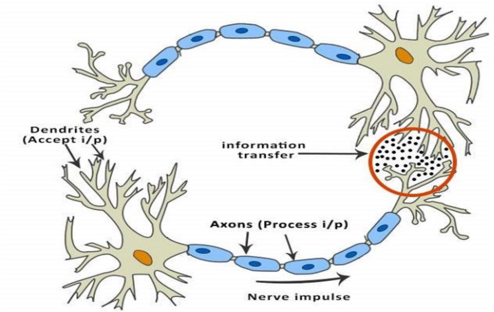
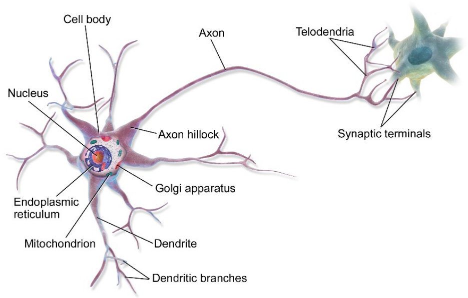
ANN and Handwritten Character Recognition

YAO ZHAO

ANN

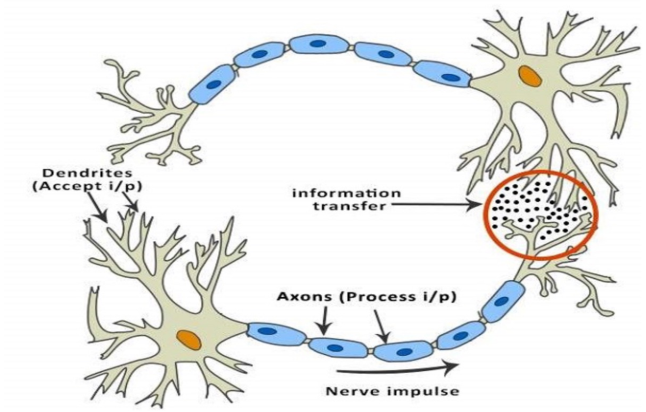
- The study of artificial neural networks was inspired by attempts to simulate biological neural systems (e. g. human brain).
- Basic structural and functional unit: nerve cells called neurons
- Work Mechanism
  - Different neurons are linked together via axons (轴突) and dendrite (树突)
  - When one neuron is excited after stimulation, it sends chemicals to the connected neurons, thereby changing the potential (电位) within these neurons.
  - If the potential (电位) of a neuron exceeds a “threshold”, then it is activated and send chemicals to other neurons.

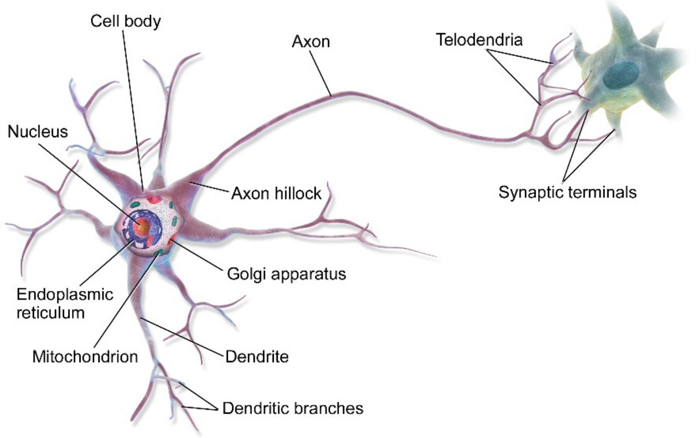




ANN

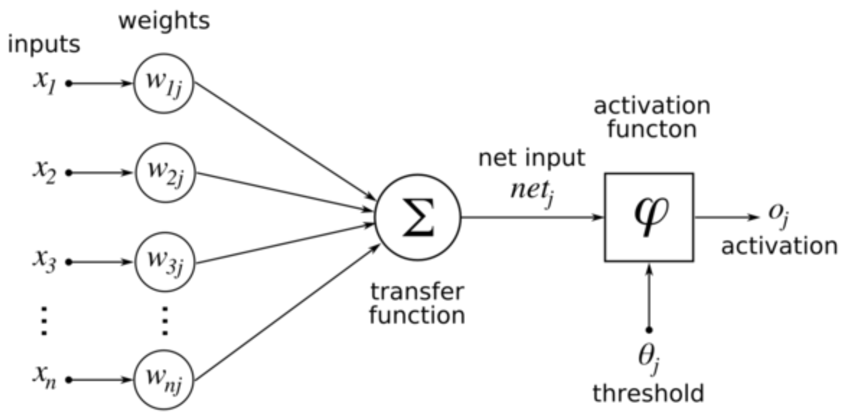
- A neuron is connected to the axons (轴突) of other neurons via dendrites (树突), which are extensions from the cell body of the neuron.
- The contact point between a dendrite (树突) and an axon (轴突) is called a synapse (突触).
- The human brain learns by changing the strength of the synaptic connection between neurons upon repeated stimulation by the same impulse.





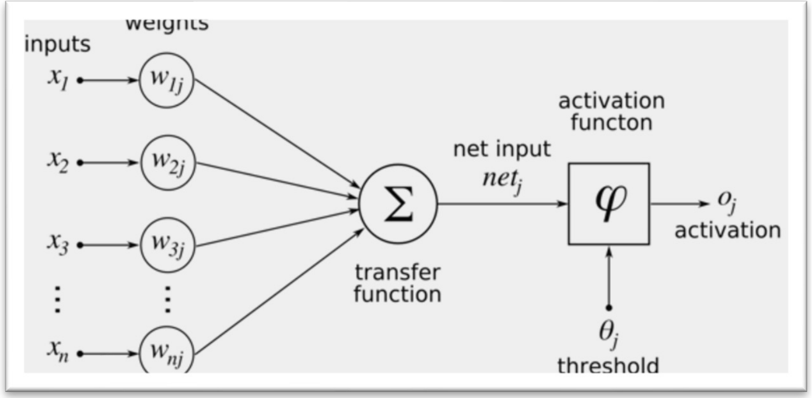
Artificial Neuron Mathematical Model

- Input: x_i from the i-th neuron
- Weights: connection weights (synapse)
- Output: $o_j = \varphi(\sum_{i=1}^{n} w_{ij}x_i - \theta_j)$



Artificial Neuron Model

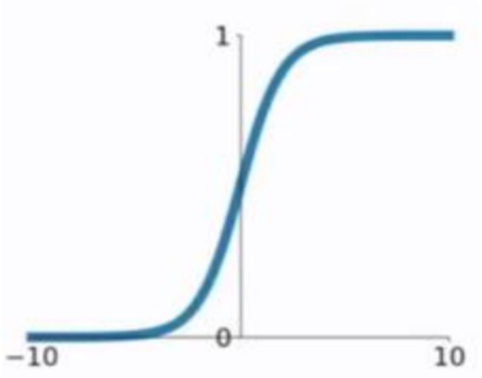
- Output: $o_j = \varphi(\sum_{i=1}^{n} w_{ij}x_i + b_j)$
- Ideal activation function: step function but inapplicable
- Common activation function：sigmoid，tanh，ReLU

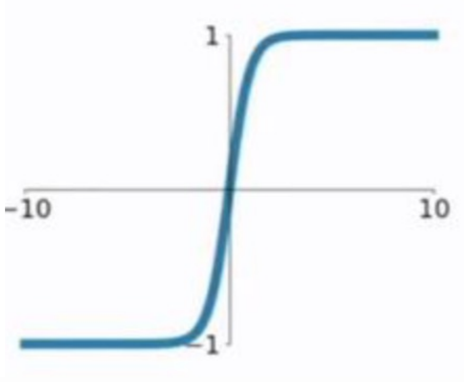


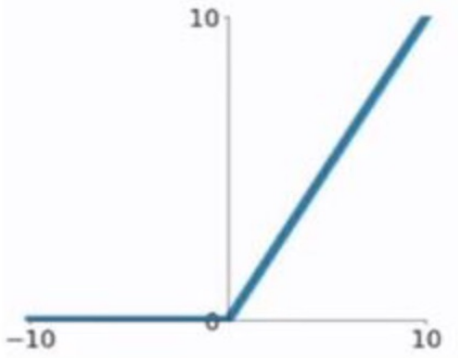
$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$g(z) = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{if } z < 0 \end{cases}$$

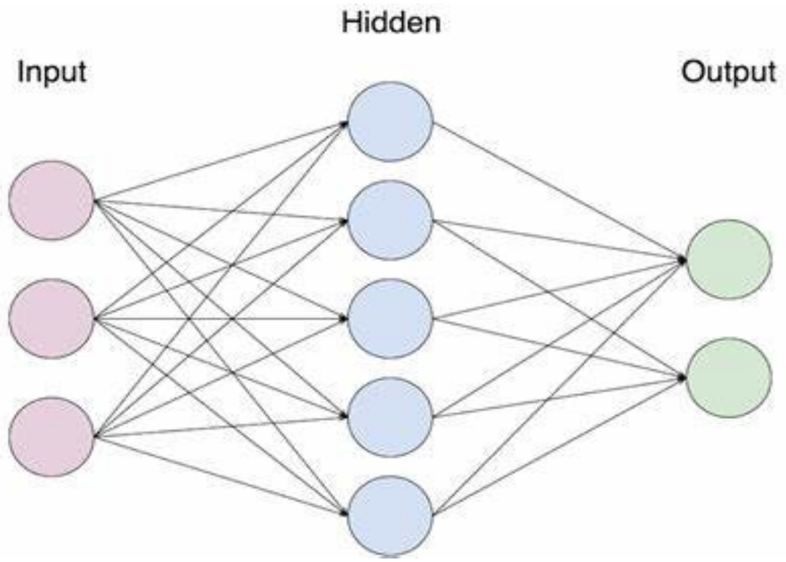






Artificial Neural Networks

- Consist of multiple artificial neurons
- Usually have the structure of an input layer, multiple hidden layer, an output layer
- The design of an NN or AutoML aims to design appropriate hidden layers and connection weights.



3-layer Feedforward neural networks

One Inference Process

$$x_j^{(1)} = f(a_j^{(1)})$$

$$x_k^{(2)} = f(a_k^{(2)})$$

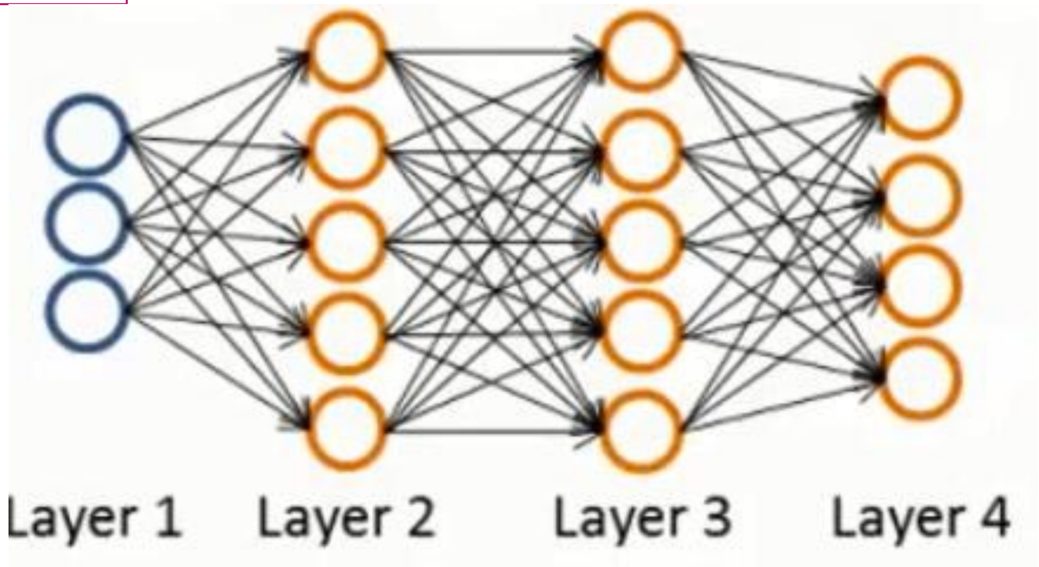
$$y_l = x_l^{(3)} = f\left(a_l^{(3)}\right)$$

$$x = x^{(0)} \to a^{(1)} \to x^{(1)} \to a^{(2)} \to x^{(2)} \to a^{(3)} \to x^{(3)} = y$$

$$a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i^{(0)} + b_j^{(1)}$$

$$a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)}$$

$$a_l^{(3)} = \sum_{k=1}^{N} w_{lk}^{(3)} x_k^{(2)} + b_l^{(3)}$$

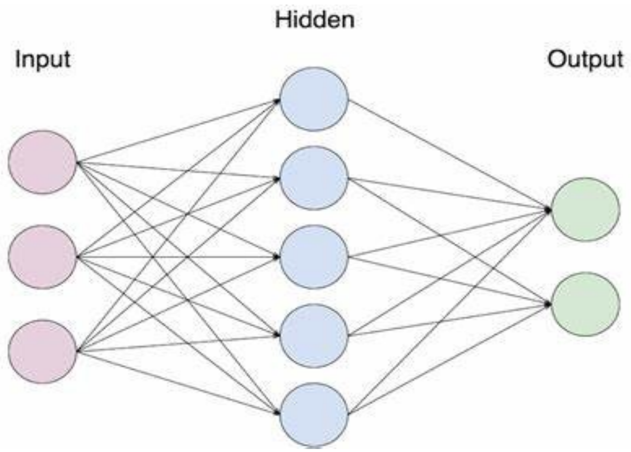


Superscript of w：layer index

Training of NN

- W and Threshold values decide the output of NN
- The training is to find appropriate values for W and Threshold
- The learning process is to tune weight matrix

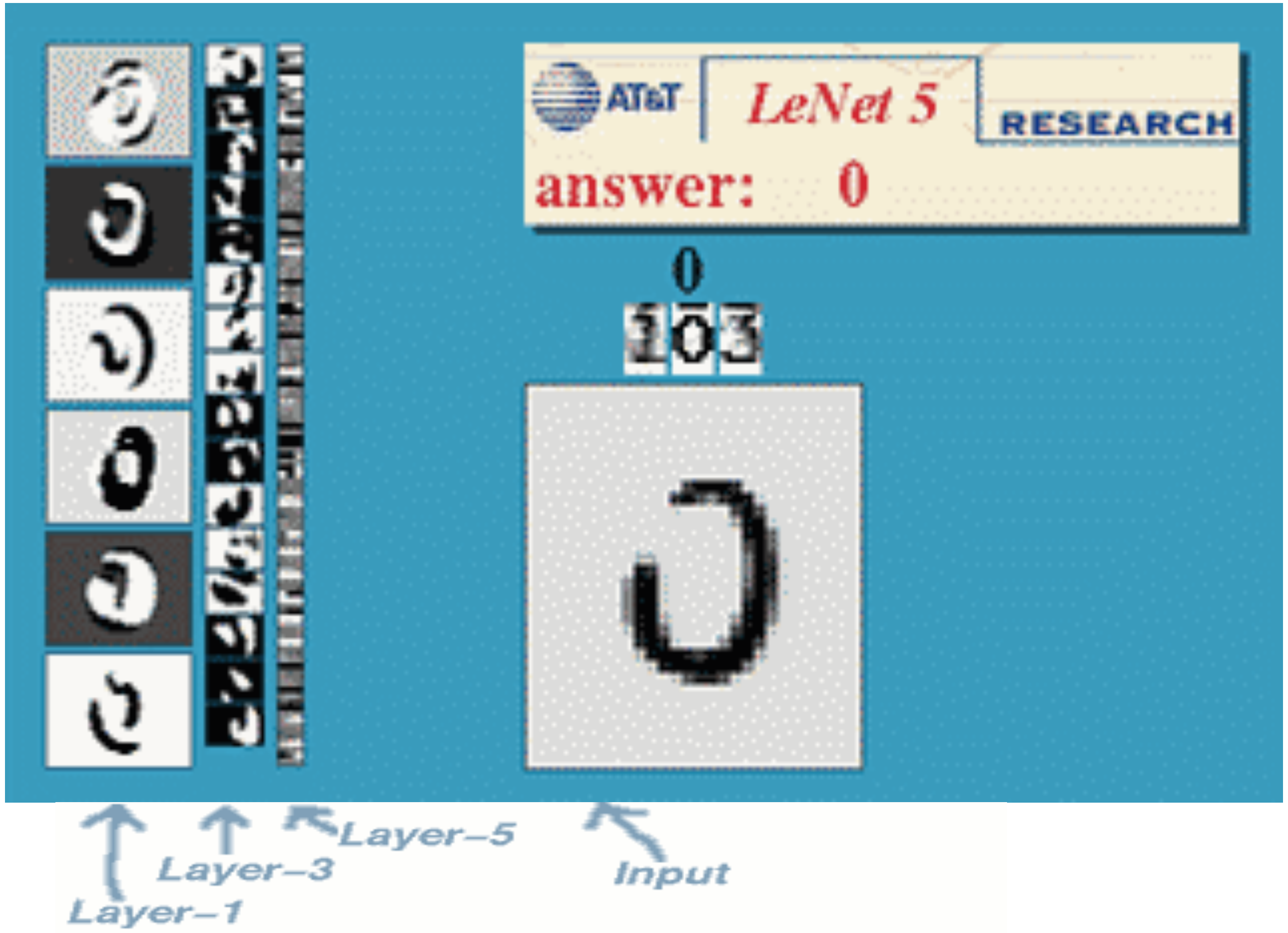
x1	x2	x3	l1	l2
1.0	0.1	0.3	1	0
0.1	1.5	1.2	1	0
1.1	1.1	2.0	0	1
0.2	0.2	0.3	0	1



$$Error\ Function\ (W,b) = \frac{1}{2}[(o1(W,b) - l1)^2 + (o2(W,b) - l2)^2]$$

Calculate the gradient for (W,b), then tune them

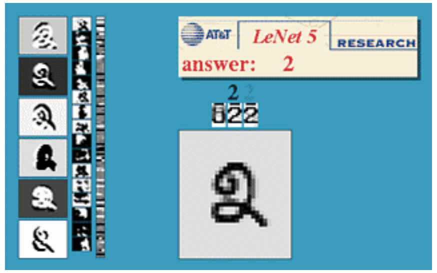
ANN Application：Handwritten character recognition

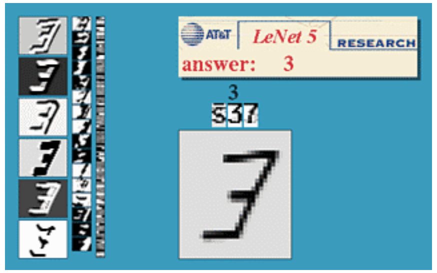


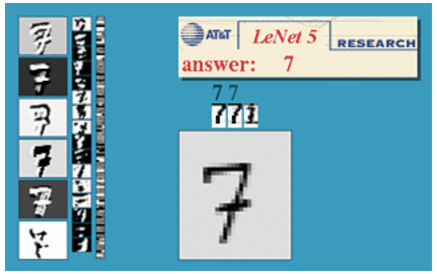
http://yann.lecun.com/exdb/lenet/

Performance of ANN

- Anti-interference ability, such as different sizes, digital distortion







The practice: Handwritten character recognition

- The dataset of this practice is a subset of MNIST

A 28*28 grayscale image:

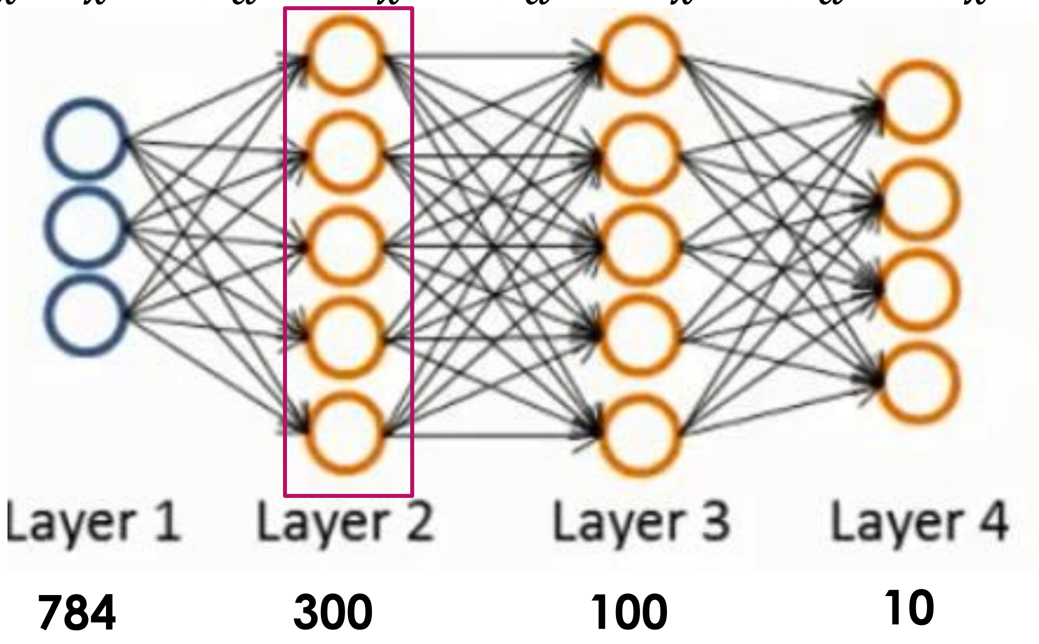
Label:

7

Handwritten character recognition ANN network structure

- Suppose 100 training samples, each sample: 28*28, feature: 784
- Layer 1: Input Layer, size D = 784
- Layer 2: Hidden Layer 1, size M = 300
- Layer 3: Hidden Layer 2, size N = 100
- Layer 4: Output Layer, size K = 10

$$x = x^{(0)} \rightarrow a^{(1)} \rightarrow x^{(1)} \rightarrow a^{(2)} \rightarrow x^{(2)} \rightarrow a^{(3)} \rightarrow x^{(3)} = y$$



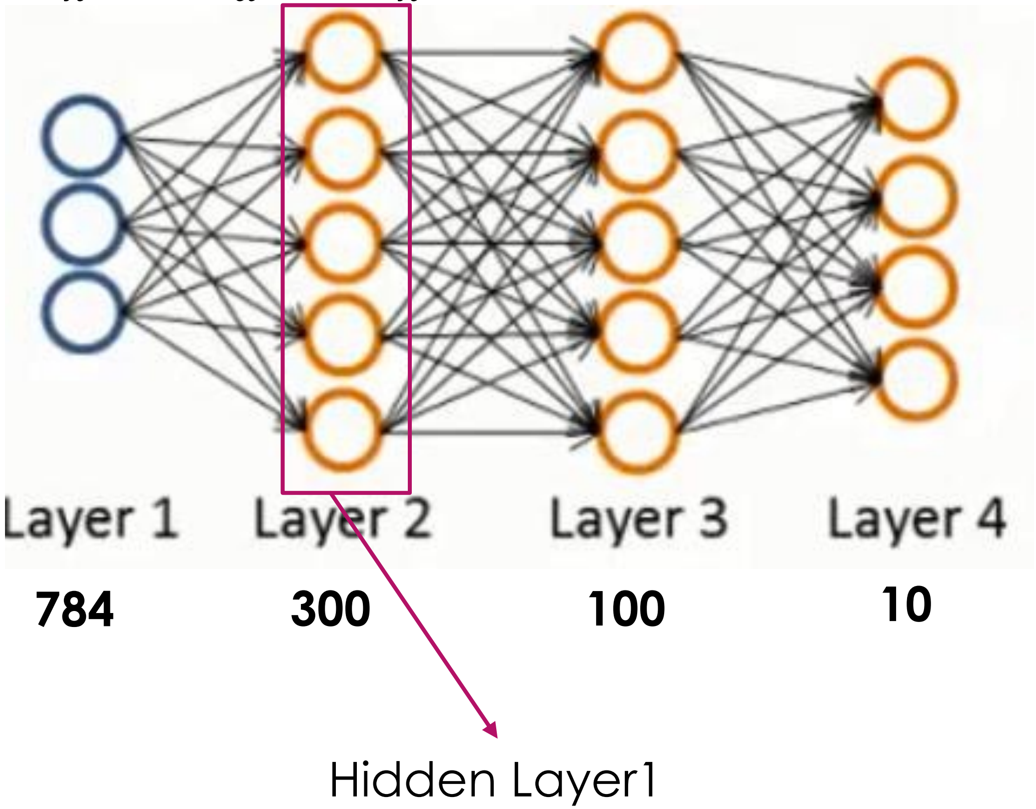
ANN-Input Layer to Hidden Layer 1

- Input: 100(samples number) * 784(feature)
- Input to Hidden Layer 1: Weight Matrix: 784*300, b: 1*300
- Weighted Sum for $j-th$ neurons of hidden layer 1:

$$a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i^{(0)} + b_j^{(1)}$$

- After activation function, output of j-th neuron: $x_j^{(1)} = f(a_j^{(1)})$
- Output of Hidden Layer 1, $x^{(1)}$: 100(samples number)* 300 (feature) matrix
- f: Relu function

$x \rightarrow a^{(1)} \rightarrow x^{(1)}$



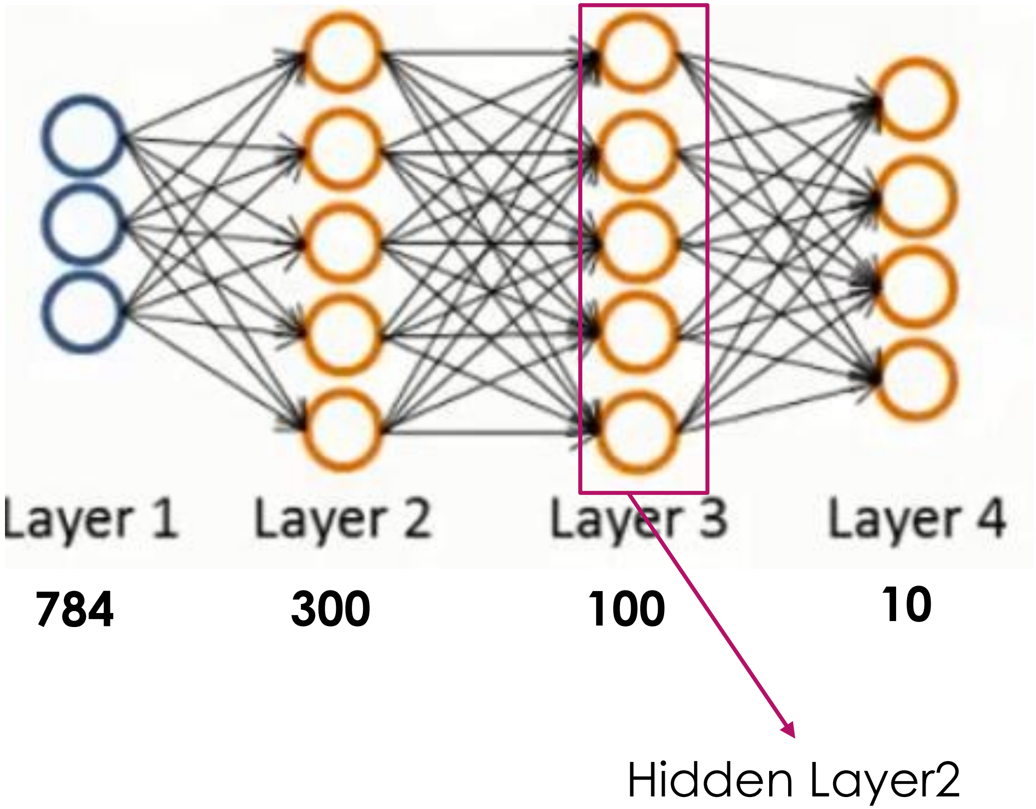
ANN-Hidden Layer 1 to Hidden Layer 2

- For Hidden Layer 2: Input data $x^{(1)}$: 100(samples number)* 300 (feature)
- Hidden Layer 1 to Hidden Layer 2: Weight Matrix: 300*100, b: 1*100
- Weighted Sum for $k - th$ neurons of hidden layer 2:

$$a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)}$$

- After activation function, output of k-th neuron: $x_k^{(2)} = f(a_k^{(2)})$
- Output of Hidden Layer 2, $x^{(2)}$: 100(samples number) *100 (feature) matrix
- f: Relu function

$x^{(1)} \rightarrow \quad a^{(2)} \rightarrow \quad x^{(2)}$



ANN-Hidden Layer 2 to Output Layer

- For ouput layer: Input data $x^{(2)}$: 100 (samples number) * 100 (feature)
- Hidden Layer 2 to Output Layer: Weight Matrix: 100*10, b: 10*1
- Weighted Sum for $l-th$ neurons of output layer:

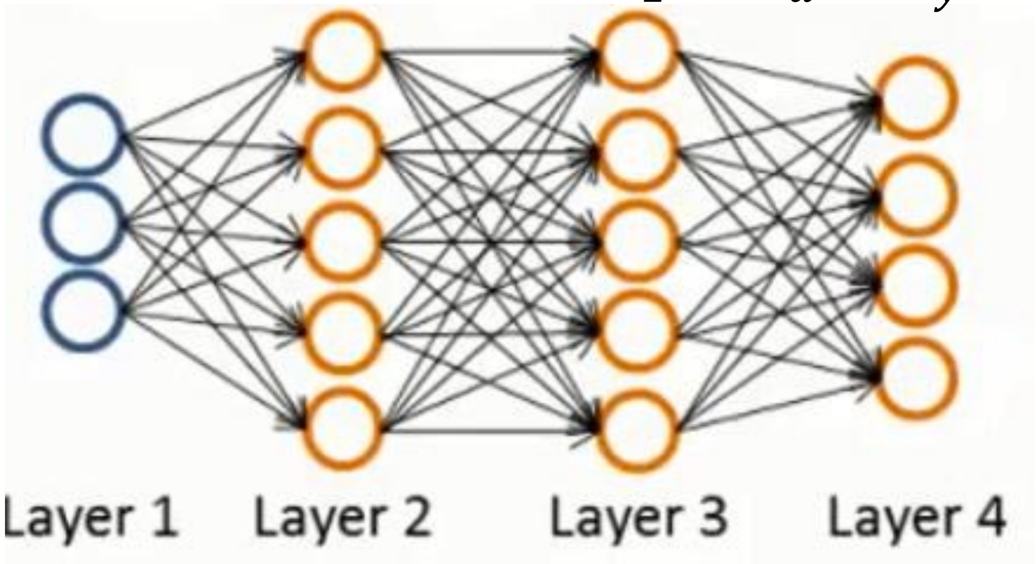
$$a_l^{(3)} = \sum_{k=1}^{N} w_{lk}^{(3)} x_k^{(2)} + b_l^{(3)}$$

- After activation function, output of l-th neuron: $y_l = x_l^{(3)} = \sigma\left(a_l^{(3)}\right)$
- Output of Output Layer, y: 100(samples number) * 10 (feature)

matrix

- σ: softmax function

$z^{(2)} \rightarrow a^{(3)} \rightarrow y$



784 **300** **100** **10**

Training of NN

- W and Threshold values decide the output of NN
- The training is to find appropriate values for W and Threshold so that the output is close to the true value
- Given the structure of NN, the learning process is to tune weight matrix in order to in order to minimize the difference between true value and prediction value.

Loss Function and Gradient Descent

- How to evaluate the difference？ Loss Function
- How to tune weight matrix？ optimization method (e. g. Gradient Descent)

As an Example

- W and b initialization:
  - Xavier: is an initialization scheme for neural networks, make w uniform distributed over $(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}})$

  (m: number of rows of the matrix, n: number of columns of the matrix)
- Loss Function: $E\ (\boldsymbol{y}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{10} -t_k^n ln y_k(x_n, w, b)$

  N is the samples number,
- To avoid overfitting, add Regularization term:

$$E_{total}(\boldsymbol{y}) = E\ (\boldsymbol{y}) + \frac{\lambda}{2}||W||_2^2$$

- Gradient Descent: min f(x) -> x(t+1)=x(t)-η*f'(x(t)) (η： learning rate)

Loss

A 28*28 grayscale image:

Label:

$\boxed{7}$

t:

$$\begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}$$

y:

$$\begin{bmatrix}0.1\\0.1\\0.1\\0.1\\0.1\\0.1\\0.1\\0.1\\0.1\end{bmatrix}$$

$E\ (W, b)$= -log0.1

Gradient Descent

- $\frac{\partial E_{total}(W^{(t)},b^{(t)})}{\partial W^{(t)}}$

$=\frac{\partial(\frac{1}{N}\sum_{n=1}^{N} E_n(W^{(t)},b^{(t)})+\frac{\lambda}{2}||W^{(t)}||^2)}{\partial W^{(t)}}$

$=\frac{1}{N}\sum_{n=1}^{N}\frac{\partial E_n(W^{(t)},b^{(t)})}{\partial W^{(t)}}+\lambda W^{(t)}$

- $\frac{\partial E_{total}(W^{(t)},b^{(t)})}{\partial b^{(t)}}=\frac{1}{N}\sum_{n=1}^{N}\frac{\partial E_n(W^{(t)},b^{(t)})}{\partial b^{(t)}}$

Chain Rule Review

- Chain rule: to find the derivative of a composite function
- If $h(x)=f(g(x))$， then $h'(x)=f'(g(x))g'(x)$
- E.g.: $f(x)=2x+2$， $g(x)=3x+3$， $g(f(x))$ is a composite function, $g'(f(x))=6$

One Inference Process

$$x_j^{(1)} = f(a_j^{(1)})$$

$$x_k^{(2)} = f(a_k^{(2)})$$

$$y_l = x_l^{(3)} = f\left(a_l^{(3)}\right)$$

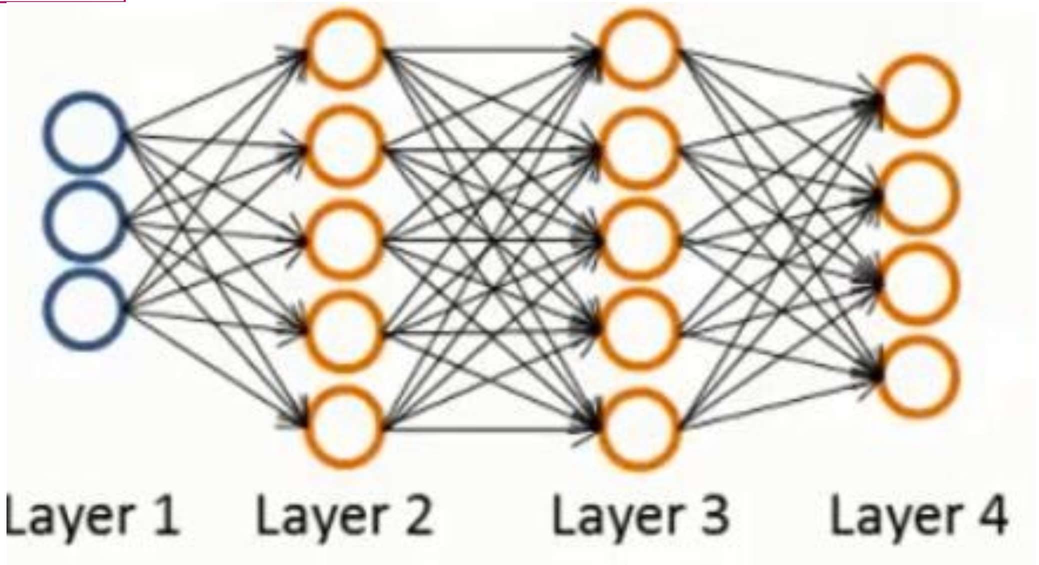
$$x = x^{(0)} \rightarrow a^{(1)} \rightarrow x^{(1)} \rightarrow a^{(2)} \rightarrow x^{(2)} \rightarrow a^{(3)} \rightarrow x^{(3)} = y$$

$$a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i^{(0)} + b_j^{(1)}$$

$$a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)}$$

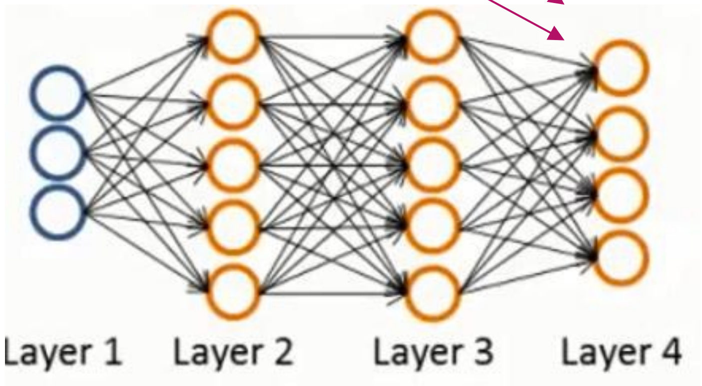
$$a_l^{(3)} = \sum_{k=1}^{N} w_{lk}^{(3)} x_k^{(2)} + b_l^{(3)}$$

Superscript of w：layer index



Derivative of E on $w^{(3)},\ b^{(3)}$

- $\dfrac{\partial E}{\partial W^{(3)}} = \dfrac{\partial E\,(y)}{\partial y}\dfrac{\partial y(a^{(3)})}{\partial a^{(3)}}\dfrac{\partial a^{(3)})}{\partial w^{(3)}}$
- $\dfrac{\partial E}{\partial b^{(3)}} = \dfrac{\partial E\,(y)}{\partial y}\dfrac{\partial y(a^{(3)})}{\partial a^{(3)}}\dfrac{\partial a^{(3)})}{\partial b^{(3)}}$



$$\boxed{x_j^{(1)} = f(a_j^{(1)})} \quad \boxed{x_k^{(2)} = f(a_k^{(2)})} \quad \boxed{y_l = x_l^{(3)} = f\left(a_l^{(3)}\right)}$$

$$x = x^{(0)} \to a^{(1)} \to x^{(1)} \to a^{(2)} \to x^{(2)} \to a^{(3)} \to x^{(3)} = y$$

$$\boxed{a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i^{(0)} + b_j^{(1)}}$$

$$\boxed{a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)}}$$

$$\boxed{a_l^{(3)} = \sum_{k=1}^{N} w_{lk}^{(3)} x_k^{(2)} + b_l^{(3)}}$$

Derivative of E on $w^{(3)},\ b^{(3)}$

- $\frac{\partial E}{\partial W^{(3)}} = \boxed{\frac{\partial E(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}}\frac{\partial a^{(3)})}{\partial w^{(3)}}$
- $\frac{\partial E}{\partial b^{(3)}} = \boxed{\frac{\partial E(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}}\frac{\partial a^{(3)})}{\partial b^{(3)}}$

$$\boxed{\text{Let } \delta^{(3)} = \frac{\partial E}{\partial a^{(3)}} = \frac{\partial E(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}}$$

$$\boxed{x_j^{(1)} = f(a_j^{(1)})} \quad \boxed{x_k^{(2)} = f(a_k^{(2)})} \quad \boxed{y_l = x_l^{(3)} = f\left(a_l^{(3)}\right)}$$

$$x = x^{(0)} \to a^{(1)} \to x^{(1)} \to a^{(2)} \to x^{(2)} \to a^{(3)} \to x^{(3)} = y$$

$$\boxed{a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i^{(0)} + b_j^{(1)}}$$

$$\boxed{a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)}}$$

$$\boxed{a_l^{(3)} = \sum_{k=1}^{N} w_{lk}^{(3)} x_k^{(2)} + b_l^{(3)}}$$

$\delta^{(3)}$ Step 1

- $\delta^{(3)} = \frac{\partial E\,(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}} = \frac{\partial \sum_{k=1}^{10} -t_k^n ln y_k}{\partial y}\boxed{\frac{\partial(\frac{e^{a_j^{(3)}}}{\sum_{k=1}^{10} e^{a_k^{(3)}}})}{\partial a^{(3)}}}$
- Suppose when k=j t_j^n is 1,
- If k=j:

$$\delta_j^{(3)} = \frac{\partial(-ln y_j)}{\partial y_j}\frac{\partial(\frac{e^{a_j^{(3)}}}{\sum_{k=1}^{10} e^{a_k^{(3)}}})}{\partial a_j^{(3)}}$$

$$=(-\frac{1}{y_j})(\frac{e^{a_j^{(3)}}\sum_{k=1}^{10} e^{a_k^{(3)}} - e^{a_j^{(3)}}e^{a_j^{(3)}}}{(\sum_{k=1}^{10} e^{a_k^{(3)}})^2})$$

$$=(-\frac{1}{y_j})(\frac{e^{a_j^{(3)}}}{\sum_{k=1}^{10} e^{a_k^{(3)}}} - \frac{e^{a_j^{(3)}}e^{a_j^{(3)}}}{(\sum_{k=1}^{10} e^{a_k^{(3)}})^2})$$

$$=(-\frac{1}{y_j})(y_j - y_j^2) = y_j - 1$$

Related derivation formula:

$$y = lnx \qquad y' = \frac{1}{x}$$

$$y = e^x \qquad y' = e^x$$

$$y = x^n \qquad y' = nx^{n-1}$$

$$\left(\frac{v(x)}{u(x)}\right)' = \frac{u(x)v'(x) - u'(x)v(x)}{(u(x))^2}$$

$\delta^{(3)}$ Step 2

- If $k \neq j$:

$$\delta_i^{(3)} = \frac{\partial(-ln y_j)}{\partial y_j} \frac{\partial\left(\frac{e^{a_j^{(3)}}}{\sum_{k=1}^{10} e^{a_k^{(3)}}}\right)}{\partial a_i^{(3)}}$$

$$=(-\frac{1}{y_j})\left(\frac{0*\sum_{k=1}^{10} e^{a_k^{(3)}} - e^{a_j^{(3)}} e^{a_i^{(3)}}}{(\sum_{k=1}^{10} e^{a_k^{(3)}})^2}\right)$$

$$=(-\frac{1}{y_j})\left(-\frac{e^{a_j^{(3)}} e^{a_i^{(3)}}}{(\sum_{k=1}^{10} e^{a_k^{(3)}})^2}\right)$$

$$=(-\frac{1}{y_j})(-y_j y_i)=y_i - 0$$

$\delta^{(3)}$

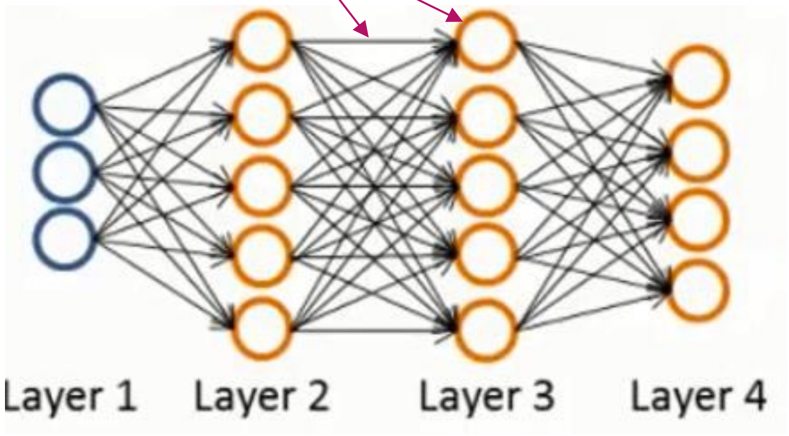
- Summary:
- $\delta^{(3)} = y - t$

Derivative of E on $w^{(3)}$, $b^{(3)}$

- $\frac{\partial E}{\partial w_{ij}^{(3)}} = \frac{\partial E}{\partial a_i^{(3)}} \frac{\partial (a_i^{(3)})}{\partial w_{ij}^{(3)}} = \frac{\partial E}{\partial a_i^{(3)}} \frac{\partial (\sum_{k=1}^{N} w_{ik}^{(3)} x_k^{(2)} + b_i^{(2)})}{\partial w_{ij}^{(3)}} = \delta_i^{(3)} x_j^{(2)}$
  - $\frac{\partial E_{\text{total}}(W^{(3)}, b^{(3)})}{\partial W^{(3)}} = \frac{1}{N} \delta^{(3)} (x^{(2)})^T + \lambda W^{(3)}$
- $\frac{\partial E}{\partial b_i^{(3)}} = \frac{\partial E}{\partial a_i^{(3)}} \frac{\partial (a_i^{(3)})}{\partial b_i^{(3)}} = \frac{\partial E}{\partial a_i^{(3)}} \frac{\partial (\sum_{k=1}^{N} w_{ik}^{(3)} x_k^{(2)} + b_i^{(2)})}{\partial b_i^{(3)}} = \delta_i^{(3)} * 1 = \delta_i^{(3)}$

Derivative of E on $w^{(2)},\ b^{(2)}$

- $$\frac{\partial E}{\partial W^{(2)}} = \frac{\partial E\,(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}\frac{\partial a^{(3)})}{\partial x^{(2)}}\frac{\partial x^{(2)})}{\partial a^{(2)}}\frac{\partial a^{(2)})}{\partial W^{(2)}}$$
- $$\frac{\partial E}{\partial b^{(2)}} = \frac{\partial E\,(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}\frac{\partial a^{(3)})}{\partial x^{(2)}}\frac{\partial x^{(2)})}{\partial a^{(2)}}\frac{\partial a^{(2)})}{\partial b^{(2)}}$$



$$x_j^{(1)} = f(a_j^{(1)})$$

$$x_k^{(2)} = f(a_k^{(2)})$$

$$y_l = x_l^{(3)} = f\left(a_l^{(3)}\right)$$

$$x = x^{(0)} \to a^{(1)} \to x^{(1)} \to a^{(2)} \to x^{(2)} \to a^{(3)} \to x^{(3)} = y$$

$$a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i^{(0)} + b_j^{(1)}$$

$$a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)}$$

$$a_l^{(3)} = \sum_{k=1}^{N} w_{lk}^{(3)} x_k^{(2)} + b_l^{(3)}$$

Derivative of E on $w^{(2)}$, $b^{(2)}$

- $$\frac{\partial E}{\partial W^{(2)}} = \boxed{\frac{\partial E\,(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}\frac{\partial a^{(3)})}{\partial x^{(2)}}\frac{\partial x^{(2)})}{\partial a^{(2)}}}\frac{\partial a^{(2)})}{\partial W^{(2)}}$$
- $$\frac{\partial E}{\partial b^{(2)}} = \boxed{\frac{\partial E\,(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}\frac{\partial a^{(3)})}{\partial x^{(2)}}\frac{\partial x^{(2)})}{\partial a^{(2)}}}\frac{\partial a^{(2)})}{\partial b^{(2)}}$$

Let $\delta^{(2)} = \frac{\partial E\,(y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}\frac{\partial a^{(3)})}{\partial x^{(2)}}\frac{\partial x^{(2)})}{\partial a^{(2)}} = \frac{\partial E}{\partial a^{(2)}}$

$$x_j^{(1)} = f(a_j^{(1)}) \qquad x_k^{(2)} = f(a_k^{(2)}) \qquad y_l = x_l^{(3)} = f\left(a_l^{(3)}\right)$$

$$x = x^{(0)} \to a^{(1)} \to x^{(1)} \to a^{(2)} \to x^{(2)} \to a^{(3)} \to x^{(3)} = y$$

$$a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i^{(0)} + b_j^{(1)}$$

$$a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)}$$

$$a_l^{(3)} = \sum_{k=1}^{N} w_{lk}^{(3)} x_k^{(2)} + b_l^{(3)}$$

$\delta^{(2)}$

- $\delta_i^{(2)} = \sum_{j=1}^{L} \delta_j^{(3)} \frac{\partial a_j^{(3)}}{\partial x_i^{(2)}} \frac{\partial x_i^{(2)}}{\partial a_i^{(2)}}$

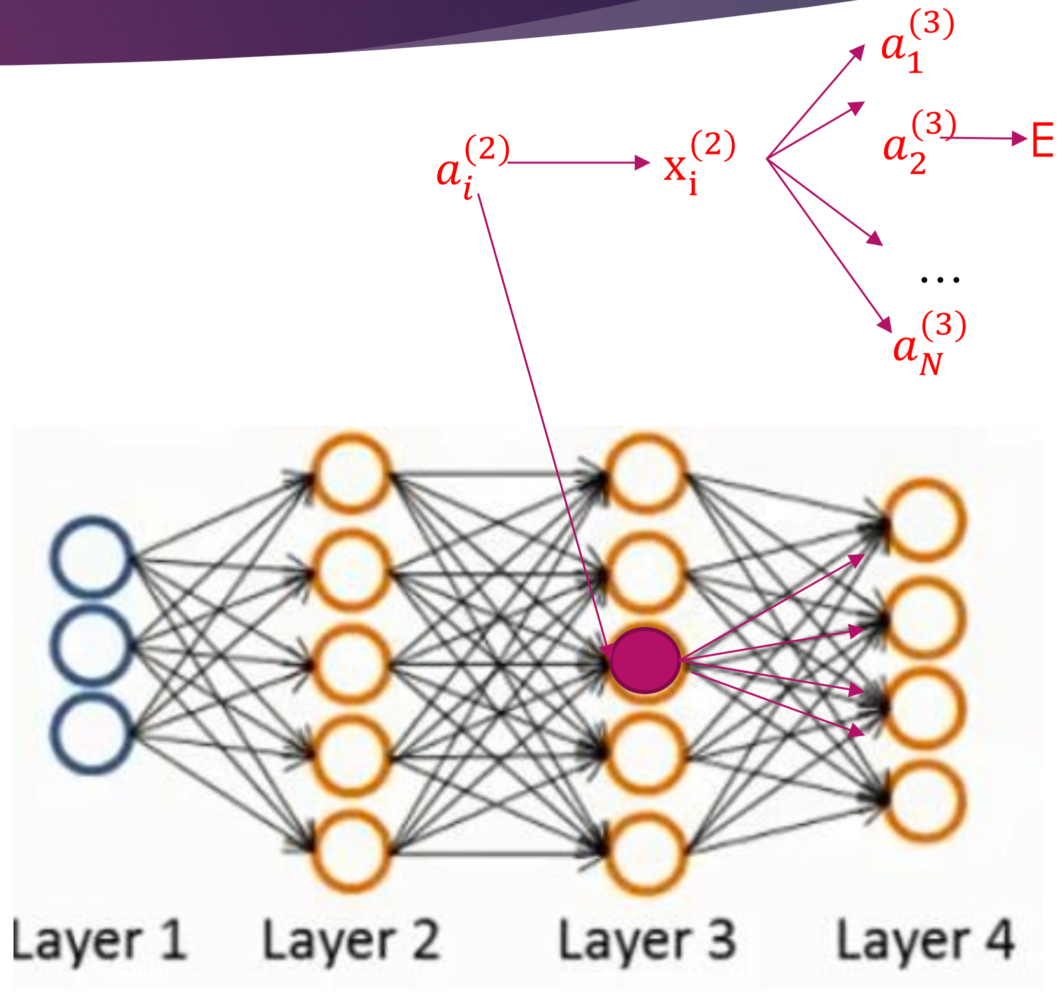
$= \sum_{j=1}^{L} \delta_j^{(3)} \frac{\partial(\sum_{k=1}^{N} w_{jk}^{(3)} x_k + b_k^{(3)})}{\partial x_i^{(2)}} \frac{\partial x_i^{(2)}}{\partial a_i^{(2)}}$

$= \sum_{j=1}^{L} \delta_j^{(3)} w_{ji}^{(3)} f'(x_i^{(2)})$

The matrix is expressed as:

$\delta^{(2)} = \delta^{(3)} (w^{(3)})^T \odot f'(x^{(2)})$

f is relu



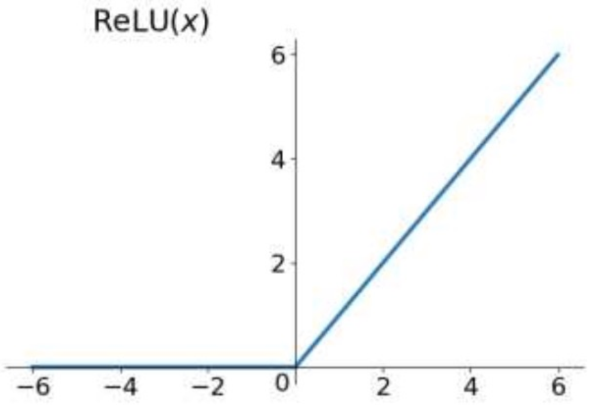
Relu and the derivative of relu

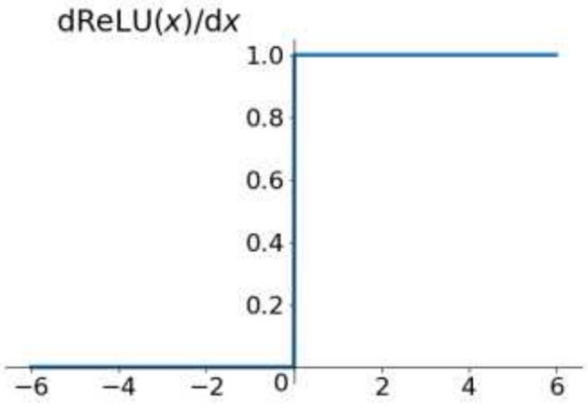
- Relu:

$$f(x) = \max(0, x)$$

- The derivative of relu:

$$f'(x) = \begin{cases} if\ x > 0, f'(x) = 1 \\ else\ f'(x) = 0 \end{cases}$$



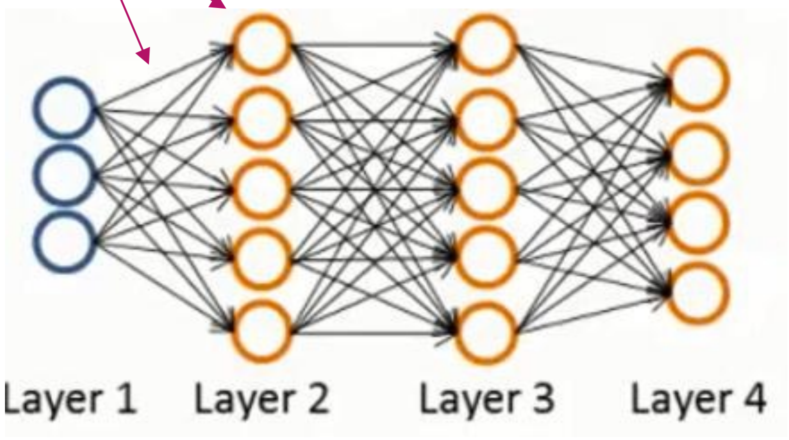


Derivative of E on $w^{(2)}$, $b^{(2)}$

- $\frac{\partial E}{\partial w_{ij}^{(2)}} = \frac{\partial E}{\partial a_i^{(2)}} \frac{\partial (a_i^{(2)})}{\partial w_{ij}^{(2)}} = \delta_i^{(2)} \frac{\partial (\sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)})}{\partial w_{ij}^{(2)}} = \delta_i^{(2)} x_j^{(1)}$
- $\frac{\partial E_{total}}{\partial W^{(2)}} = \frac{1}{N} \delta^{(2)} (x^{(1)})^T + \lambda W^{(2)}$
- $\frac{\partial E}{\partial b_i^{(2)}} = \frac{\partial E}{\partial a_i^{(2)}} \frac{\partial (a_i^{(2)})}{\partial b_i^{(2)}} = \frac{\partial E}{\partial a_i^{(2)}} \frac{\partial (\sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)})}{\partial b_i^{(3)}} = \delta_i^{(2)} * 1 = \delta_i^{(2)}$

Derivative of E on $w^{(1)},\ b^{(1)}$

- $\frac{\partial E}{\partial W^{(1)}} = \frac{\partial E\ (y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}\frac{\partial a^{(3)})}{\partial x^{(2)}}\frac{\partial x^{(2)})}{\partial a^{(2)}}\frac{\partial a^{(2)})}{\partial x^{(1)}}\frac{\partial x^{(1)})}{\partial a^{(1)}}\frac{\partial a^{(1)})}{\partial W^{(1)}}$
- $\frac{\partial E}{\partial b^{(1)}} = \frac{\partial E\ (y)}{\partial y}\frac{\partial y(a^{(3)})}{\partial a^{(3)}}\frac{\partial a^{(3)})}{\partial x^{(2)}}\frac{\partial x^{(2)})}{\partial a^{(2)}}\frac{\partial a^{(2)})}{\partial x^{(1)}}\frac{\partial x^{(1)})}{\partial a^{(1)}}\frac{\partial a^{(1)})}{\partial b^{(1)}}$



$\boxed{x_j^{(1)} = f(a_j^{(1)})}$ $\boxed{x_k^{(2)} = f(a_k^{(2)})}$ $\boxed{y_l = x_l^{(3)} = f\left(a_l^{(3)}\right)}$

$$x = x^{(0)} \to a^{(1)} \to x^{(1)} \to a^{(2)} \to x^{(2)} \to a^{(3)} \to x^{(3)} = y$$

$$\boxed{a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i^{(0)} + b_j^{(1)}}$$

$$\boxed{a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} x_j^{(1)} + b_k^{(2)}}$$

$$\boxed{a_l^{(3)} = \sum_{k=1}^{N} w_{lk}^{(3)} x_k^{(2)} + b_l^{(3)}}$$

Derivative of E on $w^{(1)}, \ b^{(1)}$

- Like $\delta^{(2)}$: $\delta^{(1)} = \delta^{(2)}(w^{(2)})^T \odot f'(x^{(1)})$

  - $\frac{\partial E_{total}}{\partial W^{(1)}} = \frac{1}{N}\delta^{(1)}(x)^T + \lambda\, W^{(1)}$

- $\frac{\partial E}{\partial b^{(1)}} = \delta^{(1)}$

Gradient Descent Process (BP Algorithm)

1. Set values to max_iterations and η
2. Randomly generate $\mathrm{W}^{(0)}$ and $\mathrm{b}^{(0)}$
3. For t = 1 to max_iterations
4. Calculate the derivatives: $\frac{\partial E_{\text{total}}\left(W^{(t)},\mathrm{b}^{(t)}\right)}{\partial W^{(t)}}$ and $\frac{\partial E_{\text{total}}\left(W^{(t)},\mathrm{b}^{(t)}\right)}{\partial \mathrm{b}^{(t)}}$
5. Update $\mathrm{W}^{(t+1)} = \mathrm{W}^{(t)} - \eta \frac{\partial E_{\text{total}}\left(W^{(t)},\mathrm{b}^{(t)}\right)}{\partial W^{(t)}}, B^{(t+1)} = b^{(t)} - \eta \frac{\partial E_{\text{total}}\left(W^{(t)},\mathrm{b}^{(t)}\right)}{\partial \mathrm{b}^{(t)}}$
6. EndFor

Note:

E can be loss for a single sample, a batch of randomly selected sample, or all training samples.