VERY DEEP CONVOLUTIONAL NEURAL NETWORKS FOR RAW WAVEFORMS

Wei Dai*[1], Chia Dai*[1], Shuhui Qu[2], Juncheng Li[3], Samarjit Das[3]

[1]Carnegie Mellon University, [2]Stanford University, [3]Bosch Research

Motivation

- Predictive modeling on environmental sound conventionally requires feature engineering based on Mel-frequency cepstrum coefficient (MFCC) and Gaussian Mixture Model features.
- End-to-end discriminative representation learning is highly effective for many areas, such as image classification and speech recognition.

Background – Raw Waveform

- Raw waveform: the sound signal in time domain represented by a 1-D vector.
- Very high dimensional, conventionally not directly used as model input.
- Prior success using acoustic waveform as model input: 2-layer convolutional neural networks (CNNs) on speech recognition [1]
- No CNNs deepr than 2-layers on environmental sounds which span much wider frequency spectrum

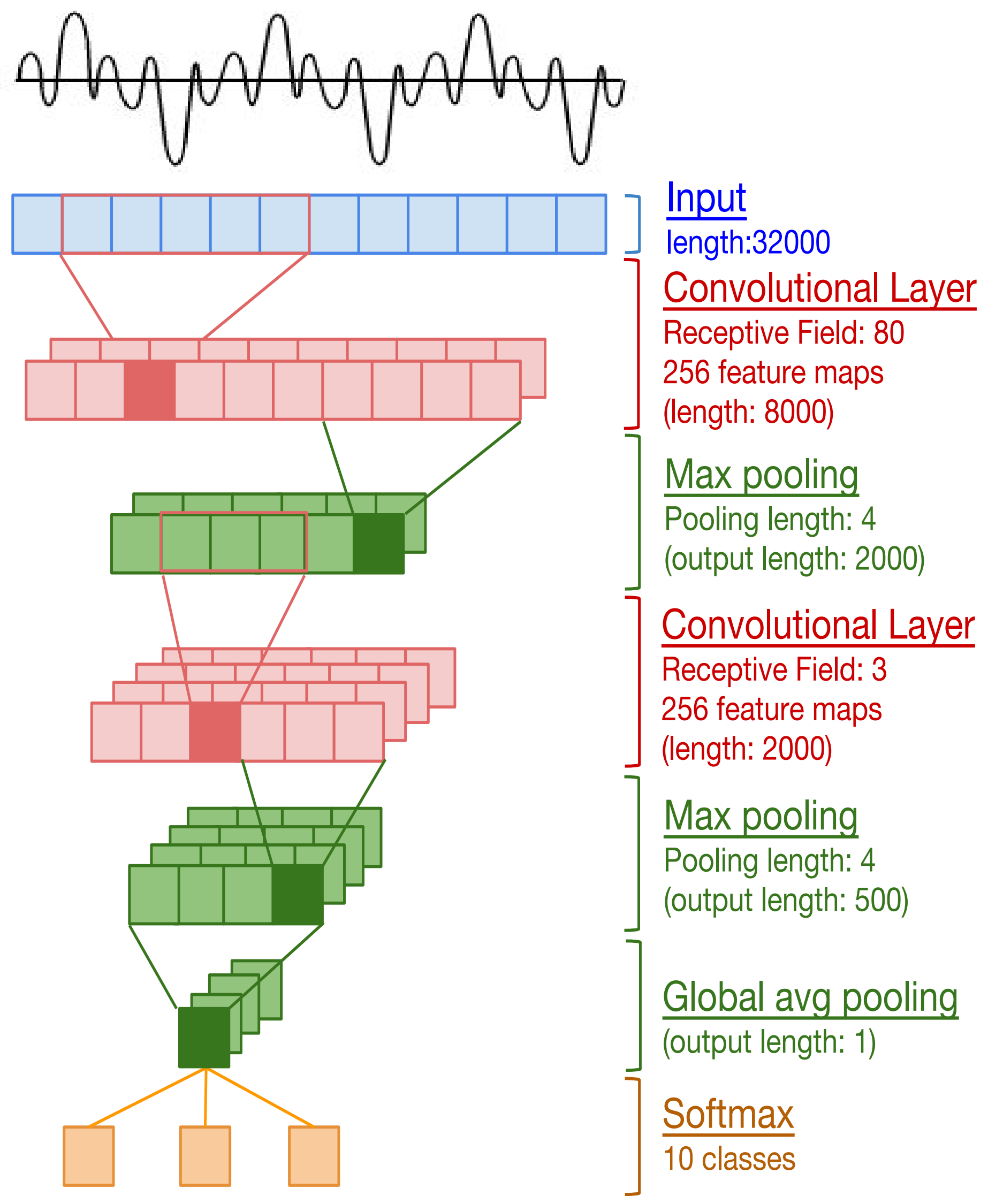
Background – Very Deep CNNs

- Very deep convolutional neural networks (CNNs) have achieved much higher accuracy than shallower networks in the visual domain.
- The deep network poses computational challenges like vanishing / exploding gradients, long training time per epoch, and overfitting.

Research Question

Which deep CNN architectures are suitable for environmental sound classification on raw acoustic waveforms? What is the performance compared with models using log-mel features?

Model Overview



Basic Building Blocks

•Inspired by VGG network design [2] and residual network [3]

•Fully convolutional (1-D convolutions)

•Max pool reduces temporal resolution by 4 and doubles the number of feature maps, comparable to vision (2x2 pooling)

•Small receptive field (3) except the first layer

Key Design Decisions

Depth of the network

•Increasing the number of convolutional layers substantially improves the accuracy, up to 18 convolutional layers.

•Deeper network increases computation time. We use stride 4 in the first convolutional layer to improve speed.

Receptive field size

•Unlike image models, large receptive field in the first layer is necessary to learn good filters.

•We find that receptive field 80 for first layer on 8kHz audio (10ms duration) works well

•Small receptive field (3) for the rest of convolutional layers

Key Design Decisions (continued)

Fully convolutional

•We remove fully connected layers to induce better representation learning in the convolutional layers.

•Substantially reduces the number of parameters and counteracts overfitting.

Residual learning

•Learn the residual mapping: F(x) = H(x) – x, where x is in the input to the layer and H(x) is the desired mapping.

•Improves convergence for very deep CNNs.

•We use the variant shown in Fig-1 (b)

Batch normalization (BN)

•Auxiliary layer to alleviate internal covariate shift and improve convergence.

•Necessary for training deeper CNNs

Architectures

<table>
<tr><td>M3 (0.2M)</td><td>M5 (0.5M)</td><td>M11 (1.8M)</td><td>M18 (3.7M)</td><td>M34-res (4M)</td></tr>
<tr><td colspan="5">Input: 32000x1 time-domain waveform</td></tr>
<tr><td>[80/4, 256]</td><td>[80/4, 128]</td><td>[80/4, 64]</td><td>[80/4, 64]</td><td>[80/4, 48]</td></tr>
<tr><td colspan="5">Maxpool: 4x1 (output: 2000 × n)</td></tr>
<tr><td>[3, 256]</td><td>[3, 128]</td><td>[3, 64] × 2</td><td>[3, 64] × 4</td><td>$\begin{bmatrix} 3, 48 \\ 3, 48 \end{bmatrix} \times 3$</td></tr>
<tr><td colspan="5">Maxpool: 4x1 (output: 500×n)</td></tr>
<tr><td></td><td>[3, 256]</td><td>[3, 128] × 2</td><td>[3, 128] × 4</td><td>$\begin{bmatrix} 3, 96 \\ 3, 96 \end{bmatrix} \times 4$</td></tr>
<tr><td></td><td colspan="4">Maxpool: 4x1 (output: 125 × n)</td></tr>
<tr><td></td><td>[3, 512]</td><td>[3, 256] × 3</td><td>[3, 256] × 4</td><td>$\begin{bmatrix} 3, 192 \\ 3, 192 \end{bmatrix} \times 6$</td></tr>
<tr><td></td><td colspan="4">Maxpool: 4x1 (output: 32 × n)</td></tr>
<tr><td></td><td></td><td>[3, 512] × 2</td><td>[3, 512] × 4</td><td>$\begin{bmatrix} 3, 384 \\ 3, 384 \end{bmatrix} \times 3$</td></tr>
<tr><td colspan="5">Global average pooling (output: 1 × n)</td></tr>
<tr><td colspan="5">Softmax</td></tr>
</table>

- 5 models studied. M3 (0.2M) represents 3 convolutional layers with 0.2M parameters.
- [3, 64] x 2 denotes two blocks of convolutional layers, each with receptive field 3 and 64 feature maps.
- Stride = 1 unless specified as [80/4, 128] for stride 4.

Architectures

- Double layers in a bracket denotes residual block:

$$\begin{bmatrix} 3, 48 \\ 3, 48 \end{bmatrix} \times 3$$

- Output 500 x n in max pool layer denotes the output temporal dimension to be 500

Experiment Results

- 64GB memory with Titan X GPU
- Trained for 150-400 epochs till convergence

Deeper is better, up to 18 layers, test accuracies increases with more layers. Training time per epoch increase mildly

Model	Test	Time
M3	56.12%	77s
M5	63.42%	63s
M11	69.07%	71s
M18	71.68%	98s
M34-res	63.47%	124s

More filters do not help shallower networks. Additional filters for shallow networks (M3, M5) marginally improve performance.

Model	Test	# Parameters
M3-big	57.55%	0.5M
M5-big	63.30%	2.2M

Performance sensitive to receptive field (RF) size in first layer. Test accuracies for M11 and M18 suffers with small RF (srf, RF=8) and large RF (lrf, RF=320).

Model	Test
M11-srf	64.78%
M18-srf	65.55%
M11-lrf	65.67%
M18-lrf	65.08%

BN is necessary for training very deep CNNs

Model	Train	Test
M11-no-bn	98.58%	69.38%
M18-no-bn	99.33%	62.48%
M34-no-bn	10.96%	11.45%

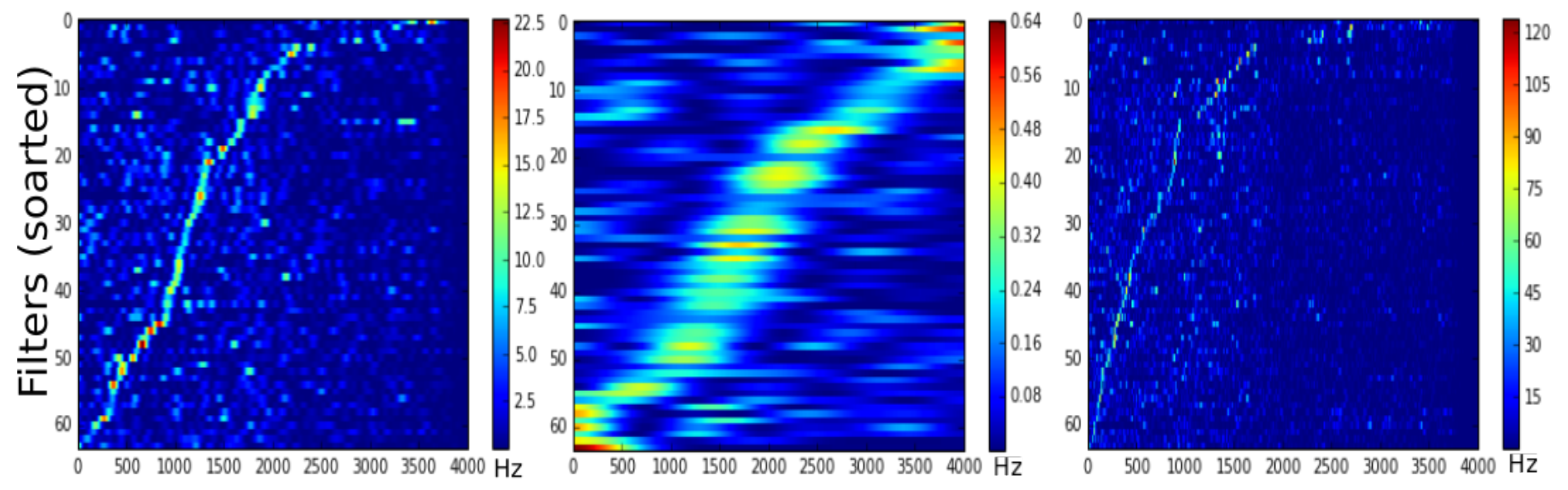
Experiments cont'd

Fully connected (FC) layer does not help. M3, M5, M11, M18 see no improvement with FC layer

Model	Test	# Parameters	Time
M3-fc	46.82%	129M	150s
M5-fc	62.76%	18M	66s
M11-fc	68.29%	1.8M	73s
M18-fc	64.93%	8.7M	100s

Visualization of Learned Kernels

- Fourier transform on the 1st convolutional layer weights of M18 show that they act as filter banks.
- Filter bank quality sensitive to receptive field (RF) size. Left: proper RF size lead to well-formed filters; Middle: small RF gives dispersed band and lower frequency resolution; Right: large RF lacks sufficient filters in high frequency range.



Conclusion

- We propose very deep fully convolutional networks, with up to 34 convolutional layers, for acoustic waveform. Our models uses large receptive field, and are efficient to train thanks to aggressive down sampling and batch normalization.
- Our CNN with 18 layers outperforms the 2-layer network by 15.56% accuracy absolutely and is competitive with models using log-mel features

References

[1] Tara N Sainath, et al, "Learning the speech front-end with raw waveform cldnns" Interspeech, (2015).

[2] Karen Simonyan, Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition" arXiv (2014).

[3] Kaiming He, et al, "Deep residual learning for image recognition" arXiv (2015).