

---

# 通信・計算機システムと 知的情報処理

(4/4)

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

# 結果を整理

sshでTSUBAMEにログイン後, 実験ディレクトリに入る

自分が前回選択したものに依じて, 以下のコマンドを実行する  
(\_はスペース)

\$ cd \_tut-asr-voicecommand/exp1p (音声認識の場合)

\$ cd \_MNIST\_Examples/exp1p (画像認識の場合)

\$ cd \_tga-egliteracy-seq2seq/exp1p (機械翻訳の場合)

```
@login1:~> cd MNIST_Examples/exp1p
@login1:~/MNIST_Examples/exp1p> ls
README.md  mnist.model  runsubame.sh  train.sh
eval.log   model       runsubame.sh.e4716010  train.time.log
eval.py    prep.py     runsubame.sh.o4716010  train_on_GoogleCoLab.ipynb
eval.sh    prep.sh     score.txt
mkgraph.py requirement.txt  train.log
mkgraph.sh run.sh      train.py
@login1:~/MNIST_Examples/exp1p> 
```

# 結果を整理

結果を確認して、メモする

\$ `less _score.txt`

他のデータ量に対しても、同様の操作を行う

<directory of the recipe>/

- exp1p/score.txt
- exp20p/score.txt
- exp60p/score.txt
- exp100p/score.txt

e.g.

Data amount	1%	20%	60%	100%
Accuracy	10.0%	20.0%	30.0%	40.0%

これらの値はダミー。自分の実験で得られた値を調べること！！！！

# 各データ量の集計

以下のコマンドで各ディレクトリへ移動し `less _score.txt` で結果を確認  
音声認識の場合

```
$ cd _~/tut-asr-voicecommand/exp20p
```

```
$ cd _~/tut-asr-voicecommand/exp60p
```

```
$ cd _~/tut-asr-voicecommand/exp100p
```

画像認識の場合

```
$ cd _~/MNIST_Examples/exp20p
```

```
$ cd _~/MNIST_Examples /exp60p
```

```
$ cd _~/MNIST_Examples /exp100p
```

機械翻訳の場合

```
$ cd _~/tga-egliteracy-seq2seq/exp20p
```

```
$ cd _~/tga-egliteracy-seq2seq/exp60p
```

```
$ cd _~/tga-egliteracy-seq2seq/exp100p
```

# グラフ作成

- レシピ内にある、**mkgraph.sh**を使用
- 使い方(例)

↓                      ↓                      ↓                      ↓

x軸ラベル      x数値      y軸ラベル      y数値

```
$/mkgraph.sh 'train(%)' '1_20_60_100' 'accuracy(%)' '10.0_20.0_30.0_40.0' 'result.png'
```

出力ファイル名

※ xの個数はyの個数と等しい

※ 入力内容を' 'で囲む

※ y数値は、自分の実験で得られたデータを入力

score.txtの内容

日本語キーボード

shift-7 : シングルクォート「'」

shift-@ : バッククォート「`」

# グラフを作る (step 1)

選択したレシピに応じて, **mkgraph.sh**のあるディレクトリに移動

\$ **cd** ~/tut-asr-voicecommand/exp1p (音声認識の場合)

\$ **cd** ~/MNIST\_Examples/exp1p (画像認識の場合)

\$ **cd** ~/tga-egliteracy-seq2seq/exp1p (機械翻訳の場合)

```
@login1:~> cd MNIST_Examples/exp1p
@login1:~/MNIST_Examples/exp1p> ls
README.md      mnist.model    runtsubame.sh      train.sh
eval.log       model          runtsubame.sh.e4716010  train.time.log
eval.py        prep.py       runtsubame.sh.o4716010  train_on_GoogleCoLab.ipynb
eval.sh        prep.sh       score.txt
mkgraph.py     requirement.txt  train.log
mkgraph.sh     run.sh        train.py
@login1:~/MNIST_Examples/exp1p>
```

画像認識の場合の例

exp20p, exp60p, exp100pの中にも同じ**mkgraph.sh**が入っている(どれを用いてもよい)

# グラフを作る (step 2)

## mkgraph.sh でグラフを作る

```
$ ./mkgraph.sh 'train(%)' '1_20_60_100' 'accuracy(%)' '10.0_20.0_30.0_40.0' 'result.png'
```

出力する画像ファイル名 (グラフが描画される)

exp1p exp20p exp60p exp100p  
それぞれのscore.txtの内容

```
@login1:~/MNIST_Examples/exp1p> ./mkgraph.sh 'train(%)' '1 20 60 100' 'accuracy(%)' '10.0 20.0 30.0 40.0' 'result.png'
figure result.png saved
@login1:~/MNIST_Examples/exp1p> ls
README.md  mkgraph.py  prep.py      run.sh        score.txt    train.time.log
eval.log   mkgraph.sh  prep.sh      runsubame.sh  train.log    train_on_GoogleCoLab.ipynb
eval.py    mnist.model requirement.txt runsubame.sh.e4716010 train.py
eval.sh    model       result.png   runsubame.sh.o4716010 train.sh
@login1:~/MNIST_Examples/exp1p>
```

mkgraph.shの実行例

コマンド実行後, 新しく  
result.png が作成される



# グラフを作る (step 3)

result.png をhomeディレクトリに移動する

```
$ mv result.png ~/
```

※ mv : ファイル等を移動するコマンド.

```
login1:~/MNIST_Examples/exp1p> mv result.png ~/
login1:~/MNIST_Examples/exp1p> ls
README.md  mkgraph.py  prep.py      runsubame.sh  train.log    train_on_GoogleCoLab.ipynb
eval.log   mkgraph.sh  prep.sh      runsubame.sh.e4716010  train.py
eval.py    mnist.model requirement.txt runsubame.sh.o4716010  train.sh
eval.sh    model       run.sh       score.txt     train.time.log
```

mvコマンドの実行例

コマンド実行後, result.png は  
実験ディレクトリからはなくなり,  
ホームディレクトリに移動  
されます

# TSUBAMEから result.png をコピーする

ローカルで新規にターミナルを開き、以下のコマンドを入力

\$ **scp** **\_学籍番号** **@login.t3.gsic.titech.ac.jp:~/result.png** **./**

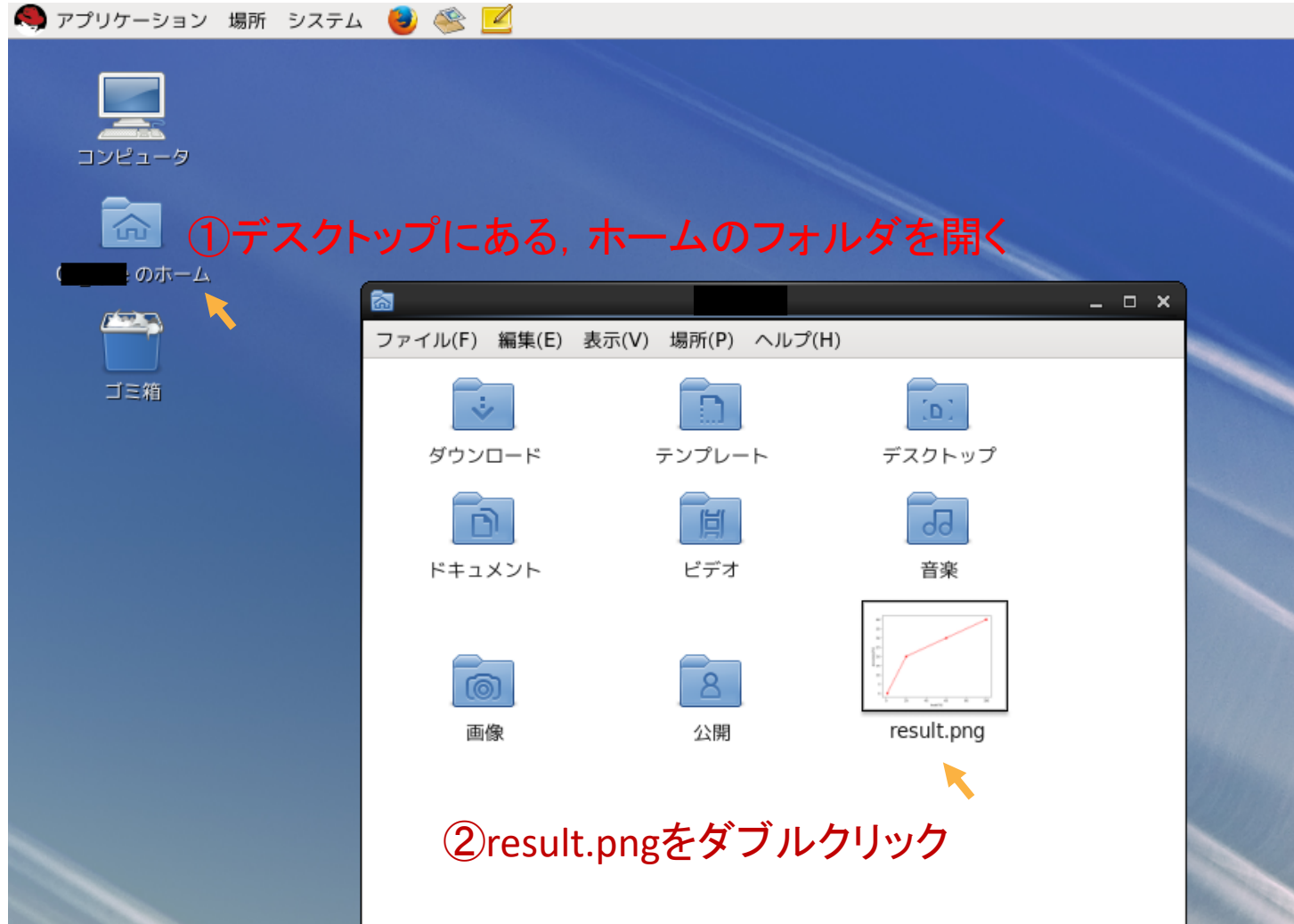


```
ファイル(F) 編集(E) 表示(V) 検索 (S) 端末(T) ヘルプ(H)
bash-4.1$ scp [REDACTED]@login.t3.gsic.titech.ac.jp:~/result.png ./
result.png                               100%  19KB  19.1KB/s   00:00
bash-4.1$ █
```

※ **scp** : 異なるマシン間でファイル等をコピーするコマンド

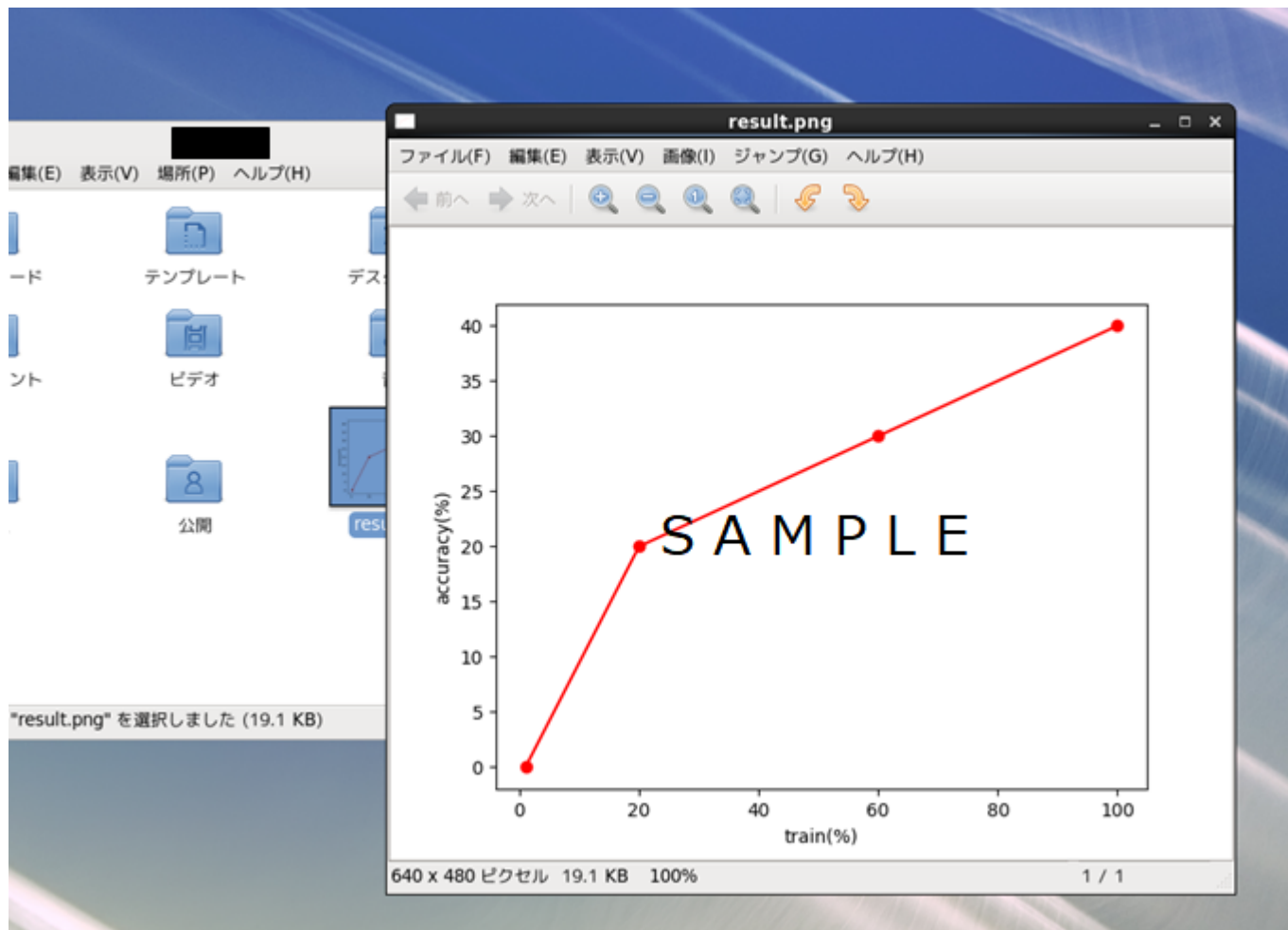
# グラフの画像を開く

グラフの画像ファイル: result.png



# グラフを表示して結果を確認する

(下の画像は実際の値とは異なります)



# 計算時間のグラフ作成

計算時間をtrain.time.log で確認。(※秒で計算時間が表示される)

```
$ less train.time.log
```

それぞれのデータ量での結果を整理

```
<directory of the recipe>/  
├── exp1p/train.time.log  
├── exp20p/train.time.log  
├── exp60p/train.time.log  
└── exp100p/train.time.log
```

mkgraph.shを用いて計算時間のグラフを作る

```
$ ./mkgraph.sh 'data size(%)' 'x_x_x_x' 'training time(sec)' 'y_y_y_y' 'time.png'
```

(※数値は実際のデータを入力！)

以下, scoreと同様にグラフをローカルにコピー, 結果を確認

# 計算時間の集計

以下のコマンドで各ディレクトリへ移動し `less _train.time.log` で確認  
音声認識の場合

```
$ cd _~/tut-asr-voicecommand/exp20p
```

```
$ cd _~/tut-asr-voicecommand/exp60p
```

```
$ cd _~/tut-asr-voicecommand/exp100p
```

画像認識の場合

```
$ cd _~/MNIST_Examples/exp20p
```

```
$ cd _~/MNIST_Examples/exp60p
```

```
$ cd _~/MNIST_Examples/exp100p
```

機械翻訳の場合

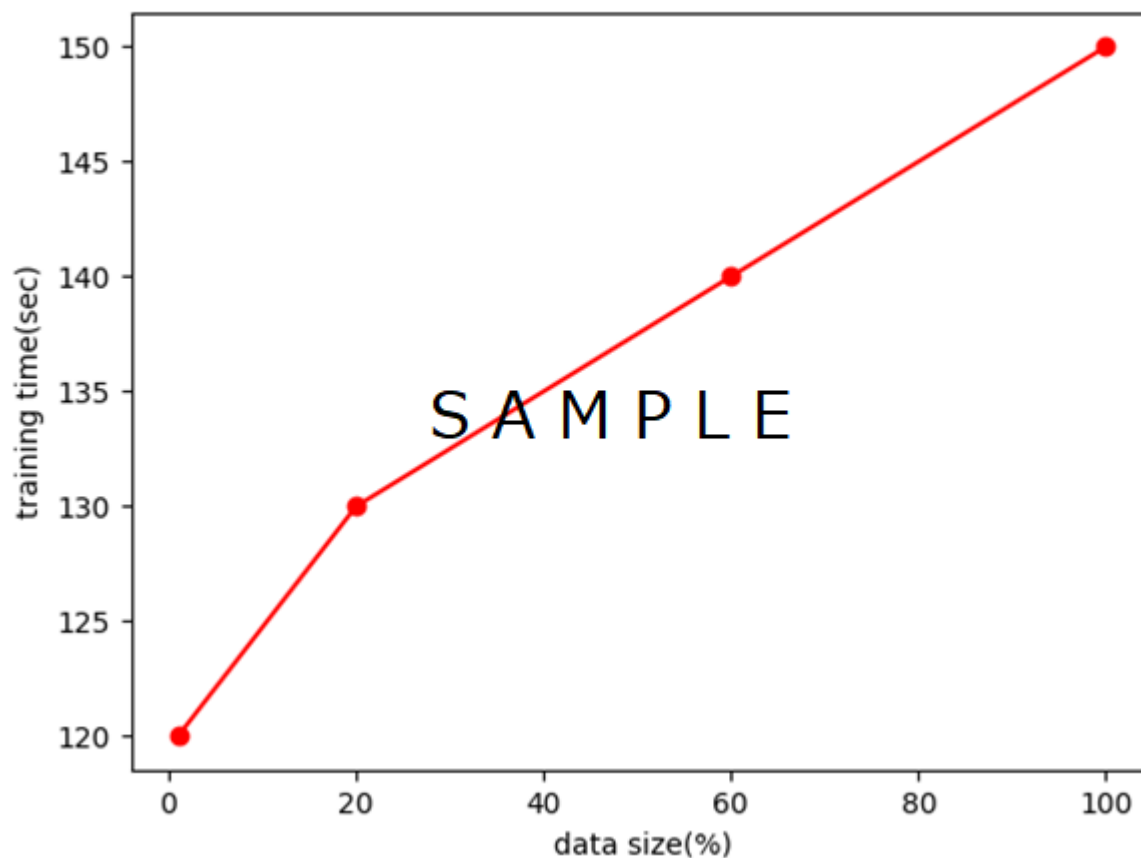
```
$ cd _~/tga-egliteracy-seq2seq/exp20p
```

```
$ cd _~/tga-egliteracy-seq2seq/exp60p
```

```
$ cd _~/tga-egliteracy-seq2seq/exp100p
```

# 時間をまとめたグラフの例

time.png (実際の値とは異なります)



# 最終レポート課題(ICT-4)

---

## Q4.0:

選択したレシピの種類を記述せよ{音声認識|画像認識|機械翻訳}

## Q4.1:

学習データ量と性能の関係を表すグラフresult.pngを提出するとともに、グラフから分かることを簡潔に説明せよ

## Q4.2:

学習データ量と計算時間の関係を表すグラフtime.pngを提出するとともに、グラフから分かることを簡潔に説明せよ



# 課題提出方法

- 締切: 本日より6日後の昼12時(OCW-iにて指定)
- 提出先: TOKYO TECH OCW-i
- 投稿タイトル: ファイル名 (フォーマット)
  - result : result.png (PNG画像ファイル)
  - time : time.png (PNG画像ファイル)
  - ICT-4 : ICT-4.txt (Text (.txt) file)
- 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  
Q4.0: 音声認識  
Q4.1: グラフから、○○○○○○  
○○○○○○○○○○○○○○○○  
○であることが分かる  
Q4.2: グラフから、○○○○○○  
○○○○○○○○○○○○○○○○  
○であることが分かる

1行目: 自分の学籍番号と名前



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

質問等の連絡先

[kamigaito@lr.pi.titech.ac.jp](mailto:kamigaito@lr.pi.titech.ac.jp)

# もしも、実験が最後まで終わらなかったら

- ① 授業時間外に自分で進める
- ② できたところまでの結果をまとめて、とにかく提出する

途中までの結果を用いてグラフを作成する例:

```
$ ./mkgraph.sh _'train(%)' _'1_20_60' _'accuracy(%)' _'10.0_20.0_30.0' _'result.png'
```

exp1p exp20p exp60p  
それぞれのscore.txtの内容