

話題提供 6

NNCを用いた音源位置の学習について

NNC: ニューラルネットワークコンソール (SONY)

2022/6/1

長崎県工業技術センター

基盤技術部 機械システム科

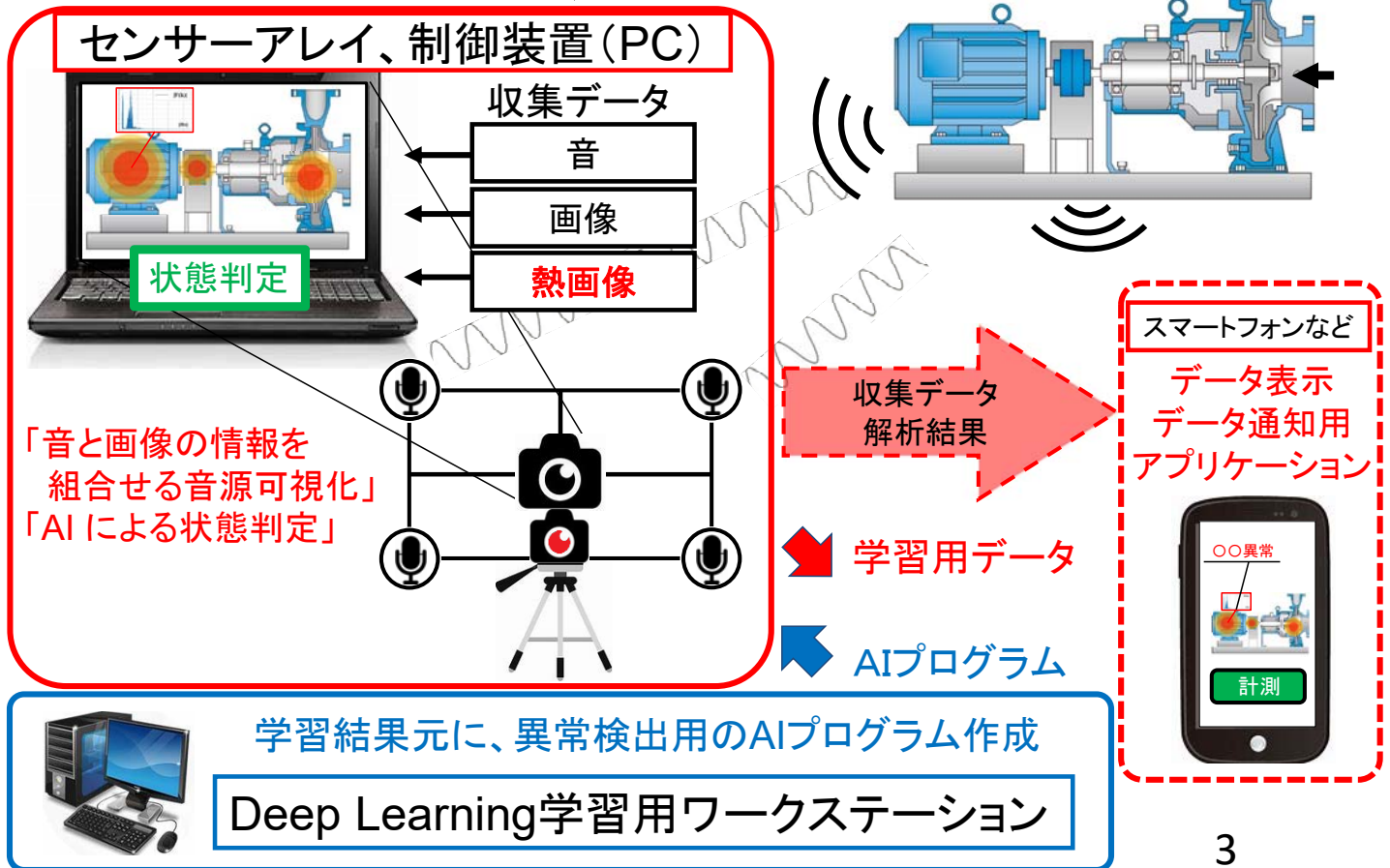
久保田 慎一

1

目次

- ・ 経常研究の内容
 - 開発装置の概要
 - 音源探査の原理について
- ・ 音源位置の学習
 - 簡易のマイクアレイ試作
 - 集音実験
 - NNCによる音声データの学習

開発装置の概要



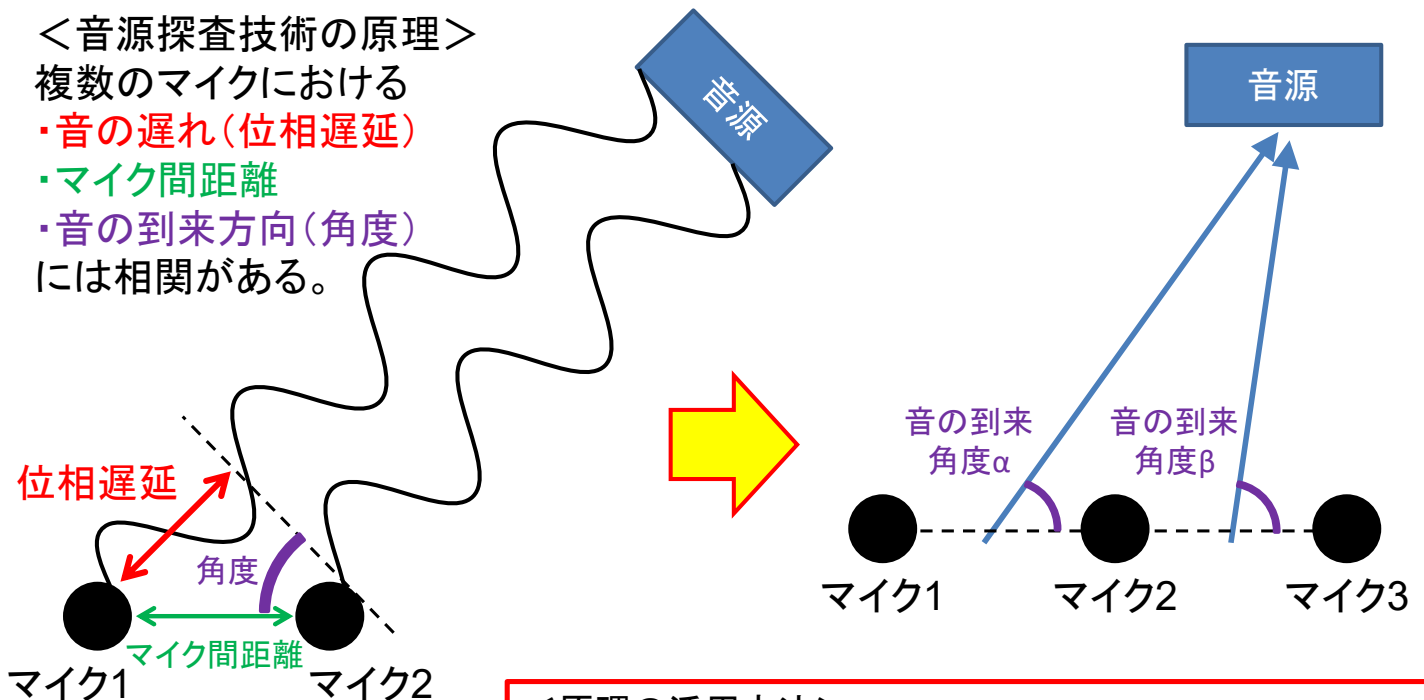
音源探査の原理について

<音源探査技術の原理>

複数のマイクにおける

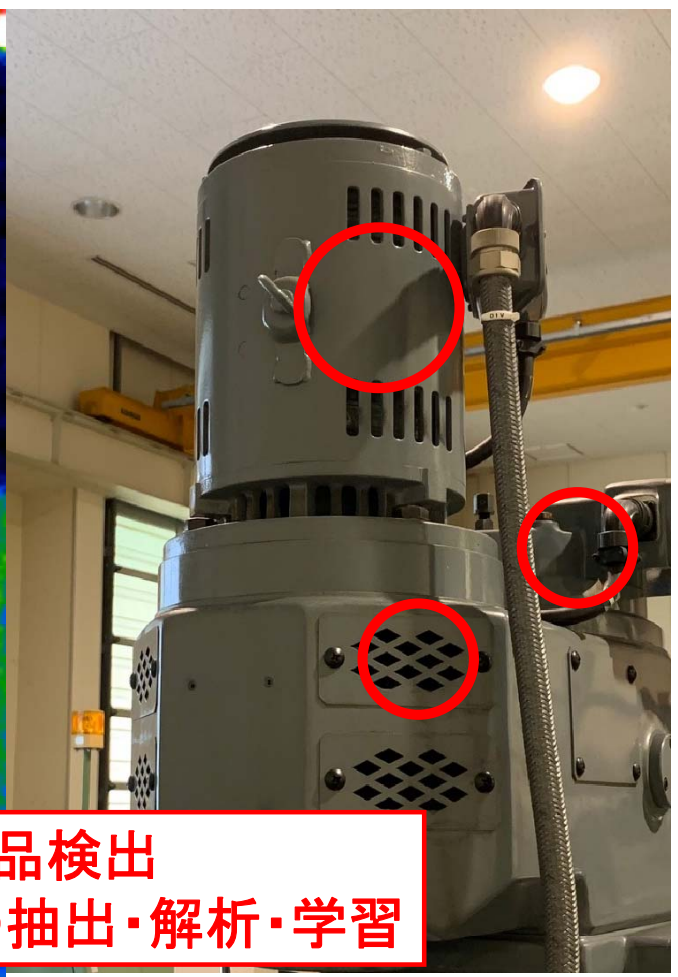
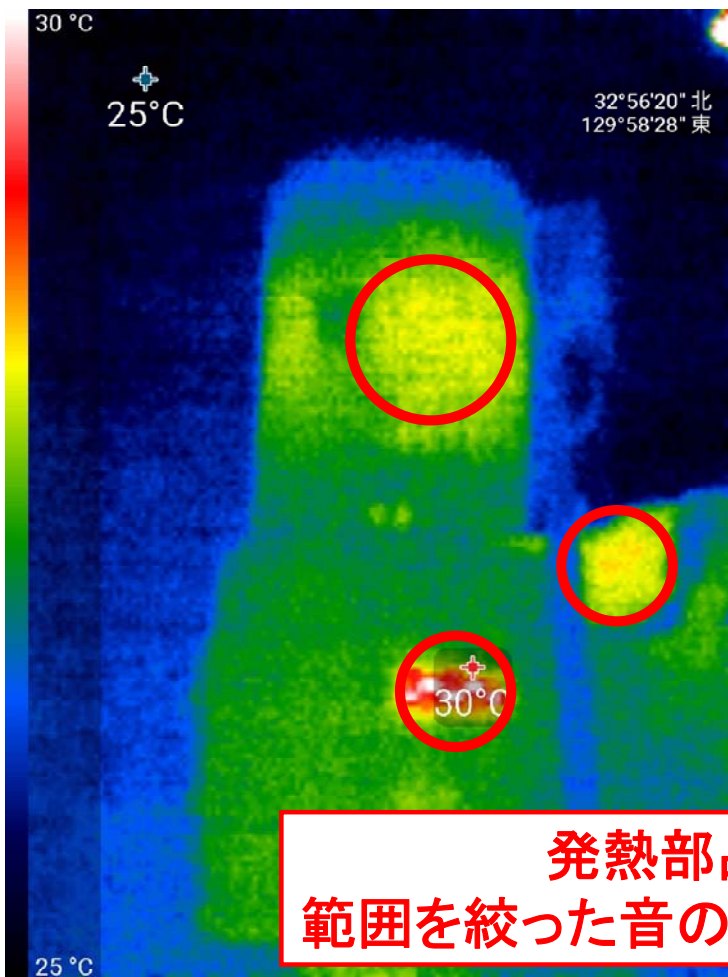
- ・音の遅れ(位相遅延)
- ・マイク間距離
- ・音の到来方向(角度)

には相関がある。



<原理の活用方法>

- ・各マイクの位相遅延を解析 \Rightarrow 角度(音源位置)の特定
- ・角度(音源位置)を指定 \Rightarrow 位相遅延を考慮した集音



発熱部品検出
 範囲を絞った音の抽出・解析・学習

<音源可視化技術の開発>

簡易のマイクアレイ試作



<仕様>

マイク: 6本

Behringer ECM8000

レコーダー:

ZOOM H8

iPad(遠隔操作)



集音実験



<仕様>

スピーカー:

ヤマハ DBR12

ノイズジェネレーター:

小野測器 DS-3000

<実験条件>

場所: 無響室

音源位置: 8箇所(A~D、A2~D2)

出力信号種類: 2種類

(RANDOM、PSEUDO RANDOM)

録音時間: 600 [s]

<音声データ>

WAVデータ(6ch) × 10分 × 16ケース

NNCによる音声データの学習

<データセット作成>

WAVデータの分割プログラム

CSVファイルで相対ファイルパスを指定

```

1 import wave
2 import struct
3 import math
4 import os
5 from scipy import fromstring, int16
6
7 # 出力に同じ名前のディレクトリがないか確認
8 file = os.path.exists("output")
9 print(file)
10
11 if file == False:
12     # 出力先のディレクトリを作成
13     os.mkdir("output")
14
15 def cut_wav(filename, time):
16     # 元のwavの長さ(sec)
17
18     # データを読み込む
19     wavf = filename + ".wav"
20     wf = wave.open(wavf, "r")
21
22     # wavファイルが持つ情報を取得
23     ch = wf.getchansels()
24     width = wf.getsampwidth()
25     fr = wf.getframerate()
26     fn = wf.getframes()
27     total_time = 1.0 * fn / fr
28     integer = math.floor(total_time*100) # 小数点以下の処理
29     t = int(time*100) # 整数化(sec)
30     frames = int(ch * fr * t/100)
31     num_cut = int(integer/t)
32
33     # 出力
34     print("Channel: ", ch)
35     print("Sample width(1:8bit, 2:16bit, 3:24bit): ", width) # 2と16を指定
36     print("Frame Rate: ", fr)
37     print("Frame num: ", fn)
38     print("Params: ", wf.getparams())
39     print("Total time: ", total_time)
40     print("Total time(integer): integer)
41     print("CUT Time: ", t)
42     print("Frames: ", frames)
43     print("Number of cut: ", num_cut)
44
45     # wavのデータを取得し、分割
46     data = wf.readframes(frames)
47     wf.close()
48     x = fromstring(data, dtype=int16)
49     print(x)
50
51     for i in range(num_cut):

```

	A	B
1	x__0	x__1
2	.¥A¥Tr1¥500.wav	.¥A¥Tr2¥500.wav
3	.¥A¥Tr1¥501.wav	.¥A¥Tr2¥501.wav
4	.¥A¥Tr1¥502.wav	.¥A¥Tr2¥502.wav
5	.¥A¥Tr1¥503.wav	.¥A¥Tr2¥503.wav
6	.¥A¥Tr1¥504.wav	.¥A¥Tr2¥504.wav
7	.¥A¥Tr1¥505.wav	.¥A¥Tr2¥505.wav

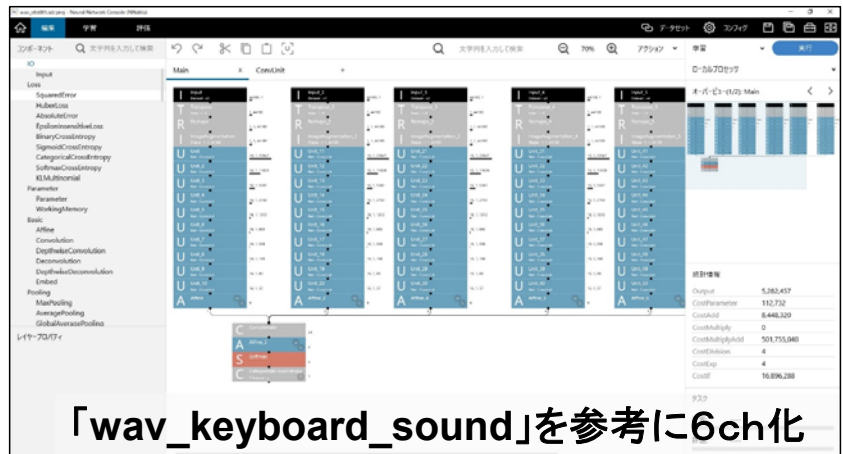
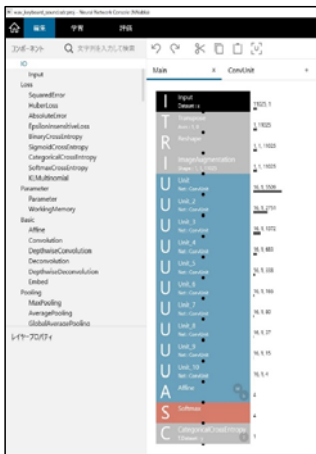
<データセット>

学習用: WAVデータ(6ch) × 1秒 × 500個

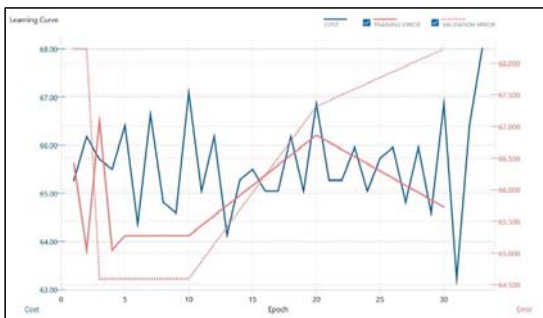
評価用: WAVデータ(6ch) × 1秒 × 100個

NNCによる音声データの学習

<NNC_サンプルプロジェクトからNNを作成>



「wav_keyboard_sound」を参考に6ch化



学習曲線

学習が収束しない原因

- ・6chの音声を個別に畳み込み演算
- ・6chをまとめる前に平均化

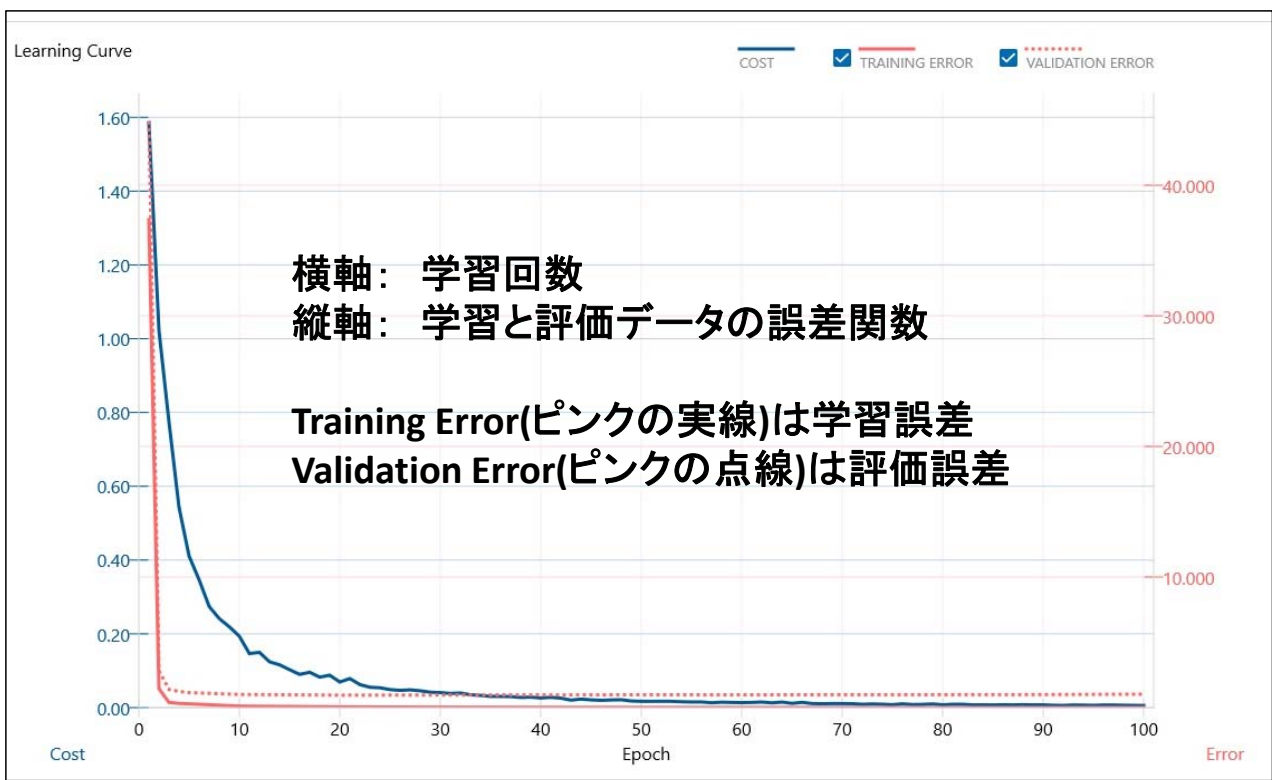


「6chデータ指定方法を、関連のある6次元化」

参考URL: <https://support.dl.sony.com/docs-ja/>

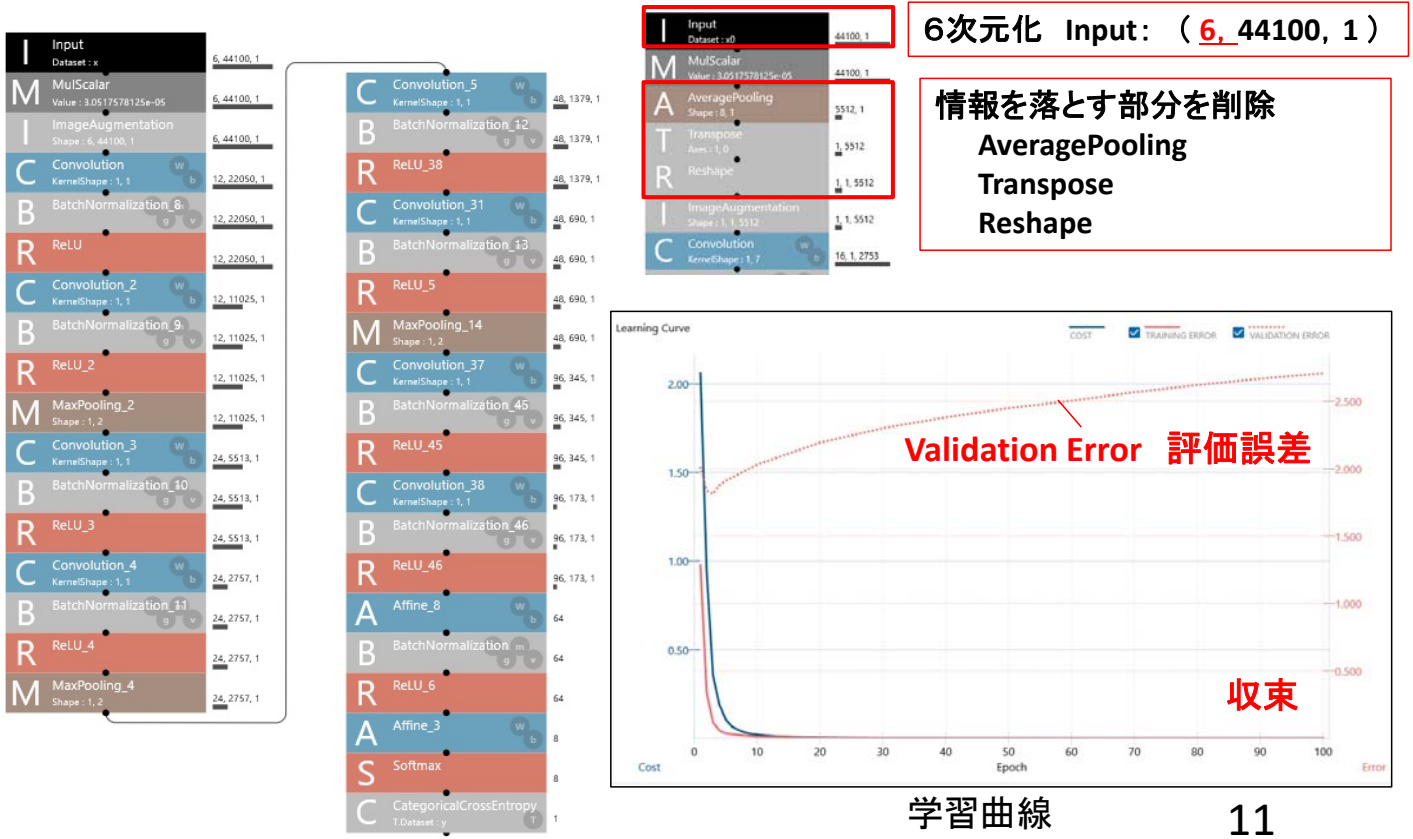
NNCによる音声データの学習

<学習曲線の見方について>



NNCによる音声データの学習

< NNC_NNを改良、データセットを増による検証 >



NNCによる音声データの学習

< NNC_NNを改良、データセットを増による検証 >

	y'_0 A	y'_1 B	y'_2 C	y'_3 D	y'_4 A2	y'_5 B2	y'_6 C2	y'_7 D2
y=0	122	1	1	0	75	1	0	0
y=1	7	132	1	3	9	48	0	0
y=2	9	0	148	1	3	0	39	0 誤認
y=3	0	0	1	97	0	2	1	99
y=4	126	2	1	0	68	1	2	0
y=5	6	108	0	0	5	81	0	0
y=6	1	0	141	0	10	0	48	0
y=7	0	1	6	83	0	5	5	100

<結果>

上下左右の4分類は精度が出たが、内側と外側の誤認識が多かった。

6chの同期精度やマイクの影響による影響の検証

NNの設計による影響の検証

熱画像を追加したデータセットの作成

角度を指定した集音プログラムの作成

などを実施していく。

正解

ご清聴ありがとうございます。

今後とも、積極的に情報交換などできたら幸いです。
よろしく願いいたします。

長崎県工業技術センター 機械システム科 久保田慎一
kubota@tc.nagasaki.go.jp