Final Project

Automatic Speech Recognition

Group 26 ***
Members : 陳昱喬 許維也 林怡秀 傅莉妮

Introduction & Motivation

unique feature of elevators

in Engineering Building 3

Related Work

Paper:
http://d-scholarship.pitt.edu/36069/1/Ramadan_Mona_PhD_dissertation.pdf

Similarity

1. feature extraction
2. Applications of LSTMs

Difference

1. processing object
2. feature selection method
3. processing time scale

Dataset

- Dataset: from “Tensorflow Speech Recognition Challenge”
- Classes: numbers 1 ~ 9, and two words “up” & “down”

link: https://www.kaggle.com/competitions/tensorflow-speech-recognition-challenge/data

Baseline

- MFCC
- Fbank
- RNN
- LSTM

MFCC (Mel Frequency Cepstral Coefficients)

MFCC extracts features from speech signals, converts speech signals into spectral features, and extracts feature vectors representing speech.

1. Pre-emphasis
2. Framing
3. Windowing
4. Fast Fourier Transform (FFT)
5. Establish a Mel frequency filter bank.
6. Take the logarithm
7. Discrete cosine transform (DCT)

Fbank (Filter Bank)

Fbank extract features from speech signals to represent the characteristics of different sounds in a compact form.

1. Pre-emphasis
2. Framing
3. Fourier transform
4. Filter composition
5. Energy calculation
6. Logarithmic conversion
7. Feature vector extraction
8. Applied to speech recognition

compare MFCC and Fbank

Difference:

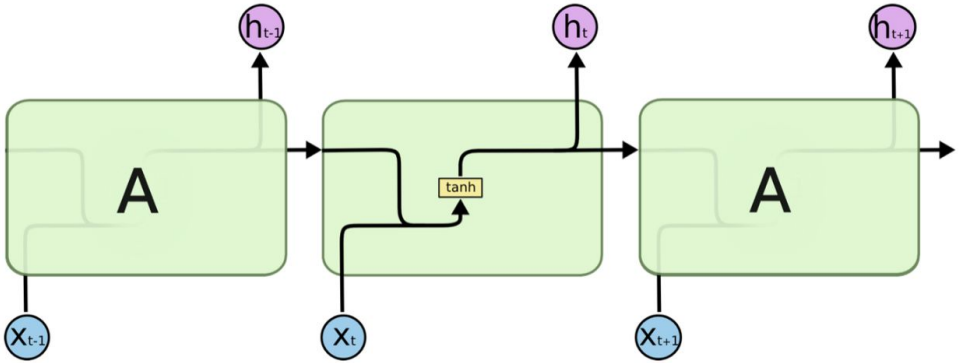
1. Calculation method

2. Feature dimension

RNN(Recurrent Neural Network)

- RNN is one of the artificial neural networks, which considers previous information of sequential data through feedback connections.
- Simplified as the formula:

$$h_t = W * h_{t-1} + U * x_t + b$$
$$y_t = g(V * h_t)$$



source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM

(Long Short Term Memory)

- Disadvantages of RNN
- Core Concept of LSTM
- Applications
- Limitations

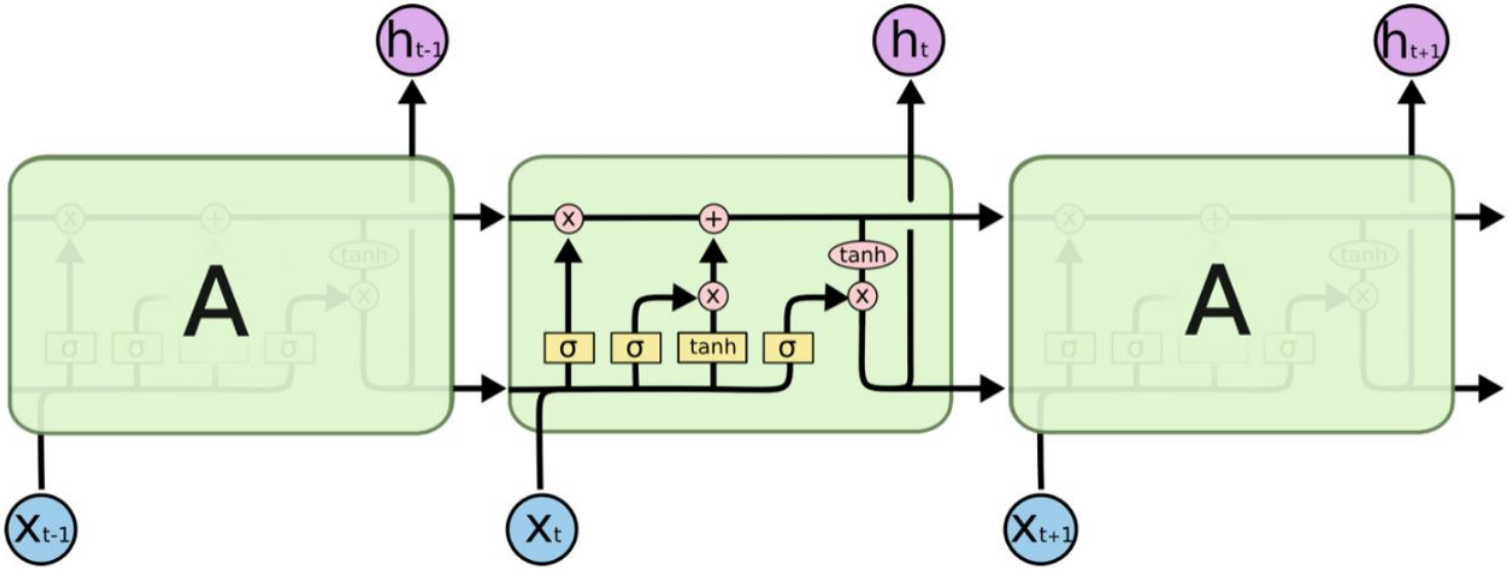
Disadvantages of RNN

Information from distant sources will be “forgotten”(the vanishing gradient problem)

➢ Solution to solving the problem

$$h_t = (W^2 * h_{t-2} + W * U * h_{t-1} + W * b) + U * x_t + b$$
$$= (W^t * h_0 + ...) + U * x_t + b$$

Core Concept of LSTM



source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Applications

- Handwriting
- Recognition,
- Natural Language
- Processing(NLP)
- Robot control
- Music composition

Main Approach

- preprocess
- main function
- Evaluation Metrics

Evaluation Metric

Since our goal is to differentiate between different numbers:

- Calculate “Accuracy”
- Comparing the model’s prediction with the true labels

Limitations

1. **Massive data and computing resource requirements:**
   Requires a large amount of data and computing resources to train and implement.
2. **Difficulty handling long data sequences:**
   The model may struggle with long sequences of data due to the vanishing gradient problem
3. **Difficult interpretability:**

   The interpretability of the model may be difficult, making it hard to understand why certain decisions are being made.

Results & Analysis

Keras

```
1/1 [==============================] - 1s 810ms/step - loss: 0.1050
Epoch 196/200
1/1 [==============================] - 1s 702ms/step - loss: 0.1050
Epoch 197/200
1/1 [==============================] - 1s 730ms/step - loss: 0.0865
Epoch 198/200
1/1 [==============================] - 1s 769ms/step - loss: 0.0864
Epoch 199/200
1/1 [==============================] - 1s 877ms/step - loss: 0.1107
Epoch 200/200
1/1 [==============================] - 1s 772ms/step - loss: 0.1137
19/19 [==============================] - 6s 245ms/step
(600, 99, 12)
(600, 1)
=== prediction ===
correct:  49
accuracy: 0.08166666666666667
start recording
end recording
19/19 [==============================] - 2s 112ms/step
zero
19/19 [==============================] - 2s 111ms/step
zero
```

Torch

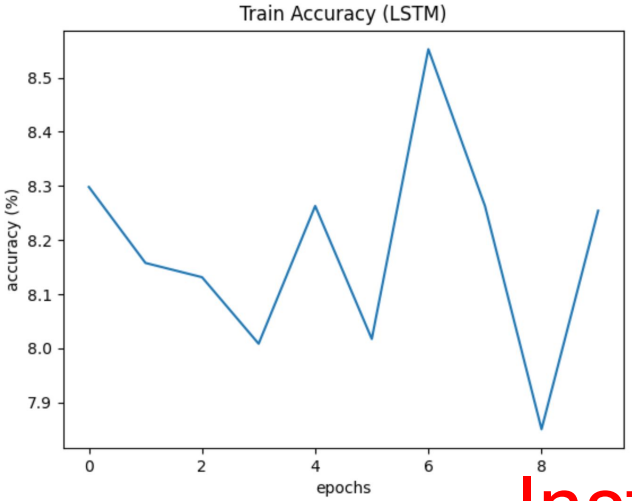
```
100%|
test_avg_acc(%):  8.5
test_avg_loss:  7.454763650894165
100%|
train_avg_acc(%):  8.017543859649123
train_avg_loss:  2.485192240330211
100%|
test_avg_acc(%):  7.000000000000001
test_avg_loss:  7.454378128051758
100%|
train_avg_acc(%):  8.552631578947368
train_avg_loss:  2.485212564468384
100%|
test_avg_acc(%):  9.166666666666666
test_avg_loss:  7.451505422592163
100%|
train_avg_acc(%):  8.263157894736842
train_avg_loss:  2.4852588469522043
100%|
test_avg_acc(%):  8.5
test_avg_loss:  7.450171751022339
```

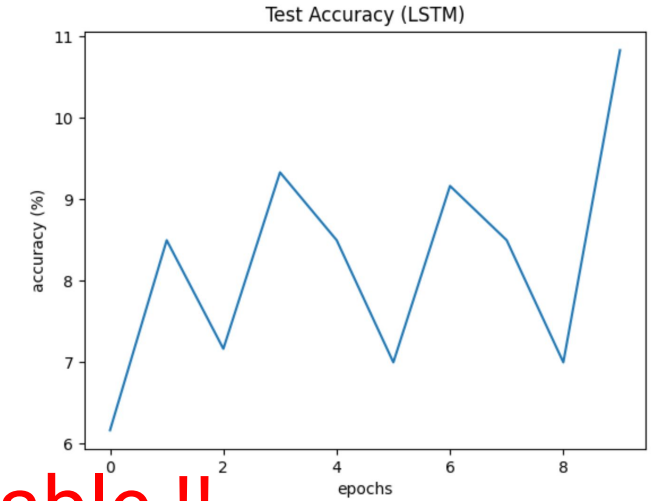
fBank

```
1/1 [==============================] - 0s 116ms/step - loss: 0.2051
Epoch 196/200
1/1 [==============================] - 0s 117ms/step - loss: 0.2030
Epoch 197/200
1/1 [==============================] - 0s 115ms/step - loss: 0.2419
Epoch 198/200
1/1 [==============================] - 0s 114ms/step - loss: 0.1869
Epoch 199/200
1/1 [==============================] - 0s 114ms/step - loss: 0.2065
Epoch 200/200
1/1 [==============================] - 0s 145ms/step - loss: 0.2870
19/19 [==============================] - 1s 26ms/step
(600, 99, 40)
(600, 1)
=== prediction ===
correct:  49
accuracy: 0.08166666666666667
```

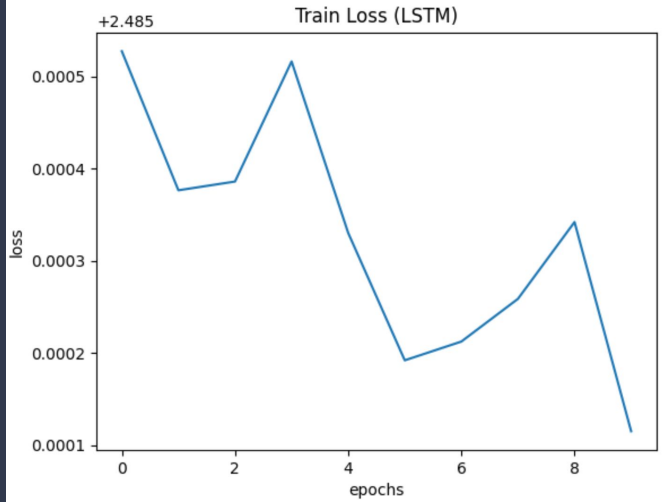
-> same accuracy

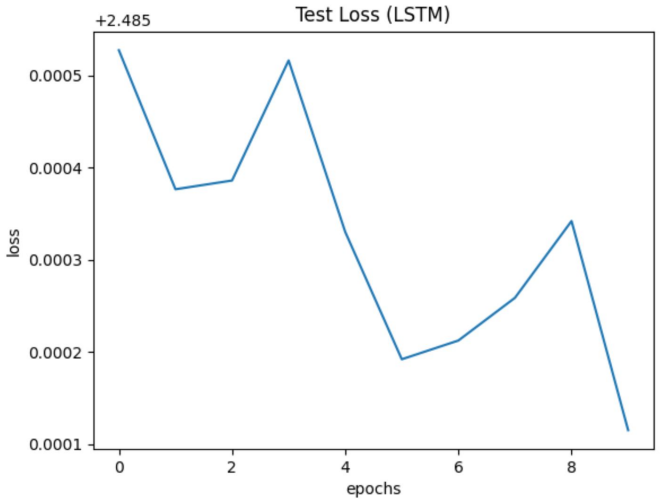
LSTM model





Instable !!

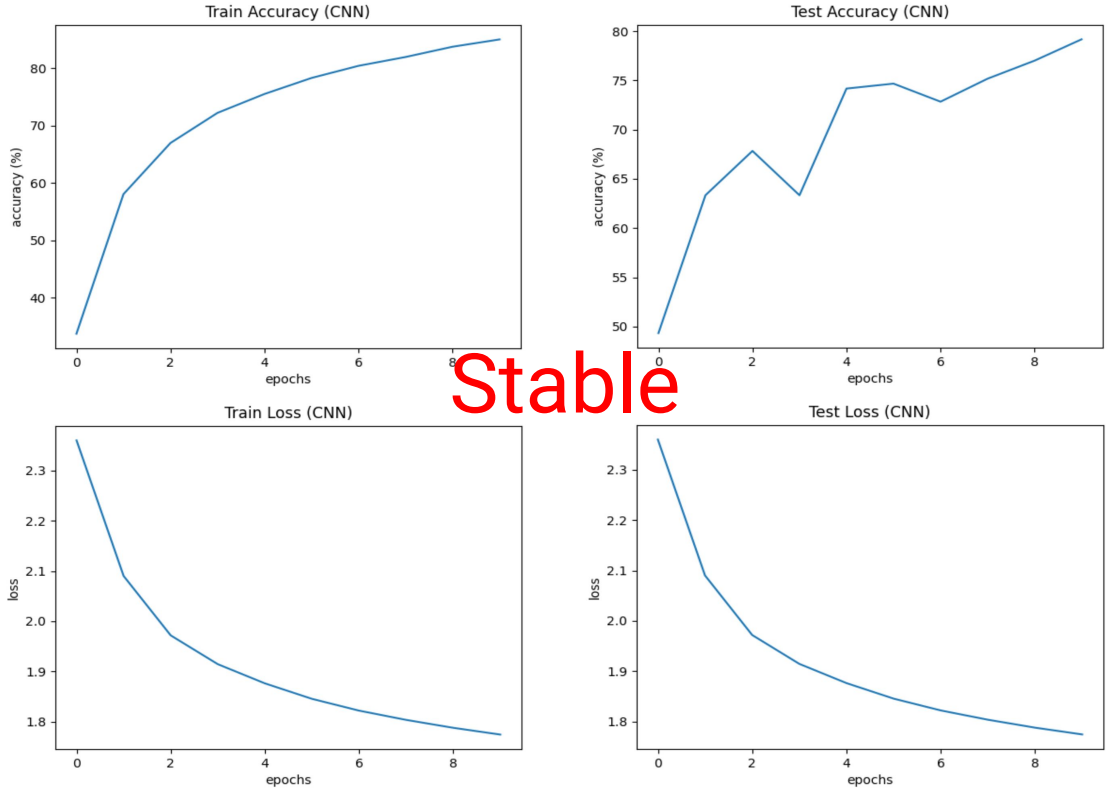




Problem:

not as we expected!

Control Group: Convolutional Neural Network




```
100%|
train_avg_acc(%):  81.11403508771929
train_avg_loss:  1.8213453585641426
100%|
test_avg_acc(%):  75.16666666666667
test_avg_loss:  5.575145959854126
100%|
train_avg_acc(%):  82.77192982456141
train_avg_loss:  1.8008365150083576
100%|
test_avg_acc(%):  79.83333333333333
test_avg_loss:  5.497155666351318
100%|
train_avg_acc(%):  84.00877192982456
train_avg_loss:  1.784987499839381
100%|
test_avg_acc(%):  80.66666666666666
test_avg_loss:  5.458284497261047
100%|
train_avg_acc(%):  85.30701754385966
train_avg_loss:  1.7722294916186416
```

-> much better accuracy

Analysis

CNN has better performance than LSTM ⇒ not as expected!

Possible reason:

1. LSTM model layer need adjustment
2. LSTM is not suitable for word classification

Conclusion

LSTM does work, but

“not suitable” in this case

Main Source & GitHub repo

Dataset: https://www.kaggle.com/competitions/tensorflow-speech-recognition-challenge/data

Related paper: http://d-scholarship.pitt.edu/36069/1/Ramadan_Mona_PhD_dissertation.pdf

GitHub repo: https://github.com/YuChiao13579/Automatic_Speech_Recognition/tree/main

Check our paper report for more details of the model: https://reurl.cc/Eor5rK

Reference

Other helpful links:

- https://www.researchgate.net/figure/Principial-block-scheme-of-MELPSEC-FBANK-and-MFCC-coefficients_fig1_286427067
- https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC
- https://www.researchgate.net/figure/Steps-to-obtain-FBANK-feature_fig2_341712278

- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://github.com/mc6666/MyNeuralNetwork/tree/master/SpeechRecognition
- https://medium.com/ai-academy-taiwan/speech-recognition-using-neural-network-d1af6f482c9b
- https://github.com/lucko515/tesla-stocks-prediction/blob/master/tensorflow_lstm.ipynb

Credit

陳昱喬 - code: preprocessing & UI design; presentation

林怡秀 - code: LSTM model; presentation

傅莉妮 - code: LSTM model; report (RNN LSTM); presentation

許維也 - paper research; report (preprocess method); presentation

Thanks for your listening! :D