“Hello world”
of deep learning
Keras

If you want to learn theano:

Very flexible
Need some effort to learn

TensorFlow or Theano

Easy to learn and use
(still have some flexibility)
You can modify it if you can write TensorFlow or Theano

Interface of TensorFlow or Theano

Keras
Keras

• François Chollet is the author of Keras.
  • He currently works for Google as a deep learning engineer and researcher.

• Keras means *horn* in Greek

• Documentation: [http://keras.io/](http://keras.io/)

• Example:
  [https://github.com/fchollet/keras/tree/master/examples](https://github.com/fchollet/keras/tree/master/examples)
使用 Keras 心得

Deep Learning研究生

朋友覺得我在
我媽覺得我在
大眾覺得我在
指導教授覺得我在
我以為我在
事實上我在

感謝 沈昇勳 同學提供圖檔
“Hello world”

- Handwriting Digit Recognition

MNIST Data: http://yann.lecun.com/exdb/mnist/
Keras provides data sets loading function: http://keras.io/datasets/
Keras: Building a Network

```
model = Sequential()
model.add(Dense(input_dim=28*28, units=500, activation='relu'))
model.add(Dense(units=500, activation='relu'))
model.add(Dense(units=10, activation='softmax'))
```

Output between 0 and 1

28x28

500

500

Softmax

\( \mathbf{y}_1 \), \( \mathbf{y}_2 \), \ldots \, \ldots \, \mathbf{y}_{10} \)
Configuration

Several alternatives: https://keras.io/objectives/

```python
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam

Pick the best function

```python
model.fit(x_train, y_train, batch_size=100, epochs=20)
```

Number of training examples

- `28 x 28 = 784`
- `10`
Mini-batch

Randomly initialize network parameters

Pick the 1\textsuperscript{st} batch
\[ L' = l^1 + l^{31} + \ldots \]
Update parameters once

Pick the 2\textsuperscript{nd} batch
\[ L'' = l^2 + l^{16} + \ldots \]
Update parameters once

Until all mini-batches have been picked

Repeat the above process
Mini-batch

Batch size influences both **speed** and **performance**. You have to tune it.

```
model.fit(x_train, y_train, batch_size=100, nb_epoch=20)
```

- **Mini-batch**
  - 100 examples in a mini-batch
  - Batch size = 1 ➞ Stochastic gradient descent
  - Repeat 20 times ➞ one epoch

**Mini-batch**

- Pick the 1\textsuperscript{st} batch
  \[ L' = l^1 + l^{31} + \ldots \]
  - Update parameters once

- Pick the 2\textsuperscript{nd} batch
  \[ L'' = l^2 + l^{16} + \ldots \]
  - Update parameters once

- Until all mini-batches have been picked
Speed

- Smaller batch size means more updates in one epoch
  - E.g. 50000 examples
  - batch size = 1, 50000 updates in one epoch
  - batch size = 10, 5000 updates in one epoch

Very large batch size can yield worse performance

Batch size = 1 and 10, update the same amount of times in the same period.

Batch size = 10 is more stable, converge faster

GTX 980 on MNIST with 50000 training examples

166s 1 epoch
17s 10 epoch

166s

17s
Speed - Matrix Operation

\[ y = f(x) \quad \text{Forward pass (Backward pass is similar)} \]

\[ = \sigma(W^L) \cdots \sigma(W^2) \sigma(W^1)x + b^1 \cdots + b^L \]
Speed - Matrix Operation

• Why mini-batch is faster than stochastic gradient descent?

*Stochastic Gradient Descent*

\[ z^1 = W^1 x \]

*Mini-batch*

\[ \begin{bmatrix} z^1 & z^1 \end{bmatrix} = W^1 \begin{bmatrix} x & x \end{bmatrix} \]

Practically, which one is faster?
Keras

Save and load models
https://faroit.github.io/keras-docs/2.0.2/getting-started/faq/#how-can-i-save-a-keras-model

How to use the neural network (testing):

```python
score = model.evaluate(x_test, y_test)
print('Total loss on Testing Set:', score[0])
print('Accuracy of Testing Set:', score[1])
```

case 1:

```python
result = model.predict(x_test)
```
Live Demo