Melody Extraction

Aman Verma
180075
Department of Electrical Engineering
amanve@iitk.ac.in

Deshpande Prasad Jayant
17103270
Department of Civil Engineering
prasadj@iitk.ac.in

Raj Prakash Gohil
19104407
Department of Electrical Engineering
rgohil@iitk.ac.in

Abstract

This paper discusses application of Gaussian Process based DNNs for the Melody Extraction task. Along with the predictions, these models provide the uncertainty estimates for both approaches classification and regression. Such uncertainty estimates become a crucial input to further applications (e.g. active learning). The implementation for both approaches with different methodologies (including Bayesian CNNs) with MIR1K dataset is illustrated in this paper. The results show that there is still scope to improve the results by improving the model and hyperparameter-tuning.

1 Introduction

Music Information Retrieval (MIR) is an interesting yet challenging task in audio signal processing. It includes pitch extraction, beat tracking, tempo estimation, source separation etc. The end-user applications of MIR include searching-by-humming, searching-by-example, searching-by-notes etc. It may also include searching songs based on mood or genre. [1]

We trained CREPE, a data-driven algorithm developed in 2018, which is highly precise, maintaining over 90 percent raw pitch accuracy (RPA) even for a strict evaluation threshold of just 10 cents[2] as a part of our baseline model where we did not made use of library and pre-trained models. We have now worked on uncertainty estimation in Melody Extraction. Model uncertainty is of crucial importance in many regression and classification tasks such as medical diagnosis, autonomous vehicle steering or high frequency trading[3].

2 Problem definition

Melody extraction problem can be defined in 2 ways -
a) Identification of dominant frequency of the music (which in measured in Hz). This problem is mathematically modelled as a regression problem.
b) Identification of Swaras/Notes of the music as a Classification problem. In this case, the western notes in terms of C, G etc. are found out in terms of absolute frequency. Indian music Swaras in terms of Sa, Re, Ga etc. are found out w.r.t. a suitable reference frequency.

In many music forms (e.g. Indian classical music), the notes played do not belong to exact location of the notes, but there are little perturbations (Vibrato / Murki), or continuous shift of one note to another note (slide). In such cases, deterministic identification of notes is not the best way to

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

identify notes, but one can identify them probabilistic way (e.g., in case of CREPE). In this case, uncertainty or confidence interval associated with the prediction. Wherever the uncertainty is high, human inspection can be carried out to identify the melody. Hence, uncertainty analysis will become an input to active learning.

In the present study, we have used both approaches for melody extraction along with uncertainty analysis. MIR1K dataset is used for training-testing-validation.

3 Methodology

3.1 Gaussian Processess

Formally, a Gaussian Process is defined as a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions. A Gaussian process therefore defines a distribution over functions and is fully specified by a mean function $m(x)$ and a covariance function $k(x, x')$. Write $f \sim GP(m, k)$ meaning f is distributed as a GP with mean m and covariance k. Given a finite number n of locations $\boldsymbol{x} = [x_1, ..., x_n]$ it is possible to sample from the GP at those locations. First compute $\mu_i = m(x_i)$, $\sum_{i,j} = k(x_i, x_j)$ and then sample a vector from the distribution $\boldsymbol{f} \sim \boldsymbol{N}(\boldsymbol{\mu}, \sum)$.

3.2 Regression approach

Amersfoort et. al. in their recent (Feb 2021) paper 'Improving Deterministic Uncertainty Estimation in Deep Learning for Classification and Regression' [4] proposed a method that provides uncertainty estimates. This model that estimates uncertainty in a single forward pass and works on both classification and regression problems. For uncertainty estimation, a wide ResNet is used as a feature extractor which becomes input to Gaussian Process(GP). Variational approximation to the full GP is made. Cost function used is ELBO. Along with usual hyperparameters of neural networks, there is a step of spectral normalization. We have tried different kernals and number of inducing points for variational GP. For regression, the output is given as prediction (mean of the GP outputs) and uncertainty (variance of the GP outputs). We have tried the following inputs strategies as below :-

- Using **Time-Domain** input : The audio input is sampled at 16kHz frequency and a set of 1024 samples in given as input to the input layer.
- Using **STFT** input : STFT is carried out on the input audio signal, keeping NFFT as 63 and hop size as 63 so that the input gets 32 features.

Following two models we have tried as a feature extractor, both were provided by the author and we have modified them

- **FCResNet** : A lightweight fully connected ResNet
- **WideResNet** : A comparitively heavier convolutional ResNet which was originally provided for classification CIFAR10 dataset. We have modified it for regression problem. Here consecutive samples are appended together to make a Nx32x32 shape input, where N is number of samples.

3.2.1 Accuracy evaluation

For visual monitoring line plots, scatter plots of ground truth frequencies and predicted values are observed. Moreover, R^2 value, coefficient of correlation, RMSE was calculated.

3.3 Classification approach

For classification we experimented with two different methods:

- We obtained uncertainty bounds for convolutional neural networks using a Gaussian process on the last-layer on features learned by the CNN that was trained in a previous step on the same test training data. The multi-class classification is typically[5] approached by

assuming the following labelling rule for y_* given $\boldsymbol{x}_*$:

$$y_* = \underset{k=1,\ldots,C}{\operatorname{argmax}} f^k(\boldsymbol{x}_*)$$

where each $f^k(.)$ is a nonlinear latent function with a GP prior and C is the number of classes. The likelihood is again non-Gaussian meaning that approximation techniques have to be used to perform inference and to optimise the hyper-parameters. We used 6 CNN layered model to learn features and Matern52 Kernel along with White noise and for making training possible for GP, we used GPFlow, which implements various approximation algorithms for GPes. We used the Sparse Variational GP Classifier[6]. The model learns a set of m inducing points which are used instead of the training points in GP. Typically, $m < n$, which makes the task tractable as its complexity is $O(nm^2)$.

- Using Bayesian convolutional neural networks with variational inference, a variant of convolutional neural networks (CNNs), in which the intractable posterior probability distributions over weights are inferred by Bayes by backpropagation (BBB) [7]. For this we implemented BBB type of layer, i.e., this layer samples all the weights individually and then combines them with the inputs to compute a sample from the activations. We used Kullback-Leibler divergence function as kernel divergence function and added a variable weight to our layers. We defined the loss function of negative log-likelihood and trained the bayesian CNN for 25 epochs with batch size 64. We then quantify the uncertainty in prediction through sampling.

4 Results and discussion

4.1 Results for regression

4.1.1 WideResNet

The results for WideResNet without spectral normalization are shown in Figures 1, 2, and 5. Figure 1 shows that WideResNet is able to model the training data well (Coeff. of correlation is 0.98). Figure 2 shows corresponding training performance. Figure 3 suggests that model is predicting non-zero frequencies where the GT corresponds to 0, but from Figure 1 we can see that the uncertainty value is maximum at such locations. The model was given almost constant output with spectral normalization, we could not find the reason. These results were obtained using 2000 samples (approx 20 audio files.

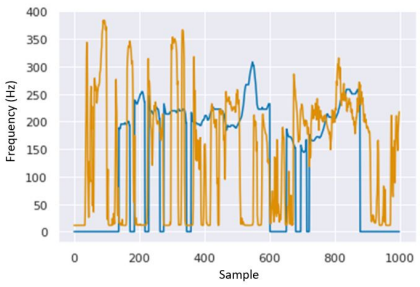

Figure 1: Predicted and GT frequencies for WideResNet

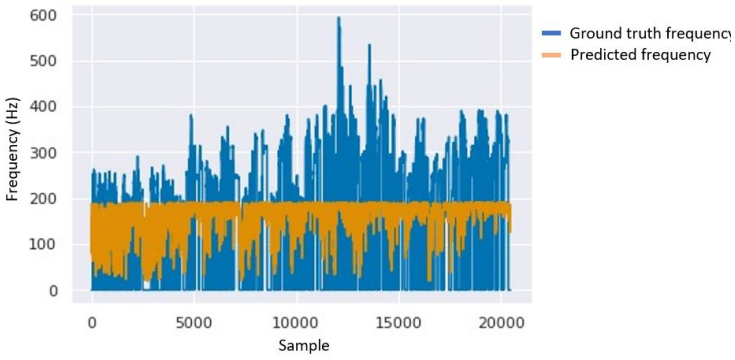

Figure 2: FCResNet predicted and GT frequencies

4.1.2 FCResNet

Though we tried different model configurations and hyperparameters but the model was not able to predict the frequencies correctly. As seen in Figure 4, the model is not able to predict the values above 200 Hz. This bound increases along with number of epochs, but on the other hand performance on validation data decreases i.e. overfitting.

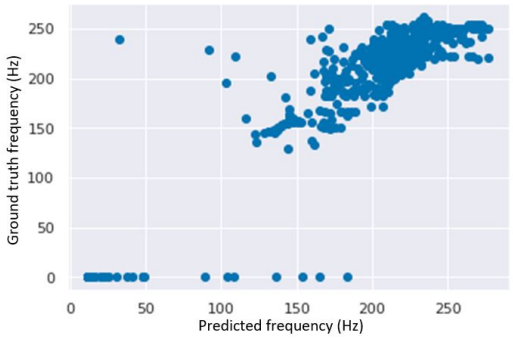

Figure 3: Scatterplot of predicted and GT values

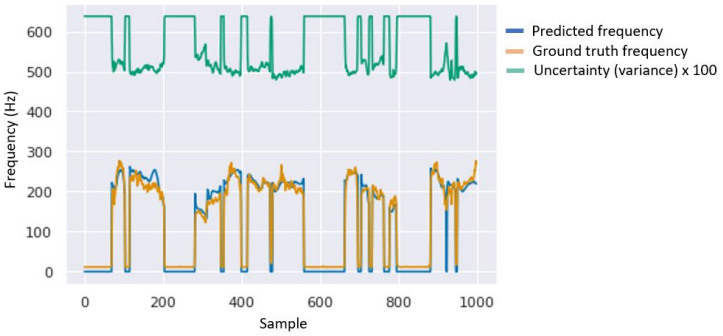

Figure 4: Predicted and GT frequencies along with uncertainty (x100)

4.2 Results for classification

We tried two different models to capture uncertainty in the classification problem. We got around 83% accuracy when we used the Gaussian process layer along with CNNs for 20 batches and we got around 77.01% accuracy when we were our baseline model CREPE[2] for the same train-test data for 20 epochs with batch size of 128. We also get a 79.5 max. Waw pitch accuracy (RPA) on our 10 test .wav files which is a massive jump from 51.78 from CREPE algorithm. We also generate a standard deviation along with our uncertainty estimate (Figure 5).

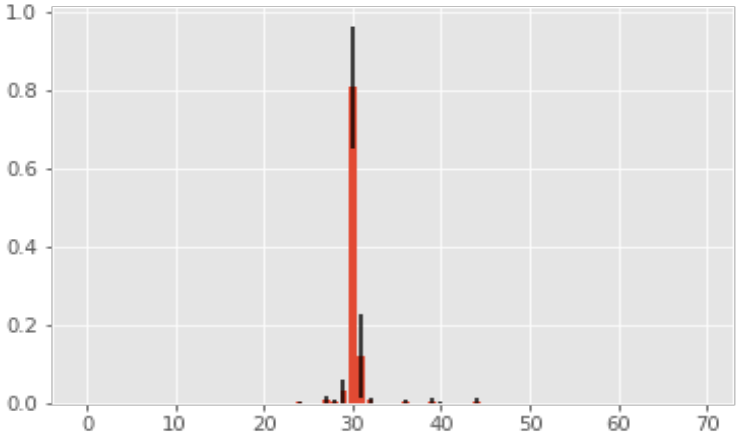

Figure 5: Predicted probabilities across all possible classes along with a standard deviation.

When using the Bayesian approach, we get nearly 67.51% validation accuracy along with a 70.29% train accuracy when trained on 25 epochs with 5 layered Bayesian Convolutional Neural Networks and batch size of 64. As seen in Fig.5 the model predicts probabilities for all possible classes for each example and if the probabilities are not above a certain threshold value then the model asks the user to annotate the data for him, so that it can learn that example too (see Fig.6)

5 Conclusion

The GP based deep learning methods have potential to estimation the prediction uncertainty [8, 4]. This estimation can be further used for various other AI applications, e.g. active learning. Moreover, the proposed models need to be improved in terms of accuracy. A deep understanding of the dataset characteristics and GPs may be helpful for tuning such models. Increasing the training data and number of epochs to train would surely help in improving the accuracy and in turn help the models learning better features from the data. Adding the annotation from the user on uncertain data to its training process would surely be a work which can be explored in future.

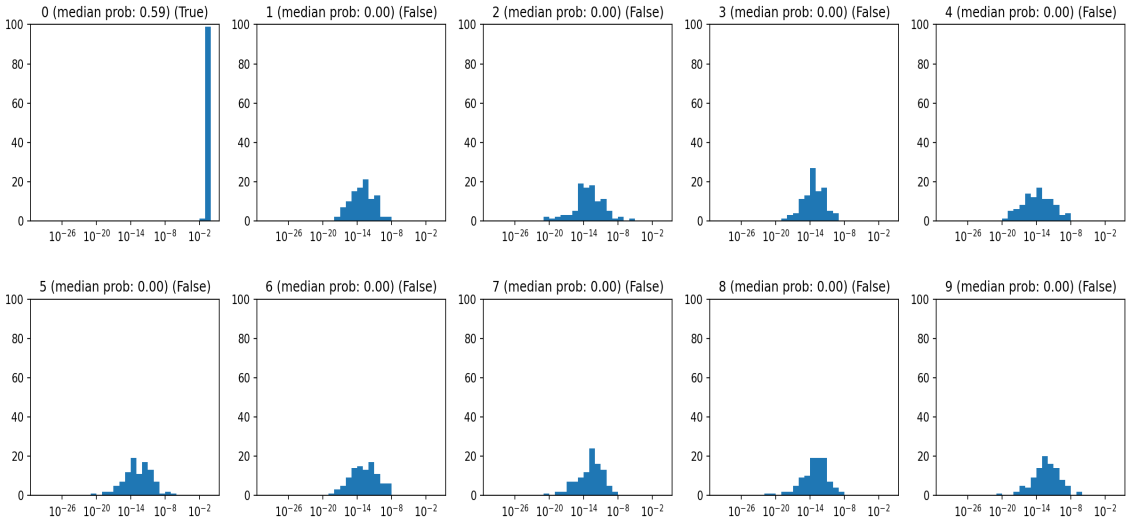

Figure 6: Plot of the histograms of predicted probabilities across all possible classes

References

[1] Roberto Raieli. 6 - the current status of mir systems. In Roberto Raieli, editor, *Multimedia Information Retrieval*, Chandos Information Professional Series, pages 175–193. Chandos Publishing, 2013.

[2] J. W. Kim, J. Salamon, P. Li, and J. P. Bello. Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165, 2018.

[3] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[4] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. Improving deterministic uncertainty estimation in deep learning for classification and regression, 2021.

[5] Carlos Villacampa-Calvo and Daniel Hernández-Lobato. Scalable multi-class gaussian process classification using expectation propagation, 2017.

[6] James Hensman, Alex Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification, 2014.

[7] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference, 2019.

[8] Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness, 2020.

6 Appendix

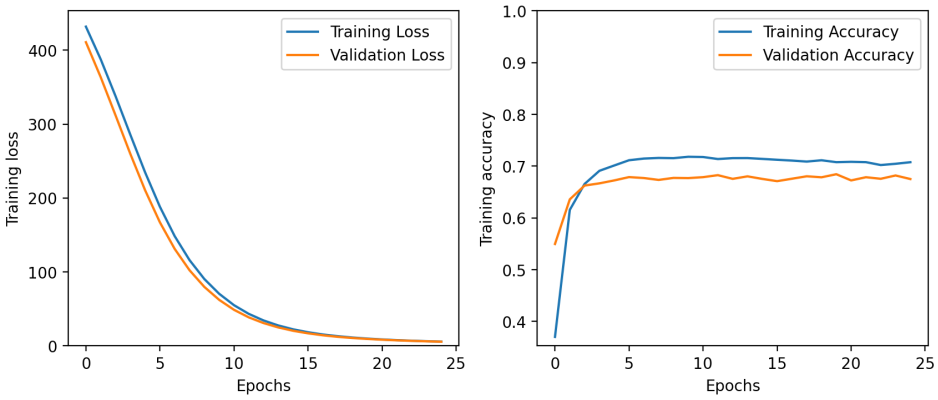

Figure 7: Training Process of Bayesian Convolutional Neural Networks approach

Algorithm 1 Algorithm for training DUE

1: **Definitions:**
- Residual NN $f_\theta : x \to \mathbb{R}^J$ with feature space dimensionality J and parameters θ.
- Approximate GP with parameters $\phi = \{l, s, \omega\}$, where l lengthscale and s output scale of k, ω GP variational parameters (including m inducing point locations Z)
- Learning rate η, loss function L

2: Using a random subset of p points of our training data, $X^{\text{init}} \subset X$, compute:
Initial inducing points: K-means on $f_\theta(X^{\text{init}})$ with $K = m$. Use found centroids as initial inducing point locations Z in GP.
Initial length scale:
$l = \frac{1}{\binom{p}{2}} \sum_{i=0}^{p} \sum_{j=i+1}^{p} |f(X_i^{\text{init}}) - f(X_j^{\text{init}})|_2$.

3: **for** minibatch $\boldsymbol{x}_b, \boldsymbol{y}_b \subset X, Y$ **do**
4: $\quad \theta' \leftarrow \text{spectral\_normalization}(\theta)$
5: $\quad \psi_b \leftarrow f_{\theta'}(\boldsymbol{x}_b)$
6: $\quad p(\boldsymbol{y}'_b|\boldsymbol{x}_b) \leftarrow \text{evaluate\_GP}_\phi(\psi_b)$
7: $\quad \mathcal{L} \leftarrow \text{ELBO}_\phi(p(\boldsymbol{y}'_b|\boldsymbol{x}_b), \boldsymbol{y}_b)$
8: $\quad \phi, \theta \leftarrow \phi, \theta + \eta * \nabla_{\phi,\theta} L$
9: **end for**

Figure 8: Pseudocode for training by [4]