Exploration of Deep Learning in Physical Layer Design

Bo Yuan
Rutgers University

May 20, 2019, Piscataway

RUTGERS

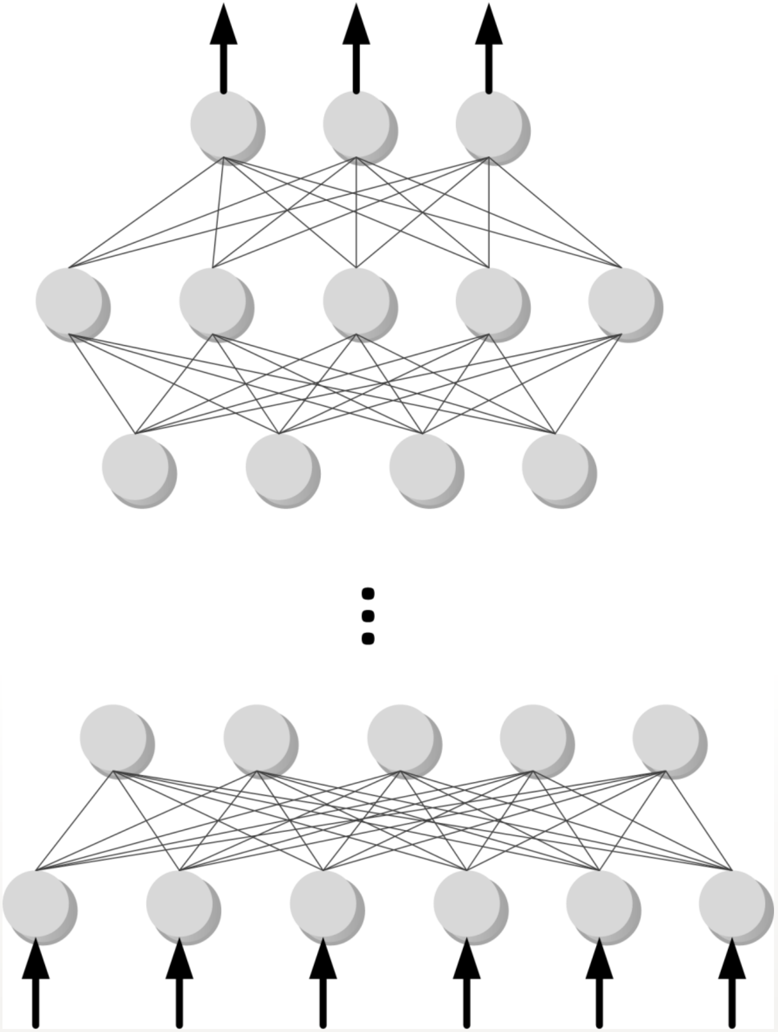
1766

1980's
Prosperous

2006
Resurgence

Deep Learning (DL)

- Popular AI technique
- Essentially neural network
- "Deep" & "Wide" (*DNN*)
- Powerful capability

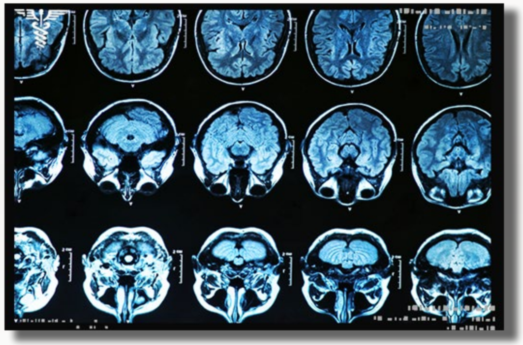


AI: Artificial Intelligence
DNN: Deep Neural Network

Deep Learning - Everywhere

AlphaGo

Siri

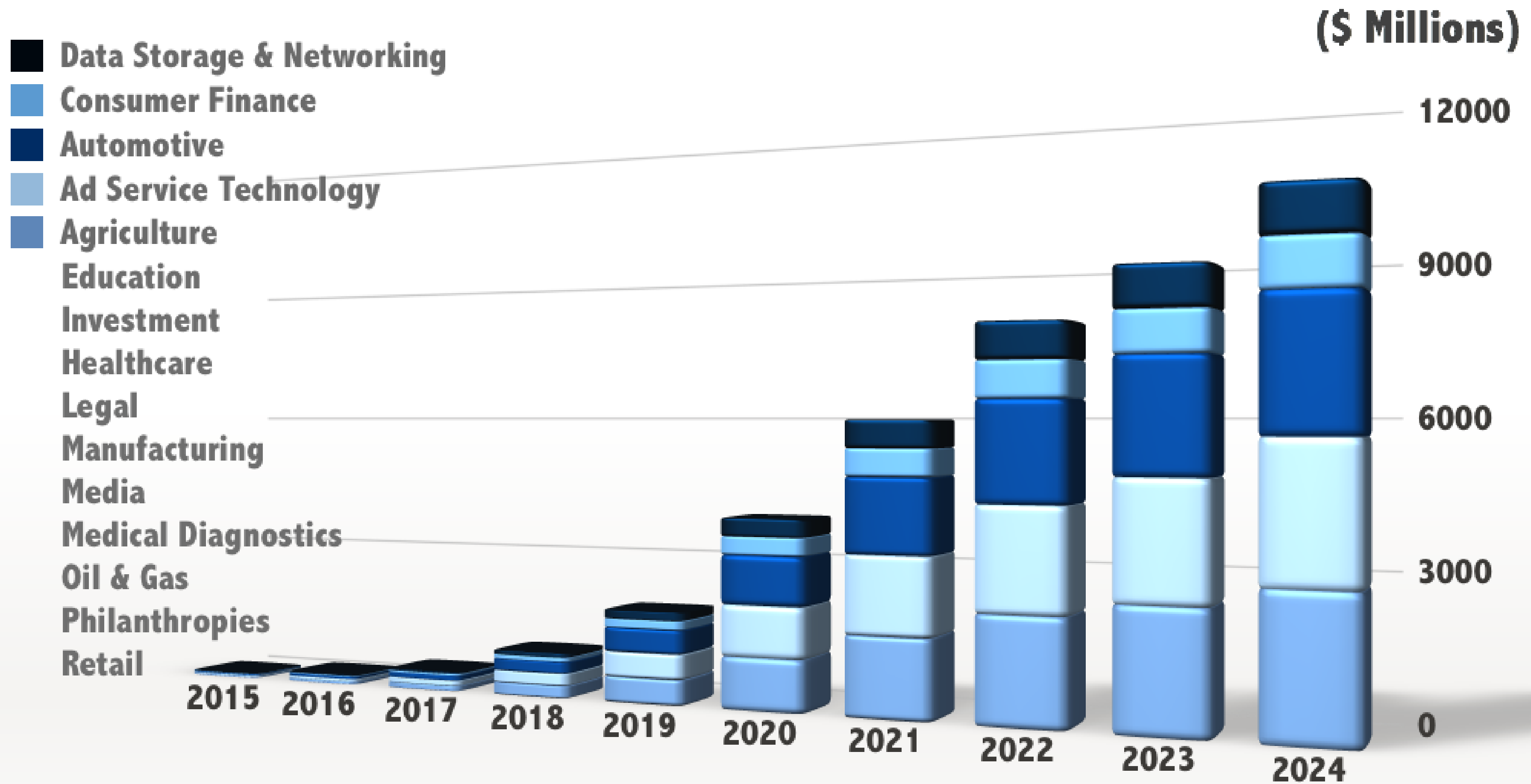


Medical Diagnosis

In courtesy of www.bigstockphoto.com

Opportunities

Deep learning software revenue



Source: Tractica

Deep Learning for Comm./Network

- **Very active** research on DL for networking
  -- Motivation: Big data, hard to model

- **Early stage** of DL for PHY in wireless comm.
  -- Machine learning (ML) is not new for PHY
  -- Existing work on DL for PHY
  decoder, detector, estimator, equalizer

Why DL for PHY

- Model-free solution
  -- Sometimes it is hard to model channel
- May improve BER performance
  -- DL works when heuristic factors exist
- Potential for on-line learning
  -- Provide flexibility and reconfigurability
- High Parallelism, avoid serial computing
  -- Successive cancellation for polar decoder
- Hardware-friendly computation
  -- Matrix Multiplication, no matrix inversion

Risks

- **Currently non-ML approach is good enough**

  -- PHY is a field with solid math. Foundation

  -- Very good codes (LDPC, polar) exist

- **Overhead of using neural network?**

  -- Unlike networking, PHY is *extremely* sensitive to latency, power, area…

Challenge of Deploying DNN

Storage intensive

Thang Luong @lmthang Follow

A new era of NLP has just begun a few days ago: large pretraining models (Transformer 24 layers, 1024 dim, 16 heads) + massive compute is all you need. BERT from @GoogleAI: SOTA results on everything arxiv.org/abs/1810.04805. Results on SQuAD are just mind-blowing. Fun time ahead!

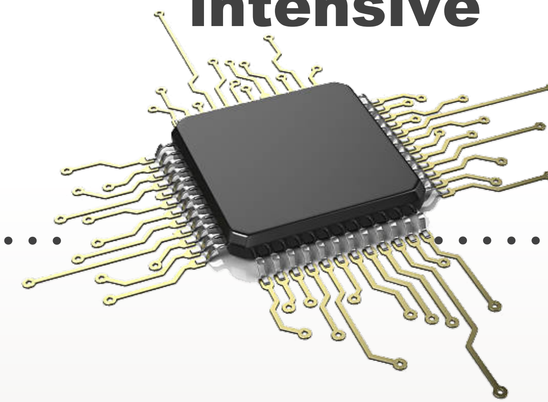
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

393M Parameters

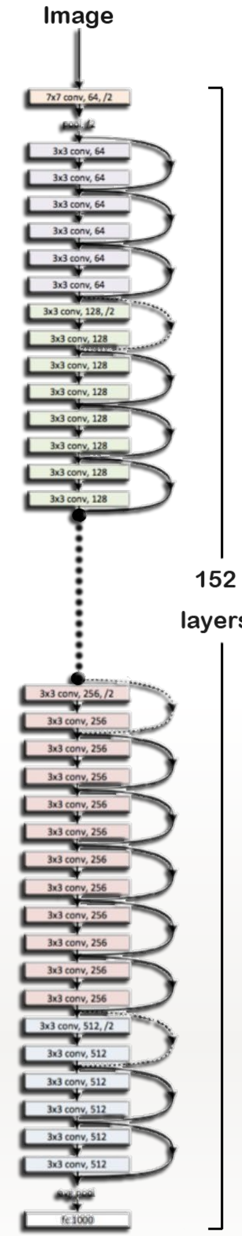
BERT from Google

Computation intensive



11.3G MAC

Resnet-152 from Microsoft



Our Viewpoint of DL for PHY

- **Algorithm-hardware co-design**
  -- Compressing neural network (NN) model
  -- Hardware-aware DL for PHY

- **Utilizing our prior experience in DL and PHY**
  -- Pioneering work on polar decoder
  (ICC’12, ICASSP’13, TCAS-I’14, TSP’14…)
  -- Recent work on both DL algo. and HW
  (ICML’17, MICRO’17/18, AAAI’18/19, ISCA’19)

Two Paths of DL for PHY

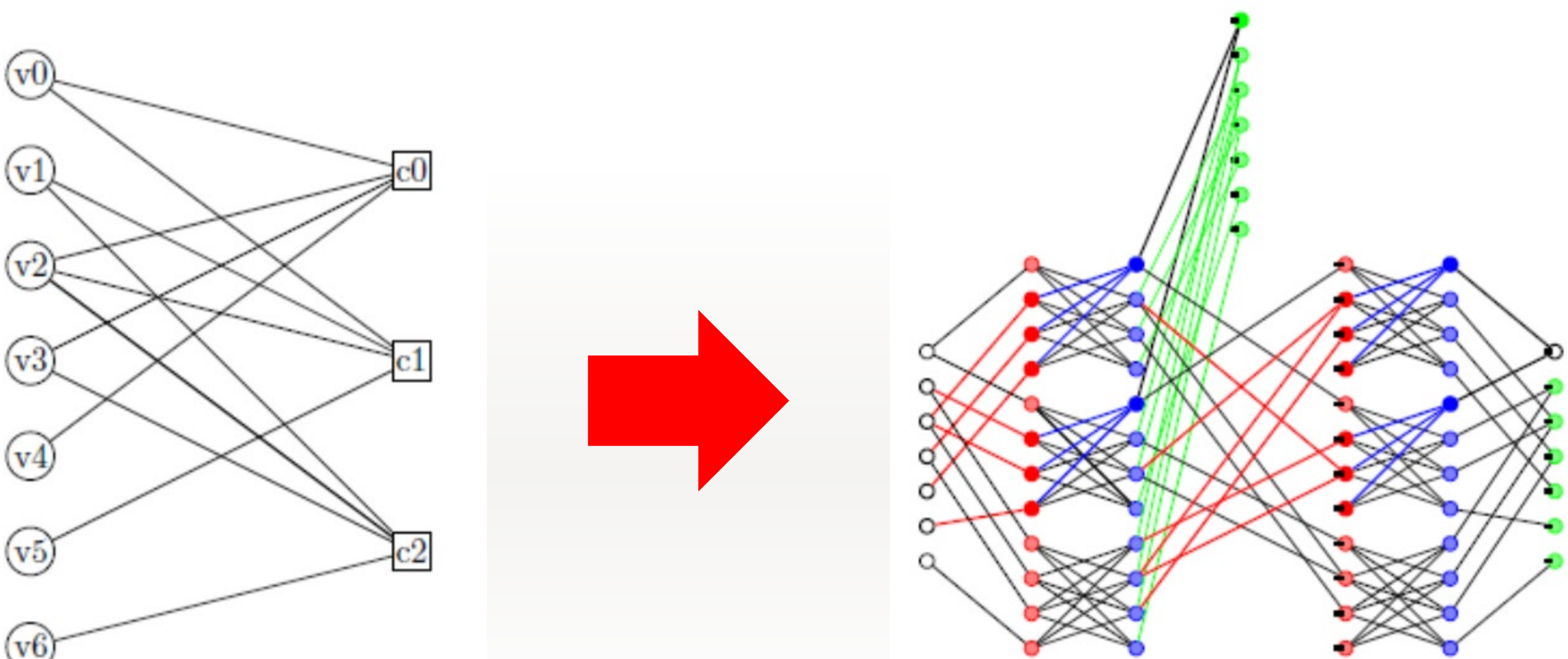
- **DL-aided solution**
  -- Reformulate existing approach to NN
  -- Underlying algorithm is still non-NN
  -- Popular in channel coding
  -- Module-level

- **DL-enabled solution**
  -- End-to-end, may not use domain info.
  -- Inter-module level (e.g. NN for joint detector/decoder)

DL for Channel Decoder

- **Current belief-propagation (BP) decoder can be viewed as a *folded* NN**

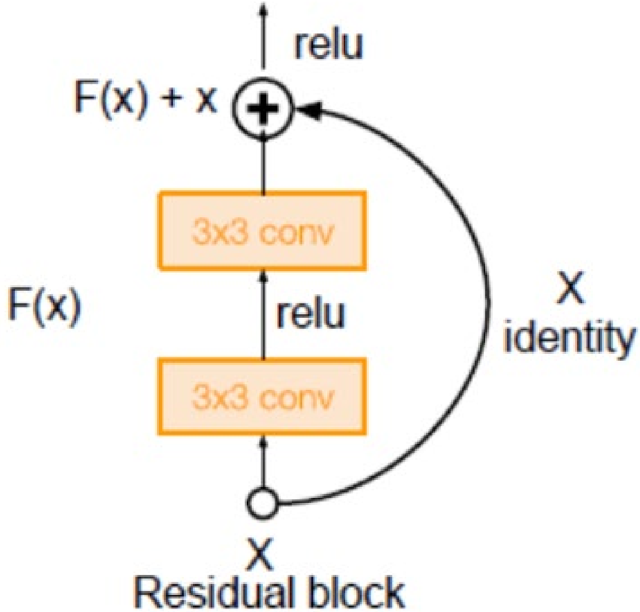
  -- Applicable to any linear codes (LDPC, BCH..)



N. Agrawal, M.S. Thesis 2017

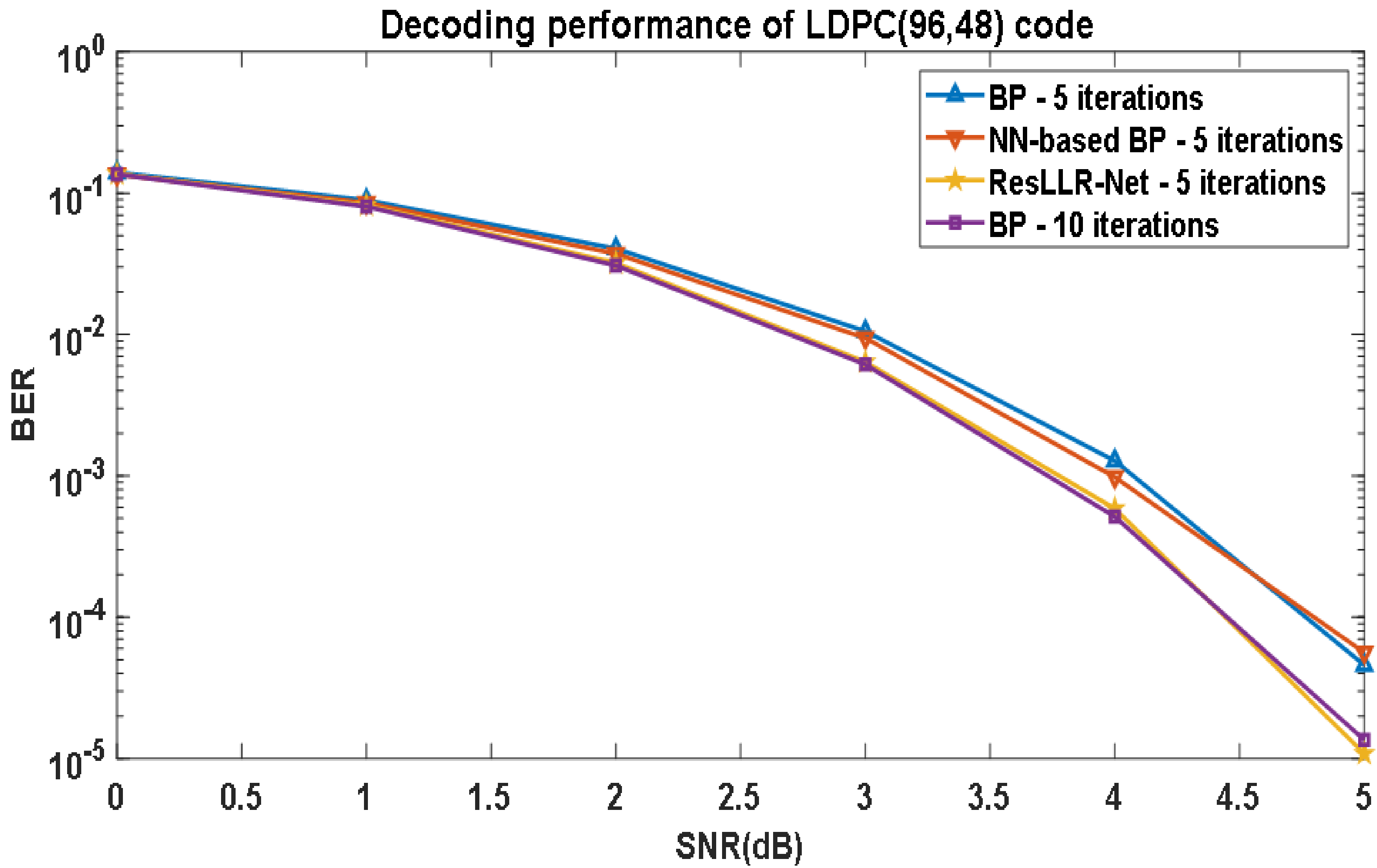
ResLLR-Net: Latency-aware NN-aided Decoder

- Inspired by the idea of residual block
  -- Unrolled NN-aided decoder can be deep
  -- Vanishing gradient problem exist
  -- Residual arch. can mitigate



Feifei Li, Stanford CS231n

Performance of ResLLR-Net

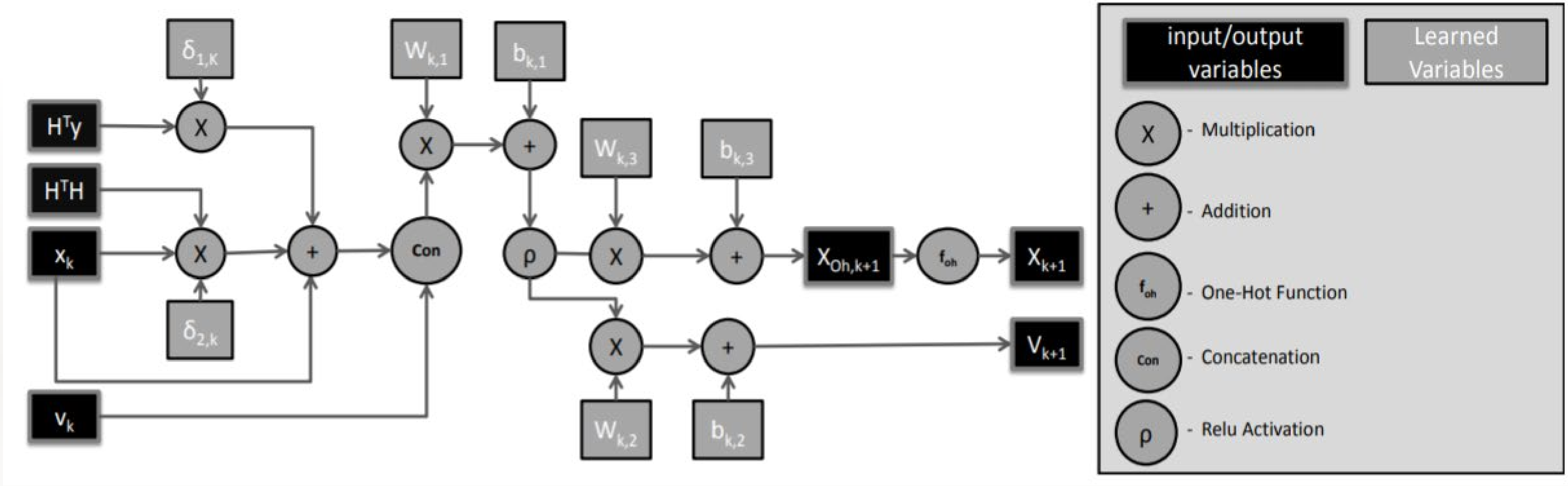


DetNet: NN-enabled MIMO Detector

- **MIMO detector**

$$\bar{\mathbf{y}} = \bar{\mathbf{H}}\bar{\mathbf{x}} + \bar{\mathbf{w}}$$

- **DetNet**

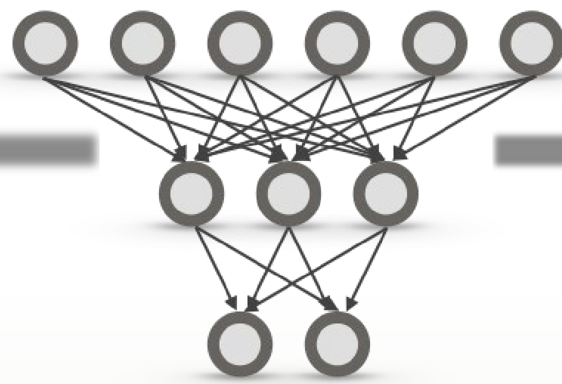


N. Samuel, Arxiv 2018

Compress NN Models using Circulant Matrix

$$\begin{bmatrix} +0.12 & +0.21 & -0.34 \\ +0.72 & -0.41 & +0.54 \\ +0.32 & +1.21 & +2.43 \\ +0.52 & +1.71 & -3.23 \\ -1.22 & +0.32 & +0.61 \\ +0.82 & -1.57 & +1.52 \end{bmatrix}$$

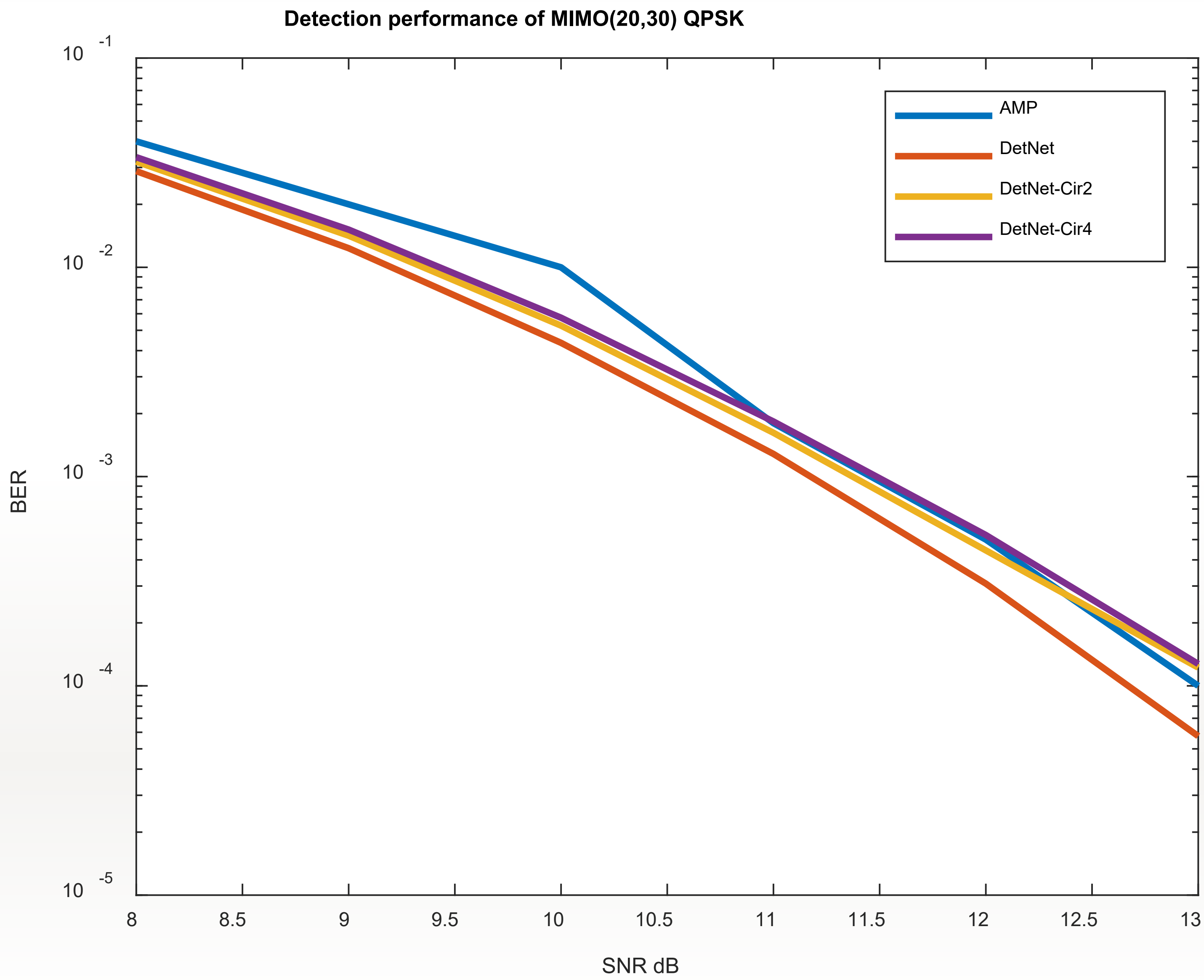
Unstructured Weight Matrix
(18 parameters)



$$\begin{bmatrix} +0.12 & +0.21 & -0.34 \\ -0.34 & +0.12 & +0.21 \\ +0.21 & -0.34 & +0.12 \\ \hline +0.52 & +1.71 & -3.23 \\ -3.23 & +0.52 & +1.71 \\ +1.71 & -3.23 & +0.52 \end{bmatrix}$$

Block-Circulant Weight Matrix
(6 parameters)

Performance Simulation



Summary

- DL for PHY is very emerging
- Interesting observation, potential huge impact

  -- If deployed, reshape landscape of PHY, especially modem chip design
- Overhead is a challenging problem
- Potential directions:

  -- Domain knowledge-based NN design

  -- Domain knowledge-based compression

  -- Cross-module NN design


Thanks!