

```
Untitled14 x Home x Untitled14 x Jupyter Server x +
localhost:8888/notebooks/Untitled14.ipynb
jupyter Untitled14 Last Checkpoint: 17 minutes ago
File Edit View Run Kernel Settings Help
JupyterLab Python 3 (ipykernel)

[1]: # توليد رقم عشوائي بين 0.5 و 0.5 بدون مكبات
def random_weight(seed):
    seed = (seed * 9301 + 49297) % 233280
    return (seed / 233280.0) - 0.5 # توليد رقم عشوائي بين 0.5 و 0.5 بدون مكبات

# تعريف كلاس الشبكة العصبية
class NeuralNetwork:
    def __init__(self, seed=123):
        self.weights = [random_weight(seed + i) for i in range(8)] # توليد 8 اوزان عشوائية
        self.b1 = 0.5
        self.b2 = 0.7

    def tanh(self, x):
        """ محسوبة بتقريب tanh بالة لقميل """
        e_pos = self.exp(x)
        e_neg = self.exp(-x)
        return (e_pos - e_neg) / (e_pos + e_neg)

    def exp(self, x):
        """ باستخدام طريقة تايلور لحساب e^x """
        n = 20
        result, factorial, power = 1.0, 1.0, 1.0
        for i in range(1, n):
            factorial *= i
            power *= x
            result += power / factorial
        return result

    def forward(self, i1, i2):
        """ تنفيذ تمرير البيانات داخل الشبكة """
        w1, w2, w3, w4, w5, w6, w7, w8 = self.weights

        # حساب مخرجات الطبقة المخفية
        h1_input = (i1 * w1) + (i2 * w3) + self.b1
        h2_input = (i1 * w2) + (i2 * w4) + self.b1

        h1_output = self.tanh(h1_input)
        h2_output = self.tanh(h2_input)

        # حساب مخرجات الطبقة النهائية
        o1_input = (h1_output * w5) + (h2_output * w7) + self.b2
        o2_input = (h1_output * w6) + (h2_output * w8) + self.b2

        o1_output = self.tanh(o1_input)
        o2_output = self.tanh(o2_input)

        return o1_output, o2_output

# إنشاء نموذج الشبكة العصبية
nn = NeuralNetwork()

# مدخلات الشبكة
i1, i2 = 0.05, 0.10

o1, o2 = nn.forward(i1, i2)
print("Output O1:", o1)
print("Output O2:", o2)

Output O1: 0.4951707552773543
Output O2: 0.5202952198401902
```

```
Untitled14 x Home x Untitled14 x Jupyter Server x +
localhost:8888/notebooks/Untitled14.ipynb
jupyter Untitled14 Last Checkpoint: 17 minutes ago
File Edit View Run Kernel Settings Help
JupyterLab Python 3 (ipykernel)

power *= x
result += power / factorial
return result

def forward(self, i1, i2):
    """ تنفيذ تمرير البيانات داخل الشبكة """
    w1, w2, w3, w4, w5, w6, w7, w8 = self.weights

    # حساب مخرجات الطبقة المخفية
    h1_input = (i1 * w1) + (i2 * w3) + self.b1
    h2_input = (i1 * w2) + (i2 * w4) + self.b1

    h1_output = self.tanh(h1_input)
    h2_output = self.tanh(h2_input)

    # حساب مخرجات الطبقة النهائية
    o1_input = (h1_output * w5) + (h2_output * w7) + self.b2
    o2_input = (h1_output * w6) + (h2_output * w8) + self.b2

    o1_output = self.tanh(o1_input)
    o2_output = self.tanh(o2_input)

    return o1_output, o2_output

# إنشاء نموذج الشبكة العصبية
nn = NeuralNetwork()

# مدخلات الشبكة
i1, i2 = 0.05, 0.10

o1, o2 = nn.forward(i1, i2)
print("Output O1:", o1)
print("Output O2:", o2)

Output O1: 0.4951707552773543
Output O2: 0.5202952198401902
```