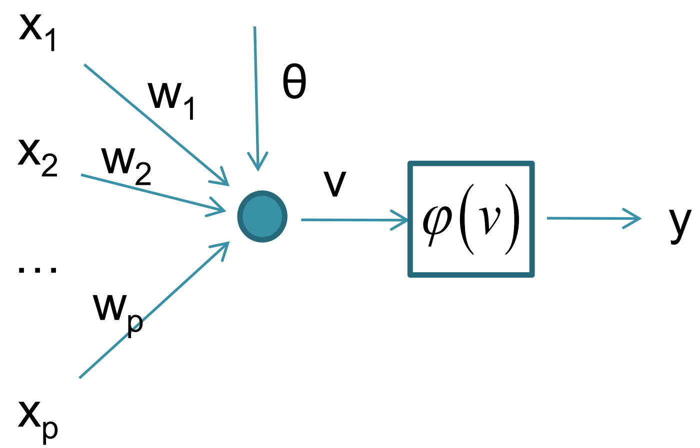
Brief summary on Lecture 2

- Linear Discriminant Analysis (LDA):

$$\delta_j(\mathbf{x}) = \boxed{\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j} + \log \pi_j - \frac{1}{2} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j$$

$$k = \arg\max_j \delta_j(\mathbf{x}).$$



- Single Layuer erceptron (SLP)

$$y = \varphi\left(\sum_{i=1}^{p} w_i x_i + \theta\right)$$

- SLP learning algorithm: $\mathbf{X}_1 \in C_1, \mathbf{X}_2 \in C_2$.

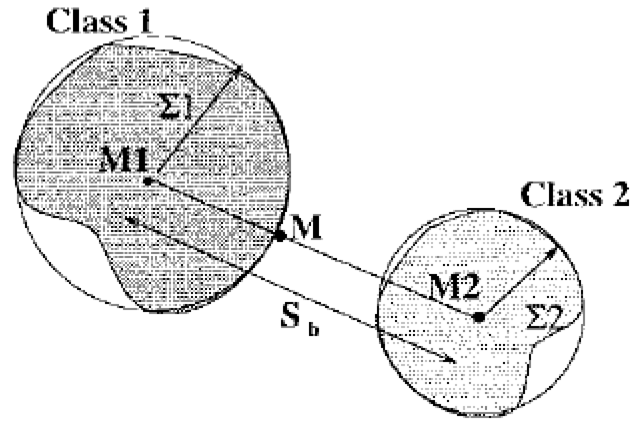
$$\begin{cases} \mathbf{w}^T \mathbf{x} \geq 0 & \mathbf{x} \in C_1 \\ \mathbf{w}^T \mathbf{x} < 0 & \mathbf{x} \in C_2 \end{cases} \Rightarrow \begin{cases} e(n) = d(n) - y(n), \\ \Delta \mathbf{w}(n) = \eta e(n) \mathbf{x}(n), \\ \mathbf{w}(n+1) = \mathbf{w}(n) + \Delta \mathbf{w}(n). \end{cases}$$

LDA, SLP, and Fisher criterion

- subsection 5.8.2

- Within-class scatter matrix:

$$\mathbf{S}_w = \sum_{j=1}^{M} \pi_j \mathbf{\Sigma}_j$$



- Between-class scatter matrix:

$$\mathbf{S}_b = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \pi_i \pi_j \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^T$$

- In order to have good separability for classification, one needs to have "large" between-class scatter and "small" within-class scatter simultaneously.

Fisher criterion:

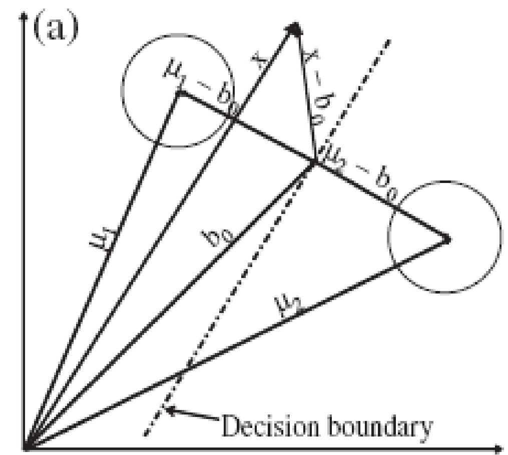
$$\mathbf{W} = \arg\max_{\mathbf{W}} \mathrm{Trace}\left(\frac{\mathbf{W}\mathbf{S}_b\mathbf{W}^T}{\mathbf{W}\mathbf{S}_w\mathbf{W}^T}\right)$$

LDA, SLP, and Fisher criterion

- For a two-class problem, the optimum LDA solution or *Fisher criterion* can be obtained by:

$$\mathbf{w} = \arg\max_{\mathbf{w}} \left(\frac{\mathbf{w}\mathbf{S}_b\mathbf{w}^T}{\mathbf{w}\mathbf{S}_w\mathbf{w}^T} \right)$$

that is known as the generalized Rayleigh quotient.



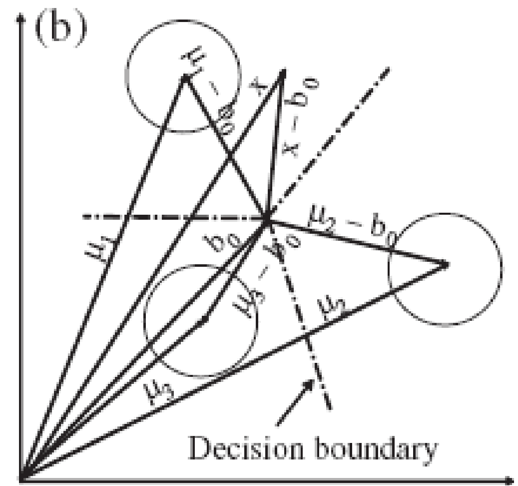
- The solution is simply a generalized eigenvalue problem:

$$\mathbf{w} = \mathbf{S}_w^{-1}\mathbf{S}_b = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2).$$

- SLP output would be:

$$y = \varphi\left(\mathbf{w}^T(\mathbf{x} - \mathbf{b}_0)\right)$$

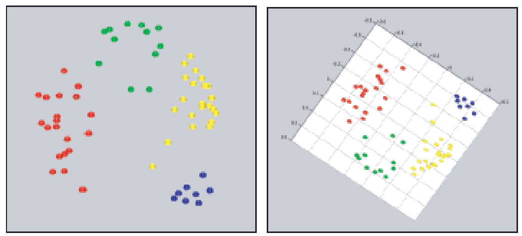
where $\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$, $\mathbf{b}_0 = \dfrac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2}$.

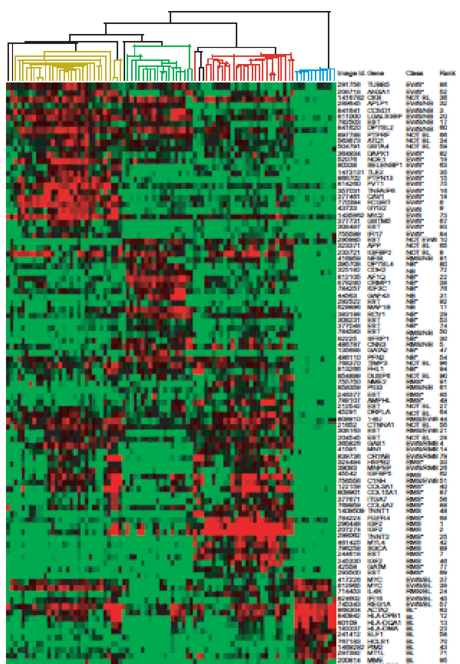


SLP real world applications

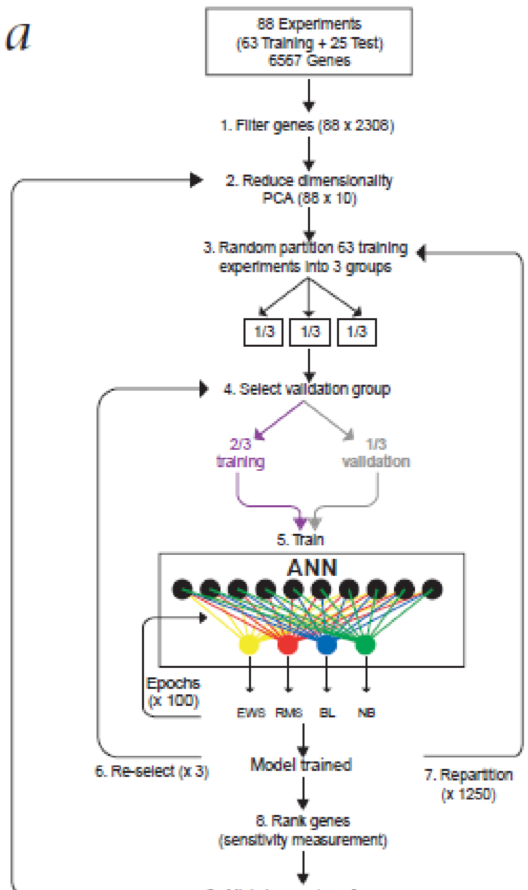
 ARTICLES

Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks

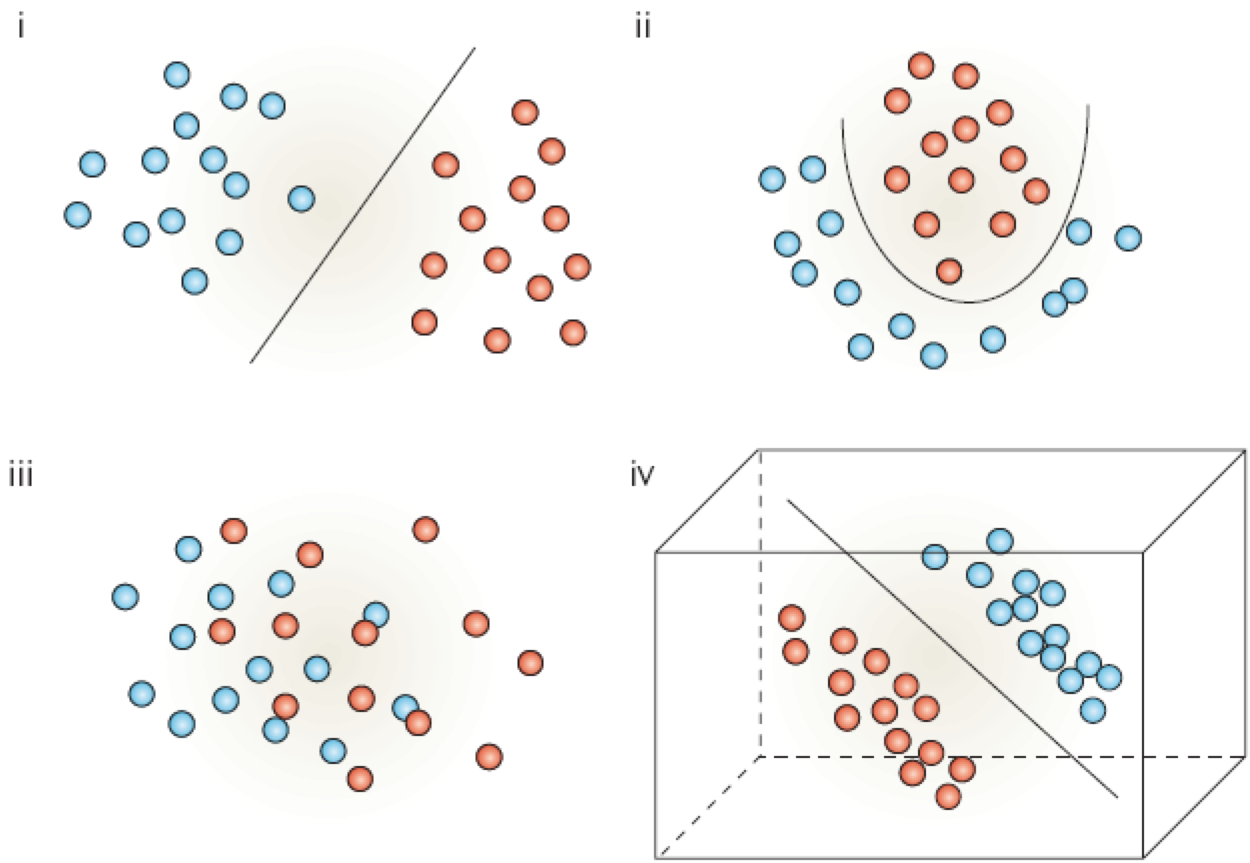




a



Approaches to nonlinear problems



Neural Networks

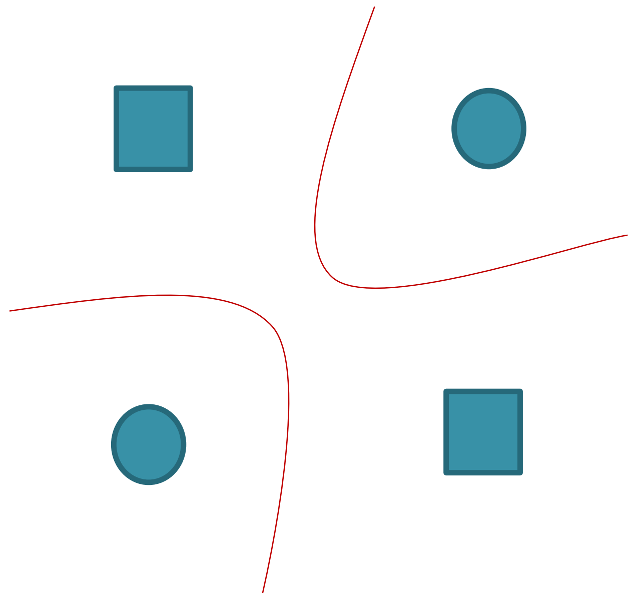
Neural Networks

Neural computation is recently developed in the information science as an outgrowth of artificial intelligence research. They differ from the conventional computer programs in that they "learn" from a set of examples rather than being programmed to get the right answer. Neural network based computational intelligence is rapidly making its way int big data sciences. .

Neural networks can be viewed as massively parallel computing systems consisting of a large number of simple processors with many interconnections. Neural network predictive models attempt to use some organizational principles in a network of weighted directed graphs in which the nodes are artificial neurons and directed edges (with weights) are connections between neuron outputs and neuron inputs.

The main advantages of neural networks are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data.

XOR (exclusive OR) problem



- True table:

input output

0+0   =0

1+1=2=0 mode 2

1+0   =1

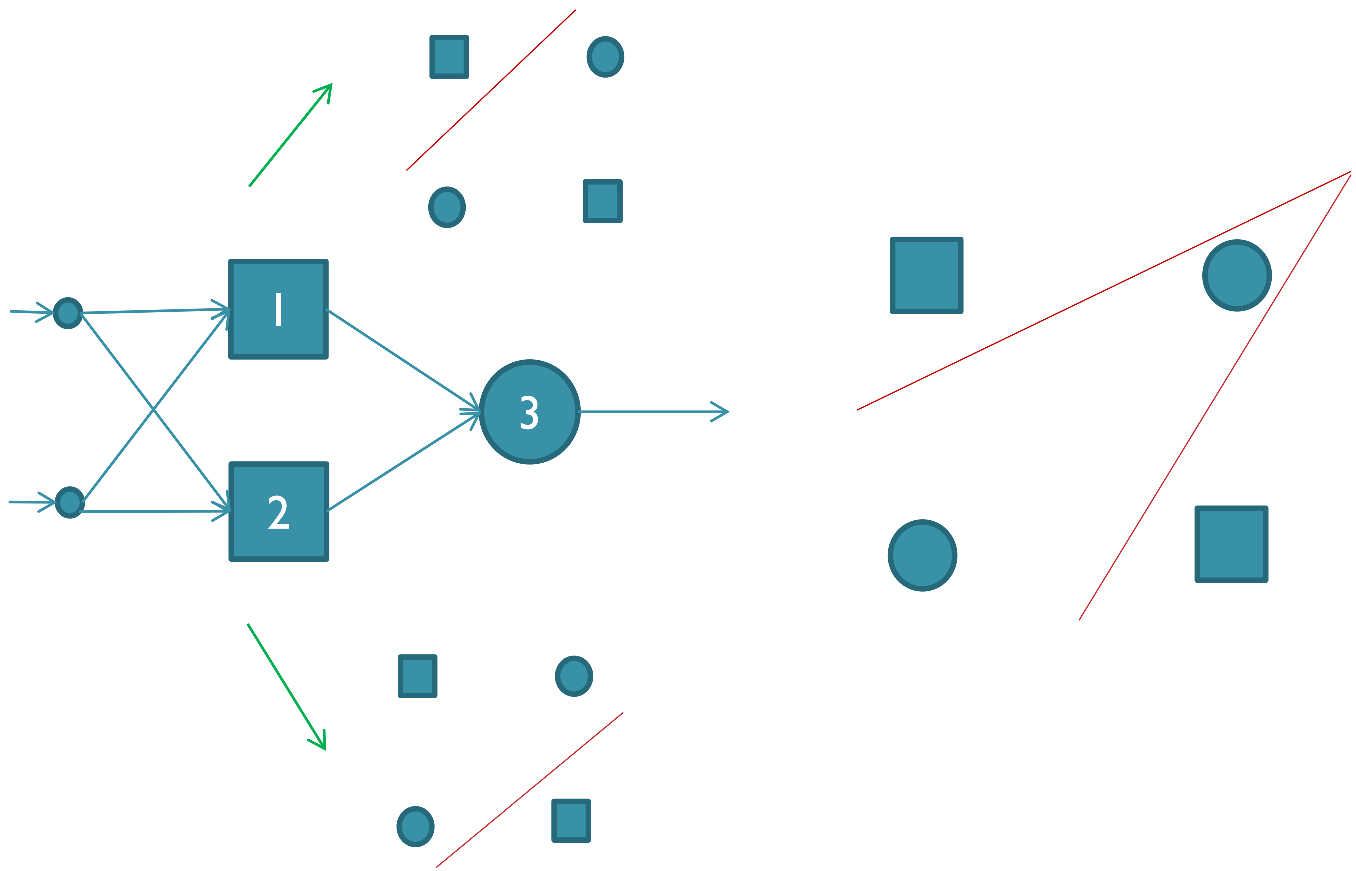
0+1   =1

- Remark: single layer perceptron provides only a linear decision boundary for a binary classification problem.

- Remark: "divide-and-conquer" strategy: (1) pseudo-class by clustering; or (2) nonlinear mapping via hidden layer.

XOR (exclusive OR) problem

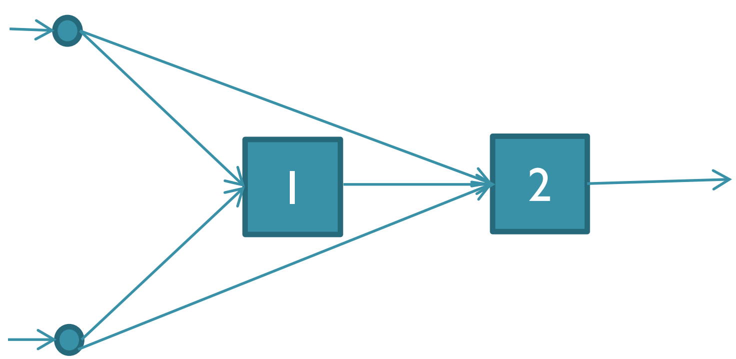
- Solution propsed by Minsky and Papert (1969)



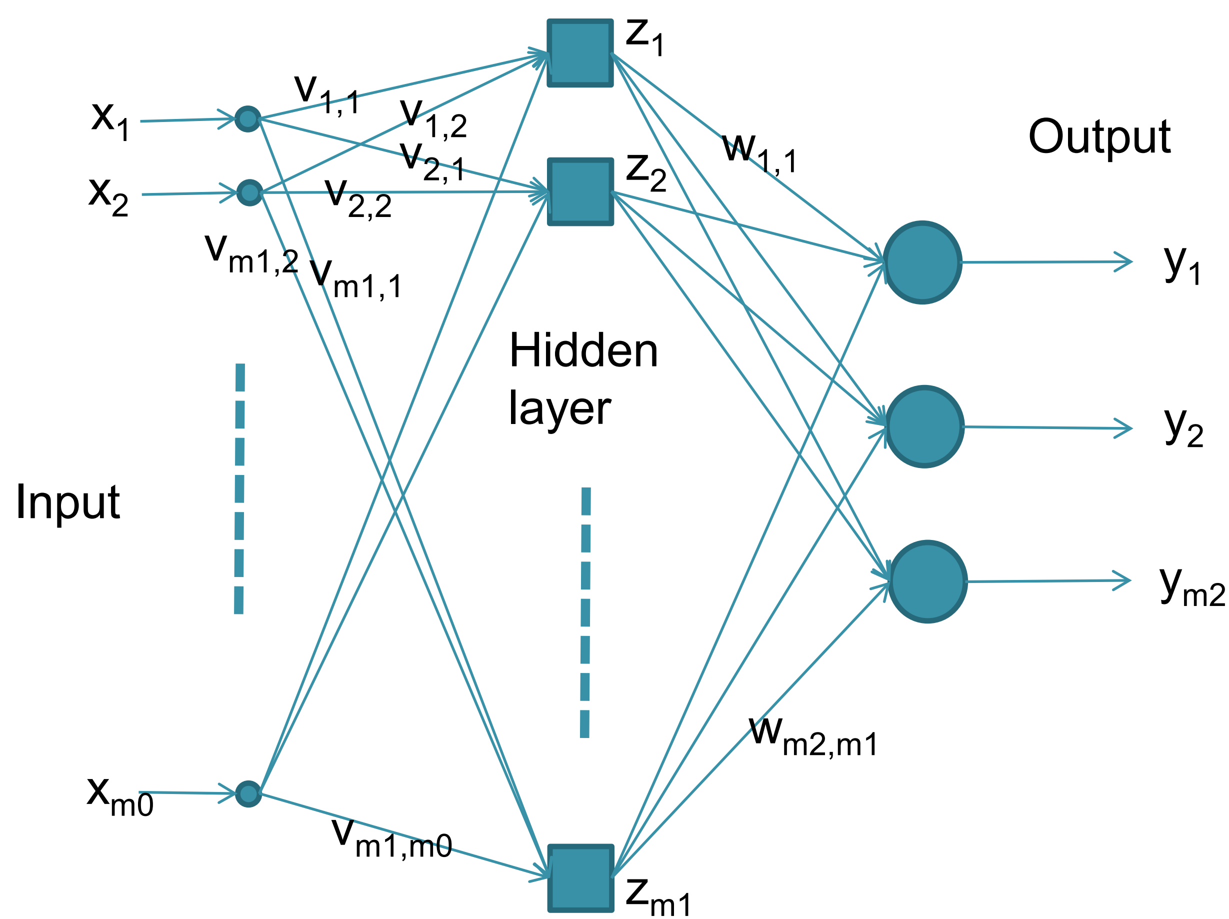
XOR (exclusive OR) problem

- Convert a nonlinearly separable problem into a linearly (pieve-wise) separable pronlem!

- How about?



Multilayer perceptron (MLP)



cascade of SLP

Properties of MLP architecture

- No connnection within a layer
- No direct connections between input and output layers
- Fully connected between layers
- Often at least 1 hidden layer(s)
- Input layer has no computational function
- # of input nodes equal dimension of feature vector
- # of output nodes equal # of classes
- # of hidden nodes can be more or less than input or output nodes

- Each hidden node is a perceptron

$$z_j = \varphi\left(\sum_{i=1}^{m0} v_{j,i} x_i + b_j\right)$$

- Each output node is a perceptron

$$y_k = \varphi\left(\sum_{j=1}^{m1} w_{k,j} z_j + b_k\right)$$

Function of MLP layers and nodes

- Hidden layer/nodes draws linear boundaries
- Output layer/nodes combines the boundaries
- Final nonlinear decision boundary can be formed

- Remark: Using (nonlinear) sigmoidal activation functions, a 3-layer MLP (feedforward neural network) can approximate any function to arbitrary accuracy, called the Property of Universal Approximation.

Practically, arbitraryily large number of hidden nodes is required.