Weekly Report

Wangwon Lee, 2019/02/16

This week	Next week
• **Audio Classification** - Previous work - 1D-CNN - Experiments - Batch normalization	• **Audio Classification** - To investigate the model more specific - 1D, 2D CNN visualization - To reduce FC layer

Interesting and new finding

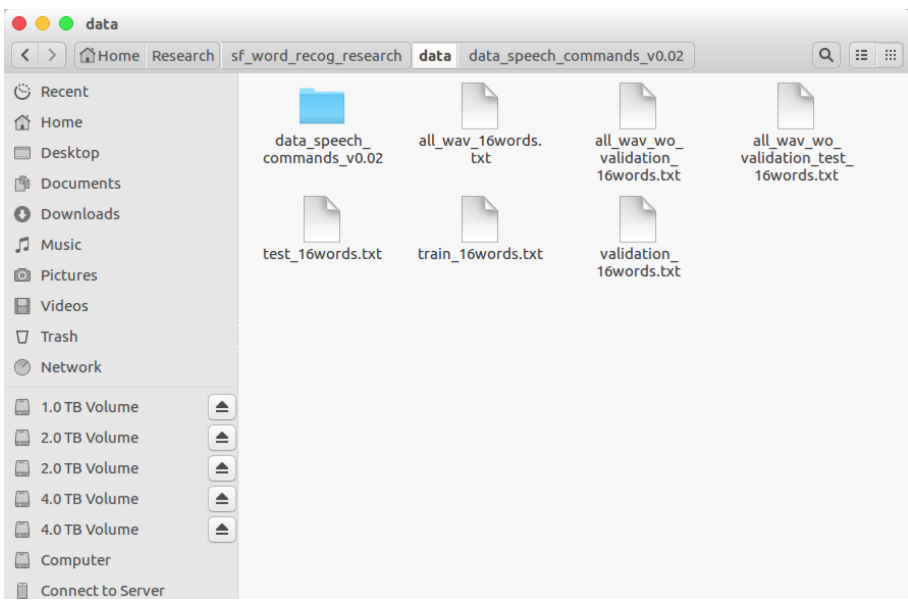
- Problems of Fully Connected Layer
- Batch normalization

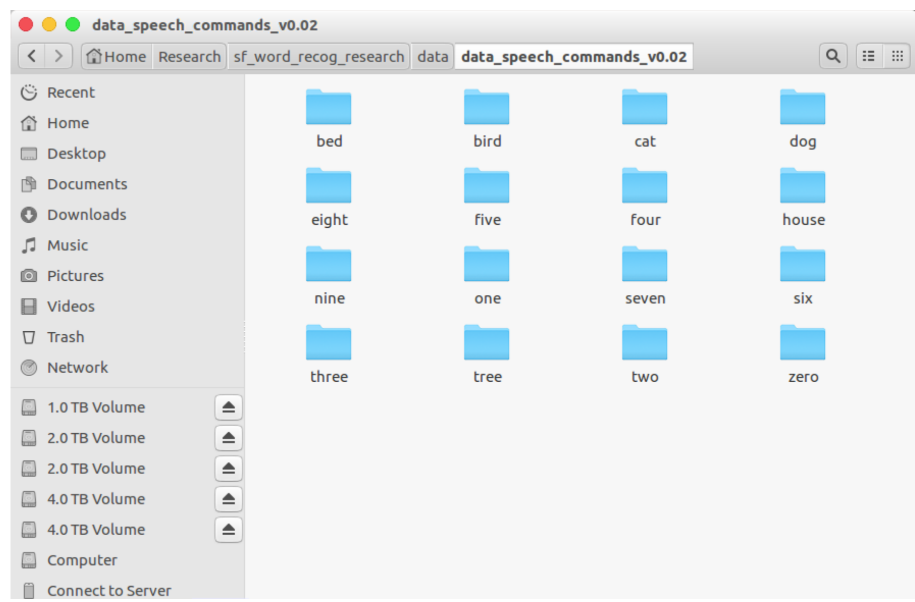
The aim of this month / Discussion

The aim of this month: To study the brain and GLM, To investigate about CNN.

Audio Classification

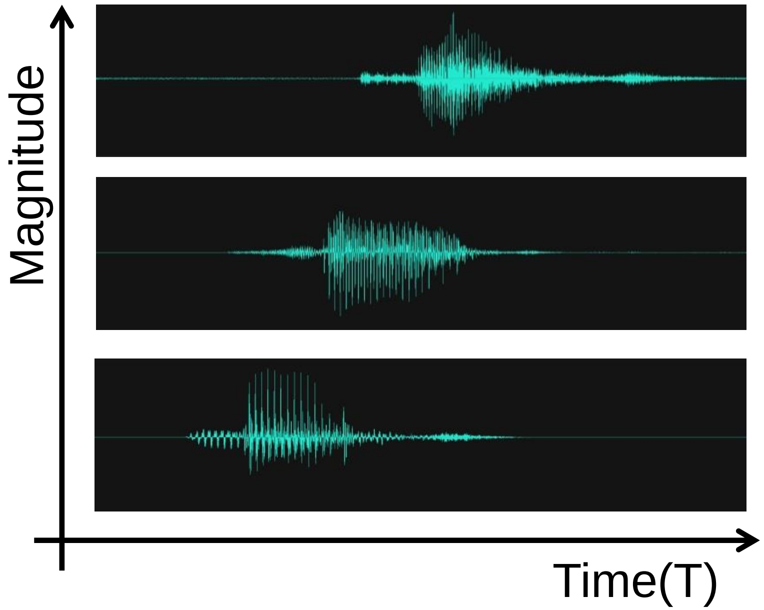
- Data is low-waveform.
  - sec: 1, sampling rate: 16000, type: float32, channel: mono
- 16 class data.
  - 'zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'bed', 'bird', 'tree', 'cat', 'house', 'dog'
- Train: 40851(≒80%), Validation: 4796(≒10%), Test: 5297(≒10%)

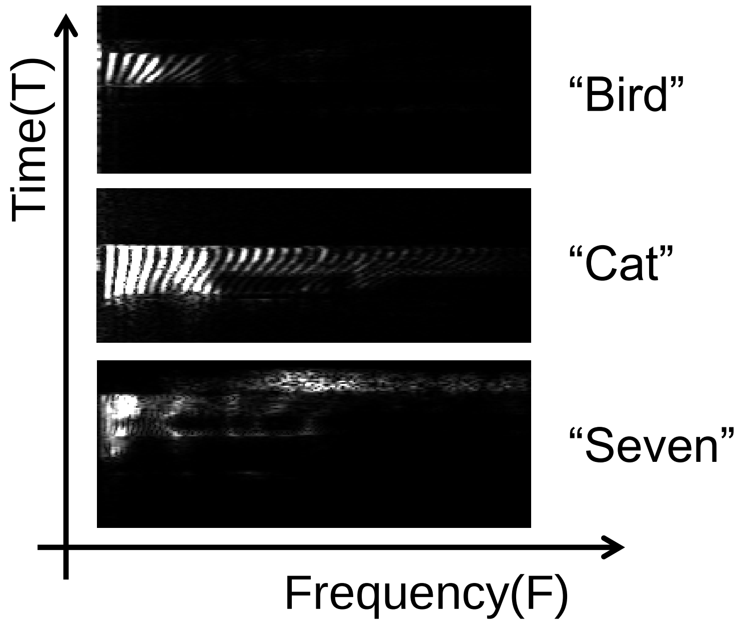




Audio Classification

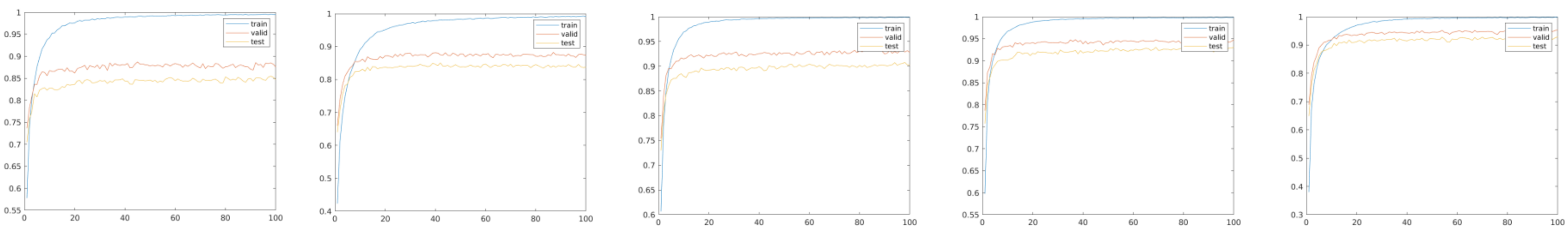
- Convert to spectrogram for using 2D-CNN
- Throught the STFT(Short Time Furier Transform)
  - Window Size: 256, Stride: 128
  - 16000(SR), 1(Ch) -> 99(Freq), 257(Time)
  - It contain the time information and frequency information at once.
- Train: 40851(≒80%), Validation: 4796(≒10%), Test: 5297(≒10%)





Audio Classification

- 2D CNN architecture.
  - 1 Conv(5x5, 8), 1 Pool(2x2), 1 FC, 1 DO(0.5) -> Test Acc: 0.8405
  - 1 Conv(5x5, 8), 1 Pool(2x2), 2 FC, 2 DO(0.5) -> Test Acc: 0.8463
  - 2 Conv(5x5, 8), 2 Pool(2x2), 1 FC, 1 DO(0.5) -> Test Acc: 0.8717
  - …
  - 3 Conv(5x5, 8), 3 Pool(2x2), 1 FC, 1 DO(0.5) -> Test Acc: 0.9119
  - …
  - 4 Conv(5x5, 8), 4 Pool(2x2), 2 FC, 2 DO(0.5) -> Test Acc: 0.9130



< Each learning curve >

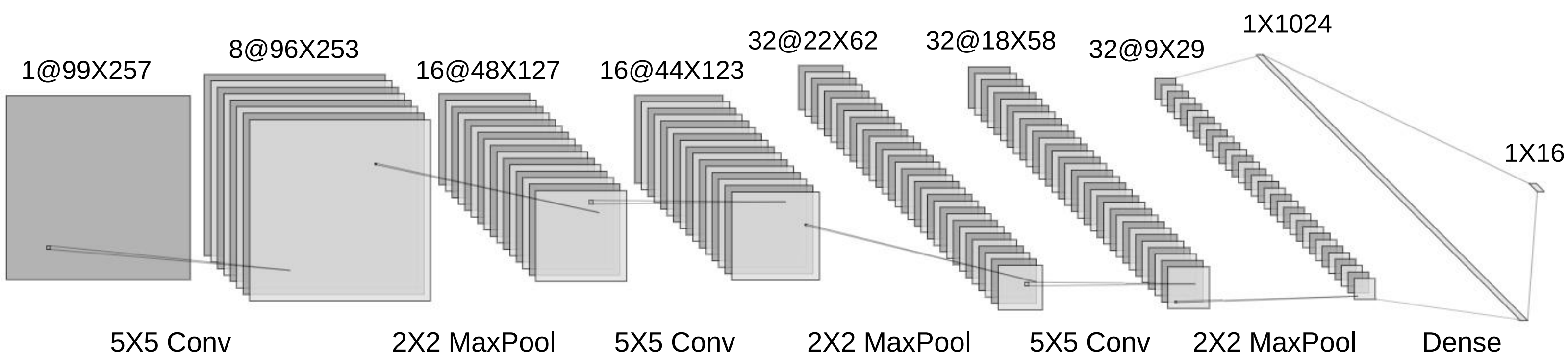
- So I decided that my research target is '3Conv, 1FC' model.
  - Good performance than the other.
  - Good Depth(?) for research. (Not too shallow and Not too deep)

Audio Classification

- '3Conv, 1FC' model's detail.

- Input: 99x257x1 spectrogram image.
- Output:1x16 labeled one hot vector. ('zero', …, 'eight', …, 'house', 'dog')
- Loss: cross entropy loss
- Obtimizer: Adam

- Final accuracy (%)
  - Train: 0.9978
  - Validataion: 0.9404
  - <u>Test: 0.9119</u>



Audio Classification

- 1D CNN architecture
  - For the research, we have to experiment in the same environment.

- So it's the same as 2D CNN's architecture.
  - 1 Conv(25, 8), 1 Pool(4), 1 FC, 1 DO(0.5) -> Test Acc: 0.6648
  - 1 Conv(25, 8), 1 Pool(4), 2 FC, 2 DO(0.5) -> Test Acc: 0.7078
  - 2 Conv(25, 8), 2 Pool(4), 1 FC, 1 DO(0.5) -> Test Acc: 0.7688
  - …
  - 3 Conv(25, 8), 3 Pool(4), 1 FC, 1 DO(0.5) -> Test Acc: 0.8717
  - 3 Conv(25, 8), 3 Pool(4), 2 FC, 2 DO(0.5) -> Test Acc: 0.8702
  - …
  - 5 Conv(25, 8), 5 Pool(4), 1 FC, 1 DO(0.5) -> Test Acc: 0.9047
  - 5 Conv(25, 8), 5 Pool(4), 2 FC, 2 DO(0.5) -> Test Acc: 0.9090

- And… the others are same
  - Input: 16000X1 low-wavform
  - Output:1x16 labeled one hot vector.
    ('zero', …, 'eight', …, 'house', 'dog')
  - Loss: cross entropy loss
  - Obtimizer: Adam

Audio Classification

- Compare 2D CNN and 1D CNN
- Every Accuracy is based on minimizing validation loss

Architecture	2D Acc	1D Acc
1 Conv(5x5, 8), 1 Pool(2x2), 1 FC, 1 DO(0.5)	0.8405	0.6648
1 Conv(5x5, 8), 1 Pool(2x2), 2 FC, 2 DO(0.5)	0.8463	0.7078
2 Conv(5x5, 8), 2 Pool(2x2), 1 FC, 1 DO(0.5)	0.8717	0.7688
2 Conv(5x5, 8), 2 Pool(2x2), 2 FC, 2 DO(0.5)	0.8941	0.8056
3 Conv(5x5, 8), 3 Pool(2x2), 1 FC, 1 DO(0.5)	0.9119	0.8717
3 Conv(5x5, 8), 3 Pool(2x2), 2 FC, 2 DO(0.5)	0.9161	0.8702
4 Conv(5x5, 8), 4 Pool(2x2), 1 FC, 1 DO(0.5)	0.9113	0.8945
4 Conv(5x5, 8), 4 Pool(2x2), 2 FC, 2 DO(0.5)	0.9130	0.9038
5 Conv(5x5, 8), 5 Pool(2x2), 1 FC, 1 DO(0.5)	X	0.9047
5 Conv(5x5, 8), 5 Pool(2x2), 2 FC, 2 DO(0.5)	X	0.9090

Audio Classification

- The accuracy is little higher than the other,
  Is it a good model than the other? No! Because of parameters

Architecture	2D Acc	Params	1D Acc	Params
1 Conv(5x5, 8), 1 Pool(2x2), 1 FC, 1 DO(0.5)	0.8405	49,956,064	0.6648	32,736,480
1 Conv(5x5, 8), 1 Pool(2x2), 2 FC, 2 DO(0.5)	0.8463	50,472,672	0.7078	33,253,088
2 Conv(5x5, 8), 2 Pool(2x2), 1 FC, 1 DO(0.5)	0.8717	22,368,624	0.7688	16,290,160
2 Conv(5x5, 8), 2 Pool(2x2), 2 FC, 2 DO(0.5)	0.8941	22,885,232	0.8056	16,806,768
3 Conv(5x5, 8), 3 Pool(2x2), 1 FC, 1 DO(0.5)	0.9119	8,586,128	0.8717	7,996,304
3 Conv(5x5, 8), 3 Pool(2x2), 2 FC, 2 DO(0.5)	0.9161	9,102,736	0.8702	8,512,912
4 Conv(5x5, 8), 4 Pool(2x2), 1 FC, 1 DO(0.5)	0.9113	2,640,848	0.8945	3,689,424
4 Conv(5x5, 8), 4 Pool(2x2), 2 FC, 2 DO(0.5)	0.9130	3,157,456	0.9038	4,206,032
5 Conv(5x5, 8), 5 Pool(2x2), 1 FC, 1 DO(0.5)	X	X	0.9047	1,338,448
5 Conv(5x5, 8), 5 Pool(2x2), 2 FC, 2 DO(0.5)	X	X	0.9090	1,855,056

Audio Classification

- Most prameters are located on Fully Connected Layer

2D_CNN_3_conv_2_fcn Model

Layer (type)	Output Shape	Param #
conv2d_101 (Conv2D)	(None, 253, 95, 8)	208
max_pooling2d_99 (MaxPooling	(None, 127, 48, 8)	0
conv2d_102 (Conv2D)	(None, 123, 44, 16)	3216
max_pooling2d_100 (MaxPoolin	(None, 62, 22, 16)	0
conv2d_103 (Conv2D)	(None, 58, 18, 32)	12832
max_pooling2d_101 (MaxPoolin	(None, 29, 9, 32)	0
flatten_39 (Flatten)	(None, 8352)	0
dense_96 (Dense)	(None, 1024)	8553472
dropout_57 (Dropout)	(None, 1024)	0
dense_97 (Dense)	(None, 512)	524800
dropout_58 (Dropout)	(None, 512)	0
dense_98 (Dense)	(None, 16)	8208

Total params: 9,102,736

2D_CNN_4_conv_1_fcn Model

Layer (type)	Output Shape	Param #
conv2d_104 (Conv2D)	(None, 253, 95, 8)	208
max_pooling2d_102 (MaxPoolin	(None, 127, 48, 8)	0
conv2d_105 (Conv2D)	(None, 123, 44, 16)	3216
max_pooling2d_103 (MaxPoolin	(None, 62, 22, 16)	0
conv2d_106 (Conv2D)	(None, 58, 18, 32)	12832
max_pooling2d_104 (MaxPoolin	(None, 29, 9, 32)	0
conv2d_107 (Conv2D)	(None, 25, 5, 64)	51264
max_pooling2d_105 (MaxPoolin	(None, 13, 3, 64)	0
flatten_40 (Flatten)	(None, 2496)	0
dense_99 (Dense)	(None, 1024)	2556928
dropout_59 (Dropout)	(None, 1024)	0
dense_100 (Dense)	(None, 16)	16400

Total params: 2,640,848

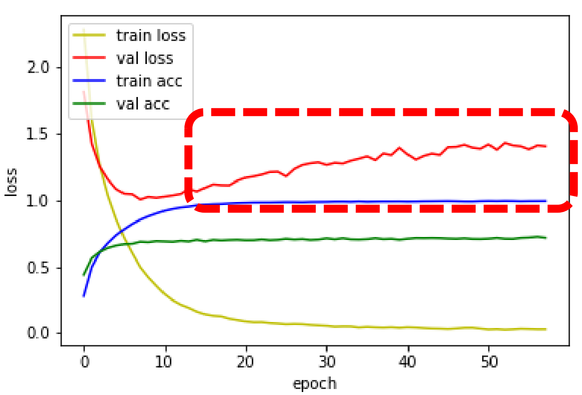
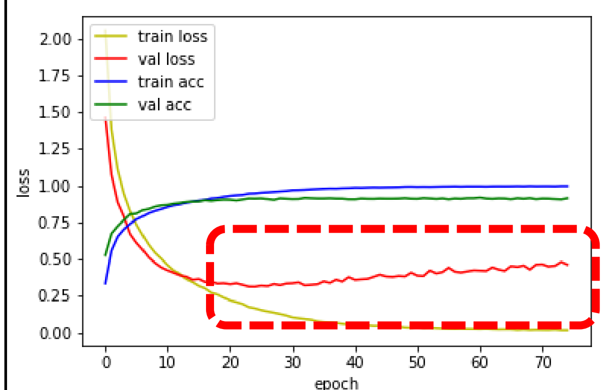
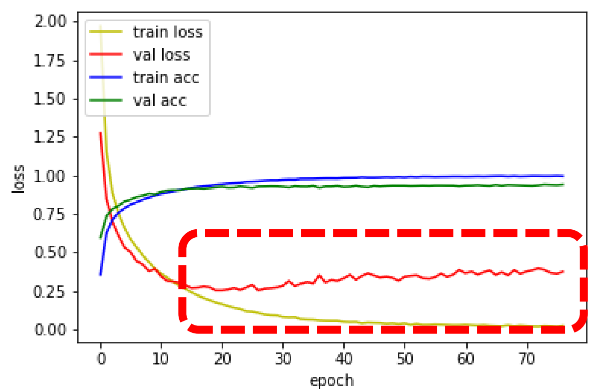
Audio Classification

- Most prameters are located on Fully Connected Layer
- It doesn't help much in the performance of the model

Architecture	2D Acc	Params	1D Acc	Params
1 Conv(5x5, 8), 1 Pool(2x2), 1 FC, 1 DO(0.5)	0.8405	49,956,064	0.6648	32,736,480
1 Conv(5x5, 8), 1 Pool(2x2), 2 FC, 2 DO(0.5)	0.8463	50,472,672	0.7078	33,253,088
2 Conv(5x5, 8), 2 Pool(2x2), 1 FC, 1 DO(0.5)	0.8717	22,368,624	0.7688	16,290,160
2 Conv(5x5, 8), 2 Pool(2x2), 2 FC, 2 DO(0.5)	0.8941	22,885,232	0.8056	16,806,768
3 Conv(5x5, 8), 3 Pool(2x2), 1 FC, 1 DO(0.5)	0.9119	8,586,128	0.8717	7,996,304
3 Conv(5x5, 8), 3 Pool(2x2), 2 FC, 2 DO(0.5)	0.9161	9,102,736	0.8702	8,512,912
4 Conv(5x5, 8), 4 Pool(2x2), 1 FC, 1 DO(0.5)	0.9113	2,640,848	0.8945	3,689,424
4 Conv(5x5, 8), 4 Pool(2x2), 2 FC, 2 DO(0.5)	0.9130	3,157,456	0.9038	4,206,032
5 Conv(5x5, 8), 5 Pool(2x2), 1 FC, 1 DO(0.5)	X	X	0.9047	1,338,448
5 Conv(5x5, 8), 5 Pool(2x2), 2 FC, 2 DO(0.5)	X	X	0.9090	1,855,056

Audio Classification

- Most prameters are located on Fully Connected Layer
- It doesn’t help much in the performance of the model
- The number of FCN parameters more smaller,
  the model becomes more stable
- And I suspect it is the cause of overfit……

1 Conv, 1 FC	3 Conv, 1 FC	5 Conv, 1 FC
		

Audio Classification

- Most prameters are located on Fully Connected Layer
- It doesn't help much in the performance of the model
- The number of FCN parameters more smaller,
  the model becomes more stable
- And I suspect it is the cause of overfit……

- So I think that FCN should be reduced as much as possible.

1 Conv, 1 FC	3 Conv, 1 FC	5 Conv, 1 FC

Audio Classification

- BTW…… why reducing the number of parameter is important?
  - just because computation complexity? No!

- We will investigate neural expression by comparing the DNN and fMRI responses to the same sensory stimuli.
  - Typically, GLM, …
- And, the parameters of CNN are important information about this.

- Therefore, If the number of parameters is more bigger,
  the quality of the information is more sparse.
- But, If the number of parameters is more smaller,
  the quality of the information is more strong.

- And, the FCN are almost impossible to analysis

- So I think, reducing the parameter of FCN is very important
  not only just engineering but also our research.

Audio Classification

- Because the training had seemed unstable, I tried Batch Nomalization
- The depth is more deeper, the performance is more better
- As mentioned in the paper, it seems to prevent the internel covariance shift

Architecture	2D Origin	2D BN	1D Origin	1D BN
1 Conv(5x5, 8), 1 Pool(2x2), 1 FC, 1 DO(0.5)	0.8405	0.8258	0.6648	0.6540
1 Conv(5x5, 8), 1 Pool(2x2), 2 FC, 2 DO(0.5)	0.8463	0.7988	0.7078	0.6800
2 Conv(5x5, 8), 2 Pool(2x2), 1 FC, 1 DO(0.5)	0.8717	0.8719	0.7688	0.7745
2 Conv(5x5, 8), 2 Pool(2x2), 2 FC, 2 DO(0.5)	0.8941	0.8835	0.8056	0.7666
3 Conv(5x5, 8), 3 Pool(2x2), 1 FC, 1 DO(0.5)	0.9119	0.9144	0.8717	0.8461
3 Conv(5x5, 8), 3 Pool(2x2), 2 FC, 2 DO(0.5)	0.9161	0.9234	0.8702	0.8744
4 Conv(5x5, 8), 4 Pool(2x2), 1 FC, 1 DO(0.5)	0.9113	0.9109	0.8945	0.8970
4 Conv(5x5, 8), 4 Pool(2x2), 2 FC, 2 DO(0.5)	0.9130	0.9205	0.9038	0.9061
5 Conv(5x5, 8), 5 Pool(2x2), 1 FC, 1 DO(0.5)	X	X	0.9047	0.9169
5 Conv(5x5, 8), 5 Pool(2x2), 2 FC, 2 DO(0.5)	X	X	0.9090	0.9240

Any Question?

Thank you