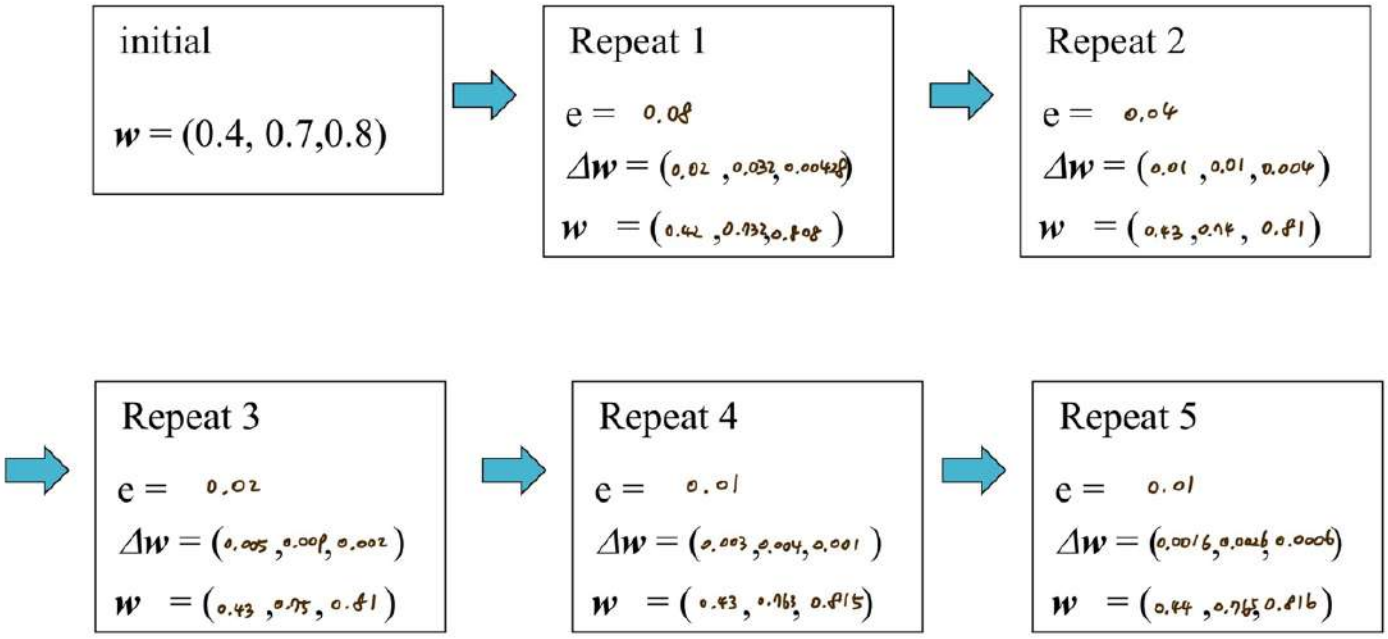
1. Learning in Neural Network

- Example 1



$$w_{ij} \leftarrow w_{ij} + \underbrace{\alpha\, e_j x_i}_{\Delta w}$$

Error is decreasing ?

```python
import numpy as np
import torch

# simple delta rule
x = torch.tensor([0.5, 0.8, 0.2])  # input
w = torch.tensor([0.4, 0.7, 0.8])   # weight
d = 1                               # target
alpha = 0.5                         # learning rate

# update w
for i in range(50):
    v = torch.sum(w * x)
    y = torch.sigmoid(v)            # PyTorch
    e = d - y
    print("error", i, e)
    w = w + alpha * y * (1 - y) * e * x  # update w
```

```
error 0 tensor(0.2850)
error 1 tensor(0.2795)
error 2 tensor(0.2742)
error 3 tensor(0.2692)
error 4 tensor(0.2644)
error 5 tensor(0.2598)
error 6 tensor(0.2553)
error 7 tensor(0.2511)
error 8 tensor(0.2470)
error 9 tensor(0.2430)
error 10 tensor(0.2392)
error 11 tensor(0.2355)
error 12 tensor(0.2320)
error 13 tensor(0.2286)
error 14 tensor(0.2253)
error 15 tensor(0.2221)
error 16 tensor(0.2191)
error 17 tensor(0.2161)
error 18 tensor(0.2132)
error 19 tensor(0.2105)
error 20 tensor(0.2078)
error 21 tensor(0.2052)
error 22 tensor(0.2026)
error 23 tensor(0.2002)
error 24 tensor(0.1978)
error 25 tensor(0.1955)
error 26 tensor(0.1933)
error 27 tensor(0.1911)
error 28 tensor(0.1890)
error 29 tensor(0.1869)
error 30 tensor(0.1849)
error 31 tensor(0.1830)
error 32 tensor(0.1811)
```

```
error 33 tensor(0.1792)
error 34 tensor(0.1774)
error 35 tensor(0.1757)
error 36 tensor(0.1740)
error 37 tensor(0.1723)
error 38 tensor(0.1707)
error 39 tensor(0.1691)
error 40 tensor(0.1676)
error 41 tensor(0.1661)
error 42 tensor(0.1646)
error 43 tensor(0.1631)
error 44 tensor(0.1617)
error 45 tensor(0.1604)
error 46 tensor(0.1590)
error 47 tensor(0.1577)
error 48 tensor(0.1564)
error 49 tensor(0.1551)
```