Keras hans-on

mishimasyk#9 @iwatobipen

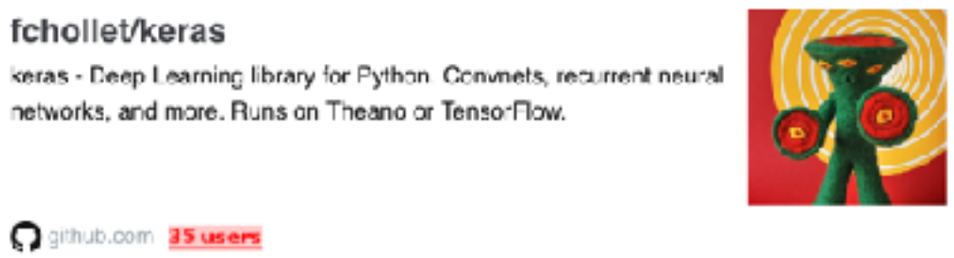
自己紹介＞＞＞カット

@iwatobipen

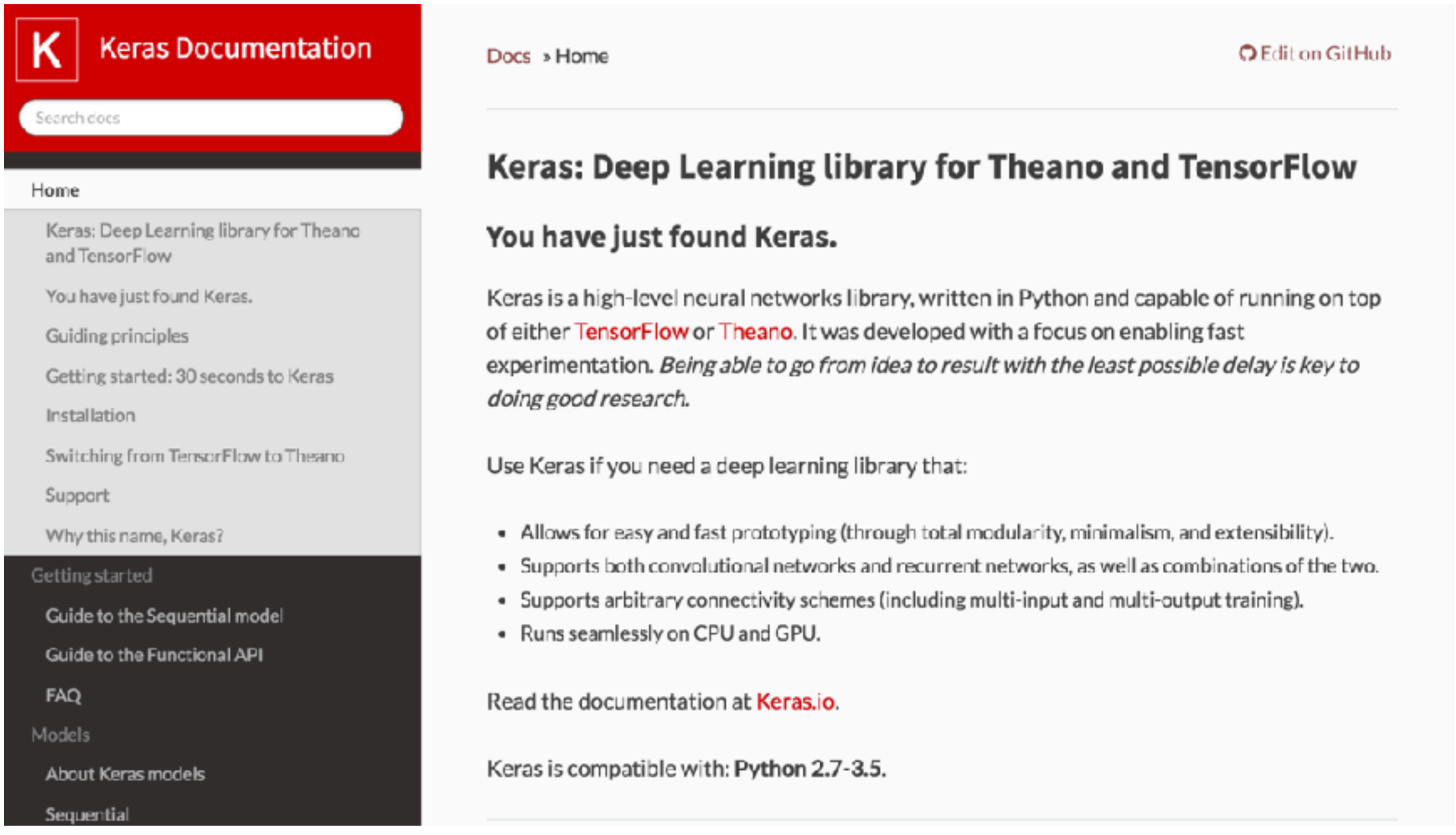
Kerasとは

- TheanoもしくはTensorFlowをエンジンとしたDLのライブラリ
- DLを簡易的に実装したい
- CNN, RNNなどを実装したいときにもオススメ
- CPU/GPU上でのシームレスな実行

https://keras.io/



ドキュメントも充実


Keras Documentation
Search docs
Home
Keras: Deep Learning library for Theano and TensorFlow
You have just found Keras.
Guiding principles
Getting started: 30 seconds to Keras
Installation
Switching from TensorFlow to Theano
Support
Why this name, Keras?
Getting started
Guide to the Sequential model
Guide to the Functional API
FAQ
Models
About Keras models
Sequential
Docs » Home
Edit on GitHub
Keras: Deep Learning library for Theano and TensorFlow
You have just found Keras.
Keras is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.
Use Keras if you need a deep learning library that:
Allows for easy and fast prototyping (through total modularity, minimalism, and extensibility).
Supports both convolutional networks and recurrent networks, as well as combinations of the two.
Supports arbitrary connectivity schemes (including multi-input and multi-output training).
Runs seamlessly on CPU and GPU.
Read the documentation at Keras.io.
Keras is compatible with: Python 2.7-3.5.

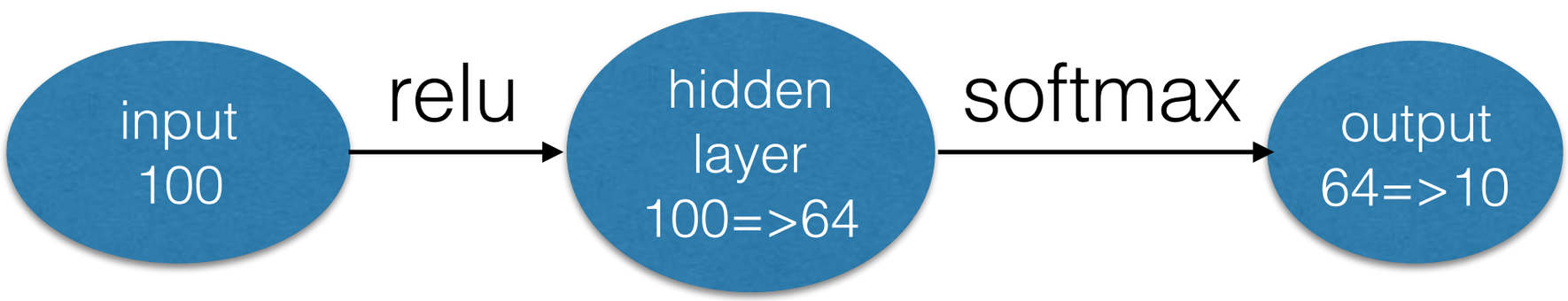
Kerasでの実装

```
from keras.models import Sequential
from seas.layers import Dense, Activation
model = Sequential()
model.add( Dense( output_dim=64, input_dim=100 ) )
model.add( Activation( “relu” ) )
model.add( Dense( output_dim=10 ) )
model.add( Activation( “softmax” ) )

model.compile( loss=“categorical_crossentropy”,
               optimizer=“sgd”,
               metrics=[ “accuracy” ] )
```

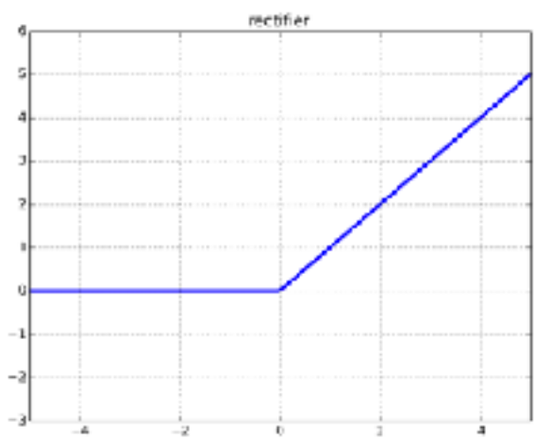
addで重ねるだけ！

compileでモデル作成

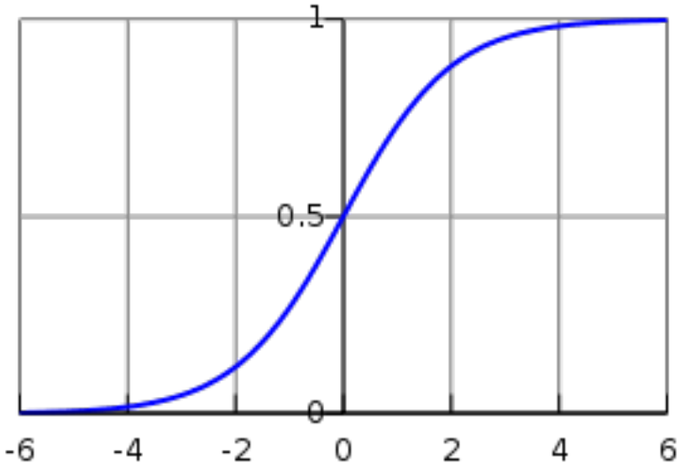


ちょっとだけ補足

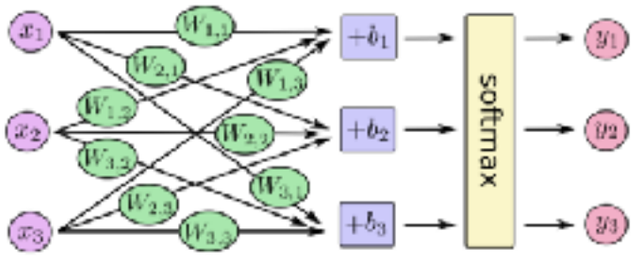
relu



sigmoid



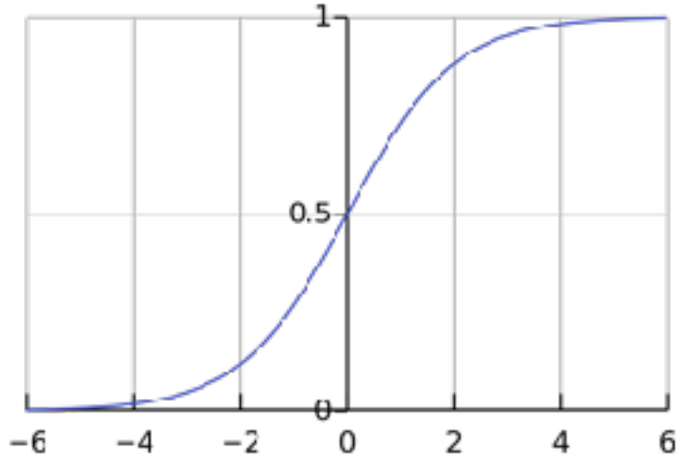
You can picture our softmax regression as looking something like the following, although with a lot more xs. For each output, we compute a weighted sum of the xs, add a bias, and then apply softmax.



If we write that out as equations, we get:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} W_{1,1}x_1 + W_{1,2}x_2 + W_{1,3}x_3 + b_1 \\ W_{2,1}x_1 + W_{2,2}x_2 + W_{2,3}x_3 + b_2 \\ W_{3,1}x_1 + W_{3,2}x_2 + W_{3,3}x_3 + b_3 \end{bmatrix}\right)$$

softmax



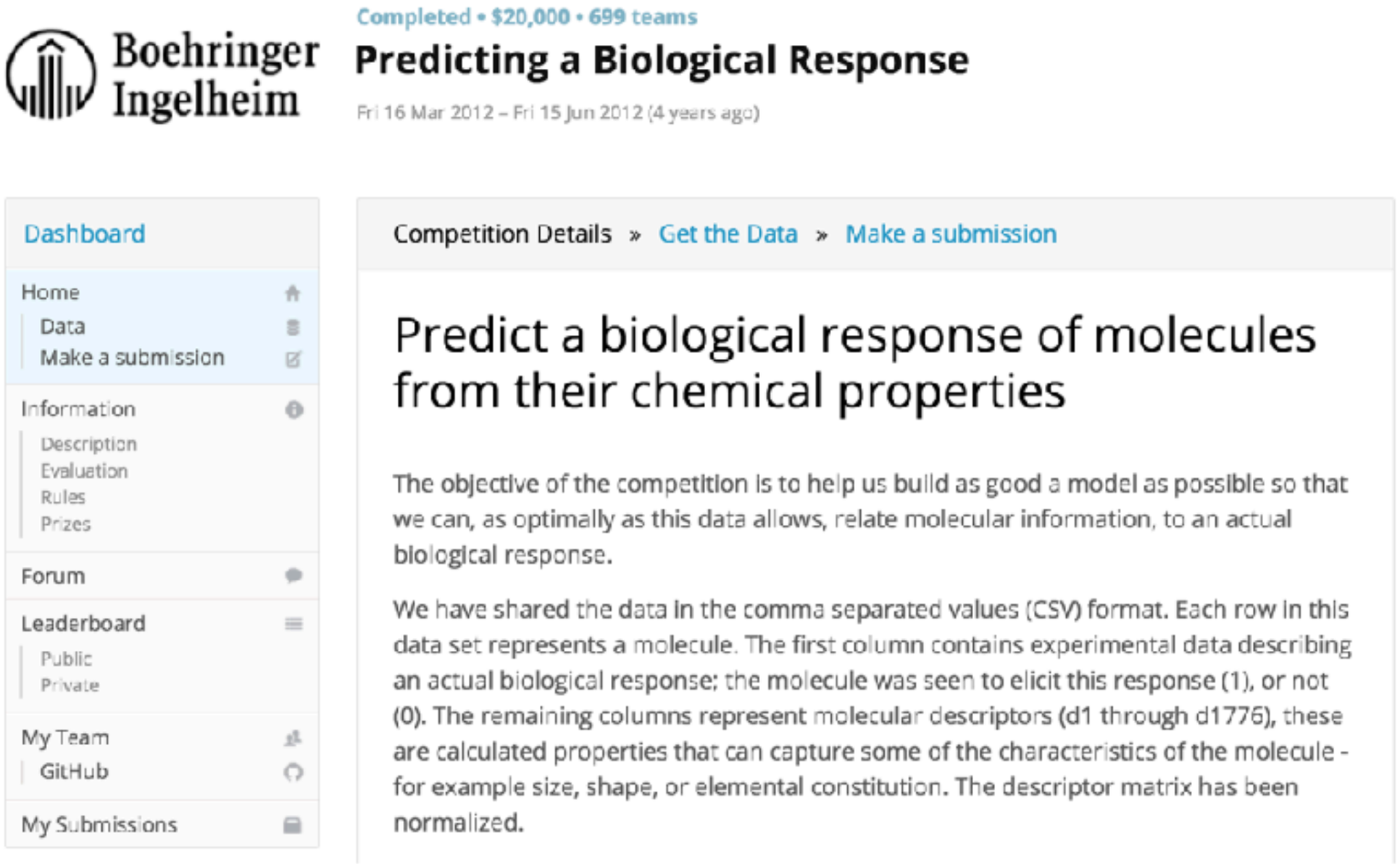
https://www.tensorflow.org/versions/r0.12/tutorials/mnist/beginners/index.html

modelができたら学習

```
model.fit(X_train, Y_train, nb_epoch=5, batch_size=32)

 # 予測もシンプル
classes = model.predict_classes(X_test, batch_size=32)
proba = model.predict_proba(X_test, batch_size=32)
```

今日のネタはとかです。



https://www.kaggle.com/c/bioresponse

ということでVMにGO

$ vagrant up

$ vagrant ssh