Published in towards data ...

Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Big Data

The Most Complete List of Best AI Cheat Sheets

Stefan Kojouharov · Follow
Technology · Machine Learning · Artificial Intelligence · Magazine

6 min read · Jun 8, 2017

Top Highlight

Over the past few months, I have been collecting AI cheat sheets. From time to time I share them with friends and colleagues and recently I have been getting asked a lot, so I decided to organize and share the entire collection. To make things more interesting and give context, I added descriptions and/or excerpts for each major topic.

This is the most complete list and the Big 4 is at the very end, enjoy...

>>> Update: We have recently redesigned these cheat sheets into a Super High Definition PDF. Check them out below:

Get AI Cheat Sheets PDF

Neural Networks

Neural Networks Cheat Sheet

>>> If you like this list, you can let me know here.<<<

Neural Networks Graphs

Neural Networks Graphs Cheat Sheet

Get AI Cheat Sheets PDF

Neural Networks Cheat Sheet

Machine Learning Cheat Sheet

Deep Learning Cheat Sheet

Big Data Cheat Sheet

Ultimate Guide to Learning AI & Machine Learning for your Career
Lizzie Kumar
Lizzie Kumar
towardsdatascience.com

Machine Learning Overview

Machine Learning Cheat Sheet

Machine Learning Explained - At Jobs Desert
Machine Jobs in AI, ML, Big Data & More
jobsdesert.com

Machine Learning: Scikit-learn algorithm

This machine learning cheat sheet will help you find the right estimator for the job which is the most difficult part. The flowchart is clear and easy to follow, the documentation and small grids of each estimator do not help you to learn more about the problems and how to solve it.

>>> See Latest Jobs by AI, ML & BIG DATA >>>

Scikit-Learn

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Get AI Cheat Sheets PDF

1 only write 2-3 articles per month.
Subscribe and get the latest
First Name
Email
Sign up

MACHINE LEARNING : ALGORITHM CHEAT SHEET

This machine learning cheat sheet from Microsoft Azure will help you choose the appropriate machine learning algorithms for your predictive analytics solution. From the large array of options, it offers the data owner and then suggests the best algorithm for the job.

MACHINE LEARNING ALGORITHMS CHEAT SHEET

>>> If you like this list, you can let me know here.<<<

Downloadable Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Data Science
Downloadable PDF of Cheat Sheets in Super High Definition
towardsdatascience.com

Python for Data Science

Python Data Science Cheat Sheet

Deep Learning Cheat Sheet

Deep Learning Cheat Sheet

Best Latest Jobs Postings for AI & ML Engineers
jobsdesert.com

TensorFlow

In May 2015 Google announced the second generation of the TPU, as well as offering access to the TPUs in Google Compute Engine. The second-generation TPUs deliver up to 180 teraflops of performance, and when organized into clusters of 64 TPUs provide up to 11.5 petaflops.

ALTOROS
About
TensorFlow
TensorFlow is an open source software library for numerical computation using data flow graphs. TensorFlow was originally developed by the Google Brain team within Google's AI organization. It comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains as well.
Installation
Install TensorFlow
pip install tensorflow
pip install --upgrade
https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-0.11.0rc0-cp27-none-darwin.whl
sudo pip install --upgrade
https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-0.11.0rc0-cp27-none-darwin.whl
How to install Jupyter
jupyter notebook
How to install Keras
sudo pip install keras
sudo pip install --upgrade git+git://github.com/fchollet/keras.git
Helpers
Python helper
Important functions
help(func)
dir(obj)
type(obj)
Get help for object list of available attributes, methods, operators and so on
vars(obj)
Get list of object attributes
locals()
Get local variables
globals()
Get global variables
id(obj)
Unique integer for given object
dir(module)
Shows documentations about the object
locals()
globals()
Return the dictionary containing the current local / global variables
Updates
Update and return a dictionary containing the current scope's local variables
Return
Return the identity of an object. This is guaranteed to be a unique and constant integer for the lifetime of the object.
help('__builtin__')
Help on built-in functions
help('__builtin__.object')
Help on built-in functions
TensorFlow
Start an session
sess = tf.Session()
tf.Session()
tf.InteractiveSession()
tf.close()
Tensorflow Constants
tf.zeros(shape, dtype=tf.float32, name=None)
tf.zeros_like(tensor, dtype=None, name=None, optimize=True)
tf.ones(shape, dtype=tf.float32, name=None)
tf.ones_like(tensor, dtype=None, name=None, optimize=True)
tf.fill(dims, value, name=None)
tf.constant(value, dtype=None, shape=None, name='Const', verify_shape=False)
TensorFlow Operations
tf.add(x, y, name=None)
tf.sub(x, y, name=None)
tf.mul(x, y, name=None)
tf.neg(x, name=None)
tf.sign(x, name=None)
tf.inv(x, name=None)
tf.abs(x, name=None)
tf.square(x, name=None)
tf.round(x, name=None)
tf.sqrt(x, name=None)
tf.pow(x, y, name=None)
tf.exp(x, name=None)
tf.log(x, name=None)
tf.ceil(x, name=None)
tf.floor(x, name=None)
tf.maximum(x, y, name=None)
tf.minimum(x, y, name=None)
tf.cos(x, name=None)
tf.sin(x, name=None)
Reduction
tf.mean(x)
Activation functions
tf.nn.relu(features, name=None)
tf.nn.relu6(features, name=None)
tf.nn.softplus(features, name=None)
tf.nn.dropout(x, keep_prob, noise_shape=None, seed=None, name=None)
tf.nn.bias_add(value, bias, data_format=None, name=None)
tf.sigmoid(x, name=None)
tf.tanh(x, name=None)
Loss
tf.nn.l2_loss(t, name=None)
tf.nn.sigmoid_cross_entropy_with_logits(logits, targets, name=None)
tf.nn.softmax(logits, name=None)
tf.nn.softmax_cross_entropy_with_logits(logits, targets, name=None)
tf.nn.sparse_softmax_cross_entropy_with_logits(logits, labels, name=None)
Matrix
tf.matmul(a, b, transpose_a=False, transpose_b=False, a_is_sparse=False, b_is_sparse=False, name=None)
tf.batch_matmul(x, y, adj_x=False, adj_y=False, name=None)
tf.transpose(a, perm=None, name='transpose')
tf.reverse(tensor, dims, name=None)
tf.reverse_sequence(input, seq_lengths, seq_dim, batch_dim=None, name=None)
tf.gather(params, indices, name=None)
tf.dynamic_partition(data, partitions, num_partitions, name=None)
tf.dynamic_stitch(indices, data, name=None)
Sparse Matrix
tf.sparse_concat(concat_dim, indices, values, shapes, name=None)
tf.sparse_reorder(indices, values, shapes, name=None)
tf.sparse_reshape(indices, shapes, shape, name=None)
tf.sparse_split(split_dim, num_split, indices, values, shapes, name=None)
tf.sparse_tile(indices, values, shapes, multiples, name=None)
Checkpoint
tf.train.Saver(var_list=None, max_to_keep=5, keep_checkpoint_every_n_hours=10000.0, name=None, restore_sequentially=False, sharded=False, shard_size=1024, max_shards=None, pad_step_number=False)
Saver.save(sess, save_path, global_step=None, latest_filename=None, meta_graph_suffix='meta', write_meta_graph=True, write_state=True)
Saver.restore(sess, save_path)
tf.train.latest_checkpoint(checkpoint_dir, latest_filename=None)
tf.train.get_checkpoint_state(checkpoint_dir, latest_filename=None)
Tensorboard
Tensorboard
Tensorboard --logdir=path/to/log-directory --port=number
Summary Operations
tf.scalar_summary(tags, values, collections=None, name=None)
tf.image_summary(tag, tensor, max_images=3, collections=None, name=None)
tf.histogram_summary(tag, values, collections=None, name=None)
tf.nn.zero_fraction(value, name=None)
Logging and Summary Operations
tf.Print(input_, data, message=None, first_n=None, summarize=None, name=None)
tf.Assert(condition, [data], summarize=None, name=None)
tf.Assert(condition, [data], summarize=None, name=None)
tf.NaN(tensor, name=None)
tf.Inf(tensor, name=None)
Tensorflow Classes
tf.Graph()
tf.Operation
tf.Tensor
tf.DtypeSize
tf.TensorShape(dims)
tf.TensorShape.concatenate(other)
tf.TensorShape.ndims
tf.TensorShape.dims
tf.TensorShape.merge_with(other)
tf.TensorShape.as_dimension()
tf.Dimension(value)
tf.Dimension.__add__(other)
tf.Dimension.__sub__(other)
tf.Dimension.__mul__(other)
tf.Dimension.__div__(other)
tf.Dimension.__mod__(other)
tf.Dimension.__lt__(other)
tf.Dimension.__gt__(other)
tf.Dimension.__eq__(other)
tf.Dimension.__str__()
tf.Dimension.__int__()
tf.Dimension.__index__()
tf.DeviceSpec(job=None, replica=None, task=None, device_type=None, device_index=None)
tf.DeviceSpec.from_string(spec)
tf.DeviceSpec.parse_from_string(spec)
tf.DeviceSpec.merge_from(other)
tf.DeviceSpec.to_string()
tf.DeviceSpec.make_merged_spec(dev1, dev2, ...)
tf.Variable(initial_value=None, trainable=True, collections=None, validate_shape=True, shape=None, name=None, variable_def=None, dtype=None)
tf.Variable.initialized_value()
tf.Variable.assign(value, use_locking=False)
tf.Variable.assign_add(delta, use_locking=False)
tf.Variable.assign_sub(delta, use_locking=False)
tf.Variable.add_to_collection(collection, [tf.GraphKeys.VARIABLES])
tf.Variable.eval(session=None)
tf.Variable.name
tf.Variable.dtype
tf.Variable.get_shape()
tf.Variable.device
tf.Variable.initializer
tf.Variable.graph
tf.Variable.op
tf.Variable.value()
tf.Variable.read_value()
tf.Variable.assign(value, use_locking=False)
tf.Variable.assign_add(delta, use_locking=False)
tf.Variable.assign_sub(delta, use_locking=False)
tf.Variable.scatter_sub(sparse_delta, use_locking=False)
tf.Variable.scatter_add(sparse_delta, use_locking=False)
tf.Variable.scatter_sub(sparse_delta, use_locking=False)
tf.Variable.scatter_mul(sparse_delta, use_locking=False)
tf.Variable.scatter_div

Downloadable Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Data Science:
DataScienceDojo.com/ai-cheat-sheets-in-a-dropbox-folder/

Keras

In 2015, Google's TensorFlow team decided to support Keras in TensorFlow's core library. Chollet explained that Keras was conceived to be an interface rather than an end to end machine learning framework. It presents a higher-level, more intuitive set of abstractions that make it easy to configure neural networks regardless of the backend scientific computing library.

Keras Cheat Sheet

Numpy

Numpy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. Numpy address the obvious performance bottleneck by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using Numpy.

Numpy Cheat Sheet

Pandas

The name 'Pandas' is derived from the term "panel data", an econometrics term for multidimensional structured data sets.

Get AI Cheat Sheets PDF

Pandas Cheat Sheet

Data Wrangling

The term "data wrangler" is starting to infiltrate pop culture. In the 2017 movie Kong: Skull Island, one of the characters, played by actor Marc Evan Jackson is introduced as "Steve Woodward, our data wrangler."

Data Wrangling Cheat Sheet

Downloadable Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Data Science:
DataScienceDojo.com/ai-cheat-sheets-in-a-dropbox-folder/

Pandas Data Wrangling Cheat Sheet

Ultimate Guide to Leveraging NLP & Machine Learning for your Enterprise
Download Now:
info.lexalytics.com

Data Wrangling with dplyr and tidyr

Data Wrangling with dplyr and tidyr Cheat Sheet

Data Wrangling with dplyr and tidyr Cheat Sheet

Discover Latest Jobs in Data Scientists, Data Engineers & Architects - AI Jobs Board
datasciencejobs.com

Scipy

Scipy builds on the NumPy array object and is part of the NumPy stack which includes tools like Matplotlib, pandas and SymPy, and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as MATLAB, GNU Octave, and Scilab. The NumPy stack is also sometimes referred to as the SciPy stack [3].

Get AI Cheat Sheets PDF

Scipy Cheat Sheet

Matplotlib

matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged [3] SciPy makes use of matplotlib.

pyplot is a matplotlib module which provides a MATLAB-like interface [6] matplotlib is designed to be as usable as MATLAB, with the ability to use Python, with the advantage that it is free.

Matplotlib Cheat Sheet

>>> If you like this list, you can list on Scoop Ipsa. <<<

Data Visualization

Data Visualization Cheat Sheet

ggplot Cheat Sheet

Discover Latest Jobs in Data Scientists, Data Engineers & Architects - AI Jobs Board
datasciencejobs.com

PySpark

Flypaper Cheat Sheet

Downloadable Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Data Science...
Downloadable PDF of Best AI Cheat Sheets in Super High Definition
leanpub.com/ai

Big-O

LEGEND
TIME Complexity
SPACE Complexity
Great
Good
Fair
Poor
Terrible
BIG-O CHEATSHEET
DATA STRUCTURE Operations
Array
Stack
Queue
Singly-Linked List
Doubly-Linked List
Skip List
Hash Table
Binary Search Tree
Cartesian Tree
B-Tree
Red-Black Tree
Splay Tree
AVL Tree
KD Tree
ARRAY SORTING Algorithms
Timsort
Clustersort
Heapsort
Blockmergesort
Libraksort
Partitioned Mergesort
Quicksort
Quicksort (3-way)
Quicksort (median-of-3)
Quicksort (random)
Quicksort (median-of-7)
Quicksort (median-of-9)
Smoothsort
Introsort
Introspective Sorting
Introspective Sorting (Musser 1997)
Introspective Sorting (Musser 1997 Improved)
Introspective Sorting (Musser 1997 Improved 2)
Dual Pivot Quicksort
Pattern-defeating Quicksort
Pattern-defeating Quicksort (Improved)
Pattern-defeating Quicksort (Improved 2)
Pattern-defeating Quicksort (Improved 3)
Pattern-defeating Quicksort (Improved 4)
Pattern-defeating Quicksort (Improved 5)
Pattern-defeating Quicksort (Improved 6)
Pattern-defeating Quicksort (Improved 7)
Pattern-defeating Quicksort (Improved 8)
Pattern-defeating Quicksort (Improved 9)
Pattern-defeating Quicksort (Improved 10)
Pattern-defeating Quicksort (Improved 11)
Pattern-defeating Quicksort (Improved