









# keras-pandas

Brendan Herger, [hergertarian.com](http://hergertarian.com)  
<http://keras-pandas.readthedocs.io/>  
Slides: <https://goo.gl/2zMKxP>

	keras/examples at master · keras-team/keras	Brendan
	GitHub, Inc. [US] <a href="https://github.com/keras-team/keras/tree/master/examples">https://github.com/keras-team/keras/tree/master/examples</a>	     
	<h2>Keras examples directory</h2>	
	<h3>Vision models examples</h3> <hr/> <p><a href="#">mnist_mlp.py</a> Trains a simple deep multi-layer perceptron on the MNIST dataset.</p>	
	<p><a href="#">mnist_cnn.py</a> Trains a simple convnet on the MNIST dataset.</p>	
	<h3>Text &amp; sequences examples</h3> <hr/> <p><a href="#">addition_rnn.py</a> Implementation of sequence to sequence learning for performing addition of two numbers (as strings)</p> <p><a href="#">conv_lstm.py</a> Demonstrates the use of a convolutional LSTM network.</p> <p><a href="#">image_ocr.py</a> Trains a convolutional stack followed by a recurrent stack and a CTC logloss function to perform optical character recognition (OCR)</p>	
	<h3>Generative models examples</h3> <hr/> <p><a href="#">lstm_text_generation.py</a> Generates text from Nietzsche's writings.</p>	
	<p><a href="#">mnist_siamese.py</a> Trains a Siamese multi-layer perceptron on pairs of digits from the MNIST dataset.</p> <p><a href="#">mnist_swwae.py</a> Trains a Stacked What-Where AutoEncoder built on residual blocks on the MNIST dataset.</p> <p><a href="#">mnist_transfer_cnn.py</a> Transfer learning toy example.</p>	

# Intro Hands On Getting Started



# Intro

DL is attainable. `keras-pandas` allows users to rapidly build and iterate on deep learning models.

- **New users:** Lowering the barrier to entry, good starting point.
- **Existing users:** Allows for rapid iteration, good starting point

Hands On

# Old way

- **Highly customizable:** Data transformations, data format, input layers
- **Heuristic driven:** Involves high amount of domain expertise, neural network theory, and heuristics
- **Repetitive:** Time consuming & repetitive to create similarly formatted layers



# keras-pandas way

- **Less customizable:** Batteries included defaults for each data type
- **Rapid:** Ability to build and iterate on models with a few function calls
- **Maintainable:** More consistent code base, with less redundancy

# Getting started

- **Example:** Try the titanic example in README.md
- **Docs:** Near total coverage, dive deeper than this talk
- **Get involved:** Actively looking for collaborators & feedback



## Getting started

We'll install `keras-pandas`

```
pip install -U keras-pandas
```

```
In [1]: from keras import Model
from keras.layers import Dense

from keras_pandas.Automater import Automater
from keras_pandas.lib import load_titanic
```

```
/Users/brendanherger/anaconda2/lib/python2.7/site-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 = np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

## Data

Let's say we want to look at the [titanic data set](#)

```
In [2]: observations = load_titanic()
observations.head(2)
```

Out[2]:


	survived	pclass		name	sex	age	siblings_spouses_abor	parents_children_abor	fare
0	0	3		Mr. Owen Harris Braund	male	22.0	1	0	7.2500

# Next steps

- **Time series:** Smart defaults for time series models
- **Time stamps:** Sine / cosine decomposition, etc
- **Iterate:** Hear and respond to user feedback
- **Examples:** Find interesting data sets w/ mixed data types



# Getting started

 keras-pandas

latest

Search docs

CONTENTS:

keras-pandas

Quick Start

Usage

Contributing


Contact

Automater

lib

constants

transformations

 Read the Docs

v: latest ▾

For more info, check out the:

- [Code](#)
- [Documentation](#)
- [Issue tracker](#)
- [Author's website](#)

## Quick Start

Let's build a model with the [titanic data set](#). This data set is particularly fun because this data set contains a mix of categorical and numerical data types, and features a lot of null values.

We'll `keras-pandas`

```
pip install -U keras-pandas
```

And then run the following snippet to create and train a model:

```
from keras import Model
from keras.layers import Dense

from keras_pandas.Automater import Automater
from keras_pandas.lib import load_titanic

observations = load_titanic()

# Transform the data set, using keras_pandas
categorical_vars = ['pclass', 'sex', 'survived']
numerical_vars = ['age', 'siblings_spouses_aboard', 'parents_children_aboard', 'fare']
text_vars = ['name']

auto = Automater(categorical_vars=categorical_vars, numerical_vars=numerical_vars, text_vars=text_vars,
                 response_var='survived')
X, y = auto.fit_transform(observations)

# Start model with provided input nub
x = auto.input_nub
```



# Thanks!

Brendan Herger, [hergertarian.com](http://hergertarian.com)  
<http://keras-pandas.readthedocs.io/>  
Slides: <https://goo.gl/2zMKxP>

# Appendix

# Lessons learned

- Thank your loved ones
- Find a few good examples you'd like to borrow (steal) from
- Stack
- Documentation: MD (docs) & RST (docstrings)
  - Documentation website: Sphinx (with m2r plugin for markdown)
  - Documentation serving: readthedocs.io
  - CI/CD: Travis for CI/CD



# Pipelines

- **Text:** String -> tokens -> embedding -> bidirectional LSTM -> Flatten
- **Numerical:** Whiten -> Dense
- **Categorical:** OHE -> Entity Embedding -> Flatten
- **Boolean:** OHE -> Entity Embedding -> Flatten