Getting started: 30 seconds to Keras  [ https://github.com/fchollet/keras ]

The core data structure of Keras is a model, a way to organize layers. There are two types of models: Sequential and Graph.

Here's the **Sequential model** (a linear pile of layers):

```python
from keras.models import Sequential

model = Sequential()
```

**Stacking layers** is as easy as `.add()`:

```python
from keras.layers.core import Dense, Activation

model.add(Dense(output_dim=64, input_dim=100, init="glorot_uniform"))
model.add(Activation("relu"))
model.add(Dense(output_dim=10, init="glorot_uniform"))
model.add(Activation("softmax"))
```

Once your model looks good, **configure its learning process** with `.compile()`:

```python
model.compile(loss='categorical_crossentropy', optimizer='sgd')
```

If you need to, you can further configure your optimizer. A core principle of Keras is to make things reasonably simple, while allowing the user to be fully in control when they need to (the ultimate control being the easy extensibility of the source code).
from keras.optimizers import SGD
model.compile(loss='categorical_crossentropy', optimizer=SGD(lr=0.01, momentum=0.9, nesterov=True))

You can now **iterate** on your training data in batches:

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

Alternatively, you can feed batches to your model manually:

```python
model.train_on_batch(X_batch, Y_batch)
```

**Evaluate your performance** in one line:

```python
objective_score = model.evaluate(X_test, Y_test, batch_size=32)
```

Or **generate predictions** on new data:

```python
classes = model.predict_classes(X_test, batch_size=32)
proba = model.predict_proba(X_test, batch_size=32)
```

Building a network of LSTMs, a deep CNN, a Neural Turing Machine, a word2vec embedder or any other model is just as fast.