

Weekly Report

Wangwon Lee, 2019/03/09

This week

- **1D-CNN Fine Tuning**
 - Previous work
 - Kernel size
 - Channel size
 - Integrate previous result
 - Dropout rate
 - Kernel size like VGG
 - Summary

Next week

- **Audio Classification**
 - To visualize 1D-CNN
 - Understanding 1D-CNN model.
 - Mapping our brain data
 - To investigate 1D model more specific

Interesting and new finding

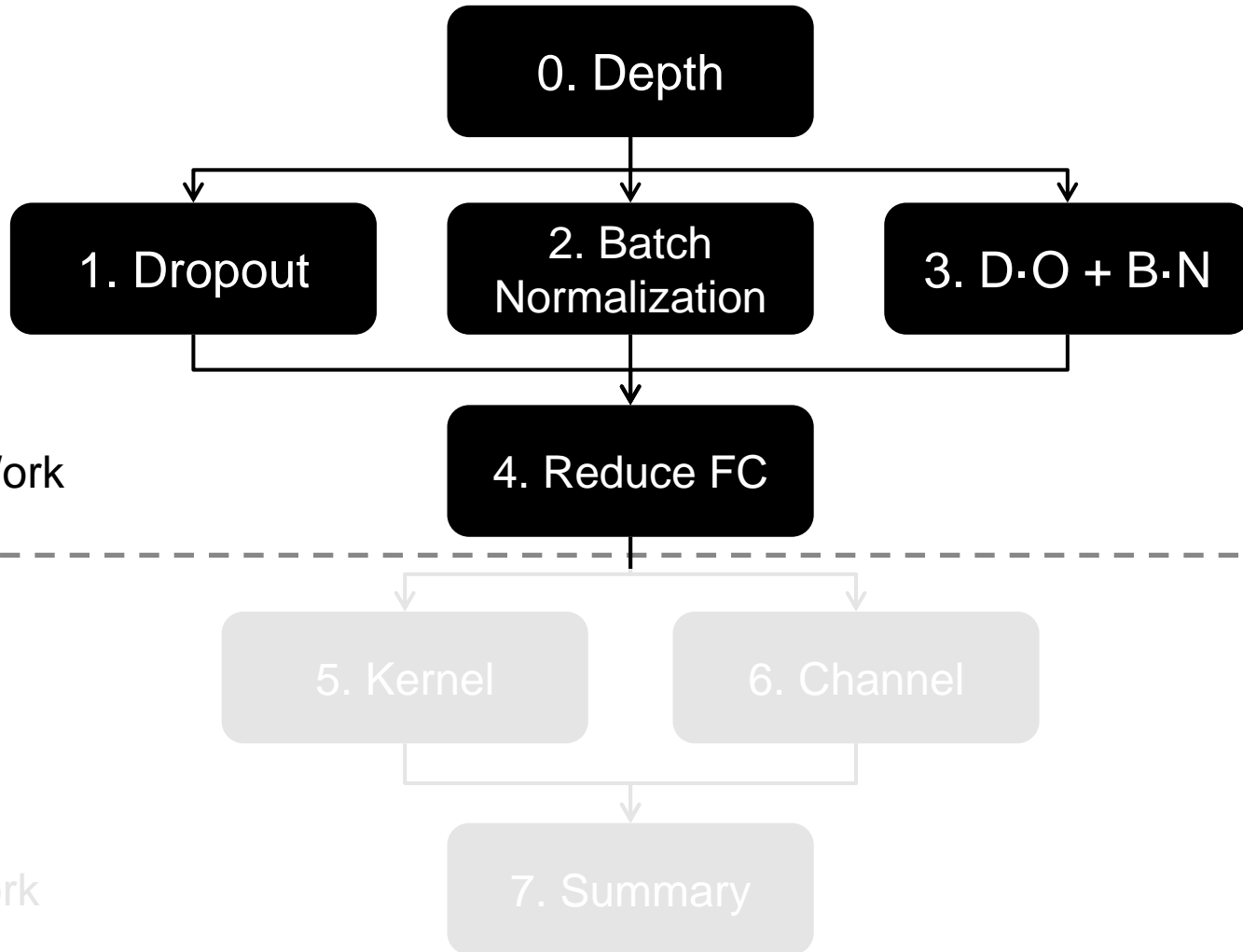
- Fine Tuning

The aim of this month / Discussion

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

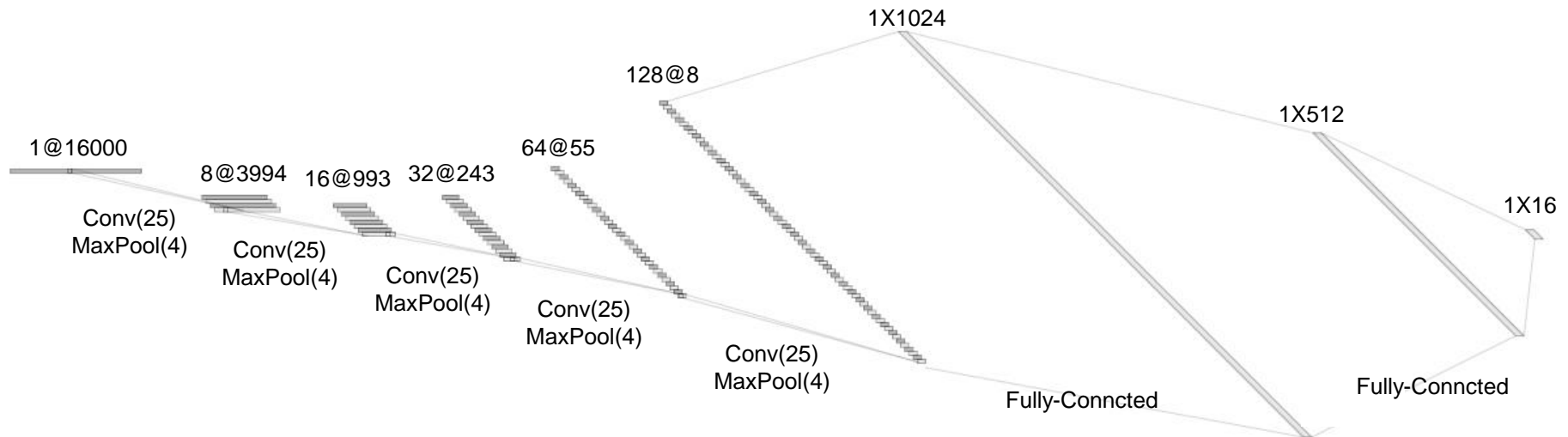
Audio Classification

- Fine tuning task in 1D-CNN



Audio Classification

- For example, '5Conv, 2FC' 1D model's detail.
 - It just flatten 2D model. (5X5 filter->1X25 filter, 2X2 stride->1X4 stride)
 - Input: 16000X1 low waveform.
 - Output: 1x16 labeled one hot vector. ('zero', ..., 'eight', ..., 'house', 'dog')
 - Loss: cross entropy loss
 - Optimizer: Adam
-

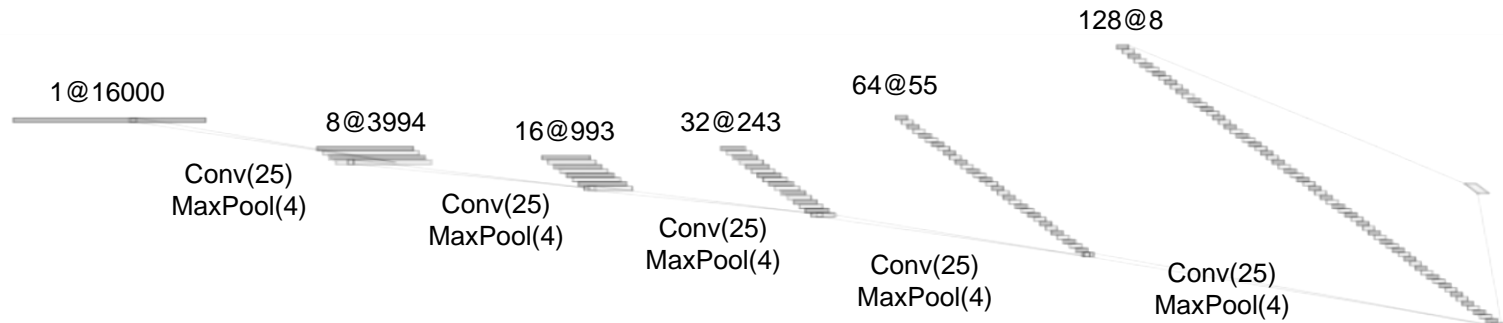


[Baseline Architecture]

Audio Classification

- 4th, I think that FC has a lot of problem (FC = Fully Connected)
- So I retry the model without FC layer (Now I call this model “only conv”)
- FC exists only between the last Conv layer and the output layer (for classification)

Architecture (i = 0,1,2...)	1D No-DO	1D DO(0.5)	1D BN	1D DO+BN
1 CONV(25, $8 \cdot 2^i$)	0.4621	0.4889	0.4289	0.4490
2 CONV(25, $8 \cdot 2^i$)	0.5321	0.6449	0.5836	0.6885
3 CONV(25, $8 \cdot 2^i$)	0.7747	0.8239	0.7890	0.8538
4 CONV(25, $8 \cdot 2^i$)	0.8866	0.9215	0.8804	0.9238
5 CONV(25, $8 \cdot 2^i$)	0.8870	0.9277	0.9288	0.9375



[“Only Conv” Architecture]

Audio Classification

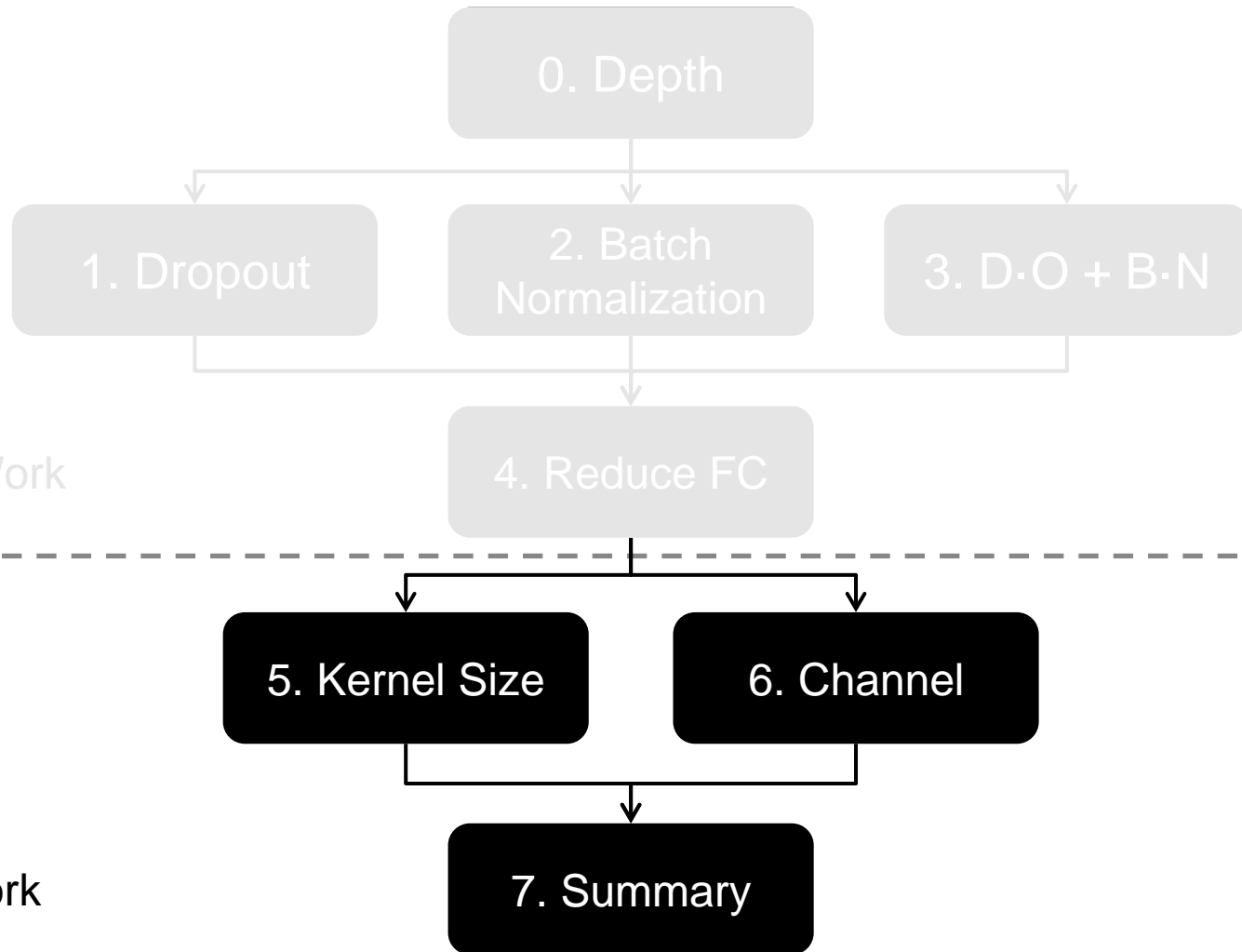
- Compared to the best accuracy, It is increased. (0.9240 → 0.9375)
- Compared to the number of parameter, It is greatly decreased. (~90.8%)
- It's a great achievement beyond what I expected.

base model	Architecture (i = 0,1,2...)	1D DO(0.5) [baseline]	1D BN	1D DO+BN	Params
	...				
	4 Conv(25, 8*2 ⁱ), 4 Pool(4), 1 FC	0.8945	0.8841	0.8970	3,689,424
	4 Conv(25, 8*2 ⁱ), 4 Pool(4), 2 FC	0.9038	0.8814	0.9061	4,206,032
	5 Conv(25, 8*2 ⁱ), 5 Pool(4), 1 FC	0.9047	0.9026	0.9169	1,338,448
	5 Conv(25, 8*2 ⁱ), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056

Only Conv model	Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
	...				
	4 CONV(25, 8*2 ⁱ)	0.9215	0.8804	0.9238	123,856
	5 CONV(25, 8*2 ⁱ)	0.9277	0.9288	0.9375	288,848

Audio Classification

- Fine tuning task in 1D-CNN



Audio Classification

- Change kernel size 25 -> 5
- If we reduce the kernel size, we can make the model deeper.
- Pooling size is also reduced.

Architecture (i = 0,1,2...)	DO(0.5)	BN	DO+BN	Params
1 Conv(5, $8 \cdot 2^i$), 1 Pool(3, 3)	0.2818	0.2735	0.3171	682,576
2 Conv(5, $8 \cdot 2^i$), 2 Pool(3, 3)	0.4671	0.4665	0.4974	455,424
3 Conv(5, $8 \cdot 2^i$), 3 Pool(3, 3)	0.6295	0.6004	0.6725	306,016
4 Conv(5, $8 \cdot 2^i$), 4 Pool(3, 3)	0.7556	0.6644	0.7807	214,560
5 Conv(5, $8 \cdot 2^i$), 5 Pool(3, 3)	0.8492	0.7674	0.8737	186,272
6 Conv(5, $8 \cdot 2^i$), 6 Pool(3, 3)	0.9180	0.8860	0.9192	301,728
7 Conv(5, $8 \cdot 2^i$), 7 Pool(3, 3)	0.9196	0.9167	0.9445	925,856
8 Conv(5, $8 \cdot 2^i$), 8 Pool(3, 3)	0.9132	0.9238	0.9310	3,517,600

Audio Classification

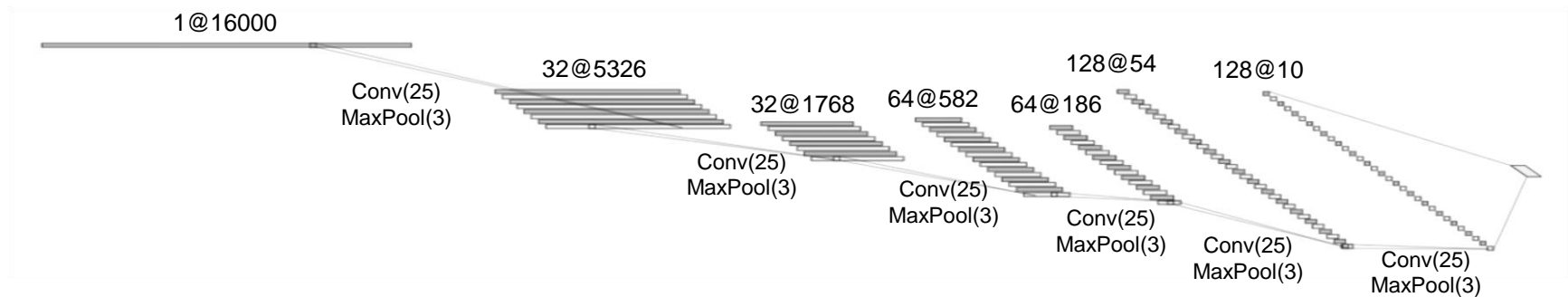
- Summarising this straight. This is current SOTA(State Of The Art) in my research.
- Accuracy is more increased. (0.9375 -> 0.9445)

	Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
baseline					
base model	5 Conv(25, $8 \cdot 2^i$), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
Accuracy and Number of parameters					
Only Conv	4 CONV(25, $8 \cdot 2^i$)	0.9215	0.8804	0.9238	123,856
Only Accuracy					
Only Conv	5 CONV(25, $8 \cdot 2^i$)	0.9277	0.9288	0.9375	288,848
And, here is new challenger					
Only Conv kernel 5	7 Conv(5, $8 \cdot 2^i$), 7 Pool(3, 3)	0.9196	0.9167	0.9445	925,856

Audio Classification

- Change channel size
- Start channel size: 32

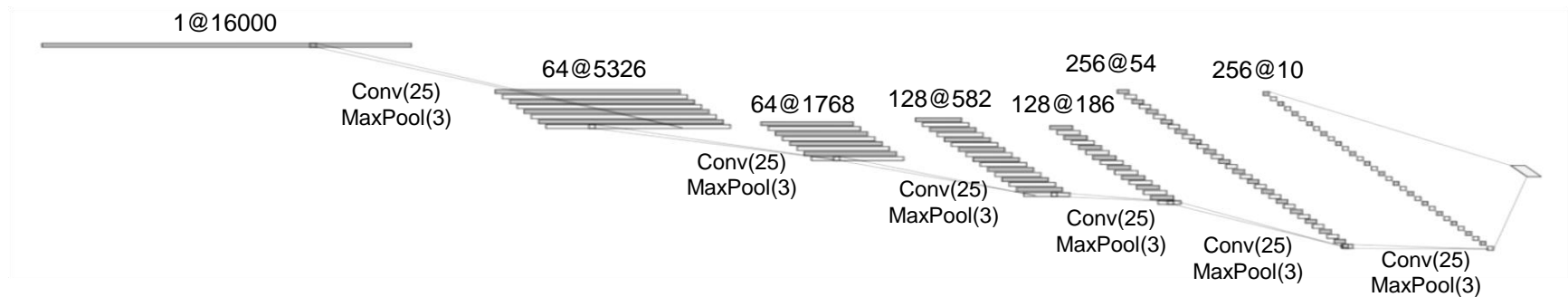
Architecture	DO(0.5)	BN	DO+BN	Params
1 Conv(25), 1 Pool(3, 3)	0.4874	0.4548	0.4453	2,727,824
2 Conv(25), 2 Pool(3, 3)	0.5948	0.5913	0.6399	931,824
3 Conv(25), 3 Pool(3, 3)	0.7886	0.7053	0.7782	673,968
4 Conv(25), 4 Pool(3, 3)	0.8953	0.8810	0.8987	371,056
5 Conv(25), 5 Pool(3, 3)	0.9375	0.9279	0.9387	496,368
6 Conv(25), 6 Pool(3, 3)	0.9364	0.9479	0.9479	816,240



Audio Classification

- Change channel size
- Start channel size: 64

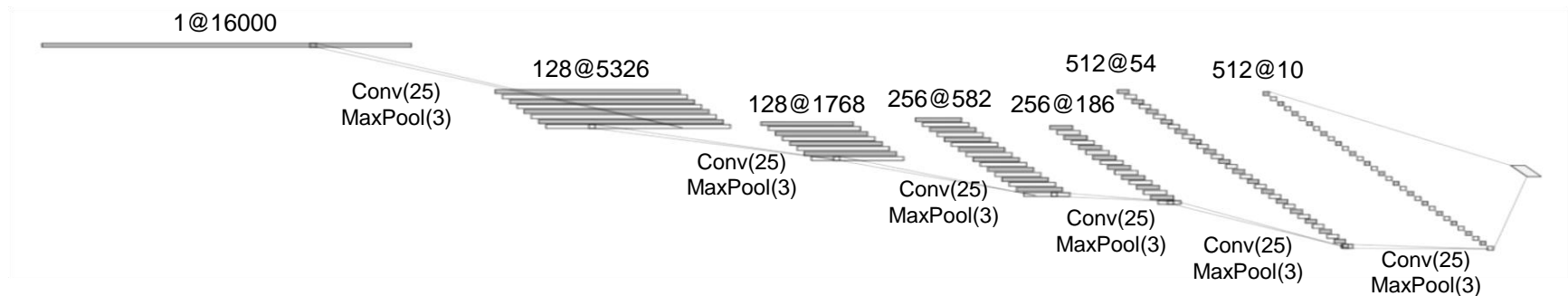
Architecture	DO(0.5)	BN	DO+BN	Params
1 Conv(25), 1 Pool(3, 3)	0.4962	0.3601	0.4469	6,137,360
2 Conv(25), 2 Pool(3, 3)	0.6486	0.5398	0.6361	2,141,136
3 Conv(25), 3 Pool(3, 3)	0.7958	0.6139	0.7965	1,650,512
4 Conv(25), 4 Pool(3, 3)	0.8945	0.8619	0.9030	1,148,112
5 Conv(25), 5 Pool(3, 3)	0.9408	0.9408	0.9466	1,788,368
6 Conv(25), 6 Pool(3, 3)	0.9412	0.9547	0.9543	3,224,784



Audio Classification

- Change channel size
- Start channel size: 128

Architecture	DO(0.5)	BN	DO+BN	Params
1 Conv(25), 1 Pool(3, 3)	0.5148	0.4417	0.4673	10,911,248
2 Conv(25), 2 Pool(3, 3)	0.6536	0.6108	0.6928	4,034,448
3 Conv(25), 3 Pool(3, 3)	0.8133	0.7502	0.7691	3,617,424
4 Conv(25), 4 Pool(3, 3)	0.9024	0.8611	0.9171	3,634,576
5 Conv(25), 5 Pool(3, 3)	0.9398	0.9412	0.9462	6,593,424
6 Conv(25), 6 Pool(3, 3)	0.9391	0.9526	0.9601	12,788,112



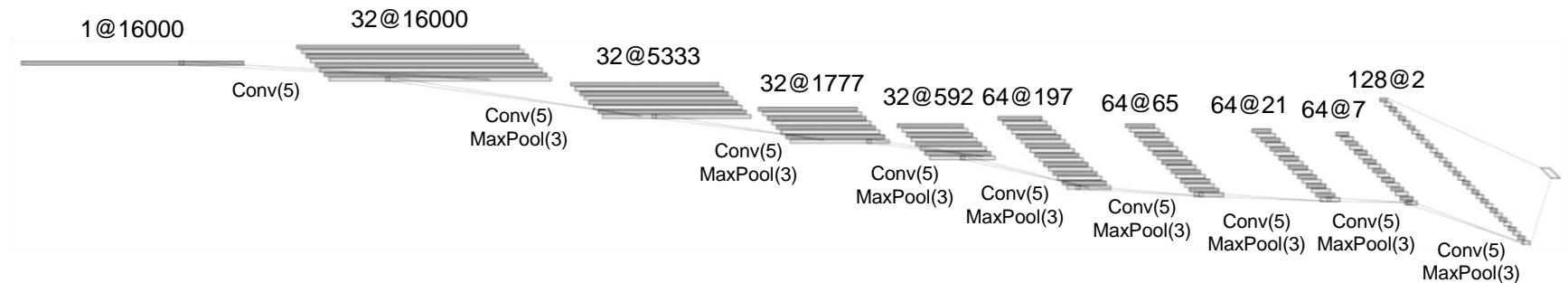
Audio Classification

- Summarising this straight. This is current SOTA(State Of The Art) in my research.
- Accuracy is more increased. (0.9445 -> 0.9601)
- The number of params is

	Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
baseline					
base model	5 Conv(25, $8 \cdot 2^i$), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
Accuracy and Number of parameters					
Only Conv	4 CONV(25, $8 \cdot 2^i$)	0.9215	0.8804	0.9238	123,856
Only Accuracy					
Only Conv	7 Conv(5, $8 \cdot 2^i$), 7 Pool(3, 3)	0.9196	0.9167	0.9445	925,856
And, here is new challenger					
Only Conv channel 128	6 Conv(25), 6 Pool(3, 3)	0.9391	0.9526	0.9601	12,788,112

Audio Classification

- Finally, I made custom model to integrate previous result.
- First, In the case of a “DO, BN, DO+BN”, I tried everything
-> because the results were different each time.
- Second, In the case of a channel, I tried everything
-> because the number of channels and the accuracy are inversely proportional.
- Third, In the case of Kernel size, I tried reducing the size
-> because reducing the kernel size could make the model deeper.
-> pooling size is also reduced. 4 -> 3
- So, the architecture below is an example of one of them



Audio Classification

- I didn't try early because the results were probably not good.
- Start channel size: 32
- The performance was close to 0.95 with less than 100,000 parameters.

Architecture	DO(0.5)	BN	DO+BN	Params
1 CONV(5, 32)	X	X	X	X
2 CONV(5, 32)	X	X	X	X
3 CONV(5, 32)	0.5169	0.5119	0.5676	920,336
4 CONV(5, 32)	0.6656	0.6253	0.7376	318,768
5 CONV(5, 64)	0.7605	0.7065	0.8039	227,696
6 CONV(5, 64)	0.8893	0.8278	0.8826	113,072
7 CONV(5, 64)	0.9373	0.9011	0.9293	88,560
8 CONV(5, 64)	0.9477	0.9285	0.9391	94,768
9 CONV(5, 128)	0.9379	0.9268	0.9424	132,784

Audio Classification

- I didn't try early because the results were probably not good.
- Start channel size: 64
- I think this is the most appropriate between the number of parameters and accuracy.

Architecture	DO(0.5)	BN	DO+BN	Params
1 CONV(5, 64)	X	X	X	X
2 CONV(5, 64)	X	X	X	X
3 CONV(5, 64)	0.5844	0.4893	0.5726	1,861,520
4 CONV(5, 64)	0.7128	0.6291	0.7333	668,752
5 CONV(5, 128)	0.7732	0.7022	0.8073	507,344
6 CONV(5, 128)	0.8901	0.8233	0.8818	319,312
7 CONV(5, 128)	0.9400	0.9061	0.9302	311,504
8 CONV(5, 128)	0.9560	0.9508	0.9458	365,136
9 CONV(5, 256)	0.9537	0.9464	0.9470	523,600

Audio Classification

- I didn't try early because the results were probably not good.
- Start channel size: 128
- Accuracy is high, but the number of parameter is too large.

Architecture	DO(0.5)	BN	DO+BN	Params
1 CONV(5, 128)	X	X	X	X
2 CONV(5, 128)	X	X	X	X
3 CONV(5, 128)	0.5718	0.4521	0.5720	3,804,944
4 CONV(5, 128)	0.6766	0.6183	0.7371	1,460,368
5 CONV(5, 256)	0.7701	0.7067	0.7985	1,219,472
6 CONV(5, 256)	0.8802	0.8588	0.8762	1,007,248
7 CONV(5, 256)	0.9387	0.9236	0.9443	1,155,472
8 CONV(5, 256)	0.9543	0.9562	0.9562	1,426,576
9 CONV(5, 512)	0.9396	0.9632	0.9674	2,071,184

Audio Classification

- Summarising this straight. This is current SOTA(State Of The Art) in my research.
- Accuracy is more increased. (0.9601 -> 0.9674)
- The number of params is decreased. (123,856 -> 94,768 And 0.9238 -> 0.9477)

	Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
baseline					
base model	5 Conv(25, 8*2 ⁱ), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
Accuracy and Number of parameters					
Only Conv	4 CONV(25, 8*2 ⁱ)	0.9215	0.8804	0.9238	123,856
Only Accuracy					
Only Conv channel 128	6 Conv(25), 6 Pool(3, 3)	0.9391	0.9526	0.9601	12,788,112
And, here is new challenger					
Custom channel 32	8 CONV(5, 64)	0.9477	0.9285	0.9391	94,768
Custom channel 64	8 CONV(5, 128)	0.9560	0.9508	0.9458	365,136
Custom channel 128	9 CONV(5, 512)	0.9396	0.9632	0.9674	2,071,184

Audio Classification

- Summarising this straight. This is current SOTA(State Of The Art) in my research.

	Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
baseline					
base model	5 Conv(25, $8 \cdot 2^i$), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
Accuracy and Number of parameters					
Custom channel 32	8 CONV(5, 64)	0.9477	0.9285	0.9391	94,768
Custom channel 64	8 CONV(5, 128)	0.9560	0.9508	0.9458	365,136
Only Accuracy					
Custom channel 128	9 CONV(5, 512)	0.9396	0.9632	0.9674	2,071,184

Audio Classification

- Summarising this straight. This is current SOTA(State Of The Art) in my research.

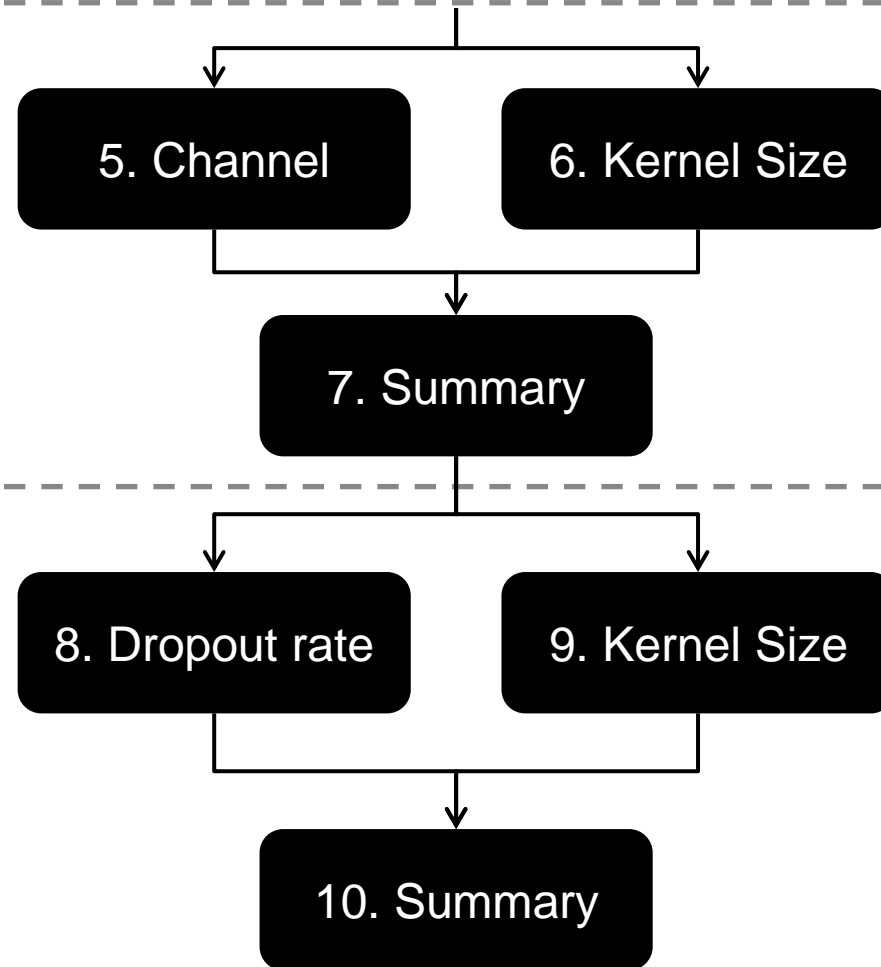
Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
baseline				
base model	0.909	0.909	0.921	1,855,056
Accuracy and Number of parameters				
Custom channel 32	0.9477	0.9285	0.9391	94,768
Custom channel 64	0.9511	0.9395	0.9505	189,536
Only Accuracy				
Custom channel 128	0.9396	0.9632	0.9628	2,071,184

The previous goal was here...

Frankly, a lot of experiment are more...

Audio Classification

- Fine tuning task in 1D-CNN



Previous goal

Added Work

Audio Classification

- I tune the dropout's rate
- The accuracy was close to 0.96

Architecture	DO(0.25)	DO(0.25)+BN	DO(0.75)	DO(0.75)+BN
1 CONV(5, 64)	0.3180	0.2069	0.3194	0.3049
2 CONV(5, 64)	0.4372	0.2760	0.4619	0.4617
3 CONV(5, 64)	0.5448	0.5639	0.6430	0.6426
4 CONV(5, 64)	0.6125	0.6625	0.7796	0.7934
5 CONV(5, 128)	0.7240	0.7653	0.8351	0.8320
6 CONV(5, 128)	0.8534	0.8633	0.9153	0.8812
7 CONV(5, 128)	0.9219	0.9269	0.9472	0.9321
8 CONV(5, 128)	0.9458	0.9512	0.9589	0.9497
9 CONV(5, 256)	0.9292	0.9578	0.9553	0.9585

Audio Classification

- Summarising this straight. This is current SOTA(State Of The Art) in my research.
- The performance is increased. (0.9560 -> 0.9589)
- But... I'm not sure it is really better...

	Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
baseline					
base model	5 Conv(25, $8 \cdot 2^i$), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
Accuracy and Number of parameters					
Custom channel 32	8 CONV(5, 64)	0.9477	0.9285	0.9391	94,768
Custom channel 64	8 CONV(5, 128)	0.9560	0.9508	0.9458	365,136
Only Accuracy					
Custom channel 128	9 CONV(5, 512)	0.9396	0.9632	0.9674	2,071,184
And, here is new challenger					
Custom channel 64 DO(0.75)	8 CONV(5, 128)	0.9589	X	0.9497	363,600

Audio Classification

- Change kernel size 1X5 (one layer) -> 1X3 and 1X3 (two layer)

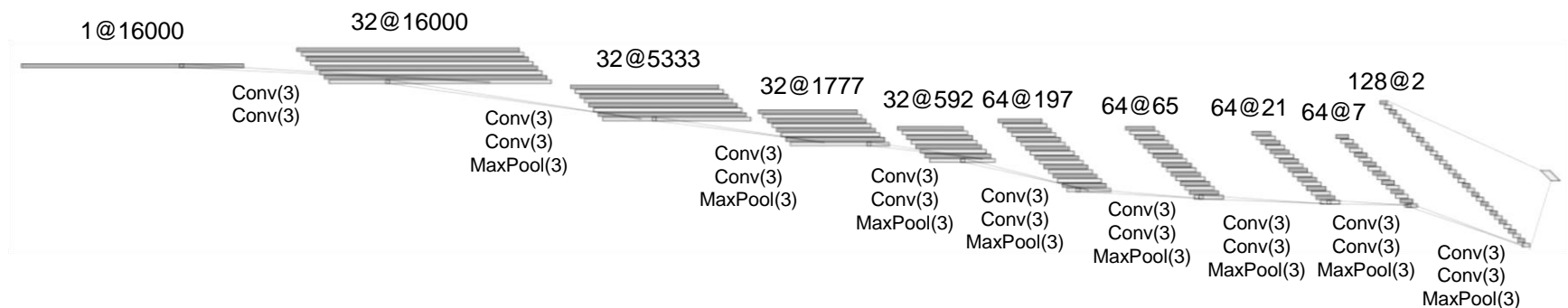
--- From VGG paper... ---

such layers have a 7×7 effective receptive field.

So what have we gained by using, for instance, a stack of three 3×3 conv layers instead of a single 7×7 layer?

First, we incorporate three non-linear rectification layers instead of a single one, which makes the decision function more discriminative.

- According to the VGG paper, it is better to replace a large-sized kernel with a combination of 3X3 kernels.
- So, I tried.



Audio Classification

- I tried VGG style
- The accuracy exceeded 0.96.

Architecture	1D DO(0.5)	1D BN	1D DO+BN	Params
2 CONV(3, 64), 1 Pool	0.3877	0.3674	0.3171	16,396,624
4 CONV(3, 64) , 2 Pool	0.4752	0.3668	0.3389	5,498,320
6 CONV(3, 64) ,3 Pool	0.5626	0.4933	0.4984	1,881,680
8 CONV(3, 64) , 4 Pool	0.6719	0.6521	0.6544	692,944
10 CONV(3, 128) , 5 Pool	0.7666	0.7047	0.7344	564,176
12 CONV(3, 128) , 6 Pool	0.9018	0.8349	0.7502	392,400
14 CONV(3, 128) , 7 Pool	0.9427	0.9286	0.8044	400,848
16 CONV(3, 128) , 8 Pool	0.9630	0.9423	0.9136	470,736
18 CONV(3, 256) , 9 Pool	0.9350	0.9493	0.9418	760,016

Audio Classification

- Summarising this straight. This is current SOTA(State Of The Art) in my research.
- The performance was increased. (0.9589 -> 0.9630)
- But, Params was also increased. (363,600 -> 470,736)

	Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
baseline					
base model	5 Conv(25, $8 \cdot 2^i$), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
Accuracy and Number of parameters					
Custom channel 32	8 CONV(5, 64)	0.9477	0.9285	0.9391	94,768
Custom channel 64 DO(0.75)	8 CONV(5, 128)	0.9589	X	0.9497	363,600
Only Accuracy					
Custom channel 128	9 CONV(5, 512)	0.9396	0.9632	0.9674	2,071,184
And, here is new challenger					
Custom VGG style	16 CONV(3, 128) , 8 Pool	0.9630	0.9423	0.9136	470,736

Audio Classification

- Summarising this straight. This is current SOTA(State Of The Art) in my research.
- The performance was increased. (0.9589 -> 0.9630)
- But, Params was also increased. (363,600 -> 470,736)

Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
-----------------------------	------------	-------	----------	--------

Frankly, a lot of experiment are more

But, because it didn't improve more,
I don't talk about some more...

base model	5 Conv(25, 3*2), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
Accuracy and Number of parameters					
Custom channel 64	8 CONV(5, 64)	0.9177	0.9285	0.9391	94,768
Custom channel 64 DO(0.75)	8 CONV(5, 128)	0.9589	X	0.9497	363,600
Custom channel 128	9 CONV(5, 512)	0.9396	0.9632	0.9674	2,071,184
And, here is new challenger					
Custom VGG style	16 CONV(3, 128) , 8 Pool	0.9630	0.9423	0.9136	470,736

Audio Classification

- Finally, This is current SOTA(State Of The Art) in my research.

	Architecture (i = 0,1,2...)	1D DO(0.5)	1D BN	1D DO+BN	Params
base model	baseline				
	5 Conv(25, $8 \cdot 2^i$), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
	Accuracy and Number of parameters				
	8 CONV(5, 64)	0.9477	0.9285	0.9391	94,768
Custom channel 32	8 CONV(5, 128)	0.9589	X	0.9497	363,600
Custom channel 64 DO(0.75)	16 CONV(3, 128) , 8 Pool	0.9630	0.9423	0.9136	470,736
Custom VGG style	Only Accuracy				
Custom channel 128	9 CONV(5, 512)	0.9396	0.9632	0.9674	2,071,184

Any Question?

Thank you