“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/RNN%20training%20(v6).ecm.mp4/index.html

幫忙算微分

or

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

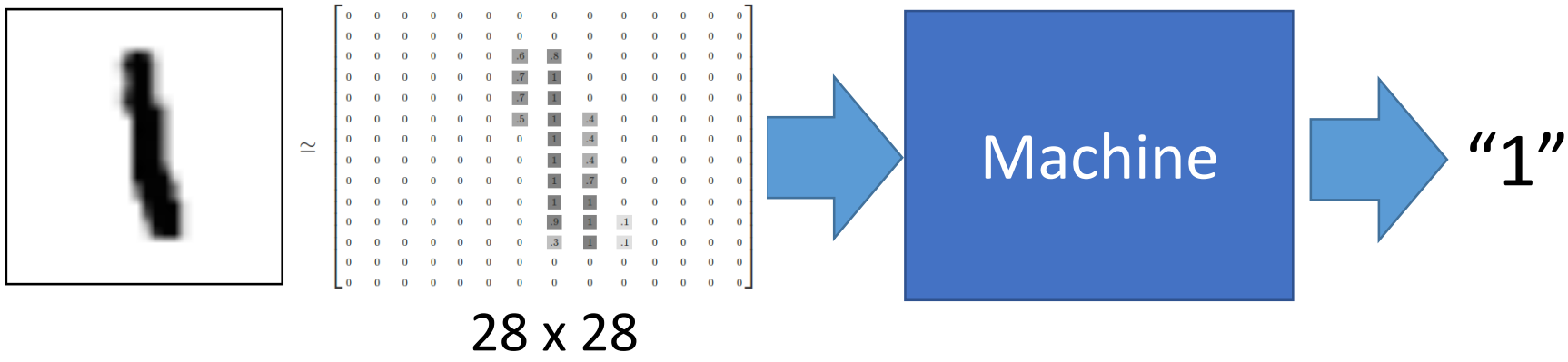
- François Chollet is the author of Keras.
  - He currently works for Google as a deep learning engineer and researcher.
- Keras means *horn* (牛角) in Greek
- Documentation: http://keras.io/
- Example: https://github.com/fchollet/keras/tree/master/examples

感謝 沈昇勳 同學提供圖檔

使用 Keras 心得

“Hello world”

- Handwriting Digit Recognition



data少但做起來有效果

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

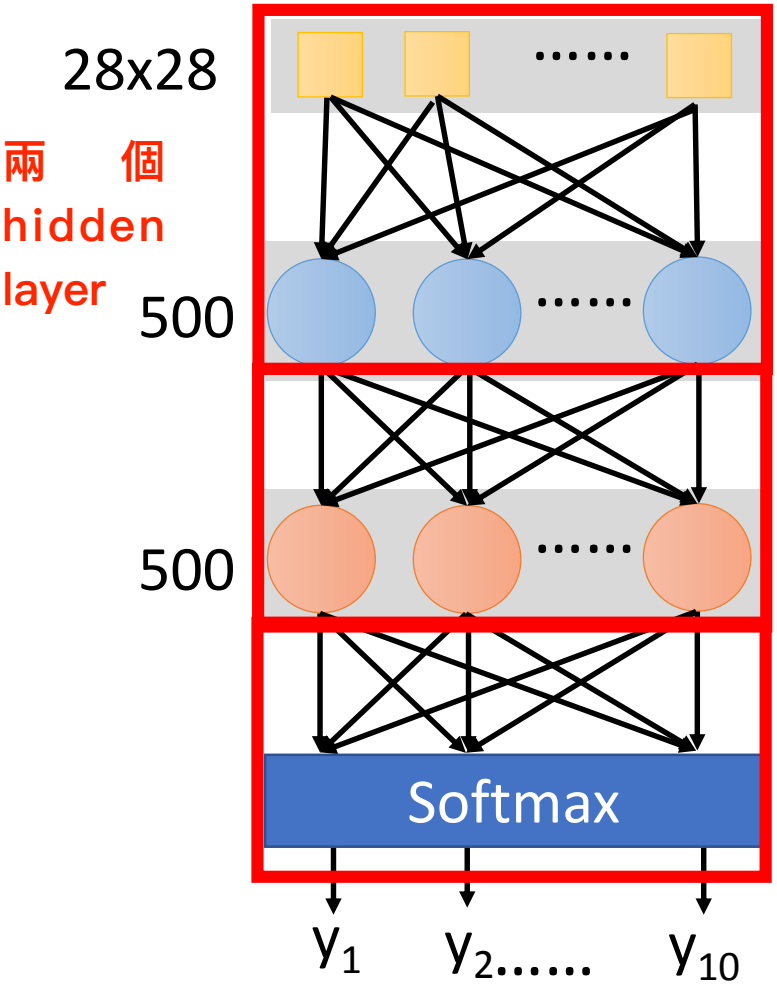
Keras provides data sets loading function: http://keras.io/datasets/

Keras

Step 1: define a set of function

Step 2: goodness of function

Step 3: pick the best function




```
model = Sequential()
```

宣告

fully connective = Dense

```
model.add( Dense( input_dim=28*28,
                  output_dim=500 ))
model.add( Activation('sigmoid') )
```

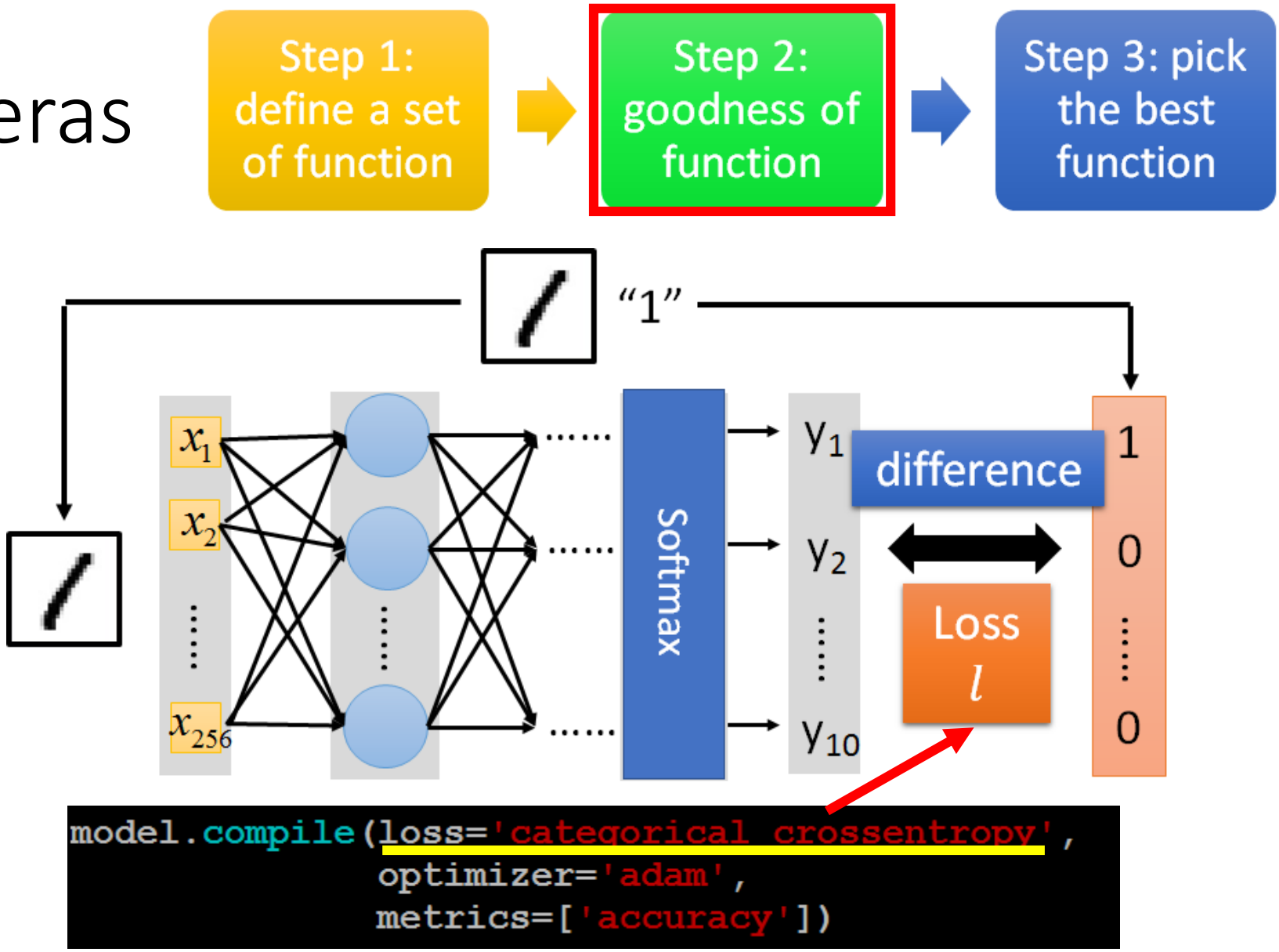
softplus, softsign, relu, tanh, hard_sigmoid, linear

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

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

output layer multi-class clssification的 activation function

Keras



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

Keras

Step 3.1: Configuration training前下的configuration

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

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

調整learning rate的方法

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

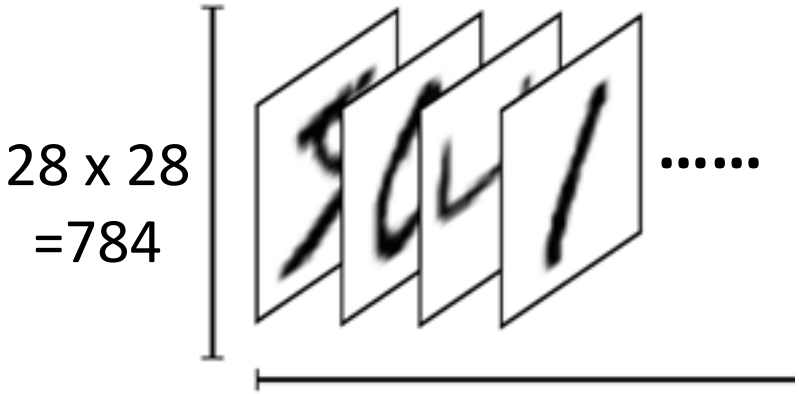
Step 3.2: Find the optimal network parameters

number of epoch

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

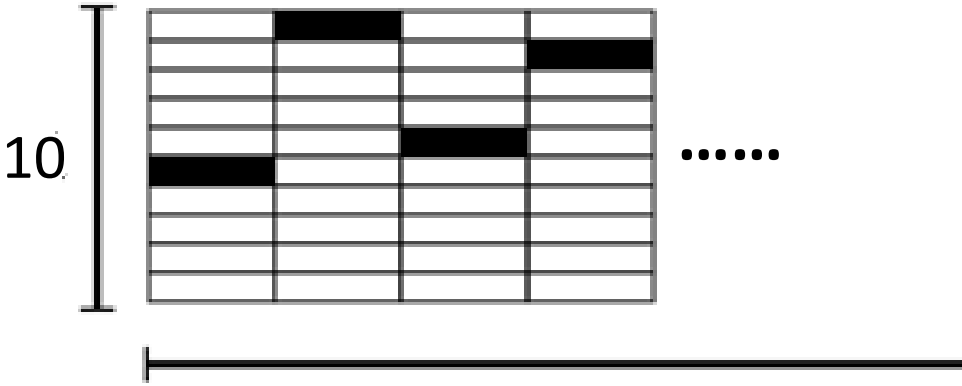
第一層dim為# of train data
第二層dim為# of pixels

numpy array



Number of training examples

numpy array

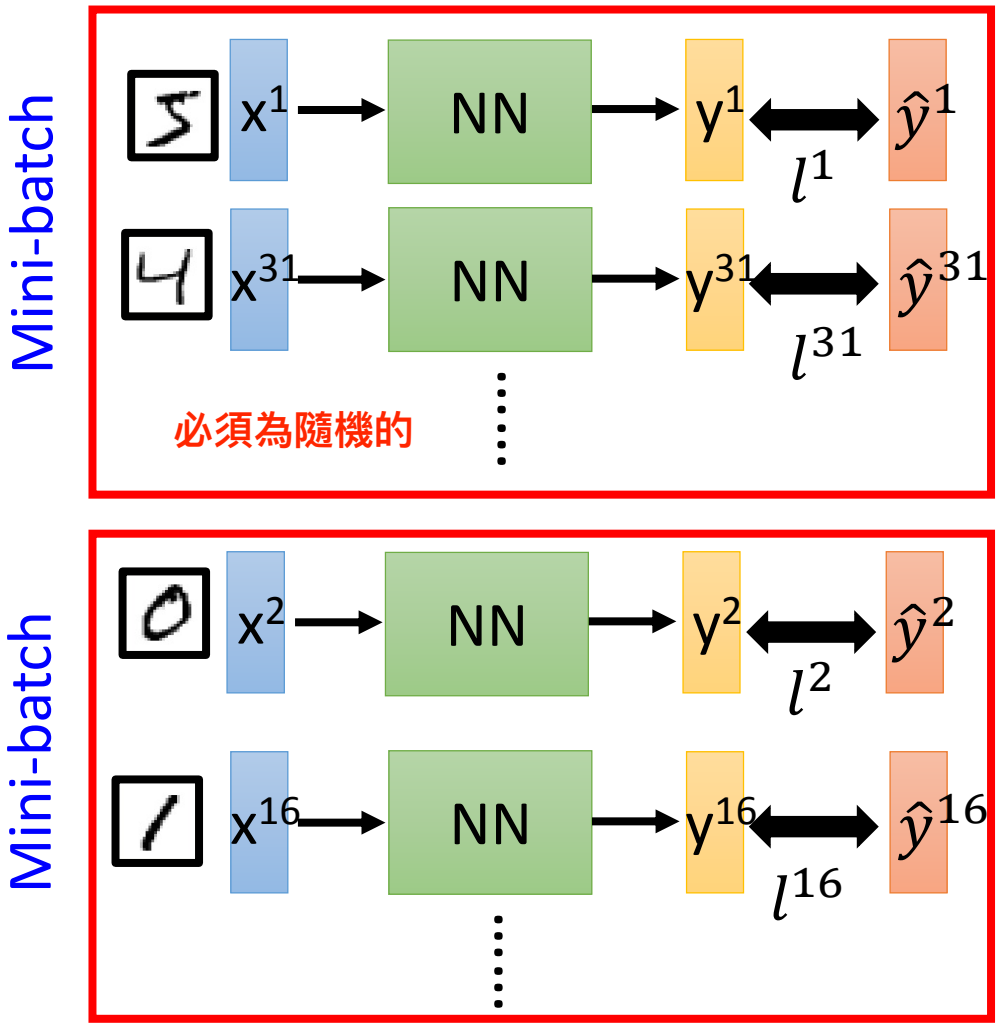


Number of training examples

https://www.tensorflow.org/versions/r0.8/tutorials/mnist/beginners/index.html

在做gradient時 We do not really minimize total loss!

Mini-batch



- Randomly initialize network parameters
- Pick the 1st batch

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

  Update parameters once
- Pick the 2nd batch

  sum of batch loss

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

  Update parameters once

  ⋮
- Until all mini-batches have been picked

one epoch

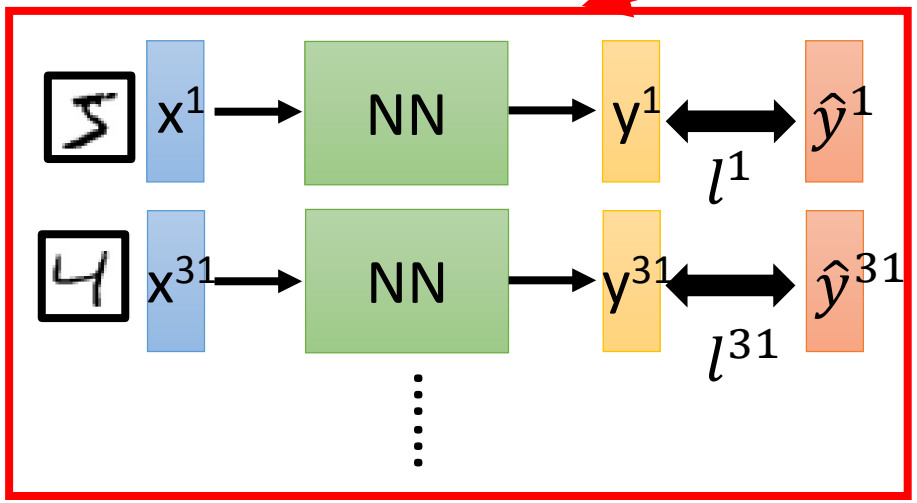
Repeat the above process

每一個batch跑一次更新一次參數,所有batch跑完稱作一個epoch

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)
```



100 examples in a mini-batch

Batch size = 1 ➡ 優點：速度快

Stochastic gradient descent

- Pick the 1st batch

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

  Update parameters once

- Pick the 2nd batch

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

  Update parameters once

  ⋮

- Until all mini-batches have been picked

每一個batch被看過20次 Repeat 20 times

one epoch

Speed

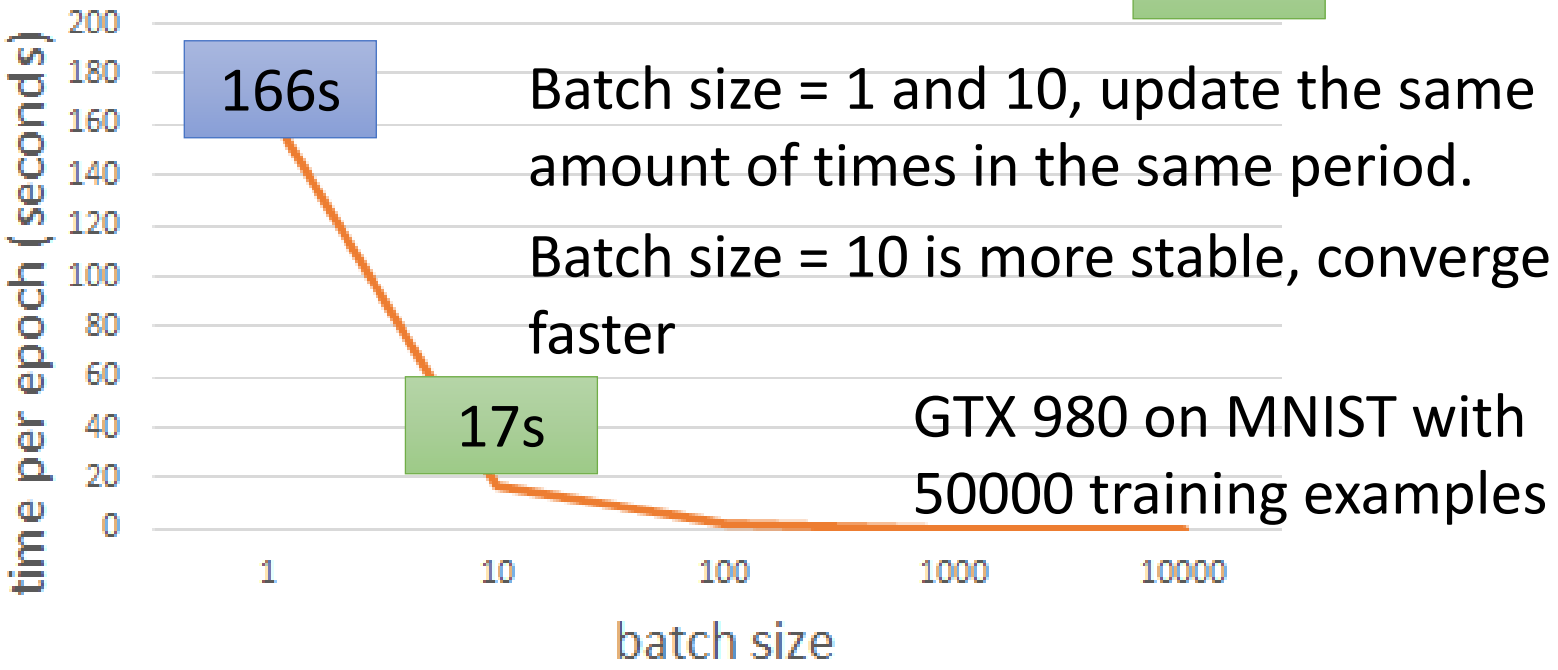
Very large batch size can yield worse performance

實作上的issue考量

- 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

166s 1 epoch

17s 10 epoch



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

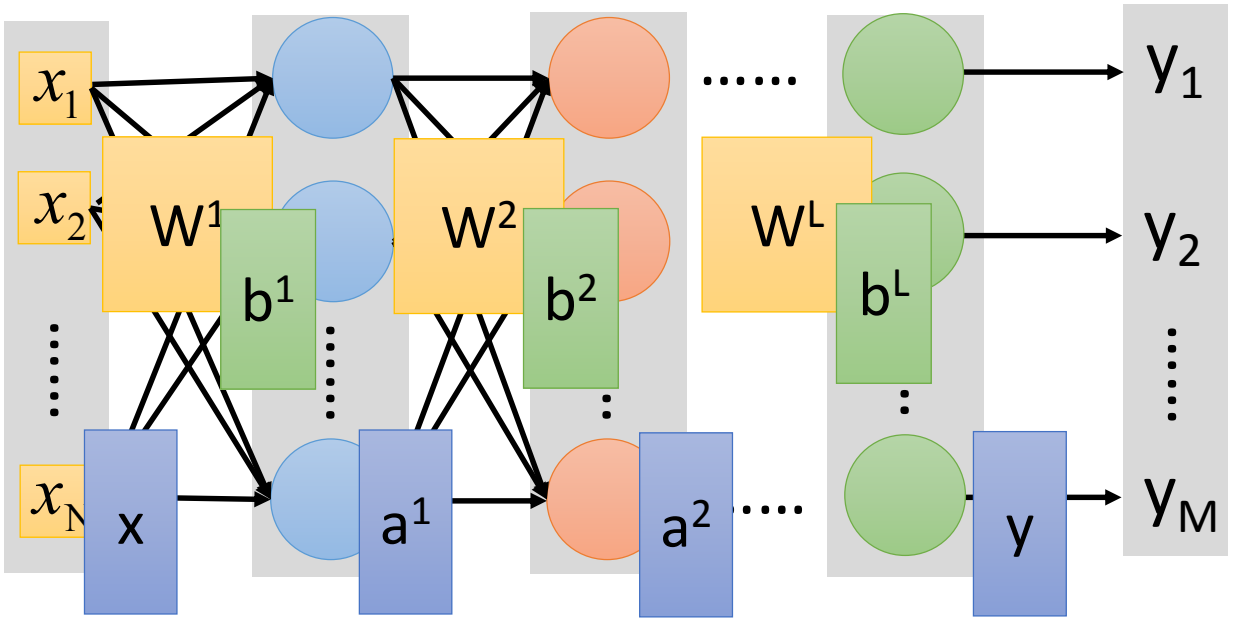
Batch size = 10 is more stable, converge faster

GTX 980 on MNIST with 50000 training examples

batch size如果太大 ex. 10000，時間還是會增長

batch size太大也會一下就卡在saddle point

Speed - Matrix Operation



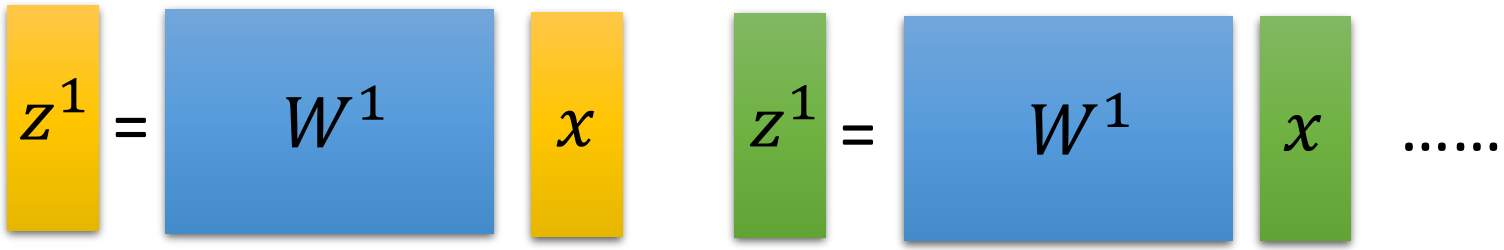
$$y = f(x)$$ Forward pass (Backward pass is similar)

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

Speed - Matrix Operation

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

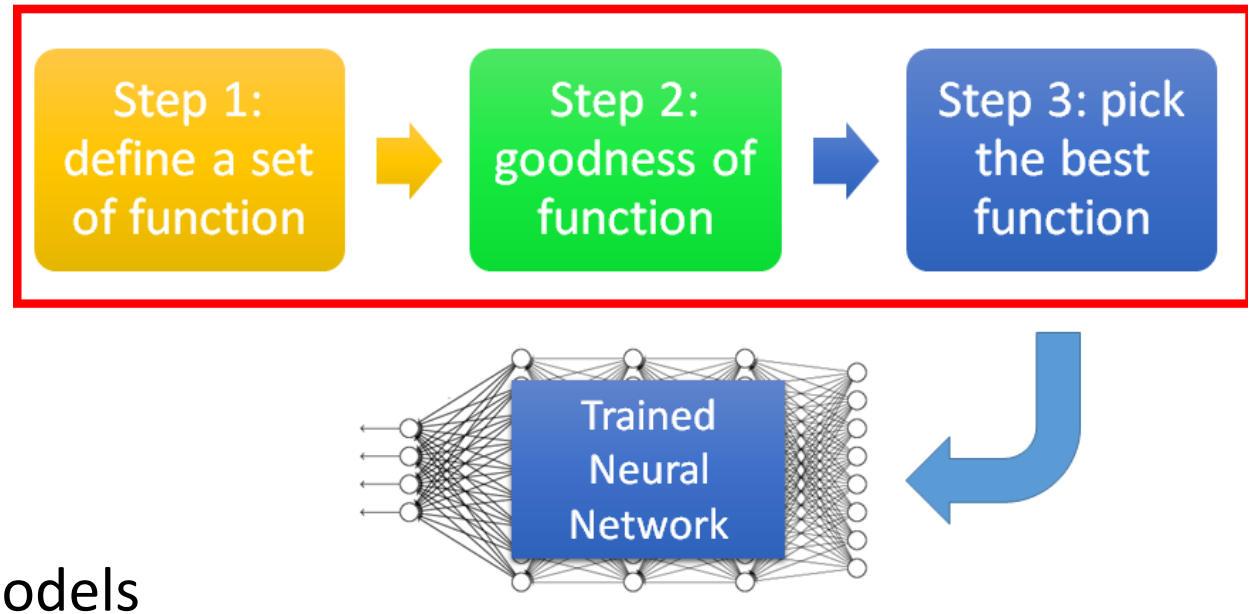
Stochastic Gradient Descent



Mini-batch 因為GPU的關係

Practically, which one is faster?

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)
```

預測結果

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