
通信・計算機システムと 知的情報処理 (0/4)

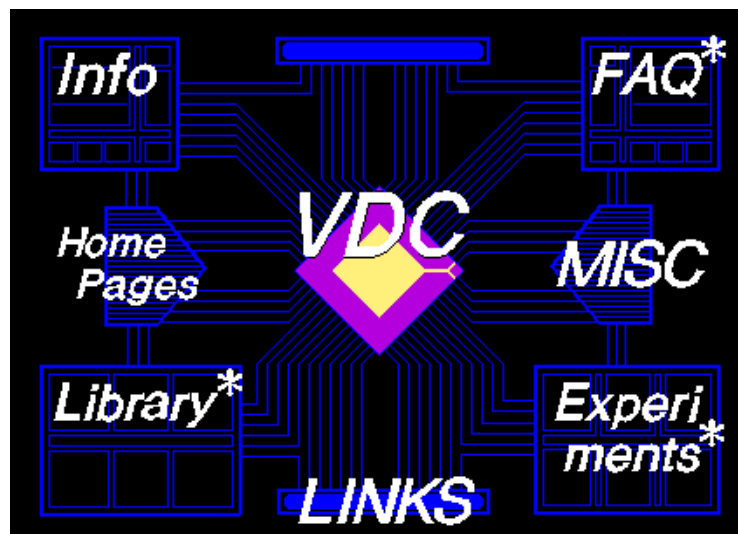
情報通信系

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

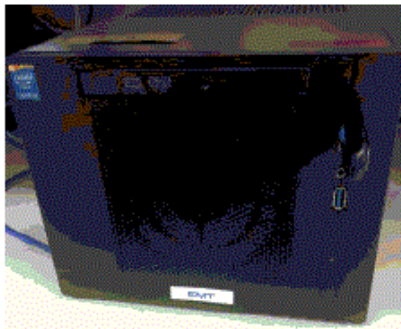
VLSI設計室の利用について

- 土足厳禁、下足入れ番号とスリッパ番号は一致
- 本講義の時間外も、他の講義の時間外は利用可
- 利用については、VLSI設計室のルールに従うこと
 - <http://www.vdc.ict.e.titech.ac.jp/index.html>



VLSI設計室の計算機の起動方法

- 「工学リテラシ」で利用する計算機は机の下に本体
 - 利用にはアカウントが必要
 - 学生証を見せて、アカウント名とパスワードを受け取る
 - Linux(ネットワークブート)を用いる
 - いくつかの機種が混在



ASUSのPC



DELLのPC

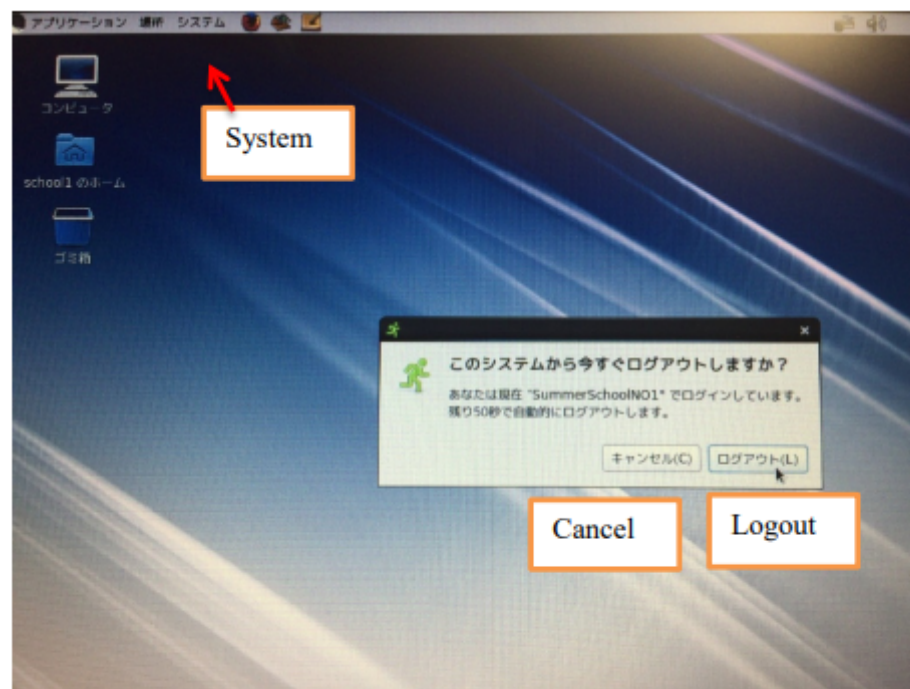


DELLのPC

- 自分のノートパソコン等を使用してもよい
 - キャンパス無線LANに接続し、東工大ポータルへのアクセスが必要
 - win/mac/linuxいずれでもよい
 - サポートはできないので自己責任で使用する

VLSI設計室の計算機の終了方法

1. 画面左上の「システム」を選択（左クリック）
2. 「ログアウト」を選択（左クリック）してログアウト
3. 電源ボタンを押して計算機をシャットダウン

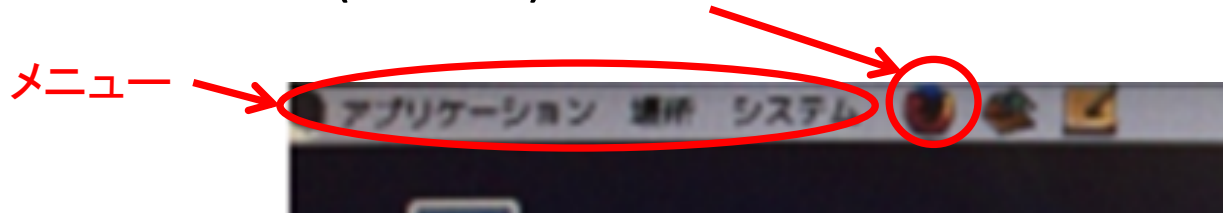


Logout: Click "System" at the top left of the screen, and then click "Logout"

ブラウザ, 端末の起動方法

- ブラウザの起動方法

- 画面左上のメニューから選ぶ(左クリック)、または
- ブラウザ(Firefox)アイコンを左クリック



- Linux「端末」の起動方法

- 画面左上のメニューの中から選ぶ, または
- デスクトップ上で右クリック
 - 表示されるメニューから「端末」を選ぶ

- PDFファイルの閲覧方法

- Evince (Document Viewer)を起動

VLSI設計室での講義資料の見方

- OCWi またはOCWで本講義のページを開く
- OCW/OCWiから参照したいPDFをダウンロード

OCWi



OCW

開講元	工学院
担当教員名	各 教員 青木 洋貴 高橋 篤司 篠崎 隆宏 中田 和秀 吉村 奈津江 中原 啓貴 上垣外 英嗣
授業形態	講義 / 演習
曜日・時限(講義室)	金5-6(<前半>, 南2号館3階VLSI設計室, <後半>, 西9号館3階311号室)
クラス	e
科目コード	XEG.B101
開講年度	H31年度
シラバス更新日	H31年3月19日
使用言語	日本語
単位数	1
開講クォーター	1Q
講義資料更新日	H31年4月17日
アクセスランキング	★★★★★

シラバス 講義ノート ユーザアンケート

第1回 ニューラルネットの原理

★★★★★ 添付資料(2715KB)

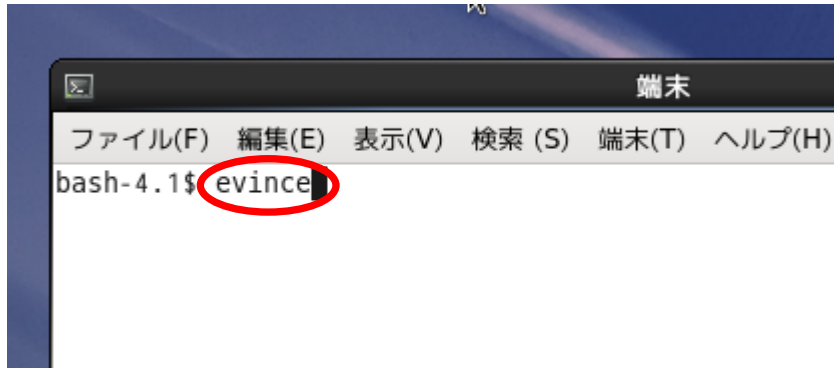
OCWの提出方法の説明を加えました。

ニューラルネットの学習と演習課題にある微分の関係の説明を加えました。

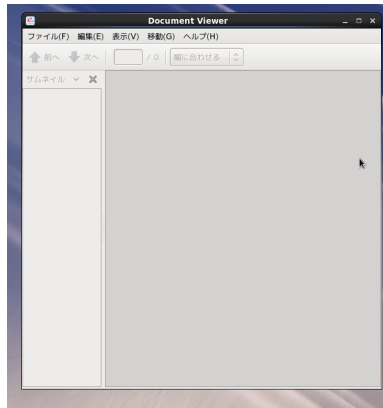
平成31年04月12日(金) 5-6時限開講

Evince (Document Viewer)の起動

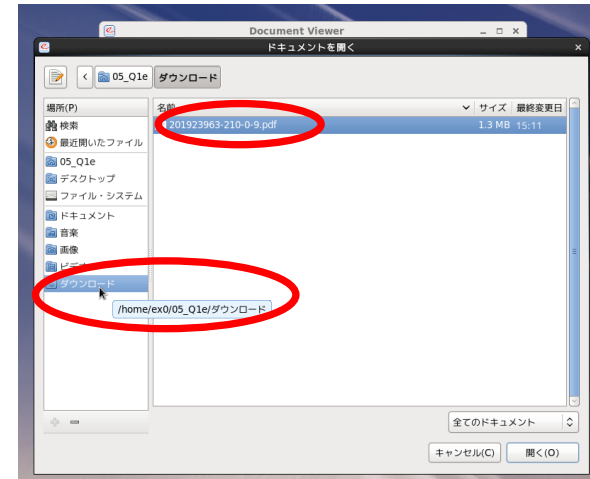
- ① 「端末」でevinceと入力、Enterキーを押す



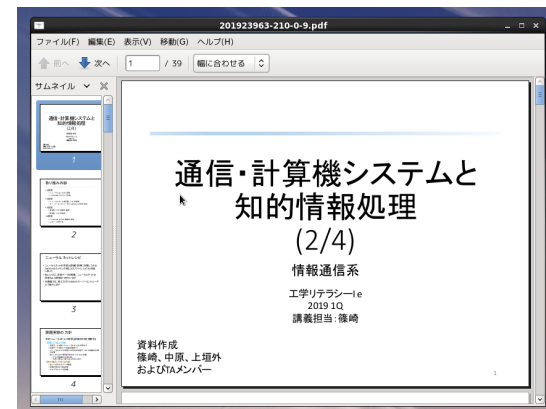
- ② Evince (Document Viewer)が起動



- ③ ファイル(PDF)を選択



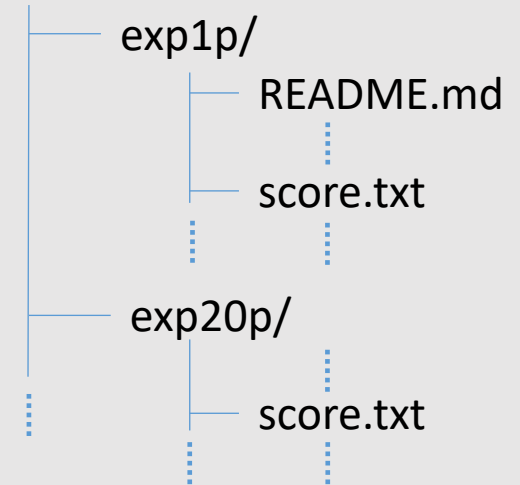
- ④ ファイル(PDF)が開く



Linux (Unix系OSカーネル)

- コマンド(命令)を入力
- 階層型ディレクトリ構造(ディレクトリ=フォルダ)
 - ルートディレクトリ `/`
 - ホームディレクトリ `~/`
 - 最初の作業ディレクトリ
 - カレントディレクトリ `./`
 - 現在の作業ディレクトリ
 - 親ディレクトリ `../`
- ディレクトリの指定
 - ルートからの絶対パス
 - カレントディレクトリからの相対パス
- プロンプト(コマンドの入力位置を示す)
 - カレントディレクトリを表示
 - 設定によって表示は異なる

~/MNIST_Examples/



```
19M1 [redacted]@login1:~> cd MNIST_Examples/exp1p
19M1 [redacted]@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
19M1 [redacted]@login1:~/MNIST_Examples/exp1p> █
```

カレントディレクト

Linuxの基本的なコマンド1

- ls (ディレクトリ名、ファイル名の表示)(list)

\$ ls : カレントディレクトリの表示

\$ ls ~/ : ホームディレクトリの表示

\$ ls .. : 親ディレクトリの表示

\$ ls path : pathで指定のディレクトリの表示

✓ pathはルートからの絶対パス、または、カレントディレクトリからの相対パス

スペース(空白)は「_」で表示

・ 引数、オプションの区切り

- echo (表示)

\$ echo text : textを表示

- pwd (カレントディレクトリのパス表示)(print working directory)

\$ pwd

- cd (カレントディレクトリの変更)(change directory)

\$ cd ~/ : ホームディレクトリへ

\$ cd .. : 親ディレクトリへ

\$ cd path : pathで指定のディレクトリへ

- mkdir (ディレクトリの作成)(make directory)

\$ mkdir dirname : dirnameの名前のディレクトリ作成

✓ dirnameで作成するディレクトリ位置も指定可能

Linuxの基本的なコマンド2

- cat (ファイルの連結、中身の表示) (concatenate)

\$ **cat** **_filename** : **filename**のファイルの中身の表示
✓ **filename**でファイルのディレクトリ指定も可能

- less (ファイルの中身の表示)

\$ **less** **_filename** : **filename**のファイルの中身の表示
✓ "q" を押す終了

- cp (ファイルのコピー) (copy)

- mv (ファイルの移動) (move)

\$ **cp** **_source** **_destination** : **source**ファイルを**destination**にコピー
\$ **mv** **_source** **_destination** : **source**ファイルを**destination**に移動
✓ カレントディレクトリのファイルの場合、ディレクトリ(へのパス)は省略可
✓ ファイル名を変更しない場合、**destination**でファイル名は省略可

- リモート計算機の利用

- ssh (リモート計算機へログイン) (secure shell)

\$ **ssh** **_ID@login.t3.gsic.titech.ac.jp** : TSUBAMEへのログイン

- scp (リモート計算機からファイルをコピー) (secure copy)

\$ **scp** **_ID@machine:source** **_destination** : リモートマシンの**source**ファイルを**destination**にコピー

- ssh-keygen (秘密鍵公開鍵の生成)

\$ **ssh-keygen** **_t_rsa**

Shell (コマンドインタプリタ) の機能

- 「Tab」キー
 - コマンドやファイル名の補完
 - 一意に定まるところまで名前を補完してくれる
- 「矢印」キー
 - コマンド履歴の表示、編集
 - ↑: 前の入力コマンド
 - ↓: 次の入力コマンド
 - ←、→: カーソル (編集位置) 移動
- 文字の削除
 - 「Del」 : カーソル位置の文字
 - 「Backspace」: カーソル位置の前の文字



TSUBAMEで用いるコマンド、スクリプト

- git (分散型バージョン管理システム)
\$ git _clone_ _URL
- qsub (TSUBAMEへのジョブの提出(queue submit))
\$ qsub _script-file-name.sh
\$ qsub _-g _tga-egliteracy _-l _h_rt=time _runsubame.sh
- qstat (ジョブの状態(queue status))
\$ qstat
- qdel (TSUBAMEからのジョブの削除(queue delete))
\$ qdel _job-ID

✓ 講義で用意したスクリプト

- mkgraph.sh (折れ線グラフの作成)
\$./mkgraph.sh _'XLabel' _'X列' _'YLabel' _'Y列' _'出力ファイル名'

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 「`」



TSUBAMEでのエディタ (nano)の起動

- 課金グループ(tga-egliteracy)参加後に利用可能
- パスの事前指定
 - 方法1 (設定ファイルで指定. 一度行えば, 次回以降は不要)
\$ `cp ~/.bashrc ~/.dot.bashrc.bak`
\$ `cp /gs/hs0/tga-egliteracy/etc/dot.bashrc ~/.bashrc`
 - 方法2 (ログアウトまで有効. ログイン後毎回行う)
\$ `alias nano=/gs/hs0/tga-egliteracy/local/bin/nano`
- 実行方法
\$ `nano`
- パスの事前指定なしで実行(パスを直接指定)
\$ `/gs/hs0/tga-egliteracy/local/bin/nano`

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設計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

Google Colaboratory

Google Colab(Colaboratory)とは



- 機械学習の普及を目的としたGoogleのサービス
- クラウドで提供されるPythonの実行環境
- 環境構築が不要
- 誰でも無料で利用可能
- プログラムを簡単に共有可能
- Googleアカウントが必要

Google Colabの使い方(目次)



新規作成する場合

(Googleにログインした状態で)

1. Colab(<https://colab.research.google.com/>)にアクセス
2. 新しいノートブックを押す
3. Pythonプログラムを書いて、実行()を押す

ファイルをGithubから読み込む場合

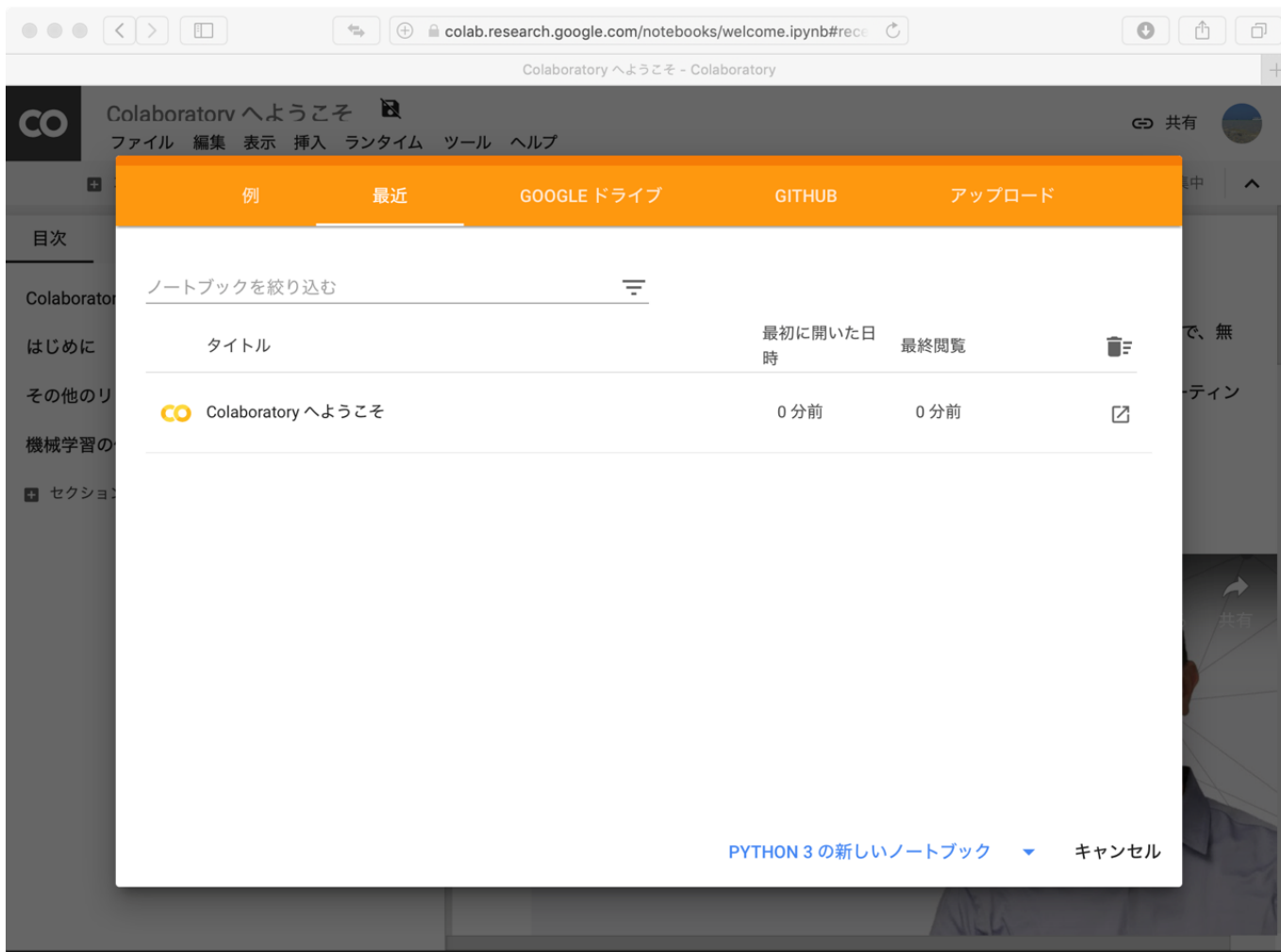
(Googleにログインした状態で)

1. Colab(<https://colab.research.google.com/>)にアクセス
2. GithubのレポジトリのURLを入力
3. 読み込むipynbファイルを選択
4. 順番に実行()を押す

[新規作成] Colabにアクセス



<https://colab.research.google.com/> にアクセス



[新規作成] 新しいノートブック CO

新しいノートブックを選択

The screenshot shows the Google Colaboratory web interface. A modal dialog box is open, titled 'ノートブックを絞り込む' (Filter notebooks). The dialog has a table with columns: 'タイトル' (Title), '最初に開いた日時' (Date and time first opened), '最終閲覧' (Last viewed), and an icon column. The table lists one notebook: 'Colaboratory へようこそ' (Welcome to Colaboratory), created '0 分前' (0 minutes ago) and last viewed '0 分前' (0 minutes ago). Below the table, there is a large blue arrow pointing down to the text 'PYTHON 3 の新しいノートブック' (New Python 3 notebook) and a 'キャンセル' (Cancel) button.

タイトル	最初に開いた日時	最終閲覧	
Colaboratory へようこそ	0 分前	0 分前	

PYTHON 3 の新しいノートブック キャンセル

[新規作成] プログラムの実行



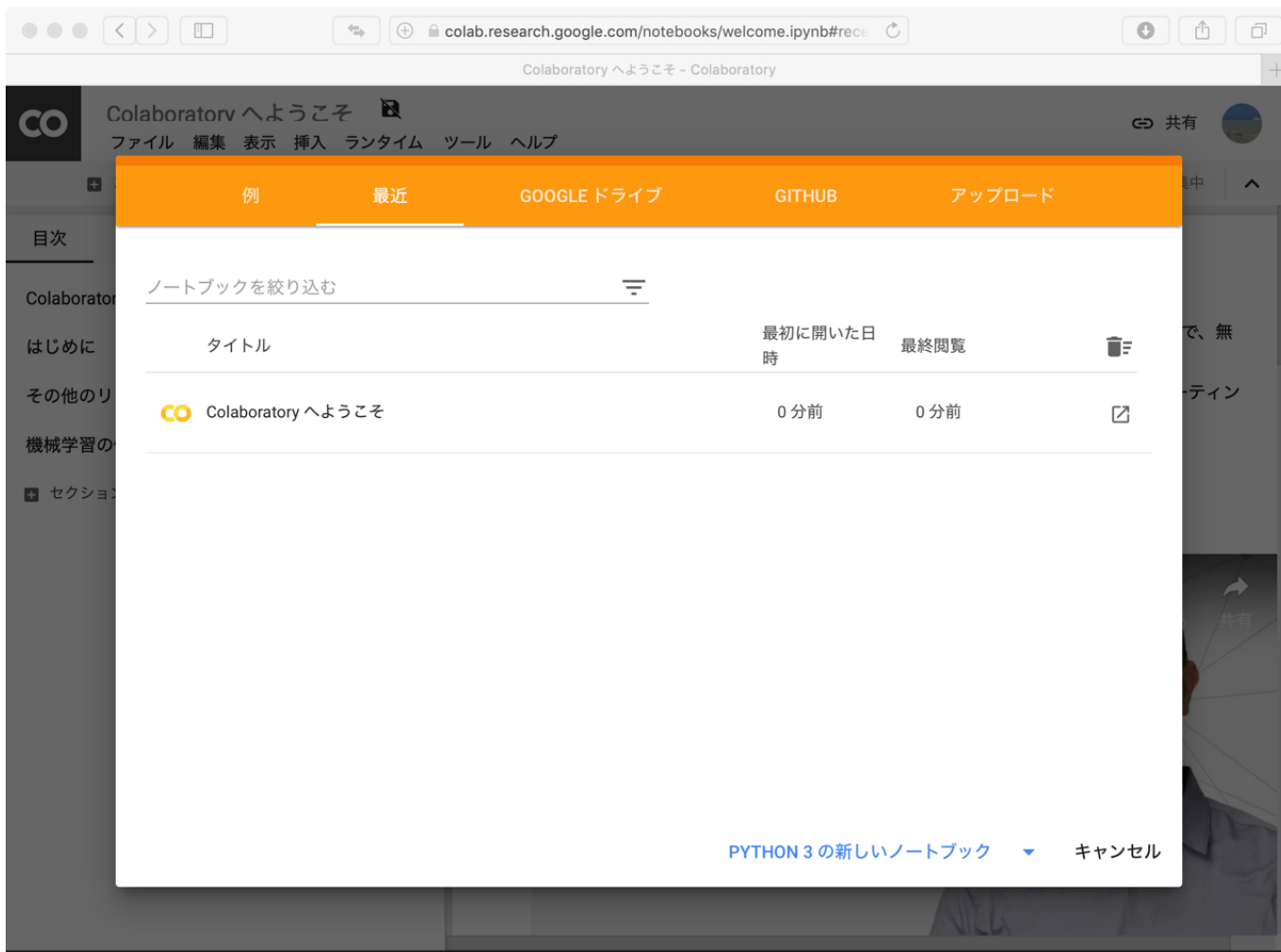
プログラムを書いて、実行()を押す

The screenshot shows the Google Colaboratory web interface. At the top, the browser address bar displays 'colab.research.google.com/drive/1Ke-Aca85-tt5ONK5ANF1'. The Colab logo is in the top left. The main area shows a code cell titled 'Untitled0.ipynb' with the code '1+2*3'. Below the code, the output '7' is displayed. A blue arrow points to the code input area, labeled 'プログラム入力欄'. Another blue arrow points to the output area, labeled '出力結果' with the subtext '画面にはプログラムの最終行の出力が表示される'. A third blue arrow points to the run button (a play icon), labeled '実行ボタン'. The interface also shows tabs for 'コード' (Code) and 'テキスト' (Text), and a status bar at the bottom indicating 'RAM' and 'ディスク' usage.

[レシピ実行] Githubから開く



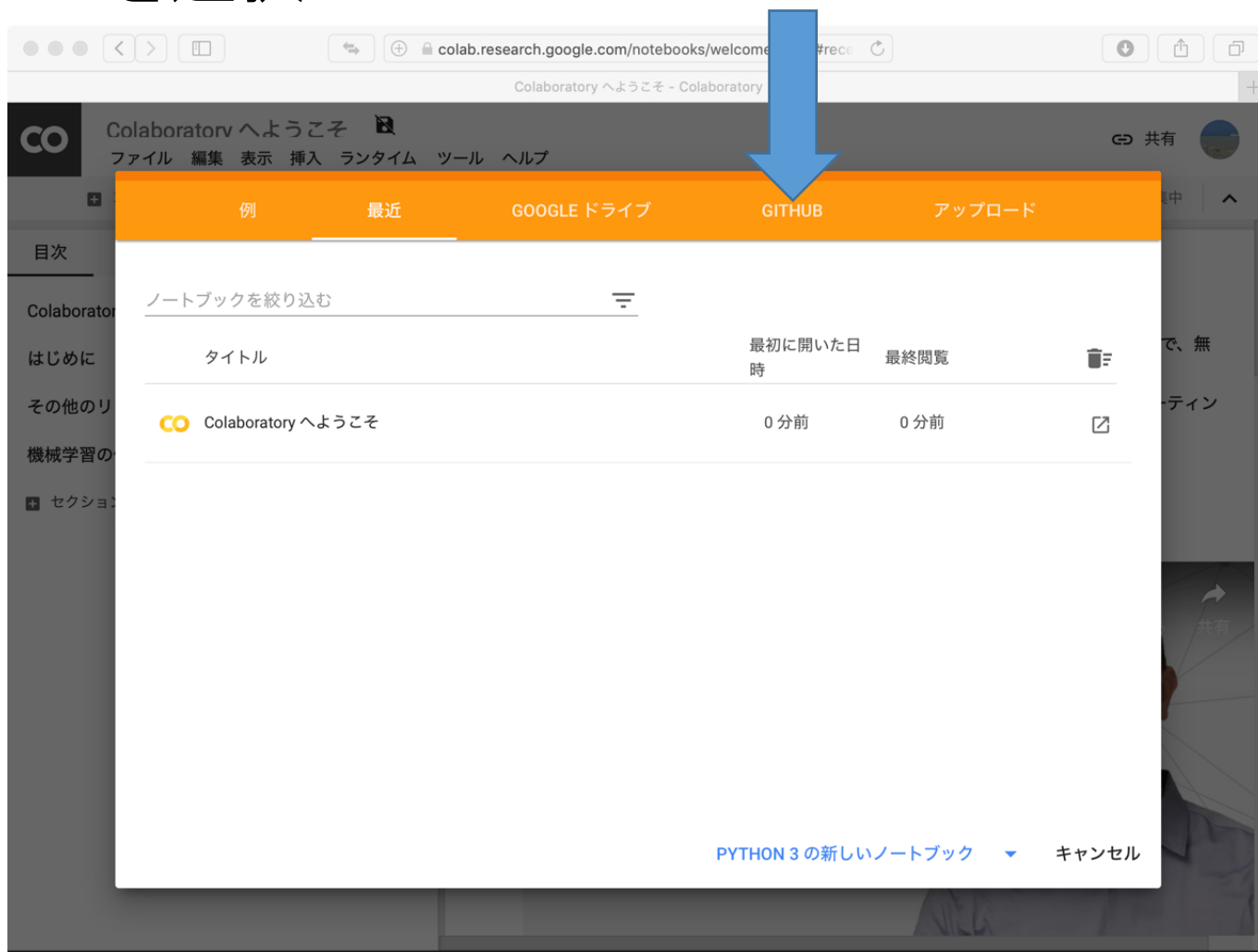
<https://colab.research.google.com/> にアクセス



[レシピ実行] Githubから開く



GITHUBを選択



[レシピ実行] Githubから開く



GithubのURLを入力してファイルを選択

Colaboratory へようこそ - Colaboratory

HirokiNakahara/MNIST_Examples: Recipes for image processing based on a tiny con...

例 最近 GOOGLE ドライブ GITHUB アップロード

Github URL を入力するか、組織またはユーザーで検索します ☐ 非公開リポジトリを含める

https://github.com/HirokiNakahara/MNIST_Examples 🔍

レポジトリ: [🔗](#) ブランチ: [🔗](#)

HirokiNakahara/MNIST_Examples master

パス

exp100p/colab.ipynb

exp1p/colab.ipynb

exp20p/colab.ipynb

exp60p/colab.ipynb

この例では画像認識の課題ファイルを開く
他の課題についても同様

PYTHON 3 の新しいノートブック ▼ キャンセル

[レシピ実行] プログラムの実行



順番に実行()を押す

colab.ipynb - Colaboratory

共有

コード テキスト セル ドライブにコピー

RAM ディスク

手書き文字認識を行うNeural Networkのレシピ

中原作成

ニューラルネットワークを使った画像認識のレシピです。今回はチュートリアルなどでよく使われている手書き文字データセットMNISTを学習してモデルを作成し、テスト画像を入力してその認識精度を確認します。

Chainer 4.5.0
CuPy 4.5.0
CUDA 9.2

環境のセットアップ (TSUBAMEでは不要?)

```
# Check GPU status
!nvidia-smi
```

```
# Install Chainer v4.5.0, Cupy v4.5.0 and CUDA
!curl https://colab.chainer.org/install | CHAINER_VERSION=="4.5.0" CUPY_VERSION=="4.5.0" sh -
```

必要なライブラリのインポート

```
# Import libraries
import chainer
import chainer.functions as F
import chainer.links as L
from chainer import training
```

[レシピ実行] プログラムの実行



初回は警告が出るので「このまま実行」を押す

The screenshot shows the Google Colaboratory interface. A warning dialog box is displayed in the center, with the following text:

警告: このノートブックは Google が作成したものではありません。

このノートブックは [GitHub](#) から読み込まれています。Google に保存されているデータへのアクセスが求められたり、他のセッションからデータや認証情報が読み取られたりする場合があります。このノートブックを実行する前にソースコードをご確認ください。このノートブックが他のセッションから状態を読み取ることができないように、ランタイムをすべてリセットできます。

Below the text, there is a checkbox labeled "実行前にランタイムをすべてリセットする" (Reset runtime before execution), which is checked.

At the bottom of the dialog, there are two buttons: "キャンセル" (Cancel) and "このまま実行" (Run anyway). A blue arrow points from the text "ランタイムをすべてリセットできます" to the "このまま実行" button.

The background of the Colab interface shows the title "手書き文字認識を行うNeural Networkのレシピ" (Recipe for handwritten character recognition using a Neural Network). The left sidebar lists the environment: Chainer 4.5.0, CuPy 4.5.0, and CUDA 9.2. The main area shows code for installing Chainer, CuPy, and CUDA, and importing libraries.

[レシピ実行] プログラムの実行



同様に「はい」を押す



[レシピ実行] プログラムの実行



プログラムの出力が表示される

colab.ipynb - Colaboratory

colab.ipynb

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

コード テキスト セル セル ドライブにコピー

RAM ディスク 編集

手書き文字認識を行うNeural Networkのレシピ

中原作成

ニューラルネットワークを使った画像認識のレシピです。今回はチュートリアルなどでよく使われている手書き文字データセットMNISTを学習してモデルを作成し、テスト画像を入力してその認識精度を確認します。

Chainer 4.5.0
CuPy 4.5.0
CUDA 9.2

環境のセットアップ (TSUBAMで

プログラム入力欄
このプログラムはGPUの状態を確認するコマンド

```
# Check GPU status
Invidia-smi
```

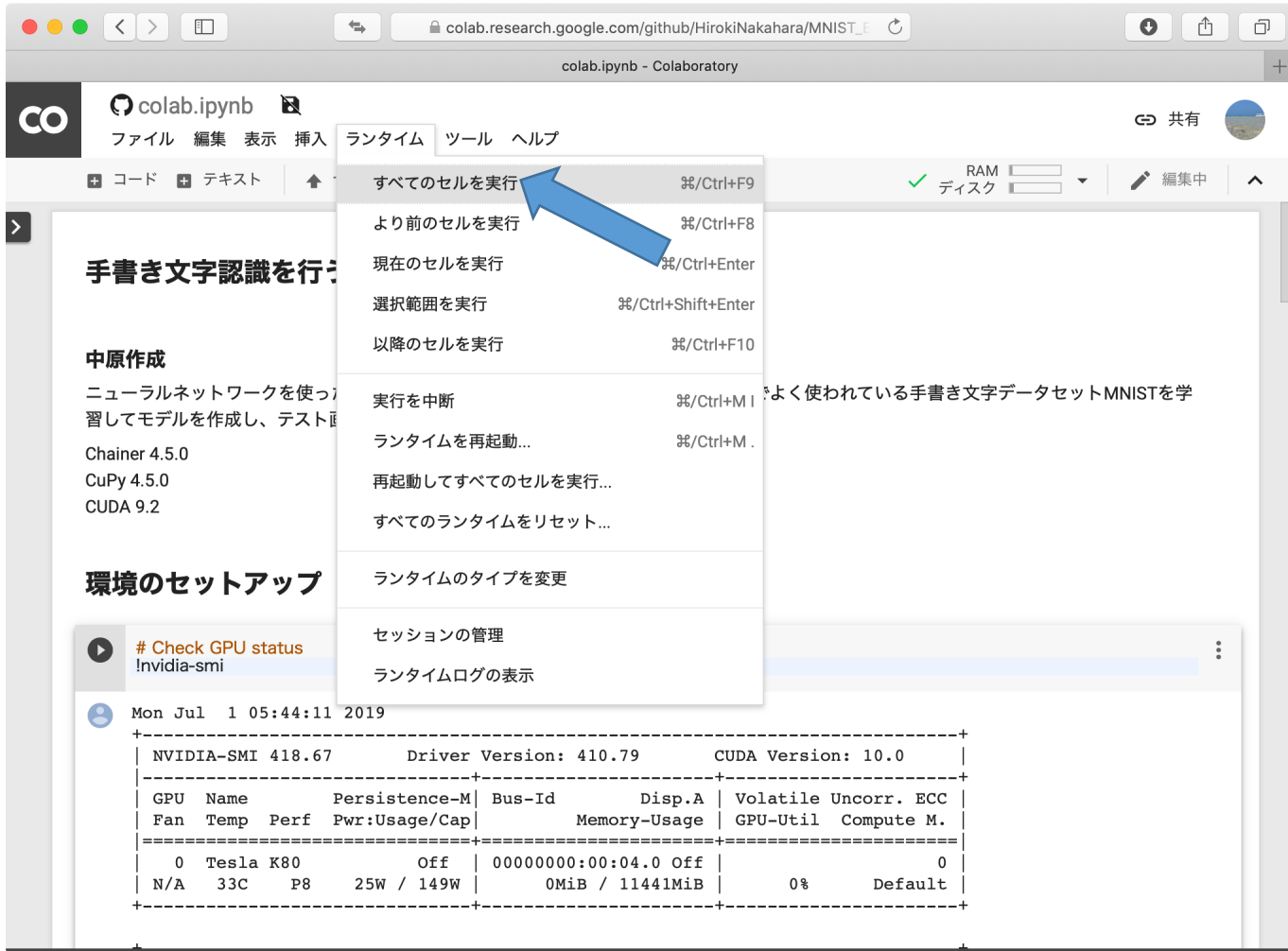
Mon Jul 1 06:18:51 2019

NVIDIA-SMI 410.79 Driver Version: 410.79 CUDA Version: 10.0			
GPU	Name	Phys	Co-M
Fan	Temp	Perf	Pwr:Usage/
0	Tesla T4	Of	
N/A	65C	P8	17W / 7

出力結果
画面にはプログラムの最終行の出力が表示される

[レシピ実行] プログラムの実行

全てのプログラムを実行することもできる



The screenshot shows the Google Colaboratory web interface. The 'ランタイム' (Runtime) menu is open, displaying various execution options. A blue arrow points to the 'すべてのセルを実行' (Run all cells) option, which is highlighted. The menu also includes options like 'より前のセルを実行' (Run previous cells), '現在のセルを実行' (Run current cell), '選択範囲を実行' (Run selection), '以降のセルを実行' (Run remaining cells), '実行を中断' (Interrupt), 'ランタイムを再起動...' (Restart runtime), '再起動してすべてのセルを実行...' (Restart and run all cells), 'すべてのランタイムをリセット...' (Reset all runtimes), 'ランタイムのタイプを変更' (Change runtime type), 'セッションの管理' (Manage session), and 'ランタイムログの表示' (Show runtime logs).

colab.ipynb - Colaboratory

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

コード テキスト

すべてのセルを実行 ⌘/Ctrl+F9

より前のセルを実行 ⌘/Ctrl+F8

現在のセルを実行 ⌘/Ctrl+Enter

選択範囲を実行 ⌘/Ctrl+Shift+Enter

以降のセルを実行 ⌘/Ctrl+F10

実行を中断 ⌘/Ctrl+M

ランタイムを再起動... ⌘/Ctrl+M

再起動してすべてのセルを実行...

すべてのランタイムをリセット...

ランタイムのタイプを変更

セッションの管理

ランタイムログの表示

手書き文字認識を行...

中原作成

ニューラルネットワークを使っ...

習してモデルを作成し、テスト...

Chainer 4.5.0

CuPy 4.5.0

CUDA 9.2

環境のセットアップ

Check GPU status

Invidia-smi

Mon Jul 1 05:44:11 2019

NVIDIA-SMI 418.67		Driver Version: 410.79		CUDA Version: 10.0	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util Compute M.
0	Tesla K80	Off	00000000:00:04.0	Off	0
N/A	33C	P8	25W / 149W	0MiB / 11441MiB	0% Default

[レシピ実行] 学習結果の確認



授業プログラムは最後のセルに学習結果が出力

```
print('val_loss:{:.04f} val_accuracy:{:.04f}'.format(
    np.mean(test_losses), np.mean(test accuracies)))

end_time = time.time()
print("-----")
print("Training complete")
chainer.serializers.save_npz('mnist.model', model)

print('accuracy:{:.04f}'.format(np.mean(test accuracies)))
print('time cost:{:.04f}'.format(end_time-start_time), 's')
```

```
epoch:01 train_loss:2.2551 train_accuracy:0.1600 val_loss:2.2796 val_accuracy:0.1749
epoch:02 train_loss:2.2331 train_accuracy:0.2900 val_loss:2.2537 val_accuracy:0.2423
epoch:03 train_loss:2.2037 train_accuracy:0.4300 val_loss:2.2286 val_accuracy:0.3202
-----
Training complete
accuracy:0.3202
time cost:7.4747 s
```

結果は実際に実行して確認してください