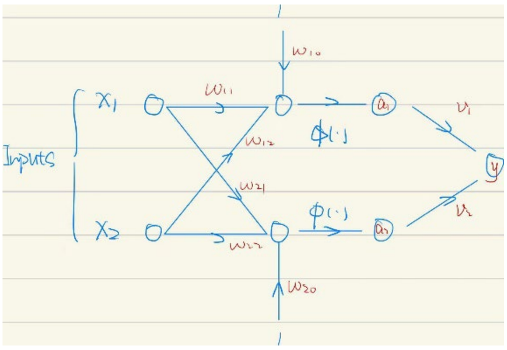
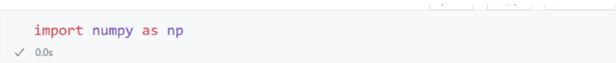
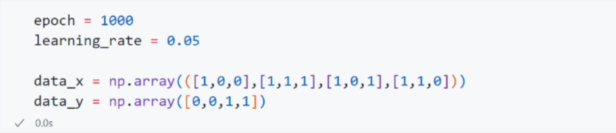
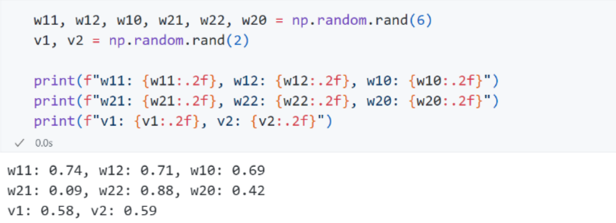




```
import numpy as np
```

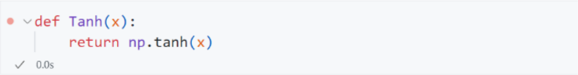


```
epoch = 1000
learning_rate = 0.05

data_x = np.array(([1,0,0],[1,1,1],[1,0,1],[1,1,0]))
data_y = np.array([0,0,1,1])
```

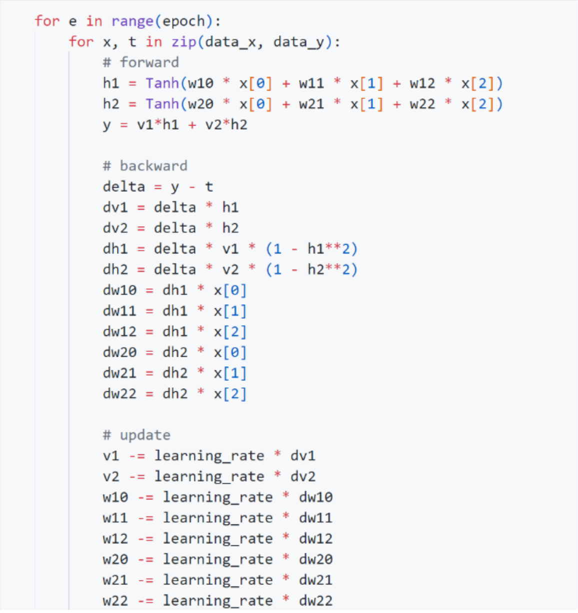


```
w11, w12, w10, w21, w22, w20 = np.random.rand(6)
v1, v2 = np.random.rand(2)

print(f"w11: {w11:.2f}, w12: {w12:.2f}, w10: {w10:.2f}")
print(f"w21: {w21:.2f}, w22: {w22:.2f}, w20: {w20:.2f}")
print(f"v1: {v1:.2f}, v2: {v2:.2f}")
```

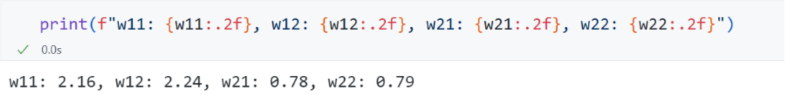
```
w11: 0.74, w12: 0.71, w10: 0.69
w21: 0.09, w22: 0.88, w20: 0.42
v1: 0.58, v2: 0.59
```



```
def Tanh(x):
    return np.tanh(x)
```

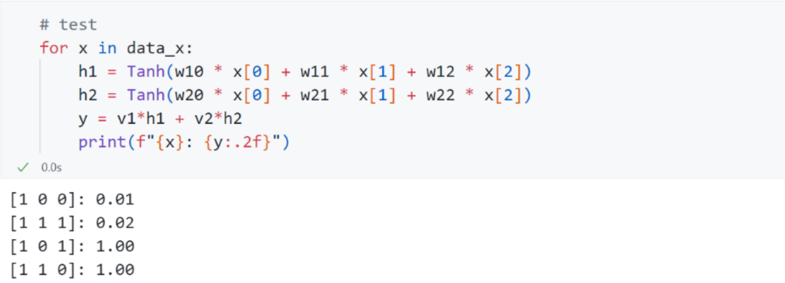


```
for e in range(epoch):
    for x, t in zip(data_x, data_y):
        # forward
        h1 = Tanh(w10 * x[0] + w11 * x[1] + w12 * x[2])
        h2 = Tanh(w20 * x[0] + w21 * x[1] + w22 * x[2])
        y = v1*h1 + v2*h2

        # backward
        delta = y - t
        dv1 = delta * h1
        dv2 = delta * h2
        dh1 = delta * v1 * (1 - h1**2)
        dh2 = delta * v2 * (1 - h2**2)
        dw10 = dh1 * x[0]
        dw11 = dh1 * x[1]
        dw12 = dh1 * x[2]
        dw20 = dh2 * x[0]
        dw21 = dh2 * x[1]
        dw22 = dh2 * x[2]

        # update
        v1 -= learning_rate * dv1
        v2 -= learning_rate * dv2
        w10 -= learning_rate * dw10
        w11 -= learning_rate * dw11
        w12 -= learning_rate * dw12
        w20 -= learning_rate * dw20
        w21 -= learning_rate * dw21
        w22 -= learning_rate * dw22
```



```
print(f"w11: {w11:.2f}, w12: {w12:.2f}, w21: {w21:.2f}, w22: {w22:.2f}")
```

```
w11: 2.16, w12: 2.24, w21: 0.78, w22: 0.79
```



```
# test
for x in data_x:
    h1 = Tanh(w10 * x[0] + w11 * x[1] + w12 * x[2])
    h2 = Tanh(w20 * x[0] + w21 * x[1] + w22 * x[2])
    y = v1*h1 + v2*h2
    print(f"{x}: {y:.2f}")
```

```
[1 0 0]: 0.01
[1 1 1]: 0.02
[1 0 1]: 1.00
[1 1 0]: 1.00
```