Application of meta-learning methods in the recognition of drums and cymbals on the basis of short sound samples

Tomasz Krzywicki

Faculty of Mathematics and Computer Science
University of Warmia and Mazury in Olsztyn
Poland
email: tomasz.krzywicki@student.uwm.edu.pl

Abstract. This article presents proposal for application of Siamese neural network in the process of classifying the sound of short music instrument samples as percussion instrument or non-percussion instrument. In the learning process 15 sound samples representing each decision classes were used. The accuracy of solution was verified by 5-fold Cross Validation test. The proposed solution has achieved a satisfactory score.

1 Introduction

Classification of sound files on the basis of sound is difficult. Today's popular methods for classification process, such as deep neural networks, require large numbers of learning examples to achieve satisfactory scores. Meta-learning [8] and transfer-learning [9] methods may be useful for small sets of learning examples.

Motivation to create the proposed method was an attempt to use meta-learning methods in the process of classification of short music instrument samples as percussion instruments being a part of basic drum kit or other instruments. In order to simplify the process of creating of dataset for learning, the solution should work correctly with small number of samples. The solution is based on the Siamese neural network architecture, which classifies the sound as a percussion or non-percussion instrument on the basis of two parallel inputs as samples of sound of music instrument. The proposed solution has achieved a satisfactory score.

In sections 2 and 3 the Reader will be familiar with basic concepts of meta-learning approach and the most common sound processing method - MFCC. Section 4 provides information on architecture of siamese neural network, which has been used in the experiment. Section 5 contains details of preparations for the experiment in the form of the way to create dataset. In section 6 the Reader will be familiar with details of the classification model used in the experiment. Section 7.1 contains information about test of model which has been used in the experiment and accuracy obtained by the model and section 8 summaries the experiment carried out and provides informations on planned future works.

2 Meta learning

Learning and meta-learning methods are used to extract knowledge from the data. Let learning process of a learning machine L will be defined by a function $A(L)$: [4]

$$A(L) : K_L \times D \to M \tag{1}$$

where:

- K_L denotes the space of configuration parameters of given learning machine
- L, D denotes the space of data streams (typically decision system [1])
- M denotes the space of goal models

Meta learning is another or rather specific learning method. In the case of meta-learning the learning phase learn how to learn, to learn as well as possible. In other words, the target model of a meta-learning (output of meta-learning) is a configuration of learning model extracted by meta-learning algorithm. The configuration produced by meta-learning method should play the goal-role (like classifier or regressor) of meta-learning task. [4]

Meta-learning can be classified in few ways, right from finding the optimal sets of weights to learning the optimizer. Currently, the term of meta-learning covers the following categories: [8]

- Learning the metric space
- Learning the initializations
- Learning the optimizer

2.1 Learning the metric space

Metric-based meta-learning process is based on learning the appropriate metric space. For example, the process is used to learn similarity between two sentences. This approach is widely used in few-shot learning, where for learning is used dataset with small number of samples in each decision classes. [8] The method of learning metric space was also used in the proposed solution.

2.2 Learning the initializations

Learning the initializations process is based on trying to learn optimal initial parameter values. Classical learning (for example neural network) approach is based on initializing random parameters, calculation loss and minimizing the loss through a gradient descent in order to find optimal parameters. Meta-learning approach is based on finding optimal values of parameters with close to optimal values of parameters in order to learn model very fast. [8]

2.3 Learning the optimizer

This method is based on learning the optimizer. In case of few-shot learning, gradient descent fails when training set has too small number of objects, so optimizer should be learn itself. In other words, there are two networks: a base network that actually tries to learn and a meta network that optimizes the base network. [8]

3 Sound processing

The first step in any automation sound recognition is is to extract features, for example identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise. [7]

Mel Frequency Cepstral Coefficents (MFCC) are a feature widely used in automatic speech and speaker recognition. They were introduced by Davis and Mermelstein in the 1980's, and have been state-of-the-art ever since. Prior to the introduction of MFCCs, Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral Coefficients (LPCCs) and were the main feature type for automatic speech recognition (ASR), especially with HMM (Hidden Markov Models) classifiers. The procedure of converting the sound spectrum into numerical vectors by MFCC method is follows: [7]

- Frame the signal into short frames
- For each frame calculate the periodogram estimate [5] of the power spectrum
- Apply the mel filterbank to the power spectra, sum the energy in each filter
- Take the logarithm of all filterbank energies
- Take the DCT of the log filterbank energies
- Keep DCT coefficients 2-13, discard the rest

4 Architecture of siamese networks

A siemese network is a special type of neural network most popularly used one-shot learning algorithms, so a siamese network is predominantly used in applications where is small number of learning objects. Siamese networks basically consist of two symmetrical neural networks both sharing the same weights and architecture, both joined together at the end using some energy function E. The objective of siamese network is to learn metric space of similarity of two objects, for example two sound samples. [8].

In the Figure 1 you can see, that input of siamese network receives two samples (sample_a, sample_b) in the form of tensors. The samples are then processed by each of twin networks, and their output is forwarded to an energy function which calculates the similarity (metric distance) of the two samples.

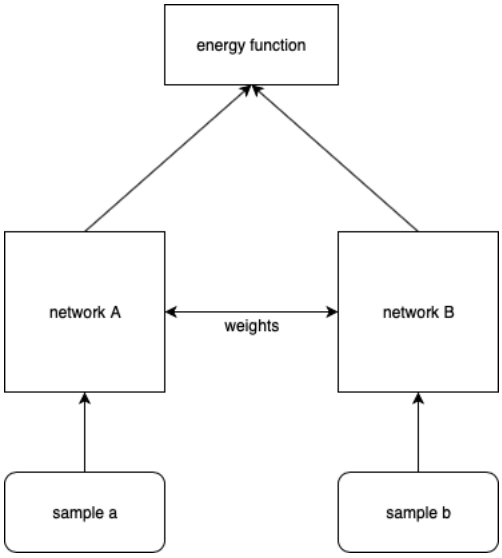

Fig. 1. Siamese neural network evaluating similarity of two samples

Siamese neural networks are commonly used not only for sound recognition. They are also used for face recognition, signature verification, object tracking, similar question retrieval and more. [8]

4.1 Detailed architecture of siamese neural networks

The detailed architecture of siamese neural network is shown in figure 2.

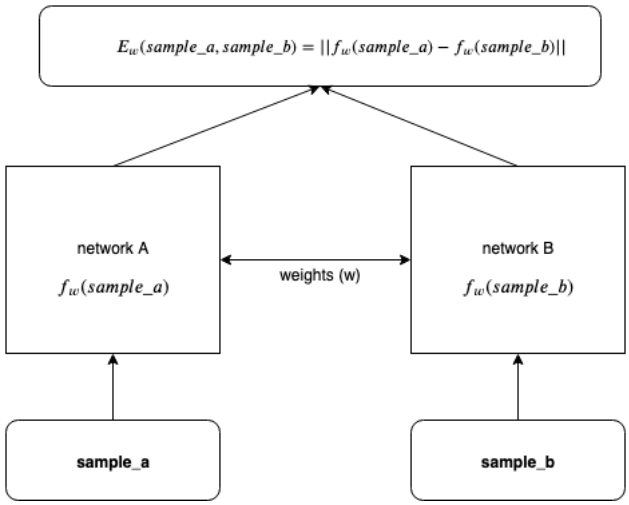

Fig. 2. Detailed architecture of siamese network

There are two inputs: $sample\_a$ and $sample\_b$. Inputs $sample\_a$ and $sample\_b$ are forwarded to networks: $networkA$ and $networkB$ respectively. Network outputs are defined by the formula $f_w(sample_x)$ where $sample_x$ denotes input of appropriate neural network. Then outputs of networks are forwarded to energy function E, which is represented by formula: [8]

$$E_w = (sample\_a, sample\_b) = ||f_w(sample\_a) - f_w(sample\_b)|| \qquad (2)$$

5 Dataset preparing

6 melodic instruments and 6 percussion instruments were used in the experiment. Group of melodic instruments consists of brass instruments, sound synthesizers, flute, guitar, organs, piano. Group of percussion instruments consist of basic version of a drum set: crash cymbal, hi-hat cymbal, kick drum, ride cymbal, snare drum, tom drums. In each group of instruments there are both acoustic and electronic samples of sounds. For each instrument there are 15 sound samples.

The aim of this paper is to use siamese naural network to evaluate the similarity of sound of group of instruments in the detection of percussion instruments. Therefore the data collection will be further processing in the form of decision system with the following attributes: **(sample_a, sample_b, similarity)**, where:

– sample_a denotes tensor of **a** sound samples after processing by MFCC method

– sample_b denotes tensor of **b** sound samples after processing by MFCC method
– decision denotes decision attribute which classifies similarity of two sound samples

Detailed overview of the sample collection for further processing was presented with table 1:

Instrument	Group of instruments	Number of samples
brass instruments	melodic	15
crash cymbal	percussion	15
sound synthesizer	melodic	15
flute	melodic	15
guitar	melodic	15
hi-hat cymbal	percussion	15
kick drum	percussion	15
organs	melodic	15
piano	melodic	15
ride cymbal	percussion	15
snare drum	percussion	15
tom drums	percussion	15

Table 1. Detailed overview of the instrument sample collection

As similar sound samples may be consider two percussion instruments, for example ride cymbal and snare drum. As dissimilar sound samples may be consider percussion instrument with non-percussion instrument, for example kick drum with piano. The procedure for the selection of similar and dissimilar sound samples is as follows:

1. For each instrument of percussion instrument group
   (a) If iteration is even, draw another melodic instrument and its sample. Add both tensors of samples to the decision system with label 0, which means dissimilar sounds.
   (b) If iteration is odd, draw another percussion instrument and its sample. Add both tensors of samples to the decision system with label 1, which means similar sounds.

Exemplary decision system based on this procedure were presented in table 2:

In order to maintain compatibility between each sound sample of music instrument, each tensor of sound sample has been reduced to shape (20, 400). In depending on sampling of sound sample may mean a sight difference of the sound processed. However, it does not affect quality of classification.

sample_a	sample_b	similarity
[[-299.0982, ..., -54.3453]] (kick drum)	[[312.765, ..., 43.8856]] (ride cymbal)	1
[[19.0841, ..., 88.5388]] (hi-hat cymbal)	[[99.0098, ..., 64.9856]] (piano)	0
[[24.0991, ..., 75.5542]] (crash cymbal)	[[246.0558, ..., 98.5436]] (snare drum)	1
[[-132.0841, ..., 45.6430]] (tom drum)	[[199.7355, ..., 99.1432]] (brass instruments)	0

Table 2. Exemplary decision system of similarity and dissimilarity of samples of sounds of music instruments

6 Classification model

The siamese neural network model was used to classify the similarity of two sounds of music instruments. A single neural network (cloned for the construction of the siamese network) of neural network was constructed as follows:

- input: (20, 400) shaped tensor containing vectorized spectrum of sound of music instrument
- hidden layers:
  - Flatten layer
  - 128 size Dense layer with ReLU activation function
  - Dropout layer with a value of factor 0.1
  - 128 size Dense layer with ReLU activation function
  - Dropout layer with a value of factor 0.1
  - 128 size Dense layer with ReLU activation function
  - Dropout layer with a value of factor 0.1
  - 64 size Dense layer with ReLU activation function
  - Dropout layer with a value of factor 0.1
- output: 64 size Dense layer with ReLU activation function

The Euclidean distance defined as follows was used as energy function in the siamese network: [6]

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(a_i(x) - a_i(y))^2} \tag{3}$$

where:

- $d(x,y)$ denotes Euclidean distance in $n-$dimensional space of real numbers
- $a_i(x)$ denotes the value of i coordinate that x objects point in $n-$dimensional space of real numbers
- $a_i(y)$ denotes the value of i coordinate that y objects point in $n-$dimensional space of real numbers

The full diagram of the siamese neural network used in the experiment, was presented in figure 3.

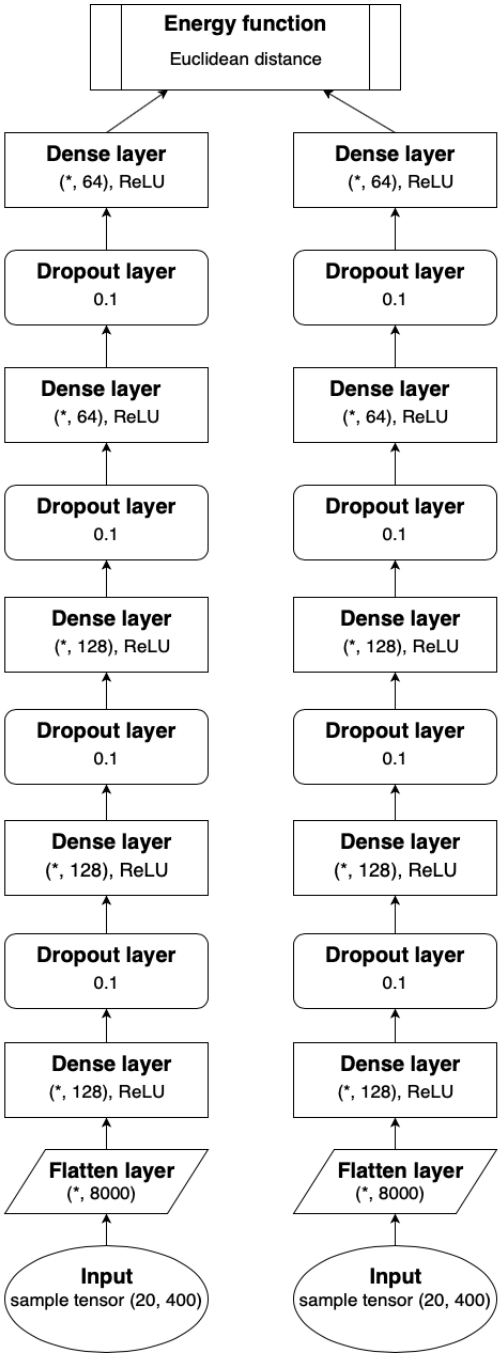

Fig. 3. Full diagram of the siamese network used in the experiment

6.1 Contrastive Loss

As a loss function during model training, the function of contrastive loss has been used. Contrastive Loss function is based on learning the parameters of a parametrized function in such a way that neighbors are pulled together and non-neighbors are pushed apart. Prior knowledge can be used to identify the neighbors for each training data point [3]. The function of loss of contrastive loss is defined by the following formula [8]:

$$L = Y(E)^2 + (1 - Y)max(\text{margin} - E, 0)^2 \tag{4}$$

where:

- L denotes Contrastive Loss function
- Y denotes expected model predictions
- E denotes energy function
- $margin$ denotes the loss function parameter, which means threshold for classifying distance calculated by the energy function as similarity

7 Model accuracy test

The siamese neural network model has been trained over 21 training epochs with 75% of the data in the training subset and 25% of the data in the validation subset. In order to verify the accuracy of the classification on small dataset, a 5-fold Cross Validation test has been performed.

7.1 Cross Validation

k-fold cross validation is based on dividing the data set into k separated subsets, and then on repeated operations of model training on k-1 subsets and checking the accuracy on the one test subset [2]. The test subsets have to be unique. The average accuracy of all k tests and its standard deviation are result of test [1].

7.2 The accuracy obtained by the model

After applying 5-fold cross validation test, the model obtained the results shown in table 3:

Subset	Accuracy	Standard deviation
training	0.902655	0.044054
test	0.85054	0.084436

Table 3. Scores obtained by the model

The accuracy obtained by the model may indicate over fitting of the model, which in this case (small number of samples in data set) may be acceptable.

8 Conclusions

The key aim of this article was to performance a proposal of recognizing a short sound samples as percussion instruments or melodic instruments based on the siamese neural network.

At the beginning of the article the Reader has been familiar with basic concepts of meta-learning approach and sound processing. Then architecture of siamese neural networks has been presented, which has been later used in the experiment. In the next step have been shown details of preparations for the experiment in the form of the way to create dataset and explanation of the classification model. The suggested solution has obtained a satisfactory effectiveness confirmed by 5-fold cross validation test: 85% of accuracy.

The proposed solution is the start of work on the method of percussion instruments recognition in full sound tracks. The method will aim at creation of musical notation for percussion instruments for any sound (if they will be there). In the future is planned to create sound classification models on the basis of other meta-learning methods and comparing their effectiveness with each other. Based on the effectiveness of these models further work will be carried out to create the planned objective.

References

1. Artiemjew, P.: Wybrane paradygmaty sztucznej inteligencji. PJATK Publishing House, (2013)
2. Chollet, F.: Deep Learning. Praca z jzykiem Python i bibliotek Keras. Helion Publishing House, (2019)
3. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality Reduction by Learning an Invariant Mapping, http://yann.lecun.com/exdb/publis/pdf/hadsell-chopra-lecun-06.pdf
4. Jankowski, N., Duch, W., Grabczewski, K.: Meta-Learning in Computational Intelligence. Springer Verlag, (2011)
5. Knopov, P. ; Bila, G.: Periodogram estimates in nonlinear regression models with long-range dependent noise. Cybernetics and Systems Analysis, 2013, Vol.49(4), pp.624-631
6. Krzywicki, T.: Weather and a Part of Day Recognition in the Photos Using a kNN Methodology. Technical Sciences, 21(4) 2018, p. 291-302
7. Mel Frequency Cepstral Coefficient (MFCC) tutorial: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/
8. Ravichandiran, S.: Hands-On Meta Learning with Python. Packt Publishing, (2018)
9. Sarkar, D., Bali, R., Et al: Hands-On Transfer Learning with Python. Packt Publishing, (2018)