Bài thực hành 1: Tạo Nơ-ron và lớp nơ-ron (Source: Build Python from Scratch)

Mạng thần kinh nhân tạo lấy cảm hứng từ mạng nơ-ron của não bộ, được tổ chức dưới hình thức dữ liệu có thể lập trình được bởi máy tính. Trong 1 mạng thần kinh nhân tạo, có các nơ-ron, các hàm kích hoạt (activation function), các liên kết giữa các nơ-ron và 1 số chức năng tính toán khác (tính toán hàm mất mát, lan truyền ngược, tối ưu hóa).

1 nơ-ron nhân tạo bao gồm: Các giá trị đầu vào (inputs), được lập trình dưới dạng vec-tơ inputs. Các trọng số (weights) tương ứng với các giá trị input, cũng được lưu dưới dạng vec-tơ và 1 biến bias.

Tín hiệu ngõ ra tương ứng với vec-tơ giá trị đầu vào được tín theo công thức sau:

output[i]= weight[i] * input[i]; [1]

output = Sum(output[i]) + bias; [2]

Câu 1: Trong chương trình Python, viết 1 đoạn code tạo 1 nơ-ron với 3 ngõ vào, 3 trọng số tương ứng và 1 giá trị bias. Tính ngõ ra tương ứng và kiểm tra tính chính xác của gía trị ngõ ra.

Code mẫu:

```
# A single neuron with 3 inputs
input = [1, 2, 3]
weight = [0.2, 0.8, -0.5]
bias = 2
output = (input[0]*weight[0]+input[1]*weight[1]+input[2]*weight[2]+bias)
print(output)
```

Câu 2: Sử dụng hàm random để tạo giá trị trọng số và bias ngẫu nhiên.

Câu 3: Tính giá trị ngõ ra sử dụng hàm dot product của nympy.

Một mạng thần kinh nhân tạo được tạo ra bởi nhiều lớp nơ-ron. Mỗi lớp nơ-ron gồm nhiều nơ-ron, với nguyên tắc tính toán giống như đã trình bày ở phương trình [1] & [2].

Câu 4: Trong chương trình Python, viết 1 đoạn code tạo 1 lớp nơ-ron với 3 nơ-ron, 3 ngõ vào, tổng công 12 giá trị trọng số, được sắp xếp dưới dạng list (danh sách) gồm 3 vec-tơ trọng số, và 1 giá trị bias. Tính ngõ ra tương ứng và kiểm tra tính chính xác của giá trị ngõ ra.

Câu 5: Thực hiện yêu cầu của câu 3, sử dụng hàm tính dot_product của numpy và khai báo 12 trọng số dưới dạng ma trận. Kiểm tra tính chính xác của giá trị ngõ ra.

Câu 6: Trong chương trình Python, viết 1 đoạn code tạo 2 lớp nơ-ron với 3 nơ-ron mỗi lớp, 4 ngõ vào. Tính giá trị của vec-tơ ngõ ra (output vector). Kiểm tra tính chính xác của giá trị ngõ ra.
Câu 7: Nhập dữ liệu ngõ vào dưới dạng batch. Quan sát giá trị ngõ ra, nhận xét và kết luận.
Câu 8: *Viết* 1 class Dense_Layer cho phép khởi tạo 1 lớp nơ-ron, với cấu trúc như sau:

```
class Dense_Layer:

    def __init__(self, n_inputs, n_neurons):
        #init eights and biases
        self.weights = ...
        self.biaises = ...

    def forward(self,inputs):
        #calculate outputs
        self.output = …
```

Câu 9: Viết 1 chương trình python, trong đó nhập lệnh khởi tạo 1 lớp nơ-ron. Và chạy hàm forward để tính giá trị ngõ ra.
Câu 10: Viết 1 chương trình python, trong đó nhập lệnh khởi tạo 2 lớp nơ-ron. Chạy hàm forward để tính giá trị ngõ ra. Kết luận.