News Classification using RNN/LSTM

Moch. Ari Nasichuddin, S.Kom., M.Eng.

Perkenalan

Riwayat Studi:

- T. Informatika UII (S.Kom)
- Magister Teknologi Informasi UGM (M.Eng)

Riwayat Pekerjaan:

- Full-stack Web Programmer (Geek Garden)
- Full-stack Web Programmer (Wonderlabs)
- ML Engineer/Web Core Programmer (Atmatech)

Outline

- Problem: Why News Classification?
- Machine Learning and Text Classification
- Method: RNN/LSTM
- Demo

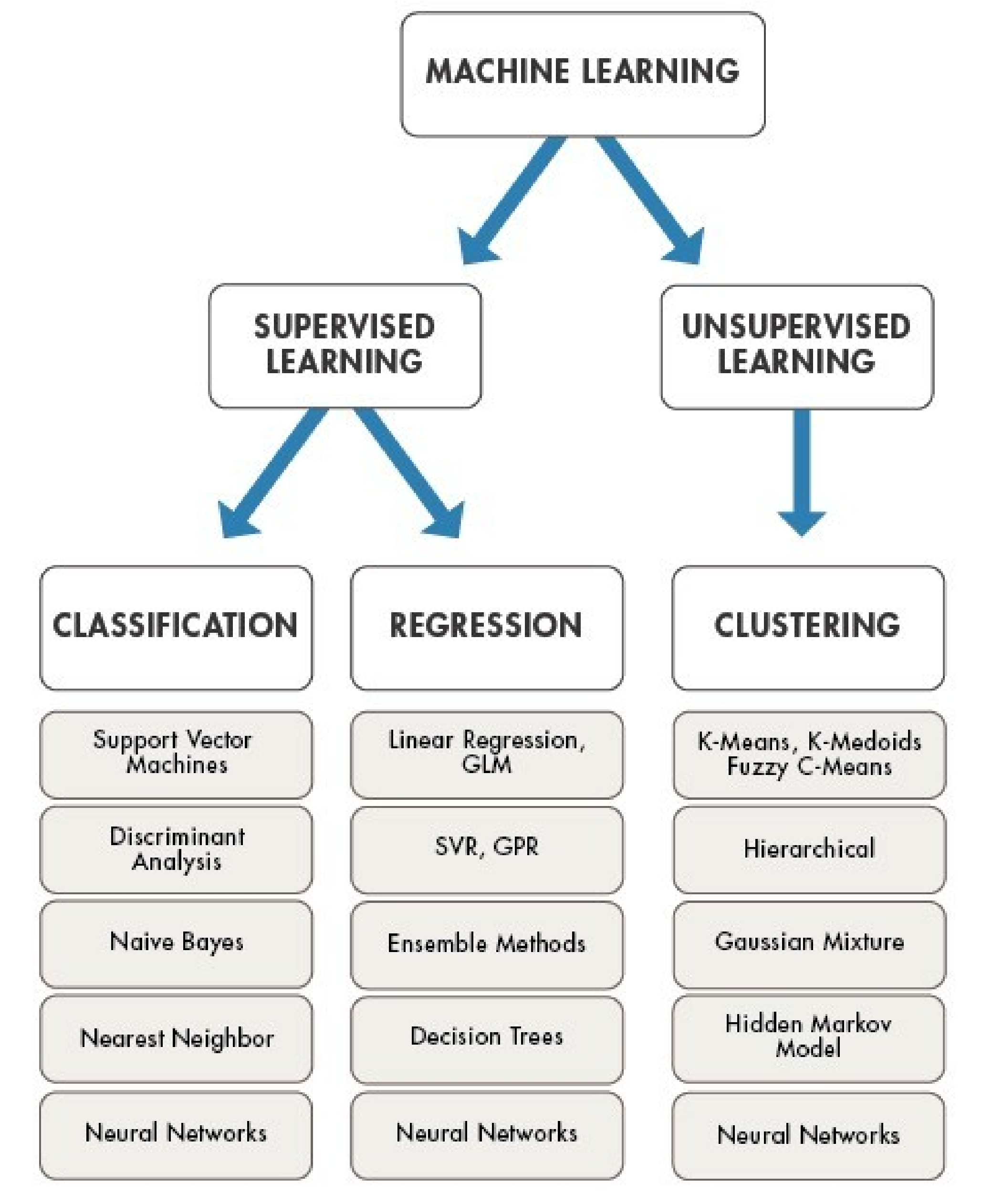
Prerequisite

- Dasar Machine Learning
- Macam-macam metode Machine Learning
- Neural Networks (Fundamental)

Problem: Why News Classification?

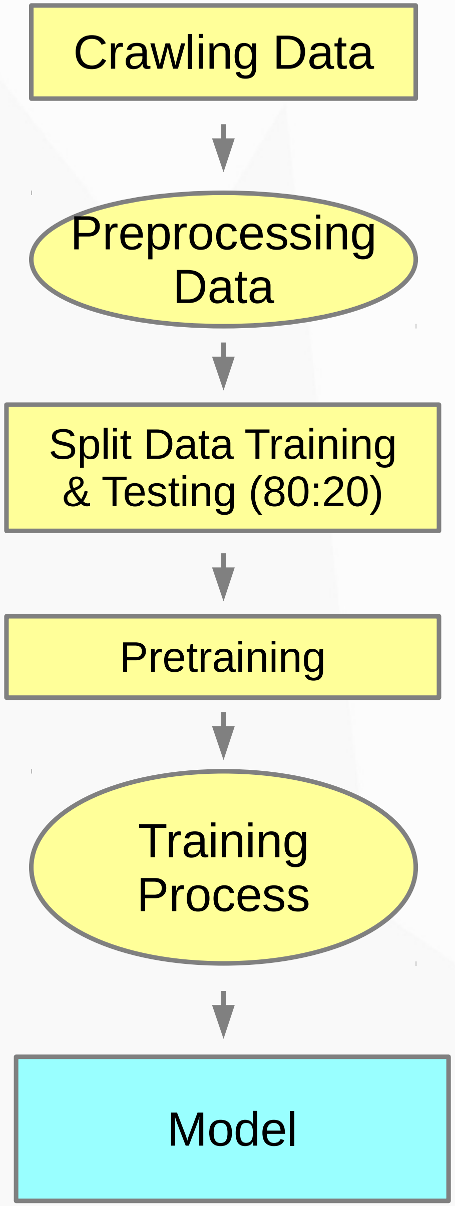
- Pertumbuhan media online sangat tinggi, semua orang dapat membuat medianya sendiri.
- Banyak konten media berbentuk straight news (berita pendek), feature (berita panjang), dan opini yang berserakan di dunia maya.
- Untuk itu diperlukan beberapa analisis agar dapat mengekstrak pengetahuan dari konten-konten media di atas
- Salah satu bentuk analisis yang dapat dipakai yakni Klasifikasi berbentuk Analisis Sentimen.

Machine Learning & Text Classification



Source:
https://goo.gl/qNZdyU

Machine Learning & Text Classification (2)



Crawling Data

Preprocessing Data

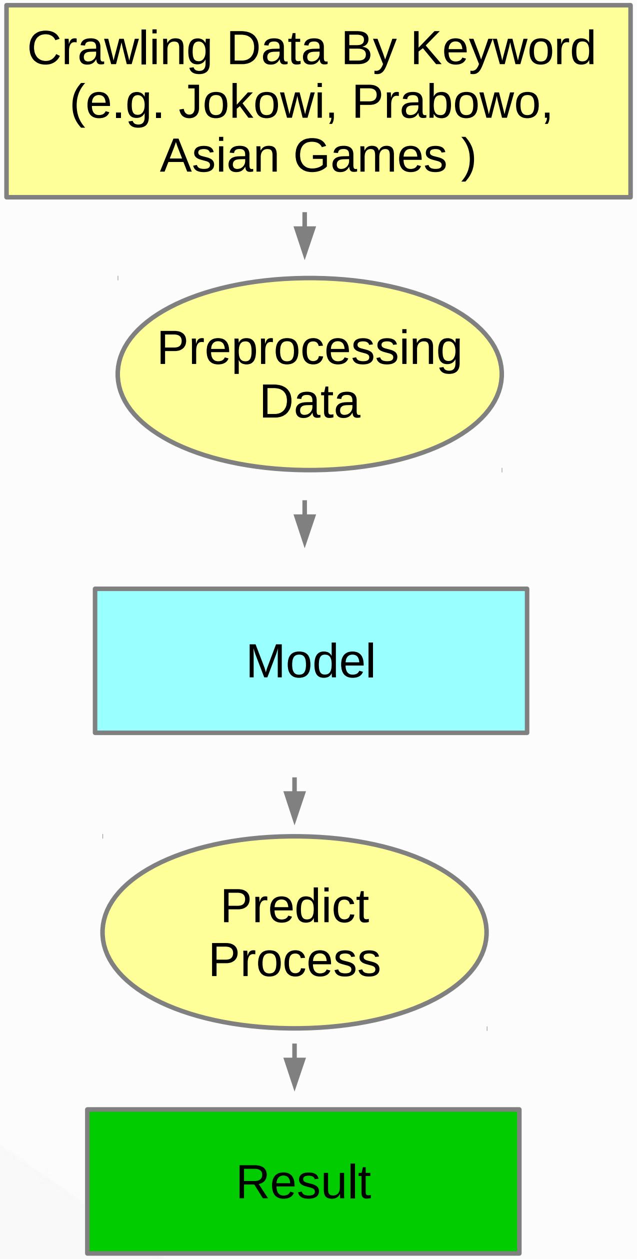
Split Data Training & Testing (80:20)

Pretraining

Training Process

Model

Proses Training



Crawling Data By Keyword (e.g. Jokowi, Prabowo, Asian Games)

Preprocessing Data

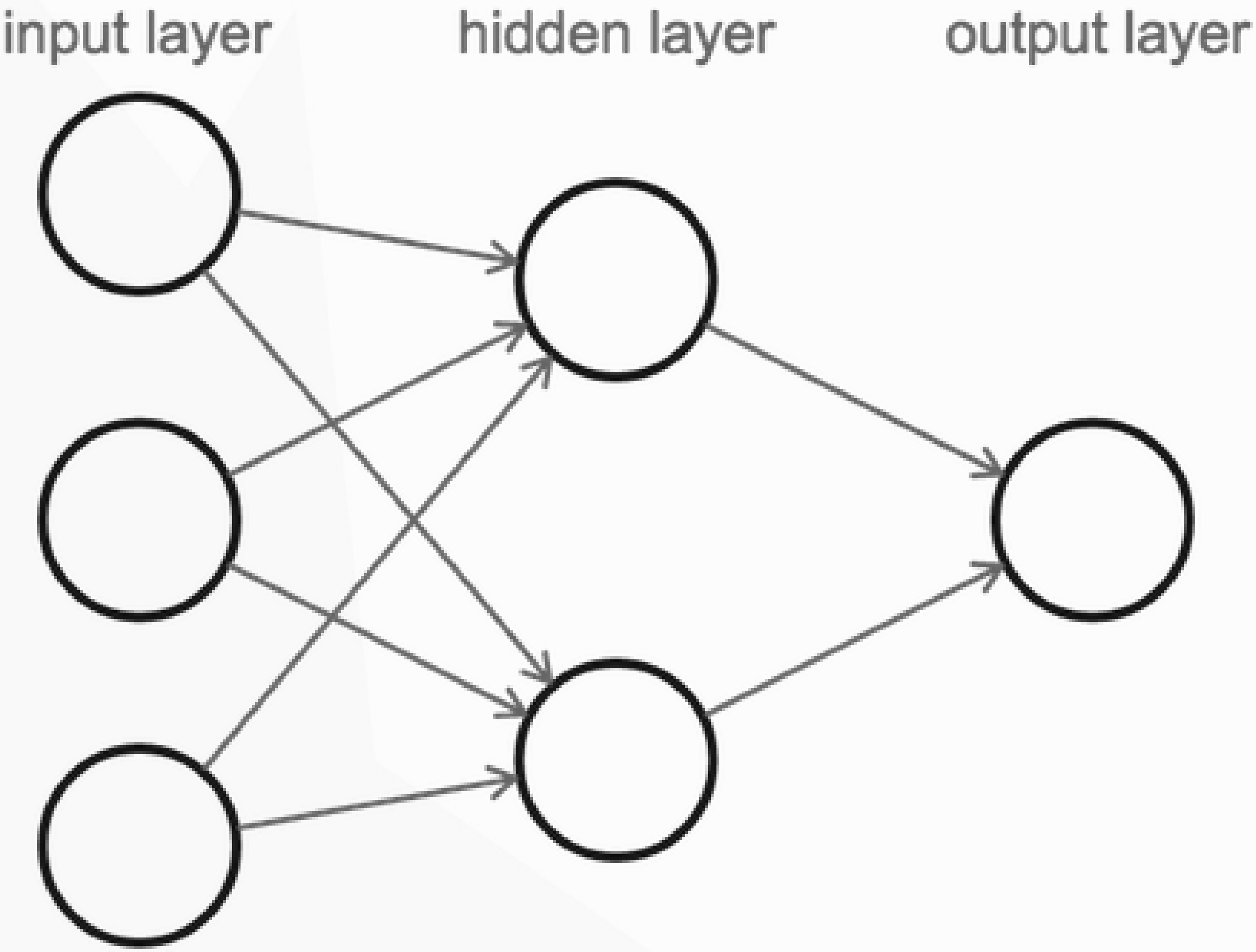
Model

Predict Process

Result

Proses Prediksi

Method: RNN/LSTM – Neural Networks

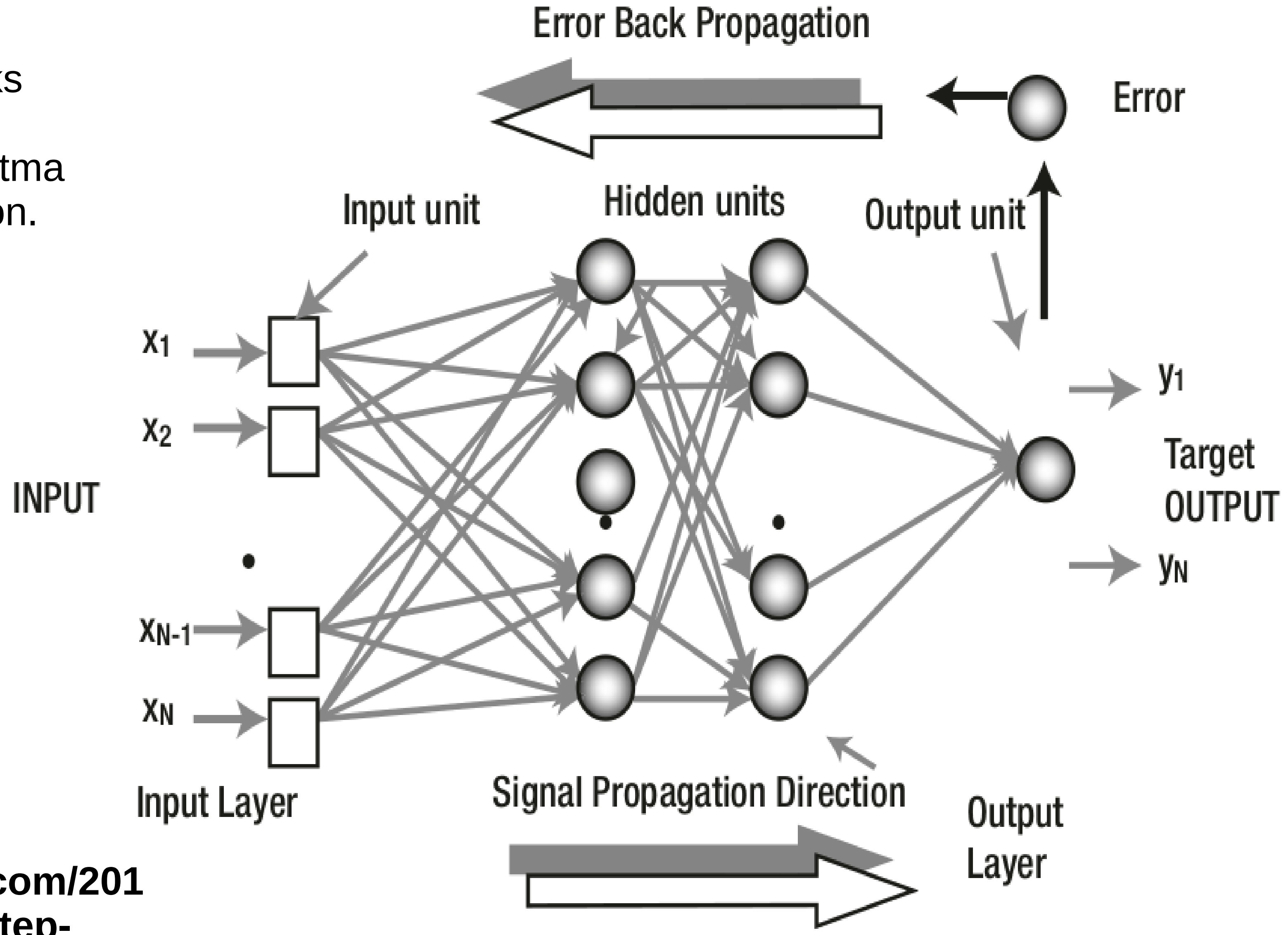


Arsitektur Neural Networks

Source:
https://goo.gl/5QCayt

Method: RNN/LSTM – Neural Networks (2)

- Neural Networks dijalankan memakai algoritma Backpropagation.



Mekanisme *training* Neural Networks

Source: Deep Learning for NLP

Bahan Latihan: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Method: RNN/LSTM – Deep Learning (Apa itu?)

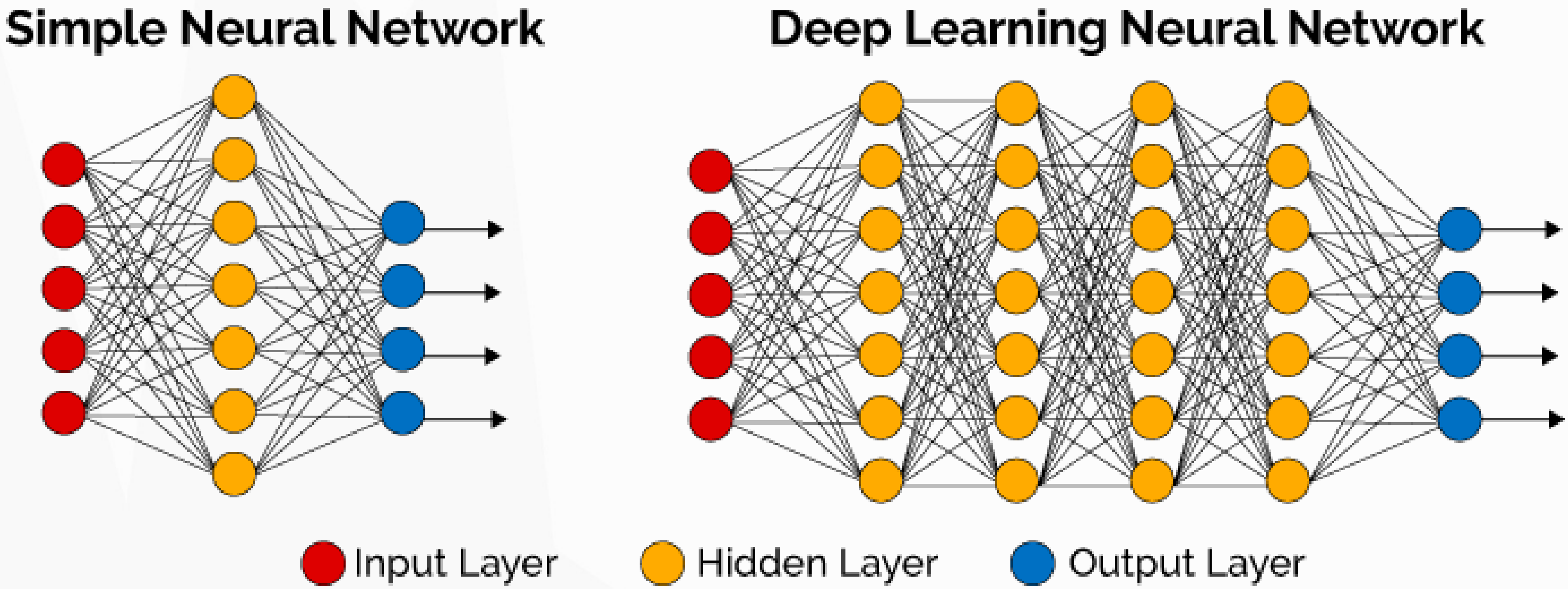
“Deep Learning is Large Neural Networks“

(Andrew Ng Chief Scientist at Baidu Research)

“When you hear the term deep learning, just think of a large deep neural net.
Deep refers to the number of layers typically and
so this kind of the popular term that’s been adopted in the press.
I think of them as deep neural networks generally.”

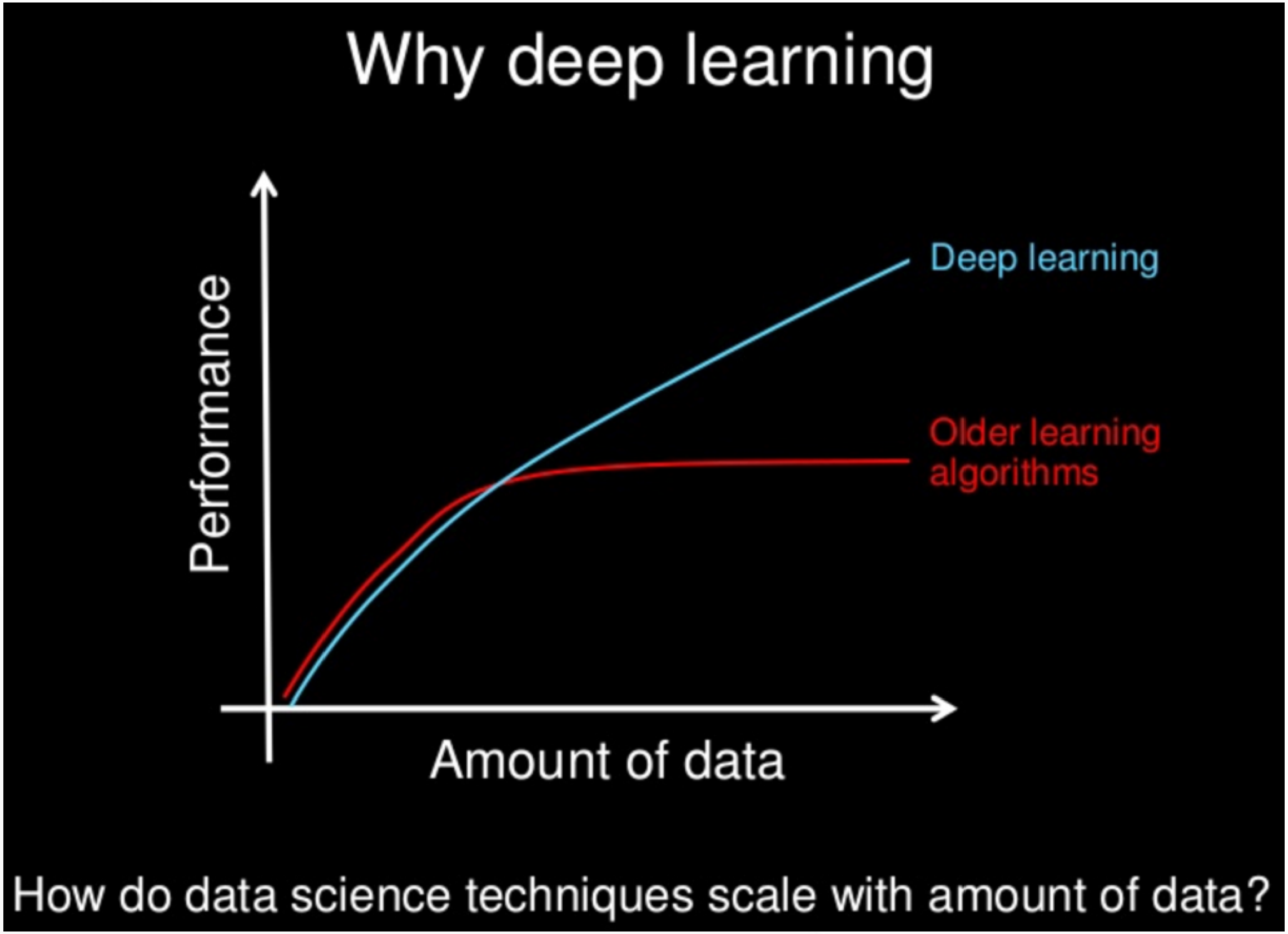
(Jeff Dean is a Wizard and Google Senior Fellow
in the Systems and Infrastructure Group at Google)

Method: RNN/LSTM – Deep Learning (Apa itu?)



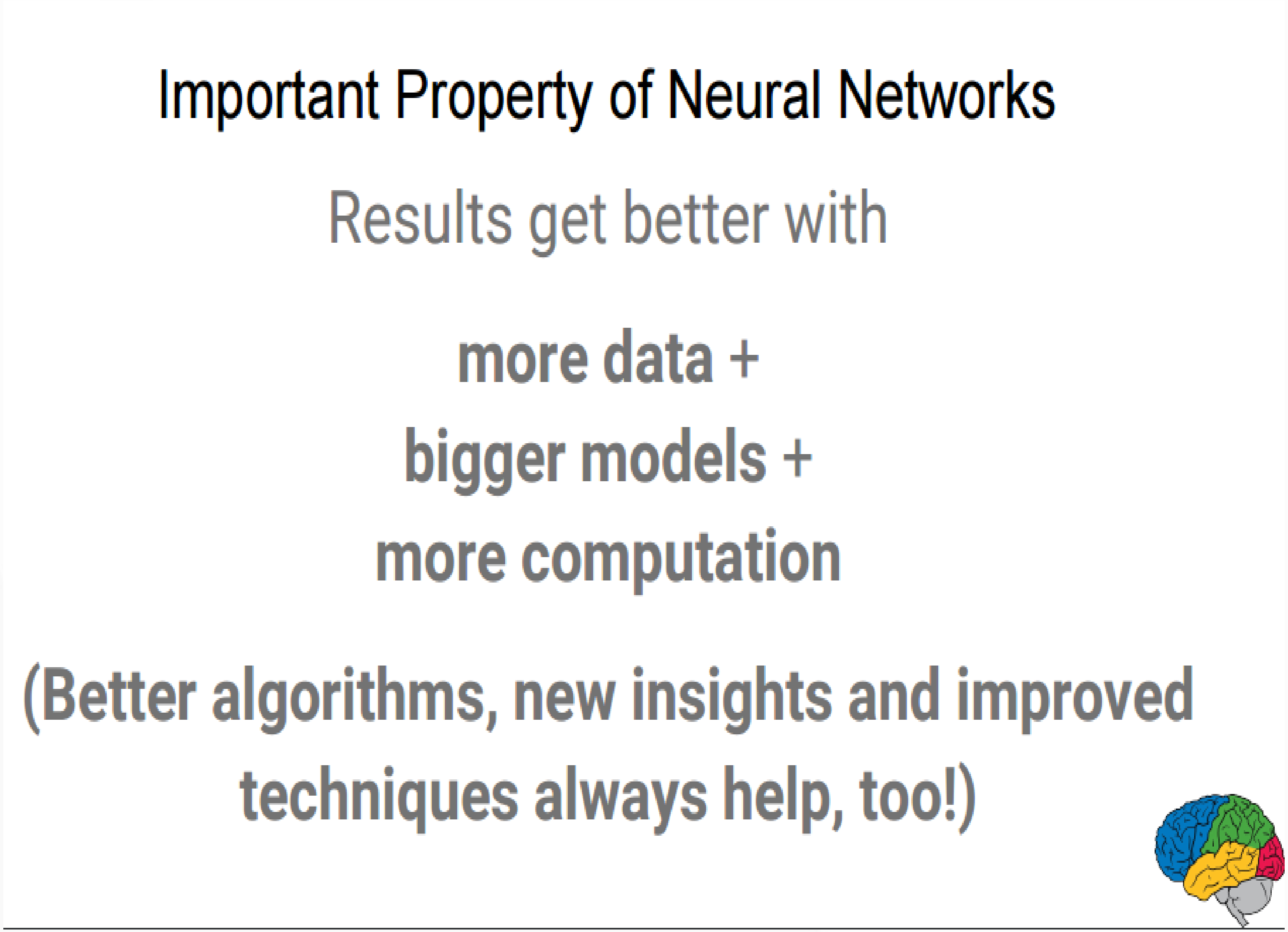
Source: Kdnuggets

Method: RNN/LSTM – Deep Learning (Mengapa?)



Source: machinelearningmastery

Method: RNN/LSTM – Deep Learning (Mengapa?)



Important Property of Neural Networks

Results get better with

more data +
bigger models +
more computation

(Better algorithms, new insights and improved techniques always help, too!)

Source: machinelearningmastery

Method: RNN/LSTM – Macam Deep Learning

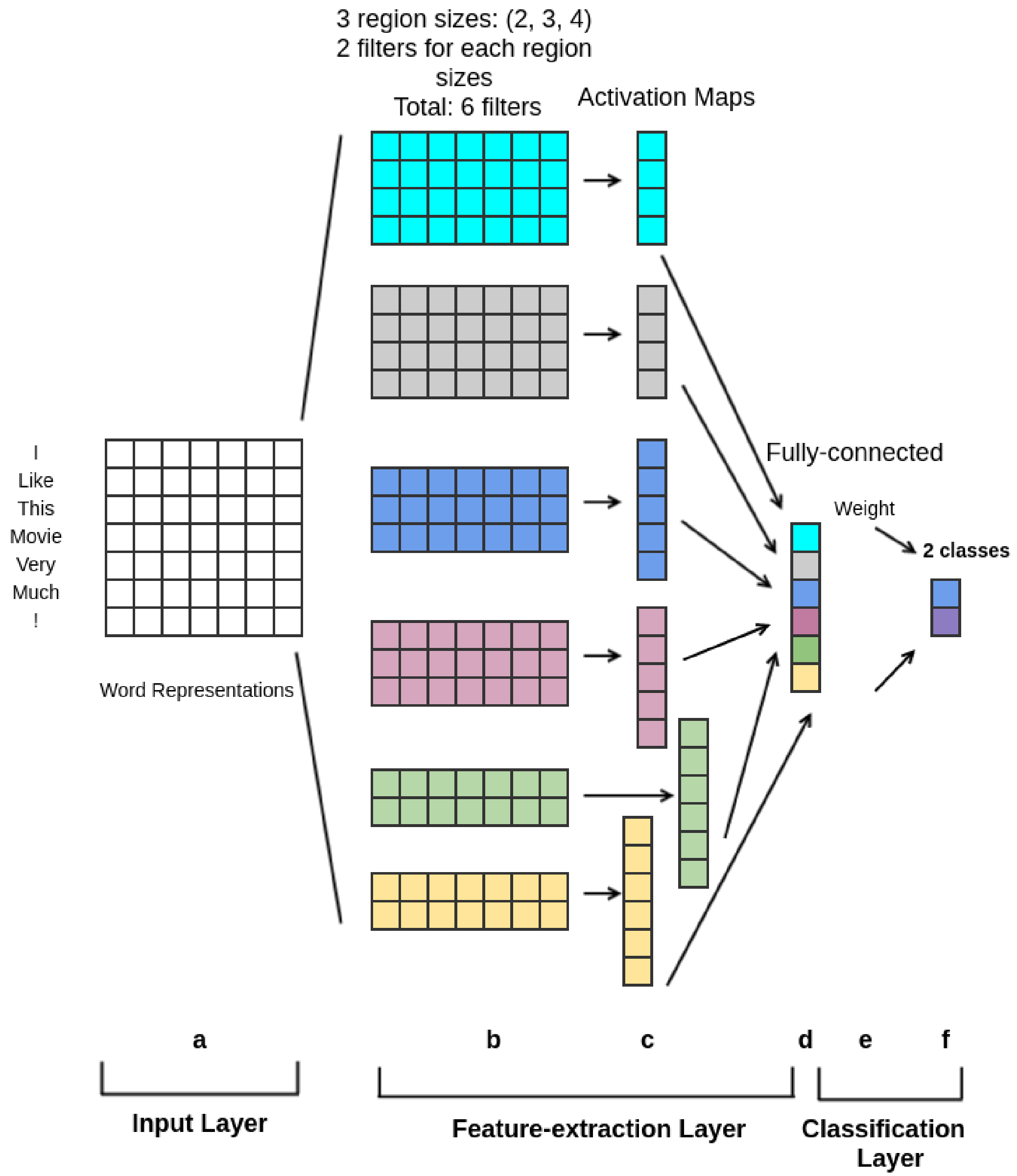
- **Convolution Neural Networks**
- **Recurrent Neural Networks**

Source:
machinelearningmastery

Method: RNN/LSTM – Deep Learning (CNN)

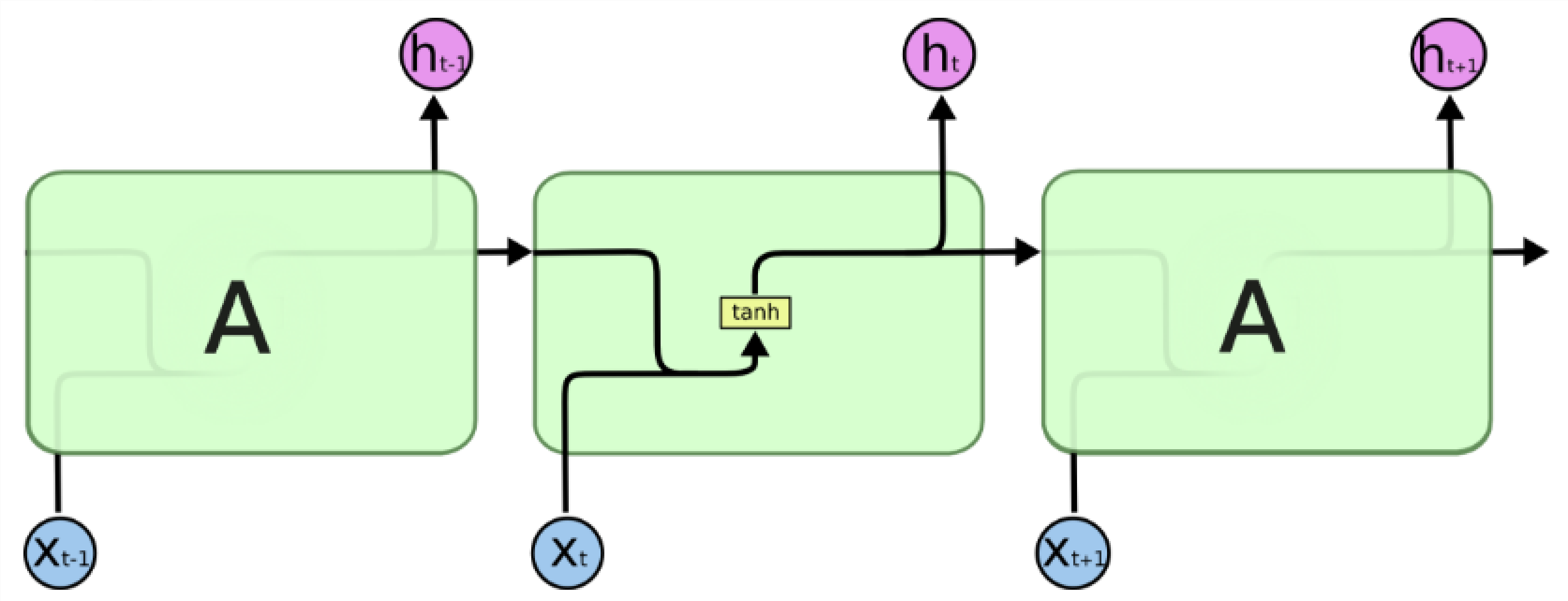
- Implementasi CNN sebelumnya banyak diterapkan pada kasus image.

- Dijalankan memakai algoritma Backpropagation.



Convolution Neural Networks

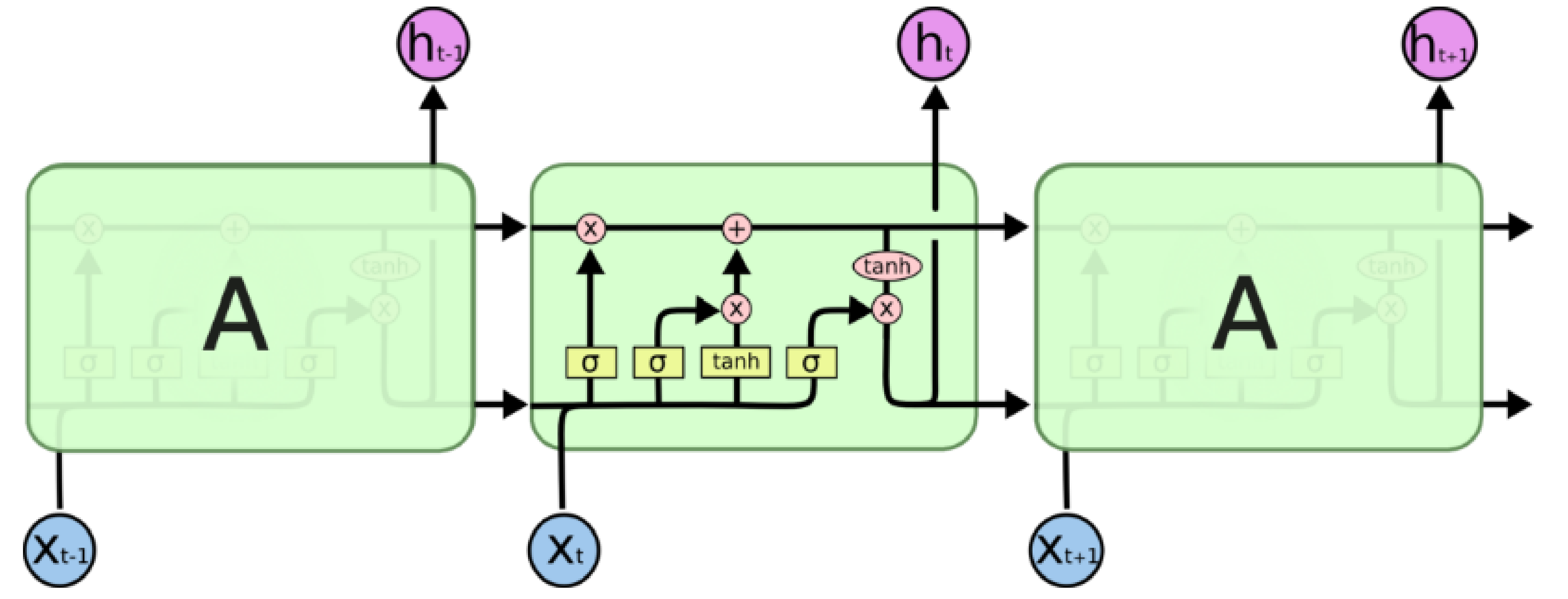
Method: RNN/LSTM – Deep Learning



Recurrent Neural Networks

- Cocok untuk *task* yang sifatnya *Sequential* (e.g. *Text*)
- Dijalankan menggunakan algoritme *Backpropagation Through Time* (BPTT)
- Mempunyai kekurangan:
  - *Vanishing Gradient*:Jika bobotnya kecil, maka nilai berikutnya akan terus semakin kecil dan cenderung mendekati 0.
  - *Exploding Gradient*: Jika bobotnya besar, nilai berikutnya akan tak terbatas.
  - Kekurangan di atas terjadi ketika *Sequence* terlalu panjang.

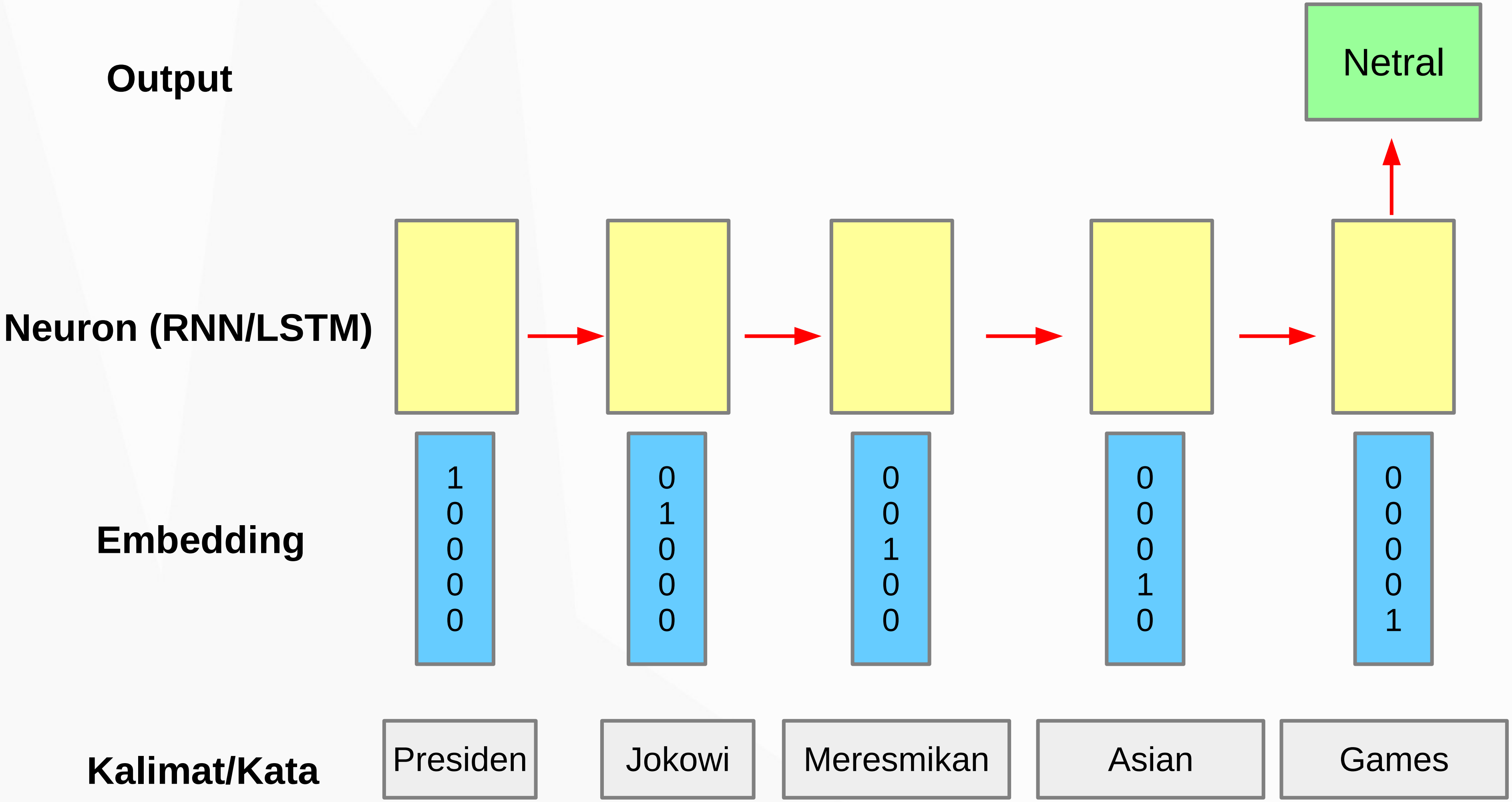
Method: RNN/LSTM – Deep Learning



Long-short term memory

- Cocok untuk *task Sequential* yang cenderung panjang
- Mampu memberikan solusi permasalahan *Vanishing Gradient* dan *Exploding Gradient* dengan menambahkan *Gate* (*Input Gate, Forget Gate, Output Gate*) pada *neuron*-nya.
- *Gate* tersebut mengontrol *gradient* dengan mengatur informasi mana yang mesti disimpan dan dihapus.
- Komputasi LSTM lebih berat daripada RNN biasa.
- Dijalankan menggunakan algoritme *Backpropagation Through Time* (BPTT)

Method: RNN/LSTM – Bagaimana RNN/LSTM bekerja?



Output

Netral

Neuron (RNN/LSTM)

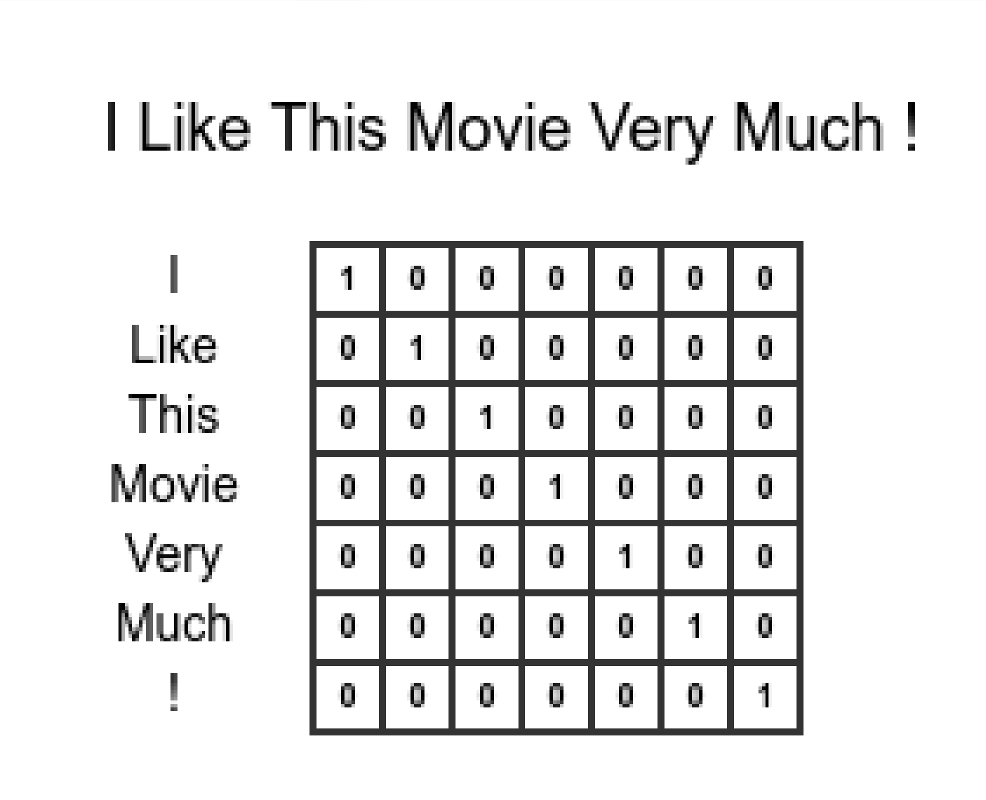
Embedding

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

Kalimat/Kata

Presiden Jokowi Meresmikan Asian Games

Method: RNN/LSTM – Bagaimana konversi kalimat ke embedding?



I Like This Movie Very Much !

I	1	0	0	0	0	0	0
Like	0	1	0	0	0	0	0
This	0	0	1	0	0	0	0
Movie	0	0	0	1	0	0	0
Very	0	0	0	0	1	0	0
Much	0	0	0	0	0	1	0
!	0	0	0	0	0	0	1

Embedding (One Hot Encoding)

Konversi dari kalimat ke angka dapat Menggunakan *One Hot Encoding* dan *Pre-training* word2vec. Contoh di atas menggunakan *One Hot Encoding.*

Panjang Matriks dipengaruhi panjang kalimat. Semakin panjang kalimat, semakin panjang pula matriksnya.

Method: RNN/LSTM – Bagaimana konversi kalimat ke embedding? (2)

1. CBOW (continuous bag of words)
2. Skipgram

Word Representations (Word2vec)

Jika memakai Word2vec kita bisa mengatur panjang dari Matriks.

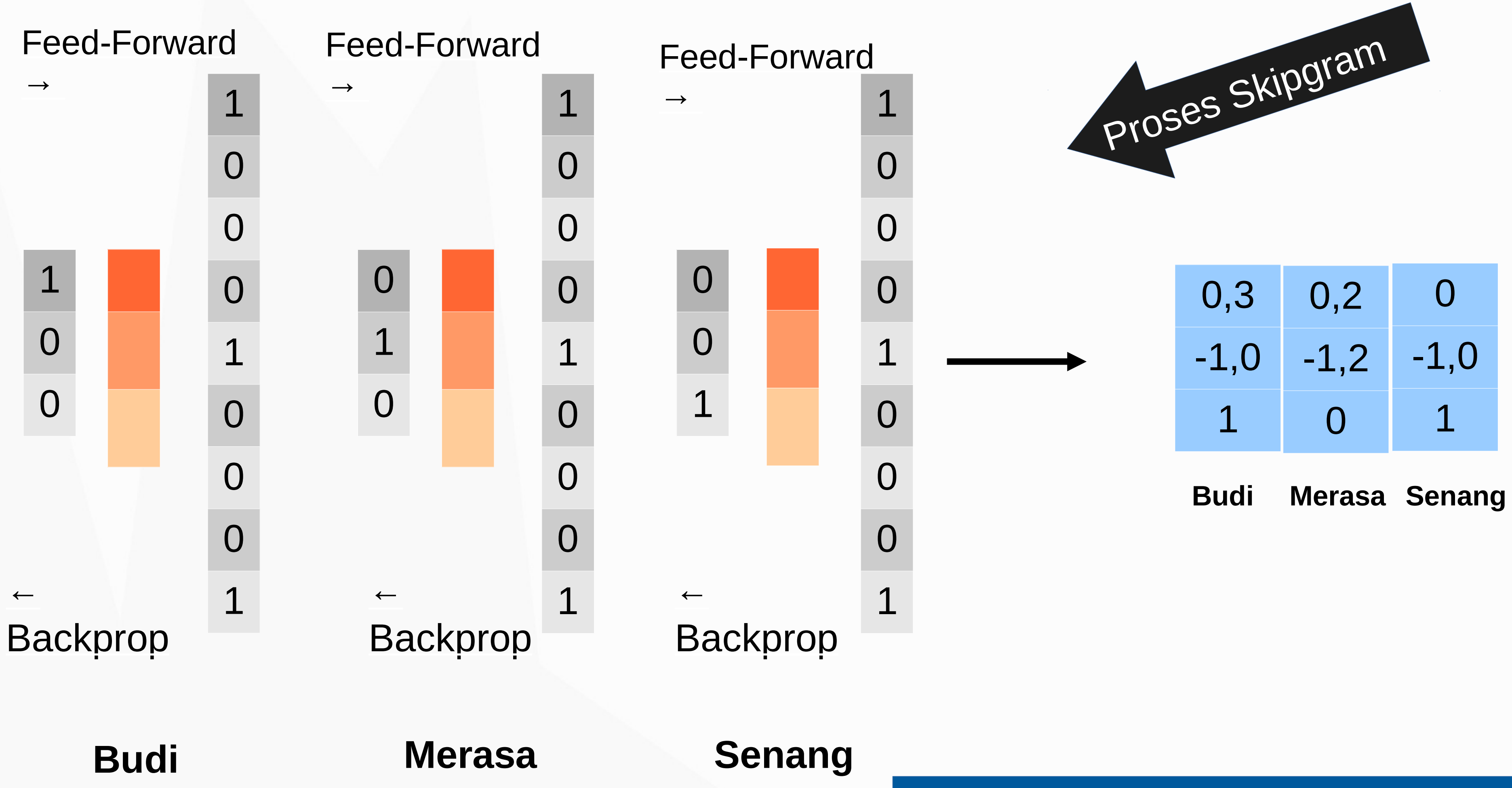
Method: RNN/LSTM – Bagaimana konversi kalimat ke embedding? (3)

Budi Merasa Senang

Budi → 1 0 0
Merasa → 0 1 0
Senang → 0 0 1

Untuk menggunakan Word2vec, terlebih dahulu mesti mengubah kalimat menjadi angka menggunakan
One Hot Encoding

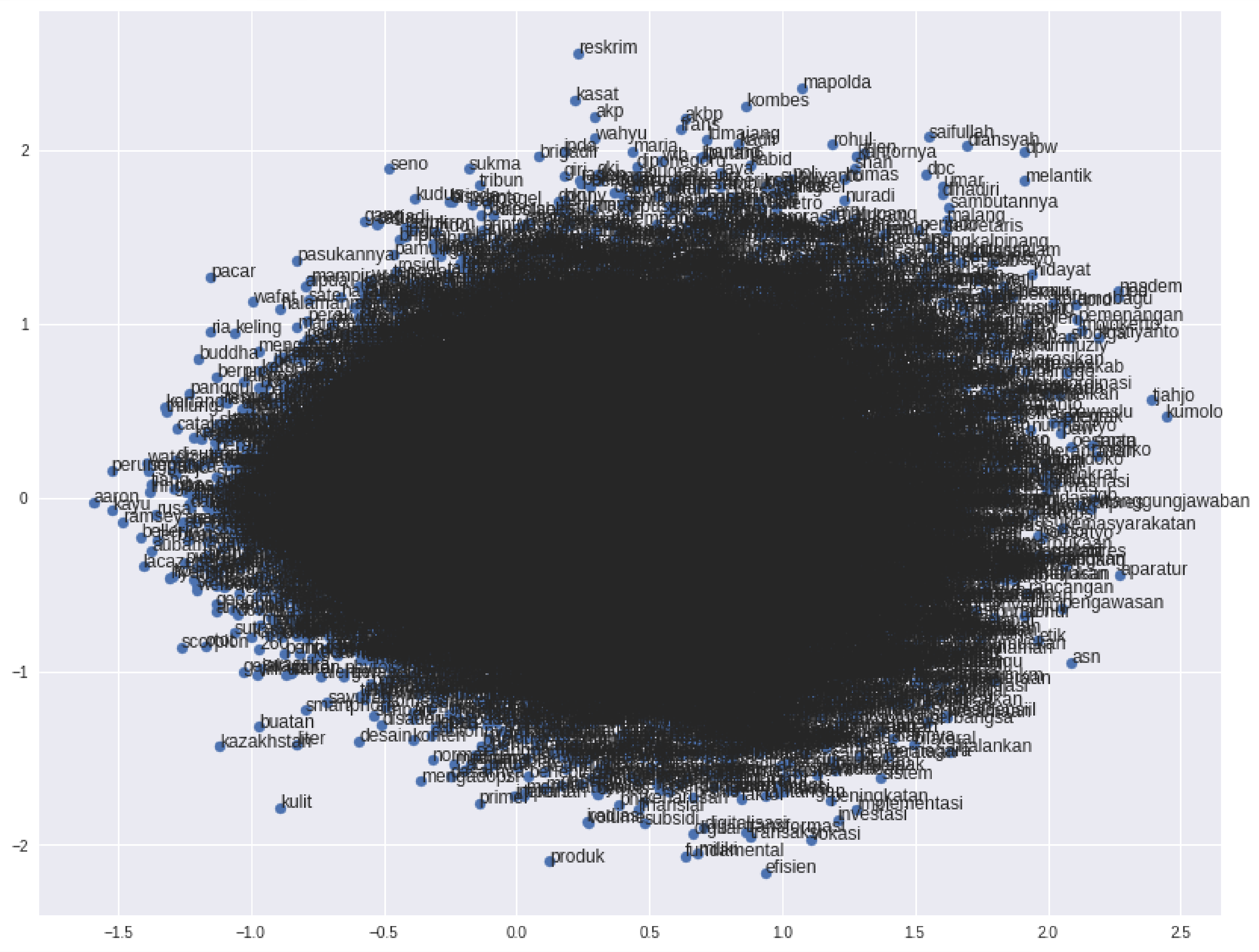
Method: RNN/LSTM – Bagaimana konversi kalimat ke embedding? (3)



Budi	Merasa	Senang
0,3	0,2	0
-1,0	-1,2	-1,0
1	0	1

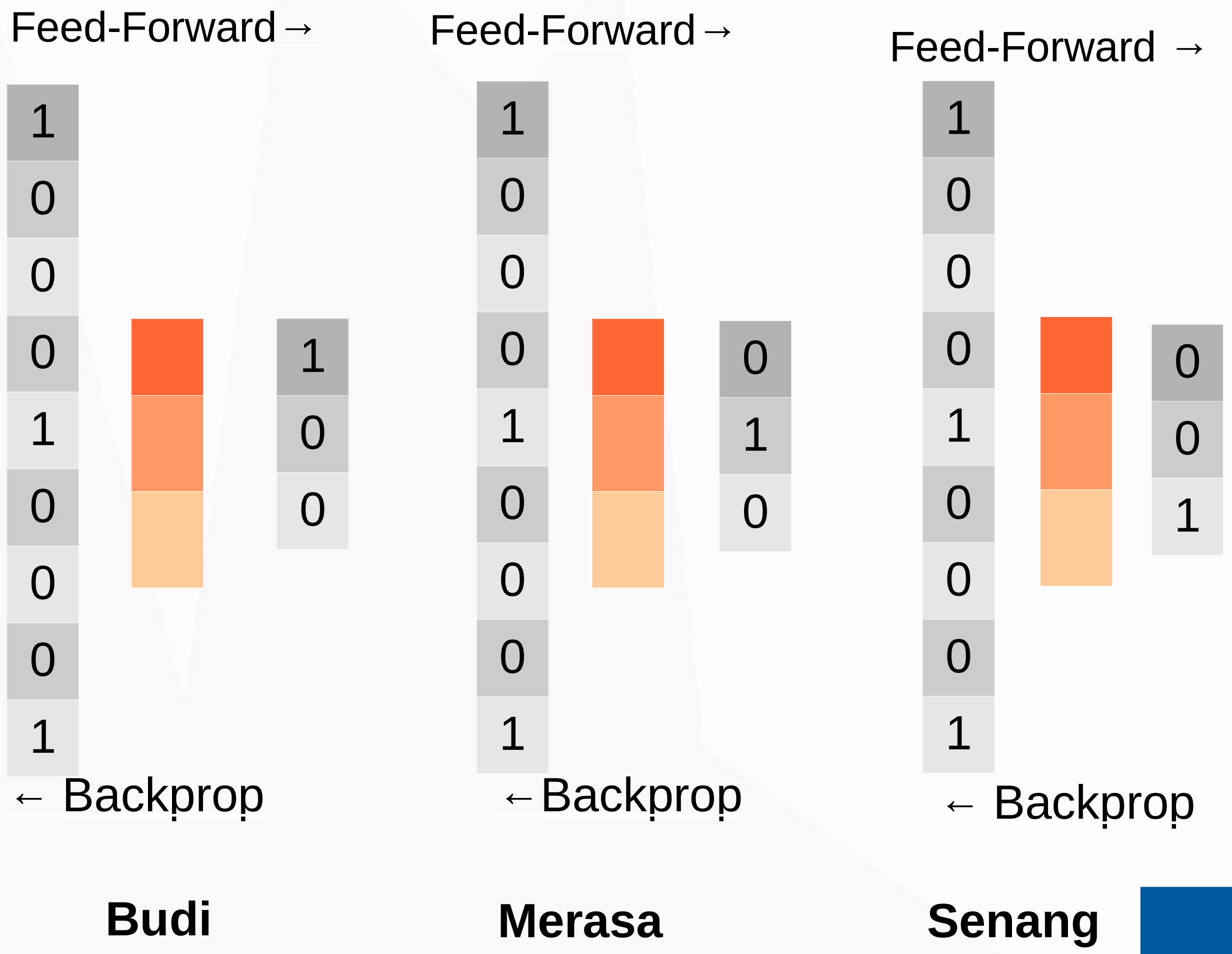
Skipgram bekerja dengan melihat konteks dari suatu kata

Method: RNN/LSTM – Bagaimana konversi kalimat ke embedding? (4)



Hasil Skipgram

Method: RNN/LSTM – Bagaimana konversi kalimat ke embedding? (5)

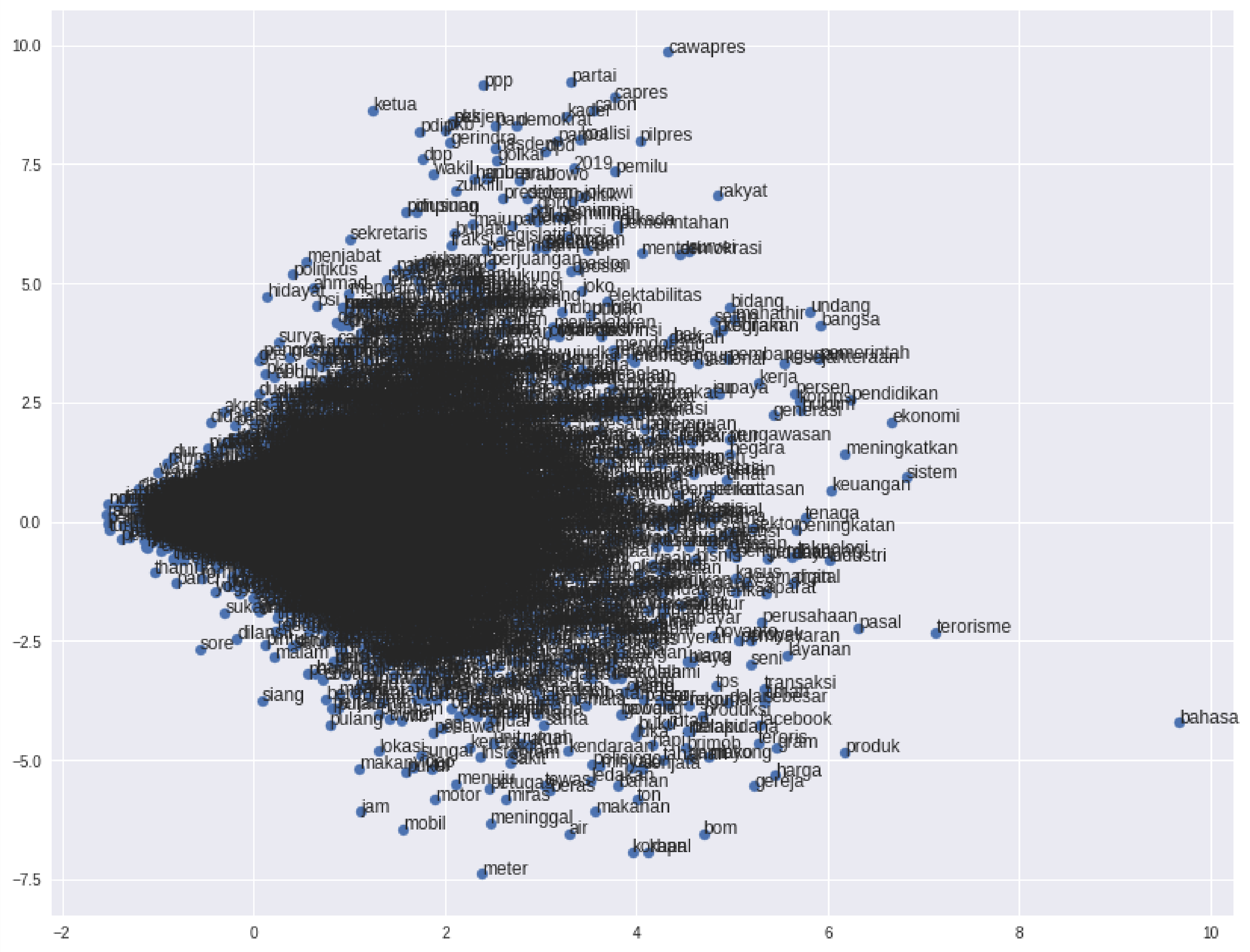


Proses CBOW

Budi	Merasa	Senang
0	0,2	0,7
-0,9	-1,1	-1,0
0	1	1

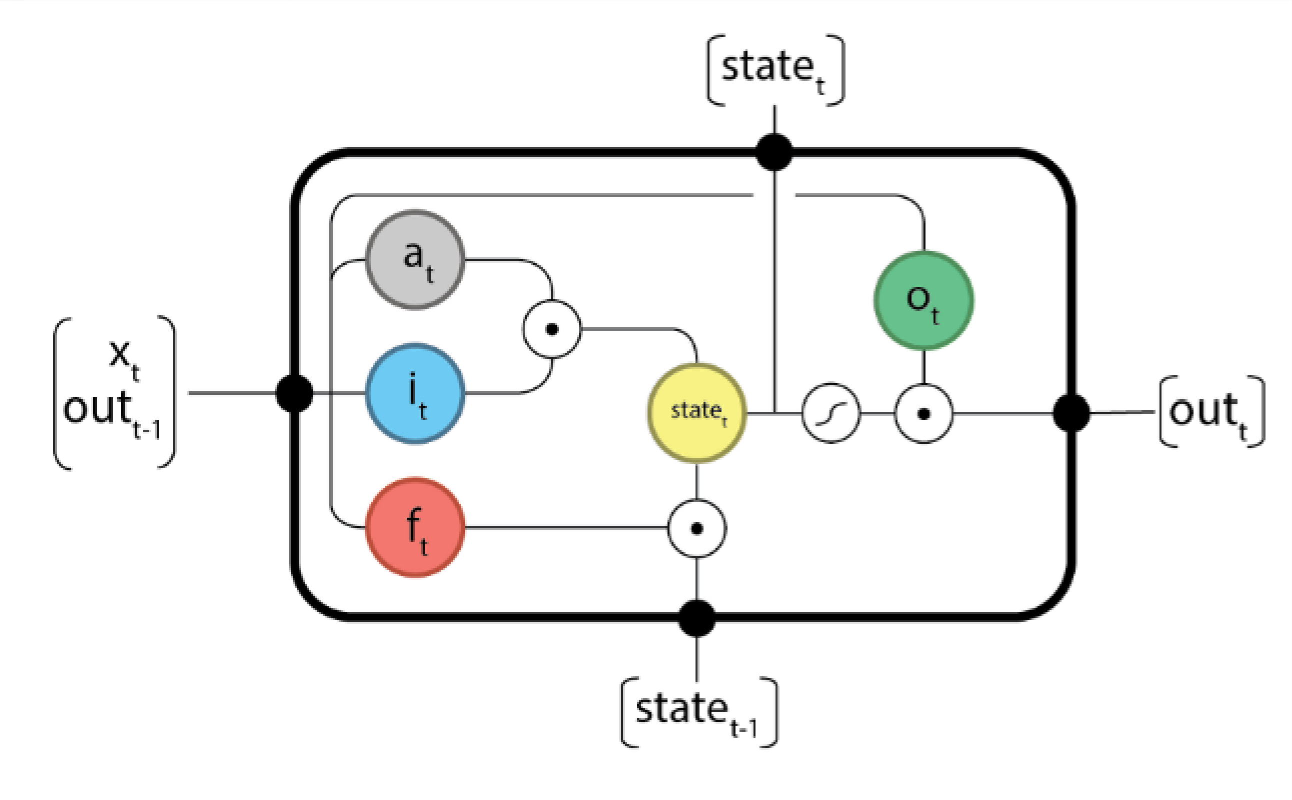
CBOW bekerja dengan memprediksi kata dari sebuah konteks

Method: RNN/LSTM – Bagaimana konversi kalimat ke embedding? (5)

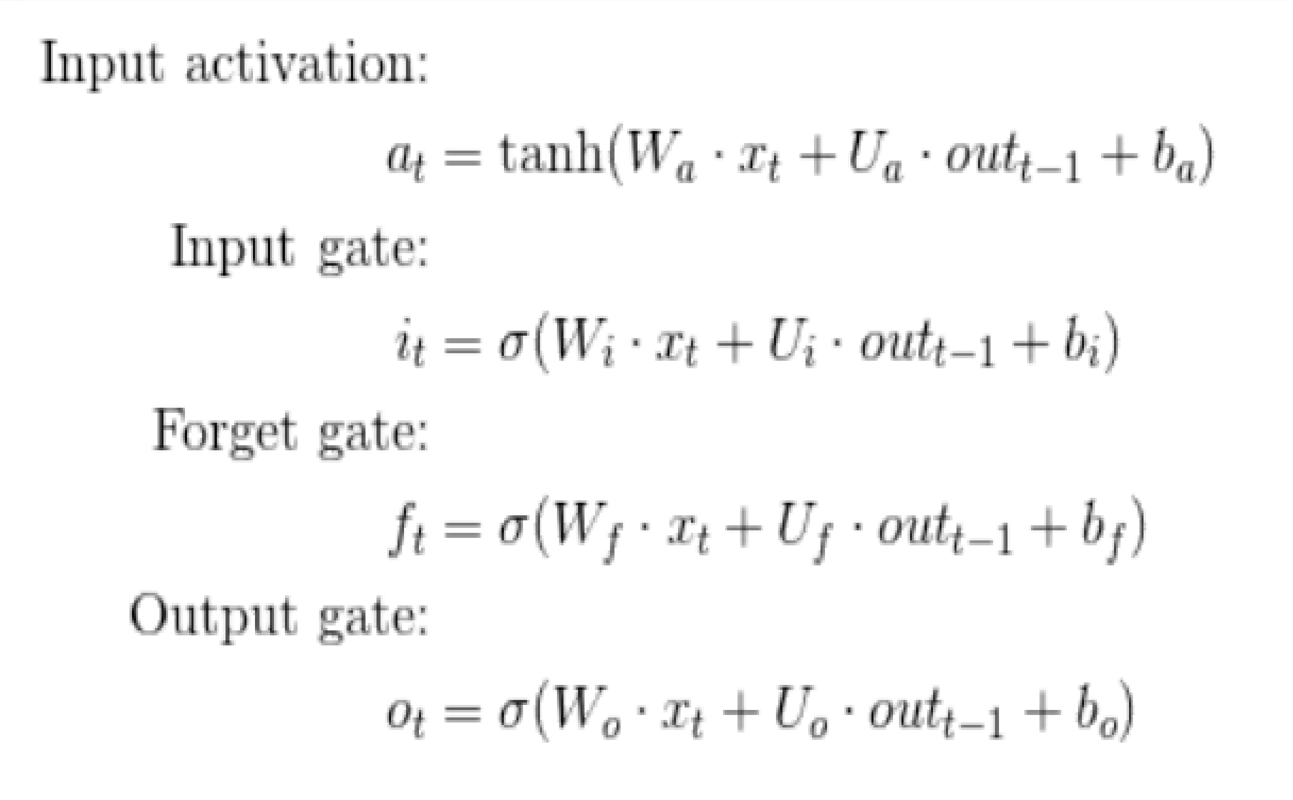


Hasil CBOW

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM?



Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (2)



Input activation:

$$a_t = \tanh(W_a \cdot x_t + U_a \cdot out_{t-1} + b_a)$$

Input gate:

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot out_{t-1} + b_i)$$

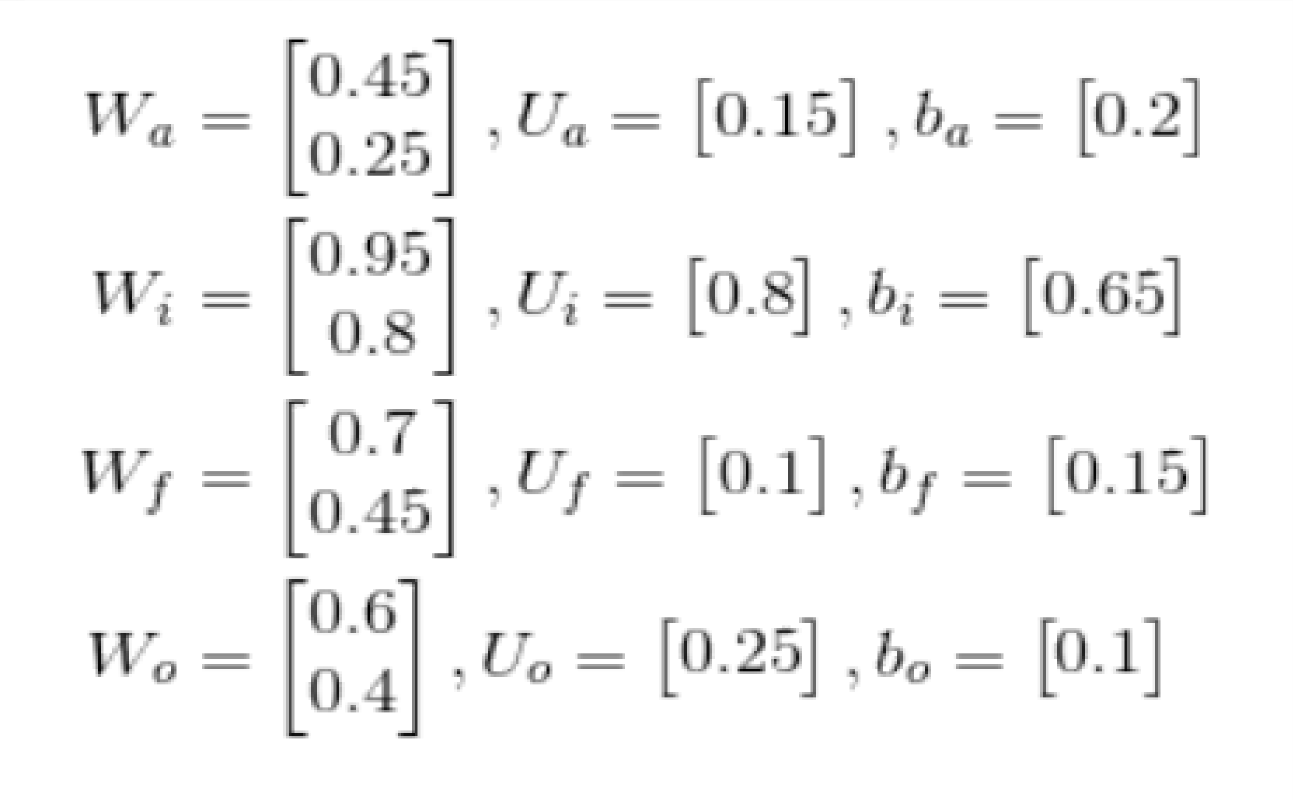
Forget gate:

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot out_{t-1} + b_f)$$

Output gate:

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot out_{t-1} + b_o)$$

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (3)



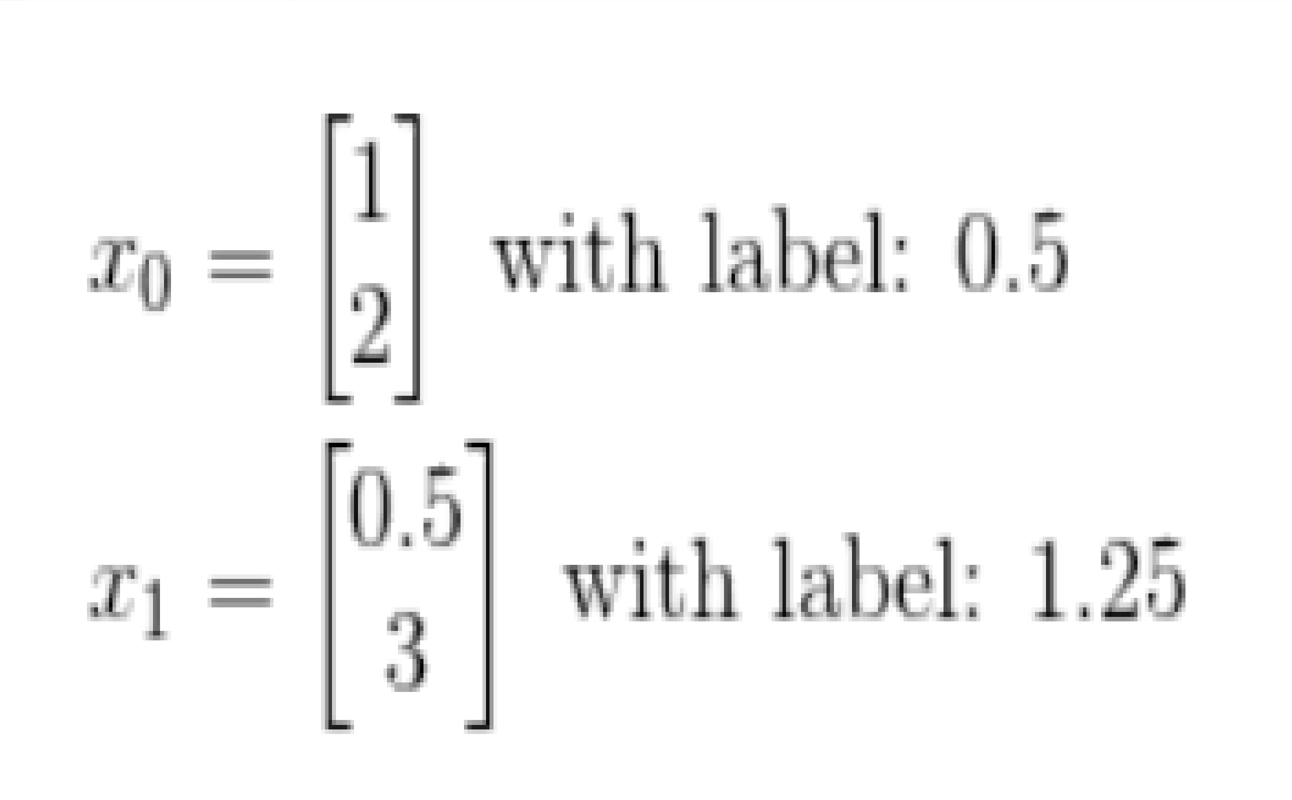
$$W_a = \begin{bmatrix} 0.45 \\ 0.25 \end{bmatrix}, U_a = \begin{bmatrix} 0.15 \end{bmatrix}, b_a = \begin{bmatrix} 0.2 \end{bmatrix}$$

$$W_i = \begin{bmatrix} 0.95 \\ 0.8 \end{bmatrix}, U_i = \begin{bmatrix} 0.8 \end{bmatrix}, b_i = \begin{bmatrix} 0.65 \end{bmatrix}$$

$$W_f = \begin{bmatrix} 0.7 \\ 0.45 \end{bmatrix}, U_f = \begin{bmatrix} 0.1 \end{bmatrix}, b_f = \begin{bmatrix} 0.15 \end{bmatrix}$$

$$W_o = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, U_o = \begin{bmatrix} 0.25 \end{bmatrix}, b_o = \begin{bmatrix} 0.1 \end{bmatrix}$$

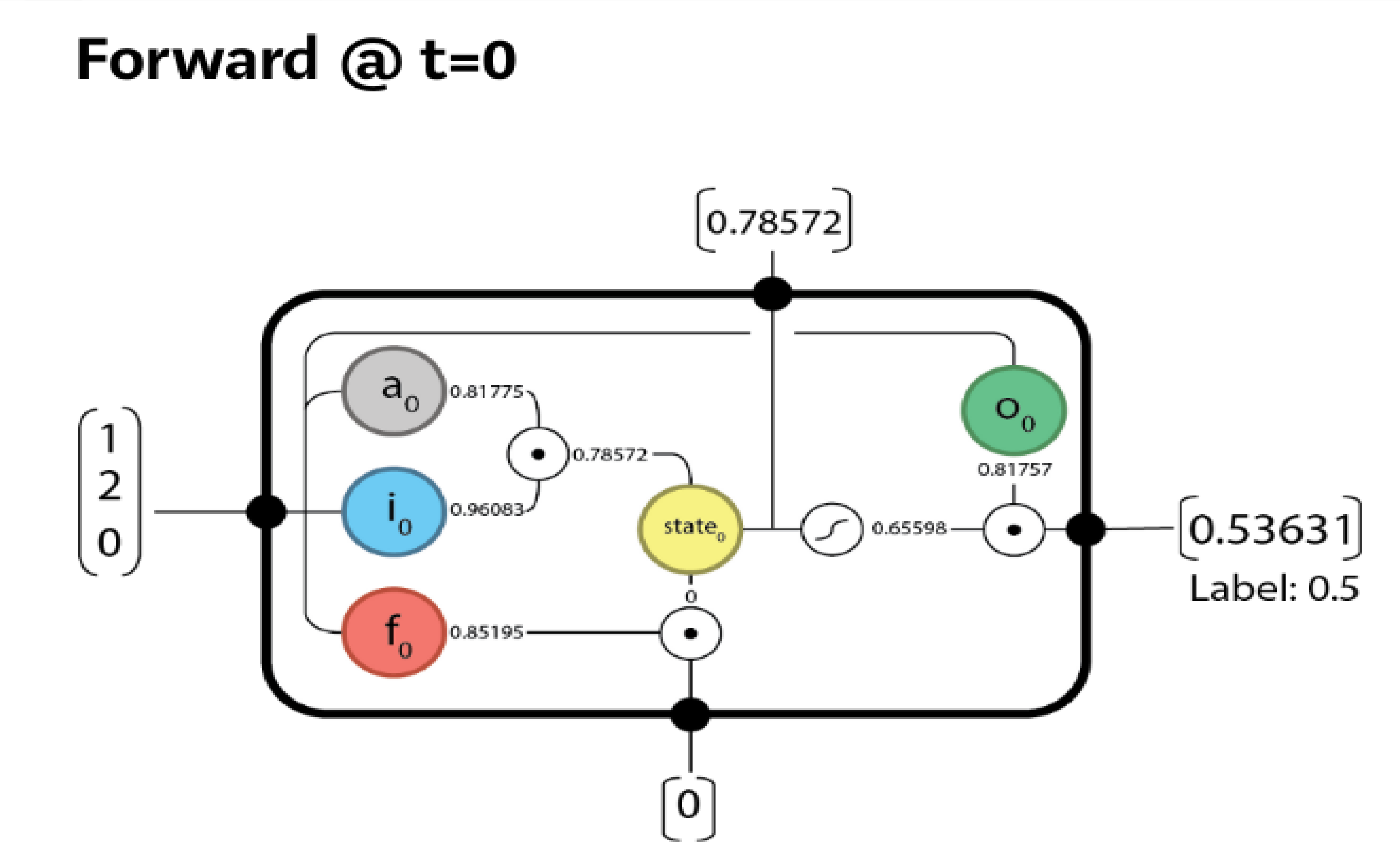
Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (4)



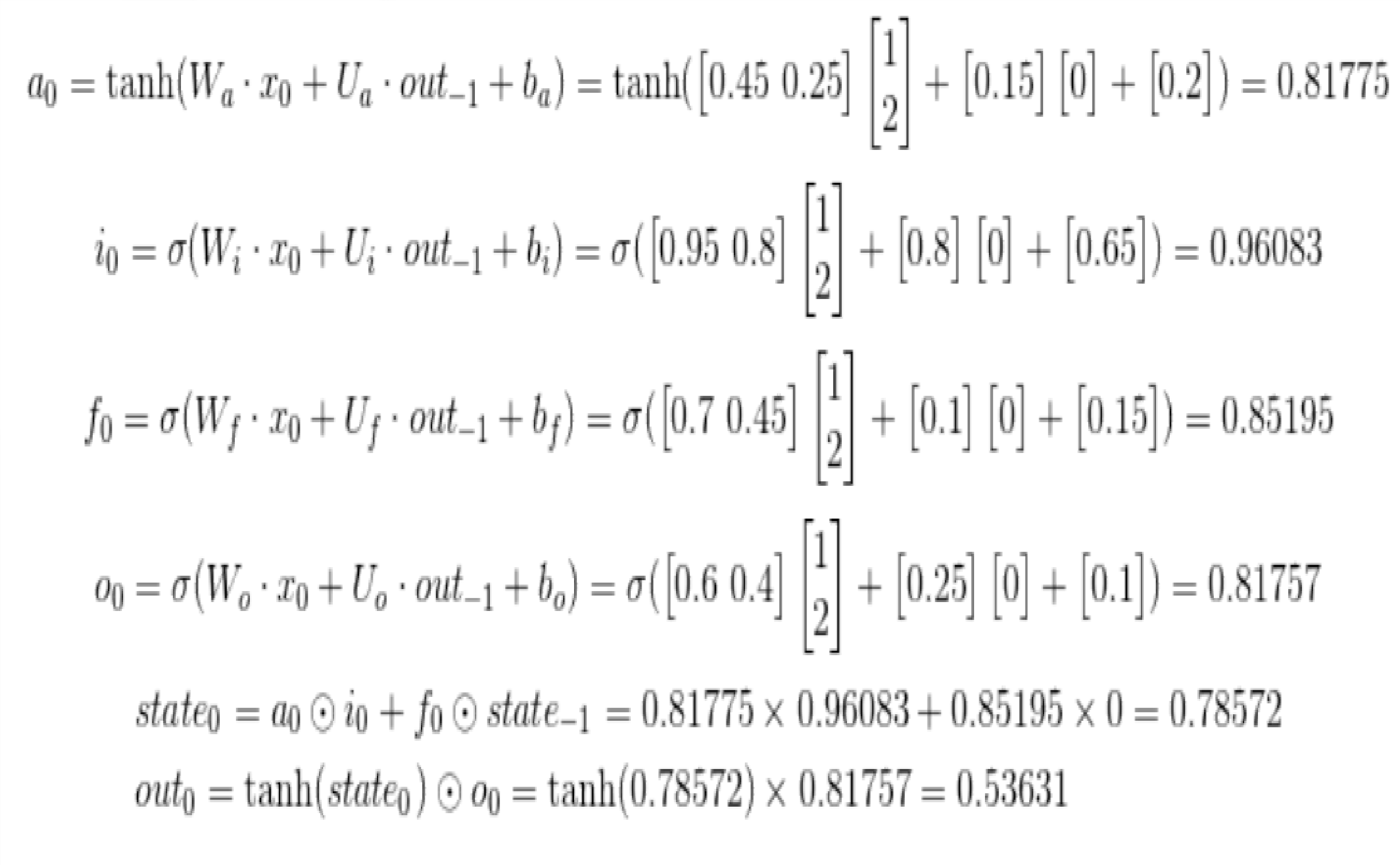
$$x_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ with label: } 0.5$$

$$x_1 = \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} \text{ with label: } 1.25$$

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (5)



Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (6)



$$a_0 = \tanh(W_a \cdot x_0 + U_a \cdot out_{-1} + b_a) = \tanh(\begin{bmatrix}0.45 & 0.25\end{bmatrix} \begin{bmatrix}1 \\ 2\end{bmatrix} + \begin{bmatrix}0.15\end{bmatrix} \begin{bmatrix}0\end{bmatrix} + \begin{bmatrix}0.2\end{bmatrix}) = 0.81775$$

$$i_0 = \sigma(W_i \cdot x_0 + U_i \cdot out_{-1} + b_i) = \sigma(\begin{bmatrix}0.95 & 0.8\end{bmatrix} \begin{bmatrix}1 \\ 2\end{bmatrix} + \begin{bmatrix}0.8\end{bmatrix} \begin{bmatrix}0\end{bmatrix} + \begin{bmatrix}0.65\end{bmatrix}) = 0.96083$$

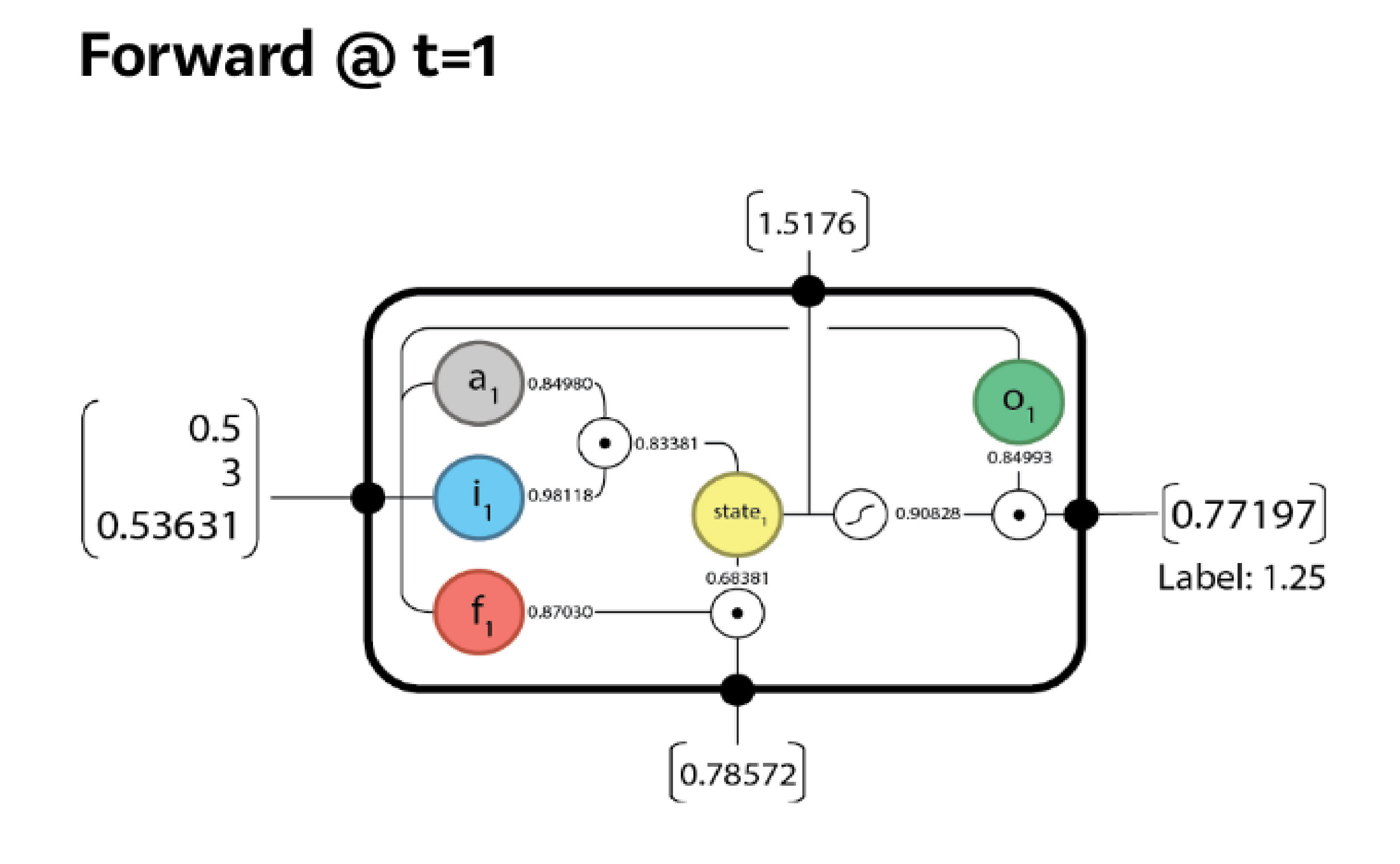
$$f_0 = \sigma(W_f \cdot x_0 + U_f \cdot out_{-1} + b_f) = \sigma(\begin{bmatrix}0.7 & 0.45\end{bmatrix} \begin{bmatrix}1 \\ 2\end{bmatrix} + \begin{bmatrix}0.1\end{bmatrix} \begin{bmatrix}0\end{bmatrix} + \begin{bmatrix}0.15\end{bmatrix}) = 0.85195$$

$$o_0 = \sigma(W_o \cdot x_0 + U_o \cdot out_{-1} + b_o) = \sigma(\begin{bmatrix}0.6 & 0.4\end{bmatrix} \begin{bmatrix}1 \\ 2\end{bmatrix} + \begin{bmatrix}0.25\end{bmatrix} \begin{bmatrix}0\end{bmatrix} + \begin{bmatrix}0.1\end{bmatrix}) = 0.81757$$

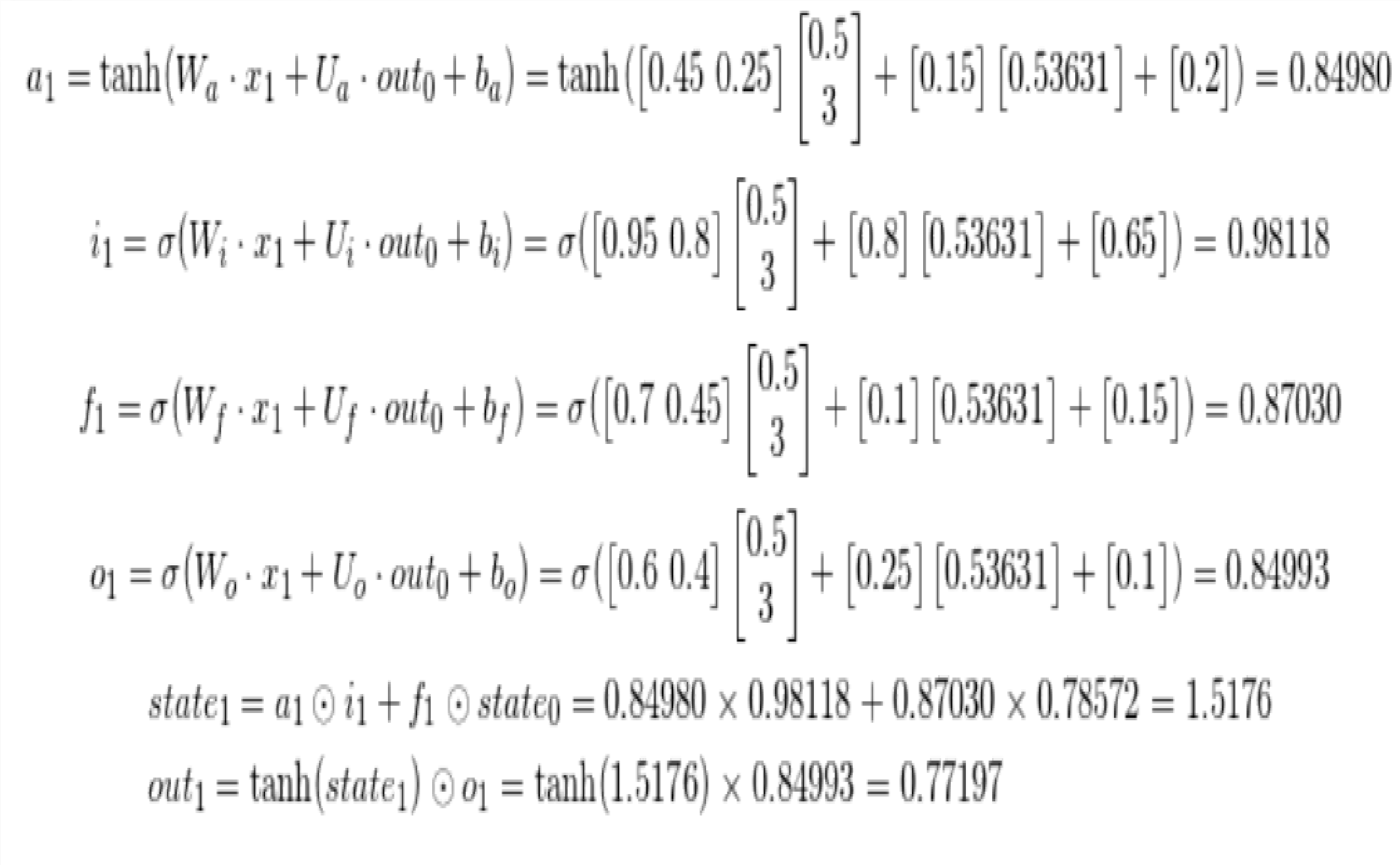
$$state_0 = a_0 \odot i_0 + f_0 \odot state_{-1} = 0.81775 \times 0.96083 + 0.85195 \times 0 = 0.78572$$

$$out_0 = \tanh(state_0) \odot o_0 = \tanh(0.78572) \times 0.81757 = 0.53631$$

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (8)



Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (9)



$$a_1 = \tanh(W_a \cdot x_1 + U_a \cdot out_0 + b_a) = \tanh\left(\begin{bmatrix}0.45 & 0.25\end{bmatrix} \begin{bmatrix}0.5 \\ 3\end{bmatrix} + \begin{bmatrix}0.15\end{bmatrix} \begin{bmatrix}0.53631\end{bmatrix} + \begin{bmatrix}0.2\end{bmatrix}\right) = 0.84980$$

$$i_1 = \sigma(W_i \cdot x_1 + U_i \cdot out_0 + b_i) = \sigma\left(\begin{bmatrix}0.95 & 0.8\end{bmatrix} \begin{bmatrix}0.5 \\ 3\end{bmatrix} + \begin{bmatrix}0.8\end{bmatrix} \begin{bmatrix}0.53631\end{bmatrix} + \begin{bmatrix}0.65\end{bmatrix}\right) = 0.98118$$

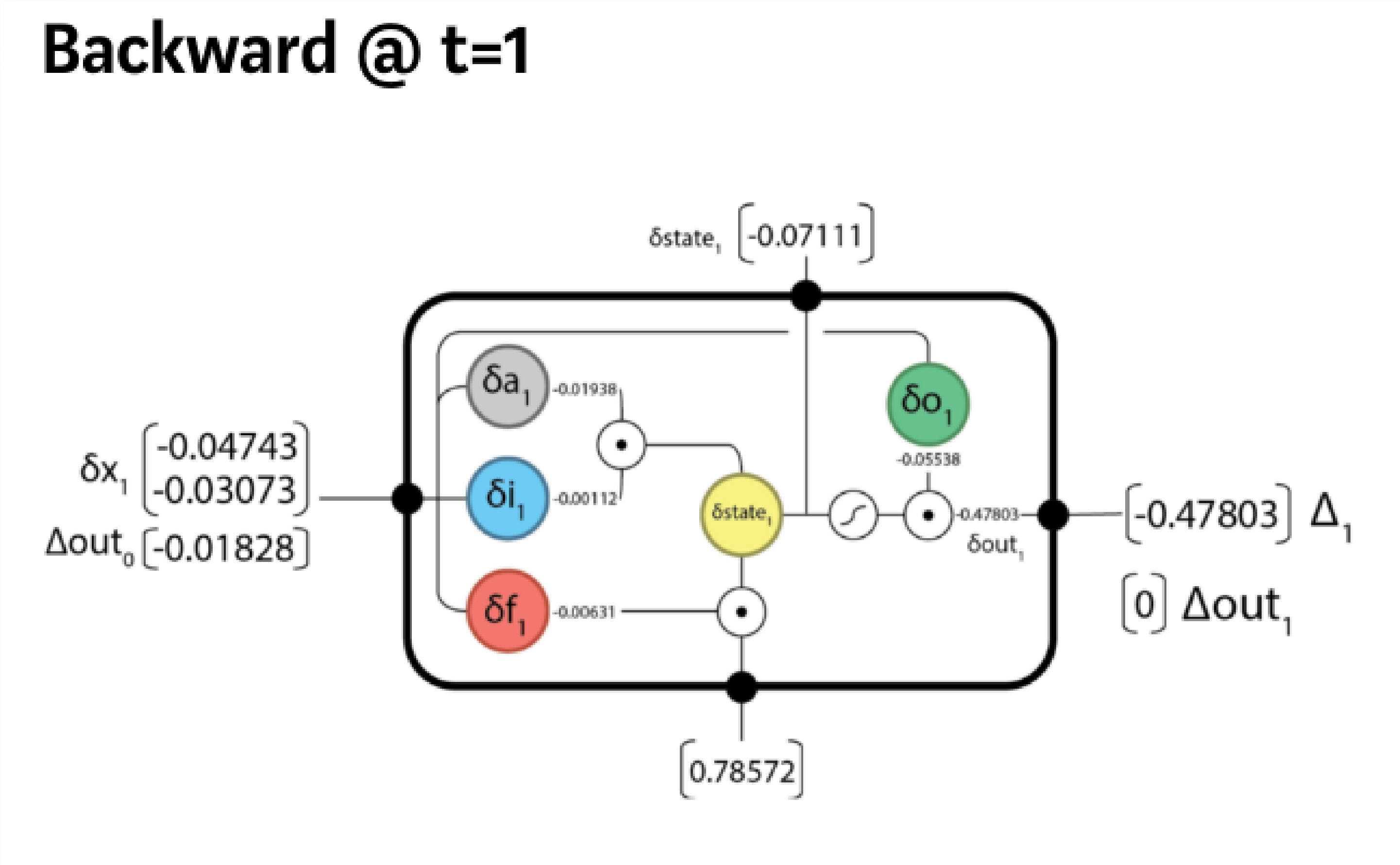
$$f_1 = \sigma(W_f \cdot x_1 + U_f \cdot out_0 + b_f) = \sigma\left(\begin{bmatrix}0.7 & 0.45\end{bmatrix} \begin{bmatrix}0.5 \\ 3\end{bmatrix} + \begin{bmatrix}0.1\end{bmatrix} \begin{bmatrix}0.53631\end{bmatrix} + \begin{bmatrix}0.15\end{bmatrix}\right) = 0.87030$$

$$o_1 = \sigma(W_o \cdot x_1 + U_o \cdot out_0 + b_o) = \sigma\left(\begin{bmatrix}0.6 & 0.4\end{bmatrix} \begin{bmatrix}0.5 \\ 3\end{bmatrix} + \begin{bmatrix}0.25\end{bmatrix} \begin{bmatrix}0.53631\end{bmatrix} + \begin{bmatrix}0.1\end{bmatrix}\right) = 0.84993$$

$$state_1 = a_1 \odot i_1 + f_1 \odot state_0 = 0.84980 \times 0.98118 + 0.87030 \times 0.78572 = 1.5176$$

$$out_1 = \tanh(state_1) \odot o_1 = \tanh(1.5176) \times 0.84993 = 0.77197$$

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (10)



Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (11)

$$\Delta_1 = \partial_x E = 0.77197 - 1.25 = -0.47803$$

$$\Delta out_1 = 0 \text{ because there are no future time-steps.}$$

$$\delta out_1 = \Delta_1 + \Delta out_1 = -0.47803 + 0 = -0.47803$$

$$\delta state_1 = \delta out_1 \odot o_1 \odot (1 - \tanh^2(state_1)) + \delta state_2 \odot f_2 = -0.47803 \times 0.84993 \times (1 - \tanh^2(1.5176)) + 0 \times 0 = -0.07111$$

$$\delta a_1 = \delta state_1 \odot i_1 \odot (1 - a_1^2) = -0.07111 \times 0.98118 \times (1 - 0.84980^2) = -0.01938$$

$$\delta i_1 = \delta state_1 \odot a_1 \odot i_1 \odot (1 - i_1) = -0.07111 \times 0.84980 \times 0.98118 \times (1 - 0.98118) = -0.00112$$

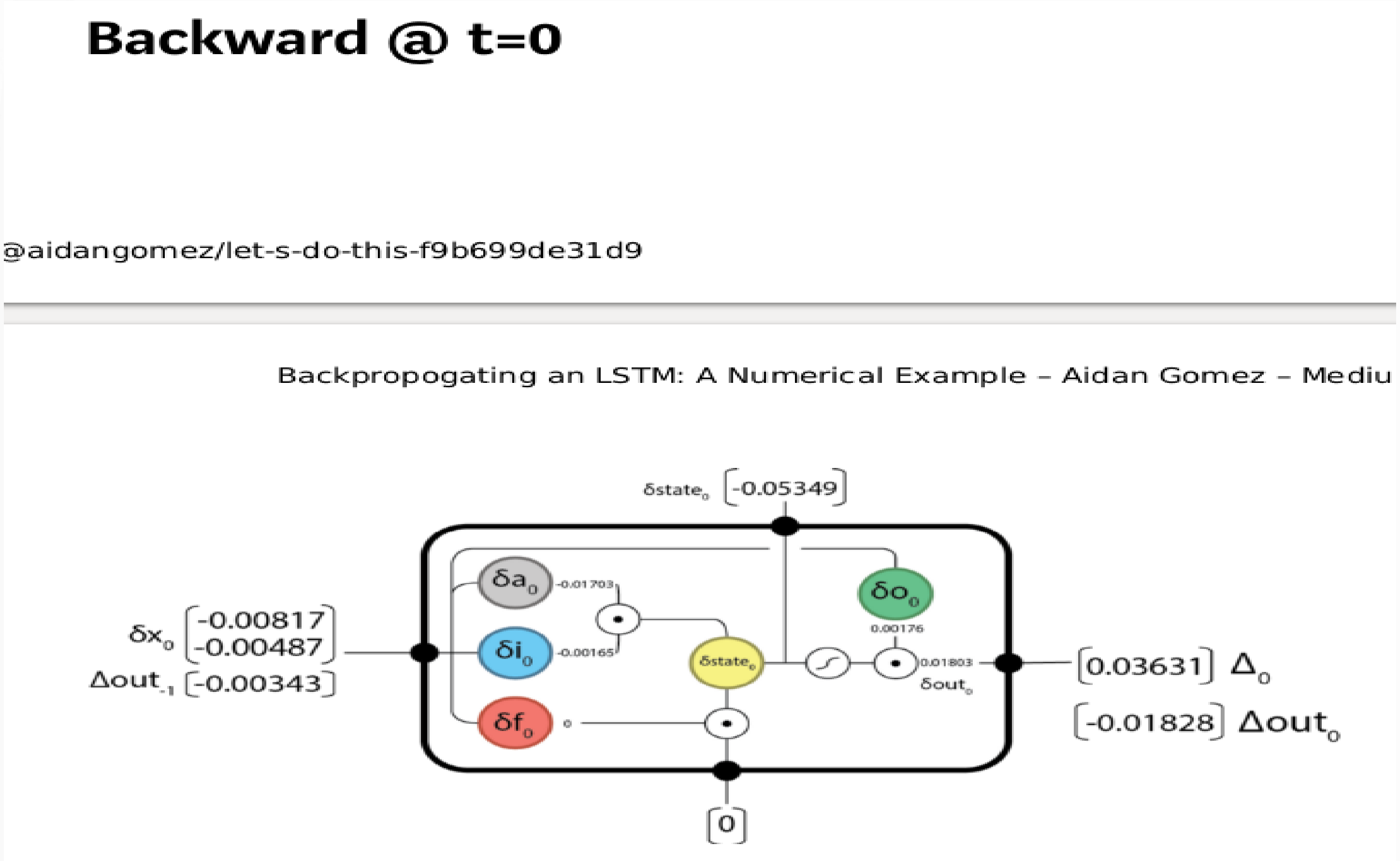
$$\delta f_1 = \delta state_1 \odot state_0 \odot f_1 \odot (1 - f_1) = -0.07111 \times 0.78572 \times 0.87030 \times (1 - 0.87030) = -0.00631$$

$$\delta o_1 = \delta out_1 \odot \tanh(state_1) \odot o_1 \odot (1 - o_1) = -0.47803 \times \tanh(1.5176) \times 0.84993 \times (1 - 0.84993) = -0.05538$$

$$\begin{aligned} \delta x_1 &= W^T \cdot \delta gates_1 \\ &= \begin{bmatrix} 0.45 & 0.95 & 0.70 & 0.60 \\ 0.25 & 0.80 & 0.45 & 0.40 \end{bmatrix} \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix} = \begin{bmatrix} -0.04743 \\ -0.03073 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \Delta out_0 &= U^T \cdot \delta gates_1 \\ &= \begin{bmatrix} 0.15 & 0.80 & 0.10 & 0.25 \end{bmatrix} \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix} = -0.01828 \end{aligned}$$

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (12)

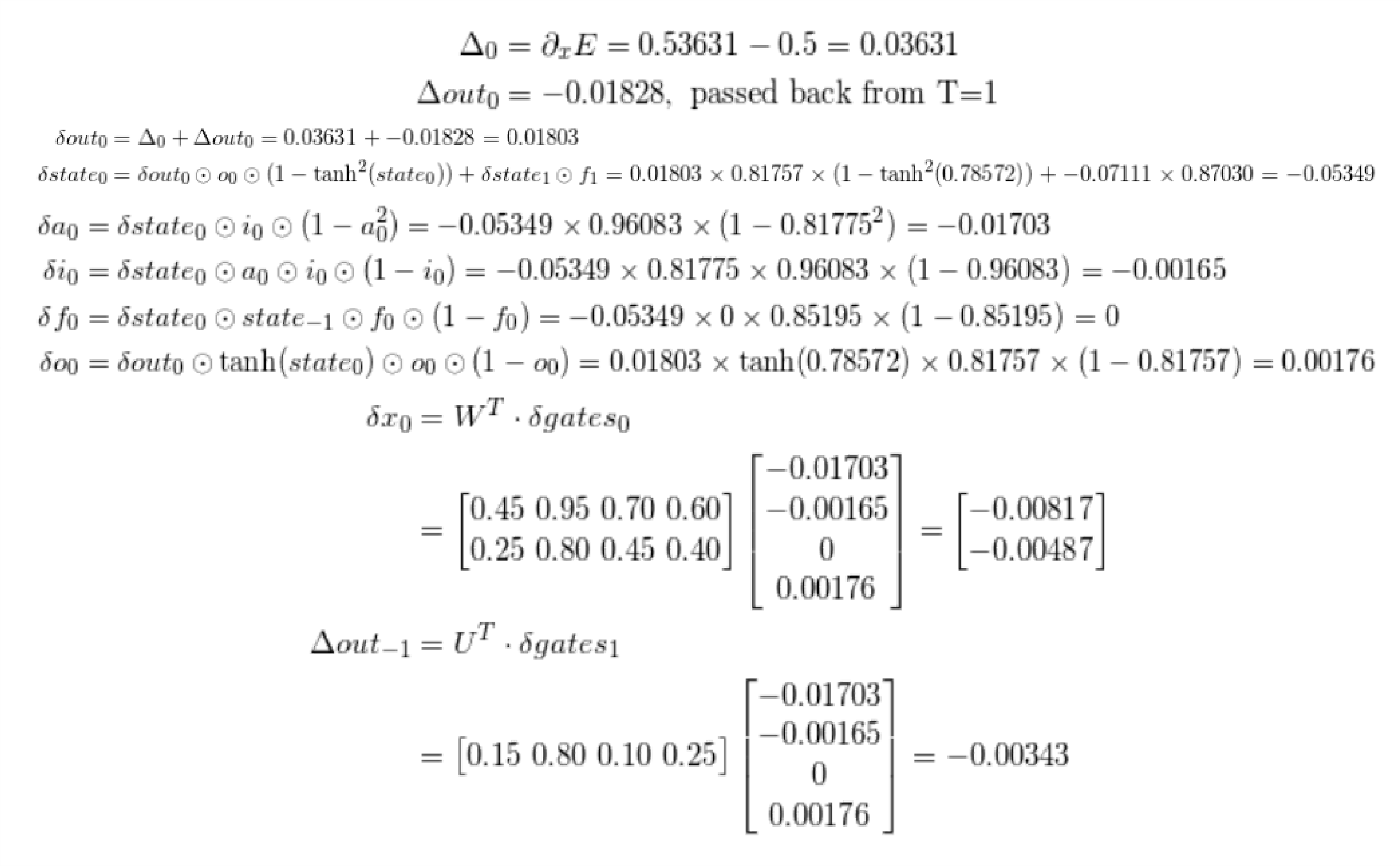


Backward @ t=0

@aidangomez/let-s-do-this-f9b699de31d9

Backpropogating an LSTM: A Numerical Example – Aidan Gomez – Mediu

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (13)



$$\Delta_0 = \partial_x E = 0.53631 - 0.5 = 0.03631$$

$$\Delta out_0 = -0.01828, \text{ passed back from T=1}$$

$$\delta out_0 = \Delta_0 + \Delta out_0 = 0.03631 + -0.01828 = 0.01803$$

$$\delta state_0 = \delta out_0 \odot o_0 \odot (1 - \tanh^2(state_0)) + \delta state_1 \odot f_1 = 0.01803 \times 0.81757 \times (1 - \tanh^2(0.78572)) + -0.07111 \times 0.87030 = -0.05349$$

$$\delta a_0 = \delta state_0 \odot i_0 \odot (1 - a_0^2) = -0.05349 \times 0.96083 \times (1 - 0.81775^2) = -0.01703$$

$$\delta i_0 = \delta state_0 \odot a_0 \odot i_0 \odot (1 - i_0) = -0.05349 \times 0.81775 \times 0.96083 \times (1 - 0.96083) = -0.00165$$

$$\delta f_0 = \delta state_0 \odot state_{-1} \odot f_0 \odot (1 - f_0) = -0.05349 \times 0 \times 0.85195 \times (1 - 0.85195) = 0$$

$$\delta o_0 = \delta out_0 \odot \tanh(state_0) \odot o_0 \odot (1 - o_0) = 0.01803 \times \tanh(0.78572) \times 0.81757 \times (1 - 0.81757) = 0.00176$$

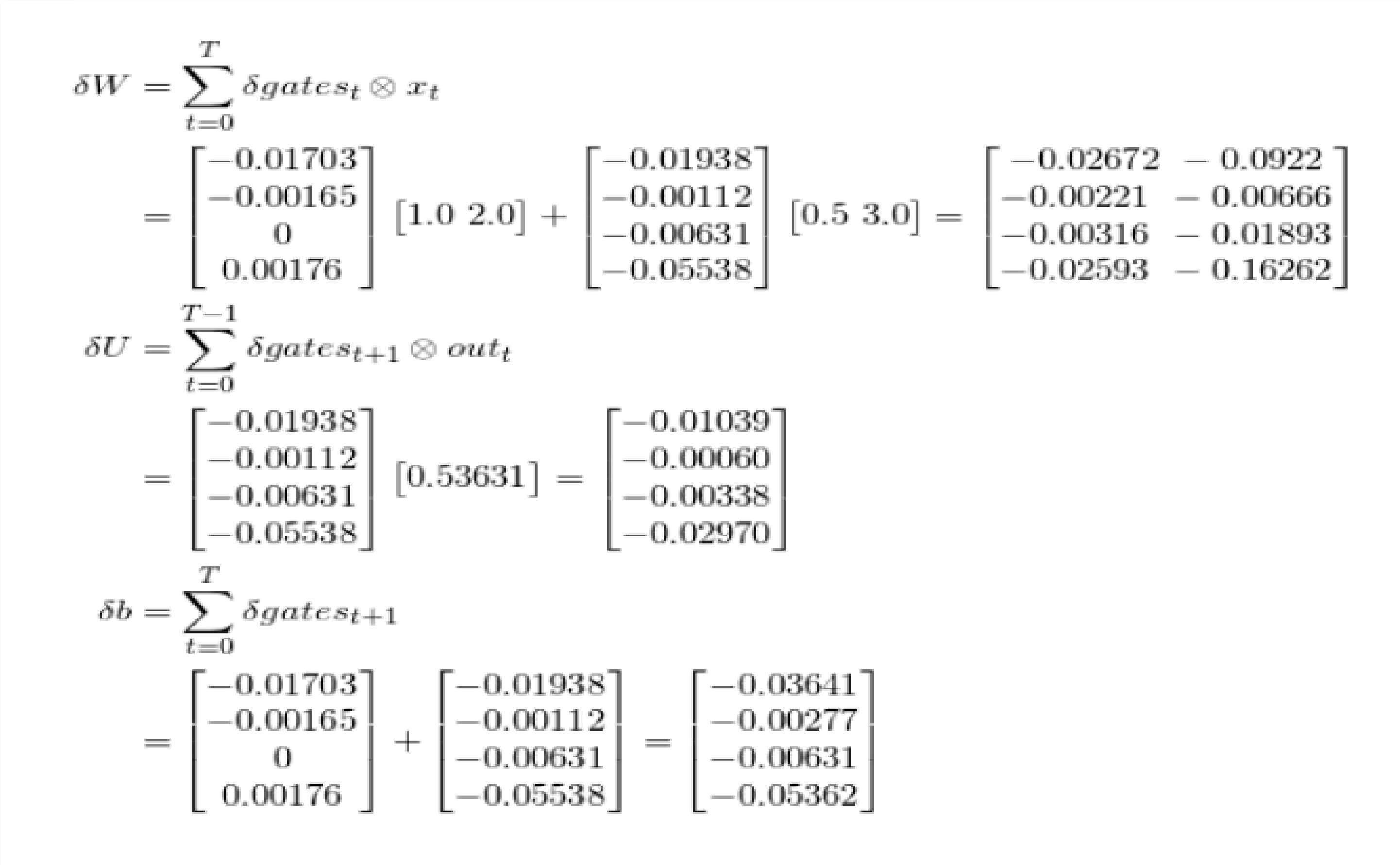
$$\delta x_0 = W^T \cdot \delta gates_0$$

$$= \begin{bmatrix} 0.45\ 0.95\ 0.70\ 0.60 \\ 0.25\ 0.80\ 0.45\ 0.40 \end{bmatrix} \begin{bmatrix} -0.01703 \\ -0.00165 \\ 0 \\ 0.00176 \end{bmatrix} = \begin{bmatrix} -0.00817 \\ -0.00487 \end{bmatrix}$$

$$\Delta out_{-1} = U^T \cdot \delta gates_1$$

$$= \begin{bmatrix} 0.15\ 0.80\ 0.10\ 0.25 \end{bmatrix} \begin{bmatrix} -0.01703 \\ -0.00165 \\ 0 \\ 0.00176 \end{bmatrix} = -0.00343$$

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (14)



$$\delta W = \sum_{t=0}^{T} \delta gates_t \otimes x_t$$

$$= \begin{bmatrix} -0.01703 \\ -0.00165 \\ 0 \\ 0.00176 \end{bmatrix} [1.0\ 2.0] + \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix} [0.5\ 3.0] = \begin{bmatrix} -0.02672 & -0.0922 \\ -0.00221 & -0.00666 \\ -0.00316 & -0.01893 \\ -0.02593 & -0.16262 \end{bmatrix}$$

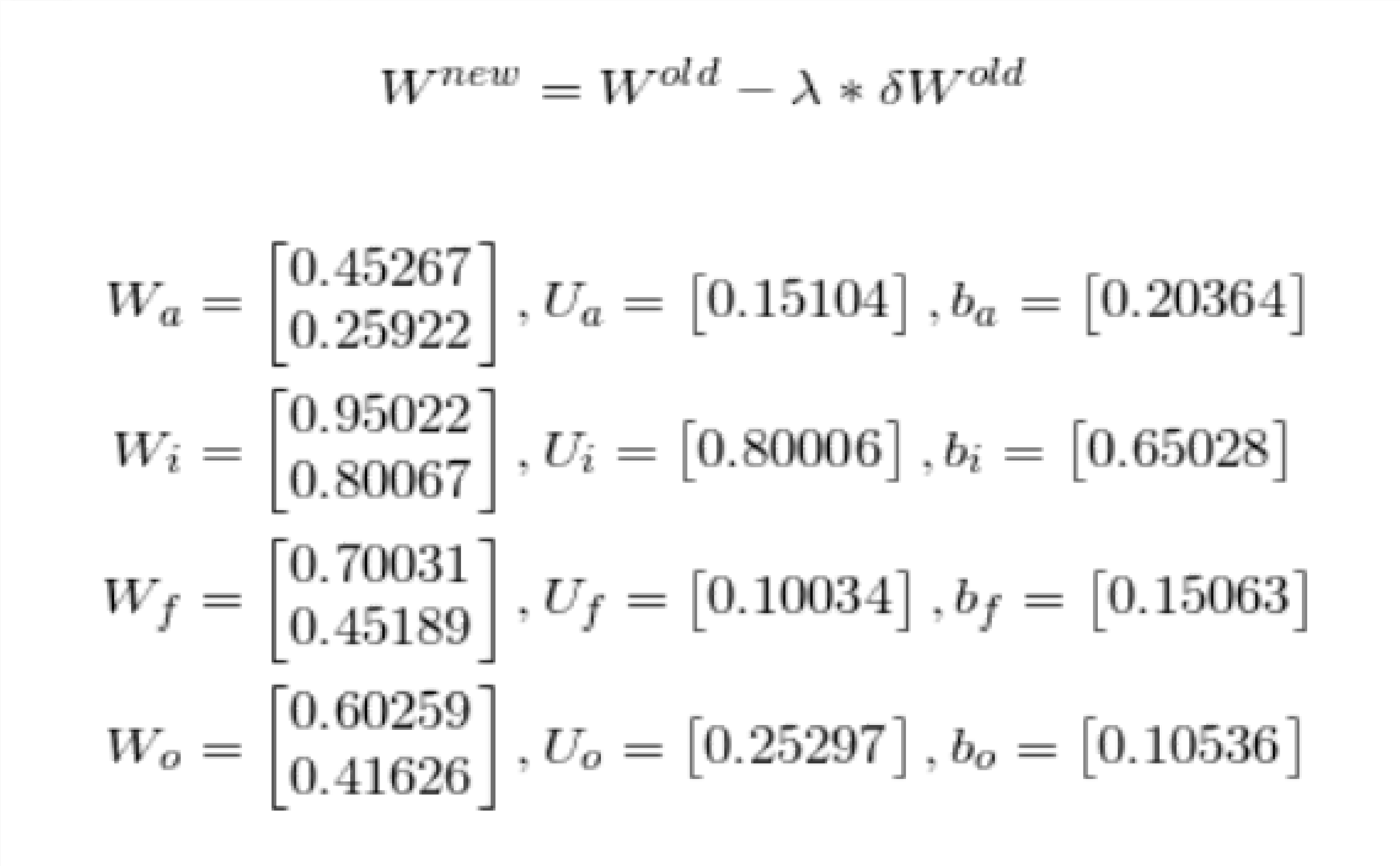
$$\delta U = \sum_{t=0}^{T-1} \delta gates_{t+1} \otimes out_t$$

$$= \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix} [0.53631] = \begin{bmatrix} -0.01039 \\ -0.00060 \\ -0.00338 \\ -0.02970 \end{bmatrix}$$

$$\delta b = \sum_{t=0}^{T} \delta gates_{t+1}$$

$$= \begin{bmatrix} -0.01703 \\ -0.00165 \\ 0 \\ 0.00176 \end{bmatrix} + \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix} = \begin{bmatrix} -0.03641 \\ -0.00277 \\ -0.00631 \\ -0.05362 \end{bmatrix}$$

Method: RNN/LSTM – Bagaimana Proses Perhitungan RNN/LSTM? (15)



$$W^{new} = W^{old} - \lambda * \delta W^{old}$$

$$W_a = \begin{bmatrix} 0.45267 \\ 0.25922 \end{bmatrix}, U_a = \begin{bmatrix} 0.15104 \end{bmatrix}, b_a = \begin{bmatrix} 0.20364 \end{bmatrix}$$

$$W_i = \begin{bmatrix} 0.95022 \\ 0.80067 \end{bmatrix}, U_i = \begin{bmatrix} 0.80006 \end{bmatrix}, b_i = \begin{bmatrix} 0.65028 \end{bmatrix}$$

$$W_f = \begin{bmatrix} 0.70031 \\ 0.45189 \end{bmatrix}, U_f = \begin{bmatrix} 0.10034 \end{bmatrix}, b_f = \begin{bmatrix} 0.15063 \end{bmatrix}$$

$$W_o = \begin{bmatrix} 0.60259 \\ 0.41626 \end{bmatrix}, U_o = \begin{bmatrix} 0.25297 \end{bmatrix}, b_o = \begin{bmatrix} 0.10536 \end{bmatrix}$$

Full process check this: https://medium.com/@aidangomez/let-s-do-this-f9b699de31d9QTib8q

DEMO

Sekian

Contact:
089620693521
Moch Ari Nasichuddin – FB/Linkedin
@AryNas04 – Twitter/IG