Dev Notes

8/18/2022

[**Python dev environment setup:**](#_1cncxg77w4gj) **2**

[**Installing packages:**](#_kyv5da23iv79) **2**

[Jupyter-Lab setup](#_xser62m1ebn9) 2

[CV2 setup](#_tcp12ny652o6) 2

[**Working directory**](#_ifhkcegq0o98) **3**

[**Deep learning book**](#_3dbs36dwbj74) **3**

[**Pytorch**](#_2bbhzzibumad) **3**

[**Neural network learning**](#_tggunaksdnb1) **3**

[**Git**](#_pty1vlw96o1r) **4**

[Create new git](#_5gsa88ug5ubh) 4

[Add selected files into git](#_nswt5ksp7e7r) 4

[See current change & undo change](#_s4x9m5shzm9t) 4

[Undo commit](#_euqfq8e3hmh2) 4

[Commit change](#_i619x95oxpdb) 4

[Git status & history](#_80b9pab674l8) 4

[Stage changes](#_gmio3tmh18t0) 4

[**Project 1: Dog & Cat Deep Neural Network Classifier**](#_wotggj2oz42s) **5**

[Load model:](#_s330q8piwplx) 5

[original:](#_rvwyy9u66n5o) 5

[Different models:](#_f7kz72nsnmag) 5

[Model 1](#_dxpvu7r3ynat) 5

[Original model (changes based off this model)](#_atw23s2e17l2) 5

[Model 2](#_6cnr8tsrgueo) 5

[changed res. to 300x300](#_1kwyvmeexi3w) 5

[Model 3](#_d784367lgksj) 6

[increased epoch by 10 and res. 300x300](#_whjpbqioxx4d) 6

[resolution 300 by 300 pixels](#_7devx4aro3ll) 6

[Model 4](#_igx5qf5a8321) 6

[changed res to 400x400](#_z24dzqbrzzbe) 6

[Model 5](#_re9hbeheope0) 6

[nothing changed](#_ve14eh6x81me) 6

[resolution 300 by 400 pixels](#_nlywmo943ejn) 6

[Model 6](#_fhpr82k4lhi9) 7

[added 1 more layers to NN](#_upibdfet67p6) 7

[resolution 300 by 400 pixels](#_xrh9gp4mbqz7) 7

[Model 7](#_b1an0i1a1p5l) 7

[added 1 more layers to NN and increased epoch by 10 and added evaluation](#_a4w3qakfgr2o) 7

[resolution 300 by 400 pixels](#_igruu7gxqt7h) 7

[Model 8](#_mqxjxwvqtkd5) 7

[added 3 more layers to NN (10, 10, 10) and increased epoch by 10](#_b419ufdn84kb) 7

[resolution 300 by 400 pixels](#_v8rw7r43vfud) 7

[Model 9](#_fzbzw9dtsnaw) 8

[Added 1 layer to NN (32)](#_2mu27ewq75wg) 8

[resolution 300 by 400 pixels](#_km2v3vuhply) 8

# Python dev environment setup:

<https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf>

Create: conda create --name Pytorch python=3.8

Activate: conda activate Pytorch

Delete: conda env remove --name Pytorch

# Installing packages:

## Jupyter-Lab setup

<https://jupyter.org/>

pip install jupyterlab

Jupyter-lab

## CV2 setup

Installing cv2: pip install opencv-python

# Working directory

cd [project name]

!pwd

!ls

Working Folder: /Users/michael/Project

# Deep learning book

git clone:<https://github.com/deep-learning-with-pytorch/dlwpt-code.git>

Deep learning book examples: cd /Users/michael/DeepLearning

# Pytorch

Official site: <https://pytorch.org/>

Transforming images: <https://pytorch.org/vision/stable/transforms.html>

Dataloader: <https://github.com/pytorch/pytorch/blob/master/torch/utils/data/dataloader.py>

# Neural network learning

visualization: <https://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=0&networkShape=4,2&seed=0.74472&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false>

Conv definition and example: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Complete tutorial: <https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html>

Converting Jpg files to jpeg file: <https://aihints.com/how-to-convert-jpg-to-jpeg-in-python-using-opencv/>

Saving & Loading a Model: <https://pytorch.org/tutorials/beginner/saving_loading_models.html>

# Git

## Create new git

Go into folder: cd [name]

git init (only needed to be run once)

## Add selected files into git

git add [file name.type] or group [\*.type]

## See current change & undo change

git diff

git checkout [file name.type] (one file change)

git checkout . (all current changes)

## Undo commit

git reflog (find change ID)

git revert [change ID]

Escape -> : -> wq (write & quit) or q! (quit and don’t revert)

## Commit change

git commit -m "description"

## Git status & history

git status

git log

## Stage changes

git add [file name] (if used again, updates the file as a ”proxy” or prototype commit file)

# Project 1: Dog & Cat Deep Neural Network Classifier

## Load model:

**Saved model must be the same as the model being used**

### original:

model = TheModelClass()

model.load\_state\_dict(torch.load(MichaelsImageClassifierModel.pt))

model.eval()

### Different models:

model = TheModelClass()

model.load\_state\_dict(torch.load(MichaelsImageClassifierModel\_V#.pt))

model.eval()

## Model 1

### Original model (changes based off this model)

* resolution 300 by 400 pixels
* 12 layer neural network
* 10 epochs
* 25,000 training data
* 198 testing data
* 98% right training data accuracy out of 25,000 images
* 74.7% right testing data accuracy out of 198 images
* V1 NN

## Model 2

### changed res. to 300x300

* resolution 300 by 300 pixels
* 12 layer neural network
* 10 epochs
* 25,000 training data
* 198 testing data
* 95.9% right training data accuracy out of 25,000 images
* 60.1% right testing data accuracy out of 198 images
* V1 NN

## Model 3

### increased epoch by 10 and res. 300x300

## resolution 300 by 300 pixels

* 12 layer neural network
* 20 epochs
* 25,000 training data
* 198 testing data
* 99.9% right training data accuracy out of 25,000 images
* 68.6% right testing data accuracy out of 198 images
* V1 NN

## Model 4 (not working)

### changed res to 400x400

* resolution 400 by 400 pixels
* 12 layer neural network
* 10 epochs
* 25,000 training data
* 198 testing data
* 98.9% right training data accuracy out of 25,000 images
* 66.1% right testing data accuracy out of 198 images
* V1 NN

## Model 5

### nothing changed

## resolution 300 by 400 pixels

* 12 layer neural network
* 10 epochs
* 25,000 training data
* 198 testing data
* 99.9% right training data accuracy out of 25,000 images
* 66.6% right testing data accuracy out of 198 images
* V1 NN

## Model 6

### added 1 more layers to NN

## resolution 300 by 400 pixels

* 15 layer neural network
* 10 epochs
* 25,000 training data
* 198 testing data
* 77.1% right training data accuracy out of 25,000 images
* 70.7% right testing data accuracy out of 198 images
* V2 NN

## Model 7

### added 1 more layers to NN and increased epoch by 10 and added evaluation

## resolution 300 by 400 pixels

* 15 layer neural network
* 20 epochs
* 25,000 training data
* 198 testing data
* 70/30 training evaluation data split
* 69.1% right training data accuracy out of 25,000 images
* 69.1% right testing data accuracy out of 198 images
* Ver 2 NN model

## Model 8

### added 3 more layers to NN (10, 10, 10) and increased epoch by 10

### resolution 300 by 400 pixels

* 21 layer neural network
* 20 epochs
* 25,000 training data
* 198 testing data
* 70/30 training evaluation data split
* 50.04% right training data accuracy out of 25,000 images
* 51.01% right testing data accuracy out of 198 images
* V3 NN

## Model 9

### Added 1 layer to NN (32)

### resolution 300 by 400 pixels

* 15 layer neural network
* 10 epochs
* 25,000 training data
* 198 testing data
* 70/30 training evaluation data split
* 83.6% right training data accuracy out of 25,000 images
* 71.7% right testing data accuracy out of 198 images
* V4 NN

## Model 10

### Changed training data and evaluation data split to 90/10

### resolution 300 by 400 pixels

* 15 layer neural network
* 10 epochs
* 25,000 training data
* 198 testing data
* 90/10 training evaluation data split
* % right training data accuracy out of 25,000 images
* % right testing data accuracy out of 198 images
* V4 NN