

# 深度学习

赵燕

## 1 深度学习概论

### 1.1 什么是神经网络

深度学习就是训练神经网络。

Example 1 – single neural network

Given data about the size of houses on the real estate market and you want to fit a function that will predict their price. It is a linear regression problem because the price as a function of size is a continuous output.

We know the prices can never be negative so we are creating a function called Rectified Linear Unit (ReLU) which starts at zero.

修正线性单元，修正的意思：取不小于0的值

### Housing Price Prediction

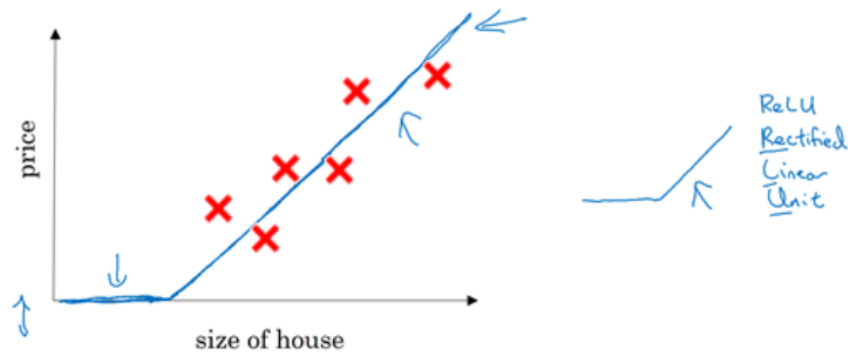


图 1: 房价预测例1

注释:

- The input is the size of the house ( $x$ )
- The output is the price ( $y$ )
- The “neuron” implements the function ReLU (blue line)

Example 2 – Multiple neural network

The price of a house can be affected by other features such as size, number of bedrooms, zip code and wealth. The role of the neural network is to predict the price and it will automatically generate the hidden units. We only need to give the inputs  $x$  and the output  $y$ .

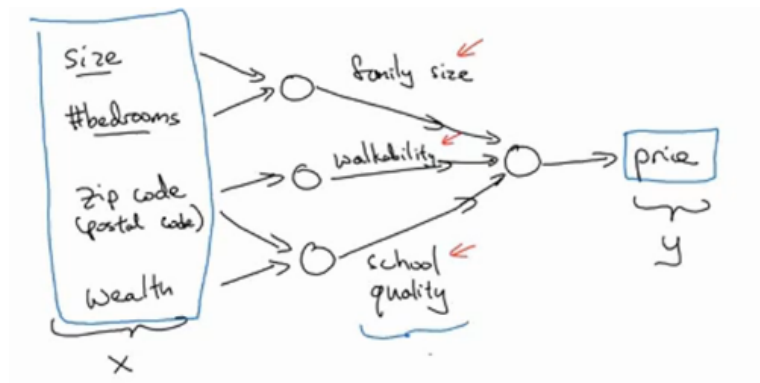


图 2: 房价预测例2

## Housing Price Prediction

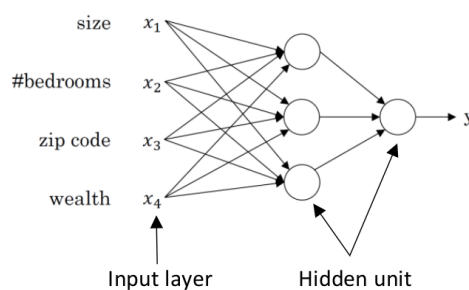


图 3: 房价预测神经网络

已知输入的特征，预测对应的price，中间的隐藏单元：每个的输入都同时来自4个特征。

给到足够的(x,y)，神经网络非常擅长于计算从x到y的精准映射函数。

### 1.2 用神经网络进行监督学习

In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

Supervised learning problems are categorized into "regression" and "classification" problems. In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function. In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.

Here are some examples of supervised learning

## Supervised Learning

Input(x)	Output (y)	Application
Home features	Price	Real Estate
Ad, user info	Click on ad? (0/1)	Online Advertising
Image	Object (1,...,1000)	Photo tagging
Audio	Text transcript	Speech recognition
English	Chinese	Machine translation
Image, Radar info	Position of other cars	Autonomous driving

图 4: 神经网络实例

There are different types of neural network, for example Convolution Neural Network (CNN) used of ten for image application and Recurrent Neural Network (RNN) used for one-dimensional sequence data such as translating English to Chinese or a temporal component such as text transcript. As for the autonomous driving, it is a hybrid neural network architecture.

## Neural Network examples

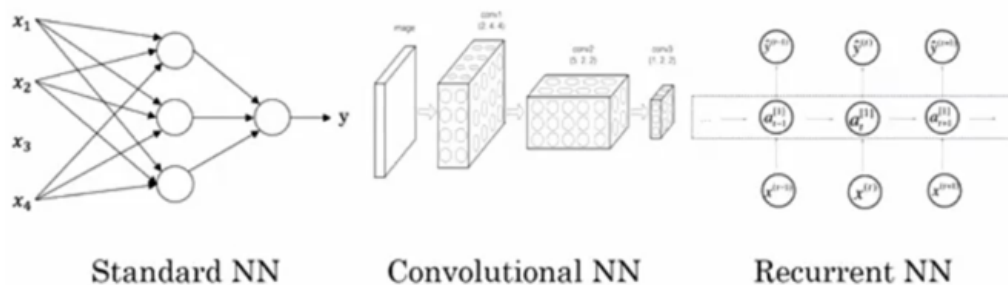


图 5: 神经网络种类

注释:

- Standard NN: 标准神经网络
- CNN: 卷积神经网络常用于图片处理;
- RNN: 循环神经网络适合处理一维数据序列, 其中包含时间成分;

Structured vs unstructured data

Structured data refers to things that has a defined meaning such as price, age whereas unstructured data refers to thing like pixel, raw audio, text.

# Supervised Learning

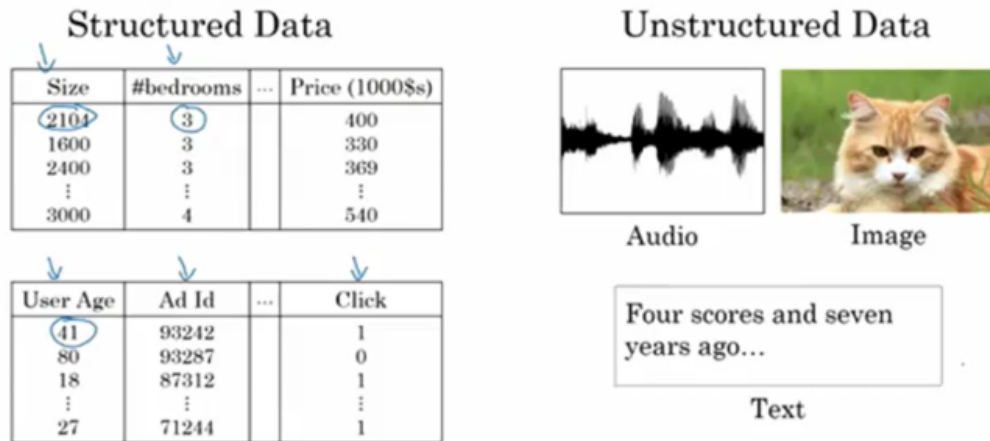


图 6: Structured vs unstructured data

相比于结构化，非结构化计算机理解起来更难，应用越来越广泛，包括语音识别，文字识别，图片识别，神经网络改变了监督学习。

## 1.3 为什么深度学习会兴起

Deep learning is taking off due to a large amount of data available through the digitization of the society, faster computation and innovation in the development of neural network algorithm.

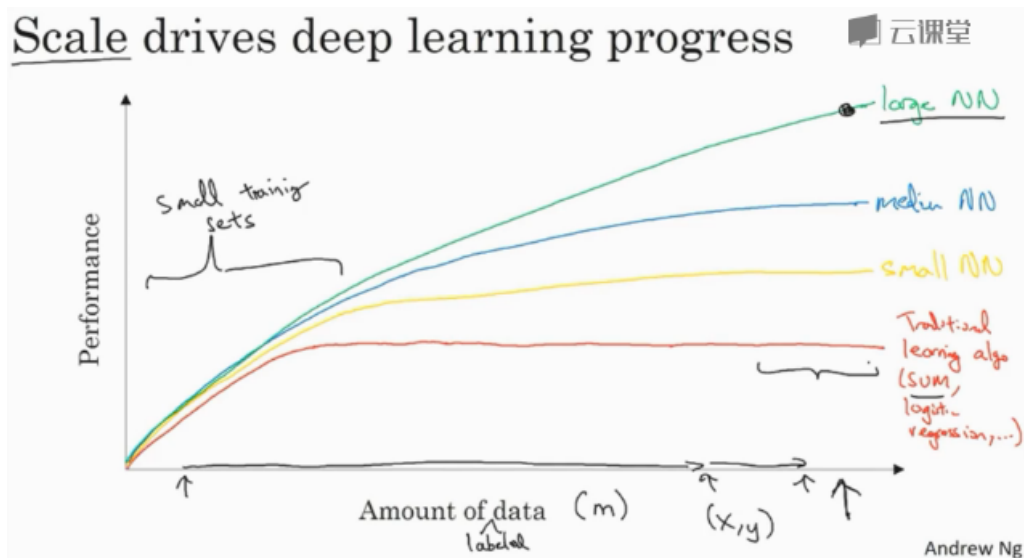


图 7: 各种算法的相对排名

Two things have to be considered to get to the high level of performance:

1. Being able to train a big enough neural network
2. Huge amount of labeled data

注释:

- x轴: 大量带有标签的数据
- y轴: Performance, 机器学习性能

在深度学习崛起的初期, 是数据和计算能力规模的发展, 训练一个特别大的神经网络的能力, 无论是在CPU还是GPU上面, 是这些发展才让我们取得了巨大的进步, 但是这几年我们也见证了算法方面的极大创新, 许多算法的创新都是为了让神经网络运行的更快。

一个重大的突破是由sigmoid函数转换为了ReLU函数, 但是使用sigmoid函数, 机器学习的问题是, 在这个区域使用sigmoid函数的斜率, 梯度会接近于0, 所以学习会变得非常缓慢, 因为用梯度下降算法时, 梯度接近于0时, 参数会变化得很慢, 学习也会变得很慢, 而通过改变激活函数, 神经网络用ReLU这个函数修正线性单元ReLU, 它的梯度对于所有为正值的输入, 输出都是1, 因此梯度不会逐渐趋向0, 而这里的梯度, 这条线的斜率, 在这左边为0。

我们发现, 只需将sigmoid函数转化为ReLU函数, 便能使得梯度下降法运行的很快。

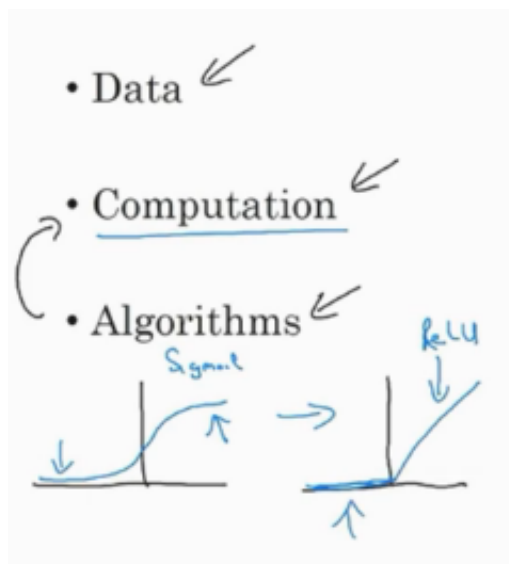


图 8: 深度学习的发展

The process of training a neural network is iterative.

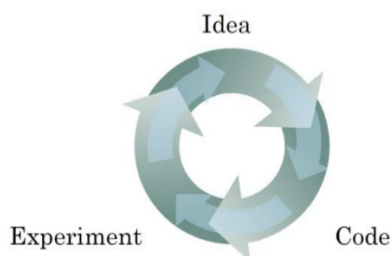


图 9: 训练神经网络

It could take a good amount of time to train a neural network, which affects your productivity. Faster computation helps to iterate and improve new algorithm.

## 2 神经网络基础

### 2.1 二分分类

### 2.2 logistic回归

### 2.3 logistic回归损失函数

### 2.4 梯度下降法

### 2.5 导数

### 2.6 更多导数的例子

### 2.7 计算图

### 2.8 计算图的导数计算

### 2.9 logistic回归的梯度下降法

### 2.10 m个样本的梯度下降

### 2.11 向量化

### 2.12 向量化的更多例子

### 2.13 向量化logistic回归

### 2.14 向量化logistic回归的梯度输出

### 2.15 Python中的广播

### 2.16 关于Python/numpy向量的说明

### 2.17 Jupyter/Ipython笔记本的快速指南

### 2.18 logistic损失函数的解释

2.Python 允许你将程序分割为不同的模块，以便在其他的 Python 程序中重用。Python 内置提供了大量的标准模块，你可以将其用作程序的基础，或者作为学习 Python 编程的示例。这些模块提供了诸如文件 I/O、系统调用、Socket 支持，甚至类似 Tk 的用户图形界面（GUI）工具包接口。

## 3 浅层神经网络

### 3.1 神经网络概览

### 3.2 神经网络表示

### 3.3 计算神经网络的输出

### 3.4 多个例子中的向量化

### 3.5 向量化实现的解释

### 3.6 激活函数

### 3.7 为什么需要非线性激活函数

### 3.8 激活函数的倒数

### 3.9 神经网络的梯度下降法

### 3.10 直观理解反向传播

### 3.11 随机初始化

## 4 深层神经网络

### 4.1 深层神经网络

1. Python 易于使用，是一门完整的编程语言；与 Shell 脚本或批处理文件相比，它为编写大型程序提供了更多的结构和支持。另一方面，Python 提供了比 C 更多的错误检查，并且作为一门高级语言，它内置支持高级的数据结构类型，例如：灵活的数组和字典。因其更多的通用数据类型，Python 比 Awk 甚至 Perl 都适用于更多问题领域，至少大多数事情在 Python 中与其他语言同样简单。

### 4.2 深层神经网络中的前向传播

### 4.3 核对矩阵的维度

### 4.4 为什么使用深层表示

### 4.5 搭建深层神经网络块

### 4.6 前向和反向传播

### 4.7 参数VS超参数

### 4.8 这和大脑有什么关系