Predicting Financial Time Series using Deep Learning

Module2. Recurrent Neural Network

Jongho Kim

NICE Pricing & Information Inc.

Fall, 2018

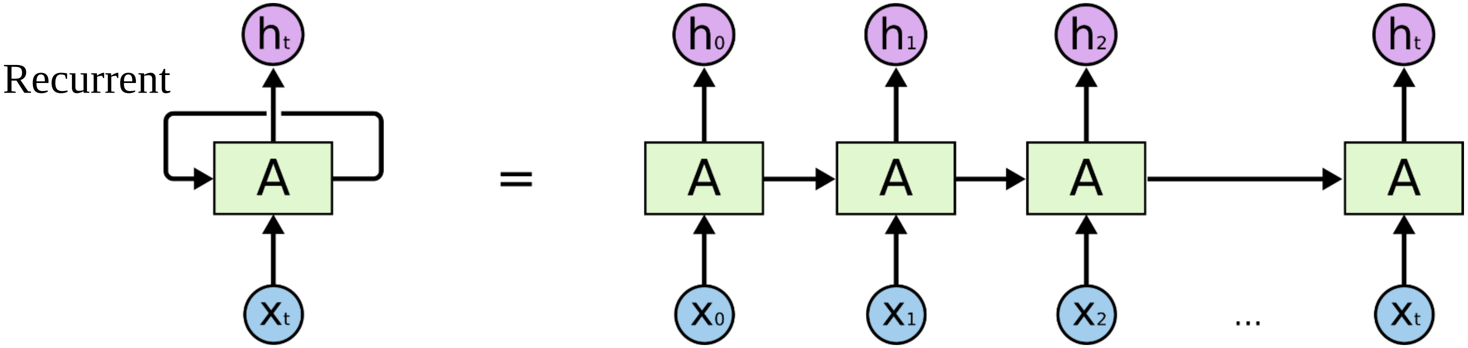
Note. This content mainly refers the summer session of KAIST organized by Jiyong Park(2018)

Recurrent Neural Network

Review: What is RNN?

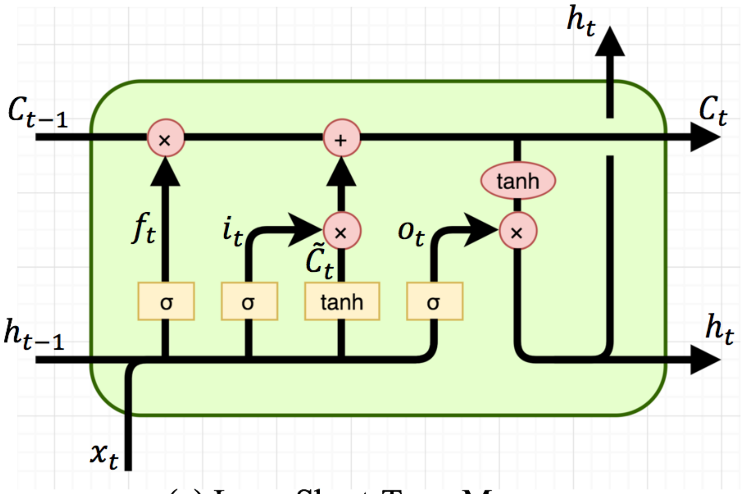
- Unlike CNN, RNN can use past information to learn the present task.
  - ➢ Example: Natural Language Processing (NLP)
    - "The clouds are in the (   )."
    - "I grew up in France …… I speak fluent (   )."
  - ➢ Vanishing gradient problem
    - As that gap grows, RNN becomes unable to learn to connect the information.

      (the past information would be vanishing or exploding)

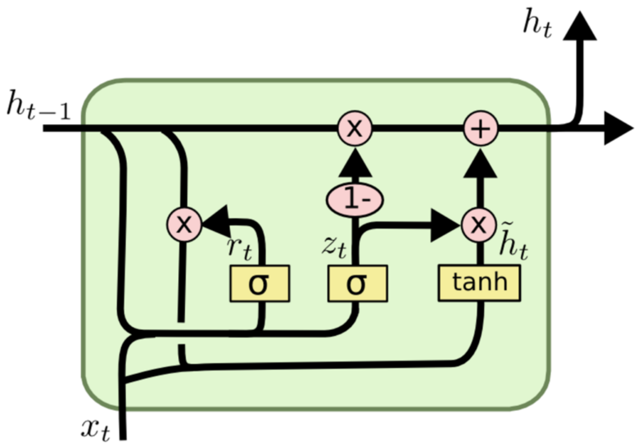


Review: What is RNN?

- Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) allows RNN to learn how much past information would pass to the next.



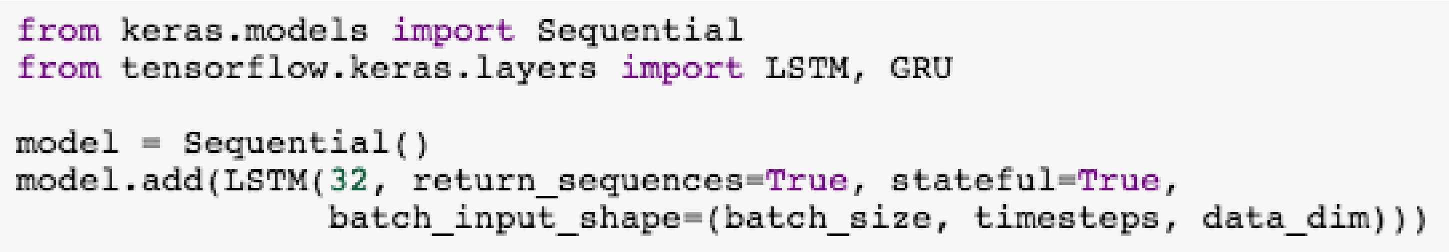
(a) Long Short-Term Memory



(b) Gated Recurrent Unit

RNN in Keras

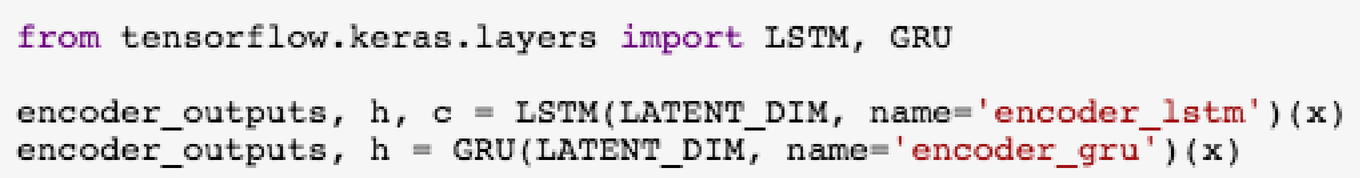
- We can stack RNN, LSTM or GRU by passing a list of layer using Keras Sequential API



```python
from keras.models import Sequential
from tensorflow.keras.layers import LSTM, GRU

model = Sequential()
model.add(LSTM(32, return_sequences=True, stateful=True,
               batch_input_shape=(batch_size, timesteps, data_dim)))
```

- Although RNN and GRU yield two states (output and hidden), LSTM yields three states (output, hidden, and cell)



```python
from tensorflow.keras.layers import LSTM, GRU

encoder_outputs, h, c = LSTM(LATENT_DIM, name='encoder_lstm')(x)
encoder_outputs, h = GRU(LATENT_DIM, name='encoder_gru')(x)
```

Hands-on-Labs Recurrent Neural Network

Recurrent Neural Network

Lab4_Recurrent Neural Network.ipynb

https://colab.research.google.com/drive/1RH5tnTrTPRZkWfNj7PAuPaM_E55coxwp

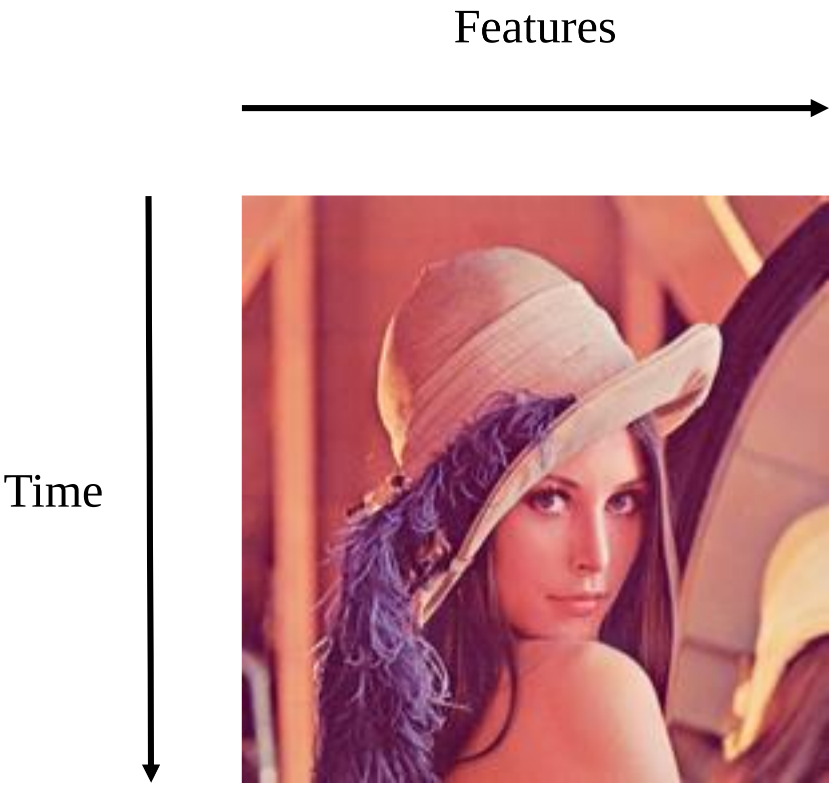
RNN Input Shape

- Regular Sequence Data Shape
  - (N, T, F)
  - N: Number of sequences
  - T: Number of time step for each sequence
  - F: Number of feature for each sample

- LSTM(32, input_shape = (T, F))
  - Latent Dimension: 32
  - Input Shape: T, F (Number of Sequences will be inferred during training)

RNN Model for Image Classification

- Let’s “pretend” the image is a sequence of vectors
- Height as Time / Width as Features

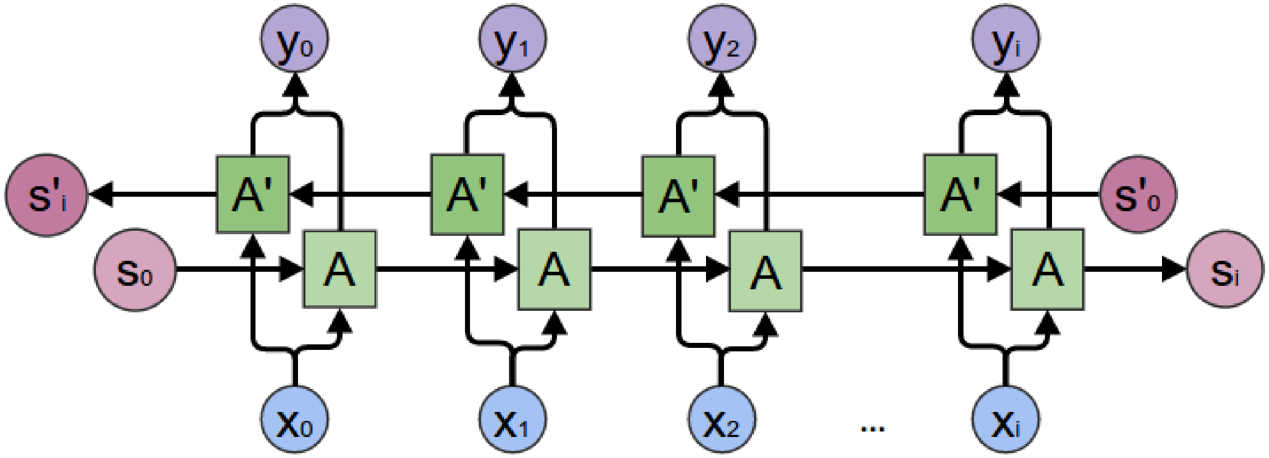


Features

Time

Bidirectional RNN

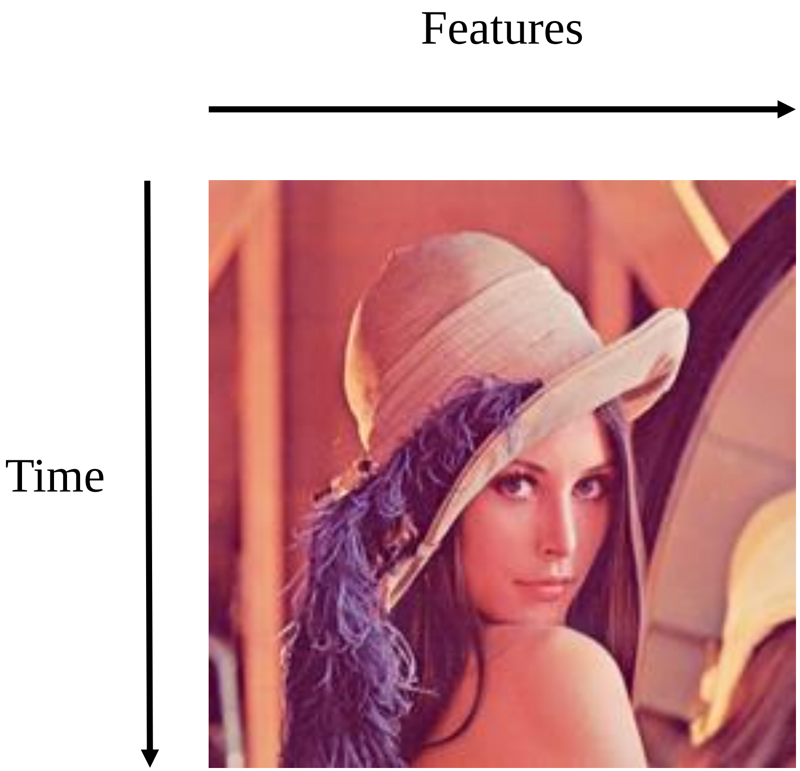
- Bidirectional Recurrent Neural Networks (BRNN) connects two hidden layers of opposite directions to the same output
- BRNNs were introduced to increase the amount of input information available to the network



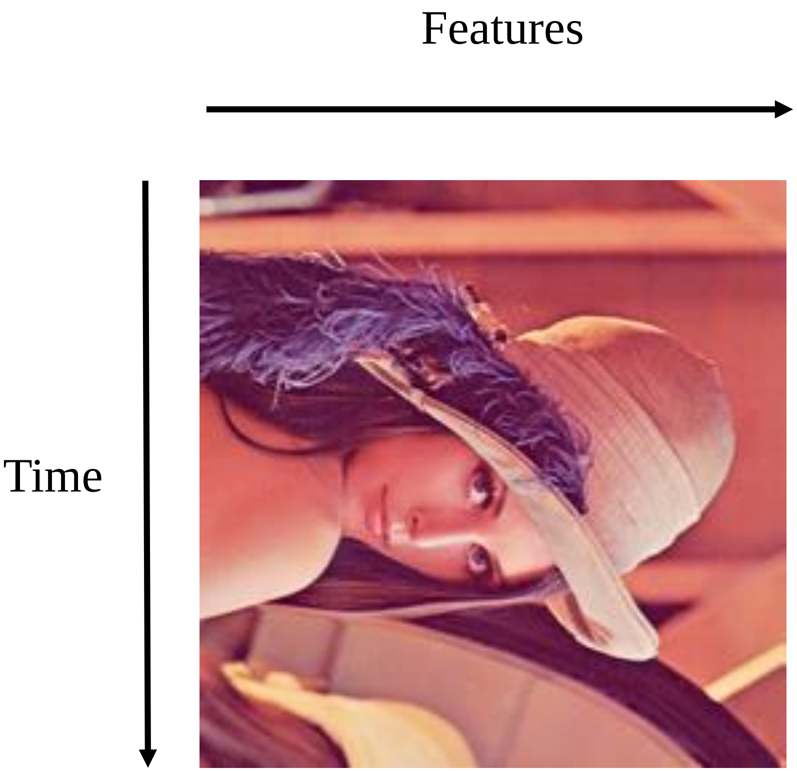
Jongho Kim (quantic.jh@gmail.com)

Bidirectional RNN

- Bidirectional RNN can improve performance
- Apply Bidirectional RNN, Top -> Bottom and Bottom -> Top
- Also, we can rotate images so that we go in 4 directions!


Features

Time


Features

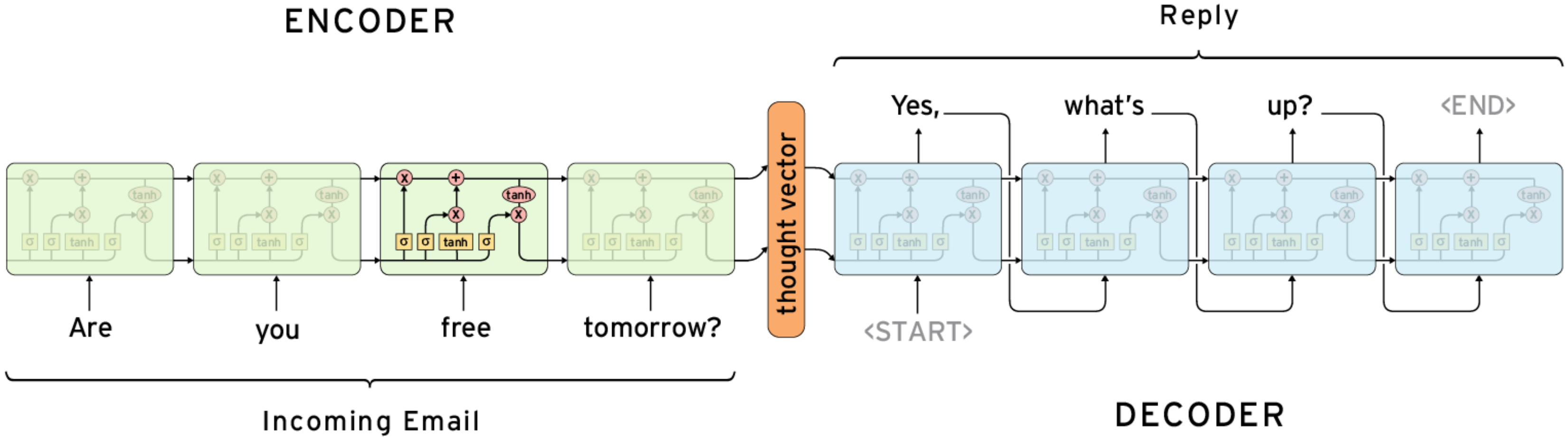
Time

Sequence-to-Sequence Model

Sequence-to-Sequence Model

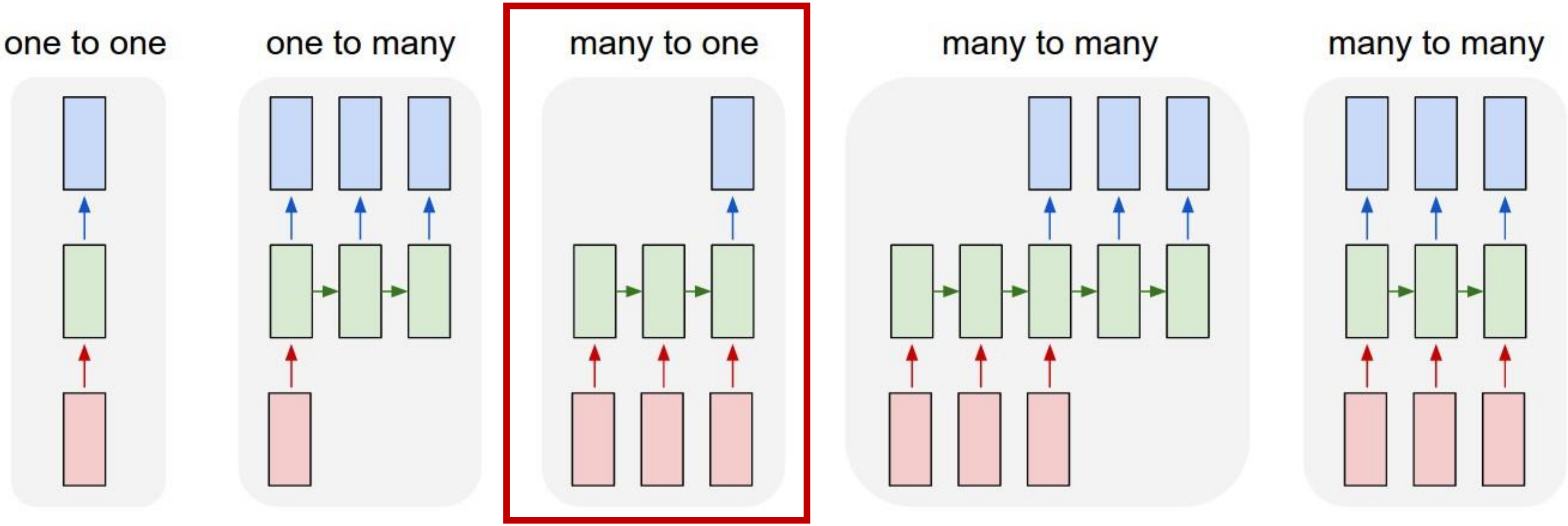
- Our understanding of contexts
  - Answer on Question / Translation etc
  - ➢ (Listening/Reading) → (Understanding) → (Speaking/Writing)

- Understanding of sequence-to-sequence models
  - ➢ (Encoding) → (Thought Vector) → (Decoding)



Seq2Seq Model for Time Series Prediction

- For time series prediction, we can apply seq2seq model
  - Input: x1, x2, … xN
  - Output: xN+1



Thank you ☺

Contact Info: quantic.jh@gmail.com

References

- Jiyong Park (2018), KAIST Summer Session, Retrieved from https://sites.google.com/view/kaist-mis-session2018/overview?authuser=0