

**TUGAS BESAR A**  
**IF3270 Pembelajaran Mesin**

**Implementasi Forward Propagation untuk**  
**Feed Forward Neural Network**

**Semester II**  
**Tahun Ajaran 2021/2022**



Disusun oleh :

Ferdy Irawan Firdaus	13519030
Ridho Daffasyah	13519038
Thomas Ferdinand Martin	13519099
Muhammad Rifky Muthahhari	13519123

**TEKNIK INFORMATIKA**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2022**

## A. Implementasi

Library yang digunakan adalah numpy dan graphviz. Numpy digunakan untuk membantu melakukan operasi matriks dan operasi matematika lain. Graphviz digunakan untuk membantu memvisualisasikan model yang telah dibangun.

Format dari input :

Input	Keterangan
1	jumlah_instance
2 2 1	jumlah_node_input jumlah_node_layer_n
0 0	Input
sigmoid	Jenis aktivasi layer n
-10 30	Bobot bias
20 -20	Bobot i
20 -20	Bobot i+1
sigmoid	Jenis aktivasi layer n+1
-30	Bobot bias
20	Bobot i
20	Bobot i+1

Kelas yang didefinisikan :

1. Activation  
Berisikan konstan jenis fungsi aktivasi.
2. Layer  
Berisikan berbagai atribut yang harus dimiliki sebuah layer pada FFNN yaitu weights, node input, node per layer, dan activations yang dipilih.
3. FFNN  
Berisikan berbagai fungsi dan beberapa atribut yang digunakan untuk perhitungan FFNN itu sendiri seperti layer yang merupakan kelas Layer dan matriks sesuai dengan isi file input.
  - a. showLayer  
Fungsi ini menampilkan layer dari model.
  - b. forward\_propagation  
Fungsi ini melakukan forward propagation dari input dan struktur model yang dibangun.
  - c. sigmoid

Fungsi aktivasi sigmoid.

d. relu

Fungsi ini merupakan fungsi aktivasi ReLU menerima input berupa sebuah matriks x, kemudian setiap nilai dalam matriks yang lebih kecil dari 0 akan diubah nilainya menjadi 0 atau dalam program yang telah dibuat, pengubahan nilai dilakukan dengan memanfaatkan  $\max(x, 0)$ .

e. linear

Fungsi aktivasi linear.

f. softmax

Fungsi softmax adalah salah satu fungsi aktivasi. Fungsi ini menerima input matriks x lalu dilakukan penjumlahan fungsi eksponensial semua

anggota matriks x ( $\sum_{i=1}^k e^i$ ) dengan i atau pangkat dari e merupakan anggota

matriks x, hasil ini akan menjadi penyebut (*denominator*) dari fungsi aktivasi softmax. Untuk pembilangnya (*numerator*) diperoleh dengan menghitung fungsi eksponensial  $f(x_i) = e^{x_i}$  dengan  $x_i$  merupakan anggota matriks x (tidak dijumlahkan seperti penyebut). Sehingga *output* dari fungsi ini adalah *numerator*  $\div$  *denominator*, perhitungan ini diulang sebanyak anggota matriks x.

g. show\_output

Fungsi ini menampilkan output hasil prediksi

h. draw\_model

Fungsi ini digunakan untuk menampilkan bentuk model dalam format pdf. Model ditampilkan dalam bentuk graf berarah dengan sisi yang diberi label berupa bobot dan simpul menyatakan node dalam layer. Penggambaran model dilakukan menggunakan library graphviz

## B. Hasil Pengujian

a. Model XOR dengan fungsi aktivasi sigmoid

```
A. Input 1 instance
----- Layer: 0 -----
Input 0: [0.0, 0.0]
----- Layer: 1 -----
Activation : sigmoid
Input-0 ==>
Node 1
Sigma : -10.0
Result : 0.0
Node 2
```

Sigma : 30.0  
Result : 1.0  
----- Layer: 2 -----  
Activation : sigmoid  
Input-0 ==>  
Node 1  
Sigma : -10.0  
Result : 0.0  
Prediction  
[[0.]]

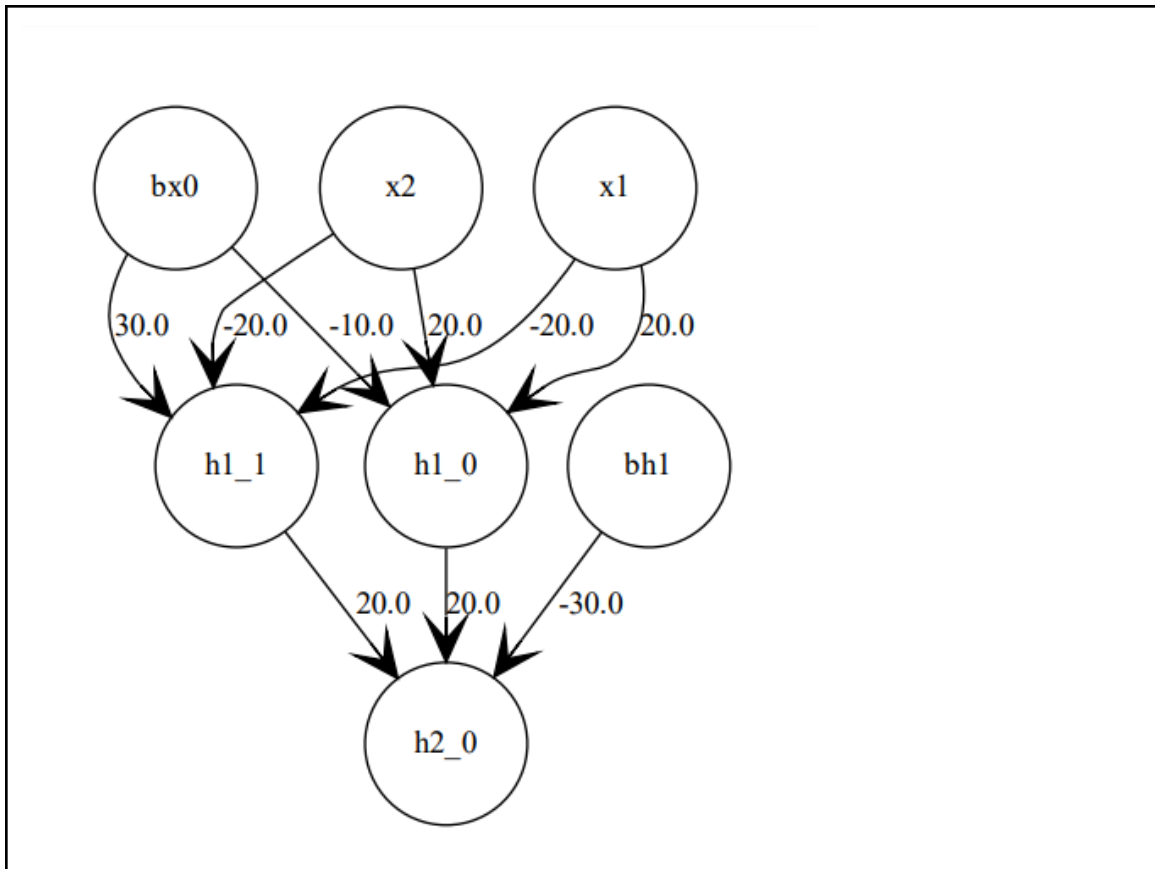
B. Input batch instance

----- Layer: 0 -----  
Input 0: [0.0, 0.0]  
Input 1: [0.0, 1.0]  
Input 2: [1.0, 0.0]  
Input 3: [1.0, 1.0]  
----- Layer: 1 -----  
Activation : sigmoid  
Input-0 ==>  
Node 1  
Sigma : -10.0  
Result : 0.0  
Node 2  
Sigma : 30.0  
Result : 1.0  
Input-1 ==>  
Node 1  
Sigma : 10.0  
Result : 1.0  
Node 2  
Sigma : 10.0  
Result : 1.0  
Input-2 ==>  
Node 1  
Sigma : 10.0  
Result : 1.0  
Node 2  
Sigma : 10.0  
Result : 1.0  
Input-3 ==>  
Node 1  
Sigma : 30.0  
Result : 1.0  
Node 2  
Sigma : -10.0

```
Result : 0.0
----- Layer: 2 -----
Activation : sigmoid
Input-0 ==>
Node 1
Sigma : -10.0
Result : 0.0
Input-1 ==>
Node 1
Sigma : 10.0
Result : 1.0
Input-2 ==>
Node 1
Sigma : 10.0
Result : 1.0
Input-3 ==>
Node 1
Sigma : -10.0
Result : 0.0
Prediction
[[0.]
 [1.]
 [1.]
 [0.]]
```

Didapatkan hasil prediksi dengan accuracy 100%

Visualisasi model :



b. Model XOR dengan fungsi aktivasi ReLU

A. Input 1 instance

----- Layer: 0 -----

Input 0: [0.0, 1.0]

----- Layer: 1 -----

Activation : relu

Input-0 ==>

Node 1

Sigma : 1.0

Result : 1.0

Node 2

Sigma : 0.0

Result : 0.0

----- Layer: 2 -----

Activation : relu

Input-0 ==>

Node 1

Sigma : 1.0

Result : 1.0

Prediction

[[1.]]

B. Input batch instances

----- Layer: 0 -----

Input 0: [0.0, 0.0]

Input 1: [0.0, 1.0]

Input 2: [1.0, 0.0]

Input 3: [1.0, 1.0]

----- Layer: 1 -----

Activation : relu

Input-0 ==>

Node 1

Sigma : 0.0

Result : 0.0

Node 2

Sigma : -1.0

Result : 0.0

Input-1 ==>

Node 1

Sigma : 1.0

Result : 1.0

Node 2

Sigma : 0.0

Result : 0.0

Input-2 ==>

Node 1

Sigma : 1.0

Result : 1.0

Node 2

Sigma : 0.0

Result : 0.0

Input-3 ==>

Node 1

Sigma : 2.0

Result : 2.0

Node 2

Sigma : 1.0

Result : 1.0

----- Layer: 2 -----

Activation : relu

Input-0 ==>

Node 1

Sigma : 0.0

Result : 0.0

Input-1 ==>

Node 1

Sigma : 1.0

Result : 1.0

Input-2 ==>

Node 1

Sigma : 1.0

Result : 1.0

Input-3 ==>

Node 1

Sigma : 0.0

Result : 0.0

Prediction

[[0.]

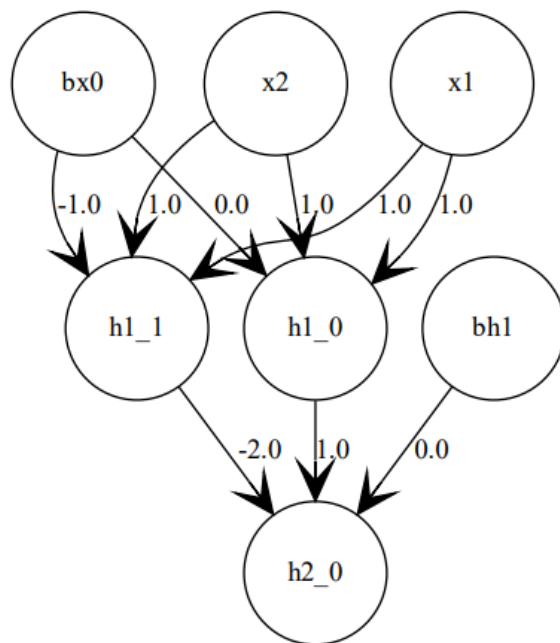
[1.]

[1.]

[0.]]

Didapatkan hasil prediksi dengan accuracy 100%

Visualisasi model :



### C. Perbandingan Hasil dengan Perhitungan Manual

a. Model 1

Perhitungan secara manual:



x1	x2	kelas	sigma_h1	h1	sigma_h2	h2	sigma_y	y
0,00	0,00	0,00	-10,00	0,00	30,00	1,00	-10,00	0,00
0,00	1,00	1,00	10,00	1,00	10,00	1,00	10,00	1,00
1,00	0,00	1,00	10,00	1,00	10,00	1,00	10,00	1,00
1,00	1,00	0,00	30,00	1,00	-10,00	0,00	-10,00	0,00

Didapatkan hasil prediksi yang sama antara perhitungan manual dengan perhitungan menggunakan program.

b. Model 2

Perhitungan secara manual:

x1	x2	kelas	sigma_h1	h1	sigma_h2	h2	sigma_y	y
0,00	0,00	0,00	0,00	0,00	-1,00	0,00	0,00	0,00
0,00	1,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00
1,00	0,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00
1,00	1,00	0,00	2,00	2,00	1,00	1,00	0,00	0,00

Didapatkan hasil prediksi yang sama antara perhitungan manual dengan perhitungan menggunakan program.

#### D. Pembagian Tugas

NIM>Nama	Pembagian
13519030/Ferdy Irawan Firdaus	Fungsi Sigmoid, fungsi softmax
13519038/Ridho Daffasyah	Kelas FFNN, kelas Layer
13519099/Thomas Ferdinand Martin	Fungsi ReLU, fungsi draw_model
13519123/Muhammad Rifky Muthahhari	Membaca file, fungsi forward propagation, fungsi linier