Secure Multi-Party Computation (MPC)

- Introduced by Andrew Chi Chi Yao [1982]
- Enables n mutually distrusting parties to jointly compute a public function on their private inputs.
- Properties:
  - Privacy: Nothing beyond function output is leaked
  - Correctness: All parties obtain the correct output of the function
- Adversarial model (Models the distrust among the parties)
  - Semi-honest: honest but curious
  - Malicious: arbitrarily deviate from protocol specification
- Corruption threshold
  - Honest Majority: majority are honest
  - Dishonest Majority: minority are honest
- Security levels
  - Security with Abort: honest parties may abort without receiving output
  - Fairness: either all parties or none get the output
  - Guaranteed Output Delivery (GOD): all parties guaranteed to obtain output

MPCLeague Protocol

- Model: 4PC, honest majority with security guarantee of GOD.
- Efficient and robust 4PC mixed-protocol framework: at least 2x better than the state-of-the-art Trident [3].
- Efficient end-to-end conversions to switch between arithmetic/Boolean/garbled representations, efficient truncation, multi-input multiplication
- Two robust frameworks: MPCLeague1 that provides a fast online phase, and MPCLeague2 that provides best overall communication.

<table>
<thead>
<tr>
<th>Ref</th>
<th>Training &amp; Inference</th>
<th>Training</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compu.</td>
<td>Time</td>
<td>Cost</td>
</tr>
<tr>
<td>Trident</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPCLeague1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPCLeague2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Compu.: Communication, Time: Runtime, Cost: Overall Monetary Cost, TP: Protocol
- Online: Throughput, on: online, tot: total (cf. Table 2)

Table 1: Comparison of Trident [14] with the two versions of MPCLeague for deep neural networks (cf. NN-4 in §6).

Privacy-Preserving Machine Learning (PPML)

- Outsourced server setting – computation outsourced to four untrusted non-colluding servers.
- Works over I-bit rings (64-bit for benchmarking).
- Follows the preprocessing paradigm.

<table>
<thead>
<tr>
<th>Ref</th>
<th>NN-1</th>
<th>NN-2</th>
<th>NN-3</th>
<th>NN-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>275S</td>
<td>1210B</td>
<td>420B</td>
<td>902B</td>
</tr>
<tr>
<td>SWIFT</td>
<td>866S</td>
<td>416S</td>
<td>420B</td>
<td>902B</td>
</tr>
<tr>
<td>MPPCLeague1</td>
<td>598S</td>
<td>436S</td>
<td>310B</td>
<td>902B</td>
</tr>
</tbody>
</table>

Table 1: Monetary Costs for Training (1 mini-batch, 10^6 iterations)

- Ref: SVM, NN-1, NN-2, NN-3, NN-4
- ABYS: 275S, 1210B, 420B, 902B
- SWIFT (S): 866S, 416S, 420B, 902B
- MPPCLeague1: 598S, 436S, 310B, 902B
- MPPCLeague2: 798S, 437S, 347S, 828B

Table 5: Monetary Costs for Inference (10^6 predictions)

Results

- Neural Networks (NN)- Training and Inference
- Support Vector Machine (SVM): Inference only
- WAN- Google Cloud with 50Mbps, Implemented in C++17 using ENCRYPTO

Neural Network Architectures:

- NN-1: 3-layered fully connected network with ReLU activation after each layer (around 118K parameters).
- NN-2: Convolutional NN with 2 hidden layers, having 100 and 10 nodes
- NN-3: LeNet [1] has 2 convolutional layers and 2 fully connected layers with ReLU activation after each layer, additionally followed by maxpool for convolutional layers (around 431K parameters).
- NN-4: VGG16 [2] has $165$ layers in total and comprises of fully-connected, convolutional, ReLU activation and maxpool layers (around 138 million parameters)

Benchmarking Parameters:

- Online and total protocol execution time.
- Online and total cumulative protocol execution time (sum of uptime of all).
- Throughput for the case of inference.
- Monetary cost using Google Cloud pricing.

Contact

Ajith Suresh and Nishat Koti
Indian Institute of Science, Bangalore
Email: [ajith, kotis]@iisc.ac.in

References