# Machine Learning Bootcamp

•••

August 19, 2021

# Overview

- Paper name: VERY DEEP CONVOLUTIONAL NEURAL NETWORKS FOR RAW WAVEFORMS.
- Authors: Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, Samarjit Das.
- Released: 1 Oct 2016

# Understanding the problem

### Item 1

Challenging and time intensive to find the right feature representation of the speech.

Might not be optimal for the predictive task.

Discuss the use of waveform direcrectly.

### Item 2

Learning acoustic models directly from the raw waveform is computational intensive.

Simpler representation allows the model to jointly learn the features and do the classification.

### Item 3

Fully convolutional net reduces the number of parameters.

Forces learn of good representation in the convolutional layer.

Deep network to deal with the long audio sequences.

# Proposed Solution

| | |
|---|---|
| Input | ● Waveform representation. |
| Models | ● Deep fully convolutional network. |
| Additional layers | ● Batch normalisation layers.<br>● Residual blocks. |
| Task | ● Environmental sound recognition |

# They did the following to allow very deep network

### Batch normalisation

Fights exploding and vanishing gradients

### Residual connection

Allows the selection of more simpler models

### Glorot initialisation

Fights exploding and vanishing gradients

### Small filter size

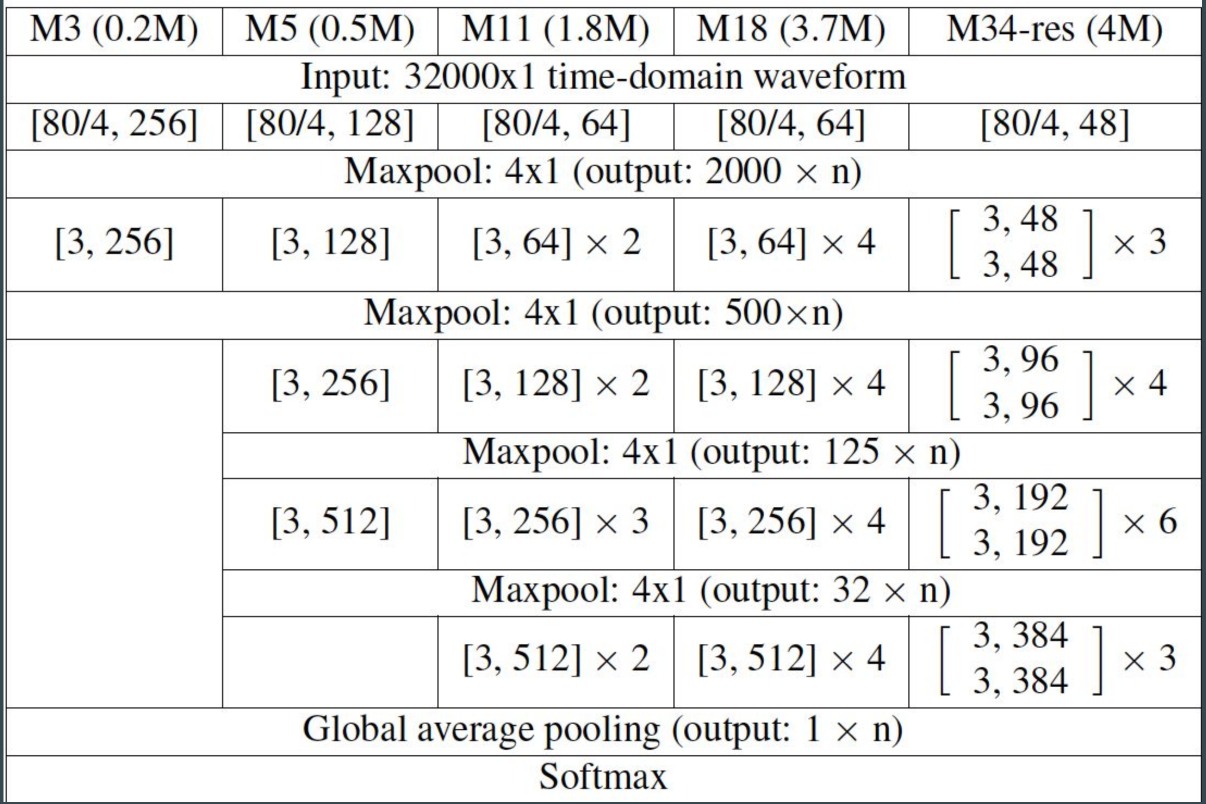
Provides larger receptive field with less number of parameters.

# Dataselt

- URBANSOUND8K DATASET.
- 8732 labeled sound excerpts (<=4s).
- 18.5 hours.
- 10 classes: air_conditioner, car_horn, children_playing, dog_bark, drilling, enginge_idling, gun_shot, jackhammer, siren, and street_music.
- Divided into 10 folds.
- Warning: don’t reshuffle the folds; will be incorrectly placing related samples in both the train and test sets; and hence it will provide wrong results.

# Experiments

# Models



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

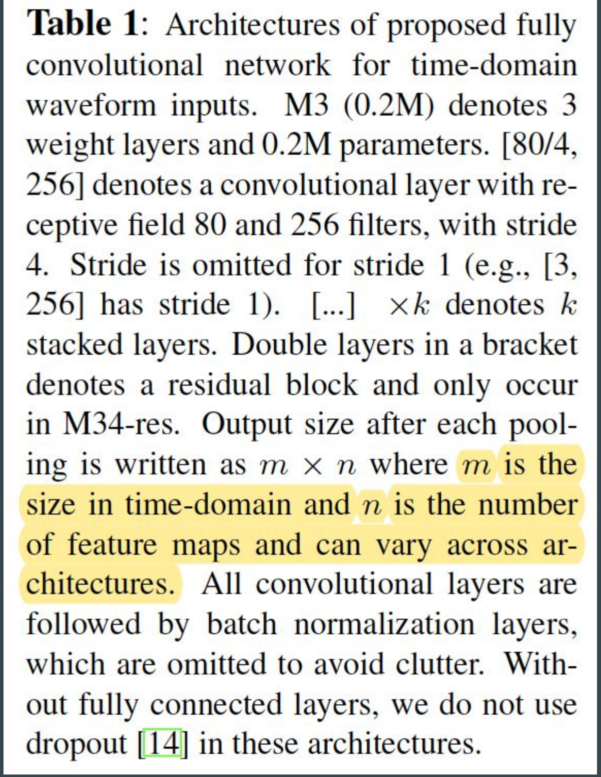


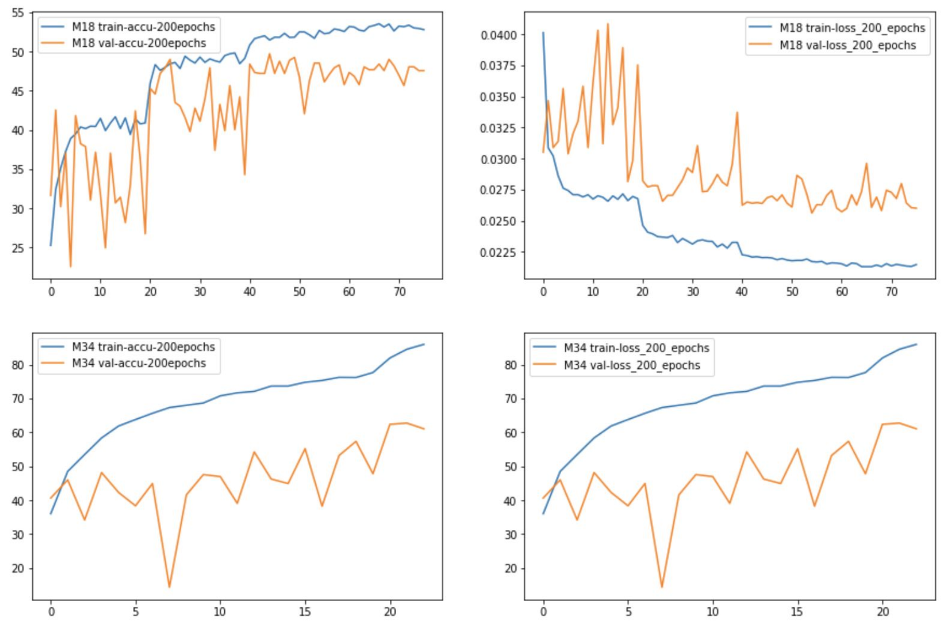
**Table 1**: Architectures of proposed fully convolutional network for time-domain waveform inputs. M3 (0.2M) denotes 3 weight layers and 0.2M parameters. [80/4, 256] denotes a convolutional layer with receptive field 80 and 256 filters, with stride 4. Stride is omitted for stride 1 (e.g., [3, 256] has stride 1). [...] $\times k$ denotes $k$ stacked layers. Double layers in a bracket denotes a residual block and only occur in M34-res. Output size after each pooling is written as $m \times n$ where $m$ is the size in time-domain and $n$ is the number of feature maps and can vary across architectures. All convolutional layers are followed by batch normalization layers, which are omitted to avoid clutter. Without fully connected layers, we do not use dropout [14] in these architectures.

# What I did so far

## Model implementation

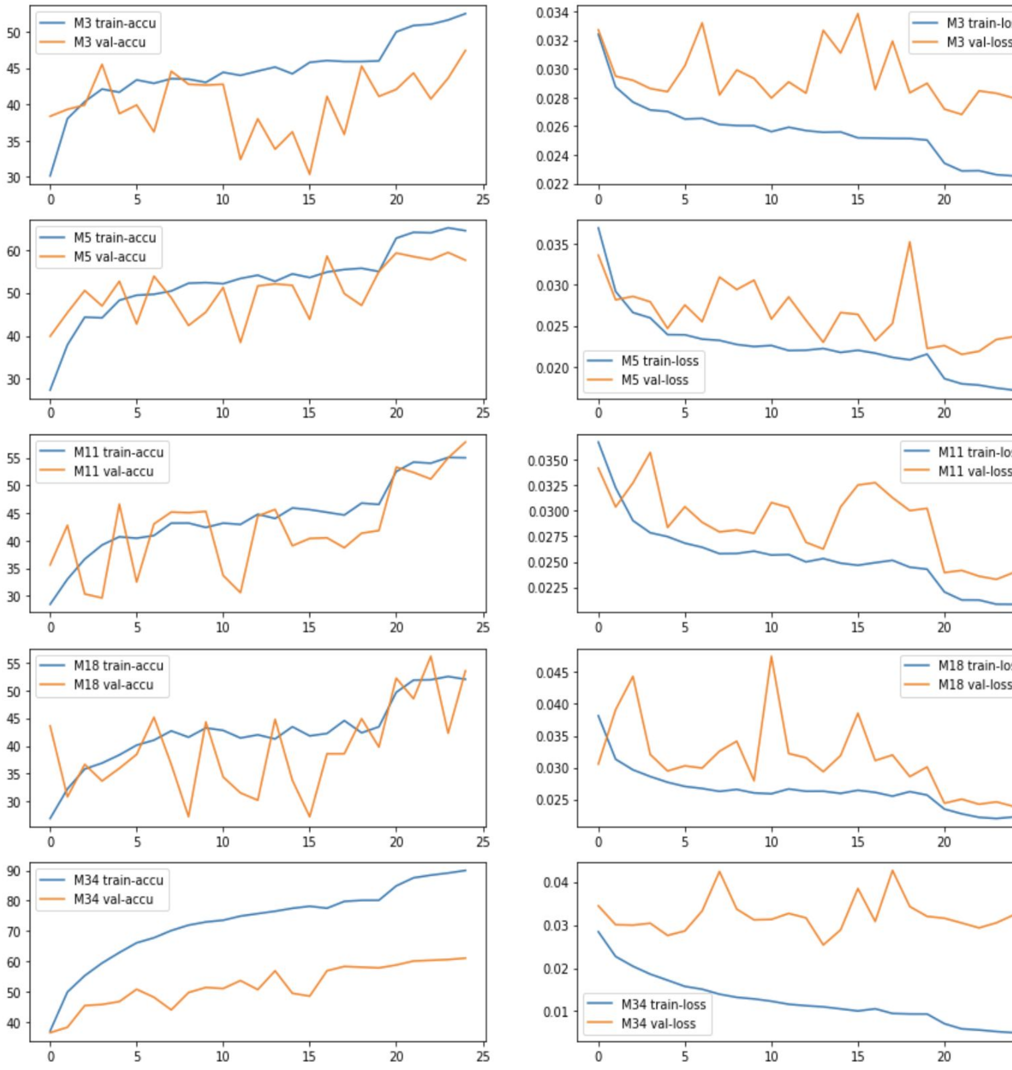
I implemented all the aforementioned architectures and test all the architectures with 25 epochs. For experiment M18 and M34 I tested it with 200 epochs and achieved accuracy of 49.22% and 61.05% respectively compared the paper’s accuracy of 71.68% and 63.47% respectively.

- I cannot guarantee that I will get the same results but the first results is promising.



# Results

The test accuracy of the different models after 25 epochs of training with 8 folds and test on fold 10.



# Future Work

- Repeat the experiments with 400 epochs to compare the results.
- Change the number of the filters and/or the number of the layers and see the results.
- Compare the results with FC network.
- Test the same models with feature representation.

# Thank you for your time !

# Questions ?