Deep Neural Networks for Radar Waveform Classification

Michael Wharton (AFRL/OSU), Anne Pavy (AFRL), Philip Schniter (OSU)

The Ohio State University — Air Force Research Laboratory Sensors Directorate

2021 Asilomar Conference on Signals, Systems, and Computers

Paper ID: 1548

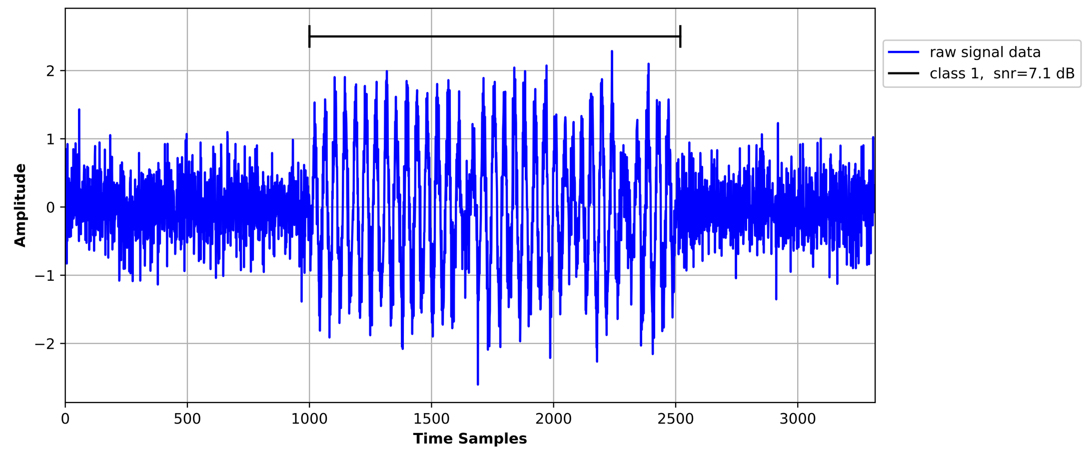
Abstract

We consider the problem of classifying phase-modulated radar pulses given raw I/Q waveforms in the presence of noise and the absence of synchronization. We also consider the problem of classifying multiple superimposed radar pulses. To tackle these problems, we design deep neural networks (DNNs) that yield more than 100x reduction in error-rate over the current state-of-the-art.

Radar Waveform Classification

Goal: Classify one or more radar waveforms that are present in a time-domain signal.

Application: Important task for cognitive radars



High-SNR radar waveform, centered around white noise.

Specifics:

1. We consider a passive sensing scenario
   - Waveforms will be subject to unknown time delays (i.e., asynchronous) and carrier shifts
2. We expect SNRs well below 0 dB
3. We have no physical model for the classes, only a dataset containing examples

Our Approach

Train a Deep Neural Network (DNN) using raw time-domain samples

Dataset:

SIDLE dataset from AFRL

- 23 classes of phase-modulated radar waveforms
- 10 000 waveforms per class
- SNRs $\in [-12, +12]$ dB
- Pulse widths $\in$ [181, 8925] time samples

Existing Work [1]:

Designed a Convolutional Neural Network (CNN) using time-domain samples

- 5 convolutional and 4 dense layers
- Only processes real part of waveform (discards imaginary)
- Noise pads each radar waveform with white Gaussian noise, up to 11 000 time samples

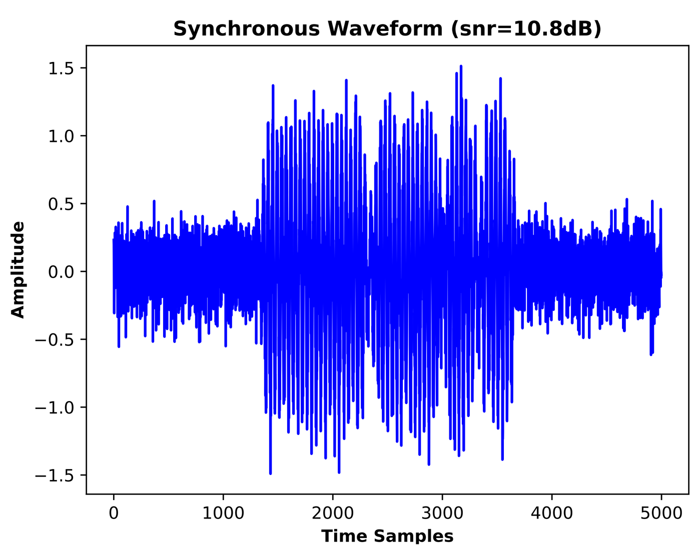
- Considers only *single-waveform* classification
- Assumes waveforms are *synchronous*

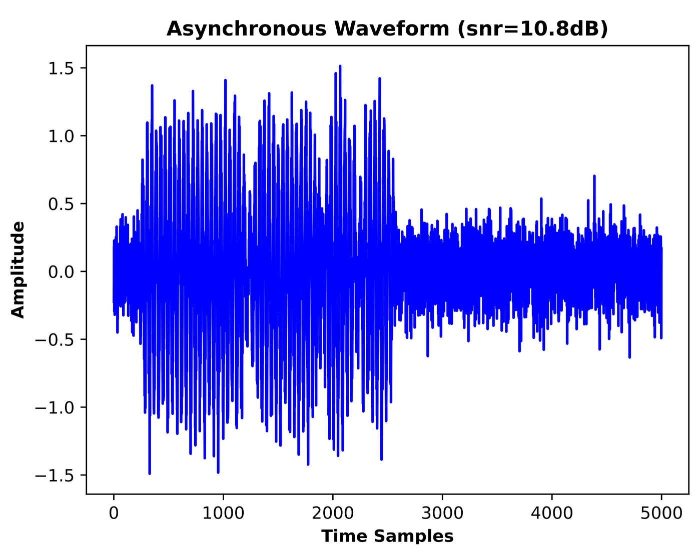
Synchronous performance [1]

- Test error: 3.6%
- Train error: 0% (overfitting)

Asynchronous performance

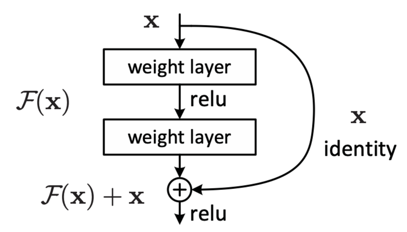
- Test error: 18.4%
- Train error: 18.2%



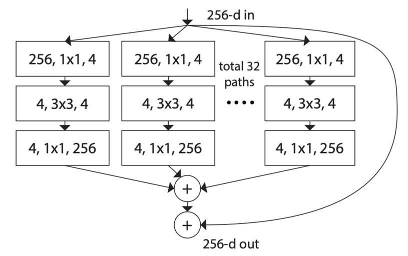


DNN Architectures Evaluated

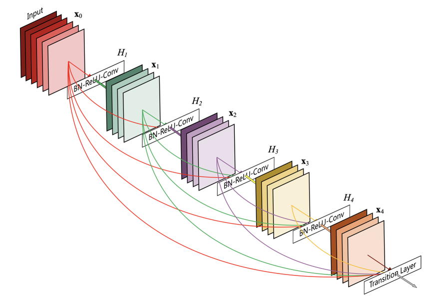
ResNet [2]



ResNeXt[3]



DenseNet[4]



Experimental Setup

- We used asynchronous SIDLE waveforms
- Noise padded to 11 000 samples
- Input I/Q samples to DNNs

Results

- ResNet outperformed other architectures

DNN Architecture	Test Error
ResNet	**1.6 %**
ResNeXt	10.5 %
DenseNet	2.8 %

ResNets for Asynchronous Waveforms

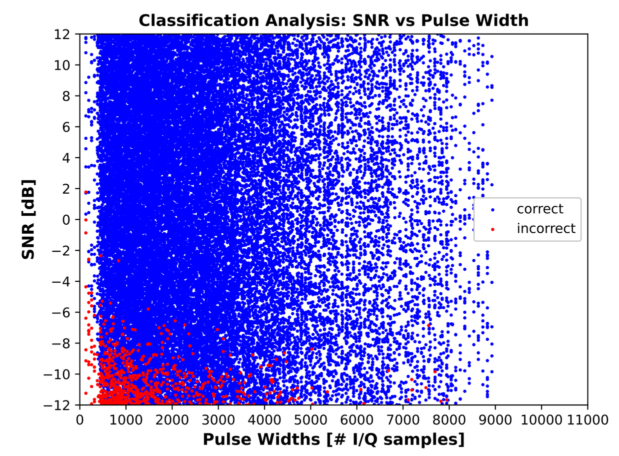
Experimental Setup

- Noise padded to 11 000 samples
- Input real channel only (baseline approach)

Results

- ResNet: 2.1% error
- Baseline: 18.4% error

- Plot shows SNR *before* noise padding
- Noise padding reduces effective SNR



30-layer ResNet results

Optimizing the Input Dimension

- We have only considered an input dimension of $D = 11\,000$ time samples
- To handle *arbitrary* values of D, we must truncate or noise-pad waveforms as needed
  - Smaller D: reduced noise padding will improve effective SNR
  - Smaller D: long-pulse truncation will discard information

Results

Input Dimension	11 000	6040	**3317**	1178	1000
Test Error	2.1%	1.4%	**1.3%**	2.2%	8.5%

- Among the tested values of D, we found 3317 to be best
  - Note: smaller D also speeds up training/processing

Complex-Valued Deep Networks

There are two approaches to linearly processing a complex-valued feature, $x = x_r + ix_i \in \mathbb{C}$:

Approach 1: 2-channel, real-valued DNN

$y_1 = k_{11}x_r + k_{12}x_i \in \mathbb{R}$

$y_2 = k_{21}x_r + k_{22}x_i \in \mathbb{R}$

Four learnable parameters:

- $k_{11}, k_{12}, k_{21}, k_{22} \in \mathbb{R}$

Approach 2: 1-channel, complex-valued DNN

$kx = (k_r x_r - k_i x_i) + \mathrm{j}(k_i x_r + k_r x_i) \in \mathbb{C}$

Only two learnable parameters!

- $k_r, k_i \in \mathbb{R}$

Experiment Setup

- 30-layer ResNet
- $D = 3317$

Real-valued vs. Complex-valued DNNs

Data	Operations	Test Error
In-phase	Real	1.52%
Complex	Real	0.39%
Complex	**Complex**	**0.36%**

Fine-Tuning

Fine-Tuning Complex-ResNet Parameters

- # layers (network depth)
- # of channels (network width)
- kernel size in convolutional layers
- Batch size
- Learning rate

Results of Network Fine Tuning

# Layers	Test Error	# Parameters	# Channels	Kernel
22	0.16%	7 721 041	32	11
26	0.16%	1 818 161	16	7
30	**0.14%**	**659 233**	**8**	**9**
34	0.15%	670 945	8	9
38	0.16%	2 228 913	16	7

Multi-label DNNs for Multi-Waveform Classification

Motivation

- The electromagnetic spectrum is very crowded!
  - Often there are multiple radar transmitting simultaneously
- The # of waveforms present in the signal will be unknown

Our Approach

- Minimize the sum of K binary-cross-entropy (BCE) losses
  - No assumption on the number of waveforms present
  - Network outputs "present" or "absent" for each class
- Re-train the fine-tuned Complex-ResNet-30 with this BCE loss

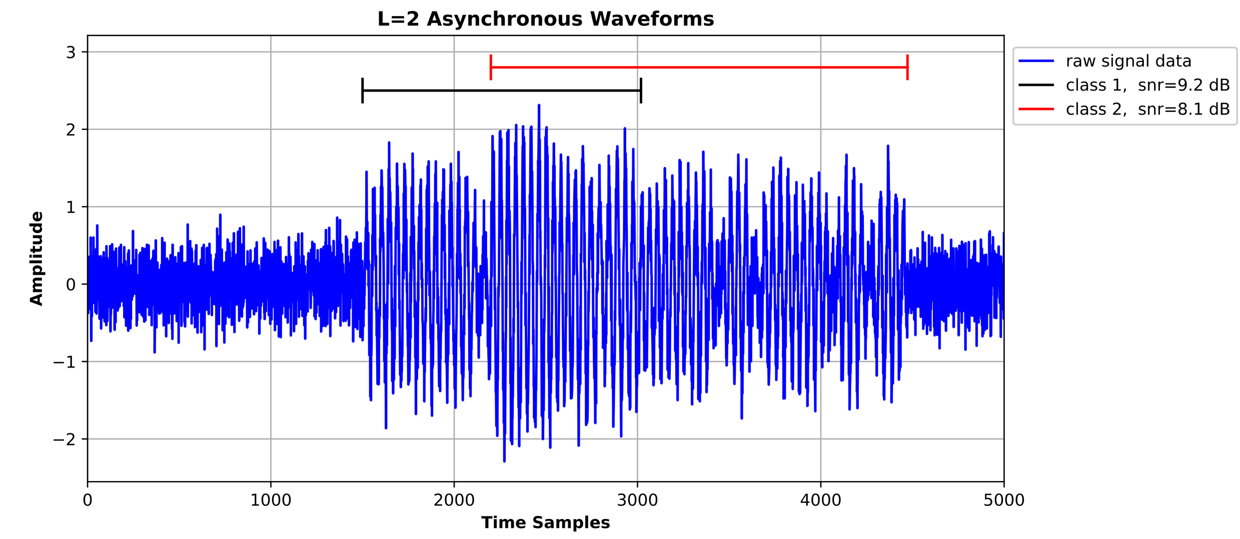
Metrics

- Absolute error: Error averaged over the K binary predictions
- Subset error: Error on the prediction vector $\in \{0,1\}^K$

$$E_{\text{sub}} \approx K E_{\text{abs}} \quad \text{for i.i.d. binary errors}$$

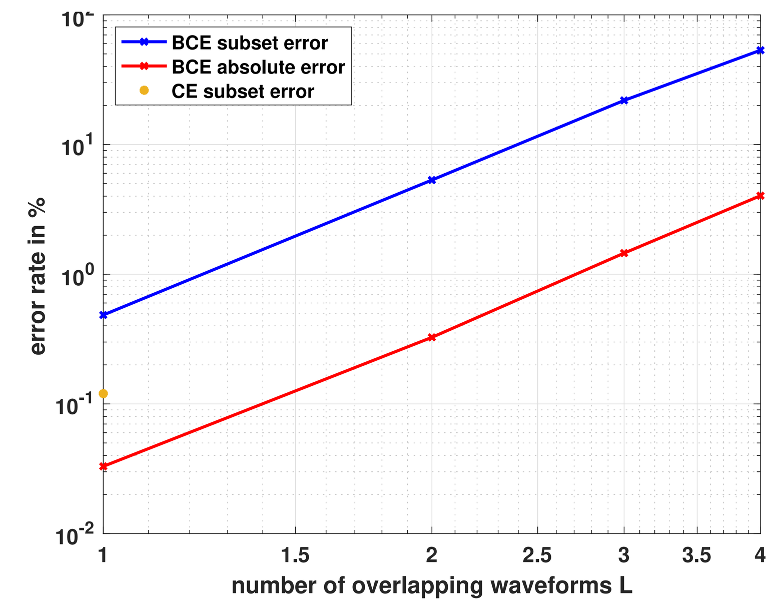
Simulating L-label waveforms:

1. Sample L uniformly from $\{1,2,3,4\}$
2. Generate one asynchronous, noise-padded waveform (as before)
3. Generate $L-1$ asynchronous, zero-padded waveforms
4. Sum all waveforms



Multi-Waveform Results

Error rates vs. Number of Overlapping Waveforms



- Errors grow linearly with $\log L$
- $L = 4$ absolute error only 4.0%
- $L = 1$ BCE subset error > CE subset error
  - CE-trained network was optimized for this case

Conclusion and Contributions

Single-Waveform Classification

- We improved the state-of-the-art error rate from 18.4% to 0.14% on asynchronous SIDLE data
  - Residual networks and optimizing the input dimension
  - Complex-valued operations & fine-tuning network parameters

Multi-Waveform Classification

- We trained a DNN to simultaneously classify up to 4 waveforms
  - Absolute error rate of only 4.0% in the case of 4 overlapping waveforms

Future Work

- Train a deep network to classify *and* localize each overlapping waveform
  - Object detection using a 1D version of YOLO
- Handle multiple radars operating in different frequency bands

References

1. R. Chakravarthy, H. Liu, and A. Pavy, "Open-set radar waveform classification: Comparison of different features and classifiers," *IEEE Radar Conf.*, pp. 542-547, 2020.
2. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE CVPR*, Jun 2016.
3. S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *IEEE CVPR*, July 2017.
4. G. Huang, Z. Lie, and L. van der Maaten, "Densely connected convolutional networks," *IEEE CVPR*, July 2017.