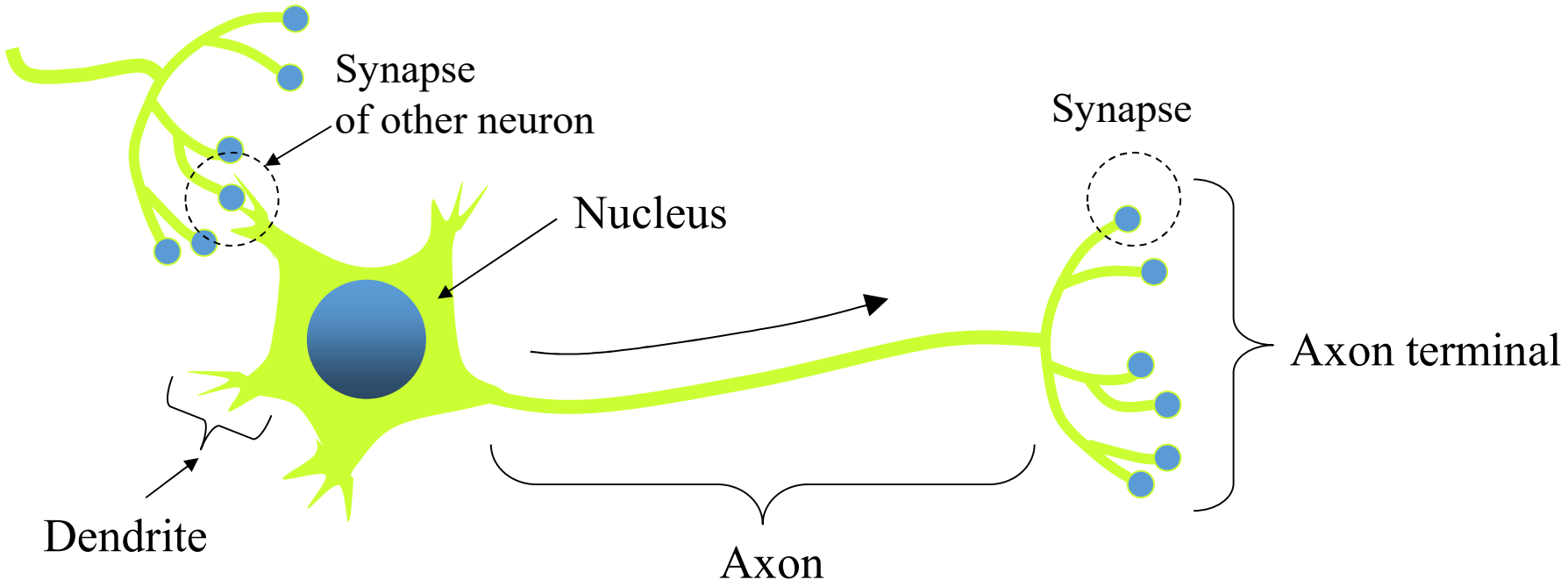
Topics in data engineering

Artificial neural network

Masaomi Kimura

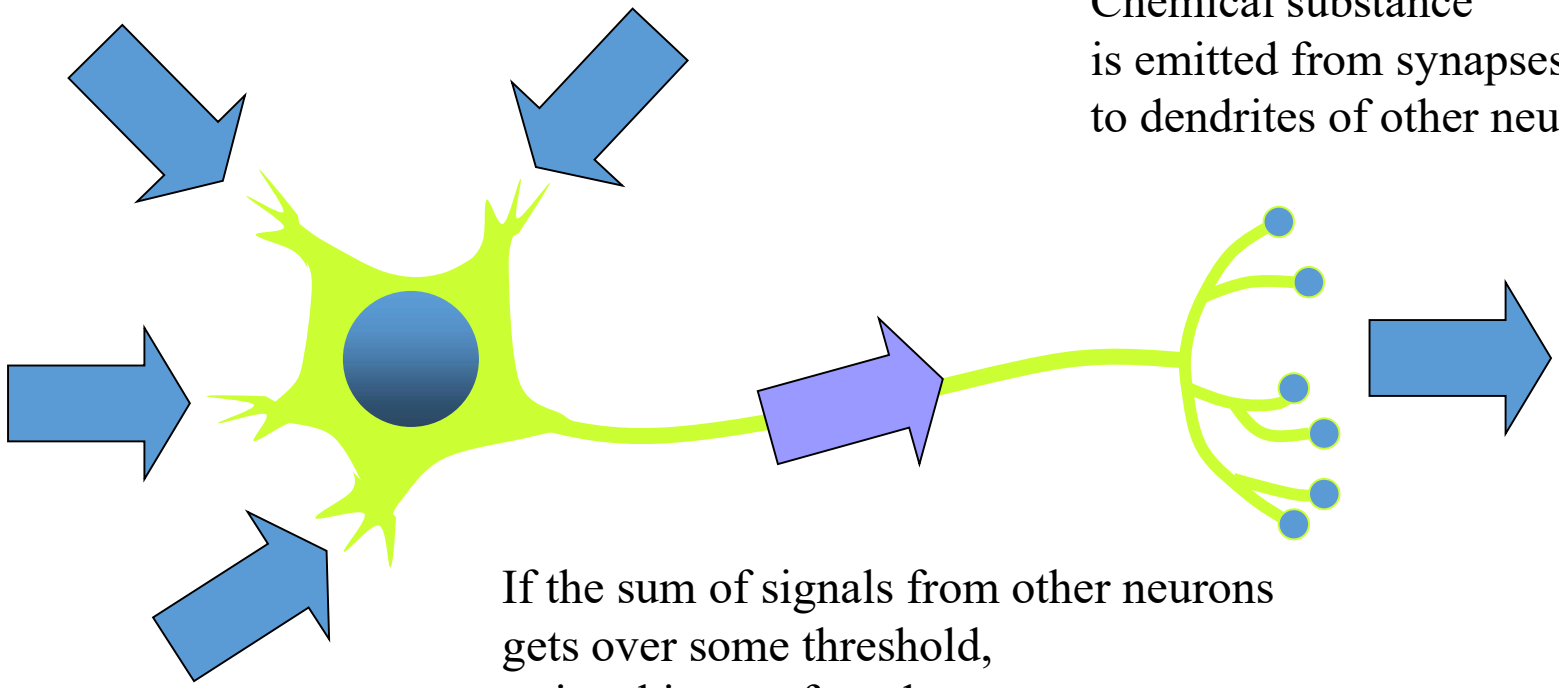
Neuron

- cells in nervous system
  - contains dendrites, a nucleus, an axon
  - Synapses at an axon terminal send a signal to dendrites in other neurons



Signal transfer

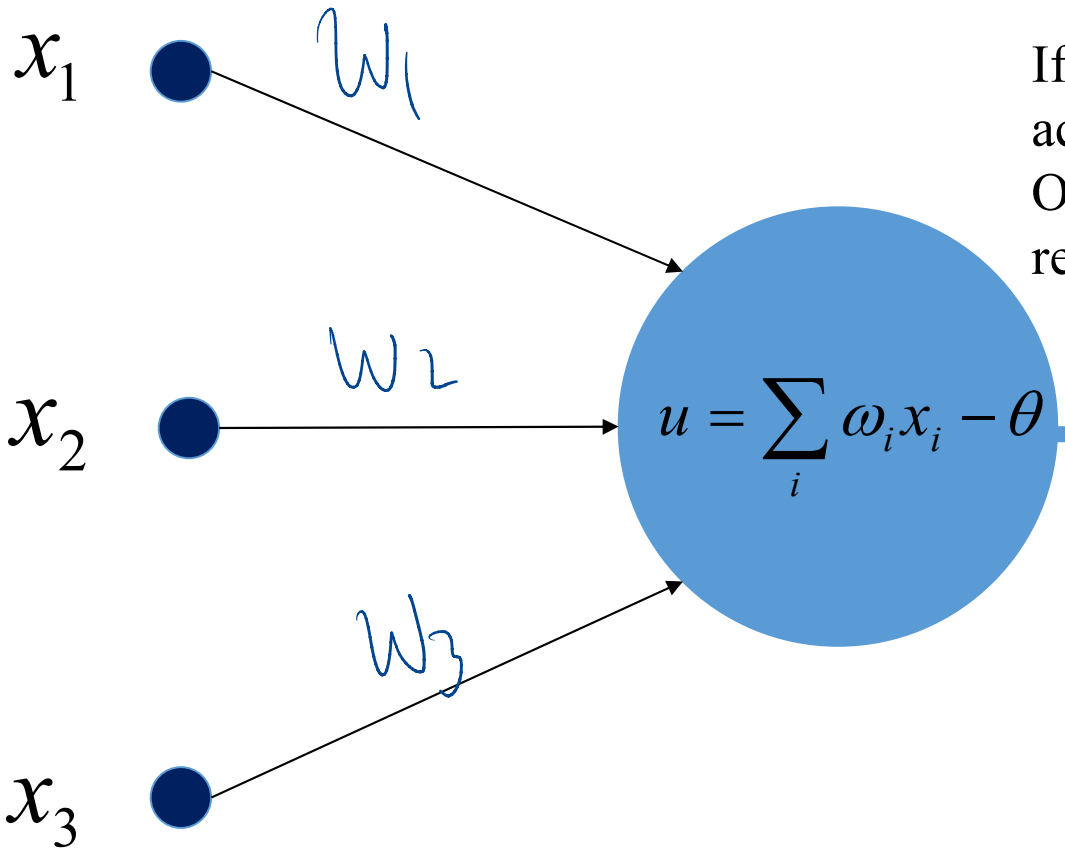
Signals from synapses of other neurons



Chemical substance
is emitted from synapses
to dendrites of other neurons

If the sum of signals from other neurons
gets over some threshold,
a signal is transferred to an axon.

Perceptron



If u is positive, the neuron is activated and returns 1 as output. Otherwise, it is not activated and returns 0.

$$z = f(u)$$

θ is a threshold, f(u) is an activation function satisfying next conditions

$$\lim_{u \to -\infty} f(u) = 0,$$

$$\lim_{u \to \infty} f(u) = 1.$$

Activation function *f(u)*

- Step function

$$f(u) = \begin{cases} 0 & (u < 0) \\ 1 & (u \geq 0) \end{cases}$$

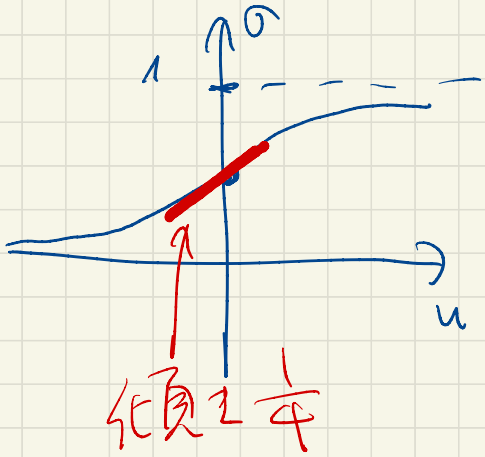
- Sigmoid function

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

- Hyperbolic tangent function

$$\tanh(u) = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}}$$

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



$$\frac{d\sigma}{du} = \frac{d}{du}\left(\frac{1}{1+e^{-u}}\right)$$

$$= \frac{d}{du}\underbrace{\left(1+e^{-u}\right)^{-1}}_{v}$$

$$= \frac{d\sigma}{dv}\frac{dv}{du}$$

$$= \left(\frac{d}{dv}v^{-1}\right)\left(\frac{d}{du}\left(1+e^{-u}\right)\right)$$

$$= -v^{-2} \times \left(-e^{-u}\right)$$

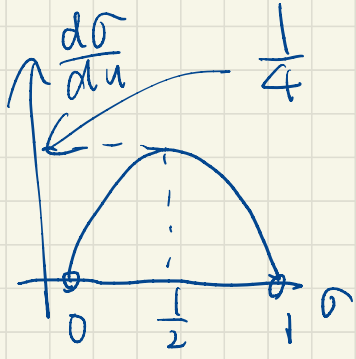
$$= \left(\frac{1}{1+e^{-u}}\right)^2 e^{-u}$$

$$= \frac{1}{1+e^{-u}}\,\frac{e^{-u}}{1+e^{-u}}$$

$$= \sigma(u)\left(1-\sigma(u)\right)$$

$$\left(1-\frac{1}{1+e^{-u}}\right)$$

$$0 < \sigma(u) < 1$$



Exercise

- Let's find the behavior of a sigmoid function $\sigma(x)$.
  - If we let x get larger ($x \to \infty$), explain the behavior of $\sigma(x)$.
  - If we let x get smaller ($x \to -\infty$), explain the behavior of $\sigma(x)$.
  - Write a graph of $\sigma(x)$ in the range $-5 < x < 5$.
- Calculate the derivative$\sigma'(x)$
  - Find the range of$\sigma'(x)$

Exercise

- Let the function

$$z(x) = \sigma(ax - 4) = \frac{1}{1+e^{-ax+4}}.$$

- Draw graphs of $f(x)$ for
  - $a = 1$
  - $a = 2$
  - $a = 4$

  and compare them.

Training of perceptrons

- A training set of an input vector$\{x_i\}$ and its expected output value z_i^* is used to determine weights $\{\omega_k\}$.
- adjust the weights $\{\omega_k\}$ in order to minimize the mean squared error R based on Gradient descent
  (z_i is an output for the an input vector x_i, z_i^* is a correct output, ε is a constant.)

$$R = \frac{1}{2}\sum_i (z_i - z_i^*)^2$$

$$\delta\omega_i = -\varepsilon\frac{\partial R}{\partial \omega_i} = \varepsilon\sum_k (z_k^* - z_k)\frac{\partial z_k}{\partial \omega_i} = \varepsilon\sum_k (z_k^* - z_k)x_k^i\frac{\partial f(u_k)}{\partial u}$$

$$z = \sigma(w \cdot x - \theta) \quad \text{-- perceptron}$$

x	$\tilde{z}$
x_1	$\tilde{z}_1$
x_2	$\tilde{z}_2$
$\vdots$	$\vdots$
x_n	$\tilde{z}_n$

$$L = \frac{1}{2}\sum_i \left(\tilde{z}_i - \sigma(w \cdot x_i - \theta)\right)^2$$

$$\Downarrow$$

$$w' = w - \varepsilon \frac{\partial L}{\partial w}$$

$$\theta' = \theta - \varepsilon' \frac{\partial L}{\partial \theta}$$

$$w \cdot x - \theta = w_1 x_1 + w_2 x_2 + \cdots + \theta(-1)$$

$$= (w_1\ w_2 \cdots\ \theta)\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ -1 \end{pmatrix}$$

$$= \hat{w} \cdot \hat{x}$$

$$L = \frac{1}{2} \sum_i \left(z_i - \sigma(w x_i) \right)^2$$

$$\frac{\partial L}{\partial w} = \sum_i \left(z_i - \sigma(w x_i) \right) \left(- \frac{\partial}{\partial w} \sigma(w x_i) \right)$$

$$w' = w - \epsilon \frac{\partial L}{\partial w} \qquad = \sum_i \left(\sigma(w x_i) - z_i \right) \sigma(w x_i) \left(1 - \sigma(w x_i) \right) \times x_i$$

Exercise

- Let $z(x) = \frac{1}{1+e^{-wx+b}}$ and $R(w) = \frac{1}{2}\left(z(2) - \frac{1}{3}\right)^2$.
  - Calculate $\frac{\partial}{\partial w} z(2)$
  - Calculate $\frac{\partial}{\partial w} R(w)$

Exercise

- Let $w' = w - \epsilon \frac{\partial}{\partial w} R(w)$, where ϵ is a small and positive constant.
- Which is larger, $R(w)$ or $R(w')$?
  - Explain why?

Exercise

- Let $z(x) = \sigma(wx - b)$.
- Train the perceptron with the training data:

X	Z*
1.0	0.1
2.0	0.9

- Draw a graph of the perceptron.

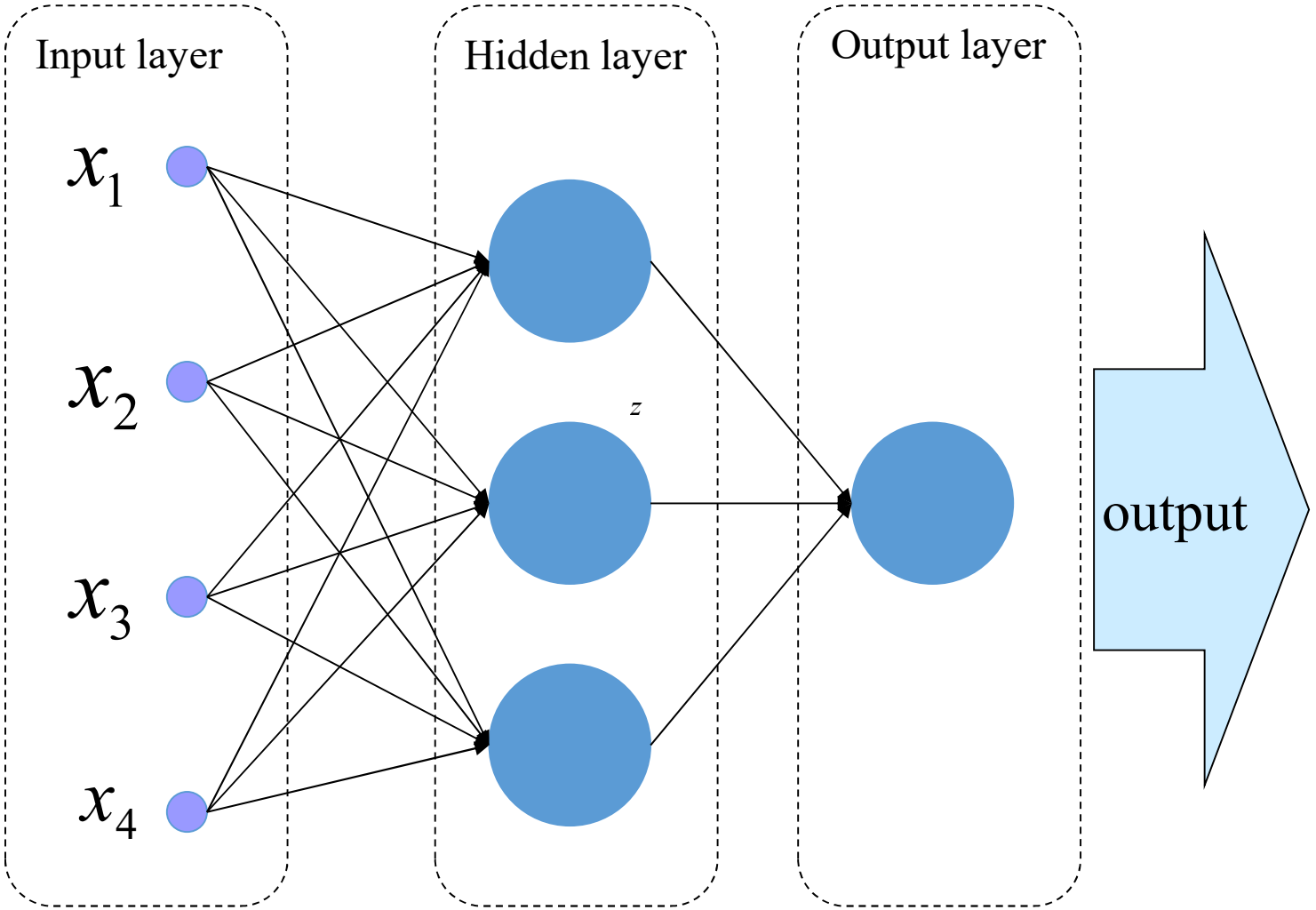
Exercise

- Let $z(x_1, x_2) = \sigma(w_1 x_1 + w_2 x_2 - b)$.
- Discuss whether we can make a perceptron which satisfies the following table:

x_1	x_2	z
0.0	0.0	0.0
1.0	0.0	1.0
0.0	1.0	1.0
1.0	1.0	0.0

- This is called XOR problem.

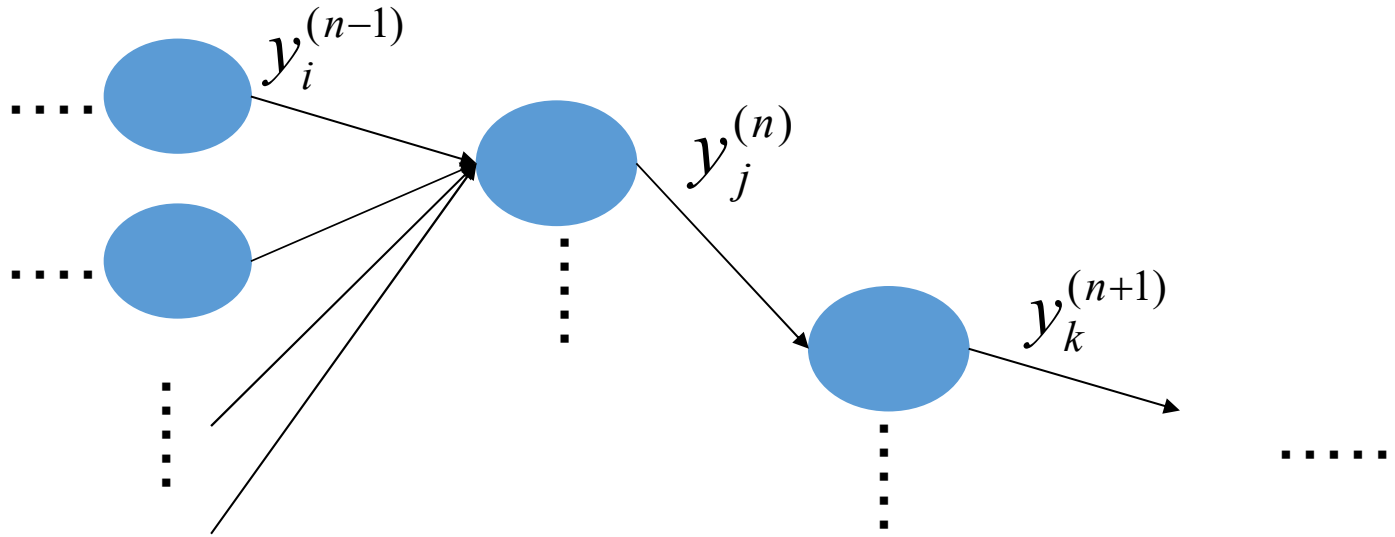
Multi layer perceptron



“Signal” propagation in NN

$$y_j^{(n)} = f\left(\sum_i \omega_{ji}^{(n)} y_i^{(n-1)} - \theta_j^{(n)}\right)$$

$$y_k^{(n+1)} = f\left(\sum_k \omega_{kj}^{(n+1)} y_j^{(n)} - \theta_k^{(n+1)}\right)$$



(n-1)th layer nth layer (n+1)th layer

Exercise

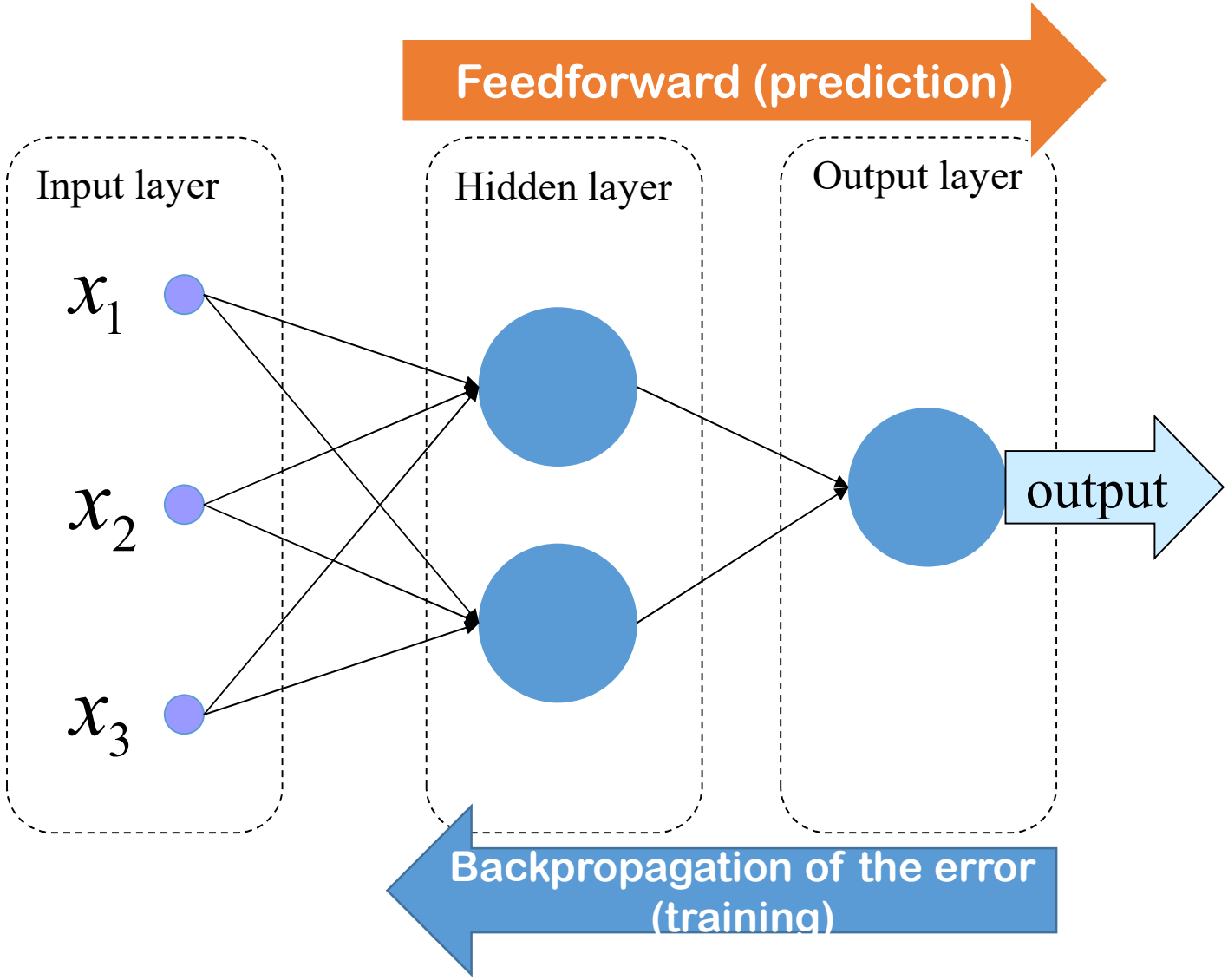
- Let

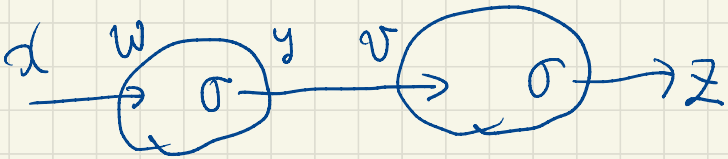
$$z(x_1, x_2) = \sigma\left(10\sigma\left(x_1 + x_2 - \frac{1}{2}\right) - 10\sigma\left(x_1 + x_2 - \frac{3}{2}\right)\right).$$

- Draw a graph of this function.
- Show this approximately solve XOR problem.

Gradient vanishing problem

Feedforward and backpropagation





$x_1 \quad \tilde{z}_1$
$x_2 \quad \tilde{z}_2$
$\vdots \quad \vdots$
$x_n \quad \tilde{z}_n$

$$L = \frac{1}{2}\sum_i \left(\tilde{z}_i - z_i\right)^2$$

$$\Downarrow = \frac{1}{2}\sum_i \left(\tilde{z}_i - \sigma(v\sigma(wx_i))\right)^2$$

$$z = \sigma(vy) \qquad y = \sigma(wx)$$

$$z_i = \sigma(v\sigma(wx_i))$$

$$\frac{\partial L}{\partial v} = \sum_i \Big[\left(\sigma(v\sigma(wx_i)) - \tilde{z}_i\right) \times \sigma(v\sigma(wx_i))\left(1 - \sigma(v\sigma(wx_i))\right) \cdot x\,\sigma(wx_i) \Big]$$

$$= \sum_i \left(z_i - \tilde{z}_i\right) z_i (1 - z_i)\, y_i$$

$$v' = v - \varepsilon \frac{\partial L}{\partial v}$$

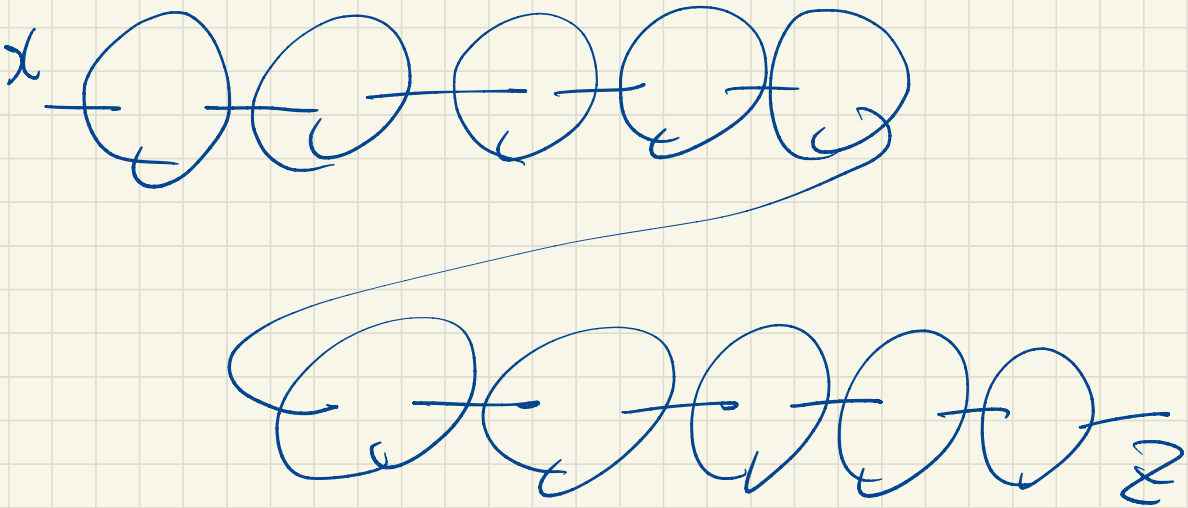
$$\frac{\partial L}{\partial w} = \frac{\partial}{\partial w} \frac{1}{2} \sum_i \left\{ \sigma\left(v\,\sigma(w x_i)\right) - \tilde{z}_i \right\}^2$$

$$= \sum_i \left\{ \left(\sigma(v\sigma(wx_i)) - \tilde{z}_i \right) \times \frac{\partial \sigma(v\sigma(wx_i))}{\partial (v\sigma(wx_i))} \cdot \frac{\partial (v\sigma(wx_i))}{\partial wx_i} \frac{\partial wx_i}{\partial w} \right\}$$

$$= \sum_i (z_i - \tilde{z}_i)\, \sigma(vy_i)(1-\sigma(vy_i)) \times v\,\sigma(wx_i)(1-\sigma(wx_i))\, x_i$$

$$= \sum_i (z_i - \tilde{z}_i)\, z_i(1-z_i) \times v\, y_i(1-y_i)\, x_i$$

$$\frac{\partial L}{\partial v} = \sum_i (z_i - \tilde{z}_i)\, z_i(1-z_i)\, y_i$$



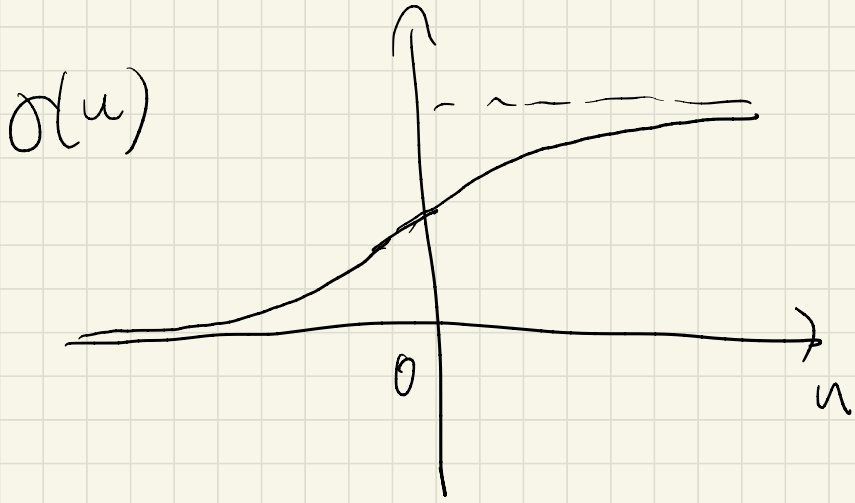
$$\frac{1}{4} \times \cdots \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4}$$

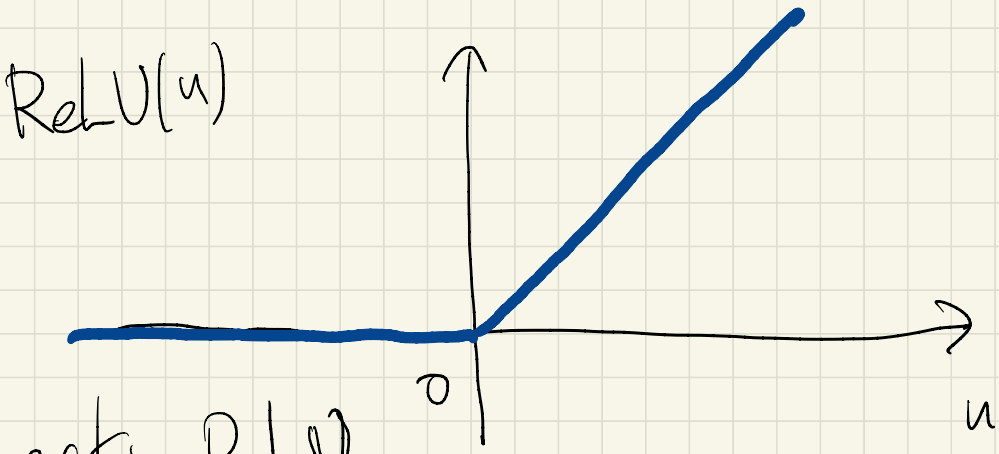
$$\left(\frac{1}{4}\right)^{10} = \left(\frac{1}{2}\right)^{20} = \frac{1}{2^{20}}$$

$$2^{10} = 1024 \approx 10^3$$

$$2^{20} \approx (10^3)^2 = 10^6$$

$$\approx \frac{1}{10^6}$$





Leaky ReLU

Back propagation

- adjusts the weights $\{\omega_k\}$ in order to minimize the mean squared error R based on Gradient descent

$$R = \frac{1}{2}\sum_k (z_k - z_k^*)^2$$

$$y_j^{(n)} = f(u_j^{(n)})$$

$$u_j^{(n)} = \sum_i \omega_{ji}^{(n)} y_i^{(n-1)} - \theta_j^{(n)}$$

$$\omega'^{(n)}_{ji} = \omega_{ji}^{(n)} - \varepsilon \frac{\partial R}{\partial \omega_{ji}^{(n)}} = \omega_{ji}^{(n)} - \varepsilon \frac{\partial R}{\partial y_j^{(n)}} \frac{\partial y_j^{(n)}}{\partial u_j^{(n)}} \frac{\partial u_j^{(n)}}{\partial \omega_{ji}^{(n)}} = \omega_{ji}^{(n)} - \varepsilon O_j^{(n)} y_i^{(n-1)}$$

$$O_j^{(n)} = \frac{\partial R}{\partial y_j^{(n)}} \frac{\partial y_j^{(n)}}{\partial u_j^{(n)}} = \frac{\partial R}{\partial y_j^{(n)}} f(u_j^{(n)})(1 - f(u_j^{(n)}))$$

Back propagation (cont'd)

$$O_i^{(n-1)} = \sum_j \frac{\partial R}{\partial y_j^{(n)}} \frac{\partial y_j^{(n)}}{\partial y_i^{(n-1)}} \frac{\partial y_i^{(n-1)}}{\partial u_i^{(n-1)}}$$

$$= \sum_j \frac{\partial R}{\partial y_j^{(n)}} \frac{\partial y_j^{(n)}}{\partial u_j^{(n)}} \frac{\partial u_j^{(n)}}{\partial y_i^{(n-1)}} \frac{\partial y_i^{(n-1)}}{\partial u_i^{(n-1)}}$$

$$= \sum_j O_j^{(n)} \omega_{ji}^{(n)} f(u_i^{(n-1)}) \left\{1 - f(u_i^{(n-1)})\right\}$$

where $y_i^{(n-1)} = f(u_i^{(n-1)})$ and $u_j^{(n)} = \sum_i \omega_{ji}^{(n)} y_i^{(n-1)} - \theta_j^{(n)}$

Exercise

- Let $z(x) = \sigma(v\,\sigma(wx - b) - d)$ and imagine we get training data (x_i, z_i) for $\mathrm{i} = 1,2,\cdots n$. Define a quadratic loss function as

$$R(v, w, b, d) = \frac{1}{2}\sum_i (z(x_i) - z_i)^2,$$

- Calculate
  - $\frac{\partial}{\partial v} z(x_i)$
  - $\frac{\partial}{\partial w} z(x_i)$ and express it with $\frac{\partial}{\partial v} z(x_i)$
  - $\frac{\partial}{\partial w} R$ and express it with $\frac{\partial}{\partial v} R$

  Hint: Note that x_i and z_i should be dealt as a constant.

Exercise

- Update rules (gradient descent) of w and v are

$$w' = w - \epsilon \frac{\partial}{\partial w} R$$

$$v' = v - \epsilon \frac{\partial}{\partial v} R$$

- Which should be calculated first to reduce calculation time?

Input data

- Numerical data
  - are usually standardized as:

$$z = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

- Categorical data
  - are converted to a set of flags:

$$\text{Male} \rightarrow (1,0),\ \text{Female} \rightarrow (0,1),\ \text{unknown} \rightarrow (0,0)$$

Examples

Estimation of fellowship grants

Organization	# of Division	# of Researchers	Total grants (actual)
団体A	29	3409	246374
団体B	499	15556	230978
団体C	73	17760	1020195
団体D	1041	133989	1367747
団体E	227	22266	188409
団体F	1961	134892	1717810

Normalization

$$z = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Organization	# of Division	# of Researchers	Total grants (actual)
団体A	0.000	0.000	0.038
団体B	0.243	0.092	0.028
団体C	0.023	0.109	0.544
団体D	0.524	0.993	0.771
団体E	0.102	0.143	0.000
団体F	1.000	1.000	1.000

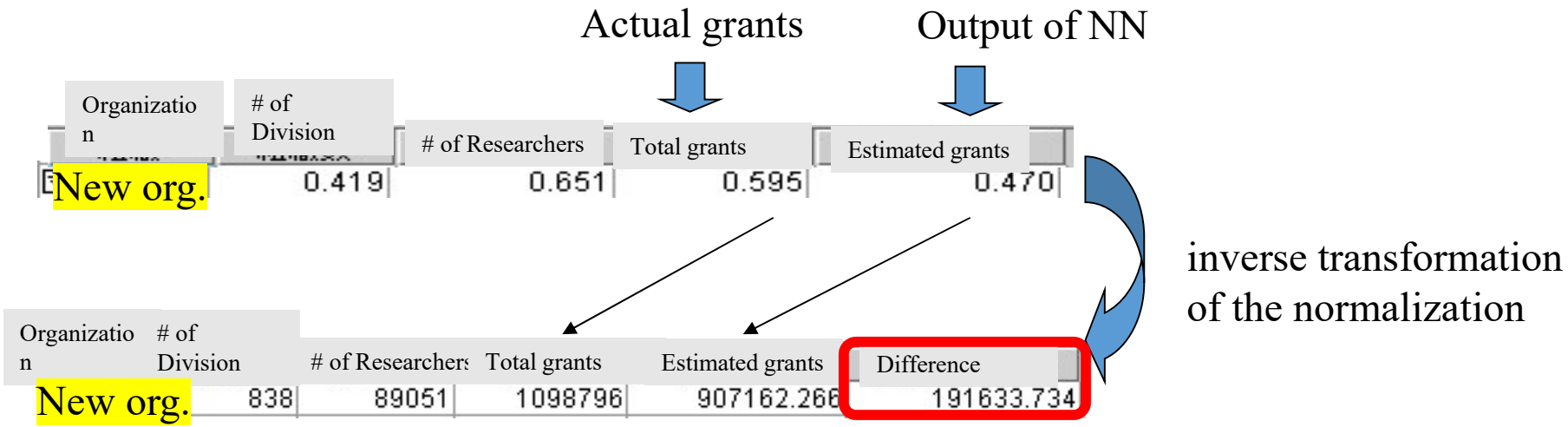
Estimation of fellowship grants (cont'd)

Estimated by NN

Organization	# of Division	# of Researchers	Total grants (actual)	Estimated grants
団体A	0.000	0.000	0.038	0.040
団体B	0.243	0.092	0.028	0.000
団体C	0.023	0.109	0.544	0.541
団体D	0.524	0.993	0.771	0.771
団体E	0.102	0.143	0.000	0.024
団体F	1.000	1.000	1.000	0.994

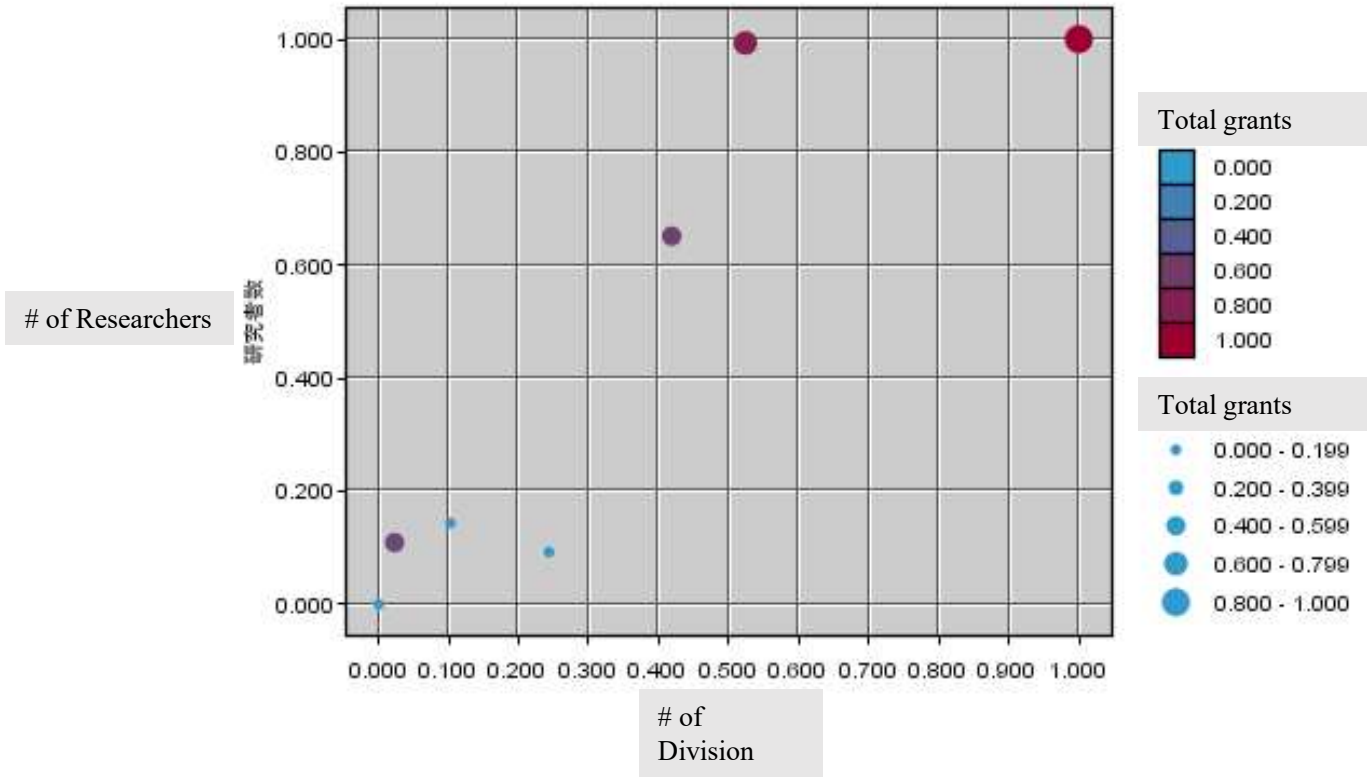
Estimation of fellowship grants (cont'd)

- Compare the estimated grants for a new organization with the actual grants



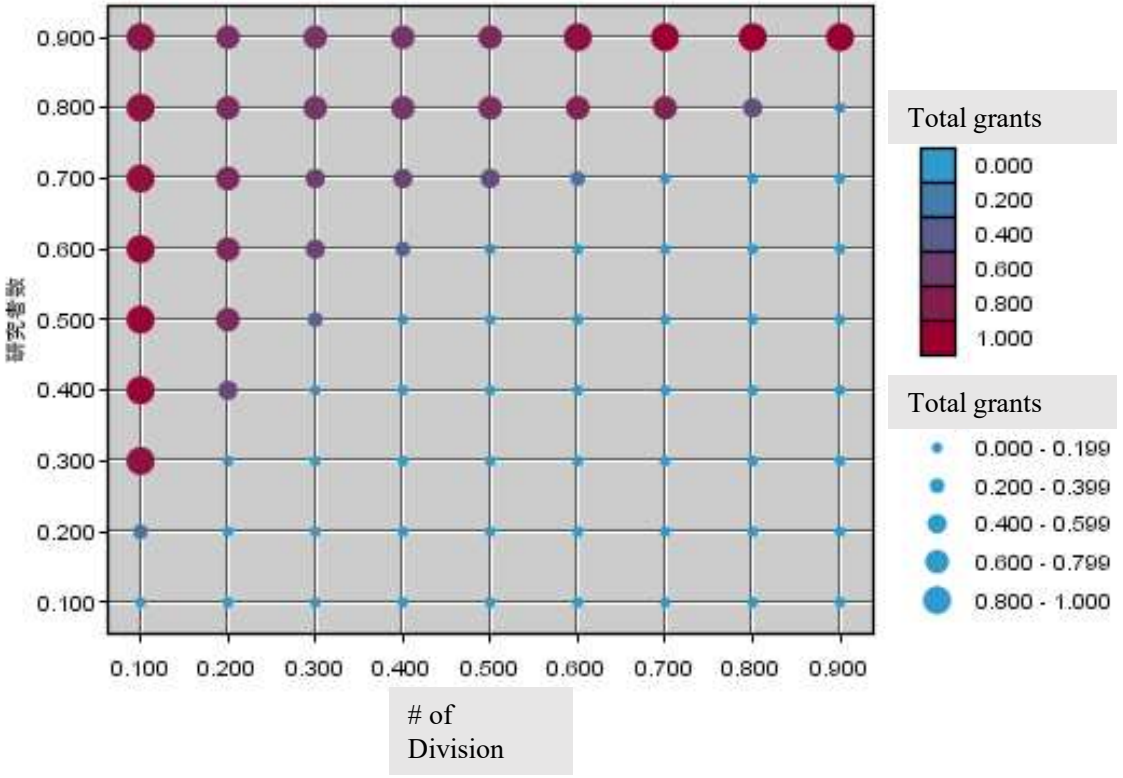
The result shows this organization receives more grants than other organizations

Distribution of the input data



The distribution of NN after learning

NN can be regarded as a non-linear interpolation function



Advantage and disadvantage

- Advantage
  - Ability to realize a highly non-linear complex function
  - Generally high precision of outputs
  - No restriction on input variables
- Disadvantage
  - Hard to understand the reason why the output is obtained
  - Long training time
  - Categorical data need to be converted to numerical data