

Artificial Neural Networks(ANNs)

ระบบโครงข่ายประสาทประดิษฐ์

โดย ดร.วรารุณ วุฒิวิณิชย์

2545

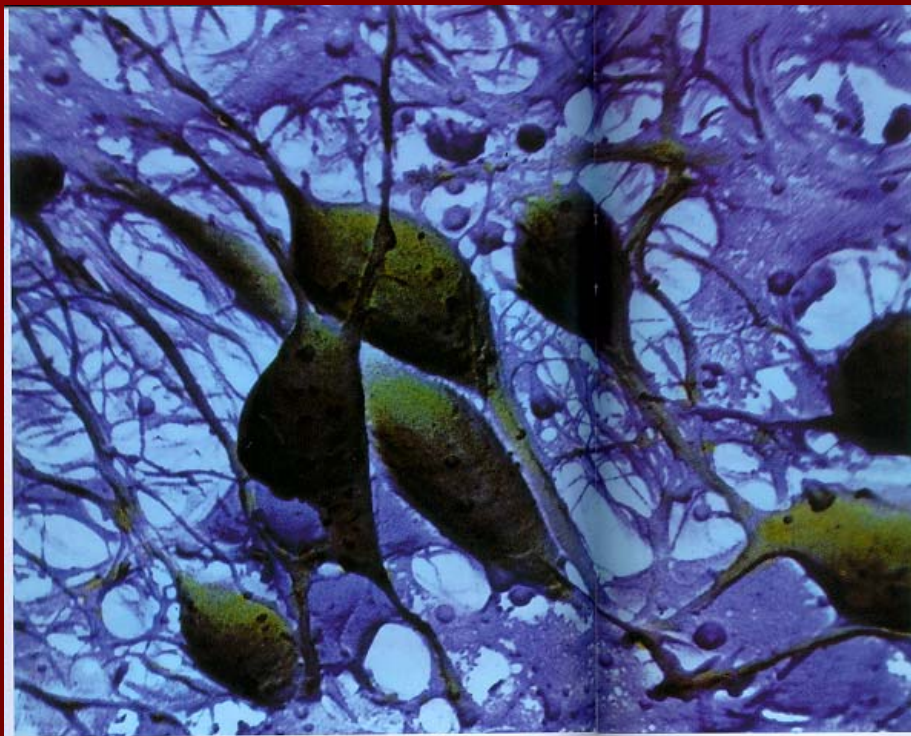
Artificial Neural Networks คืออะไร

- **ANNs** นับเป็นสาขาหนึ่งของปัญญาประดิษฐ์ (**Artificial Intelligence, AI**)
- **ANNs** คือระบบการคำนวณที่สร้างเลียนแบบการทำงานของระบบสมองมนุษย์ เพื่อใช้ประโยชน์ในการคาดคะเนเหตุการณ์จากข้อมูลที่มีอยู่เช่น การพยากรณ์อากาศ การพยากรณ์หุ้นในตลาดหลักทรัพย์
- ในปัจจุบันเริ่มมีการนำเอา **ANNs** มาใช้ในการพยากรณ์ทางด้านอุทกวิทยา (**Mason et al. 1996, Fernando and Jayawardena 1988**)
- นอกจากนี้ได้มีการพัฒนา **ANNs** ไปใช้ประโยชน์ในสาขาอื่น ๆ มากมาย เช่น **Image Processing** และ **Pattern Recognition**

ANNs มีชื่อเรียกอื่นอีกเช่น

- Natural Intelligent System
- Machine Learning Algorithms

เซลล์ประสาทในสมองคน



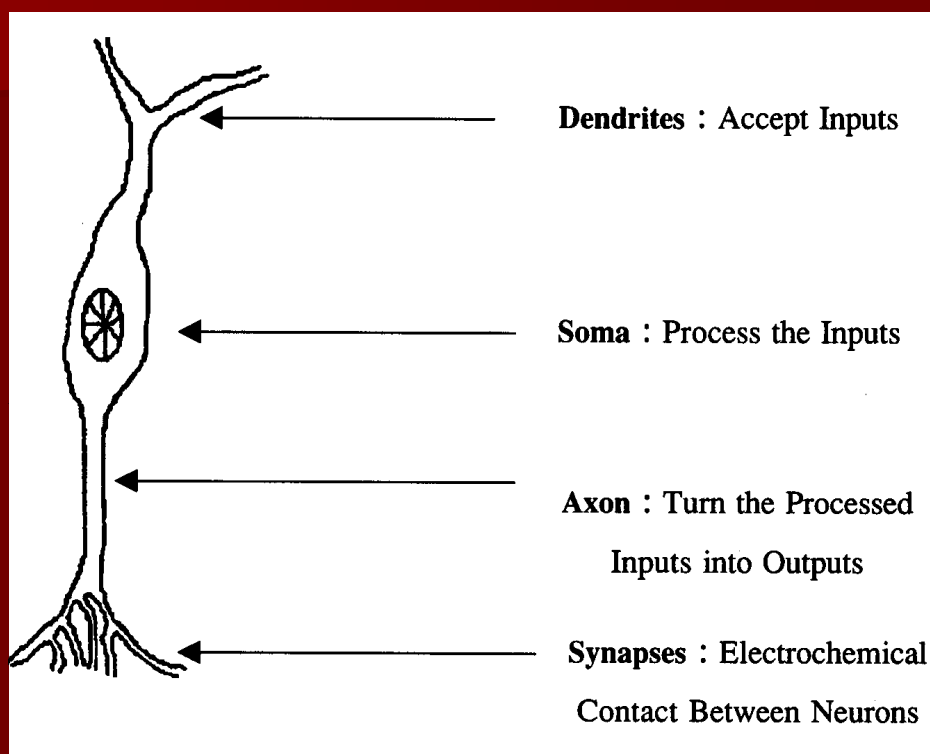
มนุษย์จำได้อย่างไร

ในเมืองย่างกุ้ง ประเทศพม่า ปี ค.ศ.
1974 ชายชื่อกันทันตะ วิจิตตสาระ
สามารถท่องจำข้อความในคัมภีร์พุทธ
ศาสนาจำนวน 16,000 หน้าได้ นับว่า
เป็นปรากฏการณ์เหนือธรรมดา คนเรา
แทบทุกคนสามารถจดจำข้อมูลจำนวน
มากได้อย่างน่าอัศจรรย์ใจ

ในช่วงการเรียนรู้ เด็กที่มีอายุถึง
6 ขวบจะรู้จักใช้ศัพท์ประมาณ
6,000 คำ อีกชั่วชีวิตต่อจากนั้นคน
ทั่วไปจะเรียนรู้ศัพท์เพิ่มขึ้นอีก
ราว 14,000 คำเท่านั้น

ข้อมูลข่าวสารที่เก็บไว้ในความทรงจำ
ระยะยาวอาจถูกแปลงเป็นภาพอะไรสัก
อย่าง แล้วเก็บไว้ในเซลล์สมองส่วน
นอก เซลล์ประสาทนี้มีจำนวนกว่า
100,000 ล้านเซลล์ แต่ละเซลล์เชื่อมกับ
เซลล์อื่น ๆ อีก 10,000 จุด เกิดเป็น
เครือข่ายโยงใยที่ซับซ้อนอย่างไม่น่าเชื่อ

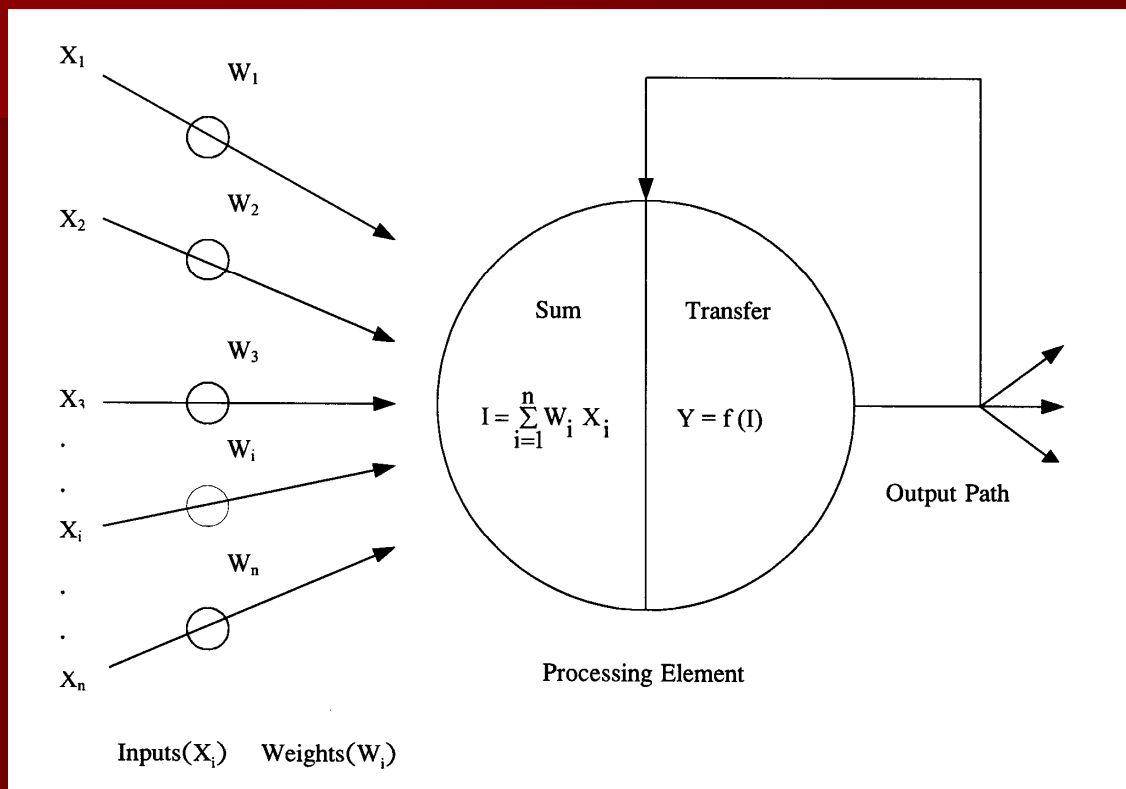
โครงสร้างของ **Neuron** ในสมองมนุษย์



Neurons ในสมองมนุษย์

- **Dendrites** คือ ส่วนที่ทำหน้าที่รับข้อมูล (Accept Inputs)
- **Soma** คือ ส่วนที่ประมวลผลข้อมูลเบื้องต้น (Process inputs)
- **Axon** คือ ส่วนที่แปลงข้อมูลที่ได้ประมวลเบื้องต้นเป็นผลลัพธ์ที่ต้องการ
(Turn the processed input into outputs)
- **Synapses** คือ เส้นประสาทที่ทำหน้าที่เชื่อมต่อเพื่อการสื่อสารกับ
Neurons อื่นในระบบสมอง
(Electrochemical contact between neurons)

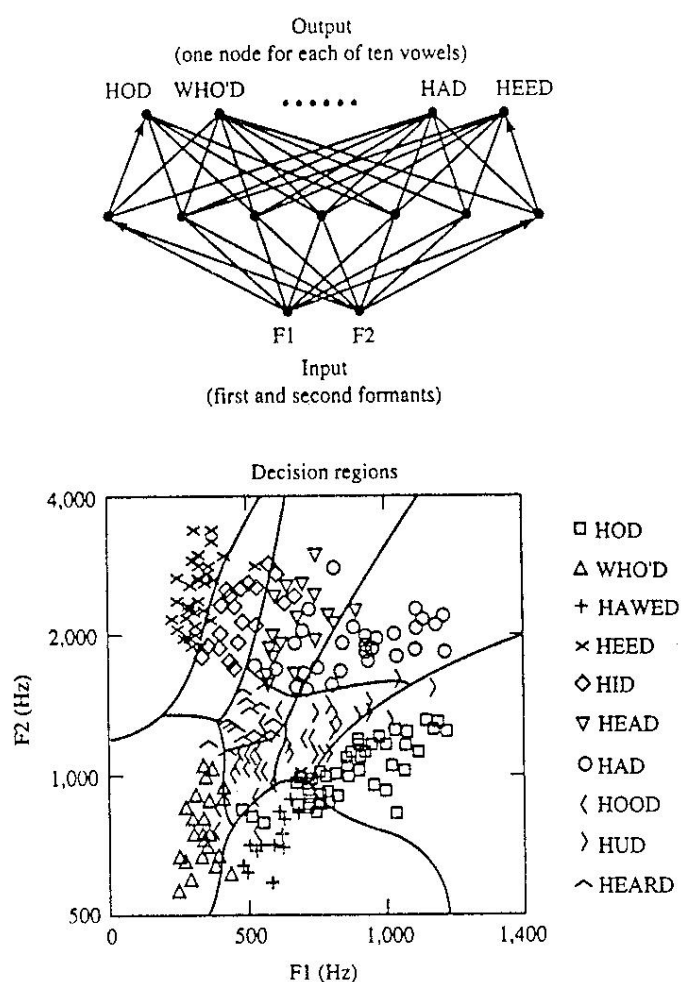
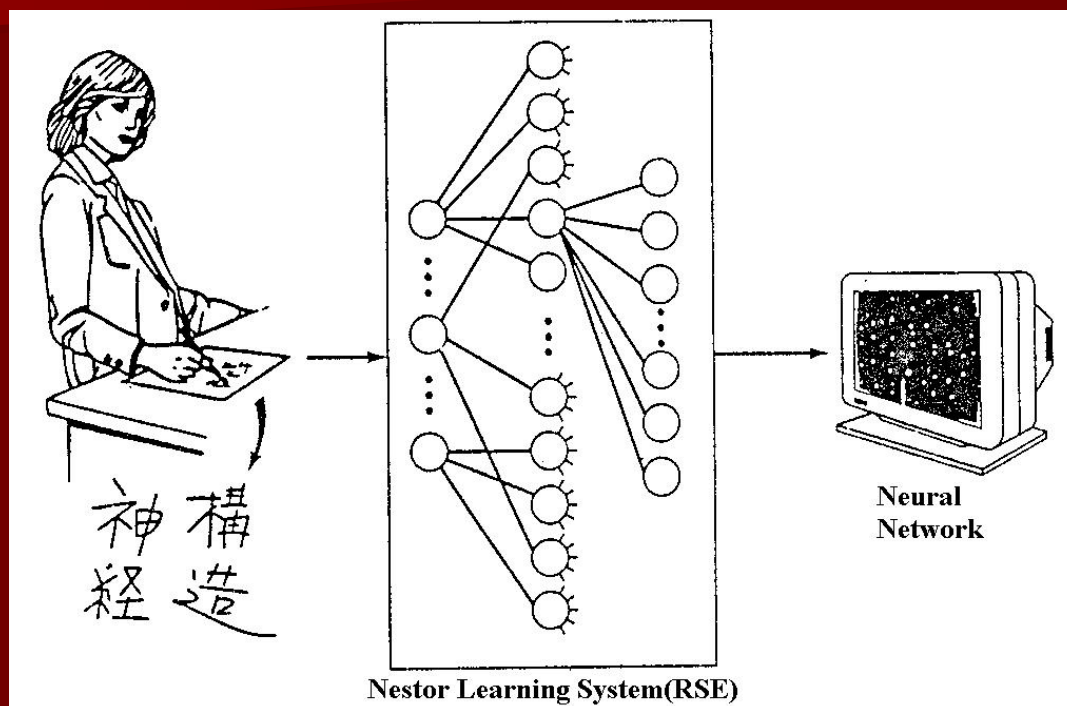
โครงสร้างของ Artificial Neuron ใน ANNs



Kenji Character Recognizer

translate Japanese handwritten to English

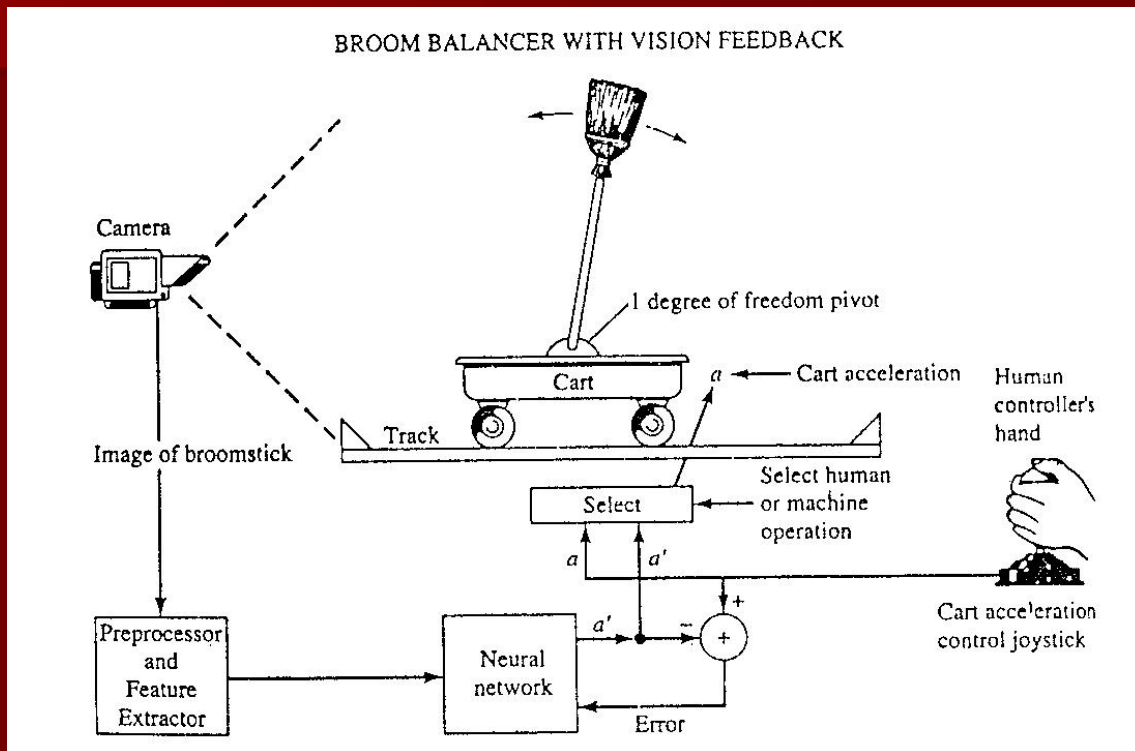
2500 characters with 92 % reliability



Neural
network
to
recognize
vowel
sounds

Richard
Lippmann of
MIT

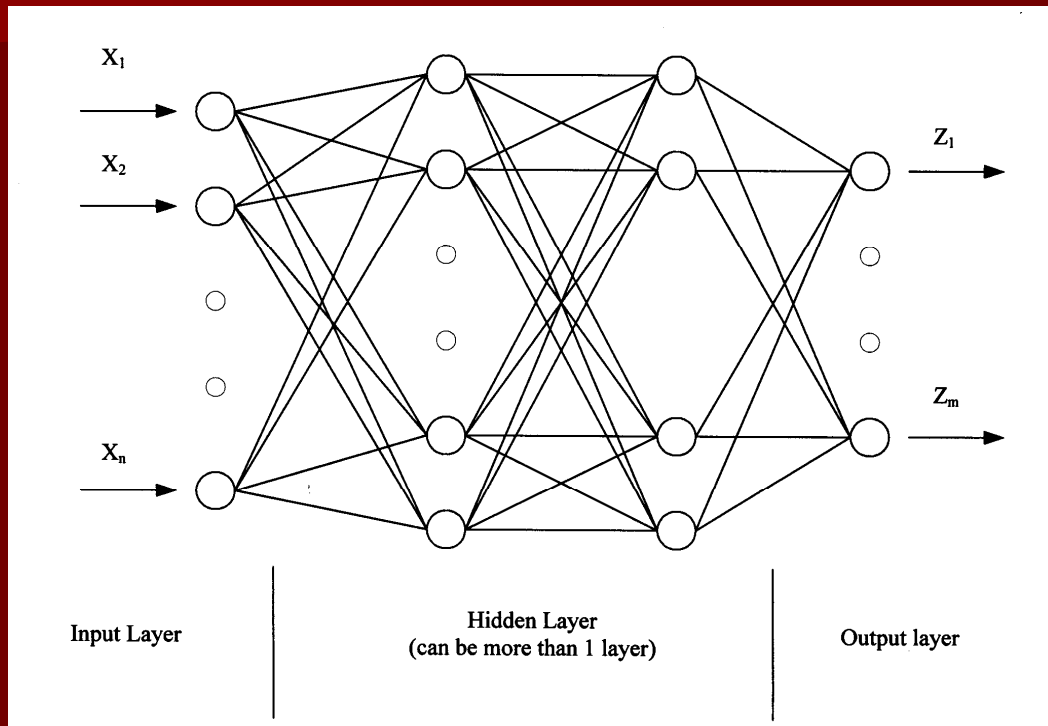
Broom Balancer with Vision Feedback



การออกแบบ Artificial Neural Networks

- กำหนด Neurons ในแต่ละ Layer โดยกำหนดให้ Layer แรกคือ Input Layer และ Layer สุดท้ายคือ Output Layer ส่วน Layer ที่อยู่ตรงกลางเรียกว่า Hidden Layer อาจมี 1 หรือมากกว่า 1 Layer ก็ได้ จำนวน Neurons ใน Input Layer จะเท่ากับจำนวน Inputs (X) และจำนวน Neurons ใน Output Layer จะเท่ากับจำนวน Outputs (Z) Neurons ใน Hidden Layers คือ Neurons ที่ทำหน้าที่ในการประมวลผลข้อมูลเพื่อแปลง Inputs เป็น Outputs
- กำหนดการเชื่อมโยง (Connection) ระหว่าง Neurons ซึ่งอยู่ต่าง Layers และ Neurons ใน Layer เดียวกัน
- สอน Network ให้เรียนรู้ (Learn) ความสัมพันธ์ระหว่างข้อมูลและผลลัพธ์ที่ต้องการ เพื่อสร้างค่า Connection Weight (W_i) ที่เหมาะสม โดยใช้ชุดข้อมูลที่เรียกว่า Training Data Set

รูปแบบการเชื่อมโยงของ Artificial Neurons ใน ANNs



รูปแบบของการเชื่อมโยงระหว่าง Neuron

■ Fully Connected

- Neurons แต่ละตัวใน Layer ที่ 1 จะเชื่อมต่อกับ Neurons ทุกตัวใน Layer ที่ 2

■ Partially Connected

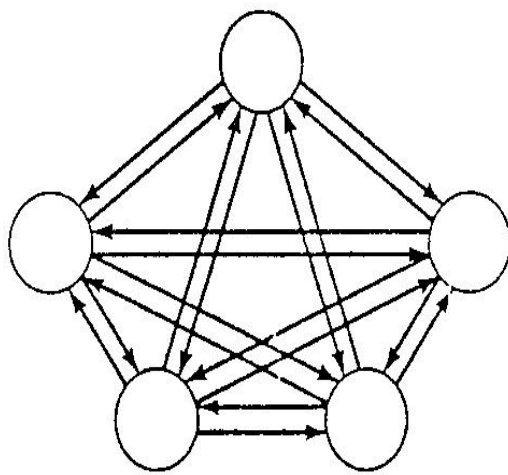
- Neurons แต่ละตัวใน Layer ที่ 1 เชื่อมต่อกับ Neurons บางตัวใน Layer ที่ 2 เท่านั้น

■ Feed Forward

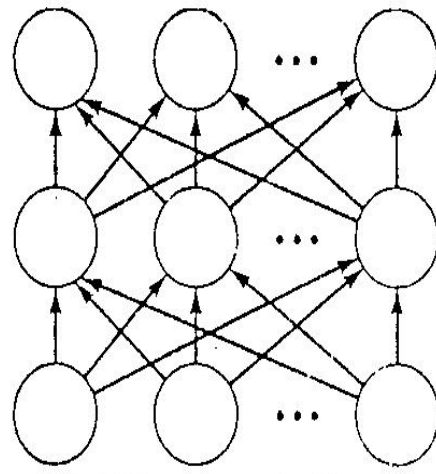
- Neurons ใน Layer ที่ 1 ส่ง Output ให้ Neurons ใน Layer ที่ 2 แต่จะไม่ได้รับ Feedback จาก Neurons ใน Layer ที่ 2

■ Bi-directional

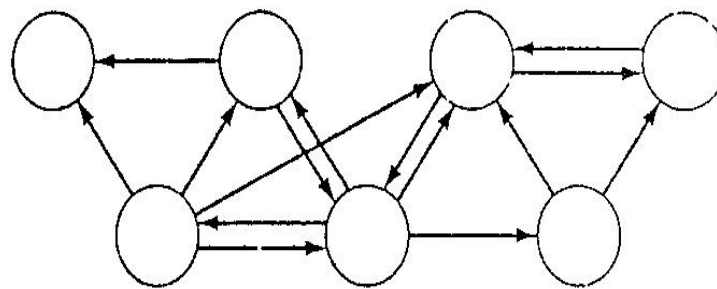
- Output จาก Neurons ใน Layer ที่ 2 จะถูกส่งกลับมาเป็น Input ของ Neurons ใน Layer ที่ 1



(a) A totally connected network



(b) A 3 layer network with total connection between layers



(c) A network of mixed connectivity

กระบวนการเรียนรู้ (Learning) ของ Artificial Neural Networks

Unsupervised Vs. Supervised

Unsupervised Learning

- ในระบบการเรียนรู้แบบ Unsupervised Hidden Neurons จะปรับตัวเองโดยไม่ต้องการความช่วยเหลือจากภายนอก ไม่ต้องมีตัวอย่างผลลัพธ์เพื่อให้ Neurons ได้เรียนรู้ กระบวนการเรียนรู้แบบนี้เรียกว่า Learning by Doing

Supervised Learning

- เป็นระบบการเรียนรู้ซึ่งต้องการครูช่วยสอน ซึ่งครูในความหมายนี้คือ ชุดข้อมูลฝึกอบรม (Training Data Set) Neurons ใน Hidden Layer จะปรับค่า Weight แบบสุ่ม (Random) หรือหมุนเวียน (Rotation) ตามผลของการคำนวณผลลัพธ์ ระบบการเรียนรู้แบบนี้อาจเรียกได้อีกอย่างหนึ่งว่า Reinforcement Learning
- ระบบการเรียนรู้แบบ Supervised ซึ่งนิยมใช้สำหรับกรณีที่มีหลาย Layer และมีระบบการเชื่อมโยงแบบ Feed Forward คือ Back Propagation ซึ่งถือเป็นระบบการเรียนรู้ที่มีระบบการส่งข้อมูลเกี่ยวกับความคลาดเคลื่อน ในการคำนวณผลลัพธ์กลับไปปรับค่ากำลังในการเชื่อมโยงระหว่าง Layer เพื่อเพิ่มประสิทธิภาพในการคำนวณของระบบ

Perceptron

Perceptron = A Learning machine

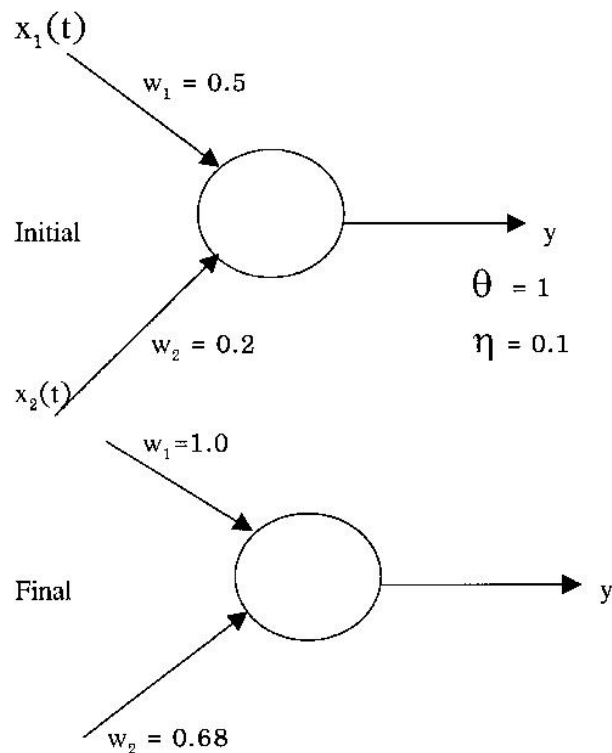
(A Single node having 2 inputs)

Bias	Learning Rate	Synaptic weights		Inputs		Argument	Sgn Function	Target	Corection Factor	Corrections	
θ	η	w_1	w_2	x_1	x_2	α	y	z	$\eta(z-y)$	Δw_1	Δw_2
1	.1	.5	.2	0	0	-1	-1	-1	0	0	0
-	-	-	-	1	1	-.3	-1	+1	.2	.2	.2
-	-	.7	.4	1	1	.1	+1	+1	0	0	0
-	-	-	-	.8	.7	-.16	-1	+1	.2	.16	.14
-	-	.86	.54	.3	.3	-.58	-1	-1	0	0	0
-	-	-	-	.7	.7	-.02	-1	+1	.2	.14	.14
-	-	1.0	.68	.7	.7	.176	+1	+1	0	0	0
-	-	-	-	.3	.3	-.496	-1	-1	0	0	0

$$y(t) = \text{sgn} \underbrace{\sum [x_i(t) w_i(t)] - \theta}_{\alpha} \leftarrow \text{Activation}$$

sgn function = +1 if argument > 0
= -1 if argument ≤ 0

$$w_i(t+1) = w_i(t) + \underbrace{\eta[z(t) - y(t)] x_i(t)}_{\Delta w_i} \leftarrow \text{Delta Rule}$$



$$\alpha = (0.5 * 0 + 0.2 * 0) - 1 = -1$$

$$y(1) = \text{sgn}(-1) = -1$$

$$\Delta W_i = 0.1(-1+1) * X_i(1) = 0$$

The structure and behavior of a perceptron are detailed here, with the perceptron as a single node having two inputs. It is trained to distinguish a pair of high inputs from a pair of low inputs. However, it would not be able to distinguish mixed pairs

$$u = (0.5 * 1 + 0.2 * 1) - 1 = -0.3$$

$$y(2) = \text{sgn}(-0.3) = -1$$

$$\Delta W_i = 0.1(1+1) * X_i(2) = 0.2 * X_i(2)$$

ตัวอย่างการใช้ Artificial Neural Network ในการประมาณค่า

Wind Run (km/day)	Fetch (m)	Relative Humidity (%)		
		≤ 40	40-70	≥ 70
< 175	1	0.55	0.65	0.75
-	10	0.65	0.75	0.85
-	100	0.70	0.80	0.85
-	1,000	0.75	0.85	0.85
175-425	1	0.50	0.60	0.65
-	10	0.60	0.70	0.75
-	100	0.65	0.75	0.80
-	1,000	0.70	0.80	0.80
425-700	1	0.45	0.50	0.60
-	10	0.55	0.60	0.65
-	100	0.60	0.65	0.70
-	1,000	0.65	0.70	0.75
> 700	1	0.40	0.45	0.50
-	10	0.45	0.55	0.60
-	100	0.50	0.60	0.65
-	1,000	0.55	0.60	0.65

- $X_1 = \text{In (Fetch)}$
- $X_2 = \text{Wind Run (km / day)}$ โดยกำหนดให้ X_2 คือค่าจุดถึงกลางชั้น ซึ่งมีค่าเท่ากับ 175, 300, 562.5 และ 700 km / day
- $X_3 = \text{Relative Humidity (\%)}$ โดยกำหนดให้ X_3 คือ จุดถึงกลางชั้นซึ่งมีค่าเท่ากับ 40, 55 และ 70% สามารถเขียนฟังก์ชันความสัมพันธ์ระหว่าง K_p และ X_1, X_2, X_3 ได้ดังนี้

$$\text{■ } K_p = f (X_1, X_2, X_3) \dots\dots\dots (3)$$

$$\text{■ จากสมการที่ 3 } K_p = f (X_1, X_2, X_3)$$

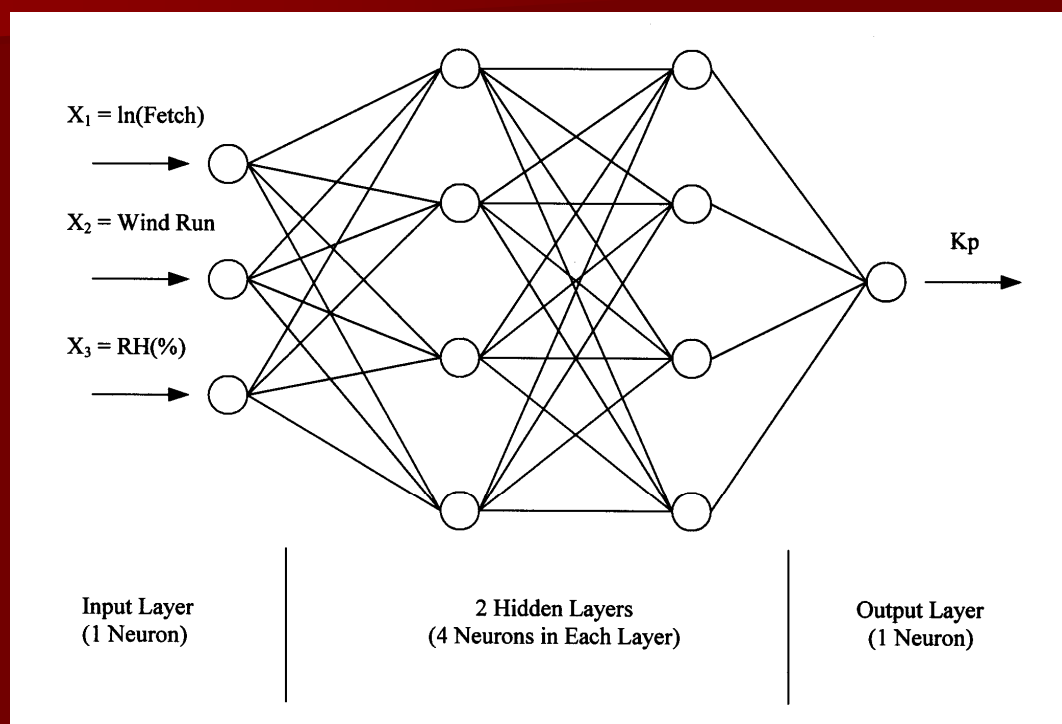
■ ให้ X_1, X_2, X_3 คือ **Inputs**

■ K_p คือ **Output**

ผลการวิเคราะห์หาสมการถดถอยของ K_p

วิธี	ฟังก์ชัน K_p	R^2	Mean Square Error
Multiple Regression	$\hat{K}_p = 0.4827 + 0.0243X_1 - 0.0004X_2 + 0.0045X_3$	0.9496	0.00069
Indicator Regression Techniques	$\hat{K}_p = 0.5944 + 0.0243X_1 - 0.0583X_2 - 0.1333X_3 - 0.2083X_4 + 0.0813X_5 + 0.1344X_6$ (วราวุธ. 2543)	0.9597	0.00055

โครงสร้าง **Artificial Neurons** และการเชื่อมโยงของ **Neurons** ในแต่ละ **Layers**



ขั้นตอนการทำงาน (คำนวณ) ของ Neurons ใน ANNs

- (1) Neurons ใน Input Layer แปลงค่า X_i ให้มีค่าอยู่ระหว่าง -1 ถึง 1 โดยใช้ Scaling Function ง่าย ๆ เช่น

$$X'_i = 2 \left(\frac{X_i - X_{\min_i}}{X_{\max_i} - X_{\min_i}} \right) - 1 \quad \dots\dots\dots(4)$$

เมื่อ $i = 1, 2, 3$

- (2) แต่ละ Neurons ใน Hidden Layer ที่ 1 จะคำนวณค่าผลรวม (Activation) ดังสมการ

$$I1_j = \sum_{i=1}^3 (W_{ij} \cdot X'_i) + \theta_j \quad \dots\dots\dots(5)$$

เมื่อ $i = 1, 2, 3$ และ $j = 1, 2, 3, 4$; θ_j = bias term ของ Neuron j

แล้วแปลงโดยใช้ Transfer Function แบบ Logistic เช่น

$$Y1_j = \frac{1}{1 + e^{-I1_j}} \quad \dots\dots\dots(6)$$

แต่ละ Neurons ใน Hidden Layer ที่ 2 จะนำผลการคำนวณของ 4 Neurons ใน Hidden Layer ที่ 1 มาคำนวณในทำนองเดียวกัน

$$I2_k = \sum_{j=1}^4 (W_{jk} \cdot Y1_j) + \theta_k \quad \dots\dots\dots(7)$$

เมื่อ $j = 1, 2, 3, 4$ และ $k = 1, 2, 3, 4$; θ_k = bias term ของ Neuron k

$$Y2_k = \frac{1}{1 + e^{-I2_k}} \dots\dots\dots(8)$$

(3) Neuron ใน Output Layer ซึ่งมี 1 Neuron จะทำการคำนวณหาผลลัพธ์ Kp โดยใช้วิธีการคำนวณแบบเดียวกับ Neurons อื่น ๆ

$$I3 = \sum_{k=1}^4 w_k Y2_k + \theta \dots\dots\dots(9)$$

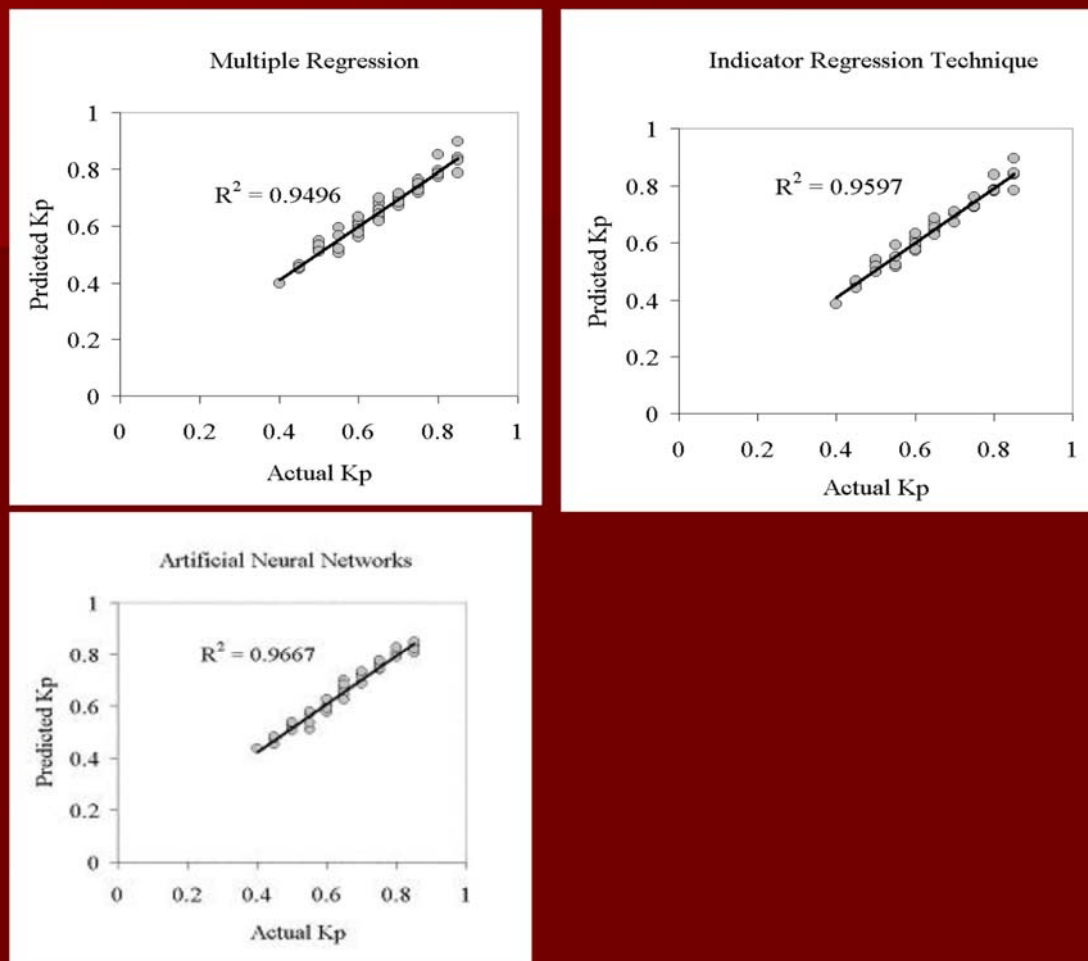
$$Y3 = \frac{1}{1 + e^{-I3}} \dots\dots\dots(10)$$

$$Kp = f(Y3) \dots\dots\dots(11)$$

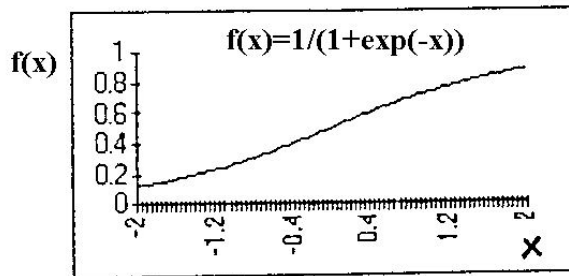
f(Y3) จะมีลักษณะคล้ายฟังก์ชันในการแปลงค่า X_i เป็น X_i'

$$f(Y3) = a_0 \left(\frac{Y3 - a_1}{a_2} \right) - a_3 \dots\dots\dots(12)$$

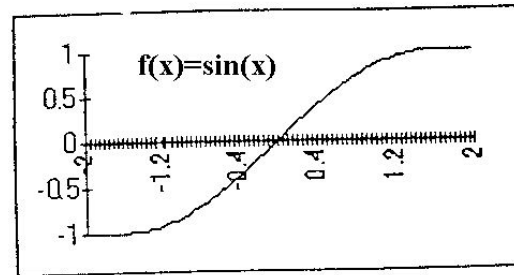
เมื่อ a_0, a_1, a_2, a_3 คือค่าสัมประสิทธิ์ของสมการ



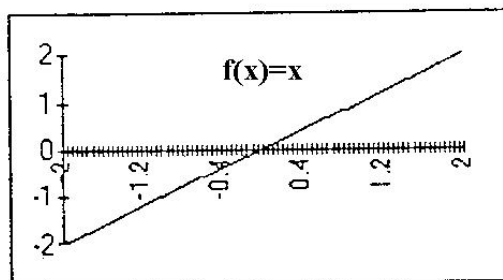
Functions



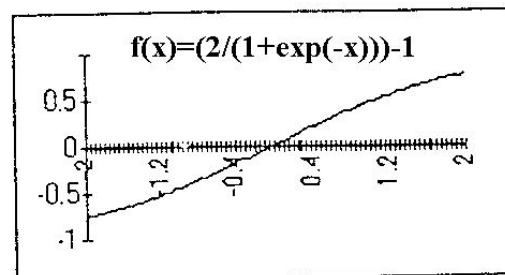
Logistic(Sigmoid Logistic)



Sine



Linear



Symmetric Logistic

Sub Fire_ANN-IRT (X(), Kp())

Dim I as double
Static Y2(4) as double
Static Y3(4) as double

' X(1) is x1
' X(2) is x2
' X(3) is x3
' Kp(1) is Kp

if (X(1) < 0) then X(1) = 0
if (X(1) > 6.9) then X(1) = 6.9
X(1) = 2 * X(1) / 6.9 - 1

if (X(2) < 175) then X(2) = 175
if (X(2) > 700) then X(2) = 700
X(2) = 2 * (X(2) - 175) / 525 - 1

if (X(3) < 40) then X(3) = 40
if (X(3) > 70) then X(3) = 70
X(3) = 2 * (X(3) - 40) / 30 - 1

I = 1.14147
I = I + X(1) * 1.328911
I = I + X(2) * .4655749
I = I + X(3) * .5334326
Y2(1) = 1 / (1 + exp(-I))

I = .3858289
I = I + X(1) * -3.603253E-03
I = I + X(2) * .6842717
I = I + X(3) * -.2884592
Y2(2) = 1 / (1 + exp(-I))

I = -.6956364
I = I + X(1) * .1388554
I = I + X(2) * -.699429
I = I + X(3) * .3695032
Y2(3) = 1 / (1 + exp(-I))

I = .6017398
I = I + X(1) * .9828456
I = I + X(2) * -.1793359
I = I + X(3) * -.2751269
Y2(4) = 1 / (1 + exp(-I))

I = -.4723979
I = I + Y2(1) * 1.16036
I = I + Y2(2) * -2.07593
I = I + Y2(3) * 1.461054
I = I + Y2(4) * .2014299
Y3(1) = 1 / (1 + exp(-I))

I = -3.154097E-02
I = I + Y2(1) * 3.641962E-02
I = I + Y2(2) * -9.839328E-02
I = I + Y2(3) * .1360031
I = I + Y2(4) * -3.594841E-02
Y3(2) = 1 / (1 + exp(-I))

I = .2317155
I = I + Y2(1) * -1.157141
I = I + Y2(2) * 1.732874
I = I + Y2(3) * -1.300422
I = I + Y2(4) * -.6180215
Y3(3) = 1 / (1 + exp(-I))

ANNs Mathematical Model

ANNs Paper#1

Deriving A General Operating Policy for Reservoirs Using ANNs

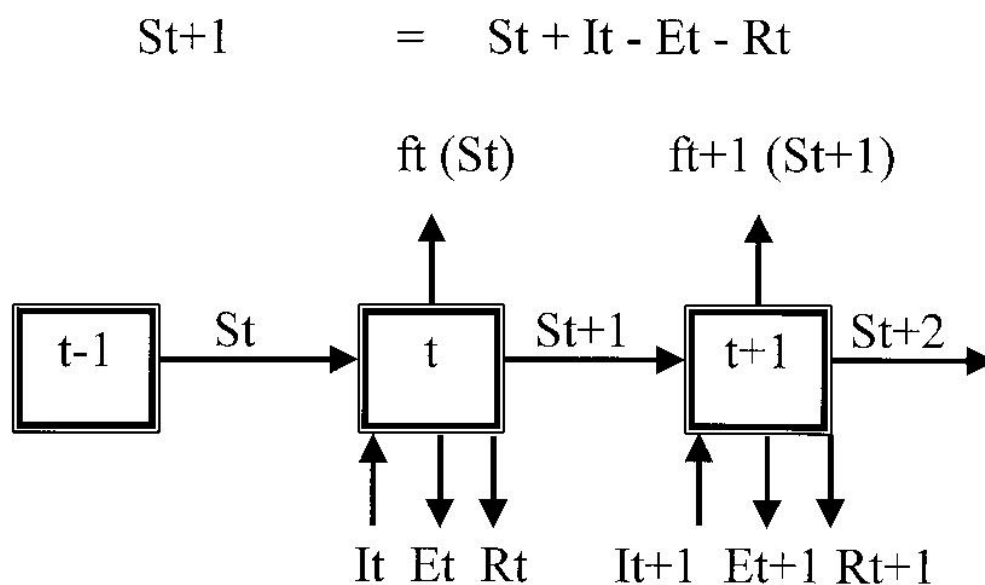
by H.Raman and V. Chandramouli(ASCE Wat.Res.Planning.&Mngt. Vol.122

Research Methodology

(1) Calculate Daily D_t

(2) Run DP at 2 week interval for 20 years of historical data to determine the optimal release(min.SSD). Using 10,5,3 and 2 mcm. Discretization levels.

DP Algorithm



R_t = Decision Variable

S_t = State Variable

Dynamic Programming Model

DP Objective Function

$$Z_t = \sum_{i=1}^T (D_i - R_i)^2 \quad (1)$$

where T = number of fortnights; R_t = release during time period t ; and D_t = irrigation demand during time period t . The recursive equation for any time period t is

$$f_t^*(S_t) = \min[Z_t + f_{t+1}^{*-1}(S_t + I_t - R_t)] \quad (2)$$

Subject to

$$0.0 \leq R_t \leq R_{t\max}$$

$$R_t \leq S_t + I_t - E_t$$

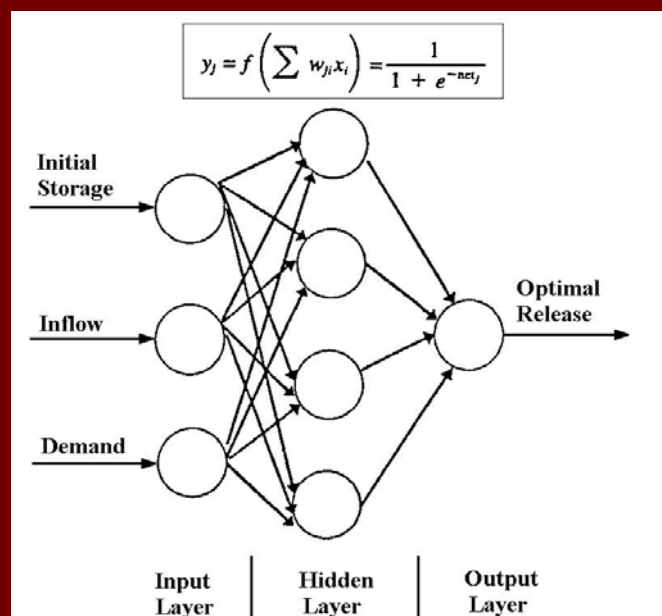
$$R_t \geq S_t + I_t - K - E_t$$

$$S_{t\min} \leq S_t \leq S_{t\max}$$

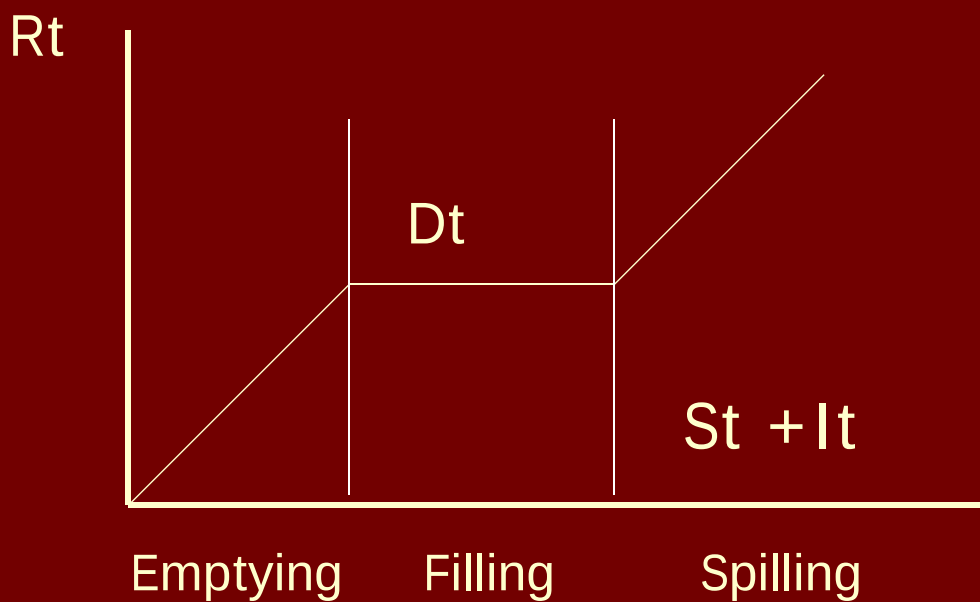
$$E_t = f(S_t, S_{t-1}, e_t)$$

- (3) Develop operating policy by
 (3.1) Multiple Regression Model
 $R_t = a \cdot S_t + b \cdot I_t + c \cdot D_t + d$

- (3.2) ANNs



(3) Standard Operating Policy(SOP)



(4) Compare the performance of derived policy DPR, DPN, SOP and SDP by simulating the reservoir operations for 3 years period and using SSD performance index as indicator

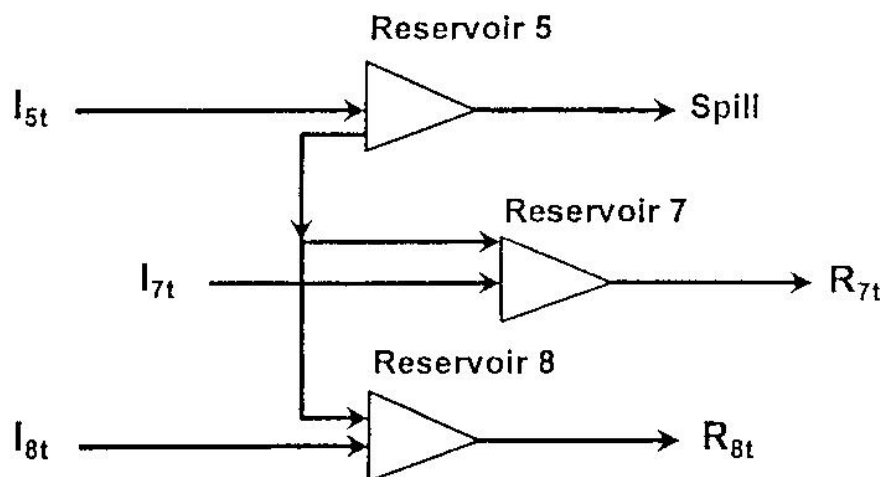
Sum of Fortnightly Squared Deficit and Total Spill for Three Years of Historical Data

Discretization step (Mm ³) (1)	DPR Model		DPN Model		SDP Model		SOP model (8)
	Total squared deficit (2)	Total spill (Mm ³) (3)	Total squared deficit (4)	Total spill. (Mm ³) (5)	Total squared deficit (6)	Total spill (Mm ³) (7)	
10	6,086.56	28.05	5,953.81	13.72	11,707.9	—	16,616.98
5	5,919.75	25.29	5,626.43	0.0	7,259.76	—	—
3	5,869.66	20.79	5,700.09	10.0	8,381.52	—	—
2	5,875.16	22.88	5,586.59	0.52	8,847.36	—	—

ANNs Paper #2

Multireservoir Modeling with Dynamic Programming and ANN

by V.Chandramouli and H.Ramman.



3-Reservoir System in Parambikulam Aliyar Project

$$\text{minimize } Z = \sum_{t=1}^T ((D_{7t} - R_{7t})^2 + (D_{8t} - R_{8t})^2) \quad (2)$$

where T = number of 2-week periods (here $T = 24$ since year-by-year optimization is considered for the three-reservoir operation model); R_{7t} and R_{8t} = release during time period t from Reservoirs 7 and 8, respectively; and D_{7t} and D_{8t} = irrigation demand during time period t in Reservoirs 7 and 8, respectively.

The recursive equation for the three-reservoir problem for a given time period t is

$$f_t^n(S_{5t}, S_{7t}, S_{8t}) = \underset{R_{5t}, R_{7t}, R_{8t}}{\text{minimize}} [Z_t + f_{t+1}^{n-1}(S_{5t+1}, S_{7t+1}, S_{8t+1})]$$

$$0.0 \leq R_{5t} \leq R_{5t \max}; \quad R_{5t} \leq S_{5t} + I_{5t} - E_{5t}$$

$$R_{5t} \geq S_{5t} + I_{5t} - E_{5t} - k_5; \quad S_{5t \min} \leq S_{5t} \leq S_{5t \max}$$

$$E_{5t} = f(S_{5t}, S_{5t-1}, e_{5t}); \quad 0.0 \leq R_{7t} \leq R_{7t \max}$$

$$R_{7t} \leq S_{7t} + I_{7t} - E_{7t} + 0.25R_{5t}$$

$$R_{7t} \geq S_{7t} + I_{7t} - E_{7t} - k_7 + 0.25R_{5t}$$

$$S_{7t \min} \leq S_{7t} \leq S_{7t \max}; \quad E_{7t} = f(S_{7t}, S_{7t-1}, e_{7t})$$

$$0.0 \leq R_{8t} \leq R_{8t \max}; \quad R_{8t} \leq S_{8t} + I_{8t} - E_{8t} + 0.75R_{5t}$$

$$R_{8t} \geq S_{8t} + I_{8t} - E_{8t} - k_8 + 0.75R_{5t}$$

$$S_{8t \min} \leq S_{8t} \leq S_{8t \max}; \quad E_{8t} = f(S_{8t}, S_{8t-1}, e_{8t})$$

DPN₃ Model

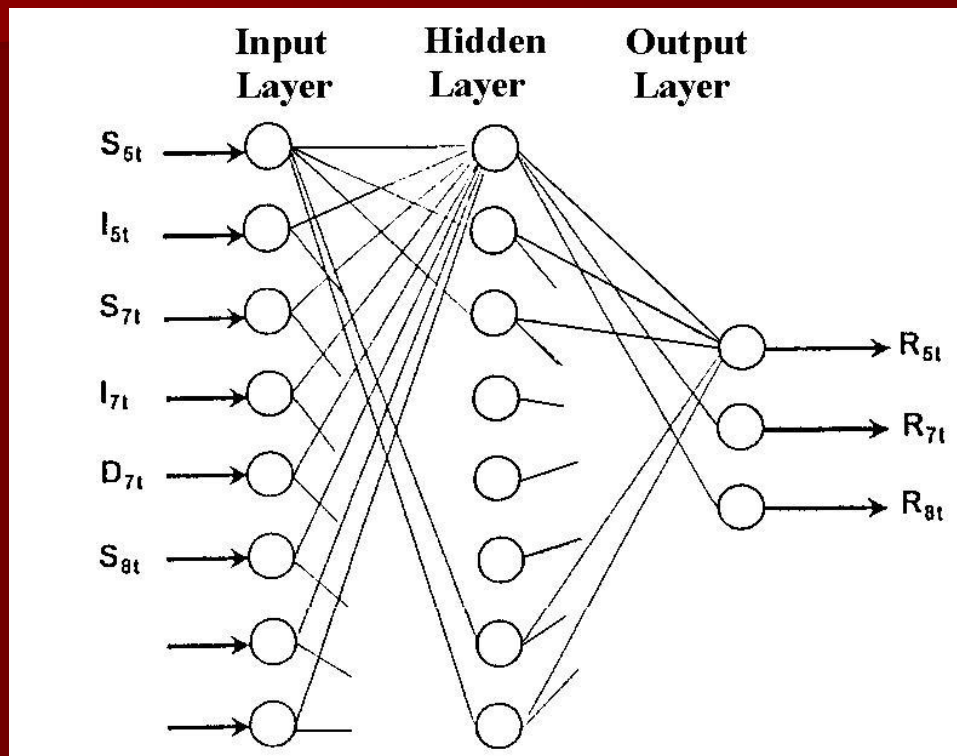
Multiple Linear Regression(DPR₃)Model

$$R_{5t} = a_1 S_{5t} + a_2 I_{5t} + a_3 S_{7t} + a_4 I_{7t} + a_5 D_{7t} + a_6 S_{8t} + a_7 I_{8t} + a_8 D_{8t} \quad (6)$$

$$R_{7t} = b_1 S_{5t} + b_2 I_{5t} + b_3 S_{7t} + b_4 I_{7t} + b_5 D_{7t} + b_6 S_{8t} + b_7 I_{8t} + b_8 D_{8t} \quad (7)$$

$$R_{8t} = c_1 S_{5t} + c_2 I_{5t} + c_3 S_{7t} + c_4 I_{7t} + c_5 D_{7t} + c_6 S_{8t} + c_7 I_{8t} + c_8 D_{8t} \quad (8)$$

Neural Network (DPN₃) Model



Performance of DPN₃ and DPR₃

Reservoir Node	Objective Function Value	
	for 19 Years of Data	
	DPN3	DPR3
7	201933.1	240599.9
8	13070.7	16372.6

สรุป

Artificial Neural Networks เป็นเทคนิคใหม่ที่น่าสนใจ ซึ่งสามารถนำมาใช้ในการประมาณค่า หรือการพยากรณ์ทางวิศวกรรมชลประทาน โดยมีความคลาดเคลื่อนต่ำ แต่อาจมีปัญหายุ่งยากในการพัฒนา **Algorithm** และการเขียนโปรแกรมเพื่อค้นหา **Weight** ที่เหมาะสม อย่างไรก็ตามปัจจุบันมี **Commercial Softwares** สำหรับช่วยออกแบบและวิเคราะห์ **ANNs** ซึ่งจะช่วยให้การใช้ **ANNs** ไม่ยุ่งยากอย่างที่คิด

Available
ANNs
Softwares

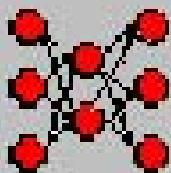


This copy is licensed to:
Asian Institute of Tech.
P.O. Box 2754
Bangkok 10501, Thailand

NeuroShell[®] 2

Release 3.0

WGS Copyright ©1993-96 Ward Systems Group, Inc.



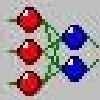
Engineering Plus, Inc.
<http://209.186.150.231>
engplus@gw.total-web.net

BackProp Plus

Neural Network Process Modeling

Beta Demo 3.41

Copyright 1996 Engineering Plus, Inc.



EasyNN^{8.01}

Day 28

The easy way to build neural networks

Thank you for evaluating EasyNN.
The standard facilities are still enabled. No restrictions have been switched on.

The full registered version has some extra facilities.
Trained neural networks can be queried using external query files.
Networks can be saved to text files for use in other applications.

[License agreement](#)

[Register and get the full version](#)

[Continue without registering](#)

Copyright _ 1999 - 2001 by Stephen Wolstenholme All Rights Reserved

WinNN

WinNN 1.2

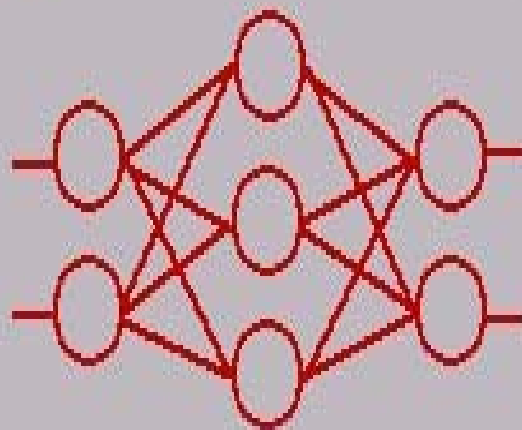
Written By Y. Danon

This is a shareware version of WinNN You may use it for 30 days. If after 30 days you would like to continue using WinNN you must register WinNN and purchase the licensed version.

[OK](#)

[Registration information](#)

QwikNet 2.23



Copyright © 1996-1999

Craig Jensen

cjensen@kagi.com

NeuNet PRO

"Discover the Patterns In Your Data"

Version 2.2

Copyright ©1998, 1999 CorMac Technologies Inc.

รายการ						
ประเภทโปรแกรม	share	share	share	com	share	share
การนำเข้าข้อมูล	text	text	dbf	spc.	spc.	spc.
แสดงภาพโครงข่าย	No	Yes	No	No	Yes	Yes
แสดงกราฟ Error	Yes	Yes	Yes	Yes	Yes	Yes
แสดง scatter graph	No	No	Yes	No	No	No
แสดง time series graph	No	No	Yes	No	No	No
แสดงข้อมูลน้ำหนัก	Yes	Yes	Yes	Yes	Yes	Yes
การส่งออกข้อมูล	text	text	dbf	spc.	spc.	spc.
การสร้างโปรแกรม	Yes	No	No	Yes	No	No



I hope that
ANNs will be
useful to
you.

Thank You

ANNs Paper#1

Deriving A General Operating Policy for Reservoirs Using ANNs

by H.Raman and V. Chandramouli

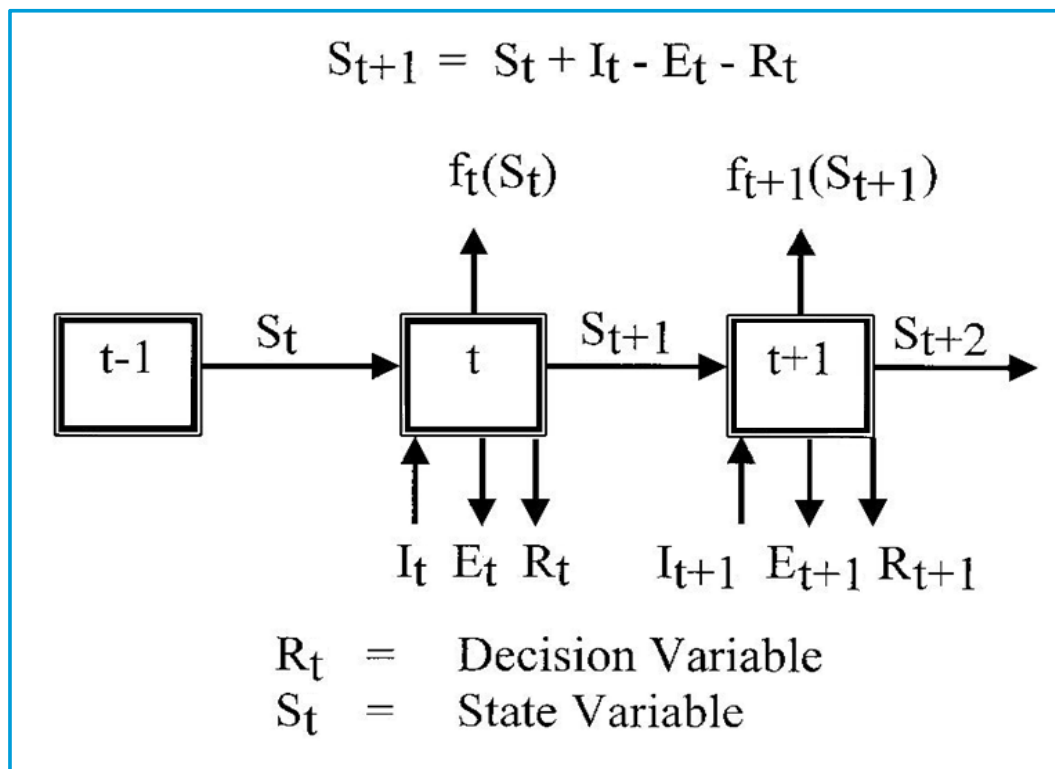
(ASCE Wat.Res.Planning.&Mngt. Vol.122)

Research Methodology

(1) Calculate Daily D_t

(2) Run DP at 2 week interval for 20 years of historical data to determine the optimal release (min.SSD). Using 10, 5, 3 and 2 mcm. Discretization levels.

DP Algorithm



Dynamic Programming Model

DP Objective Function

$$Z_t = \sum_{i=1}^T (D_i - R_i)^2 \quad (1)$$

where T = number of fortnights;

R_i = release during time period i ;

and D_i = irrigation demand during time period i .

The recursive equation for any time period t is

$$f_t^*(S_t) = \min[Z_t + f_{t+1}^*(S_t + I_t - R_t)] \quad (2)$$

Subject to

$$0.0 \leq R_t \leq R_{\max}$$

$$R_t \leq S_t + I_t - E_t$$

$$R_t \geq S_t + I_t - K - E_t$$

$$S_{\min} \leq S_t \leq S_{\max}$$

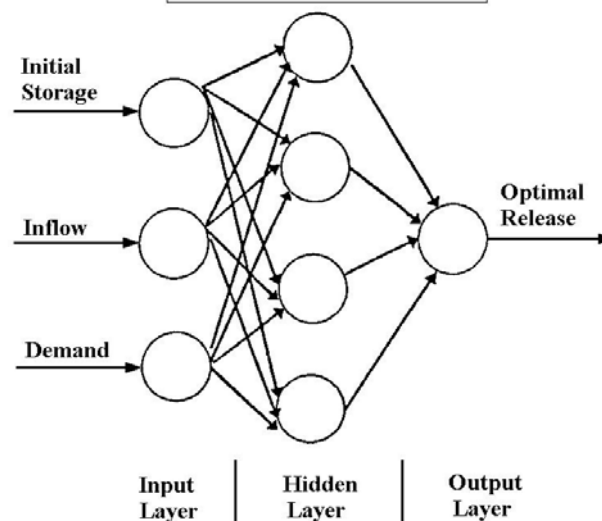
$$E_t = f(S_t, S_{t-1}, e_t)$$

(3) Develop operating policy by (3.1) Multiple Regression Model

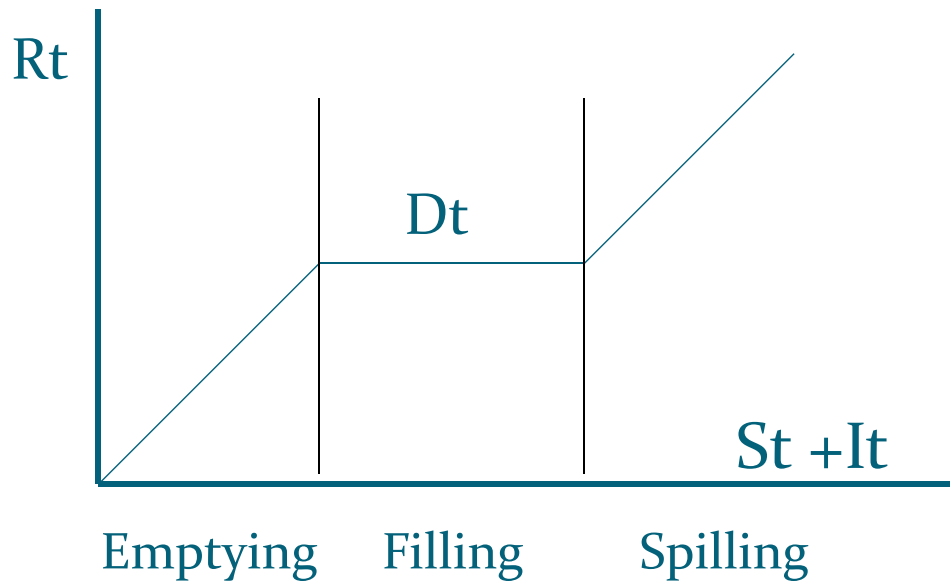
$$R_t = a * S_t + b * I_t + c * D_t + d$$

(3.2) ANNs

$$y_j = f\left(\sum w_{ji}x_i\right) = \frac{1}{1 + e^{-net_j}}$$



(4) Standard Operating Policy(SOP)



(5) Compare the performance of derived policy DPR, DPN, SOP and SDP by simulating the reservoir operations for 3 years period and using SSD performance index as indicator

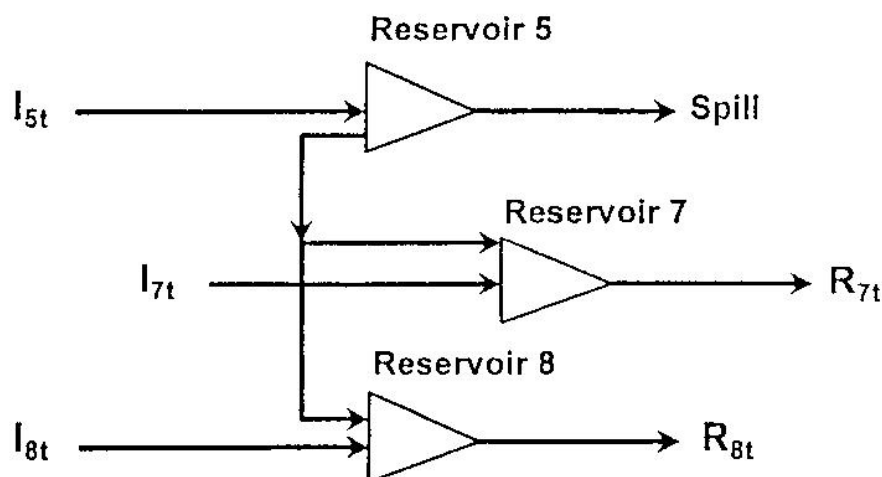
Sum of Fortnightly Squared Deficit and Total Spill for Three Years of Historical Data

Discretization step (Mm ³) (1)	DPR Model		DPN Model		SDP Model		SOP model (8)
	Total squared deficit (2)	Total spill (Mm ³) (3)	Total squared deficit (4)	Total spill. (Mm ³) (5)	Total squared deficit (6)	Total spill (Mm ³) (7)	
10	6,086.56	28.05	5,953.81	13.72	11,707.9	—	16,616.98
5	5,919.75	25.29	5,626.43	0.0	7,259.76	—	—
3	5,869.66	20.79	5,700.09	10.0	8,381.52	—	—
2	5,875.16	22.88	5,586.59	0.52	8,847.36	—	—

ANNs Paper #2

Multireservoir Modeling with Dynamic Programming and ANN

by V.Chandramouli and H.Ramman.



3-Reservoir System in Parambikulam Aliyar Project

DP Model

$$\text{minimize } Z = \sum_{t=1}^T ((D_{7t} - R_{7t})^2 + (D_{8t} - R_{8t})^2) \quad (2)$$

where T = number of 2-week periods (here $T = 24$ since year-by-year optimization is considered for the three-reservoir operation model); R_{7t} and R_{8t} = release during time period t from Reservoirs 7 and 8, respectively; and D_{7t} and D_{8t} = irrigation demand during time period t in Reservoirs 7 and 8, respectively.

The recursive equation for the three-reservoir problem for a given time period t is

$$f_t^n(S_{5t}, S_{7t}, S_{8t}) = \underset{R_{5t}, R_{7t}, R_{8t}}{\text{minimize}} [Z_t + f_{t+1}^{n-1}(S_{5t+1}, S_{7t+1}, S_{8t+1})]$$

$$0.0 \leq R_{5t} \leq R_{5t\max}; \quad R_{5t} \leq S_{5t} + I_{5t} - E_{5t}$$

$$R_{5t} \geq S_{5t} + I_{5t} - E_{5t} - k_5; \quad S_{5t\min} \leq S_{5t} \leq S_{5t\max}$$

$$E_{5t} = f(S_{5t}, S_{5t-1}, e_{5t}); \quad 0.0 \leq R_{7t} \leq R_{7t\max}$$

$$R_{7t} \leq S_{7t} + I_{7t} - E_{7t} + 0.25R_{5t}$$

$$R_{7t} \geq S_{7t} + I_{7t} - E_{7t} - k_7 + 0.25R_{5t}$$

$$S_{7t\min} \leq S_{7t} \leq S_{7t\max}; \quad E_{7t} = f(S_{7t}, S_{7t-1}, e_{7t})$$

$$0.0 \leq R_{8t} \leq R_{8t\max}; \quad R_{8t} \leq S_{8t} + I_{8t} - E_{8t} + 0.75R_{5t}$$

$$R_{8t} \geq S_{8t} + I_{8t} - E_{8t} - k_8 + 0.75R_{5t}$$

$$S_{8t\min} \leq S_{8t} \leq S_{8t\max}; \quad E_{8t} = f(S_{8t}, S_{8t-1}, e_{8t})$$

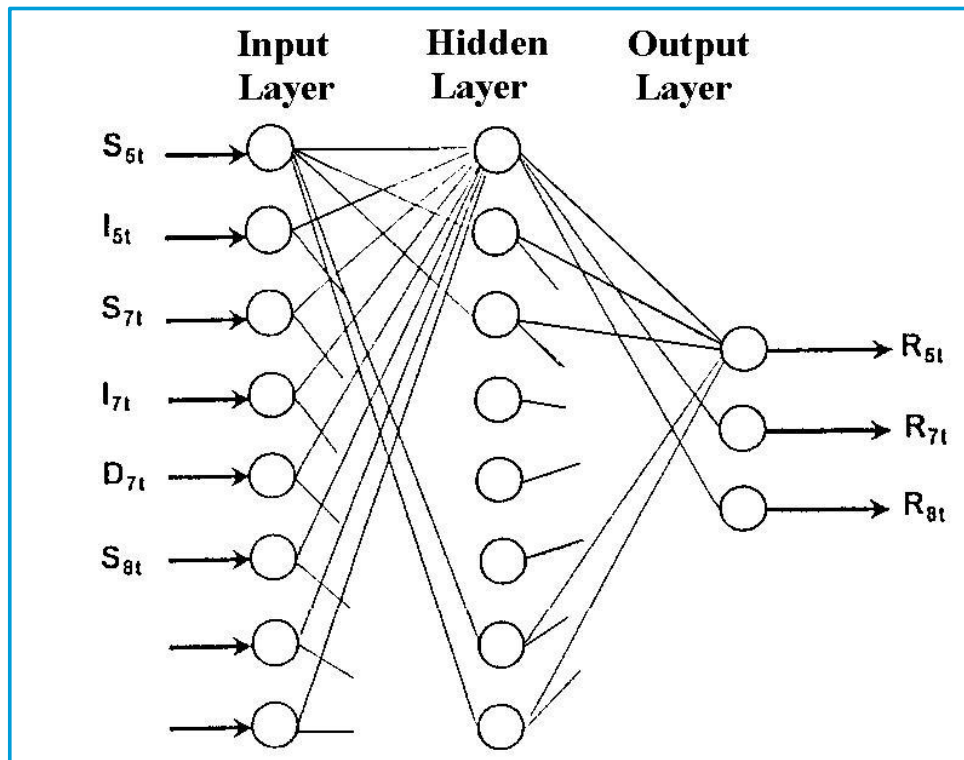
Multiple Linear Regression(DPR3)Model

$$R_{5t} = a_1 S_{5t} + a_2 I_{5t} + a_3 S_{7t} + a_4 I_{7t} + a_5 D_{7t} + a_6 S_{8t} + a_7 I_{8t} + a_8 D_{8t} \quad (6)$$

$$R_{7t} = b_1 S_{5t} + b_2 I_{5t} + b_3 S_{7t} + b_4 I_{7t} + b_5 D_{7t} + b_6 S_{8t} + b_7 I_{8t} + b_8 D_{8t} \quad (7)$$

$$R_{8t} = c_1 S_{5t} + c_2 I_{5t} + c_3 S_{7t} + c_4 I_{7t} + c_5 D_{7t} + c_6 S_{8t} + c_7 I_{8t} + c_8 D_{8t} \quad (8)$$

Neural Network (DPN3) Model



Performance of DPN3 and DPR3

Reservoir Node	Objective Function Value	
	for 19 Years of Data	
	DPN3	DPR3
7	201933.1	240599.9
8	13070.7	16372.6

Artificial Neural Networks

โดย ดร.วราวุธ วุฒินิชย์

อารัมภบท

ยุคนี้เป็นยุคคอมพิวเตอร์ ใครที่ไม่รู้จักคอมพิวเตอร์คงมีน้อยมาก ในยุคคอมพิวเตอร์มีวิทยาการใหม่ๆ เกิดขึ้นมาอย่างมากมาย สำหรับคนรุ่นใหม่อาจถือว่าวิทยาการเหล่านี้เป็นเรื่องธรรมดา แต่สำหรับคนที่เรียนจบมาหลายปีแล้ว เมื่อมาได้ยินได้ฟังวิทยาการใหม่ๆ อาจรู้สึกว่ายาก หรืออาจรู้สึกว่ามีประโยชน์กับเรา แล้วแต่จะคิด

เรื่องที่ยากจะพูดถึงในชลกรฉบับวันซุชาติปีนี้ คือ Artificial Neural Networks (ANNs) บางคนแปลเป็นภาษาไทยว่า “ระบบประสาทประดิษฐ์” ฟังดูแล้วแปลกๆ ANNs ไม่ใช่ของใหม่อะไรในยุคปี 2000 แต่ยิ่งถือว่าใหม่มากสำหรับผู้เขียนซึ่งสมัยเรียนหนังสือยังไม่เคยได้ยินใครพูดถึง ยังไม่มีการสอนในมหาวิทยาลัย จำได้ว่าเมื่อ 7 ปีที่แล้วมี Professor ทางวิศวกรรมชลประทาน จาก Colorado State University สหรัฐอเมริกาได้แวะมาเยี่ยมคณะวิศวกรรมศาสตร์ มก. และได้บรรยายพิเศษให้อาจารย์ในคณะฟังเกี่ยวกับการนำเอา Expert System มาใช้ในงานวิจัยทางด้านวิศวกรรมชลประทาน และได้กล่าวถึง ANNs ผู้เขียนได้ไปร่วมฟังด้วย รู้สึกว่า ANNs นี่น่าสนใจ แต่รู้สึกว่ายากและซับซ้อนเกินกว่าจะเอามาใช้งานได้จริง ประกอบกับไม่ค่อยได้ยินคนในวงการชลประทานบ้านเราพูดถึงเรื่องนี้จึงไม่ได้ติดตาม จนกระทั่งเร็วๆ นี้ได้อ่านบทความใน Journal of Irrigation and Drainage Engineering ของ ASCE พูดถึงการเอา ANNs มาใช้ในการ Interpolate ค่า FAO Blaney-Criddle b Factor ของสมการ Potential Evapotranspiration (Trajkovic et al. 2000) ซึ่งได้กล่าวถึง ANNs แบบฟังเข้าใจง่าย ผู้เขียนเลยมีความอยากรู้เกี่ยวกับ ANNs มากขึ้น จึงได้ศึกษาค้นคว้าโดยการหาข้อมูลจาก Internet และคุยกับคนที่พอมีความรู้ทางด้านนี้ จึงคิดว่า ANNs น่าจะมีประโยชน์ต่อการศึกษาและวิจัยทางด้านวิศวกรรมชลประทาน และไม่ยากอย่างที่คิดไว้ตอนแรก จึงอยากเผยแพร่ความรู้เบื้องต้นเกี่ยวกับ ANNs ให้พวกเราชาวชลกรได้รู้จักไว้ โดยเฉพาะนิสิตปริญญาโทสาขาวิชาวิศวกรรมชลประทานภาคพิเศษที่กำลังจะเริ่มทำวิทยานิพนธ์

Artificial Neural Networks คืออะไร

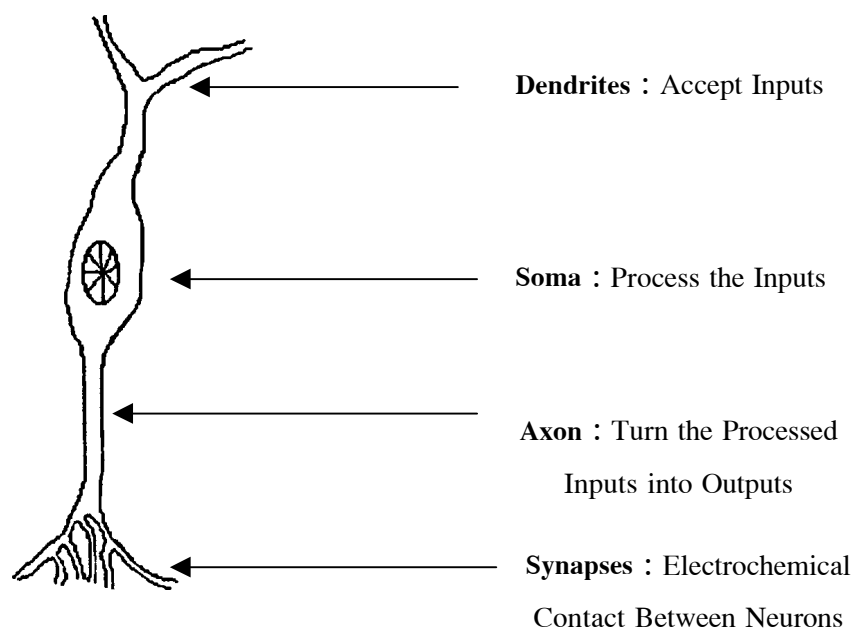
ANNs นับเป็นสาขาหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence, AI) ซึ่งถูกพัฒนาขึ้นมาในยุคคอมพิวเตอร์ ANNs คือระบบการคำนวณที่สร้างเลียนแบบการทำงานของระบบสมองมนุษย์ เพื่อใช้ประโยชน์ในการคาดคะเนเหตุการณ์จากข้อมูลที่มีอยู่เช่น การพยากรณ์อากาศ การพยากรณ์หุ้นในตลาดหลักทรัพย์ และในปัจจุบันเริ่มมีการนำเอา ANNs มาใช้ในการพยากรณ์ทางด้านอุทกวิทยา (Mason et al. 1996, Fernando and Jayawardena 1988) นอกจากนี้ได้มีการพัฒนา ANNs ไปใช้ประโยชน์ในสาขาอื่น ๆ มากมาย เช่น Image Processing และ Pattern Recognition ANNs มีชื่อเรียกอื่นอีกเช่น Machine Learning Algorithms, Natural Intelligent System

Neurons ในสมองมนุษย์

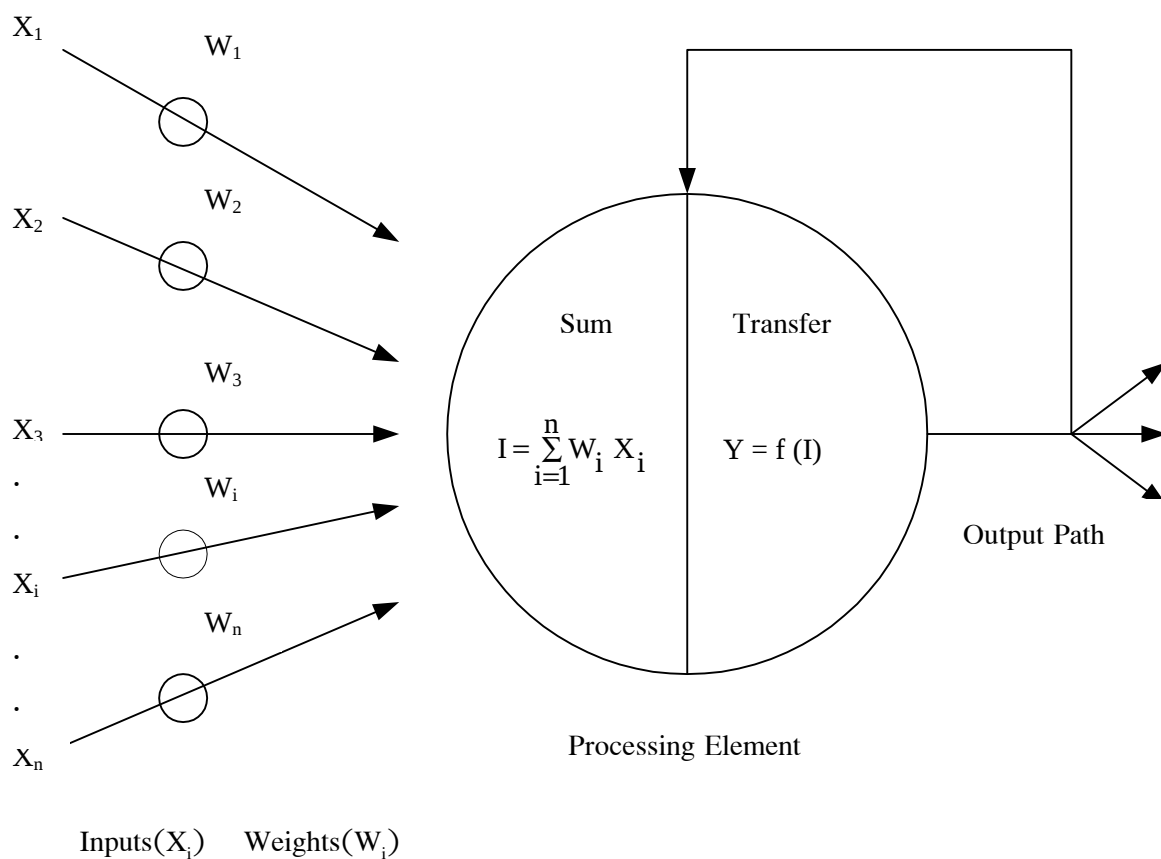
องค์ประกอบพื้นฐานของสมองมนุษย์คือ Neurons ซึ่งก็คือกลุ่มเซลล์ประสาทที่ทำหน้าที่ในการจดจำ คิด นำความรู้ และประสบการณ์ที่บันทึกไว้ในสมองมาใช้ตัดสินใจในชีวิตประจำวัน แต่ละ Neurons อาจจะเชื่อมต่อกับ Neurons อื่น ๆ มากถึง 200,000 Neurons พลังสมองของมนุษย์เกิดจาก Neurons จำนวนมากมาย และระบบการเชื่อมต่อแบบซับซ้อนของ Neurons จำนวนมากเหล่านี้ Neurons ของมนุษย์ตามหลักวิชาประสาทวิทยา (Neuroscience) จะแบ่งออกเป็น 4 ส่วน ดังรูปที่ 1 แต่ละส่วนมีฟังก์ชันการทำงานดังนี้

Dendrites	คือ	ส่วนที่ทำหน้าที่รับข้อมูล (Accept Inputs)
Soma	คือ	ส่วนที่ประมวลผลข้อมูลเบื้องต้น (Process inputs)
Axon	คือ	ส่วนที่แปลงข้อมูลที่ได้ประมวลเบื้องต้นเป็นผลลัพธ์ที่ต้องการ (Turn the processed input into outputs)
Synapses	คือ	เส้นประสาทที่ทำหน้าที่เชื่อมต่อเพื่อการสื่อสารกับ Neurons อื่นในระบบสมอง (Electrochemical contact between neurons)

ในระบบประสาทของสมองมนุษย์ Dendrites ทำหน้าที่รับข้อมูลจากแหล่งต่าง ๆ แล้วส่งให้ Soma ประมวลผลข้อมูลเบื้องต้น หลังจากนั้น Axon จะแปลงข้อมูลต่าง ๆ เป็นผลลัพธ์แล้ว Synapses จะส่งผลลัพธ์ให้ Neurons อื่นเพื่อช่วยกันสร้างผลลัพธ์ขั้นสุดท้าย ความจริงแล้วการทำงานของระบบประสาทในสมองมนุษย์มีความซับซ้อนเกินกว่าที่มนุษย์จะเข้าใจได้ ที่กล่าวมาแล้วเป็นเพียงแนวความคิดเกี่ยวกับการทำงานของ Neurons อย่างง่าย ๆ ที่จะนำมาใช้ในการพัฒนา Artificial Neural Networks ซึ่งจะกล่าวถึงในหัวข้อถัดไป



รูปที่ 1 โครงสร้างของ Neuron ในสมองมนุษย์



รูปที่ 2 โครงสร้างของ Artificial Neuron ใน ANNs

Artificial Neurons

Artificial Neurons คือหน่วยพื้นฐานของ ANNs ซึ่งจำลองการทำงานของฟังก์ชันที่ 4 ใน Neurons ของสมองมนุษย์ อย่างไรก็ตาม Artificial Neurons มีโครงสร้างการทำงานที่ง่ายกว่า Neurons ในสมองมนุษย์มาก รูปที่ 2 แสดงโครงสร้างการทำงานของ 1 Neuron ถ้ากำหนดให้ข้อมูลที่ป้อนเข้าสู่ Artificial Neurons คือ $X_1, X_2, \dots, X_n$ ข้อมูลแต่ละตัวจะถูกคูณด้วย Weight $W_1, W_2, \dots, W_n$ ตามลำดับ Artificial Neurons จะนำข้อมูลทั้ง n มารวมกัน ซึ่งเรียกว่า Activation ก่อนที่จะแปลง (Transfer) ข้อมูลเป็นผลลัพธ์แล้วส่งให้ Neurons ตัวอื่นนำไปประมวลผลในลักษณะเดียวกันเพื่อสร้างผลลัพธ์ขั้นสุดท้าย

วิธีการประมวลผลเบื้องต้นของ Artificial Neurons โดยการนำข้อมูลทั้งหมดมารวมกัน (Activation) ซึ่งสามารถแสดงในรูปของการหาผลบวก (Summation) ดังสมการ

$$I = \sum W_i \cdot X_i \quad \dots\dots\dots(1)$$

หลังจากนั้น Artificial Neurons จะแปลงค่าผลบวก หรือ Activation (I) เป็นผลลัพธ์ (Y) โดยใช้ฟังก์ชันทางคณิตศาสตร์ที่ไม่ยุ่งยากซับซ้อนมากนัก เช่น Logistic, Linear, Hyperbolic Tangent, Sine หรือ Gaussian เป็นต้น

$$Y = f(I) \quad \dots\dots\dots(2)$$

ฟังก์ชัน $f(I)$ คือ Transfer Function หรือ Response Function

ขั้นสุดท้าย Artificial Neurons ในส่วนที่ผลิตผลลัพธ์ (Output Layer) จะนำ Y ที่ได้จาก Neurons ต่าง ๆ มาประมวลเป็นผลลัพธ์ขั้นสุดท้าย (Z) ในทำนองเดียวกับ Neurons ตัวอื่นๆ คือ การ Weight การหาผลบวก (Summation) และแปลงค่า (Transfer)

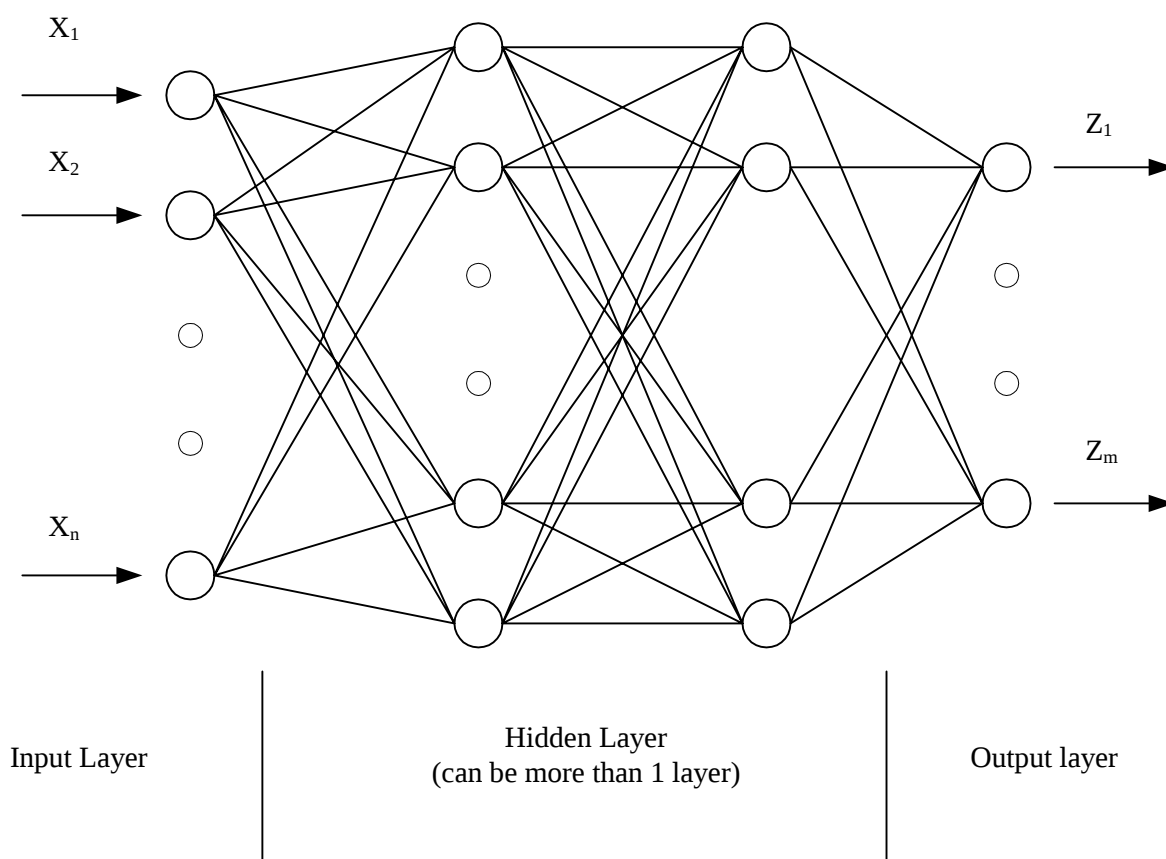
การออกแบบ Artificial Neural Networks

การออกแบบ ANNs เป็นขั้นตอนที่ยุ่งยากและซับซ้อน และต้องลองผิดลองถูก (Trial & Error) พอสมควรก่อนที่จะได้ ANNs ที่ต้องการ ขั้นตอนการออกแบบ ANNs พอจะอธิบายได้ดังนี้

- ◆ กำหนด Neurons ในแต่ละ Layer โดยกำหนดให้ Layer แรกคือ Input Layer และ Layer สุดท้าย คือ Output Layer ส่วน Layer ที่อยู่ตรงกลาง

เรียกว่า Hidden Layer อาจมี 1 หรือมากกว่า 1 Layer ก็ได้ ดังรูปที่ 3 จำนวน Neurons ใน Input Layer จะเท่ากับจำนวน Inputs (X) และจำนวน Neurons ใน Output Layer จะเท่ากับจำนวน Outputs (Z) Neurons ใน Hidden Layers คือ Neurons ที่ทำหน้าที่ในการประมวลผลข้อมูลเพื่อแปลง Inputs เป็น Outputs

- ◆ กำหนดการเชื่อมโยง (Connection) ระหว่าง Neurons ซึ่งอยู่ต่าง Layers และ Neurons ใน Layer เดียวกัน
- ◆ สอน Network ให้เรียนรู้ (Learn) ความสัมพันธ์ระหว่างข้อมูลและผลลัพธ์ที่ต้องการ เพื่อสร้างค่า Connection Weight (W_i) ที่เหมาะสม โดยใช้ชุดข้อมูลที่เรียกว่า Training Data Set



รูปที่ 3 รูปแบบการเชื่อมโยงของ Artificial Neurons ใน ANNs

รูปแบบของการเชื่อมโยงระหว่าง Neuron

Neurons แต่ละตัวจะต่อกับ Neurons ตัวอื่นตามระบบการเชื่อมต่อที่สร้างขึ้น Output ของ Neurons ตัวหนึ่งจะเป็น Input ของ Neurons ตัวอื่นต่อไป การเชื่อมต่ออาจเป็นแบบเชื่อมทางเดียว (Unidirectional) หรือแบบเชื่อม 2 ทาง (Two-way connection) ก็ได้

Neurons ใน Layer หนึ่งอาจเชื่อมโยงหรือไม่เชื่อมโยงกับ Neurons ตัวอื่นใน Layer เดียวกันก็ได้ แต่ Neurons ใน Layer หนึ่งจะต้องเชื่อมโยงกับ Neurons อย่างน้อย 1 ตัวใน Layer อื่นเสมอ

การเชื่อมโยงของ Neurons ระหว่าง Layer ซึ่งเรียกว่า Inter-Layer Connections สามารถแบ่งเป็นแบบต่าง ๆ ได้ดังนี้

◆ Fully Connected

Neurons แต่ละตัวใน Layer ที่ 1 จะเชื่อมต่อกับ Neurons ทุกตัวใน Layer ที่ 2

◆ Partially Connected

Neurons แต่ละตัวใน Layer ที่ 1 เชื่อมต่อกับ Neurons บางตัวใน Layer ที่ 2 เท่านั้น

◆ Feed Forward

Neurons ใน Layer ที่ 1 ส่ง Output ให้ Neurons ใน Layer ที่ 2 แต่จะไม่ได้รับ Feedback จาก Neurons ใน Layer ที่ 2

◆ Bi-directional

Output จาก Neurons ใน Layer ที่ 2 จะถูกส่งกลับมาเป็น Input ของ Neurons ใน Layer ที่ 1

ในโครงสร้างของ ANNs ที่ซับซ้อนมาก ๆ Neurons จะมีการเชื่อมโยงกันเองใน Layer เดียวกัน ซึ่งเรียกว่า Intra-Layer Connection ซึ่งจะไม่ขอลำถึงในที่นี้

กระบวนการเรียนรู้ (Learning) ของ Artificial Neural Networks

Artificial Neurons สามารถเรียนรู้ได้เช่นเดียวกับสมองมนุษย์ ดังนั้นบางคนจึงเรียก ANNs ว่า Machine Learning Algorithm Neurons เรียนรู้โดยการปรับค่า Connection Weight เพื่อให้ได้ผลลัพธ์ที่มีความคลาดเคลื่อน (Error) น้อยที่สุด กำลังของการเชื่อมโยงระหว่าง Neurons จะแสดงอยู่ในรูปของค่า Weight

ความสามารถในการเรียนรู้ของ ANNs ขึ้นอยู่กับโครงสร้างของ ANNs ที่ออกแบบไว้และวิธีการในการฝึกอบรม (Training) ซึ่งสามารถแบ่งวิธีการเรียนรู้ได้เป็น 2 แบบคือ

1. Unsupervised Learning

ในระบบการเรียนรู้แบบ Unsupervised Hidden Neurons จะปรับตัวเองโดยไม่ต้องการความช่วยเหลือจากภายนอก ไม่ต้องมีตัวอย่างผลลัพธ์เพื่อให้ Neurons ได้เรียนรู้ กระบวนการเรียนรู้แบบนี้เรียกว่า Learning by Doing

2. Supervised Learning

เป็นระบบการเรียนรู้ซึ่งต้องการครูช่วยสอน ซึ่งครูในความหมายนี้คือ ชุดข้อมูล ฝึกอบรม (Training Data Set) Neurons ใน Hidden Layer จะปรับค่า Weight แบบสุ่ม (Random) หรือหมุนเวียน (Rotation) ตามผลของการคำนวณผลลัพธ์ ระบบการเรียนรู้แบบนี้ อาจเรียกได้อีกอย่างหนึ่งว่า Reinforcement Learning

ระบบการเรียนรู้แบบ Supervised ซึ่งนิยมใช้สำหรับกรณีที่มีหลาย Layer และมีระบบการเชื่อมโยงแบบ Feed Forward คือ Back Propagation ซึ่งถือเป็นระบบการเรียนรู้ที่มีระบบการส่งข้อมูลเกี่ยวกับความคลาดเคลื่อน ในการคำนวณผลลัพธ์กลับไปปรับค่ากำลังในการเชื่อมโยงระหว่าง Layer เพื่อเพิ่มประสิทธิภาพในการคำนวณของระบบ

ตัวอย่างการใช้ Artificial Neural Network ในการประมาณค่า

เพื่อให้เกิดความเข้าใจในทฤษฎีของ ANNs ดียิ่งขึ้น จะขอยกตัวอย่างการใช้ ANNs ลดความคลาดเคลื่อนในการประมาณค่าสัมประสิทธิ์การวัดการระเหย K_p จากค่าความเร็วลม ความชื้นสัมพัทธ์ และระยะทางจากผิวดการระเหยแบบ เอ ถึงขอบพื้นที่ปลูกหญ้า (Upwind Fetch of Low-Growing Vegetation) โดยใช้ข้อมูล K_p ของ Doorenbos and Pruitt (1977) ดังตารางที่ 1

ตารางที่ 1 ค่าสัมประสิทธิ์การระเหย K_p (Doorenbos and Pruitt, 1977)

Wind Run (km/day)	Fetch (m)	Relative Humidity (%)		
		≤ 40	40-70	≥ 70
< 175	1	0.55	0.65	0.75
	10	0.65	0.75	0.85
	100	0.70	0.80	0.85
	1,000	0.75	0.85	0.85
175-425	1	0.50	0.60	0.65
	10	0.60	0.70	0.75
	100	0.65	0.75	0.80
	1,000	0.70	0.80	0.80
425-700	1	0.45	0.50	0.60
	10	0.55	0.60	0.65
	100	0.60	0.65	0.70
	1,000	0.65	0.70	0.75
> 700	1	0.40	0.45	0.50
	10	0.45	0.55	0.60
	100	0.50	0.60	0.65
	1,000	0.55	0.60	0.65

จากตารางที่ 1 ถ้ากำหนดให้

$$X_1 = \ln (\text{Fetch})$$

$$X_2 = \text{Wind Run (km/day) โดยกำหนดให้ } X_2 \text{ คือค่าจุดถึงกลางชั้น ซึ่งมีค่าเท่ากับ 175, 300, 562.5 และ 700 km/day}$$

$$X_3 = \text{Relative Humidity (\%)} \text{ โดยกำหนดให้ } X_3 \text{ คือ จุดถึงกลางชั้น ซึ่งมีค่าเท่ากับ 40, 55 และ 70\%}$$

สามารถเขียนฟังก์ชันความสัมพันธ์ระหว่าง K_p และ X_1, X_2, X_3 ได้ดังนี้

$$K_p = f(X_1, X_2, X_3) \dots\dots\dots(3)$$

โดยทั่วไป จะสามารถหา $f(X_1, X_2, X_3)$ ได้โดยวิธีการวิเคราะห์การถดถอยแบบพหุคูณ (Multiple Regression) ทั่วไป ซึ่งเป็นที่รู้จักกันดี หรือใช้เทคนิคการวิเคราะห์การถดถอยแบบตรรกะ (Indicator Regression Techniques) ซึ่งเป็นวิธีการวิเคราะห์การถดถอยแบบพหุคูณขั้นสูง สามารถวิเคราะห์กรณีที่ตัวแปรอิสระบางตัวไม่ได้มีค่าเป็นตัวเลข แต่เป็นช่วงค่าหรือกลุ่มค่า (Categorical หรือ Qualitative Data) ได้ โดยการแปลงตัวแปรอิสระ ซึ่งมีค่า n ช่วงค่า (Classes) เป็นตัวแปรอิสระใหม่ $n-1$ ตัว ก่อนทำการวิเคราะห์การถดถอยแบบพหุคูณตามปกติ (วรารุณ. 2543)

ผลการวิเคราะห์หาฟังก์ชันของ K_p โดยวิธีการวิเคราะห์การถดถอยแบบพหุคูณ และการวิเคราะห์การถดถอยแบบตรรกะ ได้ความสัมพันธ์ดังตารางที่ 2

ตารางที่ 2 ผลการวิเคราะห์หาสมการถดถอยของ K_p

วิธี	ฟังก์ชัน K_p	R^2	Mean Square Error
Multiple Regression	$\hat{K}_p = 0.4827 + 0.0243X_1 - 0.0004X_2 + 0.0045X_3$	0.9496	0.00069
Indicator Regression Techniques	$\hat{K}_p = 0.5944 + 0.0243X_1 - 0.0583X_2 - 0.1333X_3 - 0.2083X_4 + 0.0813X_5 + 0.1344X_6$ (วรารุณ. 2543)	0.9597	0.00055

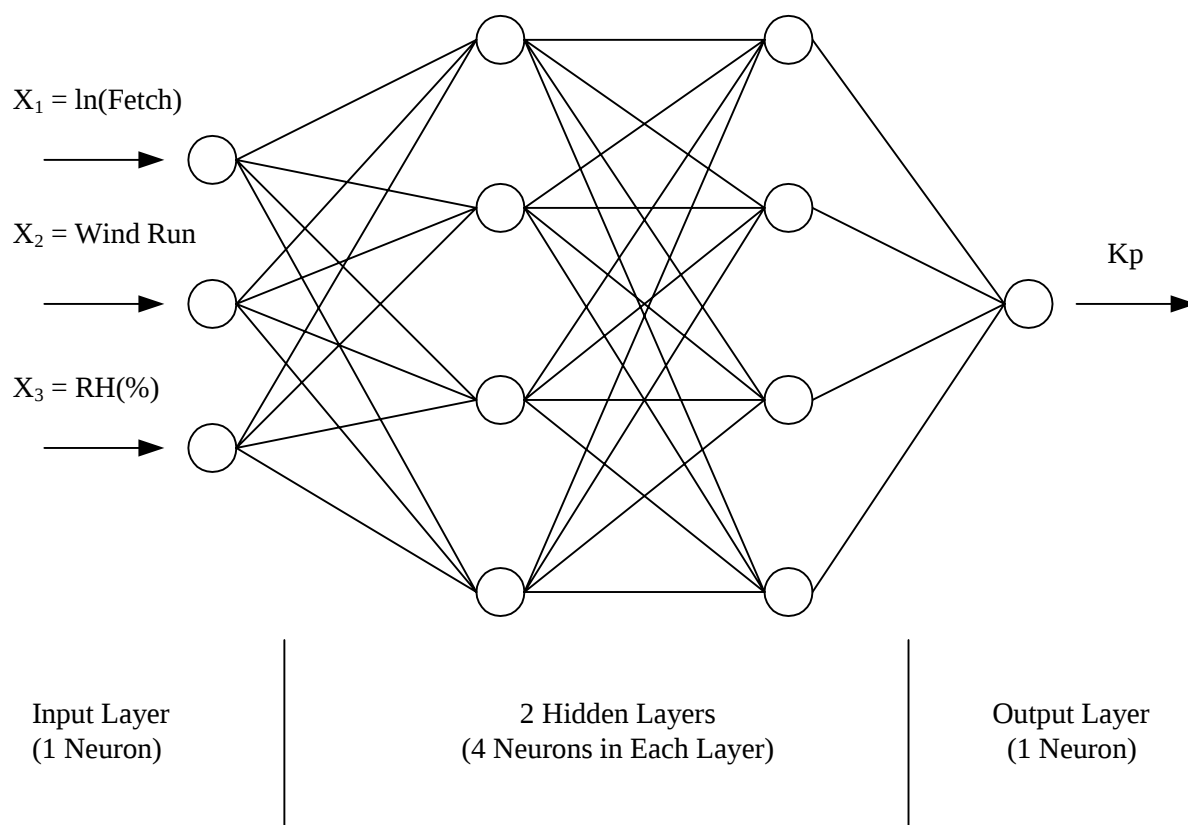
ผลการวิเคราะห์หาสมการของ $\hat{K}_p$ (Predicted K_p) ในตารางที่ 2 จะเห็นได้ว่า เทคนิคการวิเคราะห์การถดถอยแบบตรรกะให้ค่า R^2 สูงกว่า และ Mean Squared Error (MSE) ต่ำกว่า ซึ่งแสดงว่าเทคนิคการวิเคราะห์การถดถอยแบบตรรกะจะช่วยลดค่าความคลาดเคลื่อนในการประมาณค่า K_p ได้ดีกว่าสมการซึ่งได้จากการวิเคราะห์การถดถอยแบบพหุคูณธรรมดา

การใช้ Artificial Neural Networks ประมาณค่า Kp

$$\text{จากสมการที่ 3 } Kp = f(X_1, X_2, X_3)$$

ให้ X_1, X_2, X_3 คือ Inputs
 Kp คือ Output

จะสามารถออกแบบ Neurons, Input Layer, Hidden Layers และ Output Layer ได้ดังรูปที่ 4



รูปที่ 4 โครงสร้าง Artificial Neurons และการเชื่อมโยงของ Neurons ในแต่ละ Layers สำหรับการประมาณค่า Kp

จากรูปที่ 4 กำหนดให้จำนวน Neurons ใน Input Layer มี 3 Neurons ตามจำนวน Input Variables (X_1, X_2, X_3) และให้ Neurons ใน Output Layer มี 1 Neuron สำหรับ Kp ซึ่งเป็นค่าที่ต้องการให้ ANNs คำนวณ และกำหนดให้มี 2 Hidden Layers แต่ละ Layer มี 4 Neurons

ขั้นตอนการทำงาน (คำนวณ) ของ Neurons ใน ANNs ที่ออกแบบไว้ในรูปที่ 4 สามารถอธิบายพอเป็นสังเขปได้ดังนี้

- (1) Neurons ใน Input Layer แปลงค่า X_i ให้มีค่าอยู่ระหว่าง -1 ถึง 1 โดยใช้ Scaling Function ง่าย ๆ เช่น

$$X'_i = 2 \left(\frac{X_i - X_{\min_i}}{X_{\max_i} - X_{\min_i}} \right) - 1 \quad \text{.....(4)}$$

เมื่อ $i = 1, 2, 3$

- (2) แต่ละ Neurons ใน Hidden Layer ที่ 1 จะคำนวณค่าผลรวม (Activation) ดังสมการ

$$I1_j = \sum_{i=1}^3 (W_{ij} \cdot X'_i) + \theta_j \quad \text{.....(5)}$$

เมื่อ $i = 1, 2, 3$ และ $j = 1, 2, 3, 4$, ; θ_j = bias term ของ Neuron j

แล้วแปลงโดยใช้ Transfer Function แบบ Logistic เช่น

$$Y1_j = \frac{1}{1 + e^{-I1_j}} \quad \text{.....(6)}$$

แต่ละ Neurons ใน Hidden Layer ที่ 2 จะนำผลการคำนวณของ 4 Neurons ใน Hidden Layer ที่ 1 มาคำนวณในทำนองเดียวกัน

$$I2_k = \sum_{j=1}^4 (W_{jk} \cdot Y1_j) + \theta_k \quad \text{.....(7)}$$

เมื่อ $j = 1, 2, 3, 4$ และ $k = 1, 2, 3, 4$; θ_k = bias term ของ Neuron k

$$Y2_k = \frac{1}{1 + e^{-I2_k}} \dots\dots\dots(8)$$

(3) Neuron ใน Output Layer ซึ่งมี 1 Neuron จะทำการคำนวณหาผลลัพธ์ Kp โดยใช้วิธีการคำนวณแบบเดียวกับ Neurons อื่น ๆ

$$I3 = \sum_{k=1}^4 w_k Y2_k + \theta \dots\dots\dots(9)$$

$$Y3 = \frac{1}{1 + e^{-I3}} \dots\dots\dots(10)$$

$$Kp = f(Y3) \dots\dots\dots(11)$$

$f(Y3)$ จะมีลักษณะคล้ายฟังก์ชันในการแปลงค่า X_i เป็น X'_i

$$f(Y3) = a_0 \left(\frac{Y3 - a_1}{a_2} \right) - a_3 \dots\dots\dots(12)$$

เมื่อ a_0, a_1, a_2, a_3 คือค่าสัมประสิทธิ์ของสมการ

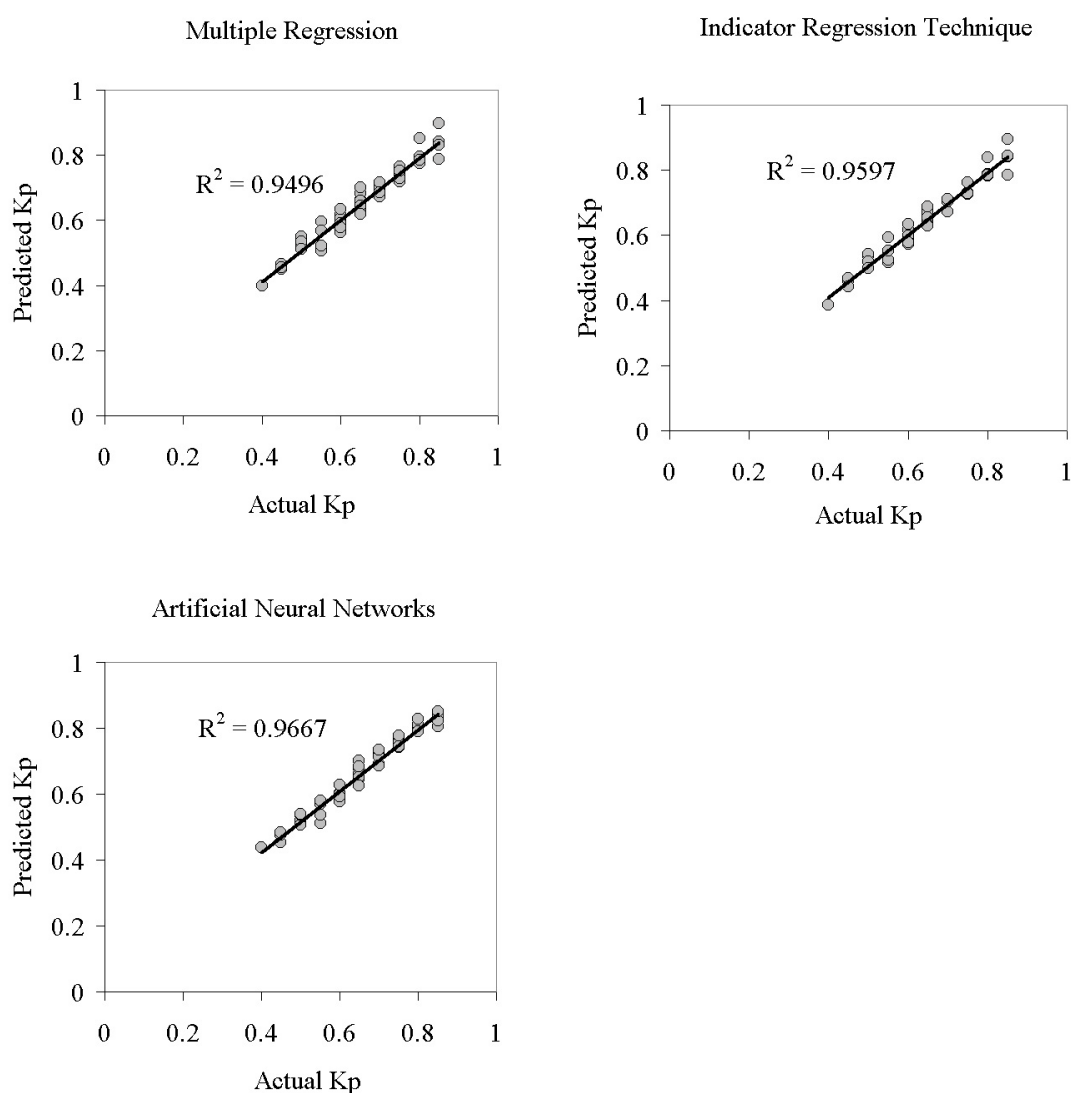
หลังจากออกแบบ ANNs แล้วจะเลือกข้อมูลบางส่วนจากตารางที่ 1 แบบสุ่ม (Random) เพื่อฝึกอบรม ANNs ตามกระบวนการเรียนรู้แบบ Supervised จนได้ความค่าคลาดเคลื่อนในการประมาณค่า Kp ตามที่กำหนด ซึ่งผลลัพธ์ที่ได้คือค่า Weight ในการเชื่อมโยง Neurons ต่าง ๆ ที่สามารถนำมาใช้ในการประมาณค่า Kp ตาม Algorithm ที่ออกแบบไว้

การออกแบบและการฝึกอบรม ANNs โดยเฉพาะการกำหนด Neurons ระบบการเชื่อมโยงระหว่าง Neurons ในแต่ละ Layer และการเลือกรูปแบบของฟังก์ชันต่าง ๆ เป็นกระบวนการทำซ้ำ ซึ่งจะต้องค่อย ๆ ทดลองเป็นขั้นเป็นตอนจนได้ผลลัพธ์ในการประมาณค่า Kp ตามที่ต้องการ

ถึงแม้ว่าฟังก์ชันในการแปลงค่าของแต่ละ Neurons จะเป็นฟังก์ชันง่าย ๆ แต่การคำนวณหา Weight ในการเชื่อมโยงในระบบที่มี Neurons จำนวนมากไม่ใช่เรื่องง่าย จึงจำเป็นต้องเขียนโปรแกรมคอมพิวเตอร์เพื่อช่วยในการค้นหาค่า Weight ที่ดีที่สุด

ผลการใช้ ANNs ที่ออกแบบไว้ในรูปที่ 4 และตามขั้นตอนที่กล่าวมาแล้ว จะทำให้ ANNs สามารถประมาณค่า Kp โดยมี Mean Squared Error เท่ากับ 0.00507 ซึ่งต่ำกว่าการประมาณค่าโดยสมการที่ได้จากการวิเคราะห์การถดถอยแบบตรรกะ

รูปที่ 5 แสดงผลการประมาณค่า Kp โดยใช้สมการถดถอยแบบพหุคูณเทียบกับ ANNs ซึ่งจะเห็นได้ว่ากำลังในการประมาณค่า Kp ของ ANNs ในรูปของค่า R^2 สูงกว่าวิธีอื่น



รูปที่ 5 ผลการเปรียบเทียบค่า Predicted Kp และค่า Actual Kp โดยสมการถดถอยและโดยวิธี ANNs

สรุป

Artificial Neural Networks เป็นเทคนิคใหม่ที่น่าสนใจ ซึ่งสามารถนำมาใช้ในการประมาณค่า หรือการพยากรณ์ทางวิศวกรรมชลประทาน โดยมีความคลาดเคลื่อนต่ำ แต่อาจมีปัญหายากในการพัฒนา Algorithm และการเขียนโปรแกรมเพื่อค้นหา Weight ที่เหมาะสม อย่างไรก็ตามปัจจุบันมี Commercial Softwares สำหรับช่วยออกแบบและวิเคราะห์ ANNs ซึ่งจะช่วยทำให้การใช้ ANNs ไม่ยุ่งยากอย่างที่คิด

เอกสารอ้างอิง

1. วรารุธ วุฒินิชย์. (2543). การวิเคราะห์การถดถอยแบบตรรกะ. ชลกรณบัณฑิต. สมาคมศิษย์เก่าวิศวกรรมชลประทาน ในพระบรมราชูปถัมภ์. มกราคม . น. 101-105.
2. Doorenbos, J. and Pruitt, W. (1977). Crop Water Requirements, FAO Irrig. and Drain. Paper 24. FAO, Rome, Italy.
3. Fernando, D. A. and Jayawardena, A. W. (1998), Runoff Forecasting Using RBF Networks with OLS Algorithm, J. Hydrologic Engrg., ASCE, 3(3), 203-209
4. Mason, J.C., Price, R. K. and Tem'me, A. (1996), A Neuron Network Model of Rainfall-Runoff Using Radial Basis Functions, J. Hydr. Res., 34(4), 537-548.
5. Trajkovic, S., Standkovi;c, M. and Todorovic, B. (2000), Estimation of FAO Blaney-Criddle b Factor by RBF Networks, J. of Irrig. and Drain. Engrg., ASCE. 111(4). 268-270.