Stochastic Process

Markov Recurrent Neural Networks

Handwriting Recognition

Ridho Gani
Ferlinda Feliana
A.R Duto Pamungkas

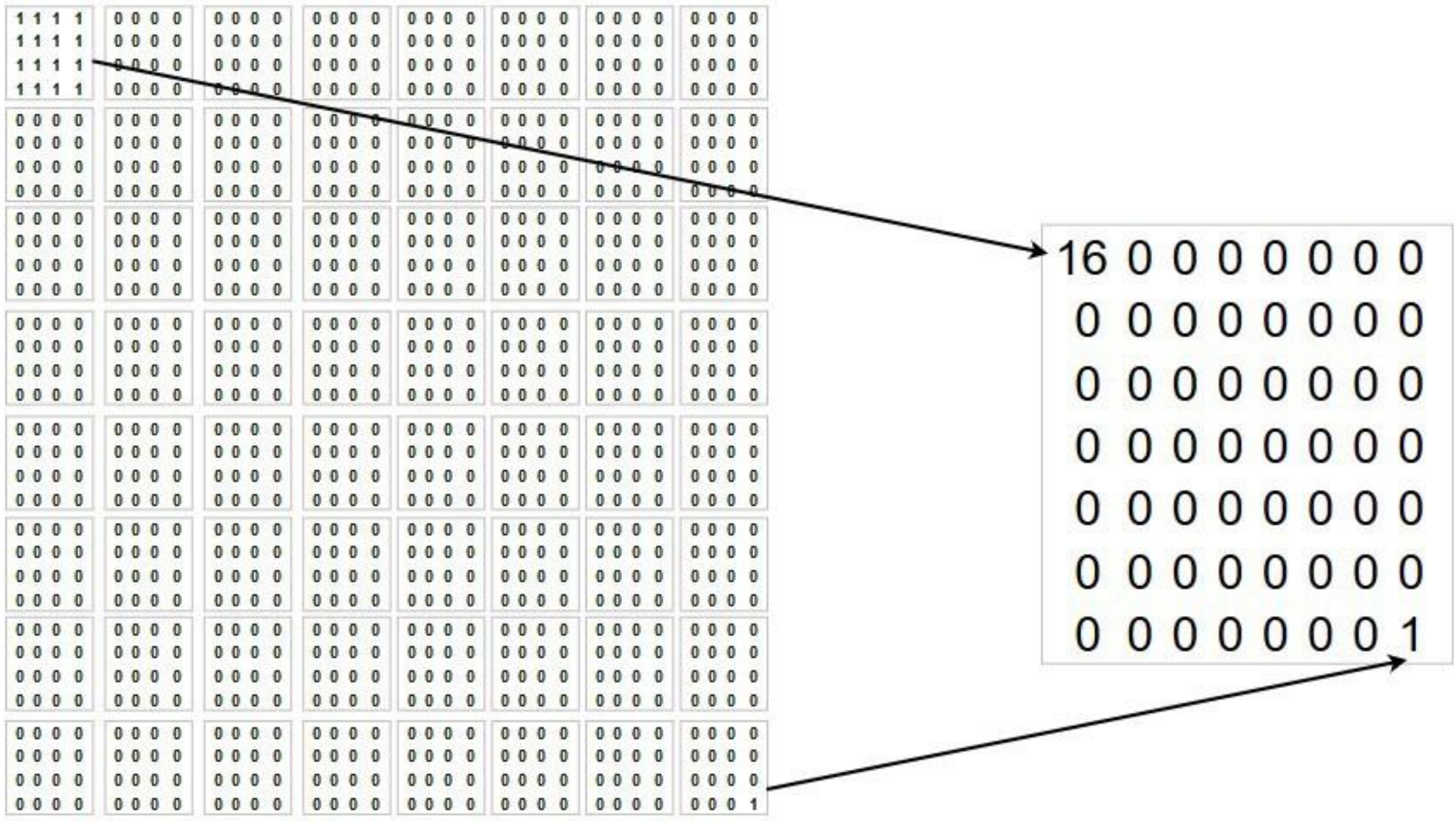
Dataset

MNIST

*modified national institute of standards and technology database

Data Processing

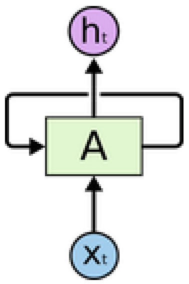
MNIST



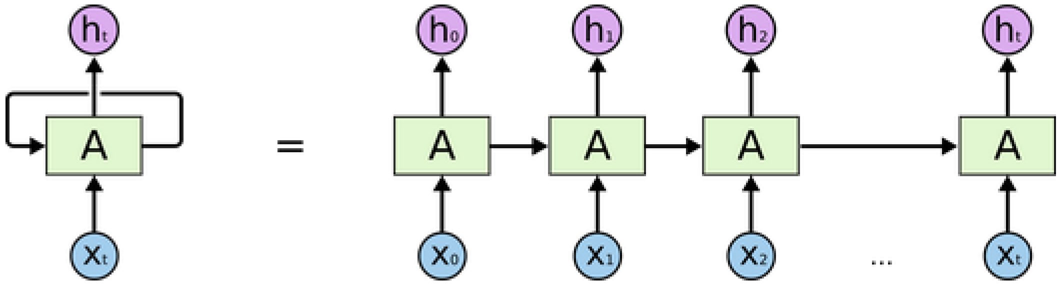
*modified national institute of standards and technology database

Data Processing

RNN



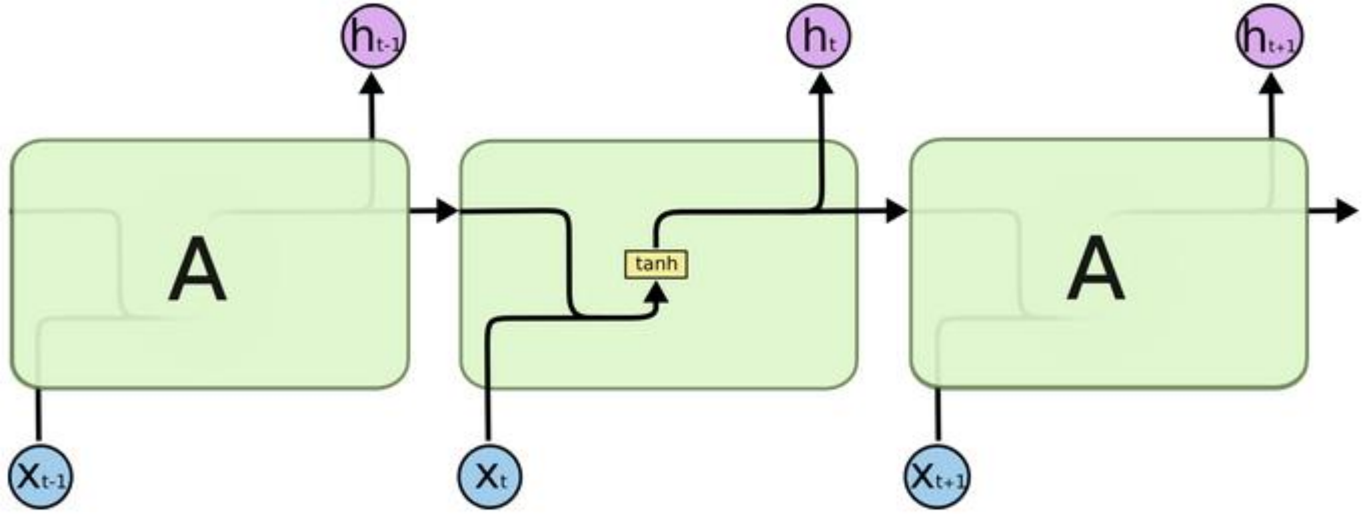
Recurrent Neural Networks have loops.



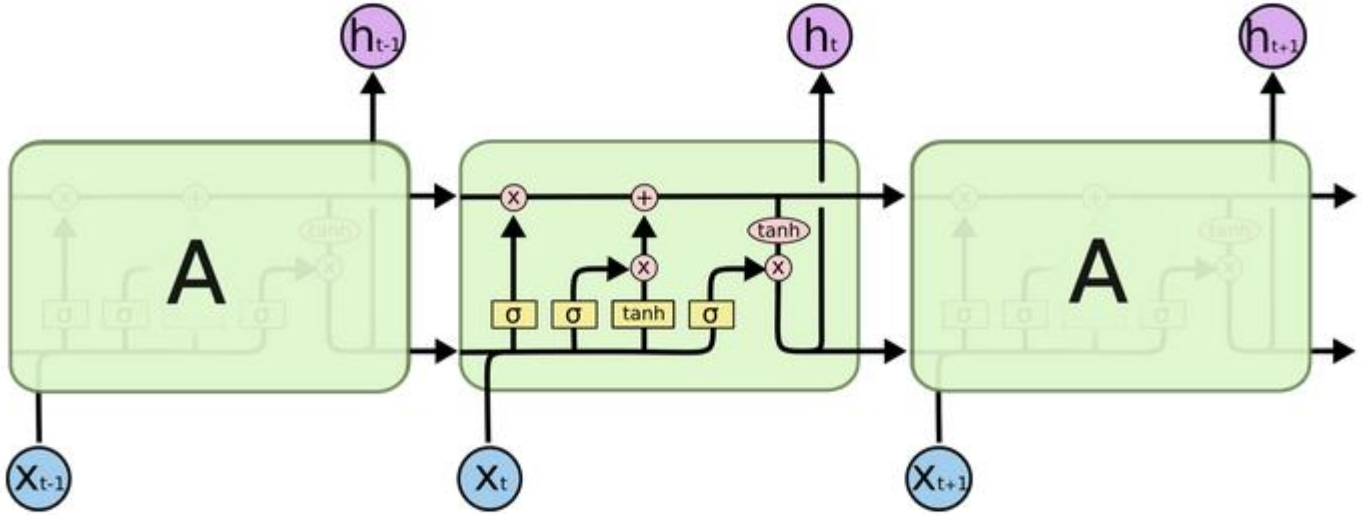
An unrolled recurrent neural network.

Data Processing

LSTM

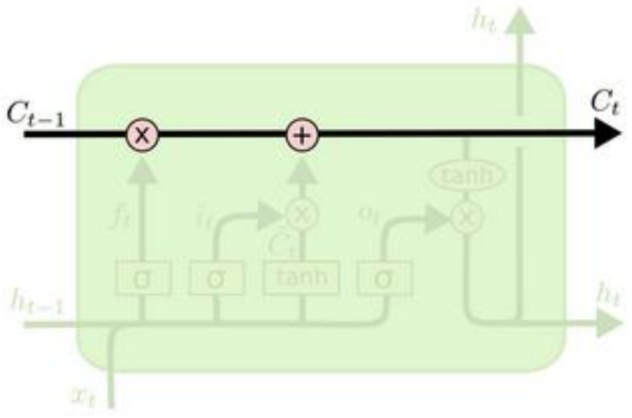


The repeating module in a standard RNN contains a single layer.

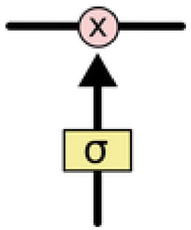


The repeating module in an LSTM contains four interacting layers.

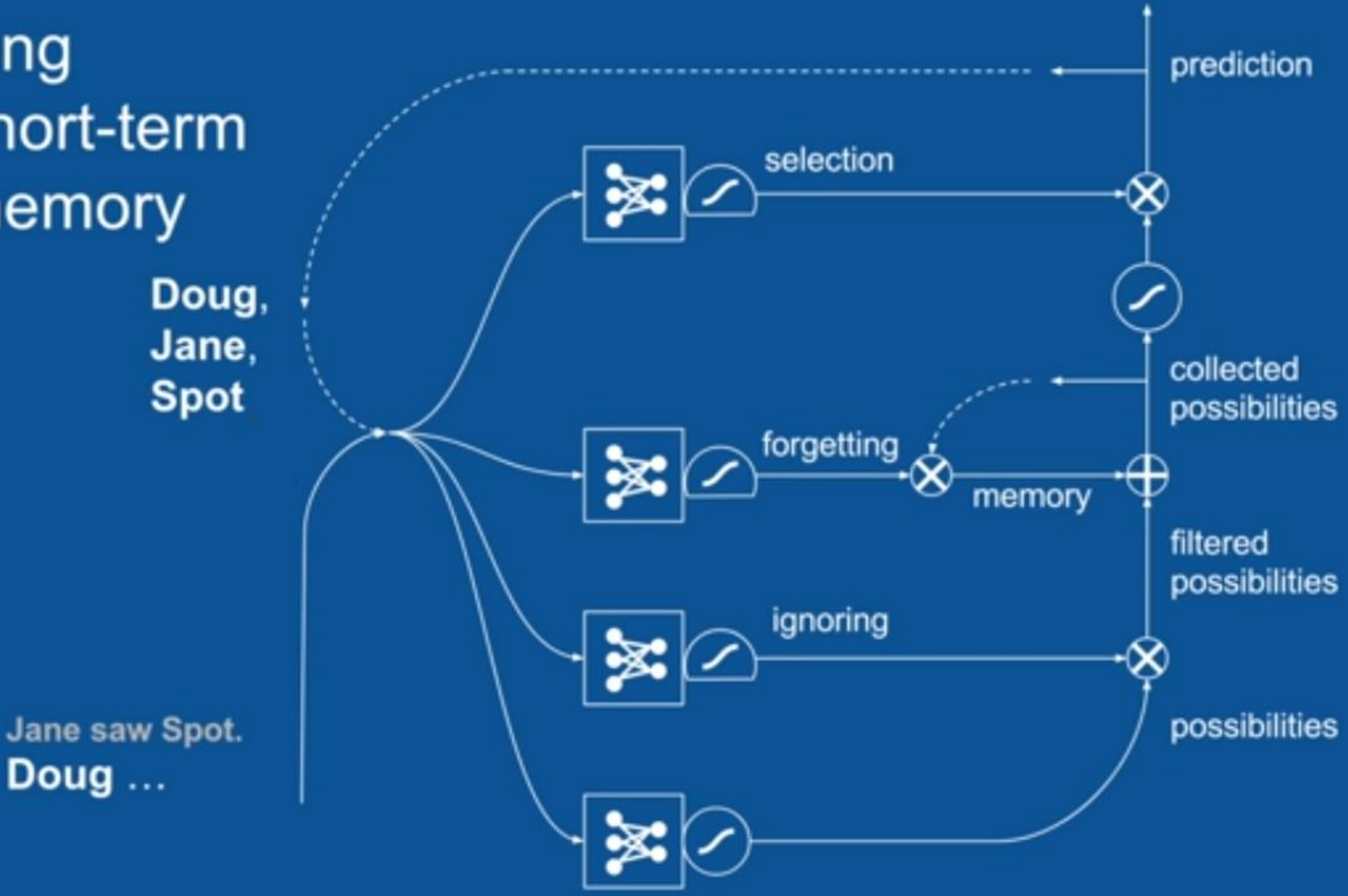
Data Processing



LSTM Key:
cell states

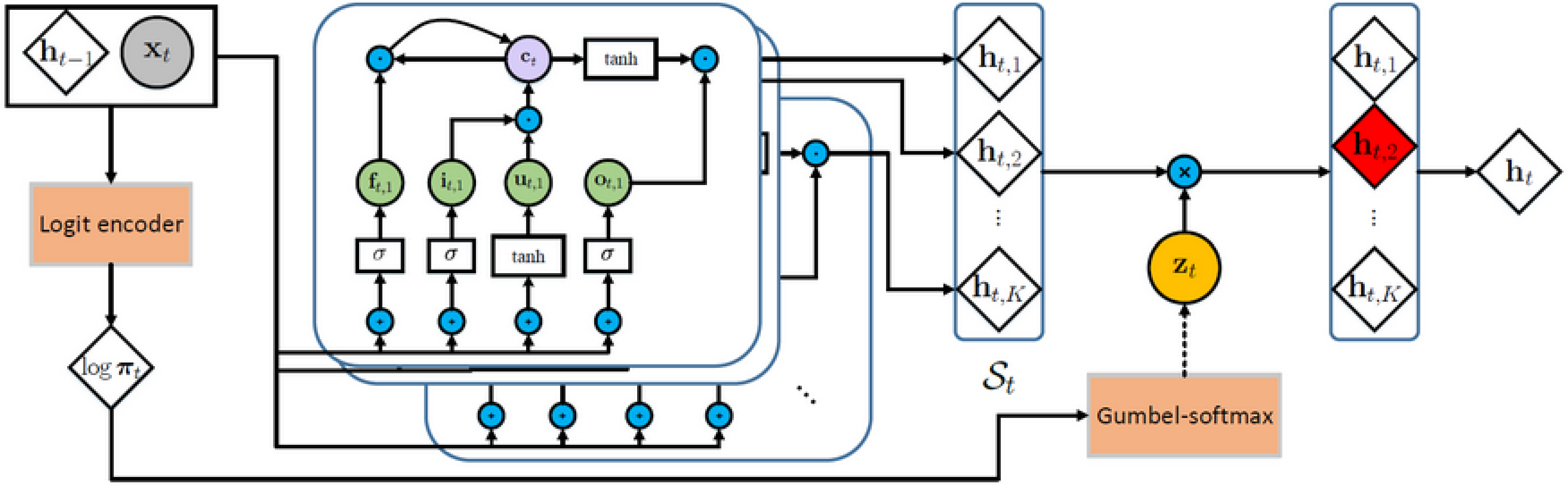


long short-term memory



Data Processing

MRNN



MARKOV CHAIN

MRNN is constructed with a state space which allows stochastic transition among multiple states {ht1, . . . , htK} at each time t. Transition of a stochastic state zt in MRNN complies with the property of Markov chain

$$p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) = p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{x}_t).$$

The probability of choosing a state $\mathbf{z}_t$ depends only on previous state $\mathbf{z}_{t-1}$ and current input $\mathbf{x}_t$.

MARKOV CHAIN

Stochastic state variable $\mathbf{z}_t = \{z_{tk}\}_{k=1}^{K}$ consisting of binary indicator $z_{tk} \in \{0, 1\}$ for each state k. The state space at each time t consists of all deterministic states $\{\mathbf{h}_{t1}, \ldots, \mathbf{h}_{tK}\}$ as basis vectors

$$\mathcal{S}_t \triangleq \begin{bmatrix} \mathbf{h}_{t1}^{\top} \\ \mathbf{h}_{t2}^{\top} \\ \vdots \\ \mathbf{h}_{tK}^{\top} \end{bmatrix} = \begin{bmatrix} \mathrm{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t, \boldsymbol{\Theta}_1) \\ \mathrm{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t, \boldsymbol{\Theta}_2) \\ \vdots \\ \mathrm{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t, \boldsymbol{\Theta}_K) \end{bmatrix}$$

Each hidden state $\mathbf{h}_{tk}$ is encoded by an individual LSTM using the previous state $\mathbf{h}_{t-1}$, the current input $\mathbf{x}_t$ and the corresponding parameters $\boldsymbol{\Theta}_k$.

MARKOV CHAIN

For simplify this problem, let's look this one

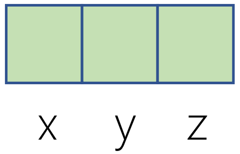
$$[next\ state] = [matrix\ of\ transition\ probabilities][current\ state]$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x0 & y0 & z0 \\ x1 & y1 & z1 \\ x2 & y2 & z2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

Data Processing

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x0 & y0 & z0 \\ x1 & y1 & z1 \\ x2 & y2 & z2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Pixel



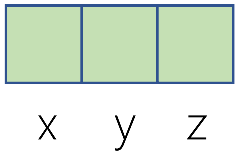
Transition Probability

X	X	X0
X	Y	Y0
X	Z	Z0
Y	Y	Y1
Y	X	X1
Y	Z	Z1
Z	Z	Z2
Z	Y	Y2
Z	X	X2

Data Processing

$$\begin{array}{cccc} \text{next state} & & \text{matrix of transition probabilities} & \text{current state} \\ \begin{bmatrix} a \\ b \\ c \end{bmatrix} & = & \begin{bmatrix} x0 & y0 & z0 \\ x1 & y1 & z1 \\ x2 & y2 & z2 \end{bmatrix} & \begin{bmatrix} x \\ y \\ z \end{bmatrix} \end{array}$$

Pixel



Transition Probability

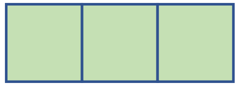
X	X	80%
X	Y	10%
X	Z	10%
Y	Y	70%
Y	X	20%
Y	Z	10%
Z	Z	60%
Z	Y	30%
Z	X	10%

Data Processing

next state   matrix of transition probabilities   current state

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x0 & y0 & z0 \\ x1 & y1 & z1 \\ x2 & y2 & z2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Pixel

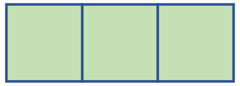


x y z

Transition Probability

From	To	value
X	X	80%
X	Y	10%
X	Z	10%
Y	Y	70%
Y	X	20%
Y	Z	10%
Z	Z	60%
Z	Y	30%
Z	X	10%

Pixel



x y z

Then, value for each pixel are
(value of pixel are determined by how much the ink fill that pixel)
For this example,

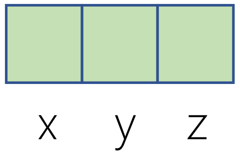
Pixel x = 200
Pixel y = 120
Pixel z = 180

Data Processing

next state — matrix of transition probabilities — current state

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x0 & y0 & z0 \\ x1 & y1 & z1 \\ x2 & y2 & z2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

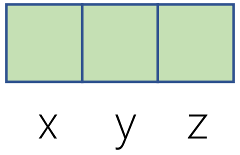
Pixel



Transition Probability

From	To	value
X	X	80%
X	Y	10%
X	Z	10%
Y	Y	70%
Y	X	20%
Y	Z	10%
Z	Z	60%
Z	Y	30%
Z	X	10%

Pixel



Pixel x = 200
Pixel y = 120
Pixel z = 180

After that, every pixel must be devided by sum of all pixel value to get the probability value of pixel fragment

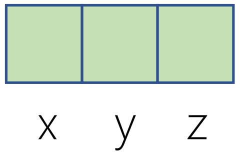
Pixel x = 200/500 = 0.40
Pixel y = 120/500 = 0.24
Pixel z = 180/500 = 0.36

Data Processing

next state, matrix of transition probabilities, current state

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 & 0.1 \\ 0.1 & 0.7 & 0.3 \\ 0.1 & 0.1 & 0.6 \end{bmatrix} \begin{bmatrix} 0.40 \\ 0.24 \\ 0.36 \end{bmatrix}$$

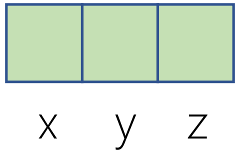
Pixel



Transition Probability

From	To	value
X	X	80%
X	Y	10%
X	Z	10%
Y	Y	70%
Y	X	20%
Y	Z	10%
Z	Z	60%
Z	Y	30%
Z	X	10%

Pixel



Pixel x = 200
Pixel y = 120
Pixel z = 180

After that, every pixel must be devided by sum of all pixel value to get the probability value of pixel fragment

Pixel x = 200/500 = 0.40
Pixel y = 120/500 = 0.24
Pixel z = 180/500 = 0.36

Data Processing

$$\begin{array}{cccc} \text{next state} & & \text{matrix of transition probabilities} & \text{current state} \\ \begin{bmatrix} a \\ b \\ c \end{bmatrix} & = & \begin{bmatrix} 0.8 & 0.2 & 0.1 \\ 0.1 & 0.7 & 0.3 \\ 0.1 & 0.1 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.40 \\ 0.24 \\ 0.36 \end{bmatrix} \end{array}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0.404 \\ 0.316 \\ 0.280 \end{bmatrix}$$

Well done! we get the probability of next state
Then, we can give this next state data to RNN model
to get the prediction what the digit is

Data Processing

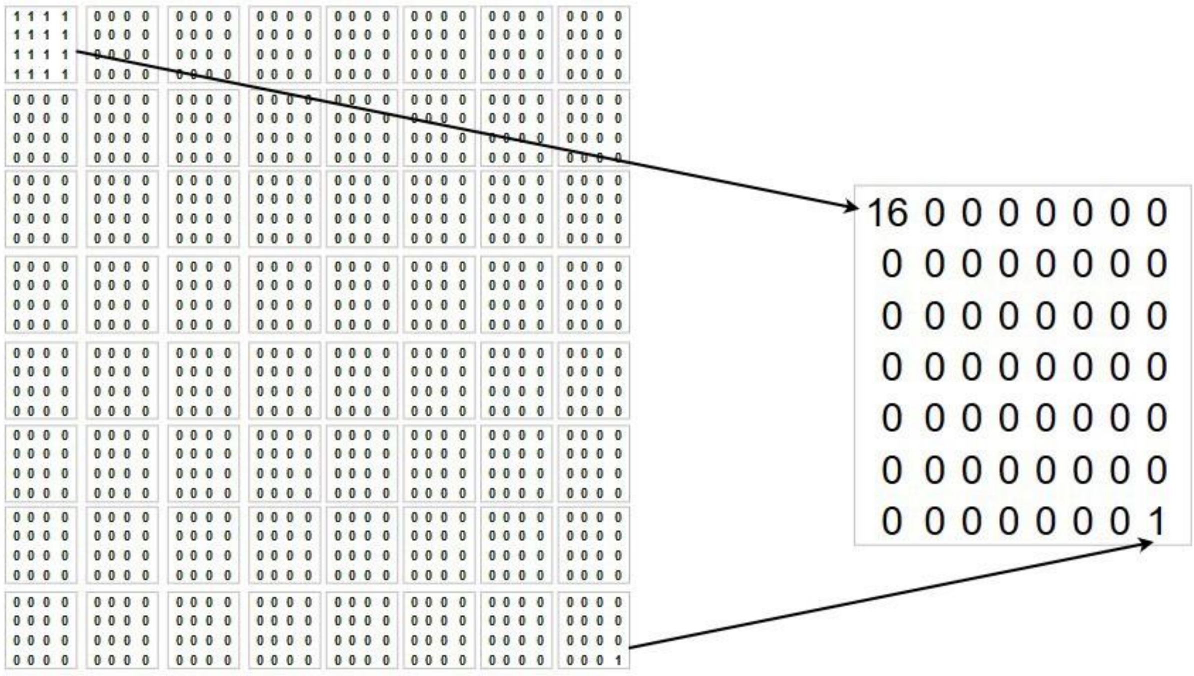
$$\overset{\text{next state}}{\begin{bmatrix} a \\ b \\ c \end{bmatrix}} = \overset{\text{matrix of transition probabilities}}{\begin{bmatrix} 0.8 & 0.2 & 0.1 \\ 0.1 & 0.7 & 0.3 \\ 0.1 & 0.1 & 0.6 \end{bmatrix}} \overset{\text{current state}}{\begin{bmatrix} 0.40 \\ 0.24 \\ 0.36 \end{bmatrix}}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0.404 \\ 0.316 \\ 0.280 \end{bmatrix}$$

Prediction is based on MNIST dataset. If prediction value equal 1 or almost 1, it means good/accurate.

Data Processing

All this example is looks simply because we only use 3-pixel 😅



And it will become a nightmare when we use real case 64-pixel.
That’s way, we also use LSTM to determine probability value of next state

Data Processing

Overall probability of which LSTM being chosen