



北京交通大学

# 图像处理与机器学习

Digital Image Processing and Machine Learning

主讲人：黄琳琳

电子信息工程学院



## 第八章 深度学习基础

- ◆ 深度学习引言
- ◆ 卷积神经网络
- ◆ 几种典型网络
- ◆ 问题及方向



# 深度学习基础

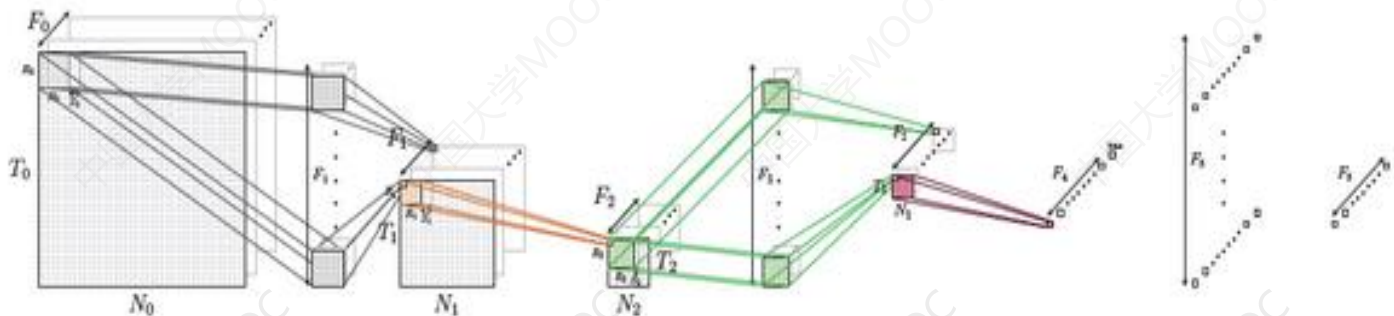
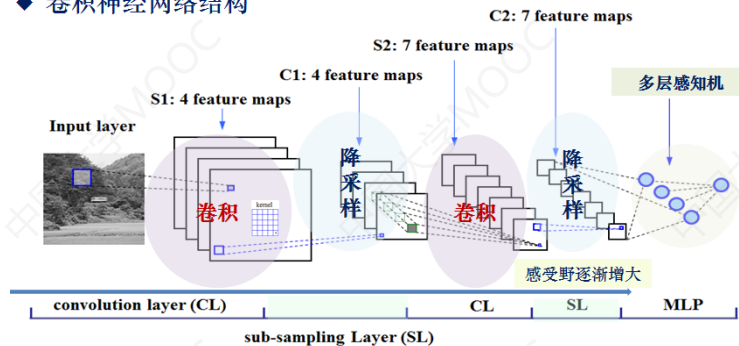
- ◆ Recurrent Neural Network, RNN  
(循环神经网络)
- ◆ Long Short-Term Memory Neural Network, LSTM  
(长短时记忆神经网络)



# 循环神经网络 (RNN)

- 卷积神经网络是前馈网络
- 网络每层之间节点无连接
- 适用静态数据，如图像等

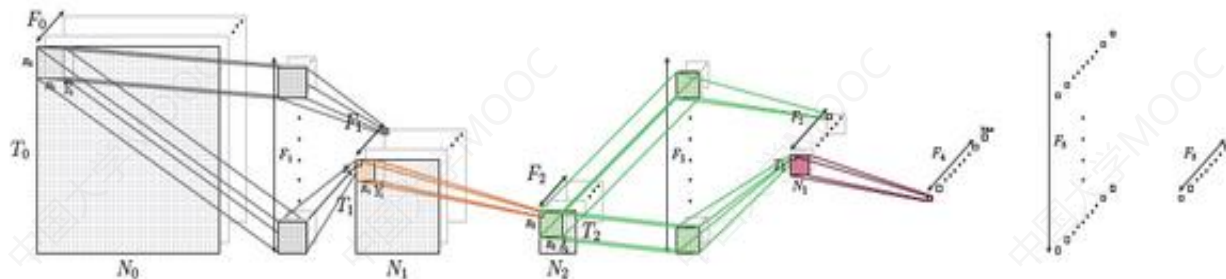
◆ 卷积神经网络结构



前馈网络输入之间完全没有关系，因此只能单独处理独立的输入

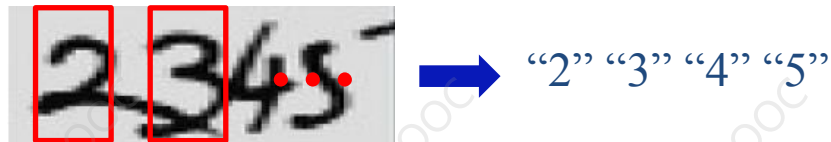


# 循环神经网络 (RNN)



- 语音识别
- 机器翻译
- 字符串识别
- 视频分析

前馈 网络 输入 之间 没有 关系 ...

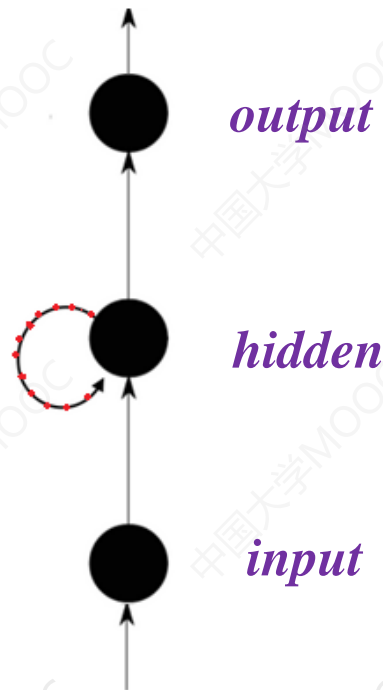


Recurrent Neural Networks, RNN  
(循环神经网络)



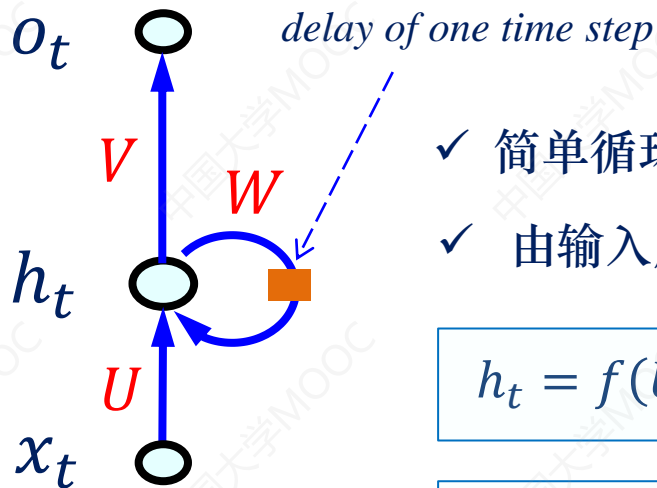
# 循环神经网络 (RNN)

- 反馈网络，模拟“人脑记忆功能”
- 对序列动态数据进行智能处理
  - ✓ 通过使用带自反馈的神经元
    - 能够处理任意长度的序列
  - ✓ 一个序列当前的输出
    - 与当前输入及之前输出有关





# 循环神经网络 (RNN)



- ✓ 简单循环网络 (Simple Recurrent Network) [Elman, 1990]
- ✓ 由输入层、一个隐藏层和一个输出层组成。

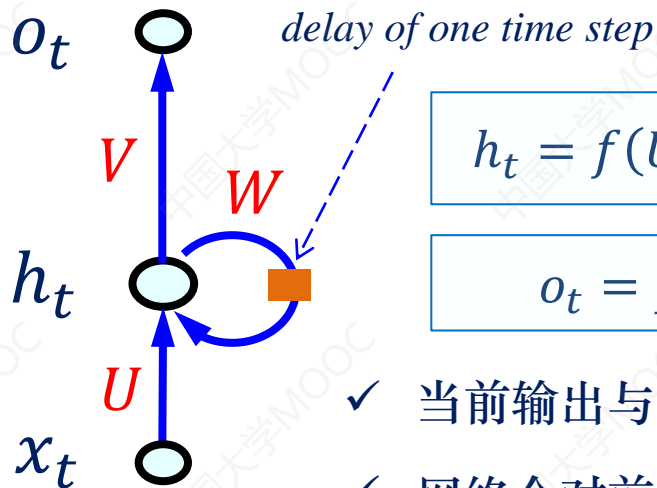
$$h_t = f(Ux_t + Wh_{t-1} + b)$$

$$o_t = f(Vh_t + b')$$

$f$  是非线性函数，通常为 *logistic* 函数或 *tanh* 函数



# 循环神经网络 (RNN)



$$h_t = f(Ux_t + Wh_{t-1} + b)$$

$$o_t = f(Vh_t + b')$$

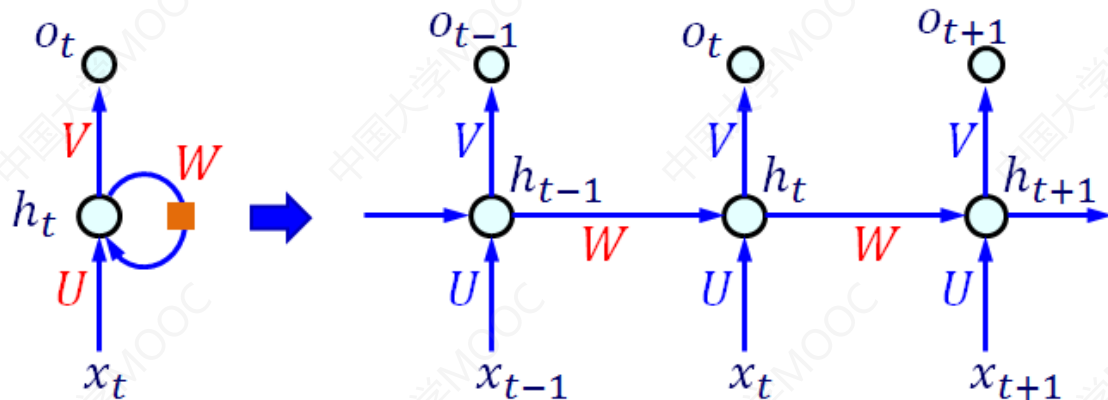
- ✓ 当前输出与当前输入以及之前输出有关
- ✓ 网络会对前面的信息进行**记忆**并应用于当前输出的计算中
- ✓ 隐藏层之间的节点不再无连接而是**有连接**的
- ✓ 隐藏层的输入包括输入层以及上一时刻隐藏层的输出





# 循环神经网络 (RNN)

- ◆ 按时间顺序展开：天然的深度神经网络



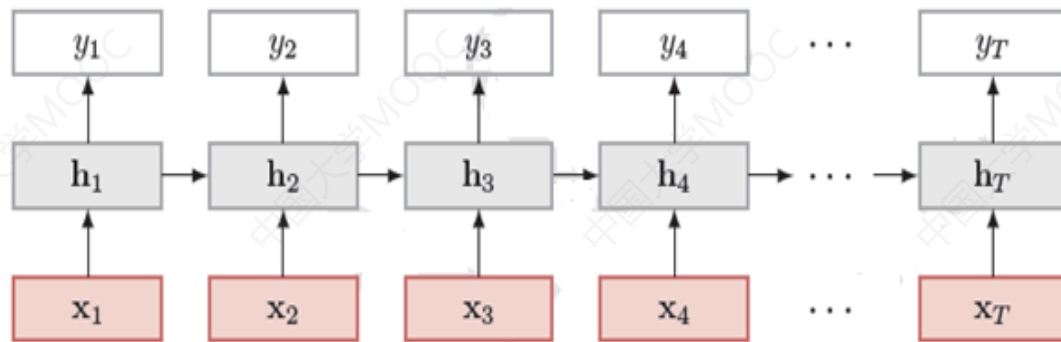
**U**: input-hidden    **V**: hidden-output    **W**: hidden-hidden

$U, V, W$ : 对每一个time-step时刻  $t$  全局共享

$U, V, W$ : 使用**BP**算法，针对特定任务进行学习优化



# 循环神经网络 (RNN)



$$h_t = f(Ux_t + Wh_{t-1} + b) \quad h_1 = f(Ux_1 + Wh_0 + b)$$

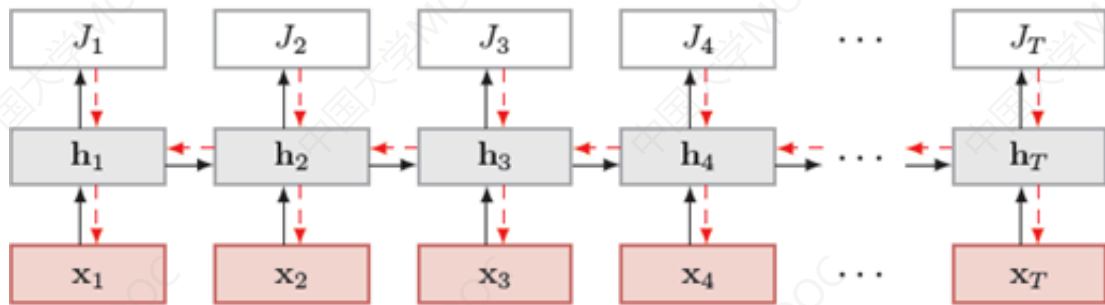
$$h_2 = f(Ux_2 + Wh_1 + b)$$

$$h_T = f(Ux_T + Wh_{T-1} + b)$$



# 循环神经网络：训练

- 网络训练：Back Propagation Through Time (BPTT) 算法



- 假设网络在每时刻  $t$  有一个监督信息，损失为  $J_t$ ，则整个序列的损失为

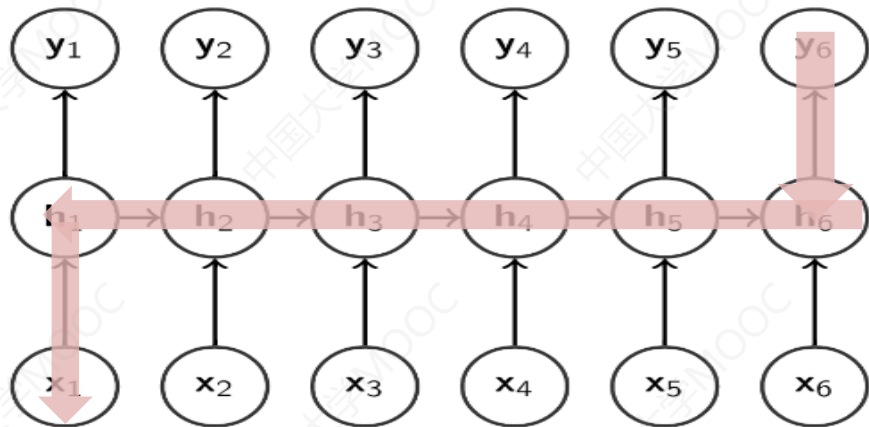
$$J = \sum_{t=1}^T J_t$$

- 损失  $J$  关于  $W$  的梯度为：
$$\frac{\partial J}{\partial W} = \sum_{t=1}^T \frac{\partial J_t}{\partial W}$$



# 循环神经网络：训练

- 目标函数有多个极小值，有很多平坦区域，也有十分陡峭的悬崖。



- 梯度消减：Jacobian矩阵奇异值  $< 1$
- 梯度爆炸：Jacobian矩阵奇异值  $> 1$

从理论上可以建立长时间间隔的状态之间的依赖关系（Long-Term Dependencies），但是由于梯度消减或爆炸问题，实际上只能学习到短周期的依赖关系。

Hochreiter and Schmidhuber [1997] 引入了门机制（Gating Mechanism）控制信息的累积速度，并可以选择遗忘之前累积的信息。

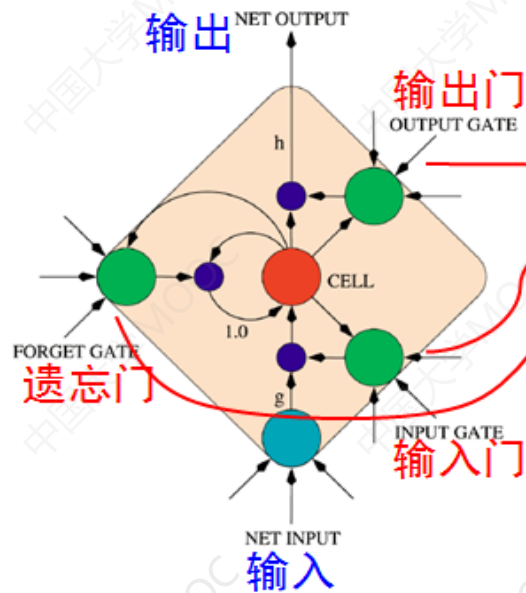
长短记忆神经网络

(Long Short-Term Memory Neural Network, LSTM)



# 长短记忆神经网络 (LSTM)

- 有选择性的记忆或者遗忘



$$\begin{aligned}o_t &= \text{sigm}(W_o x_t + U_o h_{t-1} + b_o), \\i_t &= \text{sigm}(W_i x_t + U_i h_{t-1} + b_i), \\f_t &= \text{sigm}(W_f x_t + U_f h_{t-1} + b_f), \\\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c), \\c_t &= i_t \odot \tilde{c}_t + f_t \odot c_{t-1}, \\h_t &= o_t \odot \tanh(c_t),\end{aligned}$$

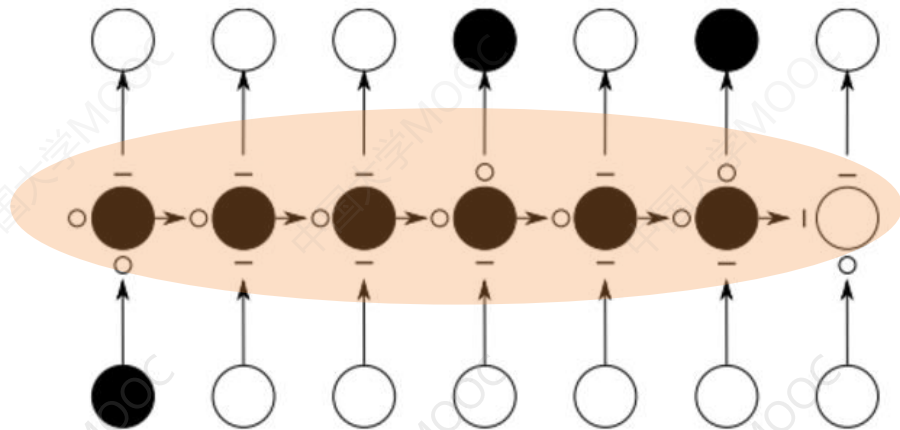


# 长短记忆神经网络 (LSTM)

## ◆ LSTM有三个门, 分别对应:

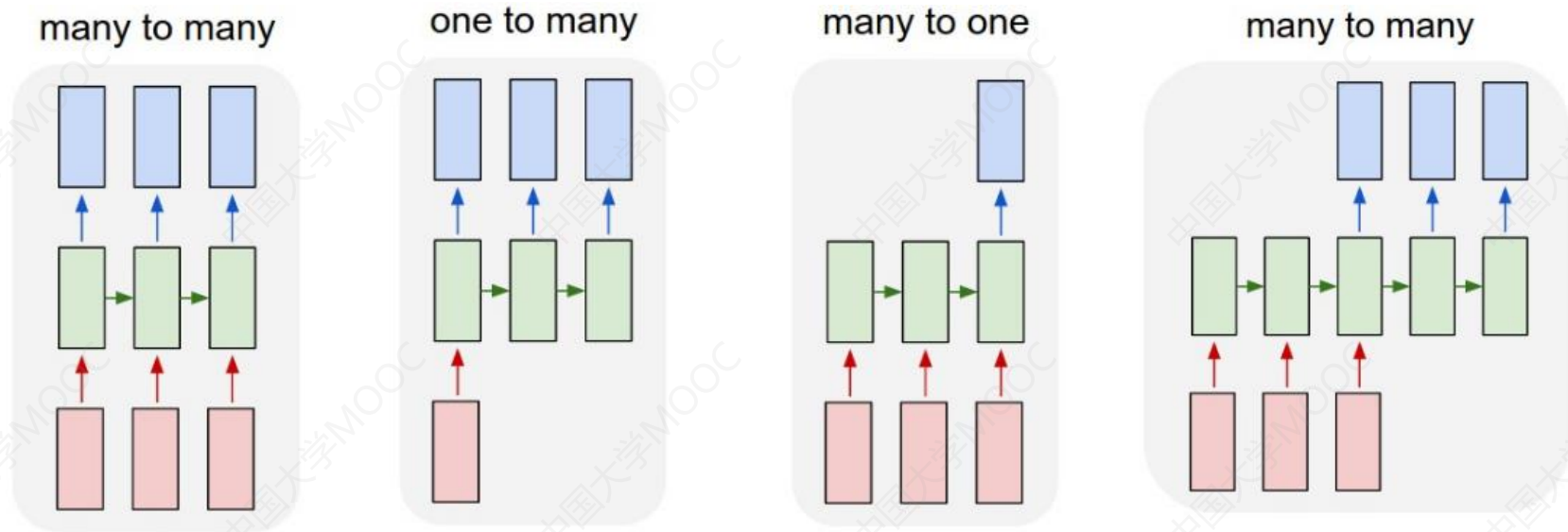
- 输入门: 输入信息是否有意义, 如否则关闭
- 遗忘门: 历史信息是否重要, 如否则关闭
- 输出门: 此时是否要回应, 如否则关闭

LSTM





# 长短记忆神经网络 (LSTM)



- ✓ 根据任务需求，灵活设定网络结构
- ✓ 完成各种不可思议的任务！



# 长短记忆神经网络 (LSTM)

传统方法

检测银行卡号位置



6 . . . 8 4 . . .

622848 0462290014713

切割字符

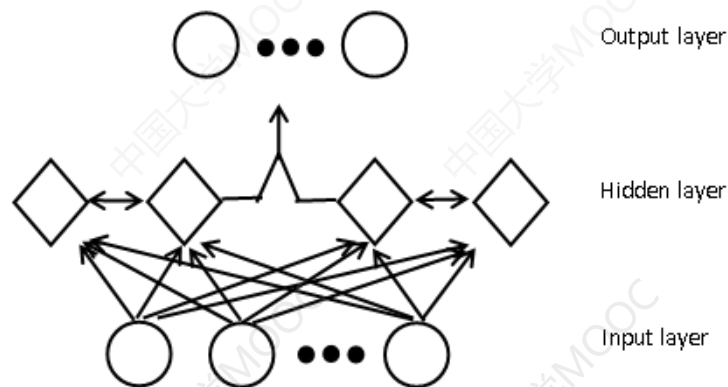
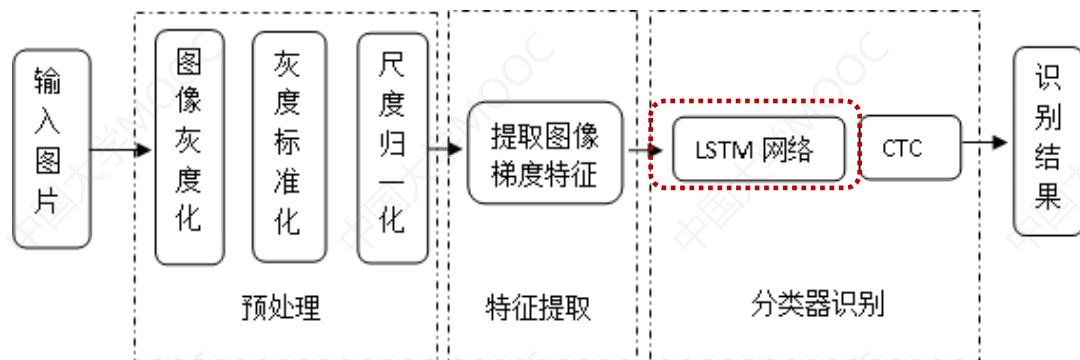
分类器分类识别字符

整合字符成串





# 长短记忆神经网络 (LSTM)





# 长短记忆神经网络 (LSTM)

- ✓ 11280张训练集，2820张测试集

85375 0535 47550 42302

650125 0259 6955789

359992 9258836201724

5394 6369 8758 6246

4397 6674 7545 3719

2965 8397 3434 1356

9 2 44 4 8 5 6 7 5 9 3 4 5 1 5

5 3 3 2 1 4 3 9 6 3 2 7 9 0 0 0 2 8

64 1 2 8 8 1 4 2 1 0 6 5 2 0 8

0144 9276 6549 3420

3 2 6 5 6 8 8 1 7 6 0 7 4 7 9 4 4 9 6

50 8 6 7 8 9 0 8 4 3 7 0 2 9 1



# 长短记忆神经网络 (LSTM)

```
C:\Windows\system32\cmd.exe
F:\OpjFile_Liuli\Bank_Card_CLSTM_1.0 -TEST\HoG_NoGuas_50_50_1e-4_1000Eco_5000Test>CLSTM "2" ".\model\HoG_NoGuas_Ping_RealPic_re45500_50_50_1e-4_4.12_training_55000" "HoG_NoGuas_Ping_RealPic_re100500_50_50_1e-4_4.15" "F:\银行卡\2.切片图(供实验)\2.银行卡切图(算法检测切图)\Ping" "F:\银行卡\2.切片图(供实验)\3.切片真值文档\PinMian_Label.txt" "F:\银行卡\2.切片图(供实验)\2.银行卡切图(算法检测切图)\Ping" "F:\银行卡\2.切片图(供实验)\3.切片真值文档\PinMian_Label.txt"
.stacked: 0.0001 0.9 in 0 40 out 0 11
.stacked.parallel: 0.0001 0.9 in 0 40 out 0 50
.stacked.parallel.lstm: 0.0001 0.9 in 0 40 out 0 25
.stacked.parallel.reversed: 0.0001 0.9 in 0 40 out 0 25
.stacked.parallel.reversed.lstm: 0.0001 0.9 in 0 40 out 0 25
.stacked.parallel: 0.0001 0.9 in 0 50 out 0 50
.stacked.parallel.lstm: 0.0001 0.9 in 0 50 out 0 25
.stacked.parallel.reversed: 0.0001 0.9 in 0 50 out 0 25
.stacked.parallel.reversed.lstm: 0.0001 0.9 in 0 50 out 0 25
.stacked.softmax: 0.0001 0.9 in 0 50 out 0 11
training: 0 gt: 4367420110118369725 pred: 2
training: 50 gt: 6227000990729032545 pred: 6227000990729032545
training: 100 gt: 625996888888888888 pred: 625996888888888888
training: 150 gt: 6226220283977402 pred: 6226220283977402
```

```
eva_Diff_GrayNom_Train...
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
*****
trained epoches: 5
trained sample number: 60000
total test time: 157s
testing time per sample: 55.6738ms

total test sample number: 2820
correct test string number: 2737
string level accuracy: 0.970567

total char number: 49729
correct char number: 49640
insert char number: 0
delete char number: 0
replace char number: 0
AR : 0.99821
CR : 0
```

单幅图像识别时间: 56ms

测试集整串识别率: 97.05%



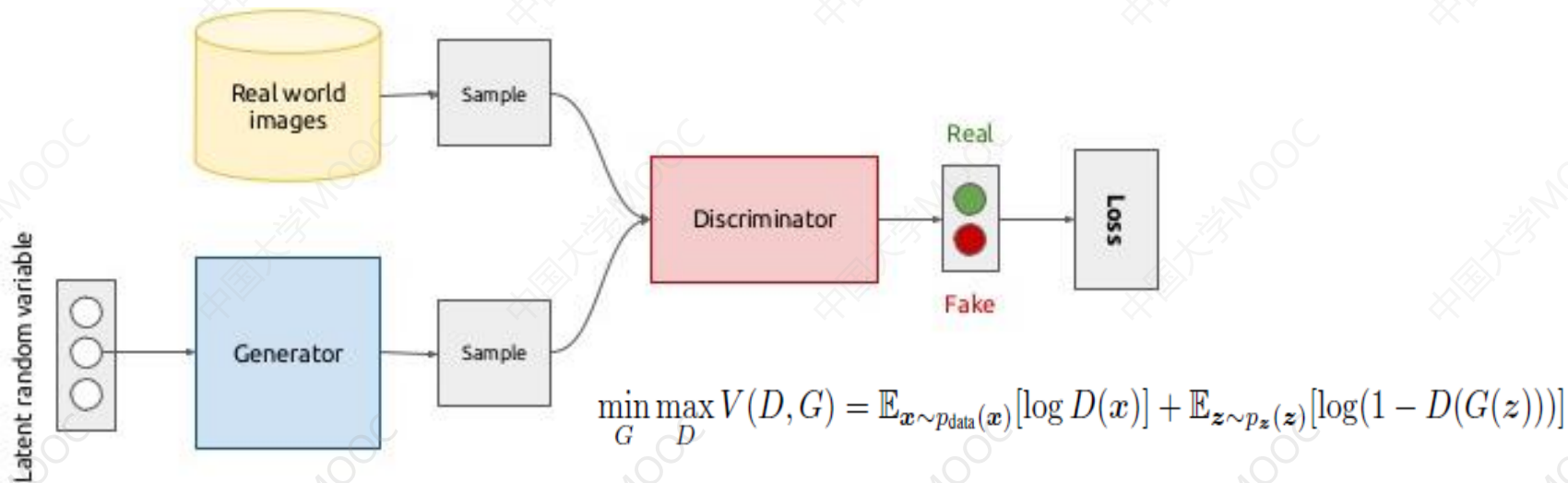
# 深度学习网络

- LeNet, AlexNet, VGG, GoogleNet
- MobileNet, ShuffleNet: 压缩存储量和计算量
- 层数极深网络: Deep Residual Network
- DenseNet
- Graph CNN
- Capsule (CapsNet)
- 生成对抗网络(GAN)



# 生成对抗网络(GAN)

- ◆ 一个生成网络：用于合成数据
- ◆ 一个判别网络：用于判别生成的数据是否真实
- ◆ 同时训练，直到判别网络不能鉴别生成数据的真假





# GAN应用: Image Inpainting



In-painting  
by GAN



Original images

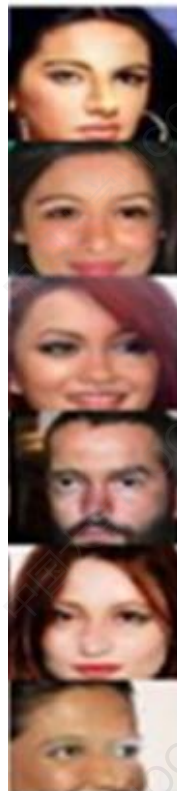
Images with 80%  
random missing pixels



# GAN应用: Image Inpainting



In-painting  
by GAN



Original images

Image with large  
central region missing





# 深度学习工具

- ◆ Caffe
  - 加州伯克利 (Yangqing Jia等人), C++编写, 主要面向视觉
  - <http://caffe.berkeleyvision.org/>
- ◆ TensorFlow
  - Google开发, C++编写
  - <https://www.tensorflow.org/>
- ◆ Theano
  - 2008年诞生于蒙特利尔工学院(Yoshua Bengio), Python编写
  - <http://www.deeplearning.net/software/theano/>
- ◆ Torch, PyTorch
  - Torch: 历史悠久的机器学习综合平台, Lua编写, <http://torch.ch/>
  - PyTorch: In Python,
- ◆ MxNet
  - 民间项目开发, C++编写, 内存使用效率较高, 分布式计算能力强
  - <https://mxnet.apache.org/>





# 存在问题

## ◆ 当前主流方法的不足

### — 学习能力

需要大量标记样本进行监督学习

人类：无监督、小样本、在线自适应

### — 解释能力

主流方法进行统计分类

结构分析、可解释性不足

### — 鲁棒性

对噪声模式的拒识能力

对对抗样本的稳定性

### — 综合信息处理能力

多模态协同

上下文、语义知识



# 存在问题



$x$

$y = \text{"panda"}$   
57.7%  
confidence

$y = \text{"gibbon"}$   
99.3 %  
confidence



$$I = x + 0.007 \times z$$



$z$

Ian Goodfellow, Deep Learning Adversarial Examples – Clarifying Misconceptions, Kdnuggets Home News, 2015



# 存在问题



+ .007 ×



=



$y = \text{"panda"}$  57.7%  
confidence

$y = \text{"gibbon"}$  99.3 %  
confidence

Deep learning is **more vulnerable** to adversarial examples than other kind of machine learning due to the **extreme non-linearity** of deep models



# 未来研究方向

## ◆ 认知计算模型

- 模式识别、学习、记忆的**认知机理**
- 认知计算模型

## ◆ 模型表示

- 模型结构：适合鲁棒在线自适应的需要，**可解释性**
- 生成模型+判别模型：如何更好地**结合**
- **结构和知识**的表示与学习？



# 未来研究方向

## ◆ 学习方法

- 无监督学习
- 监督/无监督混合自适应学习
- 对抗学习
  - 通过对抗提高小样本泛化性
  - 对抗、反对抗
- 迁移学习
- 多任务、多模态协同学习



# 谢 谢

本课程所引用的一些素材为主讲老师多年的教学积累，来源于多种媒体及同事和同行的交流，难以一一注明出处，特此说明并表示感谢！