

Contents

1 资料

2 结构

- 2.1 历史与现状 1
 - 2.1.1 历史 1
 - 2.1.2 现状 1
- 2.2 监督机器学习结构 1
- 2.3 线性回归/感知机 1
 - 2.3.1 介绍与分类原理 1
 - 2.3.2 学习与随机梯度下降 1
 - 2.3.3 特征选择问题 (多项式) 1
 - 2.3.4 正则/Overfitting 1
 - 2.3.5 感知机的局限 (异或问题) 2
- 2.4 多层感知机/ANN 2
 - 2.4.1 网络结构 (linear model + activity function) 2
 - 2.4.2 激活函数 (Sigmoid/ Tanh / ReLU etc.) 2
 - 2.4.3 前向算法 2
 - 2.4.4 后向算法 2
- 2.5 无监督学习 2
 - 2.5.1 Autoencoder 2
- 2.6 卷积网络 2
 - 2.6.1 理解 Sobel 边界检测 2
- 2.7 RNN 2
- 2.8 如何构造与应用 2

3 Refs

1 资料

1. *Deep Learning* PDF on github
2. CS231n Convolutional Neural Networks for Visual Recognition
3. Neural Networks for Machine Learning
4. UFLDL Tutorial

2 结构

2.1 历史与现状

2.1.1 历史

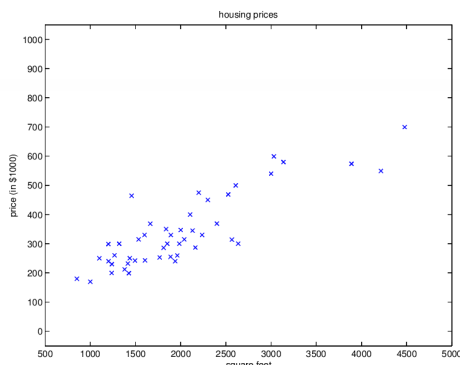
从 1940s 到 2014: <http://people.idsia.ch/~juergen/DeepLearning15May2014.pdf>

2.1.2 现状

1. Andrew Ng, Highway
2. NLP
3. Imagenet
- 4.

2.2 监督机器学习结构

假设我们获得了一部分的房价与房屋面积数据, 如



下:

对应的一条数据的格式是 $\{x, y\}$, 其中 x 为面积, y 为房价. 这些数据 (一部分) 集合 $\{(x_i, y_i); i = 1, \dots, m\}$ 可以叫做训练集 (training set)

那么, 监督机器学习可以如下理解:

利用已知的训练集, 学习一个函数 $h: X \rightarrow Y$, $h(x)$ 的结果对 y 的预测达到我们预设的”足够好”的标准.

如果我们需要根据房屋面积预测房价, 那么我们需要学习一个函数 $h(x)$, 输入 x 为面积, 输出 y 为房价.

2.3 线性回归/感知机

2.3.1 介绍与分类原理

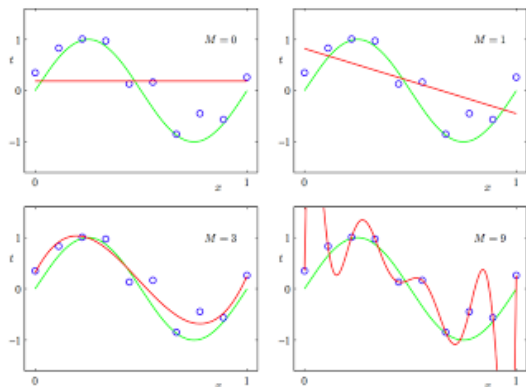
线性回归的形式: $y = W * x + b$ 其中, $W = w_0, w_1, \dots, w_n$, $x = x_0, x_1, \dots, x_n$, 展开表达式为: $y = \sum_0^n w_i * x_i + b$. 模型的参数 W 需要通过学习来获得.

2.3.2 学习与随机梯度下降

1. Cost Function

2.3.3 特征选择问题 (多项式)

2.3.4 正则/Overfitting



5. <https://colah.github.io/posts/2014-07-Conv-Nets-Modular/> Github

6. <http://cs229.stanford.edu/materials.html>

2.3.5 感知机的局限 (异或问题)

2.4 多层感知机/ANN

2.4.1 网络结构 (linear model + activity function)

2.4.2 激活函数 (Sigmoid/ Tanh / ReLU etc.)

2.4.3 前向算法

2.4.4 后向算法

2.5 无监督学习

2.5.1 Autoencoder

2.6 卷积网络

2.6.1 理解 Sobel 边界检测

2.7 RNN

1. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

2.8 如何构造与应用

3 Refs

1. How to Layout and Manage Your Machine Learning Project
2. Machine learning in 10 pictures
3. 深度学习-wiki
4. Ufldl tutorial in Python