BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Khawa Sukmawati dan Ardi Pujiyanta (2014) meneliti tentang "Deteksi Penyakit Tulang Menggunakan Jaringan Syaraf Tiruan Dengan Metode *Backpropagation*". Metodologi yang digunakan dalam penelitian ini adalah studi pustaka, observasi dan wawancara. Untuk tahap pengembangan sistem meliputi analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem dengan metode black box test dan alpha test.

Dalam penelitian ini, gejala – gejala penyakit tulang yang digunakan sebagai input untuk mendeteksi penyakit terdiri dari 42 gejala dan 10 macam jenis penyakit. Arsitektur jaringan syaraf tiruan dengan 42 neuron input, menggunakan 1 hidden layer dengan melakukan perubahan jumlah neuron hidden sebagai percobaan yaitu 21 neuron hidden kemudian diganti menjadi 42 neuron hidden, serta terdapat 10 neuron output. Perangkat lunak ini dibuat menggunakan bahasa pemrograman Borland Delphi 7 dan *Microsoft Acces 2010.*

Sistem yang dibangun mampu melakukan pendeteksian dengan tingkat akurasi yang terhitung cukup baik dengan variasi pelatihan yang terdiri dari galat yang telah ditentukan yaitu sebesar 0.05, sedangkan variasi nilai α yaitu

0.1, 0.3, 0.6, 0.9, lalu jumlah *neuron* hidden layer terhitung sebanyak 42 buah, fungsi aktivasi yang dipakai dalam sistem ini adalah fungsi logsig (*sigmoid biner*), lalu metode pelatihan yang digunakan dalam penelitian ini adalah metode *trial and error*. Tingkat akurasi kebenaran dalam proses pengujian dari sistem yang telah dibangun ini adalah sebesar 90% dengan *learning rate*(α) = 0.1 dan 42 *neuron hidden*.

Budi Soesilo (2011) melakukan sebuah penelitian yang berjudul "Pemanfaatan Jaringan Saraf Tiruan untuk Mendeteksi Gangguan Paru-paru Menggunakan Metode *Backpropagation*". Peneliti merancang sistem dengan membuat Use Case Diagram, Conseptual Data Model (CDM), Pysical Data Model (PDM), desain database, desain akses user dan desain antarmuka. Dari hasil penelitian ini diperoleh kesimpulan bahwa metode ini dapat mendeteksi gangguan paru-paru dengan hasil akhir *error* terbaik yaitu 0,025% dari 160 data masukan sebagai data pelatihan dengan jumlah *hidden layer* = 10, α = 0.01, *target error* = 0.05.

Proses uji coba dilakukan dengan diberikan 160 data masukan sebagai data pelatihan yang juga termasuk data pengujian. Proses uji coba diawali dengan memasukan variabel-variabel pelatihan yang tepat yang dkemudian mampu memberikan tingkat *error* paling kecil.

Rayendra (2019) melakukan penelitian dengan judul "Sistem Deteksi Penyalahgunaan Narkoba Menggunakan Jaringan Saraf Tiruan Model

Backpropagation". Pada penelitian ini disusun data jenis narkoba dan turunannya. Kemudian seluruh gejala psikis yang ditimbulkan untuk turunan narkoba tersebut. Data input dibuat dengan memberikan nilai 1 untuk gejala yang dimiliki pada pengguna narkoba jenis turunan tertentu, dan 0 untuk gejala yang tidak dimiliki.

Pembelajaran dan pengujian dilakukan untuk data gejala klinis dan psikis yang ada atau nilai 1 dengan beberapa kali literasi untuk mendapatkan output apakah narkoba jenis narkotika, psikotropika, zat adiktif atau tidak teridentifikasi. Menentukan variabel jumlah sel lapisan masukan, jumlah sel lapisan tersembunyi, jumlah sel lapisan keluaran, galat yang dijinkan, konstanta belajar, kenaikan konstanta belajar, penurunan konstanta belajar meomentum, rasio kesalahan dan fungsi aktivasi.

Pengujian dilakukan menggunakan perangkat lunak matlab yang diuji dengan beberapa bentuk arsitektur jaringan. Arsitektur dengan konfigurasi terbaik terdiri dari 23 lapisan masukan, 5 lapisan tersembunyi dan 5 lapisan keluaran dengan nilai learning rate sebesar 0.5, nilai toleransi error 0.01, Dapat dilihat bahwa hasil penelitian ini mencapai tingkat akurasi sebesar 95% dari data penelitian.

Dapat disimpulkan bahwa jaringan syaraf tiruan dengan *backpropagation* mampu mengenali pola dan mampu mendeteksi penyalahgunaan narkotika dengan jumlah iterasi 6341 dan MSE 0.00999889.

MSE berada di bawah nilai error yaitu 0.01, Parameter tersebut dipilih menjadi parameter terbaik, karena nilai MSE paling kecil dari arsitektur yang lain serta nilai MSE dibawah dari nilai error yang ditentukan.

Sigmoid Biner digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode backpropagation. Fungsi sigmoid memiliki nilai range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan saraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1.

Penulis melakukan uji coba terhadap 21 orang penderita. Setelah dilakukan uji coba, akan didapat hasil keluaran data yang berupa angka yang berkisar dari 0 hingga 1 untuk ketiga *layer* yang kemudian tersimpan kedalam suatu variabel bertipe data *array*. Jika hasil keluaran pada kolom ketiga menghasilkan > 0.5 maka hasil keluaran akan dianggap 0 0 1 yang kemudian diinisialisasikan sebagai Narkotika, apabila hasil keluaran pada kolom kedua menghasilkan > 0.5 maka hasil dianggap 0 1 0 yang diinisialisasikan sebagai Psikotropika lalu jika hasil keluaran pada kolom ketiga menghasilkan > 0.5 maka hasil dianggap 1 0 0 yang kemudian diinisialisasikan sebagai zat adiktif.

Dian Utari (2013) melakukan sebuah penelitian yang berjudul "*Aplikasi Prakiraan Curah Hujan Menggunakan Jaringan Syaraf Tiruan Backpropagation*". Pada penelitian ini penulis membuat suatu aplikasi atau sistem yang dapat digunakan untuk menguji metode jaringan syaraf tiruan dalam suatu prakiraan curah hujan.

Bahasa pemrograman yang digunakan untuk merancang sistem ini yaitu Java J2SE dengan menggunakan Netbeans IDE 6.7.1 sebagai *tool* untuk membangun aplikasi tersebut. Untuk perangkat lunak tambahan yaitu IBM Rational Rose enterprise edition yang digunakan untuk mendesain Sistem dengan UML dalam aplikasi ini.

Aplikasi ini dibangun dengan tujuan untuk menguji kemampuan Jaringan Syaraf Tiruan dalam melakukan prakiraan curah hujan dengan mengenali pola-pola data masa lalu, aplikasi ini menggunakan 12 node input, 36 *node hidden* dan 1 *output* data dengan *learning rate* 0-0.9 dan fungsi aktifasi sigmoid biner, secara umum aplikasi ini telah dapat melakukan prakiran curah hujan, dengan rata-rata akurasi 56 %, prakiraan dengan tingkat akurasi paling besar sebanyak 100 %.

Supryanto, Sunardi, dan Riadi (2022). “Pengaruh Nilai Hidden Layer Dan Learning Rate Terhadap Kecepatan Pelatihan Jaringan Syaraf Tiruan *Backpropagation*”. Pada penelitian ini dibutuhkan perangkat lunak dan perangkat keras untuk melakukan pelatihan dan pengujian JST backpropagation. Perangkat keras berupa laptop/netbook sedangkan perangkat lunak menggunakan sistem prediksi curah hujan yang dikembangkan menggunakan bahasa pemrograman PHP.

Setelah dilakukann proses pelatihan JST backpropagation untuk prediksi curah hujan, dapat disimpulkan JST backpropagation dalam proses

pelatihan membutuhkan kombinasi parameter yang tepat seperti hidden layer, learning rate, dan jumlah iterasi karena terkait dengan waktu proses pelatihan dan galat yang akan dihasilkan. Pada penelitian ini dilakukan pengujian terhadap parameter masukan dari kombinasi nilai hidden layer, learning rate dan jumlah iterasi untuk mengetahui sejauh mana waktu yang dibutuhkan oleh JST backpropagation dalam melakukan pelatihan terhadap data masukan.

Hasil dari pelatihan jaringan, Parameter jumlah nilai hidden layer sebesar 12 neuron untuk mendapatkan waktu pelatihan yang cepat sebanyak 3 menit 44 detik dengan galat MSE 1,654151. Pada Jumlah iterasi yang dibutuhkan dengan arsitektur hidden layer 12 neuron, dibutuhkan sebanyak 100000 iterasi dengan waktu pelatihan 21 menit 52 detik. Sedangkan pada parameter learning rate menggunakan 12 neuron dan iterasi sebesar 100000, nilai leraning rate rate yang tepat untuk pelatihan sebesar 0,5 dengan waktu yang dibutuhkan untuk pelatihan 18 menit 35 detik dengan galat MSE 0,302868.

Tabel 2.1 Tinjauan Pustaka

Parameter / Penulis	Objek	Metode	Bahasa Pemrograman	Interface
Khawa Sukmawati dan Ardi Pujiyanta (2014)	Deteksi Penyakit Tulang Menggunakan Jaringan Syaraf Tiruan Dengan Metode *Backpropagation*	JST	Borland Delphi 7 dan *Microsoft Acces 2010*	GUI
Budi Soesilo (2011)	Pemanfaatan Jaringan Saraf Tiruan untuk Mendeteksi Gangguan Paru-paru Menggunakan Metode *Backpropagation*	JST	Java	GUI
Dian Utari (2013)	Aplikasi Prakiraan Curah Hujan Menggunakan Jaringan Syaraf Tiruan *Backpropagation*	JST	Java	GUI
Supryanto, Sunardi, dan Riadi (2022)	Pengaruh Nilai Hidden Layer Dan Learning Rate Terhadap Kecepatan Pelatihan Jaringan Syaraf Tiruan *Backpropagation*	JST	PHP	Text
Rayendra (2019)	Sistem Deteksi Penyalahgunaan Narkoba Menggunakan Jaringan Saraf Tiruan Model *Backpropagation*	JST	Matlab	GUI

2.2 Dasar Teori

2.2.1 Tuberculosis Paru

Tuberkulosis didefinisikan sebagai salah satu jenis penyakit menular yang disebabkan oleh kuman *Myobacterium tuberculosis* (TB). Kebanyakan bakteri *Myobacterium tuberculosis* menginfeksi paru-paru, namun selain itu bakteri TBC ini juga dapat menyerang organ lain di dalam tubuh hingga terjadinya infeksi. Tuberkulosis adalah penyakit menular langsung yang disebabkan oleh kuman TB (Werdhani, 2009).

Tuberkulosis atau biasa disingkat dengan TBC juga termasuk golongan penyakit kronis dan penyebab utamanya adalah infeksi kompleks *Mycobacterium tuberculosis* yang penularannya terjadi melalui media dahak (droplet) dari penderita TBC kepada individu lain yang rentan (Ginanjar, 2008). Bakteri *Mycobacterium tuberculosis* merupakan kurus, batang ramping, yang tahan akan asam sehingga sering disebut dengan bakteri tahan asam (BTA). Bakteri ini dapat berbentuk lurus ataupun bengkok yang panjangnya sekitar 2-4 μm dan lebar 0,2 –0,5 μm yang bergabung membentuk rantai. Besar bakteri ini tergantung pada kondisi lingkungan (Ginanjar, 2010).

Berikut adalah gejala-gejala umum yang dialami oleh pasien penderita (Muttaqin: 2008).

1. Batuk berdahak adalah gejala pada pasien yang timbul paling awal serta menjadi gejala yang paling sering diderita oleh pasien pengidap penyakit TBC. Batuk akan disertai dengan darah Jika penyakit TBC sudah cukup parah.

2. Gejala yang kedua adalah sesak napas, gejala sesak napas ini dapat terjadi apabila terjadi kerusakan yang luas pada parenkim selain itu sesak napas terjadi akibat hal yang lain diantaranya seperti efusi adanya cairan di dada (*pleura*), rongga pleura terisi udara (*pneumothoraks*), kekurangan sel darah merah (*anemia*), dan lain-lain.

3. Nyeri dada pada TBC termasuk nyeri pleuritik ringan (dada terinfeksi bakteri). Gejala nyeri dada akan terjadi jika sistem pernapasan di pleura terkena bakteri TBC.

4. Gejala yang ke empat adalah terjadinya demam terutama pada sore atau malam hari, gejala ini sering merupakan salah satu gejala yang paling sering dijumpai dan biasanya timbul pada sore atau malam hari mirip demam *influenza*, yaitu demam yang terjadi hilang timbul, dan semakin lama semakin panjang serangannya, sedangkan masa bebas serangan semakin pendek.

5. Gejala berikutnya yang sering timbul pada pasien adalah pasien mengalami keringat terutama pada malam hari, *anoreksia* atau yang biasa

dikenal dengan gangguan makan yang ditandai dengan berat badan yang sangat rendah, dan pasien mengalami gejala badan lemah (*malaise*). Timbulnya keluhan biasanya bersifat gradual muncul dalam beberapa minggu – bulan.

2.2.2 Jaringan Syaraf Tiruan (JST)

Jaringan syaraf tiruan (*Artificial Neural Network*) merupakan metode komputasi yang mengikuti cara kerja biologis otak manusia, jaringan syaraf tiruan tersusun dari beberapa bagian pemrosesan (*neuron*) dan terdapat kaitan di antara *neuron*. *Neuron* akan mengubah bentuk dari data yang diterima oleh *neuron* satu yang kemudian akan diteruskan ke *neuron* yang lainnya, koneksi ini disebut yang kemudian disebut dengan bobot. Deboek dan Kohonen mendeskripsikan bahwa jaringan syaraf tiruan adalah kumpulan metode komputasi yang mampu melakukan pemrosesan sinyal, peramalan serta pengelompokan dan disebut sebagai metode regresi paralel non linier, *multilayer* (Shanmuganathan, 2016).

Setiap proses pengolahan berlangsung komponen akan membuat komputasi atas jumlah inputan atau masukan. Serangkaian pemrosesan elemen disebut dengan lapisan atau *layer* dalam jaringan. *Layer* atau lapisan pertama merupakan *layer input* sedangkan yang terakhir merupakan *output*. *Layer* yang berada diantara lapisan *input* dan lapisan *output* adalah lapisan tersembunyi (*hidden layer*). Salah satu algoritma yang sering dipakai dalam paradigma

jaringan syaraf tiruan adalah perambatan galat mundur atau yang biasa disebut dengan algoritma *backpropagation* (Hermawan, 2006).

2.2.3 Arsitektur Jaringan Syaraf Tiruan

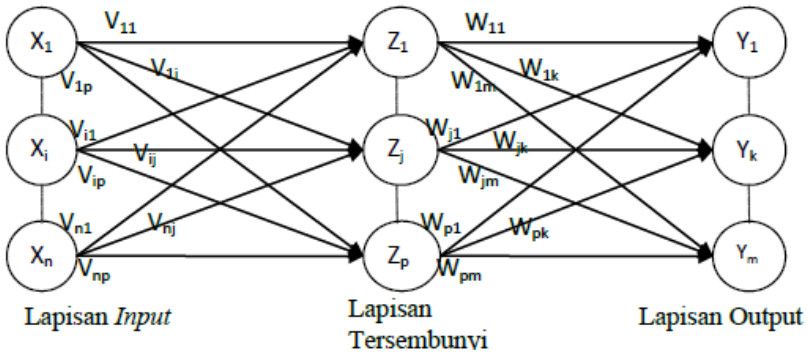
Seluruh model jaringan memiliki konsep dasar yang sama dimana Jaringan syaraf tiruan dirancang dengan menggunakan suatu aturan yang bersifat menyeluruh (*general rule*). Keberhasilan target yang akan dicapai ditentukan oleh arsitektur sebuah jaringan karena tidak semua permasalahan dapat diselesaikan dengan arsitektur yang sama.

Jaringan Saraf Tiruan (JST) memiliki beberapa pengertian, diantaranya adalah Menurut Haykin (2009), Jaringan Saraf Tiruan merupakan prosesor yang tersalur secara besar – besaran dalam bentuk parallel yang terdiri dari unit proses sederhana, unit proses sederhana ini mampu menyimpan pengetahuan berupa pengalaman hasil dari pelatihan dan dapat diperguanakan juga untuk melakukan proses yang lain.

Dalam jaringan syaraf tiruan, Neuron-neuron dikelompokkan dalam suatu lapisan yang disebut lapisan *neuron*. JST dalam penelitian ini terdiri dari tiga komponen yaitu lapisan *input*, *hidden layer*, dan *output layer*. *Neuron* pada sebuah lapisan akan terhubung ke lapisan sebelumnya begitu juga dengan lapisan berikutnya. Setiap informasi yang diteruskan pada jaringan syaraf juga akan diteruskan dari lapisan ke lapisan berikutnya, diawali dengan *input layer*

sampai ke *output layer* kemduian melintasi lapisan *hidden layer* atau yang biasa disebut dengan *multilayer network*.

Jaringan ini memiliki satu atau lebih lapisan yang terletak antara lapisan *input* dan lapisan *ouput*. Biasanya tedapat *layer* dengan bobot-bobot yang terletak diantara dua lapisan yang bersebelahan. *Multilayer network* mampu memproses dan mengolah permasalahan yang dianggap lebih kompleks dibandingkan dengan arsitektur yang lain, akan tetapi dengan proses pembelajaran yang lebih kompleks pula. Dalam banyak kasus, proses pelatihan *(training)* pada *multi layer net* ini terbukti lebih mampu menyelesaikan masalah yang dianggap kompleks atau rumit. Dibawah ini adalah gambar arsitektur JST *multi layer net*.



Gambar 2.1 Multi Layer Network

2.2.4 Backpropagation

Smith (2003), *backpropagation* adalah salah satu dari beberapa metode yang digunakan dalam JST dan telang digunakan di berbagai aplikasi dalam berbagai bidang, seperti peramalan dan optimisasi, pengenalan pola. *backpropagation* memungkinkan hal ini karena metode *backpropagation* sendiri memakai proses pembelajaran terbimbing. Pola *input* dan pola target ditetapkan sebagai sepasang data.

Bobot awal dilatih dengan melalui tahap maju (*forward)* untuk mendapatkan *output error*, selanjutnya *error* tersebut digunakan sebagai tahap (*backward*) untuk memperoleh nilai bobot yang telah ditetapkan agar nilai *error* dapat diperkecil sehingga nilai *output* yang telah ditetapkan sebelumnya terpenuhi.

Tujuan dari pemanfaatan metode ini ialah guna memperoleh keseimbangan pada kemampuan jaringan memberikan respon yang benar terhadap pola masukan yang berbeda dengan pola masukan pelatihan serta kemampuan jaringan dalam mengenali pola yang digunakan selama proses pelatihan berlangsung.

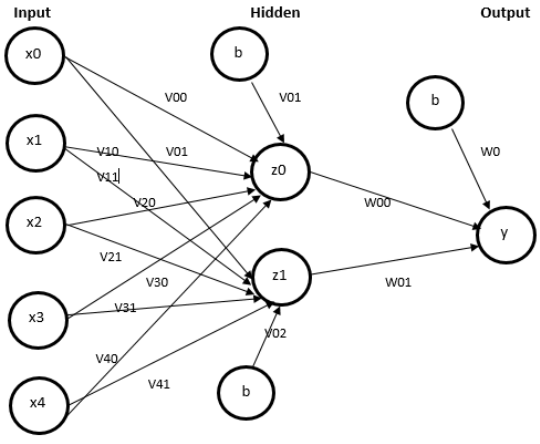
Algoritma perambatan balik (Backpropagation) merupakan salah satu algoritma JST sering dimanfaatkan dalam melakukan proses identifikasi. Dalam berbagai kasus pengenalan pola-pola kompleks metode perambatan balik merupakan metode yang sering digunakan.

Istilah perambatan balik diambil dari cara kerja jaringan ini, yaitu dilakukan berdasarkan perbedaan antara keluaran dengan target maka dihitung gradien kesalahan atau *error* pada unit-unit lapisan, yang kemudian hasilnya digunakan dalam proses komputasi gradien *error* pada unit-unit lapisan sebelumnya. (Limin Fu, 1994).

2.2.5 Arsitektur Backpropagation

Pada jaringan *backpropagation*, semua unit yang terdapat di dalam lapisan input harus saling terhubung dengan setiap unit yang ada di lapisan tersembunyi. Demikian juga dengan *hidden layer*.

Semua unit pada *hidden layer* harus saling terhubung dengan semua unit yang ada di *output layer*. Jaringan syaraf tiruan *backpropagation* tersusun atas banyak lapisan (*multilayer neural networks*) yaitu *input layer* (1 buah), *hidden layer* (minimal 1), dan *output layer* (1 buah).



Gambar 2. 2 Arsitektur Backpropagation

Keterangan:

x_i, $i \in \{0, 1, 2, 3, 4\}$: memiliki fungsi sebagai *neuron* pada *input layer.*

z_j, $j \in \{0, 1\}$: memiliki fungsi sebagai *neuron hidden layer.*

v_{ij}, $i \in \{0, 1, 2, 3, 4\}$ $j \in \{0, 1\}$: merupakan bobot yang berada di antara *input layer* dan *hidden layer*.

b: Bias merupakan sebuah unit masukkan yang nilainya *default* 1.

w_{jk}, $j \in \{0\}$ $k \in \{0, 1\}$: merupakan bobot yang terletak di antara *hideen layer* dengan *output layer*.

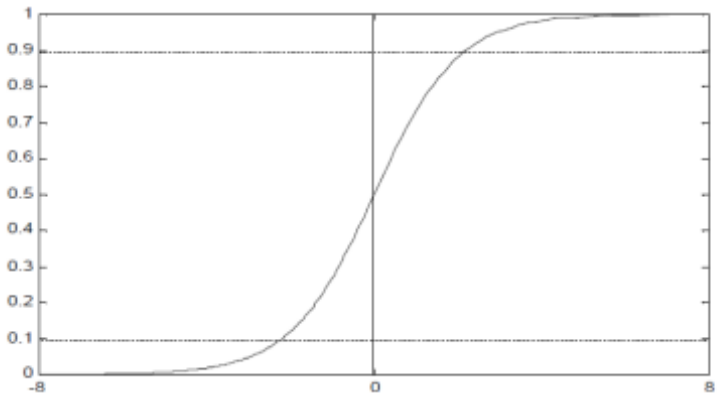
y_k (y) : berfungsi sebagai *ouput layer*.

2.2.6 Fungsi Aktivasi

Fungsi aktivasi adalah sebuah fungsi yang dipakai dalam JST yang benfungsi untuk mengaktifkan dan menonaktifkan *neuron*. Beberapa sifat yang dimiliki oleh fungsi aktivasi jaringan *backpropagation* yaitu harus berisfat *continue*, terdiferensialkan, dan tidak menurun secara monotonik (*monotonically non-decreasing*). Selain itu, pada kemampuan komputasi, turunan dari fungsi aktifasi terhitung mudah didapatkan selain itu nilai turunan dari fungsi tersebut dapat dinyatakan oleh fungsi aktivasi itu sendiri. Fungsi *sigmoid biner* adalah fungsi yang akan digunakan pada penelitian ini, fungsi

aktivasi *sigmoid biner* memiliki nilai antara 0 hingga 1. Berikut adalah gambar fungsi aktivasi *sigmoid biner*:

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



Gambar 2.1 Fungsi Sigmoid biner

2.2.7 Algoritma Pelatihan Backpropagation

Muh Insan (2019). Dalam menggunakan metode Backpropagation dilakukan 2 tahap, yaitu tahap pelatihan atau tahap pembelajaran dimana pada tahap ini digunakan untuk melatih dengan berbagai data yang diberikan, lalu dilanjutkan dengan tahap pengujian atau penggunaan, pada tahap ini dilakukan pengujian pada tahap pelatihan.

Step 0: Inisialisasi bobot dan bias baik bobot maupun bias dapat diset dengan sembarang angka (acak) dan biasanya angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif)

Step 1: Jika stopping condition masih belum terpenuhi, jalankan step 2-9.

Step 2: Untuk setiap data training, lakukan step 3-8.

- Umpan maju (*feedforward*)

Step 3: Setiap unit yang dipakai di sini adalah input training data yang sudah diskalahkan

Step 4: Setiap hidden unit $(Zj\ , j = 1, \dots, p)$ akan menjumlahkan sinyal-sinyal input yang sudah berbobot, termasuk biasnya dengan persamaan berikut.

$$Zinj = V0j + \sum_{i=1}^{n} Xi\ Vij$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari hidden unit yang bersangkutan, melalui persamaan berikut.

$$Zj = f(Zinj)$$

lalu mengirim sinyal output ini ke seluruh unit pada unit output

Keterangan

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

$V0j$ = bias pada unit tersembunyi j

Zj = unit tersebunyi j input jaringan Zj disimbolkan dengan Z_inj

$\sum$ = sigma

n = Jumlah data latih

Step 5: Setiap unit output (Yk, k=1,…,m) akan menjumlahkan sinyal-sinyal input

yang sudah berbobot, termasuk biasnya

$$y_{ink} = w_{ok} + \sum_{j=1}^{p} Z_j W_{jk}$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung

sinyal output dari unit output yang bersangkutan

$$y_{ink} = f_{(ink)}$$

Keterangan:

y_k = unit output input jaringan ke y_k disimbolkan dengan y_{ink}

w_{ok} = nilai bias

Propagasi balik error (backpropagation of error)

Step 6: Setiap unit output ($Yk, k = 1, \dots, m$) menerima suatu target (output yang diharapkan) yang akan dibandingkan dengan output dengan persamaan berikut

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

Faktor δ_k ini digunakan untuk menghitung koreksi error (ΔW_{jk}) yang nantinya akan dipakai untuk memperbaharui Wjk, dengan persamaan berikut

$$\Delta W_{jk} = \alpha \delta_k z_j$$

Faktor δ_k ini kemudian dikirimkan ke layer di depannya Keterangan:

y = output neuron

t = target data latih

δ_k = informasi tentang kesalahan pada unit Yk yang disebarkan kembali ke unit tersembunyi.

α = learning rate

Δ = delta

ΔW_{jk} = koreksi error

Step 7: Setiap hidden unit $(Zj, j = 1, \dots, p)$ menjumlah input delta (yang dikirim darilayer pada step 6) yang sudah berbobot

$$\delta_{inj} = \sum_{k=1}^{m} \delta_k \, W_{jk}$$

Hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan

jaringan untuk menghasilkan faktor koreksi error δ_j

$$\delta_j = \delta_{inj} f'(z_{inj})$$

Faktor δ_j ini digunakan untuk menghitung koreksi error (ΔV_{ij}) yang

nantinya akan dipakai untuk memperbaharui V_{ij}

$$\Delta V_{ij} = \alpha \delta_j x_i$$

Selain itu juga dihitung koreksi bias ΔV_{0j} yang nantinya akan dipakai untuk memperbaharui V_{0j}

$$\Delta V_{oj} = \alpha \delta_j$$

Keterangan:

δ_j= informasi tentang kesalahan dari lapisan output ke unit tersembunyi

Z_j X_i = unit input I

ΔV_{ij} = koreksi error bias pada unit tersembunyi j

Pembaharuan bobot dan bias:

Step 8:

1. Setiap unit output (Yk, k=1,…,m) akan memperbaharui bias dan bobotnya dengan setiap hidden unit
2. $W_{jk}(baru) = W_{jk}(lama) + \Delta W_{jk}$
3. Demikian pula untuk setiap hidden unit akan memperbaharui bias dan bobotnya dengan setiap unit input.
    1. $v_{ij}(baru) = v_{ij} + \Delta v_{ij}$
4. Keterangan:
    1. v_{0j} =bias pada unit tersembunyi j

Step 9: Memeriksa stopping condition, jika stop condition telah terpenuhi, maka pelatihan jaringan dapat dihentikan.

- Stopping Condition

Untuk menentukan stopping condition terdapat dua cara yang biasa

dipakai, yaitu:

1. Membatasi iterasi yang ingin dilakukan.

Misalnya jaringan akan dilatih sampai iterasi yang ke-500. Yang dimaksud dengan satu iterasi adalah perulangan step 3 sampai step 8 untuk semua training data yang ada.

2. Membatasi error.

Misalnya menentukan besar Mean Square Error antara output yang dikehendaki dan output yang dihasilkan oleh jaringan. Jika terdapat sebanyak m training data, maka untuk menghitung Mean Square Error digunakan persamaa berikut.

$$MSE = 0{,}5X\{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \cdots + \{t_{km} - y_{km}\}^2\}$$

Keterangan:

MSE = Mean Square Error

t = output vector target $t = (t_1, \ldots., t_k, \ldots., t_m)$

Y_k = unit output

Tahap pengujian & Penggunaan

Mengimplementasikan metode backpropagation pada tahap pengujian sama seperti proses belajar, tetapi hanya pada bagian umpan majunya saja :

Step 0: Inisialisasi bobot sesuai dengan bobot yang telah dihasilkan pada proses pelatihan di atas.

Step 1: Untuk setiap input, lakukan step 2-4.

Step 2: Untuk setiap input $i = 1, \dots, n$ skalakan bilangan dalam range fungsi aktivasi

seperti yang dilakukan pada proses pelatihan di atas.

Step 3: Untuk $j = 1, \dots, p$:

$$z_{inj} = v_{0j} + \sum_{i=1}^{m} x_i v_{ij}$$

Step 4: Untuk $k = 1, \dots, m$:

$$y_{inj} = w_{inj} + \sum_{j=1}^{p} z_j w_{jk}$$

$$y_k = f(y_{ink})$$

2.2.8 Mean Square Error

Mean Square Error (MSE) digunakan untuk menghitung nilai error set pembelajaran dengan hasil ideal yang ditentukan sebelumnya. Metode *mean square error* efektif dalam menghitung nilai *error* tanpa menghiraukan apakah nilai aktual berada di bawah atau di atas hasil yang ideal (Heaton, 2008).

MSE digunakan sebagai parameter untuk keakuratan nilai target keluaran. Semakin kecil nilai *mean square error* tidak menjamin semakin tinggi nilai akurasinya (Riztyan et al., 2013). Berikut adalah rumus untuk mendapatkan MSE:

$$MSE = 0{,}5X\{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \cdots + \{t_{km} - y_{km}\}^2\}$$

Keterangan:

MSE = Mean Square Error

t = output vector target $t = (t_1, \ldots., t_k, \ldots., t_m)$

Y_k = unit output

2.2.9 Pengenalan Pola

Abidin (2010), menurut Gonzalez dan Woods, pola adalah deskripsi kuantitatif atau struktural dari suatu objek atau entitas yang menarik dari sebuah citra. Pola biasanya dibentuk dari satu atau lebih ciri dari citra tersebut. Dengan demikian, pola dapat diartikan sebagai susunan dari beberapa ciri dari citra.

Contoh beberapa ciri yang di temukan pada sebuah citra adalah warna, histrogram, arah, dan magnitudo tepi, luas daerah dan sebagainya.

Ilmu yang mempelajari klasifikasi atau representasi pola suatu benda berdasarkan sifat-sifat benda tersebut adalah pengenalan pola. Berikut adalah dua jenis pengenalan pola yakni:

1. Deteksi

   Deteksi dapat didefinisikan sebagai proses menyelidiki atau mengidentifikasi sesuatu dengan menggunakan metode atau teknik tertentu. Deteksi dapat digunakan dalam berbagai masalah, salah satunya adalah dalam mendeteksi penyakit, dimana sistem akan menyelidiki atau melakukan proses identifikasi terhadap permasalahan yang mempunyai hubungan dengan penyakit yang dikenal dengan gejala.

   Deteksi memiliki tujuan yaitu melakukan proses pemecahan satu kasus namun dengan menggunakan banyak cara tergantung metode yang ingin dipakai sehingga menciptkan sebuah solusi untuk memecahkan masalah tersebut.

2. Pengenalan

   Tujuan dari pengenalan pola adalah untuk melakukan klasifikasi berdasarkan ciri yang terdapat pada pola tersebut.

Klasifikasi tersebut bertujuan untuk mengidentifikasi suatu objek dalam citra (Sari, 2010). Contohnya adalah, dalam mengenali wajah pada citra, wajah kemudian akan dideteksi lalu diidentifikasi yang kemudain akan dibandingkan dengan wajah yang telah dilakukan proses pengenalan sebelumnya sehingga sistem dapat menentukan siapa pemilik wajah tersebut.

2.2.10 Parameter Pelatihan

Berdasarkan penelitian Lu Yifei (2017), proses learning menggunakan parameter sebagai berikut:

1. *Epoch*

Epoch adalah ketika seluruh dataset sudah melalui proses training pada Neural Netwok sampai dikembalikan ke awal untuk sekali putaran, karena satu *Epoch* terlalu besar untuk dimasukkan (*feeding*) kedalam komputer maka dari itu kita perlu membaginya kedalam satuan kecil (*batches*).

2. *Error Goal*

Error goal adalah salah satu parameter dalam proses pelatihan dimana jika error goal atau target error tellah terpenuhi maka proses pelatihan, sehingga error goal termasuk kedalam stoping condition.

3. *Hidden Layer*

Lapisan antara input layer dan output layer, dimana artificial neuron yang memiliki sekumpulan input pembobot ‘weight’ dan prosedur untuk menghasilkan output neuron melalui *activation function.*

4. *Learning Rate*

Merupakan salah satu parameter *training* untuk menghitung nilai koreksi bobot pada waktu proses *training*. Nilai *learning rate* ini berada pada *range* nol (0) sampai (1). Semakin besar nilai *learning rate*, maka proses *training* akan berjalan semakin cepat. Semakin besar *learning rate*, maka ketelitian jaringan akan semakin berkurang, tetapi berlaku sebaliknya, apabila *learning rate*-nya semakin kecil, maka ketelitian jaringan akan semakin besar atau bertambah dengan konsekuensi proses *training* akan memakan waktu yang semakin lama.

2.2.11 Node

Node adalah tempat di mana komputasi terjadi, sebuah node menggabungkan input dari data dengan serangkaian koefisien, atau bobot, yang memperkuat input, sehingga memberikan sebuah signifikansi pada input sehubungan dengan tugas yang akan dipelajari oleh algoritma; misalnya input

mana yang paling membantu adalah mengklasifikasikan data tanpa kesalahan. Selanjutnya input ini dijumlahkan dan kemudian hasil perhitungannya diteruskan melalui fungsi aktivasi, untuk menentukan bagaimana dan sejauh mana sinyal itu berkembang melalui jaringan untuk mempengaruhi hasil akhir.

2.2.12 Matrix Laboratory (Matlab)

Matlab dapat didefinisikan sebagai suatu bahasa berkinerja tinggi yang dapat melakukan proses komputasi dalam berbagai kebutuhan di bidang teknik. Matlab melalukan penggabungan terhadap komputasi, visualisasi, dan pemrograman menjadi suatu model yang dimana masalah-masalah dan penyelesaiannya dinyatakan dalam notasi matematika yang mudah untuk pakai. Penggunaan Matlab meliputi:

1. Matematika dan komputasi.
2. Penyusunan algoritma.
3. Akusisi data.
4. Pemodelan, simulasi, dan pembuatan *prototype.*
5. Analisa data, explorasi, dan visualisasi.
6. Grafik keilmuan dan bidang rekayasa.

Nama Matlab sendiri adalah singkatan dari matrix laboratory. Di lingkungan sekolah kejuruan teknik, Matlab merupakan perangkat yang fungsi

utamanya adalah melakukan perkenalkan dan pengembangan penyajian materi matematika, teknik dan ilmiah. Di industri, Matlab adalah alat yang sempurna untuk melakukan penelitian, pengembangan, dan analisis yang produktif. Banyak fitur-fitur Matlab yang telah dikembangkan, yang dikenal dengan nama *toolbox*.

Seorang pengguna Matlab harus mengetahui toolbox mana yang harus digunakan untuk mandukung kajian ilmiah yang sedang dipelajarinya. *Toolbox* dapat didefinisikan sebagai kumpulan dari beberapa fungsi Matlab (m-files) yang telah di kembangkan ke *workspace* Matlab yang berfungsi untuk memecahkan masalah ilmiah. Berikut adalah beberapa teknik komputasi yang dapat dipecahkan mengguanakan *toolbox* saat ini diantaranya adalah pengolahan sinyal, *system kontrol*, *neural networks*, *fuzzy logic*, *wavelets*, dan lain-lain.

2.2.13 Matlab App Designer

App Designer adalah sebuah fitur pada matlab yang dapat digunakan pengguna dalam menganalisis permasalahan yang membutuhkan penanganan khusus. *App designer* sendiri menyediakan berbagai fitur, seperti menu, *pushbutton*, *slider*, dan sebagainya sesuai dengan ke kebutuhan pengguna.

App designer memberikan cara untuk efisiennya manajemen data. *App designer* sangat memungkinkan pengguna untuk membuat aplikasi profesional dengan cara *drag and drop* komponen visual untuk menyusun desain

antarmuka pengguna grafis (GUI) yang ingin dibuat kemudian gunakan editor terintegrasi untuk memprogram apliaksi.

2.2.14 Keunggulan Matlab *App Designer*

App designer mempunyai kelebihan tersendiri dibandingkan dengan bahasa pemrograman lainnya, antara lain:

1. *App designer* dinilai sangat cocok dalam mengembangkan aplikasi-aplikasi berorientasi sains, sehingga sangat membantu mahasiswa ataupun peneliti dalam menyelesaikan tugas akhir dan menyelesaikan riset mereka.
2. *App designer* memiliki berbagai fitur yang dapat digunakan pengguna dalam mengembangkan riset.
3. Ukuran file yang dihasilkan oleh *app designer* relatif berukuran kecil.
4. *App designer* memiliki tingkat visual cukup baik jika dibandingkan dengan bahasa pemrograman lainnya.

2.2.15 Machine Learning

Machine Learning adalah salah satu cabang dari disiplin ilmu kecerdasan buatan (AI) yang bekerja dengan menggunakan data. Sehingga *machine learning* dapat didefinissikan sebagai sebuah proses komputer melakukan pembelajaran melalui data masa lalu melalui proses pelatihan

(*training*). Salah satu contoh penggunaan *machine learning* pada aplikasi adalah *supervised learning*. *Supervised learning* adalah sebuah teknik untuk melakukan prediksi dimana hasil prediksinya bersifat diskrit. Model klasifikasi yang digunakan adalah membandingkan nilai aktual dengan nilai prediksi.

2.2.16 Confusion Matrix

Confusion matrix merupakan salah satu indikator performa dalam masalah klasifikasi *machine learning* yang outputnya bisa di lebih dari satu kelas. *Confusion matrix* juga dapat didefinisikan sebagai sebuah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan menggambarkan hasil dari proses klasifikasi pada *confusion matrix* yaitu *true positif*, *true negatif*, *false positif*, dan *false negatif*. (Hermoza et all., 2011), suatu *confusion matrix* merupakan alat yang berguna untuk menganalisis seberapa baik pengklasifikasi tersebut dapat mengenali *tupel* dalam kelas-kelas yang berbeda.

1. *True Positive* (TP):

   Sistem melakukan proses prediksi pada data positif TBC dan hasil dari proses deteksi tersebut adalah positif.

2. *True Negative* (TN):

   Sistem melakukan proses prediksi pada data negatif TBC dan hasil dari proses deteksi tersebut adalah negatif TBC.

3. *False Positive* (FP): (Kesalahan Tipe 1)

   Sistem melakukan proses prediksi pada data negatif TBC dan hasil dari proses deteksi tersebut adalah positif.

4. *False Negative* (FN): (Kesalahan Tipe 2)

   Sistem melakukan proses prediksi pada data positif TBC dan hasil dari proses deteksi tersebut adalah negatif.

   maka dapat dihitung nilai *akurasi dengan rumus dibawah ini:*

   *Akurasi= ((TP+TN) / (TP+FP+FN+TN)) *100%*

2.2.17 Bobot dan Bias

Bobot adalah salah satu dari beberapa faktor penting supaya jaringan mampu melakukan proses pembelajaran dengan baik (Fitrisia dan Rakhmatsyah, 2010). Pemilihan inisialisasi bobot awal akan sangat berpengaruh pada jaringan. Pembaharuan antara dua buah *neuron* begantung pada kedua turunan fungsi aktivasi yang digunakan pada *neuron* yang berada pada *layer* sebelumnya dan juga fungsi aktivasi *neuron* yang berada pada *layer* setelahnya.

Untuk inisialisasi bobot awal nilainya disarankan agar tidak terlalu besar, jika tidak sinyal pada setiap *hidden neuron* dan *neuron* keluaran akan berada pada tempat dimana turunan terkecil dari fungsi *sigmoid*. Sehingga dapat dikatakan jika *input* jaringan ke *hidden* atau *output neuron* akan

mendekati nol jika inisialisasi bobot awal terlalu kecil, sehingga membuat laju proses pelatihan menjadi agak lambat (Fausset, 1994).