BAYESIAN CONVOLUTIONAL NEURAL NETWORK BASED DOMINANT INSTRUMENT RECOGNITION IN POLYPHONIC MUSIC

Canberk Ekmekci

University of Rochester
Department of Electrical and Computer Engineering
Rochester, NY, USA

ABSTRACT

In this paper, we propose a framework to quantify the model uncertainty in neural networks used for dominant instrument recognition in polyphonic music. The proposed approach combines a convolutional neural network-based dominant instrument recognition method with recent advancements in representing uncertainty in deep neural networks. The proposed method does not introduce any additional parameters to the model, and prediction and model uncertainty can be obtained very efficiently. We tested the proposed framework on the IRMAS dataset [1] and the proposed framework achieved the micro F1 score of 0.551 and macro F1 score of 0.465.

Index Terms— Instrument recognition, convolutional neural networks, Bayesian neural networks, epistemic uncertainty

1. INTRODUCTION

Identifying the dominant instruments in a polyphonic audio plays a vital role in several applications such as music transcription and music genre classification.

Recently, there have been studies in which deep learning techniques are used to perform dominant instrument recognition (see [2, 3], and the references therein). The main idea is to use deep learning techniques to obtain feature vectors from standard features in the audio processing literature such as bandwidth, roll-off rate, zero-crossing rate, spectrogram, mel-scaled spectrogram and MFCC. Main problem with the standard deep learning techniques is that they do not offer model uncertainty information. In other words, we cannot understand whether the model is confident about the prediction itself or not. Model uncertainty is a valuable information since it can be leveraged in many applications in practice. For example, in the case of dominant instrument recognition, model uncertainty can be used to discard the frames which the model is not confident about the prediction.

To obtain the model uncertainty information, we must employ Bayesian approach, which requires evaluating the posterior distribution of the weights of the neural network and calculating the predictive distribution. In deep neural networks, representing model uncertainty is not a straightforward task because of non-linearities and deep architectures. Hence, approximation techniques such as variational inference and Monte Carlo integration is a must. In [4], Gal and Ghahramani showed that there is an efficient way to obtain the prediction and the model uncertainty information under some assumptions. We will talk about their approach in detail in Section 2.

In this paper, we propose a framework to perform dominant instrument recognition in polyphonic audio and to obtain model uncertainty information simultaneously. The proposed framework is easy to implement on existing deep learning frameworks such as PyTorch [5], and recognition and model uncertainty quantification can be performed very efficiently.

This paper is organized as follows. Section 2 provides a background on the convolutional neural network-based dominant instrument recognition and Bayesian deep learning approach introduced in [4, 6]. Section 3 introduces the proposed framework, and we test the proposed approach on IRMAS dataset [1] in Section 4. Section 5 concludes the paper.

2. BACKGROUND

2.1. CNN based Instrument Recognition

Due to the success of neural networks in different research fields, there have been some studies applying convolutional neural networks to dominant instrument recognition (see [3, 2] and references therein). The main idea is to use features, e.g., spectrogram, mel-scale spectrogram, and MFCC, as inputs to a convolutional neural network. For example, [2] concatenates spectral centroid, RMS, zero-crossing rate, spectral roll-off, bandwidth, mel-scaled spectrogram, and MFCC of a single frame to obtain a 153-dimensional feature vector. Then, by stacking those feature vectors as row vectors into a matrix, we can obtain a two-dimensional feature vector, which is suitable for being used as input to a CNN. [3] uses the same idea; the only difference is that it only uses stacked mel-scaled spectrograms as a two-dimensional feature vector.

The problem is that sound files vary in length and may

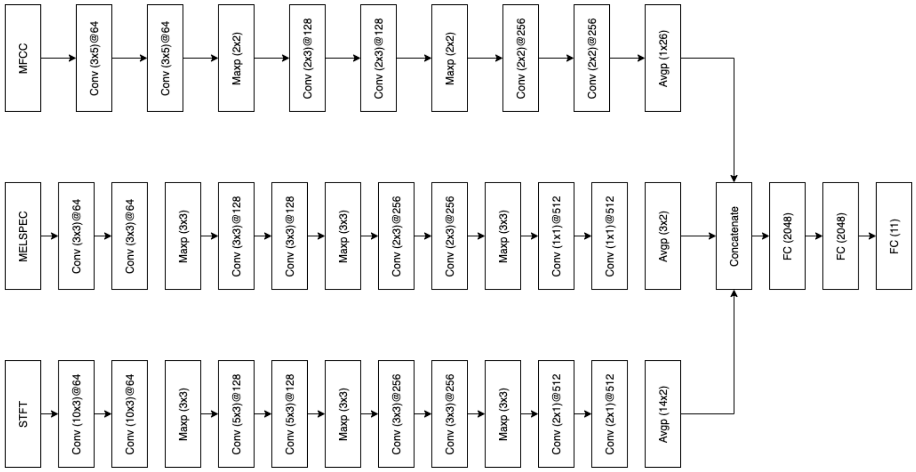

Fig. 1. Proposed Bayesian convolutional neural network model.

contain multiple dominant instruments. A straightforward solution to this problem is to divide the input audio clip into frames and feed them into the convolutional neural network separately. Then, the outputs of the neural network for each frame are used to make a file-level decision. [3] proposes a simple but efficient decision strategy: take the average of softmax outputs and perform thresholding. Then, the values of the softmax units exceeding the threshold determine the dominant instruments.

2.2. Dropout and Bayesian Neural Networks

To obtain the model uncertainty information, we have to perform Bayesian analysis. To perform Bayesian analysis in the context of neural networks, we need to find the posterior distribution of the weights of the neural network. Then, the posterior distribution can be used to evaluate the predictive distribution. However, computing the posterior is challenging, and obtaining predictive distribution relies on the correctness of the posterior and requires high dimensional integrals. To overcome these problems, we need to use approximation techniques such as variational inference. In the variational inference framework, we propose a candidate distribution parametrized by a finite number of parameters for the posterior and try to adjust the parameters of the candidate distribution so that the KL divergence between the posterior and the candidate is minimized.

In [4], Gal and Ghahramani showed that using Bernoulli variational distributions, the objective function of the problem of training a neural network with dropout is enabled is equivalent to performing stochastic variational inference on a deep Gaussian process with specific kernel functions (see the appendix of [4] for further details). In [6], they applied the same idea to convolutional neural networks by leveraging the fact that convolution operation can be written as a matrix product. Then, approximations to predictive mean and predictive variance can be computed efficiently using Monte Carlo integration.

Assume that we have an arbitrary convolutional neural network $\mathcal{D}$ and cross-entropy loss function is used during training while dropout is enabled. After the training, the resulting weights of the neural network becomes $\omega = \{\mathbf{W}_1, ..., \mathbf{W}_M\}$, where M is the number of parametric layers in the neural network $\mathcal{D}$.

Then, we can compute approximation to the predictive distribution as

$$p(\mathbf{y}^*|\mathbf{x}^*) \approx \frac{1}{L}\sum_{t=1}^{L}\mathcal{D}(\mathbf{x}^*, \tilde{\omega}_t), \tag{1}$$

where

- $\mathbf{x}^*$ is the test input,
- $\tilde{\omega}_t$ is the realizations of $\tilde{\omega}$,
- $\tilde{\omega} = \{\text{diag}(\mathbf{b}_1)\mathbf{W}_1, ..., \text{diag}(\mathbf{b}_M)\mathbf{W}_M\}$
- $\mathbf{b}_j$ is a Bernoulli distributed vector,i.e. each element of the vector $\mathbf{b}_j$ has a Bernoulli distribution with parameter p [4].

In other words, we can compute the approximation to predictive mean by passing the test input through the trained neural network while dropout is enabled and taking the average of the softmax outputs. Then, as our final prediction, we can pick the softmax output that has the highest probability value.

Computing model uncertainty is also straightforward. We need to calculate the entropy of the approximate predictive distribution calculated by Eqn.(1) [4]. High entropy value indicates that the model is not confident since it means that the probability values at the output of the neural network are spread over different classes.

3. PROPOSED

3.1. Training Procedure

Proposed method combines the idea of using convolutional neural networks for dominant instrument recognition [2, 3] and Bayesian convolutional neural networks [4, 6]. We use spectrogram, mel-scale spectrogram and MFCC as input features, where

- Frame size is 3 seconds,
- Sampling rate is 22050 Hz,
- Window length is 1024 samples,
- Hop size is 512 samples,
- Number of mel-filter banks is 128,
- Number of MFCC coefficients is 20.

The proposed Bayesian convolutional neural network model is illustrated in Figure 1. Each feature (spectrogram, mel-scale spectrogram and MFCC) is separately fed into the neural network and the result of each branch is concatenated to be used in the following fully connected layers. Note that we use dropout after each convolution operation and fully connected layers to comply with [4, 6]. Cross-entropy loss is used to train the proposed model with the stochastic optimization technique ADAM [7].

3.2. Test Procedure

In the test procedure, the incoming test example is divided into three-second excerpts. Feature extraction is performed on those excerpts using the same parameters given in Section 3.1. Features for each excerpt are used as an input to the proposed model in Figure 1. The approximation to the predictive distribution and entropy value are calculated according to Eqn.(1) for each excerpt.

After obtaining approximations to predictive distribution and entropy values for all excerpts, we need to leverage those to come up with a file-level output. One simple idea is to calculate the average of the outputs of softmax units over all excerpts and perform thresholding. This is the decision strategy used in [3]. The drawback of this decision mechanism is that it requires careful tuning of the threshold value. Furthermore, it weights all excerpts equally, i.e. it takes all excerpts into account without assessing the confidence of the prediction for the excerpt.

The proposed decision strategy utilizes the model uncertainty (i.e., entropy) to discard the excerpts that the model is not confident about compared to the other excerpts. To do that in an unsupervised manner, we propose to use the K-means algorithm to divide the entropy values of excerpts into two clusters. Then, we can discard the cluster that has the highest average entropy and use the other cluster to make file-level decision. After taking the averages of the softmax outputs of the excerpts that lie in the average-low-entropy cluster, we can again use K-means algorithm instead of thresholding to obtain the predicted labels. Hence, the proposed decision strategy uses the model uncertainty information to take only confident excerpts and eliminates the need for tuning the thresholding parameter.

4. EXPERIMENT

We tested the proposed framework on the IRMAS dataset [1] to demonstrate the ability of recognizing dominant instruments in polyphonic audio and to illustrate the benefit of using the proposed decision strategy.

4.1. IRMAS dataset

The IRMAS dataset is built for dominant instrument recognition problems in polyphonic music, and eleven different instruments are considered in the dataset: cello, clarinet, flute, acoustic guitar, electric guitar, organ, piano, saxophone, trumpet, violin, and voice.

Training set consists of 6705 audio files and each audio file is 3 seconds long. On the other hand, we have 2784 audio files in the test set, and length of each audio file ranges from 5 seconds to 20 seconds. In the training set, we have only one label for each three-second excerpt, while there are at least one label for each audio file in the test set.

4.2. Results on IRMAS dataset

We used PyTorch [5] to build, train and test the proposed model on the IRMAS dataset. Based on cross-validation, we chose

- learning rate to be 0.0001,
- batch size to be 16,
- weight decay parameter to be 0.001,
- dropout probability to be 0.2.

After obtaining the approximations of the predictive distribution and entropy values and applying the K-means based decision strategy introduced in Section 3, we obtained the predicted labels for each sound file in the test set. The performance metrics that we used are micro F1 score, micro precision, micro recall, macro F1 score, macro precision and macro recall. The resulting metrics for the proposed framework and other state-of-the-art methods are given as a bar graph in Figure 2.

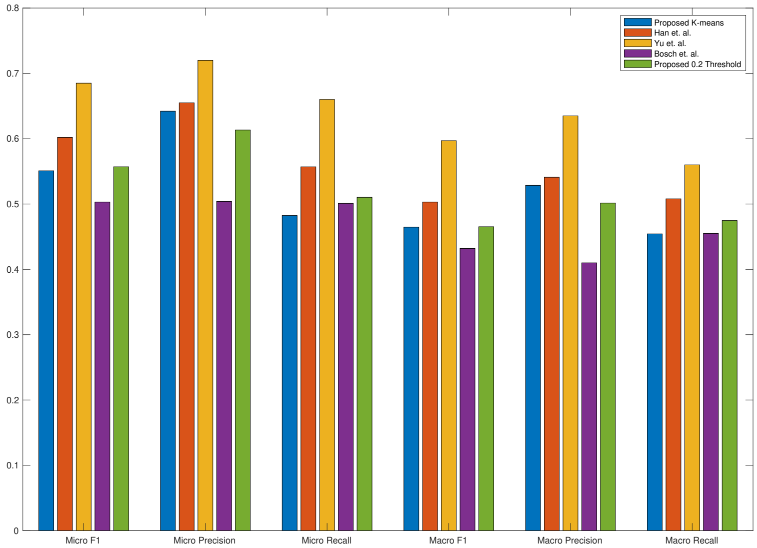

Fig. 2. Performance metrics obtained by [1], [3], [2], and the proposed method.

5. CONCLUSION

In this paper, we introduced a Bayesian convolutional neural network based dominant instrument recognition framework in polyphonic sound. The model uncertainty obtained by the proposed framework is used in decision step to discard the frames which the model is uncertain about. Because implementation of the proposed framework requires small modifications on the existing deep learning architectures, it can be implemented easily on several deep learning frameworks such as PyTorch. Finally, we showed that the proposed framework achieves the micro F1 score of 0.551, and macro F1 score of 0.465, which are close to state-of-the-art methods.

6. REFERENCES

[1] J.J. Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals," *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, pp. 559–564, 01 2012.

[2] D. Yu, H. Duan, J. Fang, and B. Zeng, "Predominant instrument recognition based on deep neural network with auxiliary classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 852–861, 2020.

[3] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, Jan 2017.

[4] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, 20–22 Jun 2016, vol. 48, pp. 1050–1059.

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

[6] Yarin Gal and Zoubin Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[7] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.