Abstract

While data provides tremendous insights, users’ personal information is often exposed with limited protection. This project aims to build a privacy preserving deep learning framework that trains and updates models without directly using raw data. Using an image classification task as a case study, the results show that similar accuracy can be achieved with only sharing a small fraction of model parameters, not data.

Data

- Data for this study comes from:
  - Microsoft Research Celebrity Face data,
  - Kaggle dogs and cats classification data
- Randomly sampled 26,142 people’s pictures from the database,
- Combined with 25,000 images for dogs and cats,
- Data is separated for initial training and PPDL model updating phase.

Table 1. Summary of datasets.

<table>
<thead>
<tr>
<th>Training</th>
<th>Validation</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Men</td>
<td>5,426</td>
<td>6,026</td>
</tr>
<tr>
<td>Women</td>
<td>6,621</td>
<td>7,357</td>
</tr>
<tr>
<td>Dogs</td>
<td>6,050</td>
<td>6,050</td>
</tr>
<tr>
<td>Cats</td>
<td>6,000</td>
<td>6,000</td>
</tr>
<tr>
<td>Total</td>
<td>23,447</td>
<td>27,807</td>
</tr>
</tbody>
</table>

CNN Initial Model

1. Suppose Company A developed a photo app. It trains an image classification model using an initial dataset.

   - Two types of CNN models were tested:
     - a built-from-scratch model with RELU activation
     - a transfer learning model with VGG16 base
   - Tried SGD, RMSProp and Adam optimizers.
   - 100 epochs, 0.0001 learning rate
   - Final initial model used VGG16 with SGD optimizer

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Training Accuracy</th>
<th>Validation Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD</td>
<td>75.9%</td>
<td>72.1%</td>
</tr>
<tr>
<td>RMSProp</td>
<td>78.3%</td>
<td>74.8%</td>
</tr>
<tr>
<td>Adam</td>
<td>79.2%</td>
<td>77.7%</td>
</tr>
</tbody>
</table>

| CNN built from scratch model | SGD | 83.5% | 79.6% |
| CNN VGG16 fix all but last layer | SGD | 84.1% | 81.4% |

Figure 2. Architecture of the built-from-scratch model.

Privacy Preserving Model Updating

Now the initial model is trained, and the photo app is published. Model needs to be continuously trained using data from users’ cellphones. Company A wants to protect users’ privacy by updating the model without collecting raw pictures.

![Diagram of local training process](image)

Day 1:
- Total training completed from central server, obtain new matrix.
- Parameter updated in central server.

Day 2:
- Local training
- “Value clipping” in Eq. 2.
- Add a differential noise as a random perturbation of parameters to uploaded.

Day 4:
- Local training
- “Value clipping” in Eq. 2.
- Add a differential noise as a random perturbation of parameters to uploaded.

Users privacy is protected through:
- The central server does not collect raw data, but only collect updated gradients from local training.
- Each participant independently trains locally, and uploads only a fraction of gradients.
- The uploaded gradients are further protected through differential privacy by adding random noise and value clipping, or through only uploading largest gradient.

Results

- By sharing only a small fraction of gradients (10%, and 20% in our case) at each gradient descent step, we can achieve similar accuracy as the privacy violating case of training in a centralized machine with 100% data.
- As expected, the “no update” training has lowest accuracy, and the centralized “all data exposing” training achieves the highest accuracy.

Future work

- Update the parameters in more layers. Currently, the framework only updates the last layer.
- Other model updating mechanism. Currently, only through Stochastic Gradient Descent.
- Improve the initial model performance.

Links

GitHub repo: [https://github.com/anuba20/PrivacyPreservingDeepLearning](https://github.com/anuba20/PrivacyPreservingDeepLearning)

Presentation: [https://youtu.be/QQ9Uuh57hWE](https://youtu.be/QQ9Uuh57hWE)

Acknowledgements: Shawn Chai, Shokri and Shmatikov’s work, CS230 teaching team, TA Kaidi Cao