Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

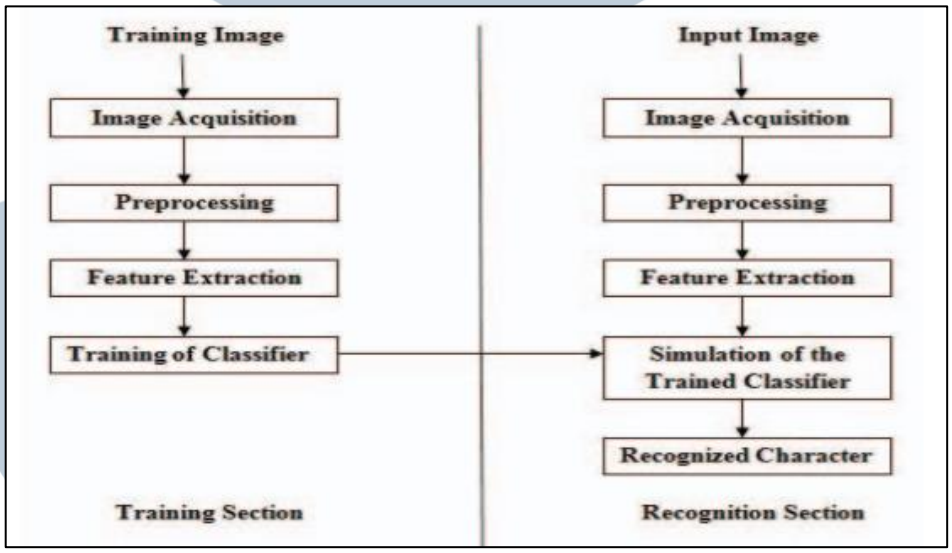
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) merupakan teknologi untuk pengenalan karakter yang biasa digunakan untuk mengubah teks dalam citra digital menjadi format teks yang dapat diolah komputer. Dalam penelitian Afroge dkk. (2016), OCR terdiri dari dua bagian yaitu bagian pelatihan dan pengenalan. Selain itu, Afroge dkk. (2016) menunjukan bahwa sistem OCR terdiri dari *image acquisition*, *preprocessing*, dan *feature extraction*, serta *training of the classifier* pada bagian pelatihan dan *simulation of the trained classifier* pada bagian pengenalan. Ilustrasi sistem OCR dapat dilihat pada Gambar 2.1.



Gambar 2.1 Sistem *Optical Character Recognition* (Afroge dkk., 2016)

1. *Image Acquisition*

   Image acquisition merupakan proses pengambilan input berupa citra digital yang diperoleh dengan menggunakan alat pemindai atau kamera.

UNIVERSITAS MULTIMEDIA NUSANTARA

2. *Preprocessing*

Preprocessing merupakan pemrosesan awal yang dilakukan untuk mengolah atau memproses input yang berupa citra digital menjadi data yang dapat dioperasikan oleh sistem OCR. Berdasarkan pada penelitian yang dilakukan oleh Afroge dkk. (2016), dalam proses ini dilakukan beberapa tahapan yaitu *gray image conversion*, *noise reduction*, *thresholding*, *binarization*, *line extraction*, *character extraction*, dan *normalization*. *Gray image conversion* digunakan untuk mengubah citra digital menjadi *gray scale image*. *Noise reduction* digunakan untuk meningkatkan kualitas dokumen. *Thresholding* digunakan untuk mengatur nilai latar belakang untuk piksel di bawah nilai ambang batas dan menggunakan serangkaian nilai yang berbeda untuk latar depan. *Binarization* digunakan untuk mengubah citra digital menjadi nilai biner. *Line extraction* digunakan untuk mengekstrasi baris dengan memindai baris dari atas ke bawah dan menghitung jumlah piksel baris. *Character extraction* digunakan untuk mengekstraksi karakter dengan memindai wilayah teks dari kiri ke kanan dan menghitung jumlah piksel kolom. Sementara, *normalization* digunakan untuk menyamakan ukuran semua bitmap karakter yang diekstraksi.

3. *Feature Extraction*

Dalam penelitian Afroge dkk. (2016), *feature extraction* digunakan untuk mengekstrak ciri numerik dari data asli. Menurut penelitian yang dilakukan oleh Nugroho (2003), *feature extraction* dapat tidak dilakukan namun mengakibatkan proses pelatihan membutuhkan waktu yang lebih lama.

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

4. *Training of the classifier*

   Training of the classifier merupakan proses pelatihan yang digunakan untuk mengklasifikasikan data menggunakan jaringan saraf tiruan *backpropagation*. Pelatihan dilakukan dengan banyak sampel agar pengenalan dapat dicapai.

5. *Simulation of the trained classifier*

   Simulation of the trained classifier merupakan proses pengujian untuk pengenalan karakter yang dilakukan pada jaringan terlatih.

2.2 Pengolahan Citra Digital

Pengolahan citra digital merupakan pemrosesan citra dua dimensi yang dilakukan melalui komputer dengan tujuan untuk memperbaiki kualitas citra, menarik informasi yang terkandung dalam citra, atau melakukan kompresi data pada citra untuk tujuan penyimpanan, transmisi, atau waktu proses data.

2.2.1 Gray Scale Image

Gray scale image merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian *RED = GREEN = BLUE* (Putra, 2010). Nilai tersebut digunakan untuk merepresentasikan intensitas warna hitam dan putih. Intensitas pada *gray scale image* berkisar antara 0 sampai 255 dimana nilai 0 sampai nilai kurang dari 255 menyatakan intensitas warna hitam semetara nilai 255 menyatakan warna putih.

UNIVERSITAS MULTIMEDIA NUSANTARA

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

2.2.2 Thresholding

Metode *thresholding* adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk objek dan background dari citra secara jelas (Malik, 2014). Menurut Hietania dkk. (2012), *thresholding* merupakan teknik yang sederhana dan efektif untuk segmentasi citra. *Thresholding* diklasifikasikan menjadi dua yaitu *global thresholding* dan *local thresholding* (*adaptive*).

1. *Global Thresholding*

   Global thresholding merupakan salah satu teknik *thresholding* yang paling sederhana. Apabila nilai ambang *t* bergantung hanya pada satu nilai aras keabuan *f*(*y*, *x*), *thresholding* disebut global (Kadir dan Susanto, 2013). *Global thresholding* digunakan ketika intensitas antara objek latar depan dan latar belakang yang sangat berbeda. Ketika perbedaan antara objek latar depan dan latar belakang sangat berbeda, nilai *threshold* global dapat digunakan untuk membedakan kedua objek (Senthilkumaran dan Vaithegi, 2016).

2. *Local Thresholding*

   Local thresholding merupakan salah satu teknik *thresholding* yang menghitung nilai *threshold* secara lokal. *Thresholding* disebut lokal jika nilai ambang *t* bergantung pada *f*(*y*, *x*) dan *g*(*y*, *x*) dengan *g*(*y*, *x*) menyatakan properti citra lokal pada titik (*y*, *x*) (Kadir dan Susanto, 2013). *Threshold* dihitung untuk setiap piksel berdasarkan pada beberapa statistik lokal dari piksel tetangga.

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

2.2.3 Operasi Morfologi

Menurut Kadir dan Susanto (2013), operasi morfologi merupakan operasi yang umum dikenakan pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra.

1. Dilasi

   Dilasi merupakan salah satu operasi morfologi yang digunakan untuk mendapatkan efek pelebaran dengan menambah piksel pada citra digital.

2. Erosi

   Erosi merupakan salah satu operasi morfologi yang digunakan untuk mendapatkan efek penipisan dengan mengurangi piksel pada citral digital.

2.2.4 Operator Canny

Operator *Canny* merupakan operator deteksi tepi yang dikemukakan pada tahun 1986 oleh John F. Canny. Menurut Hermawati (2013), operator ini diformulasikan dengan tiga tujuan utama yaitu *optimal detection* dengan tidak adanya respon yang salah (palsu), *good localization* dengan jarak minimal antara posisi tepi sebenarnya dengan posisi yang terdeteksi, dan *single response* dengan mengeliminasi *multiple response* menjadi satu garis tepi saja.

2.2.5 Gaussian Blur

Gaussian blur adalah filter blur yang menempatkan warna transisi yang signifikan dalam sebuah *image*, kemudian membuat warna-warna pertengahan untuk menciptakan efek lembut pada sisi-sisi sebuah *image* (Wedianto dkk., 2016). *Gaussian blur* memberikan efek blur dengan menggunakan filter *gaussian*.

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

Filter *gaussian* merupakan filter linear dengan nilai pembobotan pada piksel-piksel yang dipilih berdasarkan bentuk fungsi *gaussian*. Filter *Gaussian* biasanya digunakan dalam pengolahan citra digital untuk penghalusan citra atau penghilangan derau (*noise*).

2.2.6 Normalization

Normalisasi karakter dianggap sebagai operasi *preprocessing* yang paling penting untuk pengenalan karakter (Cheriet dkk., 2007). Citra dipetakan pada piksel dengan ukuran yang telah ditentukan sehingga memberikan representasi dimensi yang tetap untuk klasifikasi. Tujuan dari normalisasi adalah mengurangi resolusi citra untuk meningkatkan akurasi pengenalan karakter.

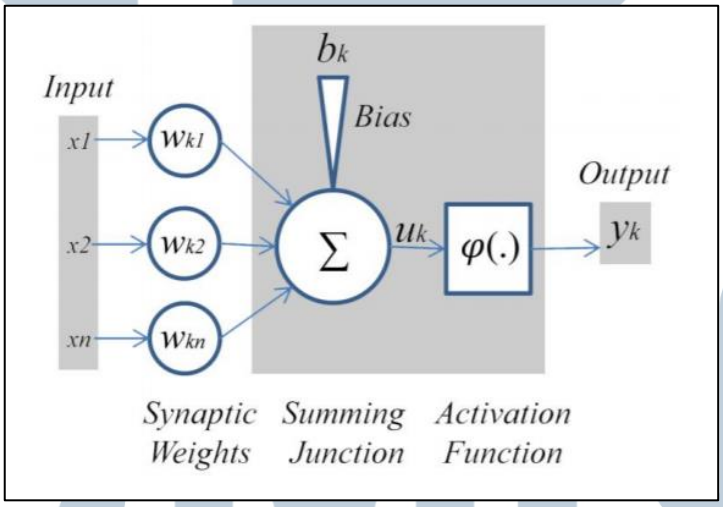
2.3 Jaringan Saraf Tiruan

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti informasi pada otak manusia (Sutojo dkk., 2011). Jaringan saraf tiruan merupakan salah satu upaya manusia untuk memodelkan fungsi dari sistem saraf manusia dalam memproses informasi untuk melaksanakan suatu tugas. Jaringan saraf tiruan tersusun atas sejumlah neuron yang dihubungkan oleh *link*. Setiap *link* memiliki bobot yang digunakan untuk mengirimkan informasi yang diterima dari satu neuron ke neuron lain yang terhubung. Bobot tersebut akan selalu diperbaharui selama proses pembelajaran.

Satu sel saraf dalam jaringan saraf tiruan terdiri dari 3 bagian yaitu *summing function*, fungsi aktivasi, dan *ouput*. *Summing function* adalah fungsi yang digunakan untuk menjumlahkan nilai bobot yang ada pada setiap *link* yang

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

terhubung dengan neuron. Sedangkan, fungsi aktivasi adalah fungsi yang digunakan untuk mengaktifkan neuron. Informasi input pada jaringan saraf tiruan akan dikirimkan ke setiap neuron dengan nilai bobot tertentu. Kemudian, nilai bobot akan dijumlahkan dengan *summing function* dan hasil dari penjumlahan akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang (*threshold*) tertentu, maka neuron tersebut akan diaktifkan. Jika tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot *output*nya ke semua neuron yang berhubungan dengannya, demikian seterusnya (Handoyo dan Susanto, 2011).



Gambar 2.2 Model Saraf (Taravat dkk., 2015)

Jaringan saraf tiruan memiliki beberapa lapisan yaitu lapisan input, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Pada lapisan input, setiap unit menerima pola inputan data dari luar. Pada lapisan *output*, *output* yang dihasilkan dari lapisan ini akan menjadi solusi jaringan saraf tiruan terhadap suatu tugas. Sementara, lapisan tersembunyi (*hidden layer*) adalah lapisan yang

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

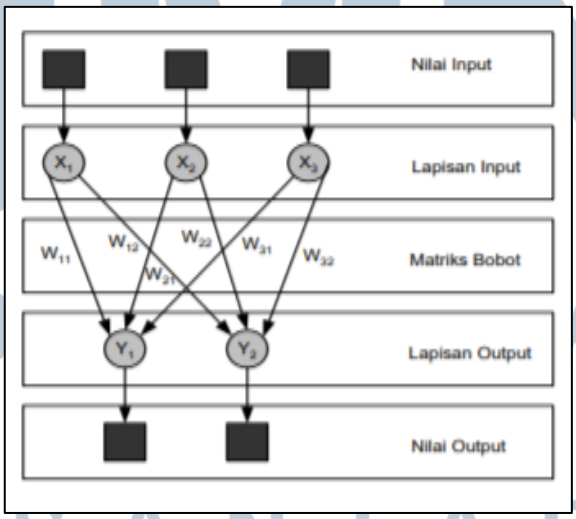
menghubungkan lapisan input dan lapisan *output*. *Output* yang dihasilkan lapisan tersembunyi (*hidden layer*) tidak dapat diamati secara langsung.

2.4 Arsitektur Jaringan Saraf Tiruan

Arsitektur jaringan saraf tiruan merupakan pola keterhubungan antar neuron dalam jarigan saraf tiruan. Jaringan saraf tiruan memiliki beberapa arsitektur yang biasa digunakan dalam berbagai aplikasi diantaranya :

1. Jaringan lapisan tunggal (*single layer network*)

   Jaringan lapisan tunggal (*single layer network*) hanya memiliki satu lapisan dengan setiap bobot yang terhubung. Menurut Winardi (2012), jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melewati lapisan tersembunyi. Dengan kata lain, Jaringan lapisan tunggal (*single layer network*) hanya memiliki satu lapisan input dan satu lapisan *output* tanpa lapisan tersembunyi. Ilustrasi arsitektur jaringan lapisan tunggal (*single layer network*) dapat dilihat pada Gambar 2.3.

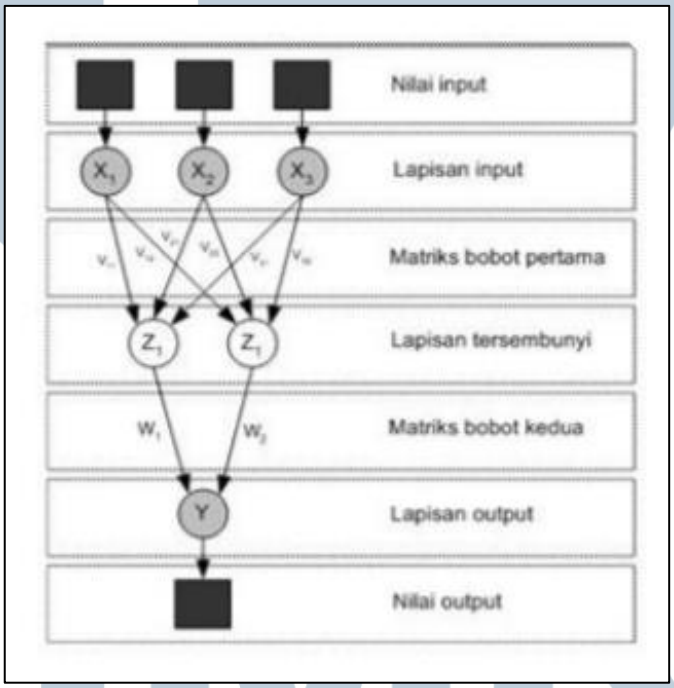


Gambar 2.3 Jaringan Saraf dengan Lapisan Tunggal (Sutojo dkk., 2011)

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

2. Jaringan lapisan banyak (*multilayer network*)

Jaringan lapisan banyak (*multilayer network*) memiliki satu atau lebih lapisan tersembunyi yang berada diantara lapisan input dan lapisan *output*. Menurut Juanita dan Sudira (2013), jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada jaringan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Ilustrasi arsitektur jaringan lapisan banyak (*multilayer network*) dapat dilihat pada Gambar 2.4.



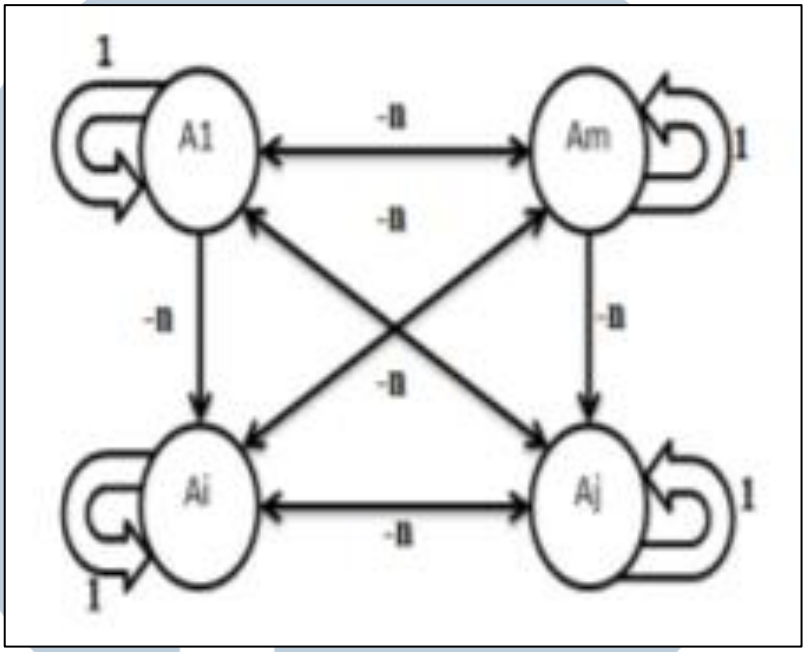
Gambar 2.4 Jaringan Saraf dengan Lapisan Banyak (Sutojo dkk., 2011)

3. Jaringan dengan lapisan kompetitif (*competitive layer network*)

Pada jaringan kompetitif, neuron saling berkompetisi sehingga hanya satu diantaranya yang aktif. Menurut Sutojo dkk. (2011), Jaringan ini digunakan untuk mengetahui neuron pemenang dari sejumlah neuron yang ada. Akibatnya, menurut Sutojo dkk. (2011), pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Ilustrasi arsitektur

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

jaringan saraf tiruan dengan lapisan kompetitif (*competitive layer network*) dapat dilihat pada Gambar 2.5.

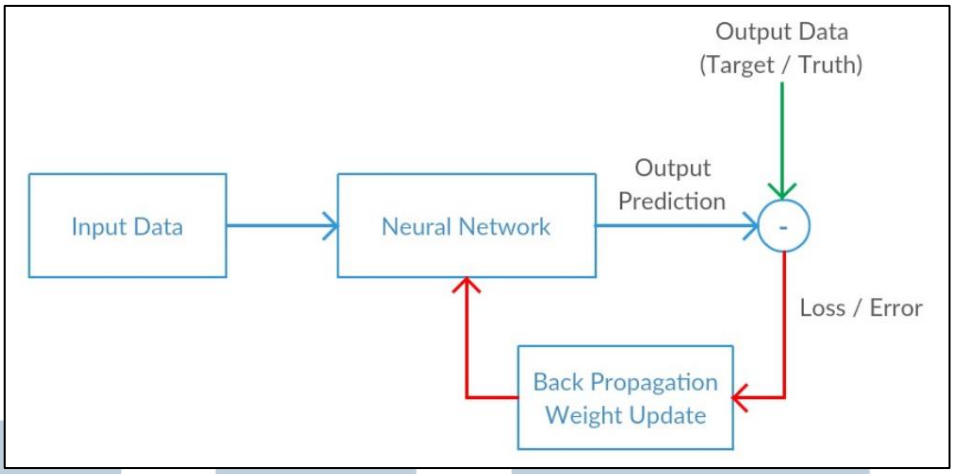


Gambar 2.5 Jaringan Saraf dengan Lapisan Kompetitif (Sutojo dkk., 2011)

2.5 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 (Wirawan, 2017). Proses pelatihan jaringan dengan algoritma ini terdiri dari tiga tahap yaitu tahap perambatan maju (*forward propagation*), tahap perambatan balik (*backpropagation*), dan tahap perubahan bobot dan bias. *Forward propagation* digunakan untuk membawa data pada input sampai ke *output*. Sedangkan, *backpropagation* digunakan untuk menyesuaikan bobot-bobot dan bias berdasarkan nilai *error* yang didapatkan dengan arah mundur. Ilustrasi dari pelatihan pada jaringan saraf tiruan dapat dilihat pada Gambar 2.6.

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019



Gambar 2.6 Pelatihan Jaringan Saraf Tiruan (Sena, 2017)

Langkah-langkah yang dilakukan pada pada proses perubahan bobot adalah sebagai berikut :

1. Inisialisasi nilai bobot dengan menetapkan nilai bobot awal secara acak.
2. Lakukan *forward propagation* selama kondisi berhenti bernilai salah atau nilai *epoch* (set pelatihan pada jaringan) lebih kecil dari nilai maksimum *epoch.*

Langkah-langkah yang dilakukan pada tahap *forward propagation* :

1. Setiap unit input (X_i , i = 1, 2, ... n) menerima sinyal X_i dan mengirim sinyal tersebut ke semua unit pada lapisan tersembunyi (*hidden layer*).
2. Setiap unit tersembunyi (Z_j , j = 1, 2, ... p) menjumlahkan bobot sinyal input dengan persamaan :

$$Zin_j = V_{0j} + \sum_{i=1}^{n} X_i V_{ij} \quad \ldots(2.1)$$

gunakan fungsi aktivasi untuk menghitung sinyal output :

$$Z_j = f(Zin_j) \quad \ldots(2.2)$$

3. Setiap unit tersembunyi (Y_k , k = 1, 2, ... m) menjumlahkan bobot sinyal input dengan persamaan :

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

$$Yin_k = W_{0k} + \sum_{j=1}^{p} Z_j W_{jk} \quad \ldots(2.3)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output* :

$$Y_k = f(Yin_k) \quad \ldots(2.4)$$

setelah itu, kirim sinyal ke lapisan *output*.

Langkah-langkah yang dilakukan pada tahap *backpropagation* :

1. Setiap unit *output* (Y_k , k = 1, 2, ... m) menerima pola target yang sesuai pada pola input pelatihan, hitung informasi *error* dengan persamaan :

$$\delta_k = (t_k - Y_k) f'(Yin_k) \quad \ldots(2.5)$$

hitung koreksi nilai bobot yang akan digunakan untuk memperbaharui bobotnya dengan persamaan :

$$\Delta W_{jk}(t) = \alpha \delta_k Z_j \quad \ldots(2.6)$$

hitung koreksi nilai bias yang akan digunakan untuk memperbaharui biasnya dengan persamaan :

$$\Delta W_{0k}(t) = \alpha \delta_k \quad \ldots(2.7)$$

Kirimkan nilai δ_k ke setiap unit yang berada pada lapisan dibawahnya.

2. Setiap unit pada lapisan tersembunyi (Z_j , j = 1, 2, ... p) jumlahkan hasil perubahan input dari setiap unit pada lapisan diatasnya dengan persamaan :

$$\delta in_j = \sum_{k=1}^{m} \delta_k W_{jk} \quad \ldots(2.8)$$

kalikan nilai dengan turunan dari fungsi aktivasinya untuk menghitung informasi error :

$$\delta_j = \delta in_j f'(Zin_j) \quad \ldots(2.9)$$

hitung koreksi nilai bobot dengan persamaan :

$$\Delta V_{jk}(t) = \alpha \delta_j X_i \quad \ldots(2.10)$$

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

hitung koreksi nilai bias dengan persamaan :

$$\Delta V_{0k}(t) = \alpha \delta_j \quad \ldots(2.11)$$

Langkah-langkah yang dilakukan pada tahap perubahan bobot dan bias :

1. Setiap unit *output* (Y_k , k = 1, 2, ... m) perbaharui nilai bias dan niliai bobot (j = 0 , ... , p) dengan persamaan :

$$\mathrm{W}_{jk}(baru) = \mathrm{W}_{jk}(lama) + \Delta \mathrm{W}_{jk} \quad \ldots(2.12)$$

untuk setiap unit lapisan tersembunyi (Z_j , j = 1, 2, ...p) perbaharui nilai bias dan niliai bobot (i = 0 , ... , n) dengan persamaan :

$$V_{ij}(baru) = V_{ij}(lama) + \Delta V_{ij} \quad \ldots(2.13)$$

2. Tes kondisi berhenti.

2.6 Fungsi Aktivasi

Fungsi aktivasi adalah fungsi yang digunakan pada jaringan saraf tiruan untuk mengaktifkan neuron. Menurut Suhardi (2007), karakteristik yang harus dimiliki oleh fungsi aktivasi jaringan *backpropagation* antara lain harus kontinyu, terdiferensialkan, dan tidak menurun secara monotis (*monotonically non-decreasing*). Beberapa fungsi aktivasi yang biasa digunakan dalam algoritma *backpropagation* diantaranya :

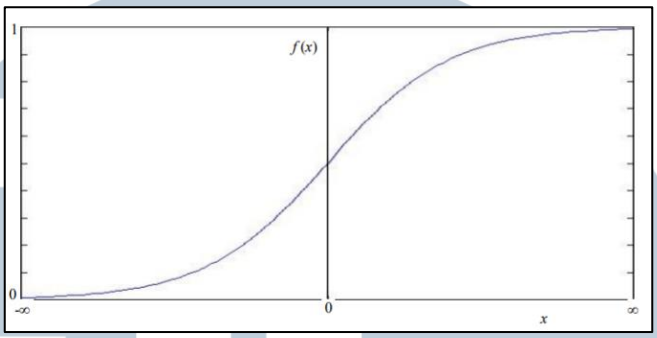
1. Fungsi Sigmoid

Fungsi ini memiliki jangkauan nilai antara 0 sampai 1. Fungsi sigmoid biner diilustrasikan pada Gambar 2.7. Fungsi sigmoid didefinisikan dengan persamaan pada Rumus 2.14 dan fungsi turunan sigmoid didefinisikan dengan persamaan pada Runus 2.15.

$$\mathrm{y} = f(x) = \frac{1}{(1 + e^{-x})} \quad \ldots(2.14)$$

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019

$$y = f'(x) = \frac{1}{(1+e^{-x})} \times (1 - \frac{1}{(1+e^{-x})}) \quad \ldots(2.15)$$



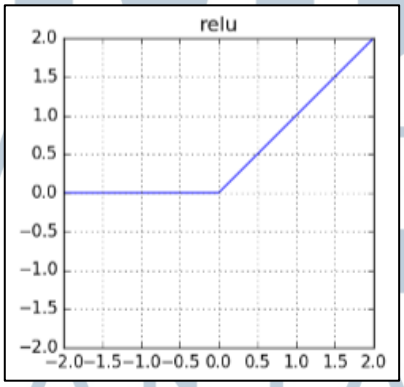
Gambar 2.7 Fungsi Aktivasi Sigmoid (Suhardi, 2007)

2. Fungsi *Rectified Linear Unit* (ReLU)

Rectified linear unit (ReLU) adalah salah satu fungsi aktivasi yang memiliki perhitungan yang sederhana. Menurut Wibawa (2017), kelebihan ReLU akan muncul saat berhadapan dengan jaringan yang memiliki neuron yang banyak sehingga dapat mengurangi waktu training dan testing dengan signifikan. Fungsi ReLU diilustrasikan pada Gambar 2.8. Fungsi ReLU didefinisikan dengan persamaan pada Rumus 2.16 dan fungsi turunan ReLU didefinisikan dengan persamaan pada Rumus 2.17.

$$y = f(x) = \max(0, x) \quad \ldots(2.16)$$

$$y = f'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad \ldots(2.17)$$



Gambar 2.8 Fungsi Aktivasi ReLU (Wibawa, 2017)

Implementasi jaringan saraf..., Dian Puspita Chandra, FTI UMN, 2019