

BAB III

METODE PENELITIAN

3.1 Metode Jaringan Syaraf Tiruan *Backpropagation*

Backpropagation adalah salah satu algoritma dengan *multilayer perceptron*, pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada jaringan syaraf tiruan (Sri Kusumadewi, 2004 : 93).

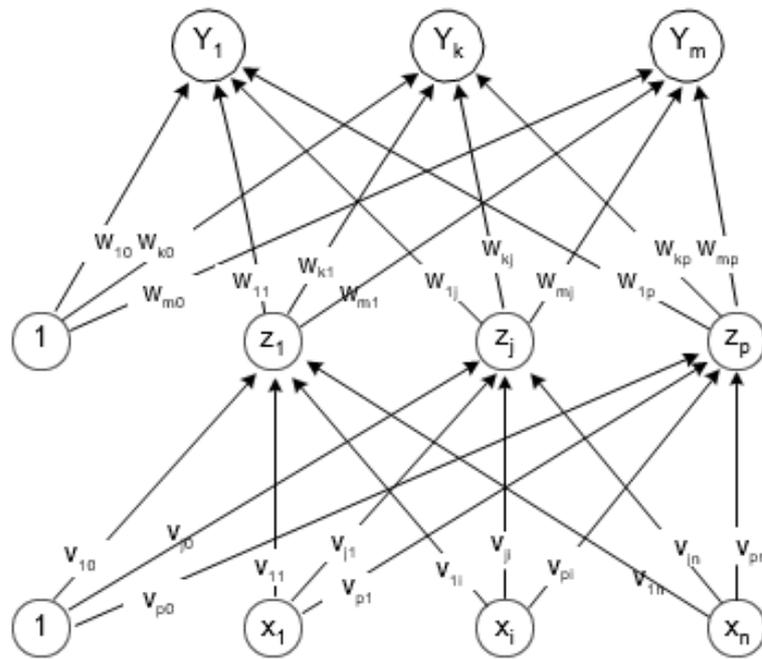
Kelemahan Jaringan Syaraf Tiruan (JST) yang terdiri dari layar tunggal membuat perkembangan JST menjadi terhenti pada sekitar tahun 1970 an. JST dengan layar tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu/beberapa layar tersembunyi diantara layar input dan layar output.

Seperti halnya model JST lain, *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan (J J Siang, 2005: 97).

1.2 Arsitektur *Backpropagation*

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. Gambar 3.1 adalah arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah layar tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran.

v_{ji} merupakan bobot garis dari unit masukan x_i ke unit layar tersembunyi z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi z_j). w_{kj} merupakan bobot dari unit layar tersembunyi z_j ke unit keluaran y_k (w_{k0} merupakan bobot dari bias di layar tersembunyi ke unit keluaran z_k).



Gambar 3.1 Arsitektur *Backpropagation*

1.3 Fungsi Aktivasi

Pemilihan fungsi aktivasi yang tepat pada sebuah aplikasi berbasis jaringan syaraf tiruan akan sangat mempengaruhi performanya baik dari segi kecepatan pemrosesan data maupun dari segi tingkat keakuratan hasilnya (Sri Redjeki, 2014). Karakteristik yang harus dimiliki fungsi aktivasi tersebut adalah kontinu dan tidak menurun secara monoton. Fungsi aktivasi diharapkan jenuh (mendekati nilai-nilai maksimum dan minimum secara asimtot) (Puspitaningrum, 2006). Dalam metode jaringan syaraf tiruan *backpropagation* terdapat beberapa fungsi aktivasi yang dapat digunakan, seperti fungsi sigmoid biner dan fungsi sigmoid bipolar.

1. Fungsi Sigmoid Biner

Fungsi sigmoid biner memiliki nilai antara 0 sampai 1 dan dirumuskan sebagai:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Fungsi ini memiliki turunan pertama:

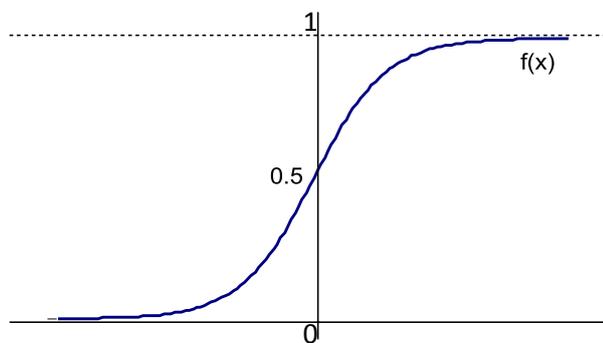
Winormia Tresnaeni, 2018

PERAMALAN TINGKAT INFLASI PROVINSI JAWA BARAT MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN BACKPROPAGATION

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

$$\begin{aligned}
 f'(x) &= \frac{0 + e^{-x} \cdot 1}{(1 + e^{-x})^2} \\
 f'(x) &= \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} \\
 f'(x) &= \frac{1}{1 + e^{-x}} \cdot \frac{1 + e^{-x} - 1}{1 + e^{-x}} \\
 f'(x) &= \frac{1}{1 + e^{-x}} \cdot \frac{1 + e^{-x}}{1 + e^{-x}} - \frac{1}{1 + e^{-x}} \\
 f'(x) &= \frac{1}{1 + e^{-x}} \left(1 - \frac{1}{1 + e^{-x}} \right) \\
 f'(x) &= f(x)(1 - f(x))
 \end{aligned}$$

Grafik fungsinya seperti pada gambar



Gambar 3.2 Fungsi Sigmoid Biner

2. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar memiliki bentuk fungsi yang mirip dengan fungsi sigmoid biner, tapi dengan range $(-1,1)$. Jika $f(x)$ adalah fungsi sigmoid biner, maka fungsi sigmoid bipolar dirumuskan sebagai:

$$\begin{aligned}
 g(x) &= 2f(x) - 1 \\
 &= \frac{2}{1 + e^{-x}} - 1 \\
 &= \frac{2}{1 + e^{-x}} - \frac{1 + e^{-x}}{1 + e^{-x}} \\
 &= \frac{2 - (1 + e^{-x})}{1 + e^{-x}} \\
 &= \frac{2 - 1 - e^{-x}}{1 + e^{-x}}
 \end{aligned}$$

Winormia Tresnaeni, 2018

PERAMALAN TINGKAT INFLASI PROVINSI JAWA BARAT MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN BACKPROPAGATION

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

$$= \frac{1 - e^{-x}}{1 + e^{-x}}$$

Dengan turunan pertama:

$$g'(x) = \frac{-e^{-x} \cdot (-1)(1 + e^{-x}) - (1 - e^{-x}) e^{-x} \cdot 1}{(1 + e^{-x})^2}$$

$$g'(x) = \frac{e^{-x} + e^{-2x} - e^{-x} + e^{-2x}}{(1 + e^{-x})^2}$$

$$g'(x) = \frac{2e^{-2x}}{(1 + e^{-x})^2}$$

$$g'(x) = \frac{2 \cdot e^{-x} \cdot e^{-x}}{(1 + e^{-x})(1 + e^{-x})}$$

$$g'(x) = \frac{2 \cdot e^{-x}}{(1 + e^{-x})} \cdot \frac{e^{-x}}{(1 + e^{-x})}$$

$$g'(x) = \frac{1 + e^{-x} + 1}{(1 + e^{-x})} \cdot \frac{1 + e^{-x} - 1}{(1 + e^{-x})}$$

$$g'(x) = \left(\frac{1 + e^{-x}}{(1 + e^{-x})} + \frac{1}{(1 + e^{-x})} \right) \cdot \left(\frac{1 + e^{-x}}{(1 + e^{-x})} - \frac{1}{(1 + e^{-x})} \right)$$

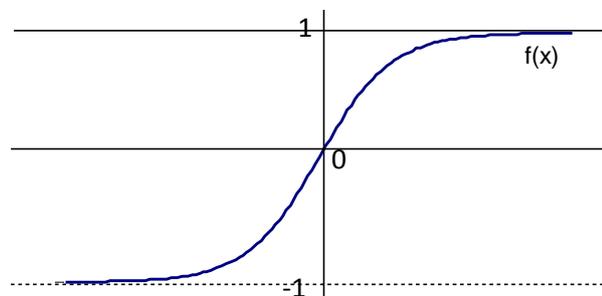
$$g'(x) = (1 + f(x))(1 - f(x))$$

$$g'(x) = \left(1 + \frac{g(x) - 1}{2} \right) \left(1 - \frac{g(x) - 1}{2} \right)$$

$$g'(x) = \left(\frac{2 + g(x) - 1}{2} \right) \left(\frac{2 - g(x) - 1}{2} \right)$$

$$g'(x) = \frac{(1 + g(x))(1 - g(x))}{2}$$

Grafik fungsinya seperti pada gambar



Gambar 3.3 Fungsi Sigmoid Bipolar

1.4 Pelatihan Standar Backpropagation

Pelatihan Backpropagation meliputi 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

Fase pertama : propagasi maju

Selama propagasi maju, sinyal masukan ($= x_i$) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi ($= z_j$) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ($= y_k$).

Berikutnya keluaran jaringan ($= y_k$) dibandingkan dengan target yang harus dicapai ($= t_k$). Selisih $t_k - y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

Fase II : Propagasi mundur

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor δ_k ($k = 1, 2, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit y_k ke semua unit tersembunyi yang terhubung langsung dengan y_k . δ_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor δ_k di setiap unit di lapisan tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

Fase III : Perubahan bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δ_k yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Algoritma pelatihan untuk jaringan dengan satu layar tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut :

- a. **Langkah 0** : Inisialisasi semua bobot dengan bilangan acak kecil
- b. **Langkah 1**: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 – 9
- c. **Langkah 2** : Untuk setiap pasang data pelatihan, lakukan langkah 3 – 8
Fase I : Propagasi maju
- d. **Langkah 3** : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya
- e. **Langkah 4** : Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$

- f. **Langkah 5** : Hitung semua keluaran jaringan di unit y_k ($k = 1, 2, \dots, m$)

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}$$

Fase II : Propagasi mundur

- g. **Langkah 6** : Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$)

Winormia Tresnaeni, 2018

PERAMALAN TINGKAT INFLASI PROVINSI JAWA BARAT MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN BACKPROPAGATION

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

$$\delta_k = (t_k - y_k) f'(y_{net_j}) = (t_k - y_k) y_k (1 - y_k)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar di bawahnya (langkah 7)

Kemudian hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α

$$\Delta w_{kj} = \alpha \delta_k z_j ; \quad k = 1, 2, \dots, m ; j = 0, 1, \dots, p$$

- h. **Langkah 7** : Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj}$$

Faktor δ unit tersembunyi :

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j)$$

Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk merubah bobot v_{ji})

$$\Delta v_{ji} = \alpha \delta_j x_i ; \quad j = 1, 2, \dots, p ; \quad i = 0, 1, \dots, n$$

Fase III : Perubahan Bobot

- i. **Langkah 8** : Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (k = 1, 2, \dots, m ; j = 0, 1, \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (j = 1, 2, \dots, p ; i = 0, 1, \dots, n)$$

- j. **Langkah 9** : Proses pelatihan berhenti.

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

1.5 Peramalan Menggunakan Jaringan Syaraf Tiruan Backpropagation

Langkah-langkah yang harus dilakukan untuk pemodelan peramalan menggunakan jaringan syaraf tiruan backpropagation adalah sebagai berikut:

a. Pembagian Data

Data dibagi menjadi 2 yaitu data yang akan digunakan untuk pelatihan dan data yang digunakan untuk pengujian dalam metode jaringan syaraf tiruan *backpropagation*.

b. Normalisasi Data

Pada proses peramalan menggunakan jaringan syaraf tiruan *backpropagation*, sebelum dilakukan pelatihan data input dan target output harus dinormalisasi terlebih dahulu. Normalisasi adalah penskalaan terhadap nilai-nilai masuk ke dalam suatu range tertentu. Hal ini dilakukan agar nilai input dan target output sesuai dengan range dari fungsi aktivasi yang digunakan dalam jaringan. Normalisasi data akan digunakan transformasi linier pada interval a sampai b (Siang,2005).

$$x' = \frac{(b - a)(x - x_{min})}{x_{max} - x_{min}} + a$$

Karena pada penulisan ini akan digunakan fungsi aktivasi sigmoid biner dimana nilainya berada pada interval (0,1) yaitu tidak pernah tepat menyentuh 0 atau 1, maka data akan dinormalisasi ke interval 0,1 sampai 0,9. Transformasi yang digunakan untuk mentransformasikan data ke interval [0,1,0,9] adalah sebagai berikut

$$x' = \frac{(0,9 - 0,1)(x - x_{min})}{x_{max} - x_{min}} + 0,1$$

$$x' = \frac{0,8(x - x_{min})}{x_{max} - x_{min}} + 0,1$$

c. Memilih dan menggunakan jaringan yang optimum

Jaringan yang dibangun akan dinilai keakuratannya dengan menentukan neuron terbaik pada lapisan tersembunyi. Indikator pemilihan penilaian yang digunakan

Winormia Tresnaeni, 2018

PERAMALAN TINGKAT INFLASI PROVINSI JAWA BARAT MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN BACKPROPAGATION

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

adalah nilai MSE. Berdasarkan nilai indikator terendah dari proses pembelajaran, maka diperoleh jaringan yang terbaik.

d. Denormalisasi data

Setelah proses pelatihan dan pengujian selesai, maka data yang telah dinormalisasi dikembalikan seperti semula atau disebut denormalisasi data.

$$x' = \frac{0,8(x - x_{min})}{x_{max} - x_{min}} + 0,1$$

$$x' - 0,1 = \frac{0,8(x - x_{min})}{x_{max} - x_{min}}$$

$$(x' - 0,1) (x_{max} - x_{min}) = 0,8(x - x_{min})$$

$$x - x_{min} = \frac{(x' - 0,1) (x_{max} - x_{min})}{0,8}$$

$$x = \frac{(x' - 0,1) (x_{max} - x_{min})}{0,8} + x_{min}$$

1.6 Kontruksi Perancangan Program

Pada bagian ini akan dibahas mengenai rancangan data input, data output, dan tampilan program menggunakan *software* Matlab R2015a.

1.6.1 Data Input

Pada program ini akan dimuat data input sebagai berikut:

1. Nama file
2. Data pelatihan
3. Data pengujian

1.6.2 Data Output

Pada program ini akan ditampilkan data output yaitu:

1. Jaringan hasil pelatihan
2. Hasil pengujian
3. Data hasil peramalan

1.6.3 Perancangan Tampilan Program

Berikut rancangan tampilan program menggunakan GUI di *software* Matlab R2015a.



Winormia Tresnaeni, 2018

PERAMALAN TINGKAT INFLASI PROVINSI JAWA BARAT MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN BACKPROPAGATION

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu