

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

Tugas Besar 1 Bagian A IF3270 Pembelajaran Mesin



Kelas 2 dan 3

Primanda Adyatma Hafiz	13520022
Vincent Prasetya Atmadja	13520099
Gerald Abraham Sianturi	13520138
Daffa Romyz Aufa	13520162

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

TEKNIK INFORMATIKA

2023

Penjelasan Implementasi

No	Bagian	Penjelasan
1	Fungsi aktivasi	Terdapat empat fungsi aktivasi yang diimplementasikan, yakni fungsi aktivasi linear, ReLU, sigmoid, dan softmax. Keempat fungsi aktivasi disimpan dalam <i>dictionary</i> agar fungsi aktivasi dapat dipanggil dengan <i>string identifier</i> yang ditentukan
2	Kelas	Secara umum, <i>design pattern</i> yang digunakan <i>composite</i> , dimana model ANN disusun dari beberapa <i>layer</i> , dan tiap layer tersusun atas beberapa <i>node</i>
	Node	Digunakan sebagai representasi neuron pada ANN, menyimpan properti <ul style="list-style-type: none"> - <code>id</code>, untuk identifikasi Node - <code>layerId</code>, untuk menyimpan <i>id</i> dari Layer yang berkorespondensi dengan Node - <code>weight</code>, - <code>bias</code>, - <code>activFunctionType</code>, Terdapat metode untuk menghitung <i>output</i> berdasarkan masukan dari <i>layer</i> sebelumnya.
	Layer	Digunakan sebagai representasi <i>layer</i> pada ANN, menyimpan properti <ul style="list-style-type: none"> - <code>id</code>, untuk identifikasi Layer - <code>nodeAmount</code> untuk menyimpan jumlah Node - <code>layerType</code> untuk menyimpan apakah Layer merupakan <i>input layer</i>, <i>hidden layer</i>, atau output layer, - <code>nodes</code> untuk menyimpan kumpulan Node pada layer. Terdapat metode untuk menambahkan <i>node</i> baru pada <code>nodes</code>
	ANNModel	Digunakan sebagai representasi model ANN, menyimpan properti <code>layers</code> untuk menyimpan kumpulan Layer yang menyusun model ANN. Terdapat metode untuk menghasilkan nilai dari <i>output layer</i> .
3	File loader	Digunakan untuk melakukan <i>setup</i> model ANN dengan struktur file text sebagai berikut: Line ke-1:

		<ul style="list-style-type: none"> - Elemen ke-1: Banyaknya <i>layer</i> - Elemen ke-2: Banyaknya node pada input layer - Elemen ke-3 s.d. kedua dari terakhir: Banyaknya node pada hidden layer ke-n - Elemen terakhir: Banyaknya node pada <i>output layer</i> <p>Line ke-2 s.d dst: (Khusus <i>node</i> pada <i>hidden</i> dan <i>output layer</i> yang memiliki <i>weight</i> dan <i>bias</i>)</p> <ul style="list-style-type: none"> - Elemen ke-1 s.d ketiga dari terakhir: <i>Weight</i> dari <i>node</i> - Elemen kedua dari terakhir: <i>Bias</i> dari <i>node</i> - Elemen terakhir: Jenis <i>activation function</i> dari <i>node</i>
4	Fungsi representasi model (gambar graf dan teks)	<p>Memanfaatkan <i>library</i> eksternal <code>networkx</code> dan <code>matplotlib</code> untuk memvisualisasikan ANN dalam bentuk graf.</p> <p>Untuk merepresentasikan data dalam bentuk teks juga dapat digunakan metode <code>describe</code> pada kelas <code>ANNModel</code></p>

Hasil pengujian

1. Model pada `model1.txt`

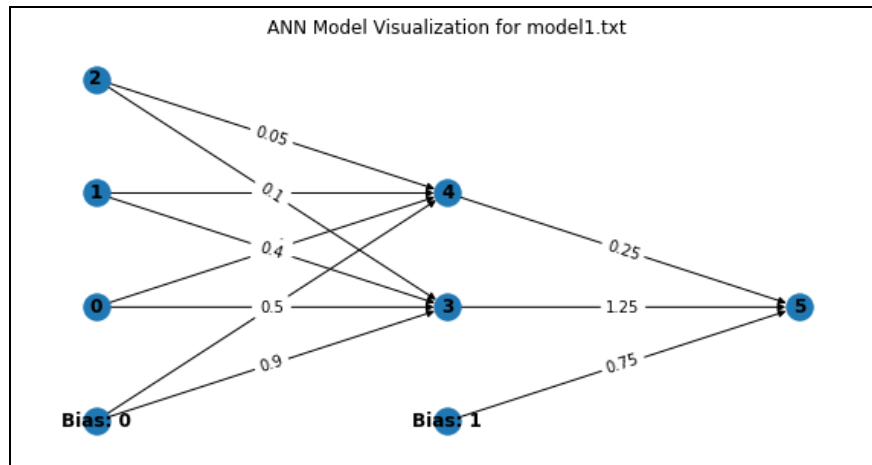
```
3 3 2 1
0.5 0.4 0.1 0.9 Sigmoid
0.25 0.2 0.05 0.8 Sigmoid
1.25 0.25 0.75 Sigmoid
```

File ini memiliki interpretasi bahwa model memiliki tiga *layer*, dimana *layer* pertama (*input node*) memiliki tiga buah *node*, *hidden layer* pertama memiliki dua buah *node*, dan *layer* ketiga memiliki 1 *output node*.

- Node pertama pada *hidden layer* memiliki weight 0.5, 0.4, dan 0.1, bias 0.9, serta fungsi aktivasi dengan tipe Sigmoid
- Node kedua pada *hidden layer* memiliki weight 0.25, 0.2, dan 0.05, bias 0.8, serta fungsi aktivasi dengan tipe Sigmoid
- Node pada *output layer* memiliki weight 1.25 dan 0.25, bias 0.75, serta fungsi aktivasi dengan tipe Sigmoid

2. Representasi model

- a. Dalam bentuk graf



b. Dalam bentuk teks

```

>>> annModelInstance.describe()
Layer with id 0:
  Amount of node -> 3
  Layer type -> LayerType.Input
Node with id 0:
  Layer id -> 0
  Weight -> None
  Bias -> None
  Activation function -> None
Node with id 1:
  Layer id -> 0
  Weight -> None
  Bias -> None
  Activation function -> None
Node with id 2:
  Layer id -> 0
  Weight -> None
  Bias -> None
  Activation function -> None

Layer with id 1:
  Amount of node -> 2
  Layer type -> LayerType.Hidden
Node with id 3:
  Layer id -> 1
  Weight -> [0.5, 0.4, 0.1]
  Bias -> 0.9
  Activation function -> Sigmoid
Node with id 4:
  Layer id -> 1
  Weight -> [0.25, 0.2, 0.05]
  Bias -> 0.8
  Activation function -> Sigmoid

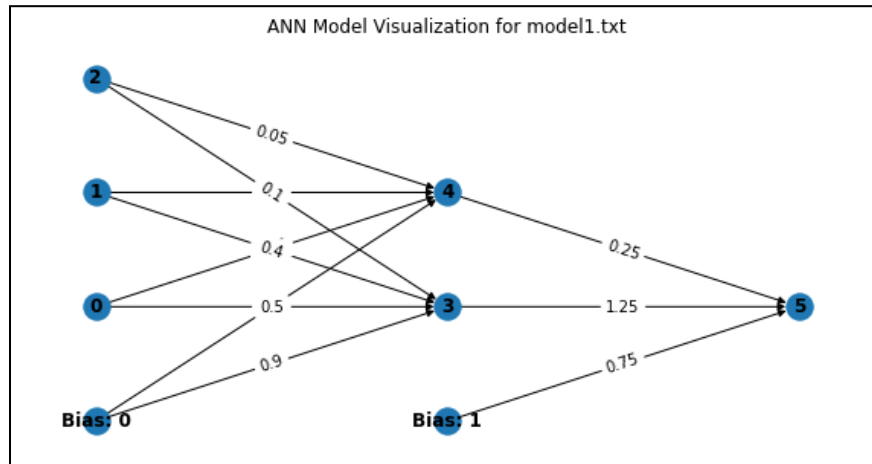
Layer with id 2:
  Amount of node -> 1
  Layer type -> LayerType.Output
Node with id 5:
  Layer id -> 2
  Weight -> [1.25, 0.25]
  Bias -> 0.75
  Activation function -> Sigmoid
  
```

3. Nilai dari *output layer*

```

>>> annModelInstance.predict([2, 1, 1])
[0.89]
  
```

Perbandingan dengan perhitungan manual



Angka pada kolom paling atas merepresentasikan *node* dengan id terkait, berikut hasil perhitungan manual dengan fungsi aktivasi semua *node* menggunakan fungsi sigmoid

0	1	2	$\Sigma 3$	3	$\Sigma 4$	4	$\Sigma 5$	5
2	1	1	(2×0.5) $+ (1 \times 0.4)$ $+ (1 \times 0.1)$ $+ 0.9 = 2.4$	$\sigma(2.4)$ $= 0.916$	(2×0.25) $+ (1 \times 0.2)$ $+ (1 \times 0.05)$ $+ 0.8 = 1.55$	$\sigma(1.55)$ $= 0.825$	0.916×1.25 $+ (0.825 \times 0.25)$ $+ 0.75 = 2.1$	$\sigma(2.1)$ $= 0.89$

Dapat dilihat bahwa hasil *output layer* pada hasil pengujian dengan metode `predict` pada program sama dengan nilai dengan perhitungan manual.

Pembagian tugas

NIM	Nama	Bagian pengerjaan
13520022	Primanda Adyatma Hafiz	Fungsi aktivasi
13520099	Vincent Prasetya Atmadja	Kelas yang digunakan
13520138	Gerald Abraham Sianturi	<i>File loader</i>
13520162	Daffa Romyz Aufa	Visualisasi graf dan teks