jupyter notebook --NotebookApp.iopub\_data\_rate\_limit=10000000000

**Chapter 5**

find / -name "vgg16\_weights\_tf\_dim\_ordering\_tf\_kernels.h5" >> output.txt

(base) popkee:~/environment $ more output.txt  
/home/ubuntu/.keras/models/vgg16\_weights\_tf\_dim\_ordering\_tf\_kernels.h5

(base) popkee:~/environment/datasets/download $ curl -O https://s3.amazonaws.com/book.keras.io/img/ch5/creative\_commons\_elephant.jpg

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 185k 100 185k 0 0 107k 0 0:00:01 0:00:01 --:--:-- 107k

(base) popkee:~/environment/datasets/download $ ls

creative\_commons\_elephant.jpg

Class Activation Map:

(Paper) Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

<https://arxiv.org/abs/1610.02391>

Class Activation Map Video:

<https://www.youtube.com/watch?v=vTY58-51XZA>

Max Pooling in Convolutional Neural Networks explained:

<https://www.youtube.com/watch?v=ZjM_XQa5s6s>

**Chapter 6**

**Download the IMDB data as raw text**

First, head to <http://ai.stanford.edu/~amaas/data/sentiment/> and download the raw IMDB dataset (if the URL isn't working anymore, just Google "IMDB dataset"). Uncompress it.

curl -O <http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz>

tar -xvf aclImdb\_v1.tar.gz

$ pwd

/home/ubuntu/environment/datasets/aclImdb

### Download the GloVe word embeddings

### Head to <https://nlp.stanford.edu/projects/glove/> (where you can learn more about the GloVe algorithm), and download the pre-computed embeddings from 2014 English Wikipedia. It's a 822MB zip file named glove.6B.zip, containing 100-dimensional embedding vectors for 400,000 words (or non-word tokens). Un-zip it.

GloVe: Global Vectors for Word Representation: (**Interesting! Worth to dive deep**)

<https://nlp.stanford.edu/projects/glove/>

curl -O <http://downloads.cs.stanford.edu/nlp/data/glove.6B.zip>

unzip glove.6B.zip

$ pwd

/home/ubuntu/environment/datasets/glovedata

**Download weather timeseries dataset**

Until now, the only sequence data we have covered has been text data, for instance the IMDB dataset and the Reuters dataset. But sequence data is found in many more problems than just language processing. In all of our examples in this section, we will be playing with a weather timeseries dataset recorded at the Weather Station at the Max-Planck-Institute for Biogeochemistry in Jena, Germany: <http://www.bgc-jena.mpg.de/wetter/>.

In this dataset, fourteen different quantities (such air temperature, atmospheric pressure, humidity, wind direction, etc.) are recorded every ten minutes, over several years. The original data goes back to 2003, but we limit ourselves to data from 2009-2016. This dataset is perfect for learning to work with numerical timeseries. We will use it to build a model that takes as input some data from the recent past (a few days worth of data points) and predicts the air temperature 24 hours in the future.

<https://www.bgc-jena.mpg.de/wetter/>

cd /home/ubuntu/environment/datasets

mkdir jena\_climate

cd jena\_climate

wget <https://s3.amazonaws.com/keras-datasets/jena_climate_2009_2016.csv.zip>

unzip jena\_climate\_2009\_2016.csv.zip

**Chapter 7**

Christian Szegedy et al., “Going Deeper with Convolutions,” Conference on Computer Vision and Pattern Recognition (2014), <https://arxiv.org/abs/1409.4842>

Kaiming He et al., “Deep Residual Learning for Image Recognition,” Conference on Computer Vision and Pattern Recognition (2015), <https://arxiv.org/abs/1512.03385>

**Chapter 8**

Question: How do you know the download directory of your Jupyter notebook?

find / -name nietzsche.txt >> Chapter8−1−output.txt

cat Chapter8-1-output.txt

/home/ubuntu/.keras/datasets/nietzsche.txt

scipy.misc.imsave is depressed in scipy, use imageio.imwrite instead.

Downloading data from <https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5>

80142336/80134624 [==============================] - 364s 5us/step

Model loaded.

**To check where to save the h5 file in my local computer**:

$ find / -name vgg19\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5 >> result.txt

$ cat result.txt

/Users/eunice/.keras/models/vgg19\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5

**8.4-generating-images-with-vaes**

**Modification:**

# z = layers.Lambda(sampling)([z\_mean, z\_log\_var])

# <https://keras.io/examples/variational_autoencoder/>

z = layers.Lambda(sampling, output\_shape=(latent\_dim,))([z\_mean, z\_log\_var])

**8.5-introduction-to-gans**

/home/ubuntu/anaconda3/lib/python3.7/site-packages/keras/engine/training.py:297:

UserWarning: Discrepancy between trainable weights and collected trainable weights, did you set `model.trainable` without calling `model.compile` after ? 'Discrepancy between trainable weights and collected trainable'