

Programvaretips for dyp læring

Ivar Thokle Hovden

Begreper

GitHub

Docker

Docker Hub

Dockerfile

nvidia-docker v2

Anaconda

Numpy

PyTorch

Programvare-repo

Virtualisering i kontainere

Repo over kontainere

Lag en kontainer

GPU-støtte fra Docker-kontainer

Python pakkebehandler

Raskt matrisebibliotek i Python

Dyp læring i Python

Setup for grafikkakselerert DL på «vanlig» PC

1. Windows 10
Anaconda
2. Windows 7
Geforce 940M
Compute Capability 5
CUDA 9
Anaconda
3. Ubuntu 18.10 “Cosmic Cuttlefish”
Geforce RTX 2070
Compute Capability 7.5
CUDA 10
Docker + nvidia-docker2
Anaconda

nvidia-docker2 støtter ikke Windows,
men Docker kan kjøre på

Windows 10: <https://docs.docker.com/docker-for-windows/install> og

Windows 7: <https://docs.docker.com/toolbox/overview>

Windows 10

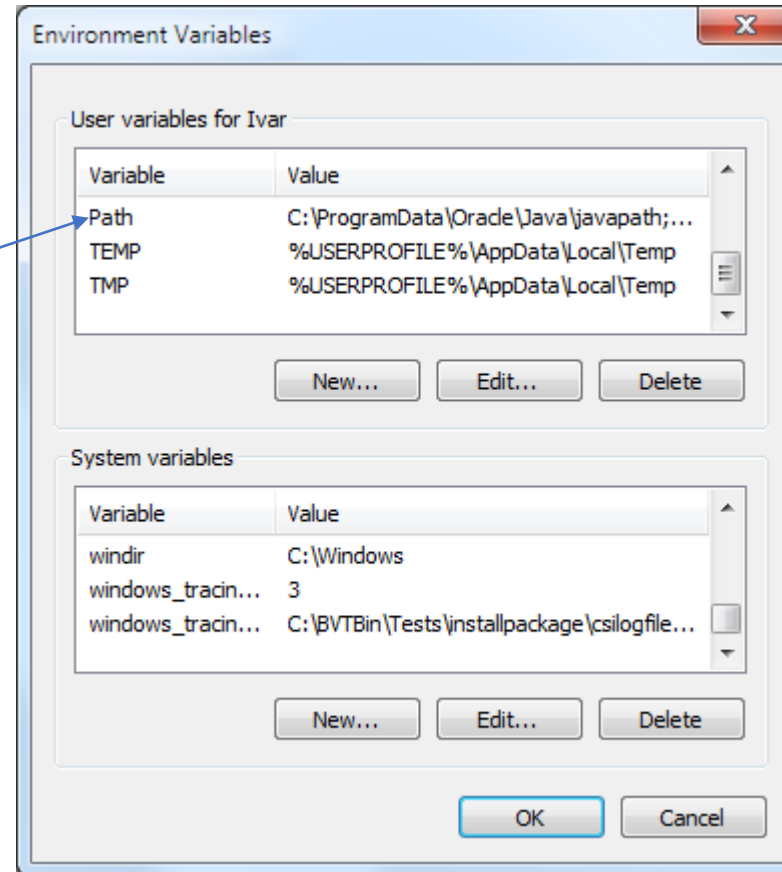
1. Installer proprietær Nvidia grafikkdriver
2. (valgfritt) Installer Windows Subsystem for Linux (WSL):
<https://docs.microsoft.com/en-us/windows/wsl/install-win10>
og installer Ubuntu fra WSL. Gir apt-get pakkebehandler.
Linux-verktøy tilsvarende Cygwin i Windows 7
3. (valgfritt) Installer chocolatey pakkebehandler:
<https://chocolatey.org/install>
<https://chocolatey.org/packages>
4. Installer Anaconda 2019.03 64 bit: <https://www.anaconda.com/distribution>
5. Installer PyTorch og andre pakker fra Anaconda:
<https://pytorch.org/get-started/locally>
6. Start Anaconda Navigator
7. Åpne JupyterLab og start eksperimenteringen

Windows 7

1. Installer proprietær Nvidia grafikkdriver
2. (valgfritt) Installer Cygwin (Linux-verktøy):
<https://cygwin.com/install.html>
3. (valgfritt) Installer chocolatey pakkebehandler:
<https://chocolatey.org/install>
<https://chocolatey.org/packages>
4. Installer Anaconda 2019.03 64 bit:
<https://www.anaconda.com/distribution>
5. Installer PyTorch andre pakker fra Anaconda:
<https://pytorch.org/get-started/locally>
6. Start Anaconda Navigator
7. Åpne JupyterLab og start eksperimenteringen

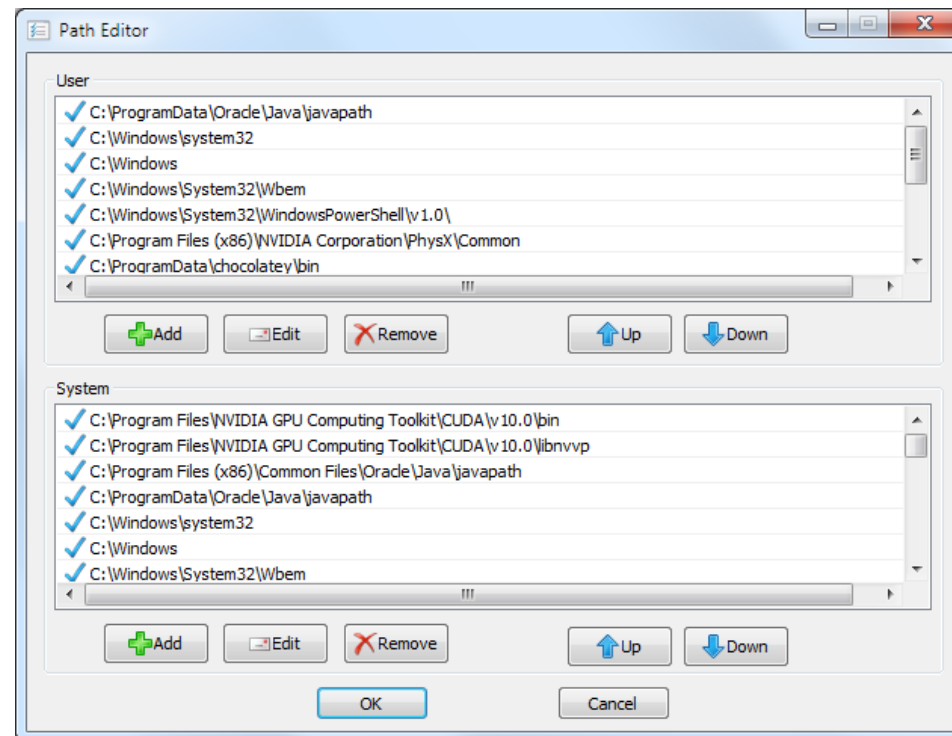
Windows 7 tips: environmental variables

- Windows + Pause
- Advanced system settings
- Environmental Variables
f. eks. Path variabel
eller JAVA_HOME

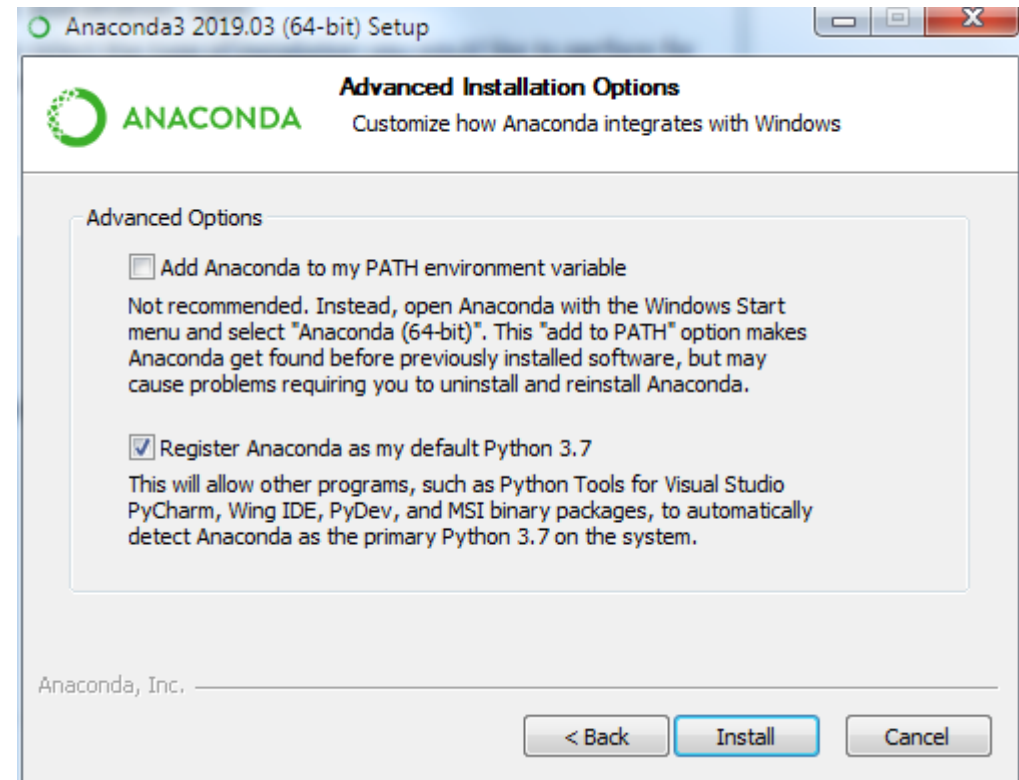
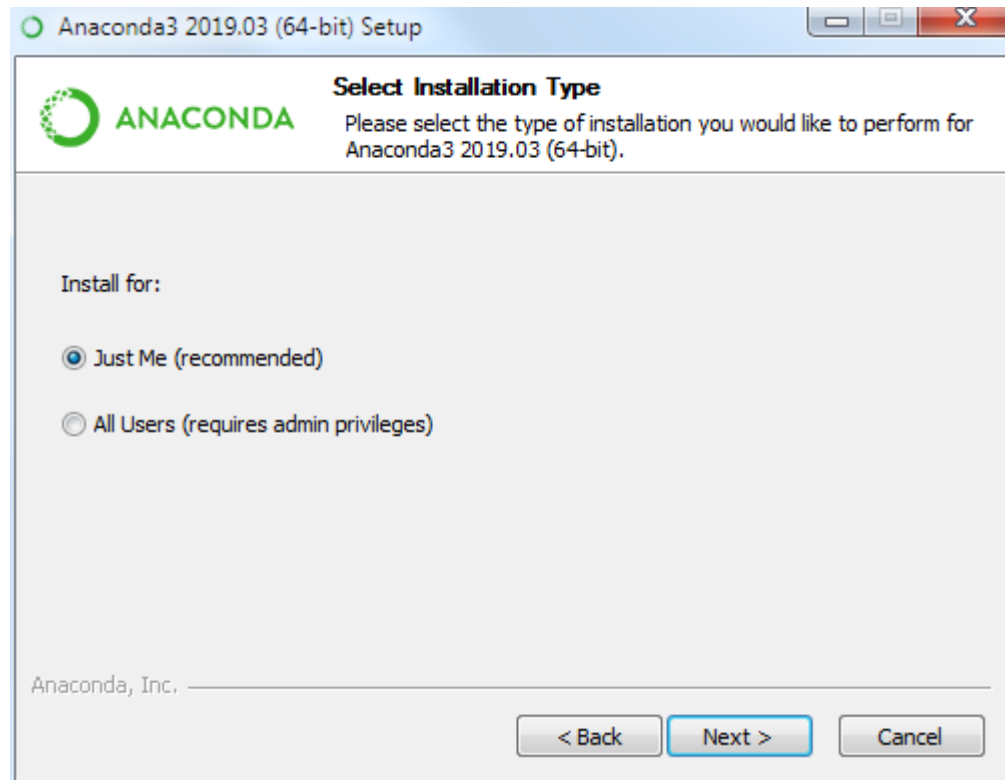


Windows 7 tips: patheditor

- Raskere måde å ændre Path-variabelen i Windows
 - > choco install patheditor
 - > patheditor



4. Installer Anaconda 2019.03 64 bit



Anaconda tips: pakkesøk

- Søk etter en pakke og hvordan installere den på <https://anaconda.org>

The screenshot shows the Anaconda Conda Forge package page for `tensorboardx` version 1.6. The page has a green header with the package name and version, and a navigation bar with icons for home, code, documentation, star, and a count of 0. Below the header, there's a section for 'tensorboard for pytorch' with tabs for Conda, Files, Labels, and Badges. The Conda tab is active, showing details like License (MIT), Home (GitHub), Development (GitHub), Documentation (ReadTheDocs), 40104 total downloads, and last upload date (3 months and 12 days ago). Below this is the 'Installers' section, which includes a warning about non-standard labels and a 'conda install' command. The command is shown in a code block with a 'noarch' label and a version 'v1.6'. Below the command, there's a description of the package: 'Write tensorboard events from PyTorch (and Chainer, MXNet, NumPy, ...)'. The page is styled with a light green and white color scheme.

conda-forge / packages / tensorboardx 1.6

tensorboard for pytorch

Conda Files Labels Badges

License: MIT
Home: <https://github.com/lanpa/tensorboardx>
Development: <https://github.com/lanpa/tensorboardx>
Documentation: <https://tensorboardx.readthedocs.io/en/latest/>
40104 total downloads
Last upload: 3 months and 12 days ago

Installers

Info: This package contains files in non-standard labels.

conda install ?

noarch v1.6

To install this package with conda run one of the following:

```
conda install -c conda-forge tensorboardx
conda install -c conda-forge/label/gcc7 tensorboardx
conda install -c conda-forge/label/cf201901 tensorboardx
```

Description

Write tensorboard events from PyTorch (and Chainer, MXNet, NumPy, ...)

5. Installer PyTorch og andre pakker

- Windows --> Anaconda Prompt
 - > conda install pytorch torchvision cudatoolkit=9.0 -c pytorch
 - > conda install -c conda-forge ipyvolum
 - > conda install -c conda-forge nodejs
 - > jupyter labextension install @jupyter-widgets/jupyterlab-manager
 - > jupyter labextension install ipyvolum
 - > jupyter labextension install jupyter-threejs
 - > conda install -c conda-forge tensorboardx
 - > conda install git

- <https://ipyvolume.readthedocs.io/en/latest/index.html>

Ubuntu 18.10

1. Installer proprietær Nvidia grafikkdriver
2. Installer Docker CE: <https://docs.docker.com/install/linux/docker-ce/ubuntu/>
3. Installer nvidia-docker v2: [https://github.com/nvidia/nvidia-docker/wiki/Installation-\(version-2.0\)](https://github.com/nvidia/nvidia-docker/wiki/Installation-(version-2.0))
4. Lag en Dockerfile basert på <https://github.com/jxcodetw/docker-jupyter-pytorch>
5. Bygg Dockerfile til bilde, kjør bilde
6. Åpne JupyterLab (<http://localhost:8888>) og Tensorboardx (<http://localhost:6006>) (kjører fra kontaineren), start eksperimentering in JupyterLab

Linux-tips: sudo uten passord

- sudo: “super user do”. Kjør kommando som administrator, som vanlig bruker.
- Krever passord hver gang som standard
- For brukere som tilhører gruppen «sudo»
- I /etc/sudoers , endre linjen
%sudo ALL=(ALL:ALL) ALL
til
%sudo ALL=(ALL:ALL) NOPASSWD:ALL
ved
- sudo gedit /etc/sudoers eller sudo visudo
- Slipper å skrive inn passord for hver gang man skriver sudo

1. Installer proprietær Nvidia grafikkdriver

```
$ sudo add-apt-repository ppa:graphics-drivers/ppa
```

Få repo med oppdaterte drivere

```
$ sudo apt update
```

```
$ ubuntu-drivers devices
```

Se hva som er tilgjengelig

```
== /sys/devices/pci0000:00/0000:00:01.0/0000:01:00.0 ==  
modalias : pci:v000010DEd00001F02sv00001458sd000037C2bc03sc00i00  
vendor   : NVIDIA Corporation  
driver   : nvidia-driver-418 - third-party free recommended  
driver   : nvidia-driver-415 - third-party free  
driver   : nvidia-driver-410 - third-party free  
driver   : xserver-xorg-video-nouveau - distro free builtin
```

```
$ sudo ubuntu-drivers autoinstall
```

Installer anbefalt driver

```
$ nvidia-smi
```

Verifiser at driver er installert

2. Installer Docker Community Edition (CE)

- \$ sudo apt-get remove docker docker-engine docker.io containerd runc (valgfritt) Avinstaller tidligere versjoner
- \$ sudo apt update
- \$ sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common Installer avhengigheter
- \$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add – Nødvendig GPG-nøkkel
- \$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable" Få repo for docker
- \$ sudo apt-get update
- \$ sudo apt-get install docker-ce docker-ce-cli containerd.io Installer siste versjon av Docker CE og containerd
- \$ sudo usermod -aG docker \$USER Legg \$USER til gruppen "docker" for å kjøre ikke-root
- \$ docker run hello-world Verifiser at docker er installert. Logg ut og inn først.

3. Installer nvidia-docker v2

```
$ docker volume ls -q -f driver=nvidia-docker | xargs -r -l{} -n1  
docker ps -q -a -f volume={} | xargs -r docker rm -f
```

(valgfritt) Fjern tidligere nvidia-docker volumer

```
$ sudo apt-get purge nvidia-docker
```

(valgfritt) Avinstaller tidligere versjoner

```
$ curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo  
apt-key add -
```

Nødvendig GPG-nøkkel

```
$ distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
```

Variabel til bruk i neste kommando

```
$ curl -s -L https://nvidia.github.io/nvidia-  
docker/$distribution/nvidia-docker.list | sudo tee  
/etc/apt/sources.list.d/nvidia-docker.list
```

Få nvidia-docker repo

```
$ sudo apt-get update
```

```
$ sudo apt-get install nvidia-docker2
```

Installer nvidia-docker2

```
$ sudo pkill -SIGHUP dockerd
```

Last inn dockerd-konfigurasjonen på nytt

```
$ docker run --runtime=nvidia --rm nvidia/cuda nvidia-smi
```

Verifiser driver at er installert inni docker

Numpy array triks

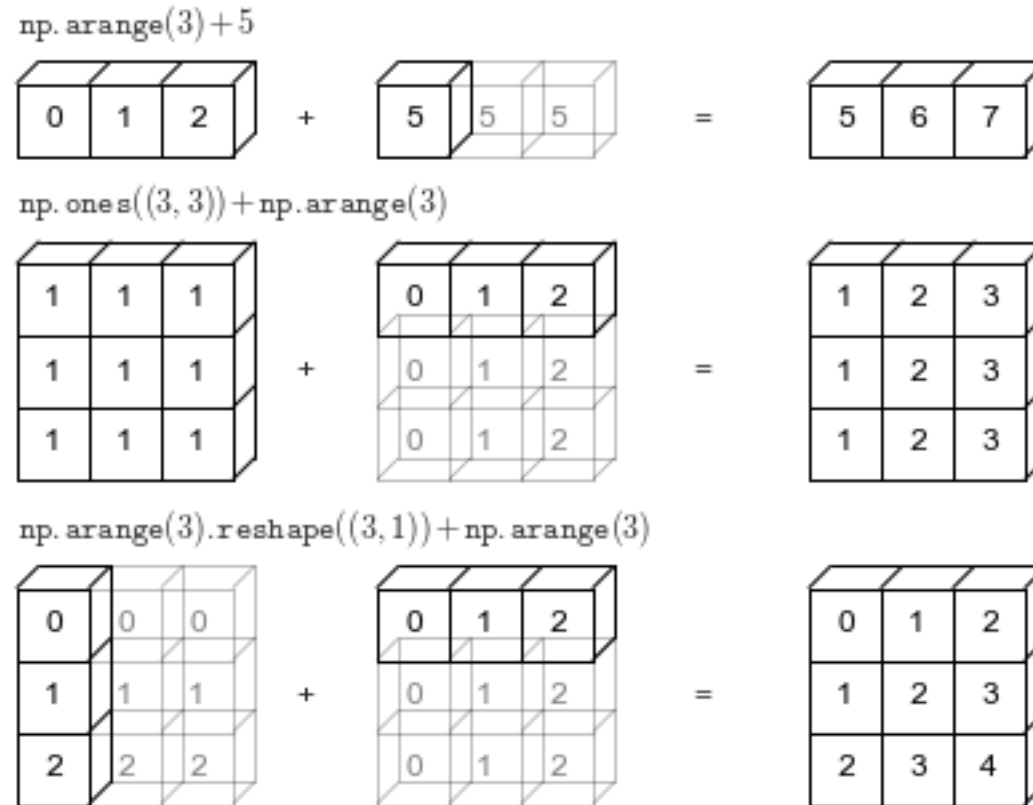
1. ufunc
2. aggregeringer
3. array broadcasting
4. slicing, masking, fancy indexing

Elementvis operasjon

min, max, sum, mean, std, etc.

Elementvis operasjon på arrayer med ulike dimensjoner (+, *, etc.)

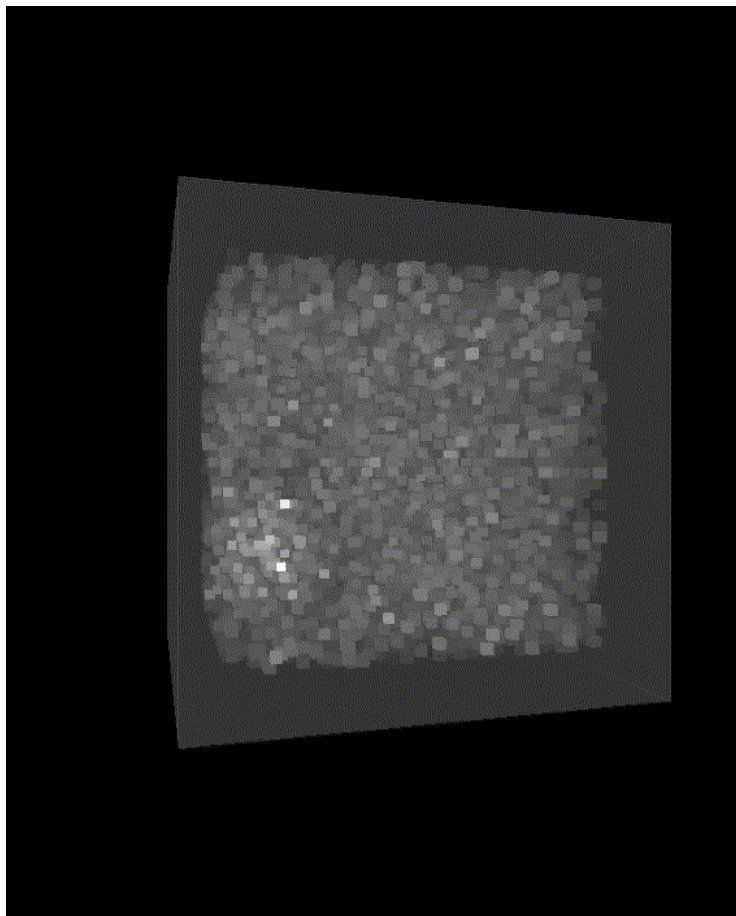
3. Array broadcasting



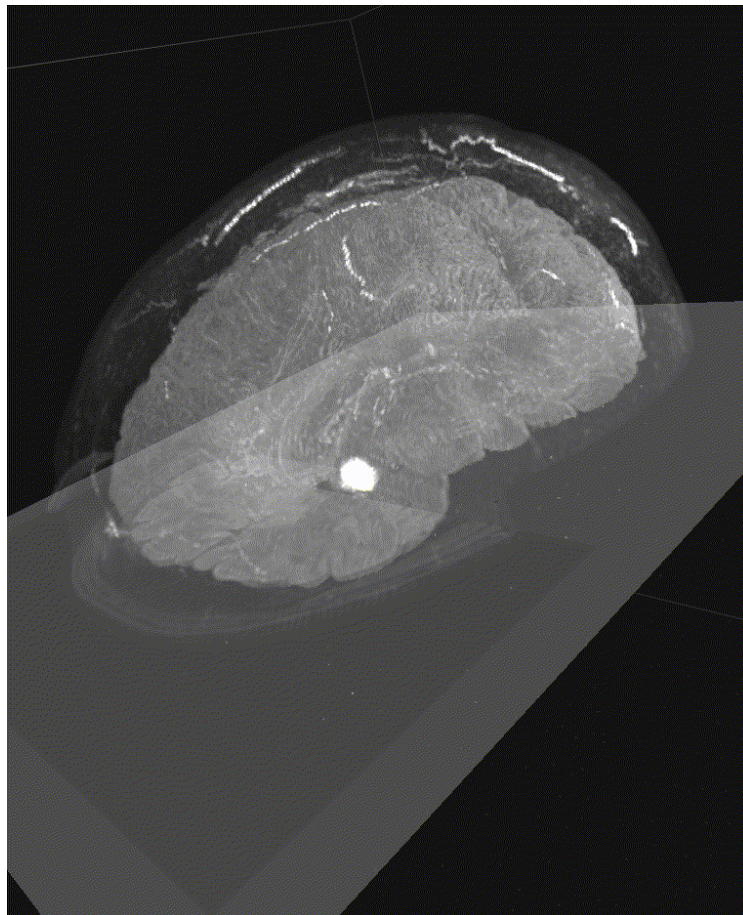
Visualisering av Numpy array

1. matplotlib, 2D
interaktiv med ipywidgets
2. ipyvolume : WebGL volumrendering, 3D, interaktiv
<https://ipyvolume.readthedocs.io/en/latest>
3. Spimagine : OpenGL volumrendering, 4D, GUI
<https://github.com/maweigert/spimagine>
I Docker: <https://github.com/ivartz/spimagine-jupyter-docker>
Til .gif med ffmpeg

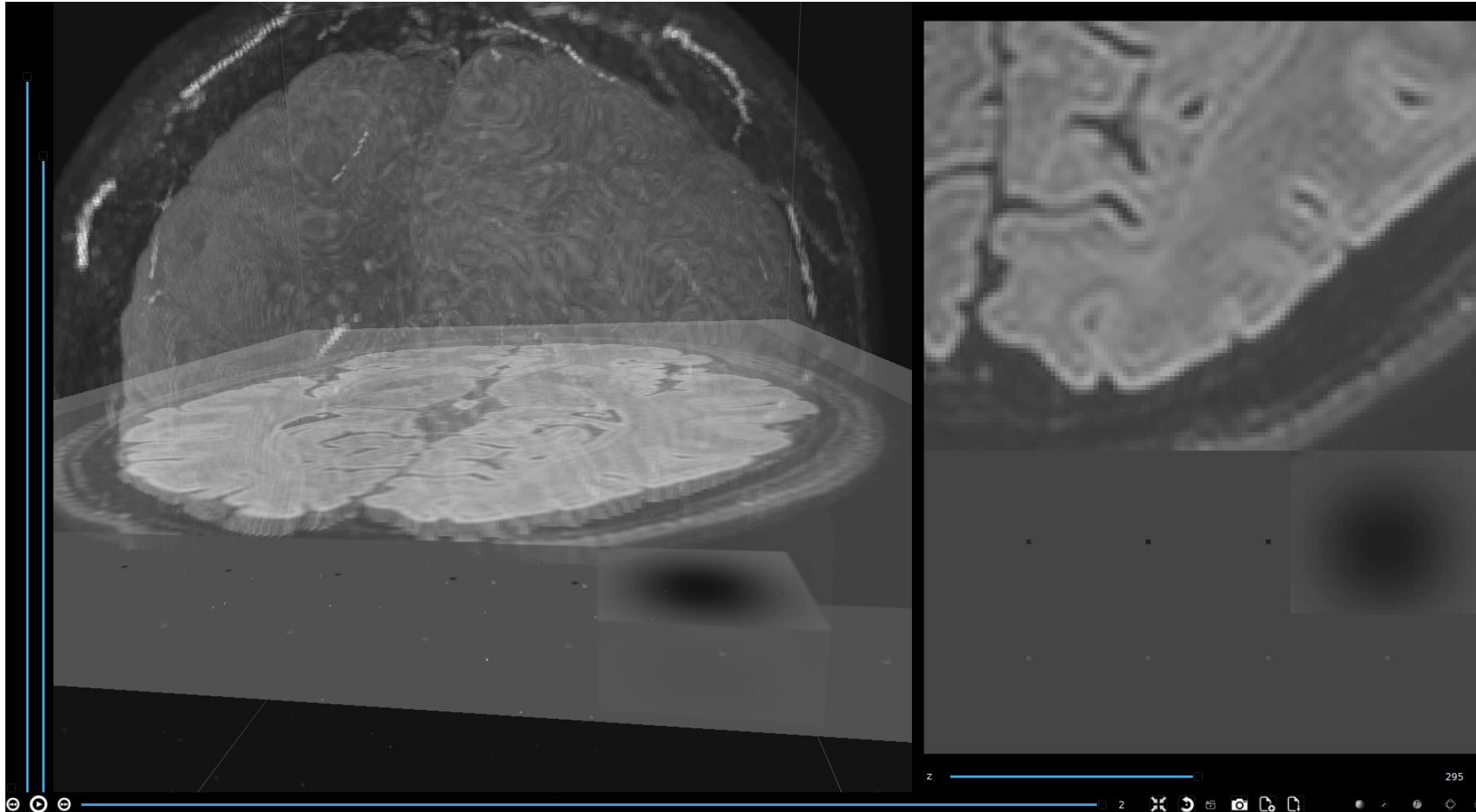
3. 3D konvolusjon av støy med gausskule



3. 3D konvolusjon av hjerne med gausskule



3. 3D konvolusjon av hjerne med gausskule 2



Python, numpy og PyTorch

- Python generelt for de som f.eks. har brukt C++:
<https://speakerdeck.com/pyconslides/transforming-code-into-beautiful-idiomatic-python-by-raymond-hettinger-1>
Video: <https://www.youtube.com/watch?v=OSGv2VnC0go>
- Rask numpy (unngå for-løkker):
<https://speakerdeck.com/jakevdp/losing-your-loops-fast-numerical-computing-with-numpy-pycon-2015>
Video: <https://www.youtube.com/watch?v=EEUXKG97YRw>
- PyTorch innføring i dype nevralt nett med Jupyter notatblokker:
https://github.com/sotte/pytorch_tutorial
Video: <https://www.youtube.com/watch?v=H3aw6wkCv0>

Etc.

- <https://hub.docker.com/r/nvidia/cuda>
- Rust (<https://doc.rust-lang.org/book>), WebAssembly (<https://rustwasm.github.io/docs/book>) og ML <http://www.arewelearningyet.com>
- Rust: Nesten like raskt som C, men finere og har innebygd pakkebehandler
- <https://docs.nvidia.com/deeplearning/dgx/pytorch-release-notes/running.html>
- <https://docs.nvidia.com/deeplearning/dgx/support-matrix/index.html>
- <https://www.drivereasy.com/knowledge/solvednvidia-display-driver-stopped-working-and-has-recovered/>
- <https://support.microsoft.com/en-us/help/2665946/display-driver-stopped-responding-and-has-recovered-error-in-windows-7>

Docker

- <https://linuxconfig.org/how-to-move-docker-s-default-var-lib-docker-to-another-directory-on-ubuntu-debian-linux>
- `systemctl stop docker`
- `systemctl status docker`