TUGAS BESAR BAGIAN A
IF3150 MACHINE LEARNING SEMESTER 2 2021/2022

Anggota:

Kelompok 02 K-01

Karlsen Adiyasa Bachtiar	13519001
Yudi Alfayat	13519051
Almeiza Arvin Muzaki	13519066
Roy H Simbolon	13519068

Tanggal Pengumpulan: 05 Maret 2022

TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

A. Implementasi Program

- Models
  Merupakan file eksternal berupa file json yang menyimpan sebuah model yang berisi layer, dan setiap layer memiliki activation(fungsi aktivasi), numOfNode (nomor node), weight dan value layer.

```
{
        "model": {
                "layers": [
                        {
                                "activation": "sigmoid",
                                "numOfNode": 2,
                                "weights": [],
                                "values": []
                        },
```

- Components:
  - Activation
    Merupakan class yang berisi method-method fungsi aktivasi yang akan dipanggil pada method active sesuai dengan fungsi aktivasi yang sesuai dengan yang diperlukan. Method yang terdapat pada class aktivasi adalah method linear, sigmoid, relu dan softmax. Fungsi-fungsi aktivasi merupakan method static.

```
class Activation:
    @staticmethod
    def linear(x) :
        return x

    @staticmethod
    def sigmoid(x):
        value = 1 / (1 + math.exp(x*(-1)))
        return round(value)

    @staticmethod
    def relu(x):
        return max(0.0, x)

    @staticmethod
    def softmax(x):
        return x
```

- FFNN
  Merupakan class yang berisi inisialisasi inputan berupa list of integer, method solve yang menghasilkan value tiap layer sesuai dengan fungsi aktivasi yang dipanggil.

```
def setInitialInput(self, input: list[int]):
    if (len(self.layers) >= 1):
        self.layers[0].values = input

def solve(self):
    for i in range(len(self.layers)-1):
        idxLayerInput = i
        idxLayerTarget = i+1
        targetValues = []
        for nodeTarget in range(self.layers[idxLayerTarget].numOfNode):
            sigma = self.getSigma(idxLayerTarget, self.layers[idxLayerInput].values, nodeTarget)
            # print(sigma)
            tempVal = Activation.active(sigma, self.layers[idxLayerTarget].activation)
            targetValues.append(tempVal)

        self.layers[idxLayerTarget].values = targetValues
        # print(targetValues)
    return self.layers[len(self.layers)-1].values
```

Pada class FFNN, terdapat juga method getSigma yang menghasilkan penjumlahan weight sebuah layer, dan method getModelFromFile yang menghasilkan model dari sebuah inputan file.

```
def getSigma(self, idxLayer, input, node):
    temp = 0
    for idx, w in enumerate(self.layers[idxLayer].weights[node]):
        if (idx != 0):
            temp += w * input[idx-1]
        else:
            temp += w
    return temp

@staticmethod
def getModelFromFile (path):
    MODELS_DIR = "models/"
    layers: list[Layer] = []
    df = pd.read_json(MODELS_DIR + path)
    for layer in df.model.layers :
        layer_ = Layer(layer['activation'], layer['numOfNode'], layer['weights'], layer['values'])
        layers.append(layer_)

    return FFNNModel(layers)
```

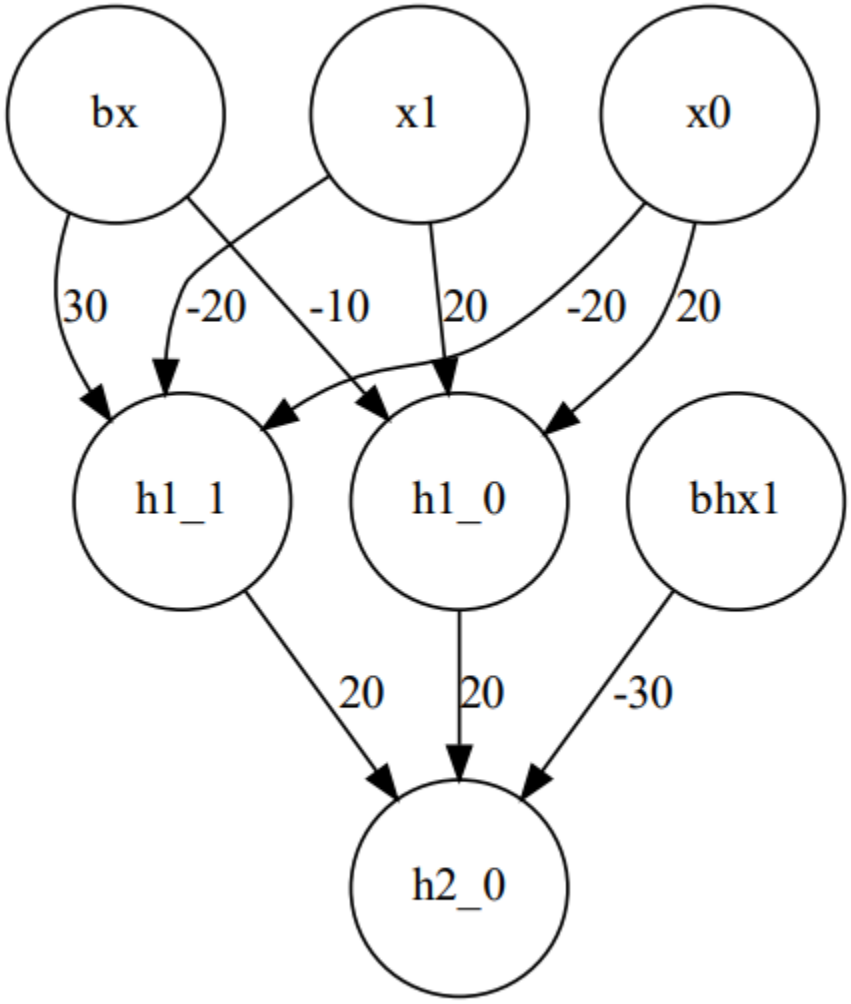
- Layer
  Merupakan class yang menginisialisasi sebuah class layer.

```
class Layer :
    def __init__(self, activation, numOfNode, weights, value) :
        self.activation : str = activation
        self.numOfNode : int = numOfNode
        self.weights : list[list[int]] = weights
        self.values : list[int] = value
```

- Main
  Merupakan driver yang akan memanggil method-method components untuk melakukan implementasi pembelajaran mesin forward propagation pada FFNN. Pada main, terdapat juga visualisasi untuk menampilkan model berupa struktur dan koefisiennya.

B. Hasil Pengujian

- Menampilkan model berupa struktur dan koefisiennya



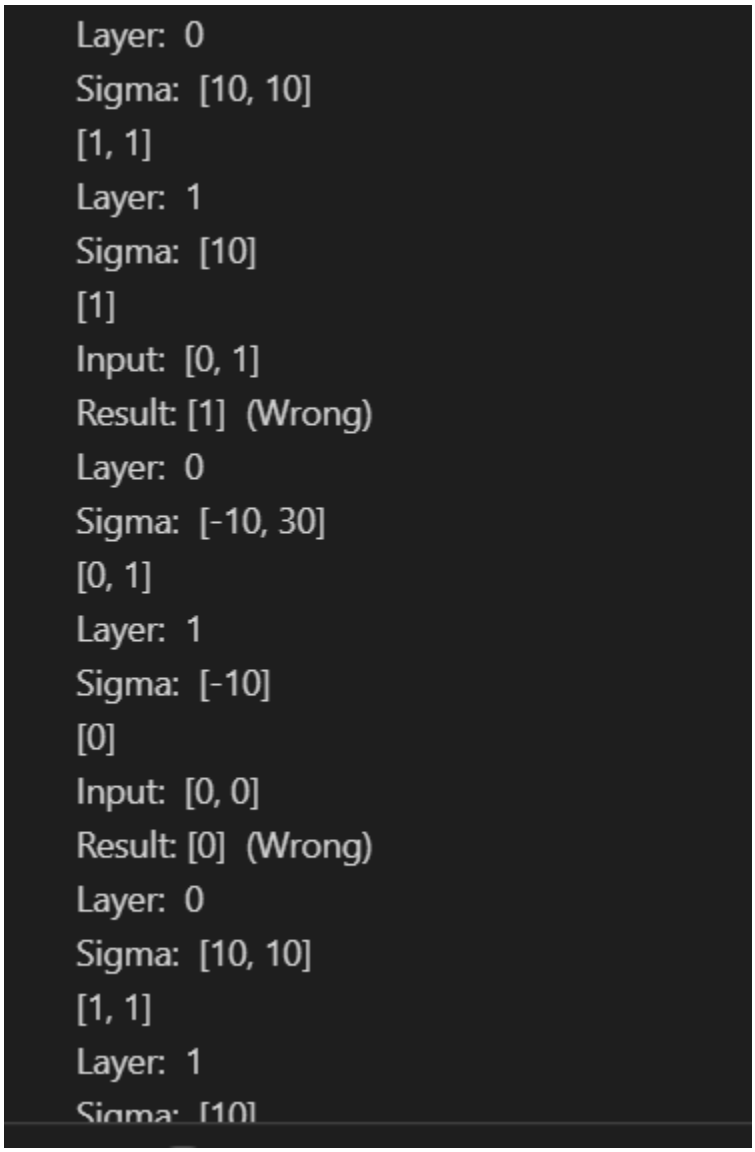
Gambar Neuron Model (Sigmoid)

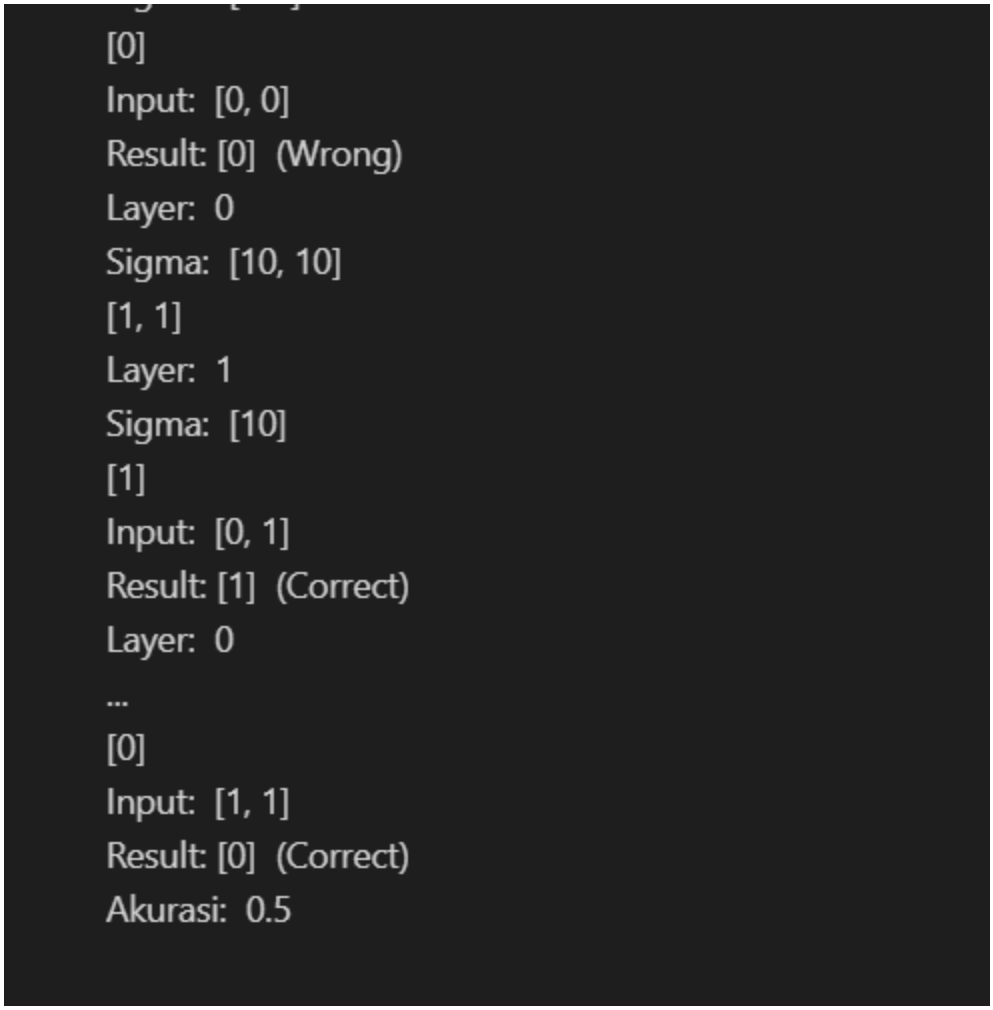
*simpul dengan huruf depan b merupakan bias

- Memprediksi output untuk input 1 instance. Pengujian dilakukan dengan kedua model XOR dari slide kuliah.
  Seperti yang dapat dilihat pada gambar di bawah, hasil yang didapat dengan menggunakan program sudah betul dan sesuai dengan targetnya

- Memprediksi output untuk input batch sejumlah instances. Pengujian dilakukan dengan kedua model XOR dari slide kuliah.

Seperti yang dapat dilihat pada gambar di bawah, hasil yang didapat untuk setiap input dengan menggunakan program sudah betul dan sesuai dengan targetnya





C. Perbandingan dengan Hasil Perhitungan Manual

x0	x1	x2	f	∑h1	h1	∑h2	h2	∑y	y
1	0	0	0	-10	0.00	30	1.00	-10.00	0.00
1	0	1	1	10	1.00	10	1.00	10.00	1.00
1	1	0	1	10	1.00	10	1.00	10.00	1.00
1	1	1	0	30	1.00	-10	0.00	-10.00	0.00

Perbandingan dengan hasil program kami menunjukkan hasil yang sama dengan hasil perhitungan secara manual.

D.Pembagian Tugas

NIM	Pekerjaan
13519001	Laporan, membaca model, visualisasi data, membuat kelas layer dan activation
13519051	Laporan, merancang dan membuat model, membuat kelas FFNNModel,
13519066	Laporan, konversi file csv ke model array agar dapat dibaca program
13519068	Laporan, menampilkan output dari prediksi dan meng-output nilai-nilai pada setiap layer