RANCANG BANGUN ALAT BANTU PENENTU POLA DISTRIBUSI INPUT DENGAN MEMANFAATKAN JARINGAN SYARAF TIRUAN DENGAN MENGGUNAKAN PERSENTIL SEBAGAI PENCIRI

Aris Tjahyanto
Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Tel. + 62 31 5939214, Fax. + 62 31 5913804
Email: arist@its-sby.edu

ABSTRAK

Penentuan pola distribusi adalah bagian yang amat penting dalam urutan proses simulasi sistem. Ada beberapa metode yang bisa digunakan untuk menguji pola distribusi, antara lain metode statistik Chi Squared dan Kolmogorov-Smirnov atau dengan menggunakan jaringan saraf tiruan (JST). Berdasar pengamatan dari terdahulu, didapatkan bahwa chi-squared tidak cocok digunakan untuk menentukan pola distribusi dari data yang berjumlah kecil. Sedangkan untuk Kolmogorov Smirnov, dalam jumlah data yang sama-sama kecil masih memiliki tingkat keberhasilan yang lebih ringgi dari pada metode Chi Squared. Kelemahan lain dari uji Chi-squared ataupun uji K-S salah satunya adalah hanya bisa mengenali pola distribusi matematis, pola-pola yang sudah diketahui rumus matematisnya. Atau dengan kata lain hanya bisa mengenali pola distribusi data yang rumusnya sudah disertakan di dalam perangkat lunak penguji.

Dengan memakai JST, kelemahan-kelemahan tersebut bisa dihilangkan. Pada penelitian sebelumnya belum ditentukan penciri khas yang bisa digunakan untuk membedakan suatu pola distribusi dengan yang lainnya. Dengan kata lain, seluruh data diinputkan langsung ke JST setelah mengalami proses pra-pengolahan BagiMax. Pada penelitian ini dicoba ditentukan penciri yang mungkin bisa dipakai untuk membedakan pola distribusi data, yaitu dengan memanfaatkan persentil sebagai ciri utama.

Pada penelitian ini pra-pengolahan berupa prosedur HitungPersentil100pth untuk mendapatkan persentil 0.1 sampai dengan 0.9. Hasil perhitungan persentil tersebut diumpankan ke JST yang telah dibuat pada penelitian sebelumnya. Dari uji coba ternyata diperoleh prosentase keberhasilan meningkat sampai 91.9%, yang menunjukkan bahwa persentil mampu dipakai sebagai penciri.

Kata kunci : *persentil, pola distribusi data, simulasi komputer, jaringan syaraf tiruan.*

1. PENDAHULUAN

Salah satu alasan pengumpulan data dari sistem adalah untuk mempelajari karakteristik proses probabilistiknya. Sebagai contoh, pada sistem antrian, bagaimana kedatangan terjadi? Dengan menganalisis data dan pemahaman tingkah laku beberapa sistem yang berbeda, dapat digambarkan dengan distribusi probabilitas untuk simulasi sistem. Pada beberapa kasus, distribusi teoritis dapat diterapkan, dan pada kasus lain, distribusi non-teoritis justru yang diperlukan. Sebagai contoh, misalnya adalah distribusi panggilan telepon pada simulasi perusahaan telepon. Salah satu hal yang menjadi perhatian adalah distribusi panggilan telepon. Sebelum data dikumpulkan, kita bisa memprediksi bahwa distribusi panggilan teleponnya akan mengikuti distribusi Poisson. Setelah data panggilan telepon didapatkan, data ini dapat digunakan untuk mengestimasi rerata kedatangan panggilan telepon untuk distribusi Poisson kemudian memeriksanya apakah data ini mengikuti distribusi Poisson.

Sebuah cara informal yang dapat digunakan untuk memutuskan apakah suatu data mengikuti distribusi tertentu adalah dengan membuat grafik data dan grafik distribusinya, kemudian memutuskan apakah kedua grafik tersebut ‘pas’ atau tidak. Bila yang diamati adalah distribusi kontinyu, kita dapat melakukan superimpose hasil histogramnya. Sedangkan untuk distribusi diskrit, grafiknya dibuat dari frekuensi relatif yang didapat dari data dan dihitung dari rumus distribusi probabilitas.

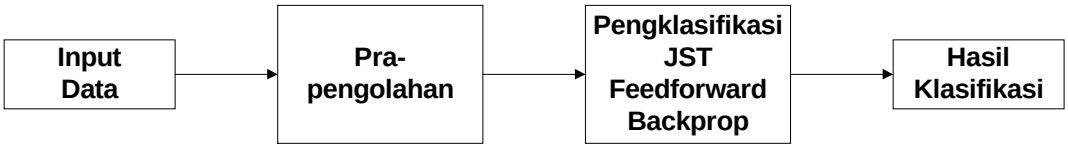
Meskipun cara visual dapat kita manfaatkan untuk menentukan pola distribusi, tetapi akan lebih mudah lagi jika menggunakan cara pemeriksaan secara statistis. Metode statistis yang biasanya digunakan untuk memeriksa pola distribusi adalah chi-squared dan Kolmogorov-Smirnov. Metode lainnya adalah dengan menerapkan jaringan syaraf dalam proses pengenalan suatu pola distribusi.

2. UJI POLA DISTRIBUSI DENGAN METODE STATISTIK

Prosedur yang sering digunakan dalam mengevaluasi asumsi distribusi adalah uji chi-squared. Kendala yang dihadapi oleh chi-squared adalah dalam menentukan interval, ini terutama pada kasus distribusi kontinyu. Sebaliknya pada uji Kolmogorov-Smirnov (K-S), membandingkan fungsi distribusi empiris dengan fungsi hipotesis distribusi . Uji K-S tidak memerlukan pengelompokan data seperti pada chi-squared, dengan demikian tidak ada informasi yang hilang [2]. Keuntungan lain uji K-S adalah cocok digunakan untuk sebarang jumlah sampel n (pada kasus semua parameter sudah diketahui).

3. UJI DENGAN JST

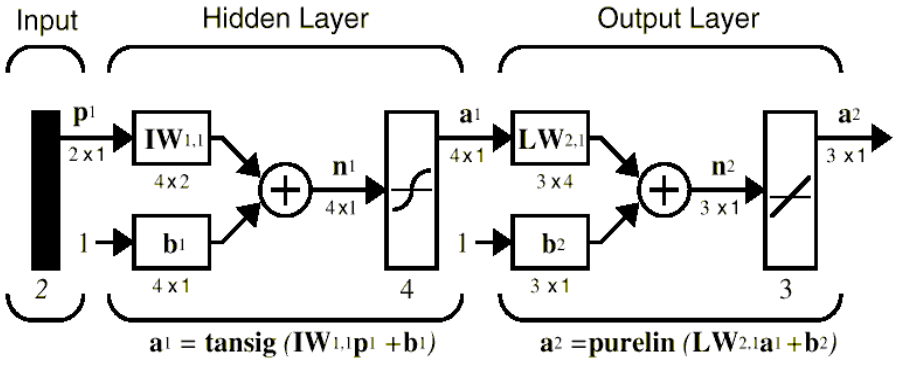
Jaringan syaraf tiruan (JST) memiliki sejumlah besar elemen pemroses neuron dan sejumlah besar hubungan terbobot antara elemen-elemen. Model komputasi JST dapat dikelompokkan menjadi beberapa kelompok aplikasi [8], yaitu (a) klasifikasi, penentuan kategori dari input data; (b) asosiasi, pengambilan suatu objek berdasar pada bagian tertentu objek itu sendiri ataupun dengan memanfaatkan objek lain pada kelompok yang berbeda; (c) optimisasi, pencarian solusi terbaik; dan (d) aplikasi lain yang memanfaatkan kemampuan pembelajaran adaptif.



Gambar 1. Diagram blok sistem

Pada gambar 1 di atas memperlihatkan diagram blok sistem pengenal pola distribusi data. Agar pelatihan JST dapat dilakukan secara lebih efisien, maka terlebih dulu dilakukan proses Pra-pengolahan pada data input maupun target. Data input dan target tsb diubah sedemikian rupa sehingga memiliki jangkauan [-1, 1]. Pada penelitian ini, pra-pengolahan memakai cara sederhana, yaitu dengan membagi input data dengan suatu nilai *xn* yang lebih besar dari $|a_i|$. Dengan cara itu, input data tidak kehilangan 'makna', lain halnya jika digunakan pra-pengolahan *prestd* atau *premnmx*. Hanya saja *xn* jangan terlalu besar, karena akan menyebabkan proses pelatihan menjadi terlalu lama untuk konvergen.

Arsitektur JST yang digunakan adalah *feedforward network* dengan algoritme pelatihan propagasi balik *traingdx*. Secara garis besar, arsitektur yang digunakan adalah sbb:



Gambar 2. Arsitektur JST feedfoward [9]

Setelah bobot dan nilai bias diinisialisasi, JST dilatih dengan memberi umpan beberapa data yang sudah diketahui pola distribusi datanya. Algoritme *traingdx* merupakan suatu *adaptive learning rate* [9] yang berusaha mempertahankan ukuran *leraning rate* *lr*. Bila *lr* terlalu besar, maka sistem akan berosilasi dan menjadi tidak stabil. Sedangkan bila *lr* terlalu kecil, sistem membutuhkan waktu yang lebih lama untuk konvergen. Proses pelatihan akan berhenti bila sudah mencapai jumlah maksimum *epoch*, atau

melebihi maksimum waktu yang ditentukan, atau mencapai performa target yang ditetapkan sebelumnya.

Agar JST bisa dipergunakan untuk mengenali pola distribusi data, perlu dilakukan terlebih dulu perubahan representasi input datanya dalam bentuk yang sesuai. Setelah itu representasi input tersebut diumpankan ke JST untuk dicocokkan dengan pola-pola distribusi data yang sudah dilatihkan sebelumnya. Dengan demikian, JST dapat digunakan untuk mengenali sembarang pola distribusi data yang sudah dilatihkan sebelumnya.

Pada waktu uji coba pelatihan JST, puluhan kali dilakukan penyesuaian paramater input untuk mendapatkan sistem yang stabil dan konvergen secara cepat. Dan ini bisa dicapai antara lain dengan melakukan pra-pengolahan *BagiMax* yaitu dengan mengubah input data bernilai [-1, 1] serta membagi setiap elemen data dengan max($|a_i|$).

Dengan memakai JST, kelemahan-kelemahan yang dijumpai pada metode Chi-squared atau K-S dapat bisa dihilangkan. Sistem bisa dilatih untuk mengenali distribusi data yang sudah diketahui polanya, tanpa perlu memasukkan rumus matematis distribusi data seperti pada sistem berbasis Chi-squared ataupun K-S. Pada ujicoba, didapatkan tingkat keberhasilan JST adalah sebesar 63% untuk jumlah data 100 buah serta 88% untuk jumlah data 500 buah.

4. PERSENTIL SEBAGAI PENCIRI

Persentil *sample 100pth* merupakan sebuah nilai dari data terurut dari kecil ke besar, paling tidak 100p% dari pengamatan adalah terletak pada atau sebelah kiri (bawah) nilai itu dan paling tidak 100(1-p)% terletak pada atau sebelah kanan (atas) dari nilai itu.

Pada penelitian kali ini, tahap pra-pengolahan diganti dengan prosedur *HitungPersentil100pth* untuk mendapatkan persentil 0.1 sampai dengan 0.9. Hasil perhitungan persentil tsb baru diumpankan ke JST yang telah dibuat pada penelitian sebelumnya [12].

Untuk menghitung persentil 100pth dari data terkelompok bisa digunakan algoritme *HitungPersentil100pth* berikut ini [11]:

a) Hitung frekuensi kumulatif
b) Tentukan interval dengan catatan frekuensi kumulatif sampai atau melebihi nilai *np* dengan *n* adalah jumlah data.
c) Hitung $\frac{(np-a)}{f}$ dengan *a* adalah frekuensi kumulatif interval sebelumnya dan *f* adalah frekuensi kelas persentil 100p
d) Hitung

   persentil100pth = $L+\frac{(np-a)}{f}h$

   dengan

   h=panjang kelas persentil100pth

Pada penelitian ini interval ditentukan secara sederhana dengan menggunakan rumus:

interval=(MAX-MIN)/JUMKELOMPOK;

Dengan MAX adalah data terbesar, MIN adalah data terkecil, dan JUMKELOMPOK adalah jumlah kelas atau kelompok data. Kali ini JUMKELOMPOK ditentukan sebagai 10.

Hasil perhitungan persentil mulai dari 0, 0.1, 0.2, ..., 1.0 kemudian diumpankan ke JST yang telah dibuat sebelumnya. Contoh hasil perhitungan persentil dari pola distribusi eksponensial dengan mean μ=5.0 dan distribusi normal dengan mean μ=5.0 serta standard deviasi *s*=1.0 yang akan diumpankan ke JST adalah sebagai berikut:

0 0.325 0.175 0.15 0.05 0.125 0.05 0.05 0 0.025 0.025 0.025 0 0 0 0

0 0.25 0.15 0.1 0.125 0.075 0.075 0.125 0.05 0.025 0 0.025 0 0 0 0

0 0.05 0.075 0.05 0.125 0.125 0.15 0.15 0.125 0.075 0.05 0.025 0 0 0 1

0 0.025 0.05 0.1 0.2 0.125 0.15 0.1 0.1 0.1 0.025 0.025 0 0 0 1

Gambar 3. Contoh data training

5. UJI COBA

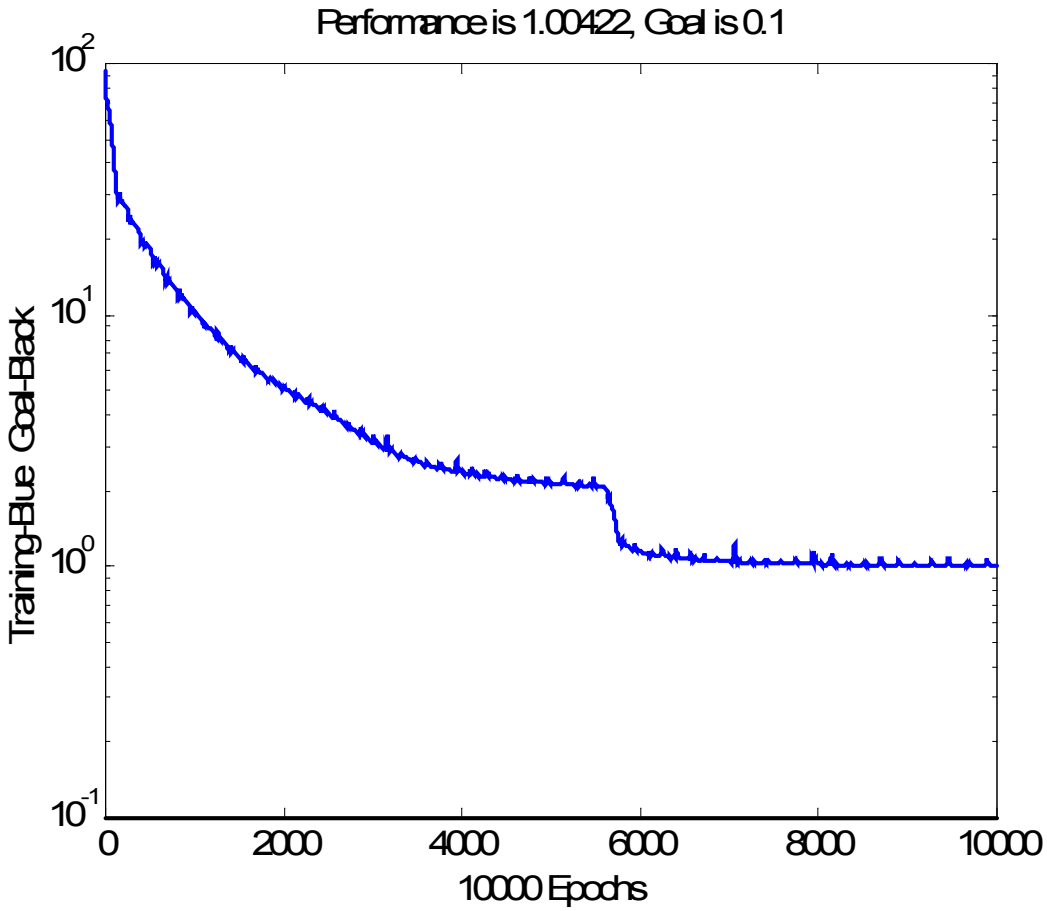
Pada uji coba kali ini digunakan algoritme pelatihan *traingdx* yang merupakan fungsi pelatihan JST yang memperbaharui nilai bobot dan nilai bias menurut momentum penurunan gradien (*gradient descent momentum*) dan laju pembelajaran adaptif (*adaptive learning rate*). Uji coba dilakukan dengan mengubah susunan layer neuron seperti yang terlihat pada tabel 1.

Tabel 1.Uji coba pelatihan JST

Susunan Layer	Jumlah Kesalahan Identifikasi	Prosentase Keberhasilan
30-4	2	90,70
25-4	3	89,53
21-4	2	90,70
20-4	1	91,86
19-4	3	89,53
18-4	12	79,07
15-4	2	90,70
10-4	7	84,88
9-4	6	86,05
8-4	7	84,88
4-4	22	67,44
9-9-4	5	87,21
6-6-4	2	90,70
4-4-4	9	82,56
5-5-4	4	88,37
6-5-4	6	86,05
7-6-4	3	89,53
5-6-4	4	88,37
7-7-4	3	89,53
7-7-7-4	2	90,70
6-6-6-4	2	90,70
4-4-4-4	12	79,07
5-5-5-4	8	83,72

JST dilatih dengan menggunakan 80 kelompok data yang memiliki pola distribusi eksponensial, normal, lognormal, erlang, poisson, dan sebagainya sampai 8 pola distribusi. Masing-masing kelompok data terdiri dari 40 buah data.

Dari hasil uji coba terlihat bahwa dengan memanfaatkan persentil sebagai penciri bisa meningkatkan taraf keberhasilan pengenalan sampai 91.9%. Selain itu proses pelatihan dengan pra-pengolahan *HitungPersentil100pth* dilakukan lebih cepat dibanding pra-pengolahan *BagiMax*. Hanya saja seluruh uji coba pelatihan yang dilakukan belum pernah mencapai target performance sebesar 0.1, dengan kata lain proses pelatihan berhenti karena telah mencapai *max epoch* sebesar 10000. Walaupun dicoba sampai 100000 epoch target performansi tetap tidak tercapai.



Gambar 4. Contoh proses pelatihan untuk susunan layer 20-4

Taraf keberhasilan pengenalan yang mencapai 91.9% tersebut kemungkinan masih bisa ditingkatkan lagi dengan memperbesar jumlah kelompok persentil, yang diharapkan akan lebih mampu memberikan ciri yang lebih detail. Akan tetapi dengan menambah jumlah kelompok persentil, dikhawatirkan akan memperlama proses pembelajaran oleh karena meningkatkan jumlah input layer JST.

6. KESIMPULAN DAN SARAN

Berdasarkan uji coba yang telah dilakukan didapat kesimpulan antara lain:

a. Pemakaian persentil sebagai penciri mampu meningkatkan prosentase keberhasilan sampai 91.9%. Pada penelitian berikutnya akan diteliti lebih lanjut apakah penambahan jumlah kelompok persentil bisa meningkatkan prosentase keberhasilan.
b. Proses pelatihan dengan menggunakan algoritme *traingdx* belum bisa digunakan untuk mencapai target performansi sebesar 0.1. Proses pelatihan berhenti karena telah mencapai batas maksimum.

Pada penelitian ini pola distribusi data yang diujicobakan masih terbatas dengan pola data yang memiliki parameter input seperti mean atau standard deviasi yang sama. Pada penelitian berikutnya perlu diperiksa apakah dengan menggunakan persentil sebagai penciri mampu mengenali pola distribusi dengan parameter yang berbeda.

7. DAFTAR PUSTAKA

1. Bratley, Paul, “A Guide to Simulation”, Springer-Verlag, New York
2. Ross, “Introduction to Probability Models”, 3rd/ed, Academic Press, New York, 1986
3. Aris Tjahyanto, “Rancang bangun alat bantu penentu pola distribusi input dengan metode Chi-Squared”, Lemlit ITS, 1999
4. Aris Tjahyanto, “Rancang bangun alat bantu penentu pola distribusi input berbasis Web untuk jumlah sampel kecil dengan metode Kolmogorov-Smirnov”, Lemlit ITS, 2000
5. Matlab Neural Network Toolbox, MathWorks Inc., 1999
6. Aris Tjahyanto & FX Arunanto, “Perbandingan sensitifitas beberapa metode penguji pola distribusi data: Chi-Square, Kolmogorov-Smirnov, dan Jaringan Syaraf Tiruan”, Proceding SITIA, Mei 2001

7. Djarwanto Ps SE, “Mengenal beberapa uji statistik”, Liberty, Yogyakarta, 1996