Notes on CNN Architectures for Large-Scale Audio Classification

Paper link Authors: Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, Kevin Wilson (Google, Inc., New York, NY, and Mountain View, CA, USA) Year: 2016

Abstract

Convolutional Neural Networks (CNNs) have proven very effective in image classification and show promise for audio. We use various CNN architectures to classify the soundtracks of a dataset of 70M training videos (5.24 million hours) with 30,871 video-level labels. We examine fully connected Deep Neural Networks (DNNs), AlexNet, VGG, Inception, and ResNet. We investigate varying the size of both training set and label vocabulary, finding that analogs of the CNNs used in image classification do well on our audio classification task, and larger training and label sets help up to a point. A model using embeddings from these classifiers does much better than raw features on the Audio Set Acoustic Event Detection (AED) classification task.

Notes

- They use log-mel spectrograms of multiple frames to create 2D-image like patches to present to the models.
- YouTube-100M: Our dataset consists of 70 million (henceforth 70M) training videos totalling 5.24 million hours, each tagged from a set of 30,871 (henceforth 30K) labels.
- Input processing: The audio is divided into non-overlapping 960 ms frames. The 960 ms frames are decomposed with a short-time Fourier transform applying 25 ms windows every 10 ms. The resulting spectrogram is integrated into 64 mel-spaced frequency bins, and the magnitude of each bin is logtransformed after adding a small offset to avoid numerical issues. This gives log-mel spectrogram patches of 96 x 64 bins that form the input to all classifiers. During training we fetch mini-batches of 128 input examples by randomly sampling from all patches.

Architectures

- Baseline: Our baseline network is a fully connected model with RELU activations, N layers, and M units per layer. We swept over N = (2; 3; 4; 5; 6) and M = (500; 1000; 2000; 3000; 4000). Our best performing model had N = 3 layers, M = 1000 units,

learning rate of 3 x 10^5, 10 GPUs and 5 parameter servers. This network has approximately 11.2M weights and 11.2M multiplies.

- AlexNet
- VGG
- Inception V3
- ResNet-50

Observations

- More params, better

Table 2: Comparison of performance of several DNN architectures trained on 70M videos, each tagged with labels from a set of 3K. The last row contains results for a model that was trained much longer than the others, with a reduction in learning rate after 13 million steps.

Architectures	Steps	Time	AUC	d-prime	mAP
Fully Connected	5M	35h	0.851	1.471	0.058
AlexNet	5M	82h	0.894	1.764	0.115
VGG	5M	184h	0.911	1.909	0.161
Inception V3	5M	137h	**0.918**	**1.969**	0.181
ResNet-50	5M	119h	0.916	1.952	**0.182**
ResNet-50	17M	356h	**0.926**	**2.041**	**0.212**

- More labels, better

Table 3: Results of varying label set size, evaluated over 400 labels. All models are variants of ResNet-50 trained on 70M videos. The bottleneck, if present, is 128 dimensions.

Bneck	Labels	AUC	d-prime	mAP
no	30K	—	—	—
no	3K	**0.930**	**2.087**	**0.381**
no	400	0.928	2.067	0.376
yes	30K	0.925	2.035	0.369
yes	3K	0.919	1.982	0.347
yes	400	0.924	2.026	0.365

- More training data, beter

Table 4: Results of training with different amounts of data. All rows used the same ResNet-50 architecture trained on videos tagged with labels from a set of 3K.

Training Videos	AUC	d-prime	mAP
70M	**0.923**	**2.019**	**0.206**
7M	0.922	2.006	0.202
700K	0.921	1.997	0.203
70K	0.909	1.883	0.162
23K	0.868	1.581	0.118

- For another task (audio event detection), they used outputs of their best model as embeddings. They got better results.

Code

VGGish

There is a tensorflow implementation of a VGG audio classifier that can be used for generating embeddings called VGGish. Torch implementation also available. It is also implemented in towhee.

Conclusions

- CNNs might be good architecture for out task
- We can replicate their input processing technique
- Size of training data is worrying
- We can attempt using VGGish in the initial layers of a larger model.