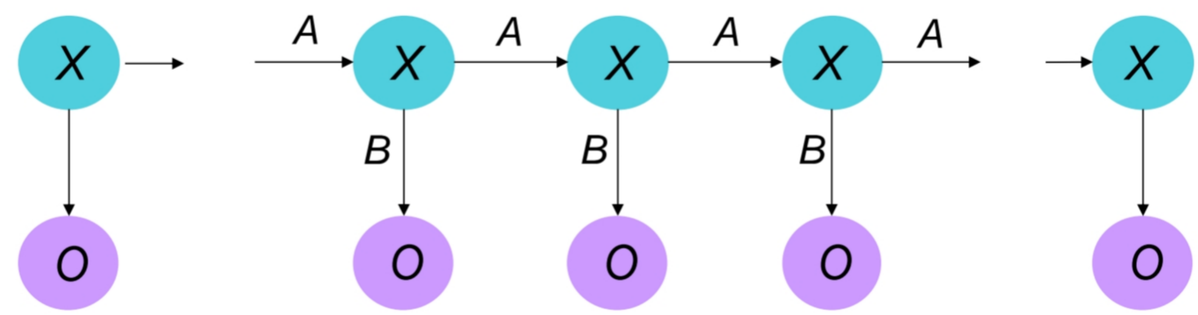
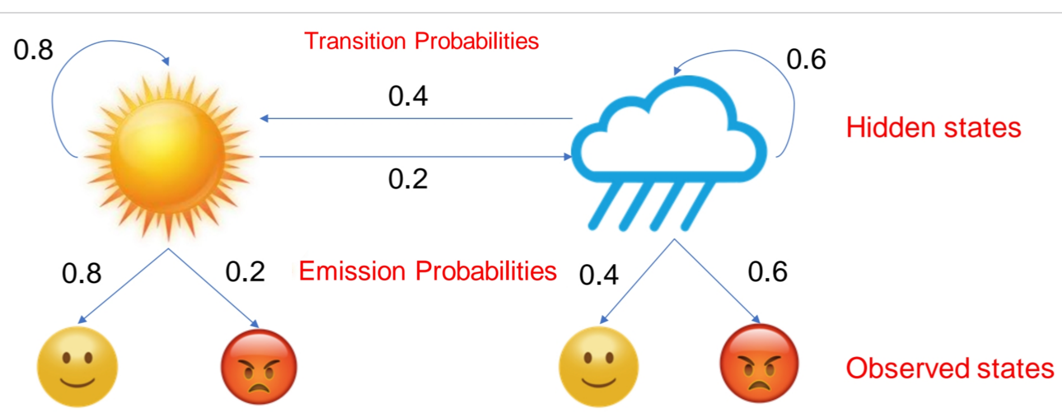
Week 8 - Hidden Markov Model

Analogi HMM :



- Graphical model
- Lingkaran = State
- Panah = Probabilities dependencies between states
- Blue nodes = Hidden states
  - ↳ Dependent only on previous state
- Purple nodes = Observed states
  - ↳ Dependent only on their corresponding hidden state

- $\{X, O, \pi, A, B\}$
  - ↳ $X = \{x_1 \ldots x_N\}$ are the values for the hidden states
  - ↳ $O = \{o_1 \ldots o_M\}$ are the values for the observed states
  - ↳ $\pi = \{\pi_i\}$ are the initial state probabilities
  - ↳ $A = \{a_{ij}\}$ are the state transition probabilities
  - ↳ $B = \{b_{ik}\}$ are the observation state probabilities

- 

Hidden Markov Model

- ☀️ ☀️ ☀️ ☀️ 🌧️ 🌧️ 🌧️ ☀️ ☀️ ☀️ ☀️ 🌧️ 🌧️ ☀️ ☀️

☀️ → ☀️	8	0.8
☀️ → 🌧️	2	0.2
🌧️ → ☀️	2	0.4
🌧️ → 🌧️	3	0.6

- ☀️ ☀️ ☀️ ☀️ 🌧️ 🌧️ 🌧️ ☀️ ☀️ ☀️ ☀️ 🌧️ 🌧️ ☀️ ☀️

  😡 🙂 🙂 🙂 😡 😡 🙂 😡 🙂 🙂 🙂 😡 🙂 🙂 🙂

☀️ → 😡	2	0.2	🌧️ → 😡	3	0.6
☀️ → 🙂	8	0.8	🌧️ → 🙂	2	0.4

- What's the probabilities that a random day is Sunny or Rainy?

  ↳ $P(\text{Sunny}) = P(\text{Sunny}\mid\text{Sunny})\cdot P(\text{Sunny}) + P(\text{Sunny}\mid\text{Rainy})\cdot P(\text{Rainy})$
  $= 0.8\cdot P(\text{Sunny}) + 0.4\cdot P(\text{Rainy})$

  ↳ $P(\text{Rainy}) = P(\text{Rainy}\mid\text{Rainy})\cdot P(\text{Rainy}) + P(\text{Rainy}\mid\text{Sunny})\cdot P(\text{Sunny})$
  $= 0.6\cdot P(\text{Rainy}) + 0.2\cdot P(\text{Sunny})$

  ∞ $P(S) = 0.8\cdot P(S) + 0.4\cdot P(R)$
  $0.2\,P(S) = 0.4\,P(R)$
  $P(S) = 2\cdot P(R)$

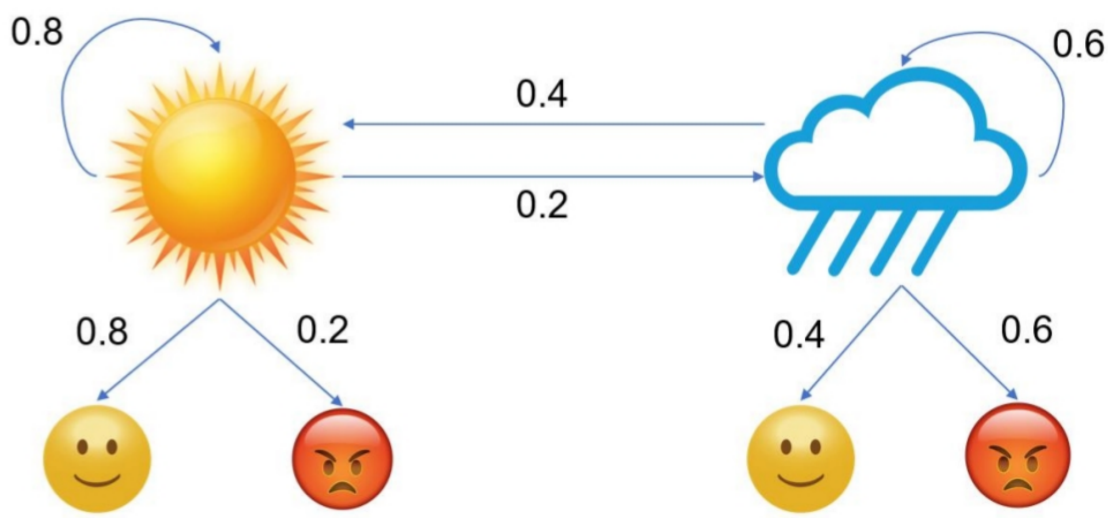
- If Bob is happy today, what's the probability that it's Sunny or Rainy?

  ↳ $P(S\mid H) = \dfrac{P(H\mid S)\cdot P(S)}{P(H)} = \dfrac{0.8\cdot\frac{2}{3}}{0.668} = 0.8$

  ↳ $P(R\mid H) = \dfrac{P(H\mid R)\cdot P(R)}{P(H)} = \dfrac{0.4\cdot\frac{1}{3}}{0.668} = 0.2$

  ↳ $P(H) = P(H\mid R)\cdot P(R) + P(H\mid S)\cdot P(S)$
  $= 0.4\cdot\frac{1}{3} + 0.8\cdot\frac{2}{3}$
  $= 0.132 + 0.536$
  $= 0.668$

- If for three days Bob is happy, grumpy, happy, what was the weather?



Viterbi Algorithm

↳

	S	R
S	0.8	0.2
R	0.4	0.6

↳

	H	G
S	0.8	0.2
R	0.4	0.6

↳ $P(S) = 2 \cdot P(R)$
$P(S) : P(R) = 2 : 1$ } Nilai phi π

S	0.67
R	0.33

$\frac{2}{3}$ $\frac{1}{3}$

↳ Mencari likelihood untuk tiap state tersembunyi :

- Likelihood pada saat t=1 : H untuk find V1

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
🙂	🙂	😡	😡	😡	🙂

0.8

0.4

S
0.67

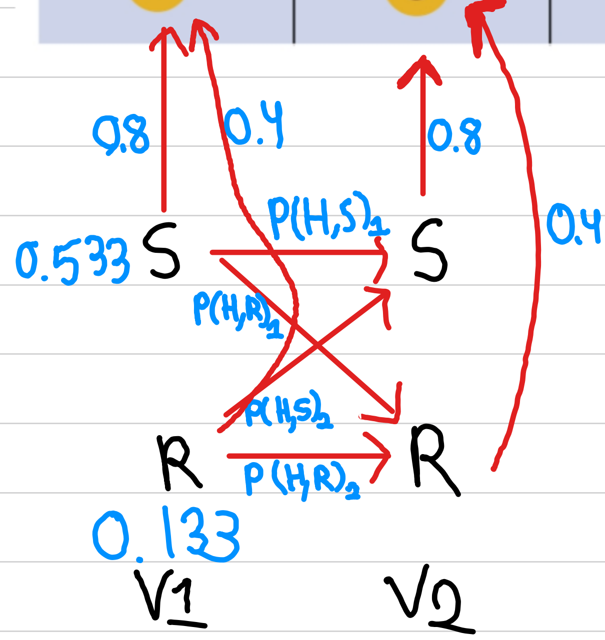
R
0.33
V_1

- $P(H,S) = P(H|S) \cdot P(S)$
$= 0.8 \cdot 0.67$
$= 0.536$

- $P(H,R) = P(H|R) \cdot P(R)$
$= 0.4 \cdot 0.33$
$= 0.132$

Likehood pada saat t=1 untuk find V2

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
🙂	🙂	😡	😡	😡	🙂

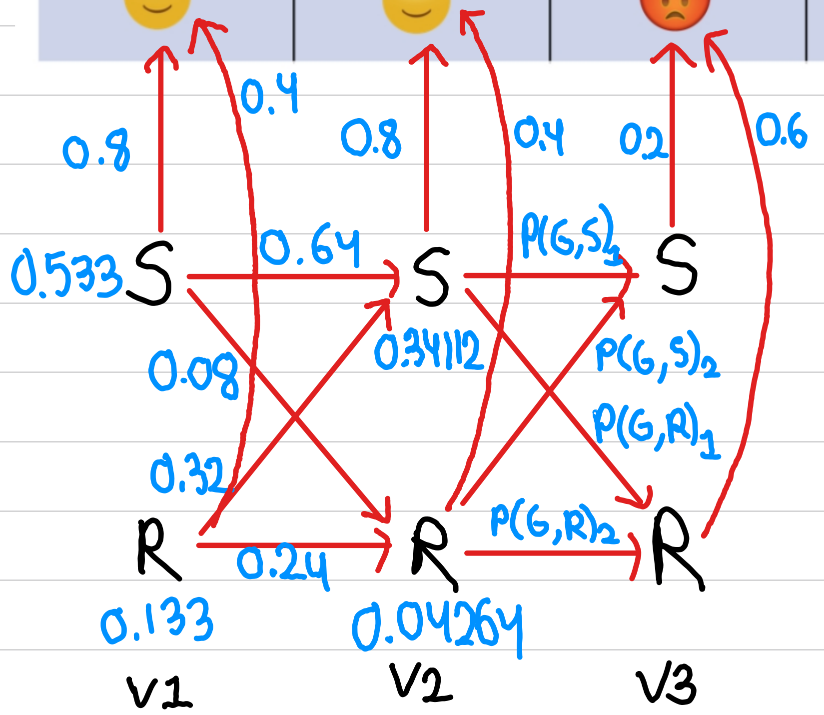


- $P(H,S)_1 = P(H|S) \cdot P(S|S)$
  $= 0.8 \cdot 0.8$
  $= 0.64$
- $P(H,R)_1 = P(H|R) \cdot P(R|S)$
  $= 0.4 \cdot 0.2$
  $= 0.08$
- $P(H,S)_2 = P(H|S) \cdot P(S|R)$
  $= 0.8 \cdot 0.4$
  $= 0.32$
- $P(H,R)_2 = P(H|R) \cdot P(R|R)$
  $= 0.4 \cdot 0.6$
  $= 0.24$

- V2 untuk S : $V2_{(1)} = P(H,S)_1 \cdot P(H,S)$
  $= 0.64 \cdot 0.533$
  $= 0.34112$
  $V2_{(2)} = P(H,S)_2 \cdot P(H,R)$
  $= 0.32 \cdot 0.133$
  $= 0.04256$
- V2 untuk R : $V2_{(1)} = P(H,R)_1 \cdot P(H,S)$
  $= 0.08 \cdot 0.533$
  $= 0.04264$
  $V2_{(2)} = P(H,R)_2 \cdot P(H,R)$
  $= 0.24 \cdot 0.133$
  $= 0.03192$
- $V2(S) = \max(0.34112 ; 0.04256) = 0.34112$
- $V2(R) = \max(0.04264 ; 0.03192) = 0.04264$

o) Likelihood pada saat $t = 3$ berdasarkan $t = 2$ untuk find V3

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
🙂	🙂	😡	😡	😡	🙂



o) $P(G,S)_1 = P(G|S) \cdot P(S|S)$
$= 0.2 \cdot 0.8$
$= 0.16$

o) $P(G,R)_1 = P(G|R) \cdot P(R|S)$
$= 0.6 \cdot 0.2$
$= 0.12$

o) $P(G,S)_2 = P(G|S) \cdot P(S|R)$
$= 0.2 \cdot 0.4$
$= 0.08$

o) $P(G,R)_2 = P(G|R) \cdot P(R|R)$
$= 0.6 \cdot 0.6$
$= 0.36$

o) V3 untuk S : $V3_{(1)} = P(G,S)_1 \cdot P(H,S)_{t-1}$
$= 0.16 \cdot 0.34112$
$= 0.05457$

$V3_{(2)} = P(G,S)_2 \cdot P(H,R)_{t-1}$
$= 0.08 \cdot 0.04264$
$= 0.0034$

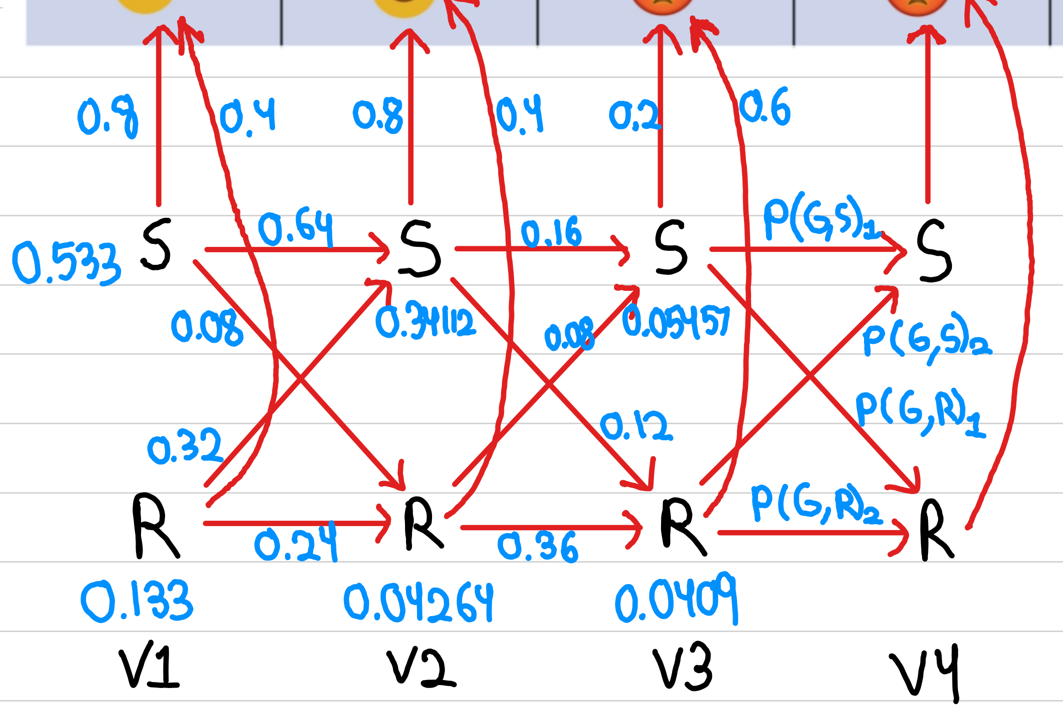
o) V3 untuk R : $V3_{(1)} = P(G,S)_2 \cdot P(H,S)_{t-1}$
$= 0.12 \cdot 0.34112$
$= 0.0409$

$V3_{(2)} = P(G,R)_2 \cdot P(H,R)_{t-1}$
$= 0.36 \cdot 0.04264$
$= 0.01535$

o) $V3(S) = \max(0.05457 ; 0.0034) = 0.05457$
o) $V3(R) = \max(0.0409 ; 0.01535) = 0.0409$

∘) Likelihood pada saat t=4 berdasarkan t=3 untuk find V4

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
🙂	🙂	😡	😡	😡	🙂



∘) $P(G,S)_1 = P(G|S) \cdot P(S|S)$
$= 0.2 \cdot 0.8$
$= 0.16$

∘) $P(G,R)_1 = P(G|R) \cdot P(R|S)$
$= 0.6 \cdot 0.2$
$= 0.12$

∘) $P(G,S)_2 = P(G|S) \cdot P(S|R)$
$= 0.2 \cdot 0.4$
$= 0.08$

∘) $P(G,R)_2 = P(G|R) \cdot P(R|R)$
$= 0.6 \cdot 0.6$
$= 0.36$

∘) V4 untuk S : $V4_{(1)} = P(G,S)_1 \cdot P(G,S)_{t-1}$
$= 0.16 \cdot 0.05457$
$= 0.00873$

$V4_{(2)} = P(G,S)_2 \cdot P(G,R)_{t-1}$
$= 0.08 \cdot 0.0409$
$= 0.00327$

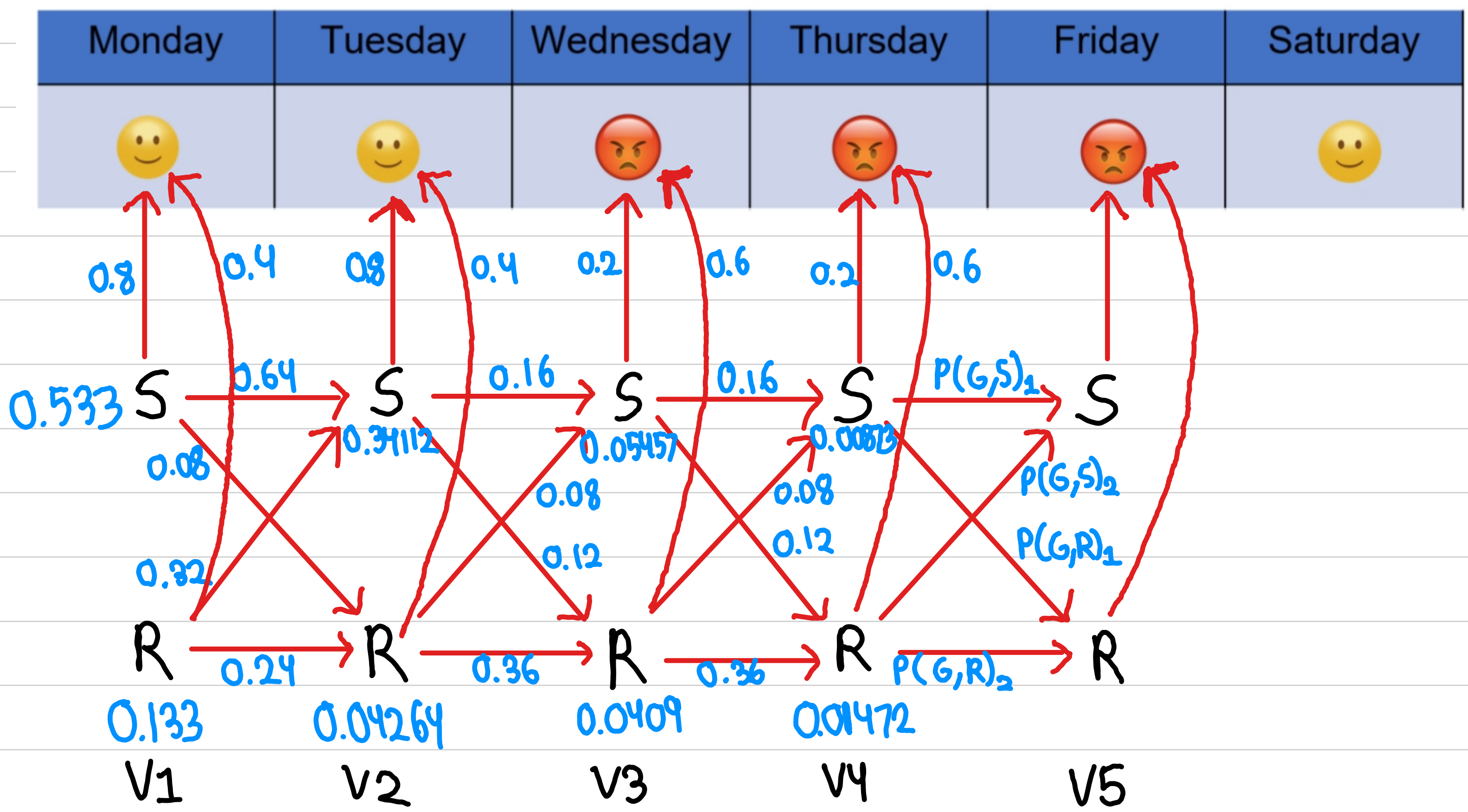
∘) V4 untuk R : $V4_{(1)} = P(G,R)_1 \cdot P(G,S)_{t-1}$
$= 0.12 \cdot 0.05457$
$= 0.00654$

$V4_{(2)} = P(G,R)_2 \cdot P(G,R)_{t-1}$
$= 0.36 \cdot 0.0409$
$= 0.01472$

∘) $V4(S) = \max(0.00873; 0.00327) = 0.00873$

∘) $V4(R) = \max(0.00654; 0.01472) = 0.01472$

Likelihood pada saat $t=5$ berdasarkan $t=4$ untuk find $V5$



$$P(G,S)_1 = P(G|S)\cdot P(S|S) = 0.2\cdot 0.8 = 0.16$$

$$P(G,S)_2 = P(G|S)\cdot P(S|R) = 0.2\cdot 0.4 = 0.08$$

$$P(G,R)_1 = P(G|R)\cdot P(R|S) = 0.6\cdot 0.2 = 0.12$$

$$P(G,R)_2 = P(G|R)\cdot P(R|R) = 0.6\cdot 0.6 = 0.36$$

$V5$ untuk S :

$$V5_{(1)} = P(G,S)_1 \cdot P(G,S)_{t-1} = 0.16 \cdot 0.00873 = 0.00139$$

$$V5_{(2)} = P(G,S)_2 \cdot P(G,R)_{t-1} = 0.08 \cdot 0.01472 = 0.00117$$

$V5$ untuk R :

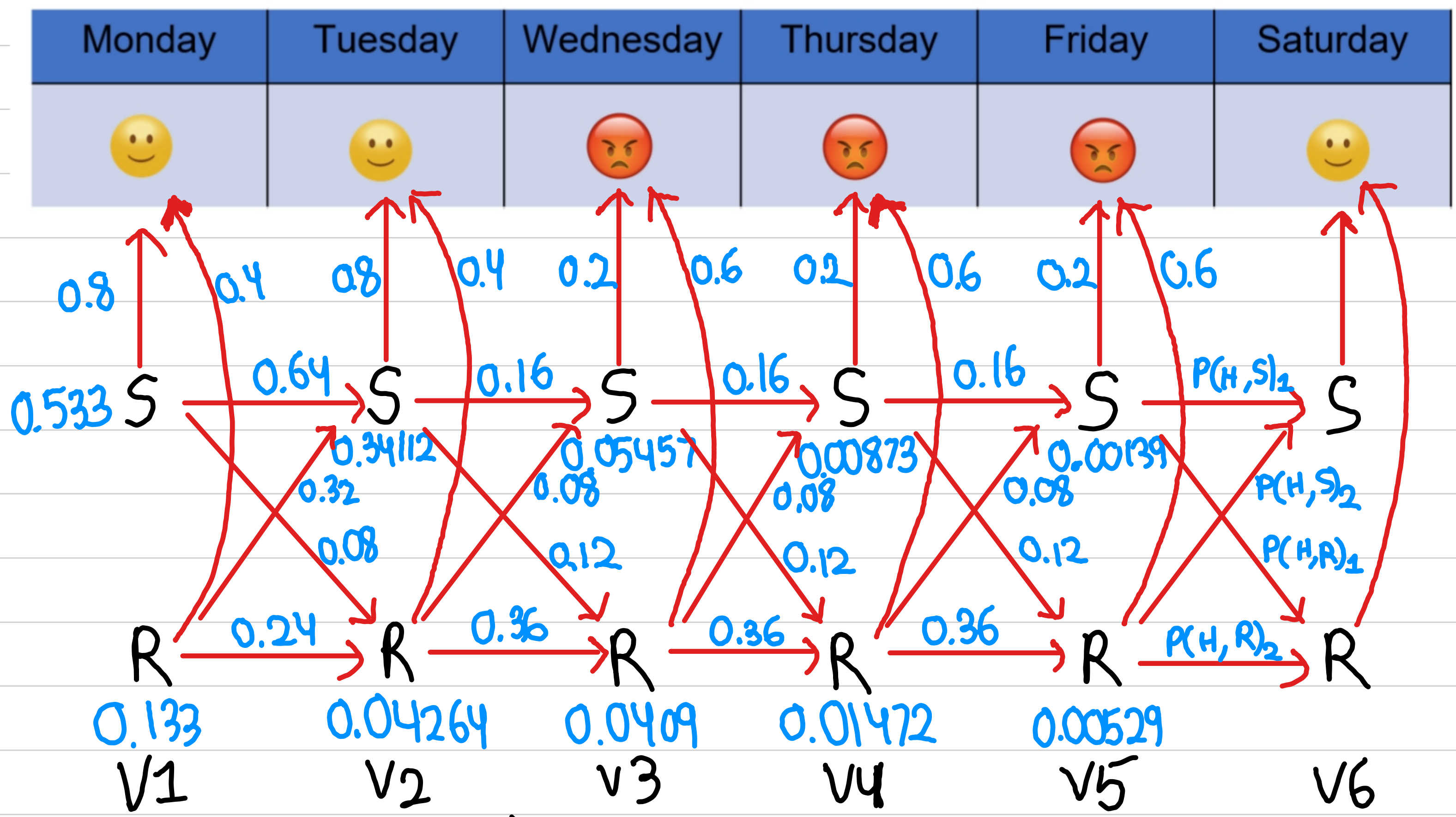
$$V5_{(1)} = P(G,R)_1 \cdot P(G,S)_{t-1} = 0.12 \cdot 0.00873 = 0.001047$$

$$V5_{(2)} = P(G,R)_2 \cdot P(G,R)_{t-1} = 0.36 \cdot 0.01472 = 0.00529$$

$V5(S) = \max(0.00139 ; 0.00117) = 0.00139$

$V5(R) = \max(0.001047 ; 0.00529) = 0.00529$

◦) Likelihood pada saat t=6 berdasarkan t=5 untuk find V6



◦) $P(H,S)_1 = P(H|S) \cdot P(S|S)$
$= 0.8 \cdot 0.8$
$= 0.64$

◦) $P(H,S)_2 = P(H|S) \cdot P(S|R)$
$= 0.8 \cdot 0.4$
$= 0.32$

◦) $P(H,R)_1 = P(H|R) \cdot P(R|S)$
$= 0.4 \cdot 0.2$
$= 0.08$

◦) $P(H,R)_2 = P(H|R) \cdot P(R|R)$
$= 0.4 \cdot 0.6$
$= 0.24$

◦) V6 untuk S : $V6_{(1)} = P(H,S)_1 \cdot P(G,S)_{t-1}$
$= 0.64 \cdot 0.00139$
$= 0.000889$

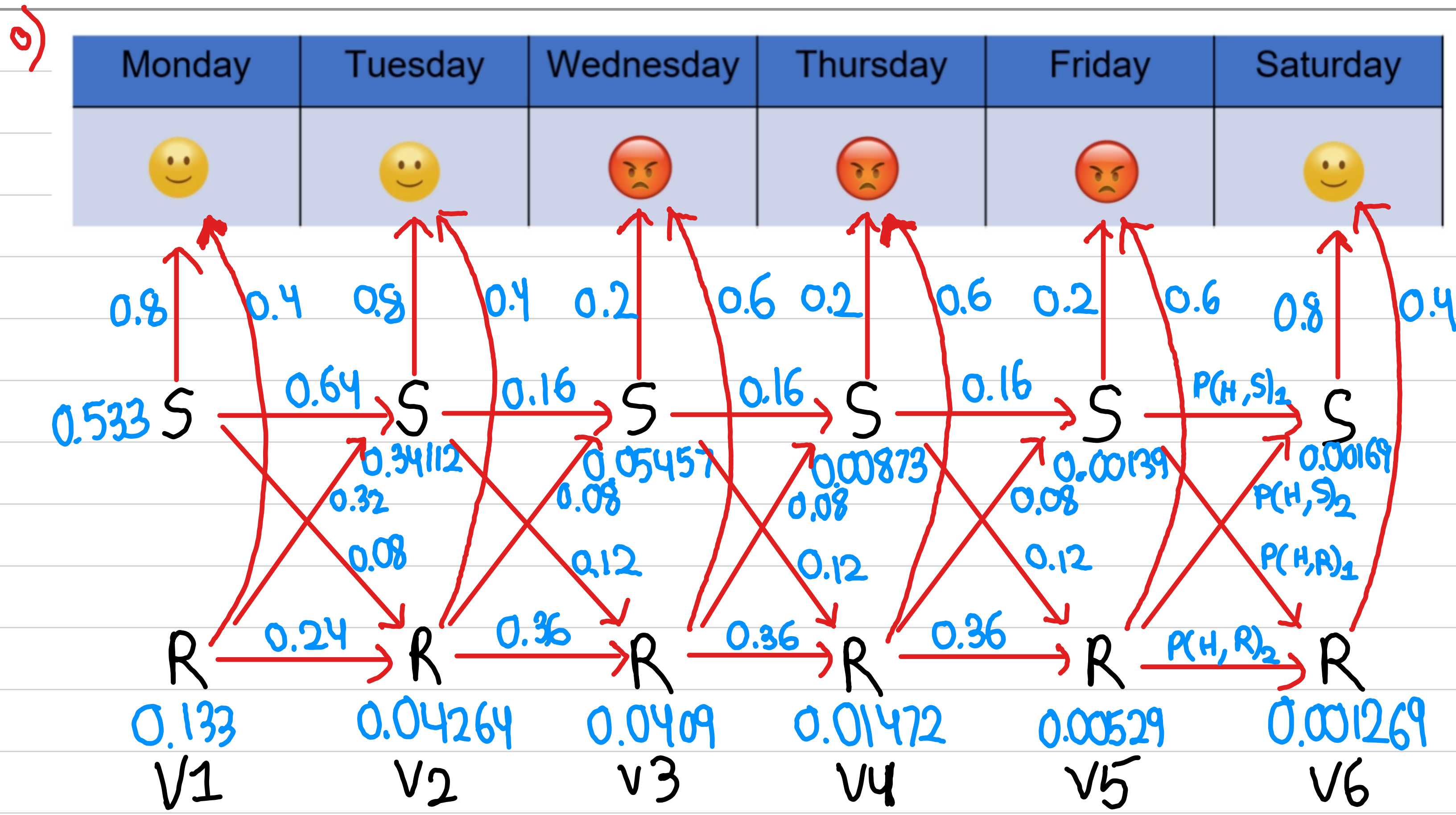
$V6_{(2)} = P(H,S)_2 \cdot P(G,R)_{t-1}$
$= 0.32 \cdot 0.00529$
$= 0.00169$

◦) V6 untuk R : $V6_{(1)} = P(H,R)_1 \cdot P(G,S)_{t-1}$
$= 0.08 \cdot 0.00139$
$= 0.0001112$

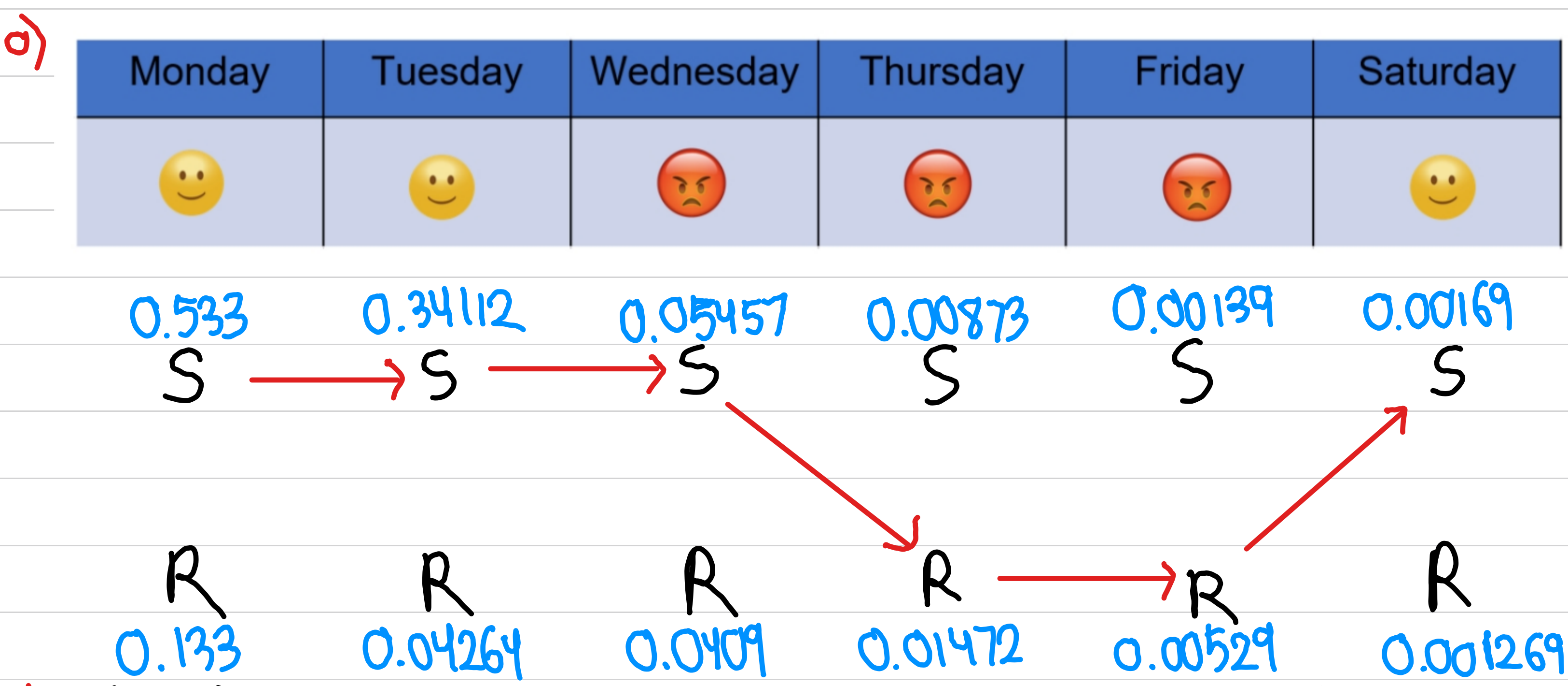
$V6_{(2)} = P(H,R)_2 \cdot P(G,R)_{t-1}$
$= 0.24 \cdot 0.00529$
$= 0.001269$

◦) $V6(S) = \max(0.000889 ; 0.00169) = 0.00169$
◦) $V6(R) = \max(0.0001112 ; 0.001269) = 0.001269$

a)



a)



a) Untuk menentukan kondisi cuaca tepat berdasarkan informasi suasana hati, ambil dengan probabilitas tinggi, maka urutan S → S → S → R → R → S

Week 10 – Expectation Maximisation Algorithm

- Merupakan iterative algorithm untuk maximizing likelihood estimation (MLE) when the model contains unobserved latent variables.

- MLE involve treating problem as optimisasi / cari mslh, dimana cari set of parameters that result in best fit for joint probability of data sample (X).

- Involve maximizing a likelihood function in order to find probability distribution and parameters that best explain observed data.

- EMA's algorithm iterate between E-Step (Expectation) and M-Step (Maximization):
  - ↳ E-Step: Create function for expectation of the log-likelihood, evaluated using current estimate for parameters
  - ↳ M-Step: Compute parameters maximizing the expected log-likehood found in E-Step

- Misalkan ada 2 koin, A dan B, each with a certain bias of landing heads, $\hat{\theta}_A$ and $\hat{\theta}_B$ dan lakukan 5 kali:
  - Pick coin randomly
  - Toss 10 kali
  - Record number of H & T

  } Lebih kurang jadi

											Coin A	Coin B
B	H	T	T	T	H	H	T	H	T	H		5 H, 5 T
A	H	H	H	H	T	H	H	H	H	H	9 H, 1 T	
A	H	T	H	H	H	H	H	T	H	H	8 H, 2 T	
B	H	T	H	T	T	T	H	H	T	T		4 H, 6 T
A	T	H	H	H	T	H	H	H	T	H	7 H, 3 T	
										Sum :	**24 H, 6 T**	**9 H, 11 T**

5 sets, 10 tosses per set

$$\hat{\theta}_A = \frac{24}{24+6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9+11} = 0.45$$

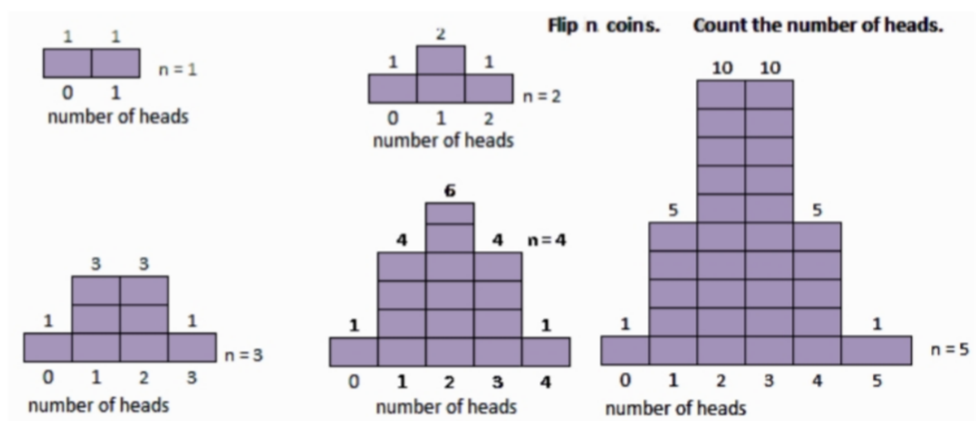
$\hat{\theta}$	The probability of getting heads
$\hat{\theta}_A$	The probability of coin A landing on head
$\hat{\theta}_B$	The probability of coin B landing on head

∘ Bagaimana kalau menentukan coin parameter tanpa mengetahui identitas dari tiap data set's coin?

↓

Solusinya pakai Expectation Maximization

∘ Binomial Distribution



∘ Jika diberikan dataset sama dari hasil coin flip, tetapi tidak ada identitas koin yg define dataset?

∘ $X = \{x_1, \ldots, x_m\}$, the observed variable

$$Z = \begin{Bmatrix} z_{1,1} & \ldots & z_{m,1} \\ \ldots & z_{i,j} & \ldots \\ z_{1,k} & \ldots & z_{m,k} \end{Bmatrix} \quad \text{where } z_{i,j} = \begin{cases} 1; & \text{if } x_i \text{ is from } j^{th} \text{coin} \\ 0; & \text{otherwise} \end{cases}$$

Z is not known / hidden / latent variable

∘ EM Algorithm:

1) Initialize some arbitrary hypothesis of parameter values $(\hat{\theta})$

$$\boxed{\hat{\theta} = \{\hat{\theta}_1, \ldots, \hat{\theta}_k\}}$$

∘ e.g.: $\hat{\theta} = \{\hat{\theta}_A, \hat{\theta}_B\} = \{0.6, 0.5\}$

2) Expectation (E-Step)

$$\boxed{E[z_{i,j}] = \frac{p(x = x_i \mid \hat{\theta} = \hat{\theta}_j)}{\sum_{n=1}^{k} p(x = x_i \mid \hat{\theta} = \hat{\theta}_n)}}$$

3) Maximization (M-Step)

$$\boxed{\hat{\theta}_j = \frac{\sum_{i=1}^{m} E[z_{i,j}] \cdot x_i}{\sum_{i=1}^{m} E[z_{i,j}]}}$$

-

H	T	T	T	H	H	T	H	T	H
H	H	H	H	T	H	H	H	H	H
H	T	H	H	H	H	H	T	H	H
H	T	H	T	T	T	H	H	T	T
T	H	H	H	T	H	H	H	T	H

5 sets, 10 tosses per set

$\rightarrow$

Set 1

H	T	T	T	H	H	T	H	T	H

- What is the probability that observe 5 H & 5 T in coin A and B given initializing parameters $\hat{\theta}_A = 0.6$, $\hat{\theta}_B = 0.5$?

  ↳ Compute likelihood of set 1 coming from coin A or B using binomial distribution with mean probability $\hat{\theta}$ on n trials with k successes

$$\boxed{p(k) = \binom{n}{k} \cdot \hat{\theta}^{k} \cdot (1 - \hat{\theta})^{n-k}}$$

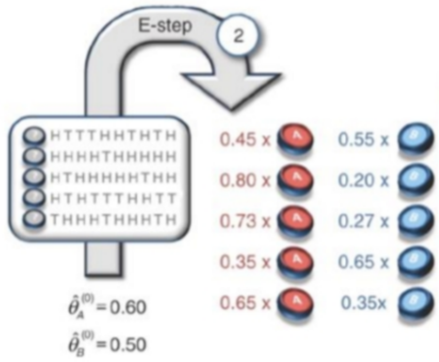
- Set 1 ada 5/10 H & 5/10 T
- Likelihood of A = $P_A(h)^h \cdot (1 - P_A(h))^{10-h} = 0.00079$
- Likelihood of B = $P_B(h)^h \cdot (1 - P_B(h))^{10-h} = 0.00097$
- Normalize to get probabilities : A = 0.45, B = 0.55

$\frac{A}{A+B}$ $\quad$ $\frac{B}{A+B}$

P(Coin=A)=0.45 $\quad$ P(Coin=B)=0.55

- E-step = Estimating likely number of H & T from :
  A ⇒ H = 0.45 · 5 heads = 2.2 heads ; T = 0.45 · 5 tails = 2.2 tails
  B ⇒ H = 0.55 · 5 heads = 2.8 heads ; T = 0.55 · 5 tails = 2.8 tails

o)



Coin A	Coin B
≈ 2.2 H, 2.2 T	≈ 2.8 H, 2.8 T
≈ 7.2 H, 0.8 T	≈ 1.8 H, 0.2 T
≈ 5.9 H, 1.5 T	≈ 2.1 H, 0.5 T
≈ 1.4 H, 2.1 T	≈ 2.6 H, 3.9 T
≈ 4.5 H, 1.9 T	≈ 2.5 H, 1.1 T
≈ 21.3 H, 8.6 T	≈ 11.7 H, 8.4 T

→ E-Step

- Compute the new probabilities for each coin $\left(\frac{H}{H+T}\right)$
- That gives you the new maximized parameter θ for each coin

o)

Coin A	Coin B
≈ 2.2 H, 2.2 T	≈ 2.8 H, 2.8 T
≈ 7.2 H, 0.8 T	≈ 1.8 H, 0.2 T
≈ 5.9 H, 1.5 T	≈ 2.1 H, 0.5 T
≈ 1.4 H, 2.1 T	≈ 2.6 H, 3.9 T
≈ 4.5 H, 1.9 T	≈ 2.5 H, 1.1 T
≈ 21.3 H, 8.6 T	≈ 11.7 H, 8.4 T

$$\widehat{\theta_A}(1) = \frac{21.3}{21.3 + 8.6} = 0.71$$

$$\widehat{\theta_B}(1) = \frac{11.7}{11.7 + 8.4} = 0.58$$

Repeat E-STEP and M-STEP until convergence

→ M-Step

The convergence stop at :

$\widehat{\theta_A}$ (10) = 0.80

$\widehat{\theta_B}$ (10) = 0.52

○ Kesimpulan EM Algorithm :

1) Choosing starting parameters
2) Estimate probability using these parameter using that each data set (x_i) came from j^{th} coin $(E[z_{i,j}])$
3) Use these probability values $(E[z_{i,j}])$ as weight on each data point when computing a new θ_j to describe each distribution
4) Summate these expected values, use maximum likelihood estimation to derive new parameter values to repeat process.

Week 11 - Certainty Factor

- CF theory is a popular alternotive to Bayesian reasoning
- Uncertainty is represented as Degree of Belief
- Express the Measure of Belief
- CF express belief in an event based on evidence/expert's assessment
  - ↳ 1.0 or 100 = Absolute truth
  - ↳ 0 = Certain falsehood

- CF bkn probabilities
- CF need not sum to 100
- Uncertain terms :

Term	Certainty Factor
Definitely not	−1.0
Almost certainly not	−0.8
Probably not	−0.6
Maybe not	−0.4
Unknown	−0.2 to +0.2
Maybe	+0.4
Probably	+0.6
Almost certainly	+0.8
Definitely	+1.0

- Syntax :

```
IF &lt;evidence&gt;/&lt;antecedent&gt;
THEN &lt;hypothesis&gt;/&lt;consequent&gt; {cf}
```

↳ cf = Belief in hypothesis H given evidence E has occurred

- CF theory is based on 2 functions :

1) Measure of Belief MB(H,E) : Number that reflects the measure of increased belief in a hypothesis H based on evidence E.

$$0 \leqslant MB(H,E) \leqslant 1$$

2) Measure of Belief MD(H,E) : Number that reflects the measure of increased disbelief in a hypothesis H based on evidence E.

$$0 \leqslant MD(H,E) \leqslant 1$$

- $$\boxed{MB(H,E) = \begin{cases} 1 & , \text{ if } p(H)=1 \\ \dfrac{\max[p(H|E), p(H)] - p(H)}{\max[1,0] - p(H)} & , \text{ otherwise} \end{cases}}$$

$$\boxed{MD(H,E) = \begin{cases} 1 & , \text{ if } p(H)=0 \\ \dfrac{\min[p(H|E), p(H)] - p(H)}{\min[1,0] - p(H)} & , \text{ otherwise} \end{cases}}$$

↳ $p(H)$ = Prior probability of hypothesis H being true
↳ $p(H|E)$ = Probability that hypothesis H is true given evidence E

- Total strength of belief / disbelief in a hypothesis :

$$\boxed{cf = \frac{MB(H,E) - MD(H,E)}{1 - \min[MB(H,E), MD(H,E)]}}$$

- CF assigned by a rule is propagated through reasoning chain. Involves establishing the net certainty of the rule consequent when evidence in rule antecedent is uncertain :

$$\boxed{cf(H,E) = cf(E) \cdot cf}$$

- Contoh : IF sky is clear
THEN the forecast is sunny {cf 0.8}
and current cf of sky is clear = 0.5
then $cf(H,E) = 0.5 \cdot 0.8 = 0.4$ → It may be sunny

- CF for conjunctive rules : IF <evidence E_1>
⋮
AND <evidence E_n>
THEN <hypothesis H> {cf}

↳ $$\boxed{cf(H, E_1 \cap E_2 \cap \ldots \cap E_n) = \min[cf(E_1), cf(E_2), \ldots, cf(E_n)] \cdot cf}$$

- CF for disjunctive rules : IF <evidence E_1>
  ⋮
  OR <evidence E_n>
  THEN <hypothesis H> {cf}

$$\boxed{cf(H, E_1 \cup E_2 \cup \ldots \cup E_n) = \max[cf(E_1), cf(E_2), \ldots, cf(E_n)] \cdot cf}$$

- Combined CF :

$$\boxed{cf(cf_1, cf_2) = \begin{cases} cf_1 + cf_2 \cdot (1 - cf_1), & \text{if } cf_1 > 0 \text{ and } cf_2 > 0 \\ \dfrac{cf_1 + cf_2}{1 - \min[|cf_1|, |cf_2|]}, & \text{if } cf_1 < 0 \text{ or } cf_2 < 0 \\ cf_1 + cf_2 \cdot (1 + cf_1), & \text{if } cf_1 < 0 \text{ and } cf_2 < 0 \end{cases}}$$

↳ cf_1 = Confidence in hypothesis H established by Rule 1
↳ cf_2 = Confidence in hypothesis H established by Rule 2
↳ $|cf_1|$ and $|cf_2|$ are absolute magnitudes of cf_1 and cf_2

- Contoh :

Evidence	KB
1. Demam cf = 0.3	R1 : IF demam AND ruam
2. Ruam cf = 0.8	THEN cacar cf = 0.6
3. Batuk darah cf = 0.7	R2 : IF batuk darah AND gatal
4. Gatal cf = 0.4	THEN cacar cf = 0.7
5. Tulang linu cf = 0.8	R3 : IF tulang linu OR pusing
6. Pusing cf = 0.4	THEN cacar cf = 0.4

1) Hitung nilai cf masing-masing rule.

R1 : $cf_1(H, E_1 \cap E_2) = \min(cf(E_1); cf(E_2)) \cdot cf$
$= \min(0.3; 0.8) \cdot 0.6$
$= 0.3 \cdot 0.6$
$= 0.18$

R2 : $cf_2(H, E_3 \cap E_4) = \min(cf(E_3); cf(E_4)) \cdot cf$
$= \min(0.7; 0.4) \cdot 0.7$
$= 0.4 \cdot 0.7$
$= 0.28$

R3 : $cf_3(H, E_5 \cup E_6) = \max(cf(E_5); cf(E_6)) \cdot cf$
$= \max(0.8; 0.4) \cdot 0.4$
$= 0.8 \cdot 0.4$
$= 0.32$

2) Hitung CF combination

cf1 & cf2 : $cf_{comb}(cf_1; cf_2) = cf_1 + cf_2 \cdot (1 - cf_1)$
$= 0.18 + 0.28 (1 - 0.18)$
$= 0.18 + 0.28 \cdot 0.82$
$= 0.4096$

cfcomb & cf3 : $cf_{comb}(cf_{comb}(cf_1; cf_2); cf_3)$
$= cf_{comb}(cf_1; cf_2) + cf_3 \cdot (1 - cf_{comb}(cf_1; cf_2))$
$= 0.4096 + 0.32 (1 - 0.4096)$
$= 0.4096 + 0.32 \cdot 0.5904$
$= 0.5986$

- Contoh :
  - If demam and ruam or pusing then cacar
    ↳ cf = max (min (demam, ruam), pusing) $\cdot cf_{rule}$
  - If demam or ruam and pusing then cacar
    ↳ cf = max (min (ruam, pusing), demam)
  - If A or B and C and D or E
    ↳ cf = max (max (min (B, C, D), A), E)
  - If A or B or C and D and E
    ↳ cf = max (max (A, B), min (C, D, E))

- Dempster - Shafer Theory :
  - Design to deal perbedaan antara uncertainty & ignorance drpd hitung probability of a proposition yg computes probability that evidence supports proposition
  - DST = Evidence theory, combines all possible outcome of the problem. Solve problems where there may be a chance that different evidence will lead to some different result
  - Consider set of propositions as a whole
  - Assign set of propositions an interval [believe, plausibility] to constraint degree of belief for each individual propositions in the set

- Belief = Probability that H is provable supported by evidence

- Belief measure bel is in [0,1]
  - 0 = No support evidence for a set of propositions / fact
  - 1 = Full support evidence for a set of propositions / fact

- Belief in fact & its negation need not sum to 1

- Plausability = Probability that H is compatible with the available evidence / can't be disapproved

- Plausability of p :
  - pl(p) = 1 - bel (not (p))
  - Reflect how evidence of not (p) relates to possibility for belief in p
  - Bel (not (p)) = 1 : Full support for not(p) No possibility for p
  - Bel (not (p)) = 0 : No support for not(p) Full possibility for p
  - Rangenya [0,1]

- Contoh : Two persons M and B with reliabilities detect a computer and claim the result independently. How you believe their claims?

Question (Q): detection claim

Related question (RQ): detectors' reliability

DFT's approach is to obtain bel for Q from subjective (prior) probabilities for RQ for each person & combine bel from 2 people

- Person M:
  - reliability 0.9, unreliability 0.1
  - Claim h1
  - Belief degree of h1 is **bel(h1)=0.9**
  - Belief degree of not(h1) is **bel(not(h1))=0.0**, different from probability theory, since no evidence supporting not(h1)
  - **pl(h1) = 1 – bel(not(h1)) = 1-0 =1**
  - Thus belief measure for M claim h1 is [0.9, 1]

- Person B:
  - Reliability 0.8, unreliability 0.2
  - Claim h2
  - **bel(h2) =0.8**, **bel(not(h2))=0**
  - **pl(h2)=1-bel(not(h2))=1-0**
  - Belief measure for B claim h2 is [0.8,1]

- Set of propositions: M claim h1 and B claim h2
  - **Case 1**: h1 = h2
    - Reliability M and B: 0.9x0.8=0.72
    - Unreliability M and B: 0.1x0.2=0.02
    - The probability that at least one of two is reliable: 1-0.02=0.98
    - Belief measure for h1=h2 is [0.98,1]

- Set of propositions: M claim h1 and B claim h2
  - **Case 2**: h1 = not(h2)
    - Cannot be both correct and reliable
    - At least one is unreliable
      - Reliable M and unreliable B: 0.9x(1-0.8)=0.18
      - Reliable B and unreliable M: 0.8x(1-0.9)=0.08
      - Unreliable M and B: (1-0.9)x(1-0.8)=0.02
      - At least one is unreliable: 0.18+0.08+0.02=0.28
    - Given at least one is unreliable, posterior probabilities
      - Reliable M and unreliable B: 0.18/0.28=0.643
      - Reliable B and unreliable M: 0.08/0.28=0.286

- Belief measure for h1
  - Bel(h1)=0.643, bel(not(h1))=bel(h2)=0.286
  - Pl(h1)=1-bel(not(h1))=1-0.286=0.714
  - Belief measure: [0.643, 0.714]
- Belief measure for h2
  - Bel(h2)=0.286, bel(not(h2))=bel(h1)=0.643
  - Pl(h2)=1-bel(not(h2))=1-0.643=0.357
  - Belief measure: [0.286, 0.357]

Week 12 – Fuzzy Logic

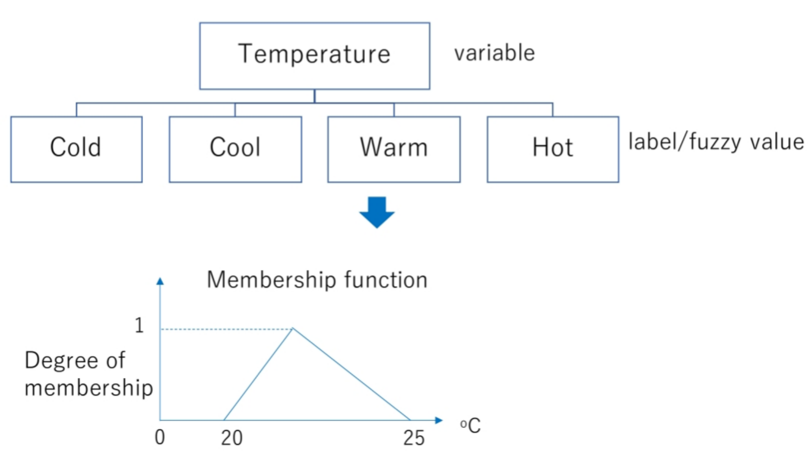
- Conventional logic : Yes or no
  True or false
  Good or bad
  {0,1}

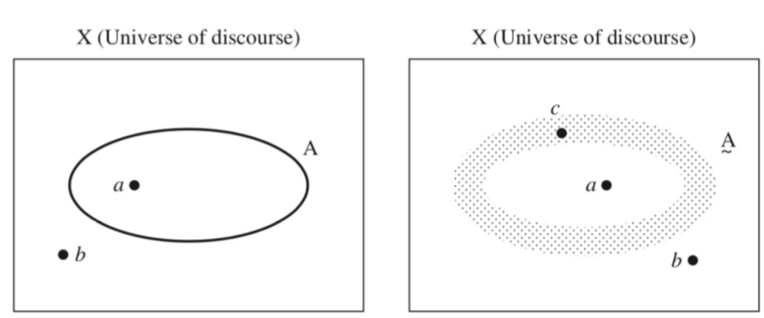
VS

- Fuzzy logic : Drive Slowly
  It's cold
  It's fast
  [0,1]
  Tidak tentu

- Crisp Sets : Classical sets, either an element belongs to set/not

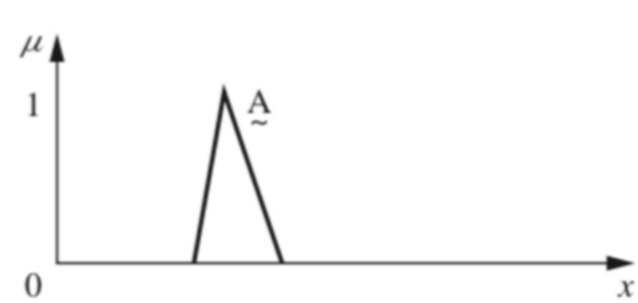
- Fuzzy Set Theory : Uses membership functions with values [0,1]
  ↳ Misalkan :





Crisp set boundary   Fuzzy set boundary

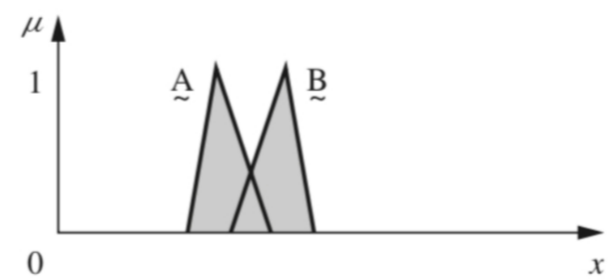
- Membership function for fuzzy set $\underset{\sim}{A}$:



- Operations on fuzzy set :

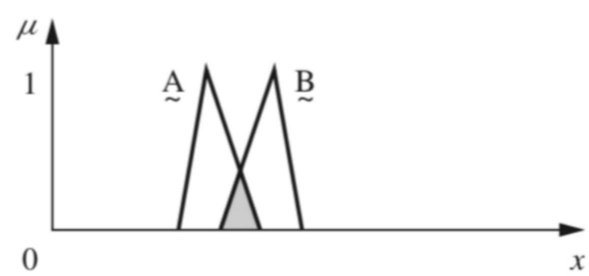
1) Union : Assume A & B are 2 fuzzy subsets of U

$\boxed{\mu_{A\cup B}(u) = \max\{\mu_A(u), \mu_B(u)\}}$

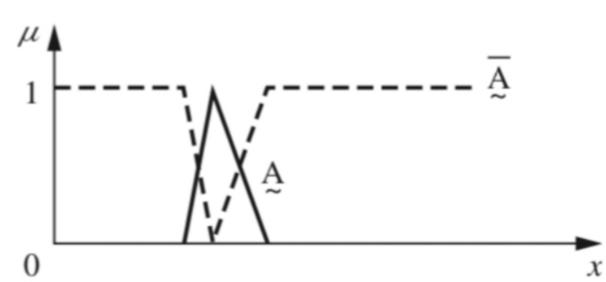


2) Intersection : Assume A & B are 2 fuzzy subsets of U

$\boxed{\mu_{A\cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}}$



3) Complement : $\boxed{\mu_{A^c}(u) = 1 - \mu_A(u)}$



4) Difference : $\boxed{A - B = A \cap B^c = \min(A, B^c)}$

- $\underset{\sim}{A} = \left\{ \frac{1}{2} + \frac{0.5}{3} + \frac{0.3}{4} + \frac{0.2}{5} \right\}$ and $\underset{\sim}{B} = \left\{ \frac{0.5}{2} + \frac{0.7}{3} + \frac{0.2}{4} + \frac{0.4}{5} \right\}$

↳ Complement :

Krn dpnnya $0 + \frac{1}{2} + \ldots$

$$\bar{\underset{\sim}{A}} = \left\{ \frac{1}{1} + \frac{0}{2} + \frac{0.5}{3} + \frac{0.7}{4} + \frac{0.8}{5} \right\}$$

+ adlh delimiter

$$\bar{\underset{\sim}{B}} = \left\{ \frac{1}{1} + \frac{0.5}{2} + \frac{0.3}{3} + \frac{0.8}{4} + \frac{0.6}{5} \right\}$$

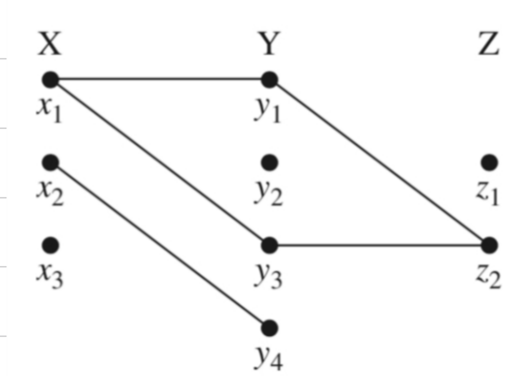
↳ Union : $\underset{\sim}{A} \cup \underset{\sim}{B} = \left\{ \frac{1}{2} + \frac{0.7}{3} + \frac{0.3}{4} + \frac{0.4}{5} \right\}$

↳ Intersection : $\underset{\sim}{A} \cap \underset{\sim}{B} = \left\{ \frac{0.5}{2} + \frac{0.5}{3} + \frac{0.2}{4} + \frac{0.2}{5} \right\}$

↳ Difference : $\underset{\sim}{A} | \underset{\sim}{B} = \underset{\sim}{A} \cap \bar{\underset{\sim}{B}} = \left\{ \frac{0.5}{2} + \frac{0.3}{3} + \frac{0.3}{4} + \frac{0.2}{5} \right\}$

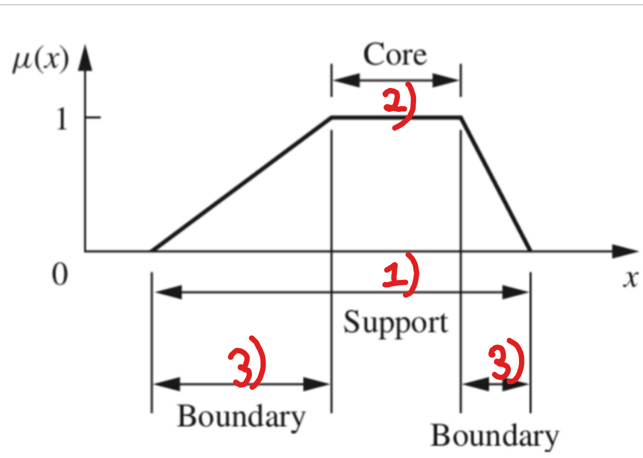
$$\underset{\sim}{B} | \underset{\sim}{A} = \underset{\sim}{B} \cap \bar{\underset{\sim}{A}} = \left\{ \frac{0}{2} + \frac{0.5}{3} + \frac{0.2}{4} + \frac{0.4}{5} \right\}$$

- Let R be relation that relates, or maps, elements from X to Y
  Let S be relation that relates, or maps, elements from Y to X

- 

$R = \{(x_1, y_1), (x_1, y_3), (x_2, y_4)\}$

$S = \{(y_1, z_2), (y_3, z_2)\}$

- 

1) Crisp subset of U which contains all elements of U that have a non-zero membership value in A.
2) Crisp subset of U which consists all elements with membership value 1
3) Elements x of the universe such that $0 < \mu_A(x) < 1$

- Max-min composition:

$$T = R \circ S$$
$$\chi_T(x,z) = \bigvee_{y \in Y} (\chi_R(x,y) \wedge \chi_S(y,z))$$

- Max-product composition:

$$T = R \circ S$$
$$\chi_T(x,z) = \bigvee_{y \in Y} (\chi_R(x,y) \bullet \chi_S(y,z))$$

- Relation matrices for R and S:

$$R = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad \text{and} \quad S = \begin{array}{c} \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{array} \begin{array}{c} \begin{array}{cc} z_1 & z_2 \end{array} \\ \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \end{array}$$

↳ $\mu_T(x_1, z_1) = \max[\min(1,0), \min(0,0), \min(1,0), \min(0,0)]$
$= \max(0, 0, 0, 0)$
$= 0$

↳ $\mu_T(x_1, z_2) = \max[\min(1,1), \min(0,0), \min(1,1), \min(0,0)]$
$= \max(1, 0, 1, 0)$
$= 1$

↳ $\mu_T(x_2, z_1) = \max[\min(0,0), \min(0,0), \min(0,0), \min(0,1)]$
$= \max(0, 0, 0, 0)$
$= 0$

↳ $\mu_T(x_2, z_2) = \max[\min(0,1), \min(0,0), \min(0,1), \min(1,0)]$
$= \max(0,0,0,0)$
$= 0$

↳ $\mu_T(x_3, z_1) = \max[\min(0,0), \min(0,0), \min(0,0), \min(0,0)]$
$= \max(0,0,0,0)$
$= 0$

↳ $\mu_T(x_3, z_2) = \max[\min(0,1), \min(0,0), \min(0,1), \min(0,0)]$
$= \max(0,0,0,0)$
$= 0$

$\hookrightarrow$ 80

$$T = \begin{array}{c|cc} & z_1 & z_2 \\ \hline x_1 & 0 & 1 \\ x_2 & 0 & 0 \\ x_3 & 0 & 0 \end{array}$$

- α - cut sets :

1) Weak : $A_\alpha = \{ u \mid \mu_A(u) \geq \alpha, \forall u \in U \}$

2) Strong : $A_\alpha' = \{ u \mid \mu_A(u) > \alpha, \forall u \in U \}$

- Example : $A = \frac{0.5}{a} + \frac{0.6}{b} + \frac{1}{c} + \frac{0.8}{d} + \frac{0.3}{e}$

$A_1 = \{c\}$ $\quad$ $A_{0.8} = \{c, d\}$

$A_{0.6} = \{b, c, d\}$ $\quad$ $A_{0.5} = \{a, b, c, d\}$

$A_{0.2} = \{a, b, c, d, e\}$ $\quad$ $A_{0.5}' = \{b, c, d\}$

$1A_1 \Rightarrow \{c\} \longrightarrow \{1 \cdot 0.5\} = \{0.5\}$

$0.8A_{0.8}' \Rightarrow \{d\} \rightarrow \{0.8 \cdot 0.8\} = \{0.64\}$

$0.6A_{0.5} \Rightarrow \{a, b, c, d\} \rightarrow \{0.6 \cdot 0.5, 0.6 \cdot 0.6, 1 \cdot 0.6, 0.8 \cdot 0.6\}$

$= \{0.3, 0.36, 0.6, 0.48\}$

$$R = \begin{bmatrix} 1 & 0.8 & 0 & 0.1 & 0.2 \\ 0.8 & 1 & 0.4 & 0 & 0.9 \\ 0 & 0.4 & 1 & 0 & 0 \\ 0.1 & 0 & 0 & 1 & 0.5 \\ 0.2 & 0.9 & 0 & 0.5 & 1 \end{bmatrix}$$

$$\lambda = 1, \; R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\lambda = 0.9,\ R_{0.9} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $\boxed{R_\lambda = \{(x,y) \mid \mu_{R(x,y)} \geq \lambda\} = \{1 \mid \mu_{R(x,y)} \geq \lambda \,;\, 0 \mid \mu_{R(x,y)} < \lambda\}}$

- Fuzzy Relations = Map elements of one universe, X, to those of another universe, Y, through the Cartesian product of 2 universes.

  Properties : $\underset{\sim}{R} \cup \overline{\underset{\sim}{R}} \neq E$ E = Complete relation

  $\underset{\sim}{R} \cap \overline{\underset{\sim}{R}} \neq O$ O = Null relation

- Suppose that we have 2 fuzzy sets :

  1) P that represents byk black pixel (Misalnya none with black pixel, C1, a few with black pixel, C2, and a lot of black pixel, C3)
  2) S that represents shape of black pixel clusters (Misalnya S1 ellipse & S2 circle)

$$P = \left\{ \frac{0.1}{C1} + \frac{0.5}{C2} + \frac{1.0}{C3} \right\} \quad \text{and} \quad S = \left\{ \frac{0.3}{S1} + \frac{0.8}{S2} \right\}$$

- Misalnya mau cari relationship between quantity of black pixels in virus and shape of black pixel clusters.

$$R = P \times S = \begin{array}{c} \\ C1 \\ C2 \\ C3 \end{array} \begin{array}{c} \begin{array}{cc} S1 & S2 \end{array} \\ \begin{bmatrix} 0.1 & 0.1 \\ 0.3 & 0.5 \\ 0.3 & 0.8 \end{bmatrix} \end{array}$$

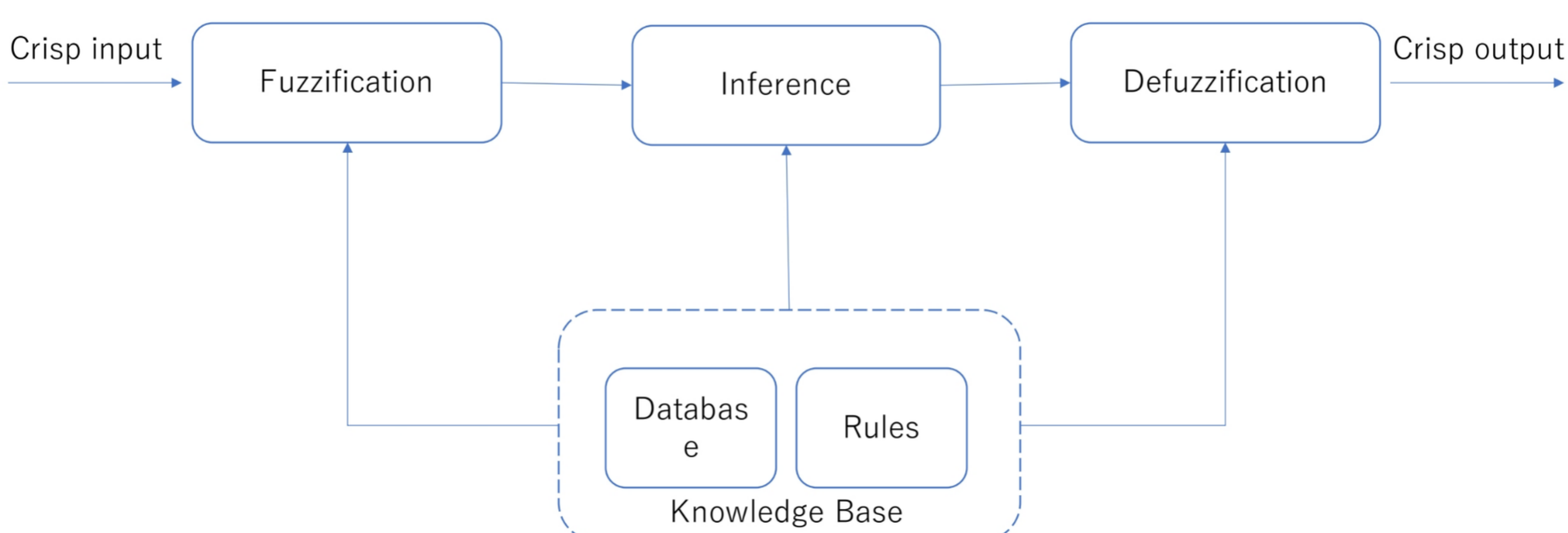
- Then, another microscope image is taken and number of black pixels is slightly different. Let new black pixel quantity be represented by fuzzy set P'.

$$P' = \left\{ \frac{0.4}{C1} + \frac{0.7}{C2} + \frac{1.0}{C3} \right\}$$

- Using max-min composition with relation R will yield a new value for fuzzy set of pixel cluster shapes that are associated with new black pixel quantity.

$$S' = P' \circ R = \begin{bmatrix} 0.4 & 0.7 & 1.0 \end{bmatrix} \circ \begin{bmatrix} 0.1 & 0.1 \\ 0.3 & 0.5 \\ 0.3 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.3 & 0.8 \end{bmatrix}$$

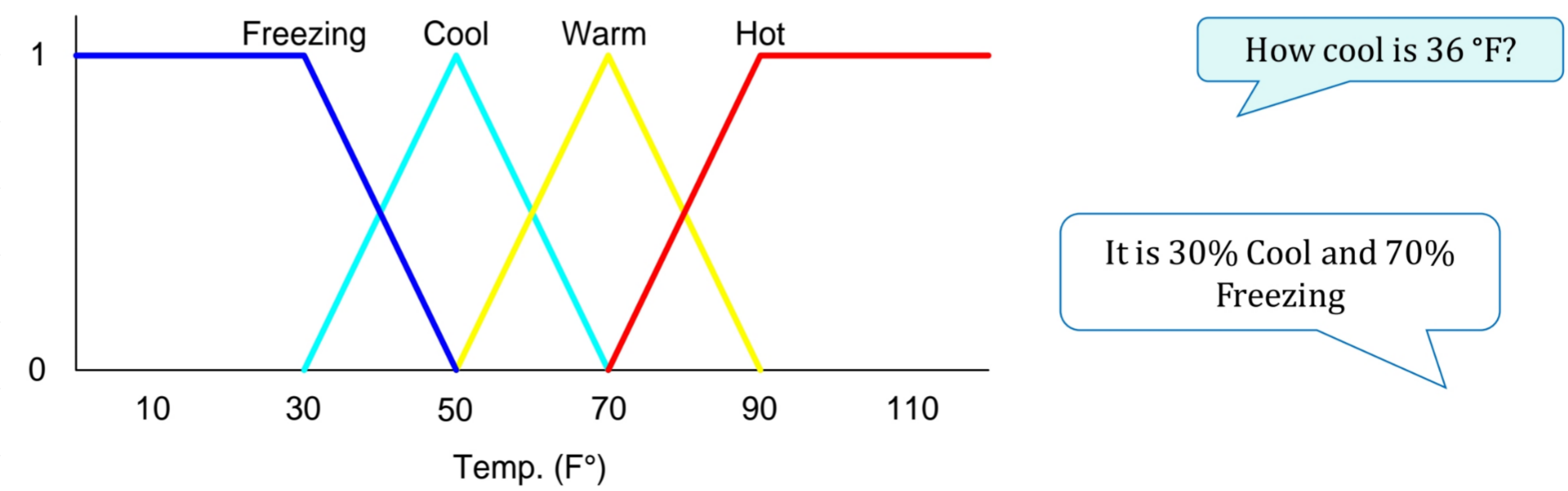
- Fuzzy inference systems = Sistem yg pakai fuzzy set theory to map inputs to outputs

- 

- **IF premise(*antecedent*), THEN conclusion (*consequent*)**

  - IF *temperature* is *freezing* THEN *heater* is *high.*
  - IF *temperature* is *cool* and *humidity* is *high* THEN *speed* is *medium.*

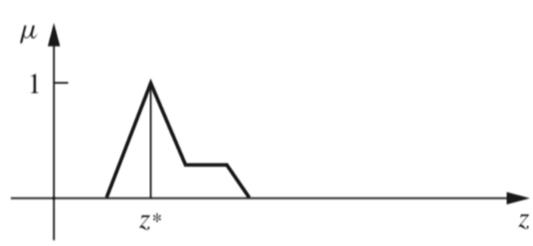
- Fuzzification = Proses of changing a real scalar value to fuzzy value using membership functions stored in knowledge base.



- Defuzzification = Process of changing fuzzy value to real scalar value
- Defuzzification terdiri atas beberapa macam yaitu :

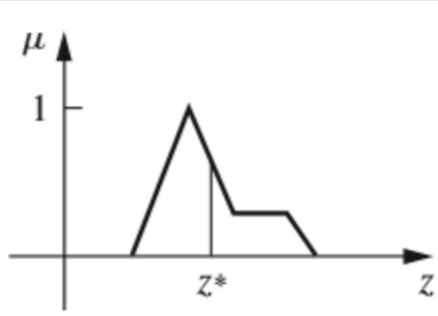
1) Max Membership Principle

$$\mu_{\underset{\sim}{C}}(z^*) \geq \mu_{\underset{\sim}{C}}(z) \qquad \text{for all } z \in Z$$



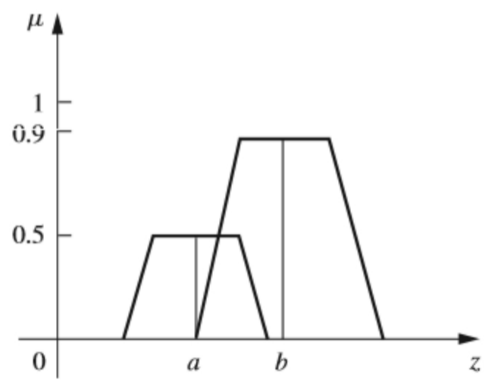
2) Centroid of Area (COA) / Center of Gravity (COG)

$$z^* = \frac{\int \mu_{\underset{\sim}{C}}(z) \cdot z \, dz}{\int \mu_{\underset{\sim}{C}}(z) \, dz}$$



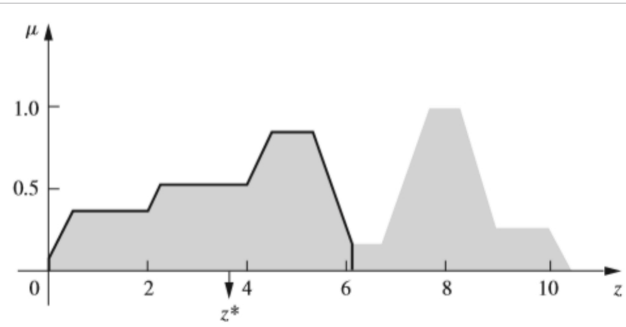
3) Weighted Average

$$z^* = \frac{\sum \mu_{\underset{\sim}{C}}(\bar{z}) \cdot \bar{z}}{\sum \mu_{\underset{\sim}{C}}(\bar{z})}$$



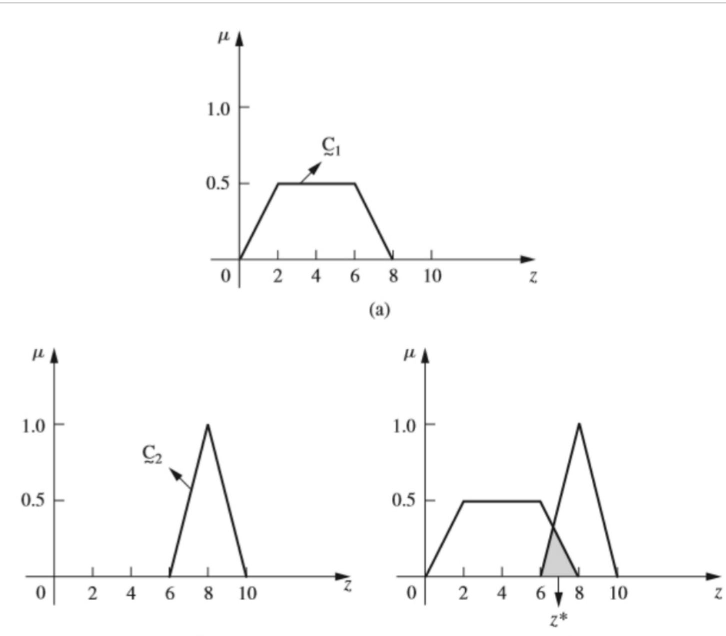
4) Center of Largest Area

$$z^* = \frac{\int \mu_{\underset{\sim}{C}_m}(z)\, z \, dz}{\int \mu_{\underset{\sim}{C}_m}(z)\, dz}$$



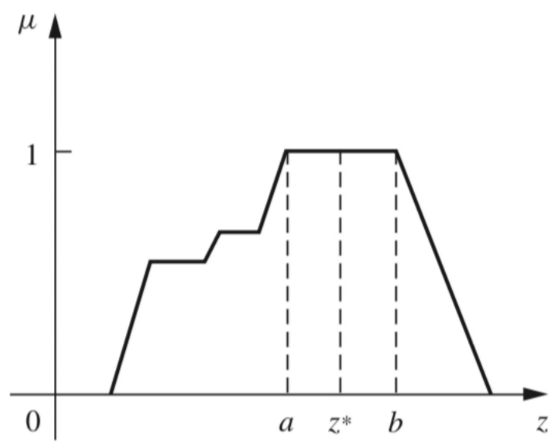
5) Center of Sums

$$z^* = \frac{\sum_{k=1}^{n} \mu_{\underset{\sim}{C}_k}(z) \int_z \bar{z}\, dz}{\sum_{k=1}^{n} \mu_{\underset{\sim}{C}_k}(z) \int_z}$$



6) Mean Max Membership

$$z^* = \frac{a+b}{2}$$



7) First of Max (and Last of Max)

- First, the largest height in the union (Denoted hgt(C_k)) is determined

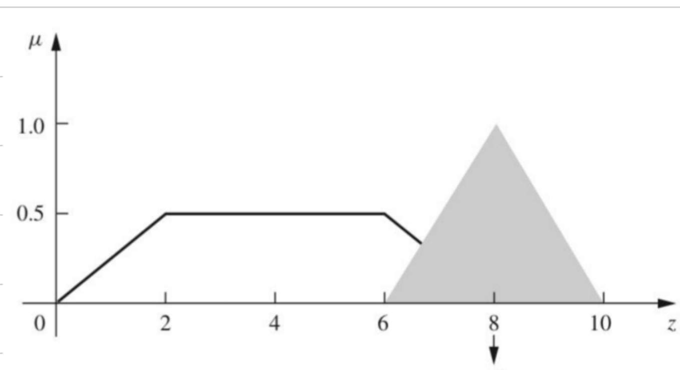
$$hgt(\underset{\sim}{C_k}) = \sup_{z \in Z} \mu_{\underset{\sim}{C_k}}(z)$$

- Then, first of the maxima is found

$$z^* = \inf_{z \in Z} \{ z \in Z \mid \mu_{\underset{\sim}{C_k}}(z) = hgt(C_k) \}$$

- An alternative to this method is called the last of max ima, and given as

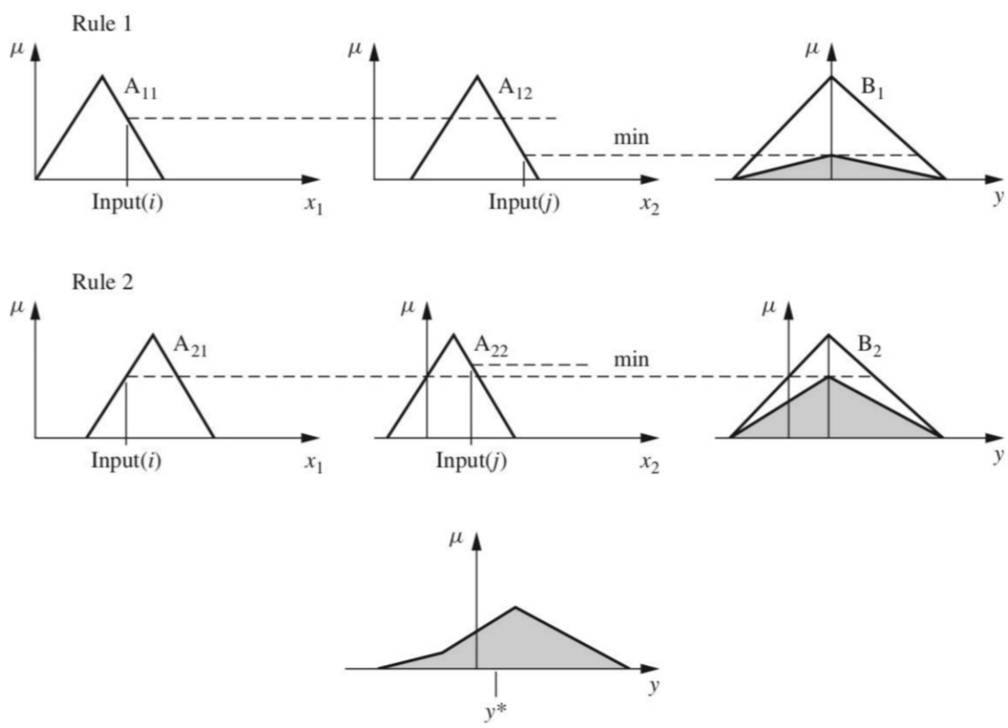
$$z^* = \sup_{z \in Z} \{ z \in Z \mid \mu_{\underset{\sim}{C_k}}(z) = hgt(C_k) \}$$



- FIS (MAMDANI)

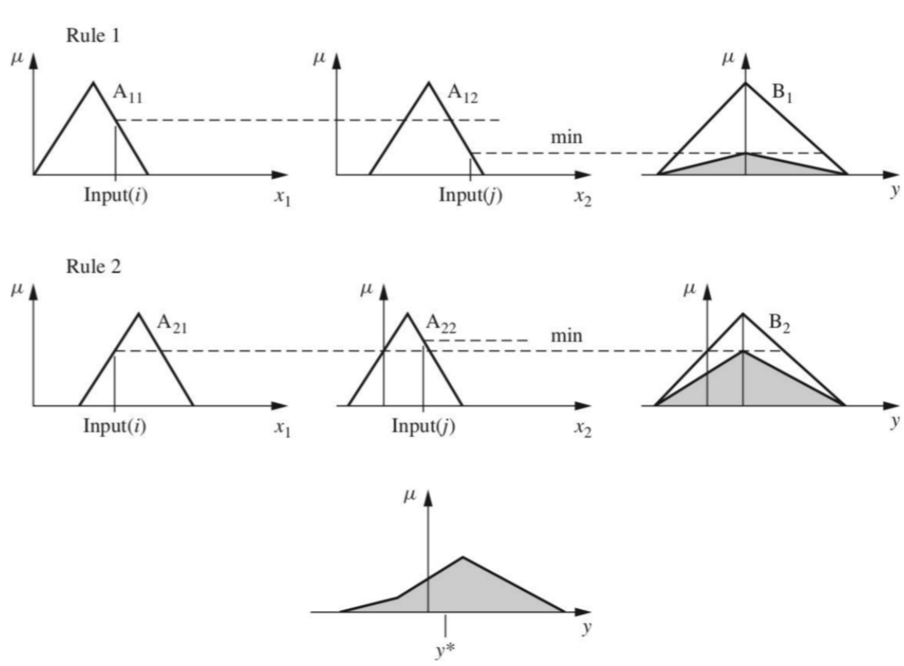
1. Inference :

i) Max - Min (Zadeh)

- 

- Fuzzy "and" $\rightarrow \mu_{A\cap B} = \min(\mu_A(x), \mu_B(x))$
- Fuzzy "or" $\rightarrow \mu_{A\cup B} = \max(\mu_A(x), \mu_B(x))$

ii) Product

- 

- Fuzzy "and" $\rightarrow \mu_{A\cap B} = \mu_A(x) \times \mu_B(x)$
- Fuzzy "or" $\rightarrow \mu_{A\cup B} = \mu_A(x) + \mu_B(x) - (\mu_A(x) \times \mu_B(x))$

O

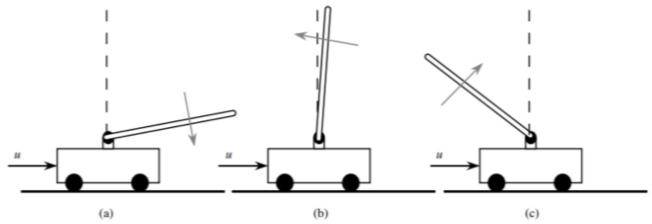
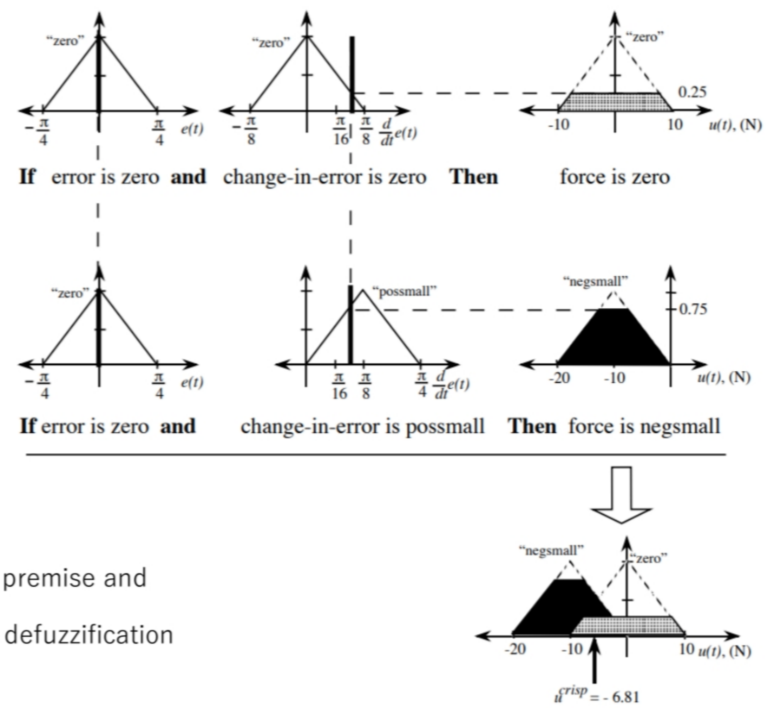


FIGURE 2.5 Inverted pendulum in various positions.



Fuzzy controller for an inverted pendulum

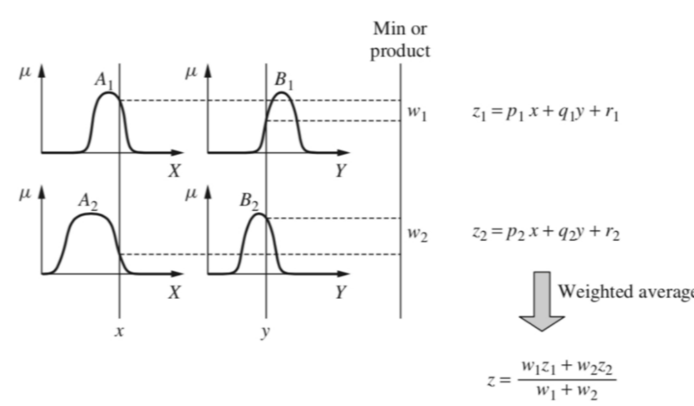
- the minimum operator to represent the "and" in the premise and
- The maximum operator for the implication and COG defuzzification

O FIS (SUGENO)

IF *x* is *A* and *y* is *B* THEN *z* is $\boxed{f(x,y)}$ → Crisp function

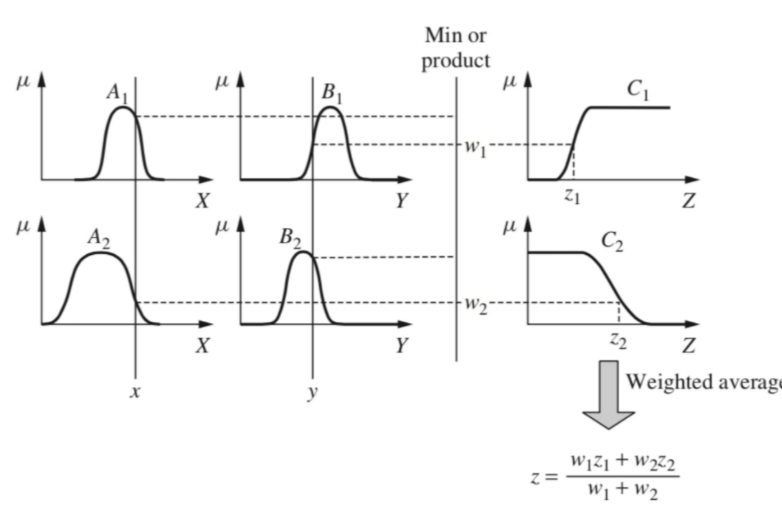
Example 5.16. An example of a two-input, single-output Sugeno model with four rules is repeated from Jang *et al.* (1997):

IF X is small and Y is small, THEN $z = -x + y + 1$.
IF X is small and Y is large, THEN $z = -y + 3$.
IF X is large and Y is small, THEN $z = -x + 3$.
IF X is large and Y is large, THEN $z = x + y + 2$.



O FIS (TSUKAMOTO)

IF x is A_1 and y is B_1 THEN z is C_1
IF x is A_2 and y is B_2 THEN z is C_2



Week 13 - Artificial Neural Network

- AI = The field of CS & engineering concerned with design & development of algorithms & systems that can exhibit human-like intelligence and behavior, including ability to perceive, reason, learn, adapt, and act autonomously

- ML = Subset of AI that involves use of statistical algorithms and models to enable computer systems to learn from and make predictions / decisions based on data, without being explicitly programmed to do so.

  ↳ Supervised Learning : Training model on labeled data, where the correct outputs are provided along with input data

  ↳ Unsupervised learning : Finding patterns in unlabeled data, without any specific guidance / feedback

  ↳ Reinforcement Learning : Training model to make decisions based on feedback from its environment, with goal of maximizing a reward signal

- DL = - Convolutional Neural Networks (CNN)
  - Recurrent Neural Networks (RNN)
  - Long-Short-Term-Memory Networks (LSTM)
  - Generative Adversarial Networks (GANs)
  - Auto-encoder Neural Networks

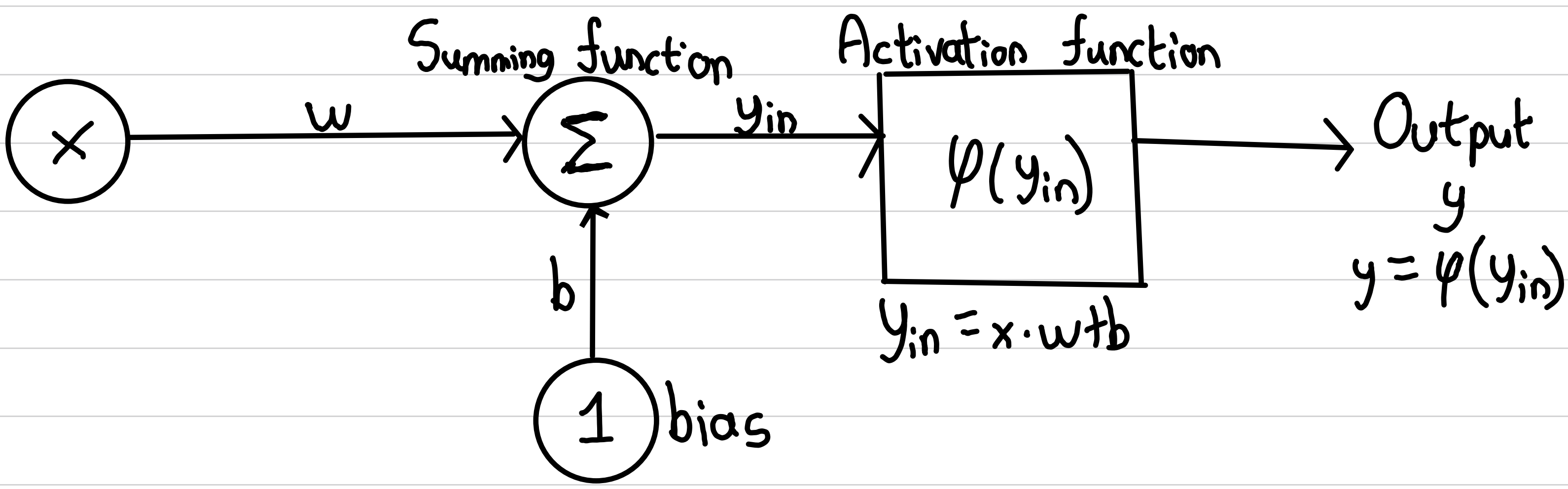
- Classification = Output is categorical. Mapping input to output labels

- Regression = Output is numerical. Mapping input to continuous output

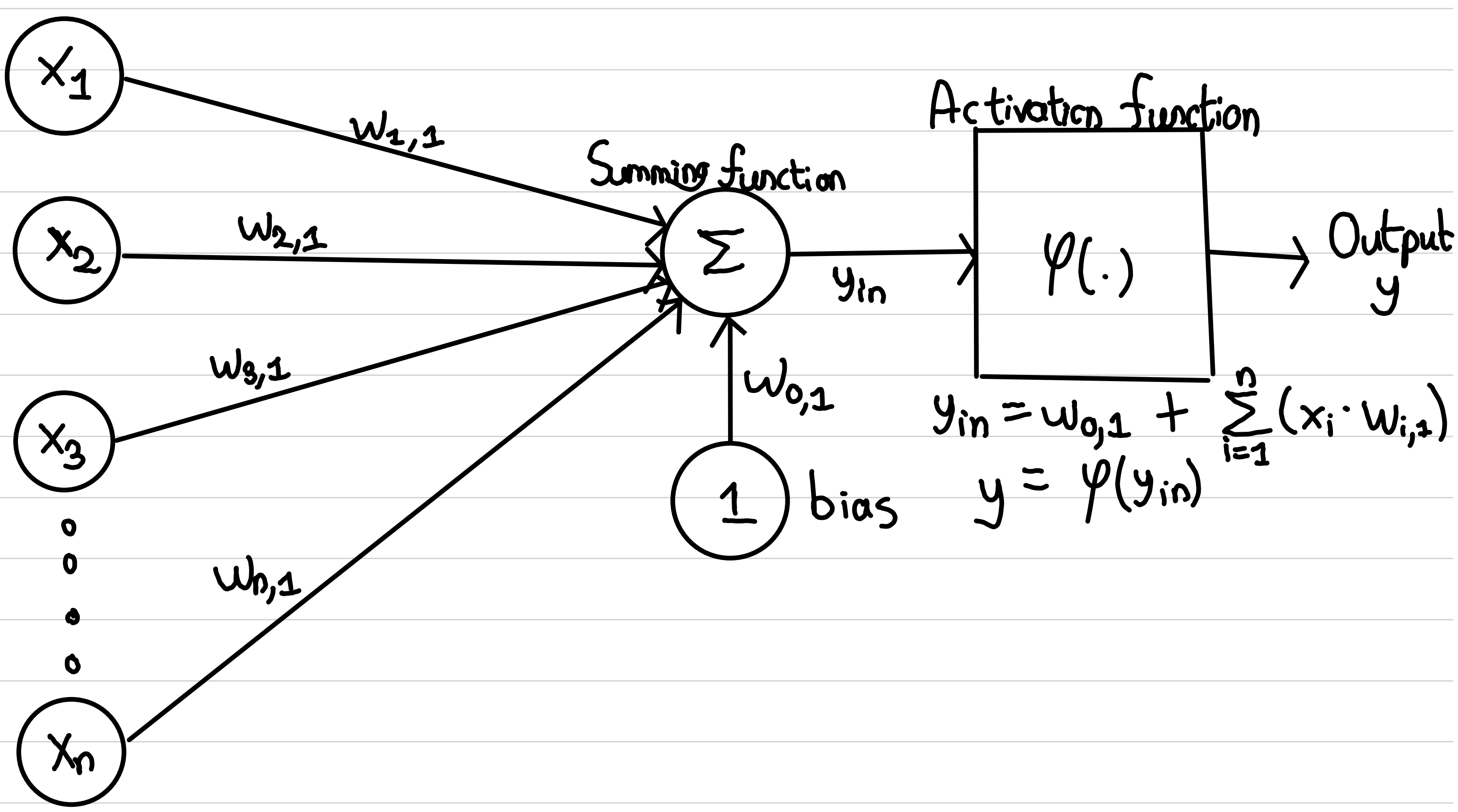
- Algorithms = Logistic regression, Naive Bayes, Support Vector Machine, Artificial neural networks, random forest

- Neuron Model :

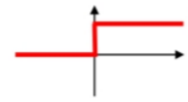
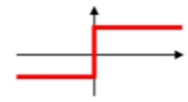
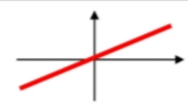
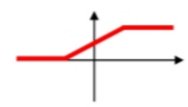
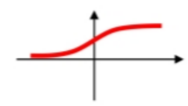
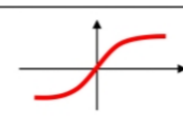
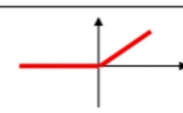
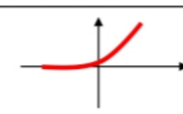
1. Single - Input Newton



2. Multi - Input Newton



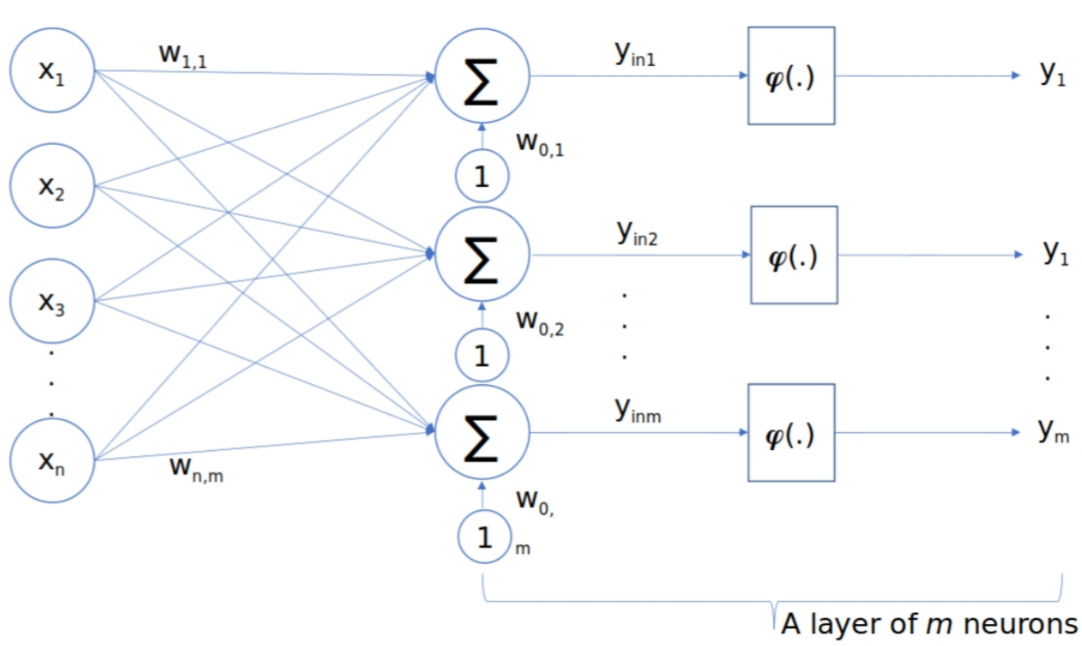
- Activation function digunakan untuk map input between required values like (0,1) or (-1,1)

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z)=\begin{cases}0, & z<0,\\ 0.5, & z=0,\\ 1, & z>0,\end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z)=\begin{cases}-1, & z<0,\\ 0, & z=0,\\ 1, & z>0,\end{cases}$	Perceptron variant	
Linear	$\phi(z)=z$	Adaline, linear regression	
Piece-wise linear	$\phi(z)=\begin{cases}1, & z\geq \frac{1}{2},\\ z+\frac{1}{2}, & -\frac{1}{2}<z<\frac{1}{2},\\ 0, & z\leq -\frac{1}{2},\end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z)=\frac{1}{1+e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z)=\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z)=max(0,z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z)=\ln(1+e^{z})$	Multi-layer Neural Networks	

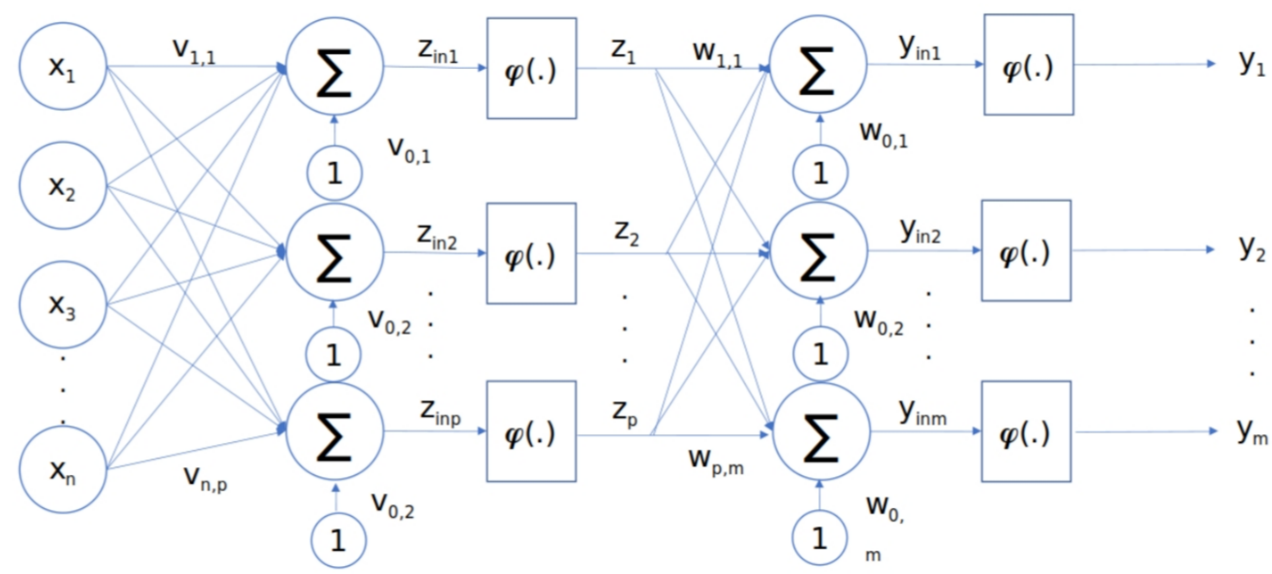
Neural - Network / NN Architecture :

- Layers = Input, hidden, output
- A pattern of connections between neurons :

1) Single - layer NN

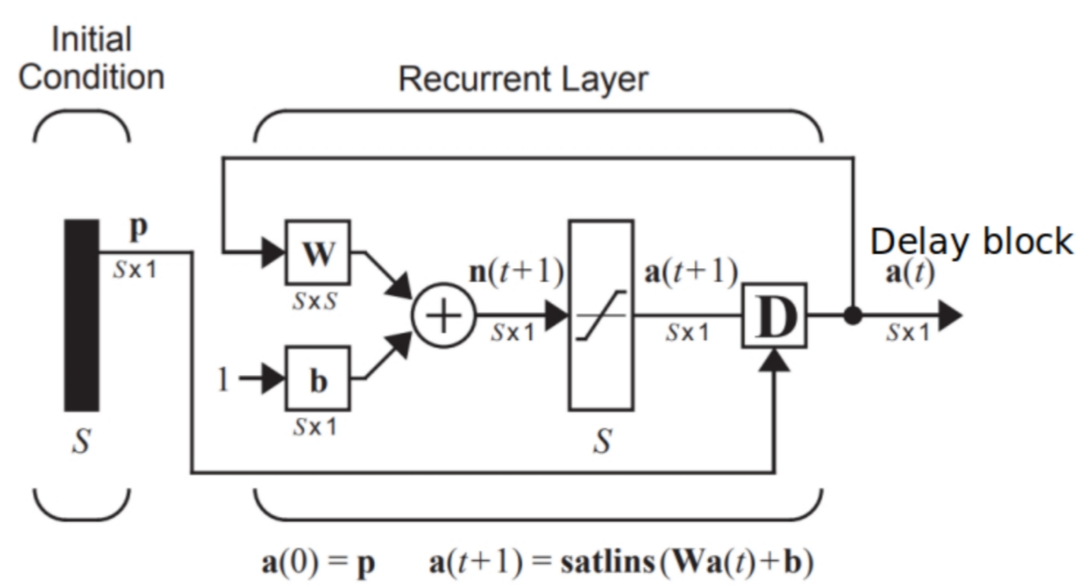


2) Multi-Layer Feedforward NN



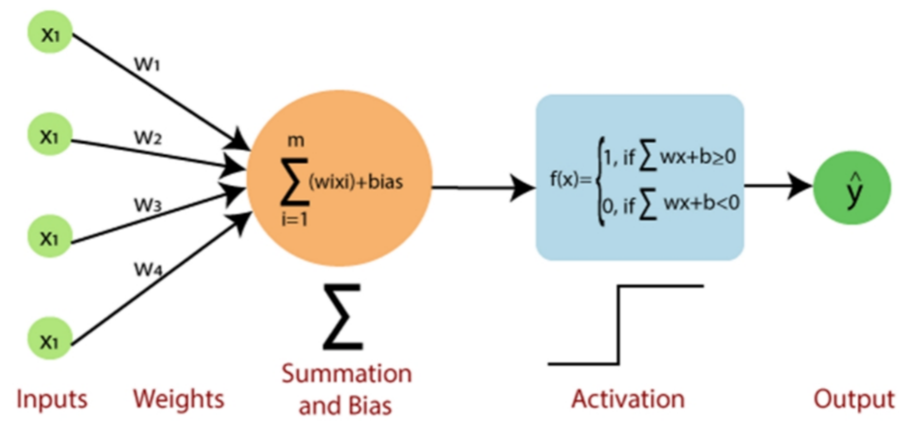
3) Recurrent NN

↳ Allow previous outputs to be used as inputs while having hidden states



- How to pick an architecture?

1) Number of network inputs = Number of problem inputs
2) Number of neurons in output layer = Number of problem outputs
3) Output layer transfer function choice at least partly determined by problem spesification of the outputs



source: `https://www.nomidl.com/deep-learning/difference-between-perceptron-and-neuron/`

- It was invented by Frank Rosenblatt in 1957.
- It is a single layer neural network.
- It is a linear binary classifier. The perceptron algorithm takes in an input vector and outputs a single binary value, representing the classification of that input.
- It can represent Boolean functions such as AND, OR, NOT but not the XOR function.
- It produces a linear separator in the input space.

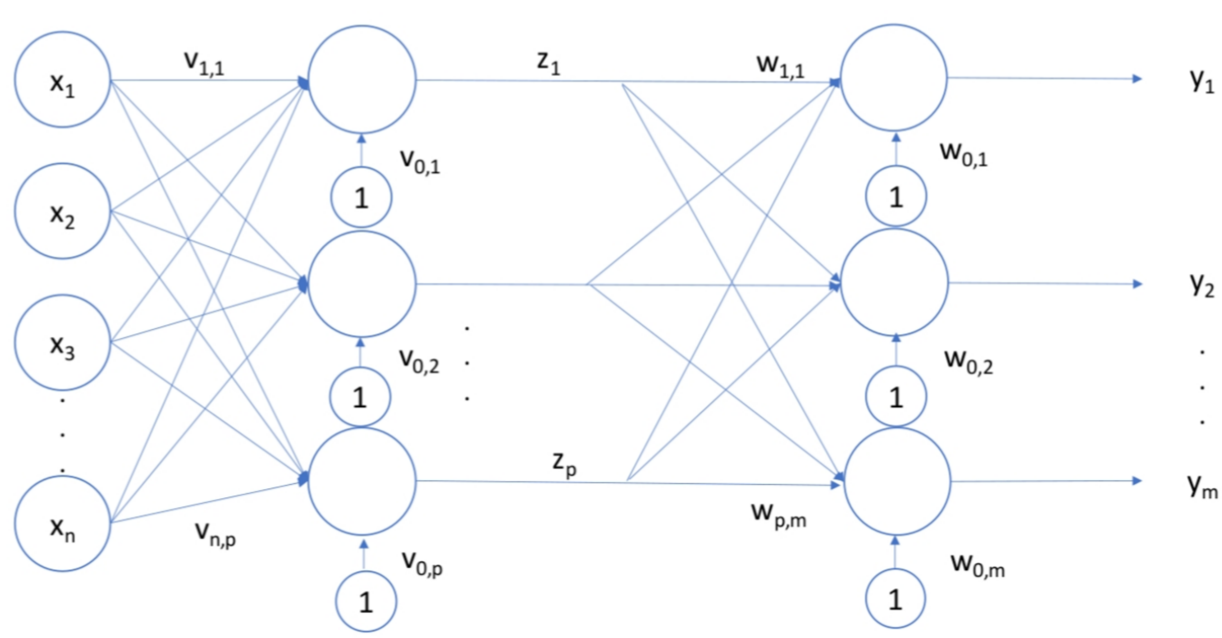
Steps :

1. Weight and bias initialisation at random
2. Learning rate $\alpha (0 < \alpha \leq 1)$. For simplicity set $\alpha = 1$.
3. Threshold θ of activation function.
4. For each input-output pair, do:
   - (a) Output unit summing function, $y_{ink} = w_{0,k} + \sum_{i=1}^{n} x_i w_{i,k}$
   - (b) Output unit, activation function
   - (c) Update weight and bias

$$w_{i,k}^{new} = w_{i,k}^{old} + \alpha \times (t_k - y_k) \times x_i$$
$$w_{0,k}^{new} = w_{0,k}^{old} + \alpha \times (t_k - y_k)$$

5. Do the iteration until each pattern has output equals/close to the target.

Backpropagation Algorithm I :



Backpropagation Algorithm II :

- Applicable to multi-layer NN.
- A popular learning method [1] capable of handling such large learning problems.
- The backpropagation algorithm looks for the minimum of the error function in weight space using the method of gradient descent.
- The algorithm involves two phases:
  1. In the **forward propagation phase**, the input data is passed through the neural network, and the output is calculated based on the current weights and biases. The output is then compared to the true output, and the error between them is calculated.
  2. In the **backward propagation phase**, the error is propagated back through the neural network, layer by layer, in order to calculate the gradient of the error with respect to the weights and biases.
- One of the more popular activation functions for backpropagation networks is the sigmoid.
  Logistic (sigmoid)

$$\phi(z) = \frac{1}{1+e^{-z}}$$

Backpropagation Algorithm III :

1. Initialise weight to small random values.
2. While stopping condition is false, do steps 3-11.
3. For each training pair (input-output) do steps 4-10.
4. Each input unit receives the input signal $x_i(i = 1, 2, 3, \ldots, n)$ and transmits this signal all units in the above i.e hidden layer.
5. Each hidden unit $\left(z_j, j = 1, \cdots, p\right)$ sums its input signals.

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

Applying activation function

$$z_j = \varphi\left(z_{inj}\right)$$

and sends this signal to all units in the layer above i.e output units.

Backpropagation Algorithm IV :

6. Each output unit $(y_k, k = 1, \cdots, m)$ sums its weighted input signals.

$$y_{\text{ink}} = w_{0,k} + \sum_{j=1}^{p} z_j w_{j,k}$$

and supplies its activation function to calculate the output signals.

$$y_k = \varphi\left(y_{\text{ink}}\right)$$

7. Calculating the total error

$$\text{error}_{\text{ total}} = \sum \frac{1}{2}\left(t_k - yk\right)^2$$

If total error $\leq$ target error then stop, else update weights and biases.

Backpropagation Algorithm V :

8 Each output unit ($y_k, k = 1, \cdots, m$) receives a target pattern corresponding to an input pattern, error information term is calculated as

$$\delta_k = (t_k - y_k)\, y_k\, (1 - y_k)$$

Each output unit ($y_k, k = 1, \cdots, m$) computes its bias and weights adjustments

$$(j = 0, \cdots, n).$$
$$\Delta w_{j,k} = \alpha \delta_k z_j$$
$$\Delta w_{0,k} = \alpha \delta_k$$

9 Each hidden unit $(z_j, j = 1, \cdots, p)$ sums its delta input from units in the layer above.

$$\delta_{inj} = \sum_{k=1}^{m} w_{j,k} \delta_k$$

the error information term is calculated as

$$\delta_j = \delta_{inj} z_j\, (1 - z_j)$$

Backpropagation Algorithm VI :

Each hidden unit $(z_j, j = 1, \cdots, p)$ computes its bias and weights adjustments ($i = 0, \cdots$, n).

$$\Delta v_{i,j} = \alpha \delta_j x_i$$
$$\Delta v_{0,j} = \alpha \delta_j$$

10 Each output unit ($y_k, k = 1, \cdots, m$) updates its bias and weights ($j = 0, \cdots, p$).

$$w_{j,k}^{new} = w_{j,k}^{old} + \Delta w_{j,k}$$

Each hidden unit $(z_j, j = 1, \cdots, p)$ updates its bias and weights ($i = 0, \cdots, n$).

$$v_{i,j}^{new} = v_{i,j}^{old} + \Delta v_{i,j}$$

11 Test the stopping condition, may be the number of epochs, etc.

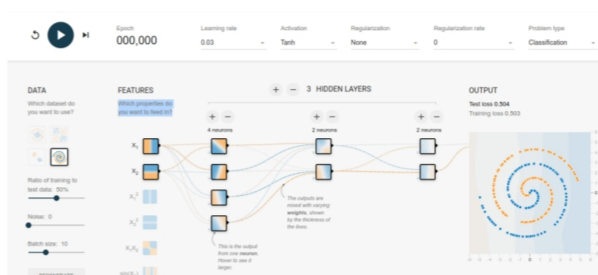
Choosing the Activation Functions :

- Match your activation function for your output layer based on the type of prediction problem that you are solving—specifically, the type of predicted variable.
  - **Regression** - Linear Activation Function
  - **Binary Classification** — Sigmoid/Logistic Activation Function
  - **Multiclass Classification** — Softmax
  - **Multilabel Classification** — Sigmoid
- As a rule of thumb, you can begin with using the ReLU activation function and then move over to other activation functions if ReLU doesn't provide optimum results.
  - ReLU activation function should only be used in the hidden layers.
  - Sigmoid/Logistic and Tanh functions should not be used in hidden layers as they make the model more susceptible to problems during training (due to vanishing gradients).
- The activation function used in hidden layers is typically chosen based on the type of neural network architecture
  - Convolutional Neural Network (CNN): ReLU activation function.
  - Recurrent Neural Network: Tanh and/or Sigmoid activation function.

Applications of NN :

- **Classification:** the aim is to predict the class of an input vector.
- **Pattern matching:** the aim is to produce a pattern best associated with a given input vector.
- **Pattern completion:** the aim is to complete the missing parts of a given input vector.
- **Optimisation:** the aim is to find the optimal values of parameters in an optimisation problem.
- **Control:** an appropriate action is suggested based on given an input vectors.
- **Function approximation/times series modelling:** the aim is to learn the functional relationships between input and desired output vectors.
- **Data mining:** with the aim of discovering hidden patterns from data (knowledge discovery).

NN Playground :



- Epoch
- Learning rate
- Activation function: Tanh, Sigmoid, ReLU or linear?
- Regularisation: L1 or L2?
- Regularisation rate?
- Problem types: classification or regression?
- Data: Which dataset is linearly separable?
- Features: Which properties do you want to feed in? Why?
- Select the number of hidden layers.
- Select the number of neurons/nodes in each layer.