Learning of a shallow ANN

$w_{10} = 0.01$

$w_{20} = -0.02$

$w_{11} = 0.1 \quad w_{21} = 0.3$

$w_{12} = -0.2 \quad w_{22} = 0.55$

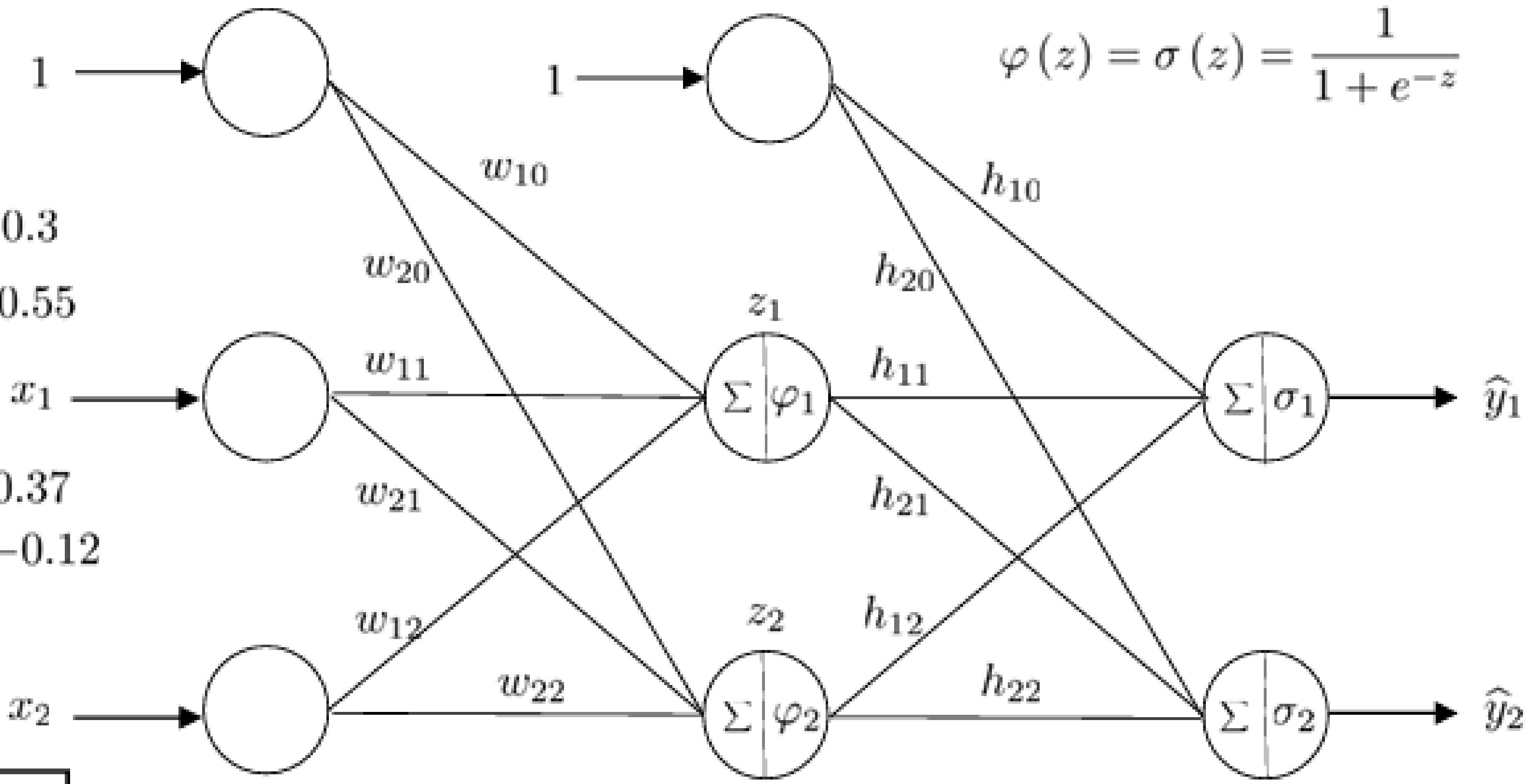
$h_{10} = 0.31$

$h_{20} = 0.27 \quad h_{11} = 0.37$

$h_{12} = 0.9 \quad h_{22} = -0.12$

$h_{21} = -0.22$

$$\varphi(z) = \sigma(z) = \frac{1}{1+e^{-z}}$$



Feature $\mathbf{x}$		Label y	
0.5	−0.5	0.9	0.1
−0.5	0.5	0.1	0.9

Input layer

Hidden layer

Output layer

Learning rate $r = 0.6$

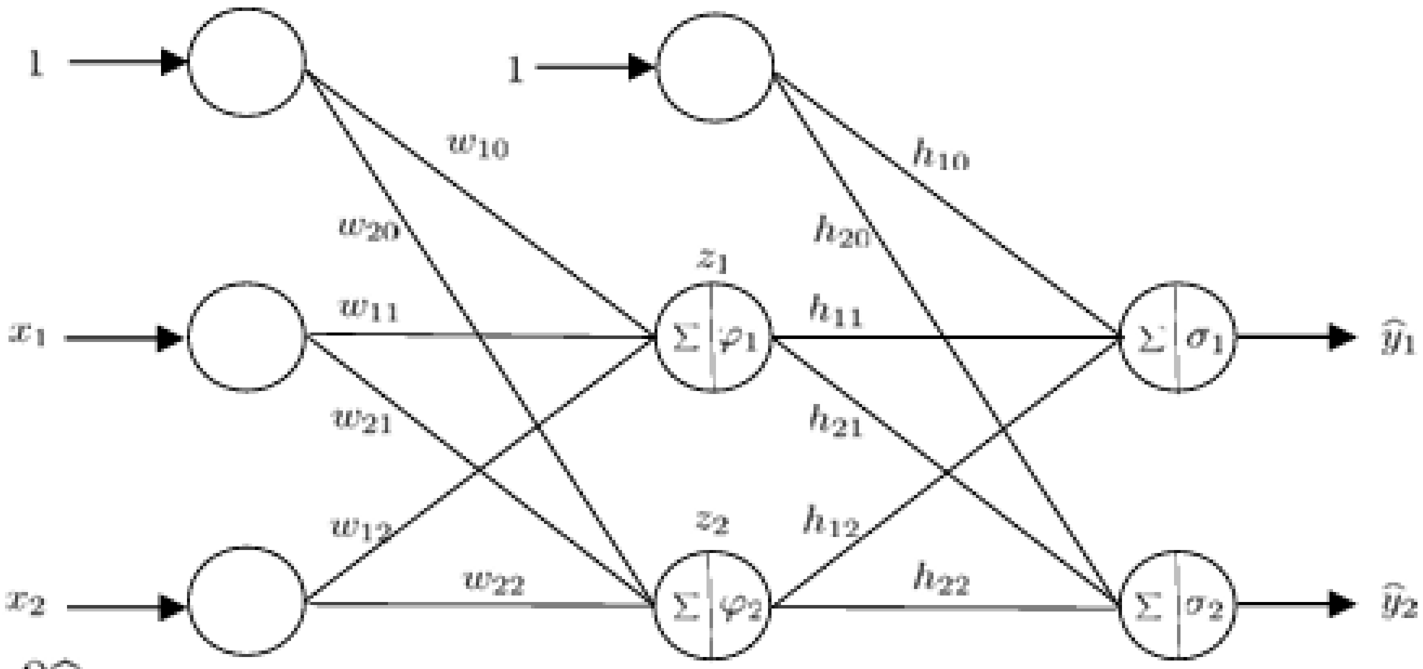
Momentum $\alpha = 0.4$

Learning of a shallow ANN

$$L = (\widehat{y}_1 - y_1)^2 + (\widehat{y}_2 - y_2)^2$$

$$\widehat{y}_1 = \sigma_1 (h_{10} + h_{11} z_1 + h_{12} z_2)$$

$$\widehat{y}_2 = \sigma_2 (h_{20} + h_{21} z_1 + h_{22} z_2)$$



$$\frac{\partial L}{\partial h_{10}} = 2(\widehat{y}_1 - y_1)\frac{\partial \widehat{y}_1}{\partial h_{10}} + 2(\widehat{y}_2 - y_2)\frac{\partial \widehat{y}_2}{\partial h_{10}}$$

$$\varphi(z) = \sigma(z) = \frac{1}{1+e^{-z}}$$

$$\frac{\partial L}{\partial h_{10}} = e_1 \sigma_1' \qquad \frac{\partial L}{\partial h_{11}} = e_1 \sigma_1' z_1 \qquad \frac{\partial L}{\partial h_{12}} = e_1 \sigma_1' z_2$$

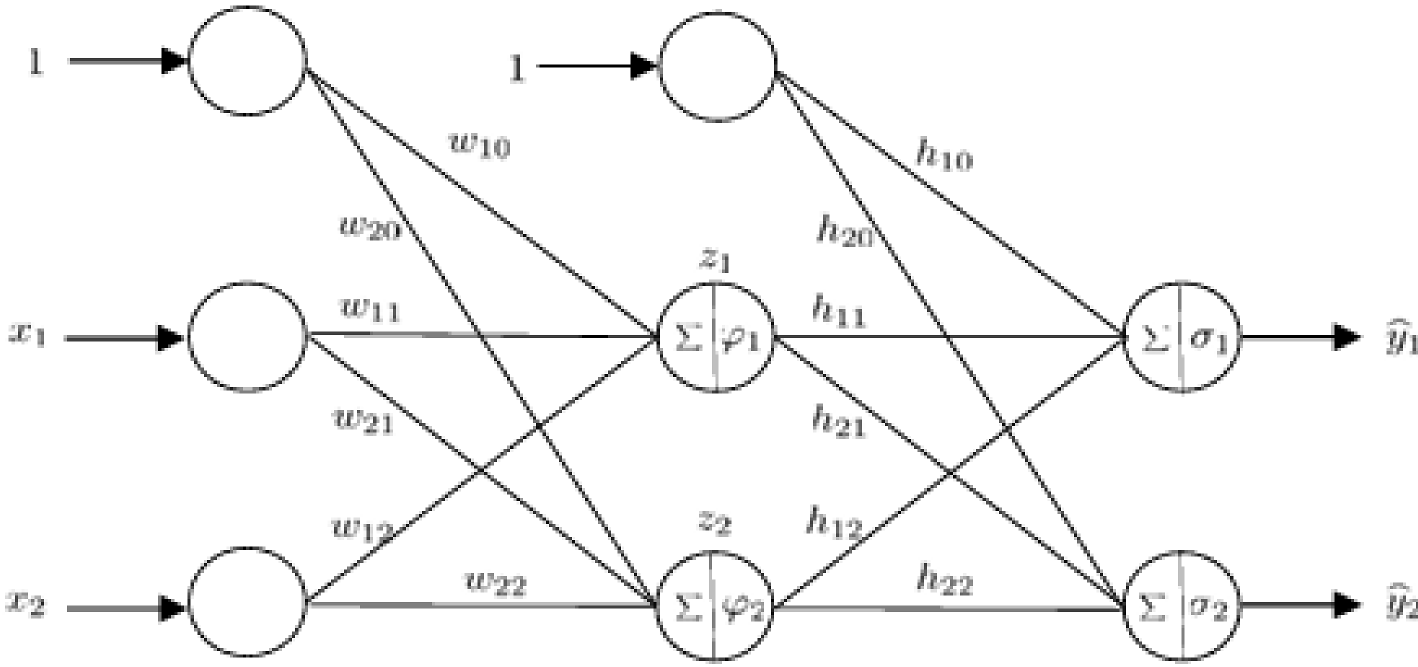
$$e_1 = 2(\widehat{y}_1 - y_1)$$

$$e_2 = 2(\widehat{y}_2 - y_2)$$

Similarly

$$\frac{\partial L}{\partial h_{20}} = e_2 \sigma_2' \qquad \frac{\partial L}{\partial h_{21}} = e_2 \sigma_2' z_1 \qquad \frac{\partial L}{\partial h_{22}} = e_2 \sigma_2' z_2$$

$$L = (\widehat{y}_1 - y_1)^2 + (\widehat{y}_2 - y_2)^2$$



$$\frac{\partial L}{\partial w_{10}} = 2\left(\widehat{y}_1 - y_1\right)\sigma_1' h_{11}\varphi_1' + 2\left(\widehat{y}_2 - y_2\right)\sigma_2' h_{21}\varphi_1'$$

$$= \left(e_1\sigma_1' h_{11} + e_2\sigma_2' h_{21}\right)\varphi_1'$$

$$\frac{\partial L}{\partial w_{11}} = \left(e_1\sigma_1' h_{11} + e_2\sigma_2' h_{21}\right)\varphi_1' x_1$$

$$\frac{\partial L}{\partial w_{12}} = \left(e_1\sigma_1' h_{11} + e_2\sigma_2' h_{21}\right)\varphi_1' x_2$$

$$\varphi(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$e_1 = 2\left(\widehat{y}_1 - y_1\right) \qquad e_2 = 2\left(\widehat{y}_2 - y_2\right)$$

Similarly

$$\frac{\partial L}{\partial w_{20}} = \left(e_1\sigma_1' h_{12} + e_2\sigma_2' h_{22}\right)\varphi_2'$$

$$\frac{\partial L}{\partial w_{21}} = \left(e_1\sigma_1' h_{12} + e_2\sigma_2' h_{22}\right)\varphi_2' x_1 \qquad \frac{\partial L}{\partial w_{22}} = \left(e_1\sigma_1' h_{12} + e_2\sigma_2' h_{22}\right)\varphi_2' x_2$$

$\mathbf{x}$	$[1 \quad 0.5 \quad -0.5]^T$
$u_1 = w_{10} + w_{11}x_1 + w_{12}x_2$	0.16
$u_2 = w_{20} + w_{21}x_1 + w_{22}x_2$	−0.145
$z_1 = \varphi_1(u_1)$	0.54
$z_2 = \varphi_1(u_2)$	0.4638
$v_1 = h_{10} + h_{11}z_1 + h_{12}z_2$	0.9272
$v_2 = h_{20} + h_{21}z_1 + h_{22}z_2$	0.09556
$\widehat{y}_1 = \sigma_1(v_1)$	0.7165
$\widehat{y}_2 = \sigma_2(v_2)$	0.52387

$e_1 = 2(\widehat{y}_1 - y_1)$	−0.366986
$e_2 = 2(\widehat{y}_2 - y_2)$	0.847744
$\sigma_1' = \widehat{y}_1(1 - \widehat{y}_1)$	0.2031
$\sigma_2' = \widehat{y}_2(1 - \widehat{y}_2)$	0.2494
$\varphi_1' = z_1(1 - z_1)$	0.2484
$\varphi_2' = z_2(1 - z_2)$	0.24869
$\delta_{h1} = -re_1\sigma_1'$	0.044726
$\delta_{h2} = -re_2\sigma_2'$	−0.126872
$\delta_{w1} = -r(e_1\sigma_1'h_{11} + e_2\sigma_2'h_{21})\varphi_1'$	0.0110443
$\delta_{w2} = -r(e_1\sigma_1'h_{12} + e_2\sigma_2'h_{22})\varphi_2'$	0.0137969

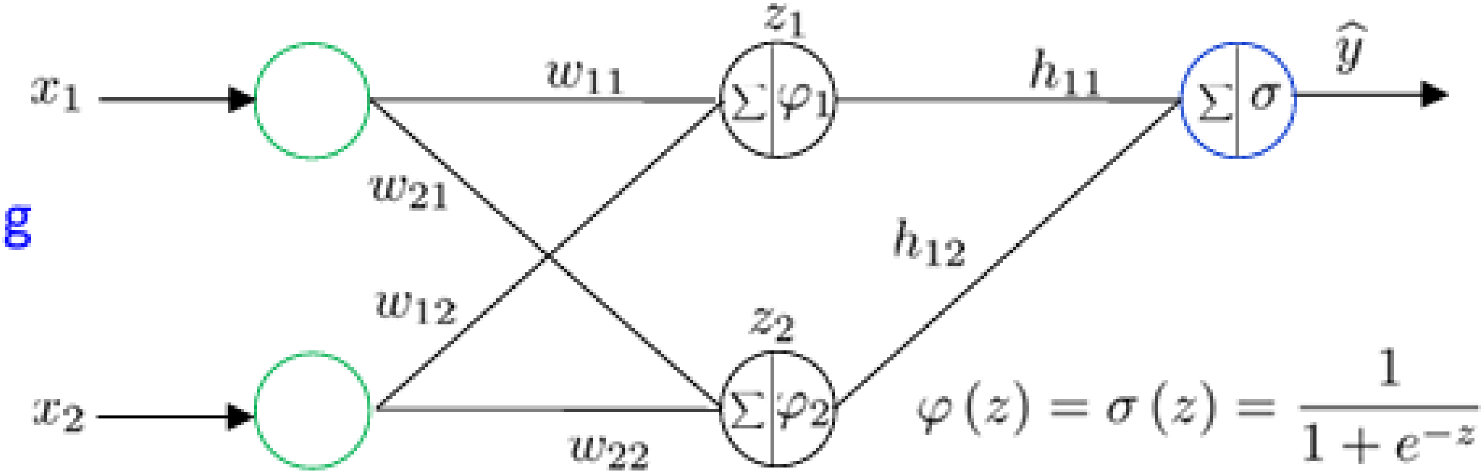
$\Delta h_{10} = \delta_{h1}$	0.044726
$\Delta h_{11} = \delta_{h1} z_1$	0.0241484
$\Delta h_{12} = \delta_{h1} z_2$	0.0207447
$\Delta h_{20} = \delta_{h2}$	-0.126872
$\Delta h_{21} = \delta_{h2} z_1$	-0.0685
$\Delta h_{22} = \delta_{h2} z_2$	-0.058845
$\Delta w_{10} = \delta_{w1}$	0.011044
$\Delta w_{11} = \delta_{w1} x_1$	0.00552215
$\Delta w_{12} = \delta_{w1} x_2$	-0.00552215
$\Delta w_{20} = \delta_{w2}$	0.0137969
$\Delta w_{21} = \delta_{w2} x_1$	0.00689847
$\Delta w_{22} = \delta_{w2} x_2$	-0.00689847

$h_{10}^{(1)} = h_{10}^{(0)} + \Delta h_{10}$	0.3547
$h_{11}^{(1)} = h_{11}^{(0)} + \Delta h_{11}$	0.3941
$h_{12}^{(1)} = h_{12}^{(0)} + \Delta h_{12}$	0.9207
$h_{20}^{(1)} = h_{20}^{(0)} + \Delta h_{20}$	0.1431
$h_{21}^{(1)} = h_{21}^{(0)} + \Delta h_{21}$	-0.2885
$h_{22}^{(1)} = h_{22}^{(0)} + \Delta h_{22}$	-0.1788
$w_{10}^{(1)} = w_{10}^{(0)} + \Delta w_{10}$	0.0210
$w_{11}^{(1)} = w_{11}^{(0)} + \Delta w_{11}$	0.1055
$w_{12}^{(1)} = w_{12}^{(0)} + \Delta w_{12}$	-0.2055
$w_{20}^{(1)} = w_{20}^{(0)} + \Delta w_{20}$	-0.0062
$w_{21}^{(1)} = w_{21}^{(0)} + \Delta w_{21}$	0.3068
$w_{22}^{(1)} = w_{22}^{(0)} + \Delta w_{22}$	0.5431

Iteration 1 completed

Weight initialization

Practical ANNs may require long computing time; good initial guess of weights accelerates convergence



$$\varphi(z) = \sigma(z) = \frac{1}{1+e^{-z}}$$

for input $\mathrm{x} = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$

$$z_1 = \varphi(w_{11}x_1 + w_{12}x_2) = \varphi_1(0) = 0.5$$

$$z_2 = \varphi(w_{21}x_1 + w_{22}x_2) = \varphi_2(0) = 0.5$$

$$\widehat{y} = \sigma(h_{11}z_1 + h_{12}z_2) = \sigma(0) = 0.5$$

$$\sigma' = 0.25 \qquad \varphi_1' = \varphi_2' = 0.25$$

Leads to $h_{11} = h_{12}$ $\quad w_{11} = w_{21}$

$w_{12} = w_{22}$ always

Initial guess: all zeroes

$$\frac{\partial L}{\partial h_{11}} = e\sigma' z_1 = 0.125e \qquad \frac{\partial L}{\partial h_{12}} = e\sigma' z_2 = 0.125e$$

$$\frac{\partial L}{\partial w_{11}} = e\sigma' h_{11}\varphi_1' x_1 = 0 \qquad \frac{\partial L}{\partial w_{21}} = e\sigma' h_{12}\varphi_2' x_1 = 0$$

$$\frac{\partial L}{\partial w_{12}} = e\sigma' h_{11}\varphi_1' x_2 = 0 \qquad \frac{\partial L}{\partial w_{22}} = e\sigma' h_{12}\varphi_2' x_1 = 0$$

Such a symmetry is unphysical

Weight initialization in ANN

Initial guess of all equal (zero or nonzero) weights lead to artificial symmetry

Remedy: randomization

Common approaches:

Gaussian or Uniform random number with zero mean, standard deviation = 1

Uniform random number in the interval $\left[-\frac{1}{\sqrt{m+n}}, -\frac{1}{\sqrt{m+n}}\right]$ (Xavier initialization)

Where $m, n:$ No. of incoming and outgoing connectors to a node

Many more approaches are available; weight initialization is an active area of ML research

1. Mean Squared Error (MSE) $E = \frac{1}{n}\sum_{i=1}^{n} L_i$ $L_i = (\widehat{y}_i - y_i)^2$

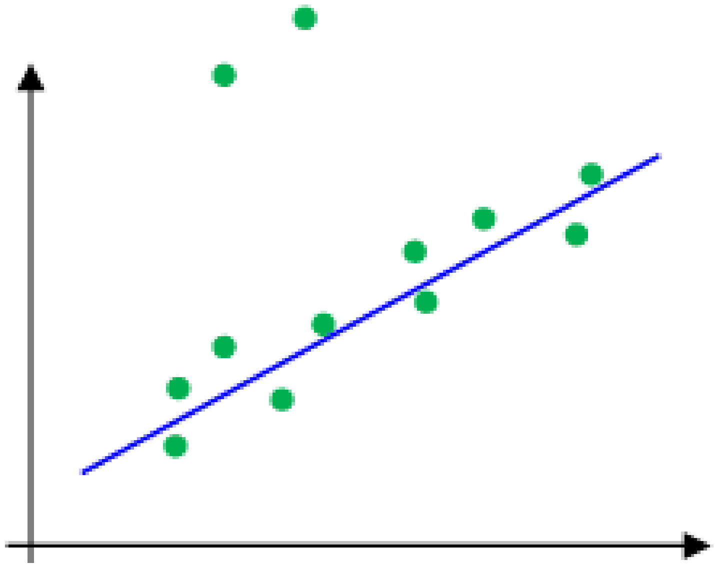
Some variations of MSE

$$E = \frac{1}{2n}\sum_{i=1}^{n} L_i$$

$$E = \sqrt{\frac{1}{2n}\sum_{i=1}^{n} L_i}$$ RMSE: Root Mean Squared Error

2. Mean Absolute Error (MAE) $E = \frac{1}{n}\sum_{i=1}^{n} |L_i|$

$$L_i = \widehat{y}_i - y_i$$



MSE is heavily influenced by outliers,
MAE avoids outliers

3. Huber Loss

$$L_i = \begin{cases} z_i^2 \ \text{ for } \ |z_i| \leq \delta \\ \\ 2\delta|z_i| - \delta^2 \ \text{ otherwise} \end{cases} \qquad z_i = \widehat{y}_i - y_i$$

At $z_i = \delta$ $\quad z_i^2 = 2\delta|z_i| - \delta^2$ and $\frac{d}{dz_i}\left(z_i^2\right) = \frac{d}{dz_i}\left(2\delta|z_i| - \delta^2\right)$

Thus Huber loss matches MSE and MAE smoothly at $z_i = \delta$

user-defined (hyperparameter)

4. Binary cross-entropy Loss

$$E = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \ln \widehat{y}_i + (1 - y_i)\ln(1 - \widehat{y}_i)\right]$$

predicted probability of being in class 0

predicted probability of being in class 1

3. Multiclass cross-entropy Loss

$$E = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(y_{ij}\ln \widehat{y}_{ij}\right)$$

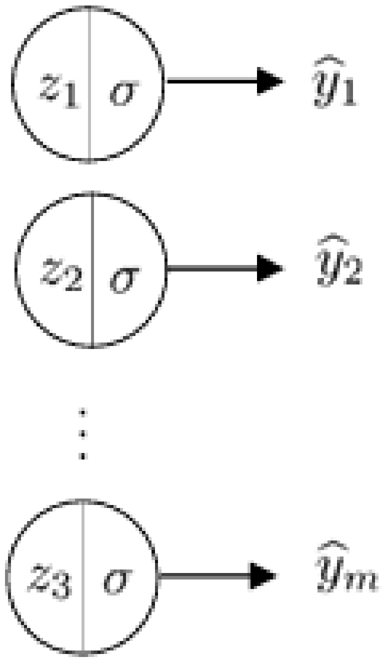
Classes
$j = 1, 2, \cdots, m$

Softmax activation if often used with cross-entropy loss

$$\widehat{y}_j = \sigma\left(z_j\right) = \frac{e^{z_j}}{\sum_{k=1}^{m} e^{z_k}}$$

brings all outputs between $0 \le \widehat{y}_j \le 1$

and enforces $\displaystyle\sum_{j=1}^{m}\widehat{y}_j = 1$



output layer

Learning rate variation

True learning rate may be obtained by solving a 1-D optimization problem

True learning rate, in most cases, diminishes as we move toward the optimum

Constant learning rate is, therefore, unphysical

We reduce learning rate via a constant factor after every k epochs

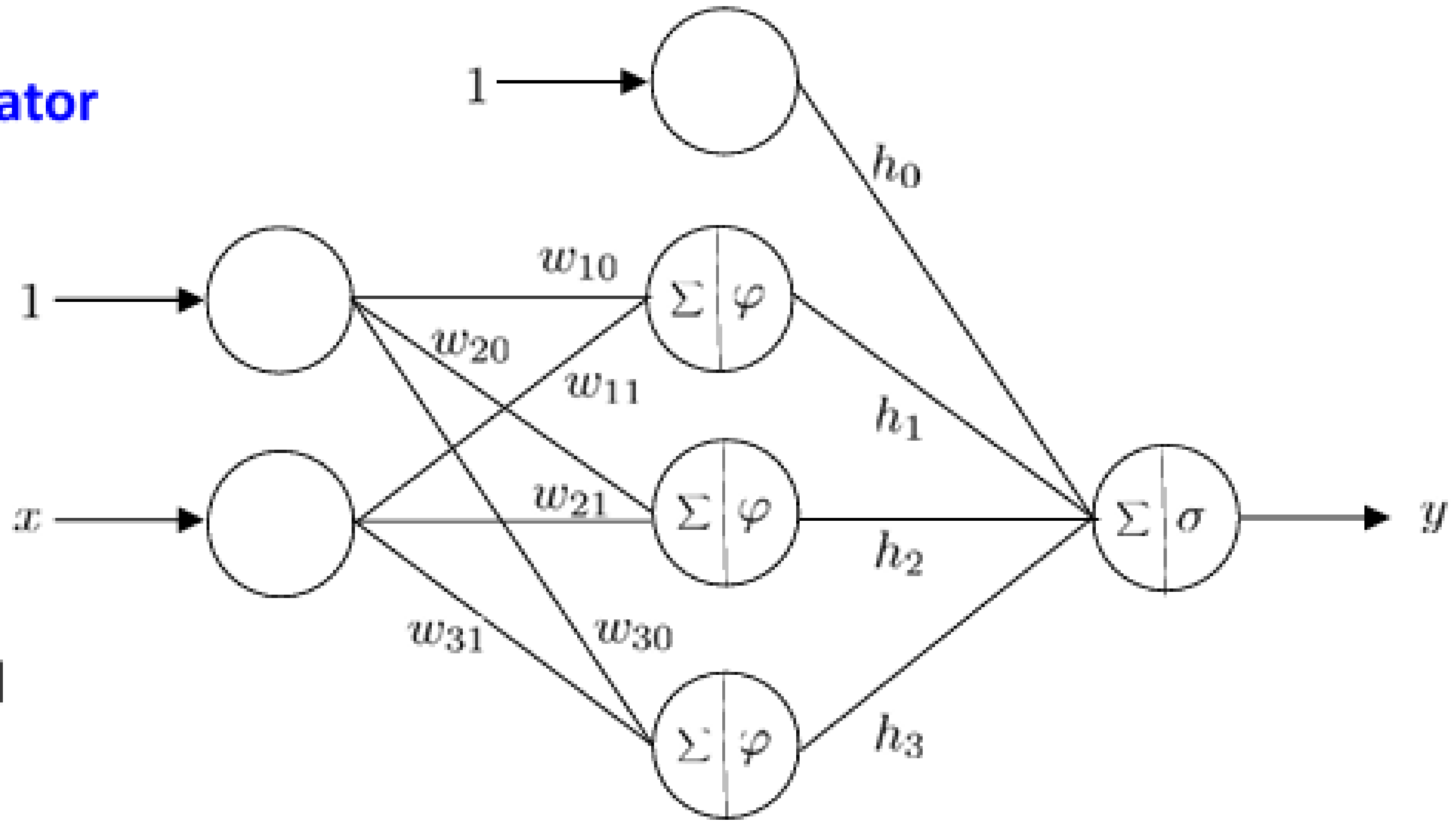
learning rate = k * learning rate

ANN as universal approximator

Consider a continuous function of one independent variable

$y = f(x)$

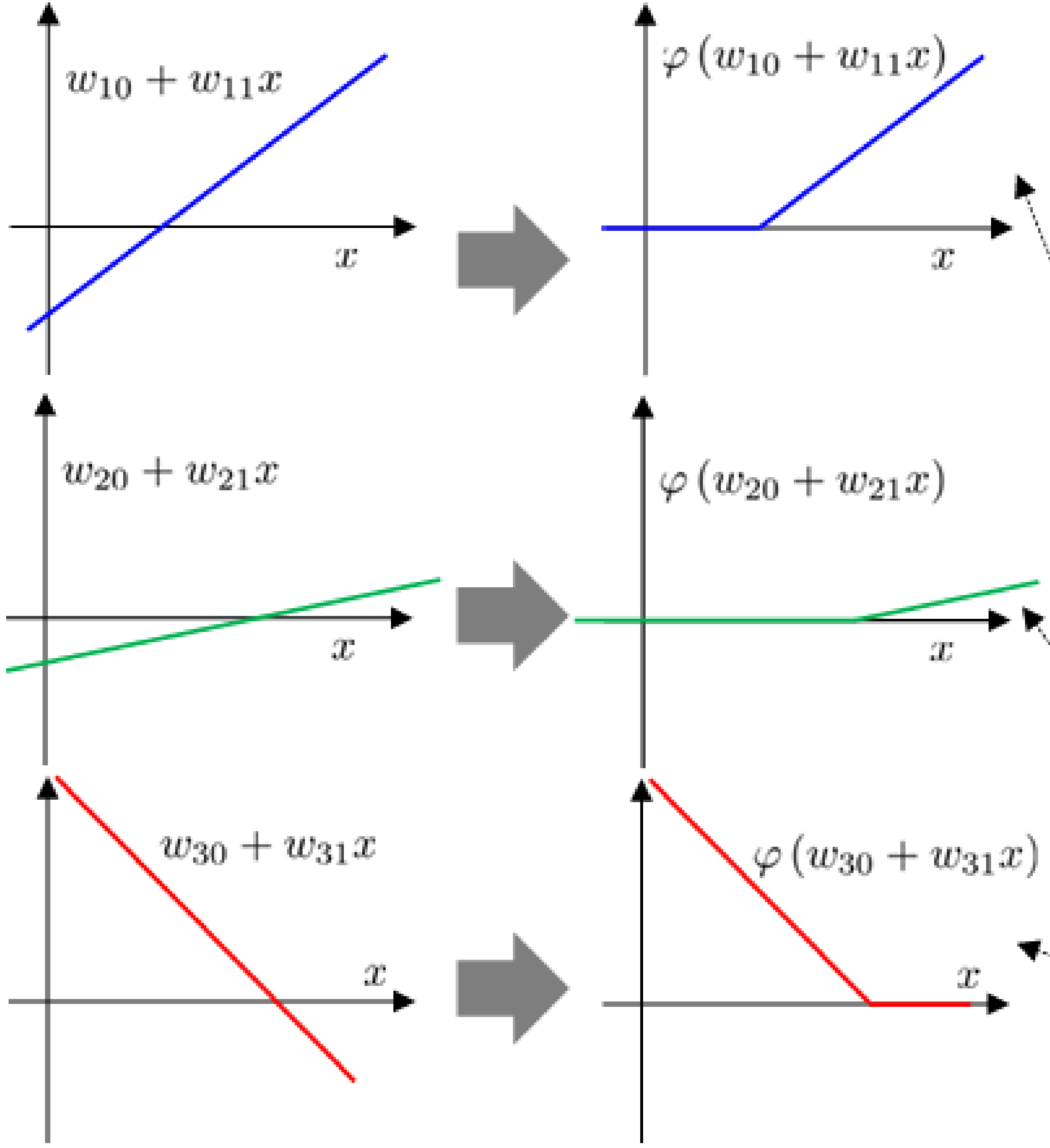
Theory suggests that a properly designed shallow (1 hidden layer) ANN should be able to approximate the above function

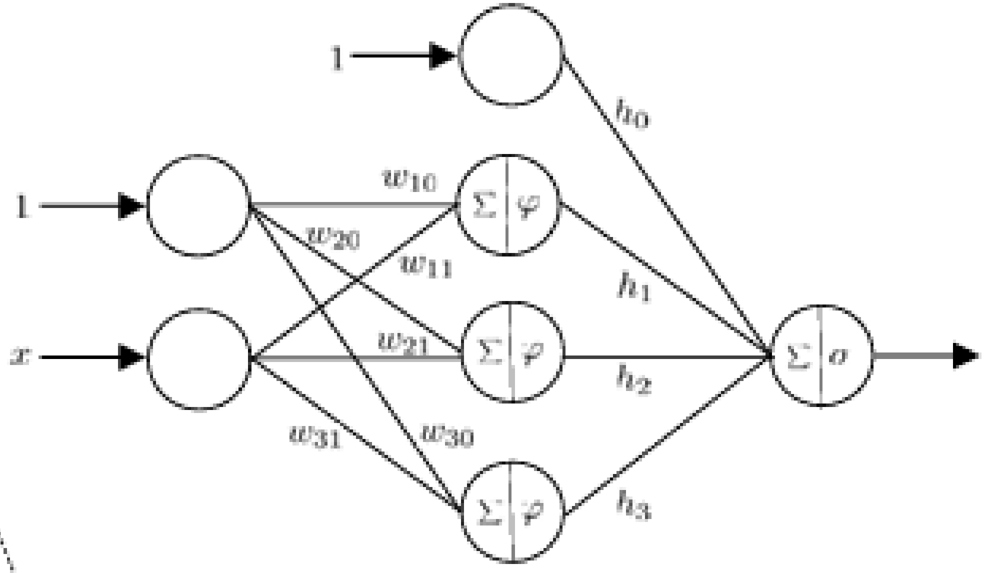


Output from i-th hidden node $z_i = \varphi_i \left(w_{i0} + w_{i1} x \right)$

ANN output $y = \sigma \left[h_0 + \sum_{i=1}^{n} h_i \varphi_i \left(w_{i0} + w_{i1} x \right) \right]$

approximates $y = f(x)$ subject to proper choice of $n, w, h, \varphi_i, \sigma$





$\varphi(z) = \max(0, z)$ $\sigma(z) = z$

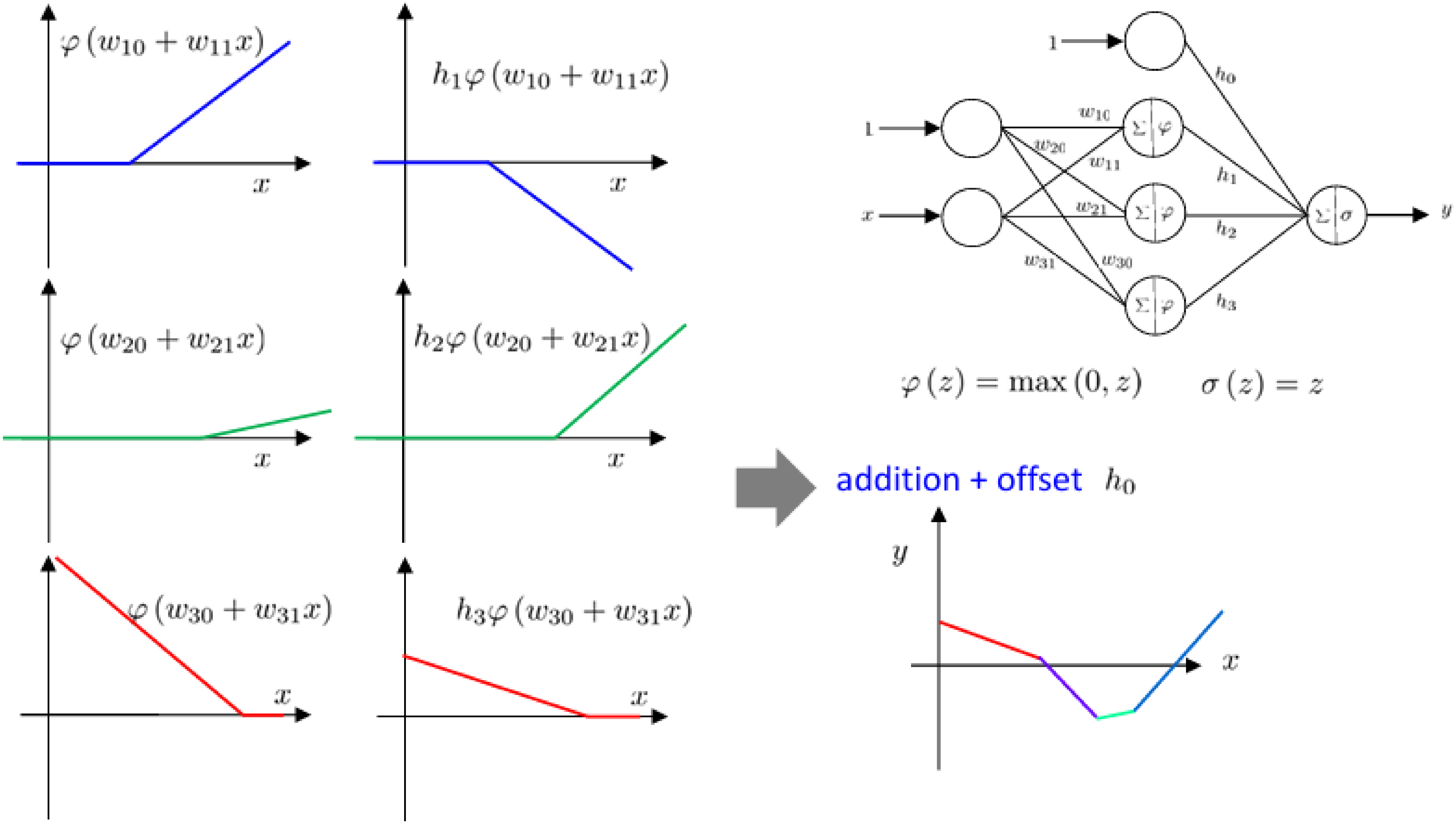
The output y

is a biased linear combination of

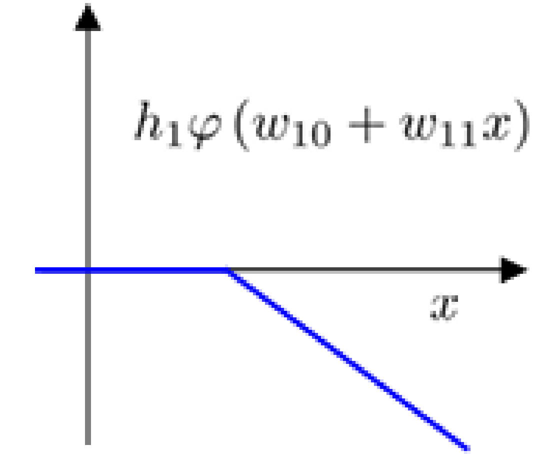
$\varphi(w_{10} + w_{11}x)$

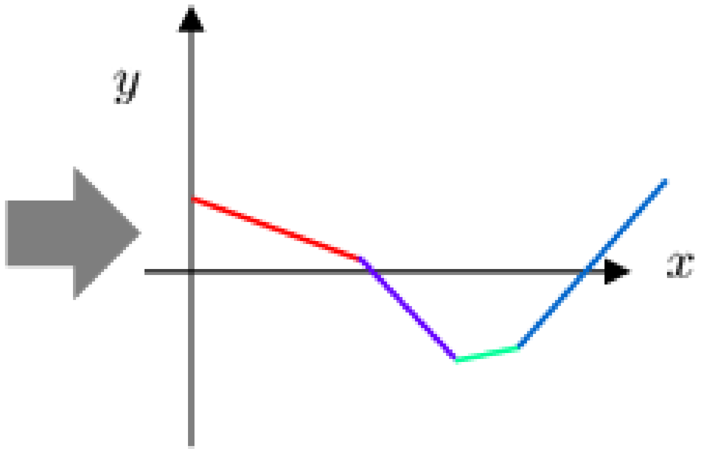
$\varphi(w_{20} + w_{21}x)$

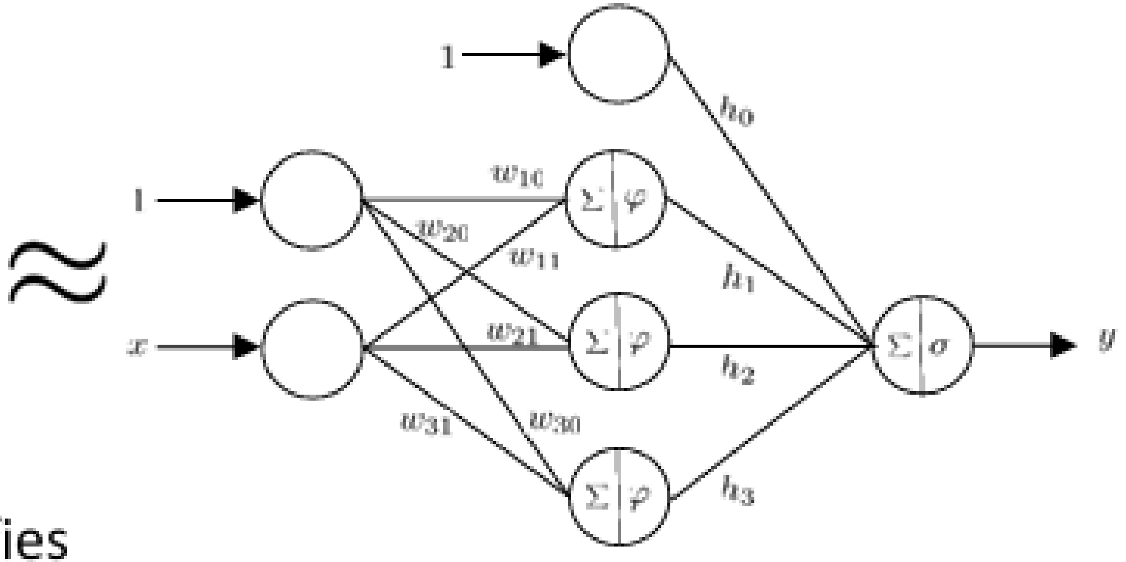
$\varphi(w_{30} + w_{31}x)$



$\varphi(w_{10}+w_{11}x)$
$h_1\varphi(w_{10}+w_{11}x)$
$\varphi(w_{20}+w_{21}x)$
$h_2\varphi(w_{20}+w_{21}x)$
$\varphi(w_{30}+w_{31}x)$
$h_3\varphi(w_{30}+w_{31}x)$
x
w_{10} w_{20} w_{11} w_{21} w_{31} w_{30}
h_0 h_1 h_2 h_3
y
$\varphi(z)=\max(0,z)$ $\sigma(z)=z$
addition + offset h_0

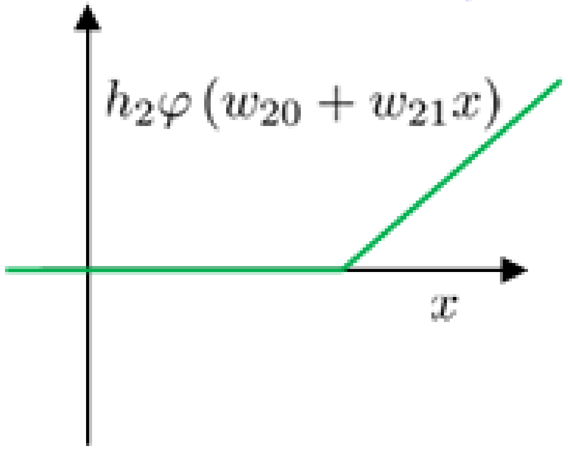




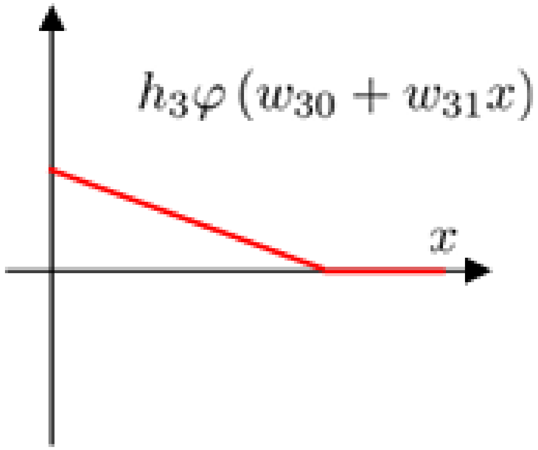


Each segment signifies on/off of some activation function

$\varphi(z) = \max(0, z) \qquad \sigma(z) = z$



The ANN with three hidden nodes represents a function with four line-segments



A continuous function may be approximated as a combination of multiple line-segments

Thus, a shallow ANN can represent a continuous function $y = f(x)$

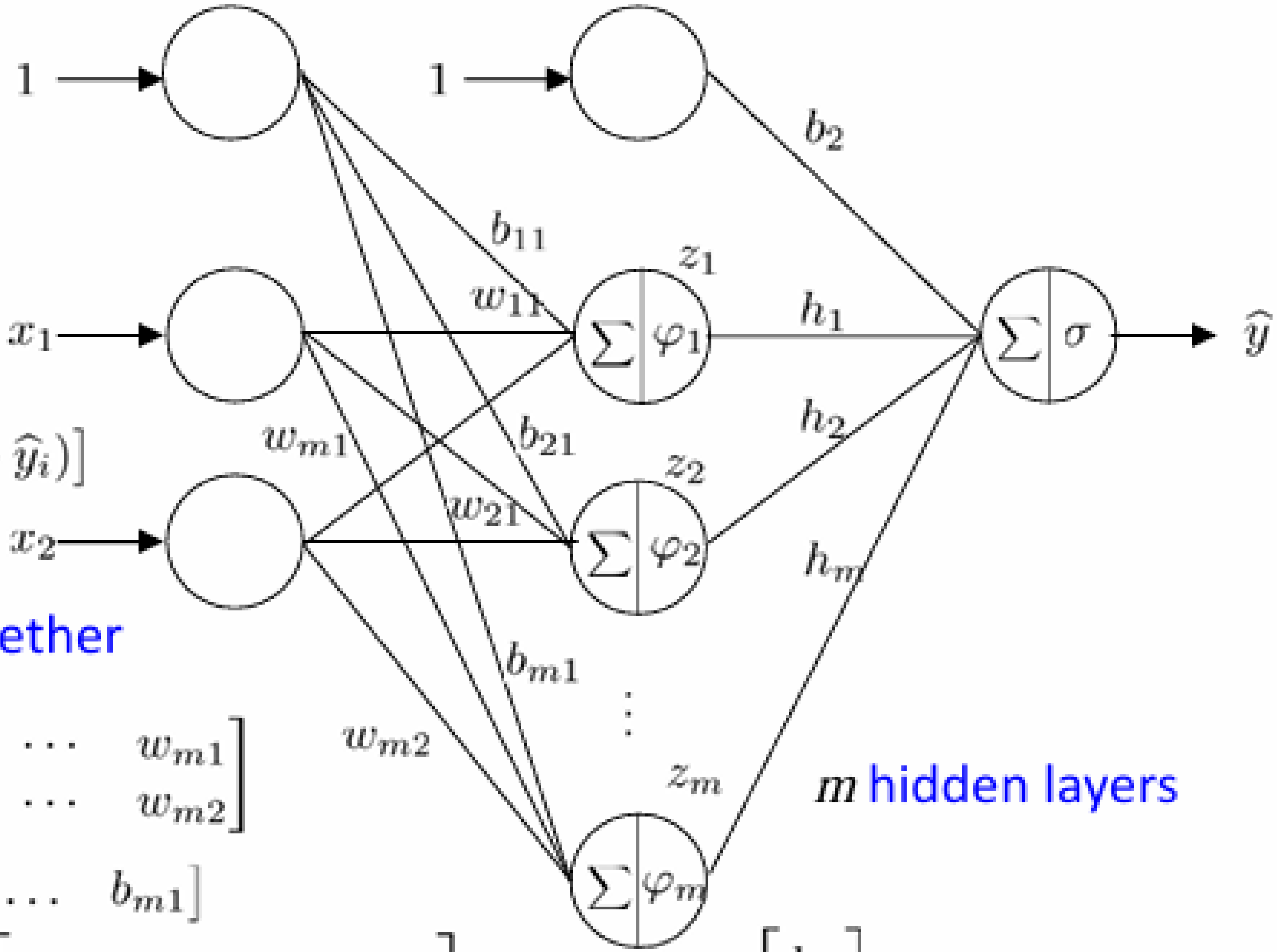
Example

Binary classification

Cross entropy cost

$$E = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \ln \widehat{y}_i + (1-y_i)\ln(1-\widehat{y}_i)\right]$$

Offline learning: using all data together



m hidden layers

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{m1} \\ w_{12} & w_{22} & \cdots & w_{m2} \end{bmatrix}$$

$$\mathbf{b}_1 = \begin{bmatrix} b_{11} & b_{21} & \dots & b_{m1} \end{bmatrix}$$

$$\mathbf{U} = \mathbf{XW} + \mathbf{b}_1 \qquad \mathbf{Z} = \varphi(\mathbf{U})$$

$$\mathbf{V} = \mathbf{Zh} + \mathbf{b}_2 \qquad \widehat{\mathbf{y}} = \sigma(\mathbf{v})$$

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1m} \\ z_{21} & z_{22} & \dots & z_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nm} \end{bmatrix} \qquad \mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_m \end{bmatrix}$$

Cross entropy cost

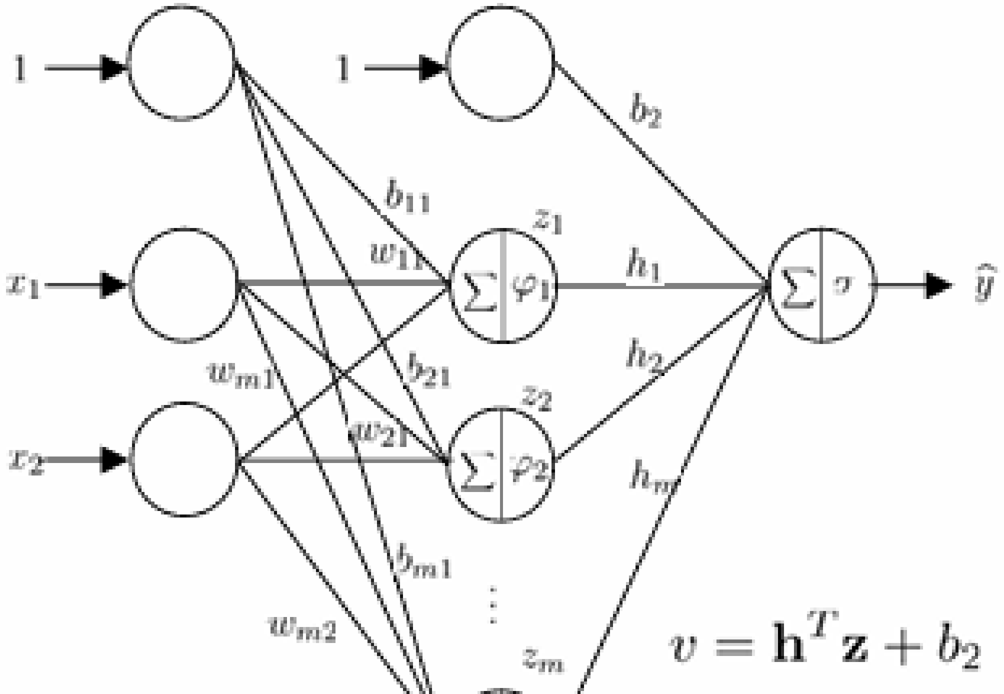
$$E = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \ln \widehat{y}_i + (1-y_i)\ln(1-\widehat{y}_i)\right]$$

$$\frac{\partial E}{\partial h_j} = \sum_{i=1}^{n}\left(e_i \frac{\partial \widehat{y}_i}{\partial h_j}\right) \qquad e_i = \frac{\widehat{y}_i - y_i}{\widehat{y}_i\left(1-\widehat{y}_i\right)}$$

$$\frac{\partial E}{\partial w_{jk}} = \sum_{i=1}^{n}\left(e_i \frac{\partial \widehat{y}_i}{\partial w_{jk}}\right)$$

$$\frac{\partial \widehat{y}_i}{\partial w_{jk}} = \frac{\partial \widehat{y}_i}{\partial v}\frac{\partial v}{\partial z_j}\frac{\partial z_j}{\partial u_j}\frac{\partial u_j}{\partial w_{jk}} = \sigma' h_j \varphi_j' x_k$$

$$\frac{\partial \widehat{y}_i}{\partial b_{j1}} = \sigma' h_j \varphi_j'$$



$$\mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{b}_1$$

$$\mathbf{z} = \varphi(\mathbf{u})$$

$$v = \mathbf{h}^T\mathbf{z} + b_2$$

$$\widehat{y} = \sigma(v) = \frac{1}{1+e^{-v}}$$