MIREX TAGGING CONTEST: A DEEP NEURAL NET APPROACH (DRAFT)

Thierry Bertin-Mahieux, Yoshua Bengio, Douglas Eck
University of Montreal, CAN
{bertinmt,bengioy,eckdoug}@iro.umontreal.ca

ABSTRACT

We present one of our submission to MIREX 2008 audio tag classification contest. The algorithm uses a two hidden layers neural network.

1 INTRODUCTION

We shortly describe our submission to MIREX 2008 audio tag classification contest based on a neural network

2 ALGORITHM

2.1 Audio Features

We compute aggregate features [3] over $5s$ segments. Features consist of a constant-Q spectrogram, an autocorrelation vector, MFCC and its first and second derivatives (delta-MFFC and delta-delta-MFCC). The size of an example (features from one segment) is 466.

2.2 Autoencoders

We train each layer of our neural network as an autoencodeur, e.g. we minimize the negative log-likelihood between an example and its reconstruction through a one layer neural network. The representation in the hidden layer can be used as the input of a second autoencoder (see [2]). Autoencoders were trained on our personal collection of approx. 120K songs. Noise was added to the inputs as in [4].

2.3 Neural Network

We stack 2 autoencoders to form a 2-hidden layers neural network. It is fine-tuned by gradient descent on the contest data. To this data, we also import some of our data: features of 4000 songs by 500 artists from our personal collection, and a subset of *Last.fm* (`www.lastfm.com`) that were applied to them [1]. The neural network is trained to predict both tag distributions by minmizing the log-likelihood between the output of the model and the target distribution. We hope this multitasking helps the learning, as well as having more data to train on.

2.4 Architecture Selection

We train different neural netwroks, varying the learning rate and proportion of contest data versus our personal data in each iteration. We keep the model that has lowest error on a validation set. One thresholds per tag is found on this validation set by minimizing $F - score$ [1] . Then, we retrain with the same parameters a network on the whole training set.

2.5 Output

The output of the network is a vector of continuous values between 0 and 1, the size of the vector is the number of tags (in the contest), plus the number of tags from *Last.fm* during training.

3 DISCUSSION

The idea between this submission is to experiment with multitasking and transfer learning (learning on one dataset improves learning on other datasets). However, time constraints are a problem, as some deep neural networks are trained for weeks before achieving their best performance.

4 ACKNOWLEDGEMENTS

The authors would like to thank everyone that pulled this contest together, in particular the IMIRSEL team, Kris West, Michael Mandel and Edith Law.

5 REFERENCES

[1] Audioscrobbler. Web Services described at http://www.audioscrobbler.net/data/webservices/.

[2] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems (NIPS) 19*. 2007.

[3] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl. Aggregate features and AdaBoost for music classification. *Machine Learning*, 65(2-3):473–484, 2006.

[4] P. Vincent, H. Larochelle, Y. Bengio, and P-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML08)*. 2008.

[1] `http://en.wikipedia.org/wiki/F-score`