E9 205: Machine Learning for Signal Processing

Introduction to Neural Network Models

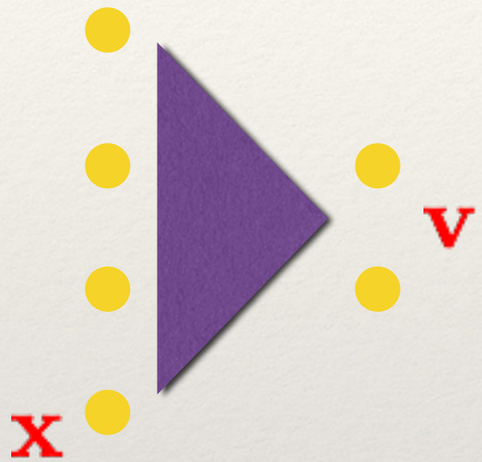
16-10-2019

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

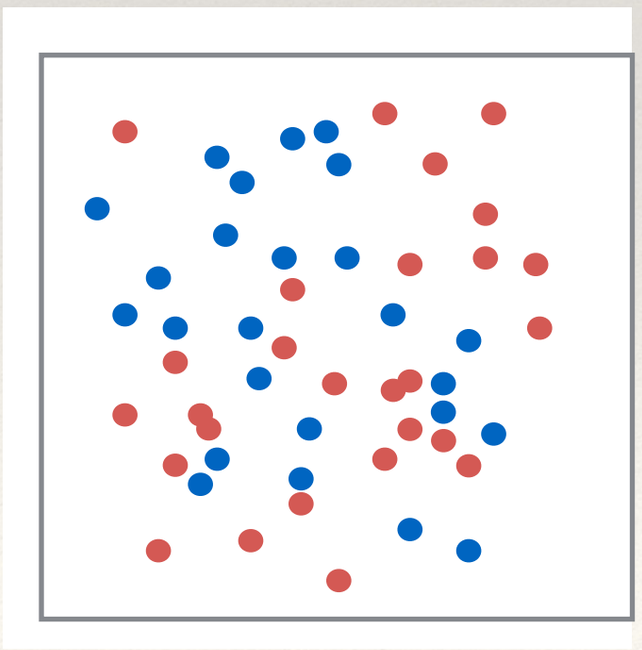
Perceptron Algorithm

Perceptron Model [McCulloch, 1943, Rosenblatt, 1957]



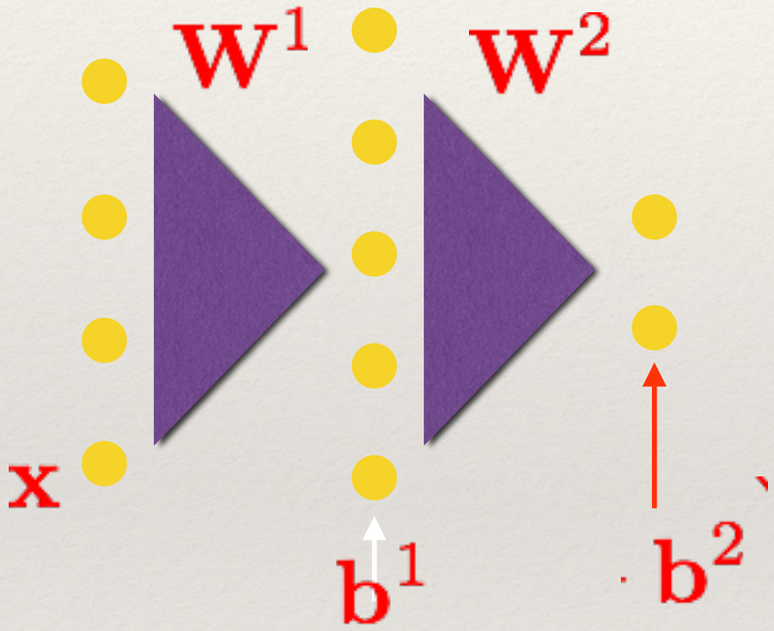
Targets are binary classes [-1,1]

What if the data is not linearly separable



Multi-layer Perceptron

Multi-layer Perceptron [Hopfield, 1982]



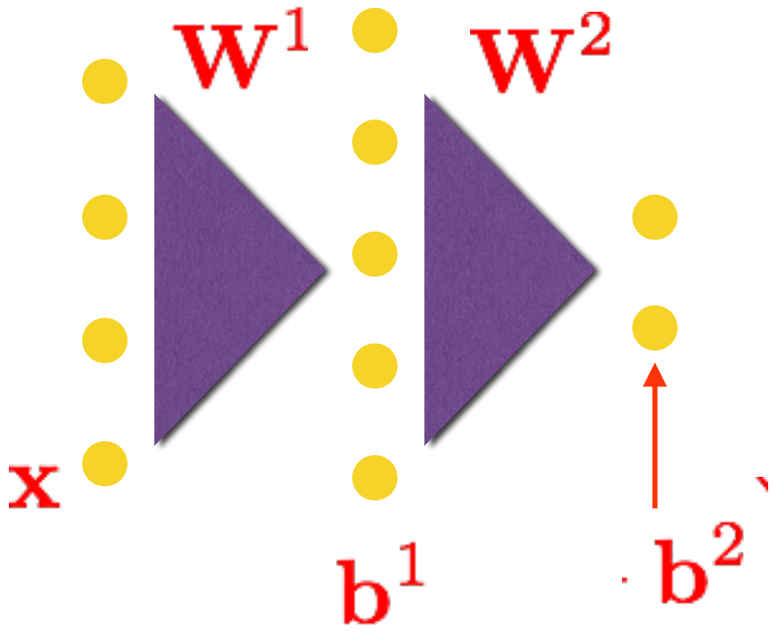
$$\mathbf{v}^2 = \psi\left(\mathbf{W}^2\phi(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1) + \mathbf{b}^2\right)$$

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

ψ thresholding function

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.

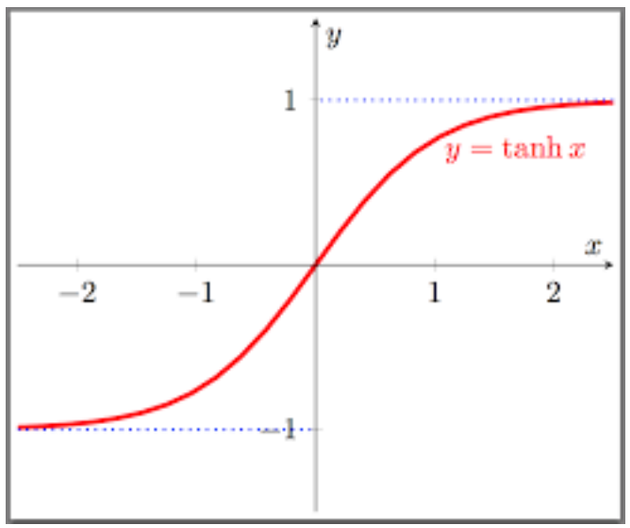
Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

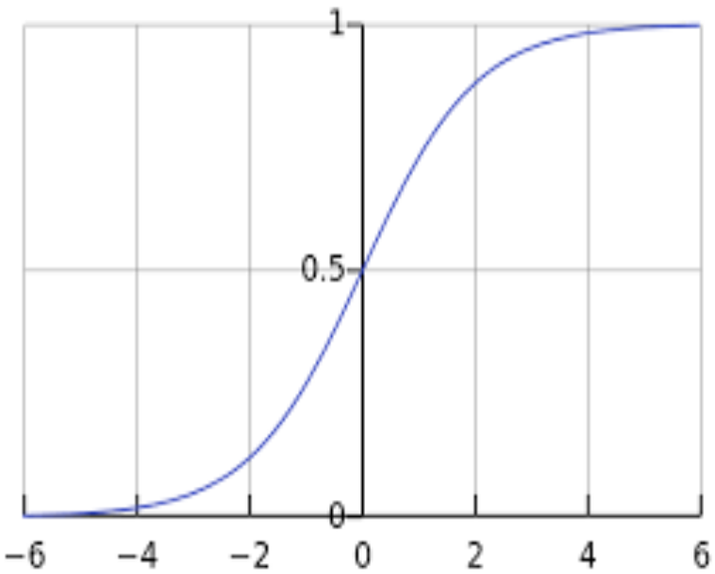
Neural Networks

Types of Non-linearities ϕ

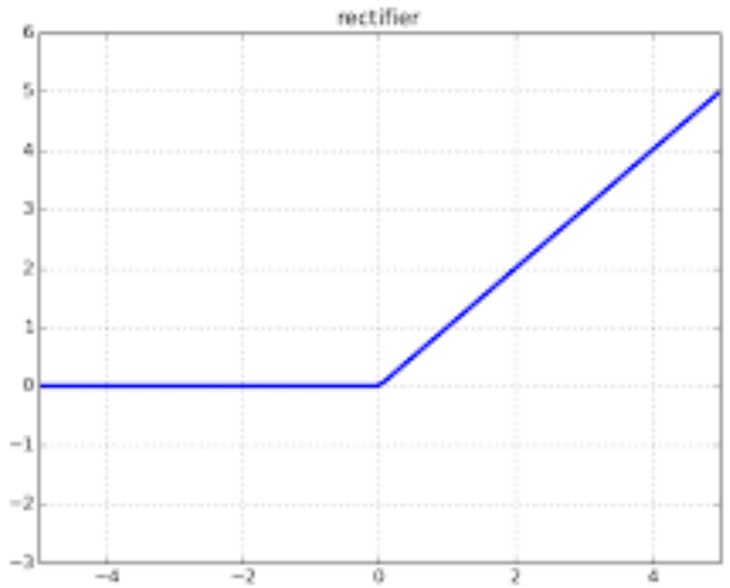
tanh



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

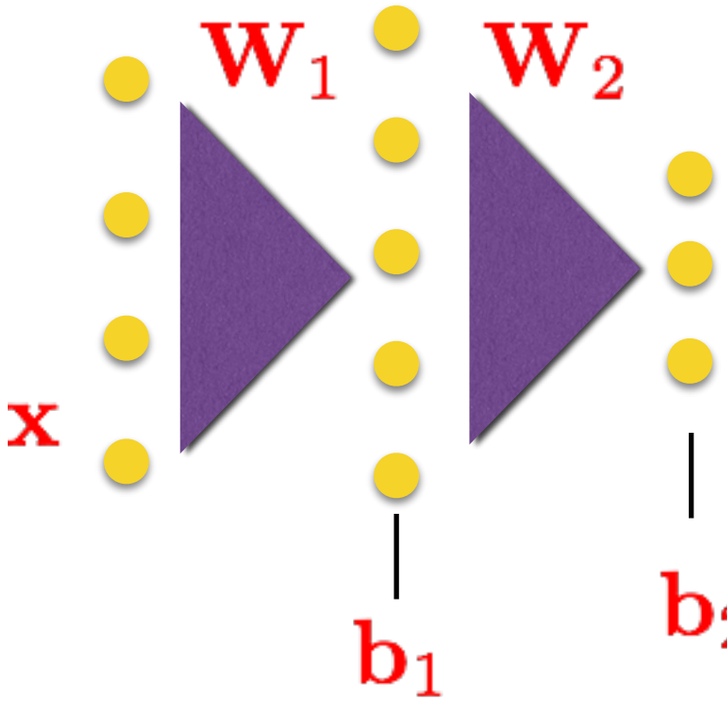
Choice of target function ψ

- 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

Parameter Learning

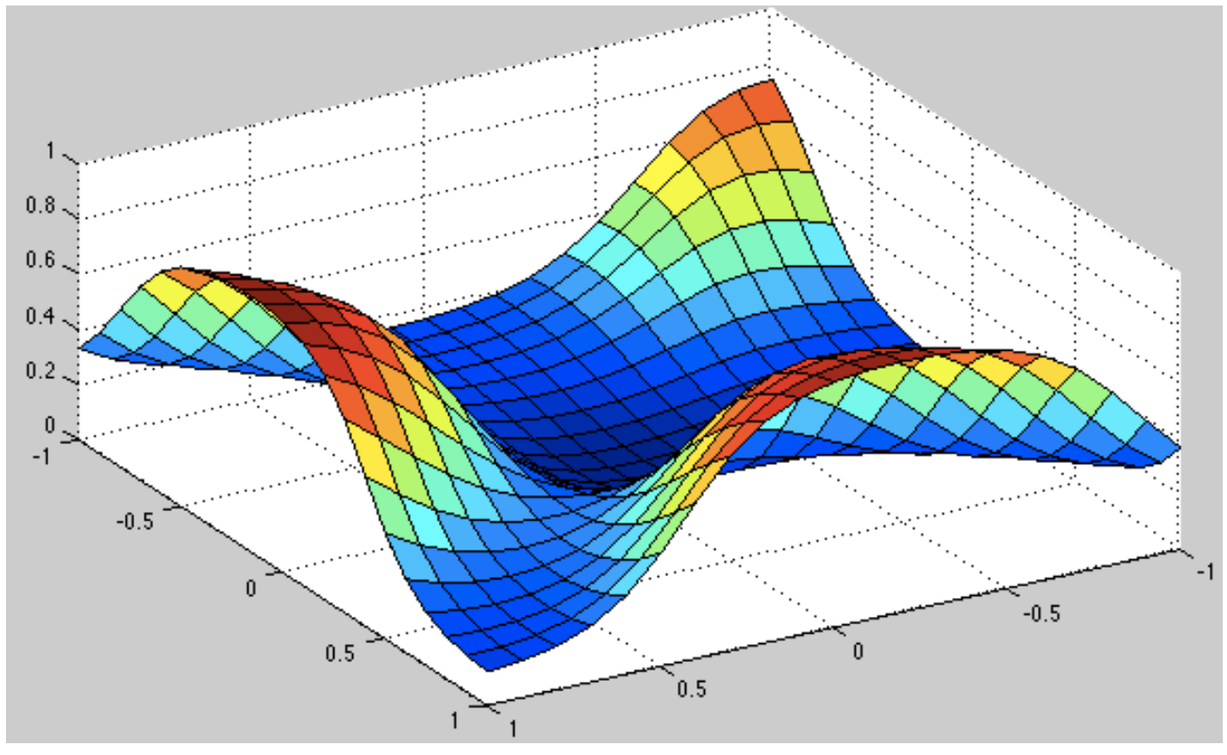


$$\mathbf{v}^2 = \psi\left(\mathbf{W}^2\phi(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1) + \mathbf{b}^2\right)$$

Error function for entire data

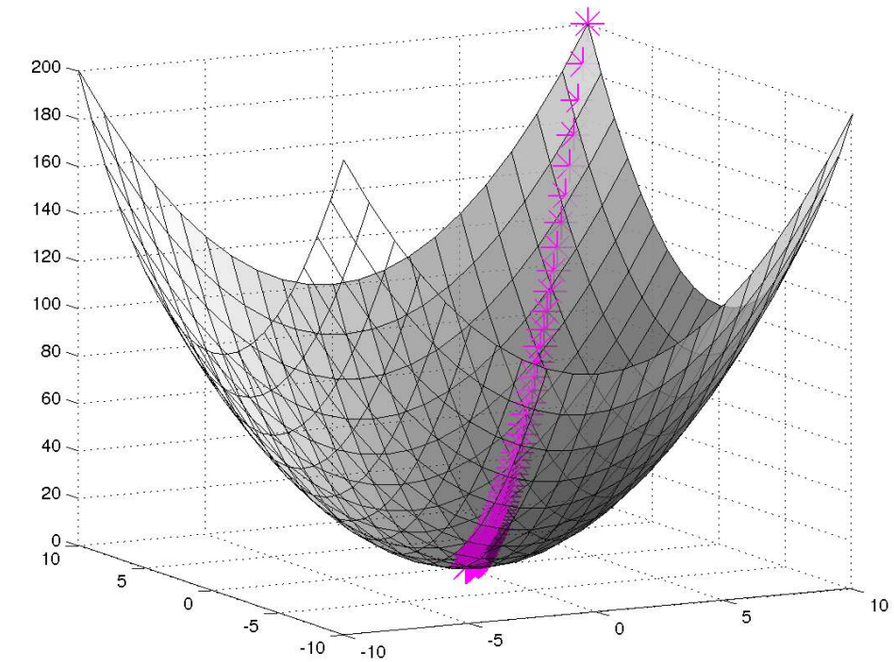
$$J_{MSE} = \sum_{i=1}^{M} ||\mathbf{v}_i - \mathbf{y}_i||^2$$

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



Parameter Learning

Error surface close to a local optima



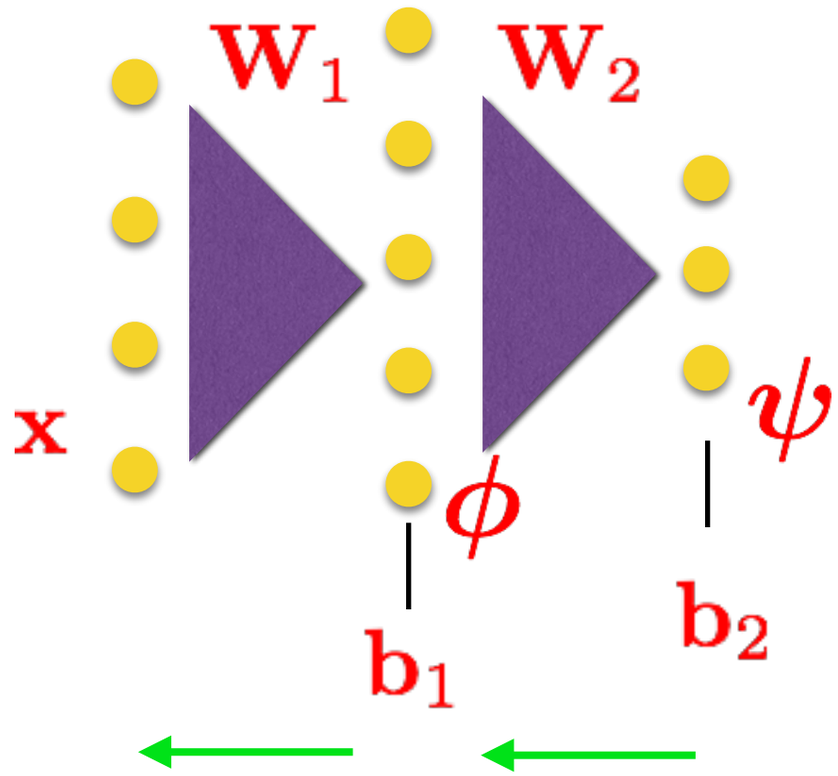
Non-linear nature of error function

- 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