Deep Learning for Audio and Music Assignment Report

Pedro Pereira Sarmento

Queen Mary University of London

April 24, 2020

1 Overview of the task

In this assignment we were requested to implement two distinct deep learning methods in order to infer two aspects from a chosen audio file. The proposed approach consisted of a first network that classifies an audio file into *music* or *non-music*. Given that the file is classified as *music*, it is then passed into the second network, where the task of music genre classification is performed. Attached to this report are two *.ipynb* files, one for each network. At the beginning of each file there is a brief description of the task, the resources and references used, as well as information about the datasets and the chosen audio file. It was implemented using *Python 3.6.6* and *PyTorch*. It is expected that this project can be made publicly available.

2 Chosen Audio File

A track from the album *Conduct*, by October Horse, a progressive metal band from Porto, Portugal was chosen[1]. The track, *Waving*, consists of two distinct sections inspired by different musical genres. This fact, combined with the fairly low budget quality of the recording, rendered the track as an interesting, difficult, test for both networks. The author of this report is a member of the band.

3 Network One

As stated before, the first network performs a binary classification task into *music* or *non-music*. To this extent it leverages pre-trained features, embeddings for audio classification models from VGGish [2].

3.1 Implementation

Regarding datasets, we used four second chunks from 1,000 files of GTZAN [5], a benchmark dataset for genre classification consisting of 100 files from ten

[1] https://octoberhorse.bandcamp.com/track/waving

different musical genres, each with a duration of 30 seconds. In our approach, the whole dataset was labelled as *music*. This was combined with four second chunks from 1,000 files of URBANSOUND8K [4], labelled as *non-music*. This dataset includes 8,000 files distributed over 10 categories, one of them labeled as *street music*. Using *vggish_input.waveform_to_examples*, a feature extraction function from the *torchvggish* package, the chunks were convereted into log mel specotgrams, each second with a dimensionality of 96 time frames x 64 mel bins. Next, we split the dataset into 70% for training and 30% for testing.

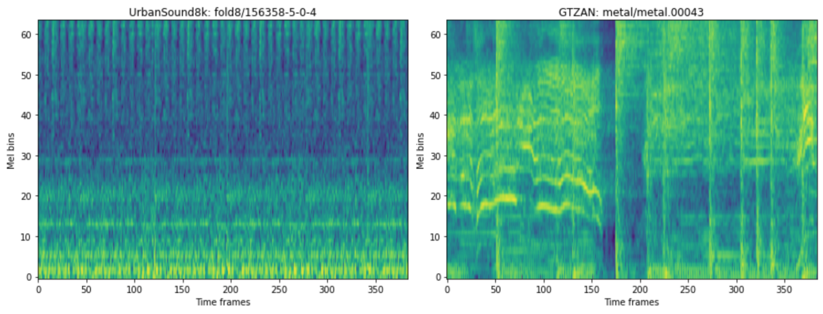


Fig. 1: Log mel spectograms generated of (left) an URBANSOUND8K example and (right) a GTZAN example. Time axis consists of four seconds (384 time frames).

In terms of network architecture, it consists of a logistic regression layer applied to the embeddings generated by the VGGish. Initially, the VGGish was loaded and the log mel spectograms were fed into it, generating 128-dimensional embeddings for each chunk.

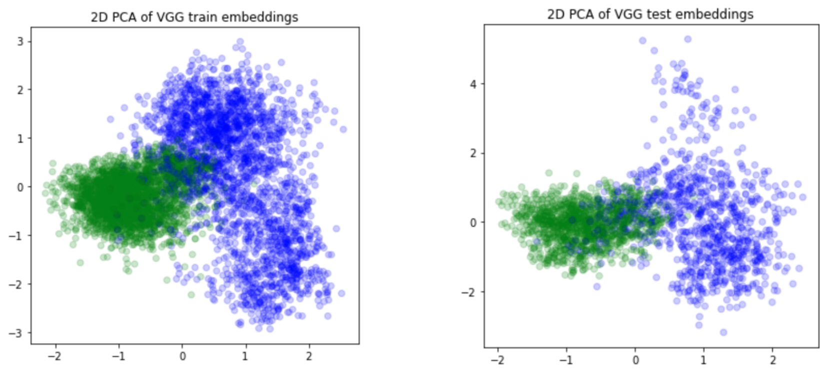


Fig. 2: Plot of 2D PCA representation of (left) VGG embeddings for the training and (right) test set. Blue points are labelled as *non-music*, green points as *music*.

Applying a principal components analysis (PCA) over the embeddings, compressing it from 128 dimensions to 2 dimensions, permitted pre-inspection of the data. Despite the fact that PCA is an unsupervised approach, thus offering no guarantee that this is the best configuration to separate both classes, it seems reasonable to assume that they are in fact separable. Furthermore, a similar intuition derives from the log mel spectograms in Figure 1. Finally, using *LogisticRegression* from *sklearn.linear_model* we performed logistic regression, classifying the principal components of the VGG embeddings as *music* and *non-music*.

3.2 Evaluation

After applying logistic regression to the VGG embeddings we plot the results, observable in Figure 3, for both training and test set.

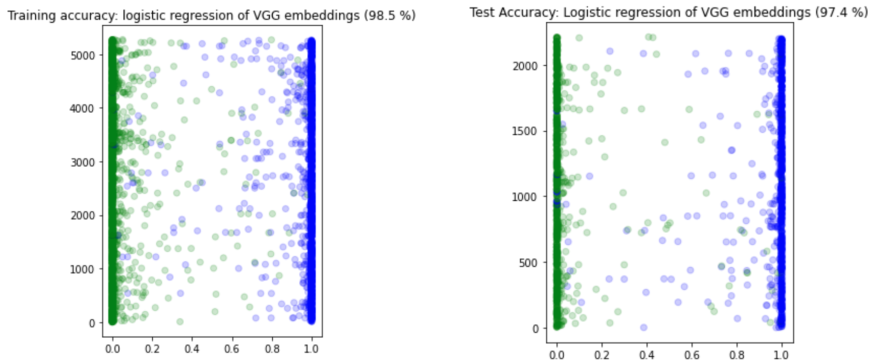


Fig. 3: Plot of predictions for (left) training set and (right) test set.

Regarding the test set, the networks yields an accurcay of 97.4%. This results were saved to be later passed through the second network. A further inspection of the classified files interestingly showed that most of the few misclassifications that it contained (i.e. files from the URBANSOUND8K that were classified as music), were labelled as *street music*. This was first interpreted as the algorithm being robust against the distortions and noises present in the *street music* examples.

Furthermore, we applied the same procedure to our chosen audio file. Out of 228 melspectogram chunks, the algorithm predicted 169 as music and 59 as non-music (72.4% probability of being music, when considering an average of probabilities for each chunk). Although this represents a satisfactory result, we might expected higher probability values given the results with the test set. A probable cause for this might be due to overfitting given the small size of the dataset used (although this should be compensated when using the model from VGGish). The audio quality of *Waving* was also thought as a potential issue, although it should not be a problem given that *street music* examples, much noisier, were correctly predicted. This too pointed towards an inability to generalize.

4 Network Two

The second network performs a multiclass classification task, identifying the musical genre of a given musical audio file. To this extent we used Convolutional Neural Networks (CNN).

4.1 Implementation

Similarly to what was done for the first network, we create a dataset out of the combination of both GTZAN and URBANSOUND8K.

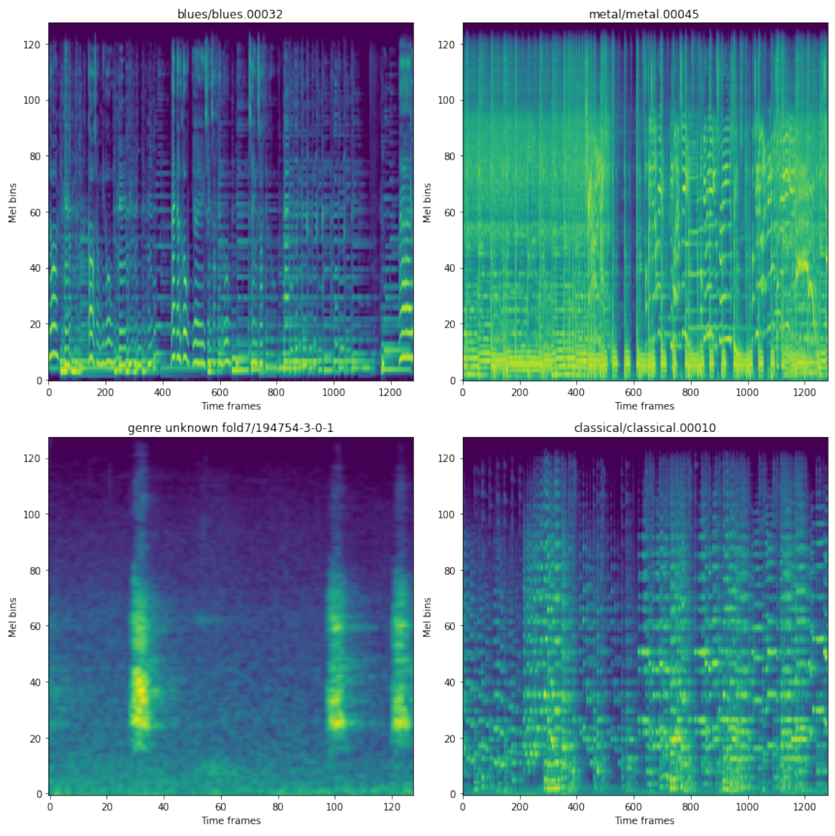


Fig. 4: Log mel spectograms generated for some examples from the dataset.

A total of 11 labels were thus generated, 10 for each musical genre of the GTZAN[2] dataset and one, labelled as *unknown*, to account for examples from URBANSOUND8K. Following a slightly different approach than in network one, here each audio file was divided into chunks of 3 seconds. In order to ensure balancing in the dataset, each genre contained a total of 1,000 chunks (GTZAN samples have a duration of around 30 seconds, while URBANSOUND8K samples last for around 4 seconds; thus, examples labelled as a musical genre consisted of 10 chunks per file, out of 100 files per genre, while example labels labelled as *unknown* consisted of 1 chunk per file out of 1,000 files). In total, the dataset consisted of 11,000 examples. In order to get the log mel spectograms we used *librosa.feature.melspectrogram*[3], obtaining data in the shape of time frames x 128 mel bins, as observable in Figure 4. For the sake of consistency, this step could be replaced by the same procedure for generating log mel spectograms applied in network one, but it was thought to be interesting to compare and use both approaches. Dataset was then shuffled and split into 70% for training, 20% for validation and 10% for testing. Regarding network architecture it consisted of a total of 5 convolutional layers with max pooling, followed by two fully connected layers. As per the convolutional size of the network, weight initialization from a normal distribuition was done, as well as batch normalization. Regarding activation functions on the fully connected layers, first layer used a ReLU and final layer used log softmax. Dropout was also applied between the two last layers. Cross entropy was used as the loss function. Regarding the optimization algorithm, *Adam*, Adaptive Moment Estmiation [3], was chosen.

4.2 Evaluation

First runs showed that the network was unable to increase performance after 100 epochs, so this value was chosen. Also, after some experimenting and constrains from GPU memory, a batch size of 16 was used.

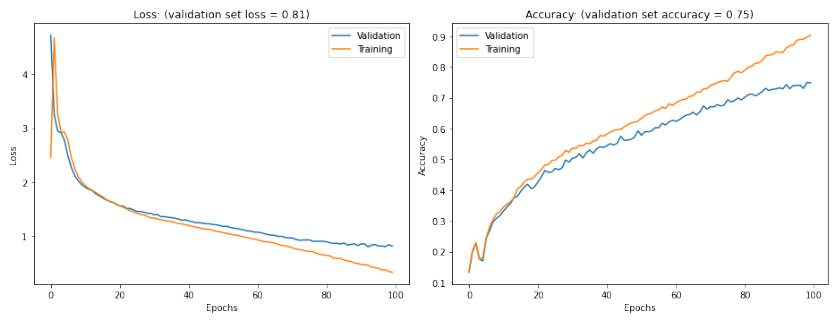


Fig. 5: Loss and accuracy for both (left) training set and (right) validation set.

[2] Namely *metal*, *reggae*, *hiphop*, *jazz*, *country*, *pop*, *blues*, *rock*, *disco* and *classical*.
[3] https://librosa.github.io/librosa/generated/librosa.feature.melspectrogram.html

Figure 5 shows that the network was able to achieve an accuracy of around 75% on the validation set. Considering the test set, an accuracy of 68% was obtained. By observing the confusion matrix in Figure 6 we can see that the network is very capable of predicting genres *unknown* and *reggae*, whilst it under performs with genre *classical*.

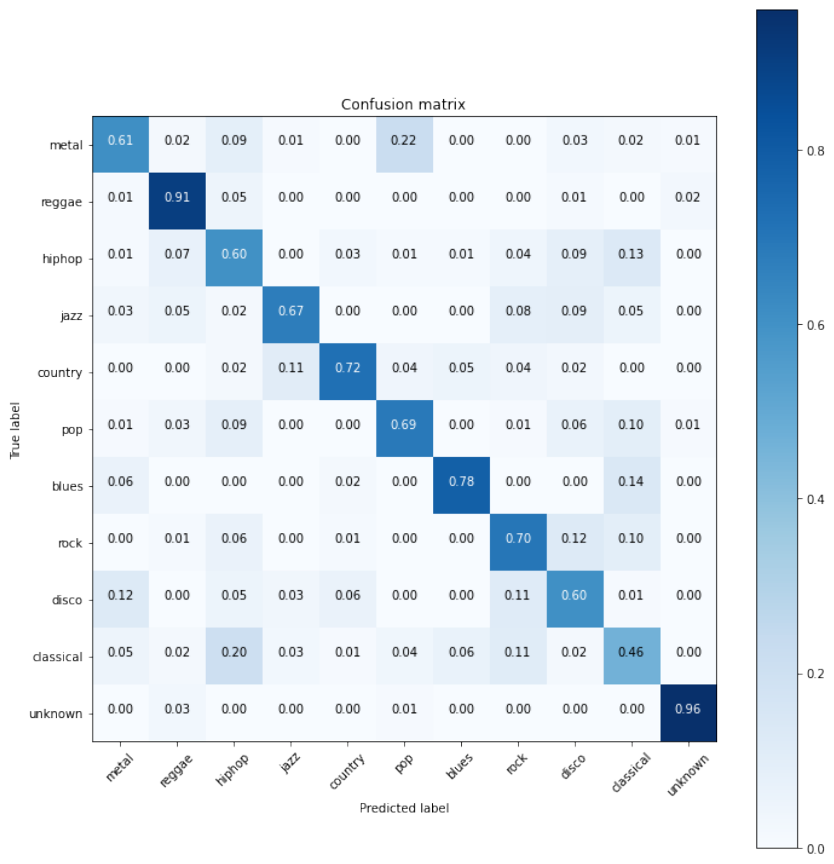
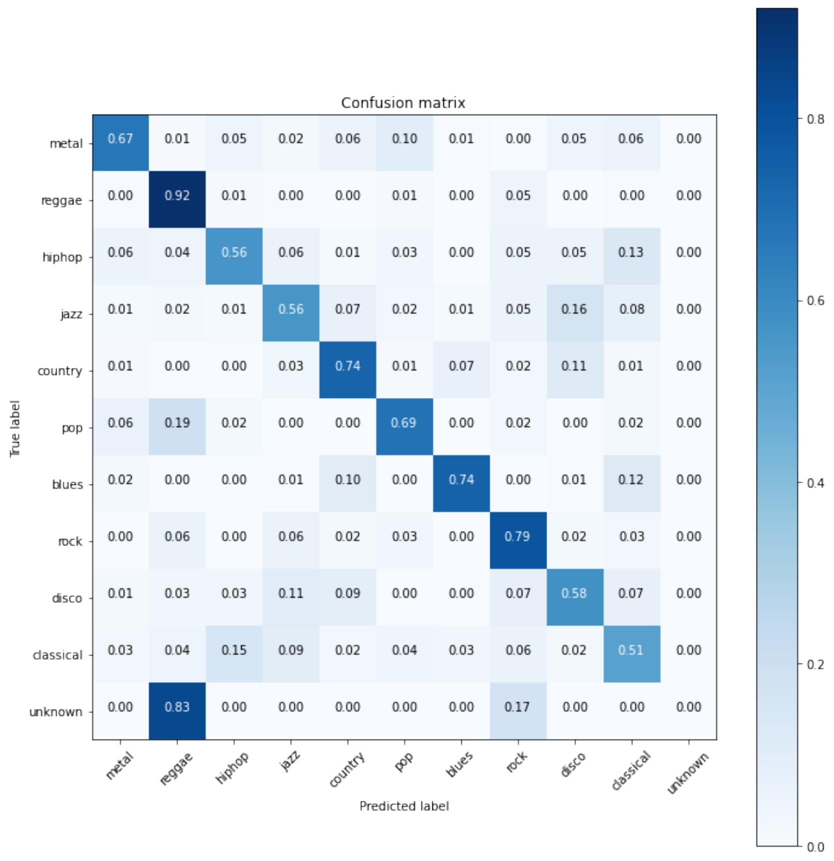


Fig. 6: Plot of the confusion matrix for the test set.

Next, with fed the network with the classifications from network one (saved as *music_keys* on the code). For this set we obtained an accuracy of 67%, very similar to the results achieved for the test set. On the other hand, the confusion matrix in Figure 7 provided interesting results. It is observable that all *unknown* labelled examples were given a genre. This could be explained by the fact that the majority of misclassifications from network one, here labelled as *unknown*, were in fact from class *street music*. By listening to some of those examples it was, although, not clear if they such a high percentage could be considered *reggae*.



	metal	reggae	hiphop	jazz	country	pop	blues	rock	disco	classical	unknown
metal	0.67	0.01	0.05	0.02	0.06	0.10	0.01	0.00	0.05	0.06	0.00
reggae	0.00	0.92	0.01	0.00	0.00	0.01	0.00	0.05	0.00	0.00	0.00
hiphop	0.06	0.04	0.56	0.06	0.01	0.03	0.00	0.05	0.05	0.13	0.00
jazz	0.01	0.02	0.01	0.56	0.07	0.02	0.01	0.05	0.16	0.08	0.00
country	0.01	0.00	0.00	0.03	0.74	0.01	0.07	0.02	0.11	0.01	0.00
pop	0.06	0.19	0.02	0.00	0.00	0.69	0.00	0.02	0.00	0.02	0.00
blues	0.02	0.00	0.00	0.01	0.10	0.00	0.74	0.00	0.01	0.12	0.00
rock	0.00	0.06	0.00	0.06	0.02	0.03	0.00	0.79	0.02	0.03	0.00
disco	0.01	0.03	0.03	0.11	0.09	0.00	0.00	0.07	0.58	0.07	0.00
classical	0.03	0.04	0.15	0.09	0.02	0.04	0.03	0.06	0.02	0.51	0.00
unknown	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.17	0.00	0.00	0.00

Fig. 7: Plot of the confusion matrix for the set consisting of the output of network one.

Finally, our test audio file, *Waving* by October Horse, was passed through the network. A snippet of 30 seconds (from 1:36 to 2:06), divided into chunks of 3 seconds was used. This excerpt encompasses the transition between section A and section B that occurs at 1:54 approximately. For evaluation, all chunks were labelled as *metal* (although section B could be considered as *non-metal*, closer to genres such as *jazz*, *blues* and *reggae*). An accuracy of 50% was obtained (5 out of 10 chunks were classified as metal), but a deeper inspection of the results showed that the last 4 seconds of the track were labelled as *blues* and as *reggae*, so subjectively this can be seen as an higher value of accuracy (in fact, during the compositional process of *Waving*, the idea of the band was indeed to create a "bluesish/reggaeish" section B).

5 Conclusions

Both networks seem to perform accurately, although some overfitting might be happening. This was patent when network one was not able to classified the entirety of *Waving* as music. Similary, despite the fairly accurate results on network two when using the *Waving* snippet, further tracks from the same album were analyzed, yielding poorer results (some classifications as *hip hop* when they should be *metal*, for example). This might be a case of overfitting given the fairly small dataset.
On a side note, in order to assess energy usage and carbon emissions related with this assignment, *experiment-impact-tracker* was used [1]. This was an unsuccessful attempt because the tracker was initially not designed to be used with *Jupyter notebooks*. This was raised as an issue in the projects' github repository[4] and the authors kindly volunteer to add a new feature to account for future usages with *Jupyter Notebooks*.

References

[1] Peter Henderson et al. *Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning.* 2020. arXiv: `2002.05651 [cs.CY]`.

[2] Shawn Hershey et al. "CNN Architectures for Large-Scale Audio Classification". In: *CoRR* abs/1609.09430 (2016). arXiv: `1609.09430`. URL: `http://arxiv.org/abs/1609.09430`.

[3] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization.* 2014. arXiv: `1412.6980 [cs.LG]`.

[4] J. Salamon, C. Jacoby, and J. P. Bello. "A Dataset and Taxonomy for Urban Sound Research". In: *22nd ACM International Conference on Multimedia (ACM-MM'14).* Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[5] G. Tzanetakis and P. Cook. "Musical Genre Classification of Audio Signals". In: *IEEE Transactions on Speech and Audio Processing* 10.5 (2002), pp. 293–302.

[4] https://github.com/Breakend/experiment-impact-tracker