- Willkommen zum Etherpad!

Mit Hilfe von Etherpad können Sie Texte zeitgleich in Gruppen bearbeiten; dabei erscheinen sämtliche Eingaben sofort bei allen Gruppenmitgliedern.
Auf diese Art und Weise lässt sich der umständliche manuelle Abgleich von Dokumenten vermeiden.

Vorhandene Pads können in diverse gängige Formate exportiert werden und somit nahtlos in OpenOffice, LibreOffice oder Microsoft Office weiter verarbeitet werden.

Wichtiger Hinweis: Bitte beachten Sie, dass Ihre Pads von jeder Person, die die URL zu Ihrem Pad kennt, bearbeitet und gelöscht werden können. Etherpads eignen sich somit nicht als langfristige Speichermethode. Inaktive Pads werden automatisch nach 180 Tagen unwiederbringlich gelöscht.
Weitere Informationen finden Sie auf der Startseite dieses Dienstes unter http://etherpad.gwdg.de

---

Welcome to Etherpad!

This pad text is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents!

Please notice that inactive pads will be deleted.

Hello world! Never used Etherpad before... you?

Wanna import something and give a shorter name to it?
> import numpy as np
then
> np.zeros(5)
instead of
> numpy.zeros(5)

reset_selective variable_name -> delete a specific variable
del(variable_name) ^^

Nice cheatsheet for numpy
https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Numpy_Python_Cheat_Sheet.pdf

Can I use function from anohter package on numpy dataset?
if the function supports numpy arrays. many functions require 'iterables', numpy arrays are iterables, but this use of them may be very slow.

matplotlib doc:
http://matplotlib.org/contents.html

scatterplot-test:
image = matplotlib.pyplot.scatter(x = data[0,:], y = data[1,:])
matplotlib.pyplot.show()

Red sticky note:
- image dataset for examples
- speed? I meant connection speed... the speed of the course is fine for me actually ^^
- sometimes trouble to catch the syntax of the language
- please don't explain something while I am still writing

Green sticky notes:
- good intro to start from scratch, +5
- notebooks on the cluster makes sure that everyone is on the same page
- preliminary data exploration
- first plotting, +1
- etherpad
- style of presenting, +5
- good speed, +2
- sticky notes
- matplotlib
- numpy
- learned a few tricks
- python is quite appealing
- interactive exercises
- questions possible
- cloud computing rocks

==========
Hashtag for twitter: #DLBC17
Standing for DeepLearningBootCamp 2017
==========

Lost in all the python language features? Here you can find it all as a single page summary: https://learnxinyminutes.com/docs/python3/

To coiterate over indices and values you could also use enumerate (see https://stackoverflow.com/questions/522563/accessing-the-index-in-python-for-loops)

https://stackoverflow.com/questions/931092/reverse-a-string-in-python good explanation

How to properly create a multi-dimensional list: https://stackoverflow.com/questions/6667201/how-to-define-two-dimensional-array-in-python
Why so complicated? Because the multiplication of a given list causes trouble... try this:

```
lst = [[1,2,3,4,5]]*5
print(lst)
lst[0][0] = 1000
print(lst)
```

Result is: [[1000, 2, 3, 4, 5], [1000, 2, 3, 4, 5], [1000, 2, 3, 4, 5], [1000, 2, 3, 4, 5], [1000, 2, 3, 4, 5]]
Sucks, eh?

So what if you do this instead:

```
lst = [[1,2,3,4,5] for muh in range(5)]
print(lst)
lst[0][0] = 1000
```

```
   print(lst)
```
Better, eh?

Now, since we want to start loving 'enumerate', look at this:
```
   lst = [ [row*15+col for col in range(15)] for row in range(2) ]
   for rownum, list in enumerate(lst):
      for colnum, elem in enumerate(list):
         lst[rownum][colnum] = elem%(1+((1+rownum*colnum)%7))   %ignore what it does,
```
since it is just some random crap
```
   print(lst)
```
Ahhhh.... sweeet!!!

also for pandas there is a nice cheat sheet
https://s3.amazonaws.com/assets.datacamp.com/blog_assets/PandasPythonForDataScience.pdf

data.loc['Italy':'Poland', 'gdpPercap_1962':'gdpPercap_1967'].idxmax() to get the index of the maximum in a data frame

Feedback of the Python introduction
---
red sticky notes:
- a bit fast (typing while listening while understanding was hard to do)
- pandas not so clear
- food for breaks
- cheat sheet
- pretty dense material for the time given
- what about classes/constructors/methods

green sticky notes

- - everything before pandas was good
- - comparison to R dataframes was good
- - cool ways to work with .csv
- - nice to have everything together in one notebook
- - learned about indexing arrays
- - good speed, good to understand
- - quite intuitive after 5hours
- - good overview
- - learnt a lot, +2
- - second best course that I have attended after 11 years

There are also different types of regularization methods depending on the network type you're using like "Dropout" (from G. Hintons lab on Neural Networks) - just google it :-)

___

While doing the minibatch thing you just take a (very bad) estimate of the gradient but it was shown (can link you the paper if needed) that on the long run you still converge to the

optima...but you speed up the calculation by a lot by not summing your error over your (supposed to have) very large Dataset
Have a look at that Link:
https://image.slidesharecdn.com/pkfetdxyqpiy2i94fvzv-signature-25ed596523f4a669f9a710899d8eaf303c4bbdce8b8e80e99e004c3f91b28bcc-poli-151222094839/95/recommender-systems-50-638.jpg?cb=1450777740

___

_

Hungry for more back-propagation??? -> https://www.youtube.com/watch?v=GlcnxUlrtek

Feedback of the first half of day2

- red sticky notes:
- - overlap between principles and implementation (confusing!)
- - improve backprop part by showing the derivatives in the graph
- - optimisation session not so clear
- - accurate math explanations
- - not so nice: "this is like ..."
- - use a mouse
- - missing: short context would be nice
- - short explanations of variables and colors on each slide
- - too simple, more hands-on would be super

green sticky notes:

- - very nice progression
- - very nice summary
- - solid math background
- - short overview on algorithmic approach
- - first 75% perfect
- - computational graphs nice
- - good intro to the basic stuff, +1
- - effective presentation
- - NN + nearest neighbors
- - liked examples with numbers (instead of formulas)
- - nice slides
- - helpful answers
- - explaining is pretty tough
- - it ended
-

Feedback of the second half of day2:

- red sticky notes:
  - - stuff not working
  - - please explain the keras code again step-by-step
  - - many info channels (slack, email, etherpad, wiki ...)
  - - missing figures in jupyter
  - - difficult to follow during lecture & run notebook at the same time
  - - GPU ran out of resource

    - - last part of presentation today was not efficient, just reviewing code is not good
    - - describing DIGITS can be done in 10 minutes at most
    - - DIGITS: rather not so useful
    - - prefer to learn about Keras
    - - no server for whole lecture
    - - please split group in advanced and beginner pythonistas
    - - for beginners just do very small project step-by-step
    -

- green sticky notes:
    - - LSTM, Recurrent NN in Keras
    - - Good hans-on exercise
    - - interesting to get to know DIGITS
    - - good to do something practical after long theory lecture
    - - morning presentation was interesting and effective
    - - liked programming part
    - - good trade-off between fundamentals, theory and hands-on
-

- I will remove these mail addresses now, to store this etherpad and put it on the timetable!!
-

-

- Deep learning course from Stanford with similar examples as those seen in our current DL course:
- https://cs231n.github.io/
-

- Feedback on the morning session of day3:
-

    - red sticky notes:
    - - not enough structured
    - - too close to the CNN course by stanford (I could have stayed at home to understand the course)
    - - slides are not uploaded yet
    - - CIFAR10 notebook should have been available, +1
    - - more time for lunch
    - - Florian Jug was missing <-- sorry... needed to give this talk... came back ASAP :) don't worry, I can habdle it!
    - - more food
    - - no practical element at all
    - - batch normalisation not clear
    -

    - green sticky notes:
    - - much clearer than yesterday
    - - very nice idea doing the tandem during the presentation, +1
    - - very nice lecture, +3

- - good to use simple program for live demo
- - less mixing of hands-on & lecture made it easier to follow
- - much, much better than yesterday
- - nice to see how to move from small blocks to larger ones
- - more interactivity, +2
- - notebooks on the cluster worked at once
- - current state of the art advices
- - very understandable
- - :)
- - better explanations
- - sessions were better organized
- - Keras commands
-
- Do different neural networks learn the same representations? -> http://proceedings.mlr.press/v44/li15convergent.pdf
-
-
- http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/
-
- https://en.wikipedia.org/wiki/List_of_datasets_for_machine_learning_research
-
- Init bits for jupyter to use just one gpu
- #NOTE: the following is MPI CBG specific, comment it out if you
- # run this notebook somewhere else
- # (otherwise you'll receive an error)
- import limit_resources
- if "limit_resources" in dir():
- limit_resources.of_tensorflow()