Institute for Advanced Studies in Basic Sciences – Zanjan
Department of Computer Science and Information Technology
By: Mahdi Vasighi

Artificial Neural Networks

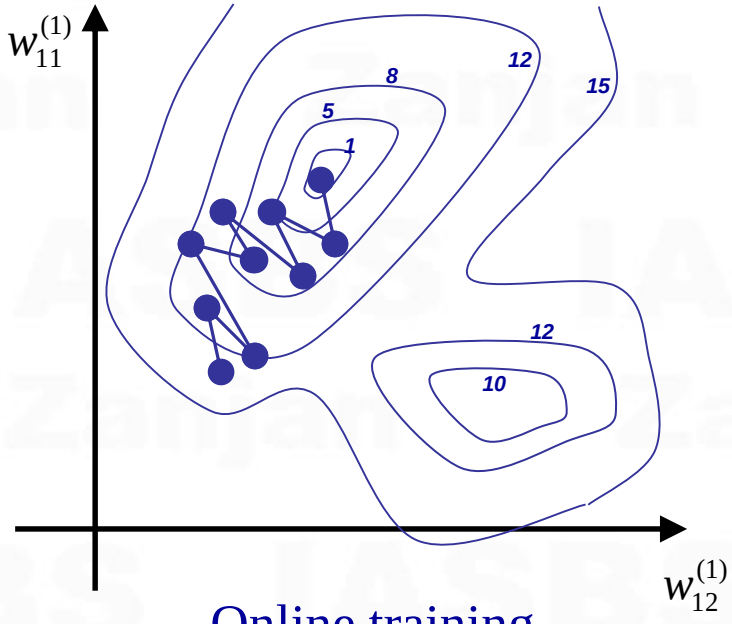
Part 7

Neural Learning

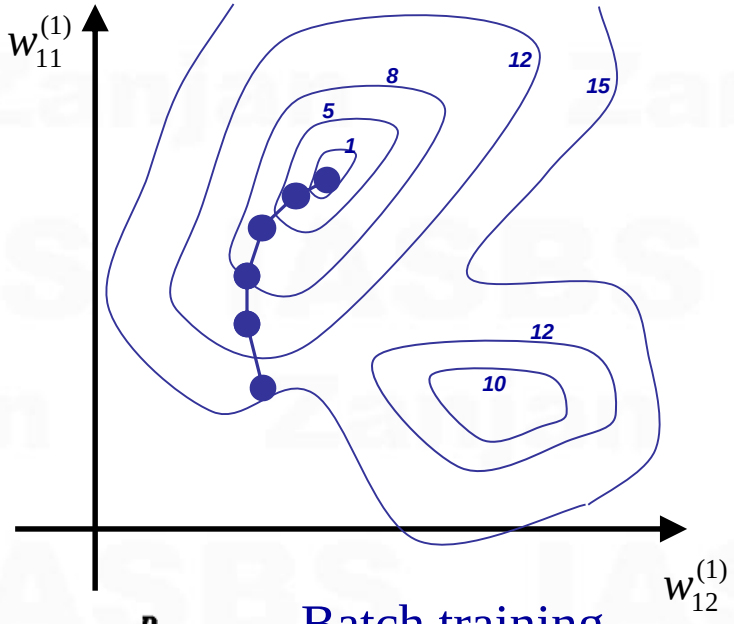
Back-Propagation Learning

Online vs. Batch training

The weights can be updated after presentation of each training pattern (Online or Incremental), rather than adding up the weight changes for all the patterns before applying them (Batch or Cumulative).



Online training



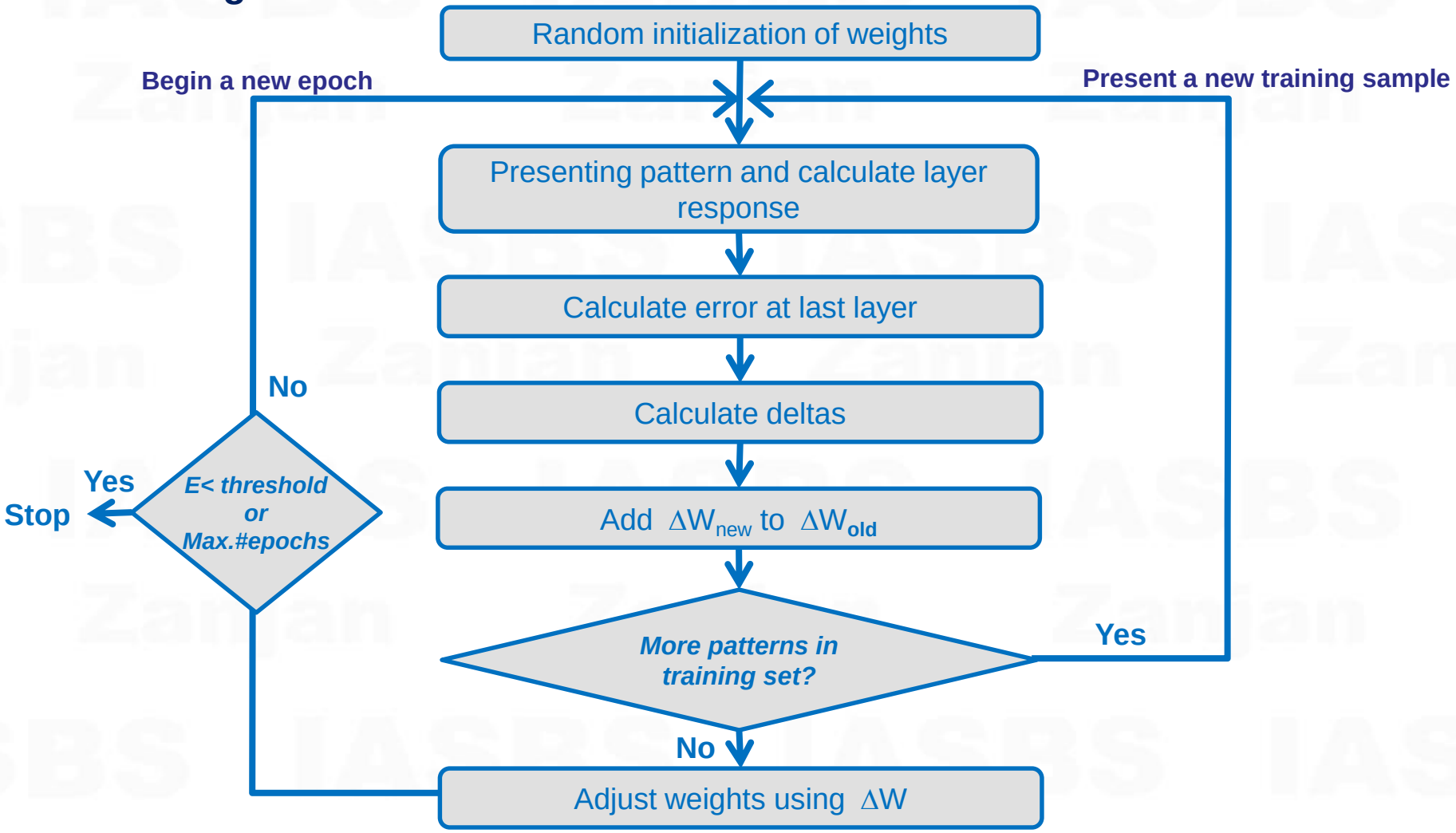
Batch training

$$\Delta w = \sum_{p=1}^{P} \Delta w_p$$

Neural Learning

Back-Propagation Learning

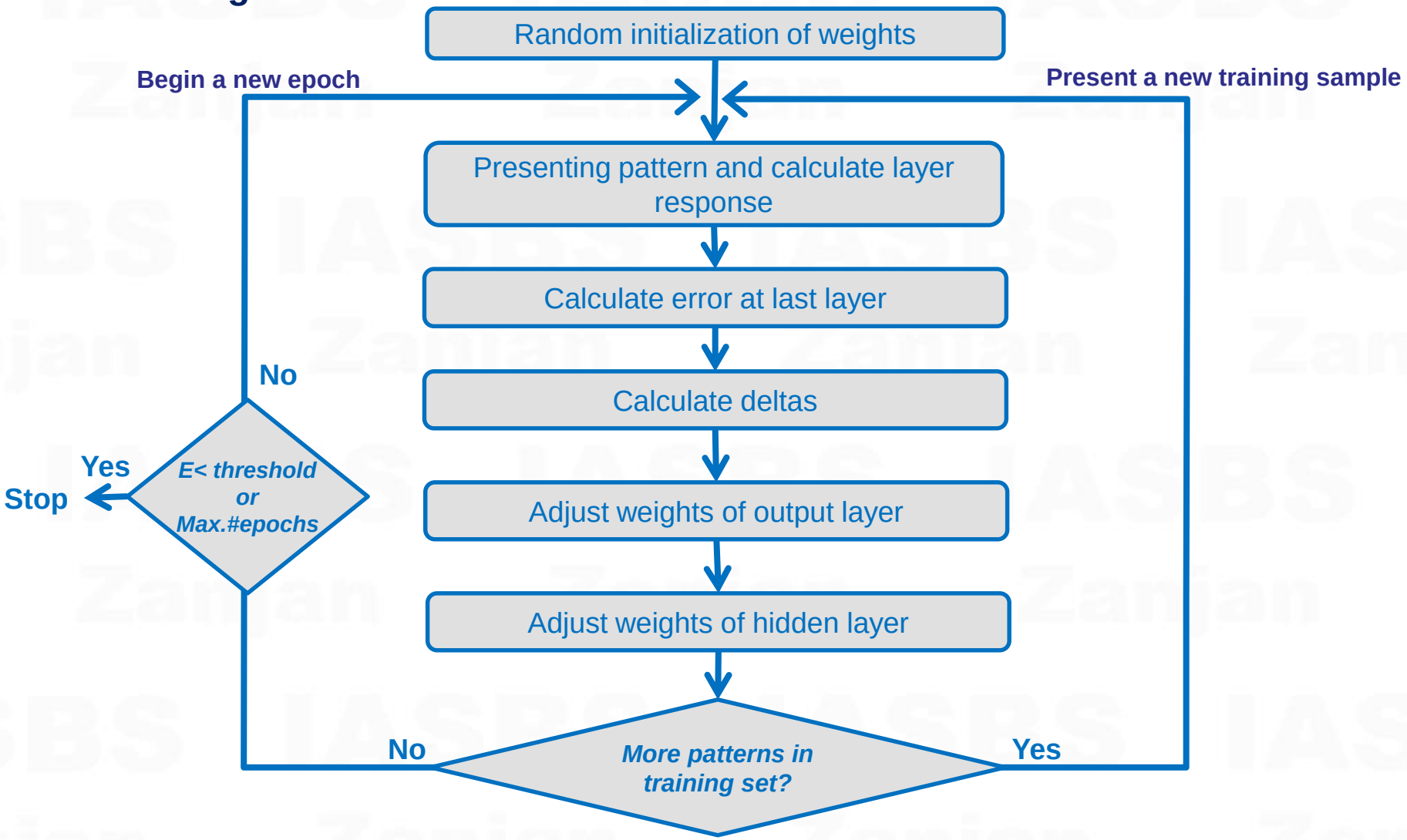
Batch training



Neural Learning

Back-Propagation Learning

Online training



Neural Learning

Back-Propagation Learning

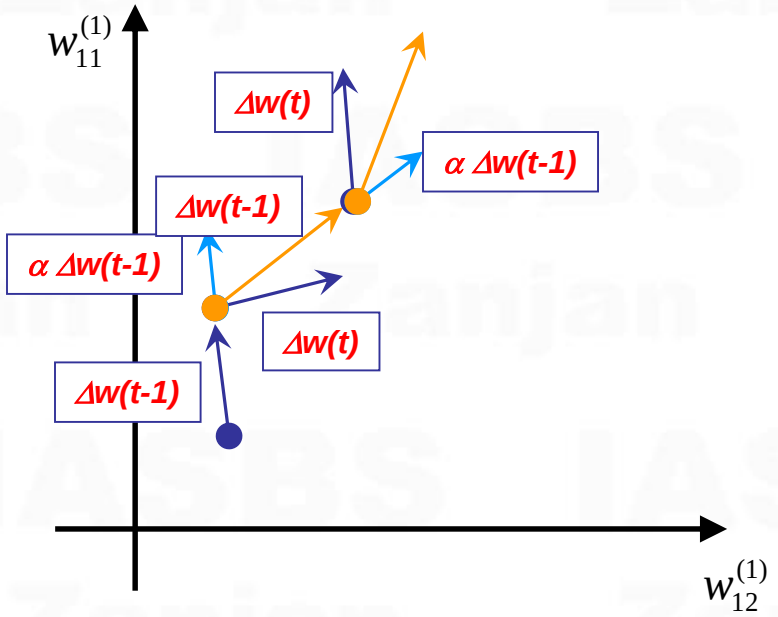
A compromise that will smooth out the irregular behavior of the on-line updates is to update the weights according to the weight change in previous step.

Momentum method

$$\Delta w_{hl}^{(n)}(t) = \eta \sum_{p} delta_{l}^{(n)}(t).out_{h}^{(n-1)}(t) + \alpha \Delta w_{hl}^{(n)}(t-1)$$

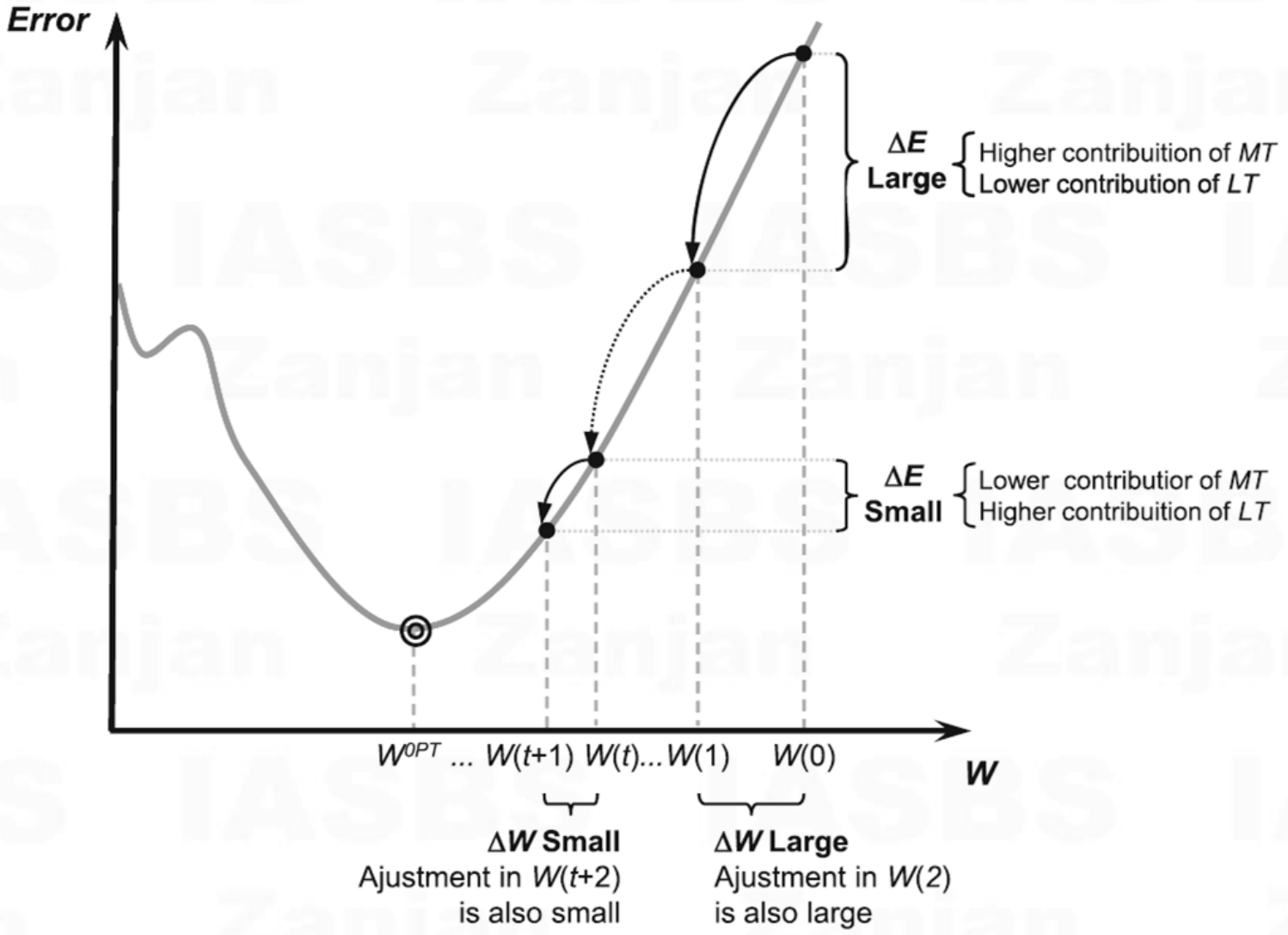
We may prevent the network from getting stuck in some local minimum by making the weight changes depend on previous states.

optimal value of alpha depends on the application and is not easy to determine a priori.



Neural Learning

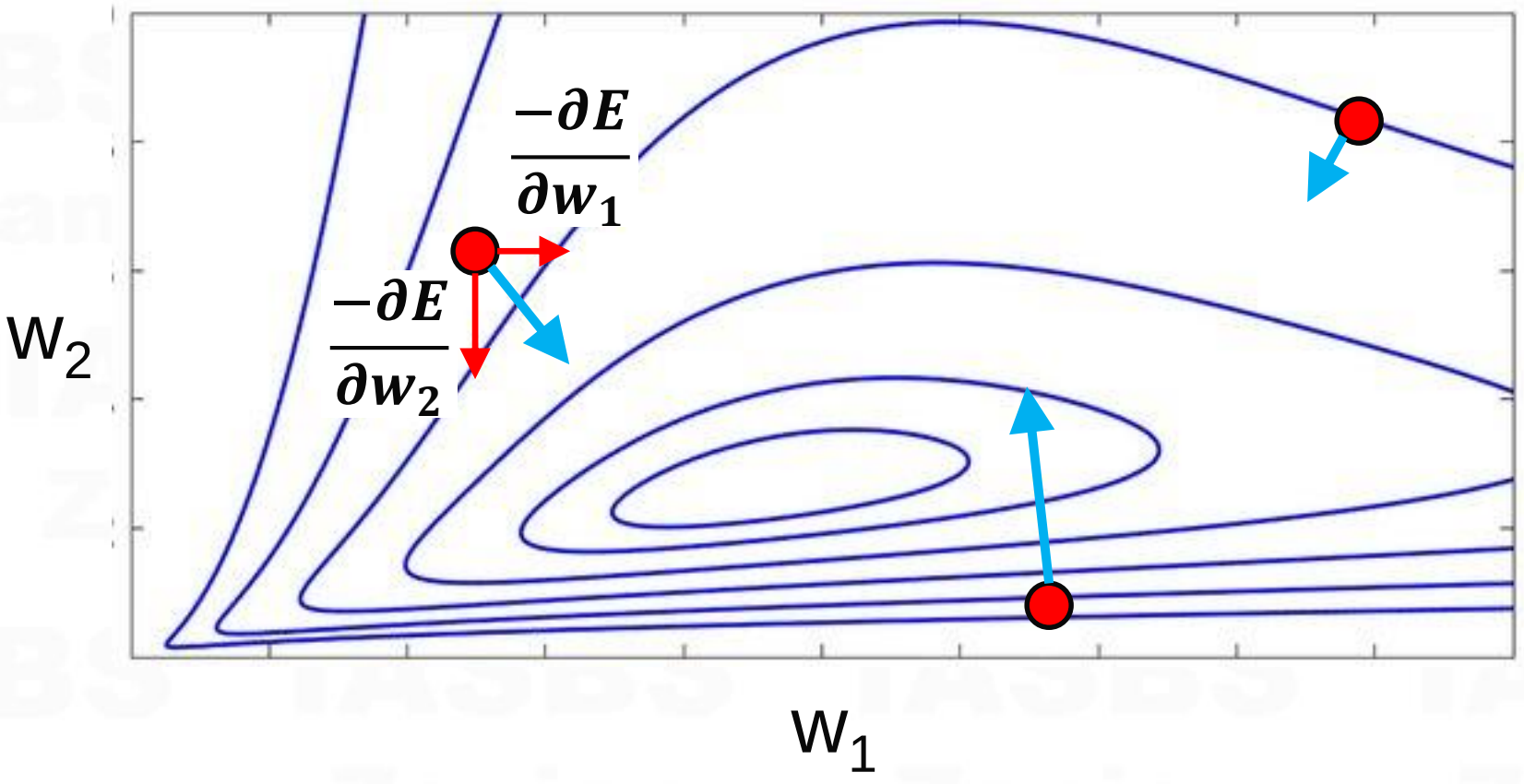
Back-Propagation Learning



Neural Learning

Back-Propagation Learning

The magnitude of the shift vector is dependent to the partial derivatives.



Neural Learning

Back-Propagation Learning

Resilient Back-Propagation Method

- Hyperbolic tangent → output between 0 and 1 or -1 and 1
- Saturation can be occurred.
- Partial derivatives would produce values next to zero.

- Effects of the magnitudes of the partial derivatives can be eliminated by Resilient BP.

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^{+} \cdot \Delta_{ij}^{(t-1)}, & \text{if } \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^{-} \cdot \Delta_{ij}^{(t-1)}, & \text{if } \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)}, & \text{otherwise} \end{cases}$$

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & \text{if } \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, & \text{if } \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, & \text{otherwise} \end{cases}$$

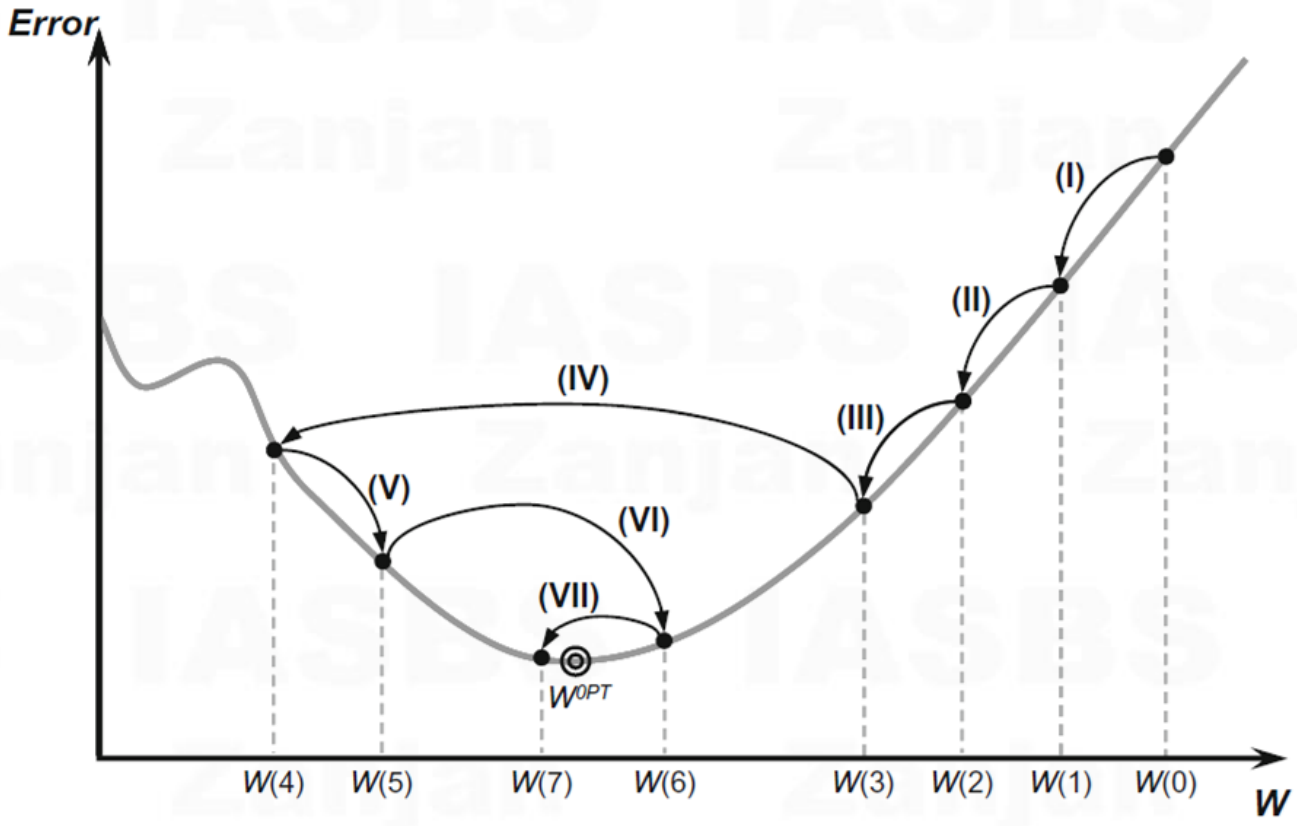
$$\eta^{+} > 1, \quad 0 > \eta^{-} > 1$$

Neural Learning

Back-Propagation Learning

Resilient Back-Propagation Method

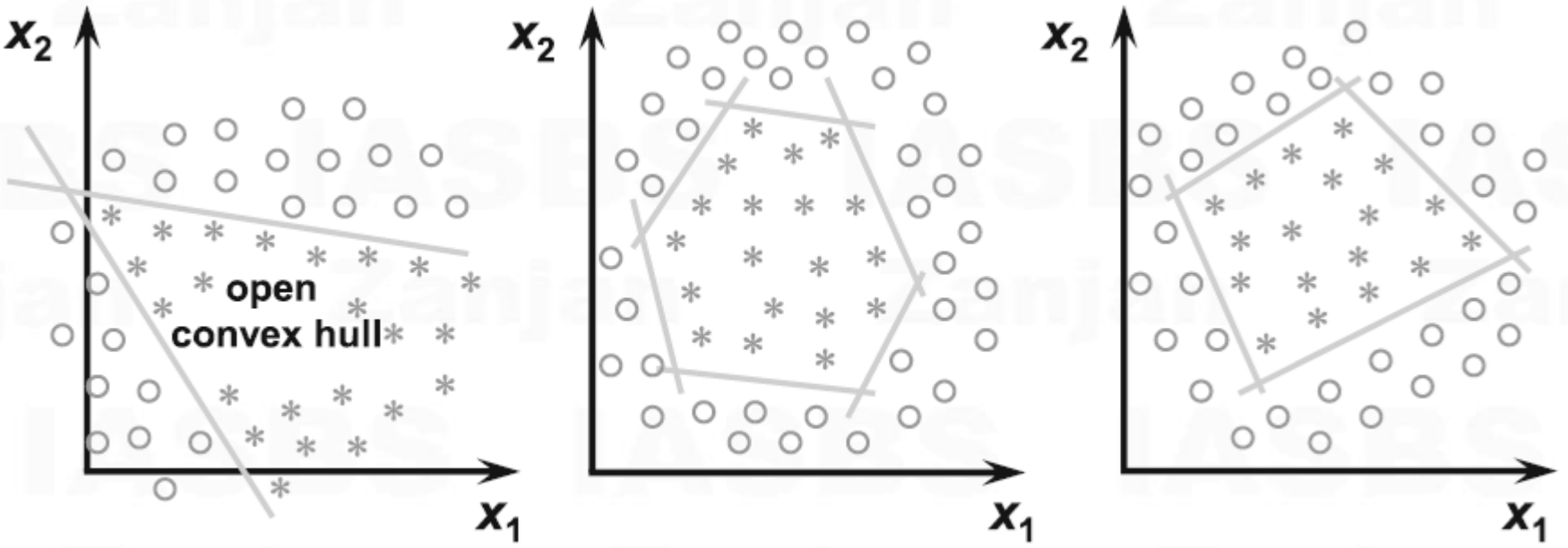
- η+ is empirically set to 1.2 and η− to 0.5.
- $\Delta max=50$, and $\Delta min=10^{-6}$



Neural Learning

Application of MLPs

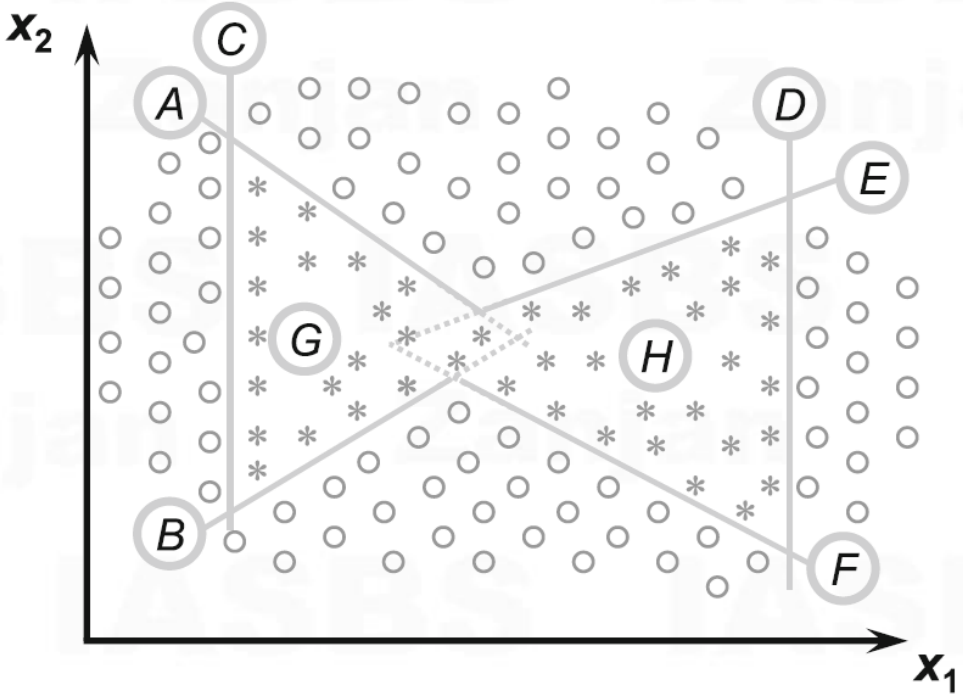
Pattern classification

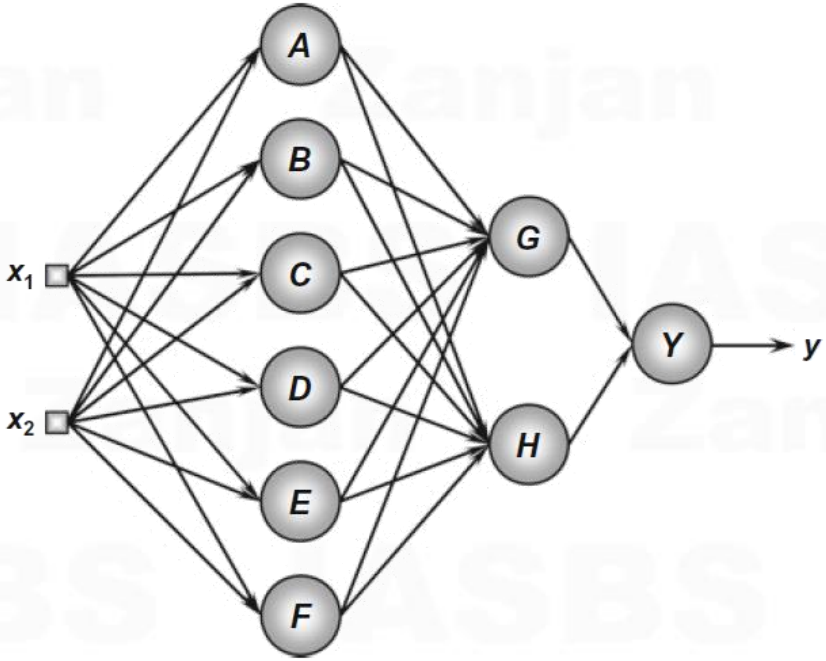


MLP networks with a single hidden layer can classify patterns placed within a convex region

Neural Learning

Application of MLPs

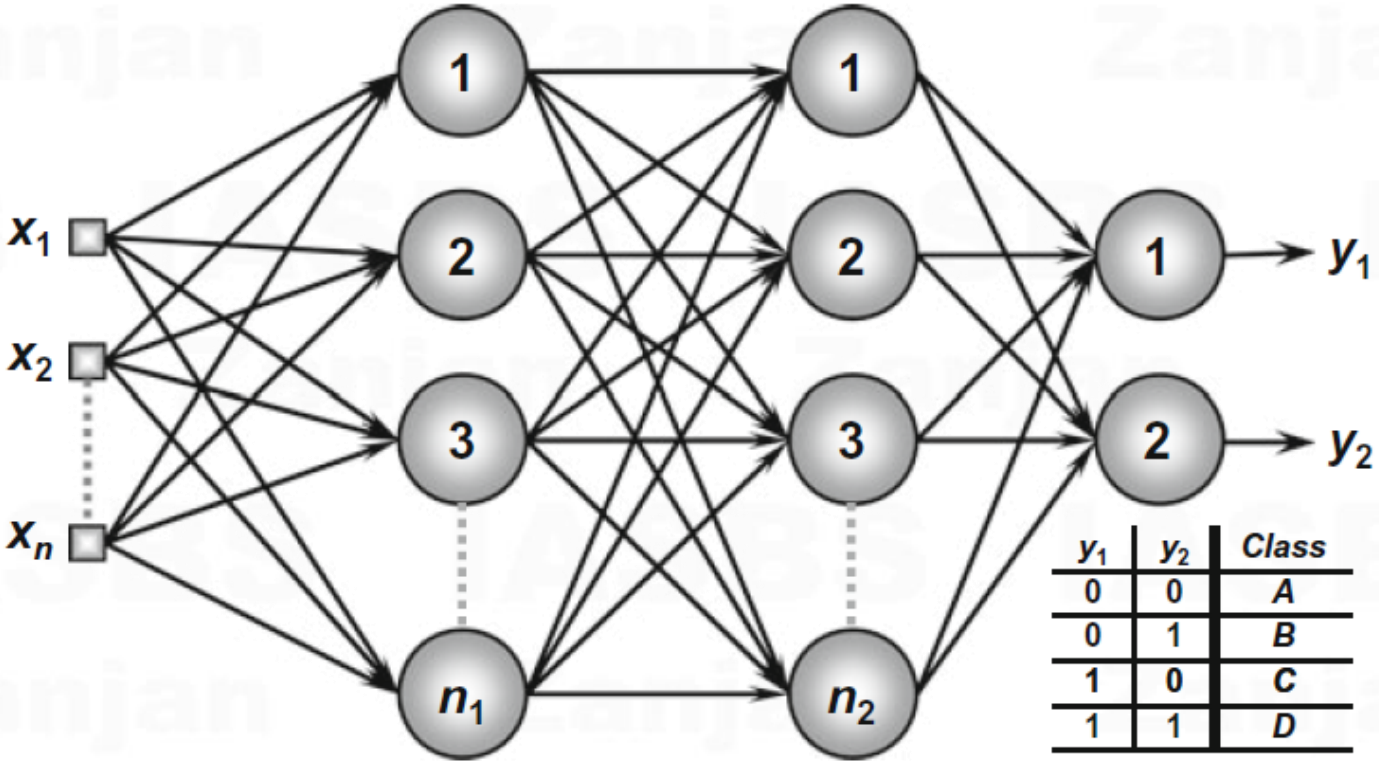




MLP networks with two hidden layers can classify patterns that are within any geometric region (Lui 1990; Lippmann 1987), including non-convex regions.

Neural Learning

Application of MLPs

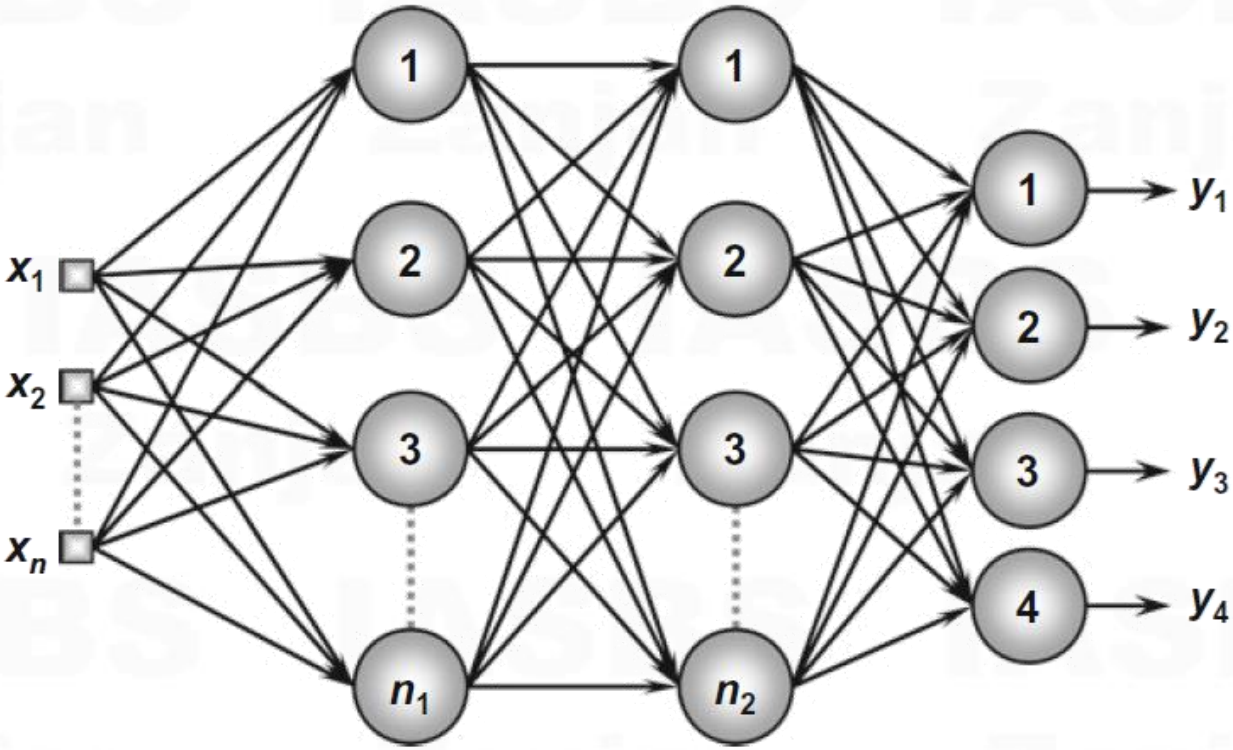


y_1	y_2	Class
0	0	A
0	1	B
1	0	C
1	1	D

An MLP with m neurons in its output layer would be able to classify, theoretically, up to 2^m classes.

Neural Learning

Application of MLPs

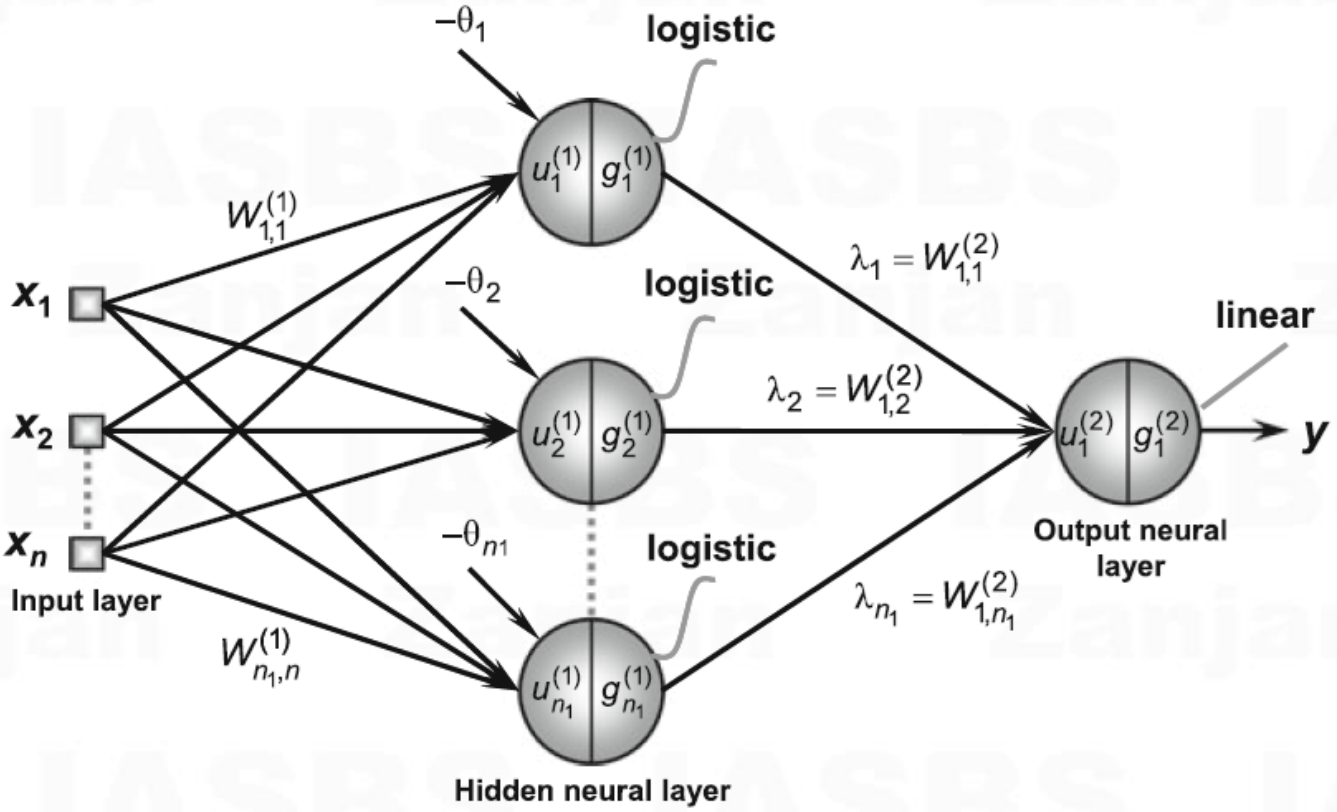


- One of the most used methods of codification is the “one of c-class”, which is an orthogonal codification.
- Output values provided by the neurons of the output layer must be post-processed

Neural Learning

Application of MLPs

Functional Approximation (Curve Fitting)

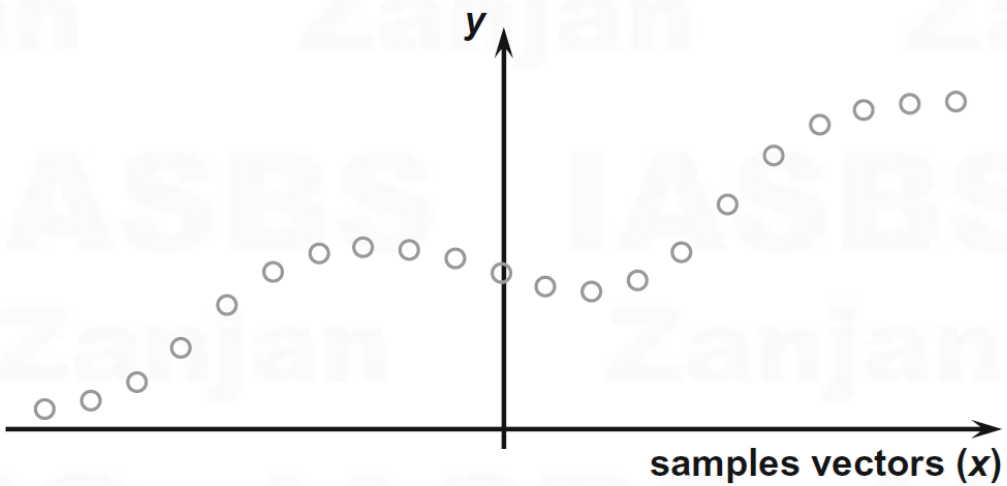


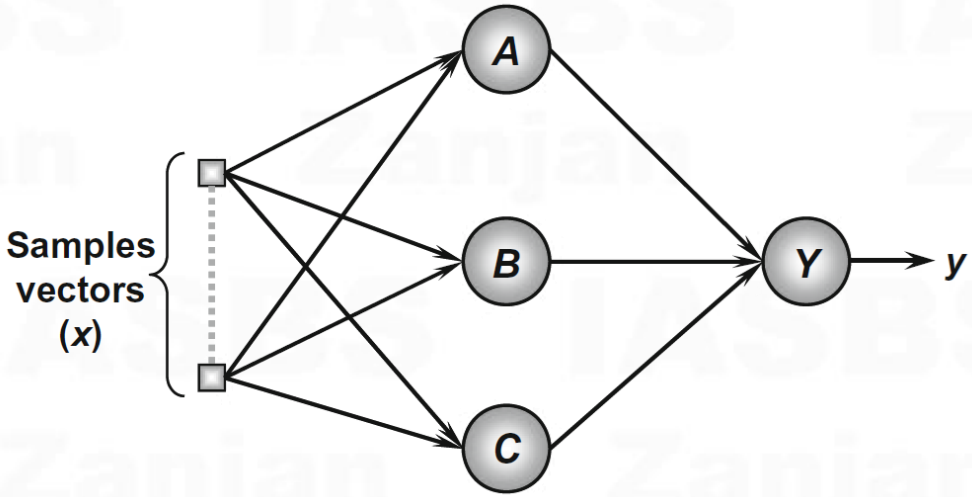
the only information available is a set of inputs/outputs.

Neural Learning

Application of MLPs

Functional Approximation (Curve Fitting)

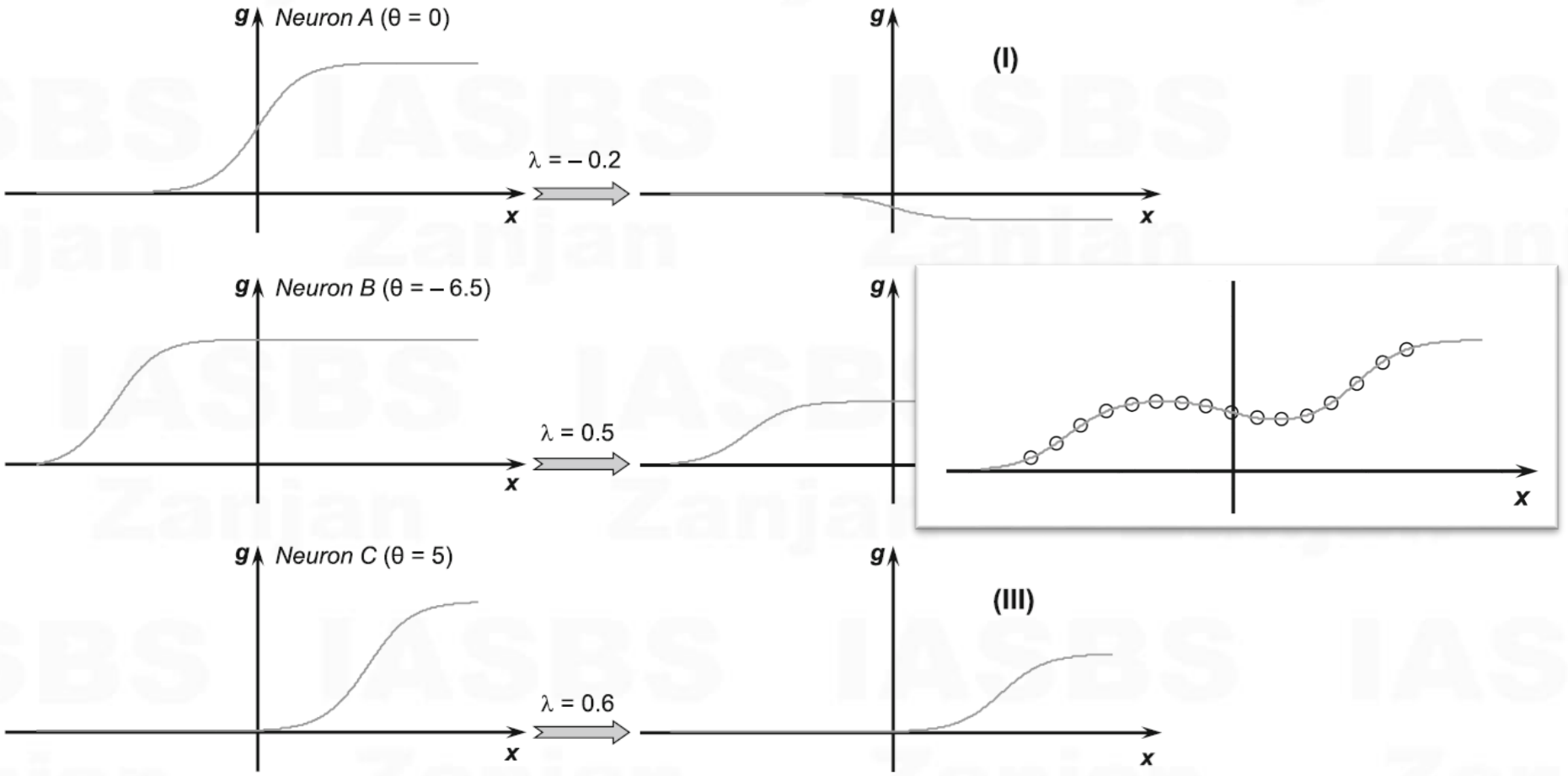




Neural Learning

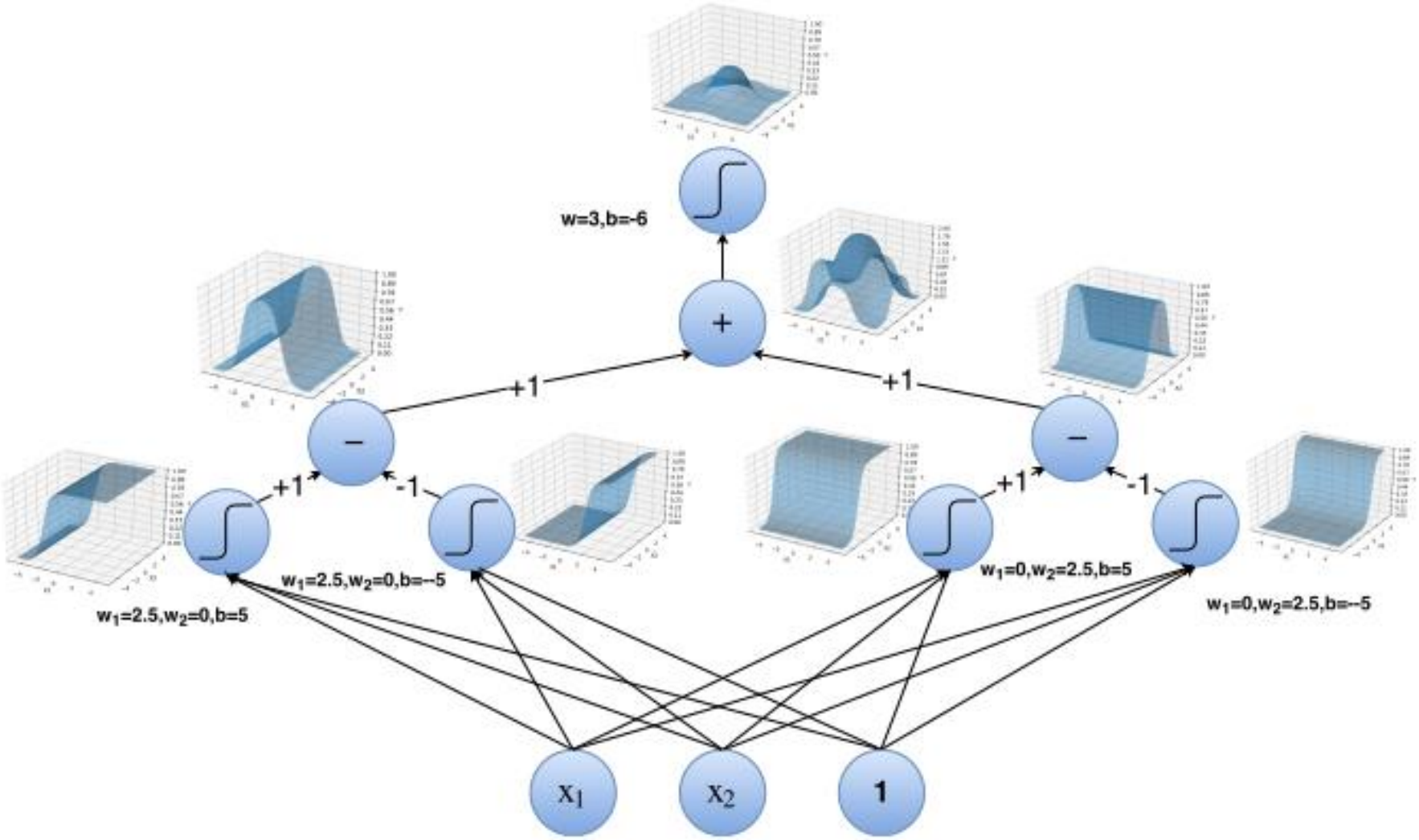
Application of MLPs

Functional Approximation (Curve Fitting)



Neural Learning

Application of MLPs



https://towardsdatascience.com/representation-power-of-neural-networks-8e99a383586