1 Dimensional CNN

2020.03.10

Seungmok Lee

Paper Review

- Serkan Kiranyaz, et al., *'1D Convolutional Neural Network and Applications – A Survey'*, (May, 2019)
  - https://arxiv.org/abs/1905.03554
- They summarize the history of CNN, and its state-of-the-art performance.

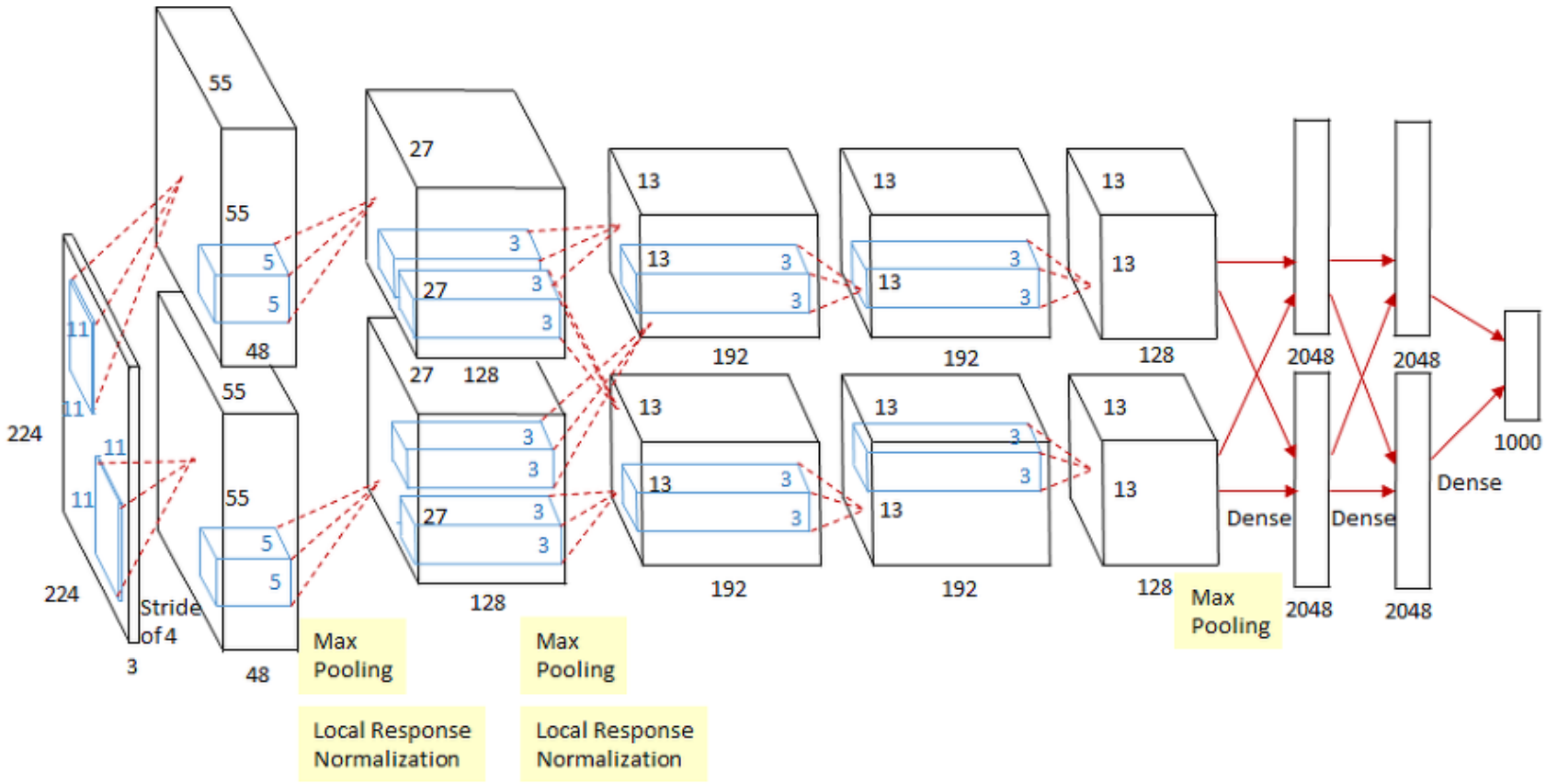
Paper Review – 1D CNN and App.

- History of CNN for image recognition
  - LeNet (Yann LeCun, 1990): First CNN. At that time, Support Vector Machine and Bayesian Network were much powerful.
  - AlexNet (Alex Krizhevsky, 2012): 8-layer CNN made 16.4% error rate for ImageNet database[†], which is 10% more accurate than SVM. ReLU, Dropout, GPU architecture introduced.
  - ZFNet (Zeiler, Fergus, 2013): Error rate 11.7%. Visualized convolutional layer.
  - GoogLeNet (Google, 2014): Error rate 6.7%. 22 layers without computational loss. Ensemble method introduced.

[†]ImageNet database: ~14M images with 1000 classes

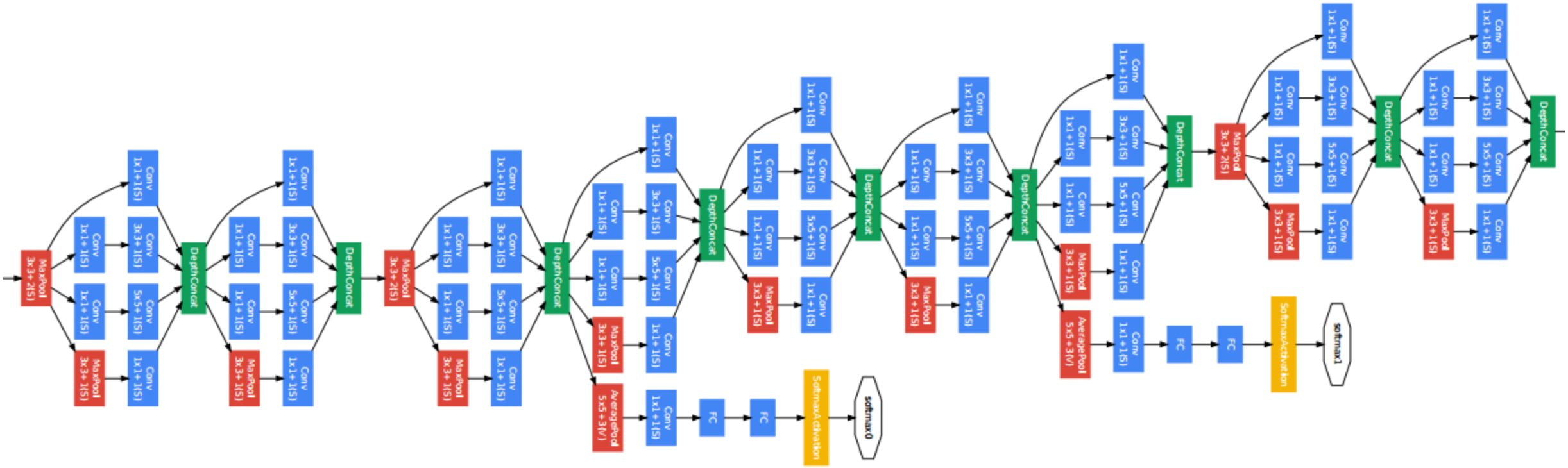
Paper Review – 1D CNN and App.

- Example; AlexNet



Paper Review – 1D CNN and App.

- Example; GoogLeNet (a.k.a. Inception)



Paper Review – 1D CNN and App.

- How about 1D data?
  - Traditionally people converted 1D data into 2D image.
    - Direct reshape

$$\boldsymbol{I} = \begin{bmatrix} x(t) \cdots x(t+n-1) \\ \vdots \\ x(t+(m-1)n) \cdots x(t+mn-1) \end{bmatrix}$$

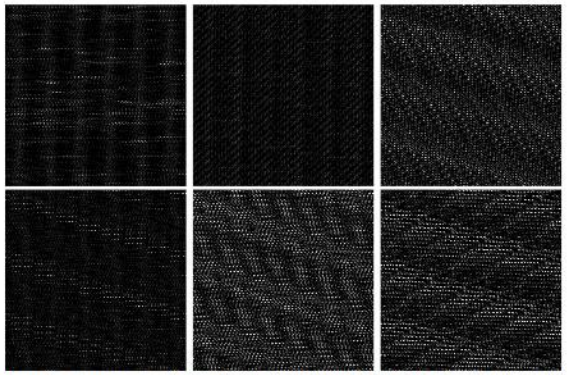
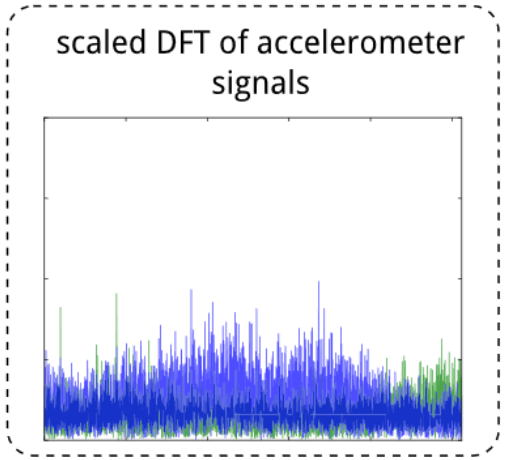


Figure 1. Images of the vibration signals for different bear faults

https://doi.org/10.1051/matecconf/20179513001

    - DFT and stack



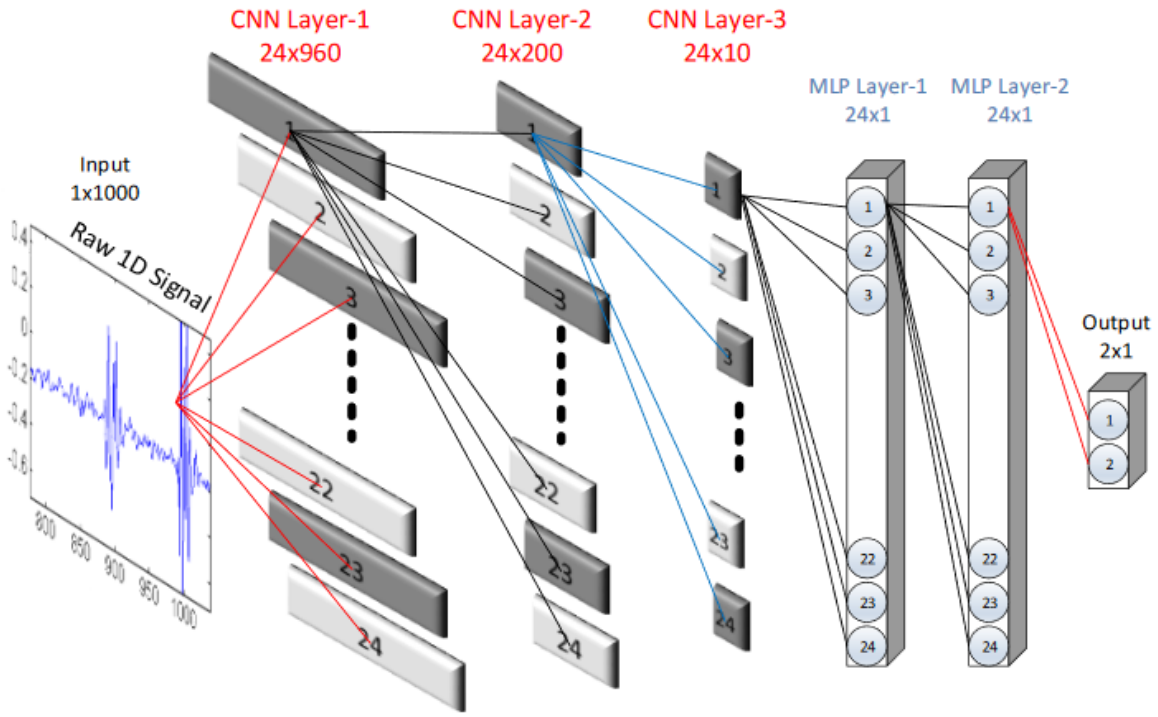
http://lps3.doi.org.libproxy.snu.ac.kr/10.1016/j.jsv.2016.05.027

Paper Review – 1D CNN and App.

- Preprocessing consumes high computational cost.
- Kiranyaz first proposed the 1D CNN to operate directly on the raw ElectroCardioGram data.
  - This made real-time health monitoring possible.
- 1D CNN is now having state-of-the-art performance in various signal analysis.

Paper Review – 1D CNN and App.

- Example 1D CNN applied to ECG.
  - Kiranyaz reported that the kernel_size = 41 was successful for a signal classification!



Paper Review – 1D CNN and App.

- And there are a lot of 1D CNN papers, now!

- If you are interested in the principle of 1D CNN (including FP, BP and other applications), refer to the paper!

First CNN for Cosine

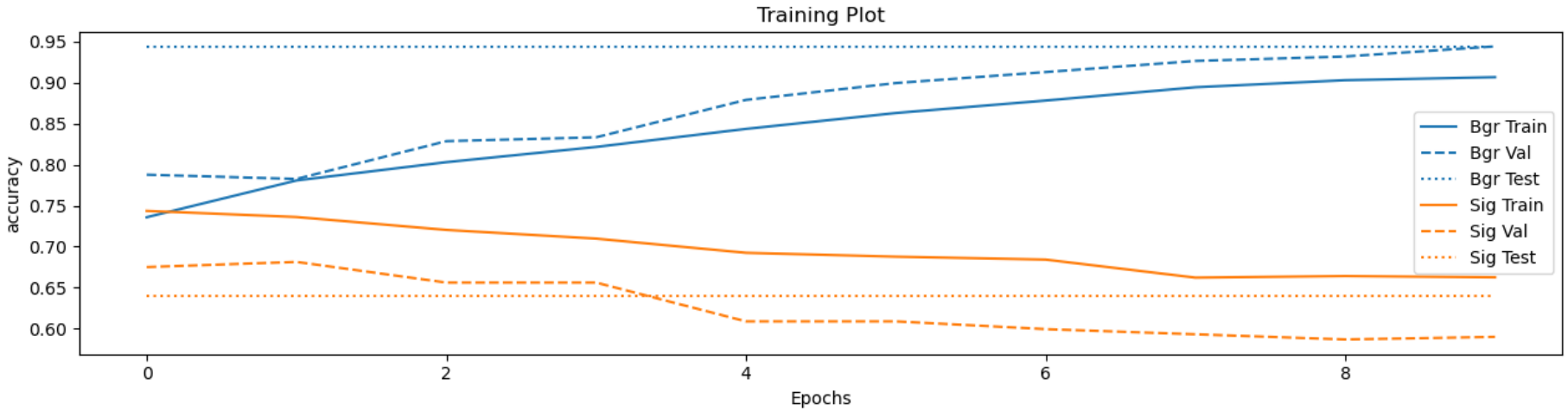
- I implemented and ran CNN for Cosine data!
  - Used data having energy 6~10 keV, from crystal 3.
  - Signal from runnum 1765, 1601 having coincidence.
    - 2540 events
  - Background from runnum 1858, 1859 without coincidence.
    - 63221 events

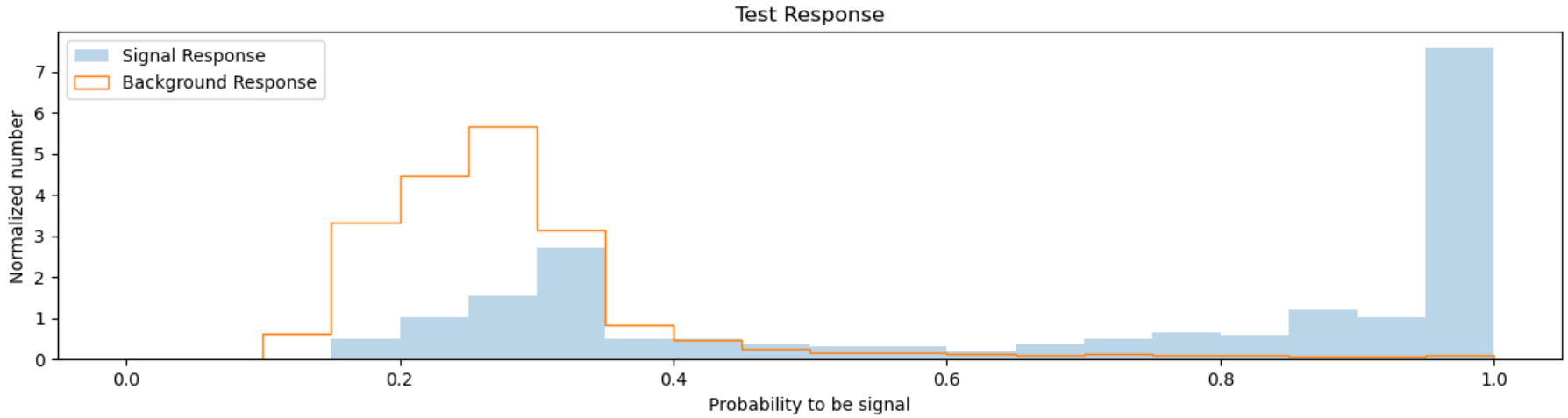
First CNN for Cosine

Input shape	Layer	Output shape
(4080, 2)	Conv1D(filters=64, kernel_size=81)	(4000, 24)
(4000, 24)	MaxPool1D(pool_size=4)	(1000, 24)
(1000, 24)	Conv1D(filters=32, kernel_size=41)	(960, 24)
(960, 24)	MaxPool1D(pool_size=4)	(240, 24)
(240, 24)	Conv1D(filters=32, kernel_size=41)	(200, 24)
(200, 24)	MaxPool1D(pool_size=4)	(50, 24)
(50, 24)	Conv1D(filters=32, kernel_size=41)	(10, 24)
(10, 24)	Flatten()	240
240	Dense(24)	24
24	Dense(24)	24
24	Dense(2)	2

Motivated from Kiranyaz ECG Network.
Batch normalization, Dropout, ReLU activation applied.

Not Satisfying Result



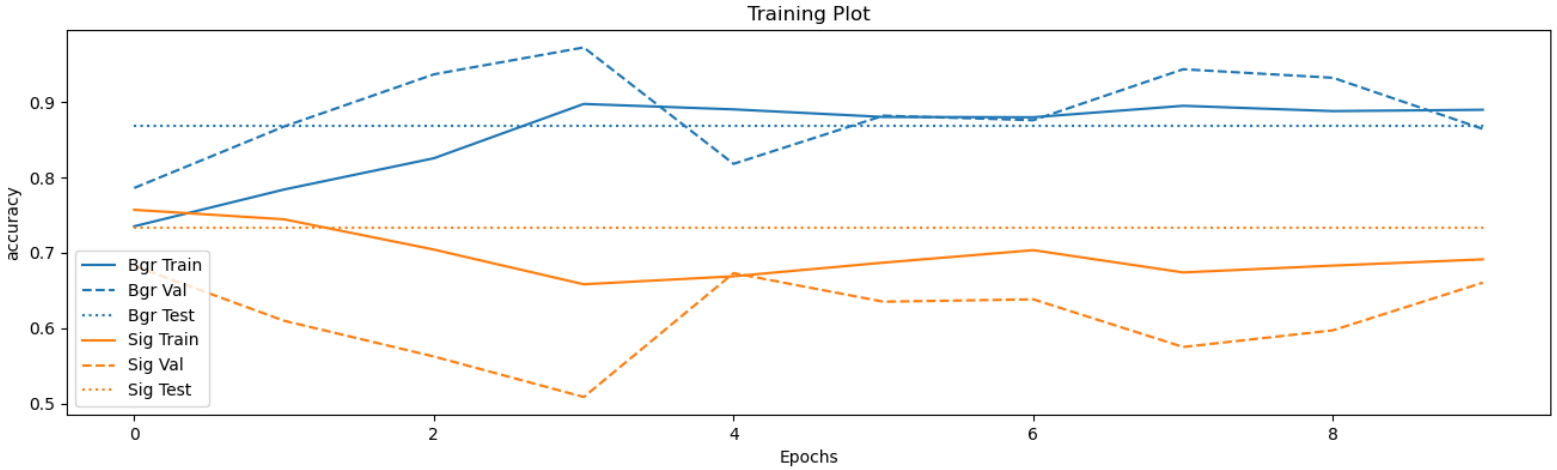


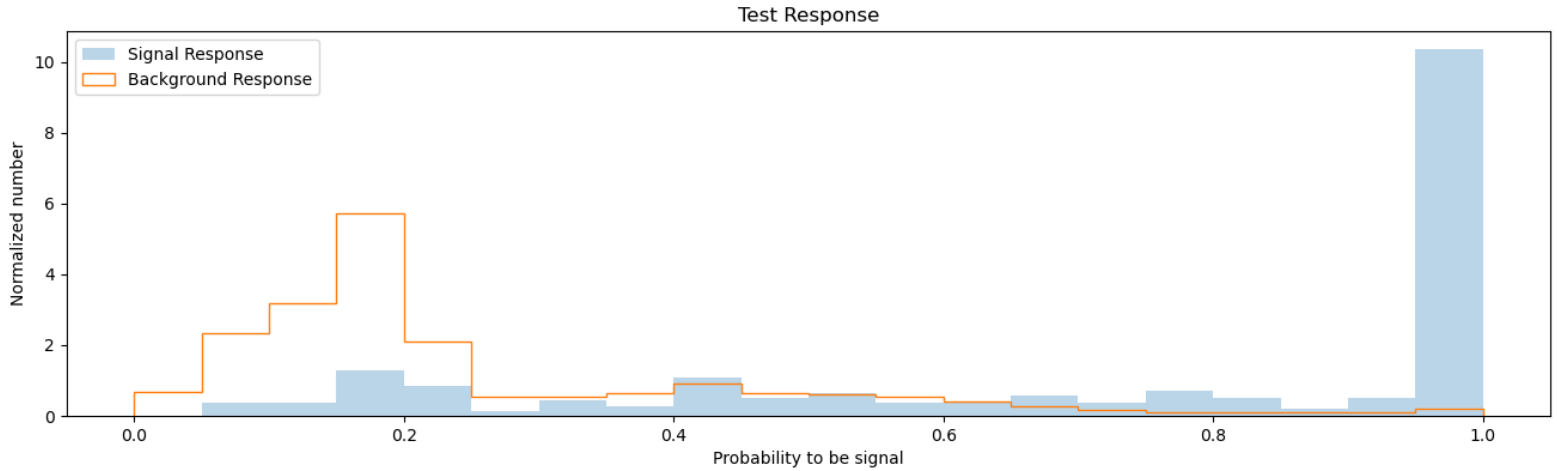
Second CNN for Cosine

Input shape	Layer	Output shape
(4080, 2)	Conv1D(filters=96, kernel_size=39, strides=3)	(1348, 96)
(1348, 96)	MaxPool1D(pool_size=4)	(337, 96)
(337, 96)	Conv1D(filters=256, kernel_size=23)	(315, 256)
(315, 256)	MaxPool1D(pool_size=3)	(105, 256)
(105, 256)	Conv1D(filters=384, kernel_size=9)	(97, 384)
(97, 384)	Conv1D(filters=384, kernel_size=9)	(89, 384)
(89, 384)	Conv1D(filters=256, kernel_size=9)	(81, 256)
(81, 256)	MaxPool1D(pool_size=3, strides=2)	(40, 256)
(40, 256)	Flatten()	10240
10240	Dense(512)	512
512	Dense(512)	512
512	Dense(512)	512
512	Dense(512)	512
512	Dense(2)	2

Motivated from AlexNet. Batch normalization, ReLU activation applied.

Still Not Satisfying

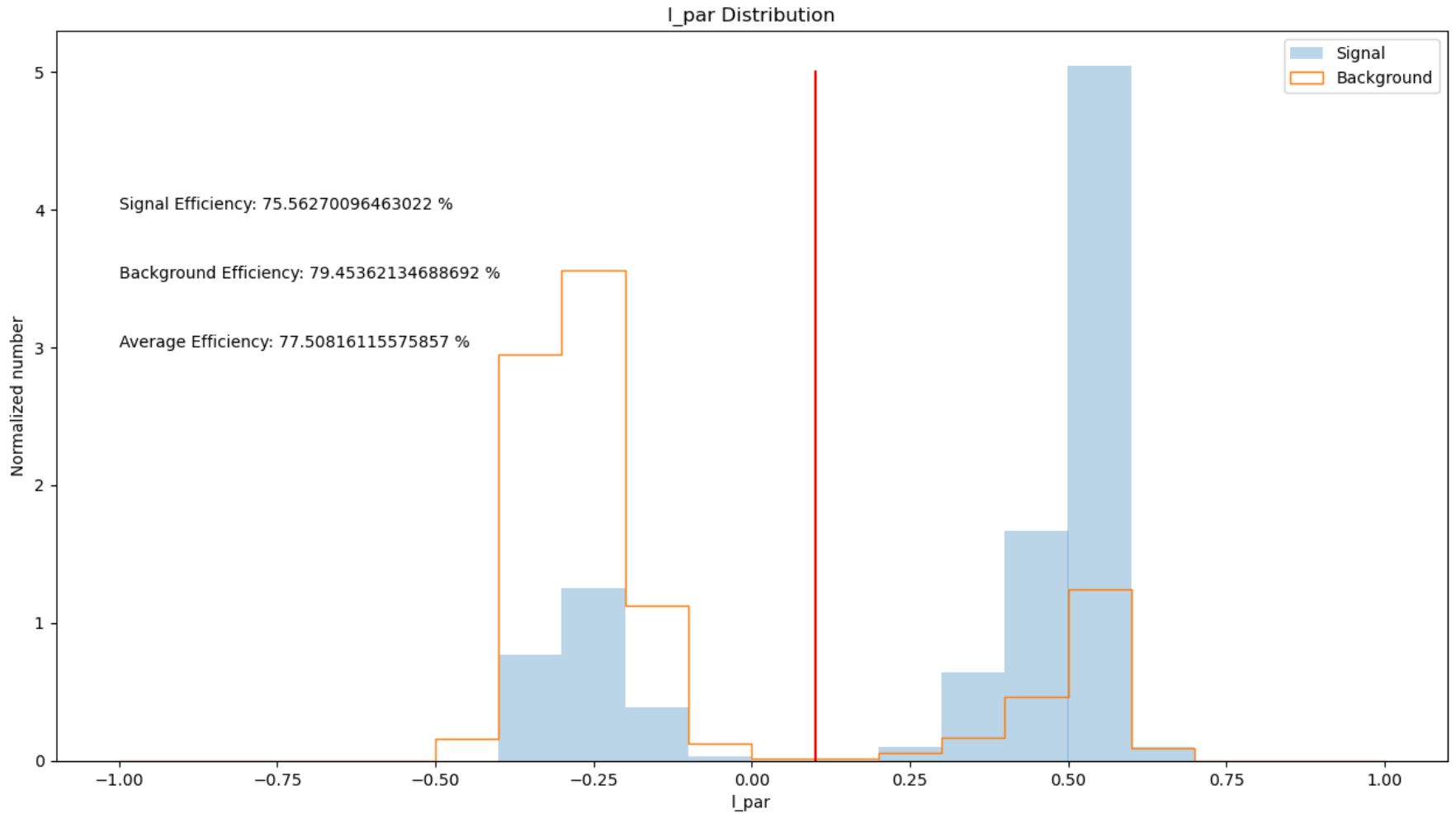




Data Impurity : Comparison With lpar

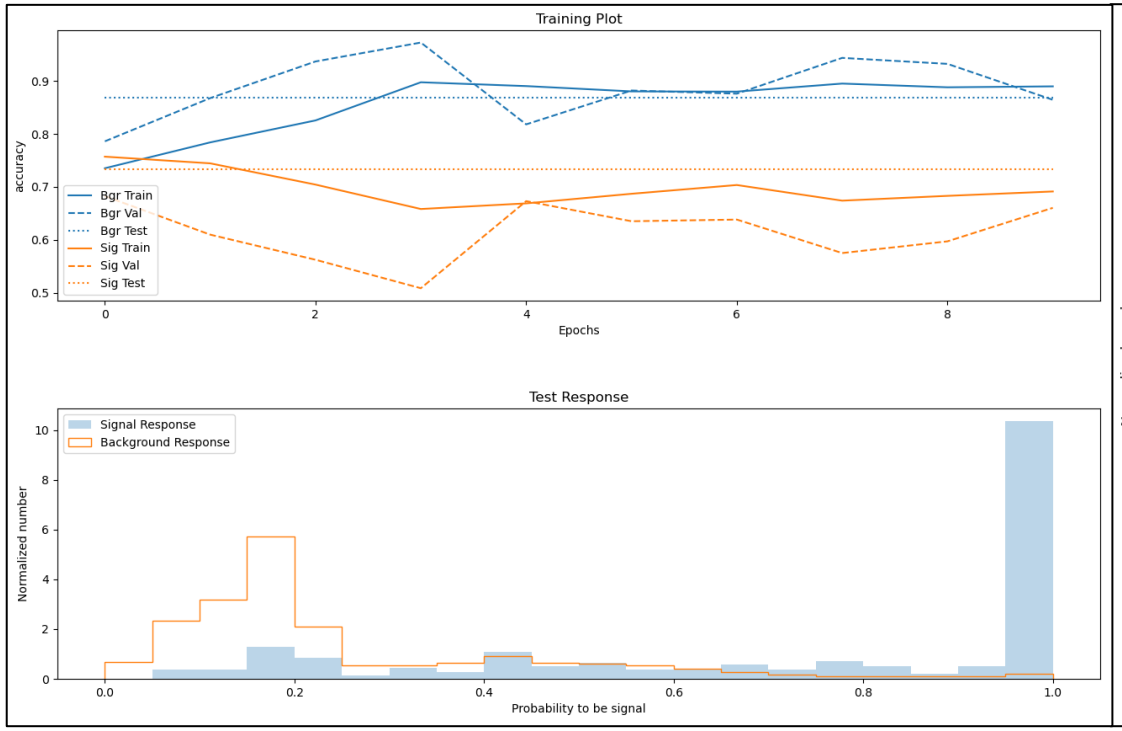
- It was awkward that CNN did not improve the performance at all.
- To check the data impurity, I observed the lpar distribution.

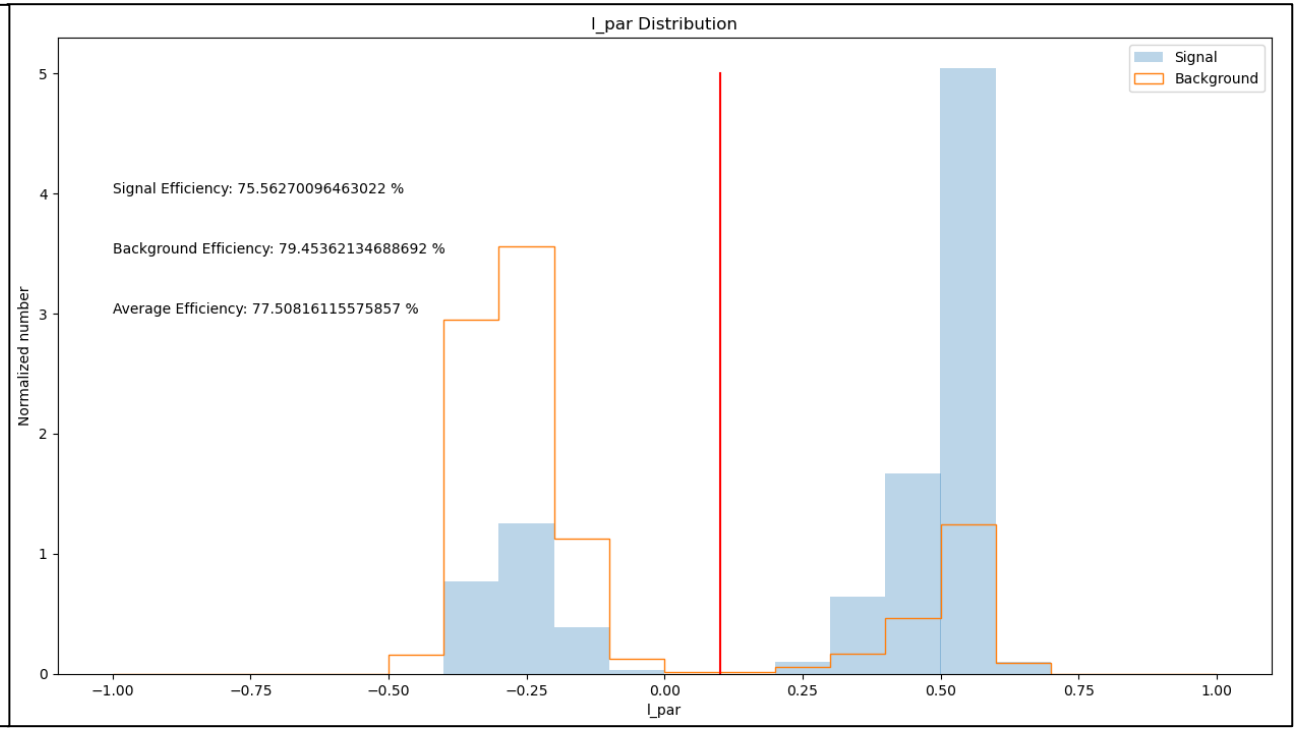
Data Impurity : Comparison With lpar



Data Impurity : Comparison With lpar

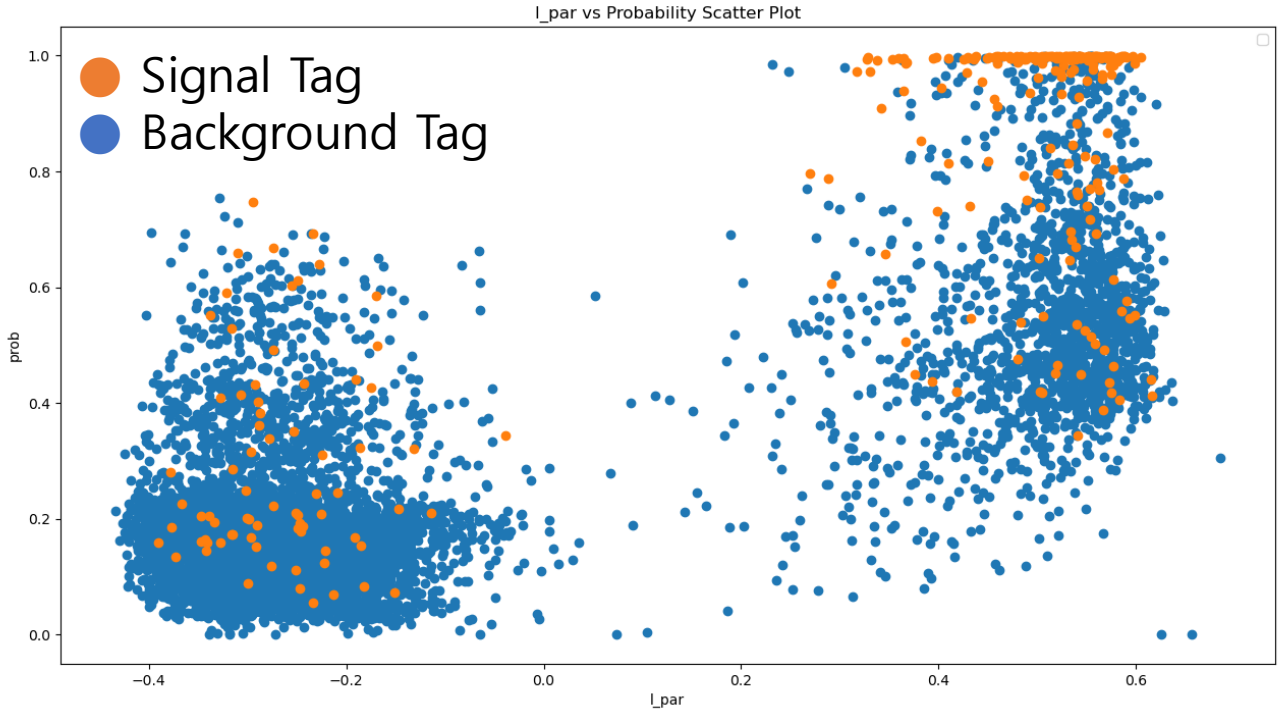
- Lpar was not consistent with my tagging!
- My CNN have already reached lpar-level performance.





Data Impurity : Comparison With lpar

- Lpar and the probability (by network) showed weak correlation.
- The first thing I have to do is maybe data checking!



Data Impurity : Code Review

- I’ve reviewed my code, but I could not find any error.
- Help!