MACHINE LEARNING FOR SIGNAL PROCESSING

24-2-2025

Sriram Ganapathy
LEAP lab, Electrical Engineering, Indian Institute of Science,
sriramg@iisc.ac.in

Viveka Salinamakki, Varada R.
LEAP lab, Electrical Engineering, Indian Institute of Science

http://leap.ee.iisc.ac.in/sriram/teaching/MLSP25/

STORY SO FAR

EM algorithm

Classification Problem

Decision Theory

Generative Modeling

Gaussian Modeling

Gaussian Mixture Modeling

Function Modeling

Linear Models for Regression and Classification

Kernel Machines & Max-margin classifiers

Support Vector Machines

Data Representations PCA, LDA

Discriminative Modeling

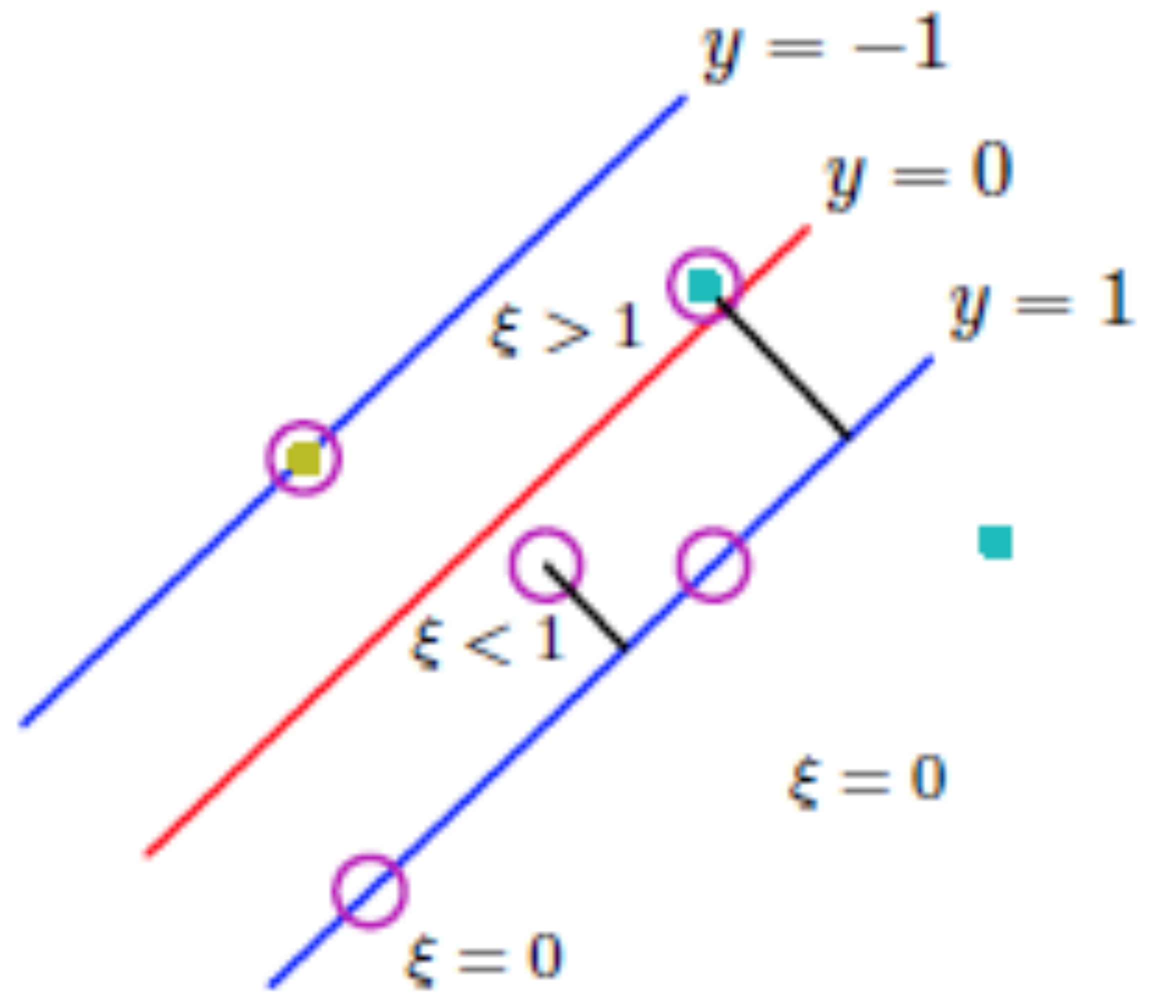
Gradient Descent

Neural Networks

Overlapping class boundaries

- The classes are not linearly separable - Introducing slack variables ζ_n
- Slack variables are non-negative $\zeta_n \geq 0$
- They are defined using

$$t_n y(\mathbf{x}_n) \geq 1 - \zeta_n$$



- The upper bound on mis-classification

$$\sum_n \zeta_n$$

- The cost function to be optimized in this case

$$C \sum_n \zeta_n + \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

SVM Formulation - overlapping classes

- Formulation very similar to previous case except for additional constraints

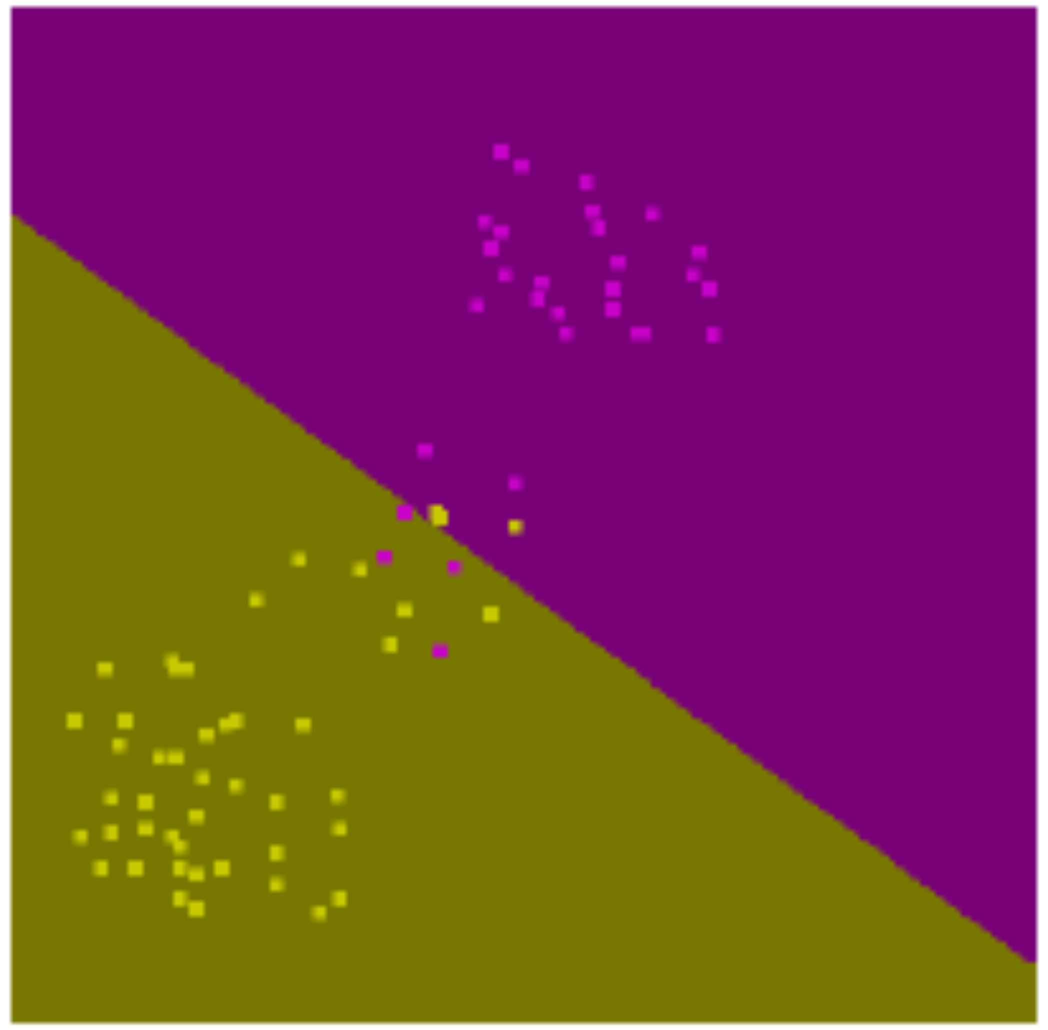
$$0 \leq a_n \leq C$$

- Solved using the dual formulation - sequential minimal optimization algorithm
- Final classifier is based on the sign of

$$y(\mathbf{x}) = \sum_{n \in S} a_n k(\mathbf{x}_n, \mathbf{x}) + b$$

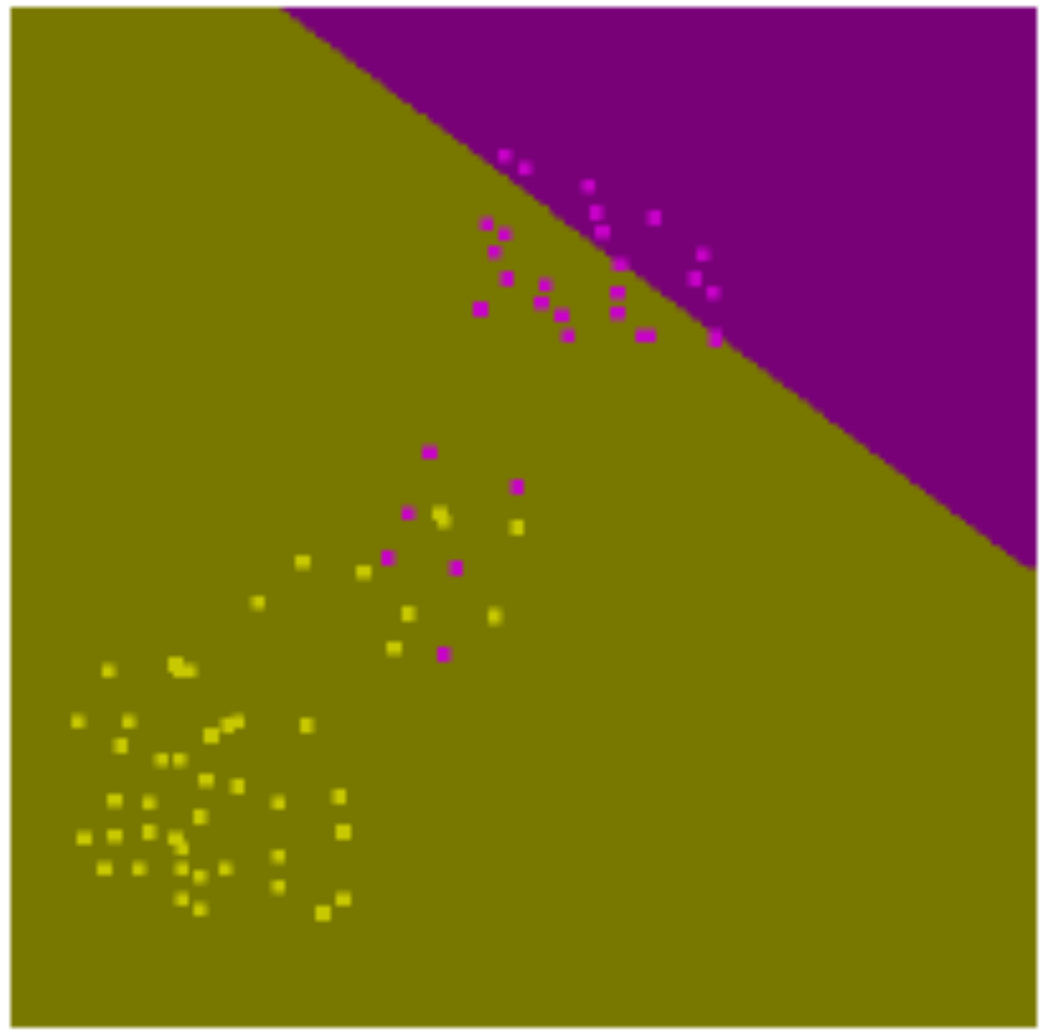
Overlapping class boundaries

C=100



C=1

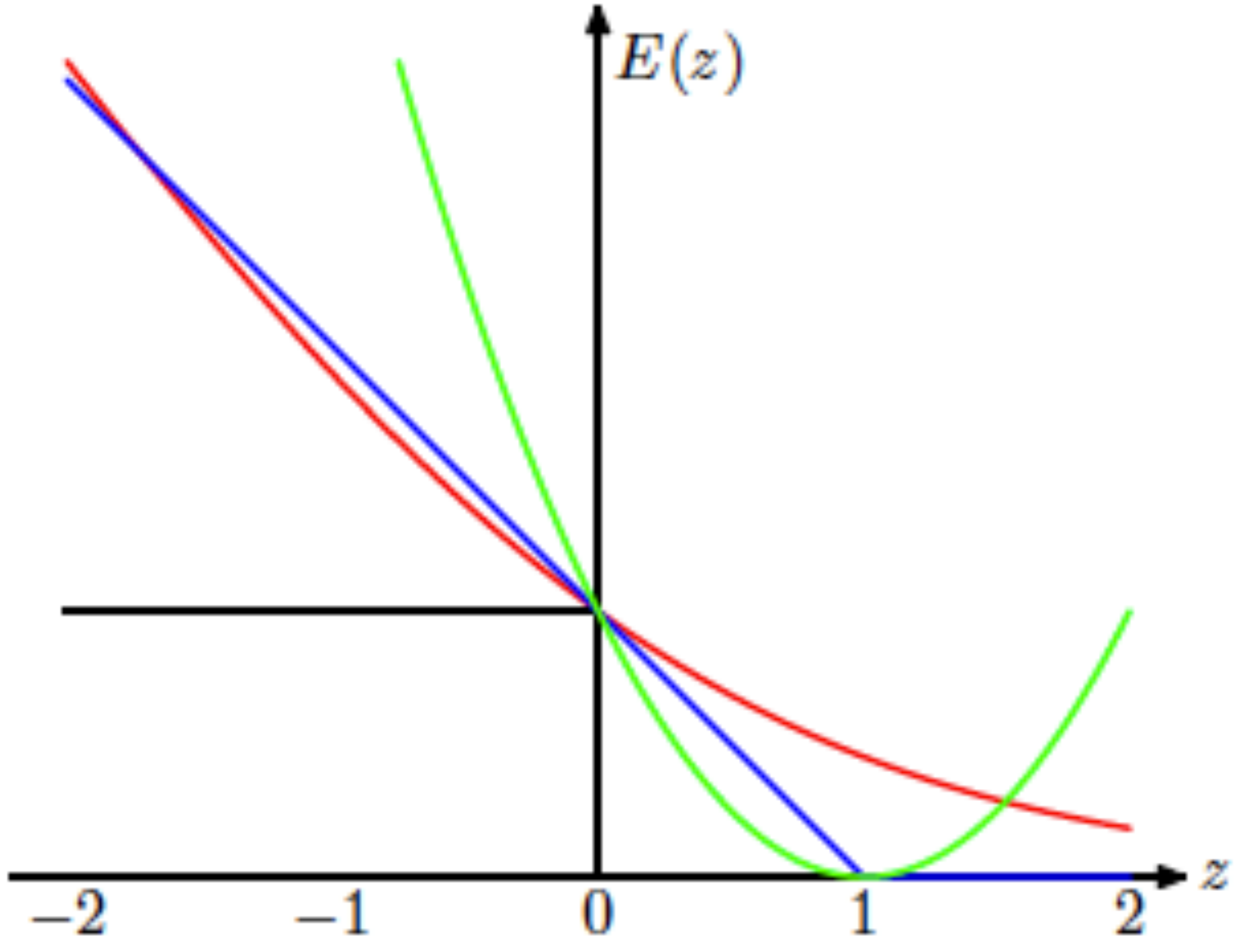
C=0.15



C=0.1

CONNECTION WITH OTHER MODELS

Plot of the 'hinge' error function used in support vector machines, shown in blue, along with the error function for logistic regression, rescaled by a factor of $1/\ln(2)$ so that it passes through the point $(0, 1)$, shown in red. Also shown are the misclassification error in black and the squared error in green.



SVM Applications

- SVM has been used successfully in many real-world problems

  - text (and hypertext) categorization

  - image classification

  - bioinformatics (Protein classification,
    Cancer classification)

  - hand-written character recognition

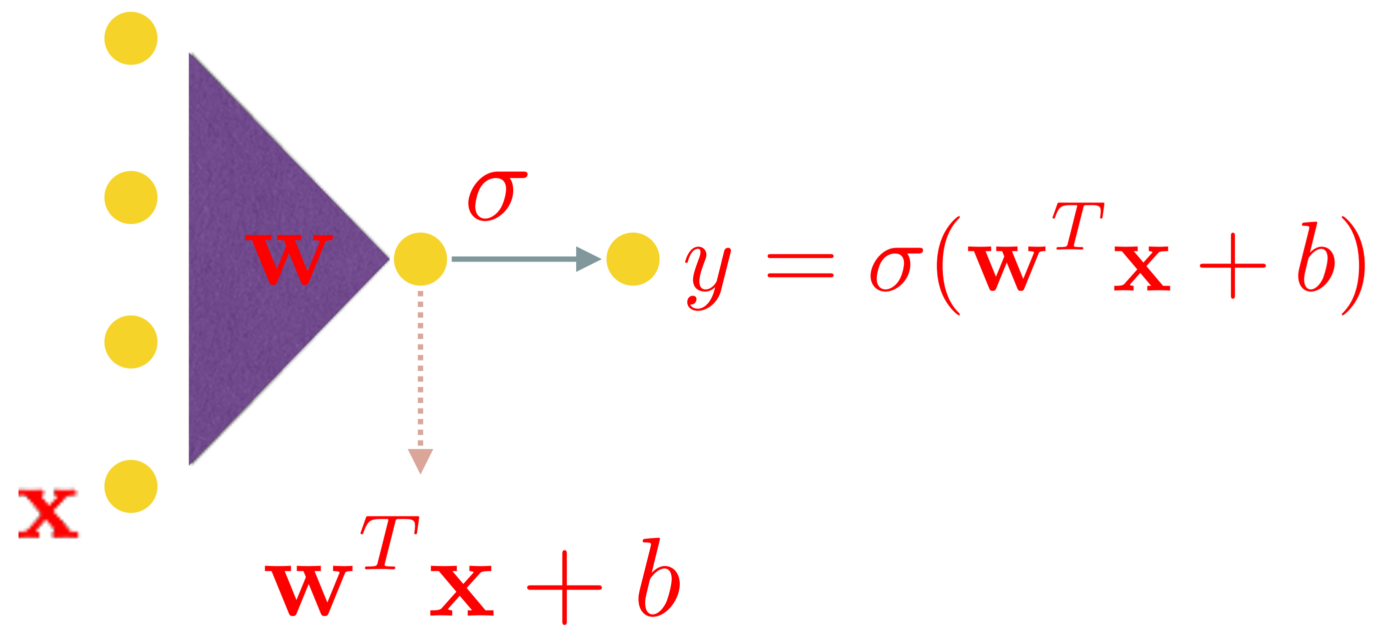
NEURAL NETWORKS AND DEEP LEARNING

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

LEAP

VISUALIZING LOGISTIC REGRESSION AS A NEURAL NETWORK

- A logistic regression is the simplest neural network
  - Number of parameters in the model - D+1

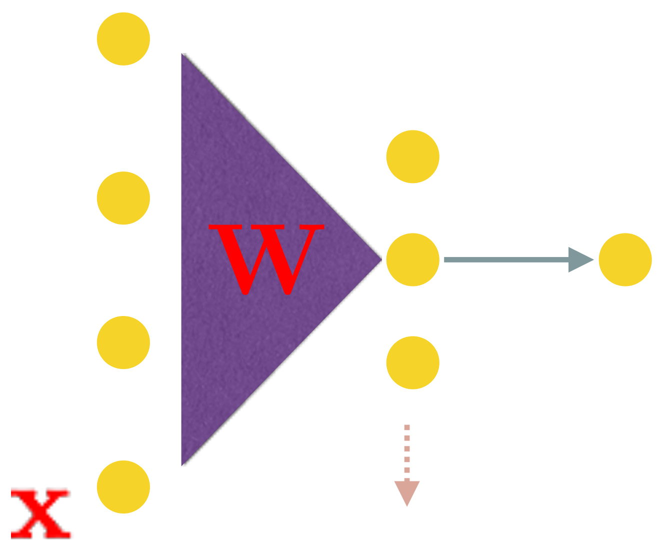


MULTI-CLASS LOGISTIC REGRESSION

- Targets are one-hot encoded vectors
  - Model approximates class posteriors using
    - softmax function

$$\begin{bmatrix} 0 \\ 0 \\ . \\ 0 \\ 1 \\ 0 \\ . \\ 0 \end{bmatrix}$$

$$softmax(\mathbf{a}) = \begin{bmatrix} \frac{e^{a_1}}{\sum_j e^{a_j}} \\ \frac{e^{a_2}}{\sum_j e^{a_j}} \\ . \\ . \\ \frac{e^{a_K}}{\sum_j e^{a_j}} \end{bmatrix}$$



$$\mathbf{y} = softmax(\mathbf{W}^T\mathbf{x} + \mathbf{b})$$

$$\mathbf{Wx} + \mathbf{b} \qquad \mathbf{W} \in \mathcal{R}^{K \times D} \quad \mathbf{b} \in \mathcal{R}^K$$

SOFTMAX FUNCTION

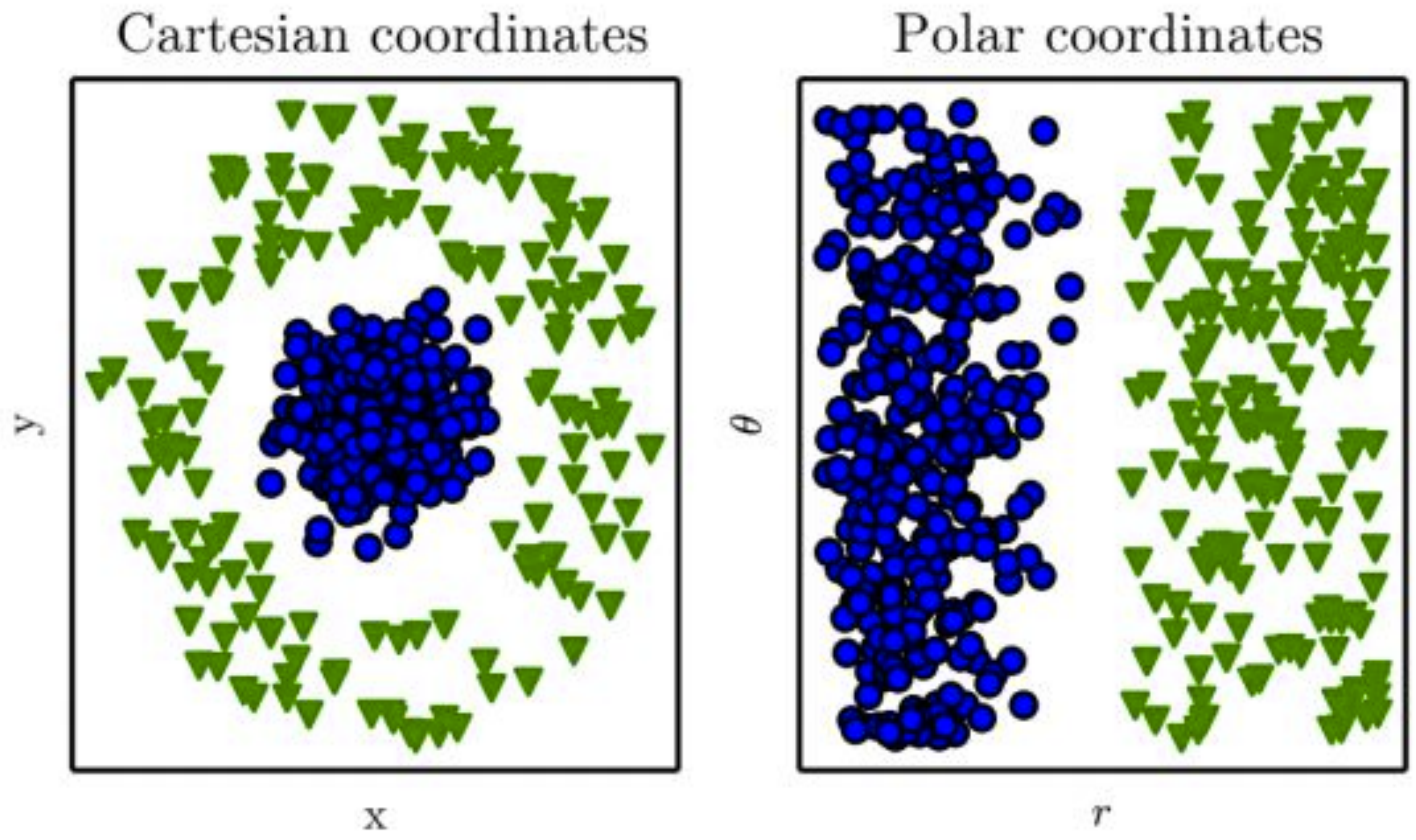
- Each value is positive
- Sum of the vector is 1.0
  - Can be interpreted as class posterior probabilities

$$\begin{bmatrix} p(C_1|\mathbf{x}) \\ p(C_2|\mathbf{x}) \\ . \\ . \\ p(C_K|\mathbf{x}) \end{bmatrix}$$

$$softmax(\mathbf{a}) = \begin{bmatrix} \frac{e^{a_1}}{\sum_j e^{a_j}} \\ \frac{e^{a_2}}{\sum_j e^{a_j}} \\ . \\ . \\ \frac{e^{a_K}}{\sum_j e^{a_j}} \end{bmatrix}$$

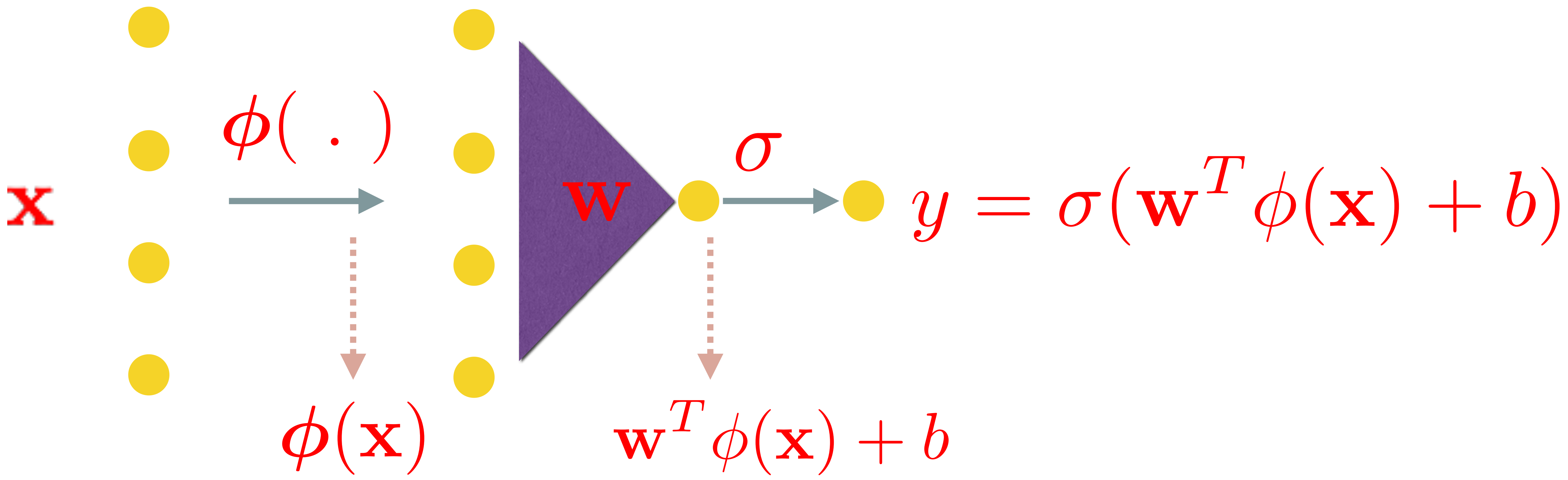
QUESTION

- **Can we transform the data to linearly separable space**
  - then apply the logistic regression to find the classifier.
    - Example



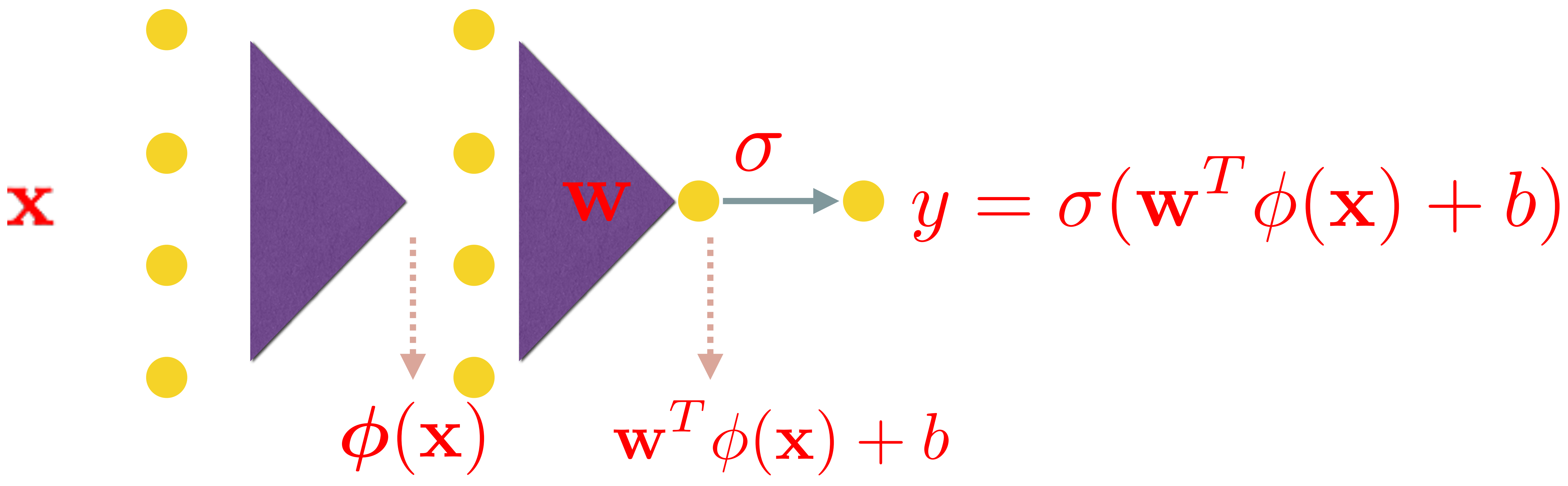
QUESTION

- **Can we transform the data to linearly separable space**
  - then apply the logistic regression to find the classifier.
- Can we learn such a transform from the data itself
  - non-linear transformation of the data is needed



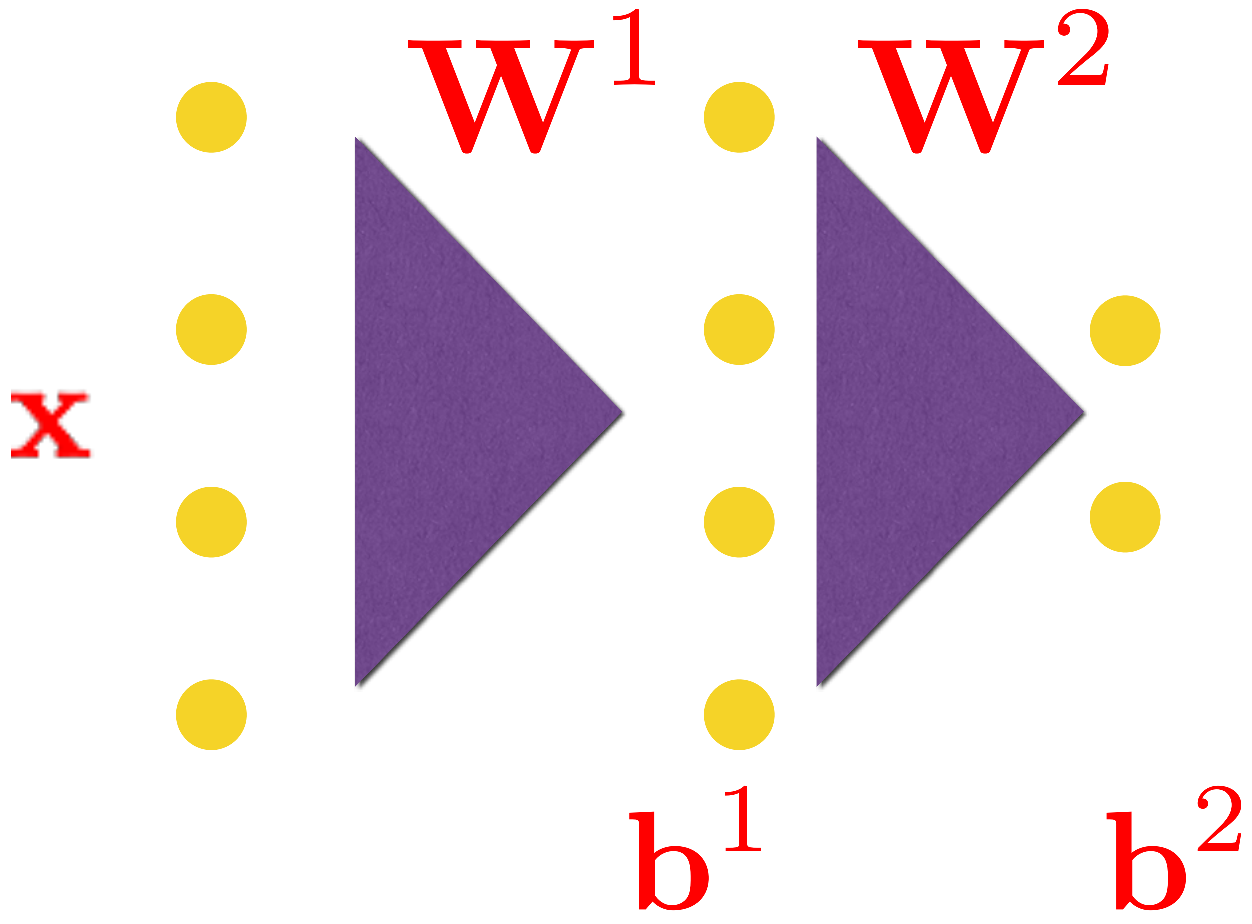
QUESTION

- **Can we transform the data to linearly separable space**
  - then apply the logistic regression to find the classifier.
- **Can we learn such a transform from the data itself**
  - non-linear transformation of the data is needed.
  - can this also be realized as neural layer



NEURAL NETWORK - 1- HIDDEN LAYER

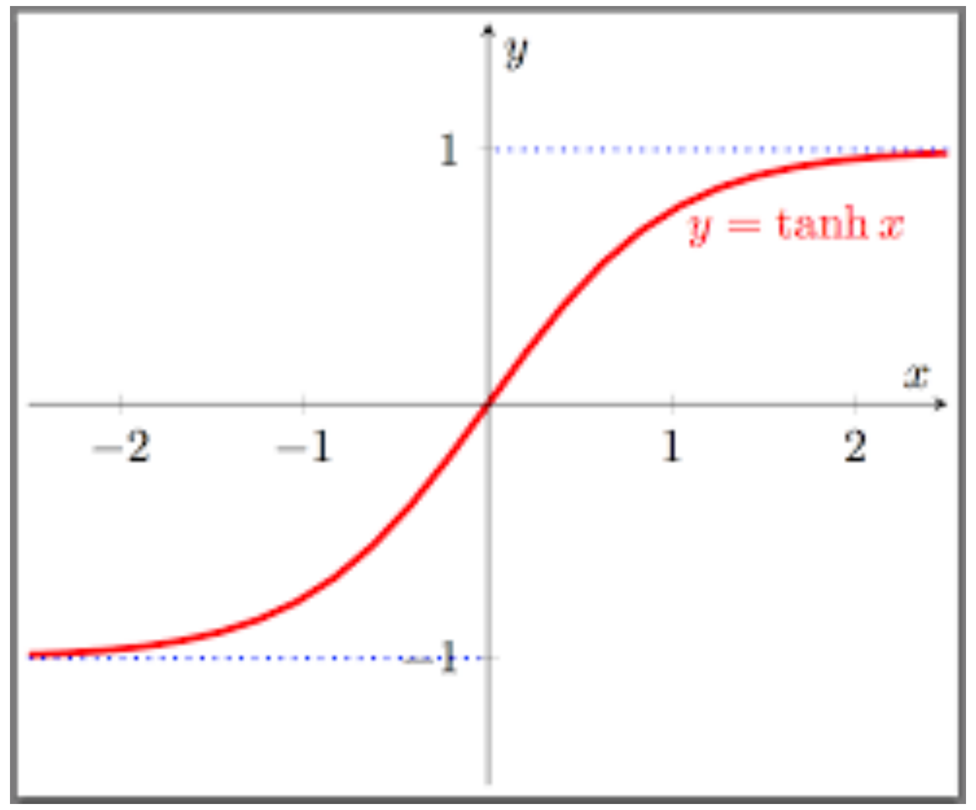
- Has more capacity than logistic regression
  - can learn non-linear data separation functions
  - both 2-class and K-class classification possible
  - can be learnt using gradient descent



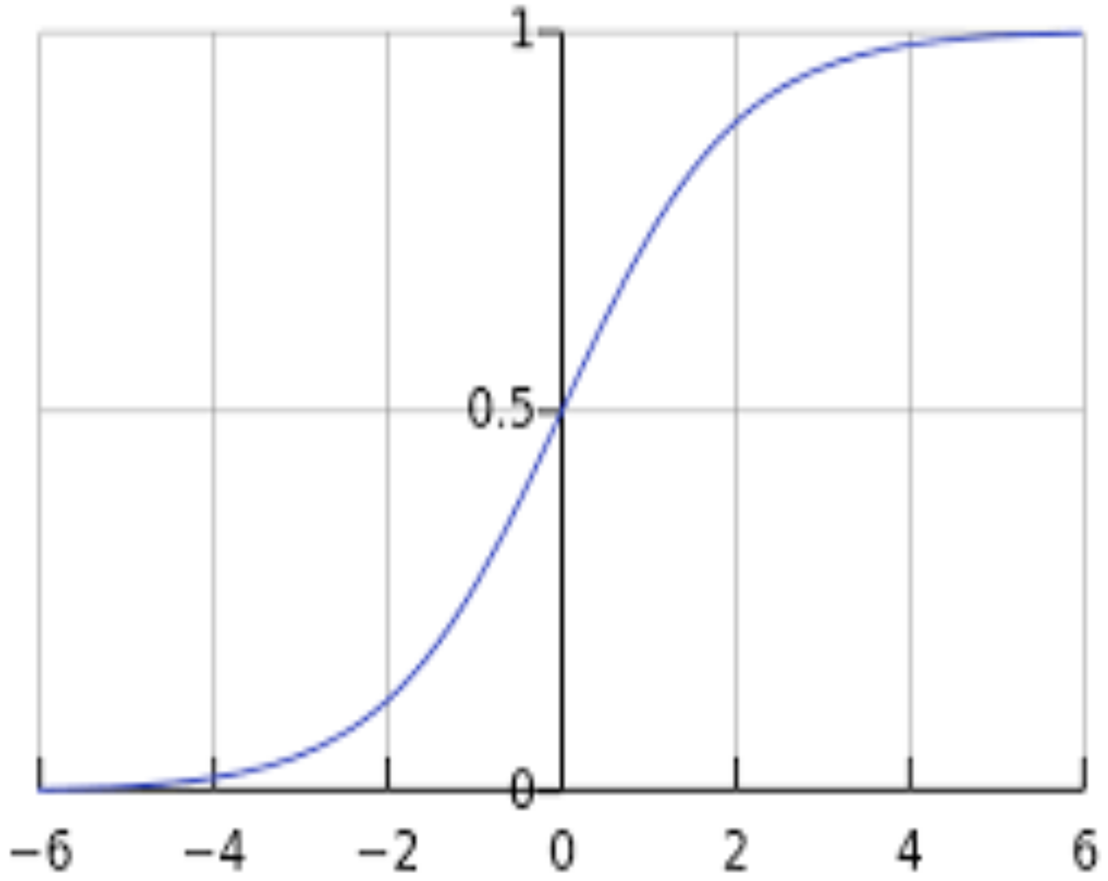
TYPES OF NON-LINEARITIES

Non-linearity in hidden layer

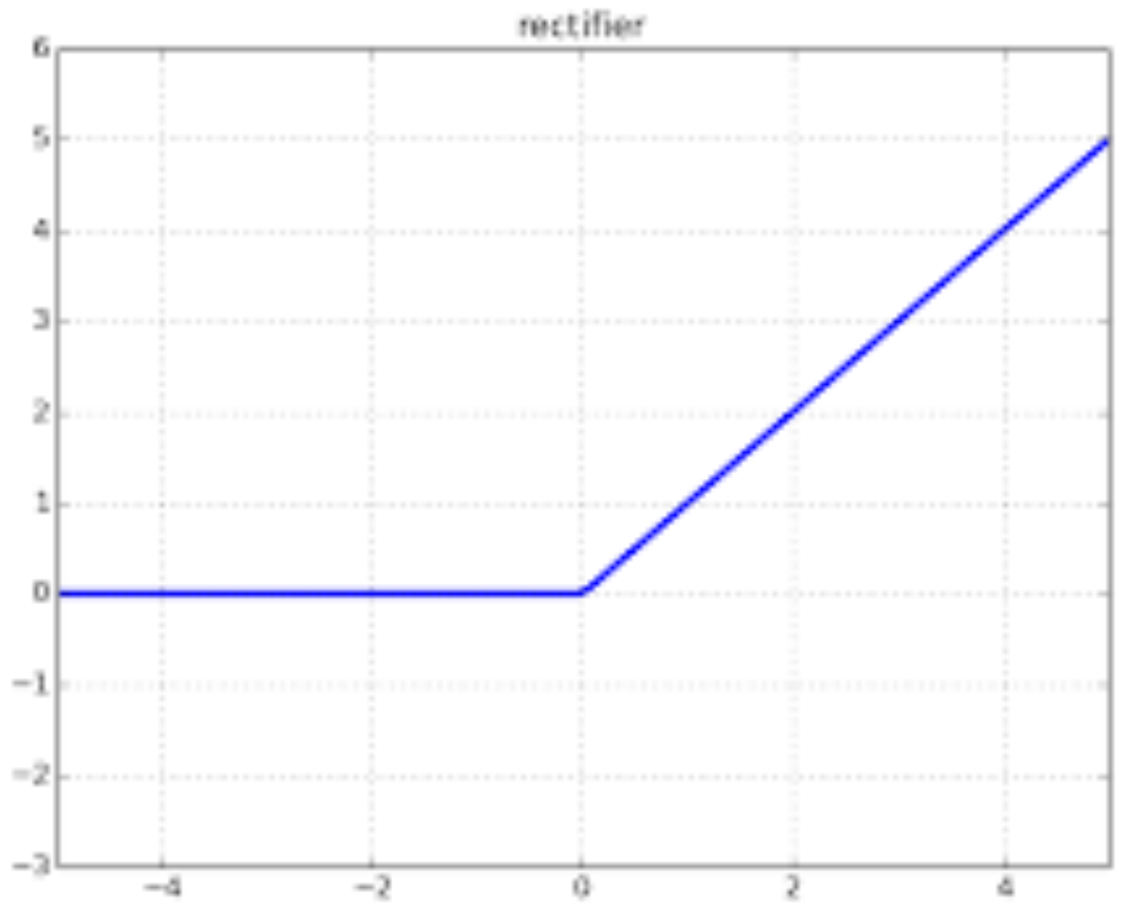
tanh



sigmoid



ReLu



OUTPUT LAYER NON-LINEARITY AND COST FUNCTIONS

- Using a softmax non-linearity
  - error function is cross entropy

$$E_{CE} = -\sum_{n}\sum_{k} t_{nk}\; log(v_{nk})$$

- For regression style tasks - output is linear
  - error function is mean square error

$$E_{MSE} = -\sum_{n}\sum_{k}(t_{nk} - v_{nk})^2$$

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

LEAP

FORWARD THROUGH THE MODEL PROPAGATION LEARNING

- Computations in the forward direction

$$\mathbf{a}^1 = \mathbf{W}^1\mathbf{x} + \mathbf{b}^1$$
$$\mathbf{z}^1 = \sigma(\mathbf{a}^1)$$
$$\mathbf{a}^2 = \mathbf{W}^2\mathbf{z}^1 + \mathbf{b}^2$$
$$\mathbf{y} = softmax(\mathbf{a}^2)$$

- Loss function

$$E_{CE} = -\sum_n \sum_k t_{nk} \, log(v_{nk})$$

$$\mathbf{\Theta} = \{\mathbf{W}^1 \, , \, \mathbf{b}^1 \, , \, \mathbf{W}^2 \, , \, \mathbf{b}^2\}$$

- Parameters in the model

Need to be updated based on the gradients w.r.t. the error

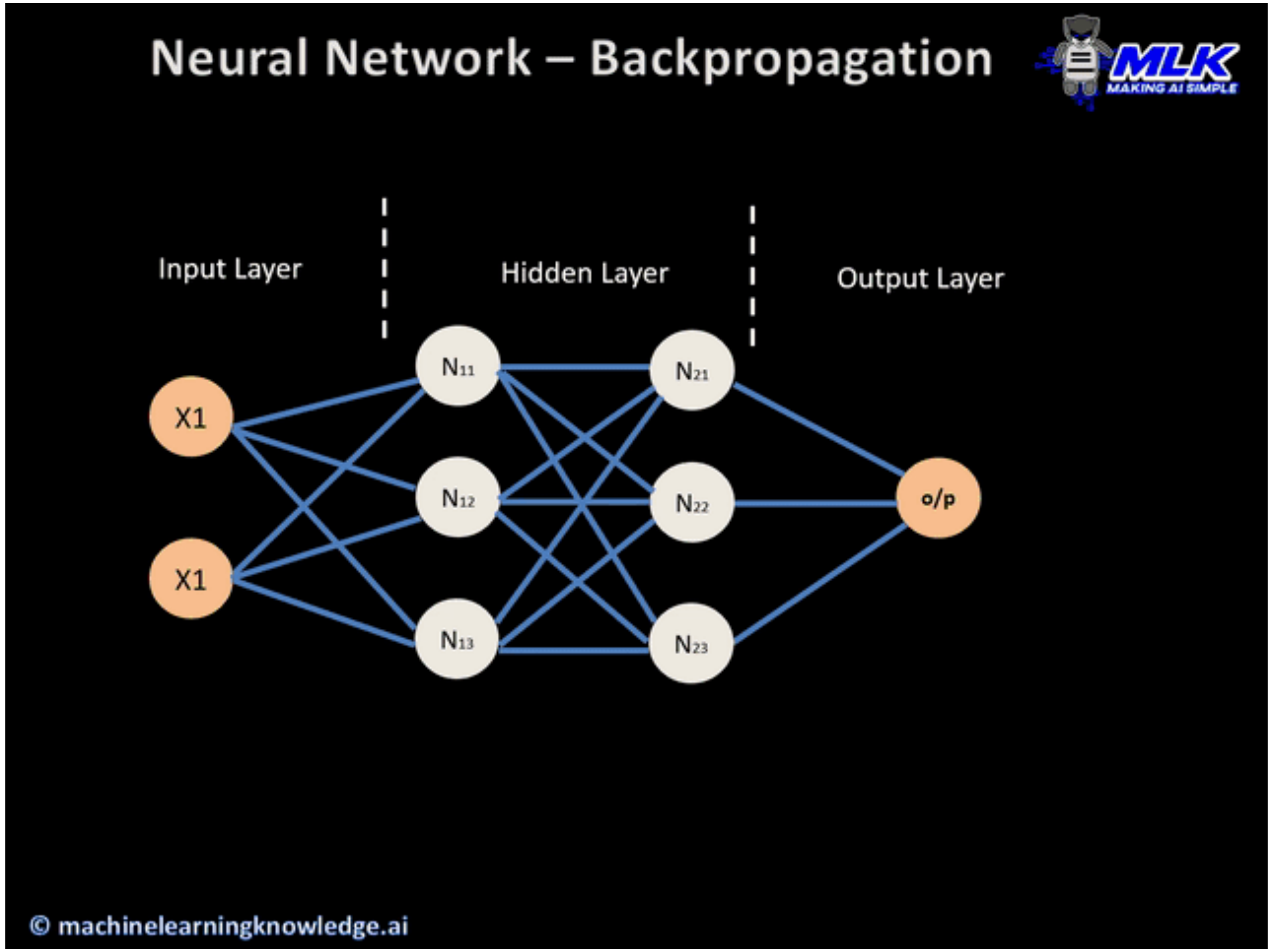
Gradient Computation in the Model

$$\mathbf{a}^1 = \mathbf{W}^1\mathbf{x} + \mathbf{b}^1$$
$$\mathbf{z}^1 = \sigma(\mathbf{a}^1)$$
$$\mathbf{a}^2 = \mathbf{W}^2\mathbf{z}^1 + \mathbf{b}^2$$
$$\mathbf{y} = softmax(\mathbf{a}^2)$$

$$E_{CE} = -\sum_n \sum_k t_{nk}\ log(v_{nk})$$

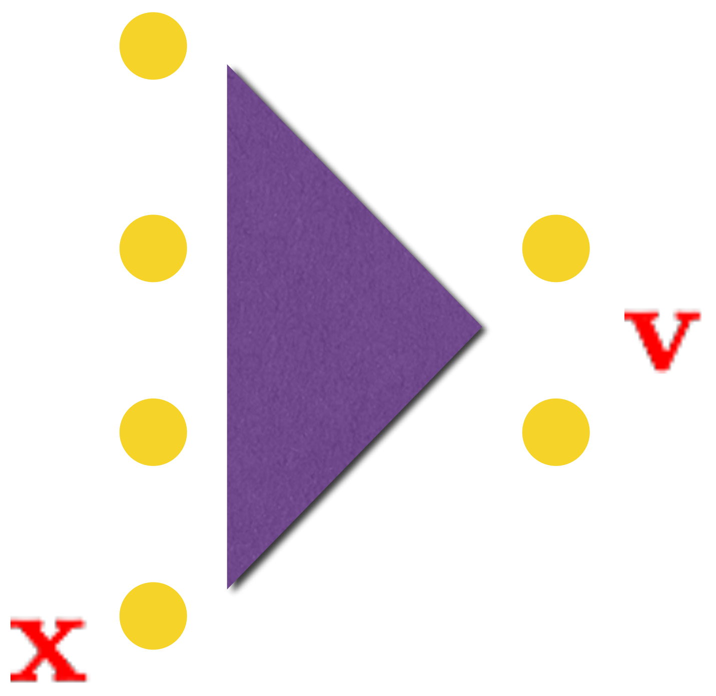
- When computing the gradients
  - Order of computations
    - The derivative of the loss function w.r.t output layer
    - The derivative of the loss function w.r.t output activation
    - The derivative of the loss function w.r.t hidden layer outputs
    - The derivative of the loss function w.r.t. hidden layer activations

BACK PROPAGATION LEARNING



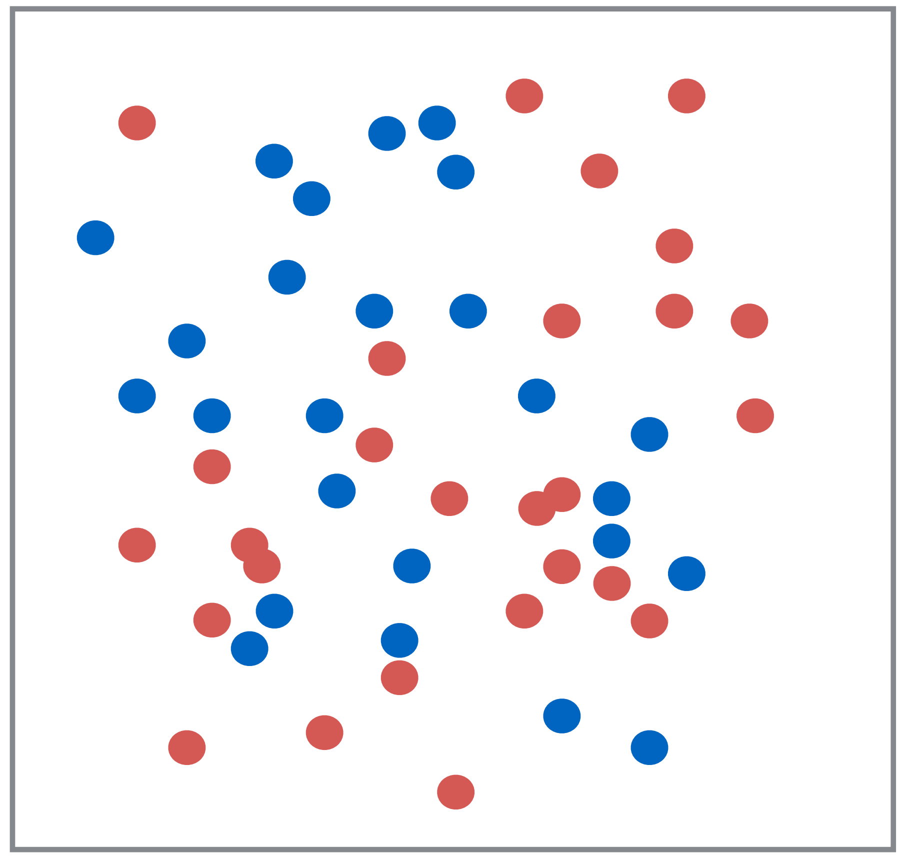
PERCEPTRON ALGORITHM

Perceptron Model [McCulloch, 1943, Rosenblatt, 1957]



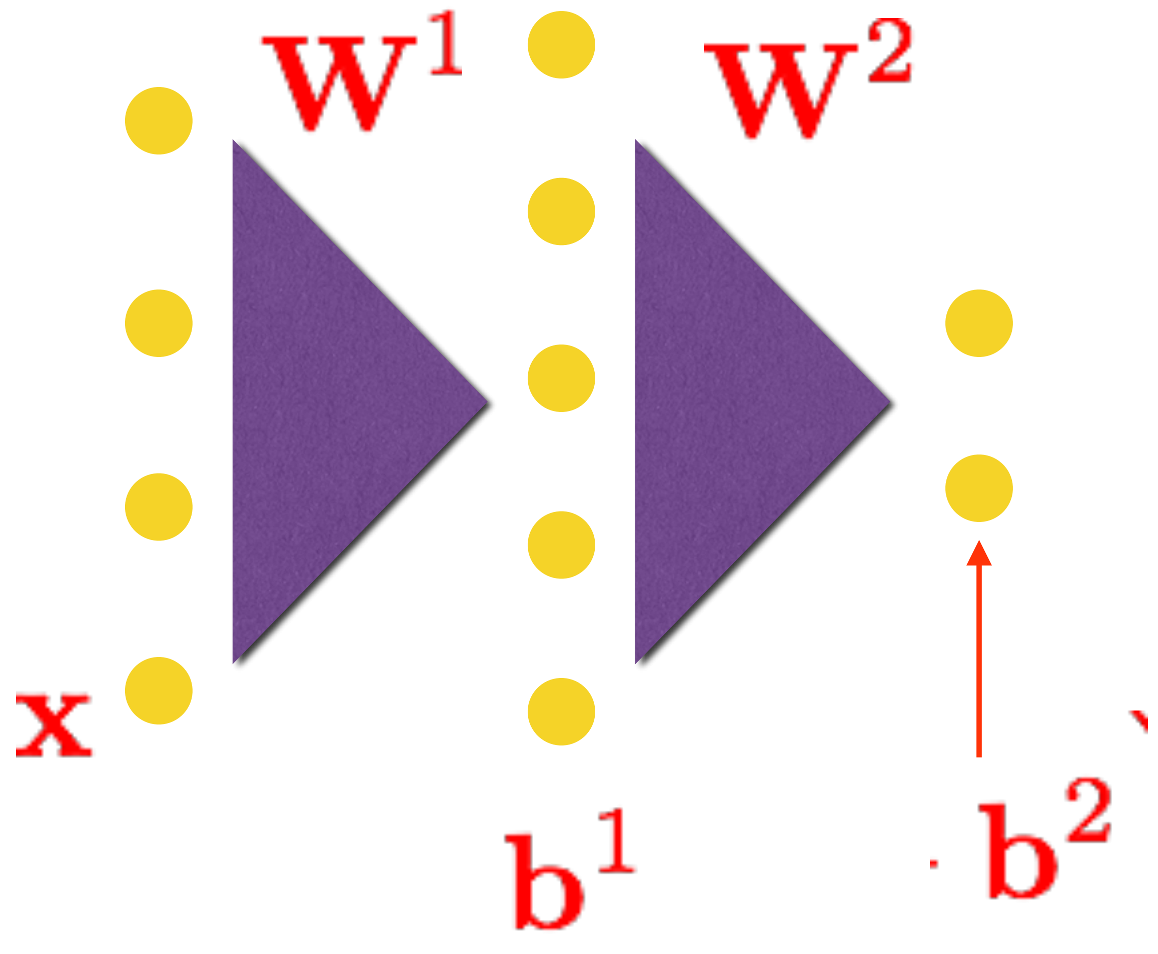
Targets are binary classes [-1,1]

What if the data is not linearly separable



MULTI-LAYER PERCEPTRON

Multi-layer Perceptron [Hopfield, 1982]



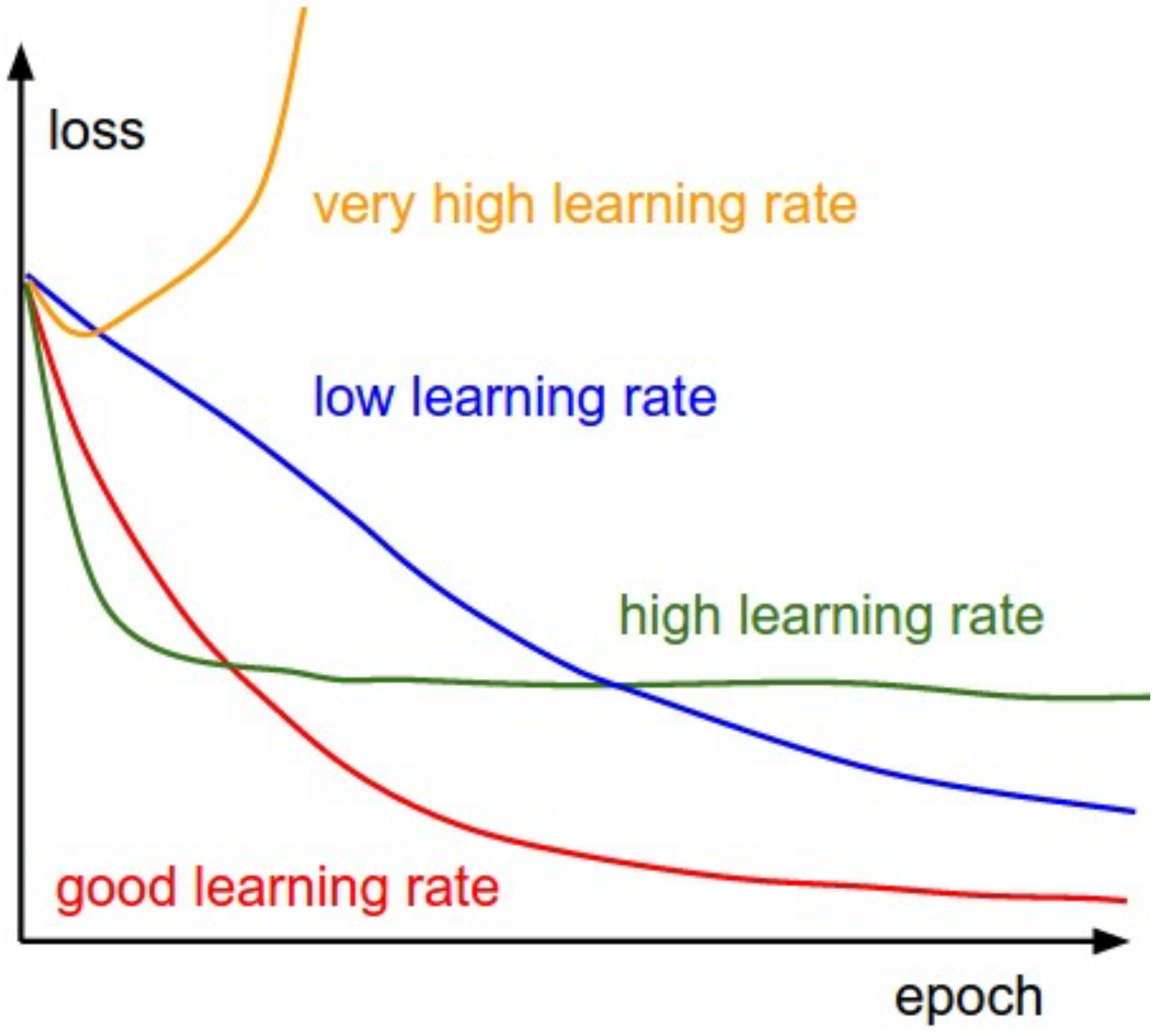
$$\mathbf{v}^2 = \psi\left(\mathbf{W}^2\phi(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1) + \mathbf{b}^2\right)$$

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

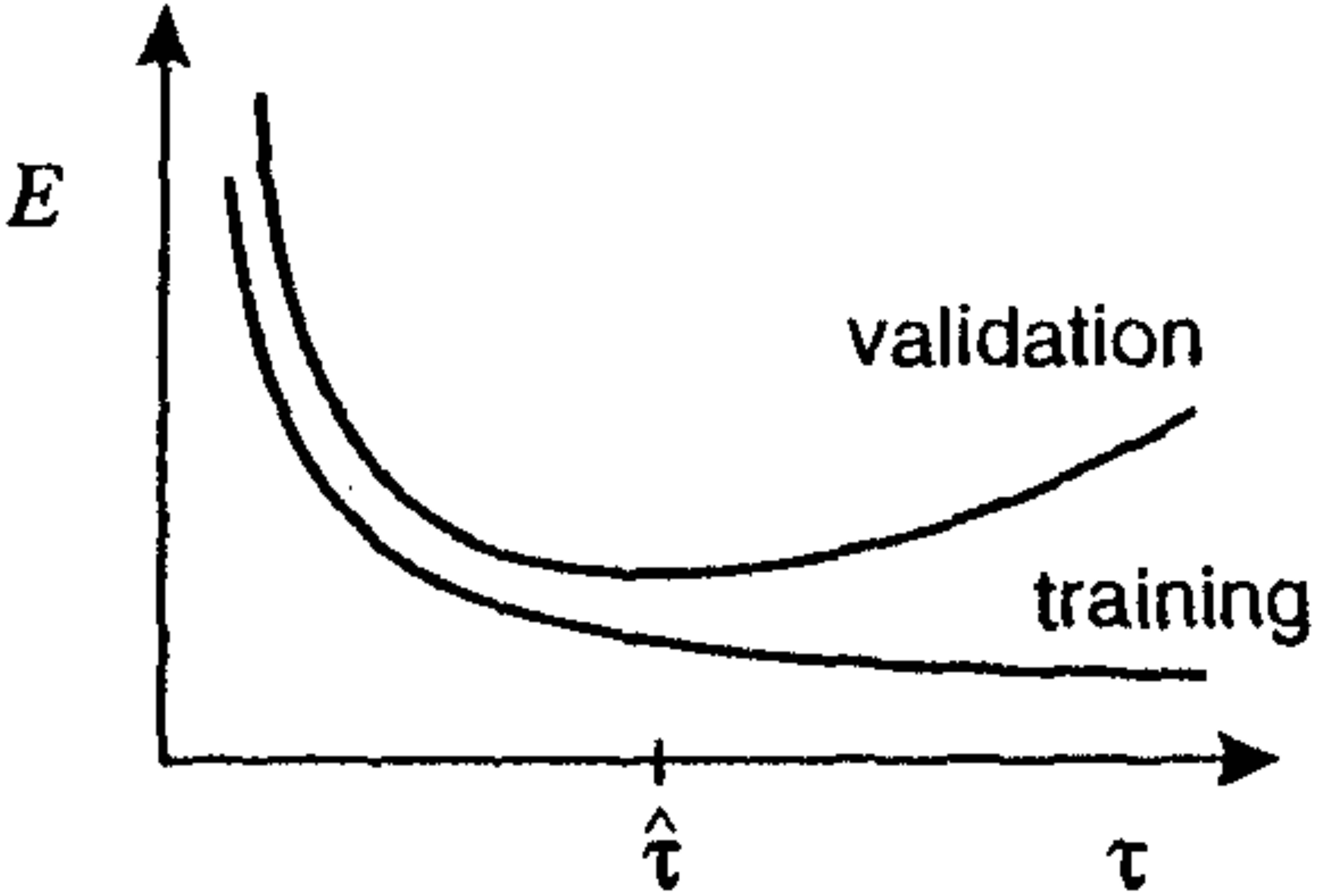
ψ thresholding function

MULTI-LAYER PERCEPTRON

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



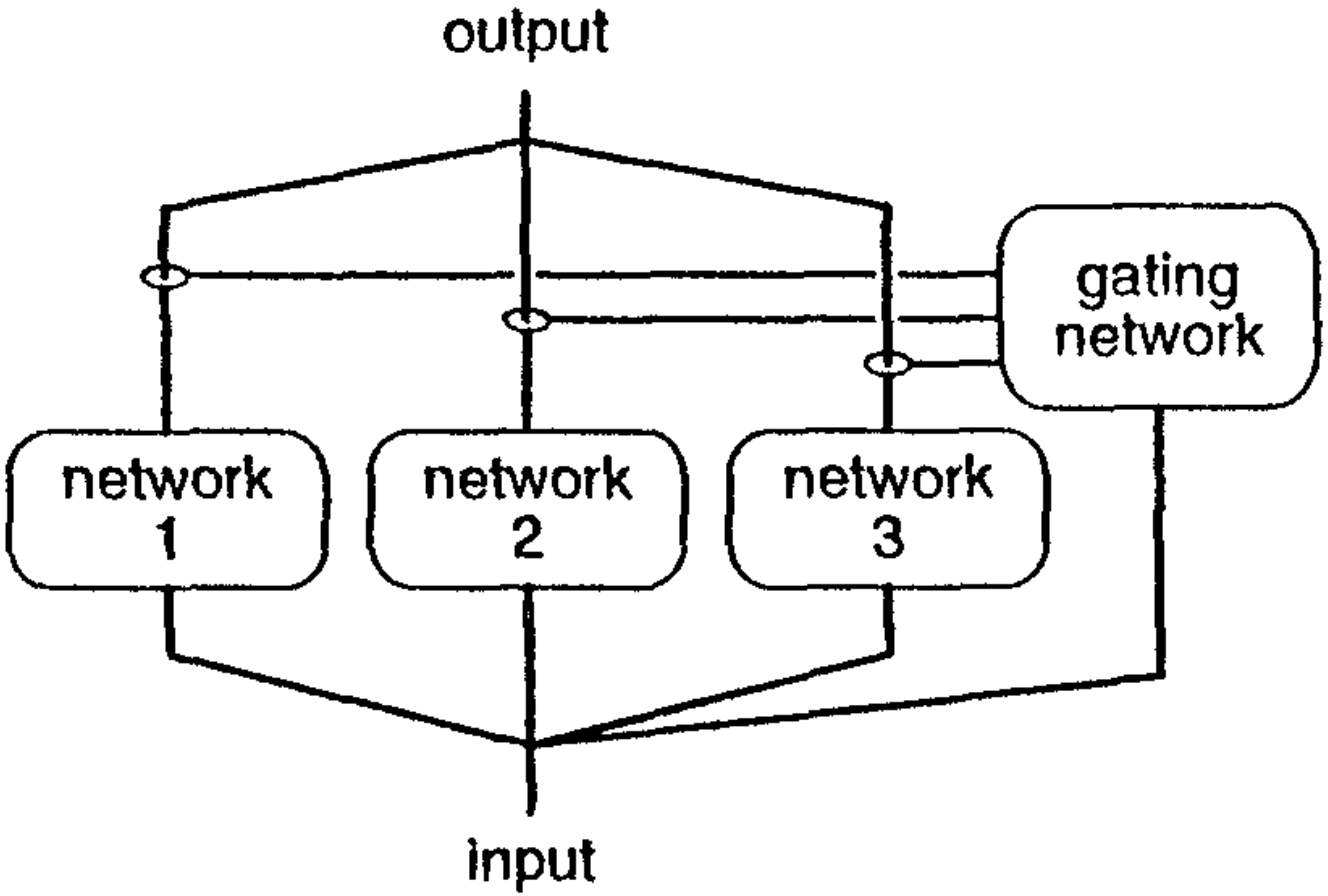
REGULARIZATION IN NEURAL NETWORKS



Other Approaches

- Training with noise
- Mixture of models
- Mixture of experts approach

MOE

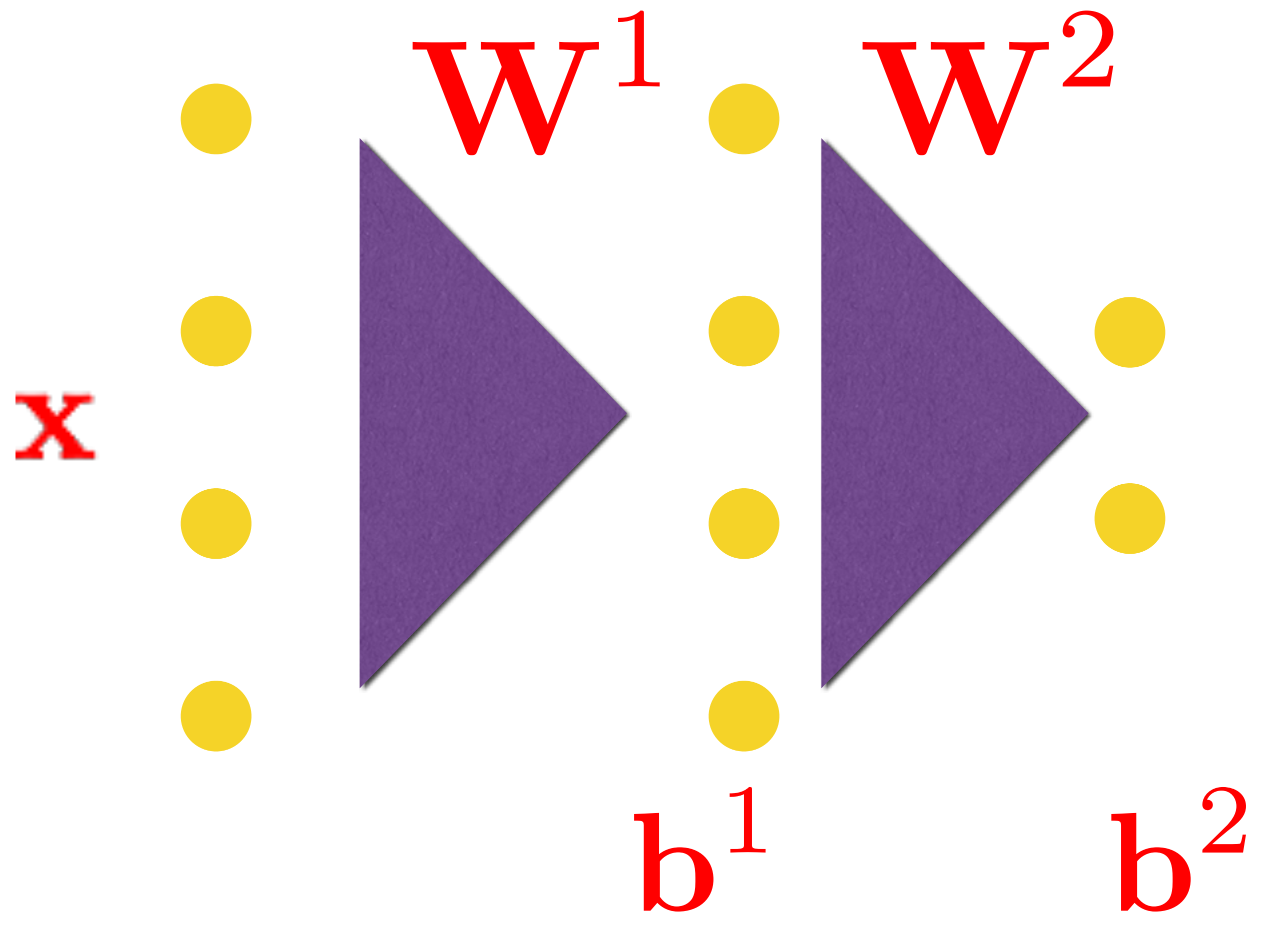


IISc logo: INDIAN INSTITUTE OF SCIENCE / भारतीय विज्ञान संस्थान

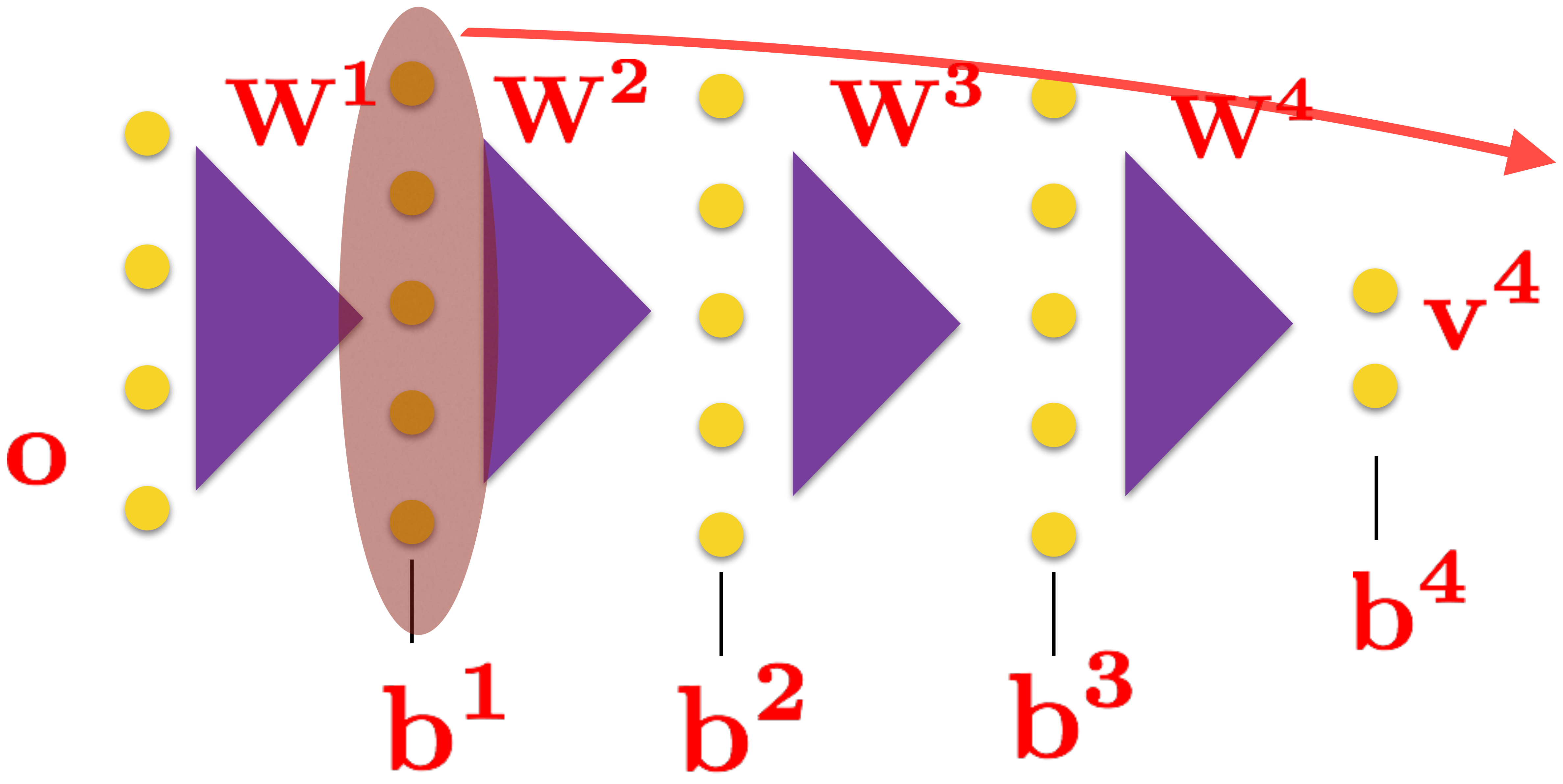
LEAP

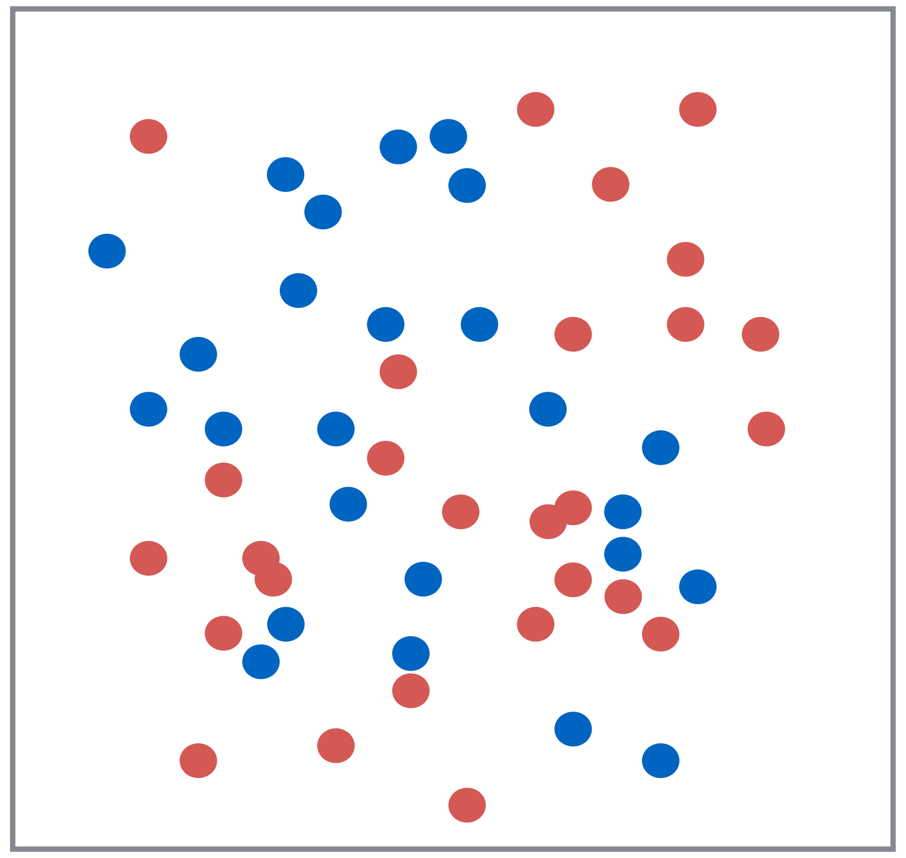
NEURAL NETWORKS - 1 HIDDEN LAYER

- For complex problems in audio/image/text
  - Single hidden layer may be too restrictive in learning the model parameters
  - May not scale up with availability of big data.

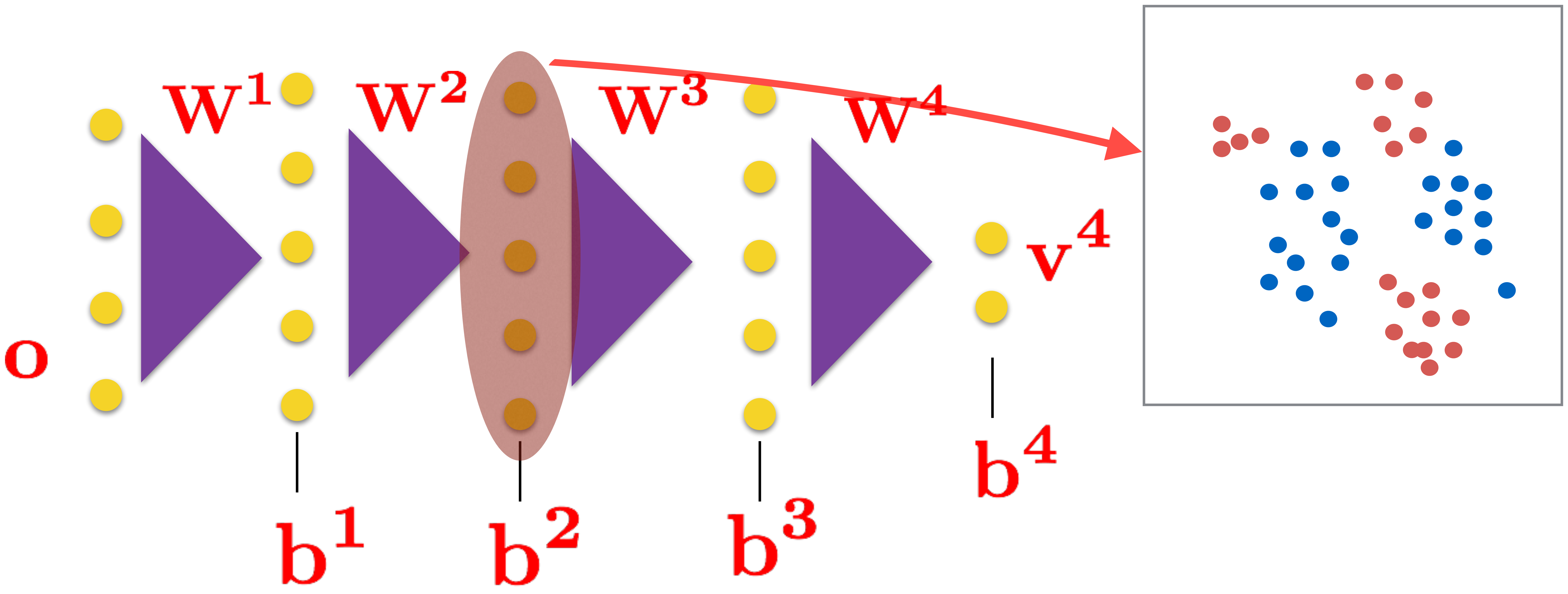


Neural Networks – Deep



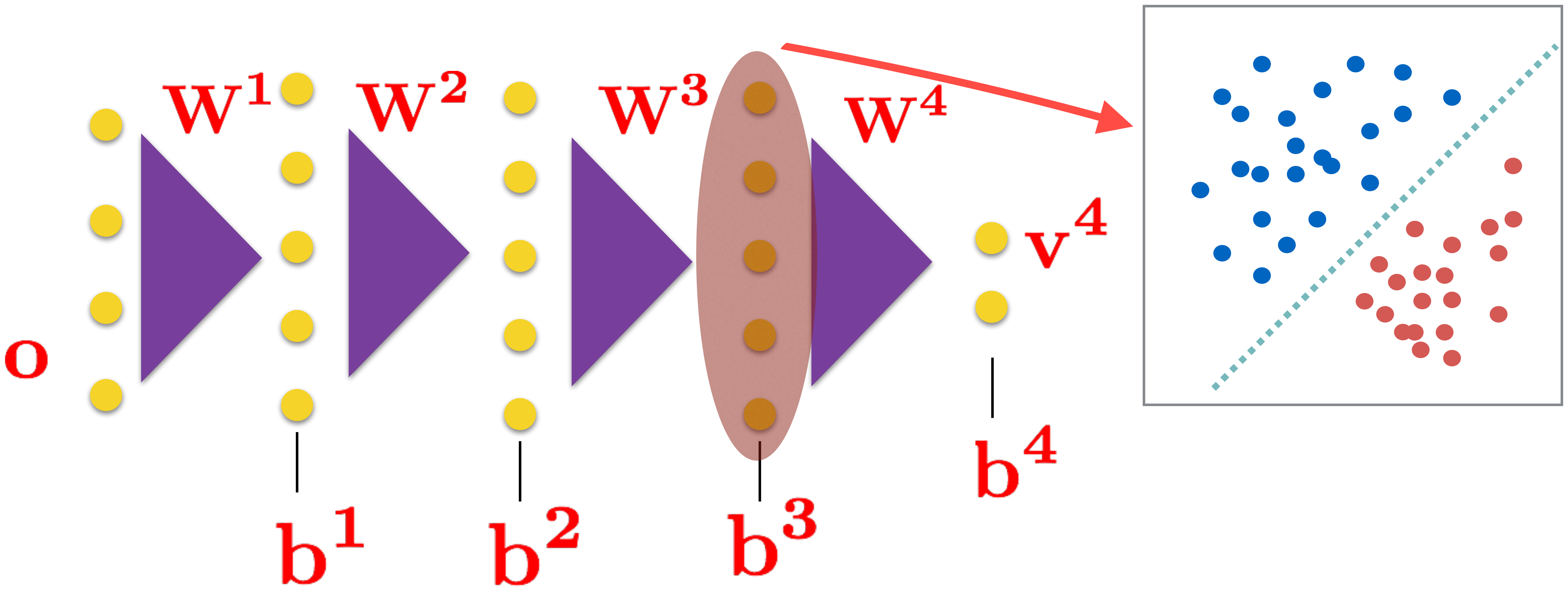


Neural networks with multiple hidden layers - Deep networks



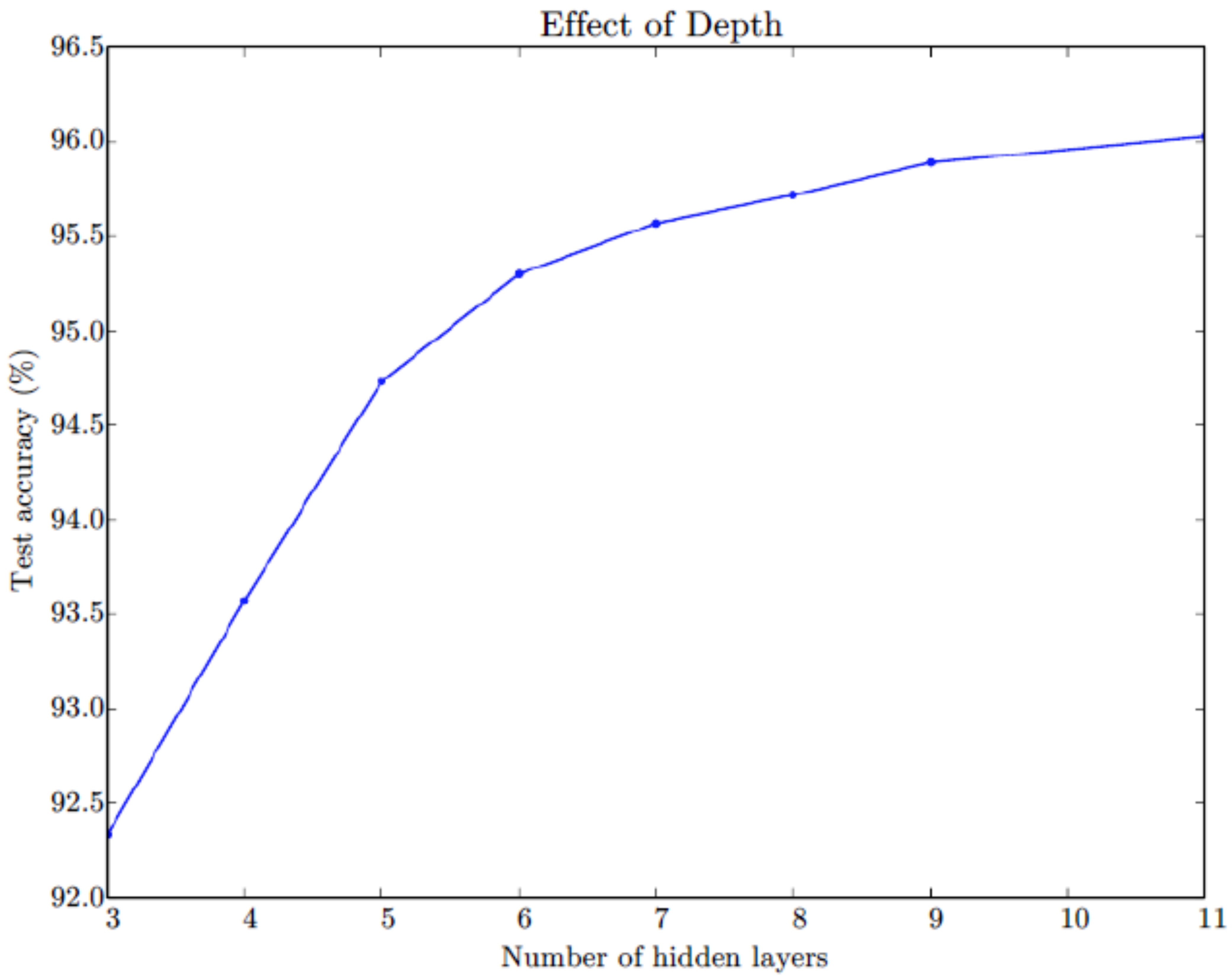
DEEP NEURAL NETWORKS

Neural networks with multiple hidden layers - Deep networks



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

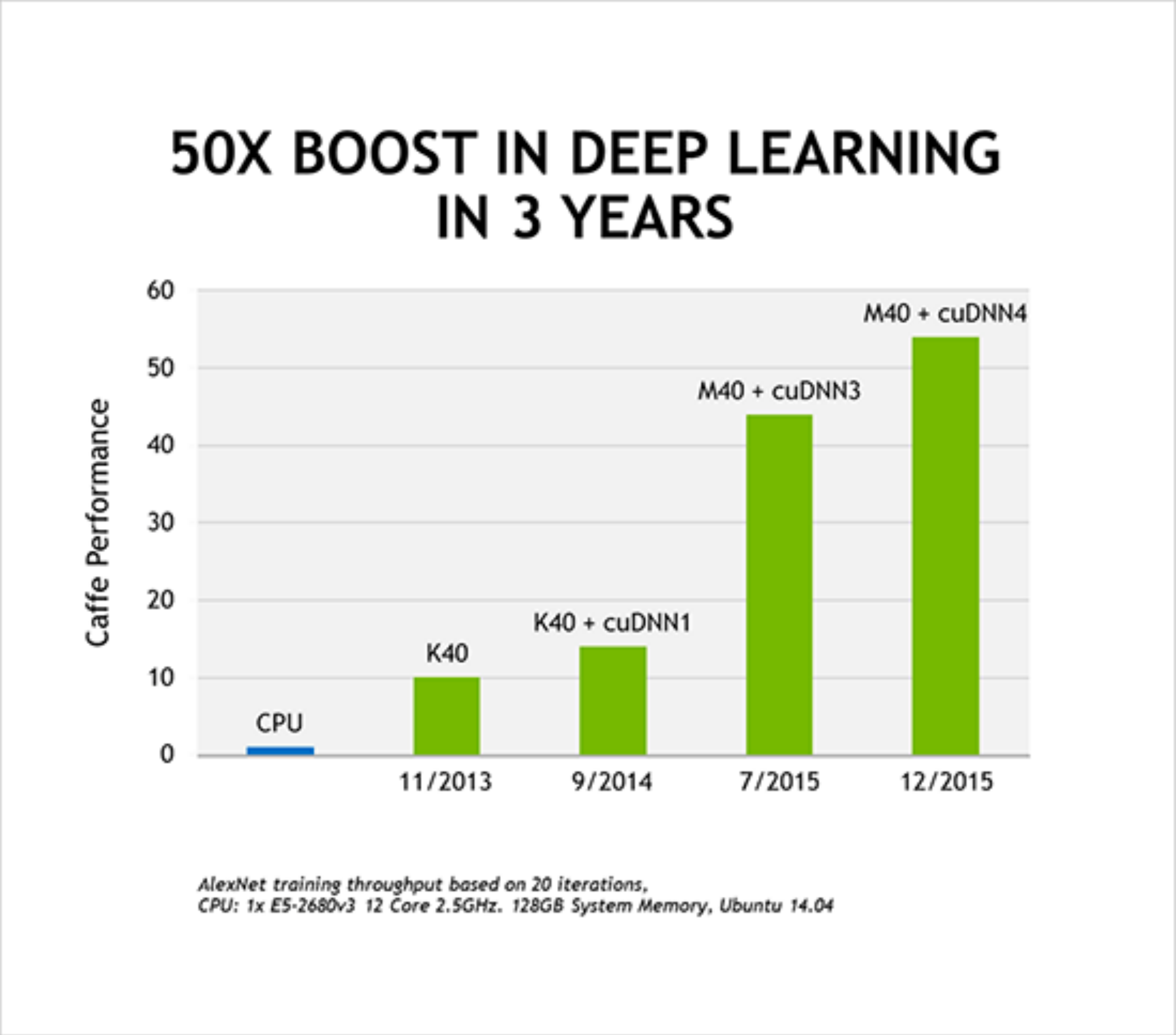
Need for Depth



$$\boldsymbol{h}^{(1)} = g^{(1)}\left(\boldsymbol{W}^{(1)\top}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right)$$

$$\boldsymbol{h}^{(2)} = g^{(2)}\left(\boldsymbol{W}^{(2)\top}\boldsymbol{h}^{(1)} + \boldsymbol{b}^{(2)}\right)$$

DEEP NEURAL 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

DEEP NEURAL 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.
  - 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.

THANK YOU

Sriram Ganapathy and TA team
LEAP lab, C328, EE, IISc
sriramg@iisc.ac.in