Audio Multi-label Classification and Applications

Khrylchenko Kirill
Mazaev Pavel
Ivanov Sergey
Kodryan Maxim

June 19, 2019

Data Preparation

Available data[1]**:**

- 4970 audio samples
- 80 audio tags: screaming, yell, bark, sigh, gasp, etc...

Multi-label classification: given an audiofile, assign probabilities of 80 independent classes (not softmax, sigmoid).

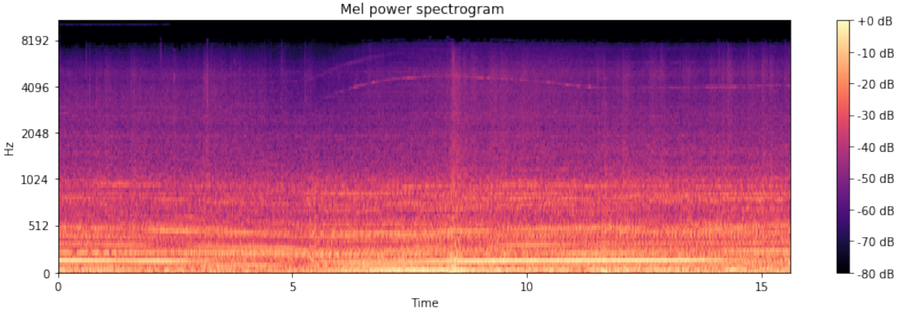
Data preparation approaches:

- melspectrograms:
  - images, 128 x 128
  - sequences of frame feature vectors, T x 128
- raw input — 2 seconds $\times$ 44100 $=$ 88200 numbers
- mu-law encoding — not going to discuss it

[1] https://www.kaggle.com/c/freesound-audio-tagging-2019/

Melspectrograms

- Audiofile is represented as a sequence of 128 overlapping frames
- Feature vector of size 128 is calculated for every frame
- **librosa.feature.melspectrogram** — calculates melspectrogram



Augmentations

- An obvious idea: merge several files and their labels
- Random samples!
- With random weights!
- **Natural filters!** – reverberation with a random IR from a set

Also attempted: Normal noise, pitch shift

- ✓ No overfitting
- × Slow (by iterations)
- × Very slow (by time)

Augmentations

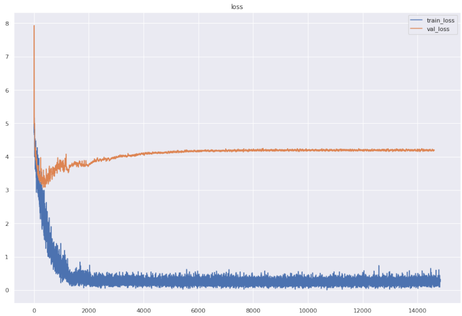


Figure: Without augmentation

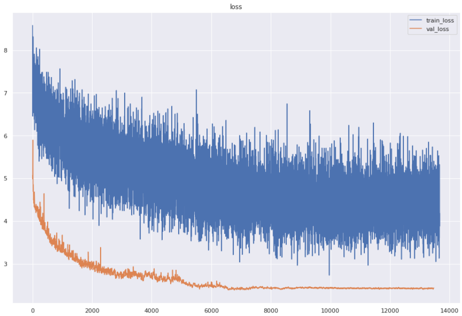


Figure: With augmentation

Classification Models

Melspectrogram-based neural networks:

- Deep Convolutional Neural Network for Environmental Sound Classification — original model and modification
- Masked Conditional Neural Networks for Audio Classification
- CNN Classifiers pretrained on ImageNet[2] — didn't work
- Kaggle-based Model
- GRU-based Model

Raw input neural networks:

- SampleCNN — analogue of VGG
- ReSE2-Multi — analogue of ResNet

[2] https://arxiv.org/pdf/1609.09430v2.pdf

Deep Convolutional Neural Network for Environmental Sound Classification[3]

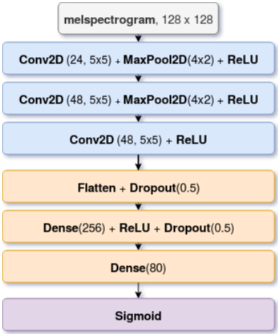


Figure: DCNN Model

Modifications:

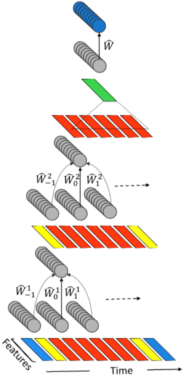
- Replace **ReLU** with **LeakyReLU**
- **BatchNorm2D** before activations
- try **InstanceNorm2D**?
- increase amount of filter maps

[3]https://arxiv.org/pdf/1608.04363v2.pdf

Masked Conditional Neural Networks for Audio Classification[4]

General ideas

- 1d convolutions along the features
- Multiple convolutions applied to a window
- $y_t = f(b + \sum_{u=-n}^{n} x_{u+t} W_u)$



[4] https://arxiv.org/pdf/1803.02421v2.pdf

MCNN

Masks

- Weights are masked
- Different channels have different source channels
- $\bar{W}_u = W_u \odot M$
- Provides a little performance boost

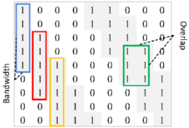


Figure: A mask for one W_u

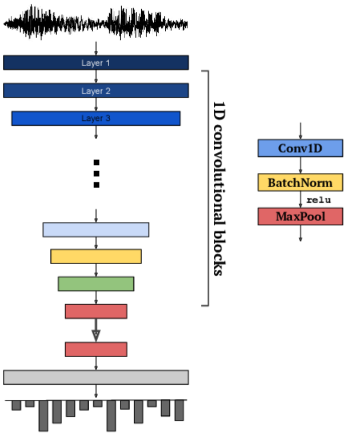
Sample-level Deep Convolutional Neural Networks[5]

SampleCNN

- ✓ may take into account phase
- × memory-heavy (look at the first layer)

Ideas

- strided convolutions at the beginning
- pooling with kernel=3 instead 2



[5] https://arxiv.org/pdf/1703.01789.pdf

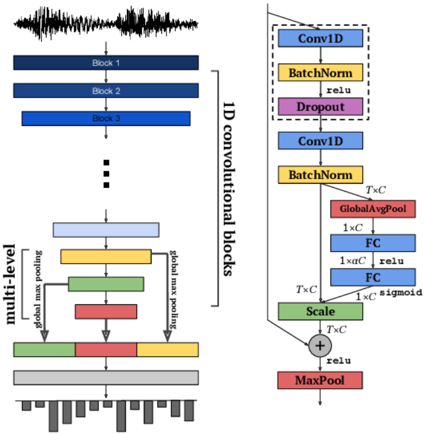
Raw Waveform-based Audio Classification[6]

ReSE-2-Multi Model

- ✓ allows to increase number of convolutional layers
- × still memory-heavy

Ideas

- add residual connections
- concatenate features from several last layers



[6]https://arxiv.org/pdf/1712.00866.pdf

Results

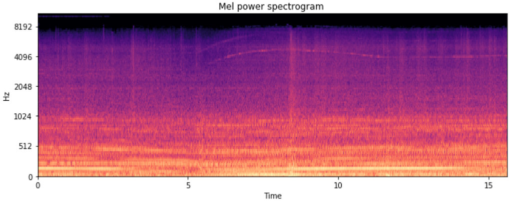
Model	**Lwlrap**[7]	**Time**[8], sec
DCNN	0.6295	1
modDCNN	0.7028	1
GRU	0.4639	2.15
Kaggle	**0.7876**	15.1
MCNN, no augmentations	0.6149	4.53
MCNN, no augmentations, no masks	0.5573	4.41
MCNN, augmentations	0.6727	76.04
MCNN, augmentations, no masks	0.6313	76.17
SampleCNN	0.6356	14.09
ReSE-2-Multi	0.6882	25.25

[7] evaluation description

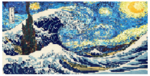
[8] 1 epoch time

Audio-image Style Transfer?

1. Audio-audio style transfer[9] is boring...
2. Why not try audio-image ST?!
3. Apply ST model to melspectrograms **images** and use Griffin-Lim to restore audio!
4. What do we get? Let's listen!



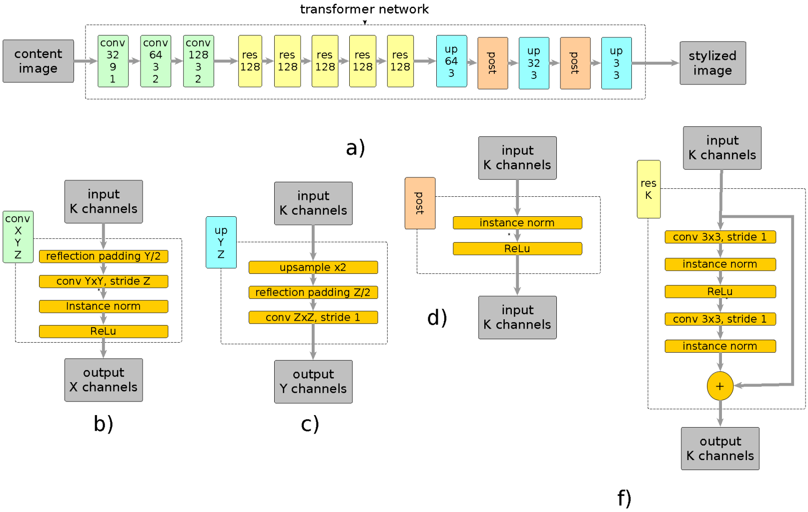
+



= ???

[9] https://github.com/inzva/Audio-Style-Transfer

ST model



transformer network
content image
conv 32 9 1
conv 64 3 2
conv 128 3 2
res 128
res 128
res 128
res 128
res 128
up 64 3
post
up 32 3
post
up 3 3
stylized image
a)
conv X Y Z
input K channels
reflection padding Y/2
conv YxY, stride Z
Instance norm
ReLu
output X channels
b)
up Y Z
input K channels
upsample x2
reflection padding Z/2
conv ZxZ, stride 1
output Y channels
c)
post
input K channels
instance norm
ReLu
input K channels
d)
res K
input K channels
conv 3x3, stride 1
instance norm
ReLu
conv 3x3, stride 1
instance norm
+
output K channels
f)

Contribution

- Khrylchenko: data preparation, training pipeline, DCNN, GRU, Kaggle models;
- Mazaev: augmentation experiments, augmentation pipeline, MCNN model and experiments;
- Ivanov: raw input pipeline, SampleCNN and ReSE-2-Multi models;
- Kodryan: style transfer, code review, article discussion.

Thanks for your **attention**[10]!

[10] Attention Is All You Need, Vaswani et al