

“Hello world”  
of deep learning

# Keras

If you want to learn theano:

[http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS\\_2015\\_2/Lecture/Theano%20DNN.ecm.mp4/index.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/Theano%20DNN.ecm.mp4/index.html)

[http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS\\_2015\\_2/Lecture/RNN%20training%20\(v6\).ecm.mp4/index.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/RNN%20training%20(v6).ecm.mp4/index.html)

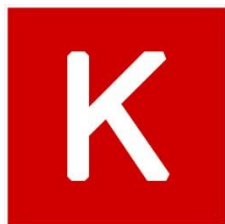


or theano

Very flexible

Need some  
effort to learn

Interface of  
TensorFlow or  
Theano



keras

Easy to learn and use

(still have some flexibility)

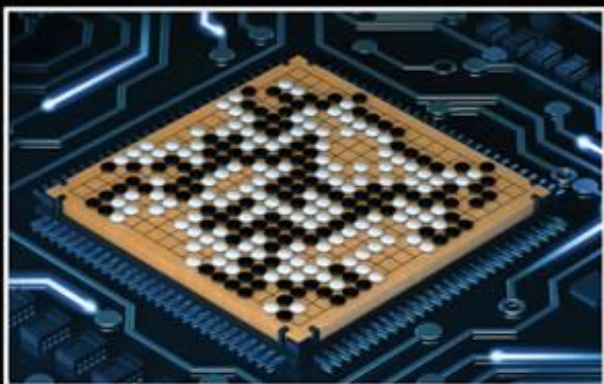
You can modify it if you can write  
TensorFlow or Theano

# 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/>
- Example:  
<https://github.com/fchollet/keras/tree/master/examples>

# 使用 Keras 心得

## Deep Learning研究生



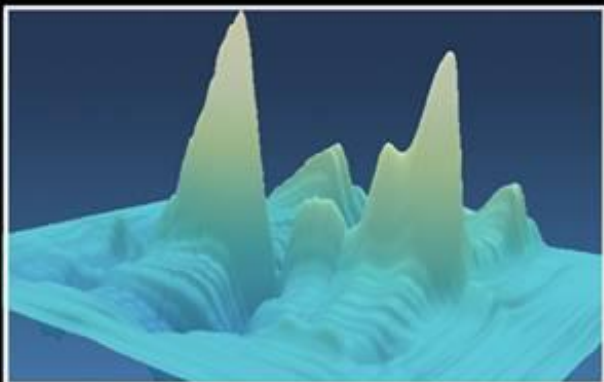
朋友覺得我在



我媽覺得我在



大眾覺得我在



指導教授覺得我在



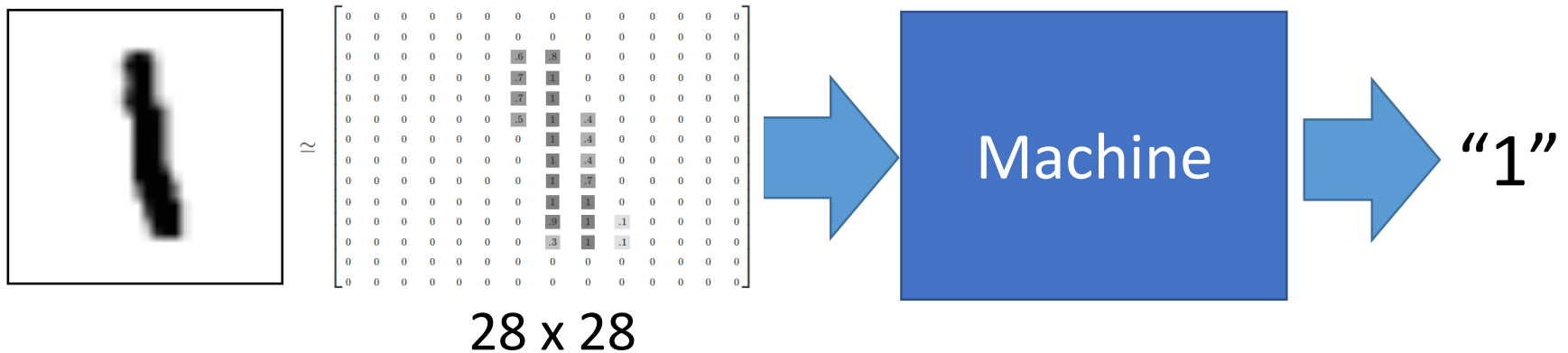
我以為我在



事實上我在

# “Hello world”

- Handwriting Digit Recognition



MNIST Data: <http://yann.lecun.com/exdb/mnist/>

Keras provides data sets loading function: <http://keras.io/datasets/>

# Keras

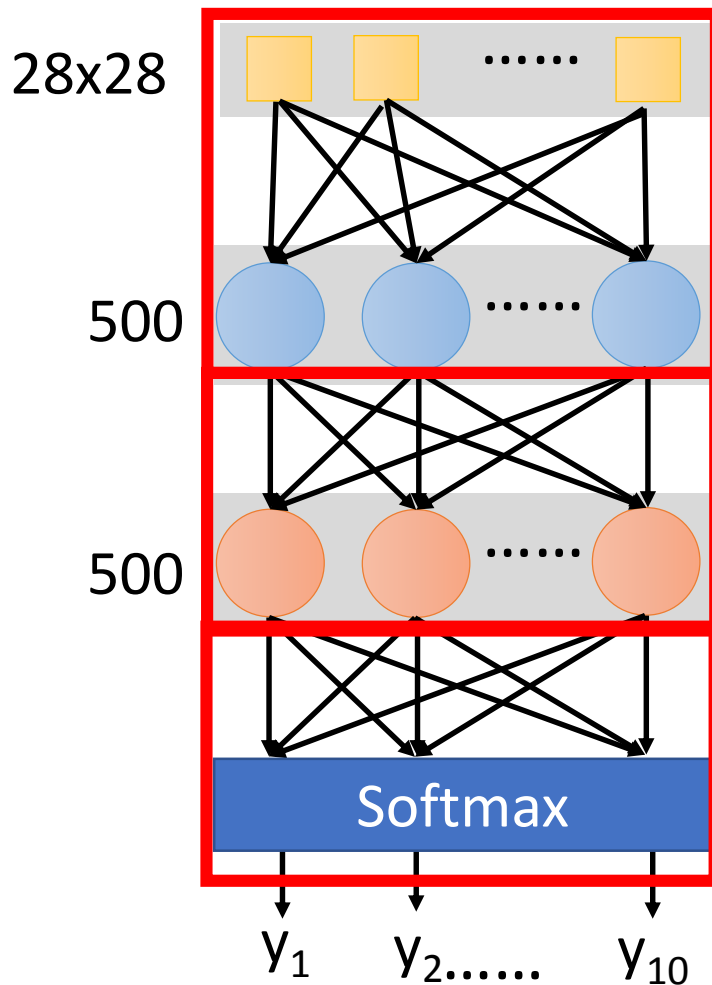
Step 1:  
define a set  
of function



Step 2:  
goodness of  
function



Step 3: pick  
the best  
function



```
model = Sequential()
```

```
model.add( Dense( input dim=28*28,  
                  output dim=500 ) )  
model.add( Activation( 'sigmoid' ) )
```

softplus, softsign, relu, tanh,  
hard\_sigmoid, linear

```
model.add( Dense( output dim=500 ) )  
model.add( Activation( 'sigmoid' ) )
```

```
model.add( Dense( output_dim=10 ) )  
model.add( Activation( 'softmax' ) )
```

# Keras

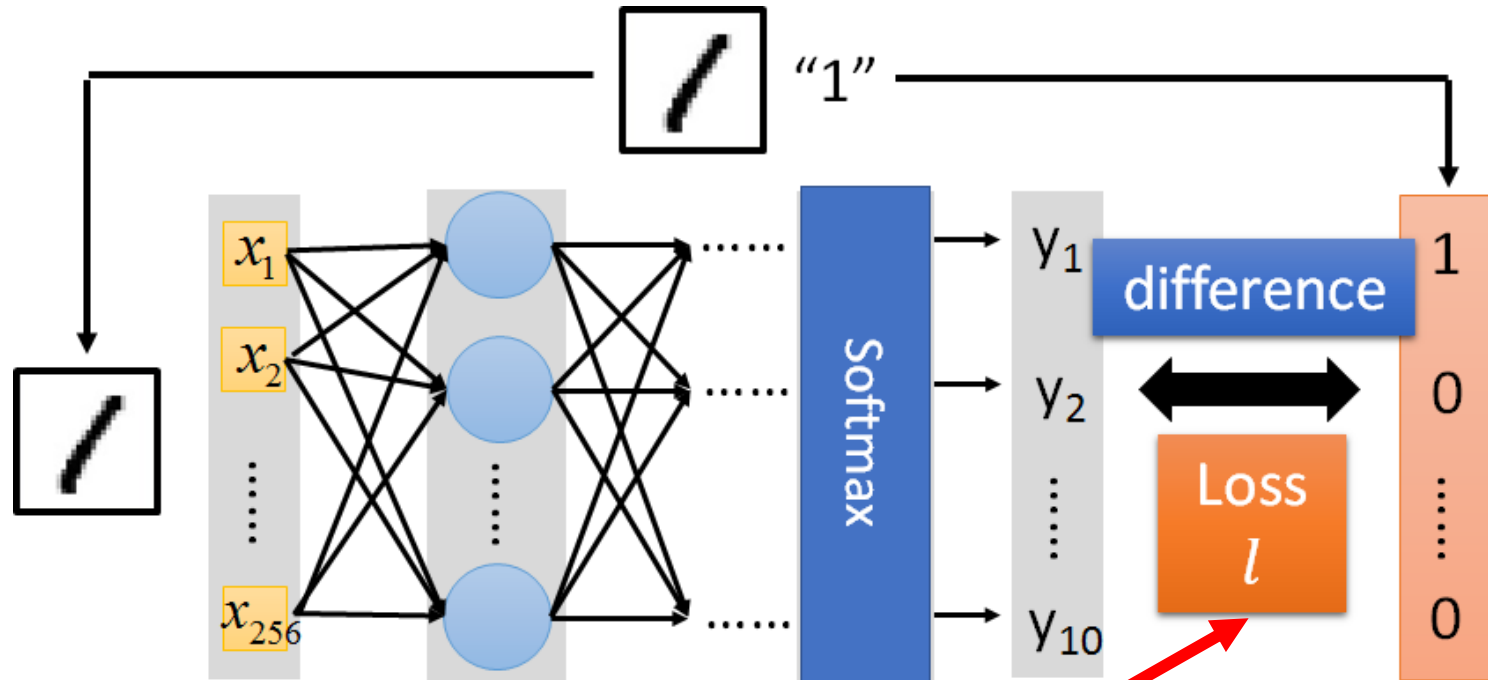
Step 1:  
define a set  
of function



Step 2:  
goodness of  
function



Step 3: pick  
the best  
function



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

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

# Keras



## Step 3.1: Configuration

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

**SGD, RMSprop, Adagrad, Adadelata, Adam, Adamax, Nadam**

## Step 3.2: Find the optimal network parameters

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

Training data  
(Images)

Labels  
(digits)

In the following slides



# Keras

Step 1:  
define a set  
of function



Step 2:  
goodness of  
function



Step 3: pick  
the best  
function

Step 3.2: Find the optimal network parameters

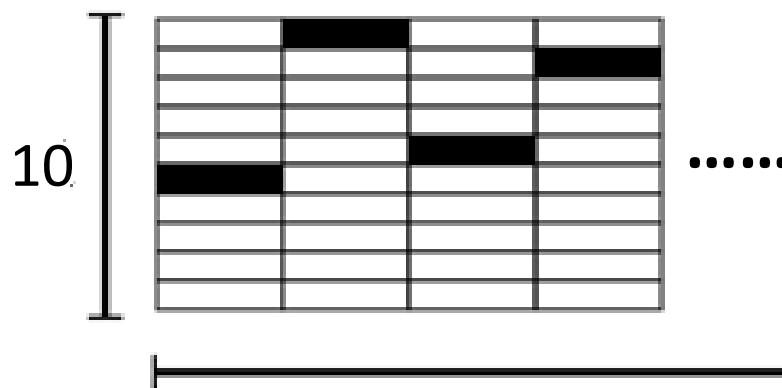
```
model.fit(x_train, y_train, batch_size=100, nb_epoch=20)
```

numpy array



Number of training examples

numpy array

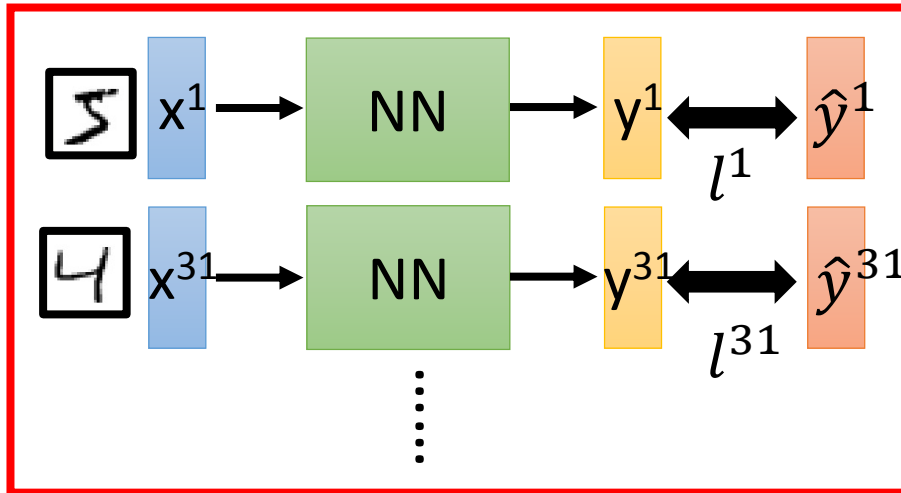


Number of training examples

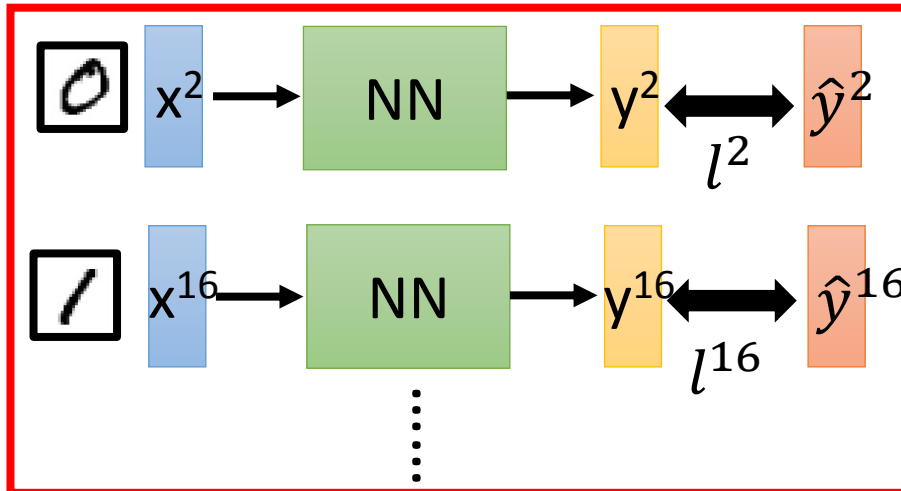
We do not really minimize total loss!

# Mini-batch

Mini-batch



Mini-batch



➤ Randomly initialize network parameters

➤ Pick the 1<sup>st</sup> batch

$$L' = l^1 + l^{31} + \dots$$

Update parameters once

➤ Pick the 2<sup>nd</sup> batch

$$L'' = l^2 + l^{16} + \dots$$

Update parameters once

:

➤ Until all mini-batches have been picked

one epoch

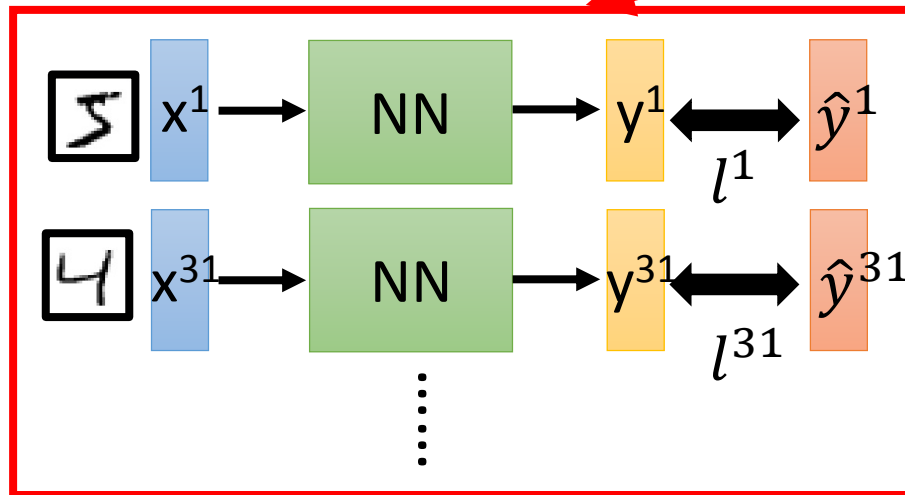
Repeat the above process

# 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

- Pick the 1<sup>st</sup> batch  
 $L' = l^1 + l^{31} + \dots$   
Update parameters once
- Pick the 2<sup>nd</sup> batch  
 $L'' = l^2 + l^{16} + \dots$   
Update parameters once
- ⋮
- Until all mini-batches have been picked

Repeat 20 times

one epoch

若batch size太大，其实可能没有遍历多少数据程序就卡到了鞍点或lm，浪费数据  
若bs太小，则机无法实现并行计算，耗时太久（取决于硬件）

# Speed

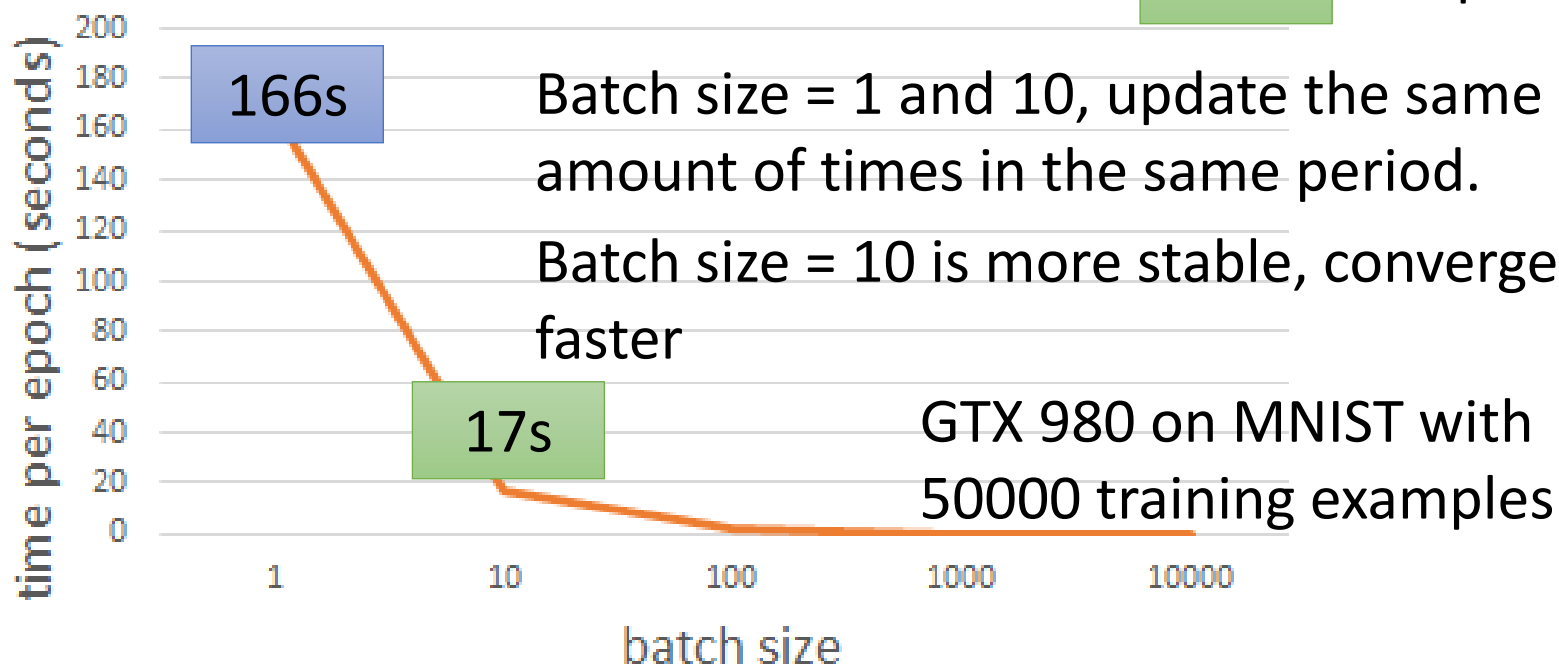
Very large batch size can yield worse performance

- Smaller batch size means more updates in one epoch

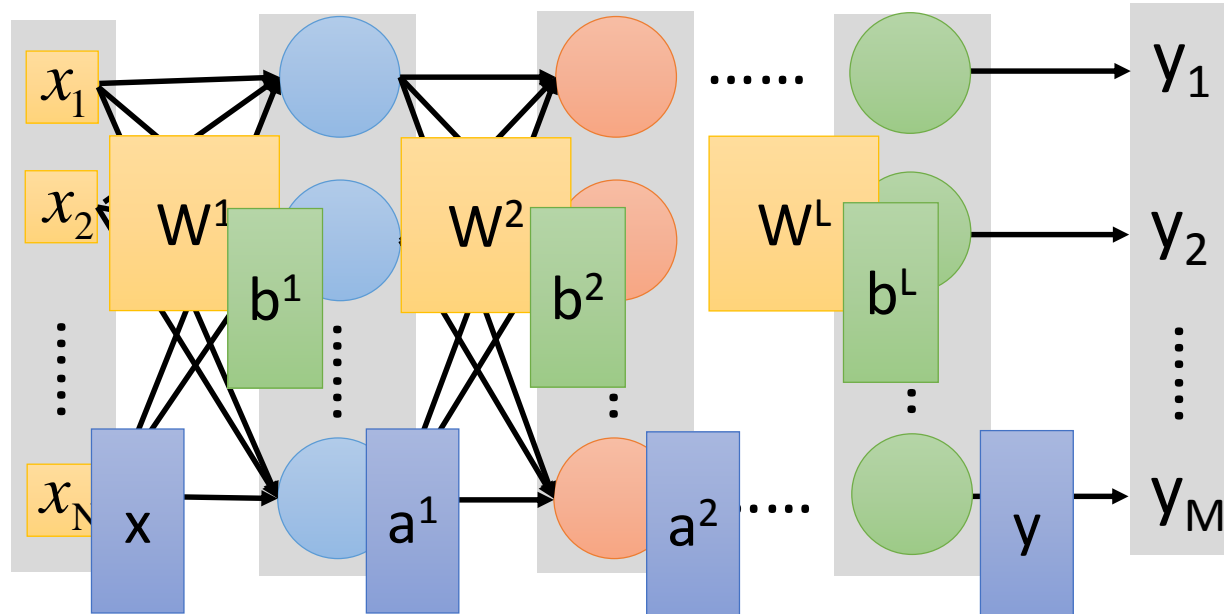
- E.g. 50000 examples

- batch size = 1, 50000 updates in one epoch 166s 1 epoch

- batch size = 10. 5000 updates in one epoch 17s 10 epoch



# Speed - Matrix Operation



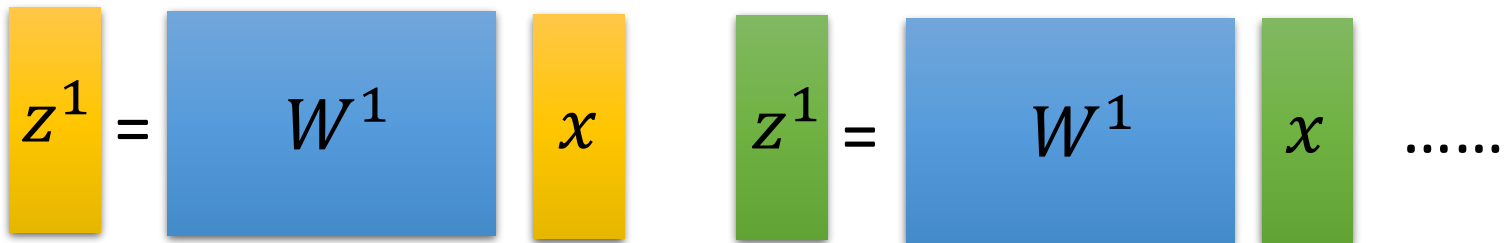
$$y = f(x) \quad \text{Forward pass (Backward pass is similar)}$$

$$= \sigma(W^L \dots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \dots + b^L)$$

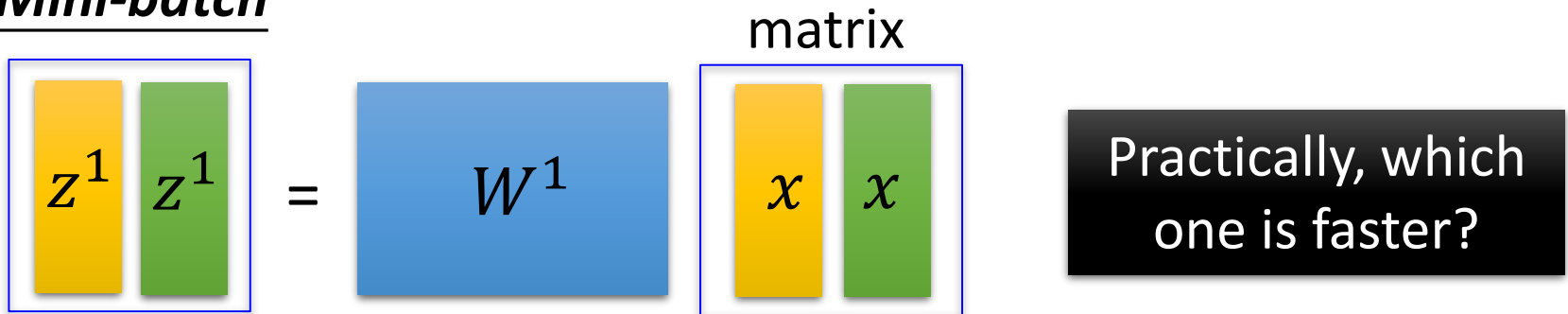
# Speed - Matrix Operation

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

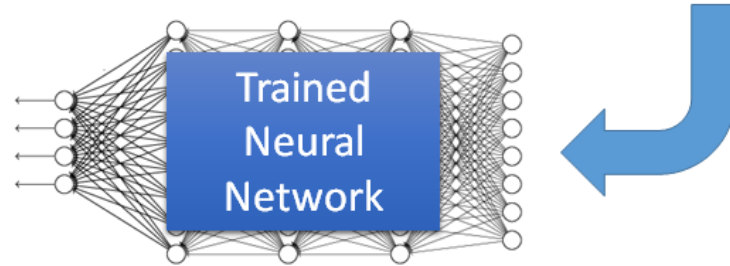
## Stochastic Gradient Descent



## Mini-batch



# Keras



Save and load models

<http://keras.io/getting-started/faq/#how-can-i-save-a-keras-model>

How to use the neural network (testing):

case 1: 

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

case 2: 

```
result = model.predict(x_test)
```

# Keras

- Using GPU to speed training
  - Way 1
    - `THEANO_FLAGS=device=gpu0 python YourCode.py`
  - Way 2 (in your code)
    - `import os`
    - `os.environ["THEANO_FLAGS"] = "device=gpu0"`



# Live Demo