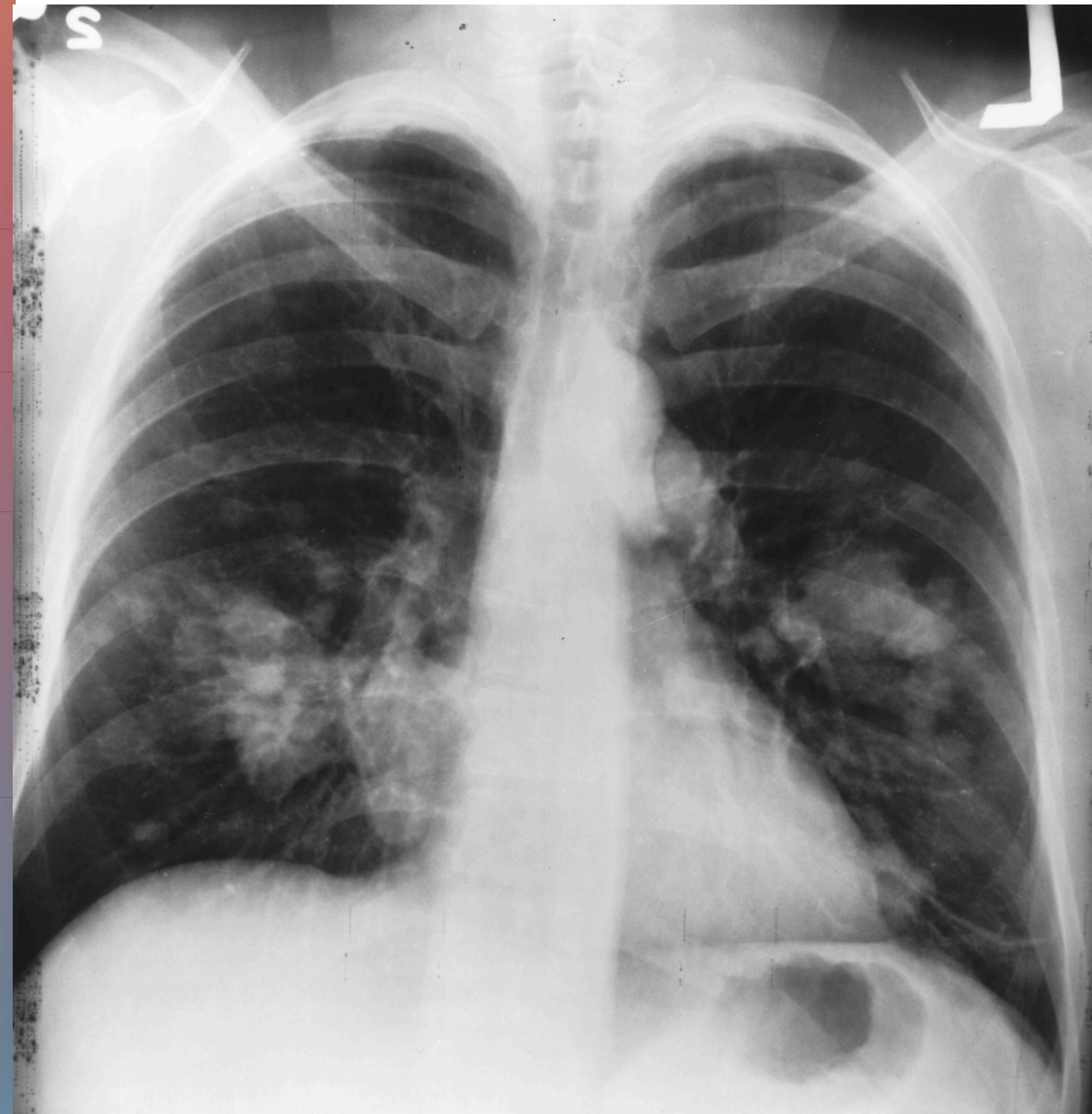# Privacy Preserving AI

IS IT POSSIBLE TO:

answer questions using data we cannot see?

What do tumors look like in humans?

What do tumors look like in humans?

Source: Wikipedia Commons

◆ Step **1**: Download millions of tumor images.

What do tumors look like in humans?

Source: Wikipedia Commons

◆ Step **0**: Buy a dataset from a hospital.

◆ Step **1**: Download millions of tumor images.

What do tumors look like in humans?

Source: Wikipedia Commons

◆ **Step -1**: Persuade a VC.

◆ Step **0**: Buy a dataset from a hospital.

◆ Step **1**: Download millions of tumor images.

What do tumors look like in humans?

Source: Wikipedia Commons

◆ **Step -2**: Create a business plan!

◆ Step **-1**: Persuade a VC.

◆ Step **0**: Buy a dataset from a hospital.

◆ Step **1**: Download millions of tumor images.

What do tumors look like in humans?

Source: Wikipedia Commons

- Step **-3**: Find a business partner!

- Step **-2**: Create a business plan!

- Step **-1**: Persuade a VC.

- Step **0**: Buy a dataset from a hospital.

- Step **1**: Download millions of tumor images.

Source: Wikipedia Commons

What do tumors look like in humans?

◆ **Step -4**: Spam all my classmates on LinkedIn!

◆ **Step -3**: Find a business partner!

◆ **Step -2**: Create a business plan!

◆ **Step -1**: Persuade a VC.

◆ **Step 0**: Buy a dataset from a hospital.

◆ **Step 1**: Download millions of tumor images.

What do handwritten digits look like?

◆ Step **1**: Download data

◆ Step **2**: Download SOTA training script

◆ Step **3**: Run script.

What do handwritten digits look like?
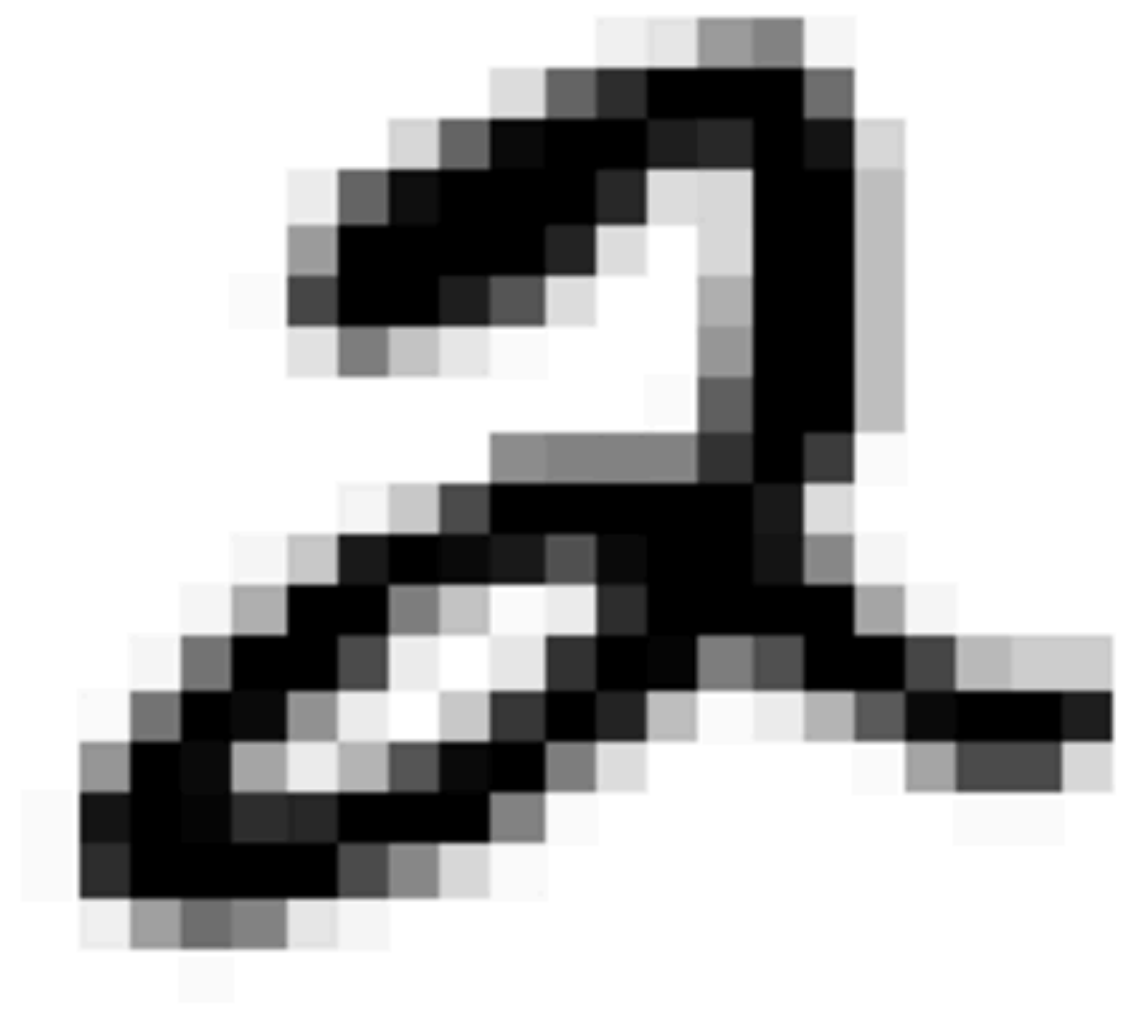
Getting access to private data is **HARD!**

We SOLVE tasks which are accessible:

✓ ImageNet

✓ MNIST

✓ CIFAR-10

✓ Librispeech

✓ WikiText-103

✓ WMT

We SOLVE tasks which are accessible:

- ✓ ImageNet
- ✓ MNIST
- ✓ CIFAR-10
- ✓ Librispeech
- ✓ WikiText-103
- ✓ WMT

... but what about?

- ◆ Cancer
- ◆ Alzheimers
- ◆ Dementia
- ◆ Depression
- ◆ Anxiety
- ◆ ... the Common Cold?

IS IT POSSIBLE TO:

answer questions using data we cannot see?

```
atrask:~ pip install the-worlds-data
```

OpenMind is a Community

OpenMined / **PySyft**

Sponsor | Used by ⌄ 16 | Unwatch ⌄ 167 | ★ Unstar 3,893 | Fork 869

<> Code | ⊘ Issues 162 | Pull requests 17 | ▶ Actions | Security | Insights | Settings

A library for encrypted, privacy preserving deep learning

Edit

deep-learning secure-computation pytorch privacy cryptography Manage topics

⊙ **5,337** commits | ⑂ **42** branches | ⬚ **5** releases | ⚡ **1** environment | 👥 **180** contributors | ⚖ Apache-2.0

Branch: dev ⌄ | New pull request

Create new file | Upload files | Find File | Clone or download ⌄

👥 jvmancuso and robert-wagner rm needless integration test (#2629)   ✓ Latest commit 8d8baa4 3 days ago

▭ .github/ISSUE_TEMPLATE	Update issue templates	4 months ago
▭ art	Improve the diagram	3 months ago
▭ docker-image	delete unecessary package installation, numpy comes already with pysyft	4 months ago
▭ docs	bumpversion 0.1.27a1 -> 0.1.28a1 (#2619)	13 days ago
▭ examples	Improvements to the Federated Recurrent Neural Network notebook (#2613)	18 days ago
▭ images	Add files via upload	3 months ago
▭ syft	Implementing Protocol (#2605)	7 days ago
▭ test	rm needless integration test (#2629)	3 days ago
▭ .flake8	changed ignore to select	6 months ago
▭ .gitbook.yaml	added gitbook.yaml	5 months ago
▭ .gitignore	Changes to socketio_server worker to allow sync communication with th...	5 months ago
▭ .pre-commit-config.yaml	change black repo from ambv-> psf (#2509)	2 months ago
▭ .travis.yml	rm needless integration test (#2629)	3 days ago
▭ CONTRIBUTING.md	Update CONTRIBUTING.md (#2449)	2 months ago

Tool 1: Remote Execution

Tool 1: Remote Execution

Tool 1: Remote Execution

```
In [1]: import syft as sy
        import torch as th
        hook = sy.TorchHook(th)
```

Tool 1: Remote Execution

```
In [1]:  import syft as sy
         import torch as th
         hook = sy.TorchHook(th)

In [2]:  hospital_datacenter = sy.VirtualWorker(hook, id="May Clinic")
```

Tool 1: Remote Execution

```
In [1]:  import syft as sy
         import torch as th
         hook = sy.TorchHook(th)

In [2]:  hospital_datacenter = sy.VirtualWorker(hook, id="May Clinic")

In [5]:  x = th.tensor([1,3,4,5])
         x = x.send(hospital_datacenter)
         x

Out[5]:  (Wrapper)>[PointerTensor | me:20069769489 -> May Clinic:27535193014]
```

Tool 1: Remote Execution

```
In [1]:  import syft as sy
         import torch as th
         hook = sy.TorchHook(th)

In [2]:  hospital_datacenter = sy.VirtualWorker(hook, id="May Clinic")

In [5]:  x = th.tensor([1,3,4,5])
         x = x.send(hospital_datacenter)
         x

Out[5]:  (Wrapper)>[PointerTensor | me:20069769489 -> May Clinic:27535193014]

In [ ]:  x.
```

```
x.abs
x.abs_
x.acos
x.acos_
x.add
x.add_
x.addbmm
```

Tool 1: Remote Execution

```
In [1]:  import syft as sy
         import torch as th
         hook = sy.TorchHook(th)
```

```
In [2]:  hospital_datacenter = sy.VirtualWorker(hook, id="May Clinic")
```

```
In [5]:  x = th.tensor([1,3,4,5])
         x = x.send(hospital_datacenter)
         x
```

```
Out[5]:  (Wrapper)>[PointerTensor | me:20069769489 -> May Clinic:27535193014]
```

```
In [6]:  y = x + x
```

Tool 1: Remote Execution

```
In [1]:  import syft as sy
         import torch as th
         hook = sy.TorchHook(th)
```

```
In [2]:  hospital_datacenter = sy.VirtualWorker(hook, id="May Clinic")
```

```
In [5]:  x = th.tensor([1,3,4,5])
         x = x.send(hospital_datacenter)
         x
```

```
Out[5]:  (Wrapper)>[PointerTensor | me:20069769489 -> May Clinic:27535193014]
```

```
In [6]:  y = x + x
```

```
In [7]:  y
```

```
Out[7]:  (Wrapper)>[PointerTensor | me:52194974528 -> May Clinic:13992236415]
```

Tool 1: Remote Execution

```
In [1]:   import syft as sy
          import torch as th
          hook = sy.TorchHook(th)

In [2]:   hospital_datacenter = sy.VirtualWorker(hook, id="May Clinic")

In [5]:   x = th.tensor([1,3,4,5])
          x = x.send(hospital_datacenter)
          x

Out[5]:   (Wrapper)>[PointerTensor | me:20069769489 -> May Clinic:27535193014]

In [6]:   y = x + x

In [7]:   y

Out[7]:   (Wrapper)>[PointerTensor | me:52194974528 -> May Clinic:13992236415]

In [8]:   y.get()

Out[8]:   tensor([ 2,  6,  8, 10])
```

Tool 1: Remote Execution

Pros:

◆ **RPC:** Data remains on remote machine

Cons:

◆ How can we do good data science without seeing the data?

Top Contributors

Tool 2: Search and Example Data

Tool 2: Search and Example Data

```
In [3]:  grid = GridClient(url="http://data.bighospital.org",
                            username="atrask",
                            password="*********")
```

Connecting to grid... Connected!

Tool 2: Search and Example Data

```
In [3]:  grid = GridClient(url="http://data.bighospital.org",
                            username="atrask",
                            password="*********")

         Connecting to grid... Connected!


In [5]:  diabetes_datasets = grid.search("#diabetes")

         Found 12 results in total.

         Tag Profile:
                 dataset found 12
                 diabetes found 12
                 #diabetes found 12
                 #data found 6
                 #target found 6
```

Tool 2: Search and Example Data

```
Found 12 results in total.

Tag Profile:
        dataset found 12
        diabetes found 12
        #diabetes found 12
        #data found 6
        #target found 6
```

```
In [10]:  dataset = diabetes_datasets[0]
          dataset
```

```
Out[10]:  (Wrapper)>[PointerTensor | me:42698983859 -> andy:47710699917]
                  Tags: #data dataset diabetes #diabetes
                  Shape: torch.Size([73, 10])
                  Description: Diabetes dataset...
```

Tool 2: Search and Example Data

```
In [12]:  print(dataset.description)
```

```
Diabetes dataset
================

Notes
-----

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

Data Set Characteristics:

   :Number of Instances: 442

   :Number of Attributes: First 10 columns are numeric predictive values
```

Tool 2: Search and Example Data

:number of Instances: 442

:Number of Attributes: First 10 columns are numeric predictive values

For more information see:
Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Le
ast Angle Regression," Annals of Statistics (with discussion), 407-499.
(http://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)

```
In [14]:  dataset.sample()
```

```
Out[14]:  tensor([[ 9.0156e-03, -4.4642e-02, -2.2373e-02, -3.2066e-02, -4.9727e-02,
                    -6.8641e-02,  7.8093e-02, -7.0859e-02, -6.2913e-02, -3.8357e-02],
                   [-7.0900e-02, -4.4642e-02,  9.2953e-02,  1.2691e-02,  2.0446e-02,
                     4.2527e-02,  7.7881e-04,  3.5983e-04, -5.4544e-02, -1.0777e-03],
                   [ 2.3546e-02,  5.0680e-02, -3.0996e-02, -5.6706e-03, -1.6704e-02,
                     1.7788e-02, -3.2356e-02, -2.5923e-03, -7.4089e-02, -3.4215e-02],
                   [-5.2738e-02,  5.0680e-02,  3.9062e-02, -4.0099e-02, -5.6968e-03,
                    -1.2900e-02,  1.1824e-02, -3.9493e-02,  1.6305e-02,  3.0644e-03],
                   [ 6.7136e-02, -4.4642e-02, -6.1174e-02, -4.0099e-02, -2.6336e-02,
                    -2.4487e-02,  3.3914e-02, -3.9493e-02, -5.6158e-02, -5.9067e-02],
                   [ 1.7505e-03, -4.4642e-02, -8.3616e-03, -6.4199e-02, -3.8720e-02,
                    -2.4487e-02,  4.4604e-03, -3.9493e-02, -6.4683e-02, -5.4925e-02],
```

Tool 2: Search and Example Data

Pros:

◆ **RPC:** Data remains on remote machine

◆ **Search/Sample:** We feature engineer w/ sample data

Cons:

◆ We can steal data using PointerTensor.get()

Top Contributors

Tool 3: Differential Privacy

Tool 3: Differential Privacy

Canonical DB

Bob: 1
Bill: 0
Sue: 0
John: 1
Joe: 1
Pat: 0
Amy: 1
Alice: 0

- ◆ **Goal:** ensure statistical analysis doesn't compromise privacy

- ◆ **Query:** function(database)

- ◆ **Perfect Privacy:** the output of our query is the same between this database and any identical database with one row removed or replaced

Tool 3: Differential Privacy

Tool 3: Differential Privacy

```
In [4]:  dataset
```

```
Out[4]:  (Wrapper)>[PointerTensor | me:74628800218 -> alice:72083270314]
                Tags: diabetes #data #diabetes dataset
                Shape: torch.Size([73, 10])
                Description: Diabetes dataset...
```

Tool 3: Differential Privacy

```
In [4]:  dataset
```

```
Out[4]:  (Wrapper)>[PointerTensor | me:74628800218 -> alice:72083270314]
                  Tags: diabetes #data #diabetes dataset
                  Shape: torch.Size([73, 10])
                  Description: Diabetes dataset...
```

```
In [5]:  dataset.get()
```

```
---------------------------------------------------------------------------
CannotRequestPrivateData                  Traceback (most recent call last)
<ipython-input-5-c3af7bfad554> in <module>()
      1 # dataset.get()
----> 2 raise CannotRequestPrivateData()

CannotRequestPrivateData: You just requsted a datapoint which is private or which
depends on data which is private. You can only query private data if noise is add
ed.

Use .get(epsilon) to add appropriate noise.
```

Tool 3: Differential Privacy

Description: Diabetes dataset...

```
In [5]: dataset.get()
```

```
---------------------------------------------------------------------------
CannotRequestPrivateData                  Traceback (most recent call last)
<ipython-input-5-c3af7bfad554> in <module>()
      1 # dataset.get()
----> 2 raise CannotRequestPrivateData()

CannotRequestPrivateData: You just requsted a datapoint which is private or which
depends on data which is private. You can only query private data if noise is add
ed.

Use .get(epsilon) to add appropriate noise.
```

```
In [6]: dataset.get(epsilon=0.1)
```

```
Out[6]: tensor([[-0.0891, -0.0446, -0.0418, -0.0194, -0.0662, -0.0743,  0.0081, -0.0395,
           0.0011, -0.0301],
         [ 0.0235,  0.0507, -0.0396, -0.0057, -0.0484, -0.0333,  0.0118, -0.0395,
          -0.1016, -0.0674],
```

Tool 3: Differential Privacy

◆ **Pros:**

◆ **Remote:** Data remains on remote machine

◆ **Search/Sample:** We can feature engineer using toy data

◆ **DP:** formal, rigorous privacy budgeting

◆ **Cons:**

◆ The data is safe, but the model is put at risk!

◆ What if we need to do a join/computation across multiple data owners?

Top Contributors

Tool 4: Secure Multi-Party Computation

Tool 4: Secure Multi-Party Computation

◆ **Definition:** multiple people can combine their private inputs to compute a function, without revealing their inputs to each other.

◆ **Implication:** multiple people can:

SHARE OWNERSHIP OF A NUMBER

Tool 4: Secure Multi-Party Computation

Tool 4: Secure Multi-Party Computation

Tool 4: Secure Multi-Party Computation

Tool 4: Secure Multi-Party Computation

Tool 4: Secure Multi-Party Computation

2

5

3

◆ **Encryption:** neither knows the hidden value

◆ **Shared Governance:** the number can only be used if everyone agrees

Tool 4: Secure Multi-Party Computation

Tool 4: Secure Multi-Party Computation

2

5

3

x

x

2

2

4

6

Tool 4: Secure Multi-Party Computation

Models and datasets are just large collections of numbers which we can encrypt

Tool 4: Secure Multi-Party Computation

```
bob = GridClient("http://bob-cloud.herokuapp.com")
alice = GridClient("http://alice-cloud.herokuapp.com")
theo = GridClient("http://sue-cloud.herokuapp.com")

crypto = GridClient("http://openmined.herokuapp.com")
```

Tool 4: Secure Multi-Party Computation

```
bob = GridClient("http://bob-cloud.herokuapp.com")
alice = GridClient("http://alice-cloud.herokuapp.com")
theo = GridClient("http://sue-cloud.herokuapp.com")

crypto = GridClient("http://openmined.herokuapp.com")
```

```
x = th.tensor([1,2,3,4,5]).share(bob, alice, theo,
                                 crypto_provider=openmined)
x
```

```
(Wrapper)>[AdditiveSharingTensor]
        -> [PointerTensor | me:75100832451 -> bob:61109349352]
        -> [PointerTensor | me:24508960736 -> alice:58174473186]
        -> [PointerTensor | me:23291943380 -> theo:84520473722]
        *crypto provider: openmined*
```

Tool 4: Secure Multi-Party Computation

```
                                        crypto_provider=openmined)
x


(Wrapper)>[AdditiveSharingTensor]
        -> [PointerTensor | me:75100832451 -> bob:61109349352]
        -> [PointerTensor | me:24508960736 -> alice:58174473186]
        -> [PointerTensor | me:23291943380 -> theo:84520473722]
        *crypto provider: openmined*


y = x + x
y


(Wrapper)>[AdditiveSharingTensor]
        -> [PointerTensor | me:61688667118 -> bob:47353472328]
        -> [PointerTensor | me:66053589763 -> alice:2058066939]
        -> [PointerTensor | me:63817030862 -> theo:90586760070]
        *crypto provider: openmined*


y.get()


tensor([ 2,  4,  6,  8, 10])
```

Tool 4: Secure Multi-Party Computation

```
In [9]:   encrypted_model = model.share(bob, alice, theo)

          encrypted_data = data.share(bob, alice, theo)
          encrypted_target = target.share(bob, alice, theo)

In [10]:  encrypted_pred = encrypted_model(encrypted_data)

In [11]:  encrypted_loss = ((encrypted_pred - encrypted_target)**2).sum()

In [12]:  encrypted_loss.backward()
```

Tool 4: Secure Multi-Party Computation

◆ **Pros:**

    ◆ **Remote:** Data remains on remote machine

    ◆ **Search/Sample:** We can feature engineer using toy data

    ◆ **DP:** formal, rigorous privacy budgeting

    ◆ **MPC:** The model can be encrypted during training!

    ◆ **MPC:** We can do joins / functions across data owners!

Top Contributors

IS IT POSSIBLE TO:

answer questions using data we cannot see?

IS IT POSSIBLE TO:

answer questions using data we cannot see?

Tool 1

Remote Execution

Tool 2

Example Data

Tool 3

Differential Privacy

Tool 4

Secure Multi-party Computation

```
atrask:~ pip install the-worlds-data
```

Lets forget these

- ImageNet

- MNIST

- CIFAR-10

- Librispeech

- WikiText-103

- WMT

Lets solve these!

✓ Cancer

✓ Alzheimers

✓ Dementia

✓ Depression

✓ Anxiety

✓ ... the Common Cold?

udacity.com/private-ai

Part 2: AI, Privacy & Society