Private Machine Learning in TensorFlow using Secure Computation

Morten Dahl, Jason Mancuso, Yann Dupis, Ben Decoste, Morgan Giraud, Ian Livingstone, Justin Patriquin, Gavin Uhma

Privacy Preserving Machine Learning workshop, NeurIPS 2018
Prediction

Sensitive?

x

Client

pred

Prediction service
Encrypted Prediction

Client → Prediction service

enc(x) → enc(pred)

Client

Prediction service
Encrypted Prediction using Secure Computation

Client

share1(x) → share1(pred) → share2(x) → share2(pred) → Complex interaction
Multidisciplinary Challenges

Engineering
(distributed, multi-core, readability)

Cryptography
(protocol, techniques, guarantees)

Machine learning
(models, activations, approx)

Data science
(use-cases, workflow, deployment)
Nice To Have

Easy to experiment

- Flexibility
- Separation of concerns
- Benchmarking

Easy to explore

- High-level interface
- Familiar framework
- Gradual adaptation

Leverage existing efforts and minimize boilerplate
TensorFlow

Backed by Google

Used for production-level model training and deployment
Dataflow Graphs
Distributed TensorFlow

[Abadi et al ‘16.] TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems
Optimized Execution

<table>
<thead>
<tr>
<th>Name</th>
<th>Total Wall Duration</th>
<th>Self Time</th>
<th>Average Wall Duration</th>
<th>Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Loop</td>
<td>0.000 ms</td>
<td>0.006 ms</td>
<td>0.006 ms</td>
<td>40</td>
</tr>
<tr>
<td>Main Loop</td>
<td>0.000 ms</td>
<td>0.000 ms</td>
<td>0.000 ms</td>
<td>0</td>
</tr>
<tr>
<td>Main Loop</td>
<td>0.007 ms</td>
<td>0.007 ms</td>
<td>0.007 ms</td>
<td>13</td>
</tr>
<tr>
<td>Main Loop</td>
<td>0.010 ms</td>
<td>0.010 ms</td>
<td>0.010 ms</td>
<td>14</td>
</tr>
<tr>
<td>Main Loop</td>
<td>0.001 ms</td>
<td>0.001 ms</td>
<td>0.001 ms</td>
<td>14</td>
</tr>
<tr>
<td>Main Loop</td>
<td>0.006 ms</td>
<td>0.006 ms</td>
<td>0.006 ms</td>
<td>14</td>
</tr>
<tr>
<td>Main Loop</td>
<td>0.006 ms</td>
<td>0.006 ms</td>
<td>0.006 ms</td>
<td>14</td>
</tr>
</tbody>
</table>

Concurrent

Batching
tf-encrypted

Open source community project for exploring and experimenting with privacy-preserving machine learning in TensorFlow
import tensorflow as tf
import tf_encrypted as tfe

def provide_weights():
    return tf.Print([], [tf.argmax(logits)])

def provide_input():
    return tf.Print([], [tf.argmax(logits)])

def receive_output(logits):
    return tf.Print([], [tf.argmax(logits)])

# get model weights
w0, b0, w1, b1, w2, b2 = provide_weights()

# get prediction input
x = provide_input()

# compute prediction
layer0 = tf.nn.relu(tf.matmul(x, w0) + b0)
layer1 = tf.nn.relu(tf.matmul(layer0, w1) + b1)
logits = tf.matmul(layer2, w2) + b2

# process result of prediction
prediction_op = receive_output(logits)

# run graph execution in a tf.Session
with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
sess.run(prediction_op)
```python
def _matmul_masked_masked(prot, x, y):
    a, a0, a1, alpha_on_0, alpha_on_1 = x.unwrapped
    b, b0, b1, beta_on_0, beta_on_1 = y.unwrapped

    with tf.name_scope('matmul'):
        with tf.device(prot.crypto_producer.device_name):
            ab = a.matmul(b)
            ab0, ab1 = prot._share(ab)

        with tf.device(prot.server_0.device_name):
            z0 = ab0 + a0.matmul(beta) + alpha.matmul(b0) + alpha.matmul(beta)

        with tf.device(prot.server_1.device_name):
            z1 = ab1 + a1.matmul(beta) + alpha.matmul(b1)

    return PondPrivateTensor(prot, z0, z1)
```
## Benchmarks

<table>
<thead>
<tr>
<th></th>
<th>Runtime, ms</th>
<th>Accuracy</th>
<th>KL-divergence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pond</td>
<td>SecNN</td>
<td>TF</td>
</tr>
<tr>
<td>A</td>
<td>14 (3.8)</td>
<td>112 (63)</td>
<td>97.35%</td>
</tr>
<tr>
<td>B</td>
<td>126 (115)</td>
<td>243 (79)</td>
<td>99.26%</td>
</tr>
<tr>
<td>C</td>
<td>124 (93)</td>
<td>293 (78)</td>
<td>99.44%</td>
</tr>
</tbody>
</table>

2.2x, 1.1x, 0.85x relative to reference custom C++ implementation
Thank you!

Common high-level framework for machine learners and cryptographers with promising performance