MACHINE LEARNING FOR WIRELESS COMMUNICATIONS AND NETWORKING:
MOTIVATIONS, CASE STUDIES, AND OPEN PROBLEMS

IEEE Communications Society Kingston/CA & Denver Chapters

Shiwen Mao
Auburn University, Auburn, AL, USA
http://www.eng.auburn.edu/~szm0001/

Joint work with: Chao Yang, Ticao Zhang, and Xuyu Wang
In This Talk

- The evolution towards 6G
- Motivating examples of machine learning (ML) for Wireless
- Three case studies
- Challenges and thoughts
Evolution of Wireless Technology

Evolution of mobile phone communications


1G - TACS
First UK mobile phone call

2G - GSM/GPRS/EDGE

3G - WCDMA/HSPA/HSPA+
3G spectrum auction

4G - LTE/LTE Advanced
2.3 GHz & 3.4 GHz auction

5G
research & standardisation
commercialisation

\( G = \text{Generation} \)

- **Generation**: defined by new, fundamental, disruptive technology, a paradigm shift, for most devices (e.g., cellphones).
- 1G to 4G: each defined by its unique, key technology
- 5G is different:
  - Technology’s view: Full duplex, cognitive radio, SDN/NFV, non-orthogonal multiple access (NOMA), small cells/HetNet, massive MIMO, mmWave communications; **OR**: spectrum expansion, spectrum efficiency enhancement, network densification
  - Standardization’s perspective: Enhanced Mobile Broadband (eMBB), URLLC (Ultra Reliable Low Latency Communications), mMTC (massive Machine Type Communications)
- How about 6G?
  - Spectrum, Terahertz communications, light (VLC, FSO), blockchain, satellite, under water, VR/AR, ..., intelligence
  - **AI is disruptive**: ML vs. model based; but seems highly suitable for wireless systems, which are historically based on probabilistic models
Evolution of Hardware Platform

Machine Learning in the Market

Technology cycle - from PC, to smartphone, to artificial intelligence?

“Pure Play” Share Price Performance

Source: Bloomberg, Jefferies
Time is Right

- **Success in other fields**: natural language processing, image recognition, gaming, ...
- **Epic events (good advertisement)**:
  - IBM Deep Blue vs. World Chess Champion Garry Kasparov (1997)
  - IBM's DeepQA project: quiz show Jeopardy! against legendary champions Brad Rutter and Ken Jennings, and won the first place prize of $1 million (2011)
  - Google DeepMind's AlphaGo/AlphaGo Zero: beat Ke Jie, the world No.1 ranked Go player (2017)
  - Facebook/CMU’s Pluribus: beat 15 of the world’s top poker players (2019)
  - Dr. Fill, Champion of the 43rd Annual American Crossword Puzzle Tournament (2021)
- **Technology is ready**:
  - Availability of: Data, Computing, and open-source Platforms
  - Smartphones and GPUs: more powerful than the computer used for moon landing/space shuttles
  - Network size increasing, heterogeneous, and more complex: hard to model, hard to solve
  - Wireless designs: historically based on probabilistic models (e.g., traffic, channel, interference, ...), and are fault tolerant
AI, ML and Deep Learning

Image source: https://zhuanlan.zhihu.com/p/43435006
AI, ML and Deep Learning (Cont’d)

Image source: https://www.zhihu.com/question/57770020
Existing Work on ML for Communications

• Best Readings in Machine Learning in Communications
  • https://www.comsoc.org/publications/best-readings/machine-learning-communications

• Surveys
  • Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, “Application of machine learning in wireless networks: Key technologies and open issues,” *IEEE Communications Surveys and Tutorials*, vol.21, no.4, pp.3072-3108, Fourth Quarter 2019
  • ...

• Problems that have been explored
  • Signal detection
  • Channel encoding and decoding
  • Channel estimation, prediction, and compression
  • End-to-end communications
  • Resource allocation
  • Selected topics: localization, signal classification, full duplex, etc.
  • ....
In This Talk

• The evolution towards 6G

• **Motivating examples of ML for Wireless**

• Case studies

• Challenges and thoughts
Wireless Channel Modeling

• Parametric models:
  • Free-space model, two-way ground model
  • Rayleigh, Rician, Lognormal

• Empirical models:
  • Okumura/Hata Model: based on experimental data collected around Tokyo, Japan, by curve fitting and adding correction factors for specific conditions
  • ...

• Ray tracing: for known environments using a simulator

• Channel estimation:
  • Complex channel function represented by basis expansion models: \( h(n; l) = \sum_{m=1}^{M} h_m(l)u_m(n) \)
  • A regression problem

Image source: https://www.tutorialspoint.com/cdma/cdma_fading.htm
Wireless Systems Are Getting More Complex

- Considers:
  - Local execution, and offloading (to which BS)
  - CPU frequency tuning
  - Energy harvesting
  - Mobility/handover
- Control knobs ($c_j$, $e_j$):
  - $c_j$: offloading or local execution
  - $e_j$: energy allocation

Traditional analytical methods may not be capable of handling such complex problems

Distributed Algorithms

- Distributed power control:

- End-to-end congestion control

- Cognitive radios

- Deep reinforcement learning


Indoor Localization: Fingerprinting

- Training locations: war-driving to collect measurements at the training locations
- Compare new measurements from an unknown location with stored fingerprints to find the best match

A classification problem


Indoor Localization: Radio Map Construction

- Construct a radio map with discrete training data
- Use the radio map for location estimation
- Deep Gaussian process

A regression problem

In This Talk

- The evolution towards 6G
- Motivating examples of ML for Wireless
- Case studies
  - Automatic Modulation Classification
  - Energy efficiency maximization
  - 3D human skeleton tracking
- Challenges and thoughts
Automatic Modulation Classification (AMC)

- An essential component of cognitive radio (CR) to detect the nearby emitters, avoid radio inference, and improve spectrum efficiency
- Classify the modulation types of received signals without a priori information of the signal and channel; an important step between signal detection and demodulation
- Applications: spectrum sensing and access, spectrum anomaly detection, classification security, and transmitter identification

Related Work

Likelihood-based:
- Bayesian estimation for modulation classification assuming prior information such as channel and noise models.
- High computational complexity and are not suited for highly dynamic environments.

Feature-based:
- Handcrafted features (i.e., cumulant, and maximum power spectral density) for classifying modulations.
- Requires reliable features and manual selection.

Deep Learning-based:
- Without assuming prior information such as channel models.
- Convolutional neural network (CNN), recurrent neural networks (RNN), and fusion methods are proposed.
- A massive amount of training samples are required and the performance hinges upon the quality of the samples.
• Dataset [1]
  • RadioML2016.10A dataset: synthetic samples with 11 different modulations, including 8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFD
  • 220,000 samples; 20,000 samples for each modulation type
  • Sampled at 20 different SNR levels (from -20dB to 18dB), with 1,000 samples for each SNR level
  • Each radio signal sample consists of 128 consecutive I/Q data units

• CNN model [1]

Generative adversarial network (GAN): a generative machine learning model
• A generative model $G$: generate samples from random noise $z$
• A discriminative model $D$: distinguish generated samples from training samples
• Unsupervised learning, and thus cannot generate labeled data

Conditional GAN (CGAN): both $G$ and $D$ are conditioned on auxiliary information, i.e., class label $y$, that act as an extension to the latent space $z$ to generate and discriminate synthesized data
• Adding class labels $y$ can control the output and guide the generator $G$ to figure out what to generate
Experiment Results (Original vs. Synthesized Data)

8PSK (SNR=16dB): (left) original data; (right) synthesized data.

AM-DSB (SNR=16dB): (left) original data; (right) synthesized data.

AM-SSB (SNR=16dB): (left) original data; (right) synthesized data.

AM-DSB (SNR=16dB): (left) original data; (right) synthesized data.

BPSK (SNR=16dB): (left) original data; (right) synthesized data.
CPFSK (SNR=16dB): (left) original data; (right) synthesized data.

GFSK (SNR=16dB): (left) original data; (right) synthesized data.

PAM4 (SNR=16dB): (left) original data; (right) synthesized data.

QAM16 (SNR=16dB): (left) original data; (right) synthesized data.
Experiment Results (Original vs. Synthesized Data, Cont’d)

QAM64 (SNR=16dB): (left) original data; (right) synthesized data.

WBFM (SNR=16dB): (left) original data; (right) synthesized data.

QPSK (SNR=16dB): (left) original data; (right) synthesized data.
CNN training can greatly benefit from the CGAN augmented data to achieve fast convergence and a smaller training loss.

(a) 1000 synthesized samples.  (b) 5000 synthesized samples.

Training performance when SNR=16 dB with (a) 1000 and (b) 5000 synthesized samples.
Confusion matrices for modulation classification when SNR = 16dB with different amount of synthesized data

Synthesized data helps
Classification accuracy with different amount of augmented data

16% to 25% gain in F1 score

Classification accuracy with different amount of augmented data
In This Talk

• The evolution towards 6G
• Motivating examples of ML for Wireless

• Case studies
  • Automatic Modulation Classification
  • Energy efficiency maximization
  • 3D human skeleton tracking

• Challenges and thoughts
Energy Efficiency Maximization

- The energy efficiency maximization problem:
  \[
  \begin{align*}
  \text{(P1)} \quad & \max_{p_i} \quad \eta_{EE}(p) = \frac{R(p)}{P(p)} \\
  \text{s.t.} \quad & p_i \in [0, p_{\text{max}}]
  \end{align*}
  \]

- Fractional format, conventional convex optimization theory does not apply

- NP-hard problem [1,2]; a global optimal incurs an exponentially growing complexity [2]

- Duality theory and fractional programming [3] can provide suboptimal solutions

Approximation Algorithms

• Branch-and-bound and Reformulation-Linearization Technique (RLT) [1]: high complexity

• Successive pseudo-convex approximation (SPCA) algorithm [2,3]
  • To approximate the objective function with functions that have specific properties (e.g., convexity)
    • Expand the nonconvex sum rate function in the numerator with a first-order Taylor series, which is positive concave
  • The objective function is approximated by a pseudo concave function, which ensures that the original problem and the approximated problem shares the same sets of stationary points
  • Search the stationary points of the approximation problem. Pseudo concavity ensures that the resulted stationary points are global optimal for the approximated problem ...
  • which is a suboptimal solution to the original problem
  • But at high computation cost

Deep Learning Models

DNN

PowerNet (CNN based)
Evaluation

- Generate random locations \( \{d_{ij}\} \), use the SPCA algorithm to compute the power allocation \( \{p_i\} \)
- Repeat, to generate a training dataset

- The DNN and PowerNet models, taking input \( \{d_{ij}\} \) and label \( \{p_i\} \), will be trained to minimize the loss function:
  \[
  \mathcal{L} = \mathbb{E}[(p_i - \hat{p}_i)^2]
  \]

- Compare the ML derived EE with SPCA computed EE
Training Process

DNN

PowerNet
EE Results Under Fast Fading Channels

5 nodes

30 Nodes
Simulation Results: EE

<table>
<thead>
<tr>
<th>N</th>
<th>Methods</th>
<th>Path loss</th>
<th></th>
<th>Shadowing</th>
<th></th>
<th>Fast fading</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>EE (kbps/Joule)</td>
<td>Percentage</td>
<td>EE (kbps/Joule)</td>
<td>Percentage</td>
<td>EE (kbps/Joule)</td>
<td>Percentage</td>
</tr>
<tr>
<td>5</td>
<td>DNN</td>
<td>0.6683</td>
<td>99.10%</td>
<td>0.6340</td>
<td>96.16%</td>
<td>0.5872</td>
<td>95.38%</td>
</tr>
<tr>
<td></td>
<td>PowerNet</td>
<td>0.6636</td>
<td>98.40%</td>
<td>0.6306</td>
<td>95.63%</td>
<td>0.5849</td>
<td>95.01%</td>
</tr>
<tr>
<td></td>
<td>SPCA</td>
<td>0.6744</td>
<td>100%</td>
<td>0.6594</td>
<td>100%</td>
<td>0.6157</td>
<td>100%</td>
</tr>
<tr>
<td>10</td>
<td>DNN</td>
<td>0.5577</td>
<td>95.69%</td>
<td>0.5136</td>
<td>91.10%</td>
<td>0.4796</td>
<td>90.12%</td>
</tr>
<tr>
<td></td>
<td>PowerNet</td>
<td>0.5518</td>
<td>94.69%</td>
<td>0.5087</td>
<td>90.24%</td>
<td>0.4757</td>
<td>89.39%</td>
</tr>
<tr>
<td></td>
<td>SPCA</td>
<td>0.5828</td>
<td>100%</td>
<td>0.5637</td>
<td>100%</td>
<td>0.5322</td>
<td>100%</td>
</tr>
<tr>
<td>20</td>
<td>DNN</td>
<td>0.4203</td>
<td>88.90%</td>
<td>0.3775</td>
<td>82.50%</td>
<td>0.3556</td>
<td>81.26%</td>
</tr>
<tr>
<td></td>
<td>PowerNet</td>
<td>0.4346</td>
<td>91.93%</td>
<td>0.3862</td>
<td>84.40%</td>
<td>0.3630</td>
<td>82.96%</td>
</tr>
<tr>
<td></td>
<td>SPCA</td>
<td>0.4728</td>
<td>100%</td>
<td>0.4576</td>
<td>100%</td>
<td>0.4356</td>
<td>100%</td>
</tr>
<tr>
<td>30</td>
<td>DNN</td>
<td>0.3378</td>
<td>83.46%</td>
<td>0.2984</td>
<td>75.76%</td>
<td>0.2822</td>
<td>74.27%</td>
</tr>
<tr>
<td></td>
<td>PowerNet</td>
<td>0.3603</td>
<td>89.09%</td>
<td>0.3158</td>
<td>80.18%</td>
<td>0.2973</td>
<td>78.22%</td>
</tr>
<tr>
<td></td>
<td>SPCA</td>
<td>0.4048</td>
<td>100%</td>
<td>0.3939</td>
<td>100%</td>
<td>0.3800</td>
<td>100%</td>
</tr>
</tbody>
</table>
Simulation Results: Execution Time

<table>
<thead>
<tr>
<th>( N )</th>
<th>Methods</th>
<th>CPU time (ms)</th>
<th>percentage</th>
<th>GPU time (ms)</th>
<th>percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>DNN</td>
<td>0.022</td>
<td>0.17%</td>
<td>0.025</td>
<td>0.33%</td>
</tr>
<tr>
<td></td>
<td>PowerNet</td>
<td>0.091</td>
<td>0.69%</td>
<td>0.109</td>
<td>1.44%</td>
</tr>
<tr>
<td></td>
<td>SPCA</td>
<td>13.268</td>
<td>100%</td>
<td>7.546</td>
<td>100%</td>
</tr>
<tr>
<td>10</td>
<td>DNN</td>
<td>0.008</td>
<td>0.07%</td>
<td>0.026</td>
<td>0.32%</td>
</tr>
<tr>
<td></td>
<td>PowerNet</td>
<td>0.160</td>
<td>1.43%</td>
<td>0.107</td>
<td>1.33%</td>
</tr>
<tr>
<td></td>
<td>SPCA</td>
<td>11.159</td>
<td>100%</td>
<td>8.066</td>
<td>100%</td>
</tr>
<tr>
<td>20</td>
<td>DNN</td>
<td>0.012</td>
<td>0.09%</td>
<td>0.026</td>
<td>0.27%</td>
</tr>
<tr>
<td></td>
<td>PowerNet</td>
<td>0.483</td>
<td>3.75%</td>
<td>0.115</td>
<td>1.20%</td>
</tr>
<tr>
<td></td>
<td>SPCA</td>
<td>12.887</td>
<td>100%</td>
<td>9.549</td>
<td>100%</td>
</tr>
<tr>
<td>30</td>
<td>DNN</td>
<td>0.019</td>
<td>0.11%</td>
<td>0.029</td>
<td>0.25%</td>
</tr>
<tr>
<td></td>
<td>PowerNet</td>
<td>1.107</td>
<td>6.10%</td>
<td>0.131</td>
<td>1.14%</td>
</tr>
<tr>
<td></td>
<td>SPCA</td>
<td>18.161</td>
<td>100%</td>
<td>11.541</td>
<td>100%</td>
</tr>
</tbody>
</table>
In This Talk

• The evolution towards 6G

• Motivating examples of ML for Wireless

• **Case studies**
  • Automatic Modulation Classification
  • Energy efficiency maximization
  • 3D human skeleton tracking

• Challenges and thoughts
Human pose tracking becomes an important topic in human-computer interaction (HCI)

- Activity Recognition
  1) Full-body sign language reading (e.g., traffic police hand signals, aircraft ground handling)
  2) Fall detection of elders
  3) Surveillance for security

- Motion capture and augmented reality

- Somatosensory games

Camera-based: privacy concerns

RF Sensing Based Related Works

**Wi-Fi based:**
1. 2D pose estimation for multiple people [1], and 3D pose generation [2]
2. Contact-free pose estimation, and wide range detection.
3. Low-cost hardware
   
   Sensitive to the interference from environment

**Radar based:**
1. Frequency-Modulated Continuous Wave (FMCW) radar-based system [1]
2. High accuracy and more robust to the environmental interference than Wi-Fi based systems
   
   Require expensive and complicated hardware

**RFID based:**
1. RF-Kinect based on RF hologram technique [4], and 3D limbs movement tracking with RFID array [5]
2. Good performance for single limb tracking
   
   Not suitable for the full body skeleton reconstruction

---

RFID-Pose/Meta-Pose System Overview

RFID Data Collection
- Kinect Data collection
- RFID Data collection

RFID Data Preprocessing
- Phase calibration
- Downsampling and synchronization
- RFID Data imputation

Skeleton Reconstruction
- Deep neural network
- Vision-aided training

RFID-Pose: Vision-aided 3D Human Pose Estimation

The same motion generates very different RF data when sampled in different environments.

Developing 3D human pose estimation techniques adaptive to the environment has become a great challenge for RF based techniques.

To analyze the influence from the environment, the RF data sampled from a different environment is considered as a different data domain.

<table>
<thead>
<tr>
<th>Position Index</th>
<th>Estimation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position 1 (Trained)</td>
<td>4.53cm</td>
</tr>
<tr>
<td>Position 2 (Trained)</td>
<td>3.82cm</td>
</tr>
<tr>
<td>Position 3 (Trained)</td>
<td>4.75cm</td>
</tr>
<tr>
<td>Position 4 (Untrained)</td>
<td>8.38cm</td>
</tr>
<tr>
<td>Position 5 (Untrained)</td>
<td>5.71cm</td>
</tr>
<tr>
<td>Position 6 (Untrained)</td>
<td>9.14cm</td>
</tr>
</tbody>
</table>

RFID phase collected in two different domains for the same motion.
Meta-Learning Based Solution

Structure of the deep learning model

- **Meta learning [1]:**
  - To learn the learning algorithm itself, i.e., **learning to learn**
  - To train a general model that can **generalize** across different tasks or datasets
  - Learn and adapt quickly from few-shot of examples, and be able to keep adjusting as more data coming in

- The network parameters should first be **well initialized** in the pretraining phase
- The network will be **fine-tuned** for a new data domain with only a few additional samples

---

The deep learning model is pretrained with data from four known data domains. A domain fusion algorithm is adopted to produce more data domains. The training variables are updated recursively by the Reptile meta-learning algorithm. When transferring to a new data domain, we only need to collect very few examples to fine-tune the generalized network.

Seven data domains are sampled in the computer lab, and the 8th domain is sampled in an empty corridor.

- D1 to D4 are used for pretraining.
- D5 to D8 are considered as new data domains for validation.

Five subjects participate in the experiments.
Overall performance in terms of mean estimation error in the eight different data domains.

Fine-tuning performance of different activities with different shots of new data in new data domain D5.

Average error comparison with the baseline method RFID-Pose.

<table>
<thead>
<tr>
<th>Domain Index</th>
<th>RFID-Pose</th>
<th>Meta-Pose</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_5$</td>
<td>6.72cm</td>
<td>3.72cm</td>
</tr>
<tr>
<td>$D_6$</td>
<td>7.62cm</td>
<td>4.32cm</td>
</tr>
<tr>
<td>$D_7$</td>
<td>5.46cm</td>
<td>3.51cm</td>
</tr>
<tr>
<td>$D_8$</td>
<td>4.62cm</td>
<td>4.11cm</td>
</tr>
<tr>
<td>$D_{all}$</td>
<td>6.27cm</td>
<td>3.97cm</td>
</tr>
</tbody>
</table>

One shot of data in Meta-Pose is defined as consecutive data samples within 6 seconds.

With few-shot finetuning, the mean error for all the new data domains is 3.98cm, which is very similar to that of the pretrained data domains.

4-shot fine-tuning is sufficient; the minimum error 4.04cm is achieved by walking.

Mean error of RFID-Pose for all the new data domains is 6.27cm, while that for Meta-Pose is only 3.97cm (a 36.68% reduction).
In This Talk

- The evolution towards 6G
- Motivating examples of ML for Wireless
- Case studies
- Challenges and thoughts
Challenges and Thoughts

• Platform, dataset, benchmark
• Need high quality (labeled) data:
  • *How many cats do you need to recognize a cat* *(sample complexity)?*
• The ML performance will be as good as your training data
  • Inconsistency between the training dataset and real deployment
• Sparse noisy data ➔ Data *augmentation*, *data imputation* *(in time, space)*
Challenges and Thoughts (cont’d)

- Although communications systems are based on probabilistic models, we do need to guarantee the worst-case performance in many cases ➔ robustness
  - Incorporate models into the ML model
- Many applications are in real-time: e.g., autonomous driving ➔ fast convergence, few-shot learning
- Dynamic environment ➔ need to retrain the models ➔ generalization
  - Transfer learning
  - Meta-learning, few-shot learning, ...
Challenges and Thoughts (cont’d)

• Reproducible results
• AI/ML for wireless and wireless for AI/ML
• Can ML/AI bring about new theory and breakthroughs in wireless communications and networking, as traditional models did?
• Explainable ML: need to know why it works or why it does not work
• Distributed ML/federated learning for resource constrained mobile devices and privacy
• The wireline network may need to be redesigned: the new bottleneck?
• Application, application, application!
  • Technologies that did not fly: need killer apps
Conclusions

• 5G gets real, 6G is on the horizon

• **Intelligence**: likely to be a key component and common theme of the new generation

• Advances in AI/ML algorithms, data, computing, and platforms: ready for wireless communications and networking

• Shared our experience of applying ML to solving several wireless problems

• Many challenges and interesting problems remain
This work is supported in part by the National Science Foundation (NSF) under Grants ECCS-1923163, ECCS-1923717, and CNS-2107190, and through the Wireless Engineering Research and Education Center (WEREC) at Auburn University.

For more information: http://www.eng.auburn.edu/~szm0001/