NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

PATTERN RECOGNITION

Preparation for Lab3:
Genre classification and Emotion recognition in Music

Explain your work comprehensively and adequately. Code without comments will not be graded. Collaboration is allowed within groups of 2 people as long as they are studying in the same study program (either undergraduate groups or postgraduate groups). Each team of 2 submits a joint report representing only the individual work of its members. If you use any source other than the books and course materials, you must mention it. The report and the code for the assignment will be submitted online in the course site. It is forbidden to post the solutions of the laboratory exercises on github, or any other websites.

On the following page you can find helper code related to the course labs. Submission of the completed scripts is necessary for grading. On this page you can also submit questions to the course TAs in the form of issues. Questions regarding the lab that will be made via email will not receive an answer.

DESCRIPTION

The purpose of this lab is the genre classification and emotion recognition from the spectrograms of music pieces. Two different datasets are being given, the Free Music Archive (FMA) genre with 3834 samples separated to 20 classes (music genres) and the database (dataset) multitask music with 1497 samples with labels regarding the emotional dimensions: valence, energy and danceability. The samples are essentially spectrograms, created by 30-second clips of the music pieces.

We are going to analyze spectrograms using deep NN architectures and specifically Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

The lab exercise consists of 5 parts:
1) Data analysis and familiarization with the spectrograms.
2) Building of classifiers for music genre detection on FMA dataset.
3) Building of regression models for valence, energy and danceability predictions on the Multitask database.
4) Utilization of advanced methods, transfer and multitask learning, for performance improvement.
5) (Optional) Visualization of music pieces representations using dimensionality reduction algorithms.

The data are available here. You can use kaggle kernels in order to have access to free GPUs and you should make use of this kernel to develop your solution (select “Copy & Edit” to create your own “clone”).

PYTHON LIBRARIES

- librosa, numpy, pytorch, scikit-learn

USEFUL LINKS

[0] https://www.kaggle.com/code/geoparslp/lab3-data-loading-tutorial
[1] https://www.kaggle.com/datasets/geoparslp/patreco3-multitask-affective-music
[2] https://www.kaggle.com/c/multitask-affective-music-lab-2022/kernels
[3] https://www.coursera.org/lecture/machine-learning-projects/transfer-learning-WNPap
[4] https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a
[5] http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf
[6] http://ruder.io/multi-task/
[7] https://arxiv.org/pdf/1706.05137.pdf
[8] https://github.com/slp-ntua/patrec-labs/tree/main/lab3
[9] https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1
[10] https://becominghuman.ai/how-to-evaluate-the-machine-learning-models-part-3-ff0dd3b76f9
[11] https://towardsdatascience.com/metrics-for-imbalanced-classification-41c71549bbb5
[12] https://twitter.com/karpathy/status/1013244313327681536
[13] http://karpathy.github.io/2019/04/25/recipe/
[14]https://www.reddit.com/r/MachineLearning/comments/5pidk2/d_is_overfitting_on_a_very_small_data_set_a/
[15] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[16] https://colah.github.io/posts/2014-07-Understanding-Convolutions/
[17] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
[18] https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/
[19] https://cs.stanford.edu/people/karpathy/convnetjs/
[20]https://towardsdatascience.com/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29

EXECUTION

During the lab preparation we will work on identifying the genre of a music piece given its spectrogram. As we have already seen, at the 2nd lab, the spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time, where the result image shows the energy of the signal at different frequency levels and time windows.

Step 0: Familiarization with Kaggle kernels

Visit Kaggle. Make use of this kernel as a starting point and select "Copy & Edit" to create your own "clone".
Run the commands:

```
import os
os.listdir("../input/patreco3-multitask-affective-music/data/")
```

to browse the content of the subfolders. Try to activate and deactivate the GPU and Save your changes.

Step 1: Familiarization with spectrograms at the Mel scale

The data that you will use for the preparation, is a subset of the FMA dataset. FMA is a database that includes free music clips with labels with regards to genre.
We have extracted the spectrograms along with their labels under the folder:
`../input/patreco3-multitask-affective-music/data/fma_genre_spectrogram`.

The file `fma_genre_spectrograms/train_labels.txt`
contains lines of the form: **“`spec_file label`”**.

a) Choose two random lines with different labels. The respective files are located under the folder `fma_genre_spectrograms/train`.

b) Read the files and get the mel-scaled spectrogram following the implementation in [0].

c) Visualize the mel-spectrograms, belonging to different classes, using the function “librosa.display.specshow”. Comment on what information they provide and what are the differences of two samples that belong to different categories. (Note: frequency on the vertical axis, time on the horizontal).

Step 2: Synchronization of the spectrograms to the music beat (beat-synced spectrograms)

a) Print the dimensions of the spectrograms you used in Step 1.

- How many time steps do they include?
- Is the training of an LSTM efficient for this data?
- Why?

b) A method to reduce the number of the time steps is to synchronize the spectrograms with the rhythm. To achieve this, we retrieve the median (frequencies) on the time steps between two music beats. The respective files can be found at the following folder: `../input/patreco3-multitask-affective-music/data/fma_genre_spectrogram_beat`. Repeat the Steps of the “Step 1” for the beat-synced spectrograms and comment on the differences you notice.

Step 3: Familiarization with the chromagrams

Chromagrams depict the energy of the signal for the frequency levels that correspond to the twelve different music notes {C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B} and they can be used as a tool for analyzing music with regards to the harmonic and melodic characteristics. They are, also, quite robust on recognizing timbre and instrumentation changes (one can say that chromagram is a spectrogram modulo the octave).

Repeat the previous Steps, 1 and 2, with the chromagrams of the respective pieces.

Step 4: Data loading and analysis

Use the helper code on [0] and [8]

a) At the helper code, you can find an implementation of a PyTorch Dataset that reads the data and returns the samples. Study the code and the values that it returns and comment on the operations that are executed.

b) At the provided implementation, we merge similar classes and we remove the ones that are represented by very few samples.

c) Create two histograms that will show the number of the samples that belong to each class, one before the aforementioned process of 4b and one after it.

Step 5: Music genre recognition with LSTM.

By utilizing the code you developed during the 2nd lab exercise:

a) Adjust the code of the LSTM model so as to accept as input the spectrograms of the Pytorch dataset from Step 4.

b) In order to accelerate the development and debugging process of your models, add a boolean parameter "overfit_batch" to the train() method. When "overfit_batch" is False, the model training is normal. When it's True, it will be trained on a small set of 3-4 batches resulting in overfitting.

- Overfitting of the model in one batch: A good practice during the development of NN models, is to ensure that the network can be trained (the gradients are backpropagated etc.). A quick method to check this, is to randomly select a small subset of the data (e.g. one batch) and train the network on this for several epochs. We expect to see the training loss taking values around 0 and the network overfitting (see also [12], [13], [14]).

c) train an LSTM [15] network that gets as input the spectrograms of the training set and predicts the different classes (music genres) of the dataset.

d) train an LSTM network that gets as input the beat-synced spectrograms of the training set and predicts the different classes (music genres) of the dataset.

e) train an LSTM network that gets as input the chromagrams of the training set and predicts the different classes (music genres) of the dataset.

f) (extra credit) train an LSTM network that gets as input the concatenated chromagrams and spectrograms of the training set and predicts the different classes (music genres) of the dataset.

Note: Use the validation set for training your models.

Note: Activate GPU for training your models.

Note: Use Adam optimizer.

Note: Use the Pytorch class Subset, in order to be able to train your models on smaller subsets and, thus, accelerate the debugging/tuning processes. Usage example.

Step 6: Evaluation of the models

Report the results of the models you trained in Step 5 on the following test sets:

- `fma_genre_spectrograms_beat/test_labels.txt`
- `fma_genre_spectrograms /test_labels.txt`

Specifically:

a) calculate the accuracy

b) calculate precision, recall and F1-score for each class

c) calculate the macro-averaged precision, recall and F1-score over all classes

d) calculate the micro-averaged precision, recall and F1-score over all classes

Explain the meaning of these metrics and comment on which one you would pick for the evaluation of a classifier in this task. Specifically, focus on the questions:

- What information does accuracy / precision / recall / f1 score give?
- What information does micro / macro averaged precision / recall / f1 score give?
- When do we expect to see a significant difference between accuracy and f1 score and what does this mean?
- When do we expect to see a significant difference between micro and macro f1 score and what does this mean?
- Are there any problems/tasks where we care more about precision than recall or the opposite? Is a good accuracy/f1 value enough in these occasions to select a model?

Note: Use the function sklearn.metrics.classification_report

Note: See also [9], [10], [11]

---------- **END OF LAB PREPARATION**----------

Step 7: 2D CNN

Another approach for building a model for audio signals processing is to interpret the spectrogram as an image and, thus, utilize Convolutional Neural Networks (CNN).

a) At link [19], you can train CNNs and see their internal operations by visualizing the activations for each layer of the network easily. Train a network on MNIST and observe the activations of each layer. Comment on the operations that take place, the patterns that seem to be learned by the model and include the respective screenshots into your report.

b) Develop a 2D CNN with 4 layers that will process the spectrogram as a 1-channel image. Train it using the training & validation sets and report its performance in the test set. Each layer consists of the following operations (with the given order):

1) 2D convolution
2) Batch normalization
3) ReLU activation
4) Max pooling

c) Explain the function and the role of convolutions, batch normalization, ReLU and Max pooling. Εξηγήστε τη λειτουργία και τον ρόλο των convolutions, batch normalization, ReLU και Max pooling. We refer to the references [16], [17], [18]

d) Apply the "overfit_batch" process of Step-5b to make sure that the network can be trained.

e) Utilize this architecture for genre recognition using spectrograms and compare with the model of Step-5a.

Note: Run the model to a different kernel to avoid memory issues.

Note: Take into consideration the notes of Step-5.

Note: Do not waste much time in the network's hyperparameters tuning. You may just check some online CNN implementations and select some "rational" values (e.g., kernel size ~ 3 or 5 etc.). If your network does not operate properly, it is more probable to have a bug in the code rather than having mistuned the hyperparameters, especially when their values are close to the default ones.

Step 8: Emotion - Behavior estimation with regression

In this step you will use the multitask dataset:
("../input/patreco3-multitask-affective-music/data/multitask_dataset/train_labels.txt").

Here, the spectrograms are available along with annotations in 3 axes with regards to the "emotion" of the song. The annotations are real numbers between 0 and 1:

- *Valence* (how positive or negative the emotion is), where negative is close to 0, positive close to 1.
- *Energy* (how powerful is the emotion), where weak is close to 0, powerful close to 1.
- *Danceability* (how danceable the song is), where not-danceable is close to 0, danceable is close to 1.

a) Adjust the best model from Step-5 and the best one from Step-7 for regression by changing the loss function.

b) Train the models to estimate valence.

c) Train the models to estimate energy.

d) Train the models to estimate danceability.

e) The final metric is the average Spearman correlation between the ground truth values and the predicted ones for all three axes.

Note: Attention. In this dataset, the test set annotations are not available and, thus, the estimation regarding the generalization ability of the model should be performed using a subset of the available data.

Step 9: Transfer Learning

An approach we follow so as to improve the NNs performance when there is not enough available data, is to transfer learning from a model, trained on a bigger dataset, to another. For this reason:

a) See the links [3], [4], [5]. Describe in short the basic conclusions of [5].

b) Choose a model from Steps 5, 7. Explain why you chose this model.

c) Train this model to the *fma_genre_spectrograms* dataset and save the weights of the network at the epoch that it had the best performance (checkpoint). Explain which metric you selected and in which set in order to judge which model state had the best performance. Why?

d) Initialize a model with these weights and use it for the task of Step 8; train it for few epochs (fine tuning) in one of the emotional axes (e.g., valence).

e) Compare the results with the ones you got at Step 8.

Step 10: Multitask Learning

At Step-8, we trained a separate model for each emotional dimension. A method so as to train more efficient models, when multiple labels are available, is to use multitask learning.

a) See the links [3], [6], [7] and describe in short the basic conclusions of [7].

b) Develop a loss function that will take as input the logits (outputs of the model) and the targets (ground truth values) for valence, arousal and danceability and will return the sum of the three individual losses (one for each task). You may use weights in order to bring the individual losses into the same order of magnitude.

c) Train a model to multitask dataset utilizing the above loss function.

d) Compare the results with the one you got at Step-8.

Step 11 (Optional): Visualization of latent representations

Select the best trained models from Step 5 (LSTM) and 7 (CNN). Use the test set of the *fma_genre_spectrograms* dataset and retrieve the final representation (a vector for each song) exactly before the output layer (of the Classifier).

a) For each model, create a new dataset that will consist of the latent representation (a vector of dimension "feature_size") for each song of the test set (as it is retrieved from the final layer of the NN before the output_layer) and its respective label (genre).

b) See the link [20]. Apply dimensionality reduction using PCA and visualize the latent representations space of each model. How much are the several classes separated in this space? Compare the spaces that are formed by different models.

c) Repeat the above using the model from the Step-5 that performed worst in the genre classification task. Do you observe any qualitative difference between this and the previous ones?

d) Try more dimensionality reduction algorithms (t-SNE etc.). Which one do you think that works better for your data?

DELIVERABLES

1. Written report in .pdf format. Write a full report using Latex / Word / Google Docs etc. We do not accept exported notebooks (.ipynb / .html files).

2. Python code for your implementation in .py files (NOT .ipynb / .html). If you use notebooks you must export to .py before submission.

For your submission create a .zip file (NOT rar / tar.* / .7z), containing the deliverables. The zip file name should follow the convention:
("patrec_FirstPartnerRegistrationNumber_SecondPartnerRegistrationNumber.zip").
If you do not have a Registration Number, use FirstNameLastName.

For group submissions, only one of two partners must submit and both partners' details must be listed on the first page of the report as well as the zip name as above.
If you do not have an institutional e-mail available / cannot complete the submission, you can contact the course.