LabMLISP: Lab Course Machine Learning in Signal Processing

Exercise I: Getting started

Prof. Veniamin Morgenshtern

Author: Kamal Nambiar

Overview of Lab Tasks

Your goal in this lab course is to create a basic lane detection system using deep learning. You will be given a template of such system with the algorithmic parts erased. At the minimum, your goal will be to re-create those parts. If you accomplish this task early, you will have the opportunity to participate in the research project by helping to extend the system and develop it further.

The tasks in the lab are divided organized into 6 exercises. Upon completion of each exercise (except exercise 1), you are required to show your work to the lab tutor and upload the results to StudOn. Here is a check list of what you should accomplish:

1. Setup your working environment. Use the instructions provided below in this document. The lane detection system that we will build in this lab uses the object oriented programming framework. If you are not comfortable with object oriented programming in Python, we provide a short tutorial for you and a link to a longer tutorial if needed. Please make sure you master the object oriented programming concepts while working on the simulator.

2. Develop a *road simulator*, that will produce artificial images of a road with lane markings depicted. For each image, the simulator must also produce a binary mask of the same resolution. The pixel in the mask is equal to 1, if the corresponding pixel in the image belongs to the road lane, and zero otherwise. You will be provided a Jupyter notebook that can be used to test various components of the simulator and this notebook has to be uploaded to StudOn after you complete the simulator task.

3. The lane detection system that you will be building is based on deep learning. Read the online book "Neural Networks and Deep Learning" by Michael Nielsen. The book contains exercises. You will be given a list of exercises to solve and you will need to timely submit well-written solutions, either in LaTeX (for theoretical exercise) or as a Jupyter notebook for programming exercises. If you are not comfortable with LaTeX, you could also solve the theoretical exercises on paper and upload a scanned version of your solutions.

4. Study *Pytorch*, a powerful and easy to use deep learning framework. We prepared a tutorial for you.

5. Create a neural network for lane detection in Pytorch.

    - Implement a custom dataloader, neural network architecture, loss function and evaluation metrics using PyTorch.
    - Train the network using the data from the road simulator. Visualize the training process in Tensorboard.

- Implement data augmentations in order to introduce more diversity in the training data.
- Test the performance of the trained network on new images (simulated and real) unseen by the network. Visualize the results. At this stage, you can upload your training Jupyter notebook to StudOn.

6. Hyperparameter search and prediction on real driving video.
    - Train the network on the mix of real world and artificial images. Test the performance of the network on real road images. Were you able to reduce the amount of false positives?
    - Perform various experiments in order to find the optimal hyperparameters for your network. Systematically record the results of each experiment that you perform in a document.
    - Perform frame-wise prediction on a real driving video using the optimal model from your hyperparameter search experiments. Upload the document with the results of your experiments and the predict Jupyter notebook with predictions from your best model.

The lane detection results for real images doesn't have to be perfect, but they should be meaningful.

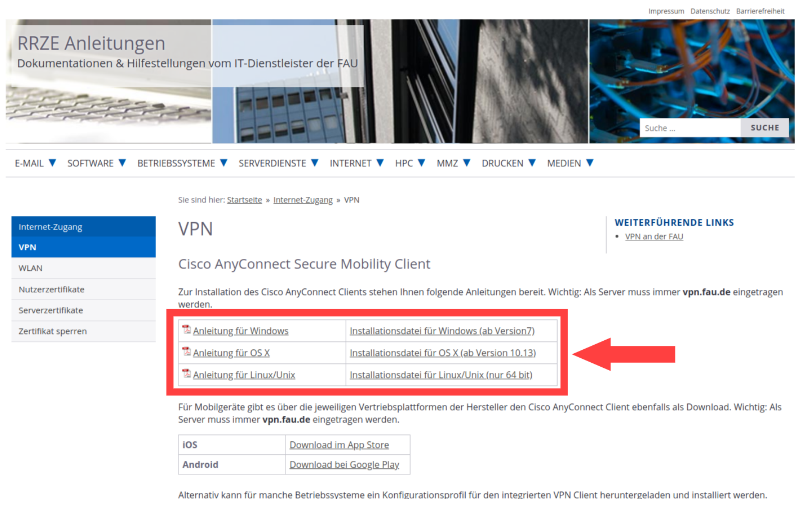
Once you are done with the plan above, you can help develop the system further. One interesting project might be to extend the road simulator to produce short artificial driving videos, and then use these artificial fragments to train a 3D network (x-y-time) that would work with a stack of images over time. This should significantly improve the performance.

Getting Started

In this lab we will be using the machine at *lms37-22.e-technik.uni-erlangen.de* to do our experiments.

Connecting to the university network

In order to access the remote machine, it is necessary to connect to the university network using a VPN. The official VPN client can be downloaded from: `https://www.anleitungen.rrze.fau.de/internet-zugang/vpn/`


Impressum Datenschutz Barrierefreiheit
RRZE Anleitungen
Dokumentationen & Hilfestellungen vom IT-Dienstleister der FAU
Suche ...
SUCHE
E-MAIL SOFTWARE BETRIEBSSYSTEME SERVERDIENSTE INTERNET HPC MMZ DRUCKEN MEDIEN
Internet-Zugang
VPN
WLAN
Nutzerzertifikate
Serverzertifikate
Zertifikat sperren
WEITERFÜHRENDE LINKS
VPN an der FAU

Sie sind hier: Startseite » Internet-Zugang » VPN

VPN

Cisco AnyConnect Secure Mobility Client

Zur Installation des Cisco AnyConnect Clients stehen Ihnen folgende Anleitungen bereit. Wichtig: Als Server muss immer **vpn.fau.de** eingetragen werden.

Anleitung für Windows	Installationsdatei für Windows (ab Version7)
Anleitung für OS X	Installationsdatei für OS X (ab Version 10.13)
Anleitung für Linux/Unix	Installationsdatei für Linux/Unix (nur 64 bit)

Für Mobilgeräte gibt es über die jeweiligen Vertriebsplattformen der Hersteller den Cisco AnyConnect Client ebenfalls als Download. Wichtig: Als Server muss immer **vpn.fau.de** eingetragen werden.

iOS	Download im App Store
Android	Download bei Google Play

Alternativ kann für manche Betriebssysteme ein Konfigurationsprofil für den integrierten VPN Client heruntergeladen und installiert werden.

Download the instructions ('Anleitung') and the installer according to your operating system, and follow the instructions to install the VPN Client.

Without being connected through the VPN, you won't be able to connect to the lab machine. Instead, you will receive an error message like the following:

```
ssh: connect to host lms37-22.e-technik.uni-erlangen.de port 22: Network is unreachable
```

Connecting to the remote system

Open the command prompt in Windows or the terminal in Linux or Mac OS, and login via ssh using the port number (`portnum`) and `username` provided to you:

```
$ ssh -L localhost:portnum:localhost:portnum username@lms37-22.e-technik.uni-erlangen.de
```

Example:

```
$ ssh -L localhost:8920:localhost:8920 mlisp-1@lms37-22.e-technik.uni-erlangen.de
```

NOTE: If you are using an older version of windows and/or are unable to access the lab machine via command prompt, please follow the PuTTY setup instructions on page 7.

Environment Setup On Lab Machine

The below instructions are for the initial setup of your working environment on the lab machine. The next time you login to this machine, we can navigate to your working directory(created below) and continue with your work.

- Create your working directory and change your current directory to this newly created directory.
  ```
  $ mkdir labmlisp

  $ cd labmlisp
  ```

- Copy files from the shared folder `/SHARED/DATA/mlisp-lab/`
  NOTE: Do not forget the dot '.' at the end of the statement. If you are copy-pasting the commands from this document into the terminal, pay attention to all special characters in the command (like the underscore '_').
  ```
  $ cp ~/SHARED/DATA/mlisp-lab/ps1-dependencies/*.txt .
  ```

- All python modules required for the lab can be installed in two stages. Install all conda modules listed in the `conda-package-list.txt` using:
  ```
  $ conda install --file conda-package-list.txt
  ```

- Install the remaining dependencies from `requirements.txt` using pip command.
  ```
  $ pip install -r requirements.txt
  ```

Jupyter Notebook

In this lab, we recommend the students to use Jupyter Notebook for code development. You will be provided with a few notebooks that will help you test your implementation. If you have not worked with Jupyter Notebook before, please take the time to familiarize yourself with Jupyter Notebook using resources available online.

- Before you launch Jupyter Notebook, make sure that you are in working directory. You can use the `pwd` command to verify if you are not sure. Also be sure that the conda environment that you had created earlier is activated.
- Launch Jupyter Notebook using the port number (`portnum`) and GPU number (`gpunum`) provided to you along with your login credentials.
  NOTE: Pay attention to the underscore '_' in the command while copy-pasting.
  ```
  $ CUDA_VISIBLE_DEVICES=gpunum jupyter notebook --no-browser --port=portnum
  ```
  Example:
  ```
  $ CUDA_VISIBLE_DEVICES=7 jupyter notebook --no-browser --port=8920
  ```
- The previous command will print a long web link, which you should now open in the web browser on your local machine.
- You can use upload button on the top right corner of the page to copy all the lab materials that you may require for your code development to your working directory.
- Once you have completed your work and are leaving for the day, close Jupyter by entering Ctrl c in the terminal.
- Jupyter has many useful hot keys, for example the hot key for commenting multiple lines of code is Ctrl / . Note that this only works on English keyboard. Here is a useful cheat sheet: `https://www.cheatography.com/weidadeyue/cheat-sheets/jupyter-notebook/pdf_bw/`
- If you are using a German keyboard, here is a tip: `https://stackoverflow.com/questions/26504796/how-can-i-block-comment-code-in-an-ipython-notebook-with-a-german-keyboard`

Study object-oriented programming

The system you will build will use the object oriented programming framework. If you are not familiar with object oriented programming in Python, begin by going through a short tutorial notebook that is available at the shared folder `/SHARED/DATA/mlisp-lab/ps1-oop-tutorial/`. Copy the notebook by running the following command in your working directory.

```
$ cp ~/SHARED/DATA/mlisp-lab/ps1-oop-tutorial/*.ipynb .
```

If needed, here is a link to a more comprehensive object-oriented programming tutorial:

`https://python.swaroopch.com/oop.html`

Study debugging in Python / Jupyter

In this laboratory you will be asked to complete a lot of code. It's best to make your work modular and after implementing each function or class test that the function you have implemented works as expected.

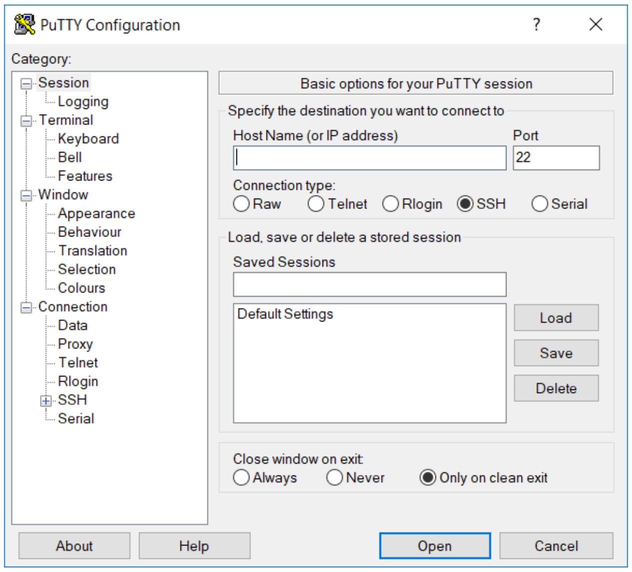
Even if your code does not work as a whole, you can still test individual functions. For this a built-in Jupyter debugger is very useful. In the end of the `OOP_Tutorial.ipynb` notebook you will find a simple example on how to use the debugger.

Alternate remote login option for Windows users

Connecting to the remote system using PuTTY

You can also connect to the lab machine from your windows computer using the ssh client ‘PuTTY’. Download and install PuTTY using the installer (.msi file) from `https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html`.

A configuration window as shown below will open when you launch PuTTY.



Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

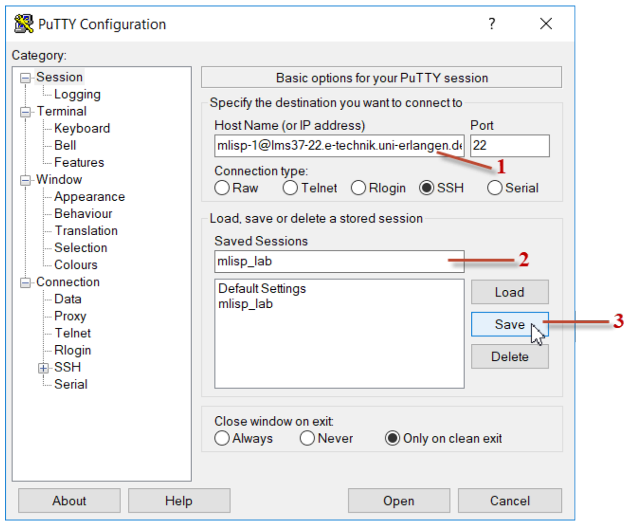
Saved Sessions:

Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

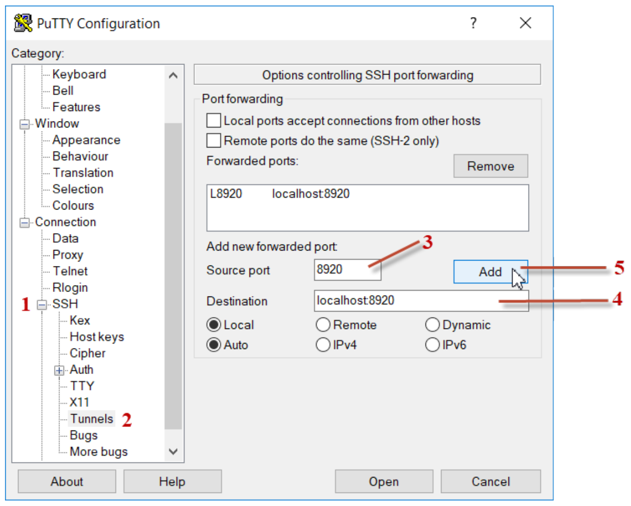
Enter the following details in the configuration window:

1. **Host Name (or IP address)** : username@lms37-22.e-technik.uni-erlangen.de
   Example: *mlisp-1@lms37-22.e-technik.uni-erlangen.de*
2. **Saved Sessions** : mlisp_lab
3. Hit the **Save** button on the right



Specify the destination you want to connect to
Host Name (or IP address): mlisp-1@lms37-22.e-technik.uni-erlangen.d
Port: 22
Connection type: ( ) Raw ( ) Telnet ( ) Rlogin [✓ SSH] ( ) Serial
Load, save or delete a stored session
Saved Sessions: mlisp_lab
Default Settings
mlisp_lab
Close window on exit: ( ) Always ( ) Never [✓ Only on clean exit]

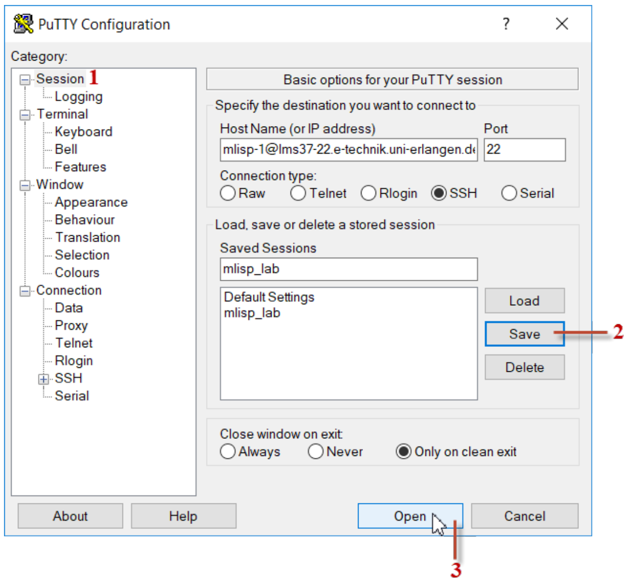
In this lab, you will be running your code in a Jupyter notebook on the lab machine. To view the notebook on your local machine, we need to set up port forwarding.



Port forwarding
[ ] Local ports accept connections from other hosts
[ ] Remote ports do the same (SSH-2 only)
Forwarded ports:
L8920 localhost:8920
Add new forwarded port:
Source port: 8920
Destination: localhost:8920
[✓ Local] ( ) Remote ( ) Dynamic
[✓ Auto] ( ) IPv4 ( ) IPv6

1. On the left ‘Category’ panel, hit the + button next to SSH option
2. From the sub-menu, select the **Tunnels** option
3. **Source port** : Enter the port number provided to you
   Example: *8920*
4. **Destination** : localhost:port_number
   Example: *localhost:8920*
5. Hit the **Add** button on the right. Once this is completed, you will find an entry in the rectangular box above.

Save and Open the session



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): mlisp-1@lms37-22.e-technik.uni-erlangen.de

Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

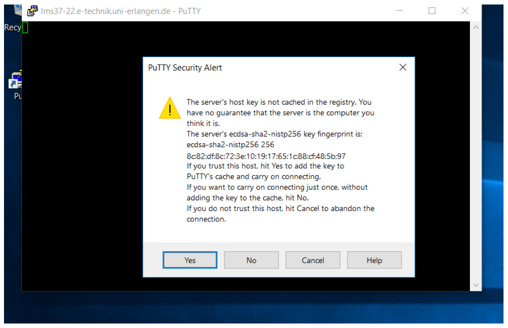
Saved Sessions: mlisp_lab

Default Settings
mlisp_lab

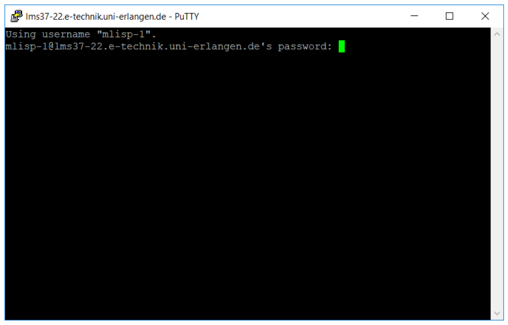
Close window on exit: Always Never [✓ Only on clean exit]

1. On the left ‘Category’ panel, hit the **Session** option
2. Hit the **Save** button on the right
3. Hit the **Open** button at the bottom

Once the connection is established, you will see a security alert window. Hit '**Yes**' and continue. Provide the password when prompted and continue with the rest of instructions from the **Environment Setup On Lab Machine** section



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's ecdsa-sha2-nistp256 key fingerprint is:
ecdsa-sha2-nistp256 256
8c:82:df:8c:72:3e:10:19:17:65:1c:88:cf:48:5b:97
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.



```
Using username "mlisp-1".
mlisp-1@lms37-22.e-technik.uni-erlangen.de's password:
```

.

NOTE: The above instructions are only for initial setup of PuTTY ssh connection. The next time you launch PuTTY, you can load your saved settings by selecting the saved session name (mlisp_lab) from the list and hitting the **Load** button on the right. Subsequently hit the **Open** button to lauch new session.