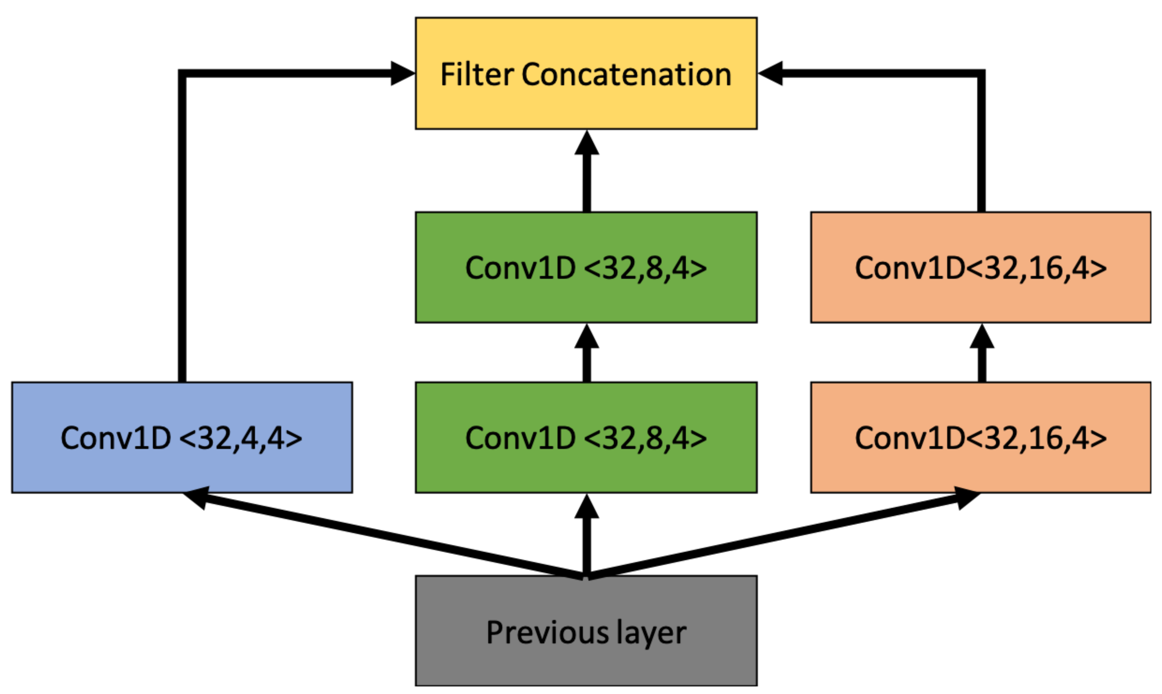
End-to-End Sound Classification On Loihi Neuromorphic Chip

Mohammad K. Ebrahimpour[1,2], Timothy M. Shea[2], Andreea Danielescu[2], David C. Noelle[1], Christopher T. Kello[1]

[1]University of California, Merced, [2]Accenture Labs

Inception Nucleus

- We proposed a novel Inception block to tune kernel sizes on the fly during training!



Network Architecture

The proposed Inception Nucleus architectures with/without Batch Normalization:

Inception Nucleus Nets Configurations			
Inception	Inception-FA	Inception-FI	Inception-BN
289 K	789 K	479 K	292 K
Input (32000 × 1)			
Conv1D,32,80,4		Inception Nucleus: Conv1D,32,60,4 Conv1D,[32,80,4]×2 Conv1D,[32,100,4]×2	Conv1D,32,80,4 with BN
Inception Nucleus: Conv1D,64,4,4 Conv1D,[64,8,4]×2 Conv1D,[64,16,4]×2	Inception Nucleus: Conv1D,64,20,4 Conv1D,[64,40,4]×2 Conv1D,[64,60,4]×2	Inception Nucleus: Conv1D,64,4,4 Conv1D,[64,8,4]×2 Conv1D,[64,16,4]×2	Inception Nucleus: Conv1D,64,4,4 - BN Conv1D,[64,8,4]×2-BN Conv1D,[64,16,4]×2-BN
Max Pooling 1D, 64,10,1			
Reshape (put the channels first)			
Conv2D,32,3 × 3,1			Conv2D,32,3 × 3,-BN
Max Pooling 2D,32,2 × 2,2			
Conv2D,64,3 × 3,1 Conv2D,64,3 × 3,1			Conv2D,64,3 × 3,1-BN Conv2D,64,3 × 3,1-BN
Max Pooling 2D,64,2 × 2,2			
Conv2D,128,3 × 3,1			Conv2D,128,3 × 3,1-BN
Max Pooling 2D,128,2 × 2,2			
Conv2D,10,1 × 1,1			Conv2D,10,1 × 1,1-BN
Global Average Pooling			
Softmax			

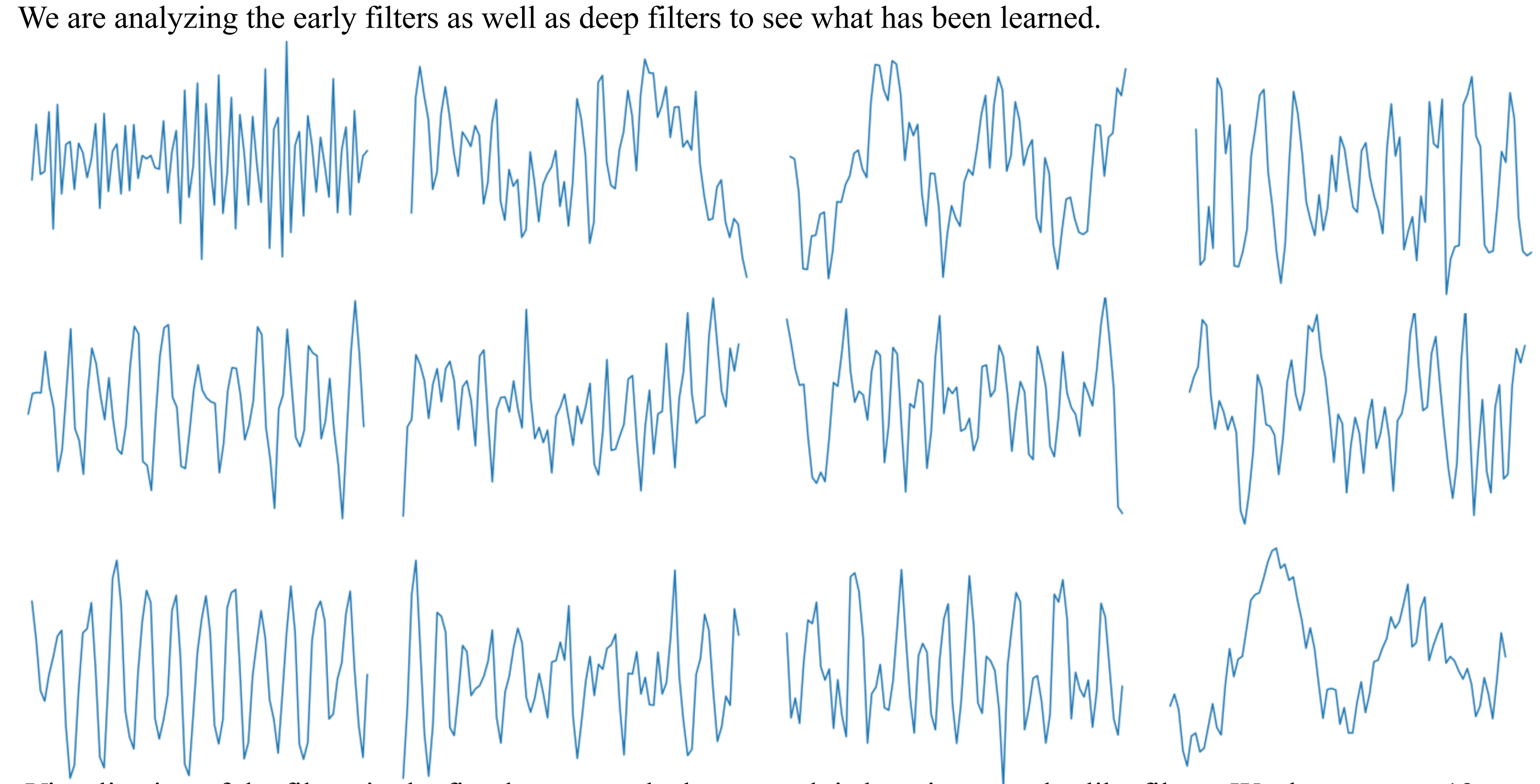
Results on Urbansound 8k Dataset

Comparing our model with the state-of-the-art approaches:

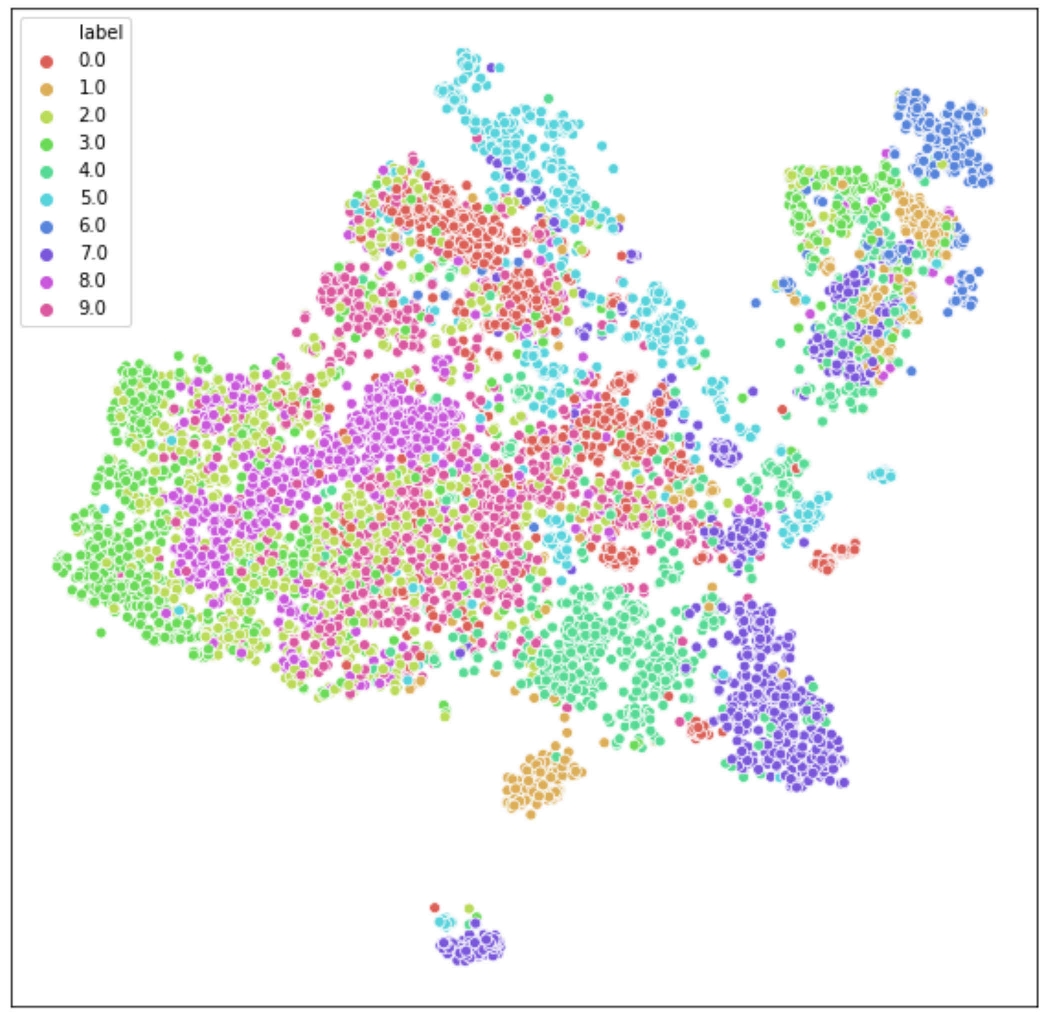
Model	Test	# Parameters
M3-fc [9]	46.82%	129M
M5-fc [9]	62.76%	18M
M11-fc [9]	68.29%	1.8M
M18-fc [9]	64.93%	8.7M
M18-fc [9]	64.93%	8.7M
RCNN [19]	71.68%	3.7M
ACLNet [11]	65.32%	2M
EnvNet-v2 [20]	78%	101M
PiczakCNN [21]	73%	26M
VGG [22]	70%	77M
Inception Nucleus-BN (Ours)	**83.2%**	**292K**
Inception Nucleus-FA (Ours)	**70.9%**	**789K**
Inception Nucleus-FI (Ours)	**75.3%**	**479K**
Inception Nucleus (Ours)	**88.4%**	**289K**

Filter Visualization

We are analyzing the early filters as well as deep filters to see what has been learned.

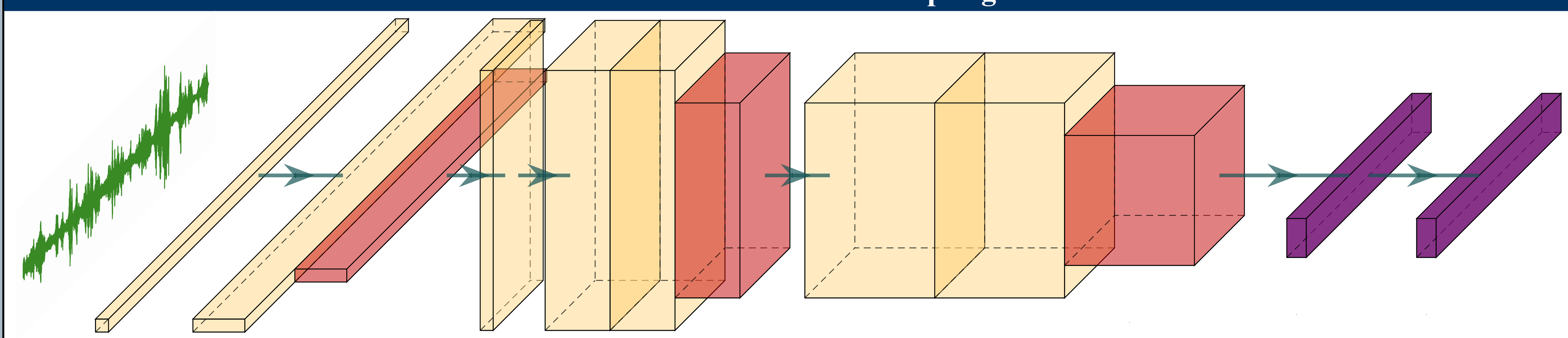


Visualization of the filters in the first layer reveals the network is learning wavelet-like filters. We demonstrate 12 random filters here.



We applied t-SNE on the most abstract features. It exhibits the network has learned semantically meaningful concepts.

Adapting Network Architecture For Loihi Hardware

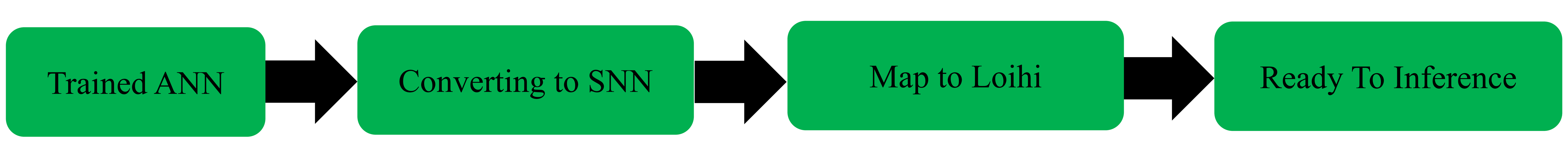


Translated network architecture. We Train the network using Gpus and then translating the learned network to spiking neural network and we will port it on the Loihi chip.

Conversion to Loihi involved 4 key network architecture adaptations:

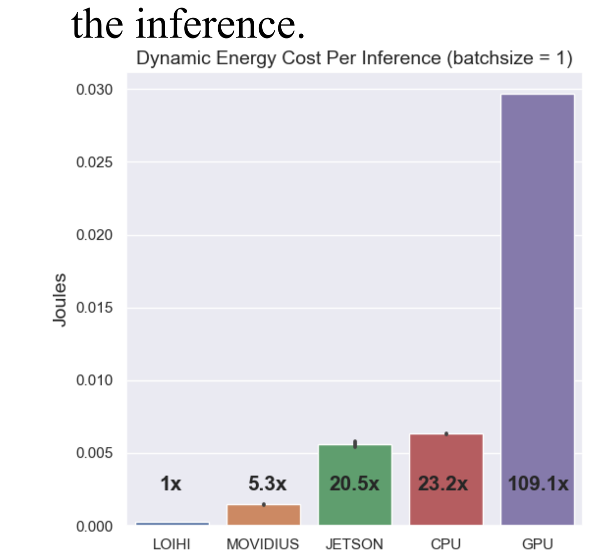
- Replace Max pooling to Average Pooling
- Reduce the input dimension from 32K to 8K
- Replacing the global average pooling to flatting and a dense layer
- Reducing total parameters from 292K to 143K

Port to Loihi



Results

A typical titan GPU needs nearly 110x more energy than a Loihi for the inference.



Mean power consumption and energy cost per inference across hardware devices.

Hardware	Idle (W)	Running (W)	Dynamic (W)	Inf/Sec	Joules/Inf
GPU	14.97	37.83	22.86	770.39	0.0298
CPU	17.01	28.48	11.47	1813.63	0.0063
Jetson	2.64	4.98	2.34	419	0.0056
Movidius	0.210	0.647	0.437	300	0.0015
Loihi	0.029	0.110	0.081	296	0.00027

Summary:

- We proposed a novel end-to-end architecture that takes a raw waveform input and maps it to labels without any feature extraction.
- We analyzed the learned filters and we noticed that the network in the very beginning is learning wavelet-like filters and deeper representations are semantically meaningful.
- We translate the network on the Loihi neuromorphic chip with some modifications.
- The results suggest that Loihi chips are very efficient in power since they are nearly 110x more efficient than GPUs on the inference.