一般社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会資料
JSAI Technical Report
SIG-Challenge-049-7 (11/25)

DNN Based Pitch Estimation Using Microphone Array*

Jani Even, Carlos Toshinori Ishi, Hiroshi Ishiguro
Hiroshi Ishiguro Laboratories, Advanced Telecommunications Research Institute International, Japan.
even@atr.jp *

Abstract

This paper presents some preliminary experiment for pitch classification of distant speech recorded with a microphone array. The pitch classification is performed by a deep neural network. Using the microphone array to perform beamforming is beneficial to the pitch classification. However it requires a larger amount of data for training the network. The network seems to be robust to data miss-matched as pretraining with close speech data improved the results for distant speech.

1 INTRODUCTION

This paper presents some preliminary results on the use of deep neural network for pitch classification. In particular, the goal is to investigate the possible improvement obtained by applying beamforming when considering distant speech. Pitch classification using neural network was applied by the authors of [1] using hand engineered features. Recent advance in neural networks make it possible to train deep architecture [2] that learns the features. In [3], the authors proposed different deep neural networks to estimate pitch. However, distant speech and the use of microphone array was not investigated.

2 OVERVIEW

Figure 1 shows the overview of the training phase and the testing phase. In the two phases, the voice of the subject was recorded using a linear microphone array (8 microphones with a spacing of 0.02 m) and a tie microphone. All the microphones are similar omni-directional microphones.

In order to access the improvement obtained by using the microphone array, the features are either extracted from one of the microphone of the microphone array or from a delay and sum beamformer (see [4] for microphone array processing). In the remainder of the paper, the features extracted from the single microphone are denoted as "far" features. The features extracted from the output of the delay and sum beamformer are denoted as "DS" features.

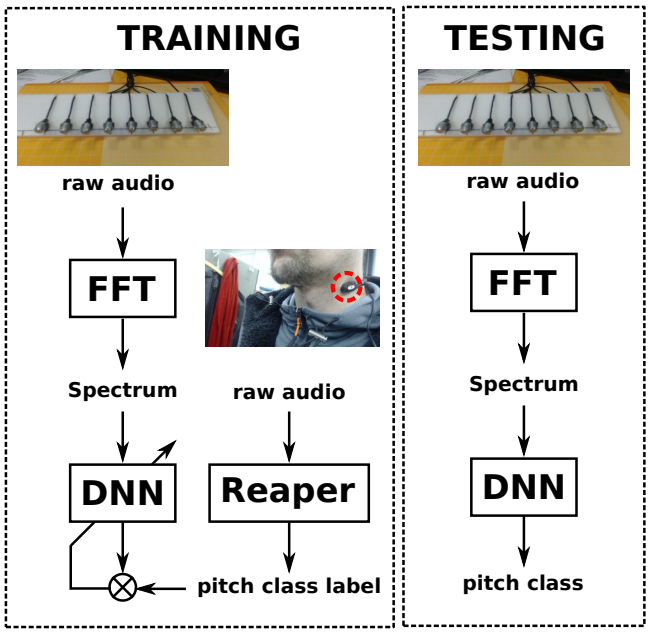

Figure 1: Overview of the training (left) and the testing (right).

The labels for pitch are extracted from a tie microphone by the "reaper" software [5]. This software simultaneously estimates the location of glottal closure instants, voicing state and pitch. Only the pitch estimation was used to label the data. Some experiments were conducted using the PRAAT software [6] to estimate the pitch label. The reaper software was preferred because it was easier to automate the labeling task with it.

3 DATA COLLECTION

The data corpus consists of approximately 42 minutes of speech data recorded from a single male speaker.

*Research supported by the JST ERATO Ishiguro Symbiotic Human-Robot Interaction Project.

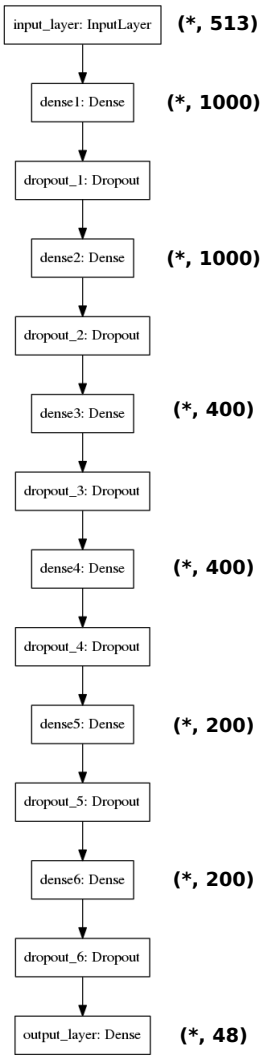

Figure 2: Architecture of the network.

The sampling frequency is 16 kHz. The audio data is transformed in the frequency domain with using a sliding hanning window of 320 ms with half overlap. The FFT size is 1024. The feature vector for each frame is the modulus of the 513 positive frequency components.

During extraction with the reaper software, the pitch values were limited to the range $[50, 300]$ Hz. The range $[50, 280]$ Hz was linearly divided in 46 bins (5 Hz per bin). A bin for non voiced frames (bin 1) and another bin (bin 48) for voiced frames over 280 Hz were also created. Thus the total number of pitch classes is 48.

The data was separated in testing set and development set. The development data is further split in training and validation set.

4 NEURAL NETWORK TRAINING

The pitch classifier is composed of 6 fully connected layers. The input layer has the feature vector size $F = 513$. The output layer is of size $C = 48$ corresponding to the number of pitch classes. Figure 2 shows the network and the number of units in the different layers. All the activations in the layers are "softplus" except for the output

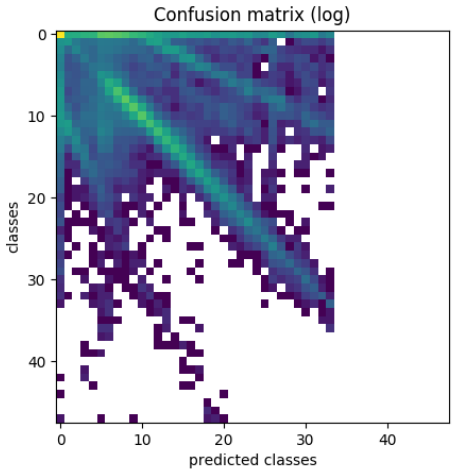

Figure 3: Confusion matrix in log scale for the "far" features without pre-training.

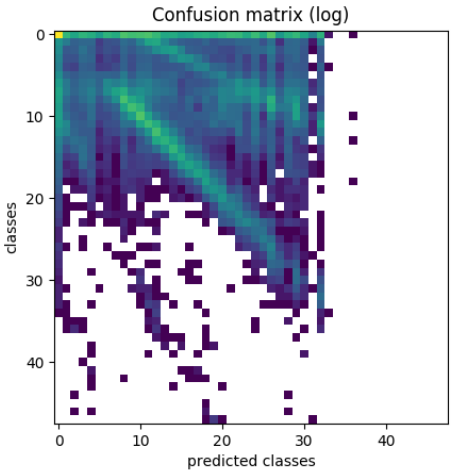

Figure 4: Confusion matrix in log scale for the "DS" features without pre-training.

layer that is using a "softmax". This neural network part is implemented using Keras [7] with Tensorflow backend [8]. The network has a total of $2,205,848$ parameters.

The network was either initialized randomly or using some pre-trained weights. The pre-trained weights were obtained by training the network with audio data from the Librispeech database [9]. LibriSpeech is a corpus of 16kHz read English speech. We used 100 hours of clean speech to train the model. During the pre-training, the pitch labels and the features are extracted from the same signal. Namely, both the features and the pitch labels are from close speech.

Since we are considering a classification problem, the network was trained to optimize the categorical cross entropy. When training from random weights, the adaptive subgradient method ("Adagrad" in Keras) was used to update the weights [10]. When using pre-trained weights, in order to move slowly away from the initial weights, a stochastic gradient descent with a small step size was used. For all the operations, batch of 100 samples were used.

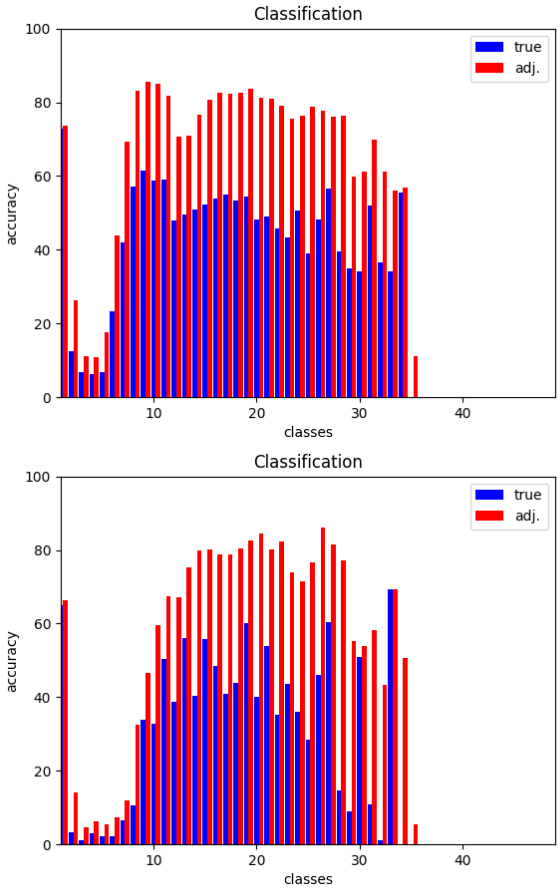

Figure 5: Classification results for the "far" (top) and "DS" (bottom) features without pre-training.

After the training, in each of the cases, the network having the best score on the validation set was selected.

5 CLASSIFICATION PERFORMANCE

5.1 Without pre-traing

Figure 3 shows the confusion matrix (log scale) for the classification of the "far" features and figure 4 shows the corresponding confusion matrix for the "DS" features.

The confusion matrices clearly show that the classification error is usually because of assignment to the neighboring bins. Thus, we give the results in term of true classification accuracy (classification in the true bin) and adjacent classification accuracy (classification in the true bin or its two immediate neighboring bins).

Figure 5 shows the classification results for both sets of features. The classification results when using the output of the delay and sum are worse. Moreover for both cases, the higher bins are not well classified.

5.2 With pre-training

The pre-training phase was added in order to improve the poor results obtained by direct training of the network.

The confusions matrices in figures 6 and 7 are still showing some errors between neighboring bins.

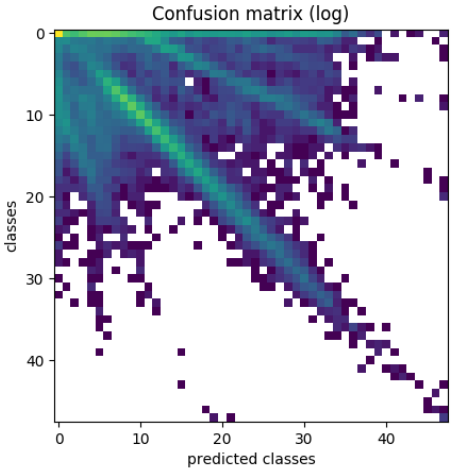

Figure 6: Confusion matrix in log scale for the "far" features with pre-training.

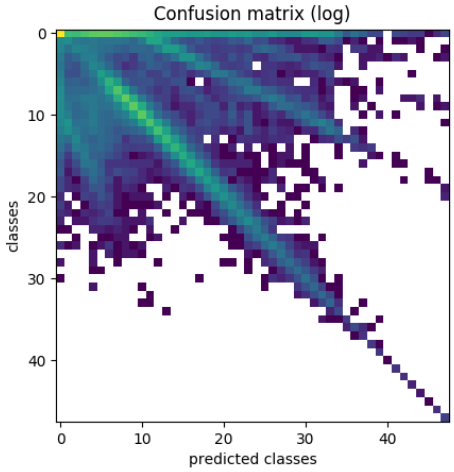

Figure 7: Confusion matrix in log scale for the "DS" features with pre-training.

However, the classification results in 8 are greatly improved by the addition of the pre-training phase. In particular, the results for the microphone array ("DS" features) are better than those for the "far" features.

The higher classes that were not well classified are now perfectly classified. The reason is that these classes were under represented in the dataset but appear in the data used for pre-training. This can be observed in the confusion matrices.

It is important to notice that the pre-training was performed using completely miss-matched data as the LibriSpeech corpus contains close talking speech recorded with a single microphone. This is an interesting results as it suggests that the DNN based pitch classifier is quite robust to data miss-match.

6 CONCLUSIONS

The experiment presented in this paper shows that using a microphone array to perform delay and sum beamforming improves the DNN based pitch classification of

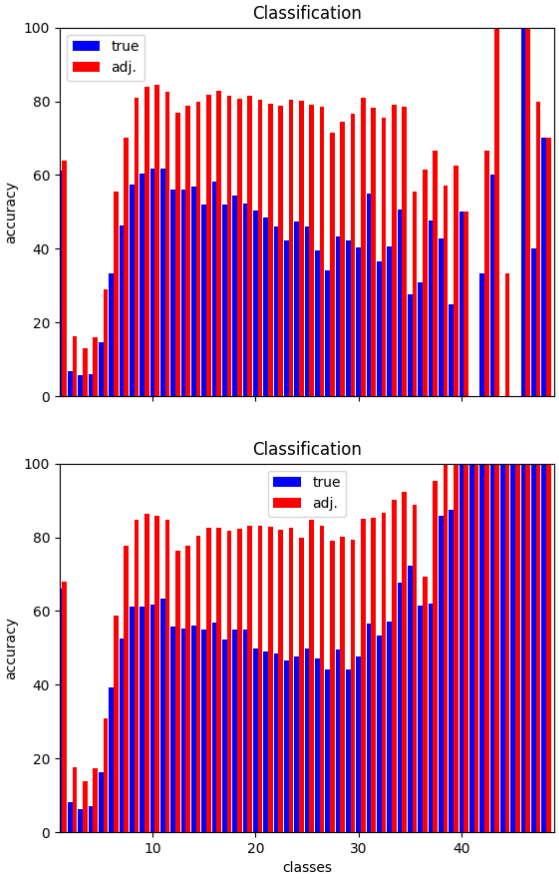

Figure 8: Classification results for the "far" (top) and "DS" (bottom) features with pre-training.

distant speech. However, it seems that a larger amount of training data is necessary as the microphone array results were only better than those of the single distant microphone when a large amount of data was used for pre-training. That improvement was very clear even if the pre-training was done with miss-matched conditions. The future research is testing this in controlled noisy environments in order to clearly assess the performance. Another point of interest is to access the robustness of the DNN based pitch classifier to data miss-match.

References

[1] E. Barnard, R. A. Cole, M. P. Vea, and F. A. Alleva, "Pitch detection with a neural-net classifier," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 298–307, 1991.

[2] G. E. Hinton, S. Osindero, and Y.W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[3] K. Han and D. Wang, "Neural network based pitch tracking in very noisy speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2158–2168, 2014.

[4] J. DiBiase, H. Silverman, and M. Brandstein, *Microphone arrays : Signal Processing Techniques and Applications*, Springer-Verlag, 2007.

[5] David Talkin, "Reaper," `https://github.com/google/REAPER`, 2015.

[6] P. Boersma and D. Weenink, "Praat version 5.3.02," `http://www.praat.org/`, 2011.

[7] François Chollet et al., "Keras," `https://github.com/fchollet/keras`, 2015.

[8] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[9] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 5206–5210.

[10] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.