

L1C9 - Tool 1: Remote Execution

0:03 so in the last several concepts we've
0:05 been learning about the motivations for
0:07 and use case examples of privacy
0:09 enhancing technologies but in this next
0:11 section of this lesson we're going to
0:13 focus instead on the actual technology
0:15 itself and what combination of
0:17 individual privacy and techniques can
0:19 make all this sort of vision possible
0:21 you know such that we can load data into
0:23 a server and other people all around the
0:25 world can use that as a part of their
0:27 distributed data science projects
0:29 and this is really going to be a series
0:31 of tools that we walk through sort of
0:33 building increasing in complexity
0:35 that sort of lead to this this this
0:37 vision of this sort of technical
0:38 ecosystem and the first tool we're going
0:40 to talk about is called remote execution
0:43 and this really does do most of the
0:45 heavy lifting and what we're going to
0:46 see is that remote execution is kind of
0:48 the dominant tool and then as there are
0:49 sort of individual kind of holes or
0:51 missing pieces that remote execution
0:53 leaves behind other more sophisticated
0:55 privacy and techniques sort of fill in
0:57 the gaps in in those areas but
0:59 let's start with mode execution so
1:00 remote execution is just really the idea
1:02 that if you want to study a data set at
1:04 a different organization
1:06 without acquiring a copy
1:08 just don't download the copy like don't
1:11 don't pull the data off of their servers
1:13 use it while it's still inside their
1:15 servers right
1:17 and
1:18 that's it it's a simple idea we just
1:20 need data science tools that can be used
1:22 remotely and
1:24 you know road execution really is mostly
1:26 heavy lifting since like if i want to
1:27 work with data at a thousand different
1:28 hospitals i just leave the data in those
1:30 hospitals and and use sort of standard
1:32 time-tested
1:34 remote procedure call uh based libraries
1:37 to be able to do data science and data
1:38 while while it's it's on a different
1:40 machine
1:41 but this does leave the question of like
1:44 what about when i actually need to get
1:45 my results when they're actually
1:46 something leaving this organization
1:47 right so to me doing data science over
1:49 there is fine but what about when
1:51 information needs to come back well
1:52 let's explore this a little bit closer

ここ数回のコンセプトでは
～の動機について学んでいます
プライバシーの使用例
テクノロジーを強化しますが、次は
このレッスンのこれから行うセクション
代わりに**実際のテクノロジー**に焦点を当てる
それ自体とその組み合わせ
個人のプライバシーと技術は、
このようなビジョンをすべて可能にする
データをロードできることはご存知でしょう
サーバーと周囲の他の人々
世界はそれを自分たちの一部として使用できます
分散型データサイエンスプロジェクト
そしてこれは本当に**シリーズ**になります
私たちが実際に使用するツールの種類
複雑さを増す建物
これがこれに**つながる**ようなもの
この種の技術的なビジョン
エコシステムと私たちが目指す最初のツール
話すことを**リモート実行**と呼びます
そしてこれは**実際にほとんどのことを実現します**
力仕事と私たちがやろうとしていること
リモート実行は一種のことです
主要なツールとその他のツール
一種の個別の穴、または
不足している部分がリモート実行される
他のより洗練されたものを置き去りにする
プライバシーとテクニックを補うようなもの
それらの領域にはギャップがありますが、
モードの実行から始めましょう
リモート実行はまさにアイデアです
データセットを研究したい場合は、
別の組織
コピーを取得せずに
コピーをダウンロードしないでください
サーバーからデータを取得しないでください
まだ**内部にある間にそれを使用してください**
サーバーは正しい
と
それはそれだけです、それは単純なアイデアです
使用できるデータサイエンスツールが必要です
リモートで、そして
道路での執行は実際にはほとんどが
力仕事をするのは好きだから
千の異なるデータを扱う
病院にはデータを残すだけです
病院ではある種の標準を使用しています
実績のある
リモート プロシージャ コール UH ベースのライブラリ
データサイエンスと**データを実行**できること
その間、それは**別の場所**にあります
マシン
しかし、これには次のような疑問が残ります
実際に取得する必要がある場合はどうなりますか
私の結果が実際にそうなったとき
何かがこの組織を離れる
そうです、私はデータサイエンスをやリ直すことになりました
それはいいけど、いつはどうなるの？
情報はきちんと戻ってくる必要がある
これをもう少し詳しく見てみましょう

1:53 so let's say i import sift and
1:56 i i get a individual variable that that
2:00 points towards this remote machine so
2:02 if you're familiar with uh with more
2:04 legacy technologies um um you know
2:07 standard data science tools they all
2:09 have one core primitive which is a
2:11 tensor a list of numbers
2:12 in order to create remote procedure
2:14 calls or in order to to do data science
2:17 in a remote execution environment we
2:18 introduce a second core primitive right
2:20 so not just a list of numbers or a
2:22 tensor or an array type but also this
2:24 this this type of a worker or a
2:26 client right where i have a client
2:29 variable that's pointing towards a
2:31 remote machine and that allows me to be
2:33 able to manipulate my core tensor data
2:35 structure my array data structure
2:37 my matrix data structure on this remote
2:40 machine so coordinate which machines are
2:42 doing what this type of stuff so
2:44 so in this case we have you know a
2:45 virtual worker um and now i can take
2:47 this virtual worker and i can
2:50 send in a tensor so if i construct you
2:52 know a list of numbers one three four
2:54 five and i call dot send it sends that
2:57 tensor into this you know hospital data
2:59 center right what gets returned back to
3:01 me is a pointer to the result right and
3:04 this this result pointer has all the
3:07 same api that a normal array would have
3:09 right in the sense that that um you know
3:12 it sort of looks and feels like a
3:15 a normal array or tensor data structure
3:16 would be however when i actually use it
3:19 you know say y equals x plus x then
3:21 instead of executing locally it sends a
3:23 little message over to a remote machine
3:25 executes there and returns back to me a
3:28 pointer to the result and this pointer
3:30 again is exactly the same idea you know
3:31 it's it's a remote data structure
3:33 um and the the
3:36 rule of thumb here that is introduced in
3:39 this remote execution paradigm but is
3:40 really going to be carried forward
3:42 through through all these different
3:43 tools is we're trying to make it feel
3:46 like for the data scientist that they
3:48 actually have all the data on their
3:50 local machine that it really really
3:51 feels like
3:52 you know they've acquired a copy of all
3:54 that information on their local machine
3:55 even though it might be distributed
3:56 across
3:57 multiple different servers around the
3:58 world
3:59 now
4:01 when we want to look at our results we
4:03 call this method dot get and this method

それで、siftをインポートして、
私はそれを示す**個々の変数**を取得します
このリモートマシンを指しているので、
ああ、もっと詳しく知っているなら
レガシーテクノロジー、ええと、ご存知ですか
標準的なデータサイエンス ツールはすべて
コアプリミティブが 1 つあります。
テンソル数値のリスト
リモートプロシージャを作成するには
電話をかけたり、データサイエンスを行うために
リモート実行環境では、
2番目の中核となる原始権を導入する
したがって、単なる数字のリストや
テンソルまたは配列型だけでなく、これも
これはこのタイプの労働者または
クライアントがいる場所に**クライアント**
を指している変数
リモートマシンのおかげで私は
コアテンソルデータを操作できる
私の配列データ構造を構造化する
このリモート上の私の行列データ構造
マシンなので、どのマシンが配置されているかを調整します
このようなことをしているので
この場合は、
バーチャルワーカー、えーっと、これで取れるようになりました
この**仮想ワーカー**ならできる
テンソルを送って、私があなたを構築したら
1、3、4という数字のリストを知っている
5 と私は dot send を呼び出します、それが送信されます
これにテンソルを適用すると、病院のデータがわかります
中央右に**返されるもの**
私は**結果へのポインタ**であり、
この結果ポインタにはすべてのものが含まれています
通常の配列が持つ API と同じ
そうですね、それはあなたが知っているという意味で
それはある種の見た目と感触です
通常の配列またはテンソル データ構造
ただし、実際に使用するとそうなります
y が x に x を加えたものに等しいと言うのは知っていますよね
ローカルで実行する代わりに、
リモートマシンへの小さなメッセージ
そこで実行され、私に返されます
結果へのポインタとこのポインタ
これもまたあなたが知っているのとまったく同じ考えです
それは**リモートデータ構造**です
ええと、そして、
で紹介されている経験則はこちら
この**リモート実行パラダイム**は、
本当に引き継がれます
これらすべての異なることを通して
ツールとは、私たちがそれを感じさせようとしているものです
データサイエンティストの場合と同様に、
実際にはすべてのデータを持っています
ローカルマシンそれは本当に本当に
のように感じている
彼らはすべてのコピーを入手したことを知っています
ローカルマシン上のその情報
配布されるかも知れませんが
横切って
周囲の複数の異なるサーバー
世界
今
結果を確認したいときは、
この**メソッド dot get を呼び出して、このメソッドを呼び出します**

4:07 is really where the bulk of our sort of
4:09 security and policy happens right so
4:11 sort of remotely controlling a data
4:13 structure it doesn't necessarily leak
4:14 any information but it's really this
4:16 this dot get method that we're going to
4:17 come back to in a little while that is
4:18 really where a lot of more sophisticated
4:20 things start to happen so what have we
4:21 done so far so pros remote procedure
4:23 road seater calls allow data to remain
4:25 in the remote machine however this sort
4:27 of opens up this question of like how do
4:29 we actually do good data science without
4:31 seeing the data like so viscerally
4:34 seeing data is a big part of the the
4:36 data science workflow and in the next
4:38 concept we're going to walk through
4:39 another tool that tries to overcome this
4:41 challenge i'll see you then

英語 (自動生成)

ここは実際、私たちの種類の大部分が存在する場所です
セキュリティとポリシーは適切に行われるため、
データを遠隔操作するようなもの
構造上必ずしも漏れないとは限りません
何か情報はあっても本当にこれだ
これから行うこの **dot get メソッド**
しばらくしてから戻ってきてください
実際には、より洗練されたものがたくさんあります
物事が起こり始めたので、私たちはどうしたのですか
これまでのところ**完了した**ので、プロのリモート手順
ロードシート通話によりデータを維持できる
ただし、リモートマシンではこのような
のような質問は、**どうやって行うか**を開きます
私たちは実際に**優れたデータサイエンス**を行っています。
データを非常に直感的に見ることができる
データの確認は、
データサイエンス ワークフローと次回
これから説明するコンセプト
これを克服しようとする別のツール
挑戦してください、それでは会いましょう



OpenMined

<https://www.youtube.com/watch?v=zIEMtYdWllw>