IEEE GLOBECOM 2020 Tutorial on

Distributed Deep Learning: Concepts, Methods & Applications in Wireless Networks

Part 3: Distributed & Federated Learning in Wireless Networks
Tutorial Outline

• Introduction (DG)

• Distributed & Federated Learning: Concepts & Methods (WS)
  • Basic concepts of federated and distributed learning
  • Reducing communication overhead
  • Clustered federated learning

• Distributed & Federated Learning in Wireless Networks (DG)
  • Distributed inference
  • Distributed training
  • Resource optimization for federated learning
  • Over-the-air distributed learning

• Distributed Learning & Neural Network Compression (WS)
  • Conceptual similarities and differences between these two problems
  • Standardization activities: ITU FG ML5G, MPEG AHG CNNMCD

• Conclusions
Distributed Machine Learning

• Distributed inference

• Distributed training
Local Inference on Device

• Most IoT devices have limited computational resources. Cannot deploy complex DNN models.

• Alternatives:
  • Shallow networks
    • Design and train simpler networks for edge devices
    • Knowledge distillation
  • Custom hardware designs for FPGAs (Microsoft Brainwave, Xilinx Everest) or ASICS (Google TPUs, IntelNervana, IBMTrueNorth)
  • Quantization: Reduce precision of weights
    • Quantization of both weights and activations possible with below 1% accuracy loss
    • Better results with layer-wise optimized precision
  • Network pruning: Remove unnecessary weights
    • Pruning weights or channels
Most IoT devices have limited computational resources: local inference not possible

Device-edge co-inference by splitting deep neural networks

Feature size of intermediate layers can be larger than input (up to 10x in VGG16)
Joint Feature Encoding and Communication

- Consider feature encoding and communication jointly in an end-to-end fashion
- Prune feature encoder network to reduce complexity

Distributed Inference at the Edge

• Edge devices typically have low computational capacity and power

• Local inference not possible
  • Decisions may rely on data (e.g., terrain information) available at an edge server (e.g., a base station), or on signals from other devices

• Latency becomes a major challenge: In self-driving cars immediate detection of obstacles is critical to avoid accidents
**Image Retrieval at the Edge**

**Goal:** match a pedestrian’s image from a wireless camera with another image in a large database

**Standard approach:**
- Transmit images to the cloud
- Determine features most relevant for re-identification over image database
- Re-ID baseline: Deep convolutional neural network, e.g., ResNet-50

Retrieval-based Compression

Joint Compression and Channel Coding

- Feature transmission is a joint source-channel coding problem
- Separation is suboptimal in general
- Joint approach provides better performance as well as ‘graceful degradation’
Person Re-identification Over Noisy Channels

- CUHK03 dataset: 14096 images of 1467 identities taken from two camera views.
- 256x128 coloured images
Distributed Training

- With the increase in data size and model complexity, ML tasks cannot be handled on a single machine
- DeepMind’s AlphaGo ran on 1920 CPUs and 280 GPUs

- Mobile edge devices employ computation power of edge servers
- Data can be huge with limited information density
- Communication links are power and bandwidth limited
- Privacy concerns

Distributed learning/computing

Federated learning
Distributed Training at the Edge

• Computationally limited edge devices can exploit edge servers for training large models
  • Increase computation speed
  • To introduce privacy and security

Challenges:
• Straggling servers
• Colluding servers
• Communication bottleneck
Distributed Linear Regression

- N labeled data points \((x_1, y_1), \ldots, (x_n, y_n)\) \(\mathbf{X} = [x_1, \ldots, x_N]^T\)
- Minimize mean squared error:

\[
L(\theta) = \frac{1}{2} \sum_{i=1}^{N} (y_i - x_i^T \theta)^2
\]

- Gradient Descent: \(\theta_{t+1} = \theta_t - \eta_t \nabla_\theta L(\theta)\)

\[
\nabla_\theta L(\theta) = \mathbf{X}^T \mathbf{X} \theta_t - \mathbf{X}^T \mathbf{y}
\]

Parameter vector
Remains constant over iterations
To be computed at each iteration
Distributed Matrix-Vector Multiplication

$$\nabla_{\theta_t} L(\theta_t) = X^T X \theta_t - X^T y$$

$$= \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix} \theta_t - X^T y$$
\[ \nabla_{\theta_t} L(\theta_t) = X^T X \theta_t - X^T y \]

\[ = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix} \theta_t - X^T y \]

Distributed Matrix-Matrix Multiplication

\[ \nabla_\theta L(\theta) = X^T X \theta_t - X^T y \]

- Remains constant over iterations
- To be computed at each iteration

\[ X^T X \theta_t = \sum_{i=1}^{N} x_i x_i^T \theta_t \]

- Can be computed in a distributed manner
Distributed Matrix-Matrix Multiplication

\[ X_1 X_1^T \theta_t \]

\[ X_2 X_2^T \theta_t \]

\[ X_3 X_3^T \theta_t \]

\[ X_4 X_4^T \theta_t \]
Gradient Coding

Computation load: \( r = 2 \)

\[
\frac{1}{2}x_1x_1^T\theta + x_2x_2^T\theta \\
\frac{1}{2}x_1x_1^T\theta + x_3x_3^T\theta \\
x_2x_2^T\theta - x_3x_3^T\theta
\]

\[
2 \left( \frac{1}{2}x_1x_1^T\theta + x_3x_3^T\theta \right) + (x_2x_2^T\theta - x_3x_3^T\theta) = x_1x_1^T\theta + x_2x_2^T\theta + x_3x_3^T\theta
\]

Coded Computation and Learning

• Distributed polynomial codes
  • S. Li et al., “Polynomially coded regression: Optimal straggler mitigation via data encoding,” 2018

• Coded computing for privacy/ security
  • D’Oliveira et al. “GASP codes for secure distributed matrix multiplication,” 2018.

• Computation scheduling

• Partial recovery

• Dynamic coded computation
  • Buyukates et al., “Gradient coding with dynamic clustering for straggler mitigation”, 2020.
Federated Edge Learning (FEEL)

- Wireless devices with their own data
- FL provides not only privacy but also communication efficiency
- Channels are time-varying and inhomogeneous across devices
- Devices interfere with each other

Device Scheduling

- Device selection should take channels into account
- Let each device compress their updates to $d$ bits
- Allocate bandwidth across users: $\mathbf{w} = (w_1, \ldots, w_K)$

\begin{align*}
\text{Minimize delay in each round:} & \quad \max_{i \in \mathcal{S}_t} \frac{d}{R_{i,t}(\mathbf{P}_t, \mathbf{h}_t, \mathbf{w}_t)} + \delta_{i,t}^\text{comp} \\
\text{Alternatively, we can fix a deadline,} & \quad \text{and maximize number of participating devices}
\end{align*}
Channel-Aware Device Scheduling

Choose 20 out of 100 devices randomly distributed in a 500m radius cell.

CIFAR-10 training
Age-based Device Scheduling

Yang et al., Age-based scheduling policy for federated learning in mobile edge networks, 2019.
Hierarchical FEEL

- Performance limited by worst user’s channel

Hierarchical FEEL over Cellular Networks

- 28 devices uniformly distributed in a circular area with radius 750m, 7 clusters
- 600 subcarriers with subcarrier spacing of 30KHz.
- Max transmit power of BS, SBSs, and devices: 20W, 6.3W, and 0.2W, respectively
- CIFAR-10 image classification
- ResNet-18 architecture

H: no. of intra-cluster SGD iterations, for each global model update

Reduction in latency (pathloss exponent 3.5):

<table>
<thead>
<tr>
<th>H</th>
<th>FL</th>
<th>HFL, H = 2</th>
<th>HFL, H = 4</th>
<th>HFL, H = 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>H=2</td>
<td></td>
<td>x35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H=4</td>
<td></td>
<td>X39.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H=6</td>
<td></td>
<td>X41.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Update-aware Device Scheduling

• Scheduled devices quantize their model updates based on their link qualities

• **Best Channel (BC):** Allocate channel resources so that each device transmits same number of bits

• **Best $l_2$ norm (BN2):** Based on the $l_2$ norm of local updates. Allocate channel proportional to $l_2$ norms

• **Best Channel - Best $l_2$ norm (BC-BN2):** Choose top $K_c$ channels, and then top $K$ among those based on updates. Bw. allocation as in BN2

• **Best $l_2$ norm - Channel (BN2-C):** Devices find quantized updates based on full channel bandwidth. Choose top $K$ devices based on $l_2$ norm of quantized updates. Bw. allocation as in BN2.
Update-aware Device Scheduling

- MNIST
- 3 local iterations
- \(M=40\) devices
- iid data distribution

Better to schedule only one device!

- BN2 and BC-BN2 best performance:
  - important to consider updates as well as channels when scheduling devices
Update-aware Device Scheduling: Non-iid Data

- MNIST dataset, multi-layer perceptron
- 3 local iterations
- M=40 devices
- Non-iid data distribution

Better to schedule 10 devices!
Federated Learning at the Wireless Edge

• Mobile devices connected to PS through wireless links: bandlimited multiple access channel (MAC)
  • Digital approach: Separate learning and communications
  • Analog approach: Let channel do gradient averaging

\[
\theta_{t+1} = \theta_t + \frac{1}{K} \sum_{k=1}^{K} \tilde{g}_k(t)
\]

Analog Distributed Gradient Descent (A-DSGD)

- All workers transmit simultaneously
- Gradient dimension typically >> bandwidth
  - (50-layer ResNet network has ~26million weight parameters)
- Thresholding to sparsify gradient estimates
- Use *pseudo-random linear projection* to reduce bandwidth
Over-the-air Federated Image Classification

- Distributed MNIST classification (single layer with 10 neurons, ADAM)
- Parameter vector size $d = 28 \times 28 \times 10 + 10 = 7850$
- Bandwidth: $d/2$ symbols
- Sparsity level: $d/2$
FEEL over Fading Wireless Channels

• Channel gains known: requires channel inversion


FEEL with Over-the-Air Majority Voting

• Analog over-the-air aggregation may not be applicable to legacy systems
• We instead use single bit quantization \textit{ala} signSGD and over-the-air majority voting:
  \[ \tilde{g}^{(t)}_k = \text{sign}(g^{(t)}_k) \]

• Map each bit to a BPSK symbol (or QPSK)
• Truncated channel inversion \(\to\) remain silent in poor channel condition with probability \(\alpha\)
  \[ \tilde{g}^{(t)} = \sum_{k=1}^{K} \sqrt{\rho_0} [\tilde{g}^{(t)}_k]_{tr} + z^{(t)} \]
• PS sends back \(\text{sign}(\tilde{g}^{(t)})\)
  \[ \theta_{t+1} = \theta_t + \eta \cdot \text{sign}(\tilde{g}^{(t)}) \]

G. Zhu, Y. Du, DG, K. Huang, One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis, 2020.
FEEL with Over-the-Air Majority Voting

FEEL with Blind Transmitters

- Channel gains unknown at the transmitters

For more information about our work: 
Information Processing and Communications Lab 
www.imperial.ac.uk/ipc-lab

IEEE JSAC Special Series on Machine Learning in Communications and Networks, Jan 2021

IEEE Communications Magazine, Special Issue on Communication Technologies for Efficient Edge Learning, Dec. 2020

D. Gündüz, D. Burth Kurka, M. Jankowski, M. Mohammadi Amiri, E. Ozfatura, and S. Sreekumar, 
Communicate to learn at the edge, IEEE Communications Magazine, 2020.

D. Gunduz, P. de Kerret, N. Sidiroupoulos, D. Gesbert, C. Murthy, M. van der Schaar, 