机器学习

Machine Learning

主讲人：张敏 清华大学长聘副教授

ML 机器学习

MACHINE LEARNING-MIN ZHANG

Unit.10

深度学习基础（II）

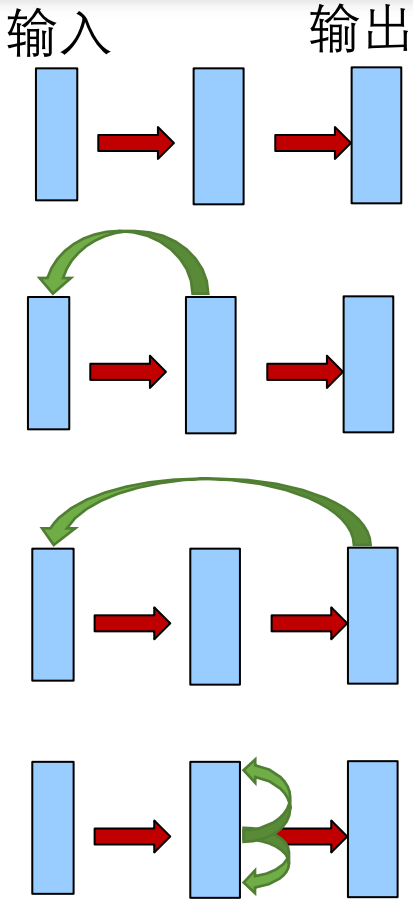
*图片均来自网络或已发表刊物

目录

- 背景
- 多层感知机（MLP）
- 卷积神经网络（CNN）
- 序列神经网络
  - 循环神经网络（RNN）
  - 长短期记忆网络（LSTM）
  - 门控循环单位网络（GRU）
- 应用举例

前向连接

- 前向网络
  - 没有反馈
- 循环网络
  - 层间反馈: 从输出层到输入层，或者从隐层到输入层
  - 层内反馈

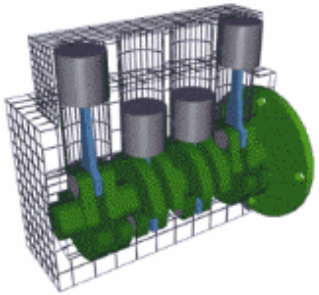
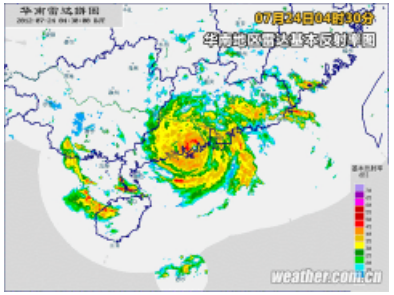


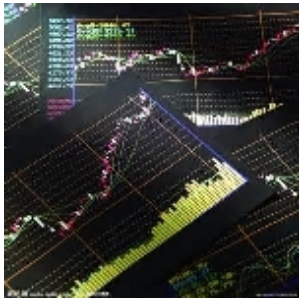
反馈连接可以使神经元的状态（包括输出）随时间变化，因为它们当前输入包含一些神经元上一个时间步的输出

动态系统

- 物理上的动态系统：



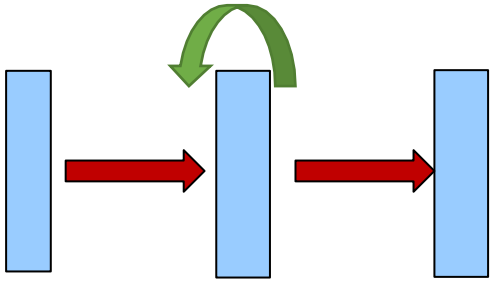






递归神经网络(Recurrent neural network,RNN)

- RNN 是一个动态系统
  - 简单 RNN (The Elman network)

$$h_t = \mathcal{H}\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right)$$
$$y_t = W_{hy}h_t + b_y$$

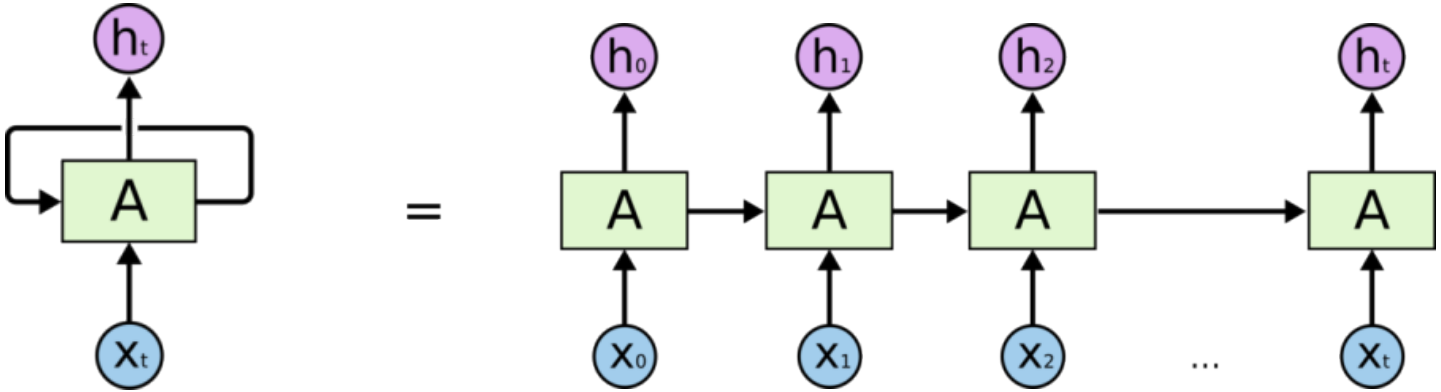


记忆单元

  - 缺点：
    - 梯度消失: 更长的输入序列意味着更多的激活层相乘，所以训练中梯度会趋近 0
    - 因此解决长序列输入问题是很难的

RNN

NN的模块A正在读取某个输入 x_i，并输出一个值 h_i



循环使得信息可以从当前步传递到下一步　可看作是同一个神经网络的多次复制

词之间的依赖: “云飘在天上”

长距离依赖: “我在法国长大 ..... 我能讲一口流利的法语”

↓

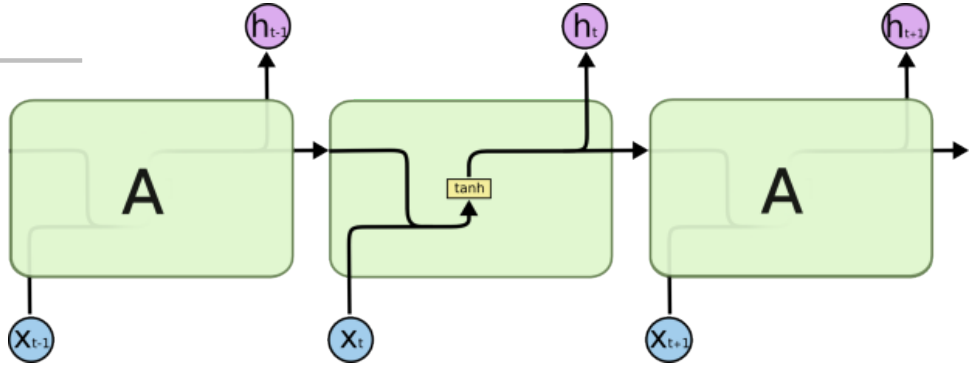
LSTM

目录

- 背景
- 多层感知机（MLP）
- 卷积神经网络（CNN）
- 序列神经网络
  - 循环神经网络（RNN）
  - 长短期记忆网络（LSTM）
  - 门控循环单位网络（GRU）
- 应用举例

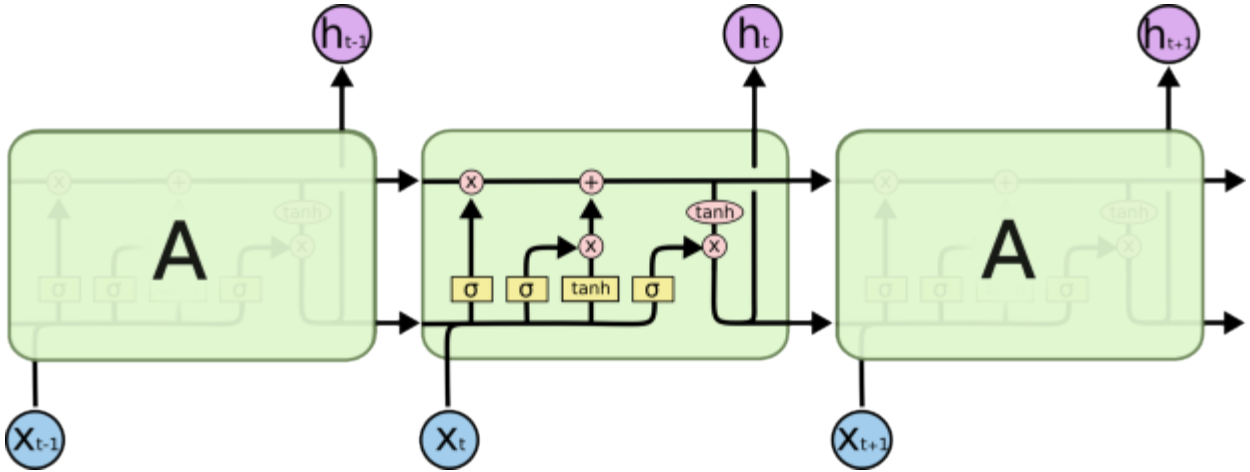
From RNN

标准 RNN 中的重复模块包含单一的层，例如一个tanh层



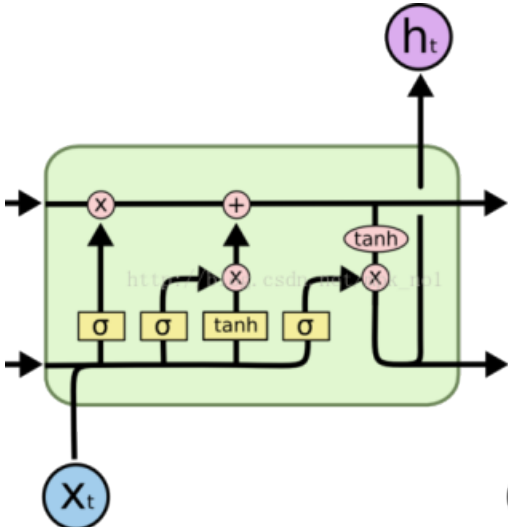
To LSTM

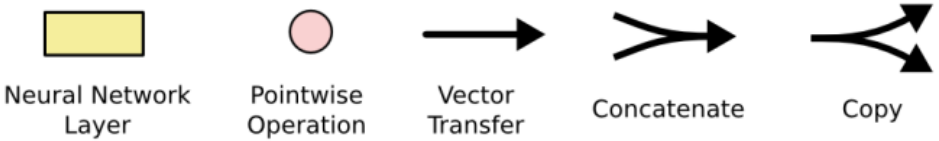
LSTM 同样是这样的结构，但是重复的模块拥有一个不同的结构。



长短期记忆网络(Long short-term memory, LSTM)

- 将长期和短期的信息都考虑进来
- 设计了门（gate）来避免梯度消失
- 三个要点:
  - 什么应该被忘记？
  - 什么应该被记住？
  - 基于当前状态和输入，输出应该是什么？



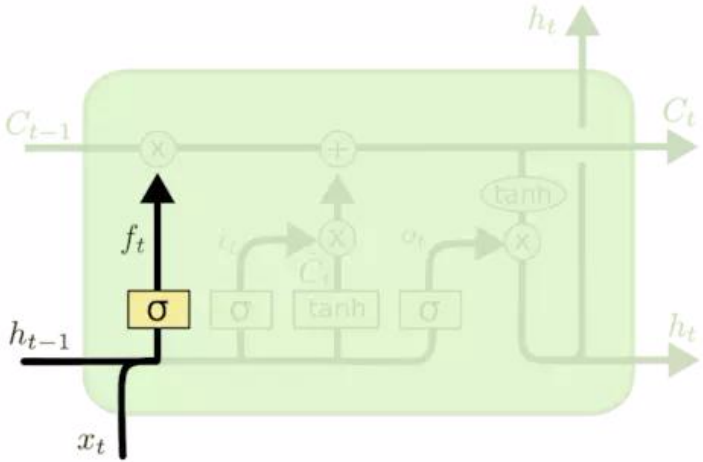


学习到的神经网络层　向量操作　从一个节点的输出到其他节点的输入　向量的连接　内容被复制，然后分发到不同的位置

Long short-term memory

LSTM 通过精心设计的“门”的结构来去除或增加信息到神经元状态的能力

- 第一部分: 什么应该被忘记 “忘记门”



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

门Gate: 一种让信息选择式通过的方法

包括sigmoid神经网络层,一个点积操作

Sigmoid 层输出 0 到 1 之间的数值，描述每个部分有多少量可以通过。

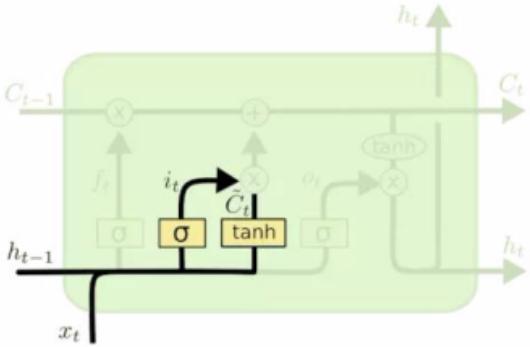
0 代表“不许任何量通过”，1 代表“允许任意量通过”

Long short-term memory

- 第二部分: 什么应该被记住
  - 计算新的输入

sigmoid 层(“输入门层”) 决定我们将要更新什么值

tanh 层创建一个新的候选值向量

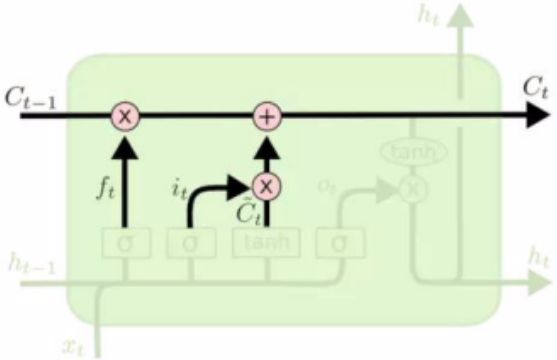


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

  - 更新当前记忆单元

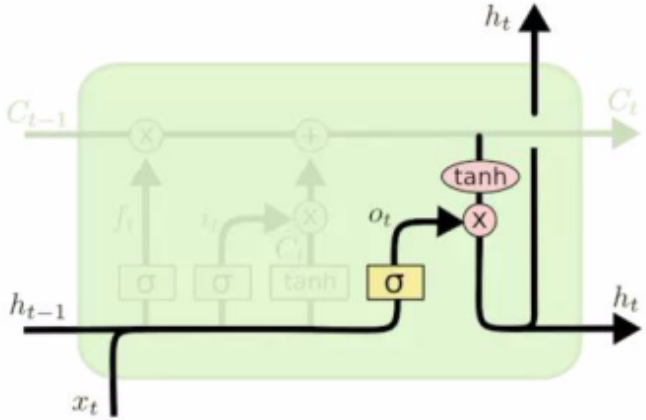
更新旧神经元状态，C_{t-1}更新为C_t



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Long short-term memory

- 第三部分: 计算输出



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

sigmoid 层用来确定神经元状态的哪个部分将被输出

把神经元状态通过 tanh 进行处理（得到一个在-1到 1 之间的值）

并将它与 sigmoid 门的输出相乘

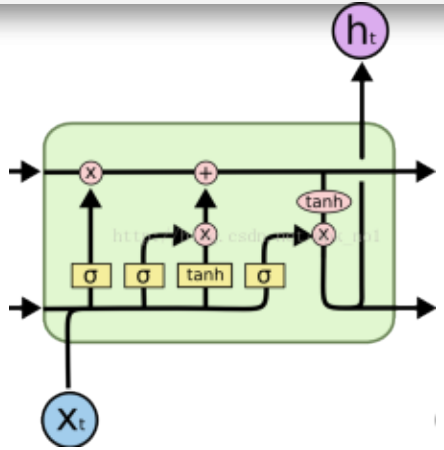
- 缺点: 参数太多，可能导致过拟合

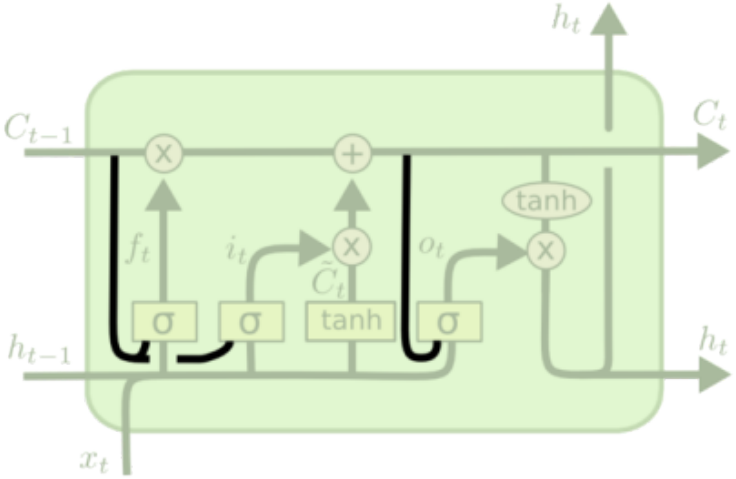
目录

- 背景
- 多层感知机（MLP）
- 卷积神经网络（CNN）
- 序列神经网络
  - 循环神经网络（RNN）
  - 长短期记忆网络（LSTM）
  - 门控循环单位网络（GRU）
- 应用举例

LSTM 的变体（1）

peephole connection [Gers & Schmidhuber (2000)]





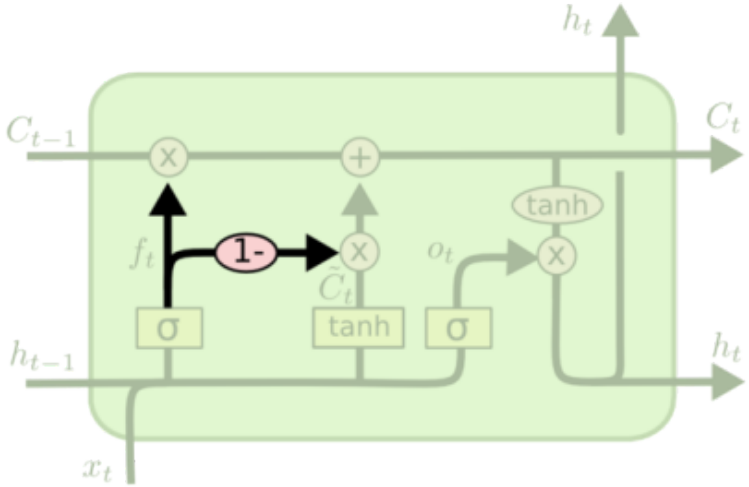
$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] + b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] + b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] + b_o\right)$$

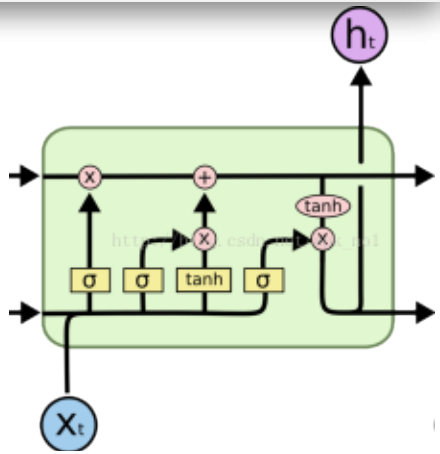
增加了“窥视孔连接”，即我们让门层也会接受神经元状态的输入

很多工作中只加入部分而非所有都加

LSTM 的变体（2）

整合输入门和遗忘门 (coupled input gate and forget gate)





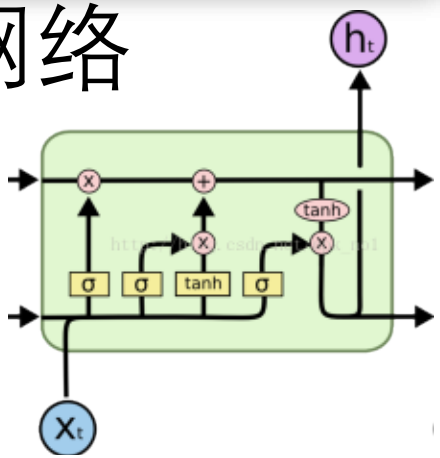
$$C_t = f_t * C_{t-1} + (1 - f_t) * \tilde{C}_t$$

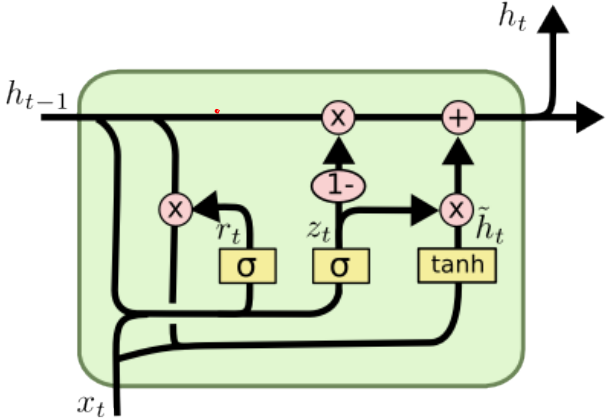
之前是分开确定什么忘记和需要添加什么新的信息，现在一同做出决定

只在要输入到当前位置时忘记；只输入新的值到已经忘记旧的信息的状态

LSTM变体（3）：门控循环单位网络

- Gated recurrent unit, GRU [Cho, et al. (2014)]
- 将输入门和遗忘门结合成一个单一的更新门
- C_t 被设为 h_t (混合了输出状态和隐藏状态，只有一个输出向量)
- 参数更少





$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

目录

- 背景
- 多层感知机（MLP）
- 卷积神经网络（CNN）
- 序列神经网络
  - 循环神经网络（RNN）
  - 长短期记忆网络（LSTM）
  - 门控循环单位网络（GRU）
- 应用举例

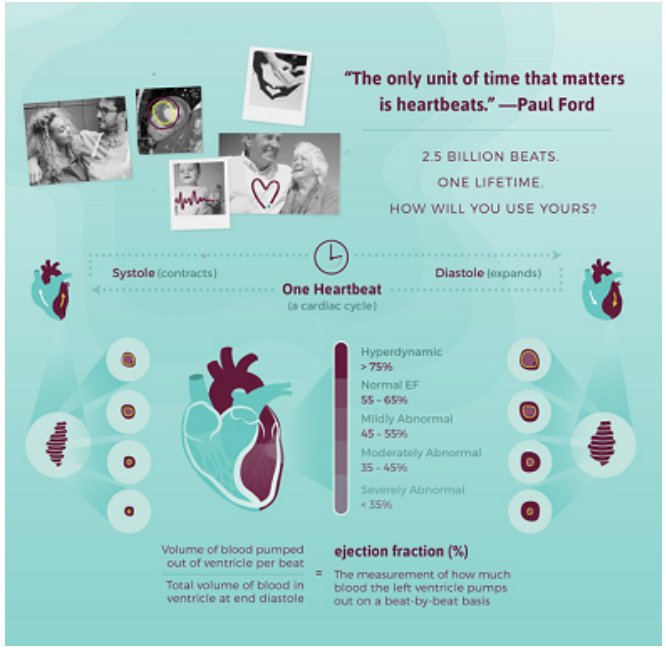
人脸识别

Model	Accuracy (%)
DeepFace (2014)	97.25
DeepID (2014)	97.45
DeepID2 (2014)	99.15
DeepID2+ (2014)	99.47
DeepID3 (2014)	99.53
FaceNet (2015)	99.63

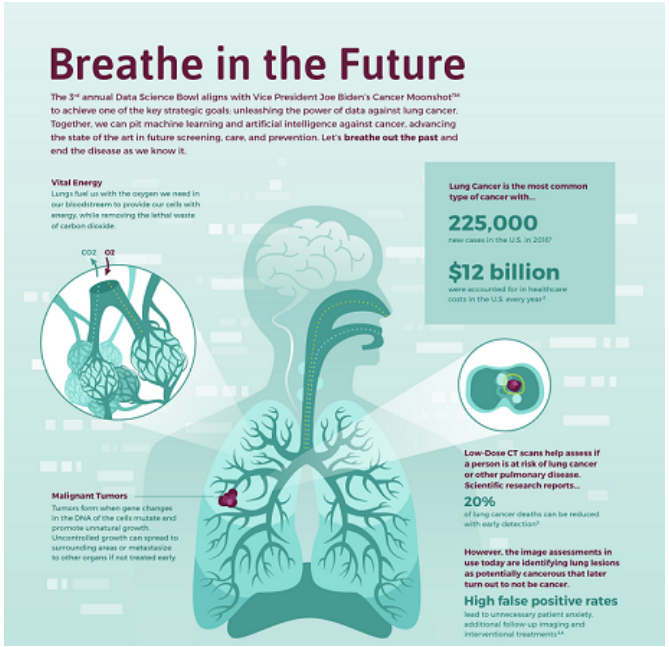
图片风格转换

Created by Prisma

医疗图像分析



Kaggle Data Science Bowl 2016



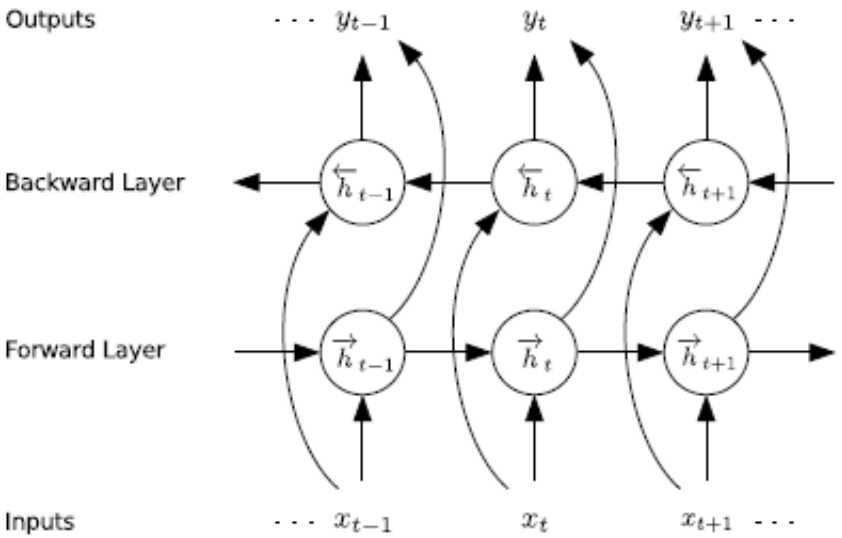
Kaggle Data Science Bowl 2017

语音识别

- 双向 RNN

$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right)$$
$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right)$$
$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

- 结合 BRNNs 和 LSTM 得到双向 LSTM
- 每个时间步 t 在输出层用 Softmax 函数 (大小为 K+1)
  - K 个音素加上一个不输出的单元



Graves et al., 2013

图片摘要

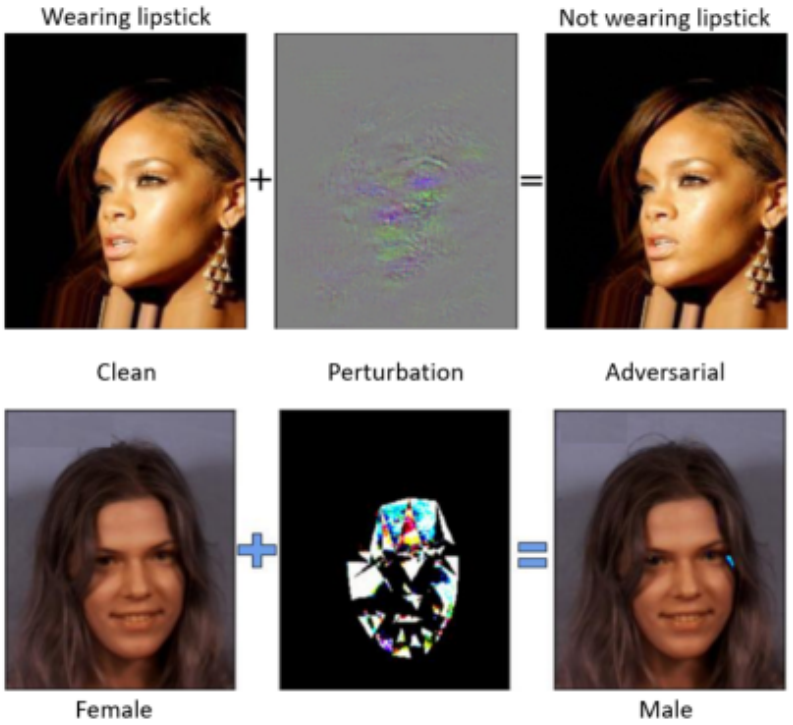
A boy is riding a bike besides a lake.

Two kids are making pizza.

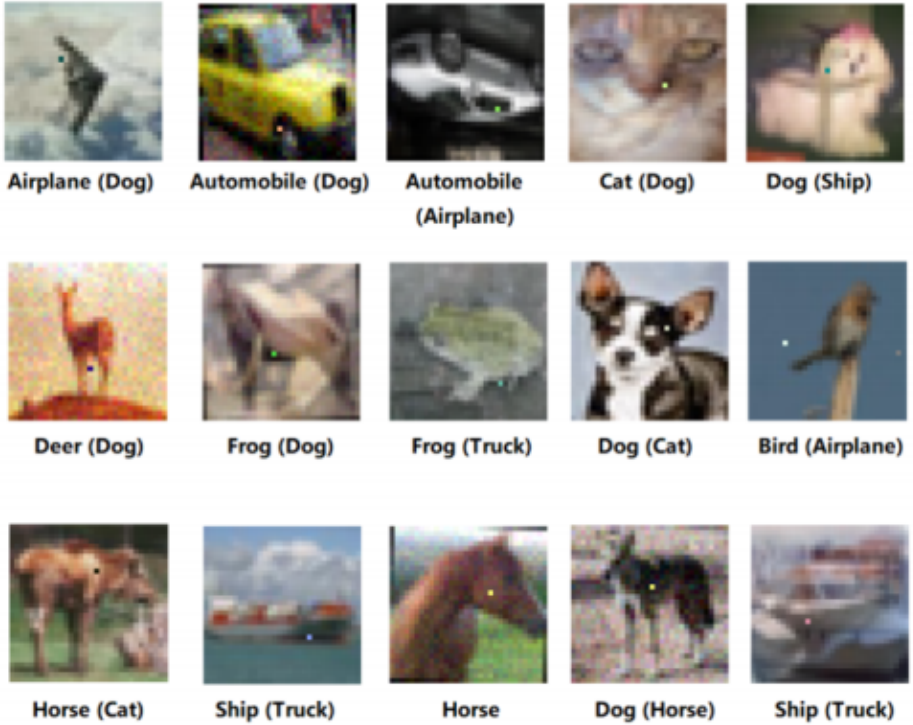
总结

- 深度学习在很多实际问题上都取得了不错的结果
- 可能是处理大量数据的一个较好选择
- 模型太大可能是个问题
  - 并行计算
- 理论基础欠缺
- 缺少可解释性
- 对恶意攻击不鲁棒

深度神经网络攻击举例



1 pixel adversarial attack



https://arxiv.org/abs/1710.08864

在线资源

- 网站: http://deeplearning.net/
  - A reading list
  - Software
  - Datasets
  - Tutorials and demos
- 编程工具
  - Theano @ University of Montreal
  - Caffe @ UC Berkley
  - Tensorflow by Google
  - Torch by Facebook
  - Deeplearning4j

深度学习基础（总结）

- 背景
- 多层感知机（MLP）
- 卷积神经网络（CNN）
- 序列神经网络
  - 循环神经网络（RNN）
  - 长短期记忆网络（LSTM）
  - 门控循环单位网络（GRU）
- 应用举例
- 总结及在线资源