
通信・計算機システムと 知的情報処理 (3/4)

情報通信系

工学リテラシー

資料作成
篠崎、中原、上垣外
およびTAメンバー

取り組み内容

- 1回目
 - ニューラルネットの原理
 - TSUBAMEアカウント作成
- 2回目
 - ニューラルネット課題レシピの説明
 - スーパーコンピュータTSUBAME3.0の利用法
- 3回目
 - 課題レシピの動作確認
 - 課題レシピの改造
- 4回目
 - TSUBAME上での実験の続き
 - レポートの作成

課題レシピの動作確認

ssh接続・レシピのダウンロード

以下のコマンドを順に入力 (＿はスペース)

(TSUBAMEにログイン. 学籍番号は自分の学籍番号)

```
$ ssh_学籍番号@login.t3.gsic.titech.ac.jp
```

(レシピのダウンロード(clone). いずれか一つ)

- 音声認識レシピ

```
$ git_clone_https://github.com/tttslab/tut-asr-voicecommand.git
```

- 画像認識レシピ

```
$ git_clone_https://github.com/HirokiNakahara/MNIST_Examples.git
```

- 機械翻訳レシピ

```
$ git_clone_https://github.com/kamigaito/tga-egliteracy-seq2seq.git
```

学習・評価を実行

(先程cloneしたディレクトリに移動)

\$ **cd** _<directory of the recipe> ディレクトリ名は **ls** コマンドで調べる

```
@login1:~> cd tut-asr-voicecommand/  
@login1:~/tut-asr-voicecommand> ls  
LICENSE  README.md  exp100p  exp1p  exp20p  exp60p  requirements.txt  
@login1:~/tut-asr-voicecommand> █
```

<directory of the recipe> は選んだタスクに応じて次から選択:

音声認識

tut-asr-voicecommand

画像認識

MNIST_Examples

機械翻訳

tga-egliteracy-seq2seq

(動作確認用のディレクトリ(exp1p)に移動)

\$ **cd** _exp1p

(TSUBAMEでジョブを実行)

\$ **qsub** _runsubame.sh



ジョブの状態は **qstat** コマンドで確認できる

学習, 評価が終わるまで待つ

(動作確認用のジョブで, 約3分ほどで学習と評価が終わる)

ssh接続・学習・評価までのまとめ

```

[redacted]:Documents Naoto$ ssh [redacted]@login.t3.gsic.titech.ac.jp
Last login: Wed May 29 20:28:31 2019 from [redacted]

-----
Last modified: Wed May 28 13:15:00 JST 2019

*** Do not run programs with high load average such as ISV ***
*** on login0 and 1.                                         ***

(The current TSUBAME 3.0 operational status)
http://www.t3.gsic.titech.ac.jp/
-----

[redacted]@login1:~> git clone https://github.com/tttslab/tut-asr-voicecommand.git
Cloning into 'tut-asr-voicecommand'...
remote: Enumerating objects: 180, done.
remote: Counting objects: 100% (180/180), done.
remote: Compressing objects: 100% (107/107), done.
remote: Total 180 (delta 104), reused 139 (delta 73), pack-reused 0
Receiving objects: 100% (180/180), 33.55 KiB | 0 bytes/s, done.
Resolving deltas: 100% (104/104), done.
Checking connectivity... done.
[redacted]@login1:~> cd tut-asr-voicecommand/
[redacted]@login1:~/tut-asr-voicecommand> ls
LICENSE  README.md  exp100p  exp1p  exp20p  exp60p  requirements.txt
[redacted]@login1:~/tut-asr-voicecommand> cd exp1p/
[redacted]@login1:~/tut-asr-voicecommand/exp1p> ls
colab.ipynb      eval.py  mkgraph.py  models.py  preprocess.py  runsubame.sh  train.sh
data_download.sh eval.sh   mkgraph.sh  prep.sh    run.sh         train.py      utils.py
[redacted]@login1:~/tut-asr-voicecommand/exp1p> qsub runsubame.sh
Your job 4731955 ("runsubame.sh") has been submitted
[redacted]@login1:~/tut-asr-voicecommand/exp1p> █
```

ssh接続してからジョブを実行するまでの画面例

評価結果の確認

以下のコマンドを入力

\$ **less** ~/<directory of the recipe>/exp1p/score.txt

less はファイルの中身を表示

ディレクトリ名はlsコマンドで調べる

```
recognition rate = 53.9%  
/home/1 /score.txt lines 1-1/1 (END)
```

この例では認識率は53.9%
数値は、乱数初期化や計算誤差などにより異なる

キーボードの "q" を押すと、もとの画面に戻る

動作確認版の精度

おおよそ下記の精度が得られる

差が数%を超えて大きく異なるときや、そもそも数字が出力されない場合、何か問題がある可能性あり

- 音声認識レシピ
 - 認識率 53.9%
- 画像認識レシピ
 - 認識率 81.6%
- 機械翻訳レシピ
 - BLEUスコア 0.93

レシピの構成（各レシピ共通）

runsubame.sh

TSUBAME上でジョブを走らせる準備
スクリプトrun.shを起動

→ run.sh

ニューラルネット実験のトップレベルのスクリプト

→ prep.sh

学習・評価データの準備
(TSUBAMEでは準備済みなのでスキップ)

→ train.sh

ニューラルネットの学習

→ eval.sh

ニューラルネットの評価

学習データ量の変更

学習データ量を変えて、学習時間や性能の変化を調べる
3種類の学習データ量(動作確認用を除く)で実験

実験のディレクトリ構成

<directory of the recipe>/

exp1p/

全体の1%のデータ(動作確認用)

exp20p/

全体の20%のデータ

exp60p/

全体の60%のデータ

exp100p/

全体の100%のデータ

<directory of the recipe>は、
音声認識レシピを選択した場合
tut-asr-voicecommand

画像認識レシピを選択した場合
MNIST_Examples

機械翻訳レシピを選択した場合
tga-egliteracy-seq2seq

学習データ量の変更 (全体の20%使用)

(ホームディレクトリに戻る)

```
$ cd ~/
```

(最初にcloneしたディレクトリに移動)

```
$ cd <directory of the recipe>
```

ディレクトリ名は `ls` コマンドで調べる

(全体の20%のデータを使用する実験のディレクトリに移動)

```
$ cd exp20p
```

```
@login1:~> cd ~/
@login1:~> cd tut-asr-voicecommand/
@login1:~/tut-asr-voicecommand> ls
LICENSE  README.md  exp100p  exp1p  exp20p  exp60p  requirements.txt
@login1:~/tut-asr-voicecommand> cd exp20p/
@login1:~/tut-asr-voicecommand/exp20p> ls
colab.ipynb      eval.sh      models.py    run.sh       train.sh
data_download.sh mkgraph.py  prep.sh      runtsubame.sh  utils.py
eval.py          mkgraph.sh  preprocess.py train.py
@login1:~/tut-asr-voicecommand/exp20p>
```

データ量を変更する例

課金グループの指定(各レシピ共通)

- 10分以上計算時間のかかるジョブ
- ⇒TSUBAMEポイントが必要
 - ✓ 課金グループを指定する必要あり
 - ✓ ジョブの最大実行時間を指定

無駄に長く設定すると無駄にポイントを消費するので注意
本実習では、どの場合も6時間あれば足りる

```
$ qsub -g tga-egliteracy -l h_rt=06:00:00 runsubame.sh
```

-g tga-egliteracy : 課金グループ(お財布)指定
-l h_rt=06:00:00 : ジョブの最大実行時間 (hh:mm:ss)指定
-p -4 : ジョブの優先度指定(すこし高めでポイントを多く消費)
runsubame.sh : バッチジョブスクリプト

音声認識の場合の例

```
$ cd tut-asr-voicecommand/exp1p/
```

```
$ qsub -g tga-egliteracy -l h_rt=06:00:00 runsubame.sh
```

音声認識の場合の例

```
$ qsub -g tga-egliteracy -l h_rt=06:00:00 runsubame.sh
```



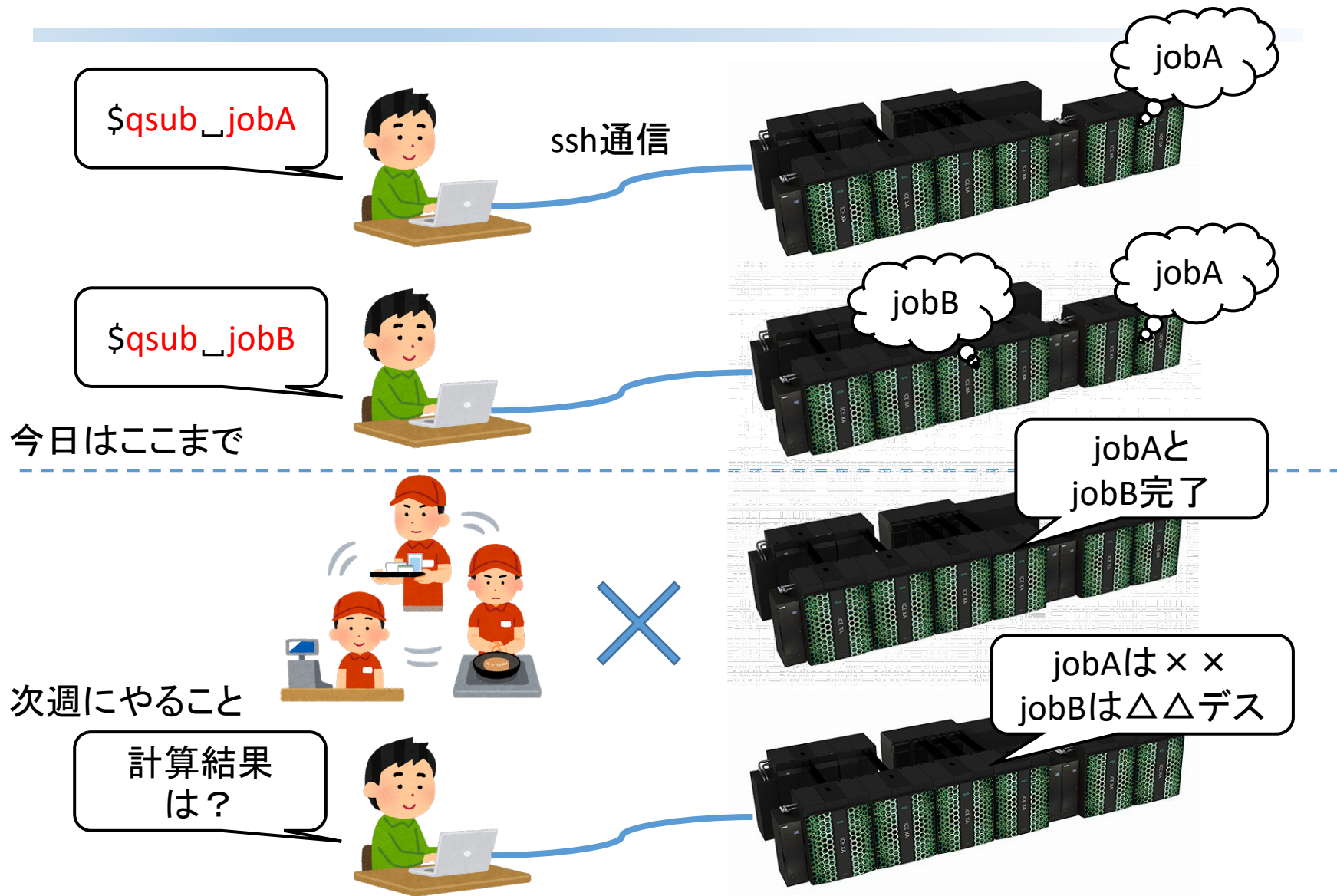
学習・評価が終わるまでしばらくかかる
途中でログアウトしても大丈夫



ジョブの終了後、評価結果を確認

```
$ less ~/tut-asr-voicecommand/exp20p/score.txt
```

バッチジョブ投入イメージ



学習データ量の変更 (60%, 100%)

60%の実験(音声認識の場合の例)

```
$ cd ~/
$ cd tut-asr-voicecommand
$ cd exp60p
$ qsub -g tga-egliteracy -l h_rt=06:00:00 runsubame.sh
```

課金サービス
を利用している
ことに留意

100%の実験(音声認識の場合の例)

```
$ cd ~/
$ cd tut-asr-voicecommand
$ cd exp100p
$ qsub -g tga-egliteracy -l h_rt=06:00:00 runsubame.sh
```



それぞれの実験は、前の実験のジョブを投入した後、その終了を待たずに進めてよい

参考: 投入したジョブの確認とキャンセル

- **qstat**コマンドで、ジョブの状態を確認できます

```
login0:tut-asr-voicecommand $ qsub runsubame.sh
Your job 4616952 ("runsubame.sh") has been submitted
login0:tut-asr-voicecommand $ qstat
```

job-ID	prior	name	user	state	submit/start at	queue	jclass	slots	ja-task-ID
4616952	0.55256	runsubame	shinozaki-t-	r	04/25/2019 12:14:38	all.q@r1i0n7		7	

Job ID

Jobの状態 (r: running)

- 投入したジョブを中止したい場合は、**qdel**コマンドを用います

キャンセル・中断したいジョブの
Job IDを指定

```
login0:tut-asr-voicecommand $ qdel 4616952
shinozaki-t-ab has registered the job 4616952 for deletion
```

参考：各レシピの学習データ量

- 音声認識レシピ
 - 30769サンプル
- 画像認識レシピ
 - 60000サンプル
- 機械翻訳レシピ
 - 329882サンプル

学習セットを100%用いた場合
デフォルトの動作確認版は1%を用いるので、こちらの100分の1

演習課題 (ICT-3)

- 別ファイル参照

課題提出方法

- 締切: 本日より6日後の朝9時(OCW-iにて指定)
- 提出先: TOKYO TECH OCW-i
- ファイル種別: Text (.txt) file
 - ✓ **NOT** docx, rtf, pdf, png, jpg, etc.
- 投稿タイトル: ICT-3
- フォーマット (**厳格に下記のフォーマットであること**)

```
19B34567, 工大 太郎
Q3.1: No
Q3.2: No
Q3.3: No
```



1行目: 自分の学籍番号と名前

2行目~: 問題ID, ':', 回答(Yes or No)

こちらの答えは(もちろん)ダミー

テンプレートをOCWにアップロードしてあるので、そちらをダウンロードして編集してください

次回予告

- 実験の続きと最終レポートの作成
 - 結果を取りまとめたグラフを作成します
- データ量を増やした実験の実行には最大6時間必要
 - 今日のうちにジョブ投入を終えていない場合、
授業時間外にTSUBAMEにアクセスする必要があるので注意
- 最終レポート締め切りは、最終回6日後の13時の予定
(次回の指示に従ってください)

注意：VLSI設計室のアカウントについて

- 本クォーターの終了後、本講義で使用したVLSI室のアカウントおよびユーザーディレクトリは使用できなくなります
- 本クォーター終了後も必要なデータがある場合は、各自でバックアップしてください