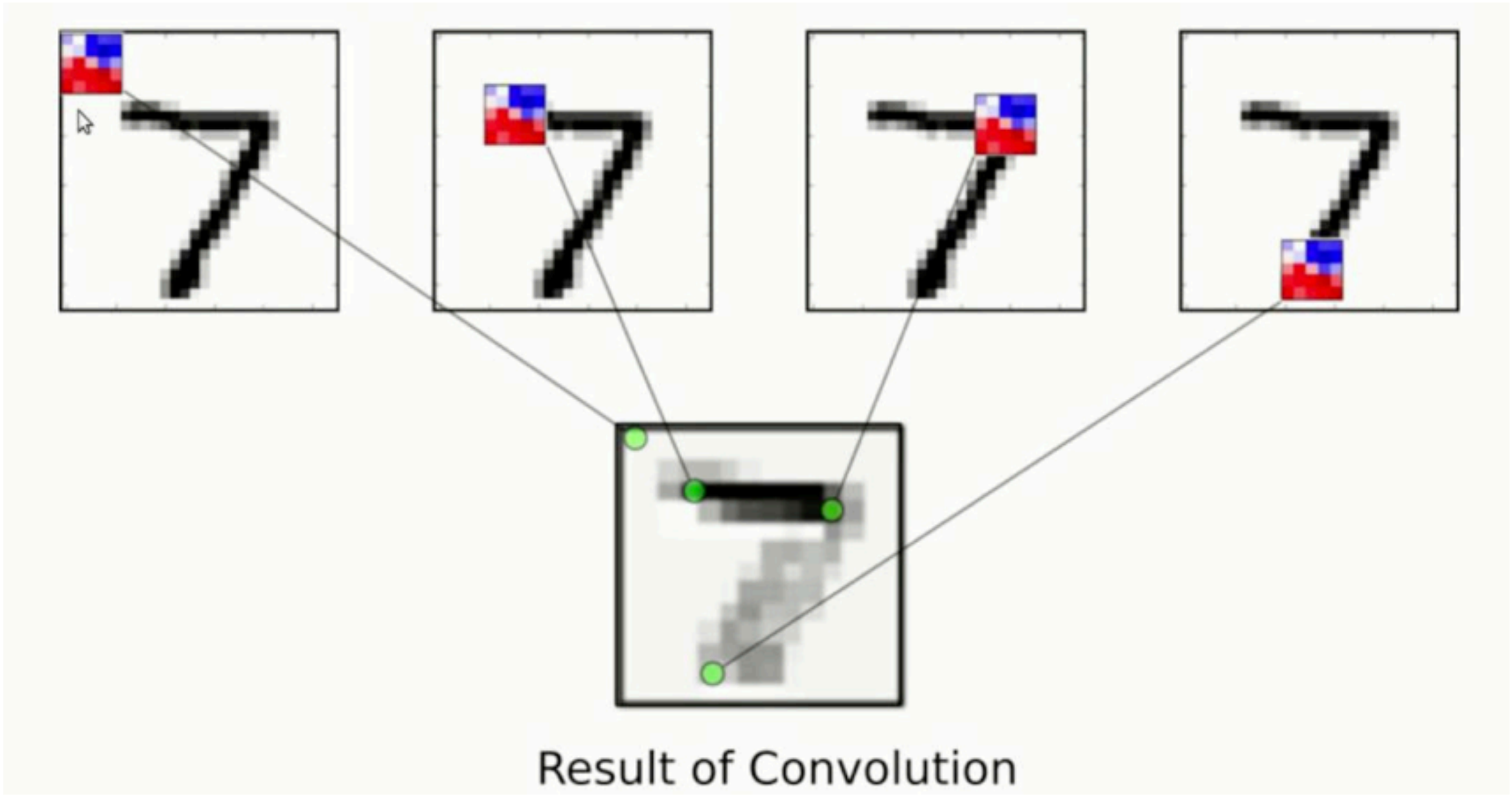
Convolution Operation



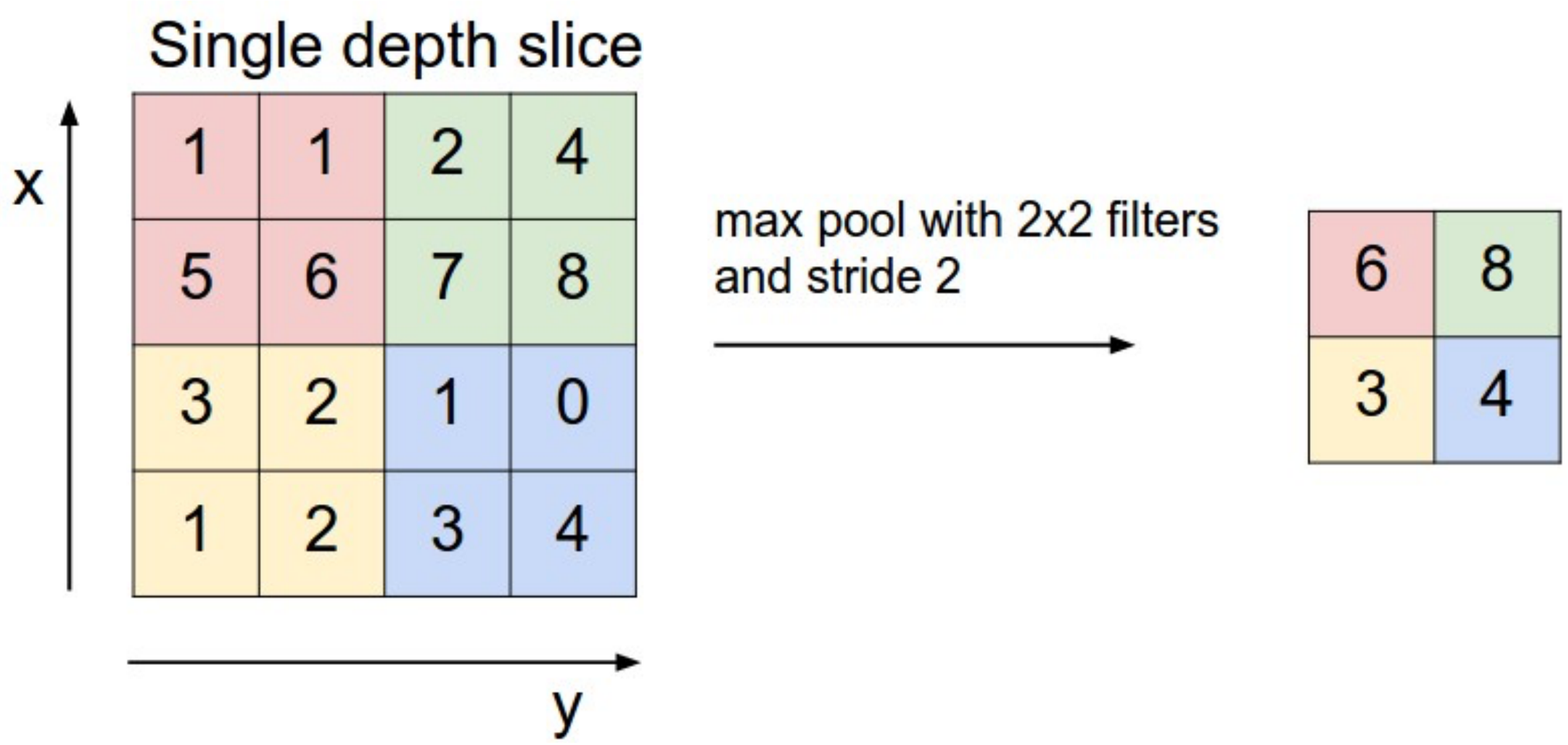
Weight sharing



Convolution Operation

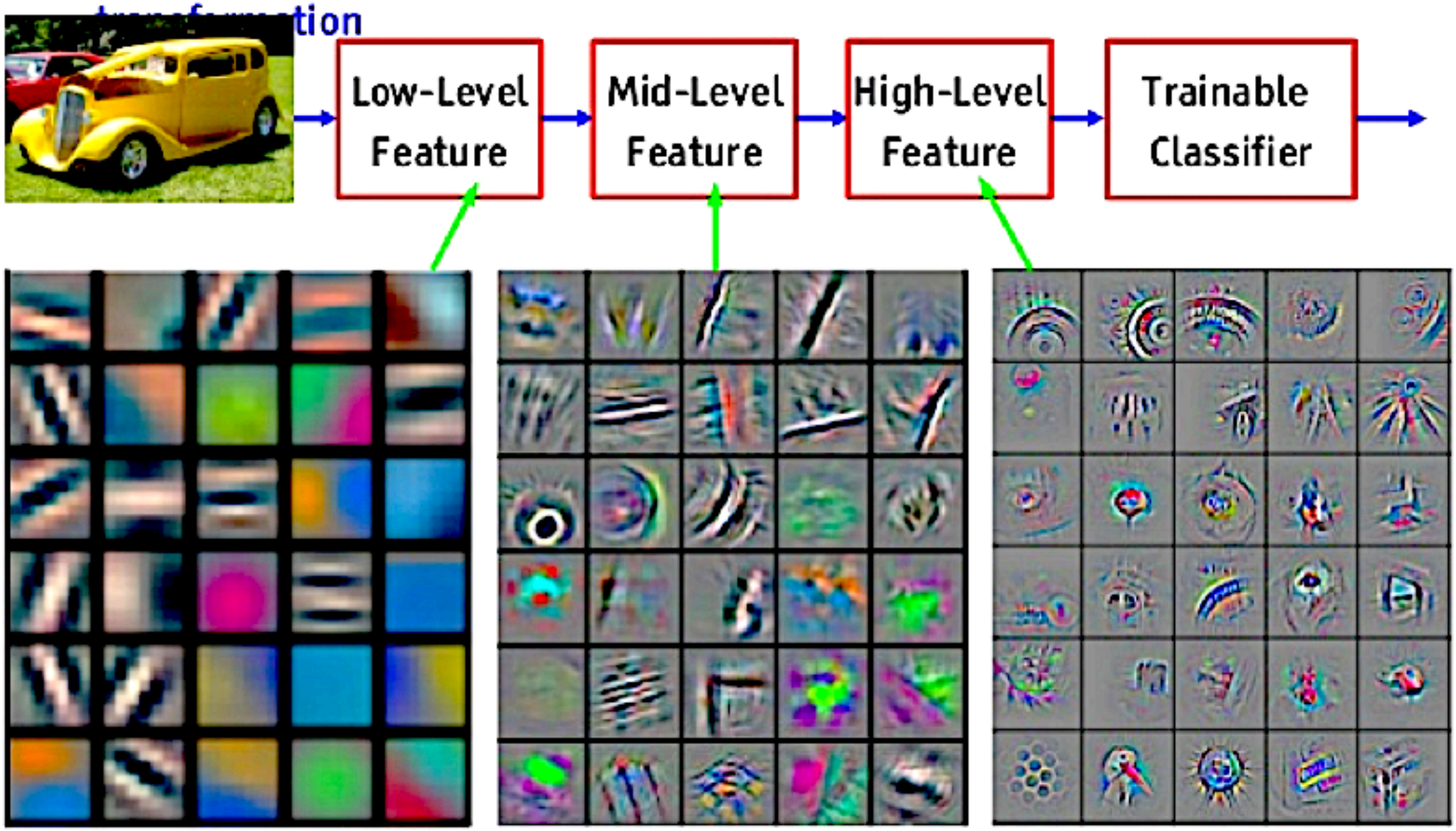


Max Pooling Operation



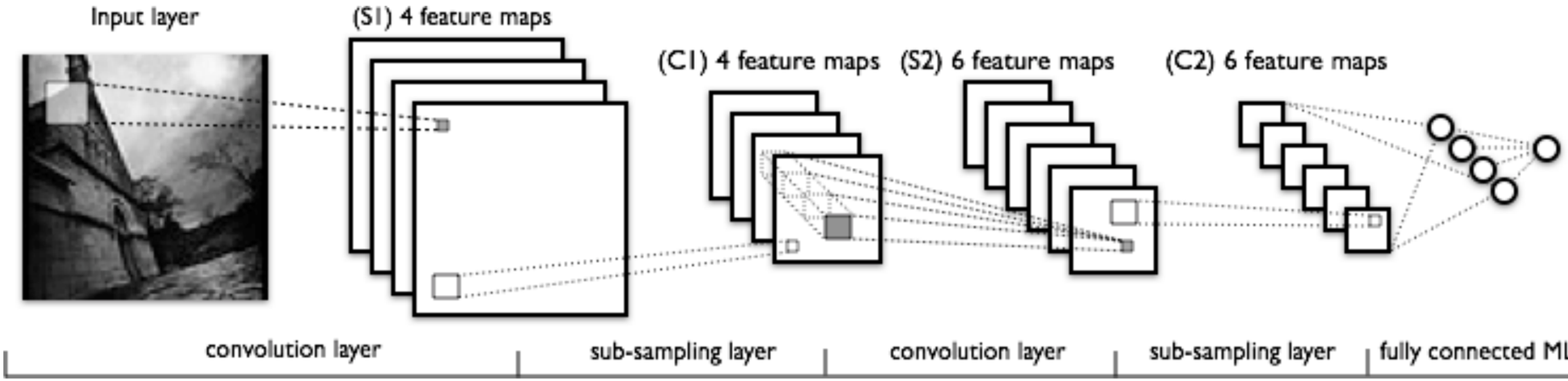
Deep Convolutional Networks

- It's deep if it has more than one stage of non-linear feature transformation



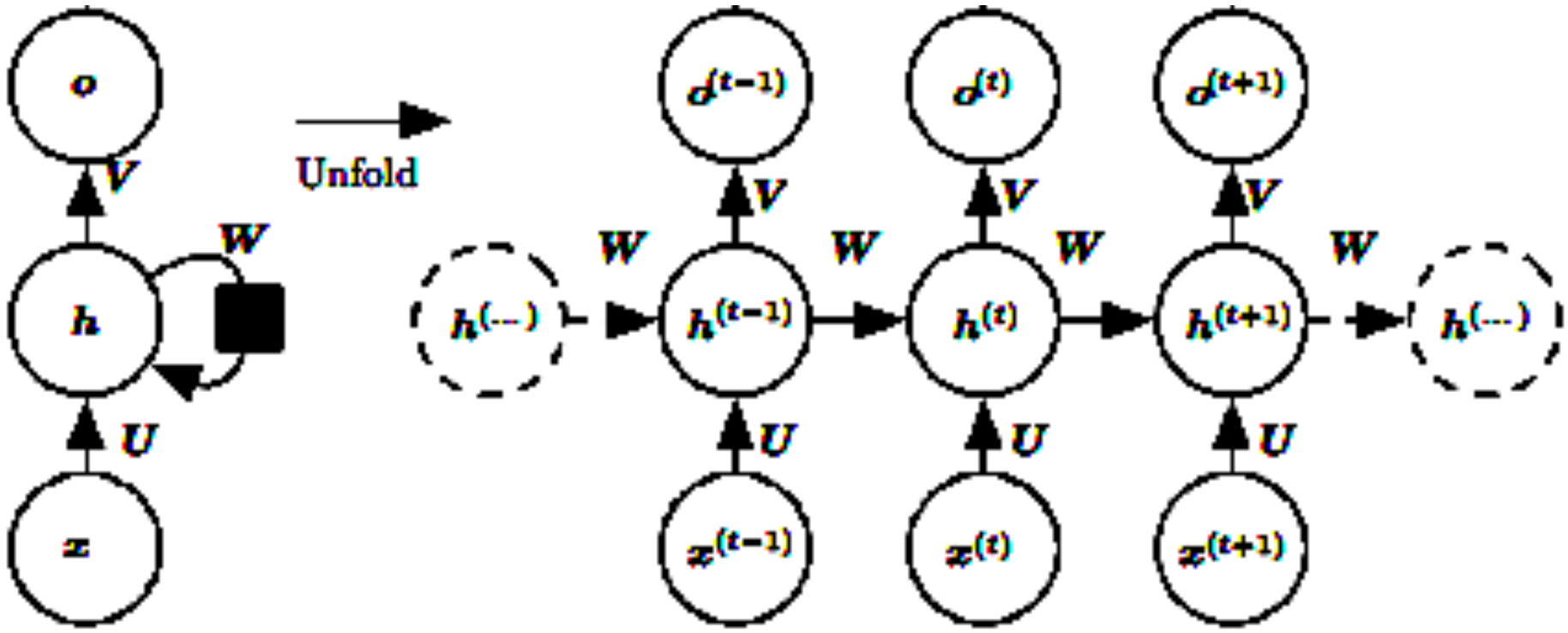
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Deep Convolutional Networks



- Multiple levels of filtering and subsampling operations.
- Feature maps are generated at every layer.

Recurrent Networks

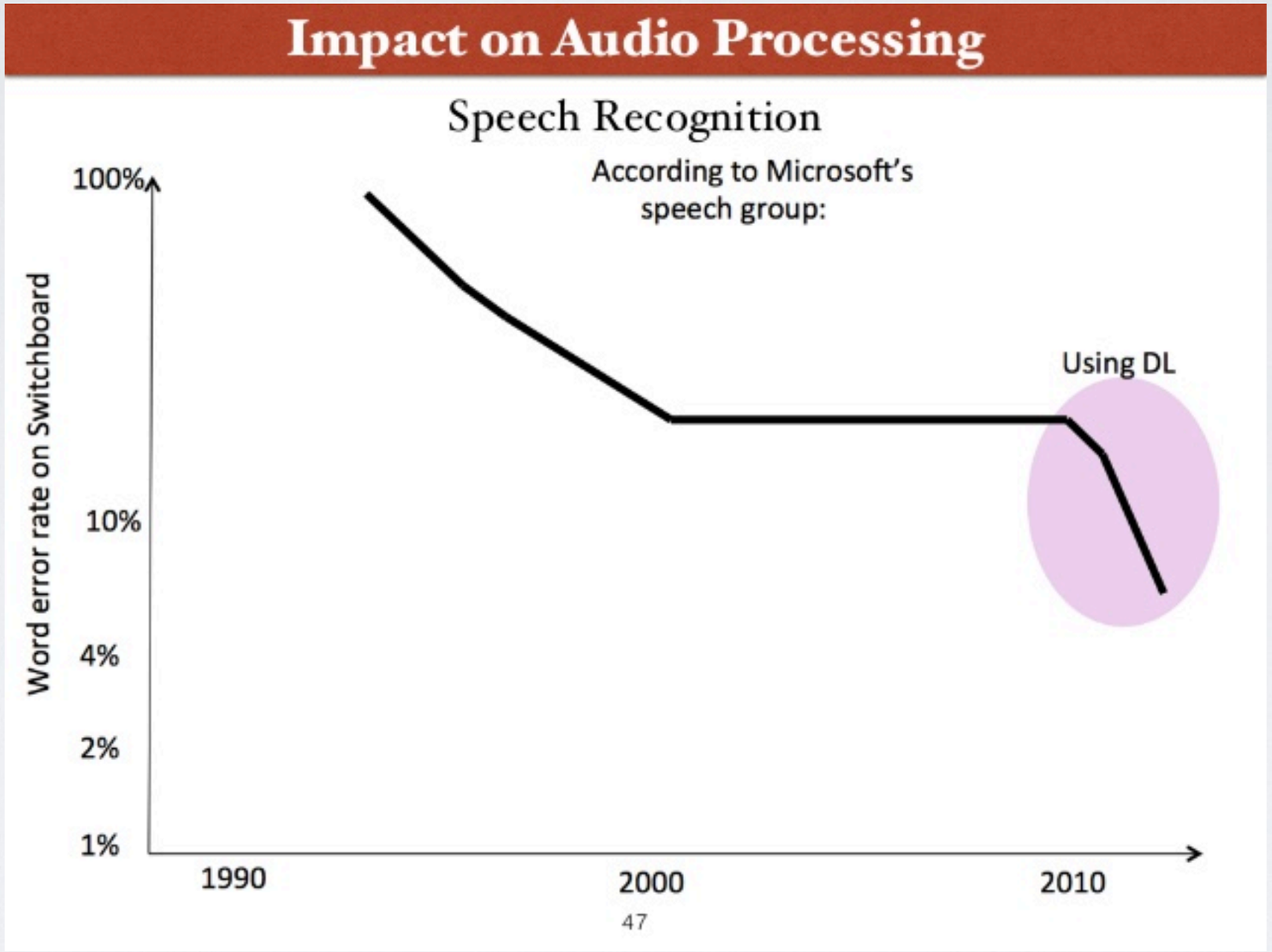


Feedback

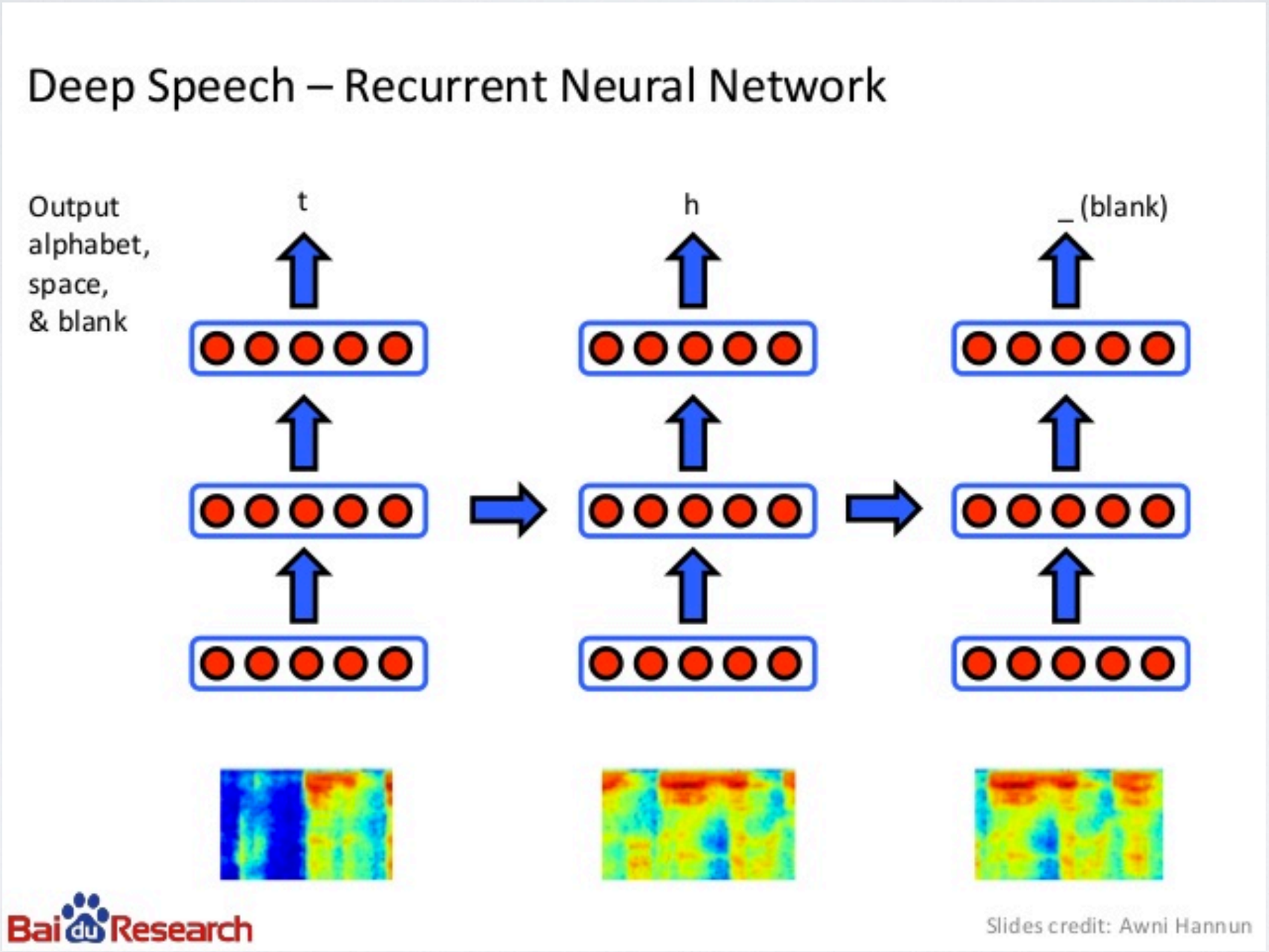
Summary so far…

- Identifying the right representation from data
- Uncovering the representation learnt in hidden layers
- Convolution and max-pooling operation
  - Convolutional neural networks (CNN)
- RNNs and dropouts.

Speech Recognition



Speech Recognition



Open Source Tools for Deep Learning

Theano

http://deeplearning.net/software/theano/

TensorFlow

https://www.tensorflow.org/

Caffe

http://caffe.berkeleyvision.org/

Kaldi

http://kaldi-asr.org/

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान