Ch. 2-4, 2-5

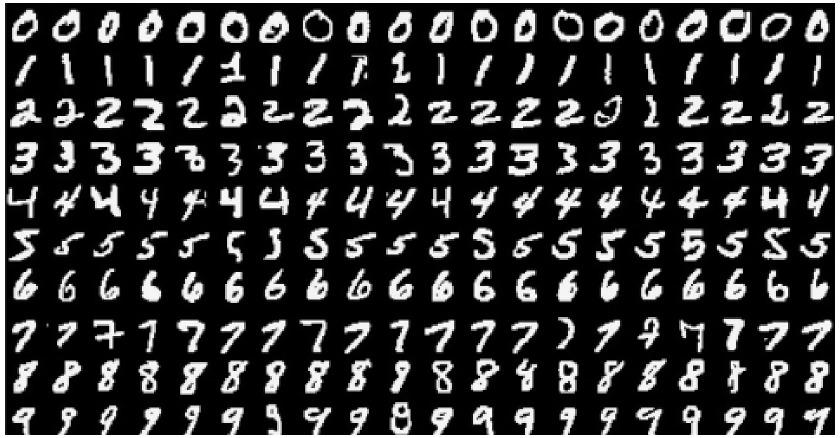
Pyo, Sanghun

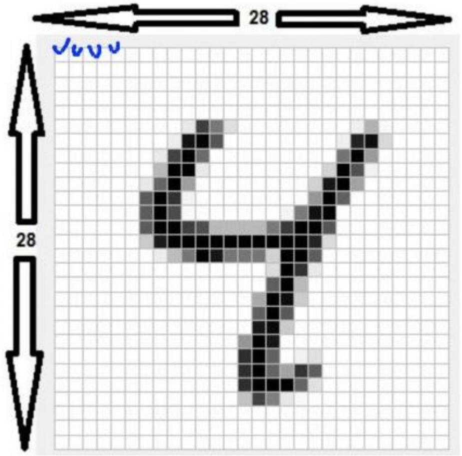
2.5 Reviewing the first example

```
from keras import models
from keras import layers
from keras.datasets import mnist
from keras.utils import to_categorical
from keras.utils import plot_model

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

MNIST data set





28X28X1 image file of handwritten digit number

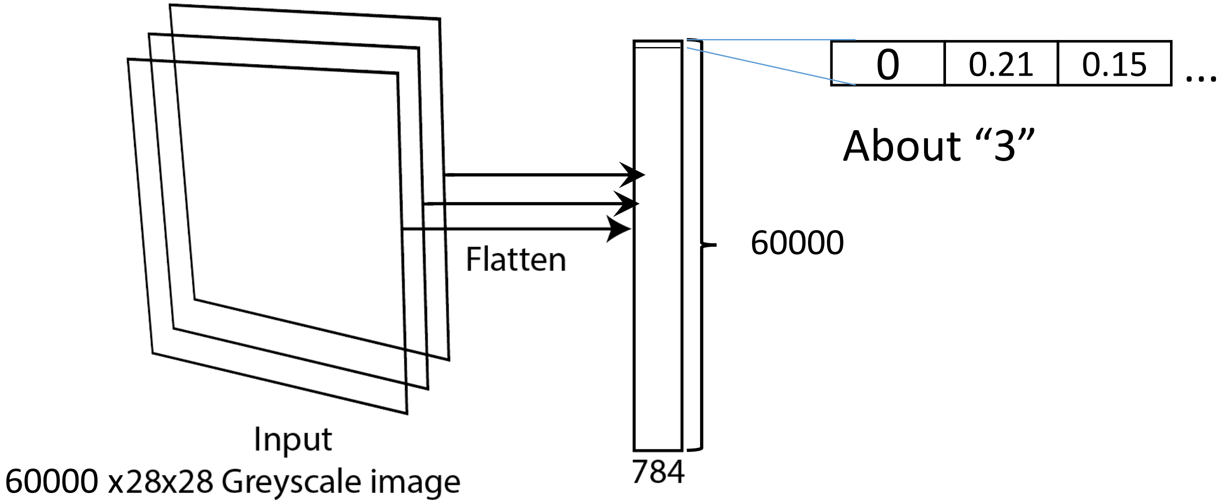
Layer

- A key component of neural networks which is a kind of data processing filter.
- Extract more meaningful expressions from input data for a given problem.

2.5 Reviewing the first example

```
print(train_images.shape)        -> (60000, 28, 28)
print(len(train_labels))         -> 60000
print(train_labels)              -> [3, 4, 2, 1....]
print(test_images.shape)         -> (10000, 28, 28)
print(len(test_labels))          -> 10000
print(test_labels)

train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype('float32') / 255     # normalization
print("train_images_reshape?")
print(train_images.shape)
```

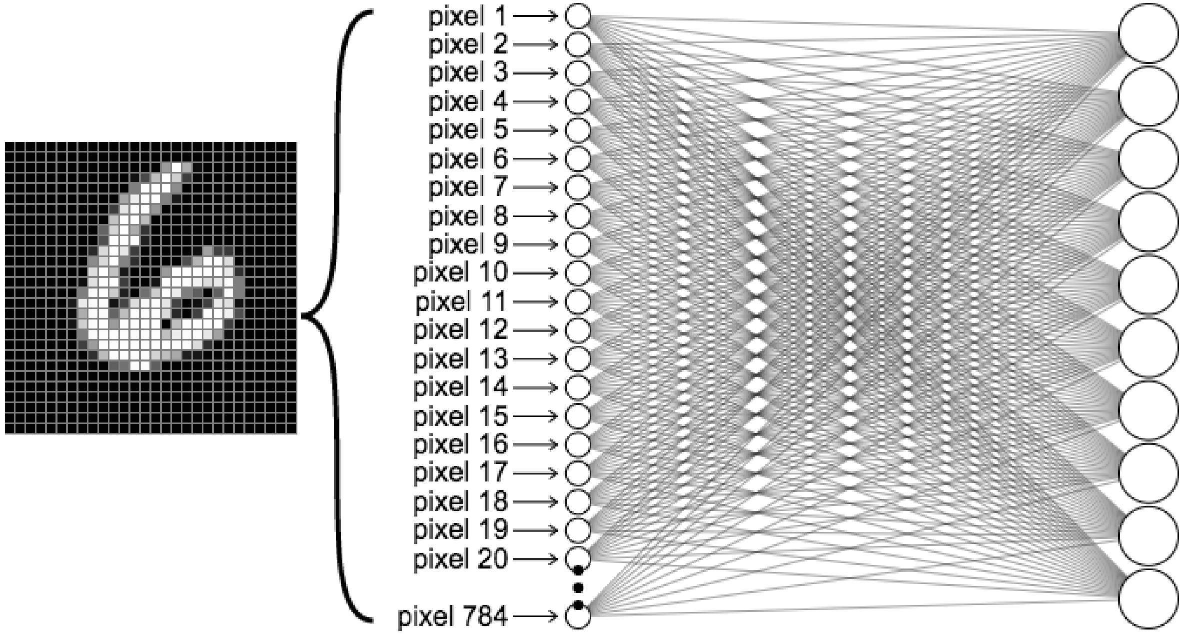


2.5 Reviewing the first example

```
train_labels = to_categorical(train_labels)
print("train_labels_categorical")
print(train_labels)
test_labels = to_categorical(test_labels)
```

[3, 4, 2, 1....]

[[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[0, 1,0 , 0, 0, 0, 0, 0, 0, 0]
....



The “dense” layer, which is fully connected to next dense layer or “softmax” layer needs the properly encoded labels

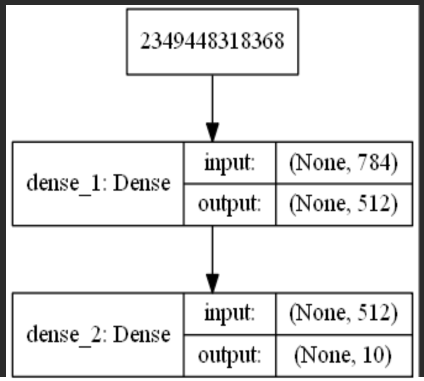
$$\frac{\exp(t_i)}{\boxed{\sum_{j=1}^{K} \exp(t_j)}} = \mathrm{softmax}(t_i)$$

=1

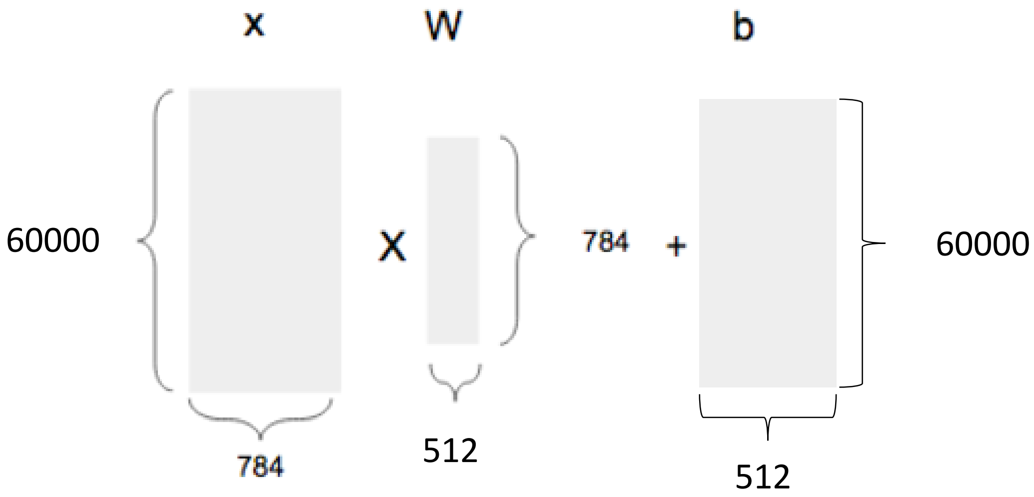
2.5 Reviewing the first example

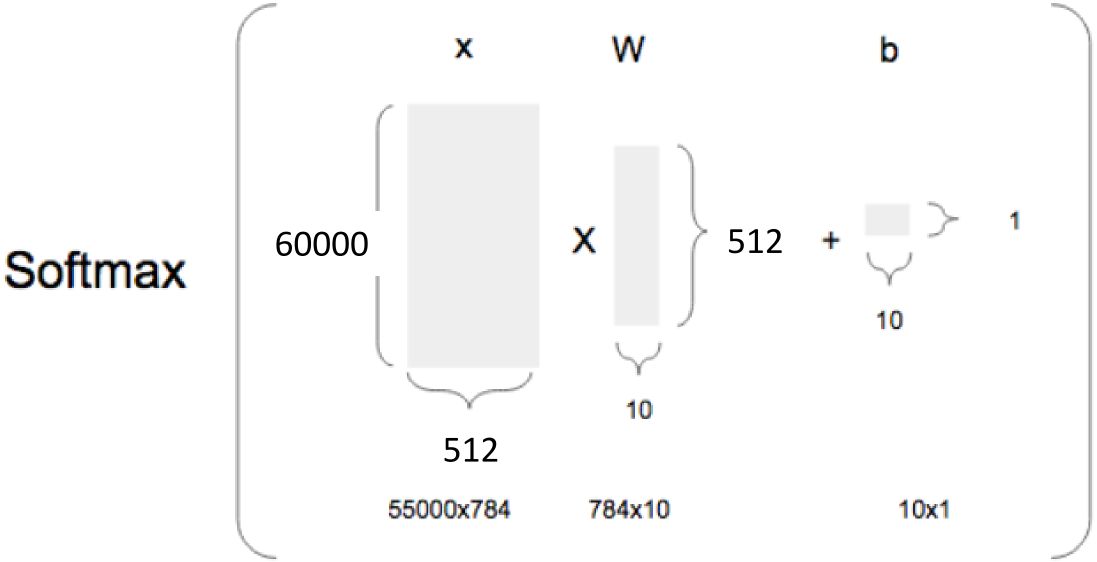
```
network = models.Sequential()
network.add(layers.Dense(512, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(10, activation='softmax'))

network.summary()
plot_model(network, to_file='model.png',show_shapes=True,
show_layer_names=True)
```



A user just need to set the shape of output. In Keras, the weigh and bias is automatically calculated.
The problem of this example is the classification problem. Thus, the later including “Softmax” needs to match up with the encoded labels.





2.4 Neural Network Engine: Gradient-based Optimization

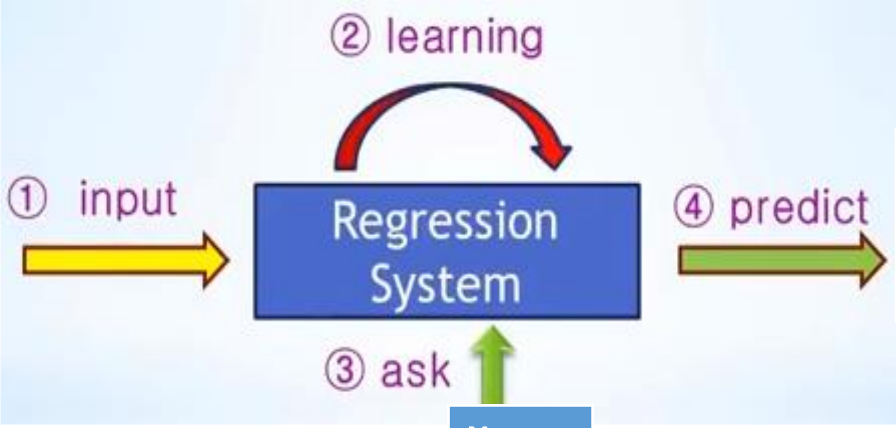
```
network.compile(optimizer='rmsprop',
                loss='categorical_crossentropy',
                metrics=['accuracy'])
network.fit(train_images, train_labels, epochs=5, batch_size=128)

test_loss, test_acc = network.evaluate(test_images, test_labels)
print('test_acc:', test_acc)
```

For example, if the batch size is 128, you will see only 128 data samples, get a gradient, and update the weight and bias. You can update weights much faster than gradient decent using the whole data. This method of obtaining weight is called Stochastic Gradient Descent method.

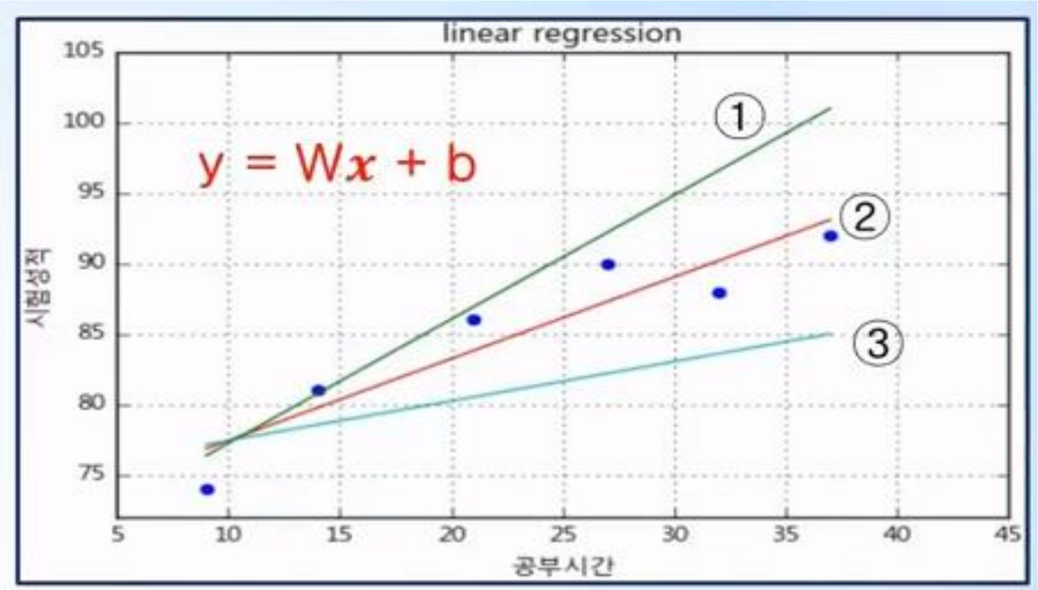
Gradient Descent for linear regression

X	t
9	74
14	81
21	86
27	90



X	t
55	?
60	?

X
55
60



2.4 Neural Network Engine: Gradient-based Optimization

Gradient Descent for linear regression

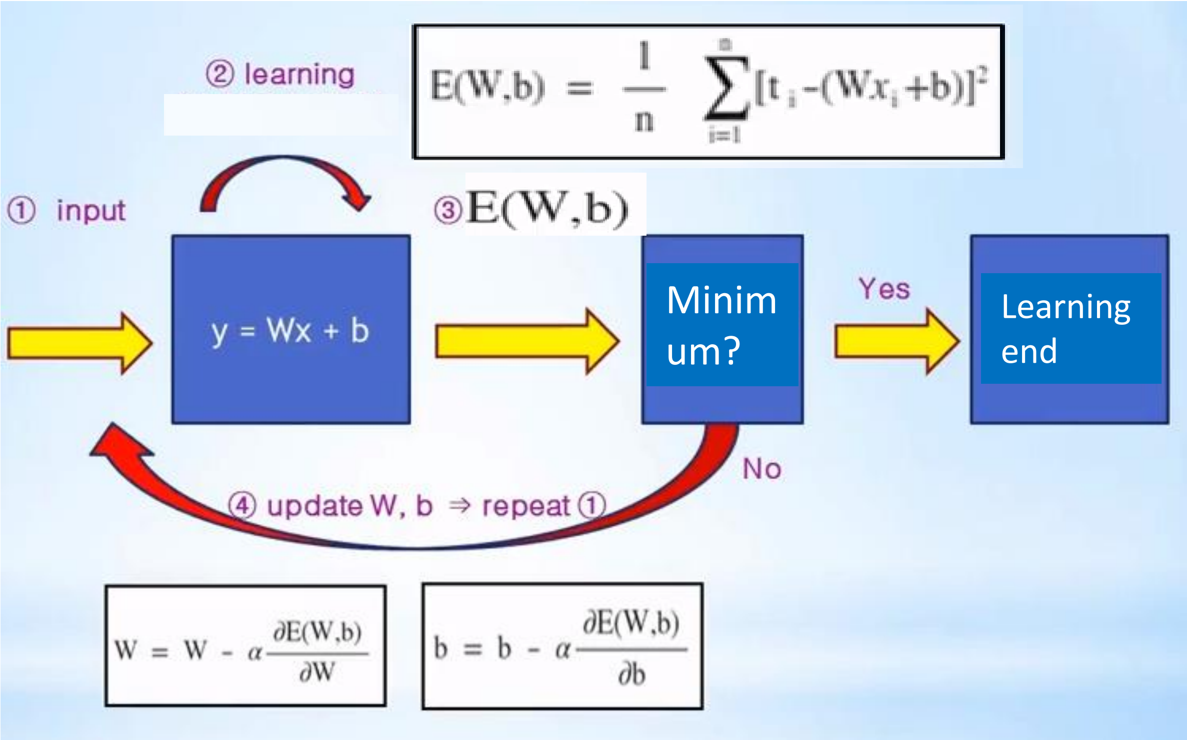
$$y = Wx + b$$

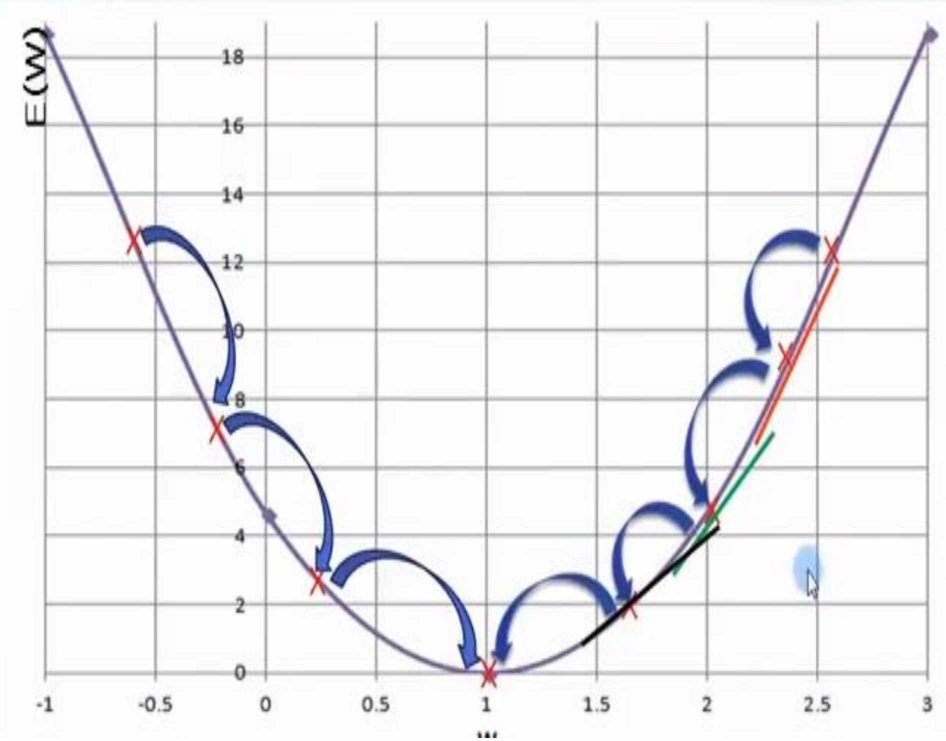
$$\text{loss function} = E(W,b) = \frac{1}{n}\sum_{i=1}^{n}[t_i - y_i]^2 = \frac{1}{n}\sum_{i=1}^{n}[t_i - (Wx_i + b)]^2$$

Linear regression is changed to convex function optimization problem

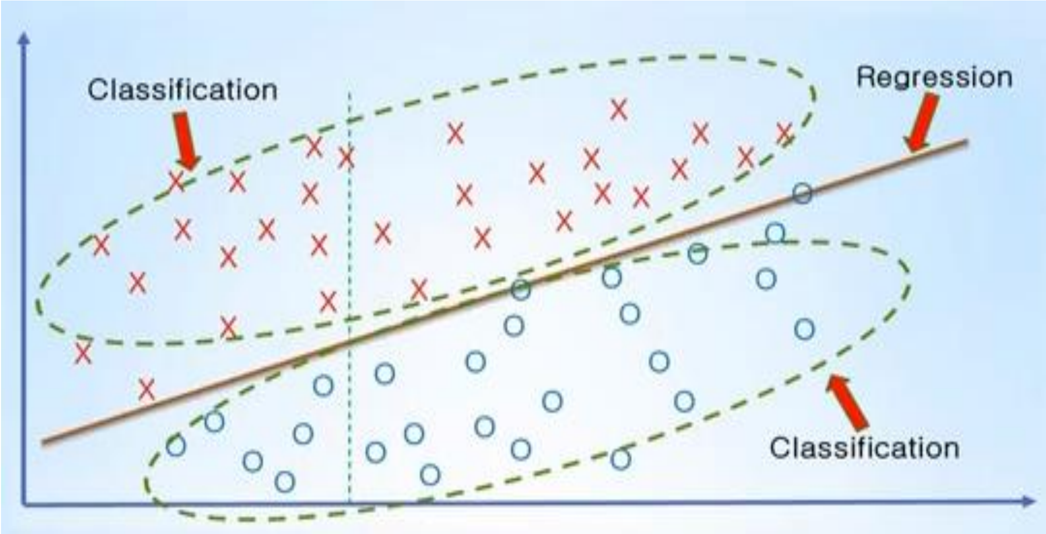
$$W = W - \boxed{\alpha}\frac{\partial E(W,b)}{\partial W} \qquad b = b - \boxed{\alpha}\frac{\partial E(W,b)}{\partial b}$$

Learning rate

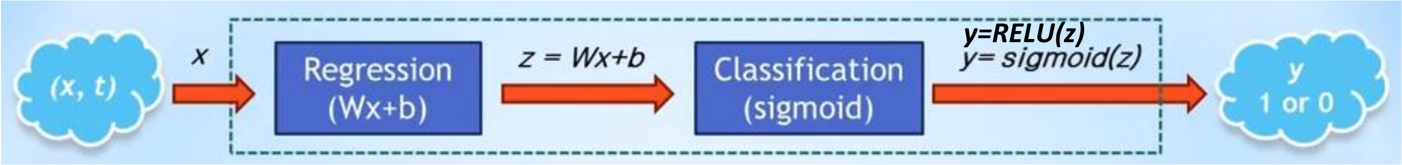




The concept of gradient descent for classification



1) Find the optimal fitted line representing the training data characteristics and distribution
2) Classify data as "up" or "down" based on its best fitted line

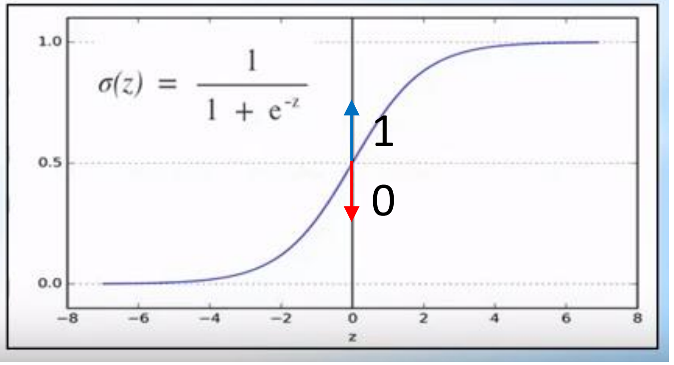


$$y = \frac{1}{1 + e^{-(Wx+b)}}$$

또는

$$z = Wx + b$$

$$y = \text{sigmoid}(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



The final output (y) of classification problem is logically 1 or 0 by the sigmoid or RELU function. Thus, a loss function is needed that is different from linear regression with continuous values.

Cross-entropy

$$y = \frac{1}{1 + e^{-(Wx+b)}}, \quad t_i = 0 \text{ or } 1$$

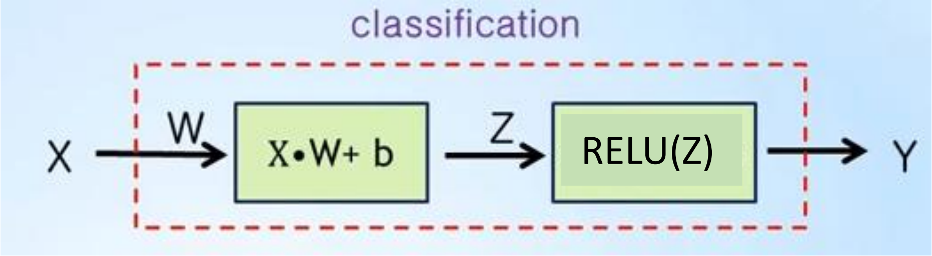
The result of linear regression

$$E(W,b) = -\sum_{i=1}^{n} \{t_i \log y_i + (1-t_i)\log(1-y_i)\}$$

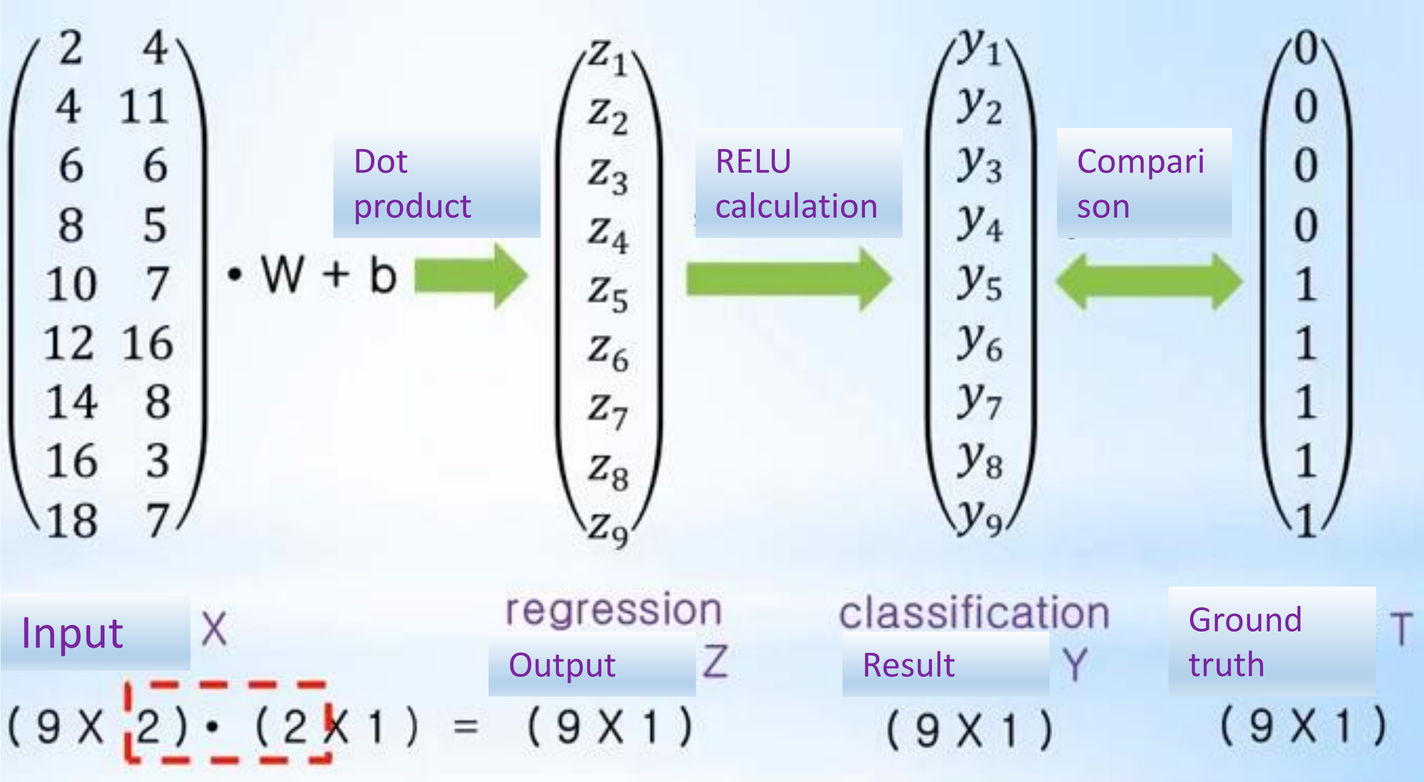
$$W = W - \alpha \frac{\partial E(W,b)}{\partial W} \qquad b = b - \alpha \frac{\partial E(W,b)}{\partial b}$$

The concept of gradient descent for classification for multi variable case

```
network.compile(optimizer='rmsprop',
                loss='categorical_crossentropy',
                metrics=['accuracy'])
network.fit(train_images, train_labels, epochs=5, batch_size=128)
```

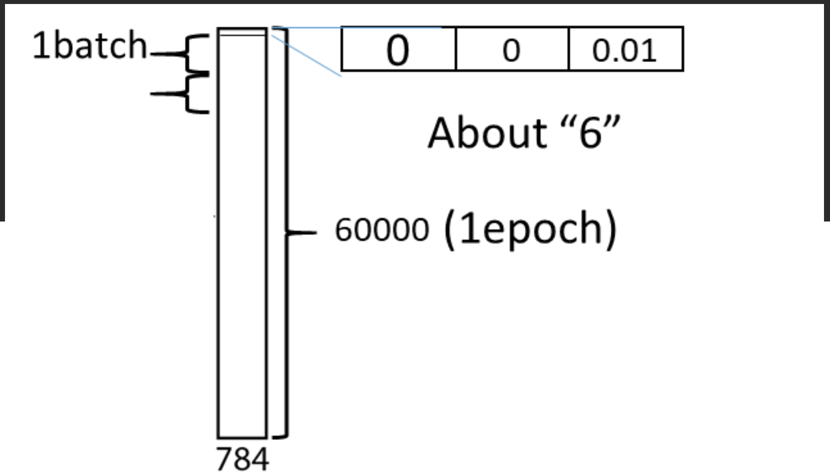


X1	X2	T
2	4	0
4	11	0
6	6	0
8	5	0
10	7	1
12	16	1
14	8	1
16	3	1
18	7	1

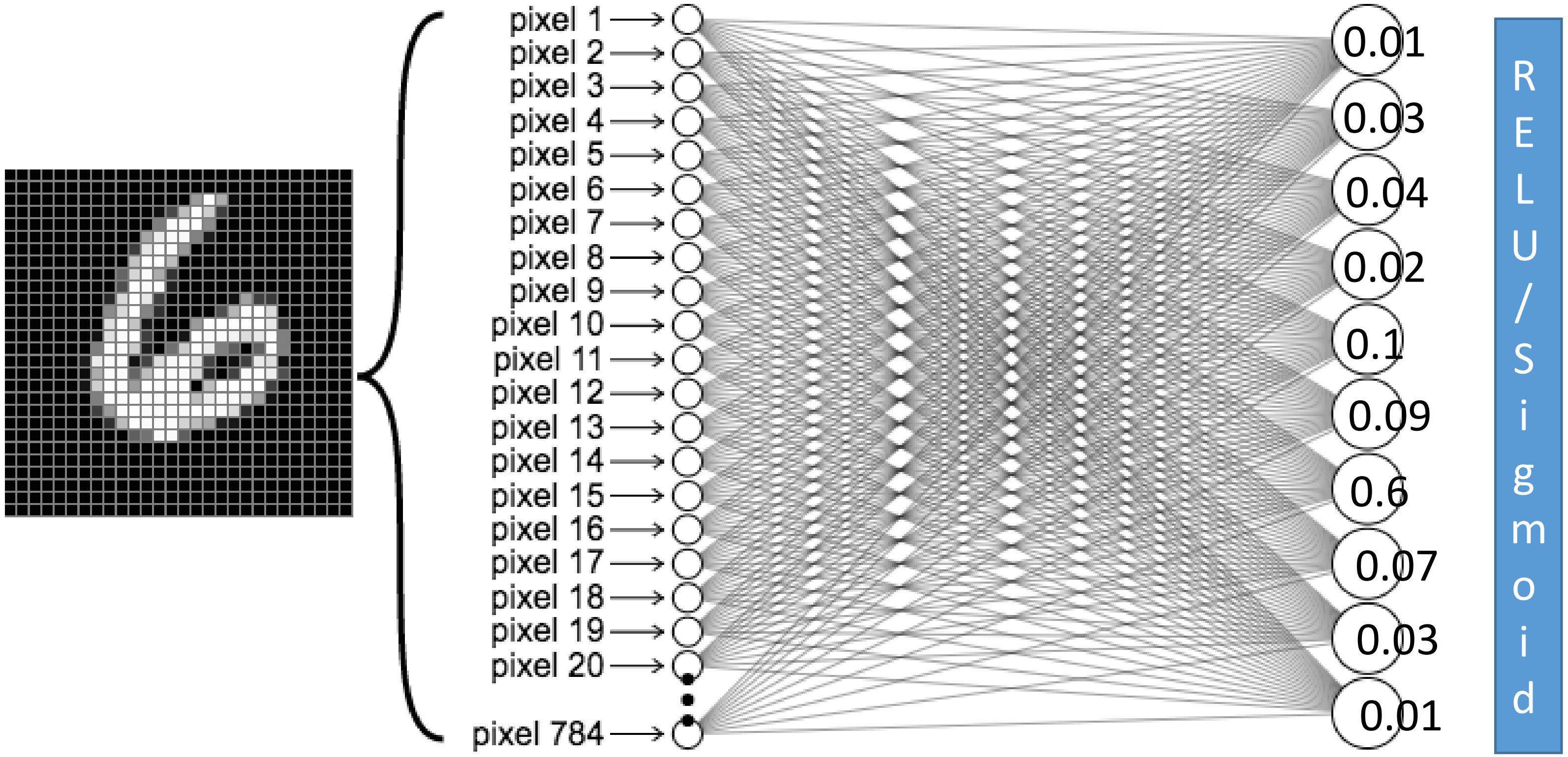


The concept of gradient descent for classification for multi variable case

```
network.compile(optimizer='rmsprop',
                loss='categorical_crossentropy',
                metrics=['accuracy'])
network.fit(train_images, train_labels, epochs=5, batch_size=128)
```

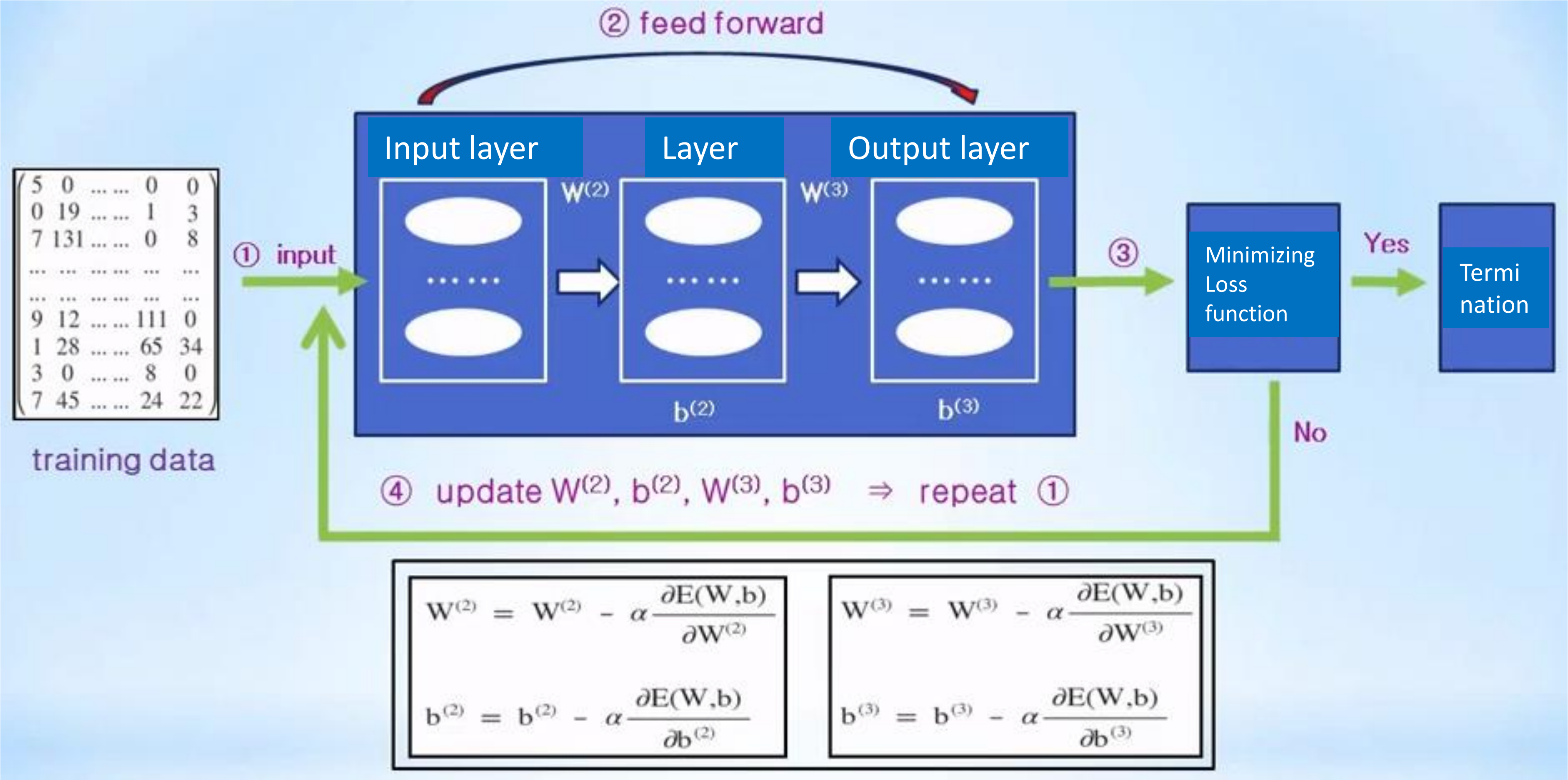


If the this code is successfully excuted..



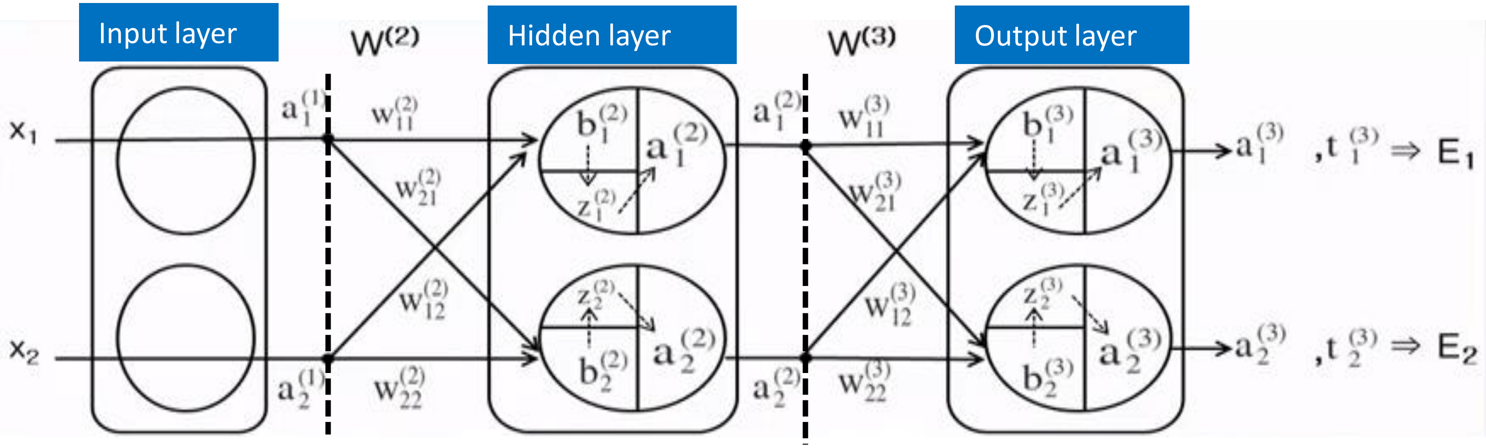
Back propagation – conceptual access with sigmoid

Problem of optimization process only using calculus



Excessive numerical differential operation causes increased computation cost!!

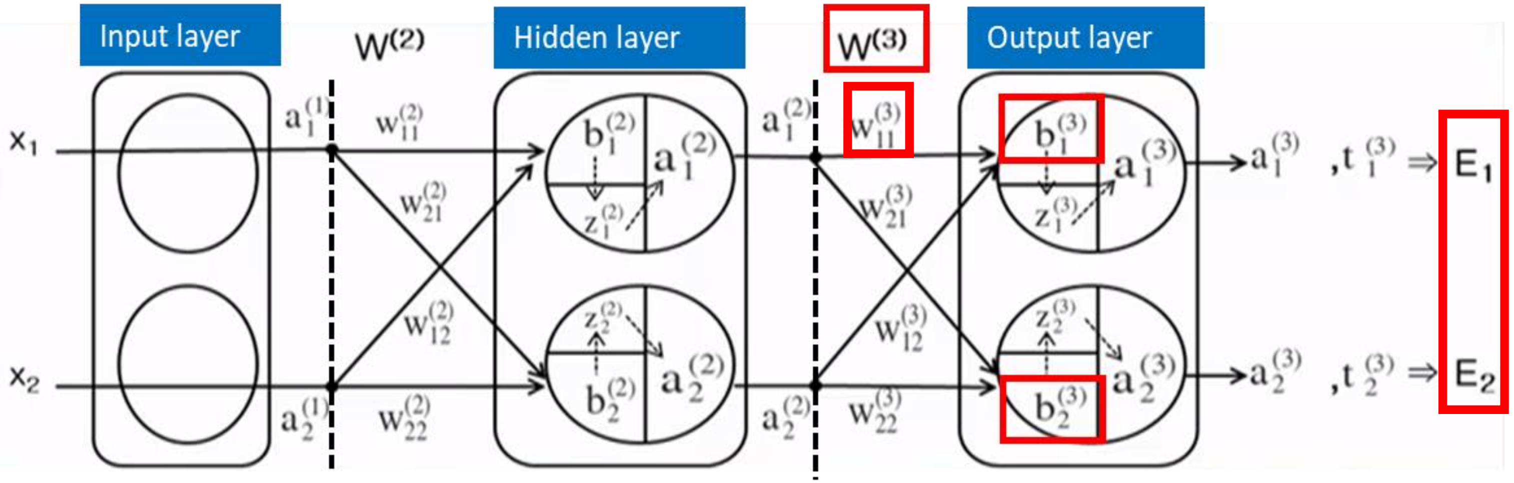
Back propagation – conceptual access with sigmoid



	Linear regression value (z)	Output (a)
Input	입력 층에는 가중치가 없기 때문에 선형회귀 값은 적용하지 않음	$a_1^{(1)} = x_1$ $a_2^{(1)} = x_2$
Hidden	$z_1^{(2)} = a_1^{(1)} w_{11}^{(2)} + a_2^{(1)} w_{12}^{(2)} + b_1^{(2)}$	$a_1^{(2)} = \text{sigmoid}(z_1^{(2)})$
	$z_2^{(2)} = a_1^{(1)} w_{21}^{(2)} + a_2^{(1)} w_{22}^{(2)} + b_2^{(2)}$	$a_2^{(2)} = \text{sigmoid}(z_2^{(2)})$
output	$z_1^{(3)} = a_1^{(2)} w_{11}^{(3)} + a_2^{(2)} w_{12}^{(3)} + b_1^{(3)}$	$a_1^{(3)} = \text{sigmoid}(z_1^{(3)})$
	$z_2^{(3)} = a_1^{(2)} w_{21}^{(3)} + a_2^{(2)} w_{22}^{(3)} + b_2^{(3)}$	$a_2^{(3)} = \text{sigmoid}(z_2^{(3)})$

$$
\begin{aligned}
\frac{\partial \text{sigmoid}(z)}{\partial z} &= \frac{\partial}{\partial z}\left(\frac{1}{1+e^{-z}}\right) \\
&= \frac{\partial}{\partial z}(1+e^{-z})^{-1} \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \frac{1}{(1+e^{-z})} \times \frac{e^{-z}}{(1+e^{-z})} \\
&= \frac{1}{(1+e^{-z})} \times \frac{(1+e^{-z})-1}{(1+e^{-z})} \\
&= \frac{1}{(1+e^{-z})} \times \left(1 - \frac{1}{(1+e^{-z})}\right) \\
&= \text{sigmoid}(z) \times (\ 1 - \text{sigmoid}(z)\ )
\end{aligned}
$$

Back propagation – conceptual access with sigmoid : example (1)



$$W^{(3)} = W^{(3)} - \alpha \frac{\partial E}{\partial W^{(3)}} \qquad b^{(3)} = b^{(3)} - \alpha \frac{\partial E}{\partial b^{(3)}}$$

$$\frac{\partial E}{\partial w_{11}^{(3)}}, \frac{\partial E}{\partial w_{21}^{(3)}}, \frac{\partial E}{\partial w_{12}^{(3)}}, \frac{\partial E}{\partial w_{22}^{(3)}} \qquad \frac{\partial E}{\partial b_1^{(3)}}, \frac{\partial E}{\partial b_2^{(3)}}$$

$$W^{(2)} = W^{(2)} - \alpha \frac{\partial E}{\partial W^{(2)}} \qquad b^{(2)} = b^{(2)} - \alpha \frac{\partial E}{\partial b^{(2)}}$$

$$\frac{\partial E}{\partial w_{11}^{(2)}}, \frac{\partial E}{\partial w_{21}^{(2)}}, \frac{\partial E}{\partial w_{12}^{(2)}}, \frac{\partial E}{\partial w_{22}^{(2)}} \qquad \frac{\partial E}{\partial b_1^{(2)}}, \frac{\partial E}{\partial b_2^{(2)}}$$

$$\boxed{\frac{\partial E}{\partial w_{11}^{(3)}}} = \frac{\partial E_1}{\partial w_{11}^{(3)}} + \cancelto{0}{\frac{\partial E_2}{\partial w_{11}^{(3)}}}$$

$$= \frac{\partial E_1}{\partial a_1^{(3)}} \times \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \times \frac{\partial z_1^{(3)}}{\partial w_{11}^{(3)}}$$

$$= \frac{\partial\left\{\frac{1}{2}(t_1^{(3)}-a_1^{(3)})^2\right\}}{\partial a_1^{(3)}} \times \frac{\partial \mathrm{sigmoid}(z_1^{(3)})}{\partial z_1^{(3)}} \times \frac{\partial(a_1^{(2)}w_{11}^{(3)}+a_2^{(2)}w_{12}^{(3)}+b_1^{(3)})}{\partial w_{11}^{(3)}}$$

$$= (a_1^{(3)} - t_1^{(3)}) \times \mathrm{sigmoid}(z_1^{(3)}) \times (1-\mathrm{sigmoid}(z_1^{(3)})) \times a_1^{(2)}$$

$$= \boxed{(a_1^{(3)} - t_1^{(3)}) \times a_1^{(3)} \times (1 - a_1^{(3)}) \times a_1^{(2)}}$$

Calculus becomes algebraic equation!

where

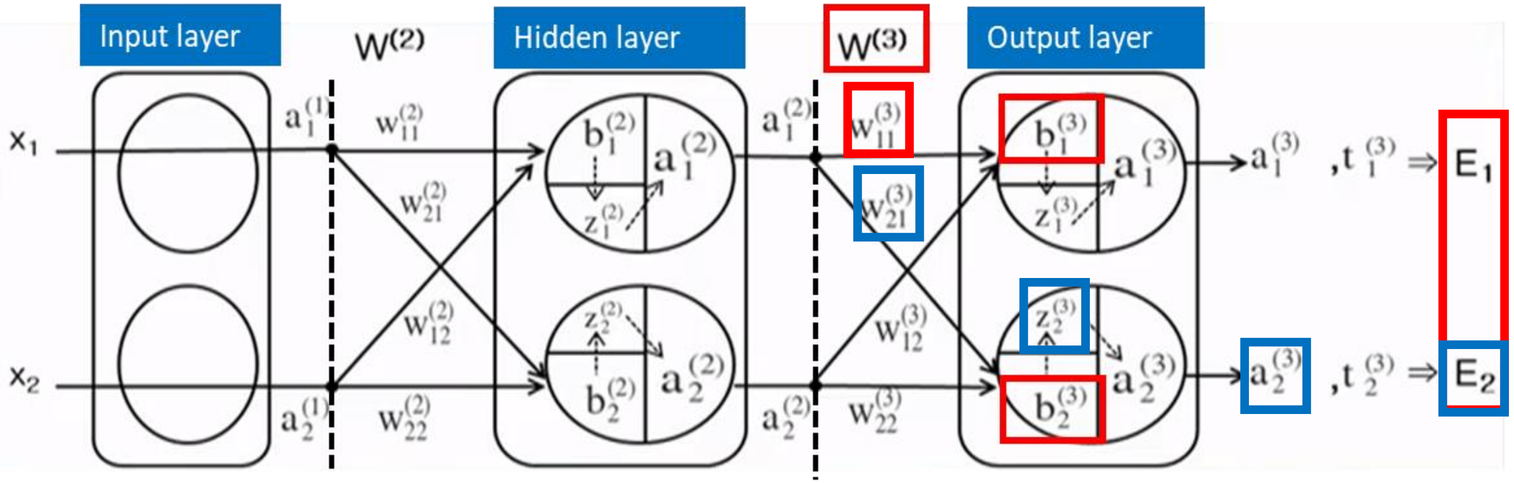
$$E = E_1 + E_2$$

$$E_1 = \frac{1}{2}(t_1^{(3)}-a_1^{(3)})^2$$

$$a_1^{(3)} = \mathrm{sigmoid}(z_1^{(3)})$$

$$z_1^{(3)} = a_1^{(2)}w_{11}^{(3)} + a_2^{(2)}w_{12}^{(3)} + b_1^{(3)}$$

Back propagation – conceptual access with sigmoid : example (2)



$$W^{(3)} = W^{(3)} - \alpha \frac{\partial E}{\partial W^{(3)}} \qquad b^{(3)} = b^{(3)} - \alpha \frac{\partial E}{\partial b^{(3)}}$$

$$\frac{\partial E}{\partial w^{(3)}_{11}},\ \frac{\partial E}{\partial w^{(3)}_{21}},\ \frac{\partial E}{\partial w^{(3)}_{12}},\ \frac{\partial E}{\partial w^{(3)}_{22}} \qquad \frac{\partial E}{\partial b^{(3)}_{1}},\ \frac{\partial E}{\partial b^{(3)}_{2}}$$

$$W^{(2)} = W^{(2)} - \alpha \frac{\partial E}{\partial W^{(2)}} \qquad b^{(2)} = b^{(2)} - \alpha \frac{\partial E}{\partial b^{(2)}}$$

$$\frac{\partial E}{\partial w^{(2)}_{11}},\ \frac{\partial E}{\partial w^{(2)}_{21}},\ \frac{\partial E}{\partial w^{(2)}_{12}},\ \frac{\partial E}{\partial w^{(2)}_{22}} \qquad \frac{\partial E}{\partial b^{(2)}_{1}},\ \frac{\partial E}{\partial b^{(2)}_{2}}$$

$$\frac{\partial E}{\partial w^{(3)}_{21}} = \cancel{\frac{\partial E_1}{\partial w^{(3)}_{21}}}^{\,0} + \frac{\partial E_2}{\partial w^{(3)}_{21}}$$

$$= \frac{\partial E_2}{\partial a^{(3)}_2} \times \frac{\partial a^{(3)}_2}{\partial z^{(3)}_2} \times \frac{\partial z^{(3)}_2}{\partial w^{(3)}_{21}}$$

$$= \frac{\partial\left\{\frac{1}{2}(t^{(3)}_2 - a^{(3)}_2)^2\right\}}{\partial a^{(3)}_2} \times \frac{\partial \mathrm{sigmoid}(z^{(3)}_2)}{\partial z^{(3)}_2} \times \frac{\partial(a^{(2)}_1 w^{(3)}_{21} + a^{(2)}_2 w^{(3)}_{22} + b^{(3)}_2)}{\partial w^{(3)}_{21}}$$

$$= (a^{(3)}_2 - t^{(3)}_2) \times \mathrm{sigmoid}(z^{(3)}_2) \times (1 - \mathrm{sigmoid}(z^{(3)}_2)) \times a^{(2)}_1$$

$$= \boxed{(a^{(3)}_2 - t^{(3)}_2) \times a^{(3)}_2 \times (1 - a^{(3)}_2) \times a^{(2)}_1}$$

where

$$\boxed{E = E_1 + E_2}$$

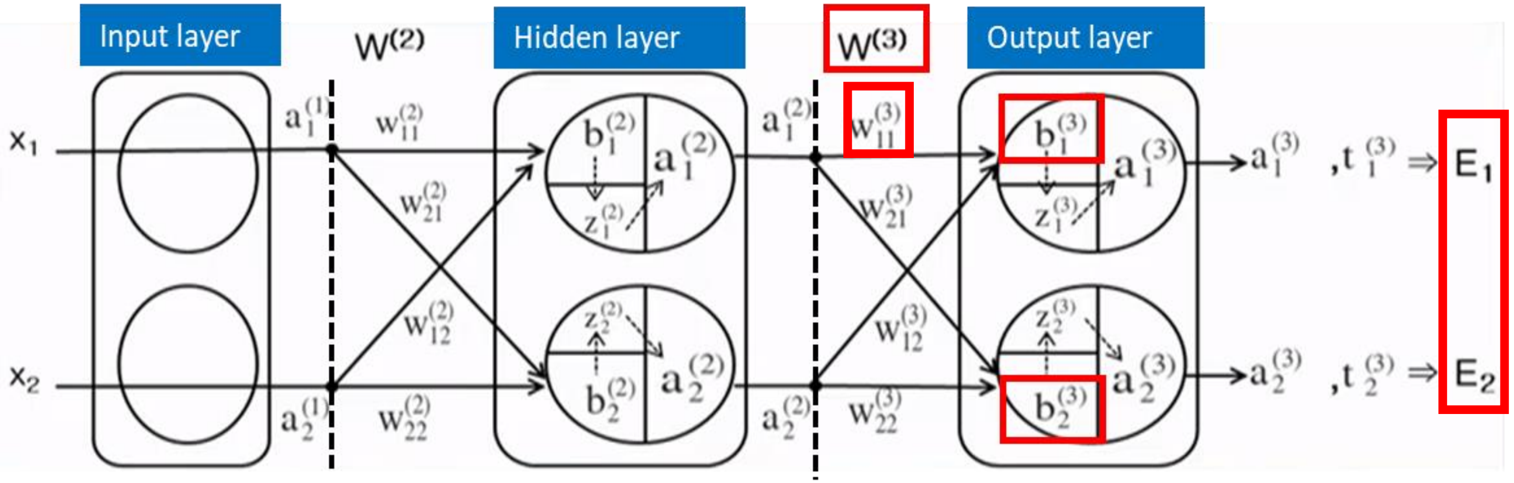
$$\boxed{E_1 = \frac{1}{2}(t^{(3)}_1 - a^{(3)}_1)^2}$$

$$\boxed{a^{(3)}_1 = \mathrm{sigmoid}(z^{(3)}_1)}$$

$$\boxed{z^{(3)}_1 = a^{(2)}_1 w^{(3)}_{11} + a^{(2)}_2 w^{(3)}_{12} + b^{(3)}_1}$$

Similarly, the others of weight in output layer (3) can be computed such as the back propagation of w_{11} and w_{12} in output layer.

Back propagation – conceptual access with sigmoid : example (3)



$$\boxed{W^{(3)} = W^{(3)} - \alpha \frac{\partial E}{\partial W^{(3)}}} \quad \boxed{b^{(3)} = b^{(3)} - \alpha \frac{\partial E}{\partial b^{(3)}}}$$

$$\boxed{\frac{\partial E}{\partial w_{11}^{(3)}},\ \frac{\partial E}{\partial w_{21}^{(3)}},\ \frac{\partial E}{\partial w_{12}^{(3)}},\ \frac{\partial E}{\partial w_{22}^{(3)}}} \quad \boxed{\frac{\partial E}{\partial b_1^{(3)}},\ \frac{\partial E}{\partial b_2^{(3)}}}$$

$$\boxed{W^{(2)} = W^{(2)} - \alpha \frac{\partial E}{\partial W^{(2)}}} \quad \boxed{b^{(2)} = b^{(2)} - \alpha \frac{\partial E}{\partial b^{(2)}}}$$

$$\boxed{\frac{\partial E}{\partial w_{11}^{(2)}},\ \frac{\partial E}{\partial w_{21}^{(2)}},\ \frac{\partial E}{\partial w_{12}^{(2)}},\ \frac{\partial E}{\partial w_{22}^{(2)}}} \quad \boxed{\frac{\partial E}{\partial b_1^{(2)}},\ \frac{\partial E}{\partial b_2^{(2)}}}$$

$$\boxed{\frac{\partial E}{\partial b_1^{(3)}}} = \frac{\partial E_1}{\partial b_1^{(3)}} + \cancel{\frac{\partial E_2}{\partial b_1^{(3)}}}\ 0$$

$$= \frac{\partial E_1}{\partial a_1^{(3)}} \times \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \times \frac{\partial z_1^{(3)}}{\partial b_1^{(3)}}$$

$$= \frac{\partial\left\{\frac{1}{2}(t_1^{(3)} - a_1^{(3)})^2\right\}}{\partial a_1^{(3)}} \times \frac{\partial \mathrm{sigmoid}(z_1^{(3)})}{\partial z_1^{(3)}} \times \frac{\partial(a_1^{(2)}w_{11}^{(3)} + a_2^{(2)}w_{12}^{(3)} + b_1^{(3)})}{\partial b_1^{(3)}}$$

$$= (a_1^{(3)} - t_1^{(3)}) \times \mathrm{sigmoid}(z_1^{(3)}) \times (1 - \mathrm{sigmoid}(z_1^{(3)})) \times 1$$

$$= \boxed{(a_1^{(3)} - t_1^{(3)}) \times a_1^{(3)} \times (1 - a_1^{(3)}) \times 1}$$

where

$$\boxed{E = E_1 + E_2}$$

$$\boxed{E_1 = \frac{1}{2}(t_1^{(3)} - a_1^{(3)})^2}$$

$$\boxed{a_1^{(3)} = \mathrm{sigmoid}(z_1^{(3)})}$$

$$\boxed{z_1^{(3)} = a_1^{(2)}w_{11}^{(3)} + a_2^{(2)}w_{12}^{(3)} + b_1^{(3)}}$$

Let's think about the back propagation between hidden layer to E