

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Peramalan**

Peramalan pada dasarnya merupakan proses menyusun informasi tentang kejadian masa lampau yang berurutan untuk menduga kejadian di masa depan (Frechtling, 2001: 8). Peramalan bertujuan mendapatkan ramalan yang dapat meminimumkan kesalahan meramal yang dapat diukur dengan *Mean Absolute Percent Error (MAPE)* (Pangestu Subagyo, 1986: 1). Peramalan pada umumnya digunakan untuk memprediksi sesuatu yang kemungkinan besar akan terjadi misalnya kondisi permintaan, banyaknya curah hujan, kondisi ekonomi, dan lain – lain.

Atas dasar logika, langkah dalam metode peramalan secara umum adalah mengumpulkan data, menyeleksi dan memilah data, memilih model peramalan, menggunakan model terpilih untuk melakukan peramalan, evaluasi hasil akhir.

Berdasarkan sifatnya, peramalan dibedakan menjadi:

1. Peramalan Kualitatif

Peramalan yang didasarkan atas dasar kualitatif pada masa lalu. Hasil peramalan kualitatif didasarkan pada pengamatan kejadian – kejadian di masa sebelumnya digabung dengan pemikiran dan penyusunnya.

2. Peramalan Kuantitatif

Peramalan yang didasarkan atas dasar kuantitatif masa lalu yang diperoleh dari pengamatan nilai – nilai sebelumnya. Hasil peramalan yang dibuat tergantung pada metode yang digunakan, menggunakan metode yang berbeda akan diperoleh hasil peramalan yang berbeda.

## 2.2. Data Time Series

*Time series* merupakan serangkaian data pengamatan yang berasal dari suatu sumber tetap yang terjadinya berdasarkan indeks waktu  $t$  secara berurutan dan dengan interval waktu yang tetap. Setiap pengamatan dapat dinyatakan sebagai variable random  $Z_t$  dengan fungsi kepadatan  $f(Z_t)$  yang dapat dipasangkan dengan indeks waktu  $t_i$ , yaitu :

$$Z_1, Z_2, Z_3, \dots, Z_n$$

$t_1, t_2, t_3, \dots, t_n$  adalah urutan waktu pengamatan.

Karena itu data time series yang diamati pada waktu  $t_1, t_2, t_3, \dots, t_n$  dapat dituliskan dalam notasi  $Z_{t_1}, Z_{t_2}, Z_{t_3}, \dots, Z_{t_n}$ .

*Time series* adalah suatu rangkaian atau seri dari nilai – nilai suatu variabel atau hasil observasi, dalam hal ini adalah nilai indeks harga saham, yang dicatat dalam jangka waktu yang berurutan (Atmaja, 2009: 29).

Hal yang perlu diperhatikan dalam melakukan peramalan adalah pola galat (*error*), yang tidak dapat dipisahkan dalam metode peramalan. Untuk

mendapatkan hasil yang mendekati data asli, maka seorang peramal berusaha membuat *error*-nya sekecil mungkin.

Dengan adanya data *time series* , maka pola gerakan data dapat diketahui.

Dengan demikian, data *time series* dapat dijadikan sebagai dasar untuk:

- Pembuatan Keputusan pada saat ini.
- Peramalan keadaan perdagangan dan ekonomi pada masa yang akan datang
- Perencanaan kegiatan untuk masa depan

### **2.3. Artificial Neural Network**

*Artificial Neural Network* (ANN) adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel syaraf biologi di dalam otak. ANN dapat digambarkan model matematis dan komputasi untuk fungsi aproksimasi nonlinear, klasifikasi data, cluster dan regresi non parametrik atau sebagai sebuah simulasi dari koleksi model syaraf biologi.

Model syaraf ditunjukkan dengan kemampuannya dalam emulasi, analisa, prediksi, dan asosiasi. Berdasarkan kemampuan yang dimiliki, ANN dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh, untuk menghasilkan *output* yang sempurna dari contoh atau input yang dimasukkan dan membuat prediksi tentang kemungkinan output yang akan muncul atau menyimpan karakteristik dari input yang disimpan kepadanya.

Artificial Neural Network(ANN) merupakan suatu pemodelan yang diilhami oleh jaringan syaraf biologis pada tubuh manusia. ANN terdiri dari unit

utama, dan neuron-neuron yang terdesain seperti susunan jaringan syaraf biologis. Neuron-neuron buatan ini mempunyai jalur input sebagaimana neuron syaraf biologis yang mempunyai *dendrit*, dan terdapat juga jalur aoutput sebagaimana neuron syaraf mempunyai *axon*. Neuron buatan maupun neuron syaraf biologis mempunyai satu kecenderungan yang menentukan nilai output. Neuron mengkombinasikan input, nilai bias dan sinyal output. Pada kedua neuron baik buatan maupun neuron syaraf biologis, terdapat proses belajar (*learning*) yang mengubah nilai-nilai kecenderungan pada neuron.

Pada neuron biologis, proses alami mengatur bagaimana sinyal inputan pada dendrit diproses dan kemudian diterjemahkan dalam aktifitas axon. Sedangkan pada neuron buatan, proses learning mengatur input-input yang digunakan dan bagaimana pemetaannya pada output.

Dalam ANN pada jalur input neuron  $i$  terdapat sinyal di dalamnya ( $P_i$ ) dan besar nilai kecenderungan pada tiap jalur diwakili oleh sebuah pembobot atau weight ( $w_i$ ). Neuron-neuron tersebut membentuk suatu jumlah perkalian antara weight dengan input yang meliputi keseluruhan weight dan juga nilai bias ( $b$ ). Output yang dibentuk ( $a$ ) biasanya merupakan bentuk *sigmoid* fungsi logistic dari penjumlahan tersebut.

Proses learning terjadi pada saat pengaturan weight dan bias. Metode yang paling umum digunakan untuk proses learning tersebut adalah *Back propagation*.

Dalam metode ini, weight-weight tersebut diatur untuk meminimalisasi nilai kuadrat beda antara output model dan output taksiran, atau secara umum disebut

SSE(*sums of square error*). Pada proses pengaturan weight biasanya berdasarkan pada algoritma sederhana *steepest descent* yang didasarkan pada turunan pertama pada fungsi *cost*. Namun pada beberapa penelitian yang pernah dilakukan, algoritma yang lebih rumit sering dipakai, dimana lebih didasarkan pada bentuk turunan kedua dari fungsi *cost*, dan lebih memberikan jaminan tercapainya SSE yang paling minimum.

Sebagaimana proses alami, beberapa neuron terkombinasi membentuk sebuah jaringan, jaringan tersebut terdiri dari sebuah lapisan input (*input layer*), sebuah lapisan output (*output layer*) dan kemungkinan satu atau lebih lapisan atau sering disebut sebagai lapisan tersembunyi (*hidden layer*). Setiap layer terdiri dari beberapa neuron, dan neuron-neuron ini dihubungkan dengan neuron-neuron lain pada layer terdekat. (Andri Kristanto 2004 : 2)

### **2.3.1. BackPropagation**

BackPropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot – bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma backpropagation menggunakan error output untuk mengubah nilai bobot – bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu.

Back propagation (BP) merupakan proses estimasi atau proses *learning* untuk menentukan weight dari suatu arsitektur ANN. BP merupakan

pengembangan dari suatu proses estimasi “*Widrow-Hoff learning rule*” yang dikembangkan oleh Rumalhart, Hinton dan Williams (1986). Aturan Widrow-Hoff digunakan untuk mengestimasi weight dari suatu single layer neuron yang menggunakan fungsi aktivasi linier. Proses BP merupakan generalisasi dari Widrow-Hoff yang digunakan untuk proses estimasi pada multi layer perceptron (MPL), dan menggunakan fungsi aktivasi non linier. Sebagaimana aturan Widrow-Hoff, BP menggunakan metode *steepest Descant* untuk menentukan sekelompok weight dan bias yang akan meminimalisasi nilai *sums of squared errors* atau secara umum dapat disebut sebagai SSE.

Metode BP ini banyak diaplikasikan secara luas. Sekitar 90% bahkan lebih BP telah berhasil diaplikasikan diberbagai bidang diantaranya diterapkan di bidang finansial, pengenalan pola tulisan, pengenalan pola suara, sistem kendali, pengolahan citra medika dan masih banyak lagi keberhasilan BP sebagai salah satu metode komputasi yang handal.

Algoritma ini banyak dipakai pada aplikasi pengaturan karena proses pelatihannya didasarkan pada hubungan yang sederhana, yaitu : Jika keluaran memberikan hasil yang salah, maka penimbang (*weight*) dikoreksi supaya galat atau errornya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar, BP juga berkemampuan untuk memperbaiki penimbang pada lapisan tersembunyi (*hidden layer*).

Secara garis besar, mengapa algoritma ini disebut propagasi balik, dapat dideskripsikan sebagai berikut : Ketika jaringan di berikan pola masukan sebagai

pola pelatihan maka pola tersebut menuju ke unit-unit lapisan tersembunyi dan diteruskan ke unit-unit lapisan keluaran. Kemudian unit-unit lapisan keluaran memberikan tanggapan yang disebut sebagai keluaran jaringan. Saat keluaran jaringan tidak sama dengan keluaran yang diharapkan maka keluaran akan menyebar mundur (*backward*) pada lapisan tersembunyi diteruskan ke unit pada lapisan masukan. Oleh karenanya maka mekanisme pelatihan tersebut dinamakan *backpropagation* / propagasi balik.

Tahap pelatihan ini merupakan langkah bagaimana suatu jaringan syaraf itu berlatih, yaitu dengan cara melakukan perubahan penimbang (sambungan antar lapisan yang membentuk jaringan melalui masing-masing unitnya). Sedangkan pemecahan masalah baru akan dilakukan jika proses pelatihan tersebut selesai, fase tersebut adalah fase *mapping* atau proses pengujian/testing.

Perlu ditekankan bahwa dalam [penelitian symbol, karakter *superscript*  $\lambda$  merupakan indek untuk layer ke- $\lambda$ , dimana  $\lambda = 1, 2, \dots, \Lambda$ . Dengan demikian terdapat sebanyak  $\Lambda$  layer dalam arsitektur ANN, dimana layer ke -  $\Lambda$  adalah output layer.

Dalam arsitektur MPL, setiap layer ke  $\lambda$  terdapat sebanyak  $S^\lambda$  neuron, sehingga matrik weight untuk layer ke- $\lambda$  mempunyai ukuran  $S^\lambda \times S^{\lambda-1}$ . Dan nilai output dari layer ke  $(\lambda-1)$  merupakan input untuk layer yang ke  $\lambda$ , seterusnya sampai pada output layer (layer ke-  $\Lambda$ ), kecuali untuk layer pertama (input) dimana ukuran matrik weight adalah  $S^1 \times R$  dengan R adalah jumlah input.

Dalam mempresentasikan bentuk fungsi yang digunakan dalam proses BP, bila  $n^\lambda$  adalah vector jumlahan pada layer ke  $\lambda$  maka :

$$n^\lambda = W^\lambda y^{\lambda-1} \quad \text{atau} \quad n^\lambda(i) = \sum W^\lambda(ij) y^{\lambda-1}(j) \dots\dots\dots(2.1)$$

dengan asumsi bahwa nilai bias telah tercakup dalam matrik weight W. Output untuk setiap lapisan adalah :

$$y^\lambda = f^\lambda(n^\lambda) \quad \text{atau} \quad y^\lambda(i) = f^\lambda(n^\lambda(i)) \dots\dots\dots(2.2)$$

dimana  $f^\lambda$  adalah fungsi aktivasi yang digunakan pada layer yang ke  $\lambda$ . Arsitektur ini diestimasi dengan menggunakan sekelompok  $Q$  pasangan in-out-output. Simbol-simbol subscript  $k$  merupakan indeks input-output untuk pasangan input-output ke  $k$  adalah  $\{x_k, t_k\}$  dimana  $k = 1, 2, \dots, Q$ . (Sri Kusumadewi 2004 : 93)

### 2.3.1.1. Simbol – simbol yang digunakan

Simbol – simbol yang digunakan pada algoritma pelatihan untuk jaringan backpropagation adalah sebagai berikut :

X input vektor pelatihan  $x=(x_1, \dots, x_i, \dots, x_n)$

T output vektor target  $t=(t_1, \dots, t_k, \dots, t_m)$

$\delta_k$  informasi tentang kesalahan pada unit  $Y_k$  yang disebabkan kembali ke unit tersembunyi

$\delta_j$  informasi tentang kesalahan dari lapisan output ke unit tersembunyi  $Z_j$

$\alpha$  *Learning rate*

$X_i$  unit input I, untuk sebuah unit input, sinyal input dan sinyal output adalah sama yaitu  $x_i$ .



$v_{0j}$  bias pada unit tersembunyi j

$Z_j$  unit tersembunyi input jaringan ke  $Z_j$  disimbolkan dengan  $z\_in_j$

$$z\_in_j = v_{0j} + \sum_i x_i v_{ij}$$

Sinyal output(aktivasi) pada  $Z_j$  disimbolkan dengan  $z_j$

$$z_j = f(z\_in_j)$$

$v_{0k}$  bias pada unit tersembunyi k

$Y_k$  Unit output input jaringan ke  $Y_k$  disimbolkan dengan  $y\_in_k$

$$y\_in_k = w_{0k} + \sum_j z_j w_{jk}$$

Sinyal output(aktivasi) pada  $Y_k$  disimbolkan dengan  $y_k$

$$y_k = f(y\_in_k)$$

### 2.3.1.2. Algoritma BackPropagation

Pelatihan sebuah jaringan yang menggunakan backpropagation terdiri dari 3 langkah, yaitu : pelatihan pola input secara feedforward, perhitungan dan backpropagation dari kumpulan kesalahan dan penyesuaian bobot. Selama kondisi feedforward, masing – masing unit input ( $X_i$ ) menerima sebuah sinyal input dan mengirimkan sinyal tersebut ke masing – masing unit tersembunyi  $Z_1, \dots, Z_p$ . Masing – masing unit tersembunyi kemudia menghitung aktivasi dan mengirimkan aktivasi dan mengirimkan sinyalnya ( $Z_j$ ) ke masing – masing unit ouyput. Masing – masing unit output ( $Y_k$ ) kemudian menghitung aktivasinya ( $y_k$ ) untuk membentuk respon dalam jaringan yang diberi pola input.

Selama pelatihan, masing – masing unit output membandingkan perhitungan aktivasi ( $y_k$ ) dengan nilai target ( $t_k$ ) kemudian menentukan kumpulan kesalahan untuk pola yang ada pada unit. Berdasarkan kesalahan tersebut, nilai faktor  $\delta_k(k=1, \dots, m)$  dihitung. Kemudian  $\delta_k$  digunakan untuk mendistribusikan kesalahan pada unit output ( $Y_k$ ) kembali ke semua unit pada lapisan sebelumnya. (unit yang tersembunyi dihubungkan dengan ( $Y_k$ )). Kondisi semacam ini nanti akan digunakan untuk memperbaiki bobot di antara output dan lapisan yang tersembunyi. Untuk hal yang sama, nilai faktor  $\delta_j(j=1, \dots, p)$  dihitung untuk masing – masing unit – unit yang tersembunyi ( $Z_j$ ). Hal ini tidak diperlukan untuk menyebarkan kesalahan ke lapisan input, tetapi  $\delta_j$  digunakan untuk memperbaiki bobot diantara lapisan tersembunyi dan lapisan input.

Sesudah semua faktor  $\delta$  ditentukan, bobot untuk semua lapisan disesuaikan secara simultan. Penyesuaian bobot  $w_{jk}$  (dari unit tersembunyi  $Z_j$  ke unit output  $Y_k$ ) didasarkan pada faktor  $\delta_k$  dan aktivasi  $z_j$  pada unit tersembunyi  $Z_j$ . Penyesuaian bobot  $v_{jk}$  (dari unit input  $X_i$  ke unit tersembunyi  $Z_j$ ) didasarkan pada faktor  $\delta_j$  dan aktivasi  $x_i$  pada unit input.

Propagasi balik merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma propagasi balik menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (backward). Untuk mendapatkan error ini, tahap perambatan maju (forward propagation) harus

dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid, yaitu :

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

Suatu fungsi aktivasi propagasi balik memiliki ciri utama yaitu berkesinambungan dan adanya peningkatan/perbaikan yang berulang-ulang. Aplikasi propagasi balik yaitu memetakan masukan (input) terhadap target keluaran (output). Tujuan pemetaan adalah untuk melatih jaringan mencapai suatu keseimbangan antara kemampuan merespon pola masukan yang digunakan dalam pelatihan dengan kemampuan untuk memberi respon masukan yang disesuaikan.

### **2.3.1.3. Implementasi BackPropagation**

Dalam proses estimasi dengan menggunakan metode BP, terdapat beberapa permasalahan yang sering kali dijumpai antara lain pemilihan inisialisasi awal, lokal minimum, jumlah neuron pada lapisan hidden dan lain-lain. Meskipun telah ada beberapa metode untuk memperkecil peluang munculnya beberapa masalah tersebut, tetapi tidak menjamin bahwa estimasi BP akan memberikan hasil yang lebih baik.

### **a. Minimum Lokal**

Sesuai dengan bentuk dari fungsi untuk arsitektur MPL, dalam menemukan SSE, terdapat beberapa nilai minimum lokal. Prosedur estimasi seperti *steepest descent* yang menggunakan gradien untuk memperbaiki nilai weight kemungkinan akan terjebak dalam likal minimum. Hal ini berarti bahwa weight yang didapat akan tetap memberikan hasil estimasi yang sesuai untuk fungsi non linier, namun mungkin tidak memberikan hasil estimasi yang baik. Hasil estimasi weight ANN dapat memberikan akurasi yang berbeda-beda dan tidak ada jaminan bahwa prosedur pembelajaran (*learning*) akan menemukan global minimum untuk SSE. Pengulangan algoritma untuk training set yang sama, dengan inisialisasi awal yang berbeda akan memberikan nilai weight yang berbeda pula.

### **b. Jumlah Neuron**

Jumlah neuron pada lapisan output adalah ditentukan dari dimensi output, namun jumlah neuron pada lapisan hidden layer harus dipilih. Ketika jumlah neuron sangat sedikit, maka akan terjadi *underfitting* dan tidak ada pilihan weight

yang mampu memodelkan bentuk fungsi ANN. Fungsi ini akan dapat di estimasi dengan baik apabila terdapat variasi yang terlalu kecil.

Jumlah neuron yang harus ditambahkan pada lapisan hidden, akan memberikan fungsi *cost*  $J$  yang semakin minimum. Dalam kasus yang ekstrim, untuk sebuah jaringan ANN dengan satu lapisan hidden, bila jumlah neuron bertambah sampai mendekati banyaknyaelemen *training set*, maka system ini akan mempunyai akurasi yang sangat tinggi dan fungsi  $J$  akan bernilai mendekati nol. Ketika terlalu banyak neuron yang terdapat pada lapisan hidden, prosedur pembelajaran ANN akan cenderung memodelkan *noise* di dalam *trainingset*, dan ini yang disebut *overfitting*. Untuk menentukan neuron yang sesuai, terdapat dua metode yang sering dipakai, yaitu :

1. *Cross Validation* (validasi silang). Metode ini digunakan untuk menentukan jumlah neuron yang terdapat pada lapisan hidden. Penambahan neuron pada lapisan hidden akan memberikan nilai SSE validasi (SSE untuk *validation set*) yang makin berkurang sampai dengan jumlah neuron tertentu, misalkan  $m$ , ketika jumlah neuron tetap ditambahkan melebihi jumlah  $m$ , maka nilai SSE validasi yang dihasilkan justru akan bertambah. Pada jumlah neuron  $m$  dengan nilai SSE validasi terkecil inilah dipilih. Dengan demikian, pada tahap *learning*, data yang digunakan sebaiknya dibagi menjadi 3 bagian yaitu :

- *Training set*, untuk menentukan nilai weight dan bias pada proses *learning*.

- *Validation set*, untuk menentukan jumlah neuron, dimana data ini tetap diikutsertakan dalam proses *learning*.
- *Prediction set*, untuk melakukan validasi akhir menggunakan data independent yang tidak terlibat dalam proses *learning*.

### **c. Overtraining**

Permasalahan yang sama dengan kasus overfitting terjadi bila terdapat training set yang terlalu banyak. Sebagaimana proses training, nilai fungsi J akan berkurang menuju lokal minimum. Untuk data yang mengandung noise tinggi, butuh waktu lama untuk mencapai lokal minimum, karena algoritma training harus berusaha memodelkan noise tersebut. Konsekuensinya, model akhir tidak dapat digeneralisasi dengan baik karena melibatkan bentuk noise.

Secara heuristic, algoritma training cenderung memodelkan weight kemudian diatur untuk memasangkan ciri-ciri data termasuk noise. Untuk mendapatkan generalisasi yang baik, disarankan untuk menghentikan proses training sebelum mencapai lokal minimum. Dan selanjutnya, metode validasi silang dapat digunakan untuk menentukan kapan harus berhenti. Pada setiap akhir iterasi (*epoch*), fungsi cost yang berhubungan dengan validasi harus diperiksa. Nilai fungsi ini akan berkurang sebagaimana proses training menentukan ciri pemetaan yang dideskripsikan oleh training set, namun saat prosedur training harus dihentikan.

### **d. Inisialisasi Awal**

Pola dari algoritma BP, baik dari segi kekonvergenan ataupun lokasi lokal minimum, sangat peka pada pemilihan inisialisasi awal. Bila matrik inisialisasi weight awal mendekati sebuah lokal minimum, maka algoritma BP akan konvergen pada titik tersebut dengan cepat. Sebaliknya, bila inisialisasi awal terletak pada daerah dimana permukaan error berbentuk relatif datar, maka proses konvergen akan sangat lambat. Karena itu, pengulangan proses traning dengan inisialisasi awal yang berbeda akan memberikan laju konvergen dan titik konvergen yang berbeda.

Penentuan inisialisasi awal dapat dilakukan secara acak, namun harus sesuai dengan kondisi data, terutama bila fungsi aktifasi yang digunakan adalah *logistic sigmoid* atau *hyperbolic tangent*. Sebuah pilihan yang sangat dapat dipertanggung jawabkan berhubungan dengan sebuah single neuron dan jumlah jalur hubungan yang terjadi pada neuron tersebut. Pada suatu layer ke- $\lambda$ , terdapat neuron sejumlah  $S^{\lambda-1}$  dari jumlah neuron pada layer sebelumnya. Nilai weight

dapat dipilih secara acak pada range  $\left[ -\frac{3}{\sqrt{S^{\lambda-1}}} \frac{3}{\sqrt{S^{\lambda-1}}} \right]$ , selain range tersebut,

Faraway dan Chatfield(1998) menjelaskan cara penentuan inisialisasi awal yang lebih mudah. Range yang digunakan adalah tergantung dari nilai traning set absolute yang terbesar. Misalkan nilai absolute terbesar adalah *max*, maka range

inisialisasi awal yang disarankan adalah  $\left[ -\frac{1}{\max} \frac{1}{\max} \right]$

#### e. Laju estimasi (*Learning*)

Nilai laju estimasi  $\eta$  sangat berpengaruh pada kecepatan BP dalam proses konvergen. Nilai laju yang sangat besar akan memberikan kekonvergenan lebih cepat. Untuk menentukan nilai laju yang menyeimbangkan antara kecepatan proses konvergen dengan langkah iterasi ketika mendekati suatu lokal minimum memang sangat sulit. Plot dari permukaan error akan sangat berguna untuk menentukan laju estimasi ini. Bila nilai error “meloncat” pada nilai yang lebih besar pada saat training, berarti nilai laju estimasi yang digunakan terlalu besar.

### **2.3.1.3. Parameter pelatihan BackPropagation**

Parameter-parameter yang turut menentukan keberhasilan proses pelatihan pada algoritma BP :

- Inisialisasi penimbang
- Jenis adaptasi penimbang
- Learning rate/laju pelatihan
- Momentum
- Penentuan jumlah lapisan tersembunyi

#### **Inisialisasi Penimbang**

Penimbang interkoneksi jaringan saraf tiruan yang akan dilatih biasanya diinisialisasikan dengan harga real kecil, penimbang bias diinisialisasi secara acak. Banyak studi empiris membuktikan bahwa meneruskan pelatihan pada saat error mencapai harga kecil yang stabil atau datar akan menghasilkan nilai-nilai penimbang yang tidak diinginkan.



Pada banyak penelitian (Hirose et al, 1991) menunjukkan bahwa konvergensi tidak akan dicapai bila penimbang kurang bervariasi, juga jika acaknya terlalu kecil. Konvergensi hampir selalu tercapai untuk inisialisasi acak pada 0.5 sampai 0,5 atau -1 sampai 1.

### **Adaptasi penimbang**

Ada dua adaptasi penimbang pada pelatihan jaringan saraf tiruan, yaitu :

#### a. Adaptasi Kumulatif (*Cumulative Weight Adjustment*)

Penimbang baru diadaptasikan setelah semua penimbang yang masuk dilatih

#### b. Adaptasi Biasa (*Incremental Updating*)

Penimbang diadaptasi pada setiap pola

### **Parameter Laju Pelatihan( $\eta$ )**

Parameter laju pelatihan (*learning rate*) sangat berpengaruh pada intensitas dalam proses pelatihan. Begitu pula terhadap efektivitas dan kecepatan mencapai konvergensi dari pelatihan.

Harga  $\eta$  yang cukup kecil menjamin penurunan gradien terlaksana dengan baik, namun ini berakibat bertambahnya jumlah iterasi. Biasanya nilai laju pelatihan tersebut dipilih mulai 0,001 sampai 1 selama proses pelatihan.

### **Momentum**

Disamping koefisien laju pelatihan, pada metoda BP ada koefisien lain yang tujuan penggunaannya adalah mempercepat konvergensi dari algoritma error *backpropagation*. Prinsip dari metode ini adalah menambahkan sebagian dari perubahan penimbang sebelumnya. Hal ini dapat dirumuskan dengan :

$$\Delta w(t) = \eta \nabla E(t) + \alpha \Delta w(t-1) \dots\dots\dots (2.21)$$

dengan  $\alpha$  adalah harga konstanta momentum yang berupa bilangan positif antara 0,5 sampai dengan 0,9.

Penggunaan koefisien momentum ini disarankan apabila konvergensi berlangsung terlalu lama, dan juga untuk mencegah terjadinya optimum local (*local optimum/minimum*).

### **Fase Pemakaian**

Pada fase pemakaian, pola yang akan dikenali pada unit lapisan masukan jaringan. Inisialisasi penimbang sambungan diambil dari nilai penimbang sambungan terakhir pada tahap pelatihan yang dianggap paling baik tentunya. Pola ini di komputasikan dengan penimbang interkoneksi hasil fase pelatihan dengan persamaan seperti berikut :

$$z_{in_j} = v_{0j} + \sum_i x_i \cdot v_{ij} \dots\dots\dots (2.22)$$

dengan menggunakan fungsi aktivasi untuk menghitung nilai aktivasi :

$$z_j = f(z_{in_j}) \dots\dots\dots (2.23)$$

dengan fungsi aktivasi adalah fungsi sigmoid maka :

$$z_j = \frac{1}{1 + \exp(-z_{in_j})} \dots\dots\dots (2.24)$$

Nilai inilah yang akan dikirimkan kelapisan keluaran, seperti proses pada lapisan tersembunyi dan didapat hasil keluaran adalah :

$$y_{in_k} = w_{0j} + \sum_i z_j \cdot w_{jk} \dots\dots\dots (2.25)$$

Hasil aktivasi sel-sel pada lapisan keluaran merupakan keputusan dari jaringan safaf tiruan.

Pada perangkat lunak yang dibuat, hasil aktivasi sel sebagai keputusan JST ditampilkan dalam skala kebenaran, dimana skala kebenaran terbesar merupakan keputusan akhir dari jaringan safar tiruan.

#### **2.3.1.4. Kelemahan yang sering terjadi pada BackPropagation**

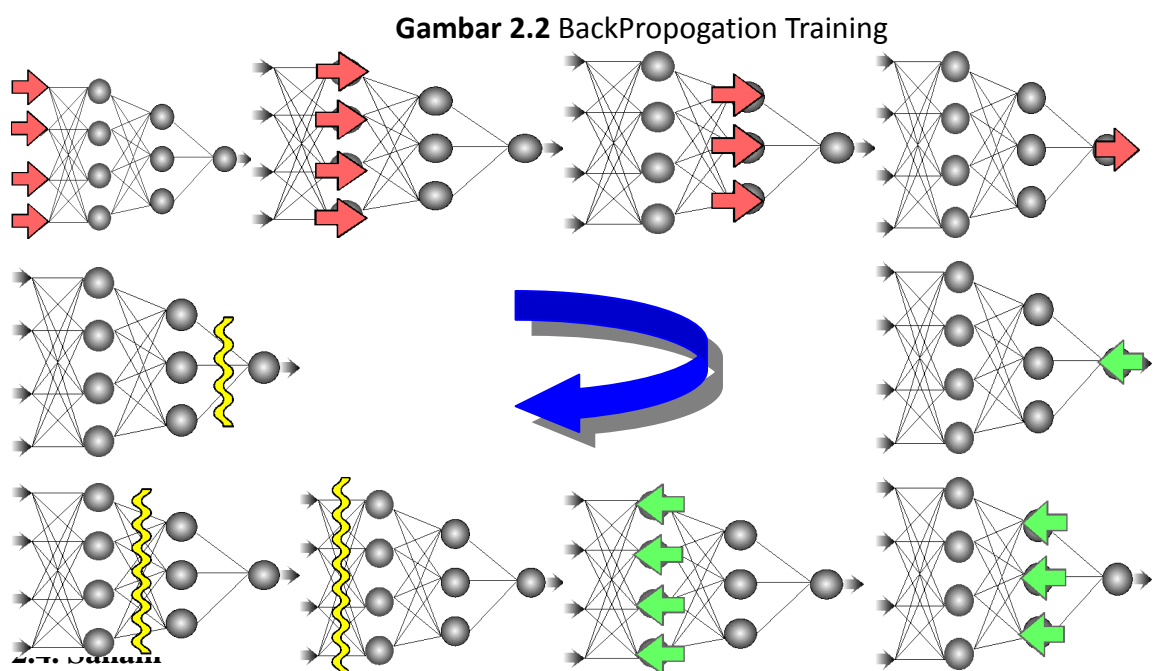
##### **Fase pelatihan**

Sering terjebak pada kondisi yang disebut sebagai kondisi optimum lokal (*local minimum*), sehingga sulit mencapai konvergen sesuai dengan error yang diinginkan. Keadaan JST tersebut sulit atau bahkan tidak dapat berkembang menjadi lebih “pintar” lagi, walaupun dilatih berulang-ulang kali. Hal ini biasanya diatasi dengan menggunakan koefisien momentum.

##### **Fase Pemakaian (Mapping)**

Pada fase ini hal tidak diinginkan yang serng terjadi adalah kondisi dimana JST kehilangan *overfitting/overtraning*, yaitu suatu kondisi dimana JST kehilangan sifat generalisainya (*lost of generalization*). Generalitas adalah kemampuan untuk memberikan jawaban yang benar, untuk suatu masukan yang berbeda namun sejenis dari data-data yang dilatihkan pada JST tersebut.

Bila JST kehilangan sifat tersebut, pada fase mapping biasanya JST hanya mengenali bagian akhir dari pasangan data yang dilatihkan pada fase pelatihan. Kondisi tersebut diakibatkan oleh jumlah pelatihan yang berlebihan, untuk mengatasinya dapat ditambahkan bias pada lapisan keluaran ataupun pada masing-masing lapisan struktur JST. Sebaliknya kondisi *undertraining* juga dapat terjadi, untuk hal ini biasanya terjadi oleh karena proses pelatihan belum sempurna.



Definisi saham adalah suatu tanda penyertaan atau kepemilikan dalam suatu perusahaan. Jika kita membeli saham artinya kita membeli bagian kepemilikan suatu perusahaan. Wujud saham hanyalah berupa selembar kertas yang menerangkan siap pemiliknya, yaitu berbentuk sertifikat saham. Karena itu

saham juga sering disebut surat berharga atau efek (dalam bahasa Inggris disebut sebagai *security*).

Saham (*stock*) merupakan salah satu instrument pasar keuangan yang paling populer dan paling banyak dipilih para investor karena mampu memberikan tingkat keuntungan yang menarik. Berdasarkan pendapat Rusdin (2005 :72) saham adalah sertifikat yang menunjukkan bukti kepemilikan suatu perusahaan, dimana pemegang saham memiliki hak klaim atas penghasilan dan aktiva perusahaan serta berhak hadir dalam Rapat Umum Pemegang Saham (RUPS)

Sedangkan Ang (1997:2) berpendapat bahwa “saham adalah surat berharga sebagai bukti penyertaan atau kepemilikan individu maupun institusi dalam suatu perusahaan berbentuk Perseroan Terbatas”

#### **2.4.1. Fungsi Pasar Saham**

Setiap negara memiliki pasar saham masing-masing. Di Indonesia adalah BEI (Bursa Efek Indonesia atau Indonesia Stock Exchange). Informasi tentang BEI dapat dilihat di <http://www.idx.co.id>. Perdagangan saham di Bursa Efek Indonesia diatur oleh PT. Bursa Efek Indonesia yang merupakan lembaga pemerintah. Pengawasan dilakukan oleh OJK (Otoritas Jasa Keuangan, dahulu Bapepan). Semua kegiatan pasar modal diatur melalui UU Pasar Modal.

Pasar saham, sering juga disebut bursa saham, atau juga disebut bursa efek adalah wahana dimana dilakukan perdagangan saham dan instrumen finansial lainnya. Pasar saham pada hakekatnya tidak berbeda jauh dengan pasar tradisional

yang selama ini kita kenal. Di sana ada pembeli penjual dan juga kegiatan tawar – menawar harga

Di pasar saham juga ada 4 peran yang sama, yaitu investor, perantara pedagang efek, Emitmen, dan pengelola bursa yaitu PT Bursa Efek Indonesia.

### **Investor(Pemodal)**

Investor disini bisa berupa investor individu (ritel) dan investasi institusi (lembaga atau perusahaan). Dilihat dari asalnya bisa dibedakan menjadi dua, yaitu investor domestik dan investor asing

### **Perantara Pedagang Efek**

Sebagai investor, kita tidak bisa langsung bertransaksi di pasar saham, tetapi melalui perantara pedagang efek, yaitu perusahaan sekuritas. Sering pula disebut broker atau pialang

### **Emitmen**

Perusahaan yang melepas sahamnya di pasar saham. Dana hasil penjualan saham tersebut umumnya digunakan untuk ekspansi atau operasi perusahaan. Di bursa saham, kita hanya bisa membeli saham perusahaan atau emitmen yang terdaftar di bursa saham tersebut, disebut perusahaan publik atau perusahaan terbuka.

### **PT. Bursa Efek Indonesia**

Pengelola bursa saham indonesia adalah PT. Bursa Saham Indonesia.

### **2.4.2. Indeks Harga Saham**

Indeks harga saham adalah indikator atau cerminan pergerakan harga saham, bisa seluruh saham atau sekelompok saham. Indeks merupakan salah satu pedoman bagi investor untuk melakukan investasi di pasar modal, khususnya saham. Bila suatu indeks harga saham bergerak naik, biasanya sebagian besar harga saham yang tergabung di dalam juga cenderung naik. Demikian pula sebaliknya.

Setiap pasar saham memiliki indeks harga sahamnya masing-masing. Tabel di bawah ini menunjukkan beberapa indeks saham dari berbagai negara.

Tabel 2.3 Indeks Saham Berbagai Negara

Negara	Nama Indeks Harga Saham
Amerika	Dow Jones, S&P500, Nasdaq
Inggris	FTSE

Jerman	DAX
Perancis	CAC
Australia	ASX
Jepang	Nikkei
China	SSE (Shanghai), HangSeng (Hong Kong)
Korea	KOSPI
India	SENSEX
Indonesia	IHSG (sering juga disebut JCI), LQ45
Malaysia	KLCI
Singapura	STI
Thailand	SET
Vietnam	VHINDEX