A HYBRID NEURAL NETWORK FRAMEWORK AND APPLICATION TO RADAR AUTOMATIC TARGET RECOGNITION

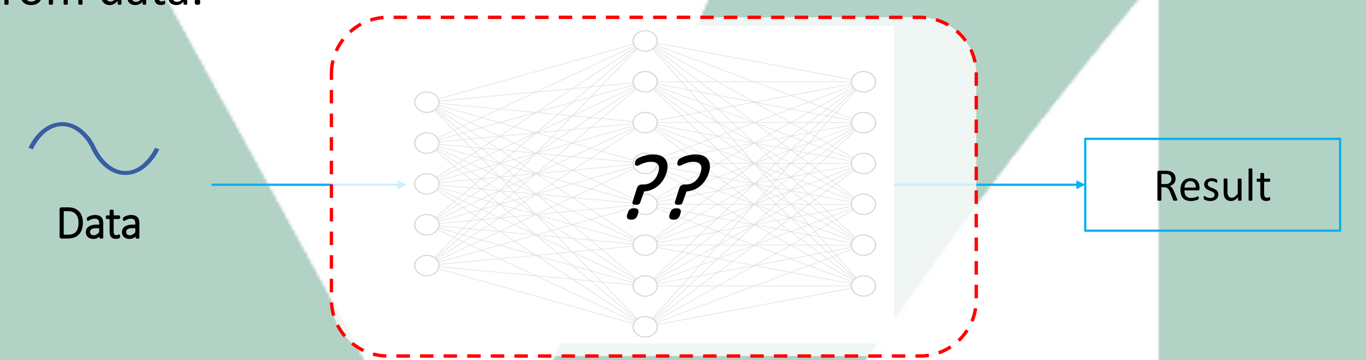
Zhe Zhang, Xiang Chen, Zhi Tian | George Mason University | {zzhang18, xchen26, ztian1}@gmu.edu

ABSTRACT

Deep neural networks (DNNs) have found applications in diverse signal processing (SP) problems. This paper presents a novel ***hybrid-NN framework*** in which one or more *SP layers* are inserted into the DNN architecture in a coherent manner to enhance the network capability and efficiency in feature extraction. These SP layers are properly designed to make good use of the available models and properties of the data. The network training algorithm of hybrid-NN is designed to actively involve the SP layers in the learning goal, by simultaneously optimizing both the weights of the DNN and the unknown tuning parameters of the SP operators. Hybrid-NN is tested on a radar automatic target recognition (ATR) problem. Compared with ordinary DNN, hybrid-NN can markedly reduce the required amount of training data and improve the learning performance.

INTRODUCTION

As a data-driven framework, traditional DNN treats the learning problem as a ``black-box'' that extracts useful features directly from data.

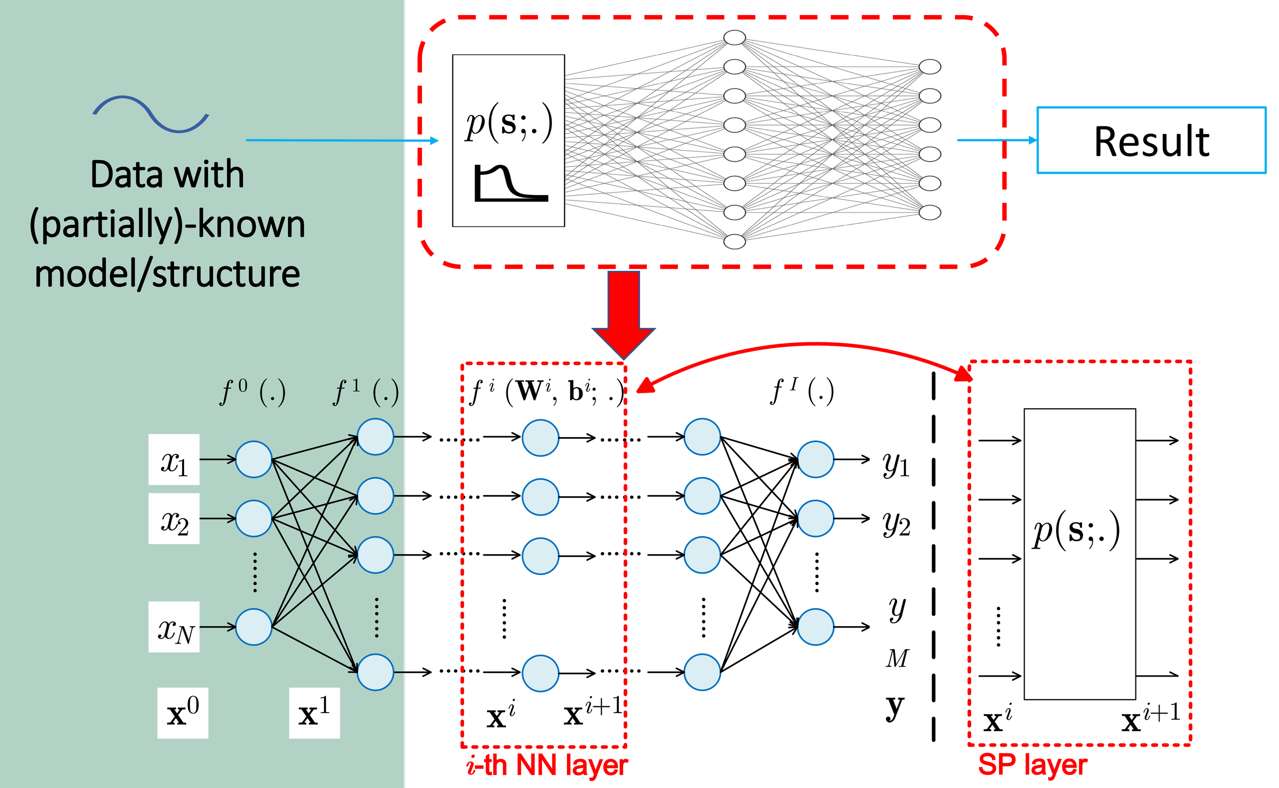


- **DNN**: does not rely on any special structure or property of the processed data
  - ✓ Universally applicable to diverse problem models;
  - × Huge amount of labeled data required;
  - × Long training time.
- **SP methods**: crafted to gainfully utilize such prior knowledge
  - ✓ Sample efficient;
  - × Only works for some known models or structures.

DNN	Real World Problems	Signal Processing
Data driven	?? In between ??	Model driven
Black-box, model free	Gray-box, Partially-known data model or structure	Clear-box, pre-known model
Universal	Can be specific	Specific
Big data	Some data, but not adequate	Small data

HYBRID NEURAL NETWORK (HYBRID-NN)

- **Motivation: combining SP operators and DNN so as to benefit from both DNN and SP.**
- ***Hybrid neural network (hybrid-NN):***
  - One or more properly-selected SP operators are inserted into the DNN architecture as embedded layers;
  - Key design parameters of each SP operator are treated as unknowns and updated during network training from data.
    - $p(\mathrm{s};\ .)$: signal processing operator;
    - s: (unknown) SP parameters;
    - $f^{\,i}$: activation function of the i-th layer.



ALGORITHM

- Based on back-propagation (BP): compute the training error of each layer based on the error of next layer.
- The output error of SP layer propagated from previous layer:

$$\underbrace{\delta^i}_{\text{Error of SP layer}} = \underbrace{\delta^{i+1}}_{\text{Error from next layer}}\ \underbrace{f^{i}}_{\text{Activation function of SP layer}}{}'\Big(\underbrace{p}_{\text{SP operator}}\big(\underbrace{\mathbf{s}}_{\text{Parameters of SP operator (weights)}};\ \underbrace{\mathbf{x}^{i-1}}_{\text{Inputs of SP layer}}\big)\Big),$$

- The (sub-)gradients of SP operators are utilized to compute the training error of each SP layer:

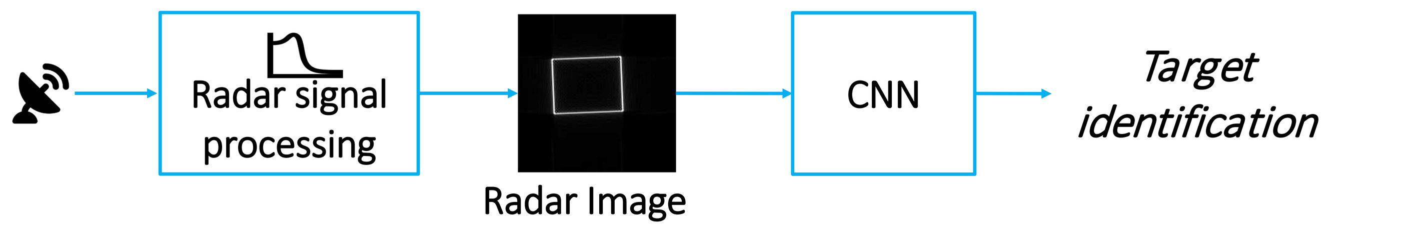
$$\underbrace{\Delta\mathbf{s}}_{\text{Update value of SP parameter}} = -\underbrace{\lambda}_{\text{Learning rate}}\underbrace{\left(\frac{\partial p}{\partial \mathbf{s}}\right)}_{\text{(Sub-)gradient of SP operator}}\delta^i,\quad \delta^{i-1} = \left(\frac{\partial p}{\partial \mathbf{x}^{i-1}}\right)\delta^i.$$

BENEFITS

- Better accuracy.
- Same computational architecture as NN; faster convergence .
- Autonomously resolved the unknown parameters in SP layer.

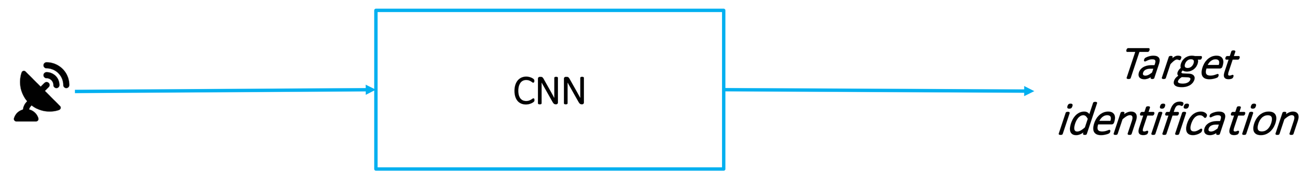
EXAMPLE: RADAR AUTOMATIC TARGET RECOGNITION (ATR)

- **Traditional ATR**:



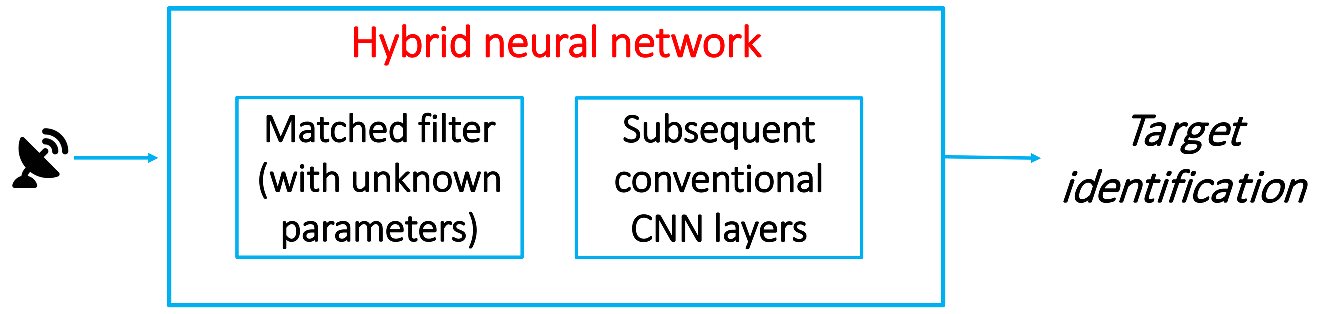
☐ *Not applicable with unknown radar parameters (e.g. passive radar, error)*

- **DNN based ATR**:



☐ *Not applicable when training data amount is limited;*
☐ *Failed to utilize the known structural information of radar echoes.*

- **Hybrid-NN based ATR**:

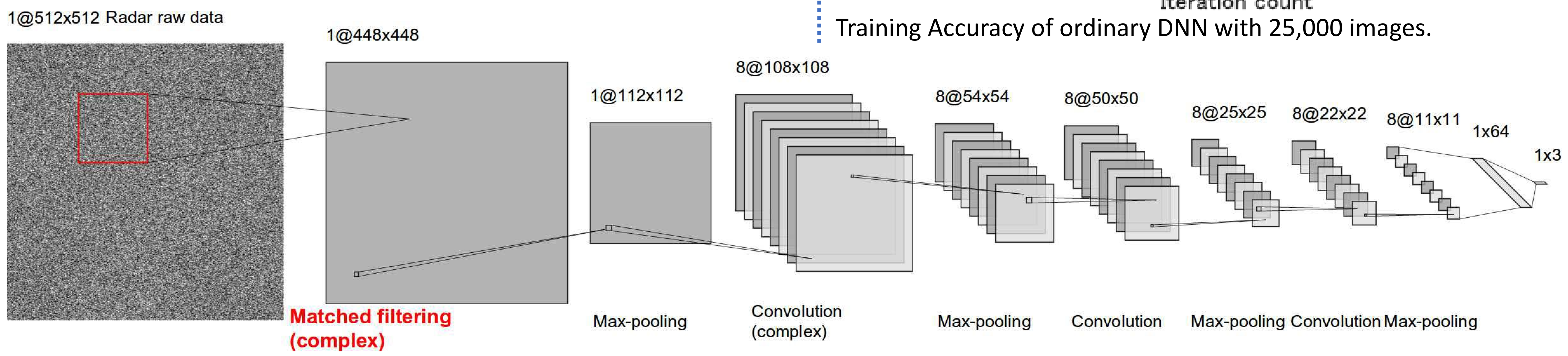


HYBRID-NN DESIGN

- Adopt conventional convolutional neural network (CNN) ;
- Replace its first convolutional layer with the **radar match filtering (MF) layer:**
  - Radar waveform is Chirp, we use a corresponding matched filter to achieve the best utilization of the known model.
  - $\hat{K}_a, \hat{K}_r$: frequency modulation rates in azimuth and range directions, needed to be learned from data.
  - M: matched filter matrix generated from K_a and K_r.

$$g(\hat{K}_a,\hat{K}_r;\mathbf{S}) = f\left(\mathrm{conv}\{\mathbf{M}(\hat{K}_a,\hat{K}_r),\mathbf{S}\}\right) = \left|\mathrm{conv}\{\mathbf{M}(\hat{K}_a,\hat{K}_r),\mathbf{S}\}\right|,$$

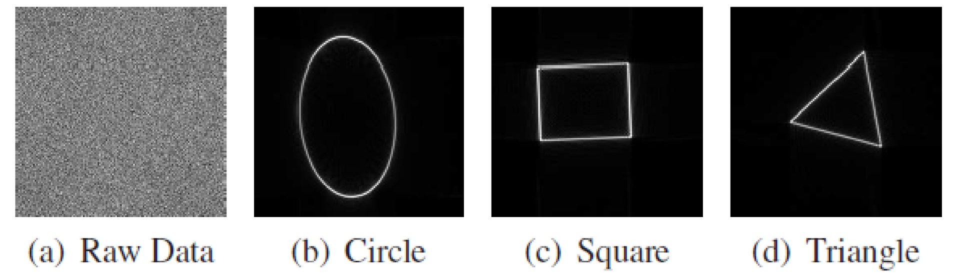
$$\left[\mathbf{M}(\hat{K}_a,\hat{K}_r)\right]_{mn} = e^{-j\pi\hat{K}_r m'^2}e^{j\pi\hat{K}_a n'^2}.$$

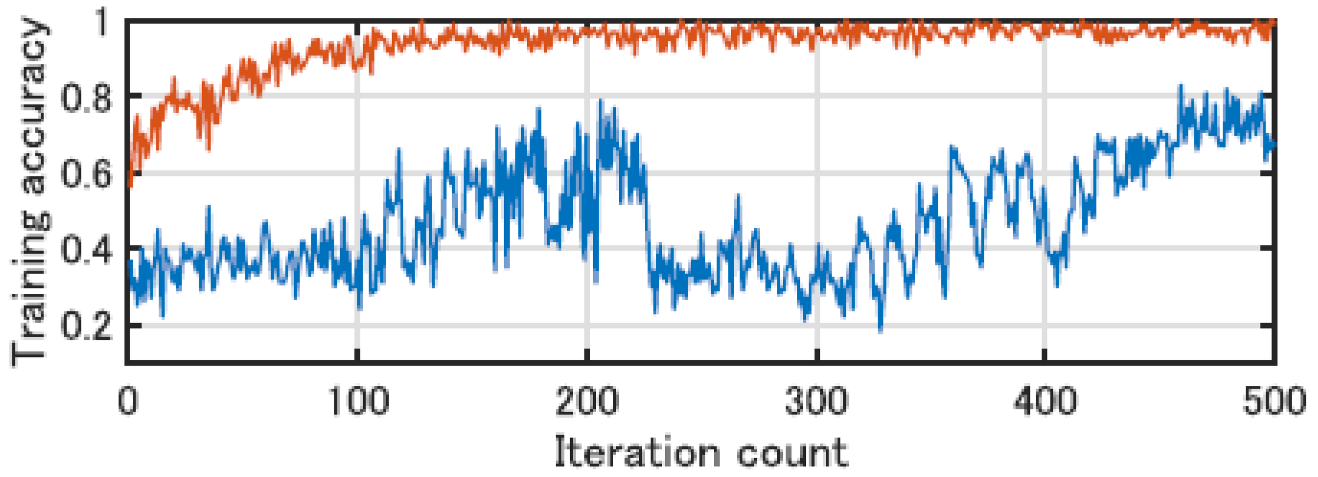


SIMULATION RESULT

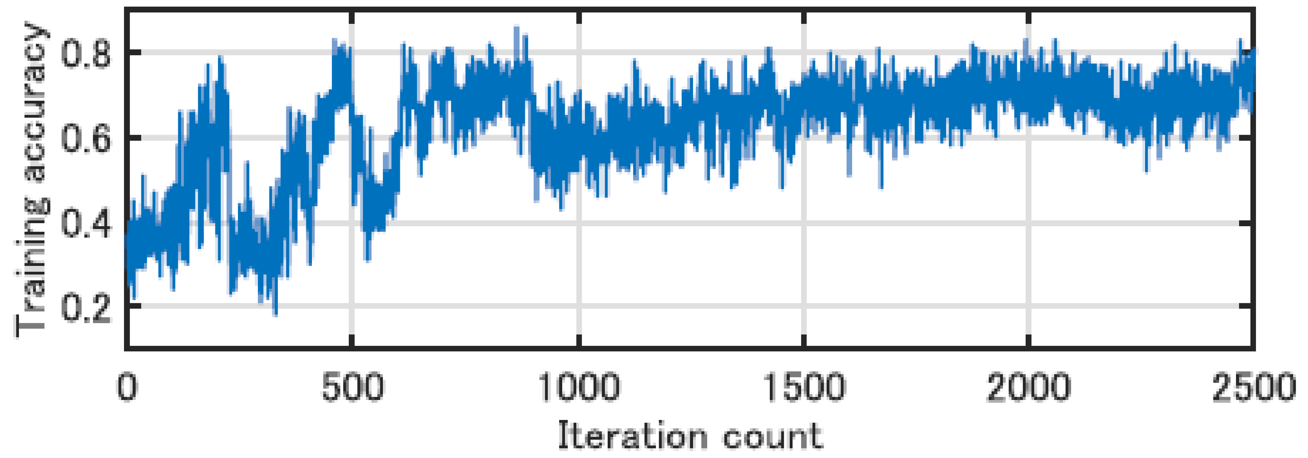
- 1 SP layer, 3 convolutional layers, 1 FC layer;
- 5,000 images training set (hybrid-NN, ordinary DNN), 25,000 images training set (ordinary DNN);
- Simulated training images.

Carrier frequency	5GHz
Range sampling rate	600MHz
Range bandwidth	500MHz
Range distance	5,000m
Target speed	100m/s
PRF	1,000Hz





Training Accuracy of Hybrid-NN and ordinary DNN with 5,000 images.



Training Accuracy of ordinary DNN with 25,000 images.