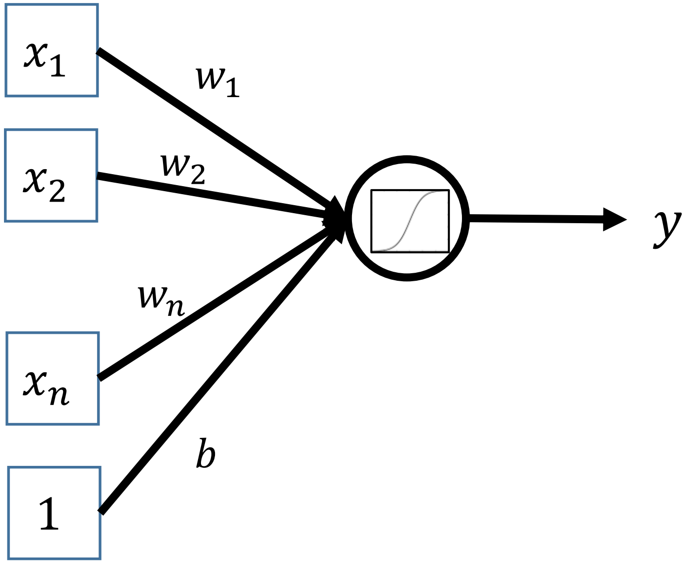
深度学习 第四讲

多层感知机与BP算法

王文中

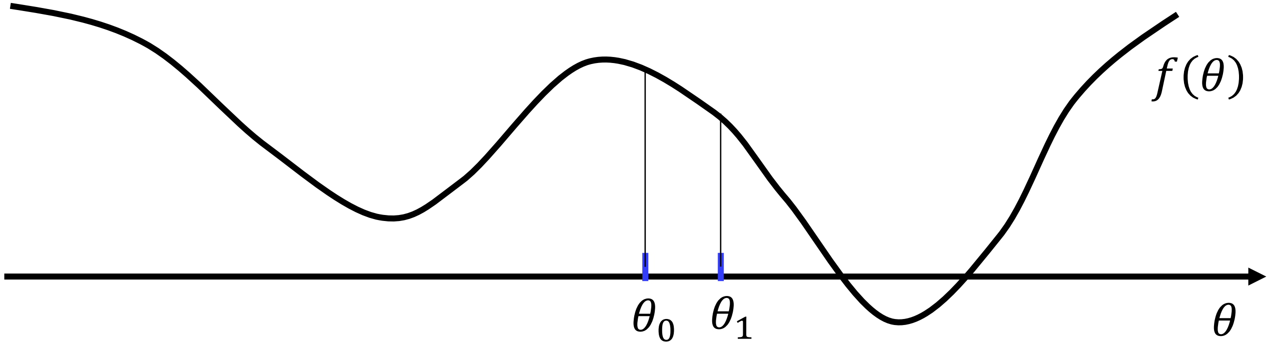
安徽大学计算机学院

回顾：神经元模型



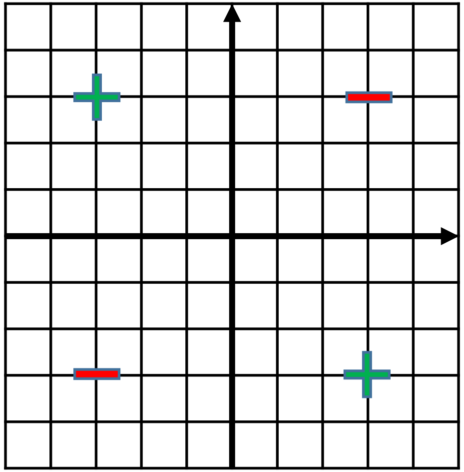
$$z=\sum_{i=1}^{n} w_i x_i+b=\mathrm{W}^{\mathrm{T}}\mathrm{X}+b, y=f(z)$$

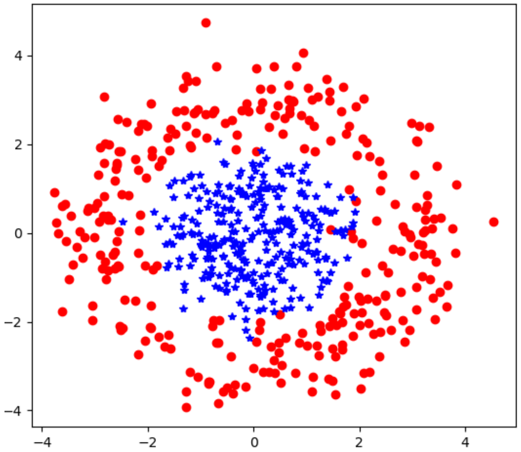
回顾：梯度下降法

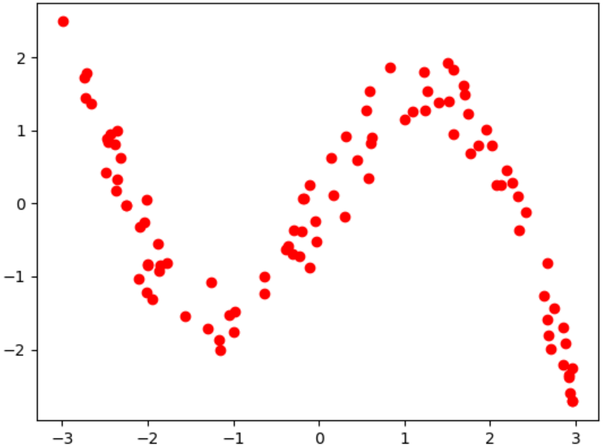


- 0：猜测一个初始值θ_0
- 1：如果θ_0是局部极小值，退出
- 2：更新θ_0为θ_1: $\theta_1 = \theta_0 - \alpha \cdot \dfrac{df}{d\theta_0}$
- 3：$\theta_0 = \theta_1$，转1

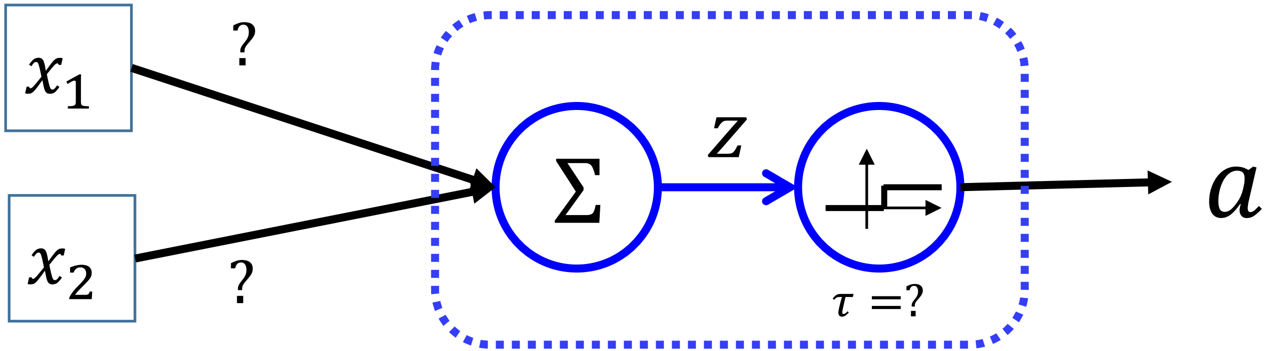
回顾：非线性问题







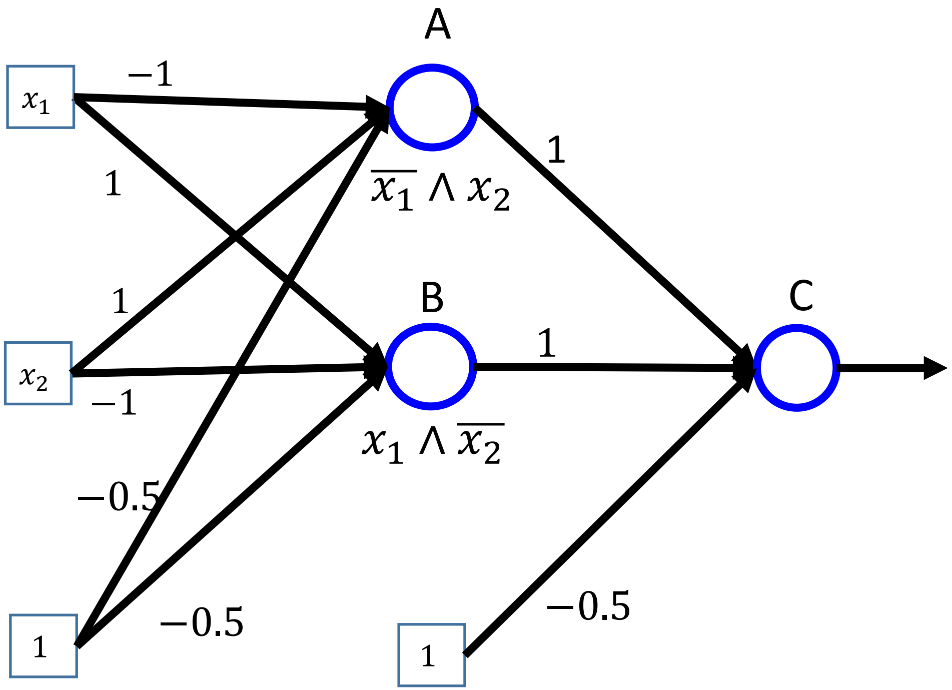
异或问题



$$y = x_1 XOR x_2 = (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2})$$

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0

用MCP神经元网络实现异或函数



$$f(x_1, x_2) = (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2})$$

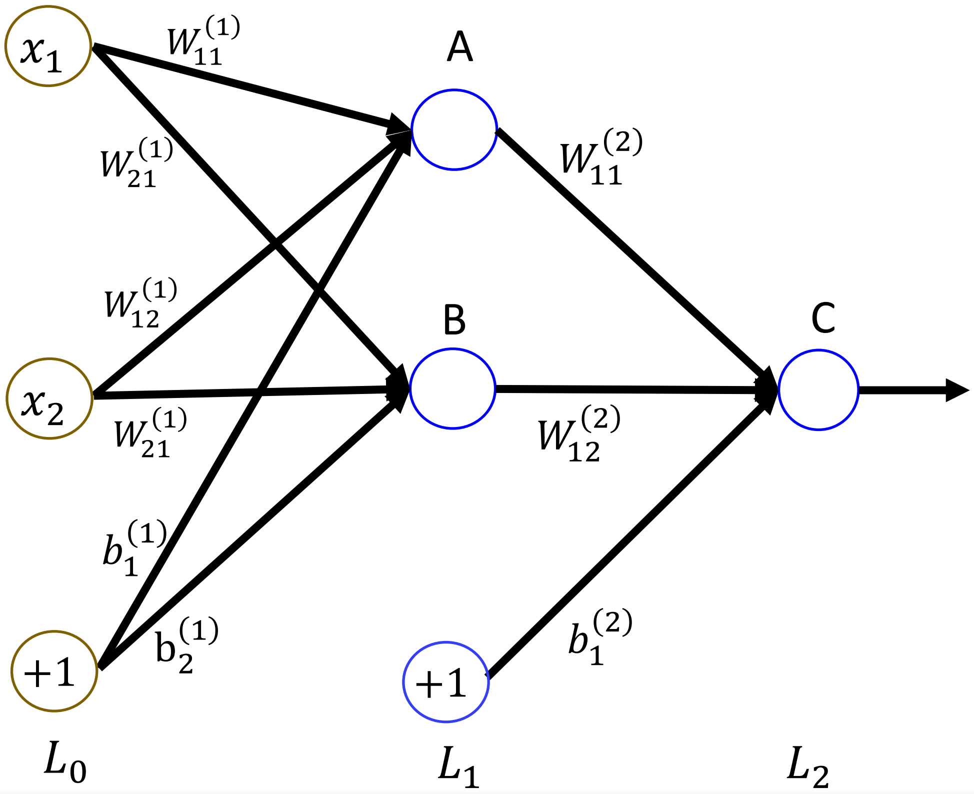
利用神经网络处理非线性问题

- 单个神经元只能实现线性分类/回归
- 使用三个MCP神经元可以实现异或运算
  - 三个MCP神经元构成一个三层神经网络：输入层、中间层以及输出层
  - 中间层神经元对输入层信号做非线性变换
  - 输出层神经元接收中间层神经元的输出结果
  - 三个神经元的参数依赖人工设计

多层感知机

Multi-Layer Perceptron

多层感知机



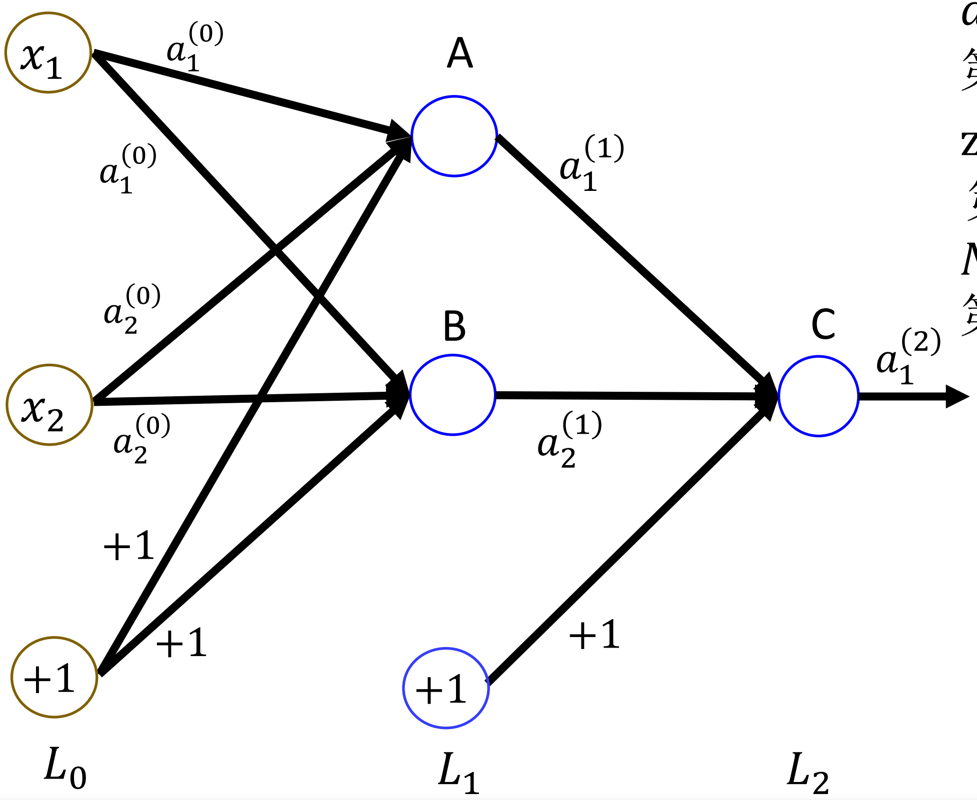
多层感知机：由多个神经元构成的一个神经元网络，这些神经元分成若干层，相邻两层的神经元之间由突触连接。

$W_{ij}^{(l)}$:
第l层第i个神经元与第$l-1$层第j个神经元之间的连接权重参数

$b_i^{(l)}$:
第l层第i个神经元的偏置参数

第一层称为输入层(Input Layer)
最后一层称为输出层(Output Layer)
中间层称为隐层(Hidden Layer)

多层感知机



$a_i^{(l)}$:
第l层第i个神经元的响应($Activation$)

$z_i^{(l)}$:
第l层第i个神经元的净响应($Activation$)

N_l:
第l层神经元数目

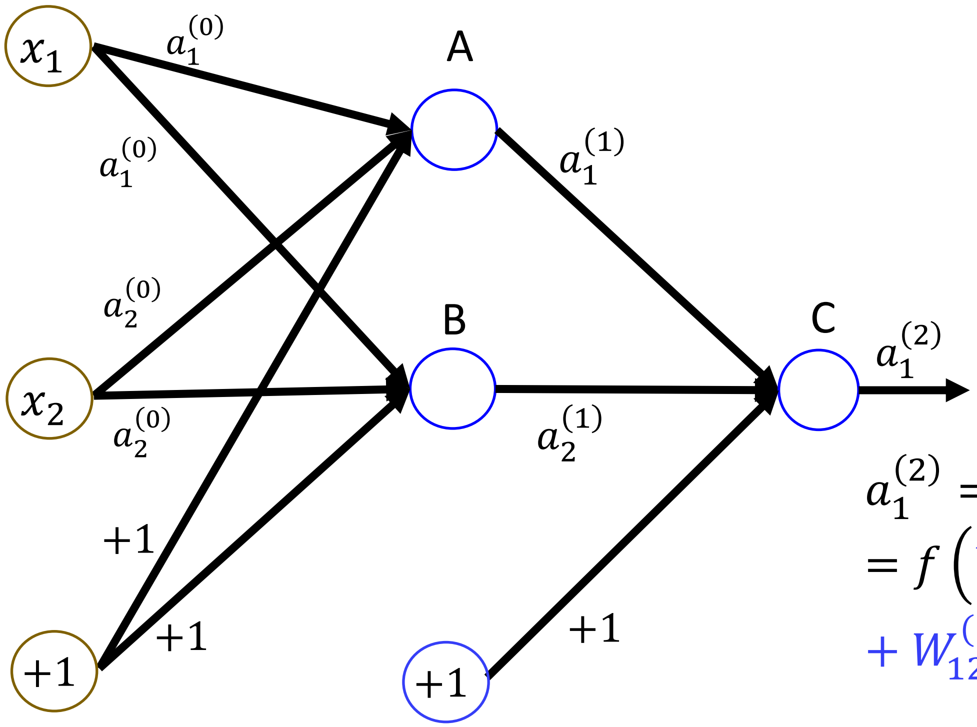
$$a_i^{(0)} = x_i, i = 1, \ldots, N_0$$

$$z_i^{(l)} = \sum_{j=1}^{N_{l-1}} W_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)}$$

$$a_i^{(l)} = f\left(z_i^{(l)}\right)$$

f: 非线性响应函数
($Activation\ Function$)

多层感知机



$$a_1^{(2)} = f\left(W_{11}^{(2)} a_1^{(1)} + W_{12}^{(2)} a_2^{(1)} + b_1^{(2)}\right)$$

$$a_1^{(1)} = f\left(W_{11}^{(1)} a_1^{(0)} + W_{12}^{(1)} a_2^{(0)} + b_1^{(1)}\right)$$

$$= f\left(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + b_1^{(1)}\right)$$

$$a_2^{(1)} = f\left(W_{21}^{(1)} a_1^{(0)} + W_{22}^{(1)} a_2^{(0)} + b_1^{(1)}\right)$$

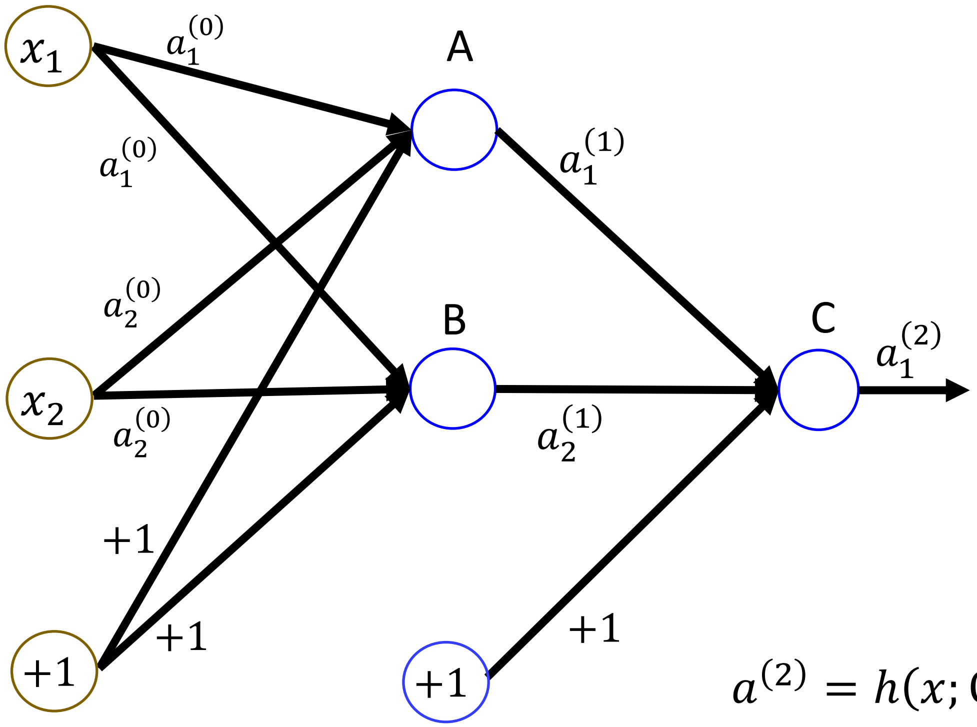
$$= f\left(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + b_1^{(1)}\right)$$

$$a_1^{(2)} = h(x_1, x_2; \Theta)$$

$$= f\left(W_{11}^{(2)} f\left(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + b_1^{(1)}\right) + W_{12}^{(2)} f\left(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + b_1^{(1)}\right) + b_1^{(2)}\right)$$

$$\Theta = \left\{W_{ij}^{(l)}, b_k^{(l)}\right\}$$

多层感知机



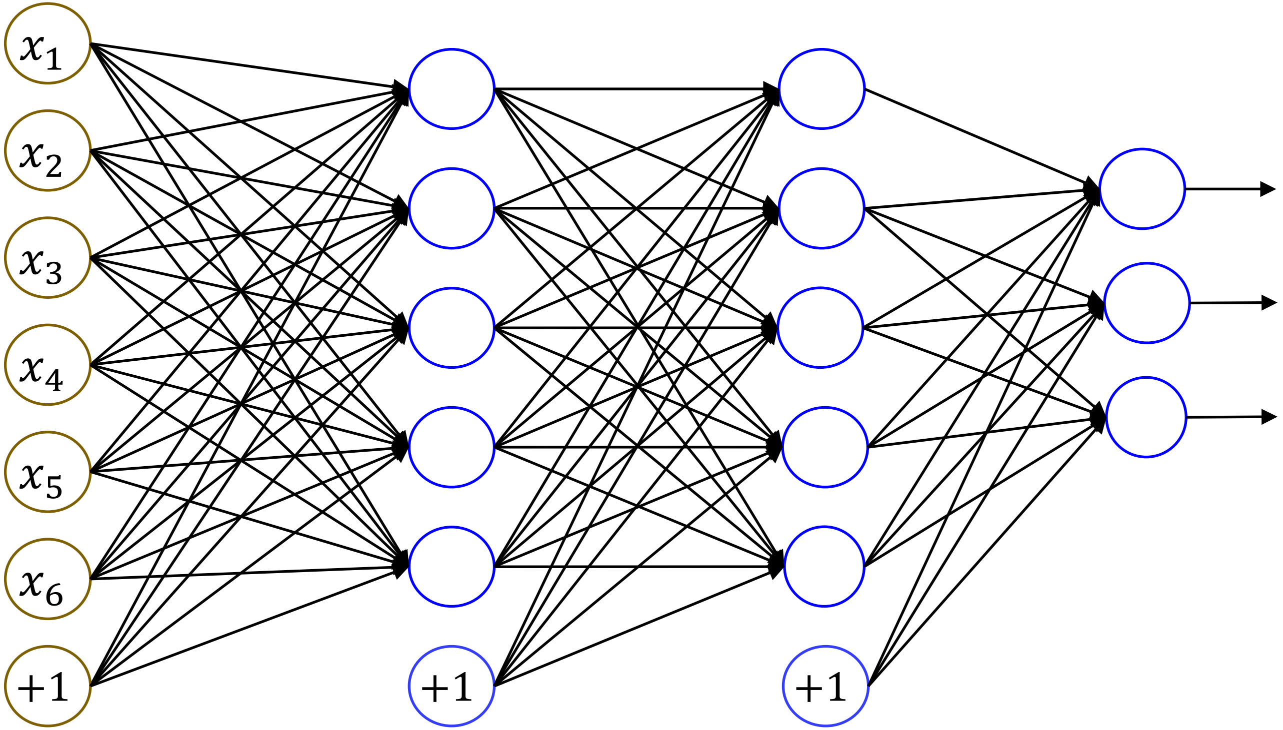
$$z^{(1)} = \left(z_1^{(1)}, z_2^{(1)}\right)^T$$
$$= \begin{pmatrix} W_{11}^{(1)} a_1^{(0)} + W_{12}^{(1)} a_2^{(0)} + b_1^{(1)} \\ W_{21}^{(1)} a_1^{(0)} + W_{22}^{(1)} a_2^{(0)} + b_1^{(1)} \end{pmatrix}$$
$$= W^{(1)} a^{(0)} + b^{(1)}$$

$$a^{(1)} = f\left(z^{(1)}\right) = \begin{pmatrix} f\left(z_1^{(1)}\right) \\ f\left(z_2^{(1)}\right) \end{pmatrix}$$

$$z^{(2)} = W^{(2)} a^{(1)} + b^{(2)}$$
$$a^{(2)} = f\left(z^{(2)}\right)$$

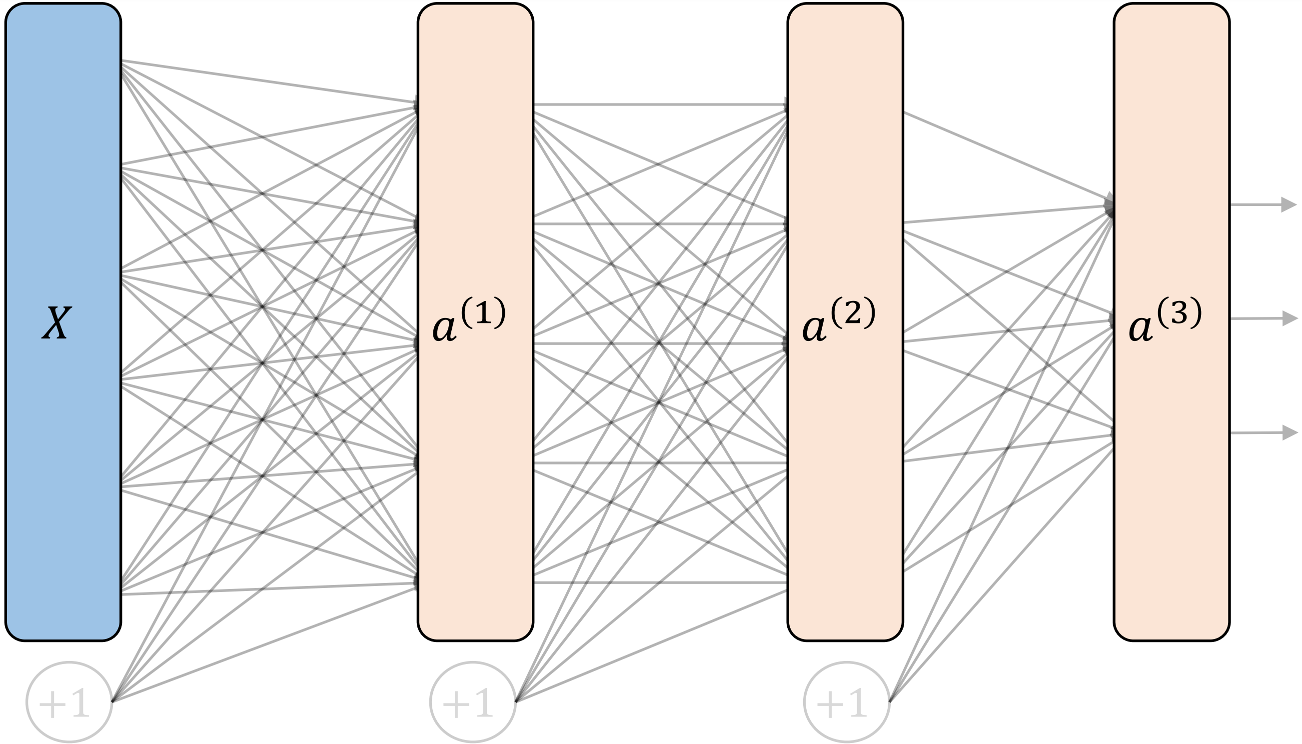
$$a^{(2)} = h(x;\Theta) = f\left(W^{(2)} f\left(W^{(1)} x + b^{(1)}\right) + b^{(2)}\right)$$
$$\Theta = \left\{W_{ij}^{(l)}, b_k^{(l)}\right\}$$

多层感知机



$$y = h(X;\Theta) = f\left(W^{(3)} f\left(W^{(2)} f\left(W^{(1)} X + b^{(1)}\right) + b^{(2)}\right) + b^{(3)}\right)$$

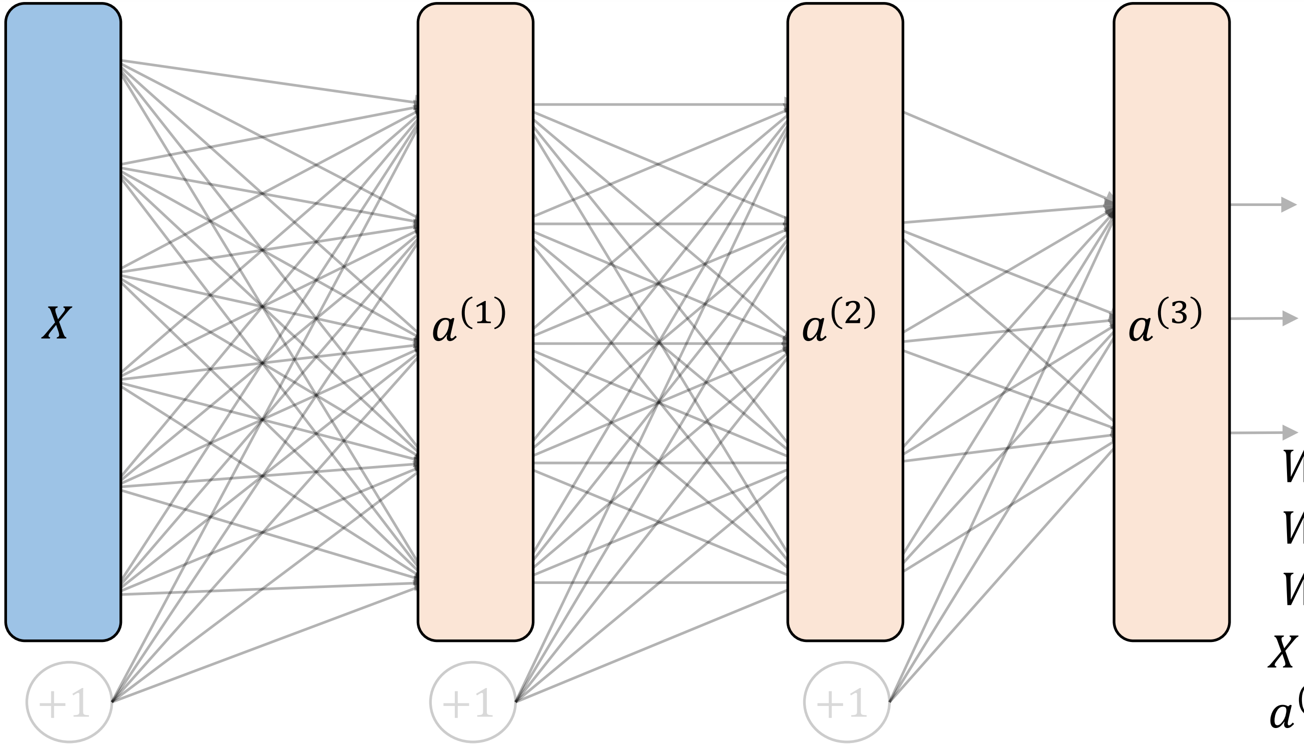
多层感知机



$$a^{(0)} = X$$
$$a^{(1)} = f\left(W^{(1)}X + b^{(1)}\right)$$
$$a^{(2)} = f\left(W^{(2)}a^{(2)} + b^{(2)}\right)$$
$$a^{(3)} = f\left(W^{(3)}a^{(3)} + b^{(3)}\right)$$

$$y = h(X;\Theta) = f\left(W^{(3)}f\left(W^{(2)}f\left(W^{(1)}X + b^{(1)}\right) + b^{(2)}\right) + b^{(3)}\right)$$

多层感知机

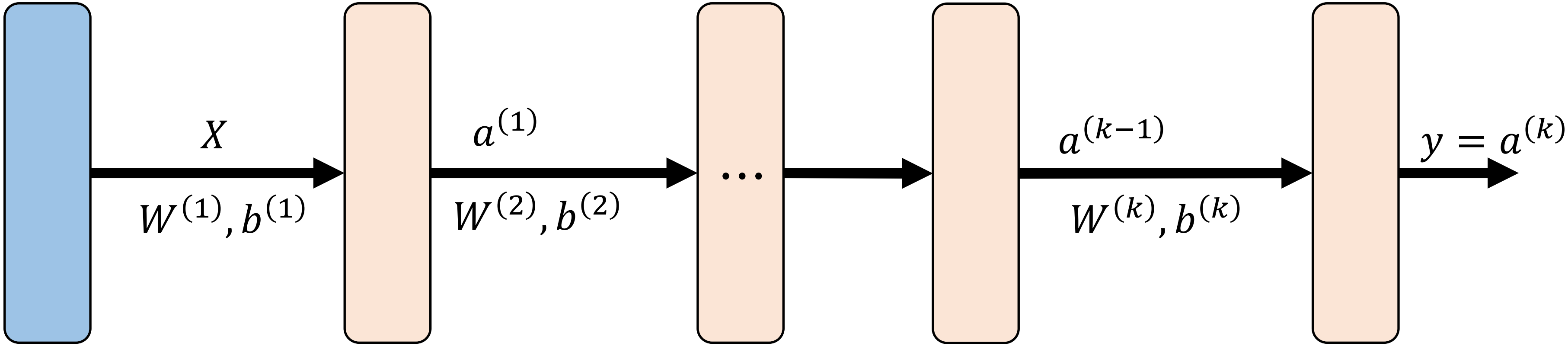


$$a^{(0)} = X$$
$$a^{(1)} = f\left(W^{(1)}X + b^{(1)}\right)$$
$$a^{(2)} = f\left(W^{(2)}a^{(2)} + b^{(2)}\right)$$
$$a^{(3)} = f\left(W^{(3)}a^{(3)} + b^{(3)}\right)$$

$$W^{(1)} \in R^{N_2 \times N_1}, b^{(1)} \in R^{N_1 \times 1}$$
$$W^{(2)} \in R^{N_3 \times N_2}, b^{(2)} \in R^{N_2 \times 1}$$
$$W^{(3)} \in R^{N_4 \times N_3}, b^{(3)} \in R^{N_3 \times 1}$$
$$X \in R^{N_1}, a^{(1)} \in R^{N_1}$$
$$a^{(2)} \in R^{N_2}, a^{(3)} \in R^{N_3}$$

$$y = h(X;\Theta) = f\left(W^{(3)}f\left(W^{(2)}f\left(W^{(1)}X + b^{(1)}\right) + b^{(2)}\right) + b^{(3)}\right)$$

多层感知机=复合函数



$$a^{(l)} = f_l\left(a^{(l)}; W^{(l)}, b^{(l)}\right) = f_l\left(W^{(l)} a^{(l-1)} + b^{(l)}\right)$$

$$y = h(X; \Theta) = f_k\left(f_{k-1}\left(\cdots f_1\left(X; W^{(1)}, b^{(1)}\right) \cdots; W^{(k-1)}, b^{(k-1)}\right); W^{(k)}, b^{(k)}\right)$$

多层感知机学习算法

用梯度下降法训练多层感知器

- 输入： $D = \{X^{(i)}, Y^{(i)}\}_{i=1}^{n}$
- 输出： 网络参数Θ
- 1. 初始化: 用随机数初始化$W^{(l)}, b^{(l)}$, $l_0 = Inf$
- 2. $While\ True$:
  - 2.1 计算损失$l_1 = l(\Theta; D)$
  - 2.2 $if\, |l_1 - l_0| < \epsilon\ Break, else\ l_0 = l_1$
  - 2.3 计算梯度: $\Delta W^{(l)}, \Delta b^{(l)}$
  - 2.4 更新参数: $W^{(l)} = W^{(l)} - \alpha \Delta W^{(l)}, b^{(l)} = b^{(l)} - \alpha \Delta b^{(l)}$
- 3. 返回Θ

损失函数

多层感知器：

$$y = h(X;\Theta) = f_k\left(f_{k-1}\left(\cdots f_2\left(X; W^{(1)}, b^{(1)}\right)\cdots; W^{(k-2)}, b^{(k-2)}\right); W^{(k-1)}, b^{(k-1)}\right)$$

- 用损失函数评估参数Θ的"好坏"
  - 根据预测的$\hat{Y}$与真实Y之间的差异计算参数Θ的损失：
  - $l(\Theta; X, Y) = l\left(\hat{Y}, Y\right), \hat{Y} = h(X;\Theta)$
- 回归问题：$Y \in R^m, \hat{Y} \in R^m$
  - 均方误差损失(MSE loss):$l(\Theta; X, Y) = \frac{1}{2}\left\|\hat{Y} - Y\right\|^2 = \frac{1}{2}\|h(X;\Theta) - Y\|^2$
- 分类问题：$Y \in \{0,1, \dots, K\}, \hat{\rho} \in (0,1)^K$
  - 对数损失(log-loss)/交叉熵损失(Cross-Entropy Loss)
  - $l(\Theta; X, Y) = -\sum_{j=1}^{K} 1(Y = j) \ln\left(\hat{\rho}_j\right), \rho = h(X;\Theta)$

计算梯度$\partial l/\partial\Theta$

$$l(\Theta;X,Y)=l\left(\hat{Y},Y\right)$$

$$\hat{Y}=h(X;\Theta)=f_k\left(f_{k-1}\left(\cdots f_1\left(X;W^{(1)},b^{(1)}\right)\cdots;W^{(k-1)},b^{(k-1)}\right);W^{(k)},b^{(k)}\right)$$

$$\Delta W^{(l)}=\frac{\partial l}{\partial W^{(l)}}=?$$

$$\Delta b^{(l)}=\frac{\partial l}{\partial b^{(l)}}=?$$

计算梯度$\partial l / \partial \Theta$

$$l(\Theta; X, Y) = l\left(\hat{Y}, Y\right)$$

$$\hat{Y} = h(X; \Theta) = f_k\left(f_{k-1}\left(\cdots f_l\left(a^{(l)}; W^{(l)}, b^{(l)}\right) \cdots\right); \cdots\right)$$

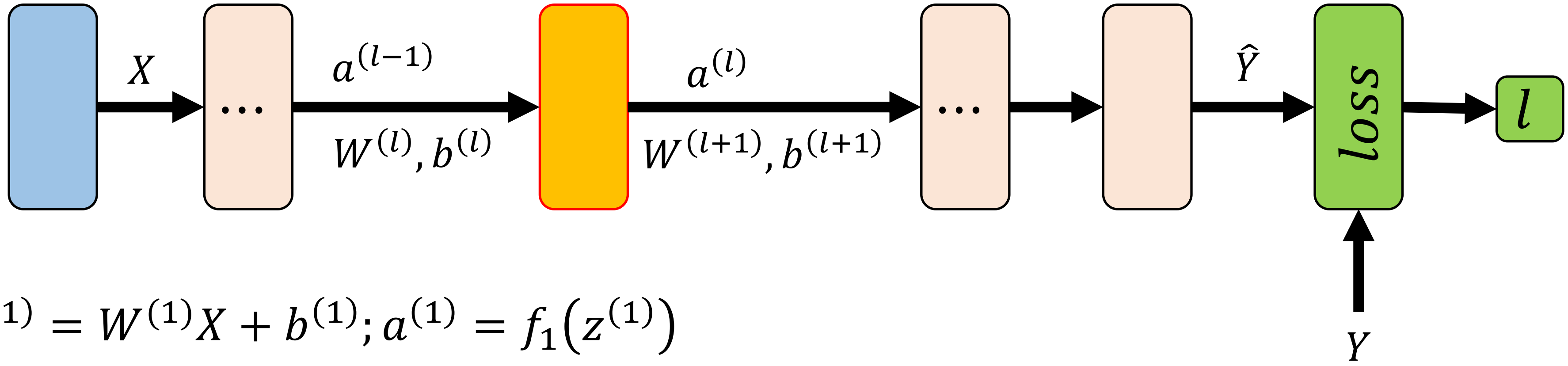
$$\Delta W^{(l)} = \frac{\partial l}{\partial W^{(l)}} = ?$$

$$\Delta b^{(l)} = \frac{\partial l}{\partial b^{(l)}} = ?$$

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}$$

$$a^{(l)} = f_l\left(z^{(l)}\right)$$

计算梯度$\partial l/\partial\Theta$

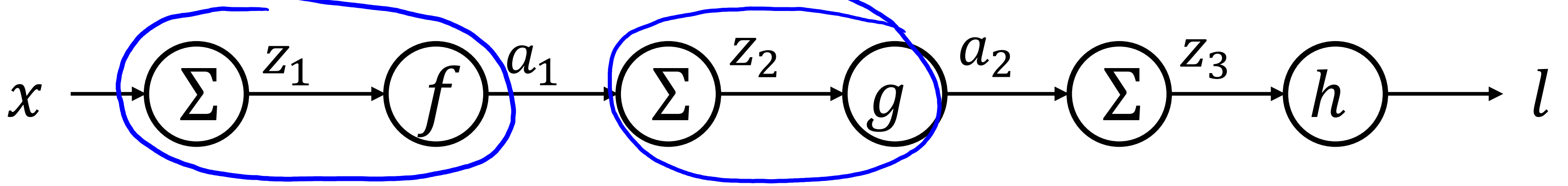


$$z^{(1)} = W^{(1)}X + b^{(1)}; a^{(1)} = f_1\left(z^{(1)}\right)$$
$$\vdots$$
$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}; a^{(l)} = f_l\left(z^{(l)}\right)$$
$$z^{(l+1)} = W^{(l+1)}a^{(l)} + b^{(l+1)}; a^{(l+1)} = f_{l+1}\left(z^{(l+1)}\right)$$
$$\vdots$$
$$\hat{Y} = f_K\left(z^{(K)}\right); l = l(\hat{Y}, Y)$$

$$\frac{\partial l}{\partial W^{(l)}} = \frac{\partial l}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial W^{(l)}}$$

复合函数求导

$$l = h(w_3 \times g(w_2 \times f(w_1 \times x + b)))$$



$l = h(z_3);$

$z_3 = w_3 \times a_2;$

$a_2 = g(z_2);$

$z_2 = w_2 \times a_1;$

$a_1 = f(z_1);$

$z_1 = w_1 \times x + b;$

复合函数求导

$$l = h(w_3 \times g(w_2 \times f(w_1 \times x + b)))$$

$$l = h(z_3);$$
$$z_3 = w_3 \times a_2;$$
$$a_2 = g(z_2);$$
$$z_2 = w_2 \times a_1;$$
$$a_1 = f(z_1);$$
$$z_1 = w_1 \times x + b;$$

$$\frac{\partial l}{\partial z_3} = h'(z_3); \frac{\partial z_3}{\partial w_3} = a_2, \frac{\partial z_3}{\partial a_2} = w_3;$$

$$\frac{\partial a_2}{\partial z_2} = g'(z_2); \frac{\partial z_2}{\partial w_2} = a_1; \frac{\partial z_2}{\partial a_1} = w_2;$$

$$\frac{\partial a_1}{\partial z_1} = f'(z_1); \frac{\partial z_1}{\partial w_1} = x; \frac{\partial z_1}{\partial b} = 1$$

复合函数求导

$$l = h(w_3 \times g(w_2 \times f(w_1 \times x + b)))$$

$$\frac{\partial l}{\partial z_3} = h'(z_3); \frac{\partial z_3}{\partial w_3} = a_2, \frac{\partial z_3}{\partial a_2} = w_3;$$

$$\frac{\partial a_2}{\partial z_2} = g'(z_2); \frac{\partial z_2}{\partial w_2} = a_1; \frac{\partial z_2}{\partial a_1} = w_2;$$

$$\frac{\partial a_1}{\partial z_1} = f'(z_1); \frac{\partial z_1}{\partial w_1} = x; \frac{\partial z_1}{\partial b} = 1$$

$$\frac{\partial l}{\partial w_3} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial w_3} = h'(z_3) \times a_2,$$

$$\frac{\partial l}{\partial w_2} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial w_2} = h'(z_3) \times w_3 \times g'(z_2) \times a_1;$$

$$\frac{\partial l}{\partial w_1} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$
$$= h'(z_3) \times w_3 \times g'(z_2) \times w_2 \times f'(z_1) \times x;$$

$$\frac{\partial l}{\partial b} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1} \times \frac{\partial z_1}{\partial b}$$
$$= h'(z_3) \times w_3 \times g'(z_2) \times w_2 \times f'(z_1) \times 1$$

复合函数求导

$$l = h(w_3 \times g(w_2 \times f(w_1 \times x + b)))$$

$$\frac{\partial l}{\partial w_3} = \boxed{\frac{\partial l}{\partial z_3}} \times \frac{\partial z_3}{\partial w_3} = h'(z_3) \times a_2,$$

$$\frac{\partial l}{\partial w_2} = \boxed{\boxed{\frac{\partial l}{\partial z_3}} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2}} \times \frac{\partial z_2}{\partial w_2} = h'(z_3) \times w_3 \times g'(z_2) \times a_1;$$

$$\frac{\partial l}{\partial w_1} = \boxed{\boxed{\frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2}} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1}} \times \frac{\partial z_1}{\partial w_1} = h'(z_3) \times w_3 \times g'(z_2) \times w_2 \times f'(z_1) \times x;$$

$$\frac{\partial l}{\partial b} = \boxed{\frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1}} \times \frac{\partial z_1}{\partial b} = h'(z_3) \times w_3 \times g'(z_2) \times w_2 \times f'(z_1) \times 1$$

复合函数求导

$$\frac{\partial l}{\partial w_3} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial w_3},$$

$$\frac{\partial l}{\partial w_2} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial w_2};$$

$$\frac{\partial l}{\partial w_1} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1};$$

$$\frac{\partial l}{\partial b} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1} \times \frac{\partial z_1}{\partial b}$$

$$\delta_3 \equiv \frac{\partial l}{\partial z_3}$$

$$\delta_2 \equiv \frac{\partial l}{\partial z_2} = \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2}$$

$$= \delta_3 \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2}$$

$$\delta_1 \equiv \frac{\partial l}{\partial z_1}$$

$$= \frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1}$$

$$= \delta_2 \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1}$$

复合函数求导

$$\frac{\partial l}{\partial w_3} = \boxed{\frac{\partial l}{\partial z_3}} \times \frac{\partial z_3}{\partial w_3},$$

$$\frac{\partial l}{\partial w_2} = \boxed{\frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2}} \times \frac{\partial z_2}{\partial w_2};$$

$$\frac{\partial l}{\partial w_1} = \boxed{\frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1}} \times \frac{\partial z_1}{\partial w_1};$$

$$\frac{\partial l}{\partial b} = \boxed{\frac{\partial l}{\partial z_3} \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1}} \times \frac{\partial z_1}{\partial b}$$

$$\delta_3 \equiv \frac{\partial l}{\partial z_3}$$

$$\delta_2 \equiv \frac{\partial l}{\partial z_2} = \delta_3 \times \frac{\partial z_3}{\partial a_2} \times \frac{\partial a_2}{\partial z_2}$$

$$\delta_1 \equiv \frac{\partial l}{\partial z_1} = \delta_2 \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1}$$

$$\frac{\partial l}{\partial w_3} = \delta_3 \times \frac{\partial z_3}{\partial w_3} = \delta_3 \times a_2,$$

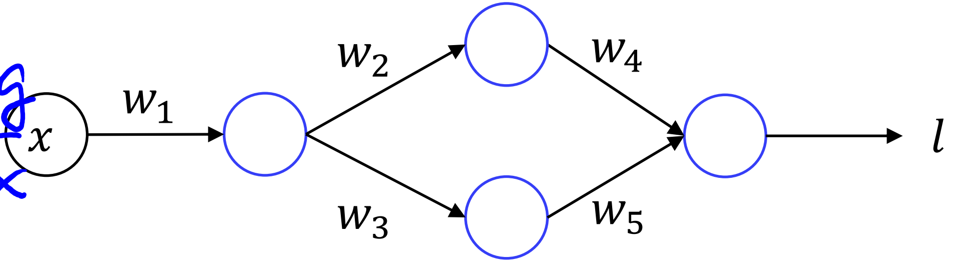
$$\frac{\partial l}{\partial w_2} = \delta_2 \times \frac{\partial z_2}{\partial w_2} = \delta_2 \times a_1;$$

$$\frac{\partial l}{\partial w_1} = \delta_1 \times \frac{\partial z_1}{\partial w_1} = \delta_1 \times x;$$

$$\frac{\partial l}{\partial b} = \delta_1 \times \frac{\partial z_1}{\partial b} = \delta_1$$

复合函数求导

$$\frac{\partial(f+g)}{\partial x} = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial x}$$



$$l = f(w_4 f(w_2 f(w_1 x + b_1) + b_2) + w_5 f(w_3 f(w_1 x + b_1) + b_3))$$

$l = f(z_4);$

$z_4 = w_4 a_2 + w_5 a_3;$

$a_2 = f(z_2); z_2 = w_2 a_1 + b_2$

$a_3 = f(z_3); z_3 = w_3 a_1 + b_3;$

$a_1 = f(z_1); z_1 = w_1 x + b_1;$

$$\frac{\partial l}{\partial w_1} = ? \quad \frac{\partial l}{\partial b_1} = ?$$

复合函数求导

$$l = f(w_4 f(w_2 f(w_1 x + b_1) + b_2) + w_5 f(w_3 f(w_1 x + b_1) + b_3) + b)$$

$$l = f(z_4);$$
$$z_4 = w_4 a_2 + w_5 a_3 + b;$$
$$a_2 = f(z_2); z_2 = w_2 a_1 + b_2$$
$$a_3 = f(z_3); z_3 = w_3 a_1 + b_3;$$
$$a_1 = f(z_1); z_1 = w_1 x + b_1;$$

$$\frac{\partial l}{\partial w_1} = \frac{\partial l}{\partial z_4}\frac{\partial z_4}{\partial w_1} = \overbrace{\frac{\partial l}{\partial z_4}}^{\delta_4}\frac{\partial z_4}{\partial a_2}\frac{\partial a_2}{\partial w_1} + \overbrace{\frac{\partial l}{\partial z_4}}^{\delta_4}\frac{\partial z_4}{\partial a_3}\frac{\partial a_3}{\partial w_1}$$

$$= \overbrace{\boxed{\frac{\partial l}{\partial z_4}\frac{\partial z_4}{\partial a_2}\frac{\partial a_2}{\partial z_2}}}^{\delta_2}\frac{\partial z_2}{\partial w_1} + \underbrace{\boxed{\frac{\partial l}{\partial z_4}\frac{\partial z_4}{\partial a_3}\frac{\partial a_3}{\partial z_3}}}_{\delta_3}\frac{\partial z_3}{\partial w_1}$$

$$= \frac{\partial l}{\partial z_2}\frac{\partial z_2}{\partial w_1} + \frac{\partial l}{\partial z_3}\frac{\partial z_3}{\partial w_1} = \delta_2 \frac{\partial z_2}{\partial w_1} + \delta_3 \frac{\partial z_3}{\partial w_1}$$

$$\delta_4 \equiv \frac{\partial l}{\partial z_4}, \delta_2 \equiv \frac{\partial l}{\partial z_2} = \frac{\partial l}{\partial z_4}\frac{\partial z_4}{\partial a_2}\frac{\partial a_2}{\partial z_2} = \delta_4 w_4 f'(z_2), \delta_3 \equiv \frac{\partial l}{\partial z_3} = \frac{\partial l}{\partial z_4}\frac{\partial z_4}{\partial a_3}\frac{\partial a_3}{\partial z_3} = \delta_4 w_5 f'(z_3)$$

复合函数求导

$$l = f(w_4 f(w_2 f(w_1 x + b_1) + b_2) + w_5 f(w_3 f(w_1 x + b_1) + b_3) + b)$$

$$\frac{\partial l}{\partial w_1} = \delta_2 \frac{\partial z_2}{\partial w_1} + \delta_3 \frac{\partial z_3}{\partial w_1}$$

$$l = f(z_4);$$
$$z_4 = w_4 a_2 + w_5 a_3 + b;$$
$$a_2 = f(z_2); z_2 = w_2 a_1 + b_2$$
$$a_3 = f(z_3); z_3 = w_3 a_1 + b_3;$$
$$a_1 = f(z_1); z_1 = w_1 x + b_1;$$

$$\frac{\partial z_2}{\partial w_1} = \frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1}; \quad \frac{\partial z_3}{\partial w_1} = \frac{\partial z_3}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial l}{\partial w_1} = \delta_2 \frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1} + \delta_3 \frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1}$$

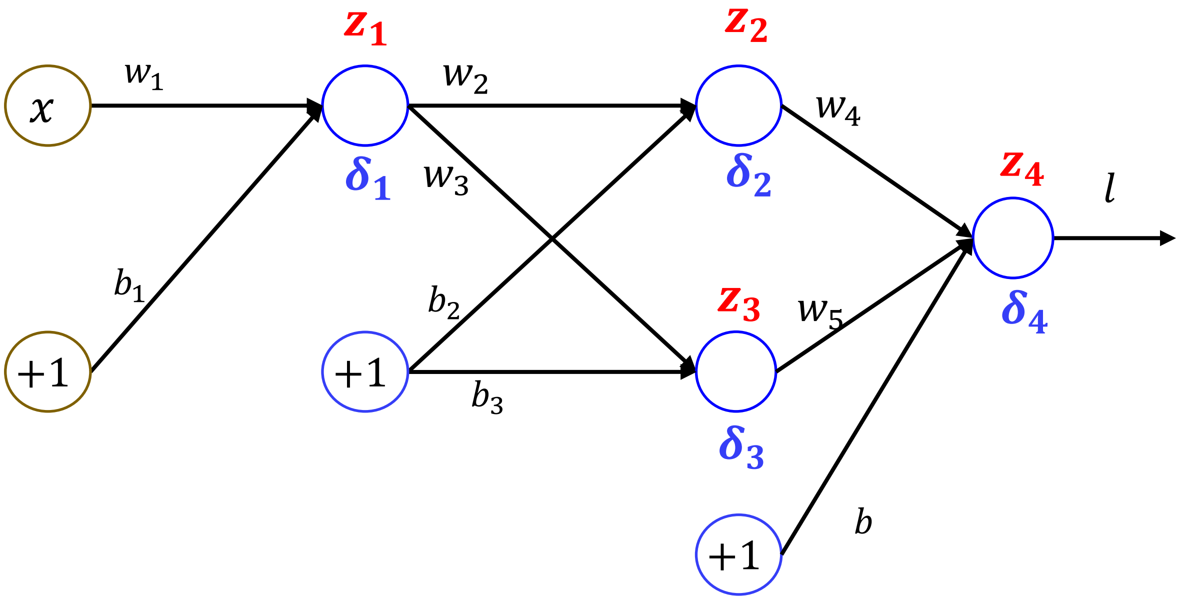
$$= \left(\delta_2 \frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1} + \delta_3 \frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\right)\frac{\partial z_1}{\partial w_1} = \delta_1 \frac{\partial z_1}{\partial w_1} = \delta_1 x$$

$$\rightarrow \frac{\partial l}{\partial z_1} = \delta_1$$

$$\delta_1 \equiv \delta_2 \frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1} + \delta_3 \frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1} = \left(\delta_2 \frac{\partial z_2}{\partial a_1} + \delta_3 \frac{\partial z_2}{\partial a_1}\right)\frac{\partial a_1}{\partial z_1} = \boxed{(\delta_2 w_2 + \delta_3 w_3) f'(z_1)}$$

复合函数求导

$$l = f(w_4 f(w_2 f(w_1 x + b_1) + b_2) + w_5 f(w_3 f(w_1 x + b_1) + b_3) + b)$$



$$\delta_4 \equiv \frac{\partial l}{\partial z_4}$$

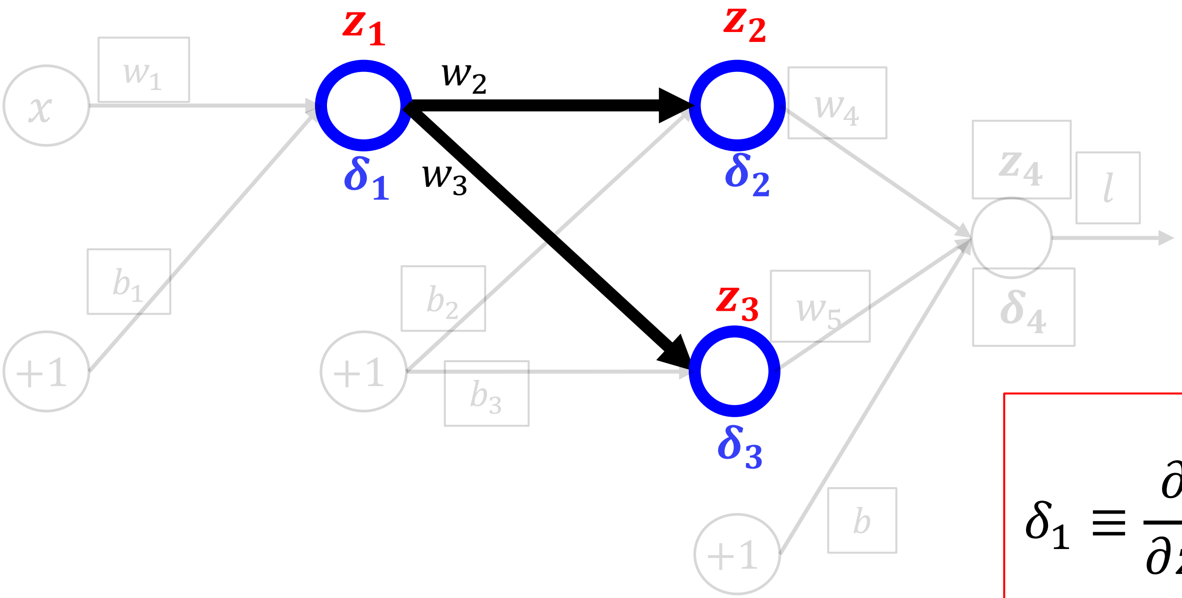
$$\delta_2 \equiv \frac{\partial l}{\partial z_2} = \delta_4 w_4 f'(z_2)$$

$$\delta_3 \equiv \frac{\partial l}{\partial z_3} = \delta_4 w_5 f'(z_3)$$

$$\delta_1 \equiv \frac{\partial l}{\partial z_1} = (\delta_2 w_2 + \delta_3 w_3) f'(z_1)$$

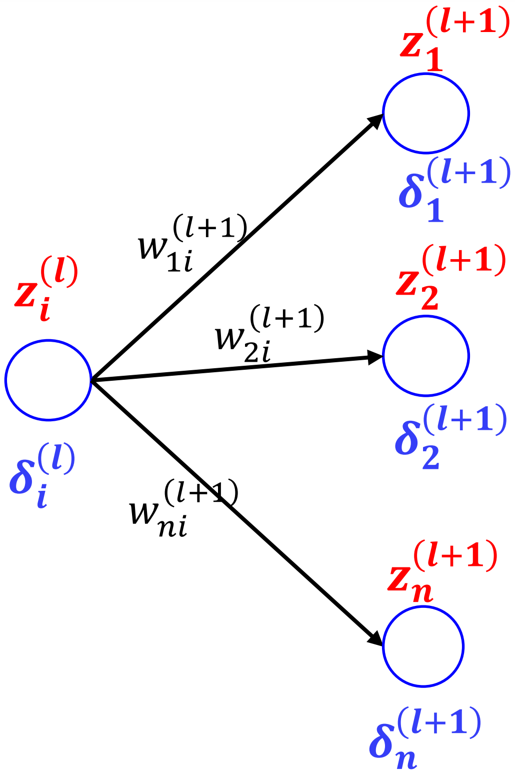
复合函数求导

$$l = f(w_4 f(w_2 f(w_1 x + b_1) + b_2) + w_5 f(w_3 f(w_1 x + b_1) + b_3) + b)$$



$$\delta_1 \equiv \frac{\partial l}{\partial z_1} = (\delta_2 w_2 + \delta_3 w_3) f'(z_1)$$

响应前向传播、误差后向传播(Error Back Propagation)

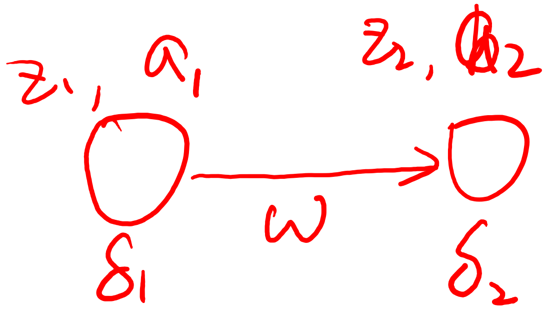


$$z_j^{(l+1)} = \cdots + w_{ji}^{(l+1)} f\left(z_i^{(l)}\right) + \cdots$$

$$a_j^{(l+1)} = f\left(z_j^{(l+1)}\right)$$

$$\delta_i^{(l)} = \left[\sum_{j=1}^{n} w_{ji}^{(l+1)} \delta_j^{(l+1)}\right] f'\left(z_i^{(l)}\right)$$

$$\Delta W_{ji}^{(l)} = \delta_j^{(l)} a_i^{(l)}$$



$$\underline{\Delta W = a_1 \delta_2} \leftarrow \text{update}$$

$$\delta_1 = \delta_2 \cdot w \cdot f'(z_1)$$

B.P.

$$a_2 = f(w a_1)$$

F.P.

误差后向传播

$$l(\Theta; X, Y) = l(\hat{Y}, Y)$$

$$\hat{Y} = h(X; \Theta) = f_k\left(f_{k-1}\left(\cdots f_1\left(X; W^{(1)}, b^{(1)}\right) \cdots; W^{(k-1)}, b^{(k-1)}\right); W^{(k)}, b^{(k)}\right)$$

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}; a^{(l)} = f_l\left(z^{(l)}\right)$$

$$\Delta W^{(l)} = \frac{\partial l}{\partial W^{(l)}} = \frac{\partial l}{\partial z^{(l)}} {a^{(l-1)}}^T = \delta^{(l)} {a^{(l-1)}}^T$$

$$\Delta b^{(l)} = \frac{\partial l}{\partial b^{(l)}} = \frac{\partial l}{\partial z^{(l)}} = \delta^{(l)}$$

$$\delta^{(l)} \equiv \frac{\partial l}{\partial z^{(l)}} \equiv \begin{pmatrix} \delta_1^{(l)} \\ \delta_2^{(l)} \\ \vdots \\ \delta_{N_l}^{(l)} \end{pmatrix}$$

$$\delta_i^{(l)} = \left[\sum_{j=1}^{N_{l+1}} w_{ji}^{(l+1)} \delta_j^{(l+1)}\right] f'\left(z_i^{(l)}\right) = \begin{bmatrix} w_{1i}^{(l+1)} \\ w_{2i}^{(l+1)} \\ \vdots \\ w_{N_{l+1}i}^{(l+1)} \end{bmatrix}^T \begin{bmatrix} \delta_1^{(l+1)} \\ \delta_2^{(l+1)} \\ \vdots \\ \delta_{N_{l+1}}^{(l+1)} \end{bmatrix} f'\left(z_i^{(l)}\right) = \begin{bmatrix} w_{1i}^{(l+1)} \\ w_{2i}^{(l+1)} \\ \vdots \\ w_{N_{l+1}i}^{(l+1)} \end{bmatrix}^T \delta^{(l+1)} f'\left(z_i^{(l)}\right)$$

误差后向传播

$$\delta^{(l)} \equiv \begin{pmatrix} \delta_1^{(l)} \\ \delta_2^{(l)} \\ \vdots \\ \delta_{N_l}^{(l)} \end{pmatrix}$$

$$\delta_i^{(l)} = \begin{bmatrix} w_{1i}^{(l+1)} \\ w_{2i}^{(l+1)} \\ \vdots \\ w_{N_{l+1}i}^{(l+1)} \end{bmatrix}^T \delta^{(l+1)} f'\left(z_i^{(l)}\right)$$

$$A \odot B = \begin{bmatrix} A_1 B_1 \\ A_2 B_2 \\ \vdots \\ A_n B_n \end{bmatrix}$$

$$\delta^{(l)} \equiv \frac{\partial l}{\partial z^{(l)}} \equiv \begin{pmatrix} \delta_1^{(l)} \\ \delta_2^{(l)} \\ \vdots \\ \delta_{N_l}^{(l)} \end{pmatrix} = \left\{ \begin{bmatrix} w_{1,1}^{(l+1)} & w_{2,1}^{(l+1)} & \cdots & w_{N_{l+1},1}^{(l+1)} \\ w_{1,2}^{(l+1)} & w_{2,2}^{(l+1)} & & w_{N_{l+1},2}^{(l+1)} \\ \vdots & \vdots & & \vdots \\ w_{1,N_l}^{(l+1)} & w_{2,N_l}^{(l+1)} & \cdots & w_{N_{l+1},N_l}^{(l+1)} \end{bmatrix} \delta^{(l+1)} \right\} \odot f'\left(z^{(l)}\right)$$

$$= W^{(l+1)^T} \delta^{(l+1)} \odot f'\left(z^{(l)}\right)$$

$$f'\left(z^{(l)}\right) = \left[f'\left(z_1^{(l)}\right) \quad f'\left(z_2^{(l)}\right) \quad \ldots \quad f'\left(z_{N_l}^{(l)}\right) \right]^T$$

使用BP算法计算梯度

$$l(\Theta;X,Y)=l\left(\hat{Y},Y\right)$$

$$\hat{Y}=h(X;\Theta)=f\left(f\left(\cdots f\left(X;W^{(1)},b^{(1)}\right)\cdots;W^{(k-1)},b^{(k-1)}\right);W^{(k)},b^{(k)}\right)$$

- *Forward Propagation(FP)*

$$a^{(0)}=X;z^{(l)}=W^{(l)}a^{(l-1)}+b^{(l)};a^{(l)}=f\left(z^{(l)}\right);\hat{Y}=a^{(L)}$$

$$l(\Theta;X,Y)=l\left(\hat{Y},Y\right)$$

- *error Backward Propagation(BP)*

$$\delta^{(L)}=\frac{\partial l}{\partial z^{(L)}};\delta^{(l)}={W^{(l+1)}}^{T}\delta^{(l+1)}\odot f'\left(z^{(l)}\right)$$

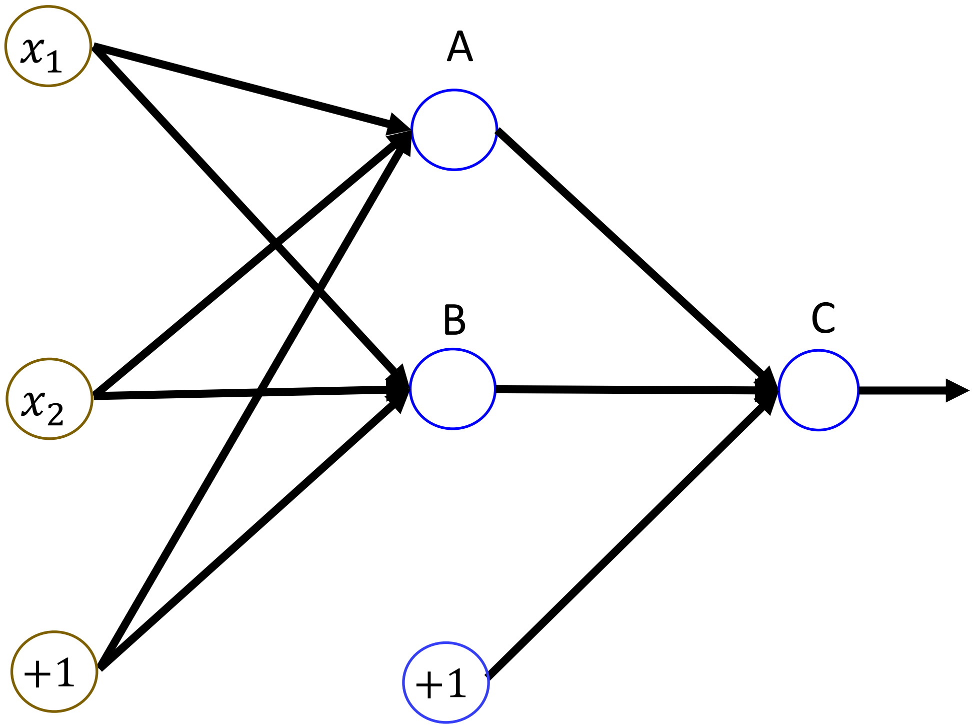
$$\Delta W^{(l)}=\delta^{(l+1)}{a^{(l)}}^{T};\Delta b^{(l)}=\delta^{(l+1)}$$

使用梯度下降法训练神经网络

- 输入：训练样本$D=\{(X^{(i)}\in R^d,Y^{(i)})\}_{i=1}^n$,学习速率$\alpha$，收敛条件$\epsilon$
- 输出：$\Theta=\{W^{(l)},b^{(l)}\}_{l=1}^L$
- 1. 初始化$\Theta, l_0=Inf$
- 2. While True:
  - 2.1 $FP{:}\, loss=0;\ for\ i=1{:}n\ \ \{a^{(0)}=X^{(i)}\ \ for\ l=1{:}L\ \{z^{(l)}=W^{(l)}a^{(l-1)}+b^{(l)};a^{(l)}=f(z^{(l)})\}\ \ l_i=loss(a^{(L)},Y^{(i)});loss\mathrel{+}=l_i;\}$
  - 2.2 $l=\frac{l}{n};if\ |l-l_0|<\epsilon,break;else\ l_0=l$
  - 2.3 $BP{:}\,\delta^{(L)}=\frac{\partial l}{\partial z^{(L)}};for\ \ l=L-1{:}1\ \{\Delta W^{(l)}=\delta^{(l)}{a^{(l)}}^T,\Delta b=\delta^{(l)};\delta^{(l)}={W^{(l+1)}}^T\delta^{(l+1)}\odot f'(z^{(l)})\}$
  - 2.4 $GD{:}\,for\ \ l=L{:}1\,\{\,W^{(l)}=W^{(l)}-\alpha\Delta W^{(l)},b^{(l)}=b^{(l)}-\alpha\Delta b^{(l)}\};$
- 3. 返回Θ

批梯度下降(Batch Gradient Descent)

权值初始化



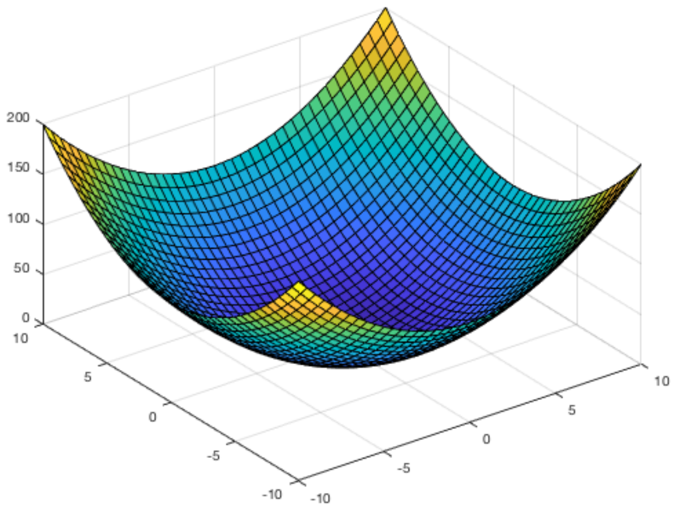
$$w_{ij}^{(l)} \sim N(0, \sigma)$$

梯度下降法(Gradient Descend)

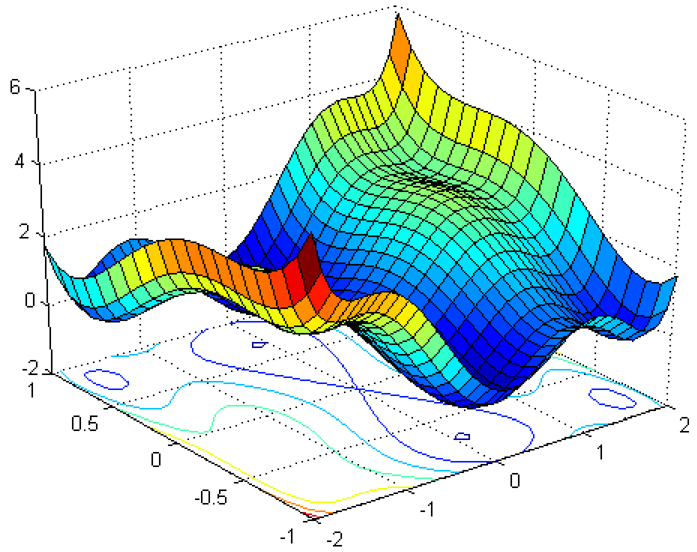
- Batch GD:
  - $l(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} l\left(\boldsymbol{\theta}; \boldsymbol{x}^{(i)}, y^{(i)}\right)$
  - $\boxed{\theta^{(t+1)} = \theta^{(t)} - \eta \nabla l\left(\theta^{(t)}\right) = \theta^{(t)} - \eta \frac{1}{n}\sum_{i=1}^{n} \nabla l\left(\theta^{(t)}; \boldsymbol{x}^{(i)}, y^{(i)}\right)}$
- Stochastic GD:
  - $\boxed{\theta^{(t+1)} = \theta^{(t)} - \eta \nabla l\left(\theta^{(t)}; \boldsymbol{x}^{(i)}, y^{(i)}\right)}$
- Mini-Batch GD:
  - $\boxed{\theta^{(t+1)} = \theta^{(t)} - \eta \nabla l\left(\theta^{(t)}; B^{(k)}\right)}$
  - $B^{(k)} = \left\{\boldsymbol{x}^{(i)}, y^{(i)}\right\}_{i=(k-1)*batch\_size+1}^{k*batch\_size}$
- Epoch: 遍历整个样本集合$\mathcal{D}$，每一个Epoch随机排列$\mathcal{D}$中元素

1次迭代(iteration)：
前向+后向传播
权值更新

损失函数$l(\boldsymbol{\theta})$的性质

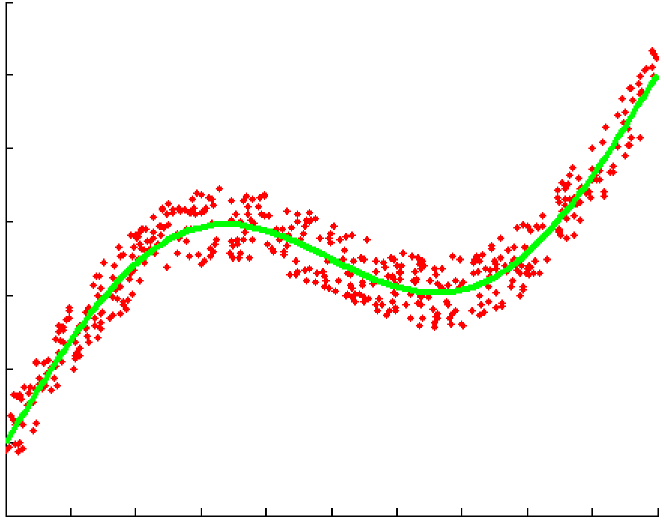


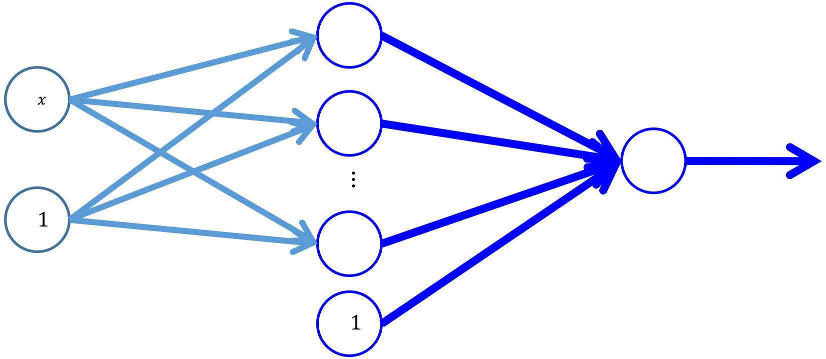
1个神经元，Logistic Regression



1个或多个隐层，非线性响应函数，MLP

实例： 曲线拟合

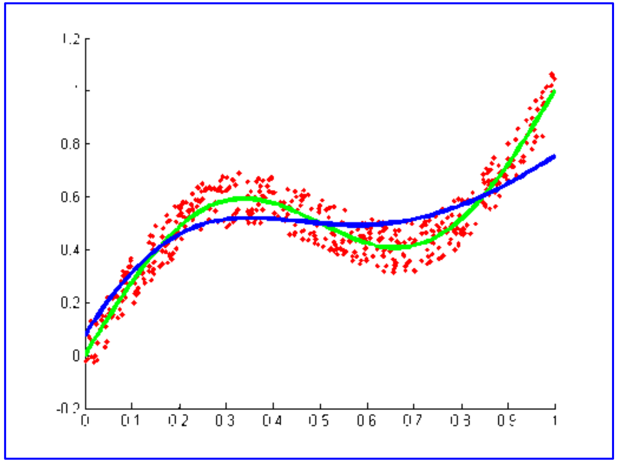


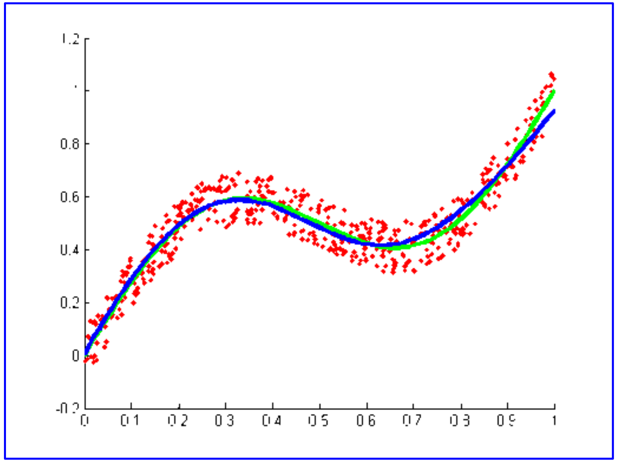


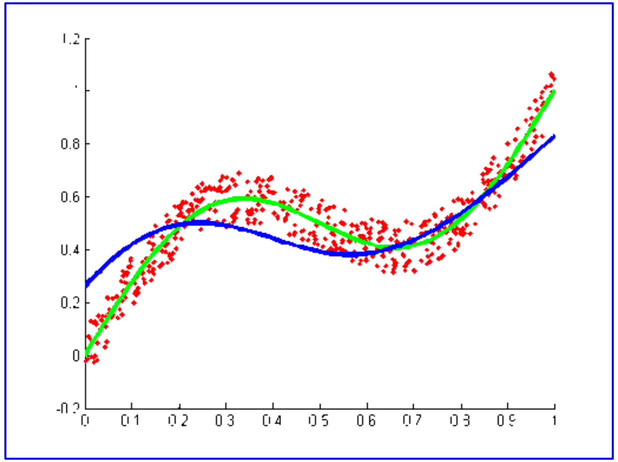
$$y = x + 0.3\sin(2\pi x) + \varepsilon \quad x \in [0 \quad 1] \quad \varepsilon \in [-0.1 \quad 0.1]$$

$$l(h; \boldsymbol{x}, y) = \frac{1}{2}(h(\boldsymbol{x}) - y)^2$$

实例：曲线拟合

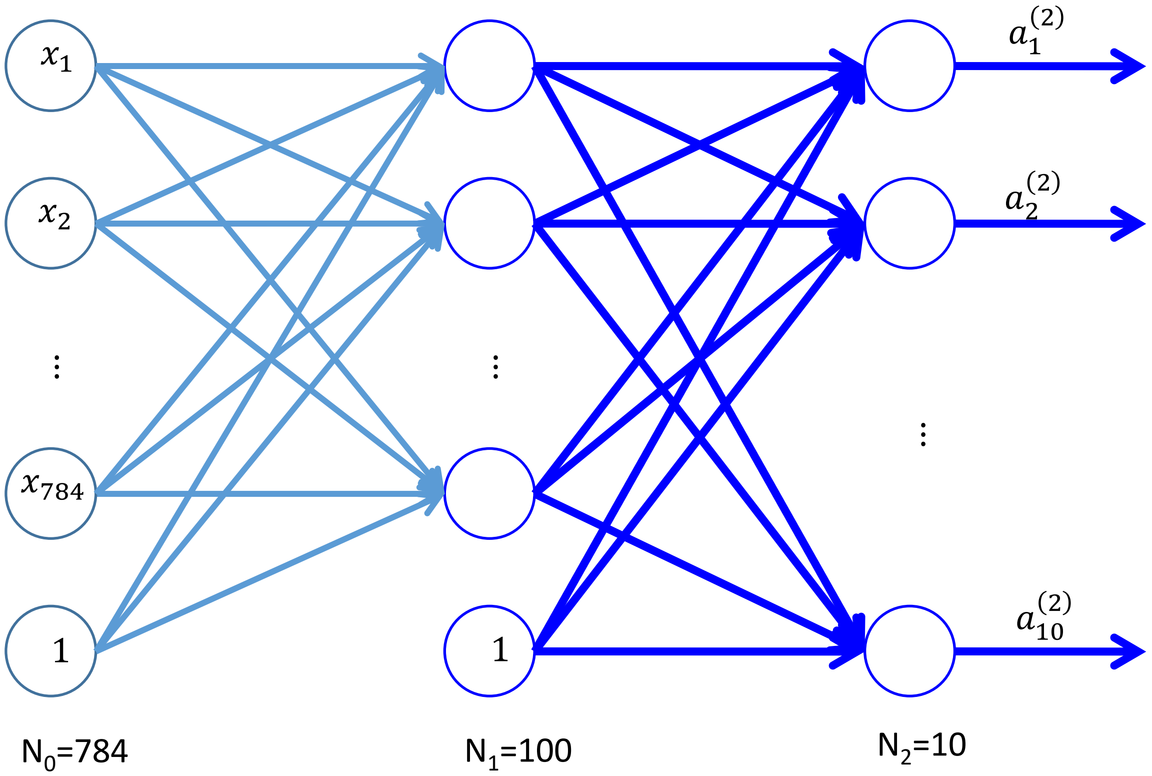






示例：MNIST手写体数字识别

实例:MNIST手写数字识别



实例:MNIST手写数字识别

$$P(y=k|x)=a_k^{(2)}(x)=\frac{\exp\left(z_k^{(2)}(x)\right)}{\sum_{i=1}^{10}\exp\left(z_i^{(2)}(x)\right)}$$

Softmax

$$y=\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{10} \end{pmatrix}; y_k=\begin{cases} 1 & x\in k \\ 0 & x\notin k \end{cases}$$

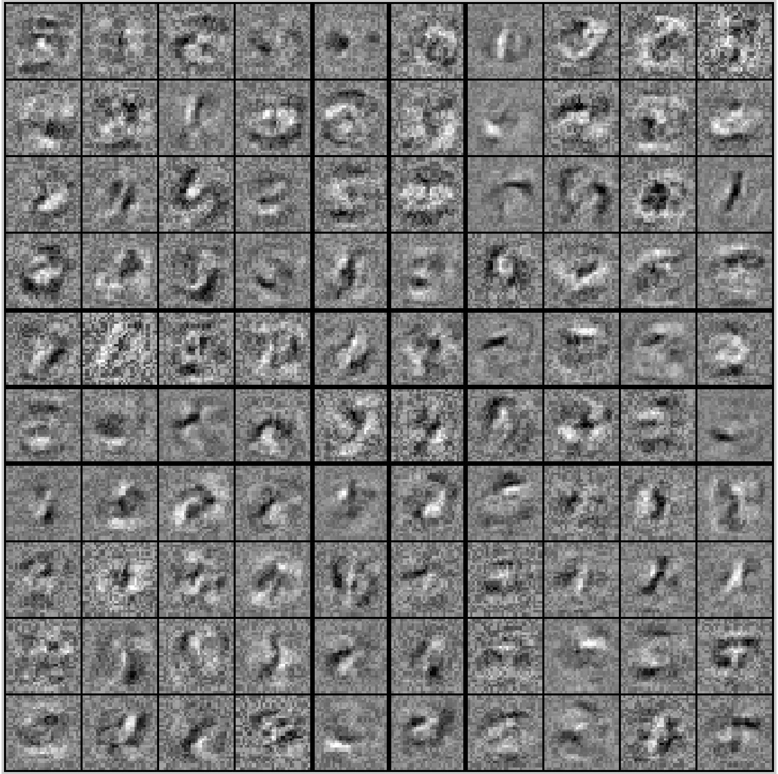
One-hot vector

$$P(y|x)=\prod_{j=1}^{10}[P(y=j|x)]^{y=j}$$

$$l(\theta\,;x\,,y)=-\log P\,(y|x)=-\sum_{j=1}^{10}\left\{y_j\log\left[a_j^{(2)}(x)\right]\right\}$$

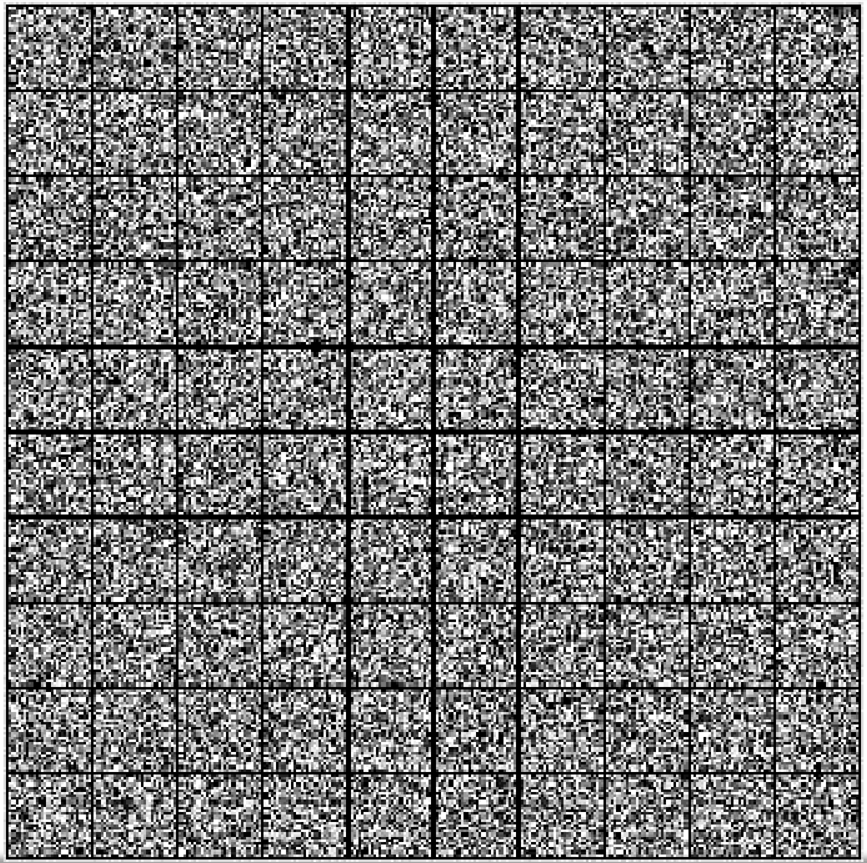
$$l_{reg}(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{10}\left\{y_j^{(i)}\log\left[a_j^{(2)}\left(x^{(i)}\right)\right]\right\}+\frac{\lambda}{2m}\left\{\sum_{i=1}^{784}\sum_{j=1}^{100}\left(w_{j,i}^{(1)}\right)^2+\sum_{i=1}^{100}\sum_{j=1}^{10}\left(w_{j,i}^{(2)}\right)^2\right\}$$

实例:MNIST手写数字识别



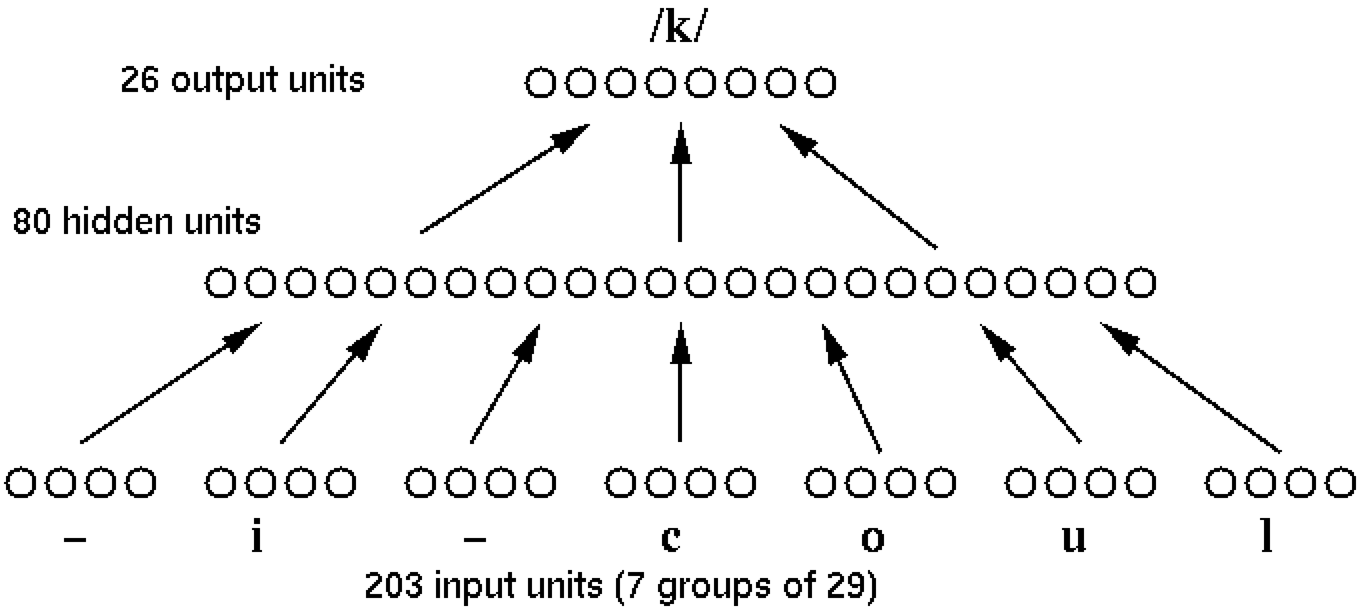
准确率=97.3%

实例:MNIST手写数字识别



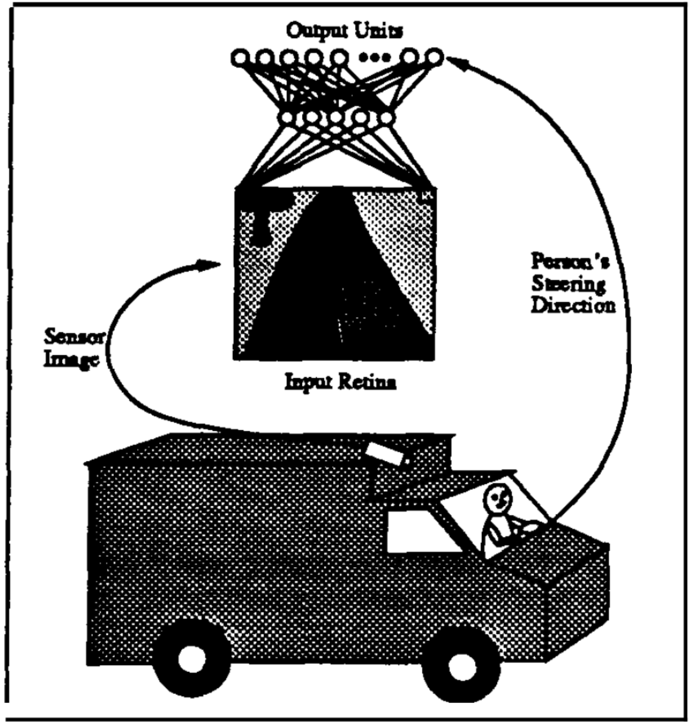
应用实例

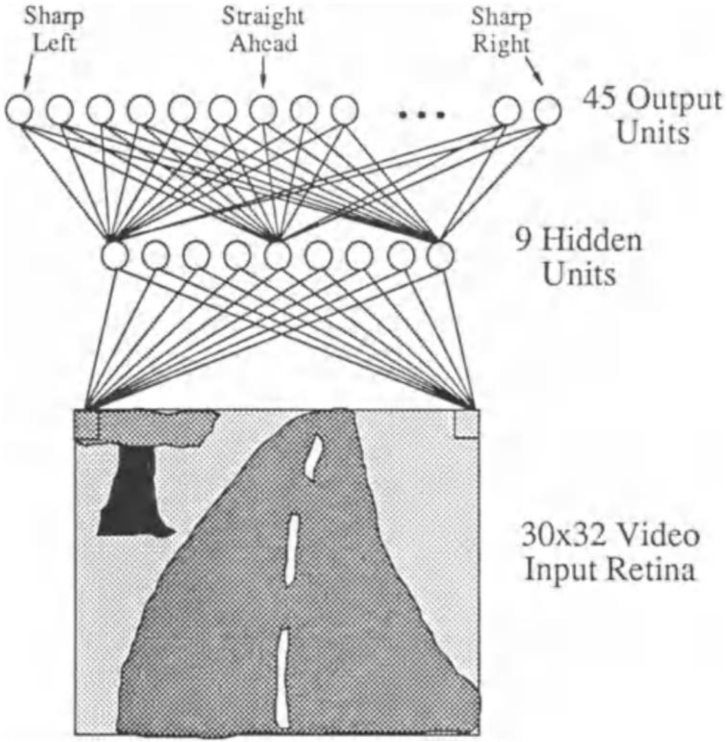
NETalk,1987



Sejnowski, T.J., and Rosenberg, C.R. (1987). "Parallel networks that learn to pronounce English text" in Complex Systems, 1, 145-168

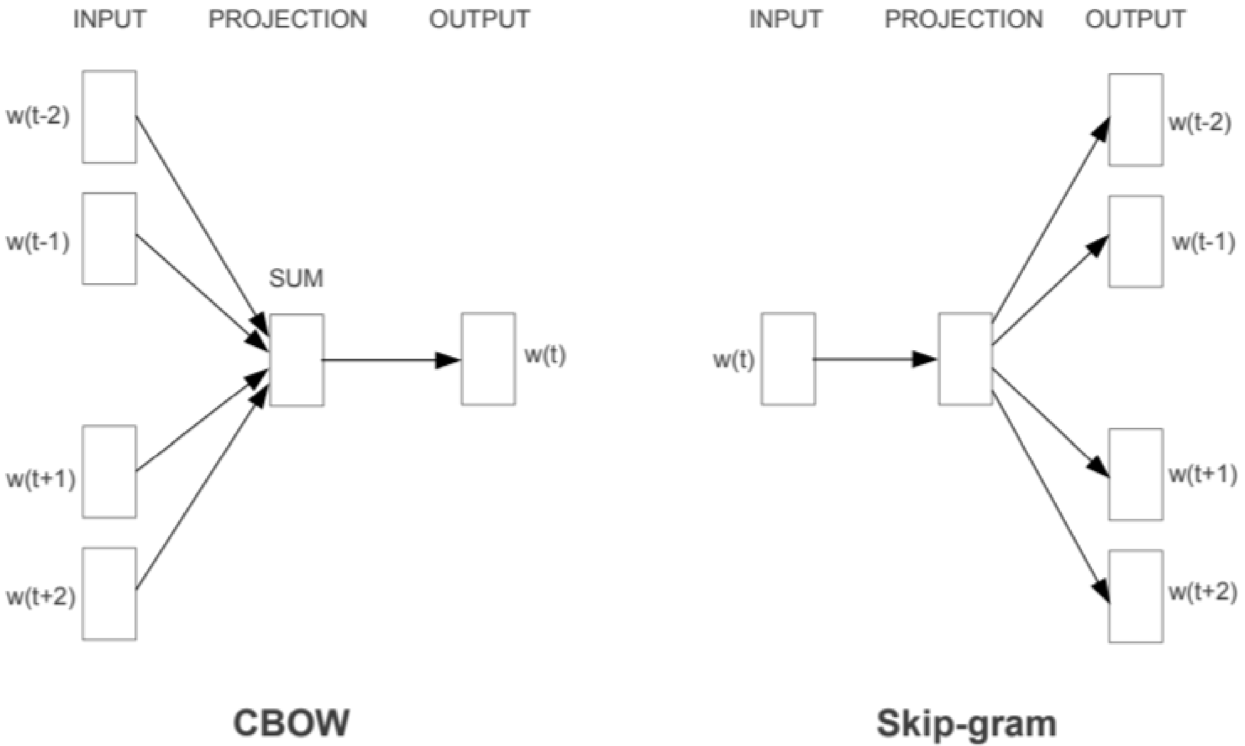
ALVINN,CMU1988





Dean Pomerleau : Autonomous Land Vehicle in a Neural Network, NIPS,1988

词嵌入(Word Embedding)



Skip-gram

Source Text

Training Samples

The quick brown fox jumps over the lazy dog. → (the, quick) (the, brown)

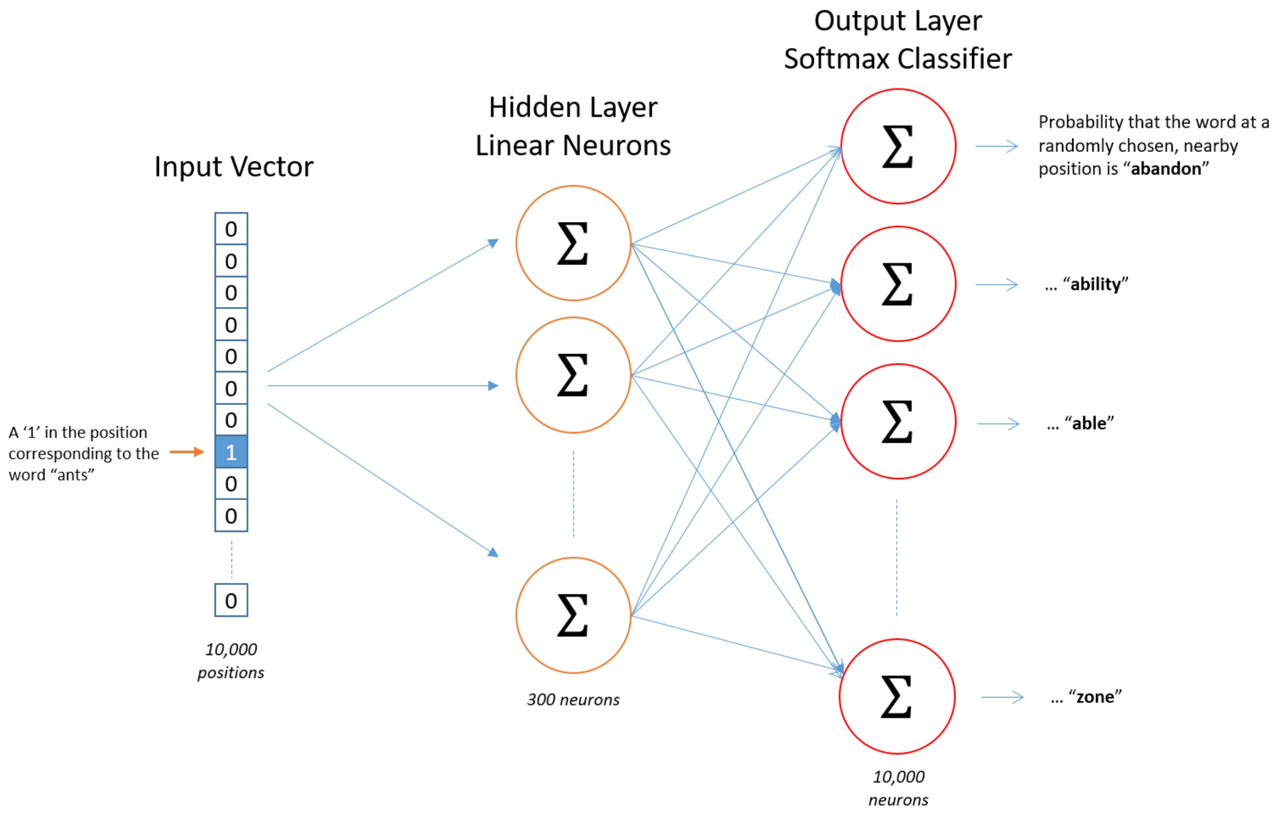
The **quick** brown fox jumps over the lazy dog. → (quick, the) (quick, brown) (quick, fox)

The quick **brown** fox jumps over the lazy dog. → (brown, the) (brown, quick) (brown, fox) (brown, jumps)

The quick brown **fox** jumps over the lazy dog. → (fox, quick) (fox, brown) (fox, jumps) (fox, over)

http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Skip-gram


Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"
A '1' in the position corresponding to the word "ants"
0
0
0
0
0
0
0
1
0
0
0
10,000 positions
300 neurons
10,000 neurons

Skip-gram

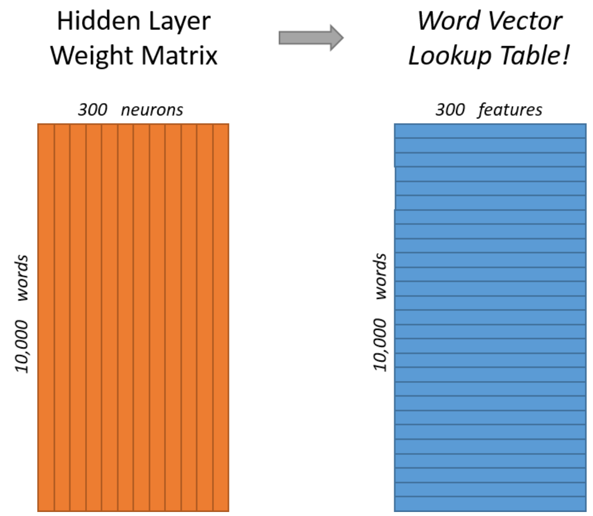
$|V| = 10{,}000 \qquad x \equiv ont\_hot('the') = [0,0,\dots,0,1,0,\dots 0] \in \{0,1\}^{10{,}000}$

$index('the') = 5793\colon x_{5793} = 1$

$E \in R^{10{,}000\times 300}\colon E = [e_1, e_2, \dots, e_{10{,}000}]$

$z = xE = e_{5793} \Leftarrow Embedding\ of\ word\ 'the'$

$$[0 \quad 0 \quad 0 \quad \boxed{1} \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ \boxed{10 \quad 12 \quad 19} \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$



Skip-gram

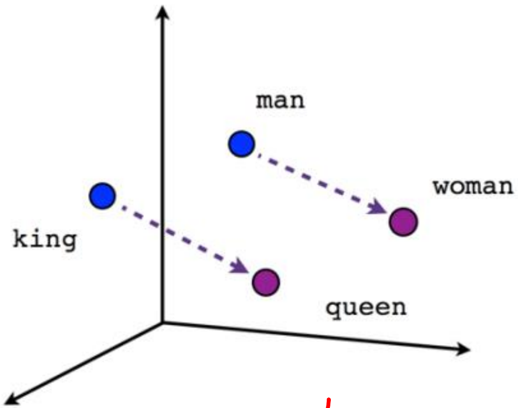
$|V| = 10{,}000$ $\qquad x \equiv ont\_hot('the') = [0,0,\ldots,0,1,0,\ldots 0] \in \{0,1\}^{10{,}000}$

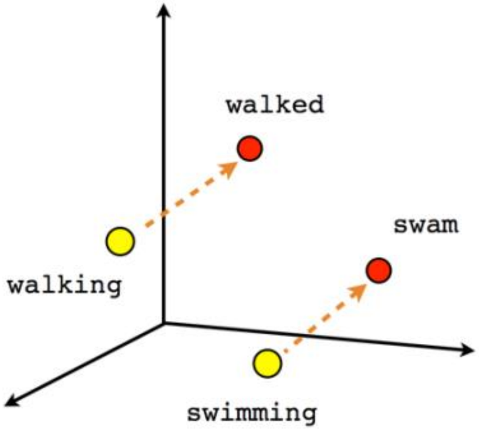
$index('the') = 5793: x_{5793} = 1$

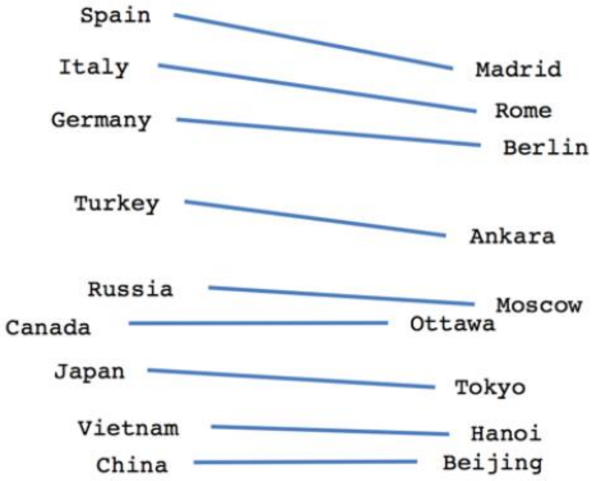
$E \in R^{10{,}000\times 300}: E = [e_1, e_2, \ldots, e_{10{,}000}]$
$z = xE = e_{5793} \Leftarrow Embedding\ of\ word\ 'the'$

输出层： $y = Wz + b, W \in R^{10{,}000\times 300}, b \in R^{10{,}000}$
$\rho = softmax(y) \equiv [\rho_1, \rho_2, \ldots, \rho_{10{,}000}], \rho_i = P(word_i\text{是}the\text{的近邻})$

词向量







Queen - King + Man

Male-Female

= Woman

Verb tense

Country-Capital

神经语言模型(Neural Language Model)

"The quick brown fox jumps over the lazy dog."

句子=单词序列： $S = (w_1, w_2, \ldots, w_T)$

语言模型：句子的概率分布模型 $P(S)$ =?

$$P(S) = P(w_1, w_2, \ldots, w_T)$$
$$= P(w_1)P(w_2|w_1)P(w_3|w_2, w_1) \cdots P(w_T|w_{T-1}, \ldots, w_1)$$

神经语言模型(Neural Language Model)

$$P(S) = P(w_1, w_2, \dots, w_T)$$
$$= P(w_1)P(w_2|w_1)P(w_3|w_2, w_1) \cdots P(w_T|w_{T-1}, \dots, w_1)$$

$$n - gram: P(w_t|w_{t-1}, w_{t-2}, \dots, w_1) = \prod_{i=1}^{T} P(w_t|w_{t-1}, w_{t-2}, \dots w_{t-n+1})$$

$$n = 1: unigram: P(w_1, w_2, \dots, w_T) = \prod_{i=1}^{T} P(w_i)$$
$$n = 2: bigram: P(w_1, w_2, \dots, w_T) = \prod_{i=1}^{T} P(w_i|w_{i-1})$$
$$n = 1: trigram: P(w_1, w_2, \dots, w_T) = \prod_{i=1}^{T} P(w_i|w_{i-1}, w_{i-2})$$

神经语言模型(Neural Language Model)

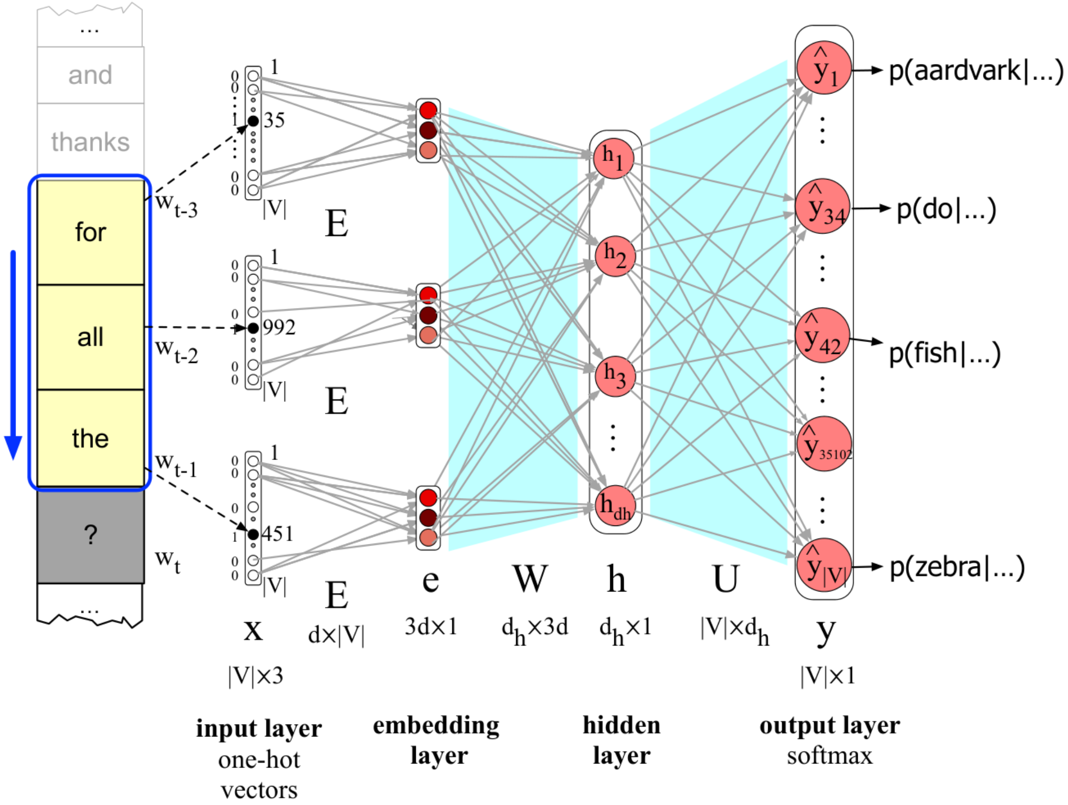
$$n-gram: P(w_t|w_{t-1},w_{t-2},\ldots,w_1)=\prod_{i=1}^{T}P(w_t|w_{t-1},w_{t-2},\ldots w_{t-n+1})$$

如何表示$P(w_t|w_{t-1},w_{t-2},\ldots w_{t-n+1})$？

输入$w_{t-1},w_{t-2},\ldots w_{t-n+1}$，输出$P(w_t|w_{t-1},w_{t-2},\ldots w_{t-n+1})$

神经语言模型：用一个神经网络表示$P(w_t|w_{t-1},w_{t-2},\ldots w_{t-n+1})$；输入层为n-1个单词$w_{t-1},w_{t-2},\ldots w_{t-n+1}$，输出层为下一个单词的概率分布$P(w_t|w_{t-1},w_{t-2},\ldots w_{t-n+1})$

神经语言模型(Bengio,2003)



Bengio, Y. et al: A neural probabilistic language model.JMLR,3:1137–1155.2003

神经语言模型(Bengio,2003)

$|V| = 10{,}000 \qquad P(w|'for','all','the')$

$$x_1 \equiv ont\_hot('for') = [0,0,\ldots 1,\ldots,0,\ldots 0] \in \{0,1\}^{10,000}$$
$$x_2 \equiv ont\_hot('all') = [0,0,\ldots,0,1,0,\ldots 0] \in \{0,1\}^{10,000}$$
$$x_3 \equiv ont\_hot('the') = [0,0,\ldots,0,0,0,\ldots 1,\ldots 0] \in \{0,1\}^{10,000}$$

$$E \in R^{10,000\times 300}: E = \left[e_1, e_2, \ldots, e_{10,000}\right]^T, e_i \in R^{300}$$
$$v_1 = x_1 E = e_{3079} \Leftarrow Embedding\ of\ word\ 'for'$$
$$v_2 = x_2 E = e_{1908} \Leftarrow Embedding\ of\ word\ 'all'$$
$$v_3 = x_3 E = e_{5793} \Leftarrow Embedding\ of\ word\ 'the'$$
$$v = (v_1, v_2, v_3) \in R^{900}$$

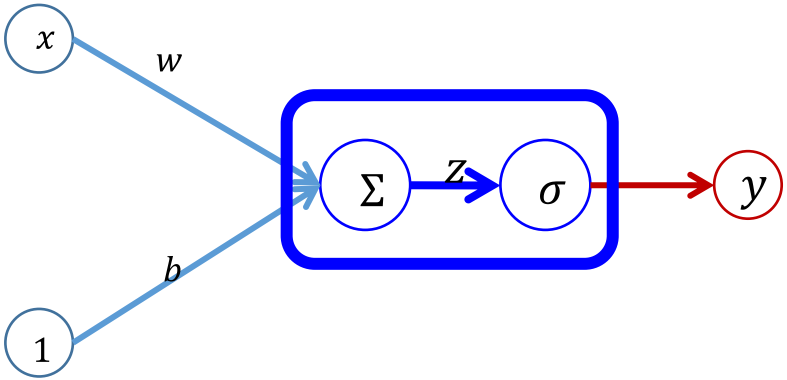
$$z_h = W_h v + b_h \in R^{1,000}, a_h = \tanh(z_h)$$
$$W_h \in R^{1,000\times 900}, b_h \in R^{1,000}$$

$$P(w|'for','all','the') = softmax(W_o a_h + b_o), W_o \in R^{10,000\times 1,000}, b_o \in R^{10,000}$$

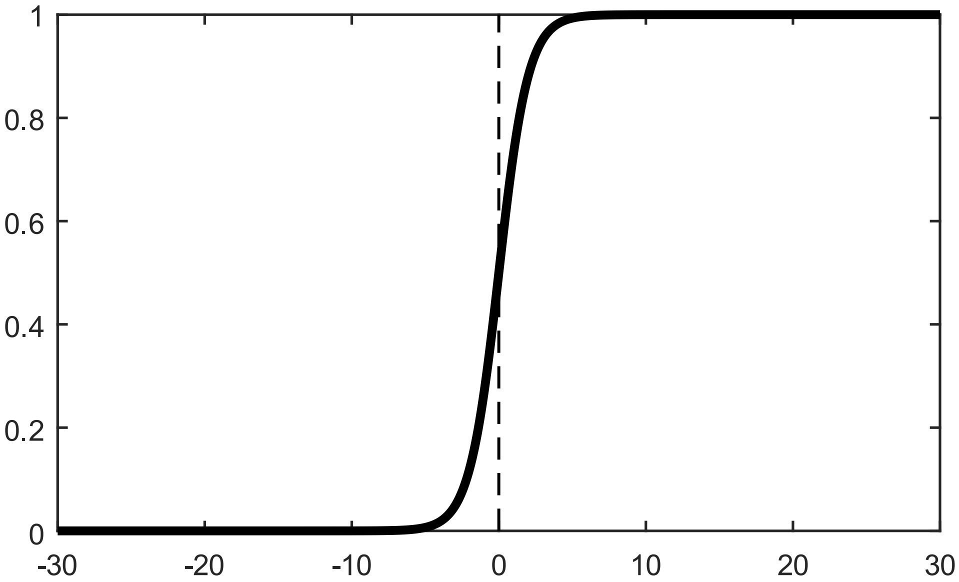
Bengio, Y. et al: A neural probabilistic language model.JMLR,3:1137–1155.2003

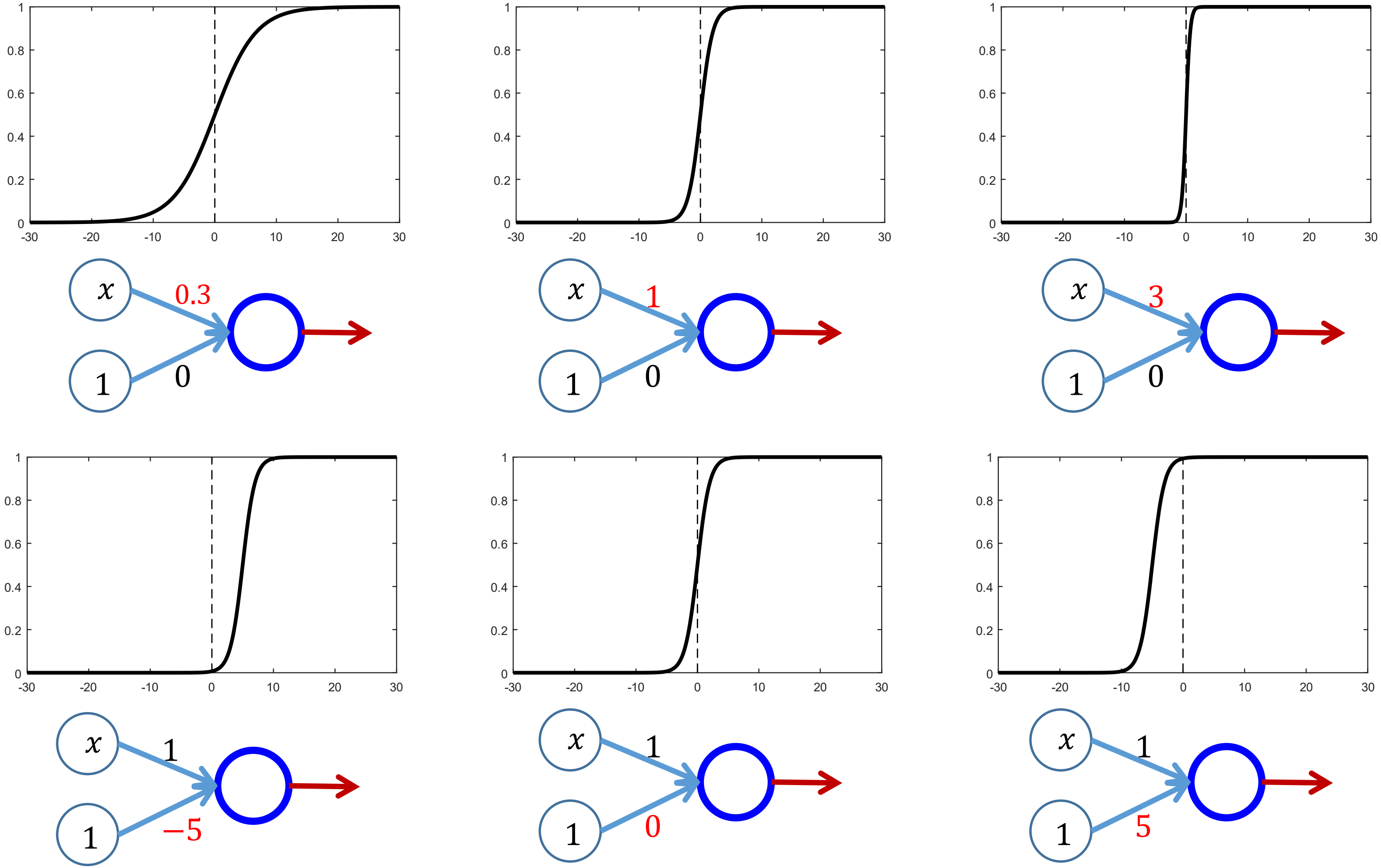
通用逼近定理

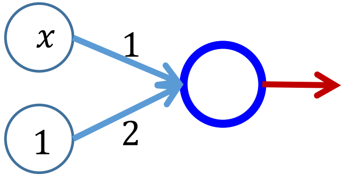
单隐层神经网络的表示能力



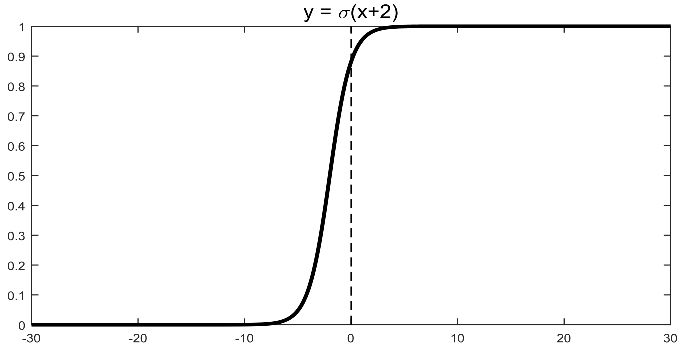
$$y = \frac{1}{1 + e^{-(wx+b)}}$$



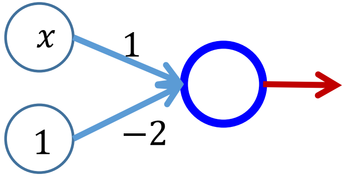

x
1
0.3
0
x
1
1
0
x
1
3
0
x
1
1
−5
x
1
1
0
x
1
1
5



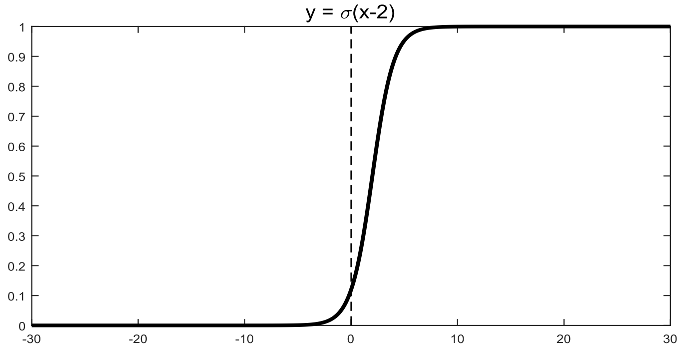
x
1
1
2



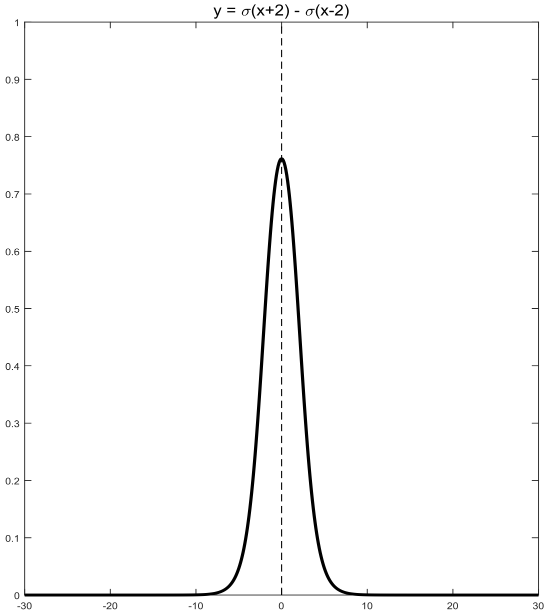
y = σ(x+2)



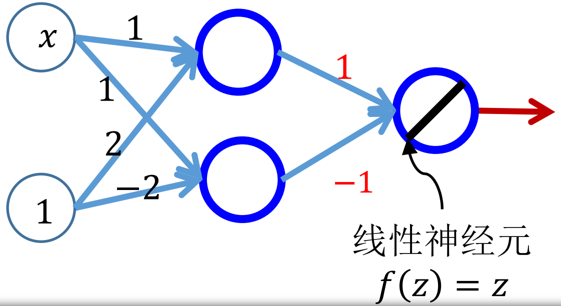
x
1
1
-2



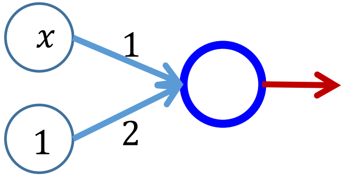
y = σ(x-2)

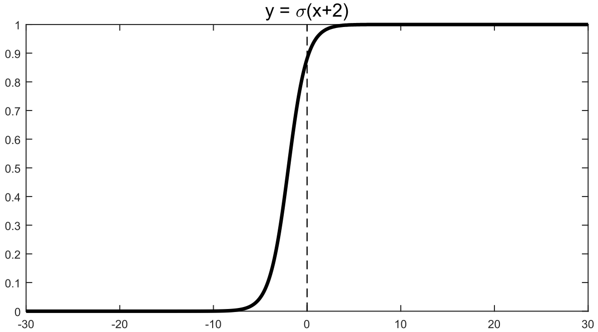


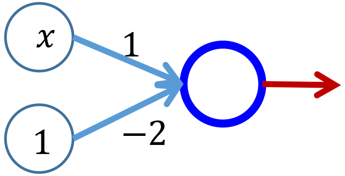
y = σ(x+2) - σ(x-2)

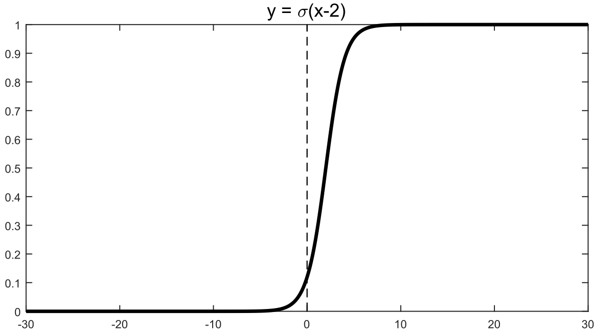


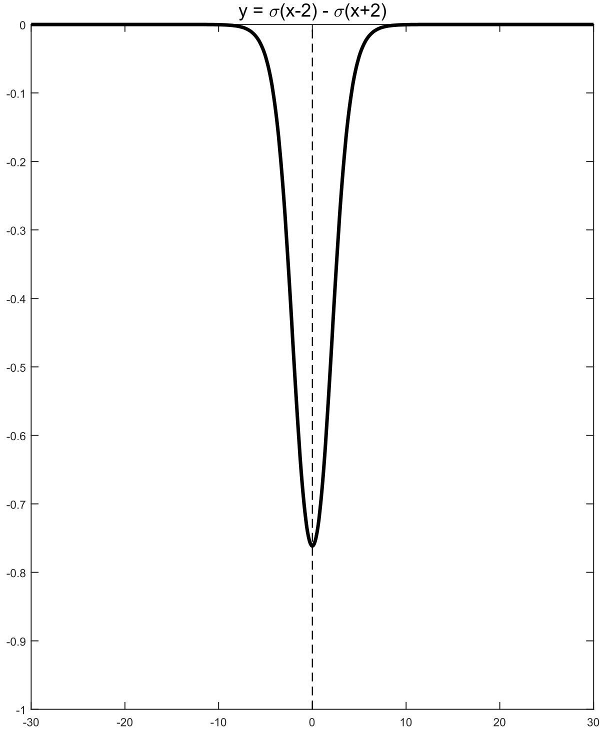
x
1
1
1
2
-2
1
-1
线性神经元
$f(z) = z$

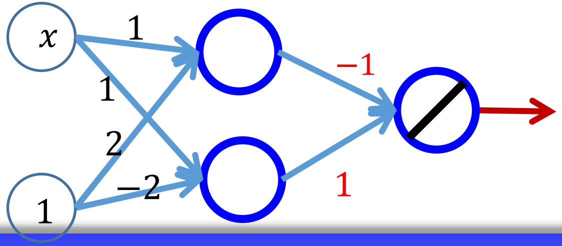

x
1
1
2

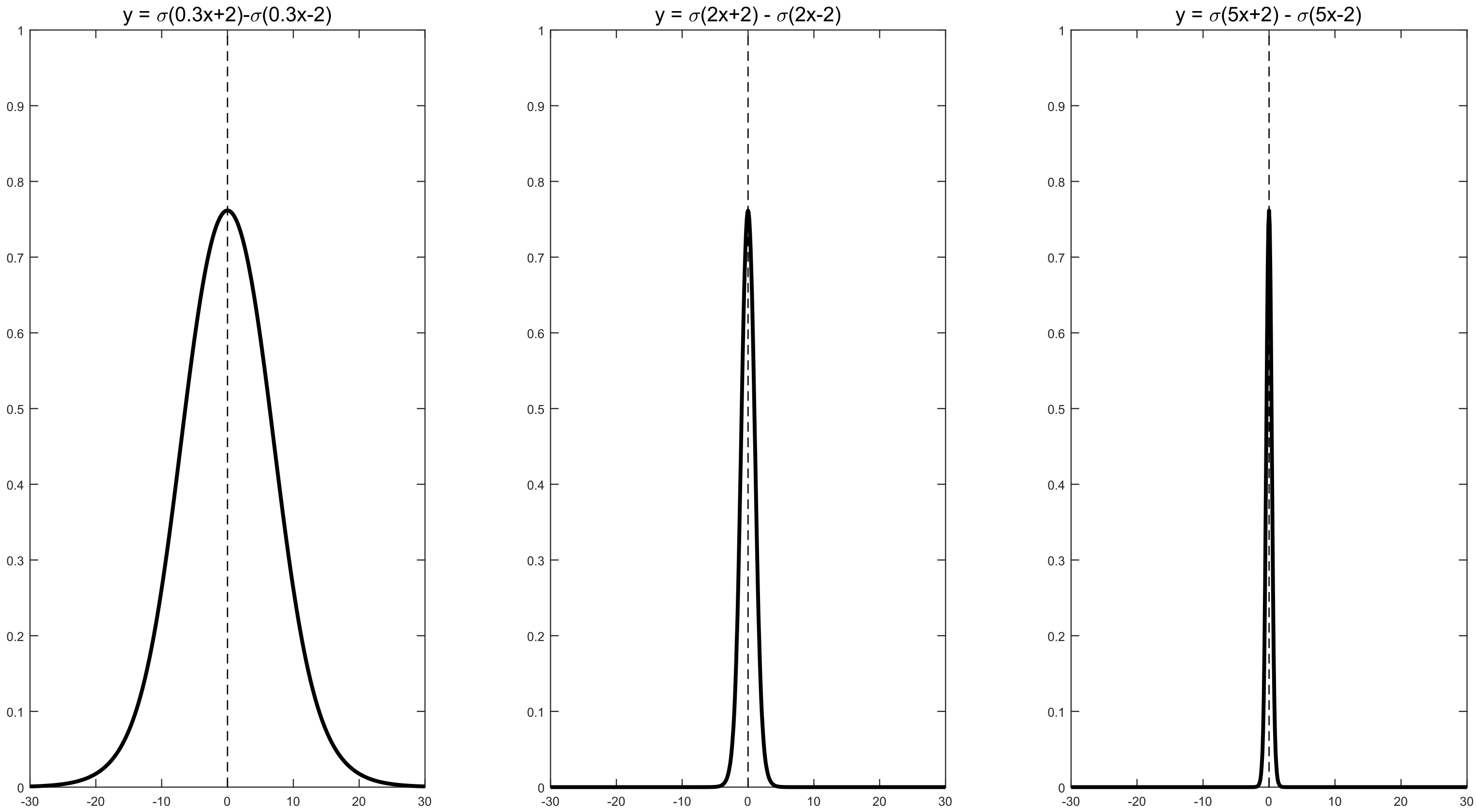

y = σ(x+2)

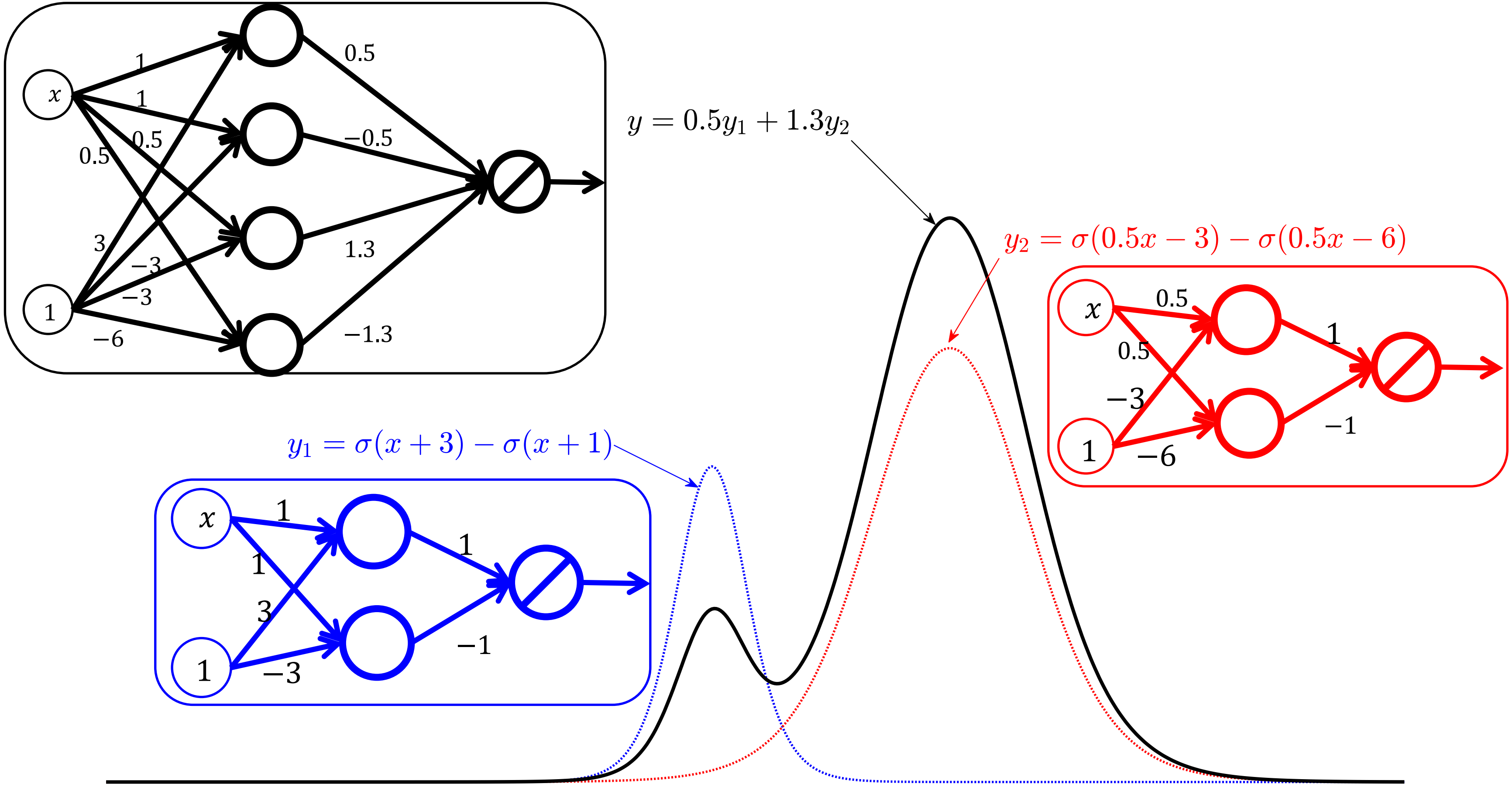

x
1
1
−2


y = σ(x-2)

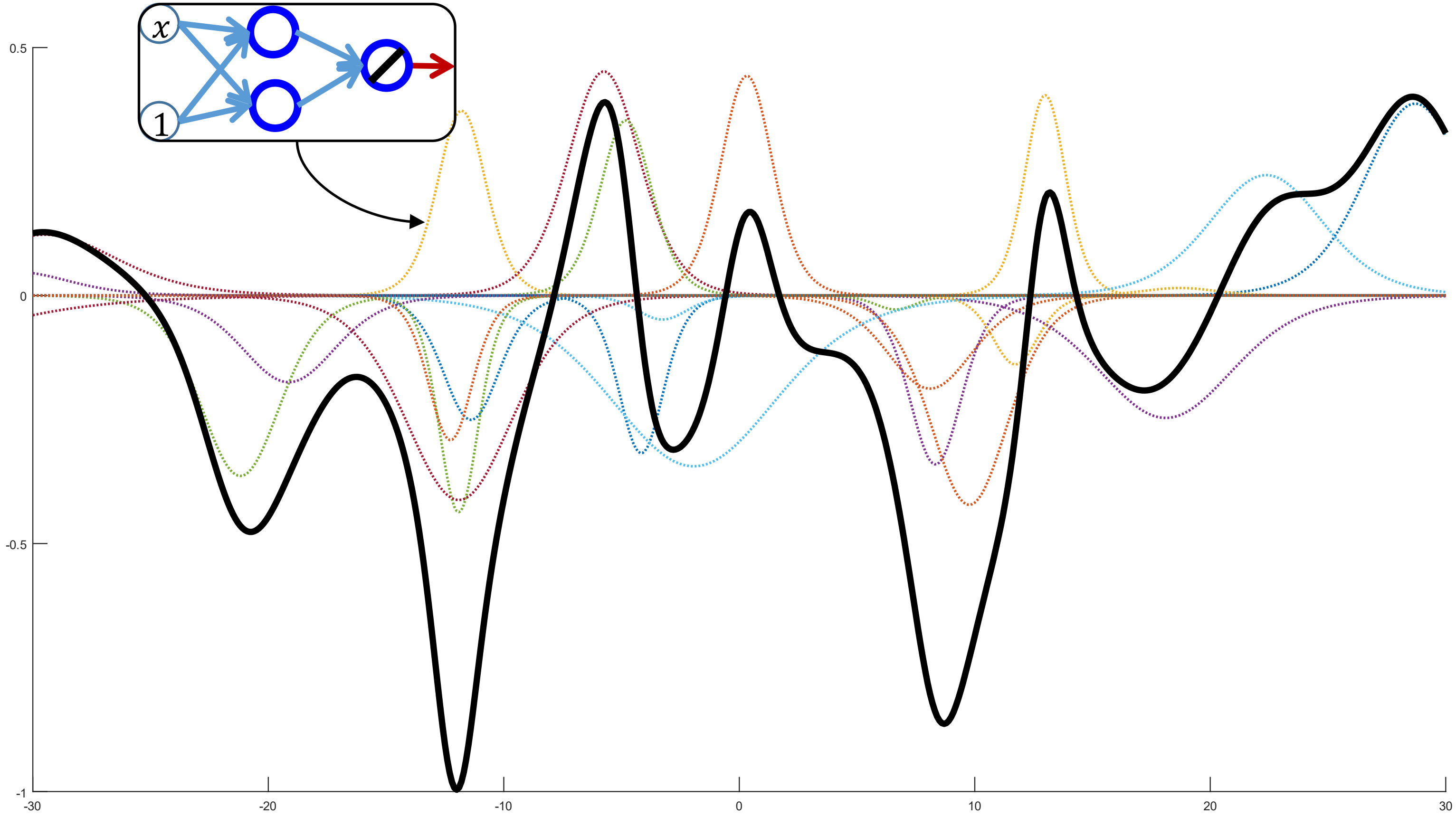

y = σ(x-2) - σ(x+2)


x
1
1
1
2
−2
−1
1

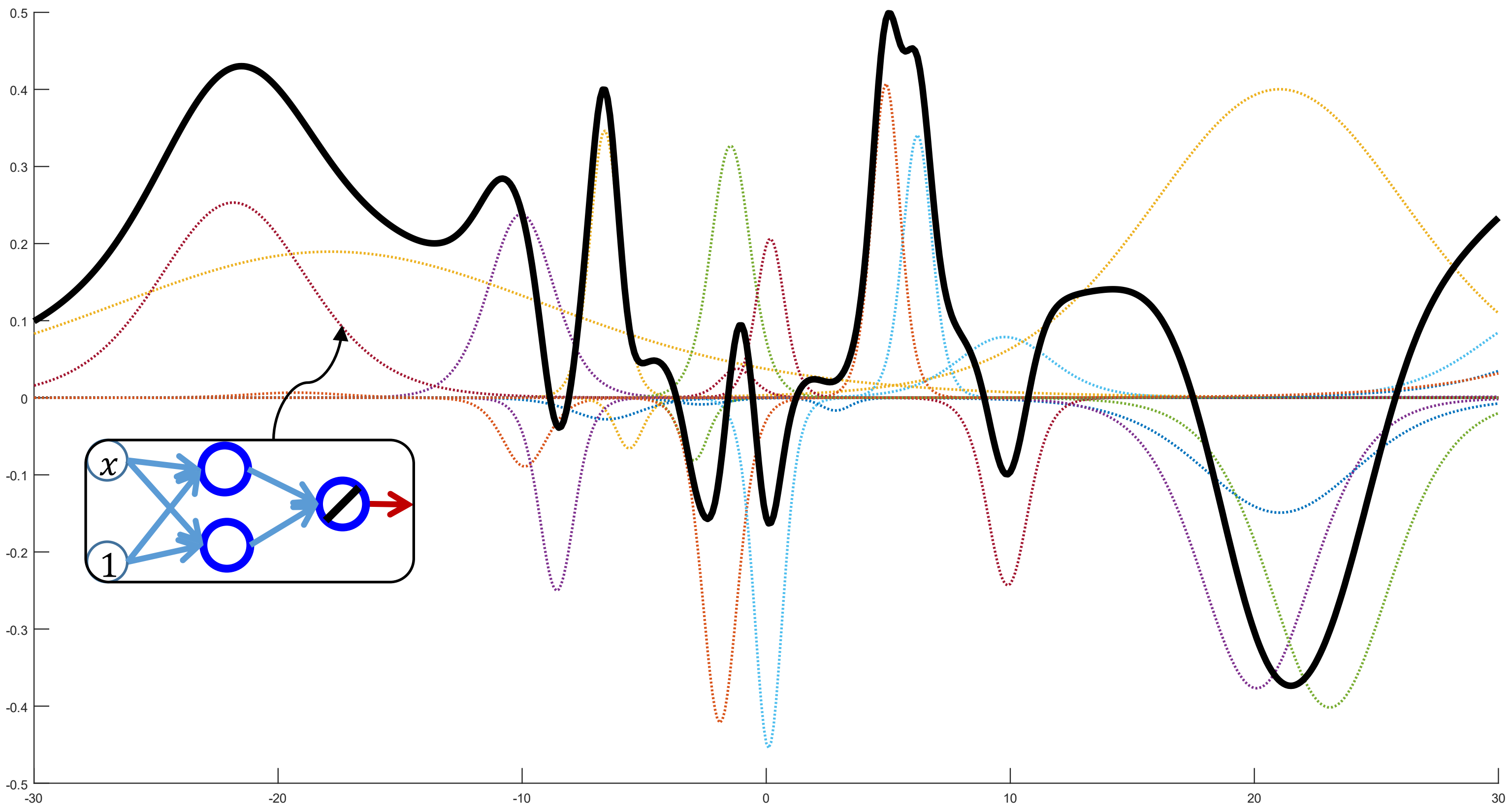

y = σ(0.3x+2)-σ(0.3x-2)
y = σ(2x+2) - σ(2x-2)
y = σ(5x+2) - σ(5x-2)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-30
-20
-10
0
10
20
30

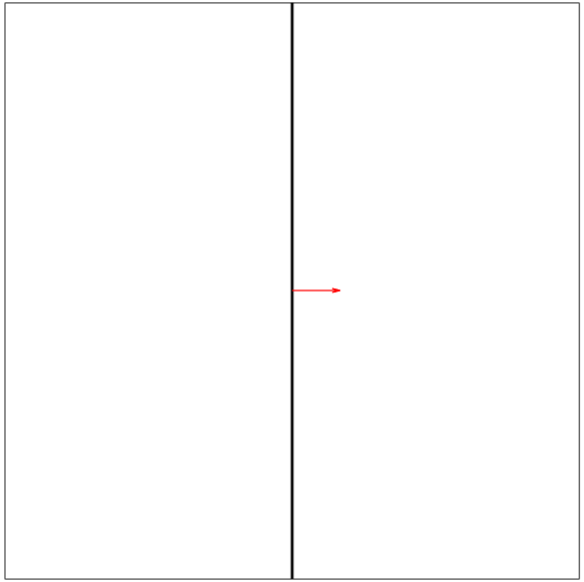


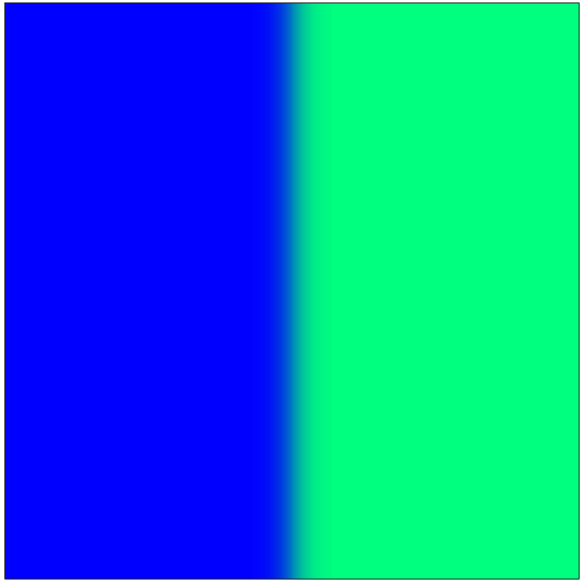
x
1
1
1
0.5
0.5
3
−3
−3
−6
0.5
−0.5
1.3
−1.3
$y = 0.5y_1 + 1.3y_2$
$y_2 = \sigma(0.5x-3) - \sigma(0.5x-6)$
0.5
0.5
−3
−6
1
−1
$y_1 = \sigma(x+3) - \sigma(x+1)$
1
1
3
−3
1
−1

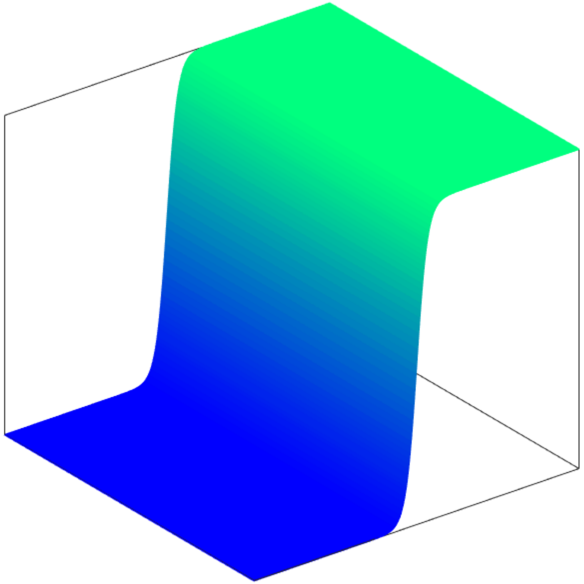


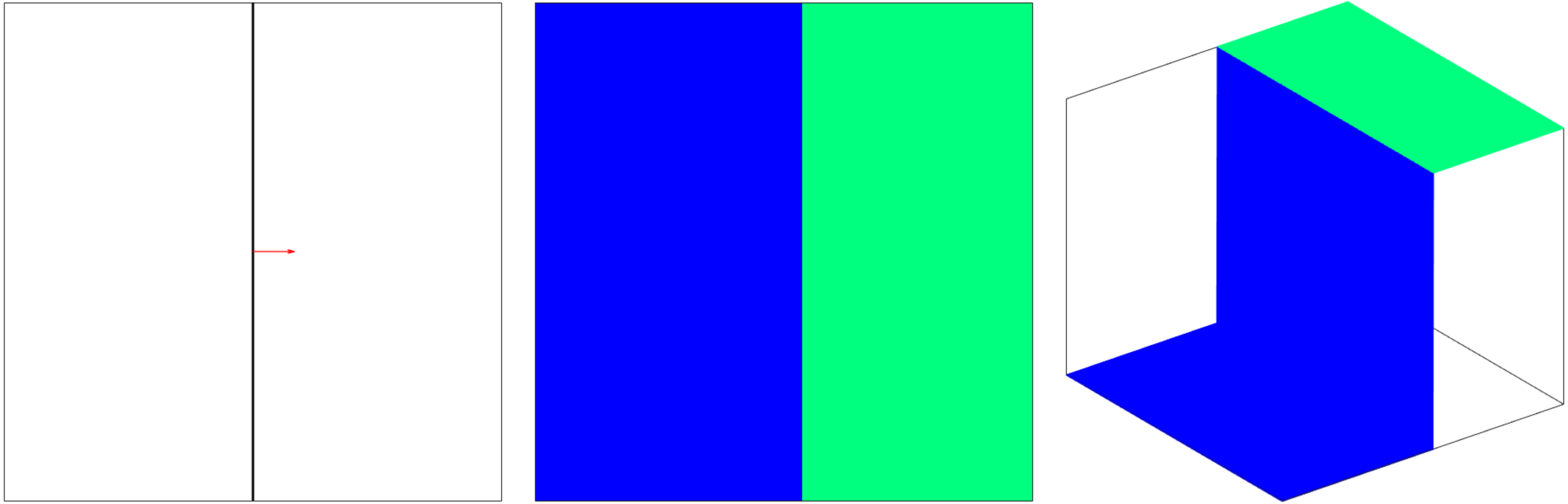
0.5
0
-0.5
-1
-30
-20
-10
0
10
20
30
x
1

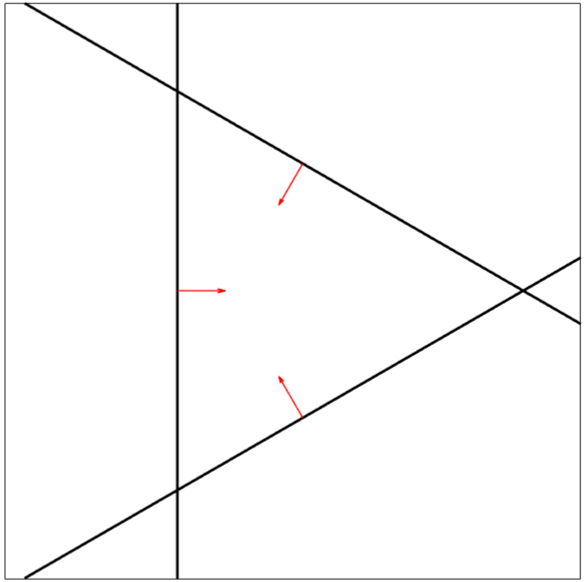
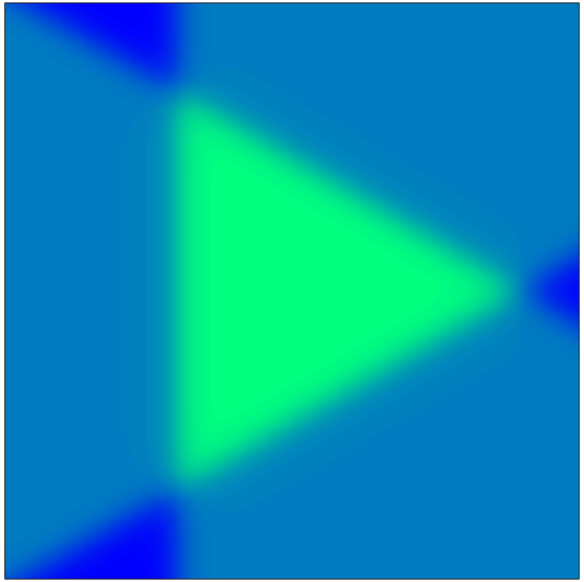
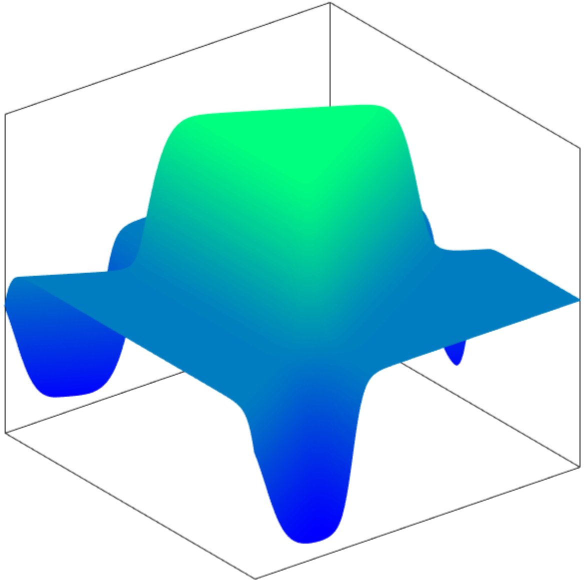

0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-30
-20
-10
0
10
20
30
x
1

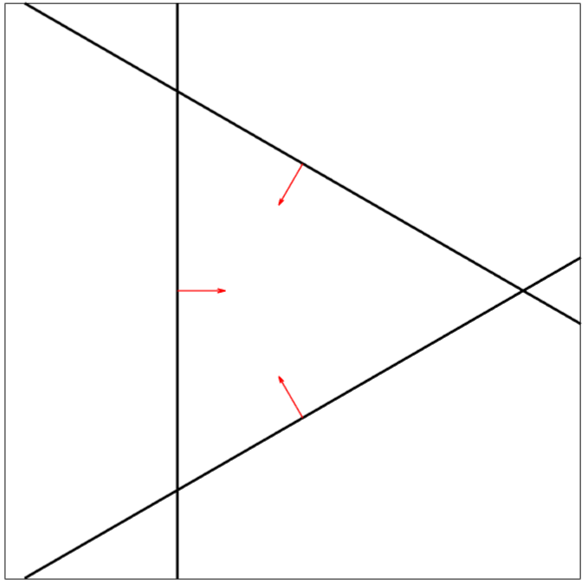


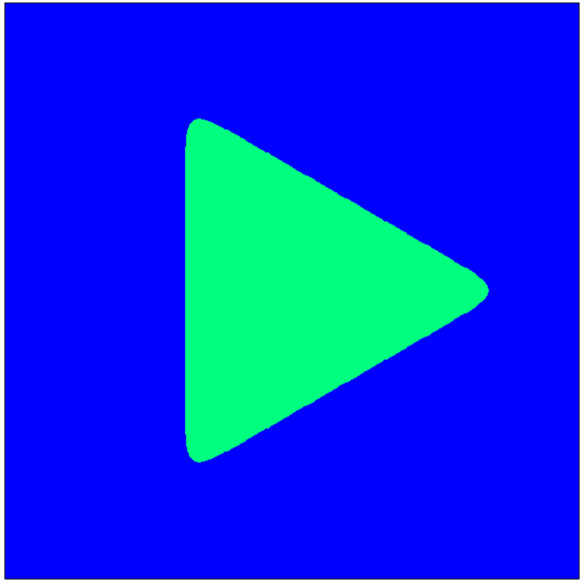


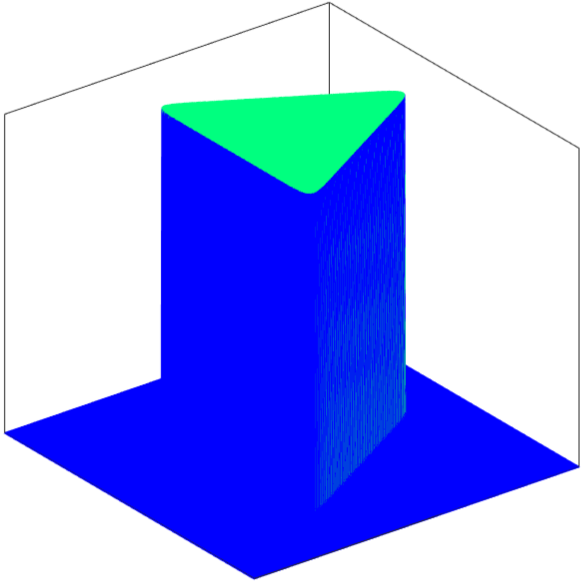


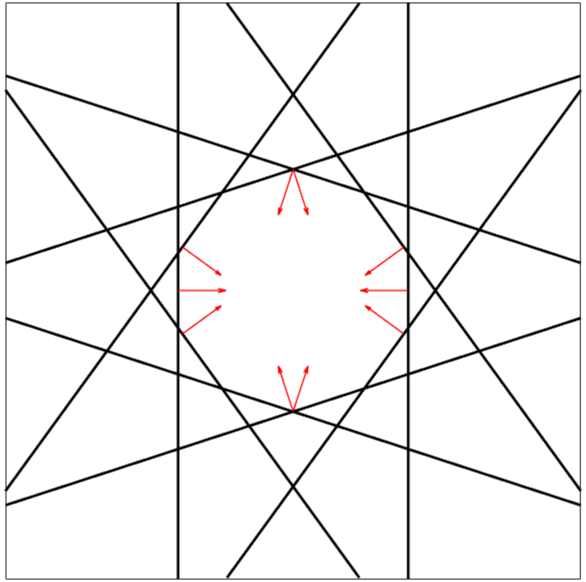


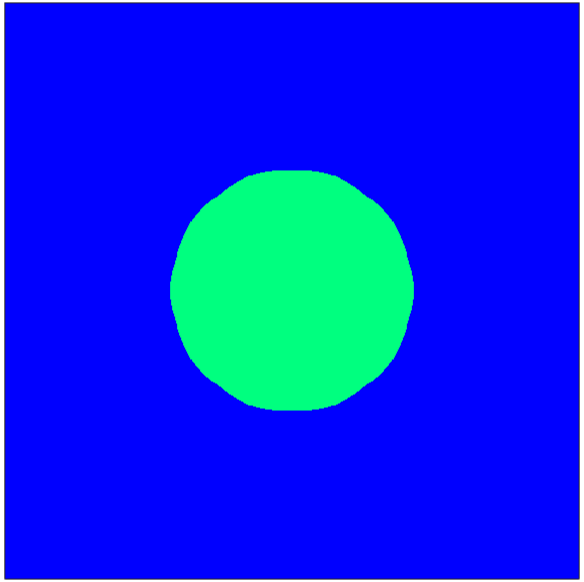




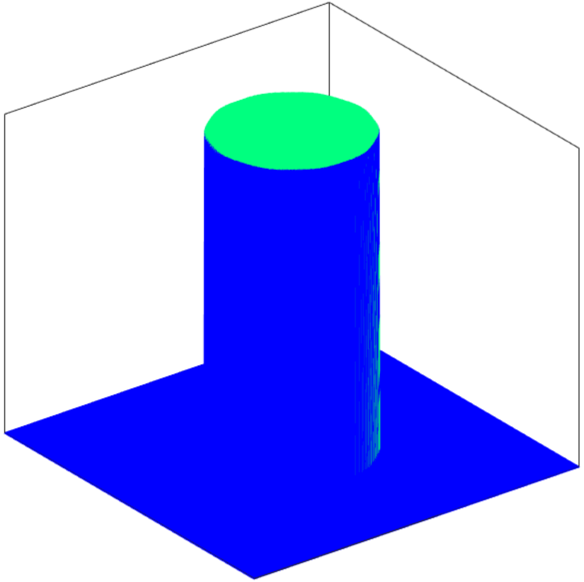


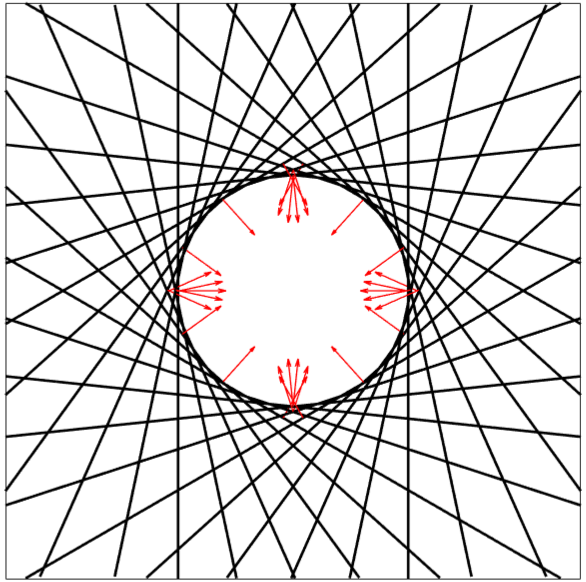


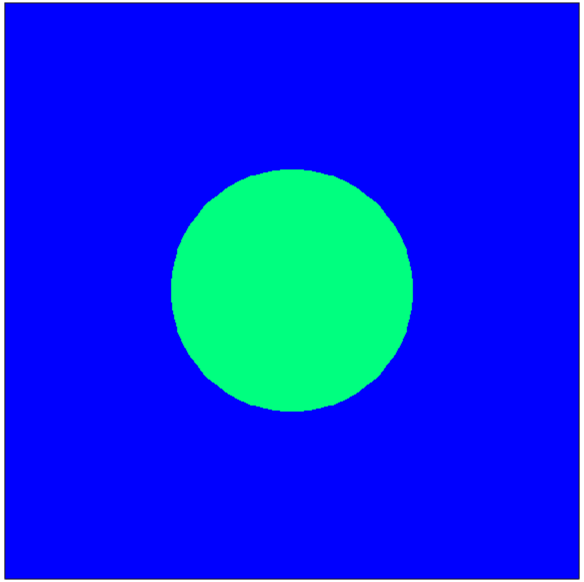


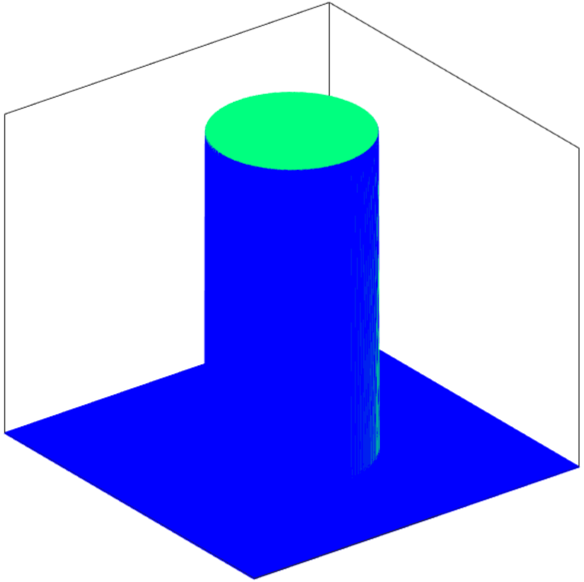


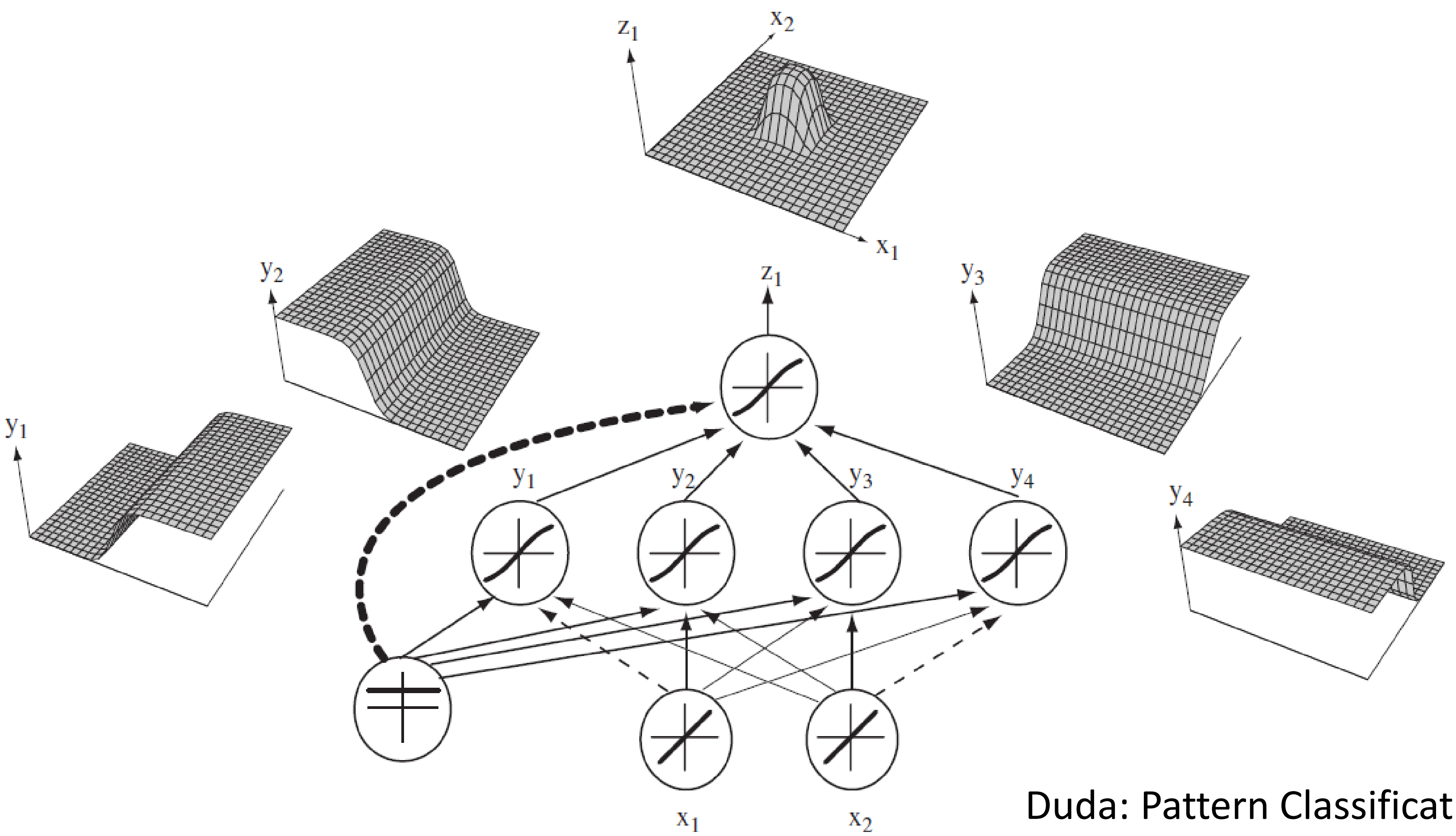




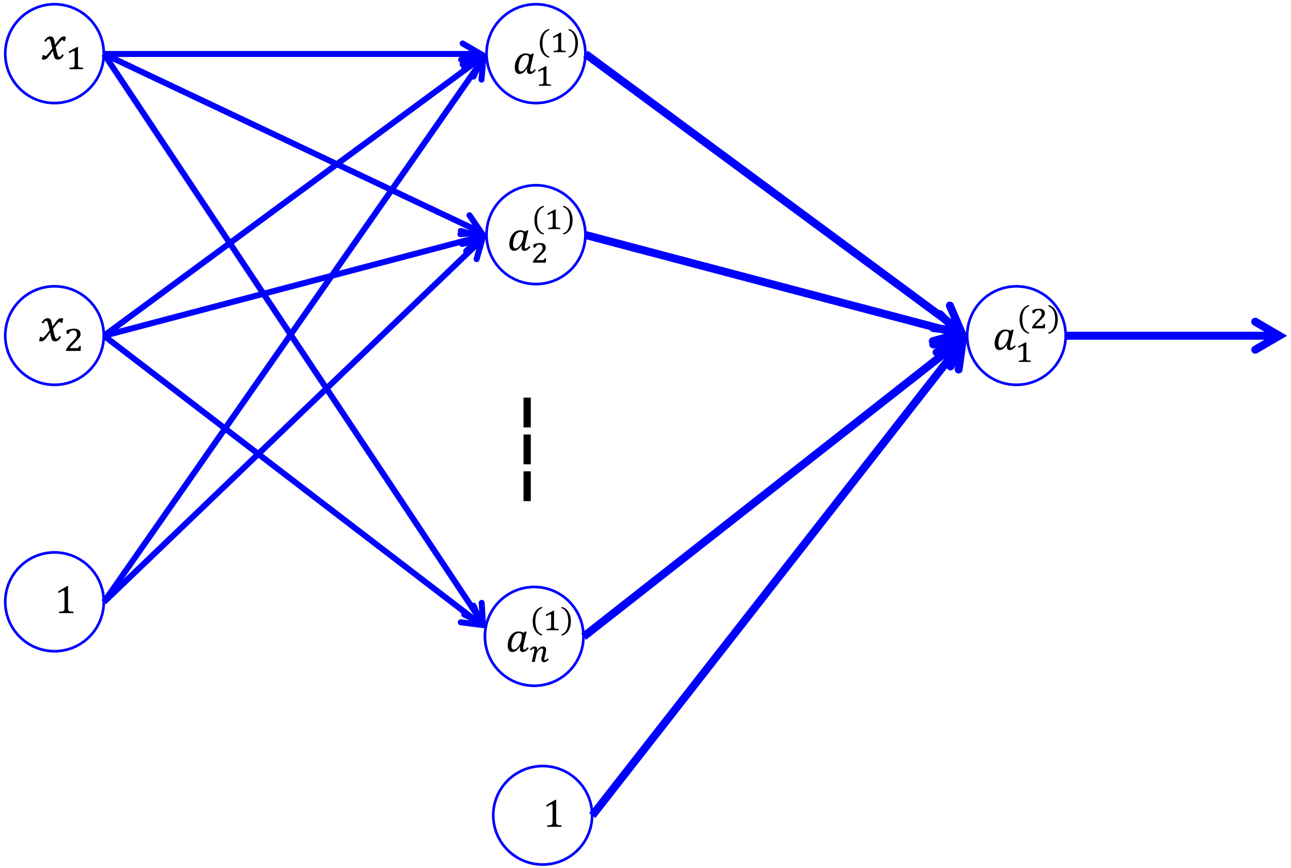


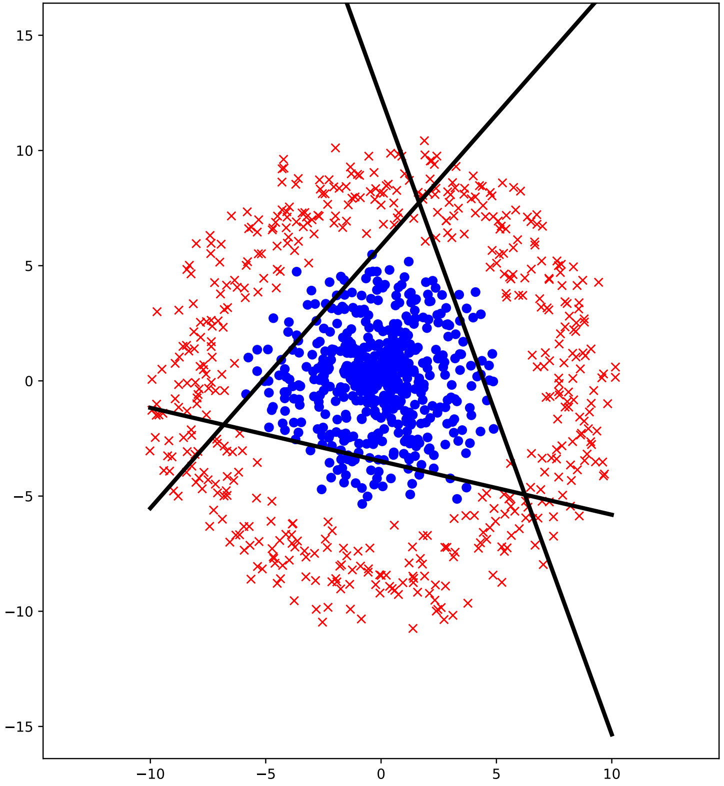


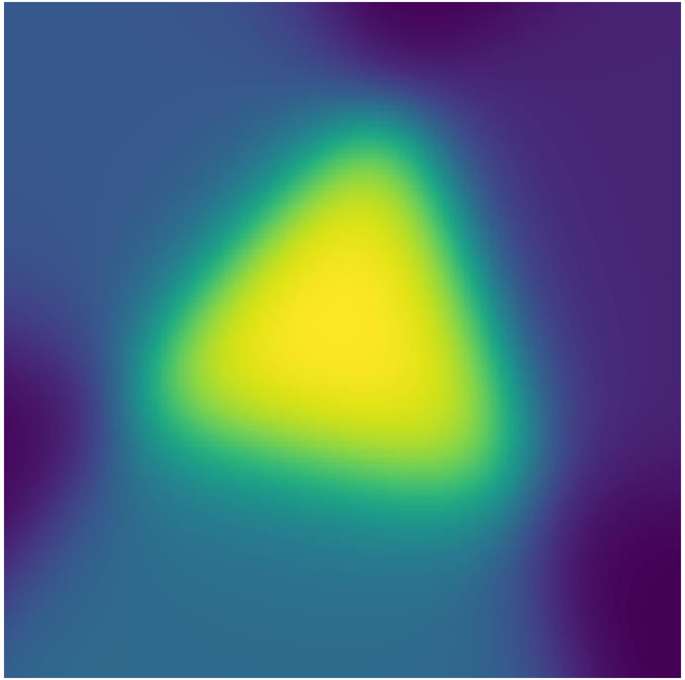




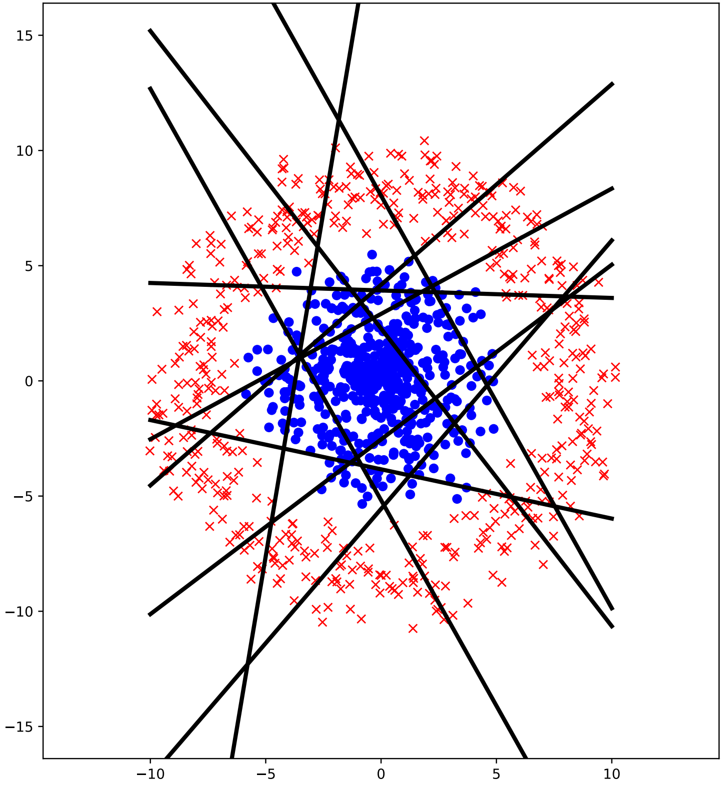
Duda: Pattern Classification

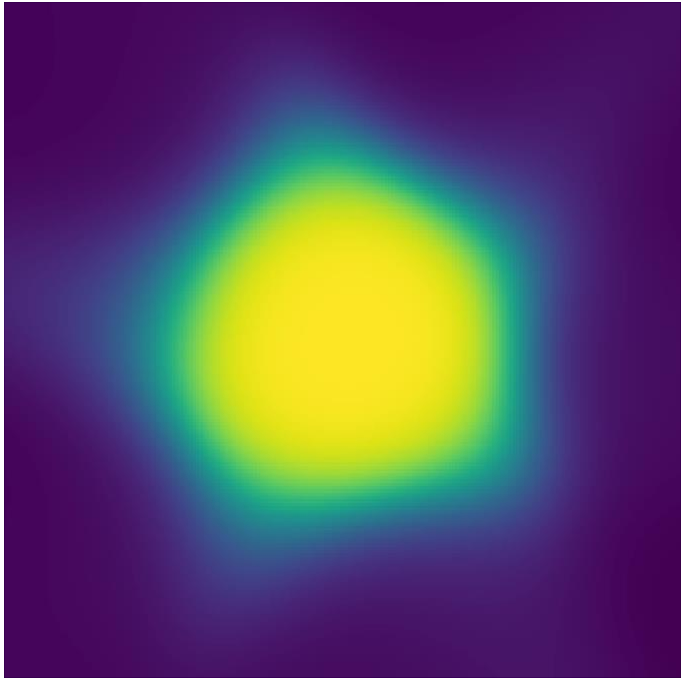

x_1
x_2
1
$a_1^{(1)}$
$a_2^{(1)}$
$a_n^{(1)}$
1
$a_1^{(2)}$


15
10
5
0
−5
−10
−15
−10
−5
0
5
10

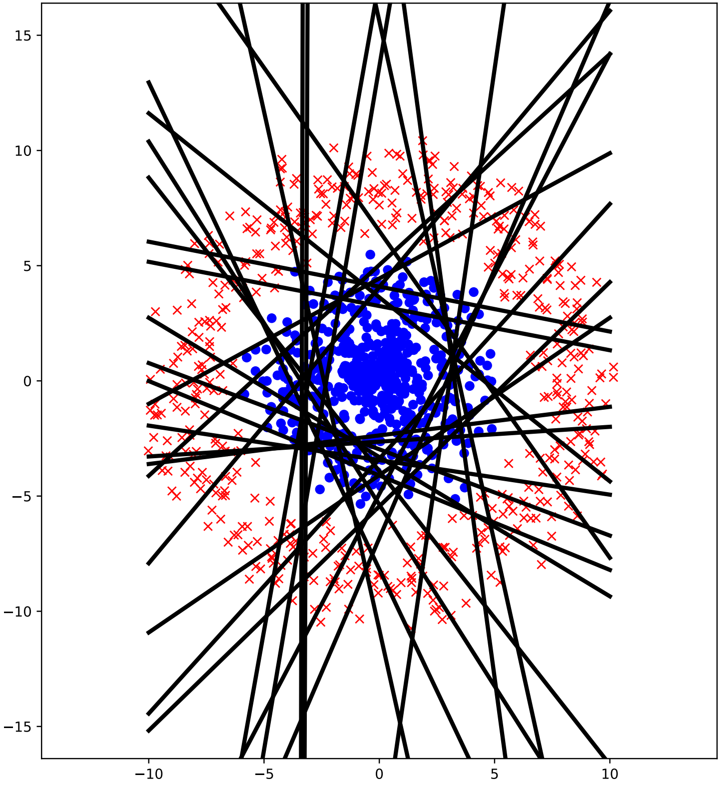


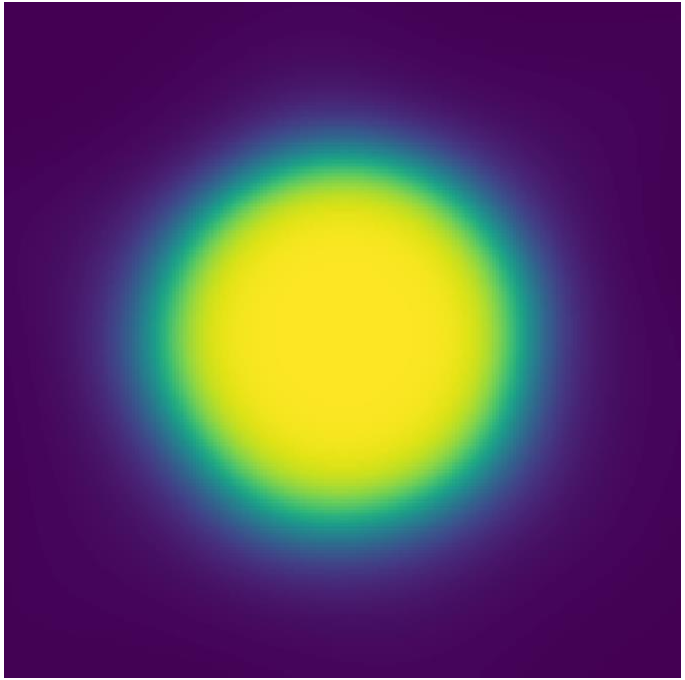
Hidden = 10



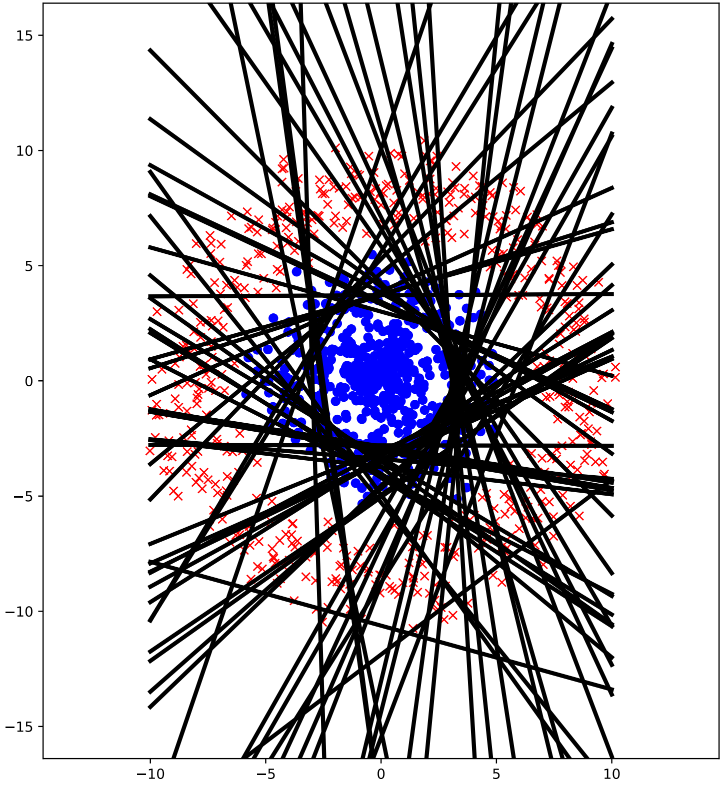


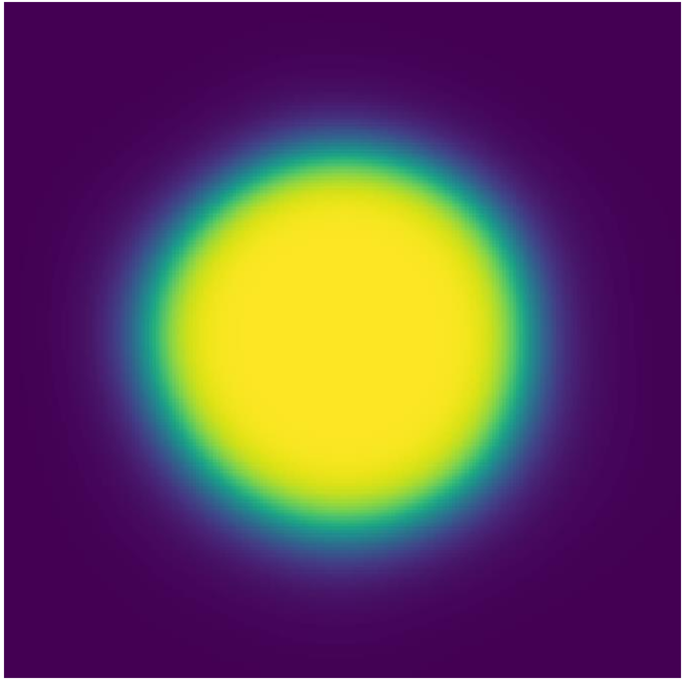
Hidden = 100

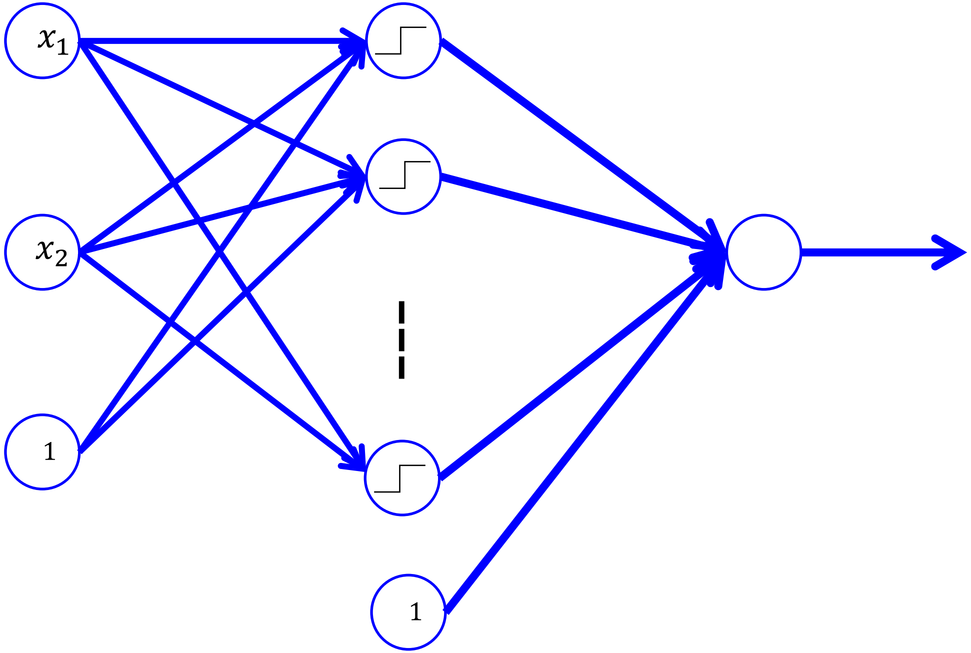


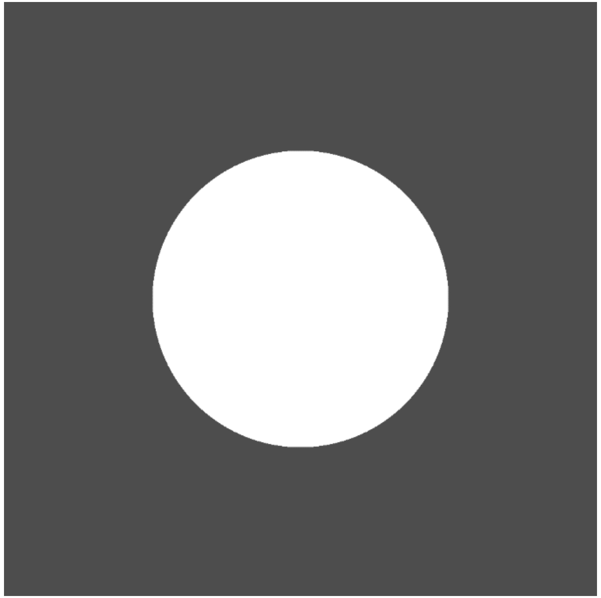


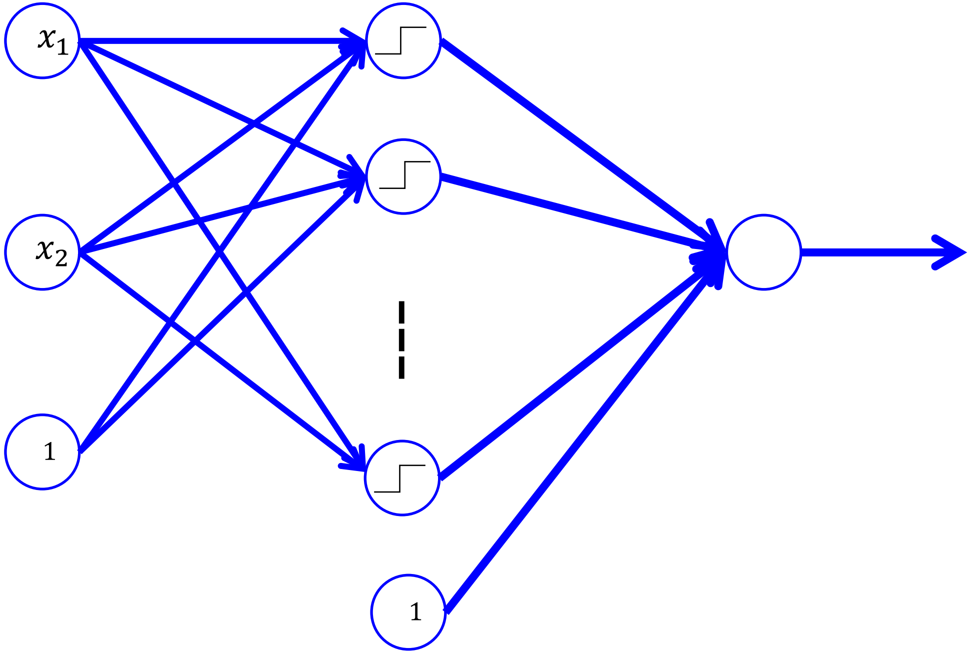
Hidden = 300

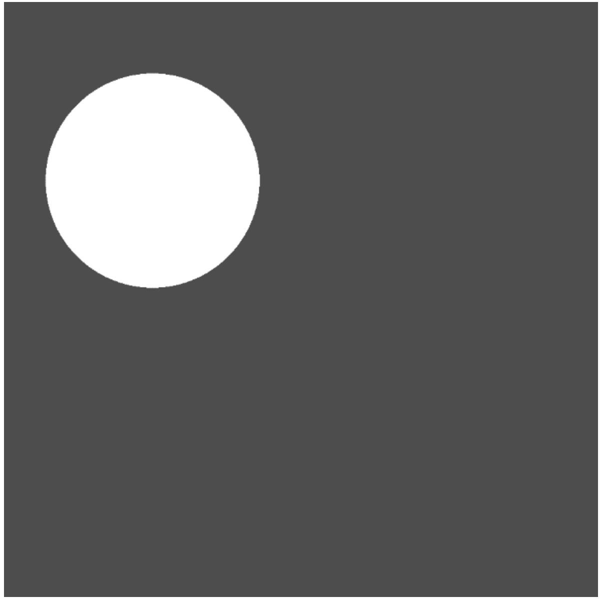


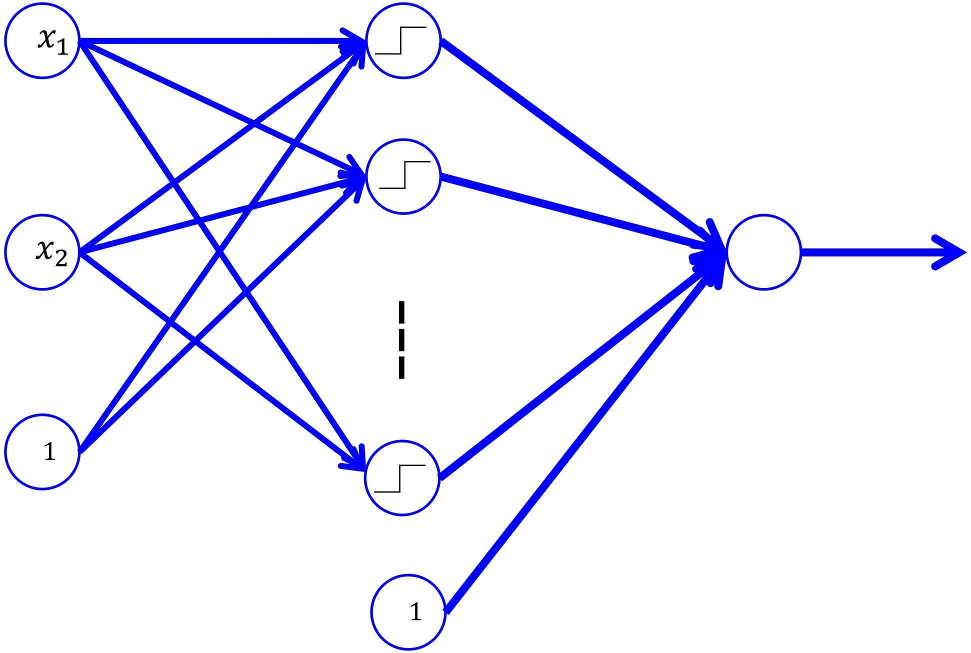



x_1
x_2
1
1

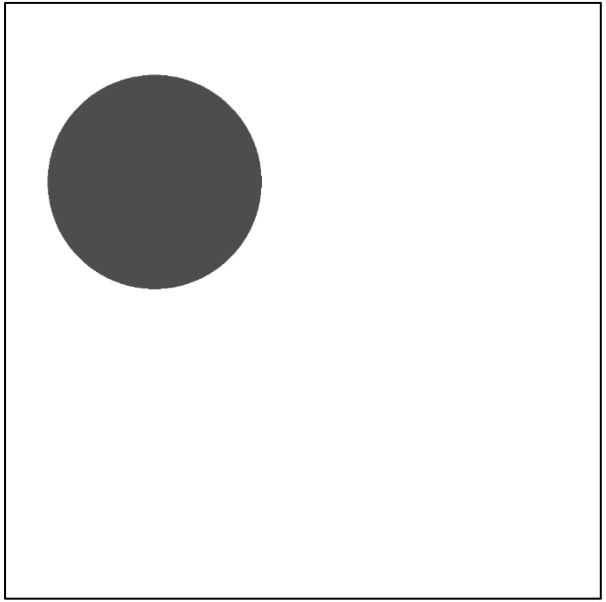


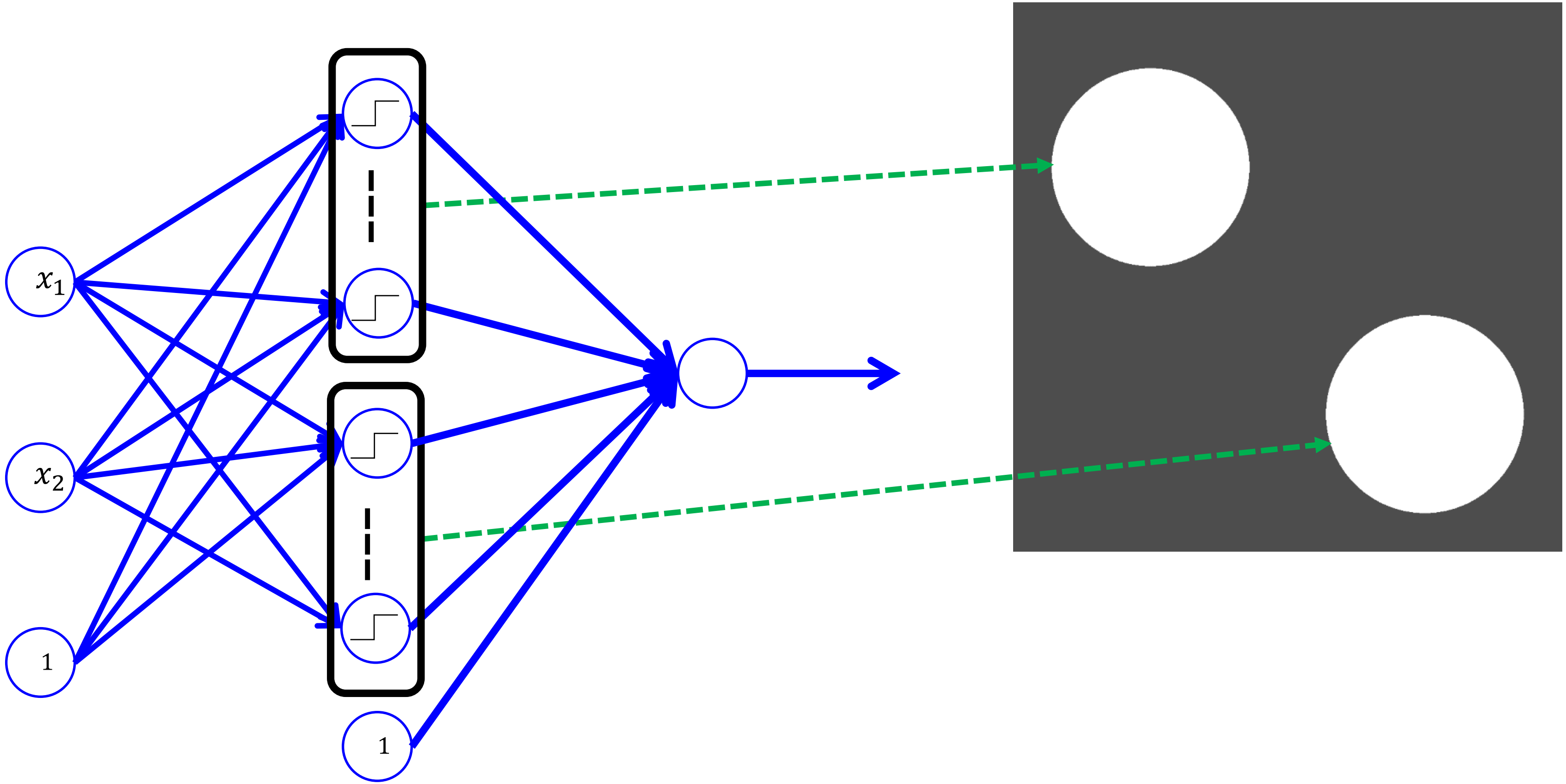

x_1
x_2
1
1

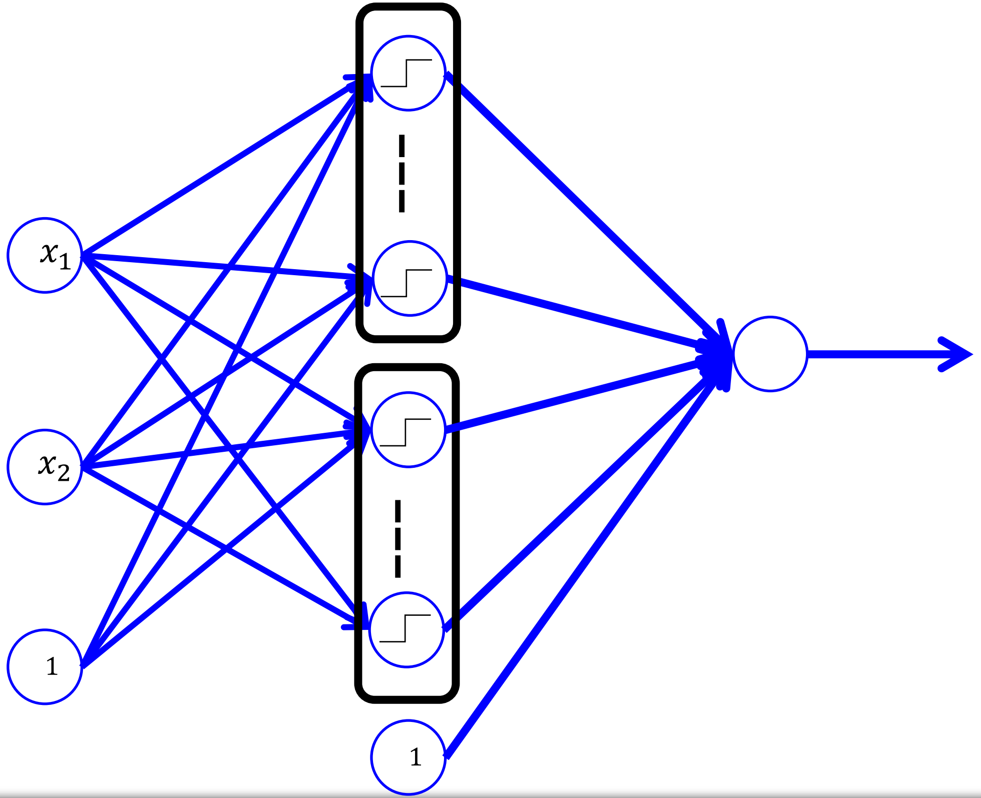




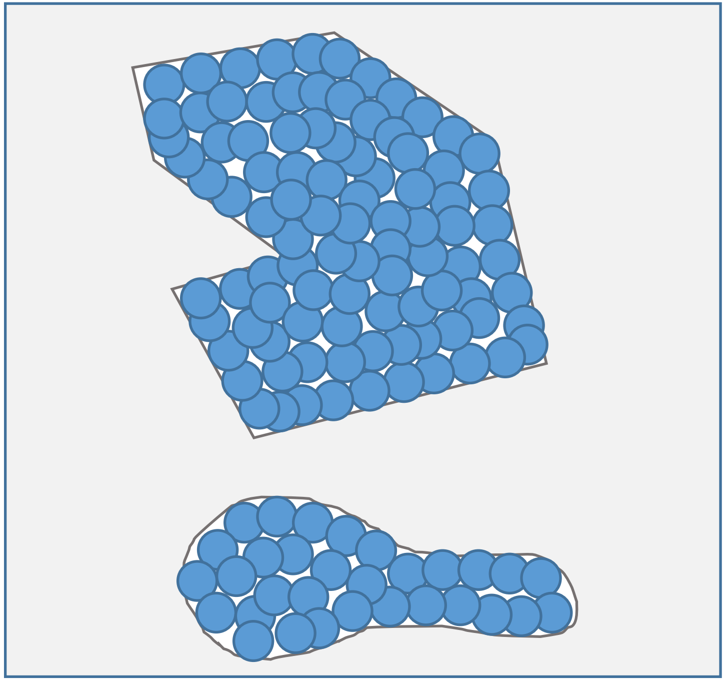
x_1
x_2
1
1

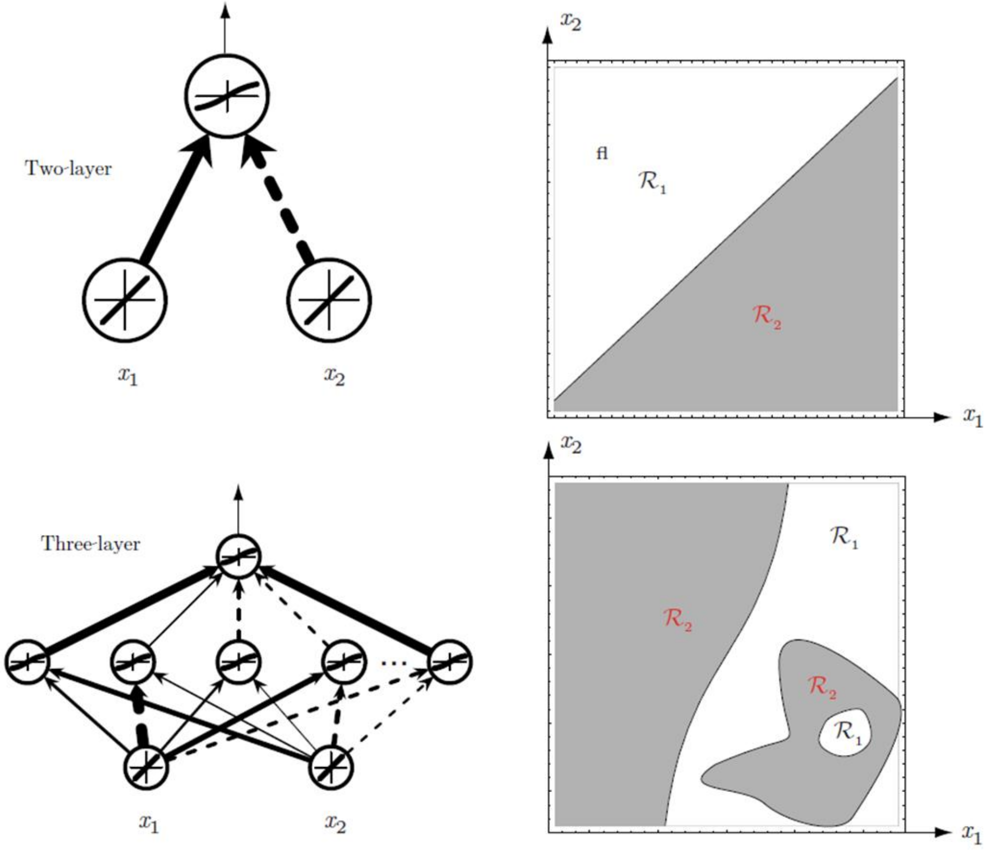



x_1
x_2
1
1



x_1
x_2
1
1





Duda: Pattern Classification

通用逼近定理(Universal Approximation Theorem)

- 单隐层神经网络（输出单元为线性响应、隐层单元为非线性（sigmoid, tanh, Relu...））
  - 可以逼近任意连续函数
  - 可以拟合任意决策边界
  - 可以表示任意布尔函数

- 困难：
  - 可能需要巨大数目（甚至无穷个）的神经元才能达到足够的逼近精度

总结

- 多层感知机(Multi-Layer Perceptron, MLP)
  - 多层神经元叠加构成的一个神经元网络
  - 通过复合(Composition)的方式表示一个函数
- 多层感知机的学习算法
  - 梯度下降法
  - 误差后向传播算法(error Back Propagation, BP算法)
    - 本质是复合函数求导的链式法则
- 通用逼近定理(Universal Approximation Theorem)
  - 单隐层神经网络可以表示任意连续函数
  - 不确定是否能真正学习出任意连续函数
  - 可能需要巨大数目的隐层神经元才能达到合适的表示精度