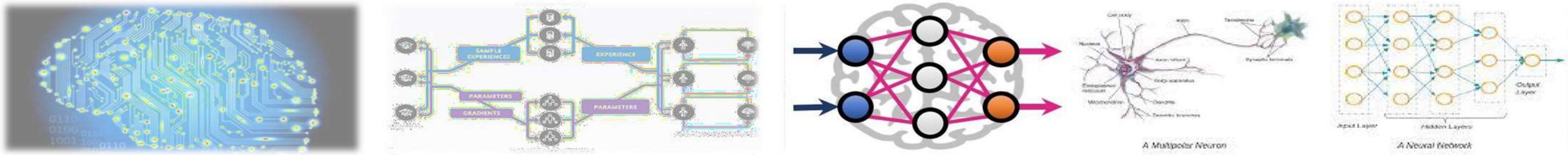


Comparison LSTM, GRU, CNN network

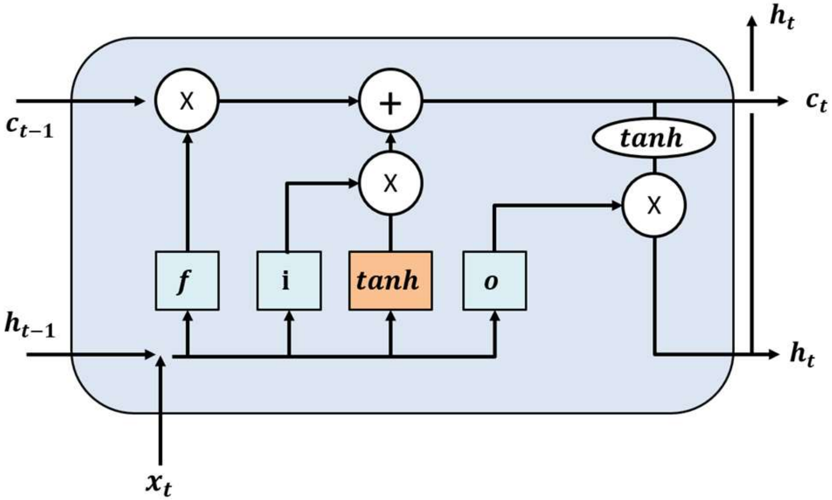
LSTM, GRU, CNN 모델 구현 및 실험

Geonil Yun,
rjsdlf45@gmail.com
2021. 07. 06.

Introduction

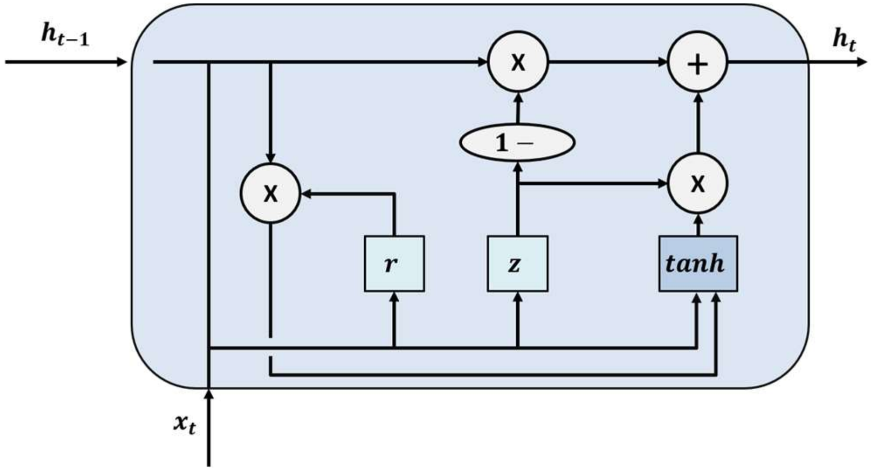
- **Gated Recurrent Unit (GRU)**
  - **Motivated by Long-Short Term Memory(LSTM)**
  - **Simple to compute and implement : less number of parameters**

LSTM Cell



- 3 gates (forget, input, output)
- output cell state, hidden state

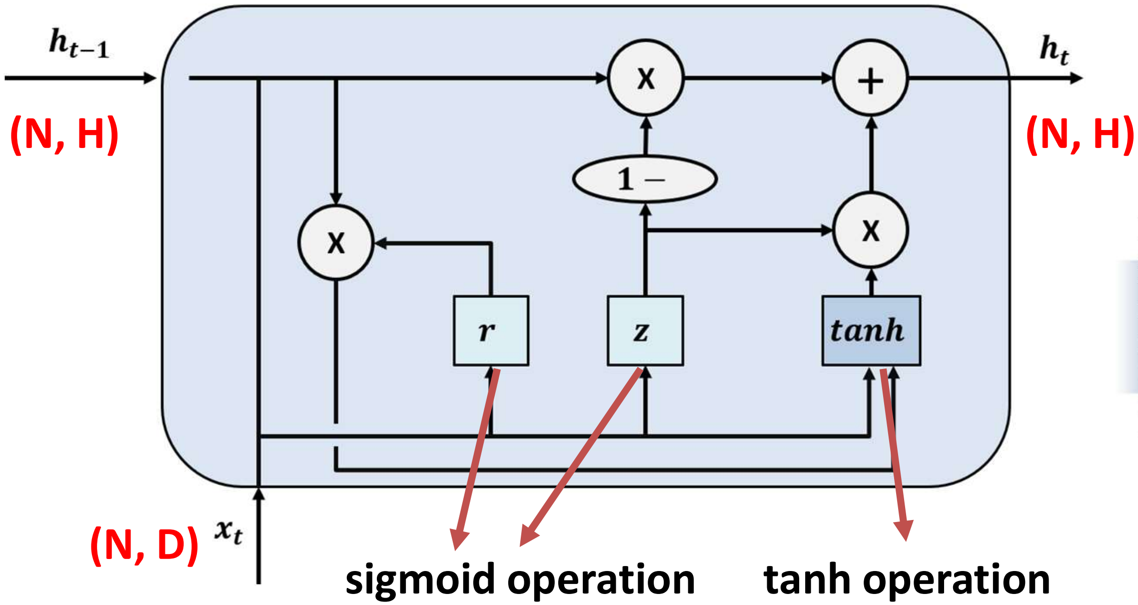
GRU Cell



- 2 gates (reset, update)
- output only hidden state

Proposed Method

GRU Architecture



$reset\ gate(r)$
$update\ gate(z)$

N : samples (mini-batch)
D : input vector dimension
H : hidden vector dimension

- Weights and bias (parameters)
  [$W_x^{(r)}$, $W_h^{(r)}$, $b^{(r)}$] for reset gate
  [$W_x^{(z)}$, $W_h^{(z)}$, $b^{(z)}$] for update gate
  [$W_x^{(h)}$, $W_h^{(h)}$, $b^{(h)}$] for current hidden state

- Equations of GRU computation

$$r = \sigma(x_t W_x^{(r)} + h_{t-1} W_h^{(r)} + b^{(r)})$$
$$z = \sigma(x_t W_x^{(z)} + h_{t-1} W_h^{(z)} + b^{(z)})$$
$$\tilde{h} = tanh(x_t W_x^{(h)} + (h_{t-1} \odot r) W_h^{(h)} + b^{(h)})$$
$$h_t = z \odot \tilde{h} + (1 - z) \odot h_{t-1}$$

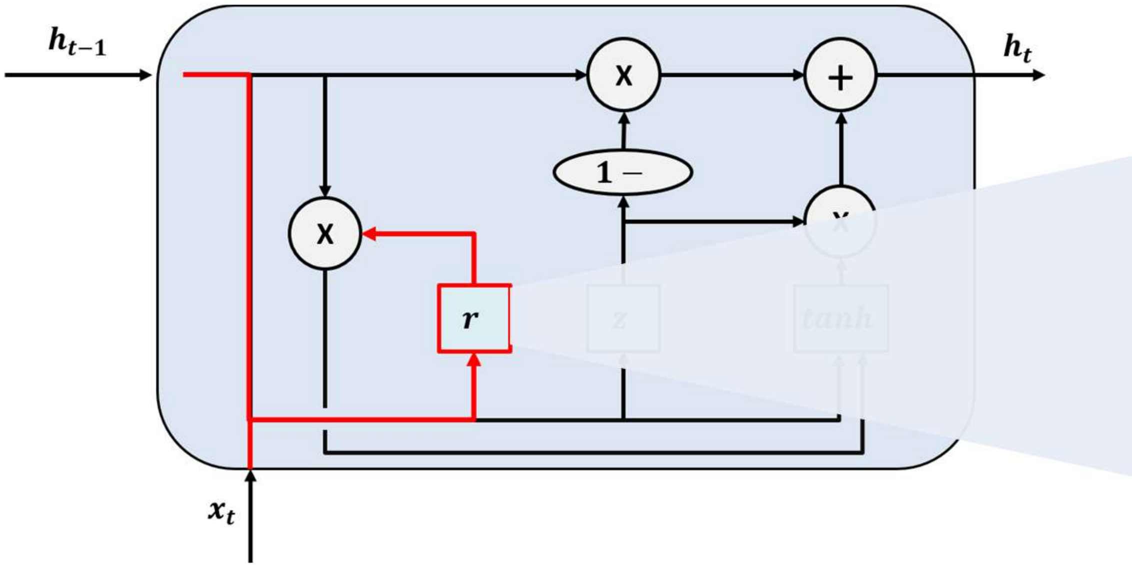
Proposed Method

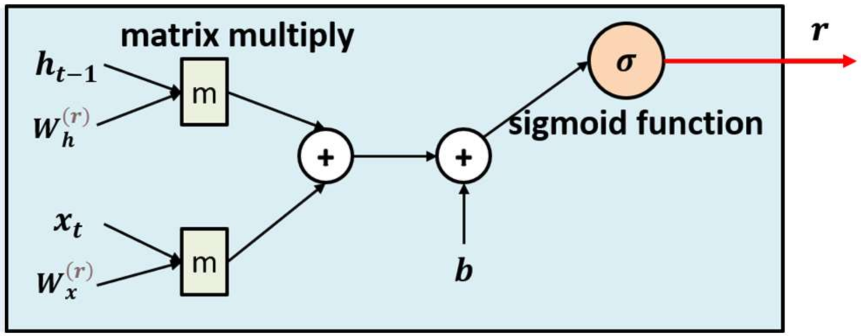
- **Gated Recurrent Unit (GRU)**

  $\boldsymbol{reset\ gate(r)}$ **: ignore previous hidden state in new hidden state(0 ~ 1)**

Weights and bias

$$r = \sigma(x_t\boxed{W_x^{(r)}} + h_{t-1}\boxed{W_h^{(r)}} + \boxed{b^{(r)}})$$



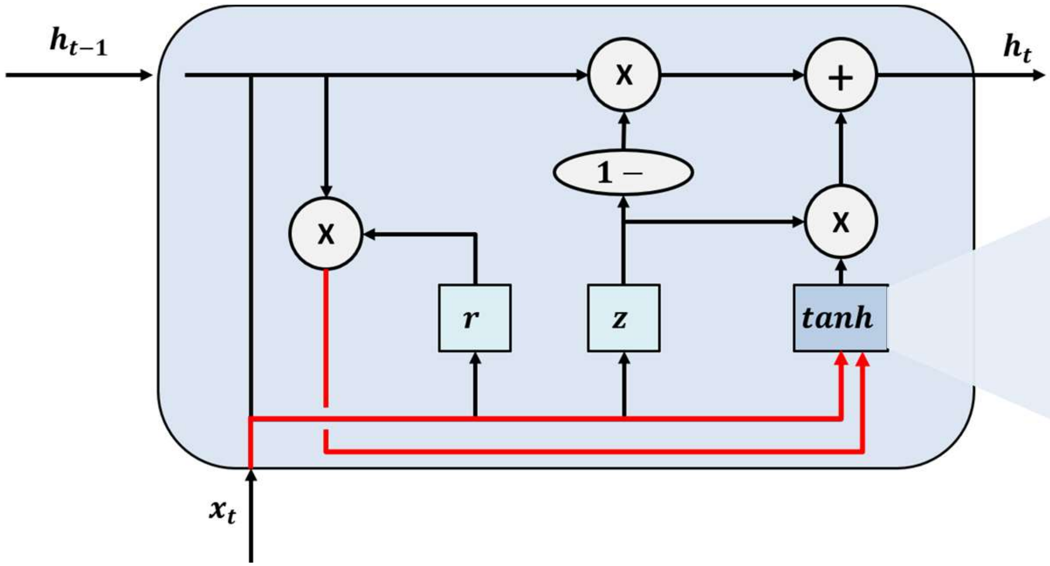


- Output of $\boldsymbol{reset\ gate(r)}$ is used as input of $\boldsymbol{new\ hidden\ state(\tilde{h})}$ with $\boldsymbol{previous\ hidden\ state(h_{t-1})}$

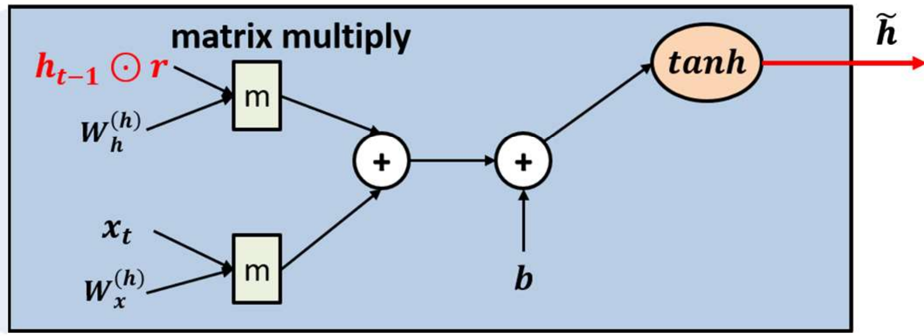
Proposed Method

- **Gated Recurrent Unit (GRU)**

$\boldsymbol{new\ hidden\ state(\tilde{h})}$ **: made up of input and previous hidden state**



$$\tilde{h} = tanh(x_t \boxed{W_x^{(h)}} + (h_{t-1} \odot r)\boxed{W_h^{(h)}} + \boxed{b^{(h)}})$$



- $\boldsymbol{new\ hidden\ state(\tilde{h})}$ is affected by $\boldsymbol{reset\ gate(r)}$

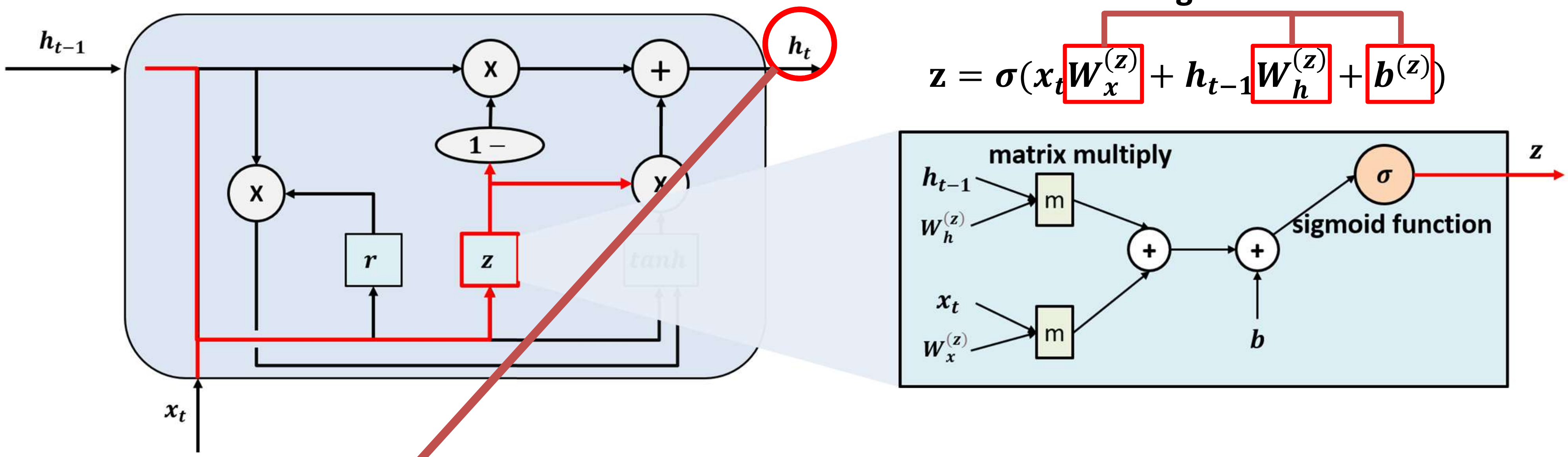
Proposed Method

- **Gated Recurrent Unit (GRU)**

$\boldsymbol{update\ gate(z)}$ **: control information of previous hidden state and new hidden state**

Weights and bias

$$\mathbf{z} = \boldsymbol{\sigma}(\boldsymbol{x_t}\boxed{\boldsymbol{W_x^{(z)}}} + \boldsymbol{h_{t-1}}\boxed{\boldsymbol{W_h^{(z)}}} + \boxed{\boldsymbol{b^{(z)}}})$$

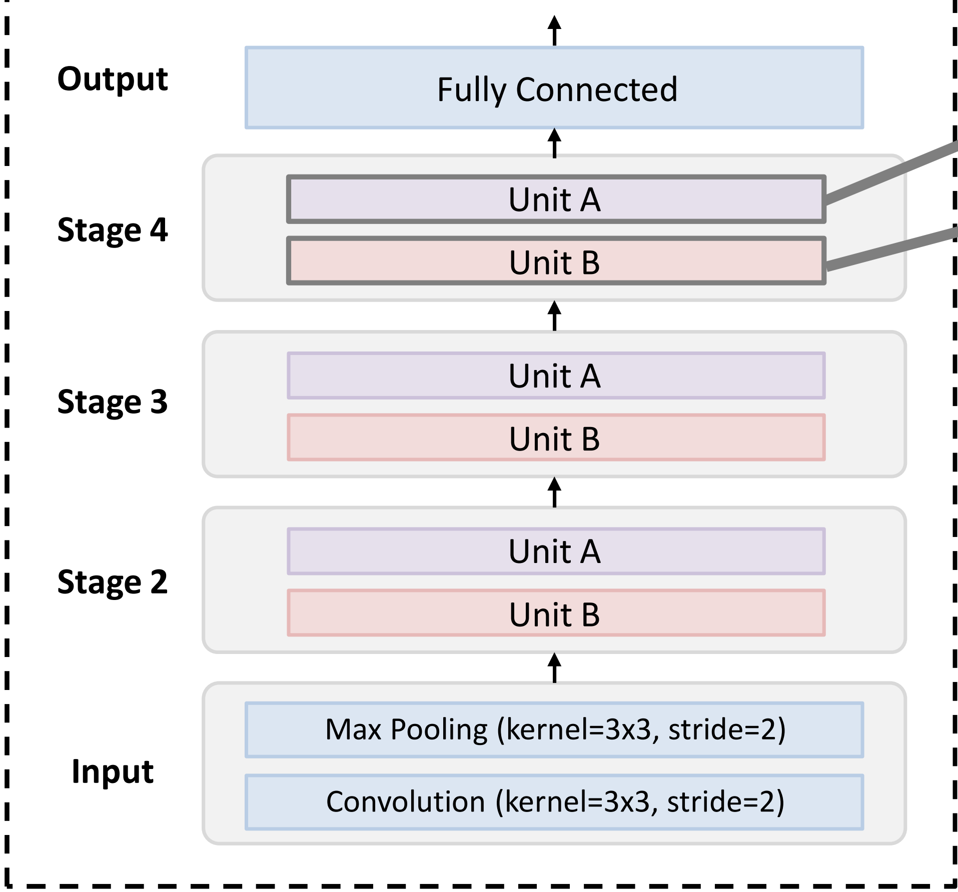


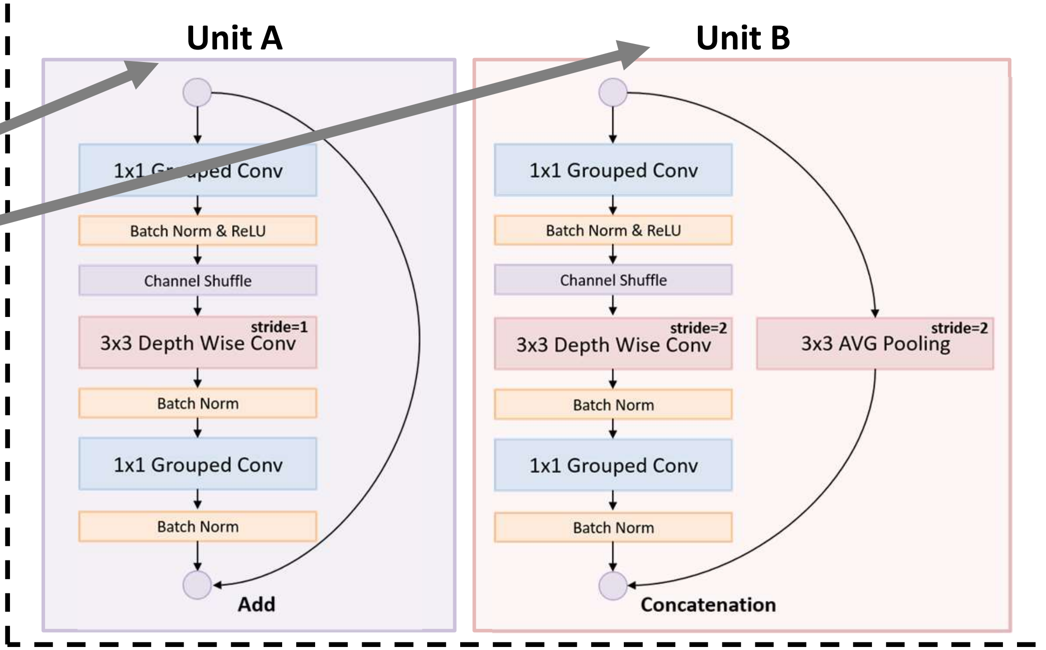
- $\boldsymbol{current\ hidden\ state(h)}$ is affected by $\boldsymbol{update\ gate(z)}$

$$\boldsymbol{h_t} = \mathbf{z} \odot \boldsymbol{\tilde{h}} + (\mathbf{1} - \mathbf{z}) \odot \boldsymbol{h_{t-1}}$$

Experimental setting

- **Model Structure**



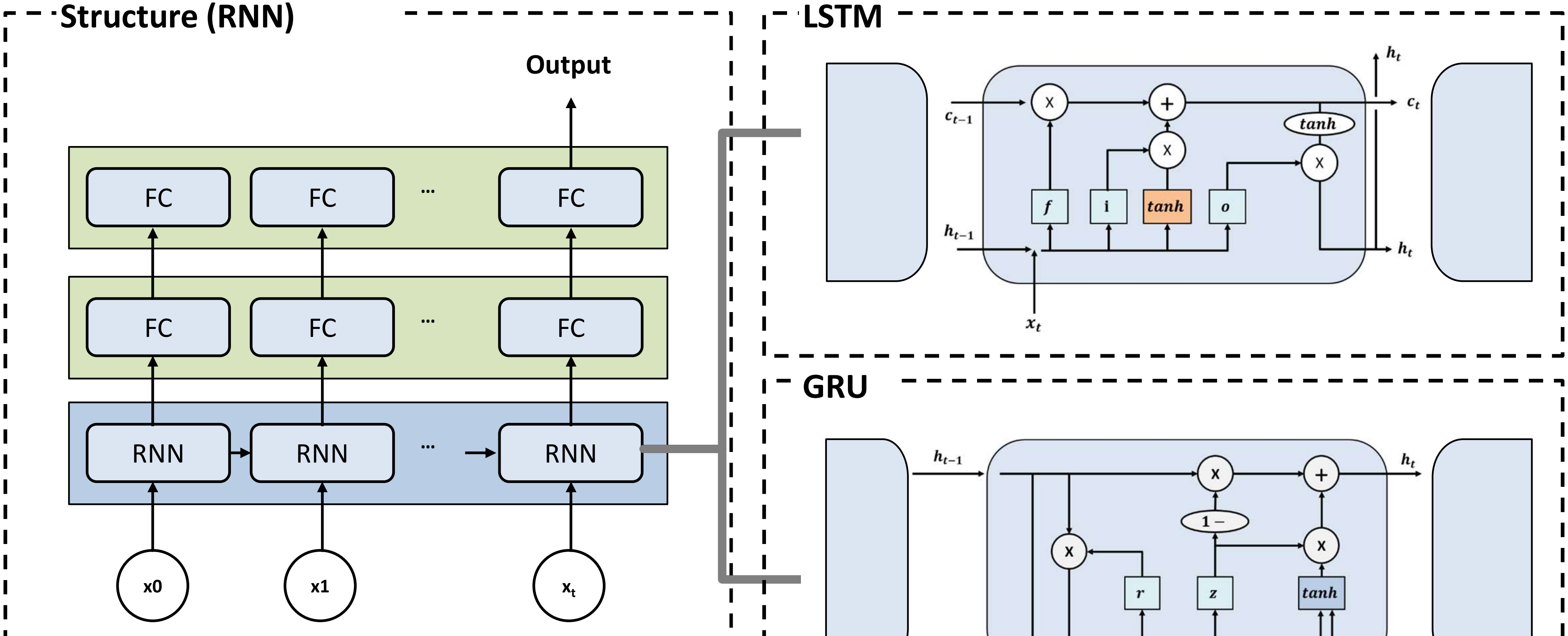


Parameter

	Stage2	Stage3	Stage4
Repeat Unit-A	**3**	**7**	**3**

Experimental setting

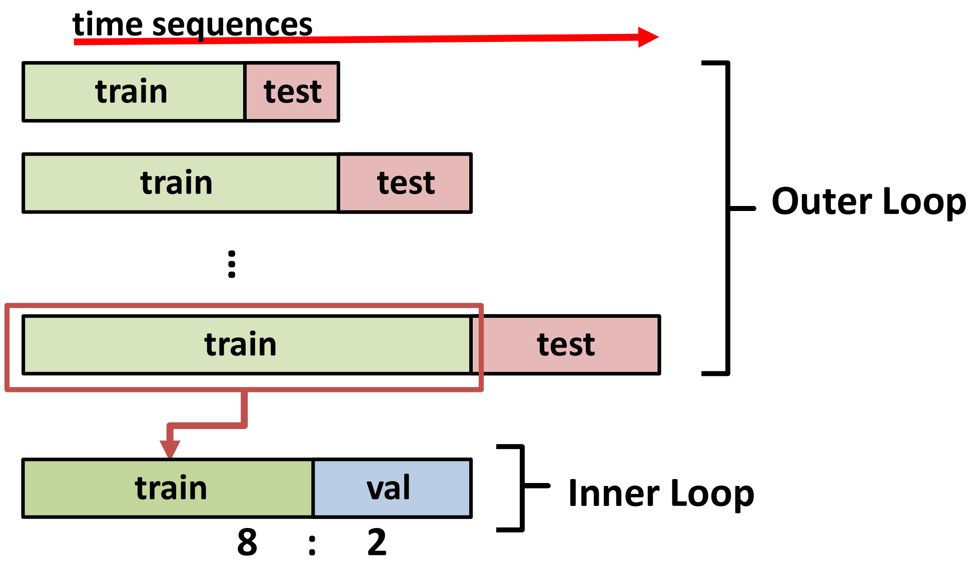
- **Model Structure**



- input data shape : (batch, time step, features)

Experimental setting

- **Nested Cross Validation for time series**



- ➢ **train past data & predict future data**
- ➢ **handling time series data in Nested CV**

- hyperparameter

Model	CNN (ShuffleNet)	RNN(LSTM, GRU)
fold num	10	
hidden size	64	
input length	24	
output length	1	
epoch	30	
batch size	32 or 64	
learning rate	**0.01, 0.001**	

- ➢ batch size 32 for **ISE, Chickenpox** dataset
- ➢ batch size 64 for others

Experimental setting

- **Statistics**

Dataset		Mean	Standard Deviation	Skewness	Kurtosis	Jarque Bera	ADF
Air Quality		1099.833	217.080	0.755	0.334	897.811	-9.823
Appliances Energy		97.694	102.524	3.386	13.664	191240.781	-21.616
Beijing Air Pollution		98.613	92.050	1.802	4.768	62162.743	-20.606
Hungarian Chickenpox		101.245	76.354	0.948	1.400	121.010	-6.933
Electric Consumption		1.483	1.194	1.141	1.732	3421.560	-5.360
Eye State		0.448	0.497	0.205	-1.957	2497.788	-4.803
Istanbul Stock Exchange	ISE	0.0016	0.016	-0.094	1.391	44.022	-22.746
	SP	0.0006	0.014	-0.090	3.068	211.013	-11.031
	DAX	0.0007	0.014	-0.106	2.104	99.894	-23.056
Metro Volume		3259.818	1986.860	-0.089	-1.309	3506.110	-28.016
Occupancy Detection		0.212	0.408	1.406	-0.020	2686.285	-3.430
SML 2010		20.493	3.312	-0.169	-0.105	21.792	-4.314

*ISE : Istanbul Stock Exchange, *SP : Standard & Poor's, *DAX : Germany

Comparison results

- Training time [s]

Data		CNN (ShuffleNet)	LSTM	GRU
Air Quality			137.668	
Appliances Energy			210.792	
Beijing Air Pollution			102.650	
Hungary Chickenpox		7.312	14.711	17.824
Electric Consumption			105.139	
Eye State			156.003	
Istanbul Stock Exchange	ISE	7.623	9.706	19.391
	SP	8.019	9.745	
	DAX	7.526	9.450	
Metro Interstate Traffic Volume				
Occupancy Detection		68.230	86.891	
SML 2010		33.614	43.461	

Comparison results

- RMSE (Root Mean Squared Error)

Data		CNN (ShuffleNet)	Standard Deviation	LSTM	Standard Deviation	GRU	Standard Deviation
Air Quality		145.833	25.764	104.545	24.832		
Appliances Energy		75.966	9.782	73.348	22.123		
Beijing Air Pollution		49.456	10.976	26.776	8.101		
Hungary Chickenpox		66.446	17.458	57.977	18.244	83.730	37.536
Electric Consumption				0.294	0.238		
Eye State		0.616	0.090	3.010	4.771		
Istanbul Stock Exchange	ISE	0.025	0.009	6.654	8.861	1.259	0.585
	SP	0.022	0.008	6.497	8.738		
	DAX	0.024	0.008	6.016	7.917		
Metro Interstate Traffic Volume				522.602	129.710		
Occupancy Detection		0.410	0.117	1.981	2.641		
SML 2010		2.695	0.962	0.285	0.192		

Comparison results

- MAE (Mean Absolute Error)

Data		CNN (ShuffleNet)	Standard Deviation	LSTM	Standard Deviation	GRU	Standard Deviation
Air Quality		111.932	22.299	80.097	24.090		
Appliances Energy		37.614	7.429	42.697	20.164		
Beijing Air Pollution		32.908	6.590	12.167	0.899		
Hungary Chickenpox		48.717	11.758	41.321	15.165	64.433	28.527
Electric Consumption				0.121	0.086		
Eye State		0.491	0.083	2.960	4.789		
Istanbul Stock Exchange	ISE	0.021	0.009	6.652	8.862	1.254	0.584
	SP	0.018	0.007	6.495	8.739		
	DAX	0.019	0.007	6.014	7.919		
Metro Interstate Traffic Volume				340.367	72.239		
Occupancy Detection		0.234	0.098	1.713	2.118		
SML 2010		2.474	1.004	0.233	0.148		

References

[1] : Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6848-6856).

[2] : Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Thank you

Appendix : Shuffle Net Architecture

Layer	Output size	KSize	Stride	Repeat	Output channels (g groups)				
					$g = 1$	$g = 2$	$g = 3$	$g = 4$	$g = 8$
Image	224×224				3	3	3	3	3
Conv1	112×112	3×3	2	1	24	24	24	24	24
MaxPool	56×56	3×3	2						
Stage2	28×28		2	1	144	200	240	272	384
	28×28		1	3	144	200	240	272	384
Stage3	14×14		2	1	288	400	480	544	768
	14×14		1	7	288	400	480	544	768
Stage4	7×7		2	1	576	800	960	1088	1536
	7×7		1	3	576	800	960	1088	1536
GlobalPool	1×1	7×7							
FC					1000	1000	1000	1000	1000
Complexity					143M	140M	137M	133M	137M

Appendix : Dataset

Dataset	references
Air Quality	https://archive.ics.uci.edu/ml/machine-learning-databases/00360/
Appliances Energy	https://archive.ics.uci.edu/ml/machine-learning-databases/00374/
Beijing Air Pollution	https://archive.ics.uci.edu/ml/machine-learning-databases/00381/
Hungarian Chickenpox	https://archive.ics.uci.edu/ml/machine-learning-databases/00580/
Electric Consumption	https://archive.ics.uci.edu/ml/machine-learning-databases/00235/
Eye State	https://archive.ics.uci.edu/ml/machine-learning-databases/00264/
Istanbul Stock Exchange	https://archive.ics.uci.edu/ml/machine-learning-databases/00247/
Metro Volume	https://archive.ics.uci.edu/ml/machine-learning-databases/00492/
Occupancy Detection	https://archive.ics.uci.edu/ml/machine-learning-databases/00357/
SML 2010	https://archive.ics.uci.edu/ml/machine-learning-databases/00274/

Appendix : Code

Model	view with code
LSTM	https://colab.research.google.com/drive/1oNW4HzBzmrjyWcWZbjFMtksP1vTV9IO2?usp=sharing
CNN	https://colab.research.google.com/drive/1Ccd7l65bJwiD2HW8XmOKTHO1PEUMuQI3?usp=sharing
LSTM-Beijing air pollution	https://colab.research.google.com/drive/1De0Tn_iD5NQBKAu_eKr3PWZwvcnNTu97?usp=sharing
CNN-Beijing air pollution	https://colab.research.google.com/drive/1HmjdyXbwUASZxzS40v4opc9myAuuhcrF?usp=sharing
LSTM-Hungarian chickenpox	https://colab.research.google.com/drive/1eeLLtCJGzQ2u_V09pXF0dHg2Oz4YHfxn?usp=sharing
CNN-Hungarian chickenpox	https://colab.research.google.com/drive/1E4PywHa-xACYhHTiMiHRwEkCHT9eBz4A?usp=sharing
LSTM-Power consumption	https://colab.research.google.com/drive/1HVLLJMPGJeixJ4H9O3IvzfUhUWp8DAXN?usp=sharing
CNN-Power consumption	https://colab.research.google.com/drive/1d7wu9u7zo_URAXYd2E01uROGcsdoSc87?usp=sharing