

ALGORITMA KLASIFIKASI : ARTIFICIAL NEURAL NETWORK

1. Tujuan

1. Mahasiswa mampu menggunakan metode Artificial Neural Network (ANN).
2. Mahasiswa mampu menerapkan ANN untuk kasus-kasus nyata.

2. Landasan Teori

Jaringan Syaraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa :

- 1) Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron)
- 2) Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung
- 3) Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal
- 4) Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

JST ditentukan oleh 3 hal yaitu :

- 1) Pola hubungan antar neuron (disebut arsitektur jaringan)
- 2) Metode untuk menentukan bobot penghubung (disebut metode training / learning / algoritma)
- 3) Fungsi aktivasi

Aplikasi Jaringan Syaraf Tiruan :

- 1) Pengenalan pola (pattern recognition)

Jaringan syaraf tiruan dapat dipakai untuk mengenali pola (misal : huruf, angka, suara atau tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpainya (mungkin wajah / bentuk tubuhnya sudah sedikit berubah).

2) Signal processing

Jaringan syaraf tiruan (model ADALINE) dapat dipakai untuk menekan noise dalam saluran telepon.

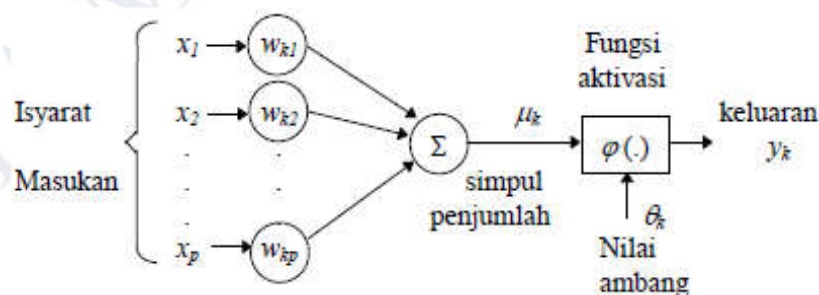
3) Peramalan

Jaringan syaraf tiruan juga dapat dipakai untuk meramalkan apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang ada dimasa yang lampau. Ini dapat dilakukan mengingat kemampuan jaringan syaraf tiruan untuk mengingat dan membuat generalisasi dari apa yang sudah ada sebelumnya.

Kelemahan Jaringan Syaraf Tiruan :

- 1) Ketidak akuratan hasil yang diperoleh.
- 2) Jaringan syaraf tiruan bekerja berdasarkan pola yang terbentuk pada inputnya.

Dalam merancang suatu jaringan syaraf selain memperhatikan struktur hubungan antara simpul masukan dengan simpul keluaran, perlu ditentukan juga cara atau metode pembelajarannya. Belajar bagi jaringan syaraf adalah cara memperbaharui bobot sinapsis disesuaikan dengan isyarat masukan dan keluran yang diharapkan. Secara umum suatu jaringan syaraf dibentuk atas sejumlah neurons sebagai unit pengolah informasi sebagai dasar operasi untuk menjalankan fungsi atau tugasnya. Model susunan neurons ditunjukkan pada Gambar 1 berikut ini.



Gambar 1. Model neurons k tak-linear

Dari model neurons untuk jaringan syaraf seperti terlihat pada Gambar 1 terdapat tiga bagian utama berikut :

- 1) Sekumpulan sinapsis atau jalur penghubung, yang mempunyai bobot atau kekuatan. Indeks yang tertulis pada notasi bobot sinapsis menunjukkan posisi masukan dan keluaran yang dihubungkan.

- 2) Unit penjumlahan untuk menjumlahkan isyarat masukan.
- 3) Fungsi aktivasi untuk membatasi amplitud keluaran.

Beberapa model neuron dalam jaringan saraf tiruan antara lain :

- 1) Neuron McCulloch – Pitts
- 2) Neuron Hebb
- 3) Neuron Perceptron
- 4) Neuron ADALINE

Sebagai catatan yang perlu diperhatikan adalah dalam model jaringan saraf tiruan, semua nilai kelas atribut berupa sebuah array yang bertipe numerik. Artinya semua data berupa bilangan baik bilangan biner, octal, maupun decimal. Jaringan saraf tiruan tidak dapat mengenali data yang bertipe character / string. Apabila terdapat suatu data yang bertipe string, maka perlu dikonversi menjadi tipe numerik. Sebagai contoh: jenis kelamin terdiri atas Pria dan Wanita, maka perlu diubah menjadi tipe bilangan biner yaitu 0 dan 1.

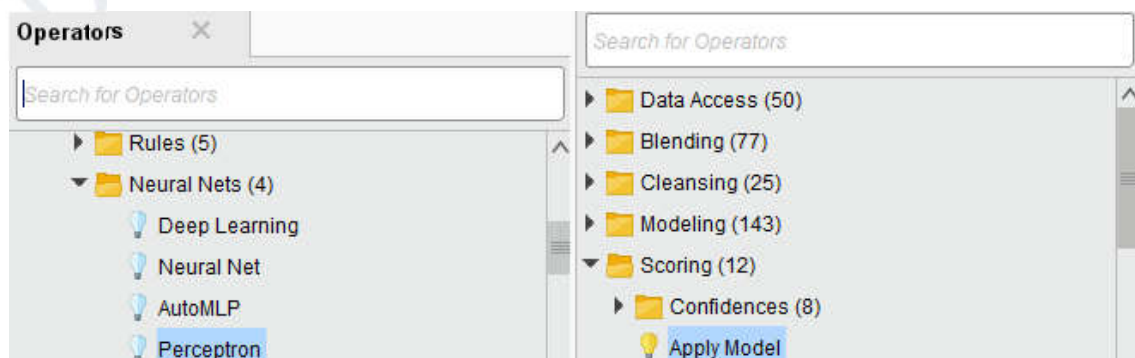
3. Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi RapidMiner.
3. Modul Praktikum Data Warehousing dan Data Mining.

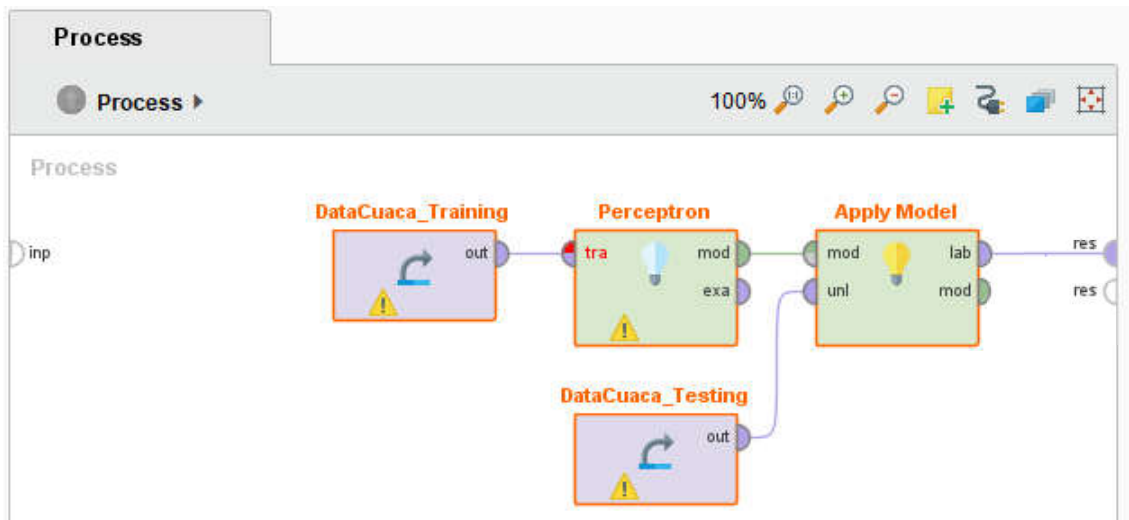
4. Langkah-langkah Praktikum


4.1. Prediksi Nilai Kelas Atribut dengan Neuron Perceptron

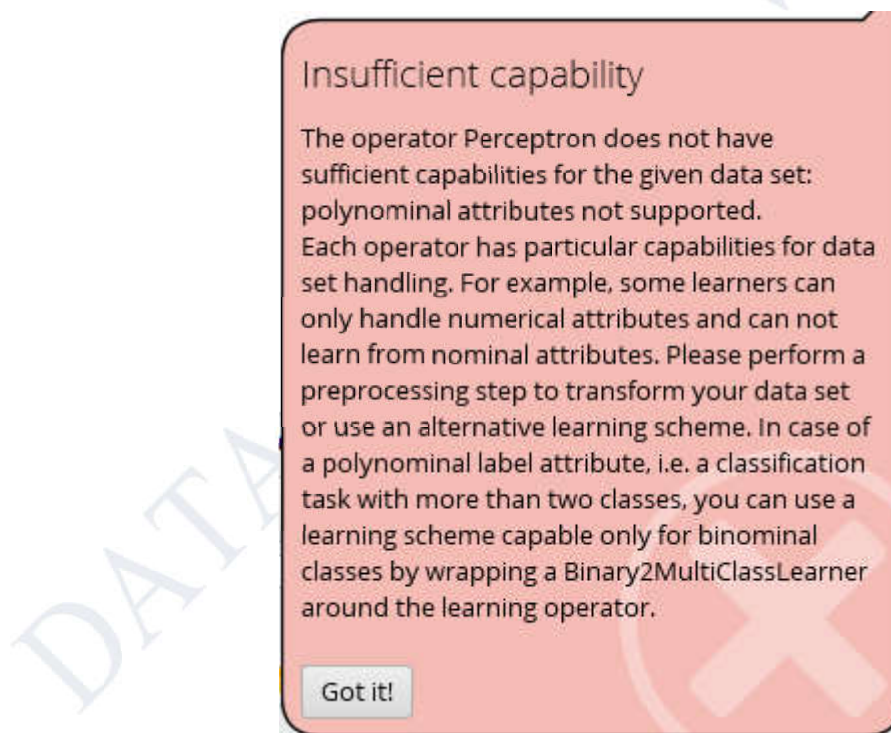
1. Jalankan aplikasi RapidMiner >> New Process.
2. Gunakan DataCuaca_Training sebagai data pelatihan dan DataCuaca_Testing sebagai data uji. Drag masing-masing data dari repository ke area Process View.
3. Drag dan masukkan operator Perceptron dan Apply Model ke dalam area Process View.



4. Hubungkan port-port input dan output masing-masing data dan operator tampak seperti pada gambar berikut.

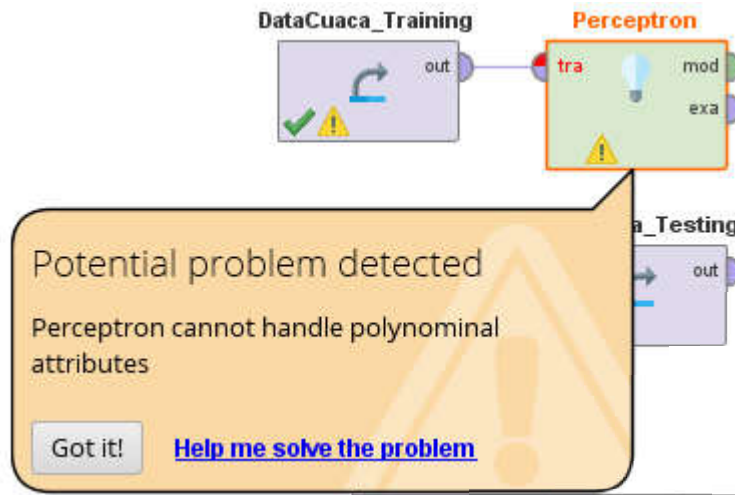


5. Jalankan proses dengan menekan tombol Run .
6. Sebuah pesan kesalahan berikut akan muncul.

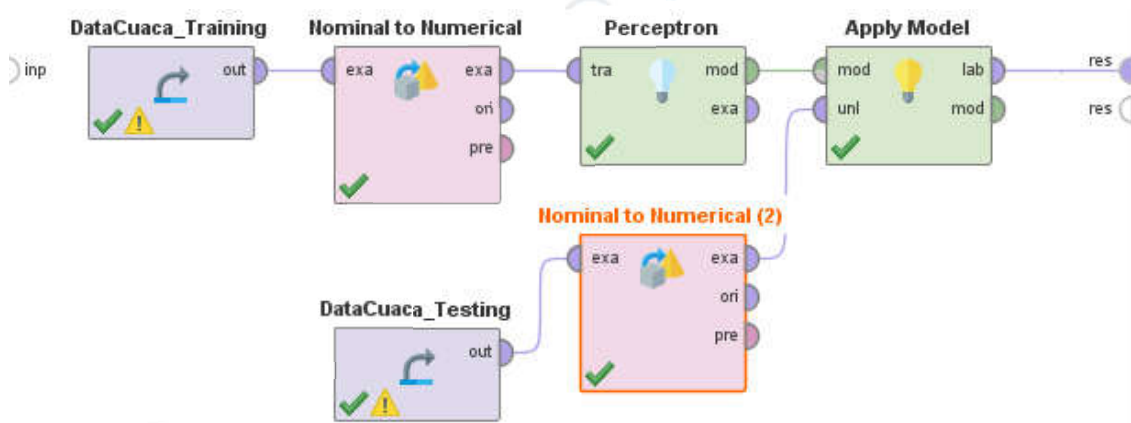


7. Pesan ini menyatakan bahwa nilai kelas data dalam data training yang bertipe polynomial tidak bisa diproses. Hal ini disebabkan karena dalam proses JST hanya bisa dilakukan terhadap data numerik. Oleh sebab itu, perlu kita tambahkan sebuah operator pengubah tipe data nominal menjadi numerik.

8. Tutup pesan kesalahan tersebut.
9. Klik tanda peringatan pada operator Perceptron. Klik ganda pada Help me solve the problem.



10. Sebuah operator Nominal to Numerical akan secara otomatis ditambahkan diantara Data training dan operator Perceptron. Tambahkan juga operator Nominal to Numerical diantara Data Testing.



11. Jalankan kembali proses dengan menekan tombol Run.
12. Akan ditunjukkan hasil prediksi data testing menggunakan JST (Perceptron)

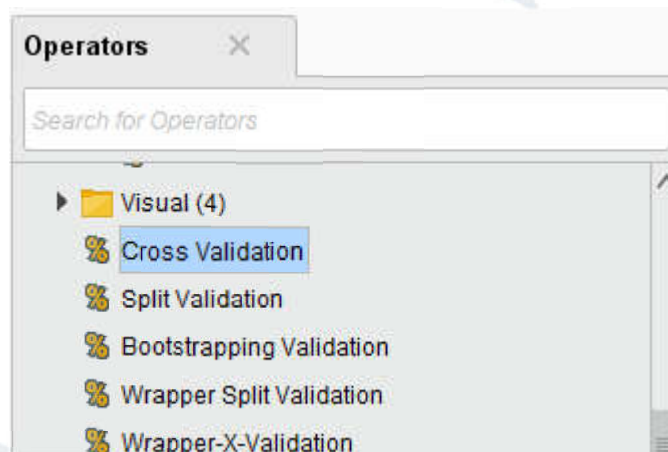
ExampleSet (Apply Model)										
ExampleSet (7 examples, 3 special attributes, 7 regular attributes)										
Filter (7 / 7 examples):										all
...	prediction(B...	confidence(Tl...	confidence(YA	Cuaca = Cer...	Cuaca = Me...	Cuaca = Huj...	berangin = T...	berangin = YA	Suhu	Kelembaban...
1	TIDAK	1.000	0.000	1	0	0	1	0	75	65
2	TIDAK	1.000	0.000	1	0	0	0	1	80	68
3	TIDAK	1.000	0.000	1	0	0	0	1	83	87
4	TIDAK	1	0	0	1	0	1	0	70	96
5	TIDAK	1.000	0.000	0	1	0	1	0	68	81
6	TIDAK	1.000	0.000	0	0	1	0	1	65	75
7	TIDAK	1	0	0	0	1	0	1	64	85

13. Hasil prediksi menunjukkan bahwa semua data akan memiliki nilai kelas Bermain_Tenis = TIDAK, berdasarkan perbandingan tingkat confidence masing-masing nilai kelas YA dan TIDAK.

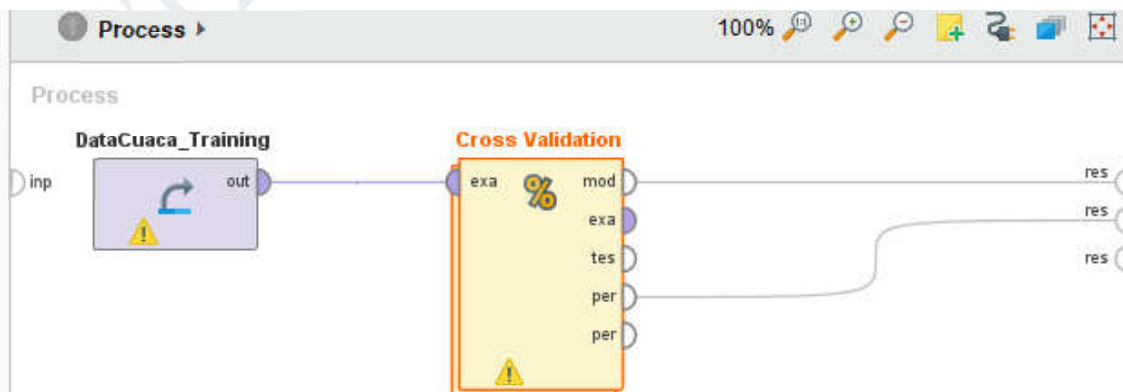
Informasi : hasil prediksi antara metode satu dengan metode lainnya tidak harus selalu sama. Tergantung dari model perhitungan, tingkat akurasi, jumlah data yang digunakan dan lain-lainnya.

4.2. Mengetahui Nilai Performance Vector pada Jaringan Saraf Tiruan

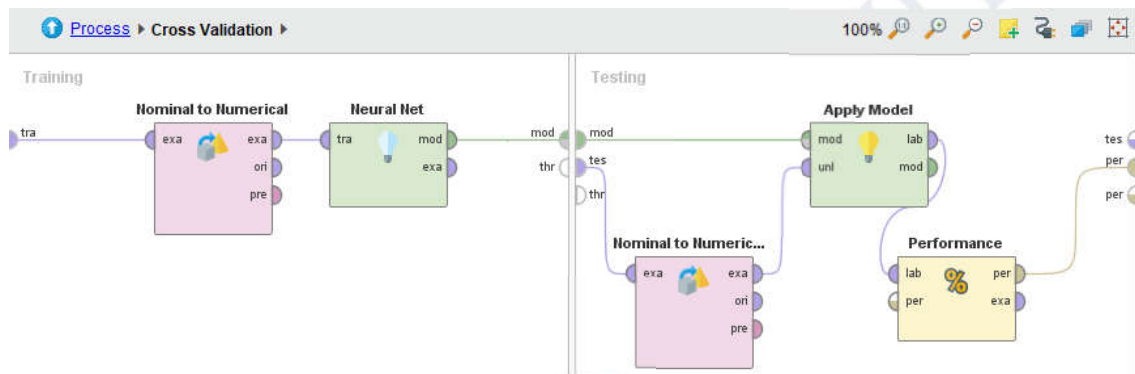
1. Masih menggunakan RapidMiner, pilih New Process.
2. Pada kegiatan ini kita hanya akan menggunakan data training saja tanpa melibatkan data testing. Tujuan dari percobaan ini untuk mengetahui performance dari data yang akan digunakan sebagai data pelatihan.
3. Gunakan DataCuaca_Training dan drag dari repository ke area Process View.
4. Drag dan masukkan operator Cross Validation ke dalam area Process View.



5. Hubungkan port output data training dengan port input cross validation.



6. Selanjutnya adalah merancang algoritma JST. Klik ganda operator Cross Validation yang terdapat pada Process View, sehingga masuk ke jendela Process – Cross Validation.
7. Masukkan operator Neural Net dalam area Training, operator Apply Model dan Performance dalam area Testing.
8. Hubungkan port input dan output masing-masing operator. Ingat bahwa dalam proses pembelajaran JST tidak dapat mengenali nilai data bertipe nominal. Data yang dikenali hanya yang bertipe numerik, sehingga anda perlu menambahkan operator Nominal to Numerical. Seperti pada gambar berikut ini.

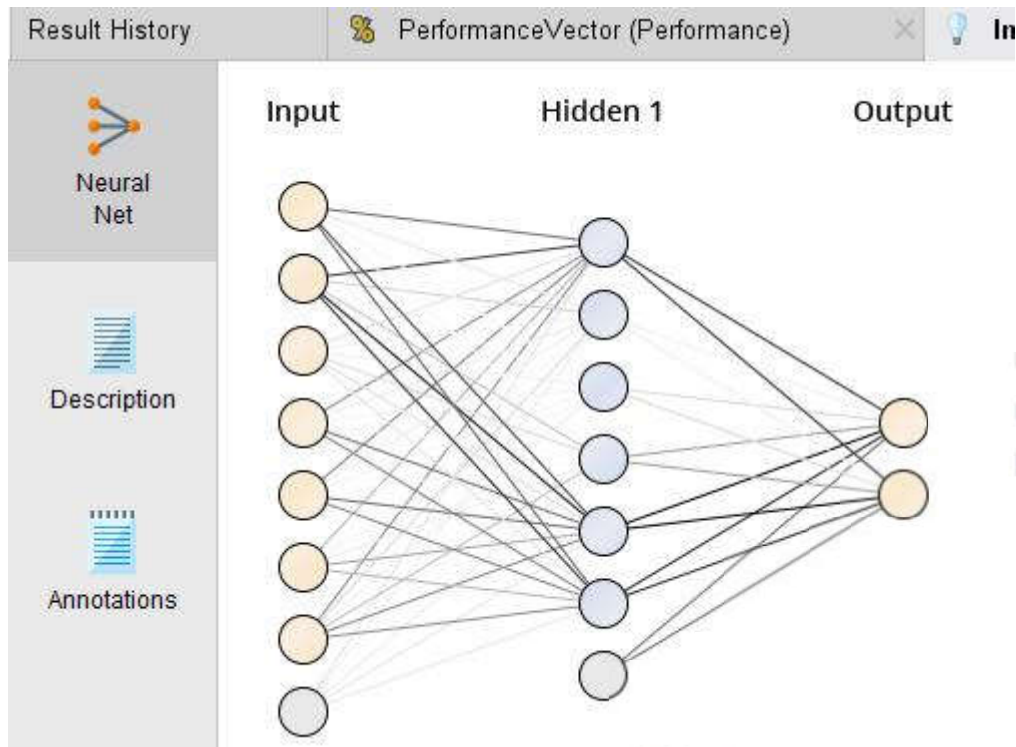


9. Tekan tombol panah ke atas atau tekan tombol Process yang terletak di atas area view untuk kembali ke desain awal.
10. Jalankan proses dengan menekan tombol Run.
11. Tab PerformanceVector (Performance), menunjukkan tingkat akurasi, presisi, recall dan lain-lain dalam bentuk tabel dan plot view.

accuracy: 50.00% +/- 44.72% (micro average: 57.14%)

	true TIDAK	true YA	class precision
pred. TIDAK	2	3	40.00%
pred. YA	3	6	66.67%
class recall	40.00%	66.67%	

12. Tab ImprovedNeuralNet (Neural Net), memperlihatkan arsitektur jaringan syaraf yang digunakan untuk proses. Ada 2 mode view yang bisa digunakan untuk melihat, yaitu :
 - a) Neural Net, digunakan untuk melihat bentuk arsitektur JST



Berdasarkan arsitektur yang terbentuk, dapat dilihat bahwa jumlah node (simpul) masing-masing layer (lapisan) sebagai berikut :

- 1) Input layer = 7 node input, dan 1 node berbobot 1
 - 2) Hidden layer = 6 node hidden, dan 1 node berbobot 1
 - 3) Output layer = 2 node (YA, TIDAK)
- b) Description, digunakan untuk melihat nilai bobot sigmoid masing-masing node (simpul) pada hidden layer dan output layer.

Berikut nilai sigmoid yang terjadi :

ImprovedNeuralNet

Hidden 1

=====

Node 1 (Sigmoid)

Cuaca = Cerah: -0.990
Cuaca = Mendung: 1.338
Cuaca = Hujan: -0.192
berangin = TIDAK: 0.742
berangin = YA: -0.794
Suhu: -0.484
Kelembaban_udara: -0.737
Bias: -0.178

Node 2 (Sigmoid)

Cuaca = Cerah: -0.157
Cuaca = Mendung: 0.313
Cuaca = Hujan: -0.092
berangin = TIDAK: 0.115
berangin = YA: -0.107
Suhu: -0.021
Kelembaban_udara: -0.235
Bias: -0.093

Node 5 (Sigmoid)

Cuaca = Cerah: -1.234
Cuaca = Mendung: 1.590
Cuaca = Hujan: -0.192
berangin = TIDAK: 0.956
berangin = YA: -0.978
Suhu: -0.598
Kelembaban_udara: -0.805
Bias: -0.151

Node 6 (Sigmoid)

Cuaca = Cerah: -1.038
Cuaca = Mendung: 1.407
Cuaca = Hujan: -0.159
berangin = TIDAK: 0.826
berangin = YA: -0.818
Suhu: -0.476
Kelembaban_udara: -0.776
Bias: -0.153

Node 3 (Sigmoid)

Cuaca = Cerah: -0.058
Cuaca = Mendung: 0.043
Cuaca = Hujan: 0.095
berangin = TIDAK: 0.066
berangin = YA: -0.070
Suhu: 0.060
Kelembaban_udara: 0.090
Bias: 0.019

Node 4 (Sigmoid)

Cuaca = Cerah: 0.047
Cuaca = Mendung: -0.372
Cuaca = Hujan: 0.229
berangin = TIDAK: -0.032
berangin = YA: 0.027
Suhu: 0.104
Kelembaban_udara: 0.402
Bias: 0.110

Output

=====

Class 'TIDAK' (Sigmoid)

Node 1: -1.370
Node 2: -0.090
Node 3: 0.253
Node 4: 0.652
Node 5: -1.742
Node 6: -1.471
Threshold: 0.979

Class 'YA' (Sigmoid)

Node 1: 1.385
Node 2: 0.106
Node 3: -0.235
Node 4: -0.662
Node 5: 1.759
Node 6: 1.456
Threshold: -0.996

5. Tugas

1. Dengan menggunakan RapidMiner, buatlah file Excel yang dikerjakan pada Tugas nomor 1 dalam Modul 6 (Pengenalan Data Mining) sebagai data training, dan file Excel pada Tugas nomor 1 dalam Modul 8 (Klasifikasi Naïve Bayes) sebagai data testing!
2. Carilah hasil prediksi terhadap data testing lama studi mahasiswa dengan menggunakan model Perceptron!
3. Dengan menggunakan Performance Vector, carilah nilai tingkat akurasi, presisi, dan recall!
4. Berdasarkan soal nomor 2, gantilah operator Perceptron menjadi Neural Net! Amati perubahan yang terjadi!
5. Berdasarkan soal nomor 4, carilah nilai tingkat akurasi, presisi, dan recall!
6. Gambarlah arsitektur jaringan saraf yang terbentuk!
7. Berapakah jumlah node (simpul) masing-masing layer (lapisan) berdasarkan arsitektur JST?
8. Tulislah nilai-nilai bobot sigmoid masing-masing node (simpul) pada hidden layer dan output layer?
9. Kesimpulan apa yang anda dapatkan dari percobaan yang anda lakukan di atas?