Artificial Neural Network

Kuliah 5 - STA1382 Teknik Pembelajaran Mesin

Septian Rahardiantoro

Outline

- Definisi Neural Network
- ANN – Feed-forward Network
- Ilustrasi

Definisi

Pemodelan Jaringan Syaraf Tiruan (Artificial Neural Network)

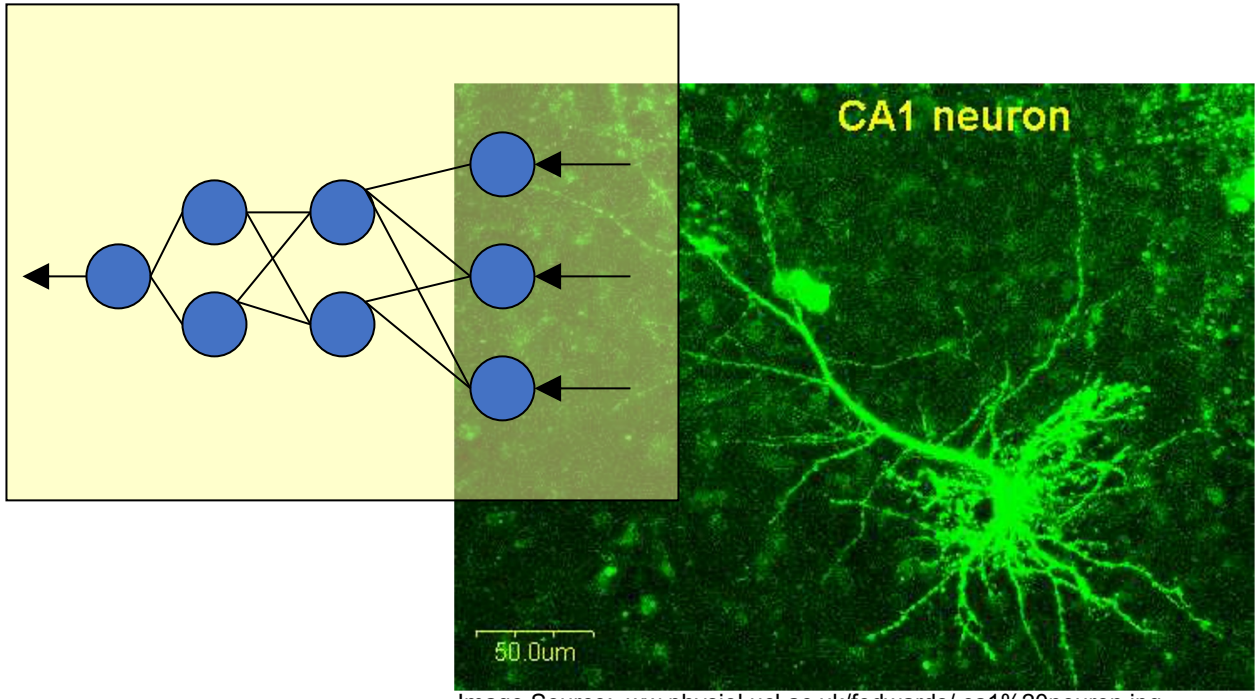


Image Source: ww.physiol.ucl.ac.uk/fedwards/ ca1%20neuron.jpg

Jaringan Syaraf (Neural Network)

Suatu upaya pemodelan yang menirukan fungsi yang ada di otak manusia

Terdapat beberapa kelas model NN yang dibedakan berdasarkan:

- ❑ Tujuan pemodelan
  Prediction, Classification, Clustering
- ❑ Struktur model
- ❑ Algoritma estimasi parameter model

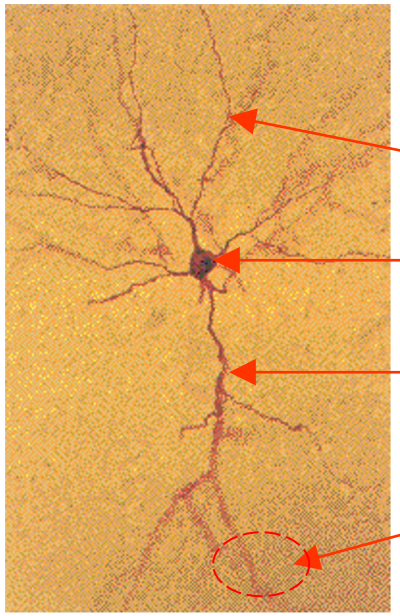
Materi ini akan fokus pada

Feed-forward Back-propagation Neural Network

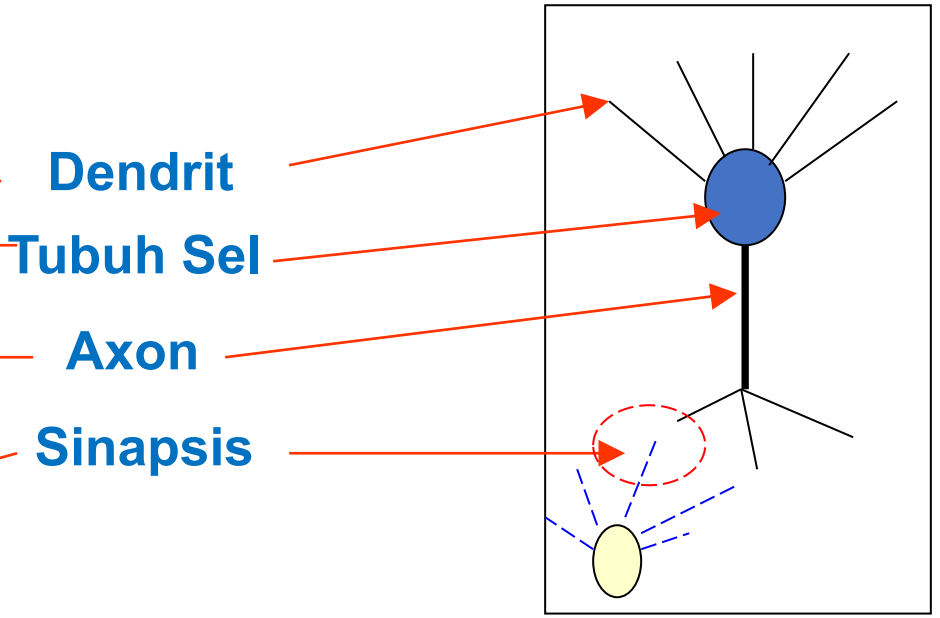
(digunakan untuk permasalahan Prediction and Classification)

A bit of biology . . .

Unit dengan fungsi terpenting dalam otak manusia – jaringan bernama– **NEURON**



Hippocampal Neurons
Source: heart.cbl.utoronto.ca/ ~berj/projects.html



Schematic

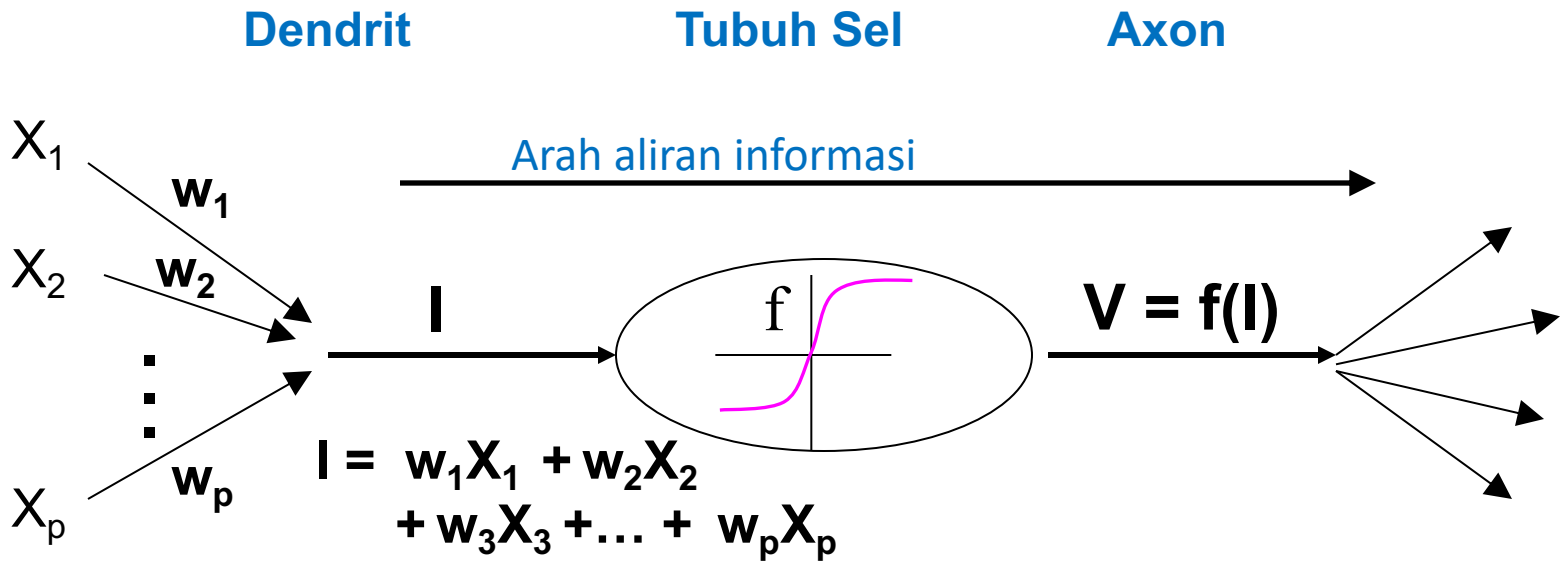
Dendrit – Menerima informasi

Tubuh Sel – Proses informasi

Axon – Membawa informasi yang telah diproses ke neuron lain

Sinapsis – Penghubung antara ujung Axon dan Dendrites neuron lain

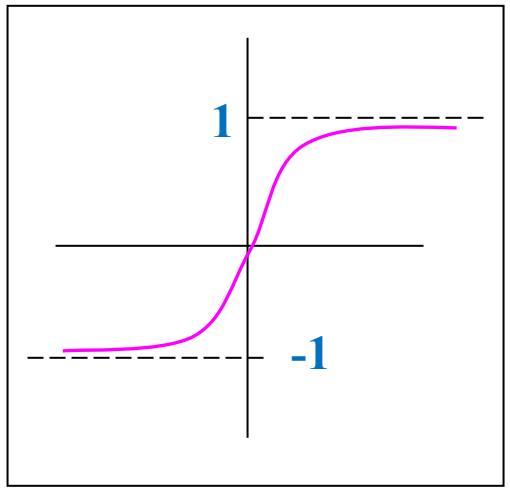
An Artificial Neuron



- Input X_1 X_2 … X_p diterima dari neuron lain atau lingkungan
- Setiap input masuk ke dalam proses ini dengan bobot masing-masing
- Total input = jumlah terboboti semua input dari semua sumber
- Transfer function (Activation function) mengkonversi input ke output
- Output masuk ke neuron lain atau lingkungan

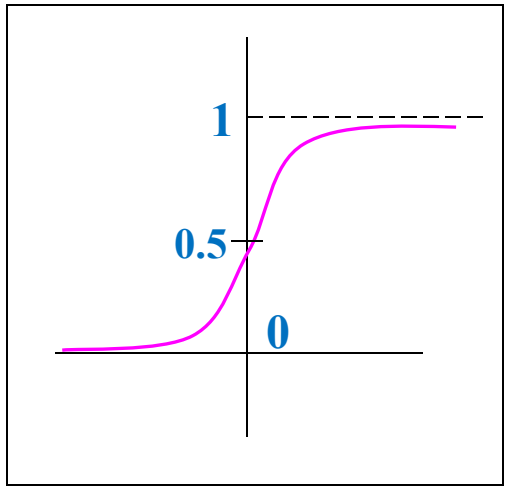
Transfer Functions

Banyak pilihan Transfer / Activation functions



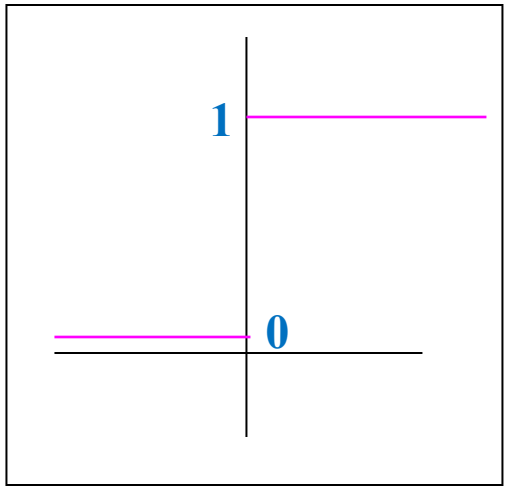
Tanh

$f(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x})$



Logistic

$f(x) = e^{x} / (1 + e^{x})$

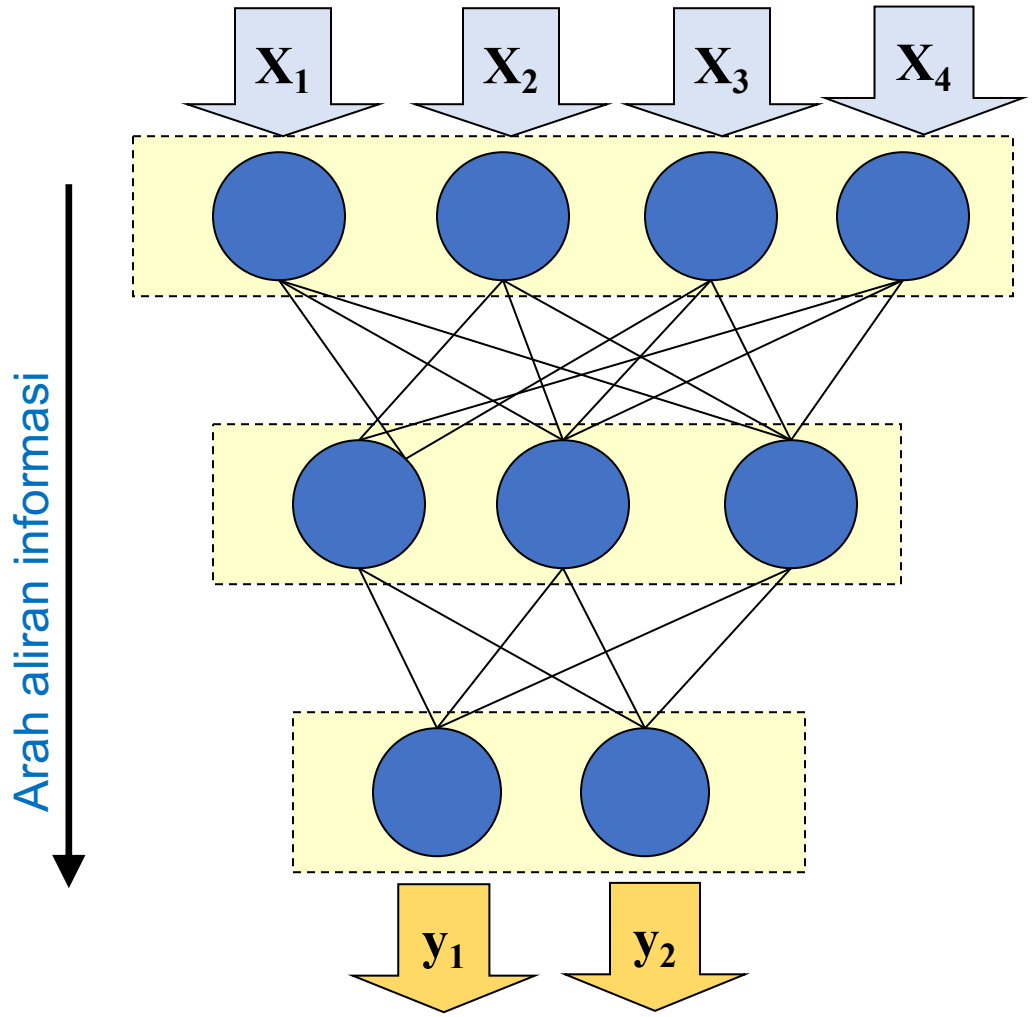


Threshold

$$f(x) = \begin{cases} 0; \text{ if } x < 0 \\ 1; \text{ if } x >= 1 \end{cases}$$

ANN – Feed-forward Network

Sekelompok neuron pada level yang sama membentuk ‘Layer’



Input Layer
- Tiap neuron mendapat HANYA satu input, langsung dari luar

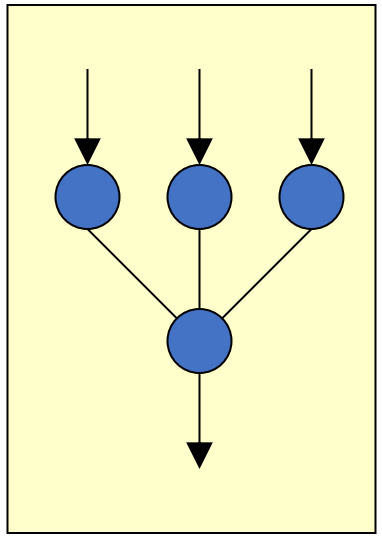
Hidden Layer
- Menghubungkan layer input dan output

Output Layer
- Keluaran dari pengolahan informasi

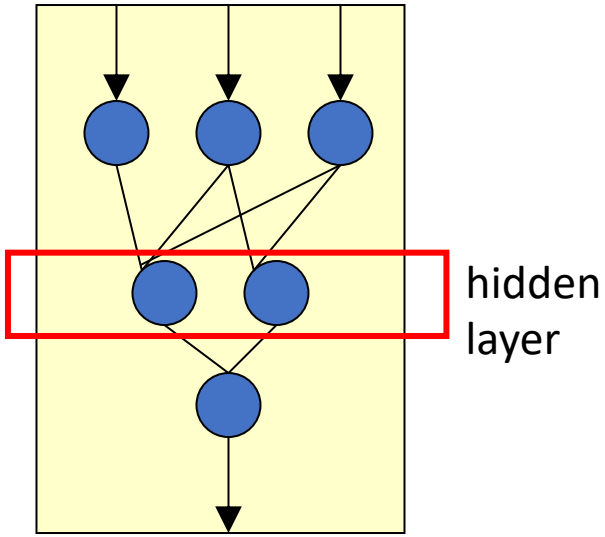
ANN – Feed-forward Network

Banyaknya hidden layer yang mungkin

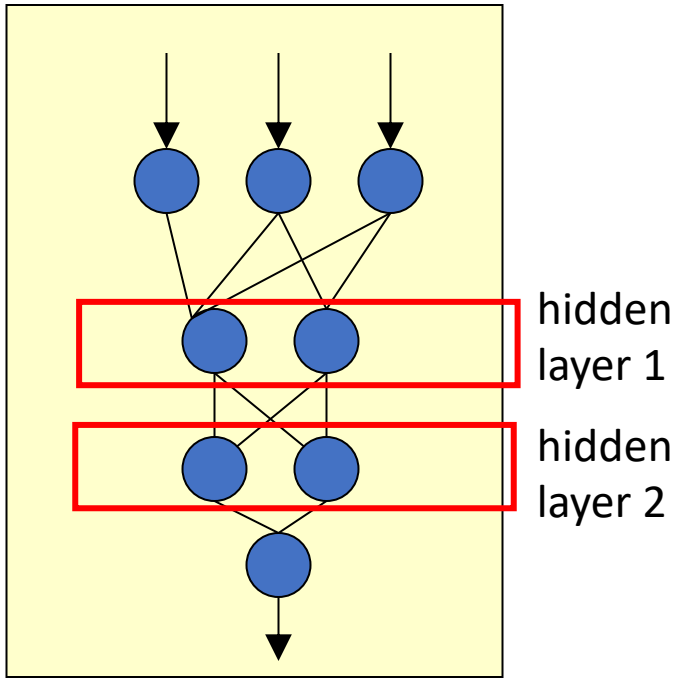
Tidak ada



Satu

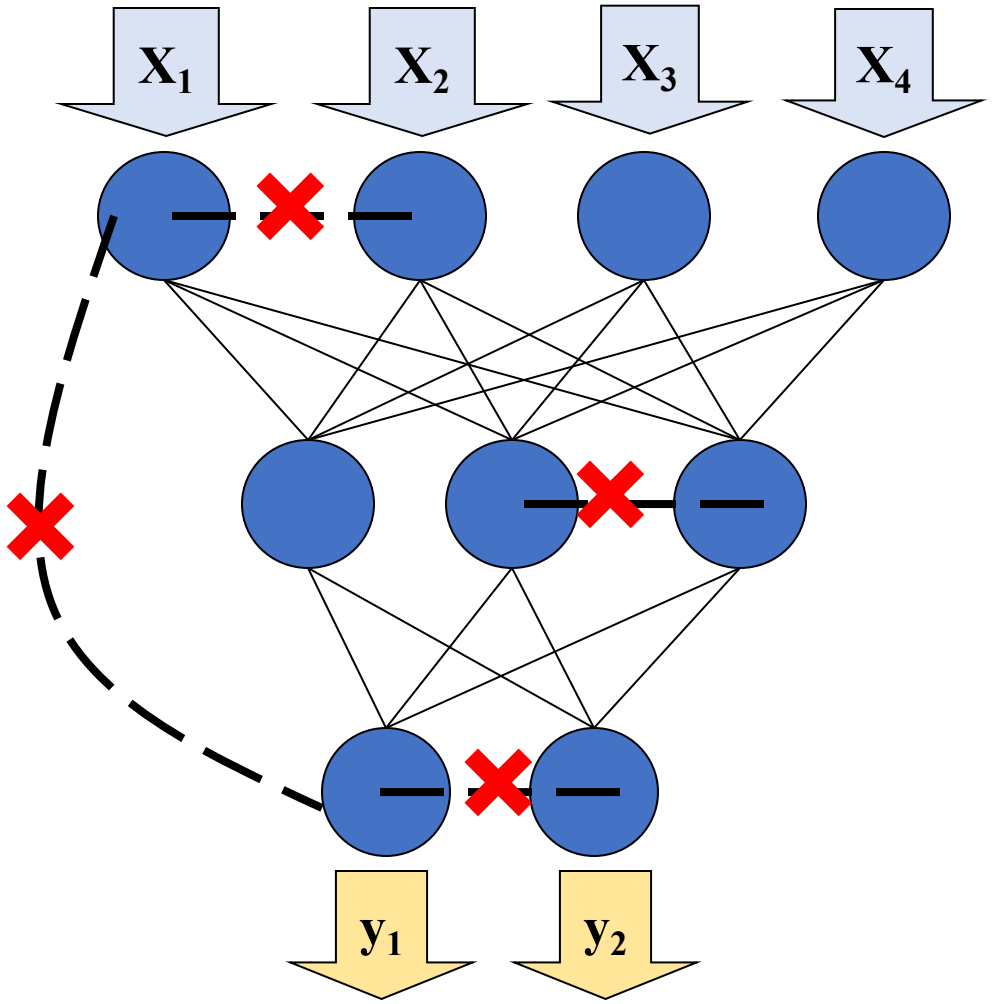


Lebih dari satu



ANN – Feed-forward Network

Beberapa hal yang perlu diperhatikan



- Neuron dalam satu layer TIDAK terhubung satu sama lain.
- Neuron dalam satu layer terhubung HANYA dengan layer BERIKUTNYA. (Feed-forward)
- Lompat layer TIDAK diperbolehkan

Model ANN

Komponen di dalam pemodelan

Input: X_1 X_2 X_3 Output: Y Model: $Y = f(X_1 X_2 X_3)$

Tidak seperti model regresi, persamaan matematika untuk ANN terlalu kompleks untuk dituliskan

Namun, ANN dicirikan dengan

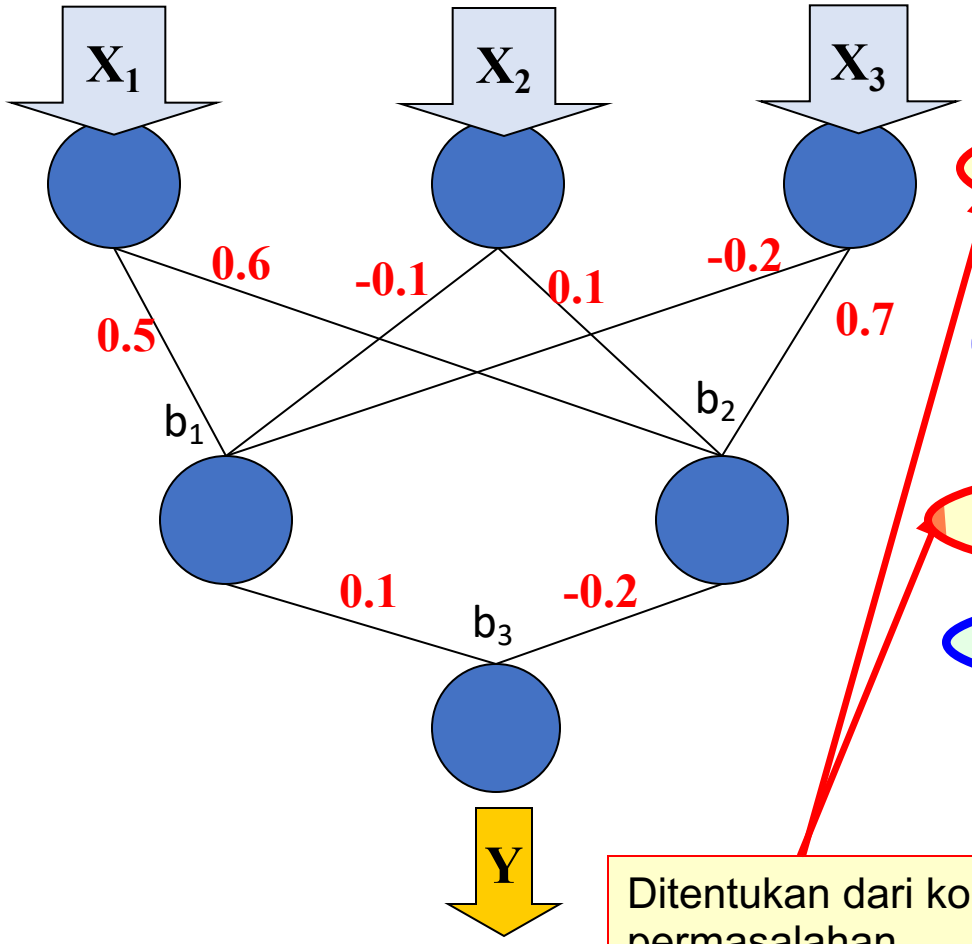
- # Neuron input
- # Hidden Layer
- # Neuron dalam setiap Hidden Layer
- # Neuron output
- BOBOT dan BIAS untuk semua koneksi

'Fitting' model ANN = menentukan nilai untuk semua parameter tersebut

Membangun model pada Neural Networks berarti menentukan parameter bobot dan bias yang optimal

Ilustrasi Model ANN

Input: X_1 X_2 X_3 **Output:** Y **Model:** $Y = f(X_1\ X_2\ X_3)$



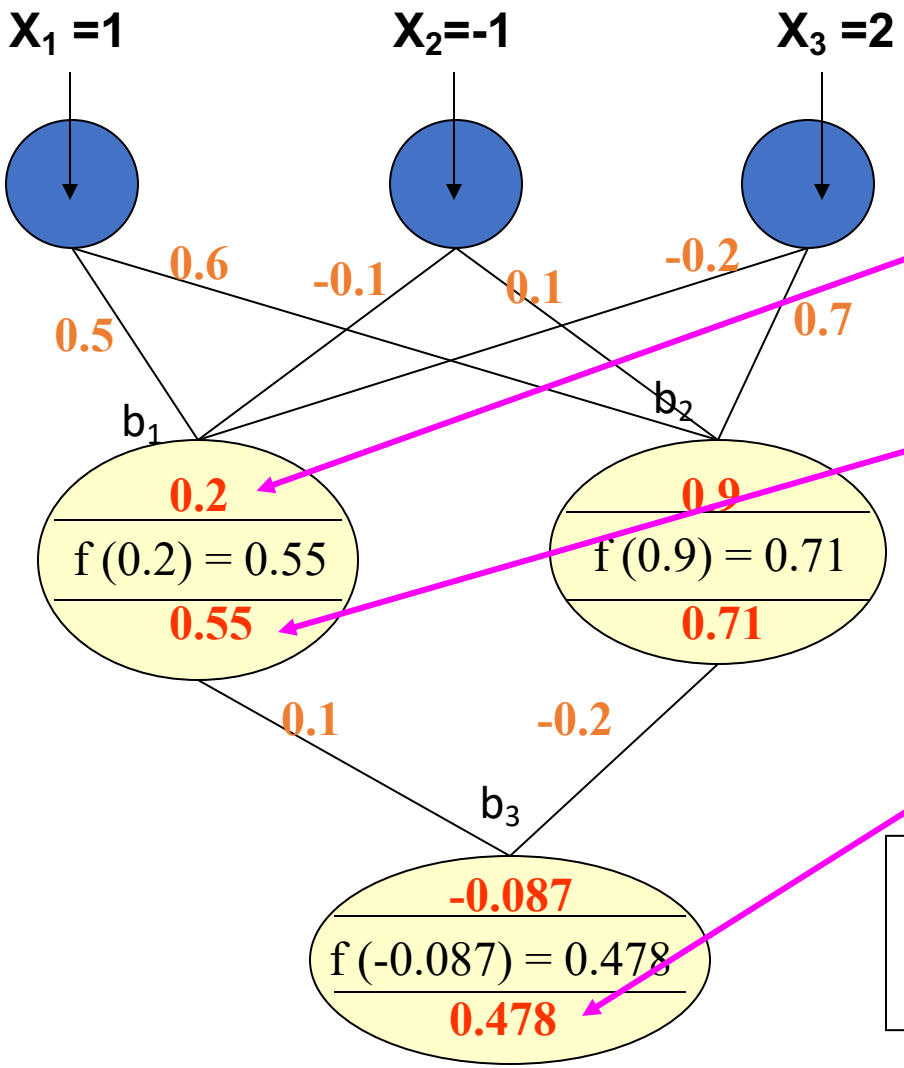
Parameter	Ilustrasi
# Neuron Input	3
# Hidden Layers	1
# Ukuran Hidden Layer	2
# Neuron Output	1
Bobot & bias	Estimasi

Ditentukan dari konteks permasalahan
Input Nrns = # of X's
Output Nrns = # of Y's

Free parameters

Prediksi menggunakan Model ANN

Input: X_1 X_2 X_3 **Output:** Y **Model:** $Y = f(X_1\ X_2\ X_3)$



Misalkan $b_1 = b_2 = b_3 = 0$

$0.2 = 0.5 * 1 - 0.1*(-1) - 0.2 * 2$

$f(x) = e^x / (1 + e^x)$

$f(0.2) = e^{0.2} / (1 + e^{0.2}) = 0.55$

Prediksi Y = 0.478

Misalkan Y Aktual = 2

maka

Prediction Error = (2-0.478) =1.522

Membangun Model ANN

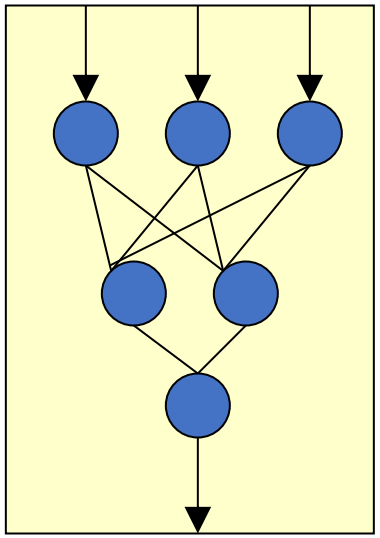
Bagaimana membangun model ANN?

Input: X_1 X_2 X_3 **Output:** Y **Model:** $Y = f(X_1\ X_2\ X_3)$

Neuron Input = # Inputs = **3**
Hidden Layer = ???
Neuron dalam Hidden Layer = ???

Neuron Output = # Outputs = **1**
Coba **1**
Coba **2**

Tidak ada strategi baku, dilakukan secara trial and error



Arsitektur model telah terdefinisi … Bagaimana bobotnya dan biasnya???

Ilustrasi di samping, terdapat 8 bobot.
$\underline{\mathbf{W}} = (W_1, W_2, \ldots, W_8)$

Training Data: $(Y_i, X_{1i}, X_{2i}, X_{3i})$ $i = 1,2,\ldots,n$
Untuk bobot $\underline{\mathbf{W}}$ tertentu, diperoleh prediksi Y

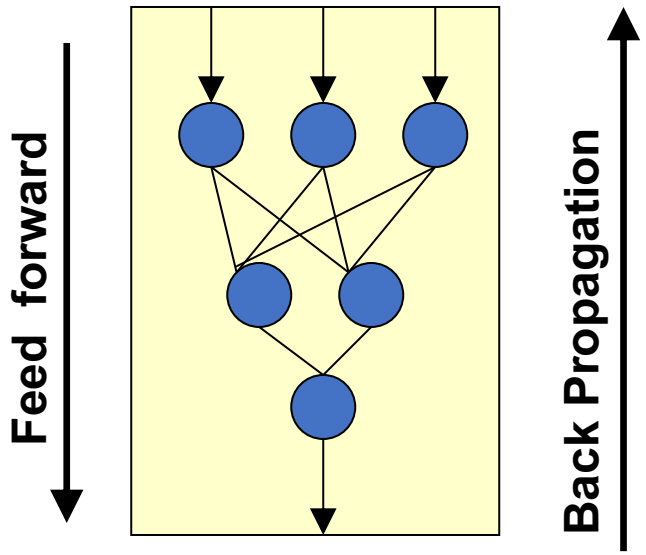
$$(V_1, V_2, \ldots, V_n)$$

Prediksi ini adalah *fungsi* dari $\underline{\mathbf{W}}$.
Bobot $\underline{\mathbf{W}}$ dipilih sedemikian rupa sehingga total prediction error ***E*** minimum

$$E = \Sigma\ (Y_i - V_i)^2$$

Training the Model

How to train the Model ?

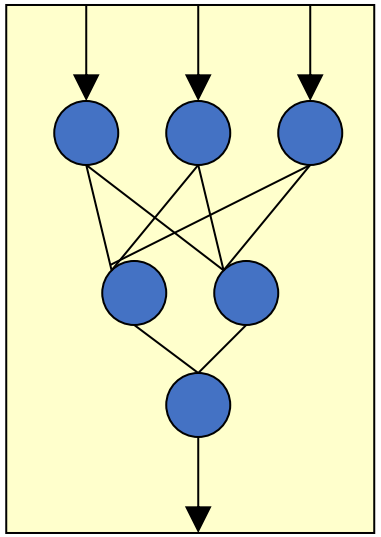


$$E = \Sigma (Y_i - V_i)^2$$

- Mulai dengan bobot awal.
- Feed forward pengamatan pertama melalui NN
  $X_1 \rightarrow$ Network $\rightarrow V_1$; Error $= (Y_1 - V_1)$
- Sesuaikan bobot sehingga error ini mengecil
  (network mengepas dengan baik pengamatan pertama)
- Feed forward pengamatan kedua.
  Sesuaikan bobot agar NN mengepas dengan baik pengamatan kedua
- Ulang terus proses ini hingga pengamatan terakhir
- Sampai di sini, training PUTARAN pertama selesai
- Lakukan beberapa putaran hingga total prediction error ***E*** kecil.

Back Propagation

Lebih jauh mengenai Back Propagation



$$E = \Sigma (Yi - Vi)^2$$

Setiap bobot ikut menyumbang kesalahan (Shares the Blame) pada prediction error bersama-sama dengan bobot lainnya.

Algoritma Back Propagation akan memutuskan bagaimana mendistribusikan kesalahan ini pada semua bobot dan sekaligus menyesuaikan nilainya.

Porsi kesalahan yang kecil berujung pada penyesuaian kecil pada bobot.

Porsi kesalahan yang besar membawa efek pada penyesuaian nilai bobot yang juga besar.

Penyesuaian bobot pada Back Propagation

Formula penyesuaian bobot pada Back Propagation

V_i – prediksi untuk pengamatan ke-i – adalah fungsi dari vektor bobot $\underline{W} = (W_1, W_2, \ldots)$
Sehingga, E, total prediction error juga fungsi dari $\underline{W}$

$$E(\underline{W}) = \Sigma [Y_i - V_i(\underline{W})]^2$$

Gradient Descent Method :

Untuk setiap bobot W_j, formula penyesuaian nilai bobot adalah

$$W_{new} = W_{old} + \alpha * (\partial E / \partial W) |_{W\ old}$$

α = Learning Parameter (antara 0 dan 1)

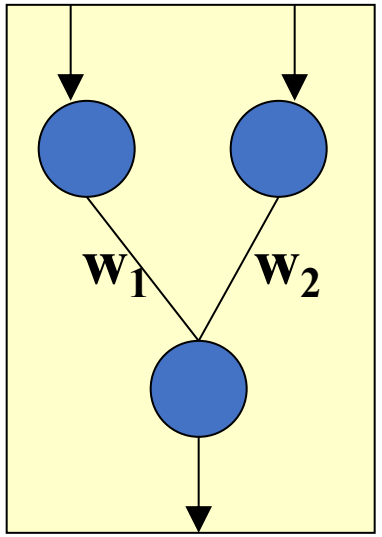
Tipe lain dari formula penyesuian ini adalah

$$W_{(t+1)} = W_{(t)} + \alpha * (\partial E / \partial W) |_{W(t)} + \beta * (W_{(t)} - W_{(t-1)})$$

β = Momentum (antara 0 dan 1)

Interpretasi geometrik dari penyesuaian bobot

Perhatikan NN dengan 2 input dan 1 output, tanpa hidden layer. NN ini memerlukan dua bobot yang harus ditentukan nilainya.



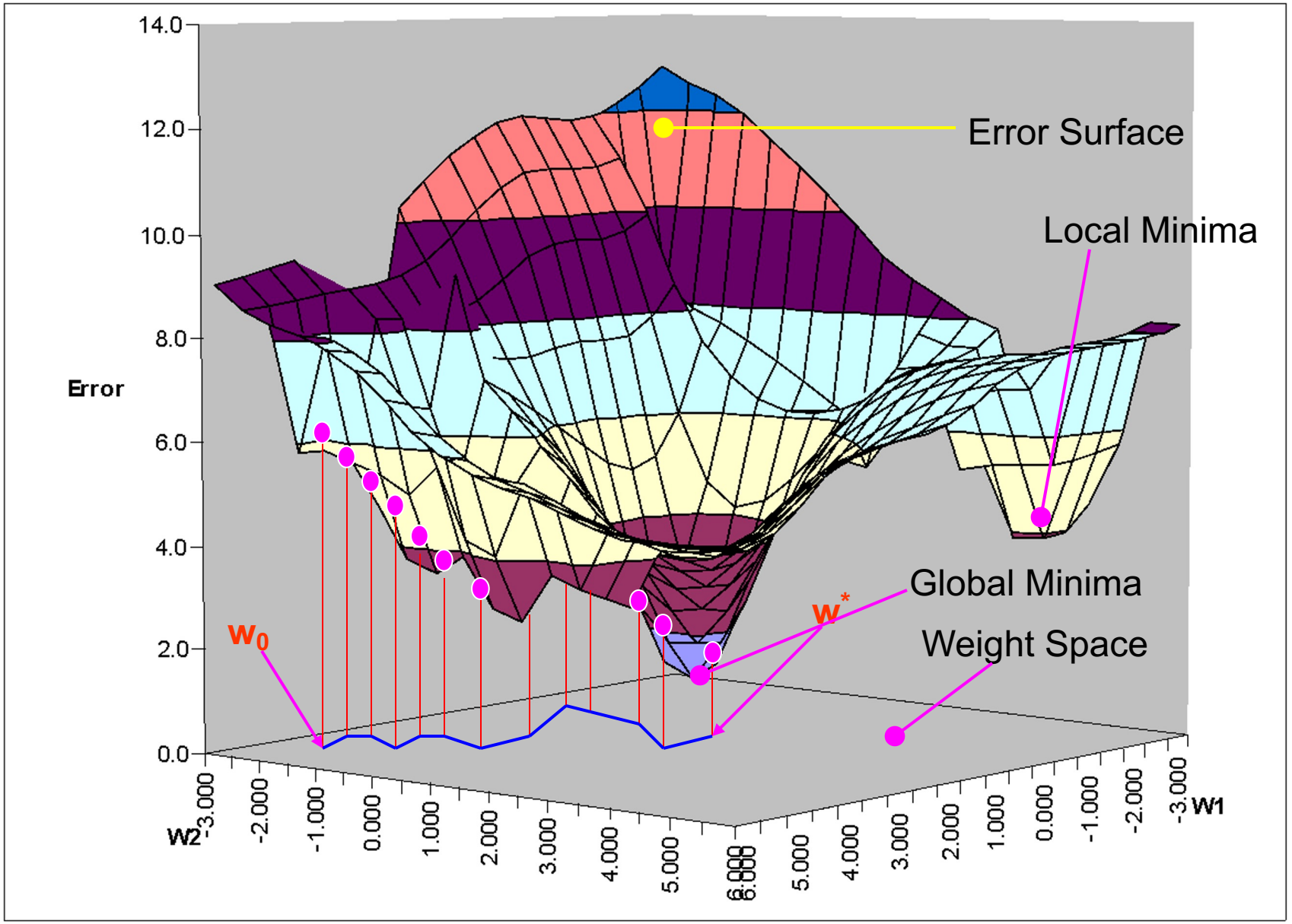
$$E(w_1, w_2) = \Sigma [Yi - Vi(w_1, w_2)]^2$$

- Pasangan (w_1, w_2) adalah sebuah titik pada bidang 2-D.
- Untuk setiap titik dapat dihitung nilai E.
- Buat Plot E vs (w_1, w_2) pada ruang 3-D - '**Error Surface**'
- Tujuan kita adalah menentukan pasangan bobot dengan nilai E minimum
- Ini berarti kita berupaya mencari pasangan bobot dengan tinggi error surface minimum.

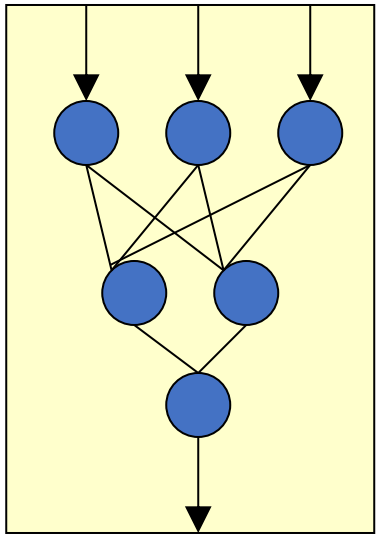
Gradient Descent Algorithm

- Mulai dengan sembarang titik (w_1, w_2)
- Pindah pada titik baru (w'_1, w'_2) dengan tinggi error surface lebih rendah.
- Terus cari titik baru hingga (w^*_1, w^*_2), di mana error minimum.

Crawling the Error Surface



Training Algorithm



$$E = \Sigma (Y_i - V_i)^2$$

Tentukan Network architecture
(# Hidden layer, # Neuron pada setiap Hidden Layer)

Tentukan Learning parameter and Momentum

Initialize NN dengan sembarang bobot

Do till Convergence criterion is met

For I = 1 to # Training Data points

Feed forward pengamatan ke-i
Hitung prediction error pada pengamatan ke-i

Back propagate error dan sesuaikan nilai bobot

Next I

Check for Convergence

End Do

Convergence Criterion

Kapan diputuskan training NN selesai?

Idealnya – ketika tercapai global minima pada error surface

Bagaimana kita tahu titik ini telah tercapai? Kita tidak pernah tahu …

Suggestion:

1. Stop jika penurunan total prediction error (dari putaran terakhir) *kecil*.
2. Stop jika keseluruhan perubahan bobot (dari putaran terakhir) *kecil*.

Drawback:

Error terus menurun. Pada titik ini kita mendapatkan model yang sangat cocok dengan data training.

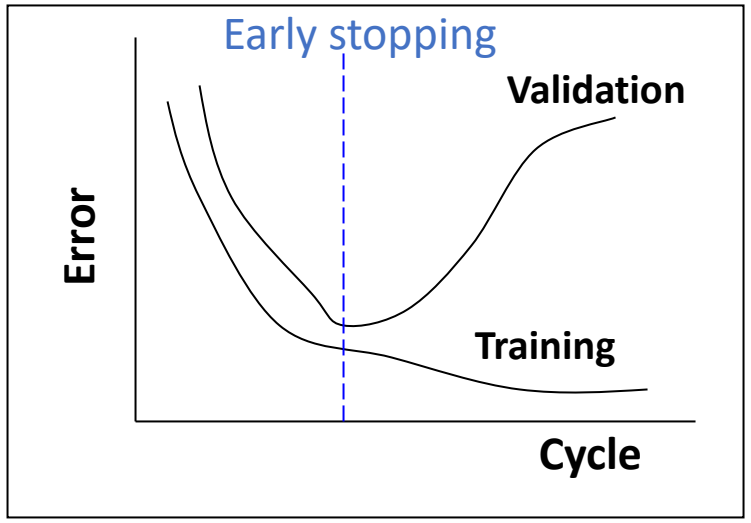
NAMUN … Model NN memiliki performa yang tidak bagus (poor generalizing power) pada data lain (unseen data)

Fenomena ini dinamakan - **Over fitting** pada data training

Overfitting: Good performance on the training data, poor generliazation to other data.
Underfitting: Poor performance on the training data and poor generalization to other data.

Cara mengatasi Overfitting dalam Neural Network:

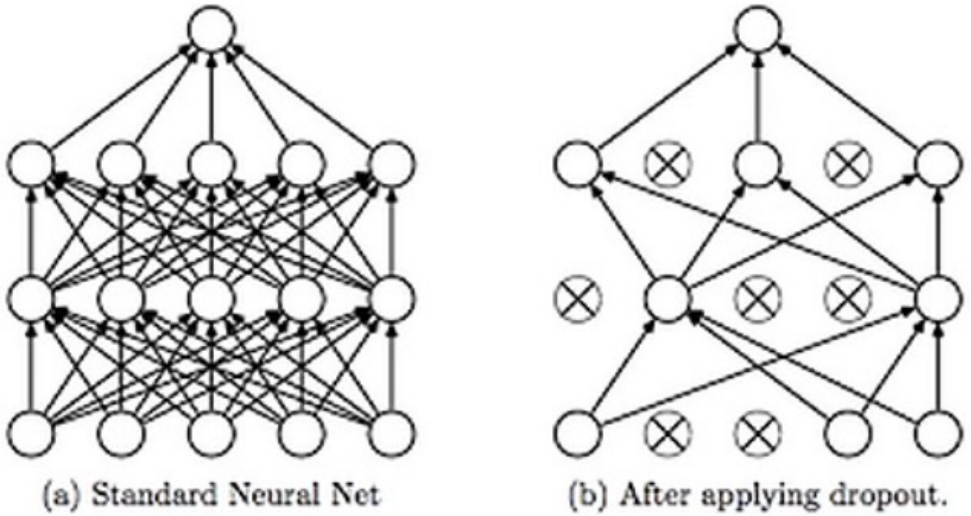
1. Early Stopping



2. Regularization

$$L(x,y) \equiv \sum_{i=1}^{n} \left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{i=1}^{n} |\theta_i|$$

3. Dropout



(a) Standard Neural Net

(b) After applying dropout.

Convergence Criterion

Modified Suggestion:

Partisi data menjadi **Training set** dan **Validation set**

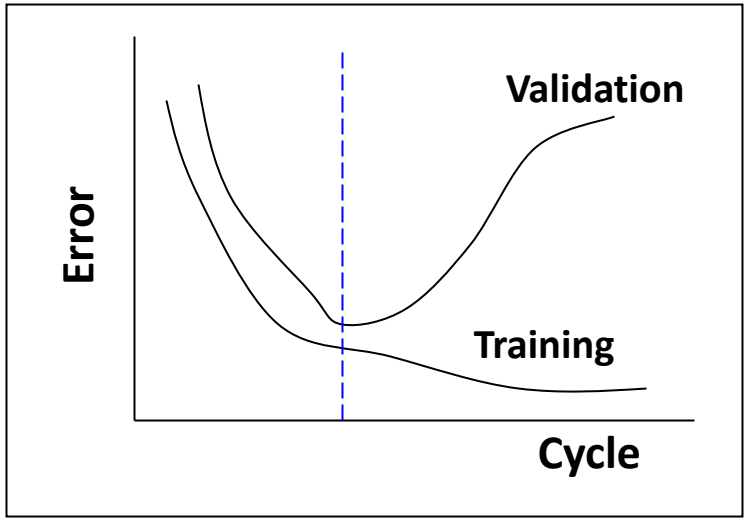
Gunakan Training set – membangun model

Validation set – Uji performa model pada data lain

Biasanya dengan semakin banyaknya putaran training

Error pada Training set terus menurun.

Error pada Validation set awalnya menurun kemudian meningkat.



Stop training ketika error pada Validation set mulai naik

Pemilihan Parameter Training

Learning Parameter dan Momentum

- harus ditentukan oleh pengguna, dengan nilainya antara 0 dan 1

Berapa nilai yang optimal bagi kedua parameter ini?

- Tidak ada strategi baku penentuan nilai optimal.
- Hanya saja, efek dari kesalahan penentuan nilai kedua parameter ini dapat ditelusuri.

Learning Parameter

Terlalu besar – Perubahan titik bobot setiap putaran sangat besar – risiko global minima terlewatkan.

Terlalu kecil –

- Memakan waktu lama untuk konvergen pada global minima
- Sekali terperangkap pada local minima, sulit untuk keluar.

<u>Suggestion</u>

Trial and Error – Cobakan berbagai nilai Learning Parameter dan Momentum dan perhatikan nilai mana yang berujung pada prediction error minimum

Catatan

- Training neural network dilakukan untuk mendapatkan bobot (w_i) optimal dengan menggunakan gradient descent dengan bantuan metode backpropagation untuk proses menghitung gradient.
- Metode gradient descent memungkinkan untuk mendapatkan bobot (w_i) optimal dengan meminimumkan suatu cost function.
- Banyaknya iterasi (ulangan) yang dilakukan untuk meng-update bobot menggunakan seluruh training data disebut **epoch**.
- Dalam satu epoch, training data bisa dibagi menjadi beberapa bagian yang disebut **batch**.
- Kecepatan konvergensi proses train bisa diatur dengan learning rate. Learning Rate nilainya 0,1.
- Semakin kecil learning rate, **epoch** yang dibutuhkan akan semakin banyak.
- **Batch size** merupakan hyperparameter untuk mengontrol banyaknya amatan yang masuk dalam 1 batch

Wrap Up

- Artificial Neural network (ANN)
  – Pemodelan yang terinspirasi dari prinsip kerja jaringan syaraf

- Digunakan untuk berbagai tujuan pemodelan – Prediction, Classification, Clustering, ..

- Salah satu metode dalam ANN – Feed forward Back propagation networks

  - Dibentuk dari beberapa layer – Input, hidden, Output
  - Tiap layer adalah sekumpulan artificial Neurons
  - Neuron dalam satu layer terhubung dengan neuron pada layer berikutnya
  - Hubungan antar neuron memiliki bobot tertentu

- Fitting model ANN pada dasarnya adalah mencari nilai optimal bobot ini.

- Berdasarkan data training set – bobot diperoleh dengan algoritma Feed forward Back propagation, yang menerapkan metode Gradient Descent Method – yaitu suatu teknik yang populer digunakan untuk meminimumkan suatu fungsi.

- Arsitektur Network maupun parameter training ditentukan secara trial and error. Cobakan berbagai nilai yang mungkin dan pilih yang menghasilkan prediction error minimum.

Ilustrasi 1

A Neural Network Playground – TensorFlow

https://playground.tensorflow.org

Ilustrasi 2

Gunakan data iris di R

1. Lakukan ANN untuk Binary classification untuk Species = "setosa"

2. Lakukan ANN untuk Multiclass classification untuk Species

3. Lakukan ANN untuk Binary classification untuk Species = “setosa” dengan fungsi aktivasi:

$$\ln(1 + e^x)$$

```
library(neuralnet)

# Binary classification
nn <- neuralnet(Species == "setosa" ~ Petal.Length + Petal.Width, iris,
linear.output = FALSE)

print(nn)
plot(nn)

# Multiclass classification
nn2 <- neuralnet(Species ~ Petal.Length + Petal.Width, iris, linear.output = FALSE)
names(nn2)
nn2$act.fct

print(nn2)
plot(nn2)

# Custom activation function
softplus <- function(x) log(1 + exp(x))
nn3 <- neuralnet((Species == "setosa") ~ Petal.Length + Petal.Width, iris,
linear.output = FALSE, hidden = c(3, 2), act.fct = softplus)
print(nn3)
plot(nn3)
```

Tugas individu

- Suatu perusahaan perbank-kan meneliti 75 jenis skema pinjaman yang telah diberi rating oleh para customernya pada **data ann.csv** (data terlampir di newlms)
- Variabel yang digunakan ialah:
  - Besar pinjaman (dalam juta rupiah)
  - Lama pembayaran (dalam tahun)
  - Tambahan bunga yang ditetapkan (dalam %)
  - Pembayaran per bulan (dalam 10000)
  - Banyak cash back yang diterapkan pada skema tersebut
- Tujuan penelitian yang dilakukan ialah memprediksi rating skema pinjaman berdasarkan variabel-variabel tersebut

- Bantulah peneliti tersebut untuk memprediksi reating skema pinjaman dengan neural network, hitunglah RMSE prediksinya

Ketentuan Tugas:

- Kirimkan kode R beserta output dan interpretasinya pada file dengan format pdf

- Submit tugas individu pada newlms STA1382 – Teknik Pembelajaran Mesin, pada topik Artificial Neural Network, Tugas Individu 1

- Batas waktu pengiriman adalah **Hari Senin, tanggal 26 Februari 2024 jam 13:00 WIB**

- Komponen Penilaian:
  - Kecepatan pengiriman
  - Kesesuaian jawaban
  - Orisinalitas

Terima kasih ☺