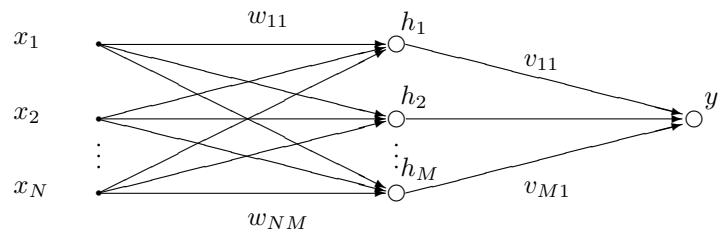


Machine Learning

TBMI26 / 732A55

Computer Assignments

2023



Contents

General Information	2
Lab assignments	3
A1: Supervised learning	3
A2: Deep learning	3
A3: Boosting	3
A4: Reinforcement learning	3
Lab submission	4
Written submission	4
Oral examination	4
Python setup	5
Azure setup	36

General information

This compendium contains instructions for the 4 computer assignments. Additional resources are found on the LISAM course page (lisam.liu.se).

You have to pass all assignments to get the lab credits. The assignments will be graded (pass/fail) based on your solutions and answers, which you submit in LISAM. All files needed for the assignments are found in the zip file available on LISAM. The solved assignments should be submitted individually or in groups of two students. The submissions will be checked for plagiarism. Plagiarized submissions will be reported to the LiU disciplinary board.

The deadlines for the reports can be found in LISAM. If a report is sent in late we might not look at it until the next re-examination period. *So, submit your reports in time!*

Important: This year, all four assignments have gone through major updates. In case you discover any technical problems with the provided code, we would be grateful if you notified us as soon as possible so the issue can be fixed.

Common setup: All four assignments are written in python notebooks. This is a file type used for interactive python coding. We recommend that you use Jupyter Lab and that you install the Variable Inspector extension. This will give you a list of all current variables and their values, which will make debugging much more convenient. Once you are up and running with your first notebook, you can open the Variable Inspector by right-clicking anywhere in the notebook window and selecting "Open Variable Inspector". In the same way, you can also open a python console for the notebook, which can be useful when testing your code.

The step-by-steps instructions for setting up the lab environment is included at the end of this pdf.

It is also possible to run the notebooks in Visual Studio Code, but we do not provide the instructions for that environment and cannot guarantee that it will work as expected.

Lab assignments

This course has four assignments. These are, in order:

A1: Supervised learning

This assignment contains three notebooks. First you will implement and test the k-nearest neighbour algorithm. Then you will implement and train a simple single-layer neural network. Finally, you will implement and train a multi-layer neural network.

Note that this assignment is often regarded as the most difficult and time consuming, since it covers three different models. Do not expect to finish this lab in the scheduled timeslot, you have to prepare in advance!

A2: Deep learning

Here you will use the Keras framework for deep learning to train increasingly powerful models for object classification. To get you started using Keras, we have provided a separate notebook with some explanations and examples, called `MNIST-Demo.ipynb`. It is highly recommended that you go through this demo well in advance.

Note that this assignment requires a different Python environment than the other three (either `tf_env` or `tf_gpu_env`), due to the inclusion of several large deep learning packages. The details are in the Python setup instructions below.

A3: Boosting

This assignment will cover a method for supervised ensemble learning, specifically the algorithm AdaBoost, in which many weak classifiers are combined to make a single strong classifier. You will use this technique to build a face detector.

A4: Reinforcement learning

In the last assignment you will implement and train a reinforcement learning agent to find the optimal paths through different grid worlds.

Lab submission

Written submission

You submit your solved assignments in Lisam. To ensure that you submit a working solution we will run your code. Therefore, you must submit both your notebook file (and any required external files), as well as an exported version of the notebook (HTML-format). In short, follow these steps to ensure a valid submission:

1. Export your notebook using the menu bar option `File -> Export Notebook As... -> Export Notebook to HTML / PDF`
2. Use the menu bar option `Kernel -> Restart Kernel` and `Run All....` This ensures that all code is run from a clean state and actually works. This is for your own benefit; if the code does not run, you will get it back.
3. Make sure nothing big changed in the solution. Some variance is expected due to randomness, but you should get comparable results. If you did not you probably need to optimize your hyperparameters to get more stable solutions.
4. Zip all files *except* the exported HTML file.
5. Submit the Zip and HTML in Lisam. The exported HTML-file will be checked for plagiarism, and we will run your code.

Oral examination

As an alternative to the default written submission, we will offer the option to instead do the lab examination orally during the scheduled lab session. If you pass the lab this way, you will not have to wait for us to correct the written report and will instead be passed directly (or within a day or two). You should still submit your solutions for reference in Lisam. In addition to a faster approval, we teachers also get the opportunity to discuss your solutions in order to clear up any misunderstandings. Although note that this examination session is not meant to help you solve the labs. You should be fully done and well prepared before you request an oral examination. In case we require you to make revisions you will have to submit them in the default written format in Lisam.

Again note, this is only an option. We do not require you to present the labs orally if you do not want to.

We will give more details about this before the first lab session.

Python setup

The following pages will go through the installation process for the lab environment.

(There are more instructions after these, don't miss them.)

Setting up your computer for the labs

Please read fully before starting

Setting up the environment

This only needs to be done once

Miniconda — Conda document

docs.conda.io/en/latest/miniconda.html#latest-miniconda-installer-links

Latest Miniconda Installer Links

Latest - Conda 4.10.3 Python 3.9.5 released July 21, 2021

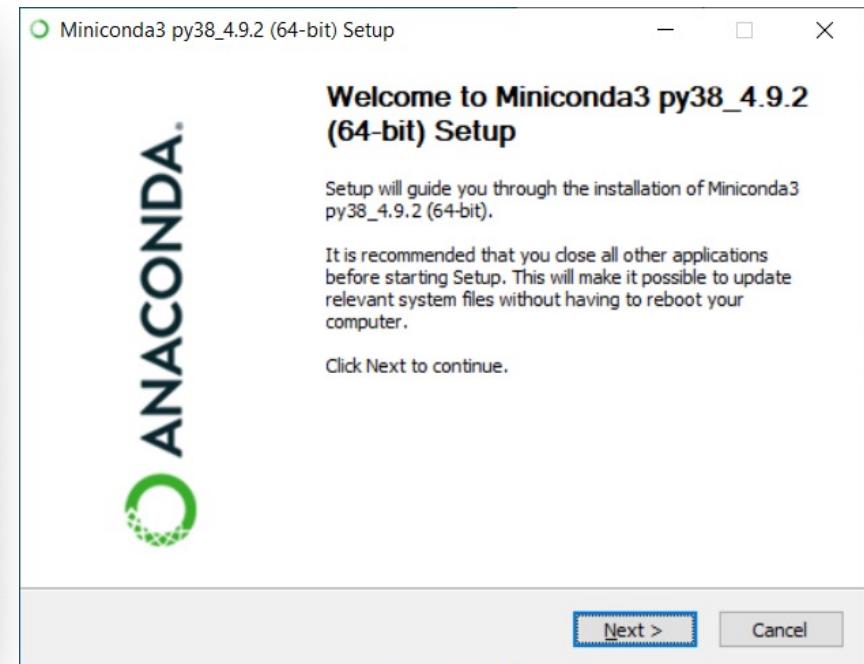
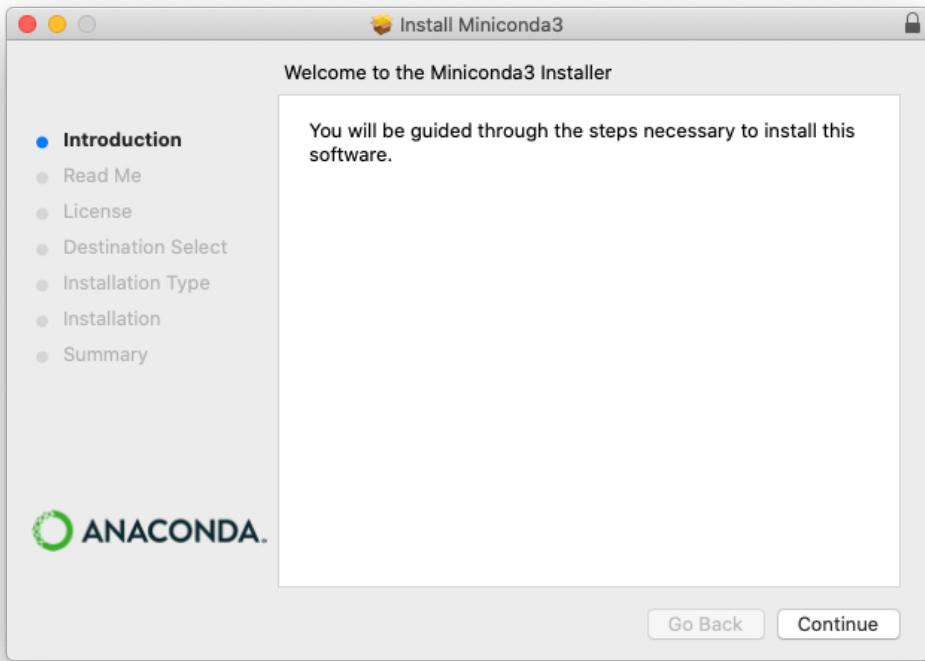
Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	b33797064593ab2229a0135dc69001bea05cb56a20c2f243b1231
	Miniconda3 Windows 32-bit	24f438e57ff2ef1ce1e93050d4e9d13f5050955f759f448d84a40
MacOSX	Miniconda3 MacOSX 64-bit bash	786de9721f43e2c7d2803144c631
	Miniconda3 MacOSX 64-bit pkg	8fa371ae97218c3c005cd5f04b1
	Miniconda3 macOS Apple M1 64-bit bash (Py38 conda 4.10.1 2021-11-08)	4ce4047065f32e991eddbb63b3
Linux	Miniconda3 Linux 64-bit	1ea2f885b4dbc3098662845560b
	Miniconda3 Linux-aarch64 64-bit	4879820a10718743f945d88ef14
	Miniconda3 Linux-ppc64le 64-bit	fa92ee4773611f58ed9333f977d
	Miniconda3 Linux-s390x 64-bit	1faed9abecf4a4dd4e0d8891fc

Windows installers

Windows

Python version	Name	Size	SHA256 hash
Python 3.9	Miniconda3 Windows 64-bit	58.1 MiB	b33797064593ab2229a0135dc69001bea05cb56a20c2f243b1231
Python 3.8	Miniconda3 Windows 64-bit	57.3 MiB	8940cdd621557bc55743d6bb4518c6d343a4587127e76de808fb0
Python 3.7	Miniconda3 Windows 64-bit	55.8 MiB	9c031506bfcb0428a0ac46c9152f9bdd48d5bd8aa83046691bf8e0
Python 3.9	Miniconda3 Windows 32-bit	55.3 MiB	24f438e57ff2ef1ce1e93050d4e9d13f5050955f759f448d84a40

Download the right version of Miniconda for your OS.



Install Miniconda. The default settings should be fine.

Neural Networks and Learning x +

liuonline.sharepoint.com/sites/Lisam_TBMI26_2022VT_8A/CourseDocuments/Forms/AllItems.aspx?id=%2Fsites%2FLisam_TBMI26...

SharePoint Search this library

Lisam / Neural Networks and Learning Systems (TBMI26 2022VT 8A) Adjust course room Courses and programs

NN Neural Networks and Learning Systems (TBMI26 2022VT 8A) Private group

Home + New Upload Edit in grid view Share ... All Documents

Course plan Course documents Collaborative workspace Members and groups Schedule Teams Submissions Signup Lecture videos Edit

Course documents > Assignments > environment_files

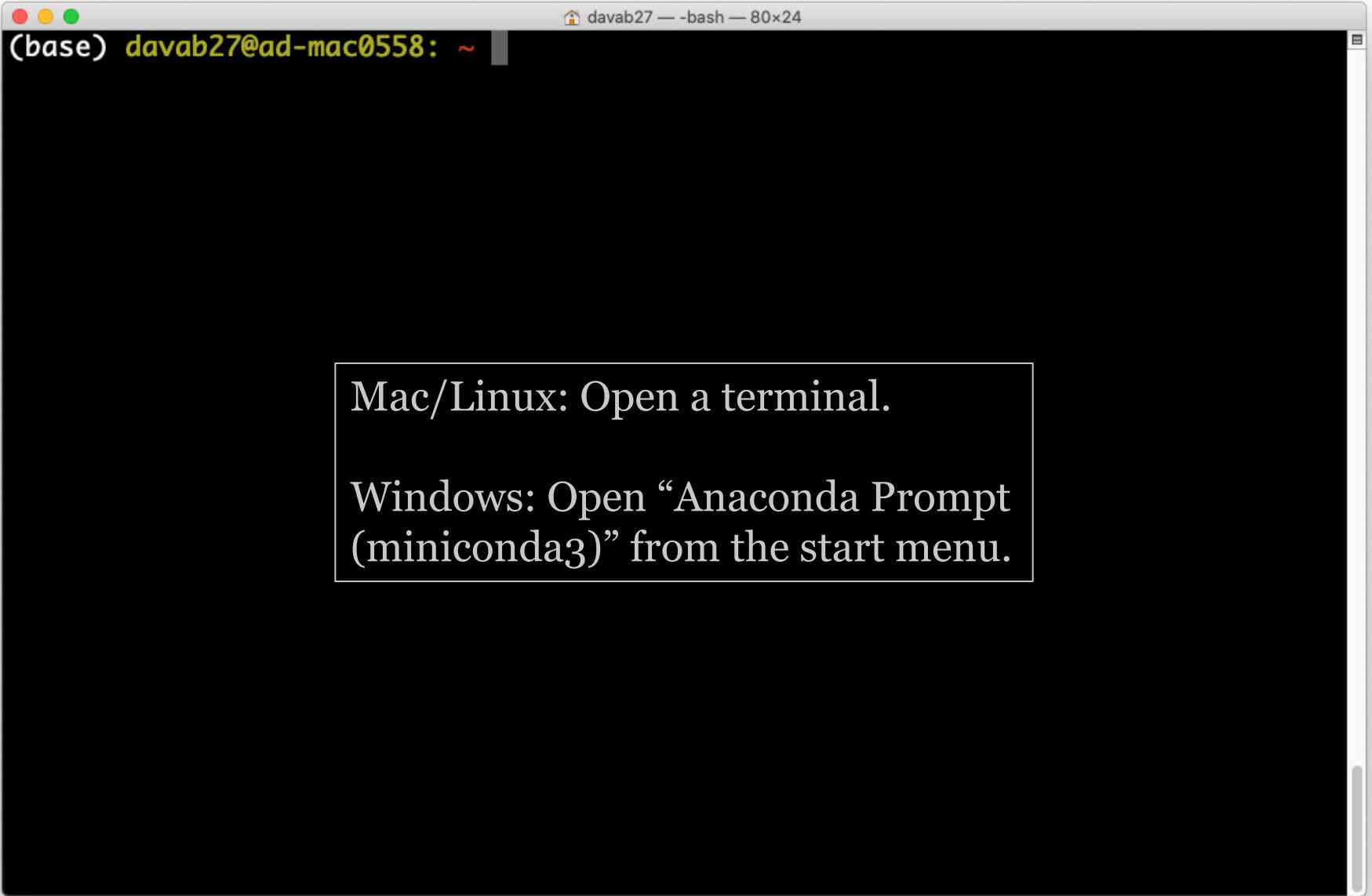
Name	Modified	Modified By	Version	Description
tbmi26_env.yml	21 minutes ago	David Abramian	1.0	
tbmi26_tf_env.yml	21 minutes ago	David Abramian	1.0	
tbmi26_tf_gpu_env.yml	21 minutes ago	David Abramian	1.0	

Return to classic SharePoint

Download the right yml file for your situation:

- Default environment for labs 1, 3, 4. Recommended!
- Usable also for lab 2 if you don't have an Nvidia GPU.
- Usable also for lab 2 if you have an Nvidia GPU.

Only the last of these will take advantage of a GPU.
The rest will run on the CPU.



```
 davab27 — bash — 80x24
(base) davab27@ad-mac0558: ~ conda config --set auto_activate_base false
(base) davab27@ad-mac0558: ~ conda deactivate
davab27@ad-mac0558: ~
```

Mac/Linux: Without going into explanations, run the commands above. You will not have to do this again.

Windows: Skip this and go to next step.

```
[davab27@ad-mac0558: ~ conda env create -n tbmi26 -f Downloads/tbmi26_env.yml ]
```

We create the conda environment used in this course by running the command above.

- Name of the conda environment.
- Path to the yml file downloaded earlier.

```
[davab27@ad-mac0558: ~ conda env create -n tbmi26 -f Downloads/tbmi26_env.yml ]
```

Collecting package metadata (repodata.json): done
Solving environment: -

This will download and install many packages and will probably take a while.

```
matplotlib-inline-0. | 12 KB    | ##### | 100%
bleach-4.1.0        | 123 KB   | ##### | 100%
importlib_metadata-4 | 12 KB    | ##### | 100%
jinja2-3.0.2        | 110 KB   | ##### | 100%
pip-21.2.2          | 1.8 MB   | ##### | 100%
prompt-toolkit-3.0.2 | 259 KB   | ##### | 100%
argcomplete-1.12.3  | 35 KB    | ##### | 100%
freetype-2.11.0     | 589 KB   | ##### | 100%
giflib-5.2.1         | 70 KB    | ##### | 100%
debugpy-1.5.1        | 1.7 MB   | ##### | 100%
mkl_fft-1.3.1       | 161 KB   | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate tbmi26
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

davab27@ad-mac0558: ~

We now have an environment with Jupyter and all the other packages we will need.

Neural Networks and Learning x +

liuonline.sharepoint.com/sites/Lisam_TBMI26_2022VT_8A/CourseDocuments/Forms/AllItems.aspx?id=%2Fsites%2FLisam_TBMI26...

SharePoint Search this library

Lisam / Neural Networks and Learning Systems (TBMI26 2022VT 8A) Adjust course room Courses and programs

NN Neural Networks and Learning Systems (TBMI26 2022VT 8A) Private group

Home + New Upload Edit in grid view Share ... All Documents

Course plan Course documents Collaborative workspace Members and groups Schedule Teams Submissions Signup Lecture videos Edit

Course documents > Assignments

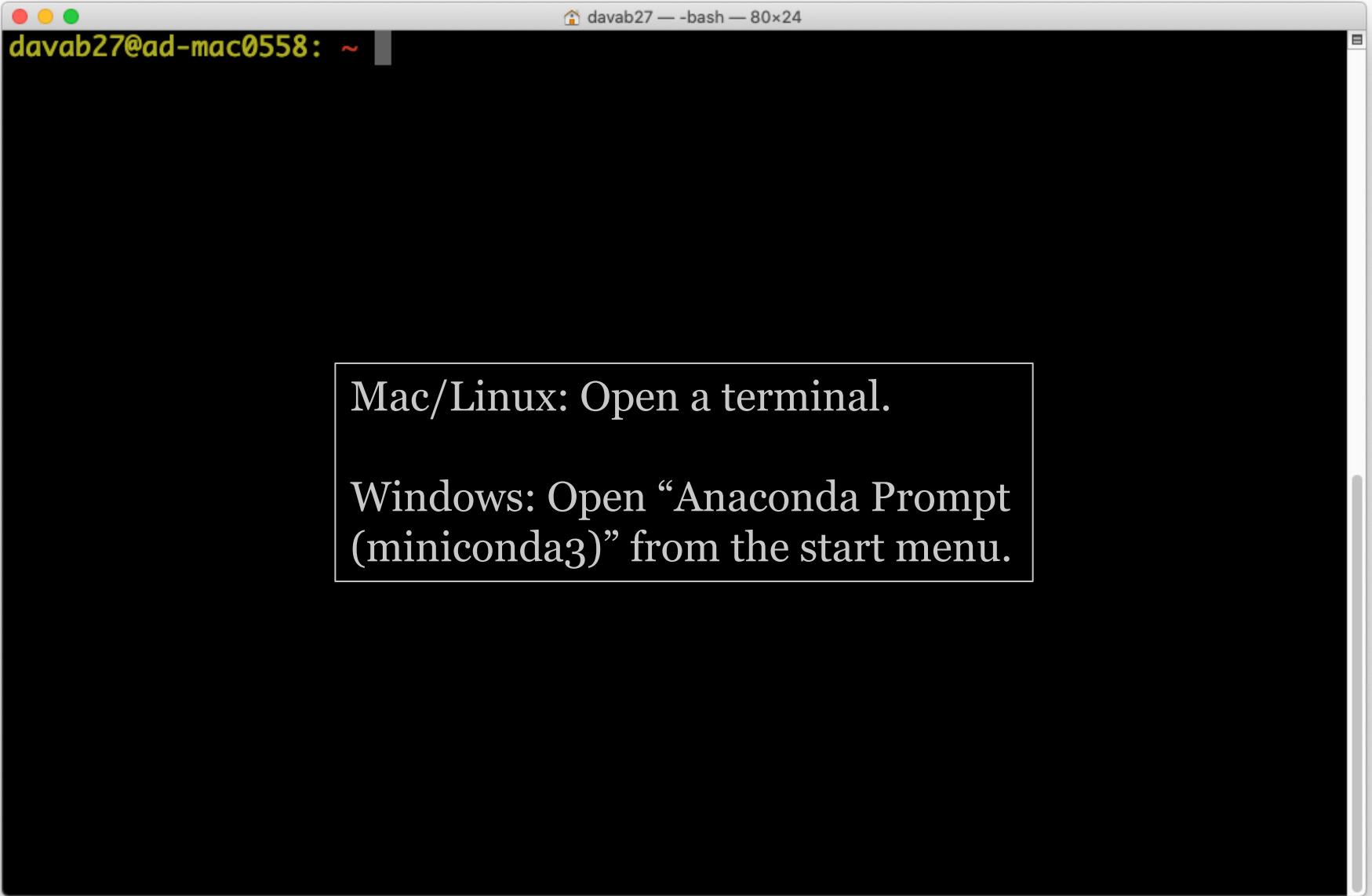
Name	Modified	Modified By	Version	Description
environment_files	21 minutes ago	David Abramian	1.0	
Lab_2	4 days ago	David Abramian	1.0	
Assignments.zip	4 days ago	David Abramian	1.0	
ComputerCompendium.pdf	4 days ago	David Abramian	1.0	

If you don't have the assignment files already, download them from Lisam and extract them somewhere.

Return to classic SharePoint

Launching the labs

Do this every time you want to work on the lab



davab27 — bash — 80x24

```
[davab27@ad-mac0558: ~ conda activate tbmi26  
(tbmi26) davab27@ad-mac0558: ~ ]
```

Activate the conda environment.

```
Assignments — -bash — 80x24
[davab27@ad-mac0558: ~ conda activate tbmi26
(tbmi26) davab27@ad-mac0558: ~ cd Downloads/Assignments
(tbmi26) davab27@ad-mac0558: Assignments ]
```

Navigate to the folder containing the lab files.

```
Assignments — node < python3.7 ~/opt/miniconda3/envs/tbmi26/bin/jupyter-lab — 80x24
(tbmi26) davab27@ad-mac0558: Assignments jupyter lab
[I 11:59:55.695 LabApp] JupyterLab extension loaded from /Users/davab27/opt/miniconda3/envs/tbmi26/lib/python3.7/site-packages/jupyterlab
[I 11:59:55.695 LabApp] JupyterLab application directory is /Users/davab27/opt/miniconda3/envs/tbmi26/share/jupyter/lab
[I 11:59:55.700 LabApp] Serving notebooks from local directory: /Users/davab27/Downloads/Assignments
[I 11:59:55.700 LabApp] Jupyter Notebook 6.4.6 is running at:
[I 11:59:55.700 LabApp] http://localhost:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663
[I 11:59:55.700 LabApp] or http://127.0.0.1:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663
[I 11:59:55.700 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:59:55.734 LabApp]
```

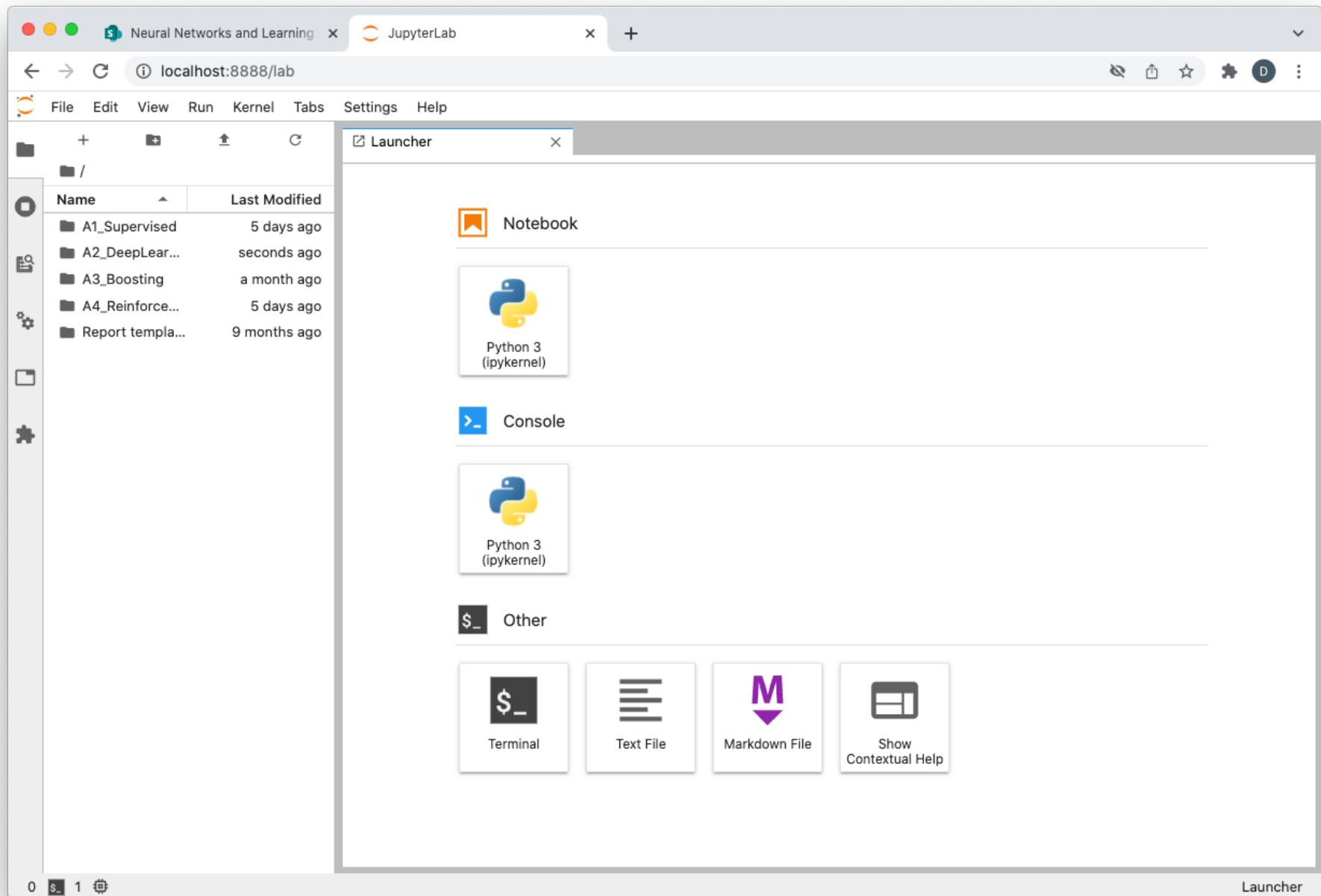
To access the notebook, open this file in a browser:

file:///Users/davab27/Library/Jupyter/runtime/nbserver-17472-open.html

Or copy and paste one of these URLs:

http://localhost:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663
or http://127.0.0.1:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663

Start Jupyter with the given command.



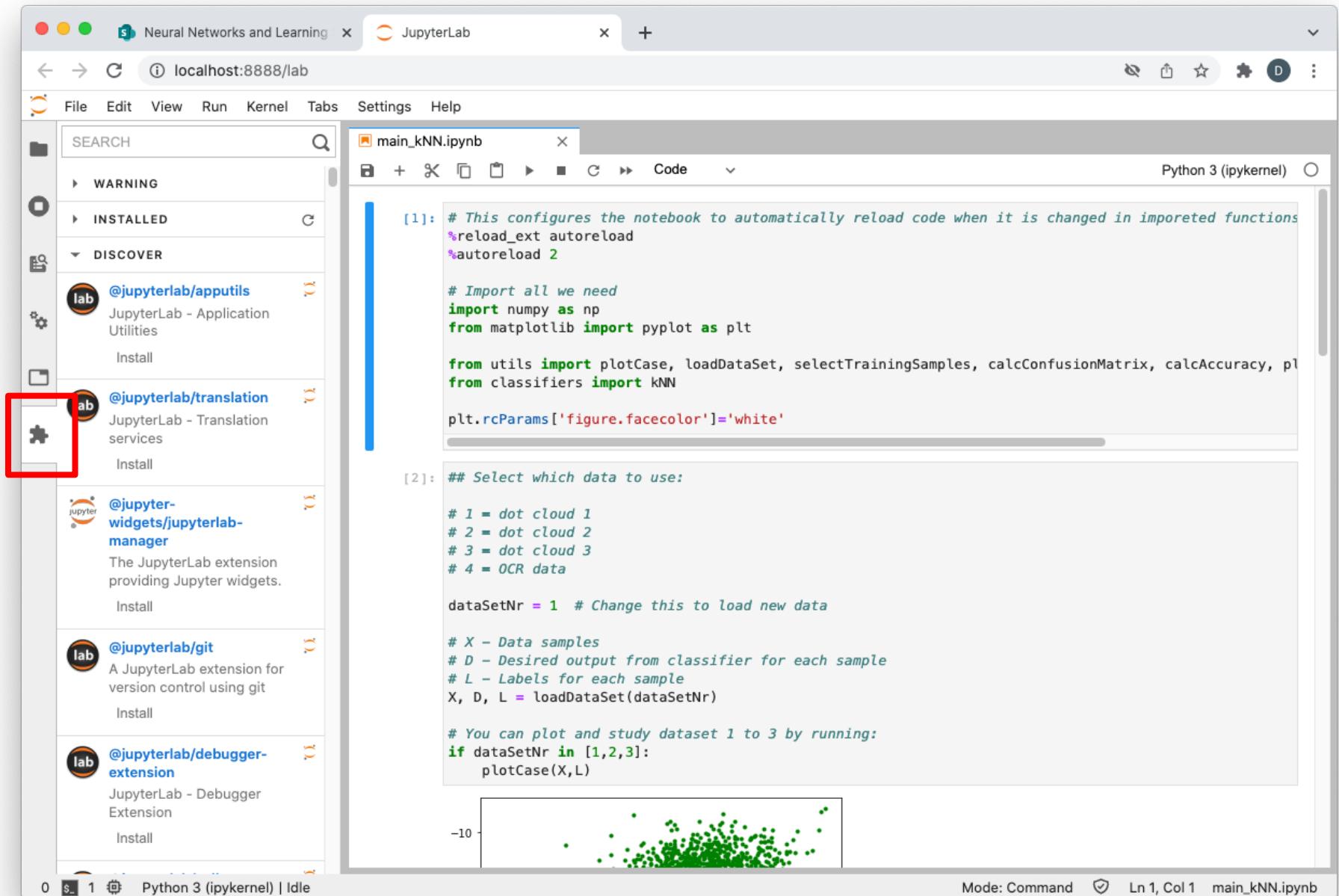
Jupyter will launch in a browser window automatically.

```
[I 12:00:06.814 LabApp] Build is up to date
[I 12:00:07.735 LabApp] Kernel started: 48f5dfed-97b8-4af3-ab55-6070c7fbubb5, na
me: python3
[W 12:01:52.272 LabApp] Notebook A1_Supervised/main_kNN.ipynb is not trusted
[I 12:01:52.653 LabApp] Kernel started: eff4d800-4a83-4209-b7bd-302f07cdf914, na
me: python3
[I 12:01:55.938 LabApp] Starting buffering for eff4d800-4a83-4209-b7bd-302f07cdf
914:dc4a55bb-3203-4c1e-95f1-664999d40786
[I 12:02:01.786 LabApp] Starting buffering for 48f5dfed-97b8-4af3-ab55-6070c7fb
bc5:25db727f-3035-4c78-9060-dcb4dcc1d7cf
^C[I 12:02:13.651 LabApp] interrupted
Serving notebooks from local directory: /Users/davab27/Downloads/Assignments
2 active kernels
Jupyter Notebook 6.4.6 is running at:
http://localhost:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663
or http://127.0.0.1:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc866
3
Shutdown this notebook server (y/[n])? ^C[C 12:02:13.864 LabApp] received signal
2, stopping
[I 12:02:13.865 LabApp] Shutting down 2 kernels
[I 12:02:14.069 LabApp] Kernel shutdown: 48f5dfed-97b8-4af3-ab55-6070c7fbubb5
[I 12:02:14.070 LabApp] Kernel shutdown: eff4d800-4a83-4209-b7bd-302f07cdf914
[I 12:02:14.071 LabApp] Shutting down 0 terminals
(tbmi26) davab27@ad-mac0558: Assignments
```

You can shutdown Jupyter by pressing
[Ctrl + C] twice in the terminal.

Install Jupyter variable inspector

This is strongly recommended!



Open the Extensions tab.

Neural Networks and Learning x JupyterLab x localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

inspector

WARNING

INSTALLED

No entries

SEARCH RESULTS

@lckr/jupyterlab_variableinspector

Variable inspector extension for JupyterLab

Install

@almond-sh/jupyterlab_variableinspector

Variable inspector extension for JupyterLab

Install

main_kNN.ipynb

Code

Python 3 (ipykernel)

```
[1]: # This configures the notebook to automatically reload code when it is changed in imported functions
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectTrainingSamples, calcConfusionMatrix, calcAccuracy,
from classifiers import KNN

plt.rcParams['figure.facecolor']='white'

## Select which data to use:

# 1 = dot cloud 1
# 2 = dot cloud 2
# 3 = dot cloud 3
# 4 = OCR data

dataSetNr = 1 # Change this to load new data

# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by running:
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```



Mode: Command Ln 1, Col 1 main_kNN.ipynb

Search for and install the variable inspector extension (make sure to get the indicated one). This will take some time.

Neural Networks and Learning x JupyterLab x localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

inspector

A build is needed to include the latest changes

Ignore Rebuild

WARNING

INSTALLED

@lckr/jupyterlab_variableinspector Variable inspector extension for JupyterLab Uninstall Disable

SEARCH RESULTS

@almond-sh/jupyterlab_variableinspector Variable inspector extension for JupyterLab Install

main_kNN.ipynb

Python 3 (ipykernel)

```
[1]: # This configures the notebook to automatically reload code when it is changed in imported functions
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectTrainingSamples, calcConfusionMatrix, calcAccuracy,
from classifiers import KNN

plt.rcParams['figure.facecolor']='white'

## Select which data to use:

# 1 = dot cloud 1
# 2 = dot cloud 2
# 3 = dot cloud 3
# 4 = OCR data

dataSetNr = 1 # Change this to load new data

# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by running:
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```



Mode: Command Ln 1, Col 1 main_kNN.ipynb

After installation you will need to rebuild Jupyter. This will take some time.

Neural Networks and Learning x JupyterLab x localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

inspector

WARNING

INSTALLED

@lckr/jupyterlab_variableinspector Variable inspector extension for JupyterLab Uninstall Disable

SEARCH RESULTS

@almond-sh/jupyterlab_variableinspector Variable inspector extension for JupyterLab Install

main_kNN.ipynb Python 3 (ipykernel)

```
[1]: # This configures the notebook to automatically reload code when it is changed in imported functions
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectTrainingSamples, calcConfusionMatrix, calcAccuracy,
from classifiers import KNN

plt.rcParams['figure.dpi'] = 100
plt.rcParams['font.size'] = 10
plt.rcParams['font.family'] = 'serif'

## Set the number of data sets to load
# 1 - Iris dataset
# 2 - Wine dataset
# 3 - Breast Cancer dataset
# 4 - MNIST dataset

dataSetNr = 1 # Change this to load new data

# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by running:
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```

Build Complete

Build successfully completed, reload page?

Cancel Reload

Saving completed

Mode: Command

Ln 1, Col 1 main_kNN.ipynb

Reload the page.

The screenshot shows the JupyterLab interface with a floating sidebar containing extension management. A context menu is open over a code cell, with the 'Open Variable Inspector' option highlighted by a red box.

File Edit View Run Kernel Tabs Settings Help

SEARCH

WARNING

INSTALLED

- @lckr/jupyterlab_variableinspector Variable inspector extension for JupyterLab [Uninstall](#) [Disable](#)
- @jupyterlab/apputils JupyterLab - Application Utilities [Install](#)
- @jupyterlab/translation JupyterLab - Translation services [Install](#)
- @jupyter-widgets/jupyterlab-manager The JupyterLab extension providing Jupyter widgets. [Install](#)
- @jupyterlab/git A JupyterLab extension for version control using git [Install](#)
- @jupyterlab/debugger-extension

main_kNN.ipynb

```
[1]: # This configures the notebook to automatically reload on changes
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectTrainingData
from classifiers import kNN

plt.rcParams['figure.facecolor']='white'

[2]: ## Select which data to use:

# 1 = dot cloud 1
# 2 = dot cloud 2
# 3 = dot cloud 3
# 4 = OCR data

dataSetNr = 1 # Change this to load new data

# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by running:
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```

Cut Cells X
Copy Cells C
Paste Cells Below V
Delete Cells D, D
Split Cell ⌘ ⌘ -
Merge Selected Cells ⌘ M
Create New View for Output
Clear Outputs
Clear All Outputs
Enable Scrolling for Outputs
Disable Scrolling for Outputs
Undo Cell Operation Z
Redo Cell Operation ⌘ Z
Restart Kernel...
New Console for Notebook
Show Contextual Help ⌘ I
Show Log Console
Open Variable Inspector Open Variable Inspector
Shift+Right Click for Browser Menu

Mode: Command Ln 1, Col 1 main_kNN.ipynb

You can right click anywhere to open the variable inspector.

The screenshot shows the JupyterLab interface with the "Variable Inspector" extension open. The left sidebar displays a list of installed extensions, including @lckr/jupyterlab_variableinspector, @jupyterlab/apputils, @jupyterlab/translation, @jupyter-widgets/jupyterlab-manager, @jupyterlab/git, and @jupyterlab/debugger-extension. The main area shows the Variable Inspector panel with the following data:

	NAME	TYPE	SIZE	SHAPE	CONTENT
D	ndarray	32000	2000 x 2		array([[0.99, -0.99], [0.99, -0.99], [0.99, -0.99], ..., [-0.99, 0.99], [-0.99, 0.99], [-0.99, 0.99]])
L	ndarray	2000	2000		array([1, 1, 1, ..., 2, 2, 2], dtype=uint8)
X	ndarray	32000	2000 x 2		array([[-0.90412345, 3.26041993], [0.09863015, 4.35740643], [-0.45150727, 2.41539265], ..., [0.22865578, -7.56594632] ...)
dataSetNr	int	28			1

At the bottom, the status bar indicates "Mode: Command" and "Ln 1, Col 1 main_kNN.ipynb".

This will allow you to examine the defined variables and their shapes (the “size” column is not so useful, as it gives the size of the variable in bytes).

The screenshot shows the JupyterLab interface with several tabs open. The main tab is 'main_kNN.ipynb' containing Python code for k-Nearest Neighbors classification. The code imports numpy, matplotlib.pyplot, and classifiers from utils. It defines variables D, L, and X, and loads data from dataSetNr. A plot is shown at the bottom of the code cell.

```
[1]: # This configures the notebook to automatically reload extensions
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectData
from classifiers import kNN

plt.rcParams['figure.facecolor']='white'

## Select which data to use:

# 1 = dot cloud 1
# 2 = dot cloud 2
# 3 = dot cloud 3
# 4 = OCR data

dataSetNr = 1 # Change this to load new data

# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```

The Variable Inspector panel on the right shows the state of variables:

NAME	TYPE	SIZE	SHAPE	CONTENT
D	ndarray	32000	2000 x 2	array([[0.99, -0.99], [0.99, -0.99], [0.99, -0.99], ... [-0.99, 0.99], [-0.99, 0.99], [-0.99, 0.99]])
L	ndarray	2000	2000	array([1, 1, 1, ..., 2, 2, 2], dtype=uint8)
X	ndarray	32000	2000 x 2	array([-0.90412345, 3.26041993], [0.09863015, 4.35740643], [-0.45150727, 2.41539265], ... [0.22865578, -7.56594632] ...)
dataSetNr	int	28		1

The left sidebar includes a 'SEARCH' bar and sections for 'WARNING', 'INSTALLED' (listing '@lckr/jupyterlab_variableinspector'), and 'DISCOVER' (listing '@jupyterlab/apputils', '@jupyterlab/translation', '@jupyter-widgets/jupyterlab-manager', and '@jupyterlab/git').

You can also view it side-by-side with the code by dragging the tab to the right.

Deleting the environment

Do this if you want once the course is finished

```
davab27 — -bash — 80x24
davab27@ad-mac0558: ~
```

Once the course is finished, you may want to remove the environment we used.

Mac/Linux: Open a terminal.

Windows: Open “Anaconda Prompt (miniconda3)” from the start menu.

```
 davab27@ad-mac0558: ~ conda env remove -n tbmi26
Remove all packages in environment /Users/davab27/opt/miniconda3/envs/tbmi26:
davab27@ad-mac0558: ~
```

Remove the conda environment. This will not delete the downloaded packages.

```
[davab27@ad-mac0558: ~ conda env remove -n tbmi26
Remove all packages in environment /Users/davab27/opt/miniconda3/envs/tbmi26:
[davab27@ad-mac0558: ~ conda clean --all
Cache location: /Users/davab27/opt/miniconda3/pkgs
Will remove the following tarballs:
/Users/davab27/opt/miniconda3/pkgs
-----
parso-0.8.3-pyhd3eb1b0_0.conda          70 KB
ipykernel-6.4.1-py37hecd8cb5_1.conda    191 KB
decorator-5.1.0-pyhd3eb1b0_0.conda       14 KB
argcomplete-1.12.3-pyhd3eb1b0_0.conda    35 KB
jupyter_core-4.6.3-py37_0.conda          70 KB
importlib-metadata-4.8.2-py37hecd8cb5_0.conda 39 KB
wcwidth-0.2.5-pyhd3eb1b0_0.conda        26 KB
pyrsistent-0.18.0-py37hca72f7f_0.conda   93 KB
scipy-1.7.3-py37h8c7af03_0.conda        15.5 MB
matplotlib-3.5.0-py37hecd8cb5_0.conda    28 KB
pip-21.2.2-py37hecd8cb5_0.conda         1.8 MB
pyparsing-3.0.4-pyhd3eb1b0_0.conda       81 KB
giflib-5.2.1-haf1e3a3_0.conda            70 KB
blas-1.0-mkl.conda                      6 KB
```

Delete the conda packages used in the environment,
freeing up some hard drive space.

Azure setup

The following pages will go through the process for running Lab 2 on Microsoft Azure. Lab 2 (deep learning) requires a computer with GPU to train the models in a reasonable time. If you have a decent home computer you can run it on your own machine and skip these instructions.

Starting lab 2 in Azure

Please read fully before starting

Joining the lab and starting the machines

Mail - David Abramian - Outlook

outlook.office.com/mail/inbox/id/AAQkADU1MjFmYzqxLTliMDUtNDRiZi1iYWQyLTFkNDliNmJIZWE1MQAQAIHNWOsXhspCmm0tdZwjFtI...

LINKÖPINGS
UNIVERSITET

Outlook Search

New message Delete Archive Junk Sweep Move to Categorize Snooze ...

Favorites Register for Lab - TBMI26 2021

Sent Items Microsoft Azure Tue 2021-01-19 5:13 PM To: David Abramian

Drafts

Inbox 2 Microsoft Azure

Add favorite

Folders

Inbox 2 Press to access your virtual machine.

Drafts

Sent Items

Deleted Items 60 WM paper

Junk Email 53

Archive

Notes

Microsoft Azure

David Abramian invited you to the lab:
TBMI26 2021

Register now to access the virtual machines in the lab.

[Register for the lab >](#)

f t v in

[Privacy Statement](#)

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

Microsoft

Reply | Forward

Mail - David Abramian - Outlook My virtual machines - Azure Lab Services

labs.azure.com/virtualmachines

Azure Lab Services ? DA

My virtual machines

TBMI26 2021

0.4 / 4 hour(s) used

Stopped

Start

Start the virtual machine.

Feedback

Mail - David Abramian - Outlook x My virtual machines - Azure Lab Services x +

labs.azure.com/virtualmachines

Azure Lab Services ? DA

My virtual machines

TBMI26 2021

0.4 / 4 hour(s) used

Starting... ...

It may take a few minutes.

Feedback

Mail - David Abramian - Outlook My virtual machines - Azure Lab Services

labs.azure.com/virtualmachines

Azure Lab Services

?

DA

My virtual machines

TBMI26 2021

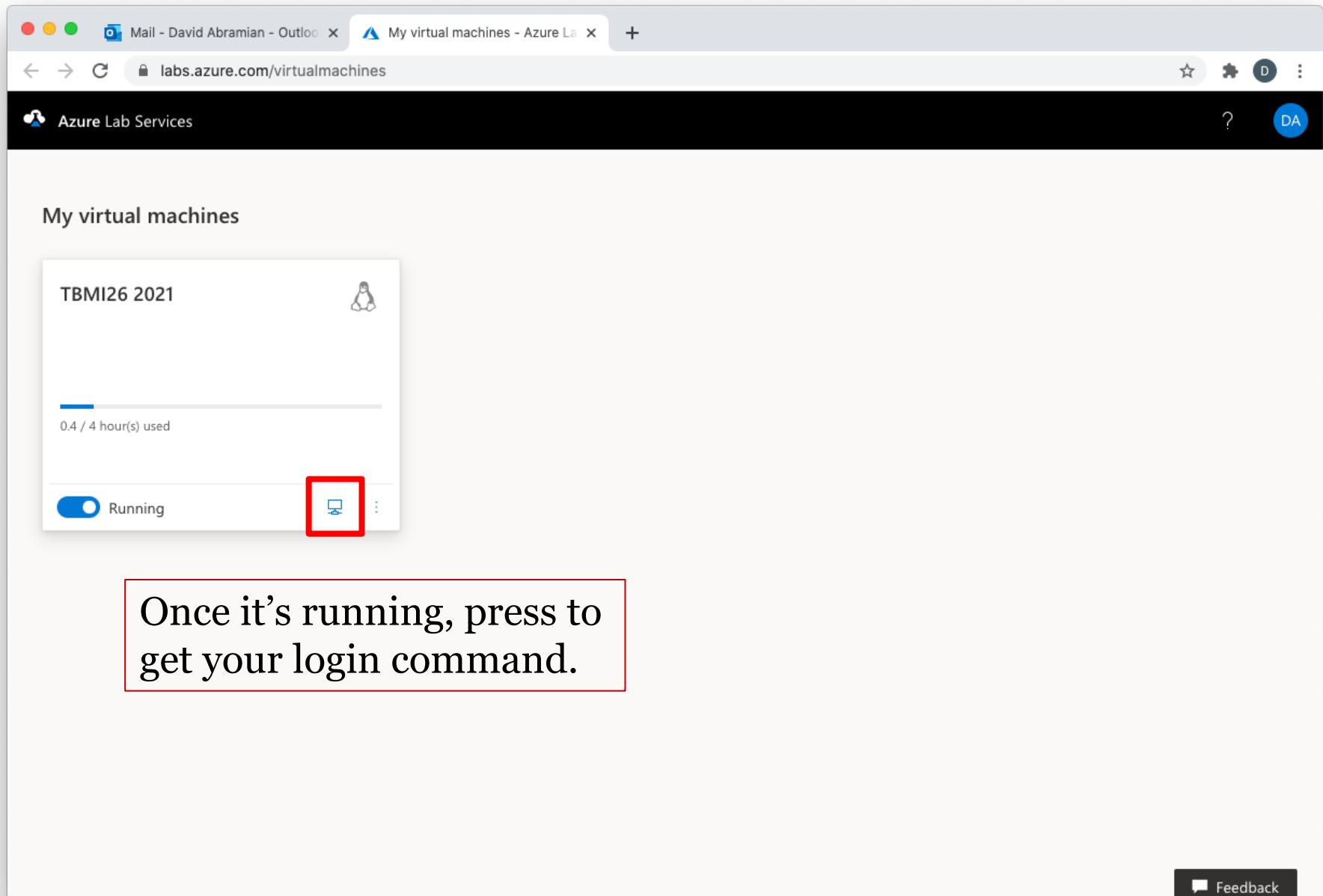
0.4 / 4 hour(s) used

Running

... 

Once it's running, press to get your login command.

Feedback



A screenshot of a web browser window titled "My virtual machines - Azure Lab Services" at the URL "labs.azure.com/virtualmachines". The main content area shows a list of virtual machines, with one named "TBMI26 2021" highlighted. A modal dialog box is overlaid on the page, titled "Reset password". The dialog contains instructions: "Enter a new password to be used when logging in. Resetting the password may take several minutes." It has fields for "Username" (set to "tbmi26") and "Password" (an empty input field). At the bottom are "Reset password" and "Cancel" buttons.

The first time you will be prompted to set your password. This will take a few minutes.

Feedback

Mail - David Abramian - Outlook My virtual machines - Azure Lab Services

labs.azure.com/virtualmachines

Azure Lab Services DA

My virtual machines

TBMI26 2021

0.4 / 4 hour(s) used

Running

Connect to your virtual machine

To connect to your Linux virtual machine using SSH, use the following command:

```
ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
```

Copy

Done

This is your login command.
Copy it entirely.

Feedback

Logging in with Linux/Mac

Mail - David Abramian - Outlook | My virtual machines - Azure Lab | Neural Networks and Learning | +

liuonline.sharepoint.com/sites/Lisam_TBMI26_2021VT_OR/CourseDocuments/Forms/AllItems.aspx?viewid=390fc32a-6dce-4888-b226-9a... ☆ D :

SharePoint Search this library

Lisam / Courses and programs / Neural Networks and Learning Systems (TBMI26 2021VT OR)

NN Neural Networks and Learning Systems (TBMI26 2021VT OR) Private group

Home

Course plan

Lecture videos

Course documents

Collaborative workspa...

Members and groups

Schedule

Teams

Signup

Submissions

Assessment record

Edit

+ New ▾ Upload ▾ Edit in grid view Share Copy link Sync ... All Documents ▾

Course documents > Assignments > Lab_2 > files

Name	Modified	Modified By	Version	Description
azure_launch_jupyter.bat	16 minutes ago	David Abramian	1.0	
azure_launch_jupyter.sh	16 minutes ago	David Abramian	1.0	
lab_gpu.yml	17 minutes ago	David Abramian	1.0	
lab_macos.yml	17 minutes ago	David Abramian	1.0	
lab_nogpu.yml	17 minutes ago	David Abramian	1.0	

Go to Lisam and download this file. Note where it is saved.

 davab27 — bash — 80x24

davab27@ad-mac0558: ~

Open a terminal.

```
davab27@ad-mac0558: ~ bash Downloads/azure_launch_jupyter.sh ssh -p 63459 tbmi26  
@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
```

Input the following command:

- bash
- the path to the launch script
- the entire Azure login command

```
[davab27 — ssh + bash Downloads/azure_launch_jupyter.sh ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.c...]  
[davab27@ad-mac0558: ~ bash Downloads/azure_launch_jupyter.sh ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com  
The authenticity of host '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459 ([51.124.10.211]:63459)' can't be established.  
ECDSA key fingerprint is SHA256:JIALBD7yf+MhNNEZGZ1A9+798AgafrKpBBc7ozHr400.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

You will get a warning about
accessing an unknown host. Say **yes**.

```
davab27 — ssh + bash Downloads/azure_launch_jupyter.sh ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.c...
[davab27@ad-mac0558: ~ bash Downloads/azure_launch_jupyter.sh ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
The authenticity of host '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459 ([51.124.10.211]:63459)' can't be established.
ECDSA key fingerprint is SHA256:JIALBD7yf+MhNNEZGZ1A9+798AgafrKpBBc7ozHr400.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459,[51.124.10.211]:63459' (ECDSA) to the list of known hosts.
tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com's password: 
```

You will be prompted for your password (the one from slide 7). Note that **nothing** will be printed as you type!

```
davab27 — ssh + bash Downloads/azure_launch_jupyter.sh ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.c...
[davab27@ad-mac0558: ~ bash Downloads/azure_launch_jupyter.sh ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
The authenticity of host '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459 ([51.124.10.211]:63459)' can't be established.
ECDSA key fingerprint is SHA256:JIALBD7yf+MhNNEZGZ1A9+798AgafrKpBBc7ozHr400.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459,[51.124.10.211]:63459' (ECDSA) to the list of known hosts.
[tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]
's password:
[I 17:20:20.441 LabApp] JupyterLab extension loaded from /anaconda/envs/lab_test/lib/python3.8/site-packages/jupyterlab
[I 17:20:20.441 LabApp] JupyterLab application directory is /anaconda/envs/lab_test/share/jupyter/lab
[I 17:20:20.443 LabApp] Serving notebooks from local directory: /home/tbmi26/lab
[I 17:20:20.443 LabApp] Jupyter Notebook 6.1.5 is running at:
[I 17:20:20.443 LabApp] http://localhost:8888/
[I 17:20:20.443 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Once you log in, Jupyter will be launched automatically. This can take a little time.

Logging in with Windows 10

Mail - David Abramian - Outlook | My virtual machines - Azure La | Neural Networks and Learning | +

liuonline.sharepoint.com/sites/Lisam_TBMI26_2021VT_OR/CourseDocuments/Forms/AllItems.aspx?viewid=390fc32a-6dce-4888-b226-9a... ☆ D :

SharePoint Search this library

Lisam / Courses and programs / Neural Networks and Learning Systems (TBMI26 2021VT OR)

NN Neural Networks and Learning Systems (TBMI26 2021VT OR) Private group

Home

Course plan

Lecture videos

Course documents

Collaborative workspa...

Members and groups

Schedule

Teams

Signup

Submissions

Assessment record

Edit

+ New ▾ Upload ▾ Edit in grid view Share Copy link Sync ... All Documents ▾

Course documents > Assignments > Lab_2 > files

Name	Modified	Modified By	Version	Description
azure_launch_jupyter.bat	16 minutes ago	David Abramian	1.0	
azure_launch_jupyter.sh	16 minutes ago	David Abramian	1.0	
lab_gpu.yml	17 minutes ago	David Abramian	1.0	
lab_macos.yml	17 minutes ago	David Abramian	1.0	
lab_nogpu.yml	17 minutes ago	David Abramian	1.0	

Go to Lisam and download this file. Note where it is saved.

Command Prompt

C:\Users\David>.

Open a command line.

```
Command Prompt
C:\Users\David>Downloads\azure_launch_jupyter.bat ssh -p 63459 tbmi26@ml-lab-47c4597
b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
```

Input the following command:
- the path to the launch script
- the entire Azure login command

```
Command Prompt - Downloads\azure_launch_jupyter.bat ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
C:\Users\David>Downloads\azure_launch_jupyter.bat ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
The authenticity of host '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459 ([51.124.10.211]:63459)' can't be established.
ECDSA key fingerprint is SHA256:JIALBD7yf+MhNNEZGZ1A9+798AgafKpBBc7ozHr400.
Are you sure you want to continue connecting (yes/no)? yes
```

You will get a warning about
accessing an unknown host. Say **yes**.

```
C:\ Command Prompt - Downloads\azure_launch_jupyter.bat ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com - X
C:\Users\David>Downloads\azure_launch_jupyter.bat ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
The authenticity of host '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459 ([51.124.10.211]:63459)' can't be established.
ECDSA key fingerprint is SHA256:JIALBD7yf+MhNNEZGZ1A9+798AgafKpBBc7ozHr400.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459,[51.124.10.211]:63459' (ECDSA) to the list of known hosts.

tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com's password: ■
```

You will be prompted for your password (the one from slide 7). Note that **nothing** will be printed as you type!

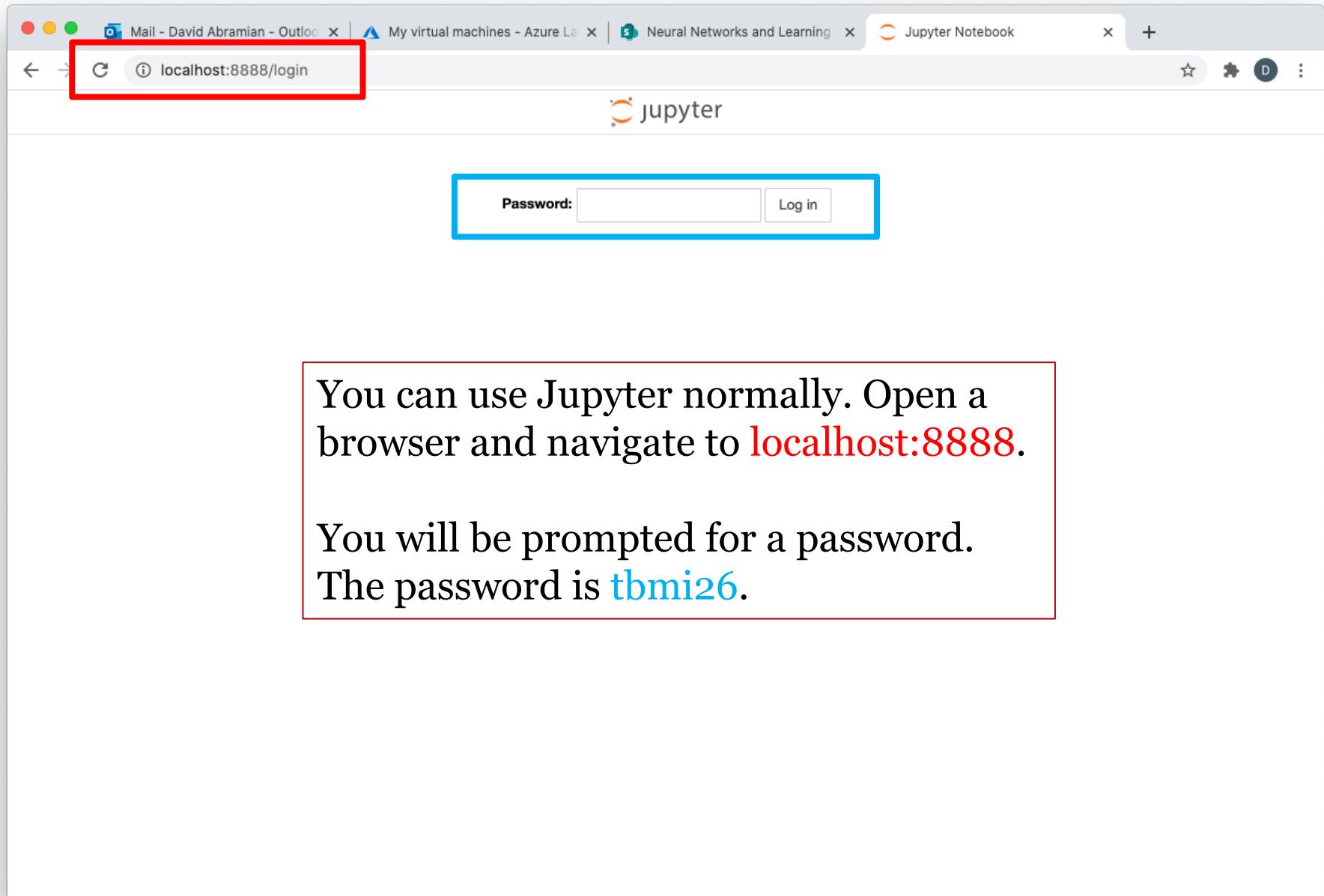
OpenSSH SSH client

```
C:\Users\David>Downloads\azure_launch_jupyter.bat ssh -p 63459 tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com
The authenticity of host '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459 ([51.124.10.211]:63459)' can't be established.
ECDSA key fingerprint is SHA256:JIALBD7yf+MhNNEZGZ1A9+798AgafKpBBc7ozHr400.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com]:63459,[51.124.10.211]:63459' (ECDSA) to the list of known hosts.

tbmi26@ml-lab-47c4597b-3725-457f-bf9c-0038ff934f32.westeurope.cloudapp.azure.com's password:
[I 17:41:38.501 LabApp] JupyterLab extension loaded from /anaconda/envs/lab_test/lib/python3.8/site-packages/jupyterlab
[I 17:41:38.502 LabApp] JupyterLab application directory is /anaconda/envs/lab_test/share/jupyter/lab
[I 17:41:38.504 LabApp] Serving notebooks from local directory: /home/tbmi26/lab
[I 17:41:38.504 LabApp] Jupyter Notebook 6.1.5 is running at:
[I 17:41:38.504 LabApp] http://localhost:8888/
[I 17:41:38.504 LabApp] Use Control-C to stop this server and shut down all kernels
(twice to skip confirmation).
```

Once you log in, Jupyter will be launched automatically. This can take a little time.

Running the lab



The screenshot shows a JupyterLab interface running on a Mac OS X system. The top bar displays several open tabs: 'Mail - David Abramian - Outlook', 'My virtual machines - Azure Lab', 'Neural Networks and Learning', and 'JupyterLab'. The main window has a toolbar with File, Edit, View, Run, Kernel, Tabs, Settings, and Help. A sidebar on the left shows a file tree with three files: 'CIFAR10-Lab.ipynb', 'Custom.py', and 'MNIST-Demo.ipynb' (which is currently selected). The main content area displays a notebook cell with the title '0. Quick introduction to jupyter notebooks' and a bulleted list of instructions:

- Each cell in this notebook contains either code or text.
- You can run a cell by pressing Ctrl-Enter, or run and advance to the next cell with Shift-Enter.
- Code cells will print their output, including images, below the cell. Running it again deletes the previous output, so be careful if you want to save some results.
- You don't have to rerun all cells to test changes, just rerun the cell you have made changes to. Some exceptions might apply, for example if you overwrite variables from previous cells, but in general this will work.
- If all else fails, use the "Kernel" menu and select "Restart Kernel and Clear All Output". You can also use this menu to run all cells.

0.5 Some hardware setup

Keras uses all available GPUs in your computer. The following `os.environ` commands configures that only one of them should be used. If you are on a system with several GPUs and want to use more than one, you can change or comment out these commands.

By default, Keras will allocate all of the available memory in the device. The last two lines will have Keras allocate memory as needed.

```
[ ]: import os
import tensorflow as tf

# If there are multiple GPUs and we only want to use one/some, set the number in the visible device
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"

# This sets the GPU to allocate memory only as needed
physical_devices = tf.config.experimental.list_physical_devices('GPU')
if len(physical_devices) != 0:
    tf.config.experimental.set_memory_growth(physical_devices[0], True)
```

0 5 2 Python 3 | Idle Mode: Command ⌘ Ln 1, Col 1 MNIST-Demo.ipynb

When you are done

Mail - David Abramian - Outlook | My virtual machines - Azure Lab Services | Neural Networks and Learning | JupyterLab | +

labs.azure.com/virtualmachines

Azure Lab Services ? DA

My virtual machines

TBMI26 2021

0.5 / 4 hour(s) used

Stopping...

Remember to shut down the virtual machine! Your time is used as long as it is on.

Feedback