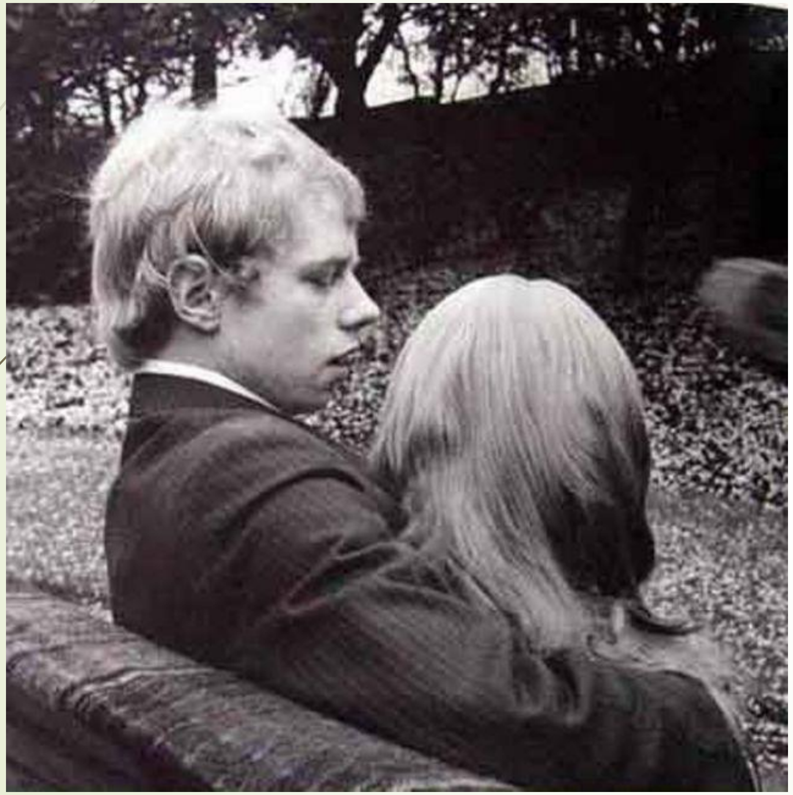
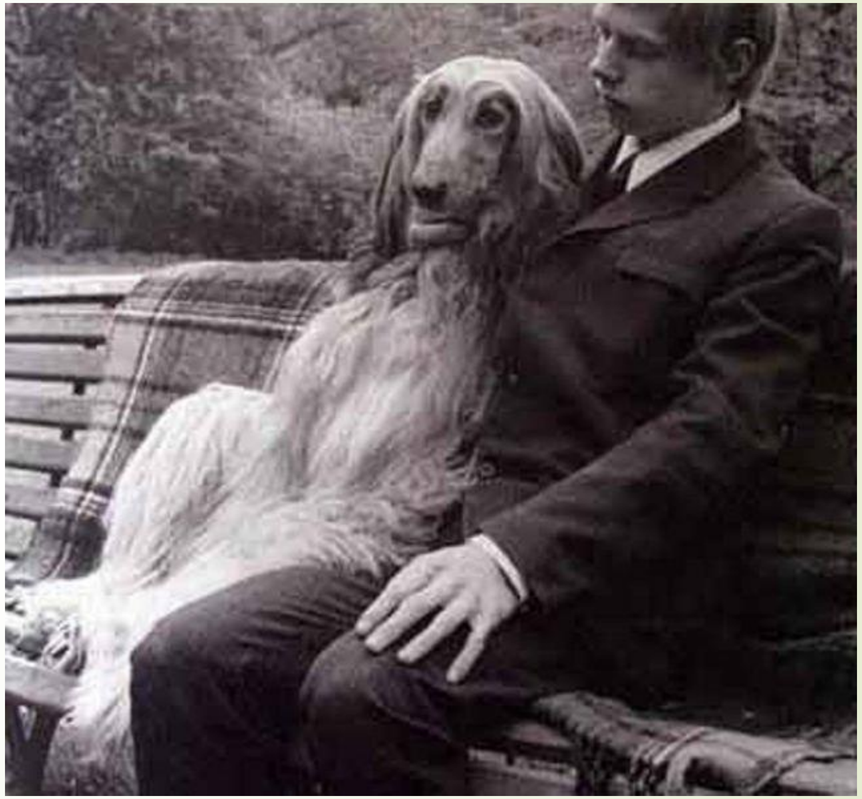
Tensor Flow for Deep Learning

Outline

- Tensor Flow 簡介
- 安裝 TensorFlow
  - Lab1: Tensorflow安裝練習
- Lab2: Implementing Softmax Regression
- Lab3: Deep Convolutional Networks in Tensorflow
- Final Report
  - Implementing a pattern recognition problem using Tensorflow

Object Recognition is not Easy





單一視角畫面存在太大不確定性

甚麼是Tensor Flow?

- Google (Brain Team)為機器學習(machine learning)提供的open source software library
- 很適合用以訓練(training)及製作深度學習類神經網路(deep neural networks)
- Example applications
  - image recognition, automated translation. Think Google Photos, Translate
  - Used in production at Uber, SnapChat, Google (obv.), others

深度學習製作套件

- Torch
- Caffe
- Theano (Keras, Lasagne)
- CuDNN
- TensorFlow
- Mxnet
- Etc.

深度學習模型製作方式

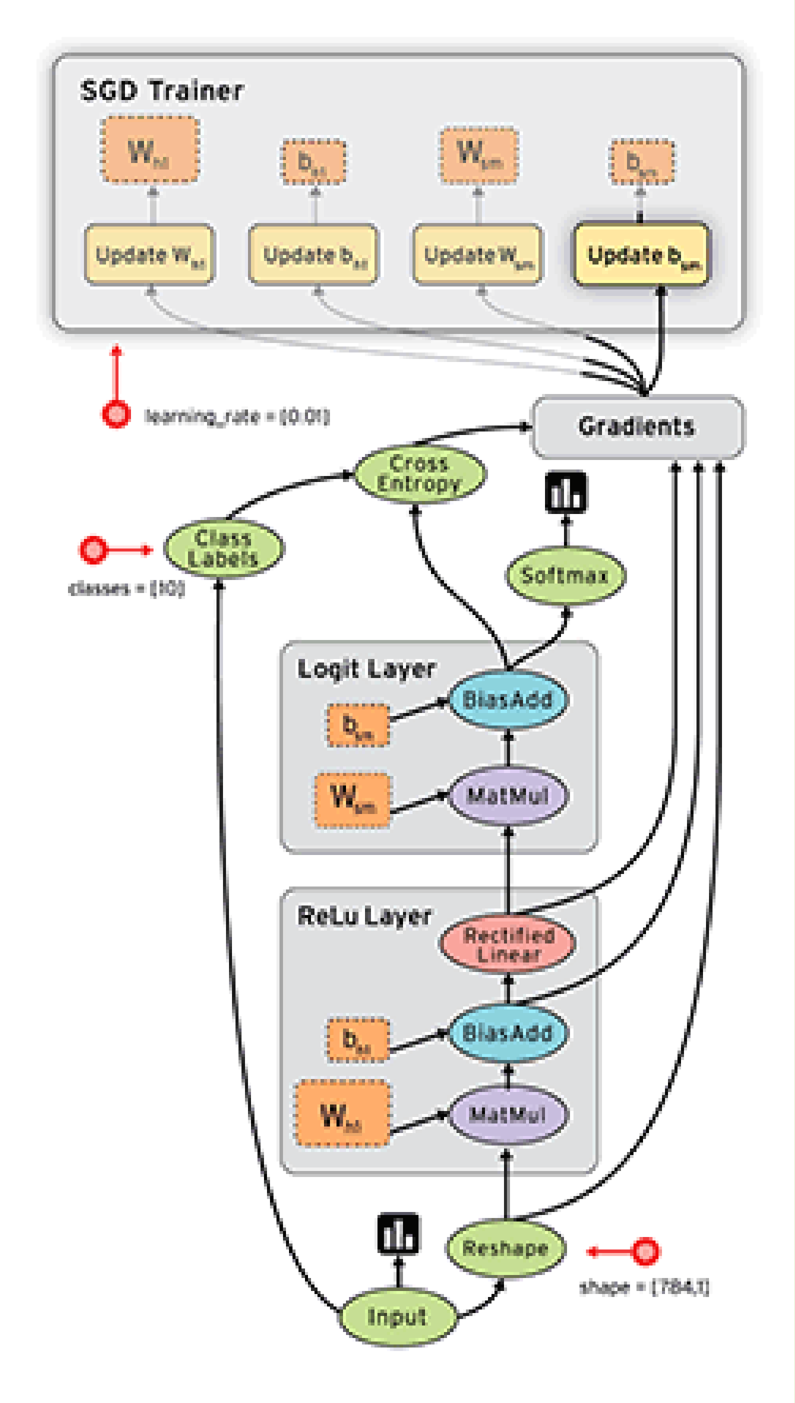
- 使用Configuration file
  - e.g. Caffe, DistBelief, CNTK
- 使用程式設定 (programmatic generation)
  - Torch, Theano, Tensorflow
- 製作深度學習的程式語言
  - Lua (Torch) vs. Python (Theano, Tensorflow) vs others.
- Python 是很好的選擇
  - because of rich community and library infrastructure

TensorFlow vs. Theano

- Theano is another deep-learning library with python wrapper (was inspiration for Tensorflow)
- Theano and TensorFlow are very similar systems.
  - TensorFlow has better support for distributed systems though, and has development funded by Google, while Theano is an academic project.

TensorFlow 的運作原理

- In Tensorflow computation represented using Graphs
- Graph comprises of nodes and edges
  - Nodes represent mathematical operations (op)
  - Edges represent multi-dimensional data arrays (tensors)
    - Data is represented as an edge (Tensor)
- Op takes Tensors and returns Tensors.
- Variables maintain state across executions of the graph.
- Two phases in the program:
  - Construct the computation graph.
  - Executes a graph in the context of a Session.
- Feed/fetch data to/from the graph.
- C based with Python and C++ APIs

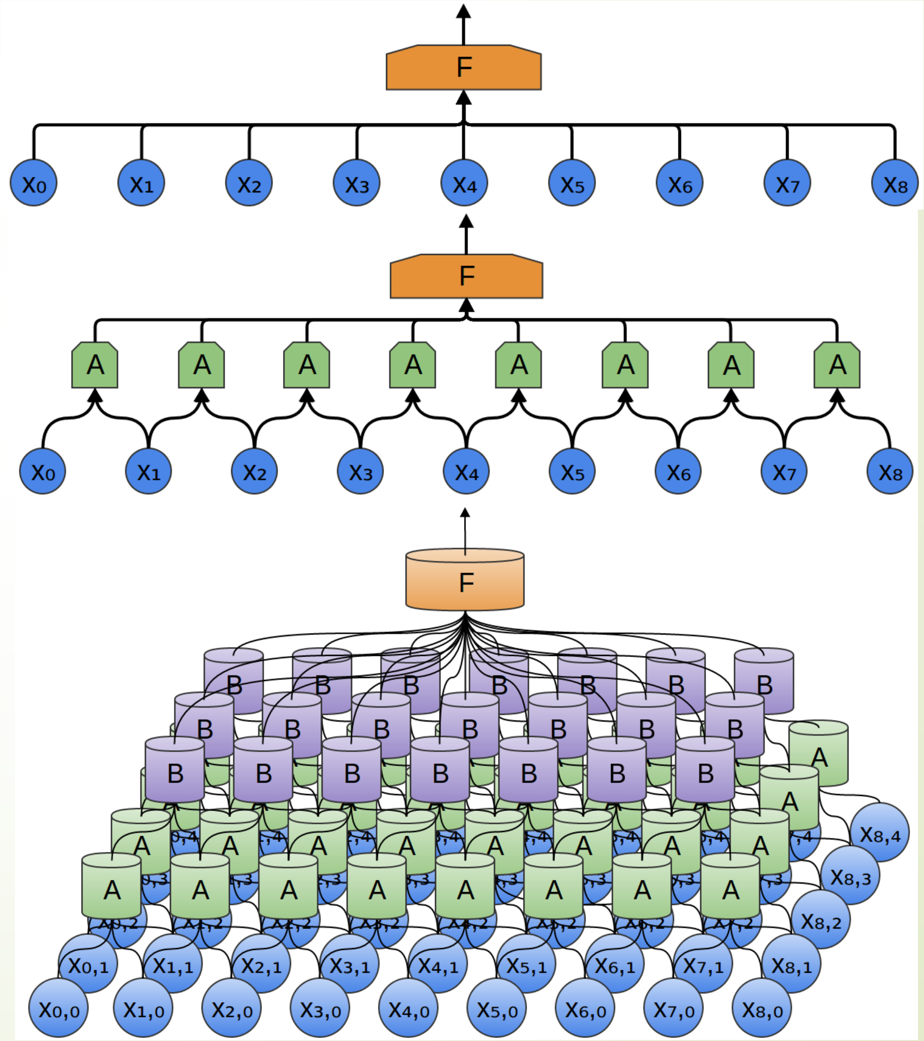


Construction of Computation Graph

- Always the same 3-steps pattern:
  - 1. inference() - Builds the graph as far as is required for running the network forward to make predictions.
  - 2. loss() - Adds to the inference graph the ops required to generate loss.
  - 3. training() - Adds to the loss graph the ops required to compute and apply gradients.

Deep Learning Neural Networks

- Neural networks algorithm with more than one layer
- Now possible with cheap HW, huge data sets and better techniques
    - Cross platform: Android, Linux, etc.
    - Quick turn around to production
    - Efficient computation utilizing CPUs, GPUs



甚麼是Tensor?

- Formally, tensors are multilinear maps from vector spaces to the real numbers (V vector space, and V^* dual space)

$$f : \underbrace{V^* \times \cdots V^*}_{p \text{ copies}} \times \underbrace{V \times \cdots V}_{q \text{ copies}} \to \mathbb{R}$$

- A scalar is a tensor: $(f\colon \mathbb{R} \to \mathbb{R}, f(e_1) = c)$
- A vector is a tensor: $(f\colon \mathbb{R}^n \to \mathbb{R}, f(e_i) = c_i)$
- A matrix is a tensor: $(f\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, f\left(e_i , e_j\right) = A_{ij})$
- Common to have fixed basis, so **a tensor can be represented as a multidimensional array of numbers**.

Outline

- Tensor Flow 簡介
- 安裝 TensorFlow
  - Lab1: Tensorflow安裝練習
- Lab2: Implementing Softmax Regression
- Lab3: Deep Convolutional Networks in Tensorflow
- Final Report
  - Implementing a pattern recognition problem using Tensorflow

可安裝Tensorflow有哪些平台?

- Installing TensorFlow on Ubuntu
- Installing TensorFlow on Mac OS X
- Installing TensorFlow on Windows
- Installing TensorFlow from Sources

Installing Tensorflow on Windows

- Determine which TensorFlow to install
  - **TensorFlow with CPU support only**.
  - **TensorFlow with GPU support**.
    - NVIDIA software must be installed on the system
    - CUDA® Toolkit 8.0. For details
- Determine how to install TensorFlow
  - "native" pip: Native pip installs TensorFlow directly on your system without going through a virtual environment.
  - Anaconda: use conda to create a virtual environment
    - Install TensorFlow with the pip install command

Installing with native pip

- If the following version of Python is not installed on your machine, install it now:
  - Python 3.5.x from python.org
- TensorFlow only supports version 3.5.x of Python on Windows.
- Note that Python 3.5.x comes with the pip3 package manager, which is the program you'll use to install TensorFlow.
- To install the CPU-only version of TensorFlow, enter the following command:
  - `C:\> pip3 install –upgrade tensorflow`
- To install the GPU version of TensorFlow, enter the following command:
  - `C:\> pip3 install –upgrade tensorflow-gpu`

Installing with AnaConda

- Create a virtual environment with anaconda (it takes some time)
- $ conda update conda
- $ conda create -n tensorflow python=3.5 anaconda
  - (tensorflow is the name of the environment, it can be whatever we want)
- Activate our new environment, prompt changes to (tensorflow)$
- $ activate tensorflow
- To deactivate the environment you have to write (do it at the end of the session) $ deactivate tensorflow
- To install the CPU-only version of TensorFlow, enter the following command:

```
(tensorflow)c: pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-1.2.1-cp35-cp35m-win_amd64.whl
```

- To install the GPU version of TensorFlow, enter the following command:

```
(tensorflow)c: pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-1.2.1-cp35-cp35m-win_amd64.whl
```

Validate Your Installation

- Start a terminal: C:\
- activate your Anaconda environment: $ activate tensorflow
- (tensorflow)$ python
- >>> import tensorflow as tf
- >>> hello = tf.constant(’Hello, TensorFlow!’)
- >>> sess = tf.Session()
- >>> print sess.run(hello)

  Hello, TensorFlow!
- >>> a = tf.constant(10)
- >>> b = tf.constant(32)
- >>> print sess.run(a + b)

  42
- >>>

Lab1: Tensorflow環境安裝

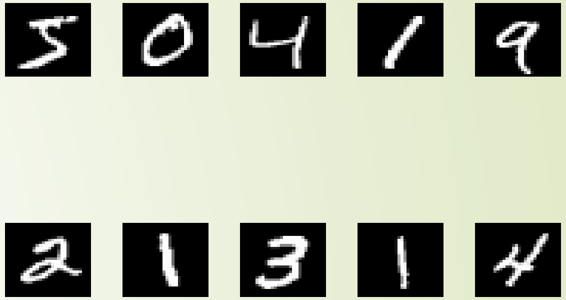
- The lab sessions focus on the implementation of a digital recognition system using Tensorflow throughout doing experiments on the MNIST digits dataset.
- **Task 1: Installing Tensorflow on your computer**
- **Task 2: Practice programming in python**
- **Task 3: Activate your Tensorflow environment**
- **Task 4: Write a lab report including the results of all tasks**

Outline

- Tensor Flow 簡介
- 安裝 TensorFlow
  - Lab1: Tensorflow安裝練習
- Lab2: Implementing Softmax Regression
- Lab3: Deep Convolutional Networks in Tensorflow
- Lab 4:
  - Implementing a pattern recognition problem using Tensorflow

MNIST Dataset: The Hello World of Pattern Recognition and Machine Learning

- Each image is 28 pixels by 28 pixels.
- 55,000 data points of training data (mnist.train)
- 10,000 points of test data (mnist.test)
- 5,000 points of validation data (mnist.validation).



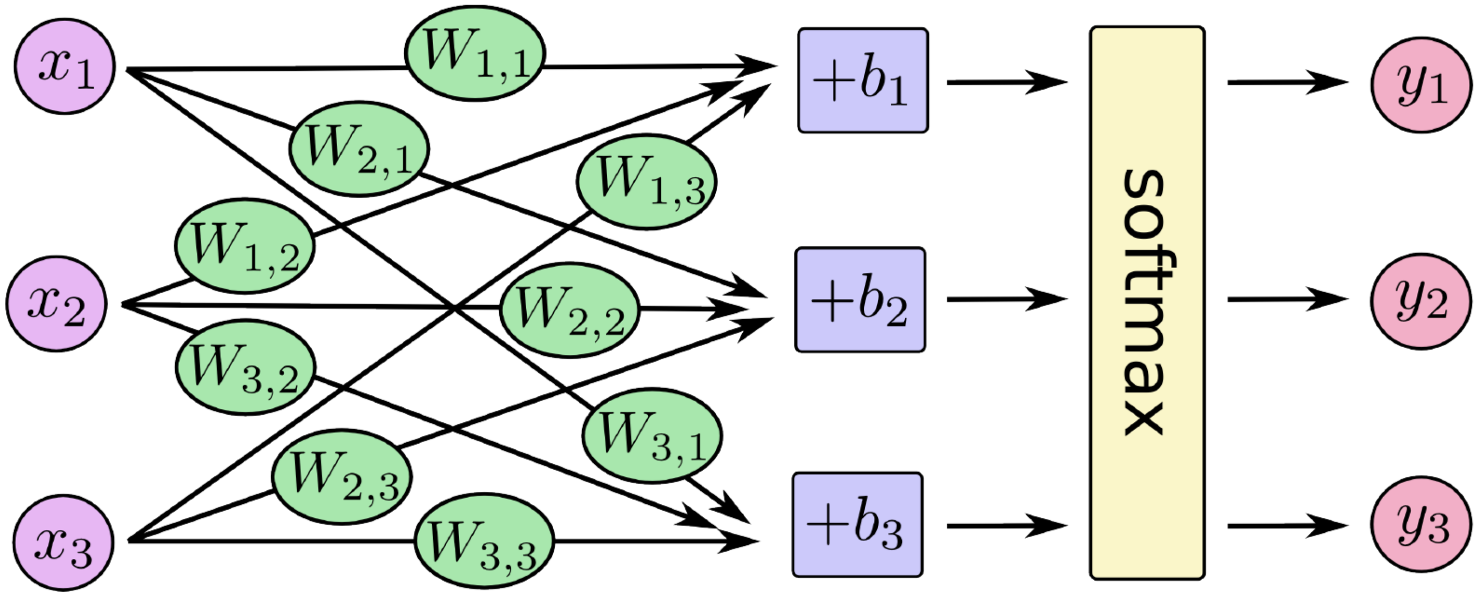
Use tensorflow.googlesource.com/tensorflow/+/master/tensorflow/examples/tutorials/mnist/input_data.py to download the data.

Softmax Regression

$$y = \mathrm{softmax}(Wx + b)$$

$$\mathrm{softmax}(x)_i = \frac{\exp x_i}{\sum_j \exp x_j}$$

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



Loading The MNIST Dataset

- 建立 mnist_softmax.py

```
import tensorflow as tf
import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
```

- 啟動Anaconda Tensorflow Terminal
- 執行 Python mnist_softmax.py

加入Placeholder Variables

- Mnist_softmax.py: 變數宣告

```
x = tf.placeholder(tf.float32, [None, 784])
y_ = tf.placeholder(tf.float32, [None, 10])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
```

組裝Computation Graph (I)

- mnist softmax.py: Inference

```
y = tf.nn.softmax(tf.matmul(x, W) + b)
```

- mnist softmax.py: Loos computation

```
cross_entropy = -tf.reduce_sum(y_*tf.log(y))
```

組裝Computation Graph (II)

- mnist softmax.py : Training

```
train_step = tf.train.GradientDescentOptimizer(0.01).
          minimize(cross_entropy)
```

- 也可使用其他Tensorflow提供的optimization algorithms, using one is as simple as tweaking one line:
  - class tf.train.AdagradOptimizer
  - class tf.train.MomentumOptimizer
  - class tf.train.AdamOptimizer

Evaluation

- To verify the correctness of the regression classifier
- mnist softmax.py: Evaluation

```
correct_prediction = tf.equal(tf.argmax(y,1), tf.argmax(y_,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float"))
```

執行Computation Graph

- mnist softmax.py: Initialize all the variables

```
 init = tf.initialize_all_variables()
```

- mnist softmax.py: Start a new session

```
 sess = tf.Session()
 sess.run(init)
# Let's train -- we'll run the training step 1000 times!
for i in range(1000):
    batch_xs, batch_ys = mnist.train.next_batch(100)
       sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})
print sess.run(accuracy, feed_dict={x: mnist.test.images,
y_: mnist.test.labels})
```

Result around 91 %: VERY BAD for MNIST

Lab2: MNIST_SOFTMAX效能改善

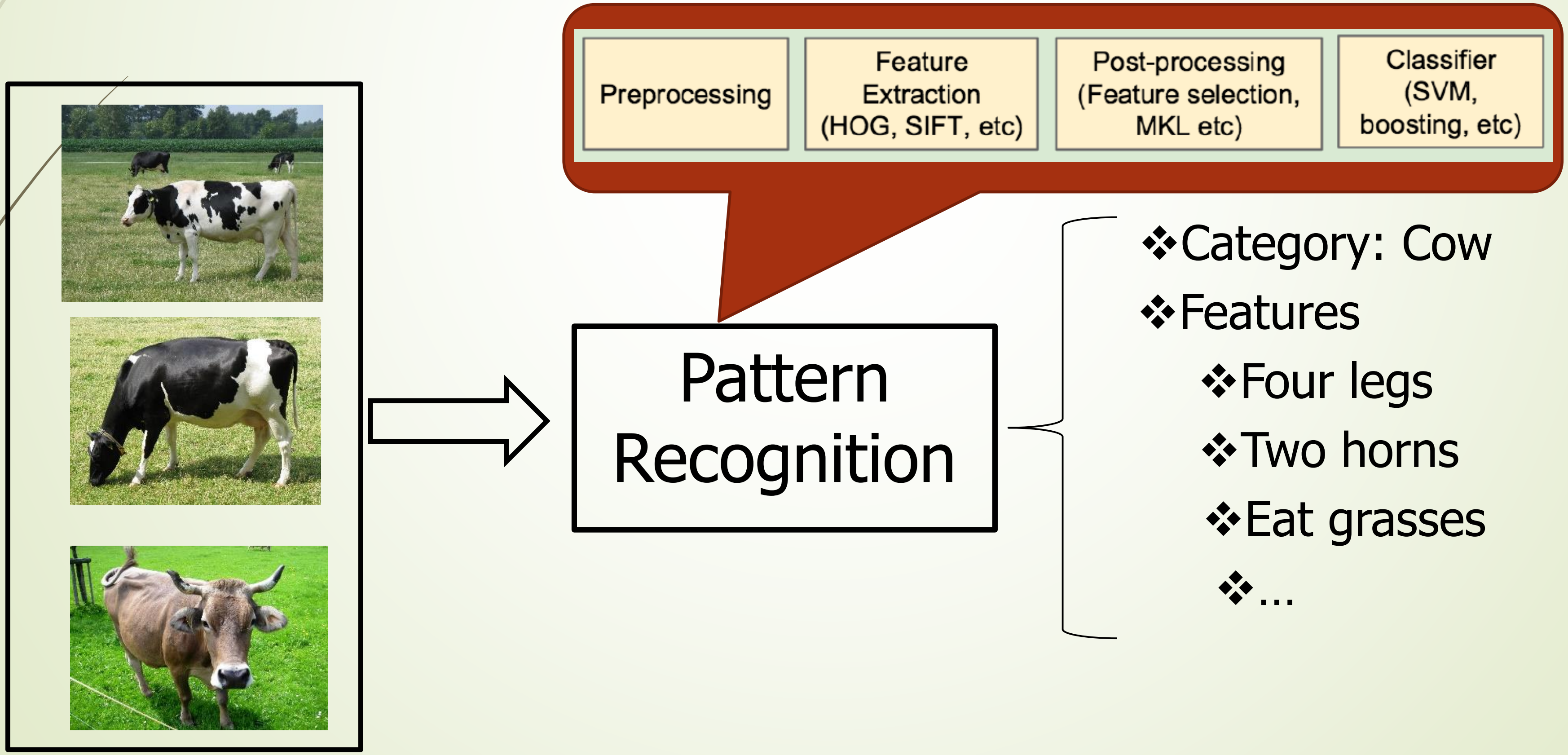
- 如前面所述，簡易版的MNIST_SOFTMAX的效能普通，本實驗的任務是修改參數改善效能
- Task 1: 實作MNIST_SOFTMAX.PY
- Task 2: 定義可以改善效能的參數
- Task 3: 改寫MNIST_SOFTMAX.PY以利改善效能
- Task 4: Write a lab report including the results of all tasks

Outline

- Tensor Flow 簡介
- 安裝 TensorFlow
  - Lab1: Tensorflow安裝練習
- Lab2: Implementing Softmax Regression
- Lab3: Deep Convolutional Networks in Tensorflow
- Final Report
  - Implementing a pattern recognition problem using Tensorflow

Deep Convolutional Neural Networks in Tensorflow (I)

- State-of-the-art of Image Recognition.
- Traditional Approach: Handmade features



Preprocessing

Feature Extraction (HOG, SIFT, etc)

Post-processing (Feature selection, MKL etc)

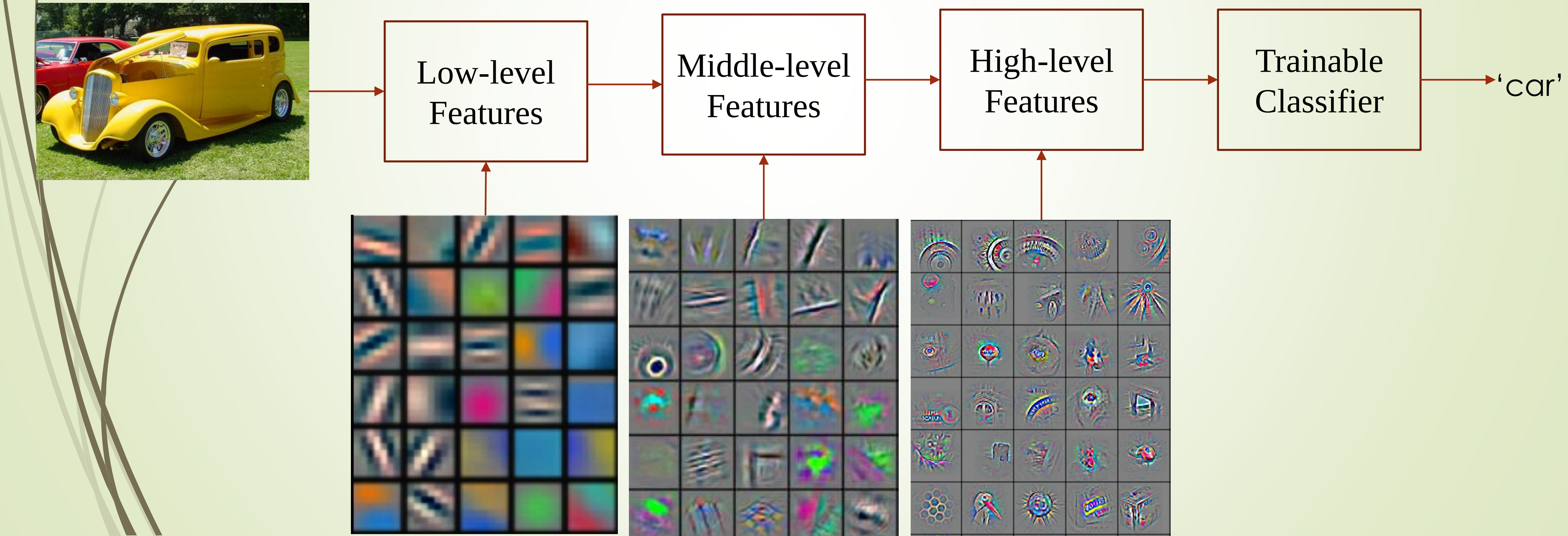
Classifier (SVM, boosting, etc)

Pattern Recognition

- ❖Category: Cow
- ❖Features
  - ❖Four legs
  - ❖Two horns
  - ❖Eat grasses
  - ❖…

Deep Convolutional Neural Networks in Tensorflow (II)

- Deep Learning learns features + classifiers.
- Using complex non-linear map from pixels to labels



Low-level Features

Middle-level Features

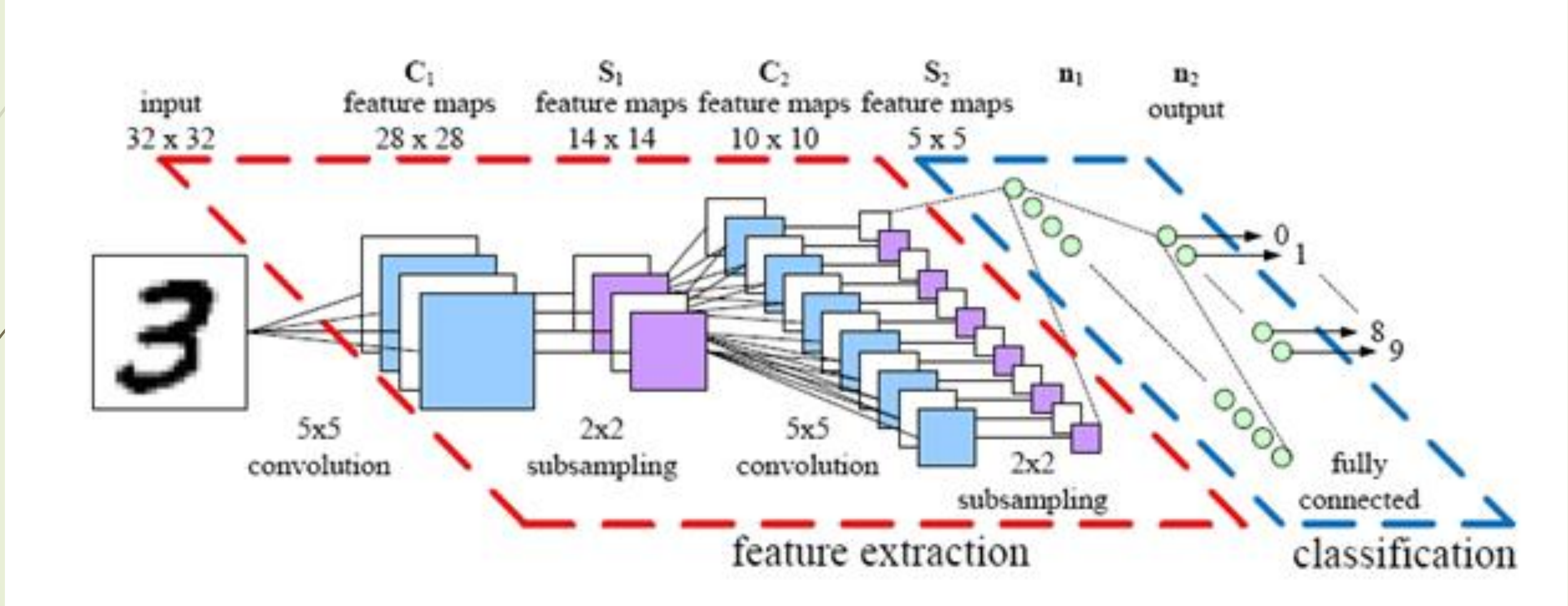
High-level Features

Trainable Classifier

'car'

Deep Convolutional Neural Networks in Tensorflow (II)

Source: http://parse.ele.tue.nl



- Use convolutional operator to extract features
- Use subsampling to do dimension redup tion
- Train using Backpropagation
- This works very well. Why?

  Rick Baraniuk \opinion": A Probabilistic Theory of Deep Learning.

Cnn for Mnist in Tensorflow

- Mnist_deep.py

```
 import tensorflow as tf
 import input_data
 mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)

 x = tf.placeholder("float", shape=[None, 784])
y_ = tf.placeholder("float", shape=[None, 10])
```

Cnn for Mnist in Tensorflow

- We will have to initialize a lot of weights.
- Mnist_deep.py: weight initialization

```
def weight_variable(shape):
    initial = tf.truncated_normal(shape, stddev=0.1)
return tf.Variable(initial)
def bias_variable(shape):
    initial = tf.constant(0.1, shape=shape)
 return tf.Variable(initial)
```

Cnn for Mnist in Tensorflow

- Convolution and pooling operations. We will use them in different layers.
- Mnist_deep.py: Convolution and pooling

```
def conv2d(x, W):
return tf.nn.conv2d(x, W, strides=[1, 1, 1, 1], padding='SAME')
def max_pool_2x2(x):
return tf.nn.max_pool(x, ksize=[1, 2, 2, 1],
          strides=[1, 2, 2, 1], padding='SAME')
```

Cnn for Mnist in Tensorflow

- First layer: From input data to second layer
- Mnist_deep.py: First Convolutional Layer

```
W_conv1 = weight_variable([5, 5, 1, 32])
b_conv1 = bias_variable([32])
x_image = tf.reshape(x, [-1, 28, 28, 1])
h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1) + b_conv1)
h_pool1 = max_pool_2x2(h_conv1)
```

Cnn for Mnist in Tensorflow

- Second layer: From ouput of first layer to FC layer
- Mnist_deep.py: Second Convolutional Layer

```
W_conv2 = weight_variable([5, 5, 32, 64])
b_conv2 = bias_variable([64])
h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2) + b_conv2)
h_pool2 = max_pool_2x2(h_conv2)
```

Cnn for Mnist in Tensorflow

- Mnist_deep.py: Fully Connected layer

```
# Densely connected layer
W_fc1 = weight_variable([7 * 7 * 64, 1024])
 b_fc1 = bias_variable([1024])
 h_pool2_flat = tf.reshape(h_pool2, [-1, 7*7*64])
 h_fc1 = tf.nn.relu(tf.matmul(h_pool2_flat, W_fc1) + b_fc1)
```

Cnn for Mnist in Tensorflow

- Mnist_deep.py: Train and Evaluate

```
cross_entropy = -tf.reduce_sum(y_*tf.log(y_conv))
train_step = tf.train.AdamOptimizer(1e-4).minimize(cross_entropy)
correct_prediction = tf.equal(tf.argmax(y_conv,1), tf.argmax(y_,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float"))
```

Cnn for Mnist in Tensorflow

- Mnist_deep.py: Execute

```
  init = tf.initialize_all_variables()
sess = tf.InteractiveSession()
 sess.run(init)
 for i in range(20000):
    batch = mnist.train.next_batch(50)
   if i%100 == 0:
      train_accuracy = accuracy.eval(feed_dict={
           x:batch[0], y_: batch[1], keep_prob: 1.0})
     print "step %d, training accuracy %g"%(i, train_accuracy)
   train_step.run(feed_dict={x: batch[0], y_: batch[1], keep_prob: 0.5})
 print "test accuracy %g"%accuracy.eval(feed_dict={
        x: mnist.test.images, y_: mnist.test.labels, keep_prob: 1.0})
```

- Go back to work while it finishes: Accuracy 99;2 %.

What else?

- Tensorboard.
  https://www.tensorflow.org/versions/0.6.0/how_tos/summaries_and_tensorboard/index.html
- Vector Representation of Words (word2vec).
- Recurrent Neural Networks (Long short-term memory Networks, seq2seq models).
- General Mathematics (Mandelbrot Set, Partial Dierential Equations)
- Udacity free online course

Lab3: MNIST_DEEP人臉辨識系統製作

- **Task1: MNIST_DEEP的效能卓越，本實驗的任務是置換 MNIST Dataset 為任一人臉辨識DATASET，以利完成一基於深度學習之人臉辨識系統**
  - **請自行上網蒐尋DATASET**
- **Task 1: 實作MNIST_DEEP.PY**
- **Task 2: 修改Dataset載入功能**
- **Task 3: 改寫MNIST_DEEP.PY為FACE_CNNPY，以利完成人臉辨識系統建置**
- **Task 4: Write a lab report including the results of all tasks**

Outline

- Tensor Flow 簡介
- 安裝 TensorFlow
  - Lab1: Tensorflow安裝練習
- Lab2: Implementing Softmax Regression
- Lab3: Deep Convolutional Networks in Tensorflow
- Lab4: 題目自訂
  - Implementing a machine learning problem using Tensorflow

References

- Source code - https://www.tensorflow.org/
- Convolution neural networks - http://colah.github.io/posts/2014-07-Conv-Nets-Modular/
- Pattern recognition and ML reading - https://github.com/rasbt/pattern_classification, https://github.com/rasbt/python-machine-learning-book
- Retraining last layer of model - https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/#0