AUDIO DEEPDREAM: OPTIMIZING RAW AUDIO WITH CONVOLUTIONAL NETWORKS.

Diego Ardila **Cinjon Resnick** **Adam Roberts** **Douglas Eck**
Google Brain, Mountain View, California, USA
{ardila,cinjon,adarob,deck}@google.com

ABSTRACT

The hallucinatory images of DeepDream [8] opened up the floodgates for a recent wave of artwork generated by neural networks. In this work, we take first steps to applying this to audio. We believe a key to solving this problem is training a deep neural network to perform a perception task on raw audio. Consequently, we have followed in the footsteps of Van den Oord et al [13] and trained a network to predict embeddings that were themselves the result of a collaborative filtering model. A key difference is that we learn features directly from the raw audio, which creates a chain of differentiable functions from raw audio to high level features. We then use gradient descent on the network to extract samples of "dreamed" audio. Examples are available at http://tiny.cc/78qqdy.

1. INTRODUCTION

The work of Mordvintsev et al [8], widely known as DeepDream, combined three key steps towards getting neural networks to "dream" images.

1. A meaningful and challenging perceptual task that covers a large portion of the stimulus space. [9]

2. The right architecture and training procedure to solve this task with a deep neural network. [7] [10] [11]

3. Applied constraints on the stimulus space to align the gradient descent optimization with natural images.

The first two steps are critical and produce a model with a hierarchical understanding of perception, which is a key foundation for more recent style transfer work [2] [6] [12]. The last step is more ad-hoc, but also a novel way to attain models with interesting generative capabilities.

We believe that all three of these questions are much further from resolution in audio than in vision, and we are presenting our first attempt at each of them. For our task, we predicted collaborative filter embeddings from audio. For our network, we used a six-layer convolutional architecture. And for optimization, we used gradient descent and experimented with some regularization functions based on the output of the first layer.

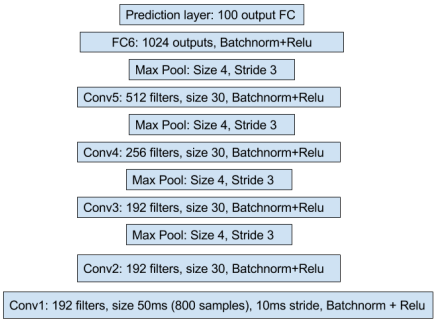

Figure 1. Architecture of convolutional network. Layers not illustrated to scale.

2. DETAILS

2.1 Task: Predict collaborative filtering embeddings

We trained a network to predict 100-dimensional collaborative filtering (CF) track embeddings from 30-second clips of music audio. The audio had a sample rate of 16khz, set to a mean of zero and then normalized by the maximum value of each mini-batch.

The target CF embeddings were generated by applying the Weighted Alternating Least Squares (WALS) [4] factorization algorithm to a sparse matrix containing user music listening history. User embeddings were not stored.

One challenge with this task is that the embeddings vary greatly in scale, which can cause problems for the L2 loss function. In addition, the norm of each embedding is correlated with popularity. Therefore, we decided to divide each embedding by its L2 norm. Across a wide variety of networks, we found empirical evidence (R^2 coefficient) that it was better to use normalized embeddings than unnormalized embeddings.

2.2 Network: six-layer convolutional network with raw audio input

The architecture of the network we trained is shown in Figure 1. It's fairly straightforward, but we found several de-

Thanks to Jenny Liu, Ron Weiss and Lucas Abend.

tails to be important to performance.

- Strided convolution in the first layer. Starting with raw audio means that there needs to be significant downscaling. Most of it occurs at the first layer by applying 50ms filters with 10ms strides.
- Batch normalization [5]: We use batch normalization after every layer, a necessary addition.
- Overlapping pooling: We overlap the max pooling windows after each step, which we found to improve performance.
- No Logrelu: In contrast with previous work [3], we did not find any significant boost from applying a logarithm after the first layer.
- 192 units in the first layer: We found that performance suffers when there are not enough filters in the first layer. Too few filters led to clustering in the lower end of the spectrum. It may be possible to add even more.

We used a learning rate of .03, and trained for 300k minibatches, each of size 64. Our peak performance on a held out validation set was an R^2 (coefficient of determination) of .23. We have not yet compared this to previously existing attempts at this task.

As has been seen in other studies applying feature learning directly to raw audio [1] [3], the first layer of filters learned a roughly log-spaced frequency selective bank (See figure 2).

There is still a large amount of noise in the learned first layer filters. This can be seen in Figure 3. Audio generated using this input and output will always contain noise. Finding a way to produce a network with cleaner filters in the first layer will be critical in improving upon the audio we generate.

2.3 Optimization: Gradient descent plus a couple of tricks

In order to generate sounds, we start from either noise or silence and then select different targets within the network to optimize. An example target could be to maximize the mean output of the last layer. We then use gradient descent on the input of the network, as in [8]. The output we attain is not musical, but it is also not purely noise. Here are some approaches we've explored:

- Normalizing gradient by its maximum, then multiplying by a learning rate.
- Adding a "sparsity constraint" by taking the L1 norm across the entire output of the first layer and adding it as a regularization term in the optimization.
- Adding a "continuity cost" where neighboring frames are penalized for having different first layer outputs. The total sum of square differences between neighboring frames in the output of the first layer is added as a regularization term.

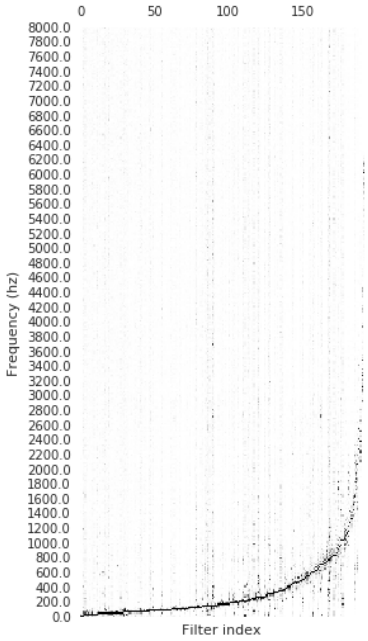

Figure 2. Spectra of filters learned in the first layer arranged by dominant frequency

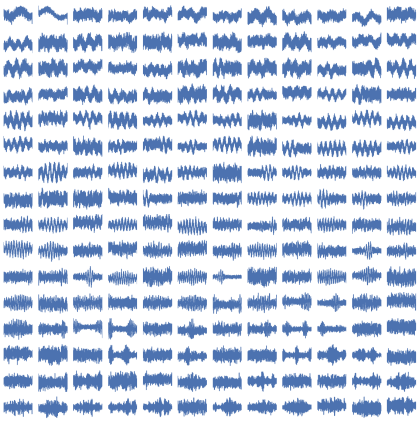

Figure 3. Filters learned in the first layer arranged by dominant frequency. Placed into row-major order for brevity.

3. CONCLUSION

In this work, we laid out our first attempts at building a network and procedure for applying DeepDream to audio. Possible future directions include alternative or additive perception tasks, better training procedures, and more advanced constraints in the optimization procedure.

Sample output can be found at http://tiny.cc/78qqdy.

4. REFERENCES

[1] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *ICASSP*, 2014.

[2] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[3] Y. Hoshen, R. J. Weiss, and K. W. Wilson. Speech Acoustic Modeling from Raw Multichannel Waveforms. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, April 2015.

[4] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *In IEEE International Conference on Data Mining (ICDM 2008*, pages 263–272, 2008.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[6] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[8] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015.

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *CoRR*, abs/1603.03417, 2016.

[13] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2643–2651. Curran Associates, Inc., 2013.