深度学习

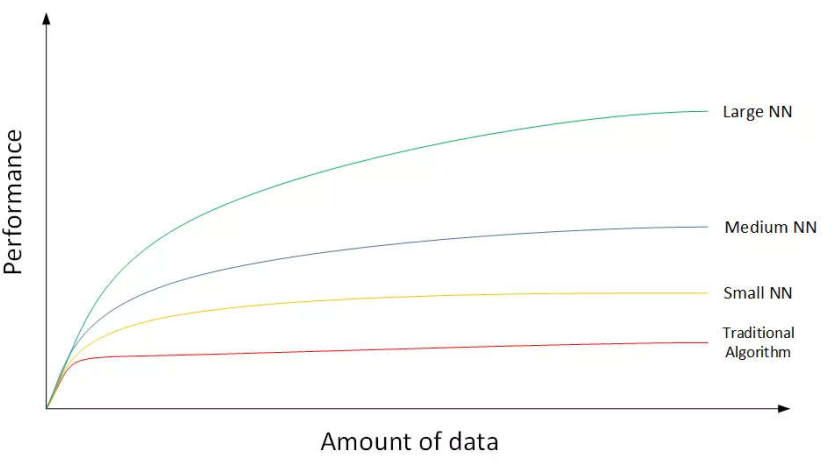
1. 深度学习基础
2. **为什么如此之强？**

* 不用再进行特征提取
* 处理线性不可分

1. **深度学习得以实现的依据**

* DATA，数据量的几何级数增长。
* Computation，GPU出现，运算能力大大提升。
* Algorithm，算法的创新与改进让深度学习的性能与速度大幅提升。如将激活函数由sigmod函数改成relu函数，大幅提高收敛速度。

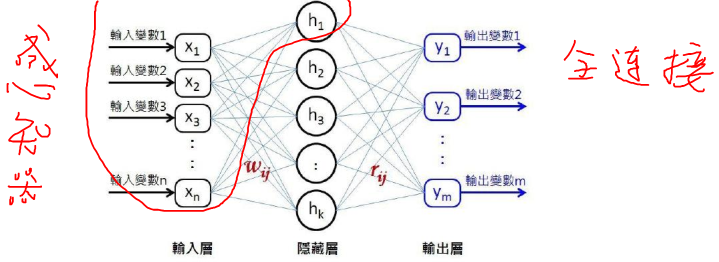
1. **不同学习模型性能表现：**



1. **深度学习概念：**

* 深度学习的概念源于人工神经网络的研究。
* 含有多隐层的感知器就是一种深度学习结构。
* 深度学习通过组合低层特征形成更加抽象的高层表示属性类别或特征。

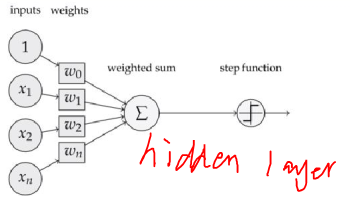
1. **感知器：**共有k个感知器：



* 第一个感知器：

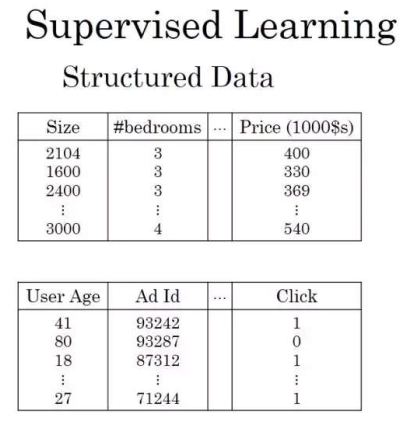


* 单感知器神经网络：

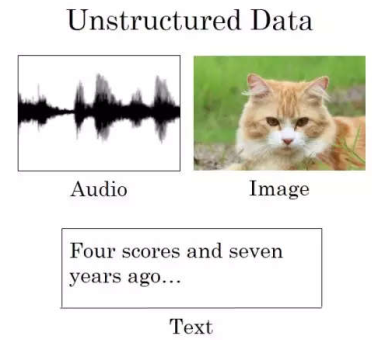


1. **数据类型**

* **structured data：有实际意义的量化数据**

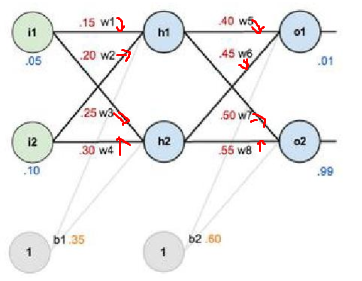


**unstructured data：比较抽象的数据，人类容易识别，计算机难以识别，由于深度学习的兴起，非结构化数据也变得越来越容易处理。**



* 1. 基本组件
     1. 感知器（基本神经单元）
     2. 激活函数（去线性化，构建更复杂系统）
     3. 正则化（防止过拟合）
     4. Dropout（防止过拟合）
     5. 卷积层（降低维度）
     6. 池化层
     7. 全连接层
     8. Softmax回归（实现分类）
  2.  前向传播+反向传播的推导

1. **初始化网络结构：**



**前向传播：(h = x1\*w1 + ... + xn \* wn + b E=(1/n)\*（target-output）^2)，激活函数为sigmod=1/(1+e^-(wx+b))，n为样本数，此处为2**

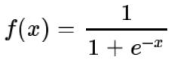
**h1 = 0.05\*0.15+0.10\*0.20+0.35\*1=0.3775 o1=0.3775\*0.40+0.3925\*0.45+0.60\*1=0.927625 e1=0.5\*(0.01-0.927625)**

**h2 = 0.05\*0.25+0.10\*0.30+0.35\*1=0.3925 o2=0.3775\*0.50+0.3925\*0.55+0.60\*1=1.004625**

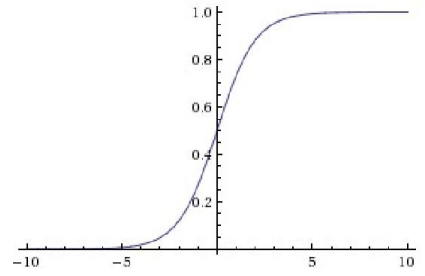
**反向传播：（链式求导）**

* 1. 激活函数
* 激活函数的作用是能够给神经网络加入一些**非线性因素**
* 使得神经网络可以更好地**解决较为复杂的问题**
  + 1. Sigmod （逻辑函数）

Sigmoid函数曾被广泛地应用。

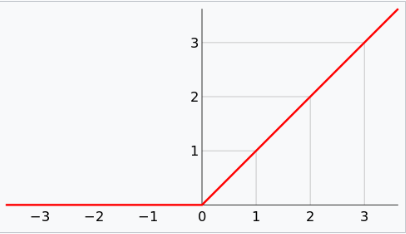


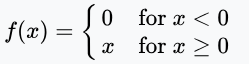
* **去线性化**
* Sigmoid函数的输出映射在(0,1)之间，单调连续，输出范围有限，**优化稳定，可以用作输出层。**
* 求导容易。



* + 1. ReLU（线性整流函数）

线性整流函数（Rectified Linear Unit, ReLU）,又称修正线性单元, 是一种人工神经网络中常用的激活函数（activation function），通常指代以斜坡函数及其变种为代表的非线性函数。





ReLU是最近几年非常受欢迎的激活函数。被定义为：

* 相比起Sigmoid， **ReLU在SGD中能够快速收敛**。
* Sigmoid涉及了很多运算大的操作（比如指数）， **ReLU可以更加简单**的实现。
* **有效缓解了梯度消失的问题。**
  + 1. Tanh（双曲正切函数）

graphic graphic （-1， 1）

* 1. 深度学习防止过拟合

在深度学习模型中，由于其模型参数非常多非常容易产生过拟合，比较常用的技术包括：

1. **参数添加约束，例如L1、 L2范数等**

L1正则： 参数约束添加L1范数惩罚项 graphic

L2正则： 参数约束添加L2范数惩罚项 graphic

1. **训练集合扩充，例如添加噪声、数据变换等**

防止过拟合最有效的方法是**增加训练集合**，训练集合越大过拟合概率越小。数据集合扩充是一个省时有效的方法，但是在不同领域方法不太通用

* 在目标识别领域常用的方法是将图片进行旋转、缩放等
* 语音识别中对输入数据添加随机噪声
* NLP中常用思路是进行近义词替换
* 噪声注入，可以对输入添加噪声，也可以对隐藏层或者输出层添加噪声(Noise)

1. **Dropout：**

**一类通用的正则化方法**

* Dropout在训练过程中**随机的丢弃一部分输入**，
* 此时丢弃部分对应的参数不会更新

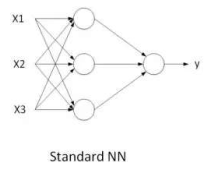
**相当于Dropout是一个集成方法**

* 将所有子网络结果进行合并，通过随机丢弃
* 输入可以得到各种子网络

**通过不同的输入屏蔽相当于学习到所有子网络结构。**

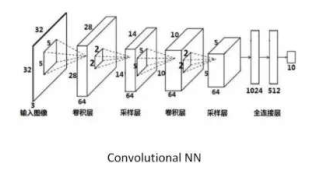
1. 因此前向传播过程变成如下形式：相当于每层输入多了一个屏蔽向量μ
2. 来控制该层有哪些输入会被屏蔽掉。

1. 神经网络（NN）



* 1. 人工神经网络概述
  2. 人工神经网络发展历程
  3. 深度信念网络
     1. BM（玻尔兹曼机）
     2. RBM（受限玻尔兹曼机）
     3. DBN（深度信念网络）
  4. 堆积的自动编码器
  5. CNN（卷积神经网络）
     1. 演进
  6. 卷积自动编码器

1. 卷积神经网络（CNN ）



CNN一般处理图像问题，

* 1. LeNet（1998 YangLecun）

1998, 杨力坤，

1. 过拟合

2. 计算量大

* 1. AlexNet

辛顿及其团队，2012

1. 数据增强

2. ReLU

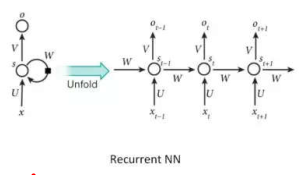
3. Dropout

4. LRN

5. 多GPU

* 1. VGG
  2. NIN
  3. GoogleNet
  4. ResNet
  5. CNN 与 GPU
     1. 副主题

1. （循环神经网络（RNN）

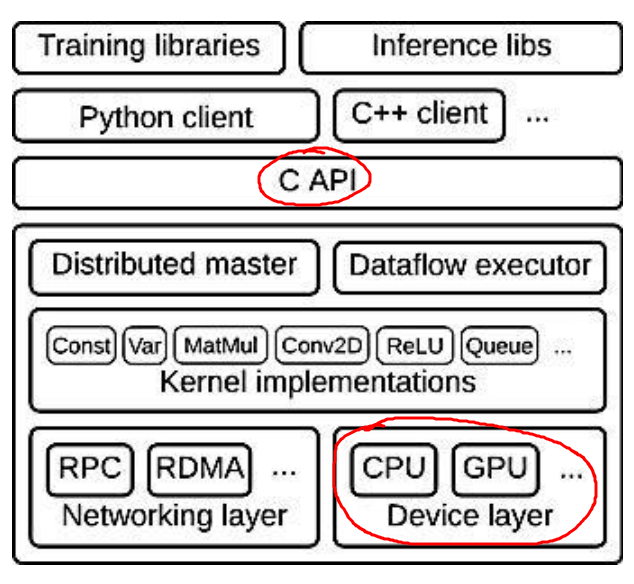


RNN一般处理语音信号

* 1. BiRNN
  2. LSTM 与 GRU
  3. Seq2seq

1. 深度学习框架
   1. tensorflow
      1. 原理架构

基础架构



TensorFlow特点

高度的灵活性

可移植性

综合了科研及产品

自动计算梯度导数

性能最优化

多语言支持

TensorFlow支持C++、Python、Java、Go、JavaScript

* + 1. 计算模型和计算图

计算图

nodes：数据的数学计算

edges：数据的流动方向

查看计算图

Tensorboard

计算图的本质

OpDef

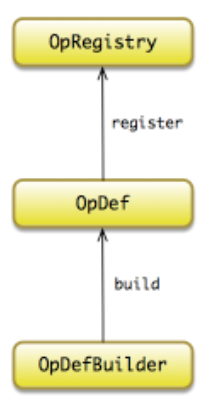
NodeDef

符号编程

**（1）注册OP**

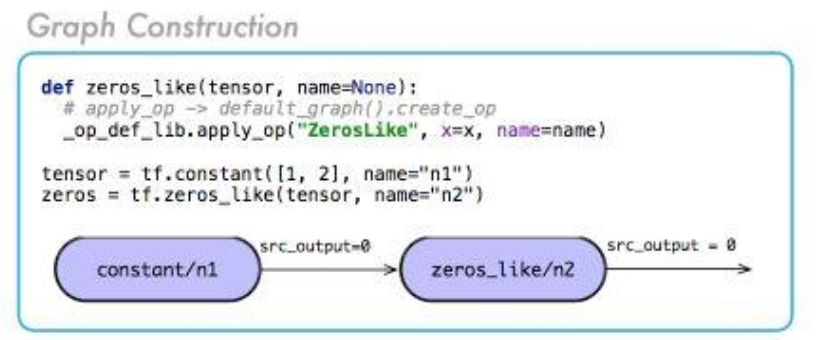
在系统初始化时，系统实现对所有**OP进行扫描注册**，并保存于OpRegistry之中。

在系统实现中， OP的元数据使用Protobuf格式的OpDef描述，实现前端与后端的数据交换，及其领域模型的统一。



**（ 2）构造图**

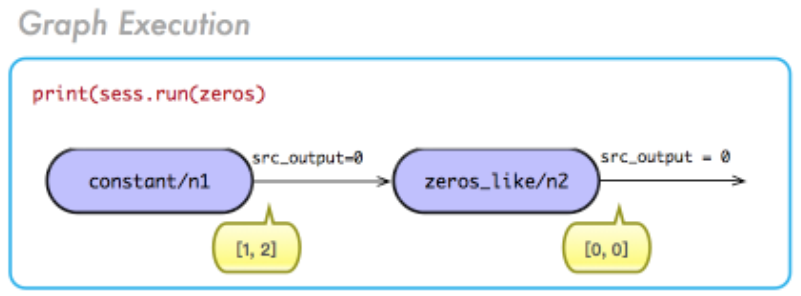
在前端，用户使用OP构造器实现OP的构造，并将OP注册到计算图中。在计算图构造期间， OP的输入/输出的类型， Shape得以确定， OP属性值也得以确定。



**（ 3）执行图**

在计算图执行期间，输入由上游OP流入得以确定，根据特定设备类型，输入输出类型，多态选择合适的Kernel实现，并启动Kernel的计算过程。

例如，如果zeros\_like上游输入为[1, 2]， 经过zeros\_like的OP运算，输出为[0, 0]。



图构造期

图执行期

* + 1. 会话、张量、变量与图

张量、变量

神经网络变量定义方式如下：

**import tensorflow as tf**

**import numpy as np**

X1 = tf.placeholder(tf.float32,tf.zeros)

W1 = tf.Variable(tf.zeros([4,4]),

W2 = tf.constant(np.ones(4,4))

张量、 变量运算



会话

在用Python描绘神经网络的过程中其实并未直接执行， 我们需要一个会话来启动整个定义的计算图。 同时需要我们对计算图中的变量进行初始化。

sess = tf.Session()

#sess = tf.InteractiveSession()

init = tf.global\_variables\_initializer()

sess.run( init)

使用 TensorFlow, 你必须明白 TensorFlow:

* 使用图 (graph) 来表示计算任务.
* 在被称之为会话（ Session） 的上下文 (context) 中执行图.
* 使用张量（tensor ） 表示数据.
* 通过变量（ variable） 维护状态.

使用 feed 和 fetch 可以为任意的操作(arbitrary operation) 赋值或者从其中获取数据.

变量共享

当训练复杂模型时，可能经常需要共享大量的变量。

使用TensorFlow提供variable scope机制，通过这个机制，可以让我们在构建模型时轻松共享命名变量。

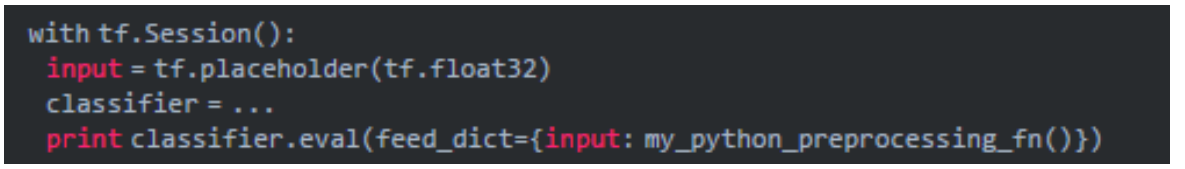
tensorflow中的变量共享是通过 tf.variable\_scope() 和 tf.get\_variable() 来实现的,变量共享主要涉及到两个函数：

* tf.get\_variable([name], [shape], [initializer])
* tf.variable\_scope([scope\_name])
  + 1. TensorFlow读取数据

供给数据(Feeding)

TensorFlow的数据供给机制允许你在TensorFlow运算图中将数据注入到任一张量中。因此，python运算可以把数据直接设置到TensorFlow图中。

通过给run()或者eval()函数输入feed\_dict参数，可以启动计算过程



从文件读取数据

* Csv文件格式。需要使用TextLineReader和decode\_csv 操作
* 固定长度的记录(Fixed Length Record)
* 标准TensorFlow格式(TFRecords)

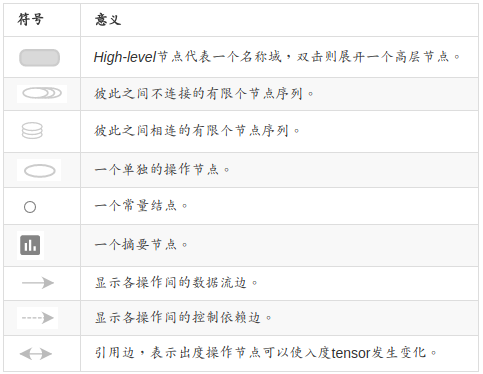
预加载数据

这仅用于可以完全加载到存储器中的小的数据集。有两种方法：

ü -存储在常数中。使用常数更简单一些，但是会使用更多的内存（因为常数会内联的存储在数据流图数据结构中，这个结构体可能会被复制几次）

ü 存储在变量中，初始化后，永远不要改变它的值。

* + 1. TensorFlow实例
    2. Tensorboard使用



副主题

* + 1. TensorFlow回归
    2. TensorFlow逻辑回归
    3. 人工神经网络

FeedForward（前馈神经网络）

前馈神经网络（ Feedforward neural network） 的特点是**不存在循环，** 常见的CNN， 全连接网络均属于前馈神经网络， 其可以完成包括人脸识别、 文字识别在内的一些工作。

循环神经网络(RNN)

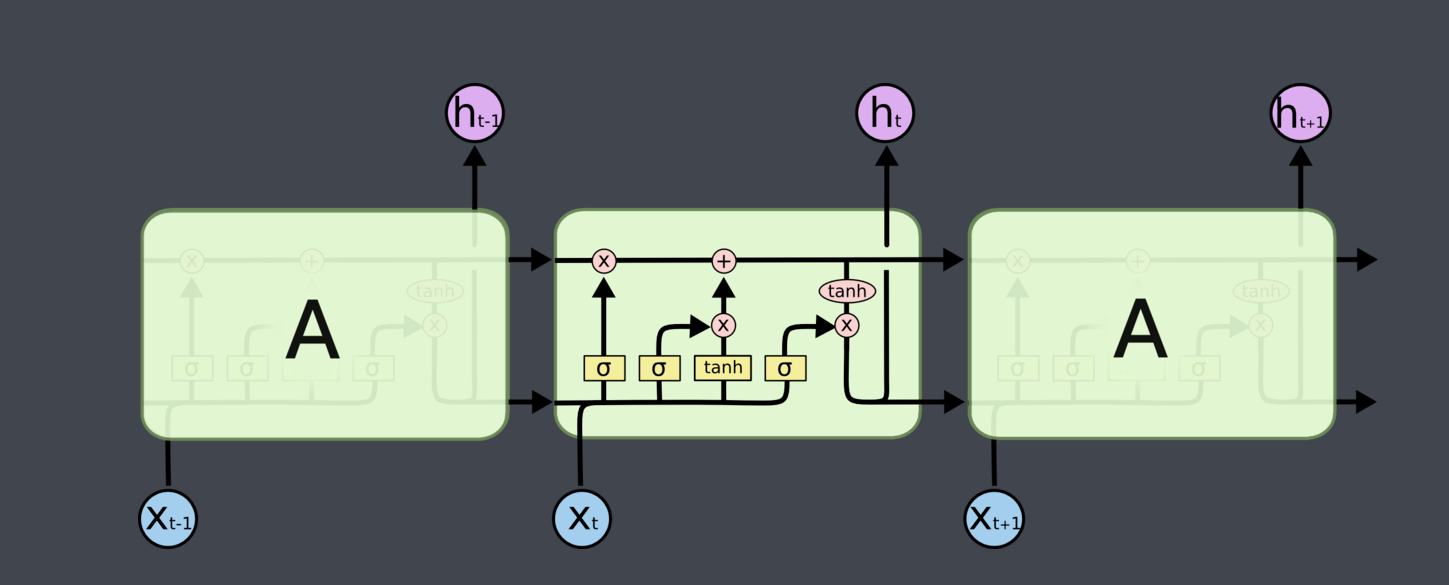
循环神经网络(RNN)在计算中**存在循环**， 常见的LSTM就属于循环神经网络， 循环神经网络常与前馈神经网络相结合， 可以完成包括文本识别、 对话机器人在内的一些工作。

LSTM（Long Short-Term Memory）是**长短期记忆网络**，是一种时间递归神经网络，适合于处理和预测时间序列中间隔和延迟相对较长的重要事件。

LSTM 已经在科技领域有了多种应用。基于 LSTM 的系统可以**学习翻译语言、控制机器人、图像分析、文档摘要、语音识别图像识别、手写识别、控制聊天机器人、预测疾病、点击率和股票、合成音乐**等等任务。

LSTM区别于RNN的地方，主要就在于它在算法中加入了一个判断信息有用与否的“处理器”，这个处理器作用的结构被称为**cell**。

一个cell当中被放置了三扇门，分别叫做**输入门**、**遗忘门**和**输出门**。一个信息进入LSTM的网络当中，可以根据规则来判断是否有用。只有符合算法认证的信息才会留下，不符的信息则通过遗忘门被遗忘。

说起来无非就是**一进二出**的工作原理，却可以在反复运算下解决神经网络中长期存在的大问题。目前已经证明，LSTM是解决长序依赖问题的有效技术，并且这种技术的普适性非常高，导致带来的可能性变化非常多。各研究者根据LSTM纷纷提出了自己的变量版本，这就让LSTM可以处理千变万化的垂直问题。

单层神经网络+ 多层神经网络

深度神经网络

损失函数

二范数（监督学习）

熵（无监督学习）

* + 1. TensorFlow传统神经网络
    2. TensorFlow卷积神经网络
  1. Keras
* 深度学习框架
* 底层可用Tensorflow等
* 简化深度学习构建
* 功能丰富

1. 项企业级目
   1. 人脸识别
      1. 简介及应用场景
      2. 人脸识别任务

**十大关键任务 :**

* **人脸检测**
* **人脸对齐**
* 人脸属性识别
* 人脸提特征
* 人脸对比
* 人脸验证
* **人脸识别**
* 人脸检索
* 人脸聚类
* 人脸活体

**活体检测：**

* 注入应用绕过活体检测
* 视频攻击绕过活体检测
* 三维建模绕过云端检测
* 脸部模具绕过云端检测
* 利用接口保护不当和各种奇葩设计缺陷

 人脸检测

人脸检测是整个人脸识别算法的第一步。

我们将整个人脸检测算法分为3个阶段，分别是：

* 早期算法

* AdaBoost框架

* 深度学习时代

在接下来将分这几部分进行介绍。

**1、早期算法：**

* EigenFace（PCA）
* FisherFace（LDA）

**2、人工特征 + 分类器**

* 纹理特征：HOG、 Cabor、 LBP
* 局部特征：SIFT
* 其他： Haar-like

**3、基于深度学习的**

* MTCNN
* Face-RCNN
* PyramidBox

 人脸对齐

人脸对齐做的是将检测到的脸和 Landmark 通过几何变换，将五官变换到图中相对固定的位置，提供较强的先验。

广泛使用的对齐方法为 **Similarity Transformation**。

 人脸识别

**核心思想：**

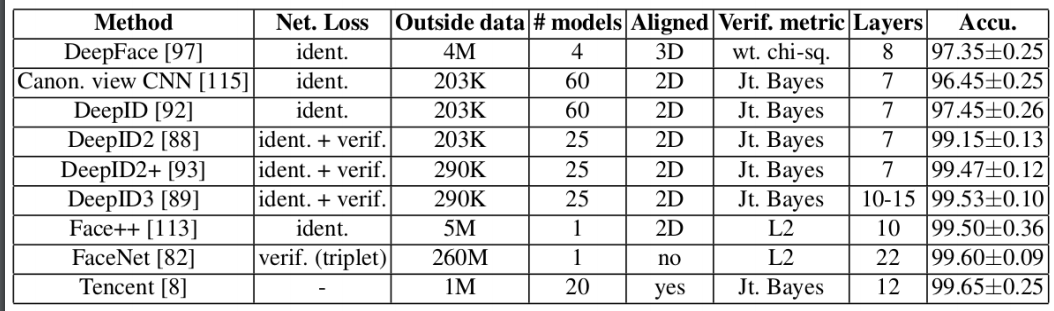
不同人脸由不同特征组成，根据不同特性达到识别的目的

**识别容易受外界因素干扰：**

* 侧脸
* 模糊
* 光照
* 遮挡
* 变形

模型：

* + 1. 深度学习算法



DeepFace

DeepID

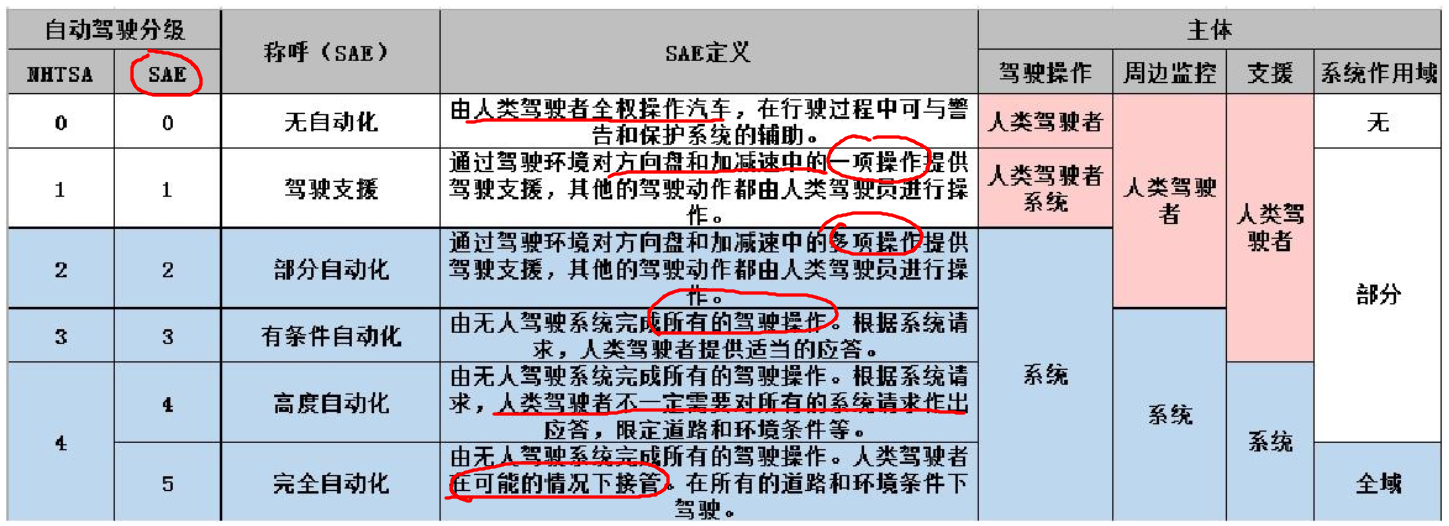
FaceNet

WebFace

* 1. 对话机器人
  2. 语音识别
  3. 自动驾驶
     1. 目标检测
* **目的：**

目标检测主要解决**分类（SVM）**与**回归（Bbox reg）**问题。先使用SVM等分类算法筛选出类别，然后利用Bbox reg等回归方法找出坐标，画出边界。

* **自动驾驶分级：**



* **业界发展：**

特斯拉：L3级别

百度：L4级别，特定区域，高度自动化

Apple：L5级别（强人工智能）

* **图像识别技术涵盖：**

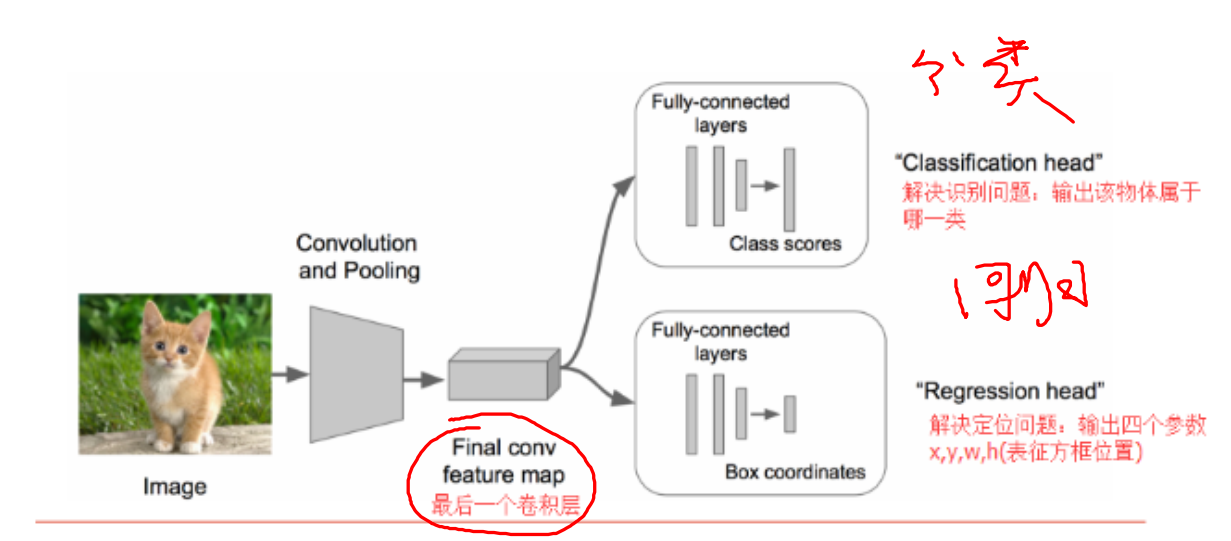
图像分类 --> 图像检测 --> 图像分割

RCNN

**介绍：**

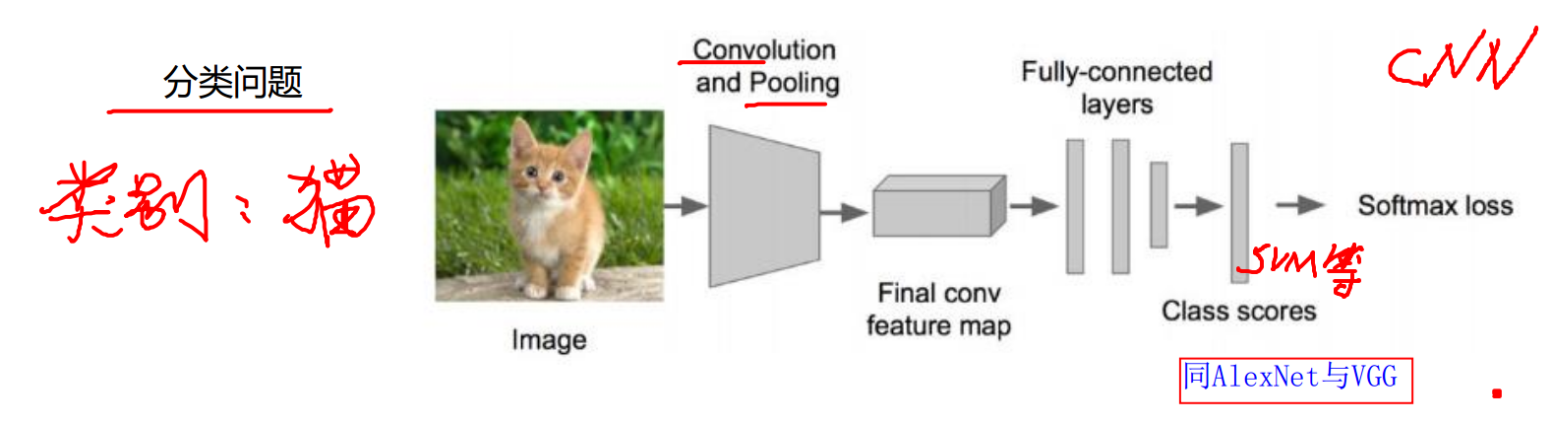
**Region CNN(RCNN)**可以说是利用深度学习进行**目标检测**的开山之作。分为**RCNN**, **Fast RCN**N, **Faster RCNN。**

* **https://github.com/rbgirshick/rcnn**
* **https://github.com/rbgirshick/fast-rcnn**
* **https://github.com/ShaoqingRen/faster\_rcnn**

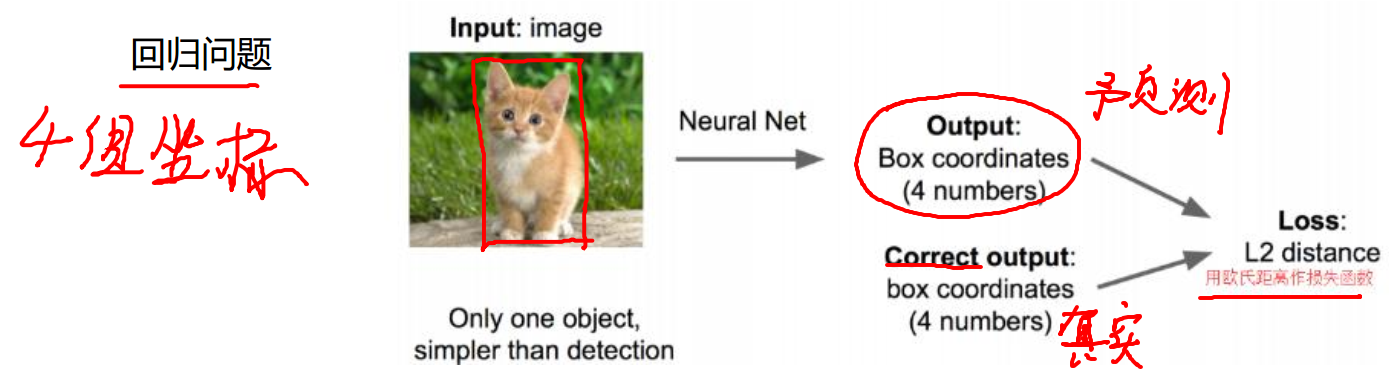


**四部曲：**

* **生成候选框**
* **卷积神经网络提取特征**
* **分类**



* **回归**



速度瓶颈

性能瓶颈

四部曲

生成候选框

卷积神经网络提取特征

分类

回归

速度瓶颈

重复为每个region proposal提取特征是极其费时的，Selective Search对于每幅图片产生2K左右个region proposal，也就是意味着一幅图片需要经过**2K**次的完整的CNN计算得到最终的结果。

性能瓶颈

对于所有的region proposal放缩到固定的尺寸会导致我们不期望看到的几何形变，而且由于速度瓶颈的存在，不可能采用多尺度或者是大量的数据增强去训练模型。

SSP Net

Fast RCNN

Faster RCNN

比较RCNN、 Fast RCNN、 Faster RCNN

YOLO

YOLO v1

YOLO v2

YOLO v3

SSD

SSD

DSSD

* + 1. 增强学习

支撑算法

Rule based

**系统复杂性：**

人工设计大量的模块

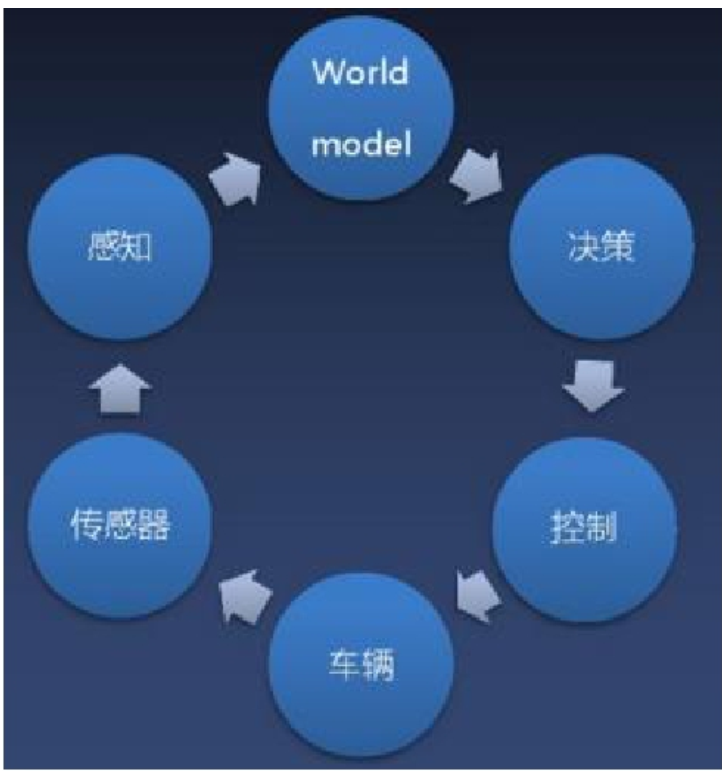
**高精度地图成本：**

广铺、更新

**车载硬件计算能力：**

各个子模块自己的网络

**难度大、需要协作生态**



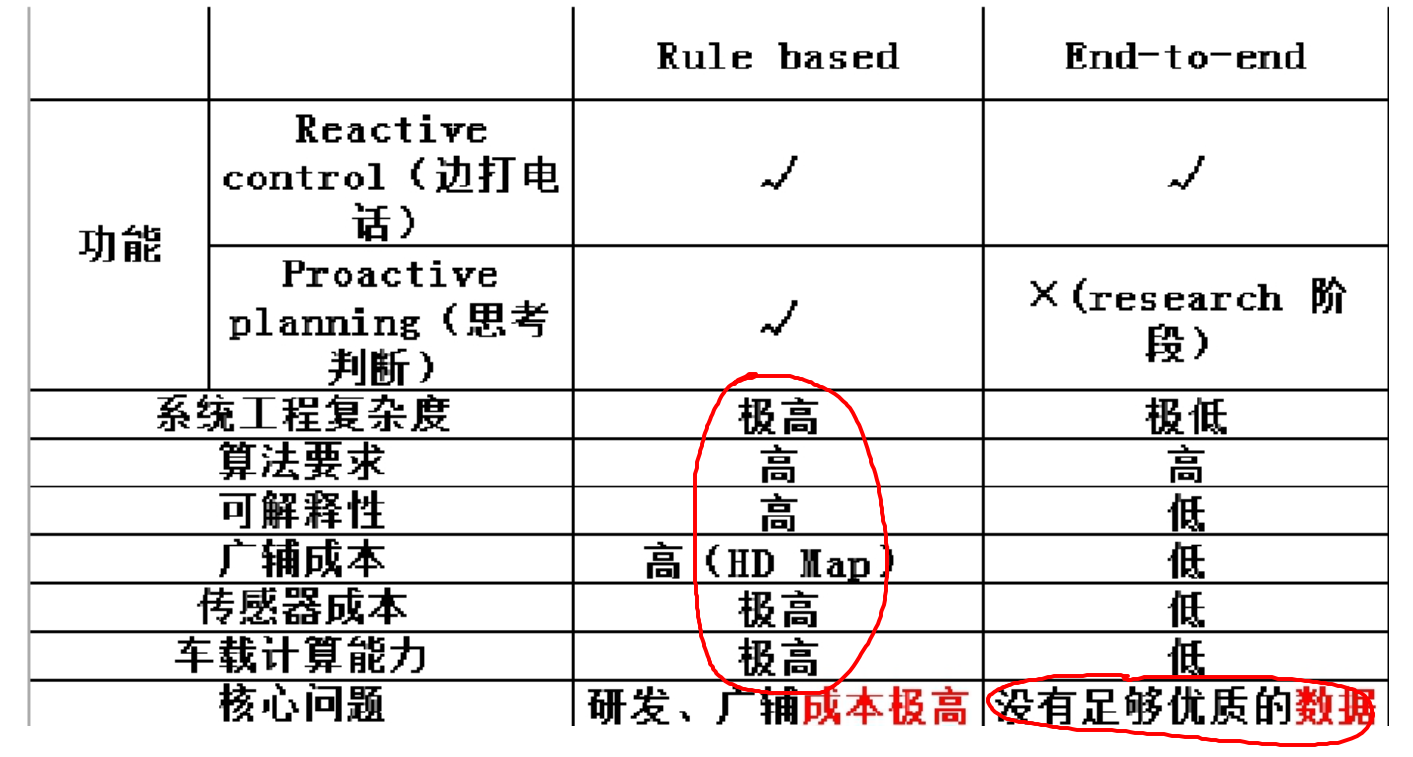
End2end

**1988 ALVINN**

**2005 DAVE**

**2015 Deep Driving**

**2016 DAVE2 NVIDIA**



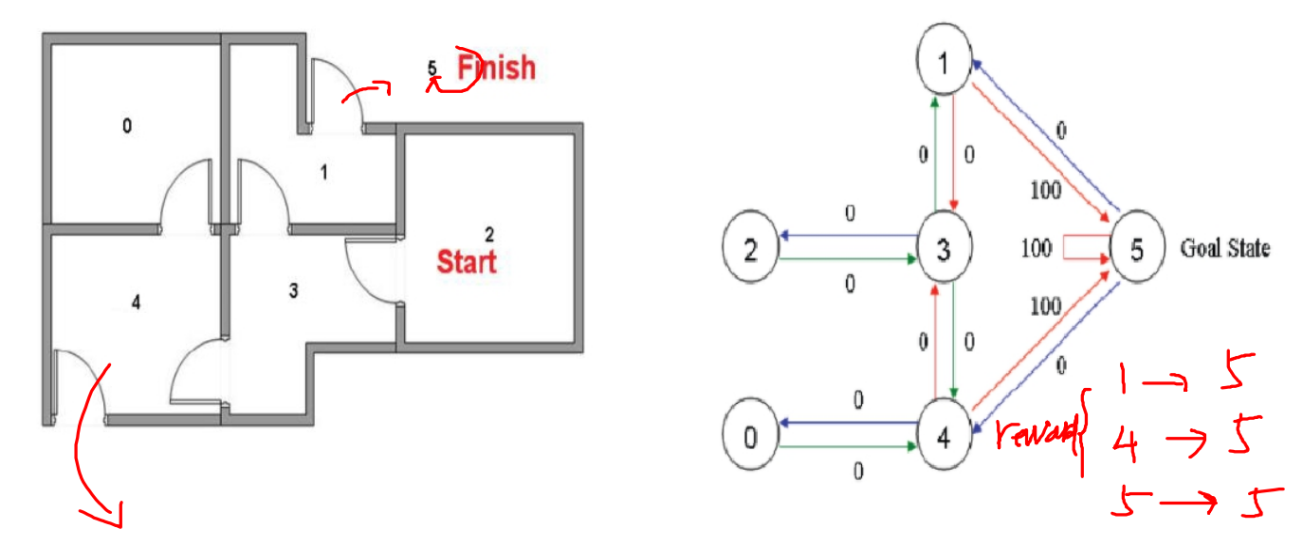
强化学习

**RL**(Reinforcement Learning)，根据结果进行反馈，不断调整以得到最好的算法。



DQN原理

Q-learn



DQN自动驾驶

**Q-learn挑战**

* 使用表格来表示Q值，显示的很多问题上几乎不可行的，因为状态太多。
* **价值函数近似** Value Function Approximate
* 深度学习**网络近似Q函数**

End2end自动驾驶

**end2end整体架构：**

* 视频处理模块
* 图像预处理模块
* NVIDIA网络构建模块
* 模型训练模块
* 模型推断模块

**End to end：**

* 英伟达的DRIVE解决方案
* 采用端到端的方法
* 以摄像头的原始输入作为输入
* 通过前期使用数据训练出来的模型直接输出车辆的速度和方向

**问题陈述：**

该项目是一个**监督回归**的问题。

**数据集：**对应每一帧图像都给定了相应的方向盘转向角度‘

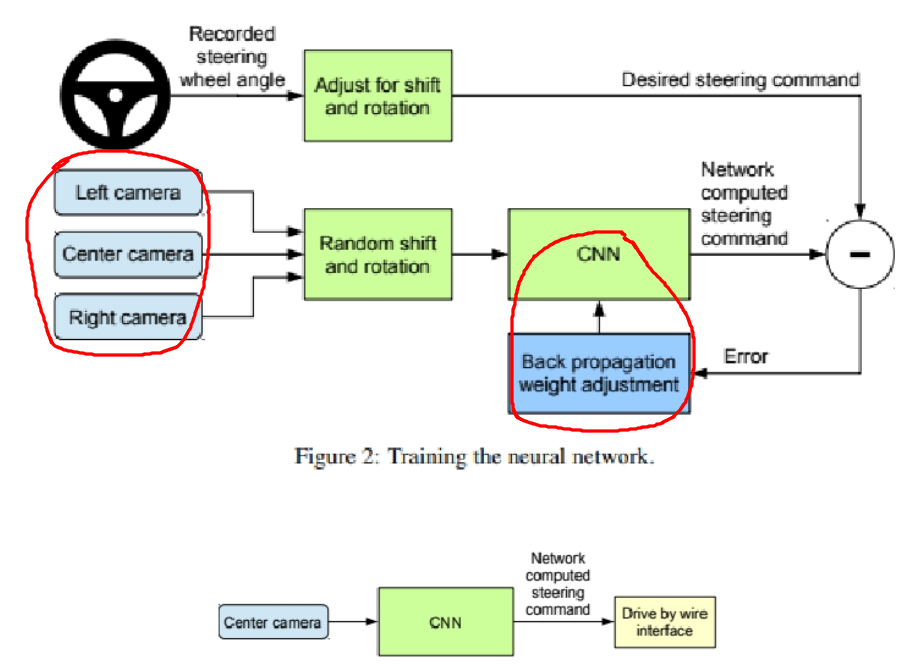
在本问题中，**输入x是前置内置摄像头获取的单帧图片**，而最后的结果的**y为转向角度**

**模型：端到端（end to end）模型**

端到端模型指的是输入的是原始数据，输出的是最后的结果

**目标：**训练一个端到端的模型f，该模型可以预测转向角度Y，其关系为Y=f(x)

**DAVE2:**

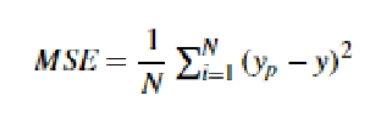


**数据集：**

10段由前置摄像头采集的MKV格式的原始图像数据，其中前9段用于训练，第10段用于测试，第1到第9个视频经提取后，共有24300张图片。图片的大小为（720,1200, 3）

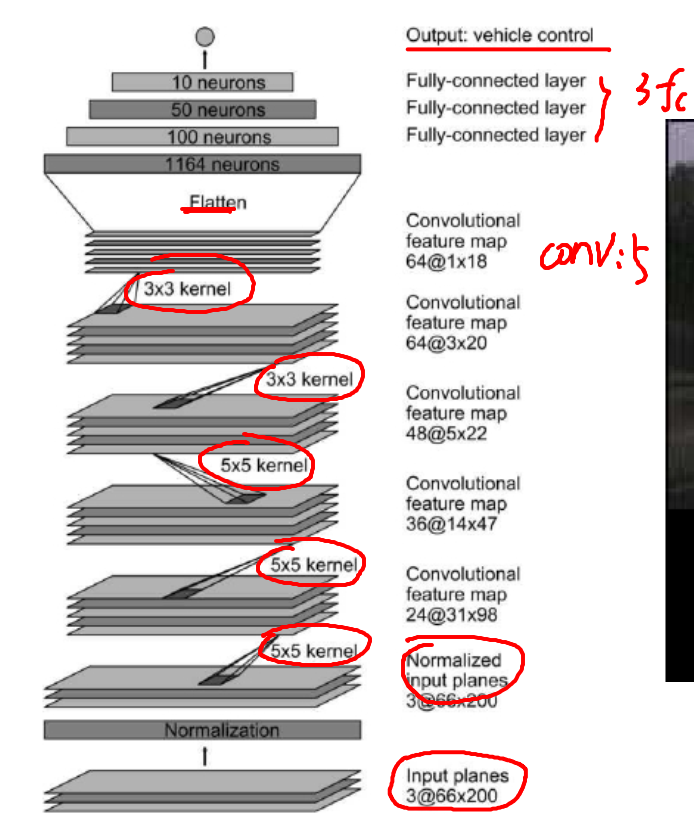
**评价指标：**

* **平均损失函数（MSE）**

，其中**N为样本总数**，**yp为预测转向角度**，**y为实际转向角度**，MSE值越小，则模型训练效果越好。

* **模型的训练和预测时长**

**英伟达模型：**



* + 1. 迁移学习

**TL(Transfer Learning）**，迁移学习主要可以用来解决两大类问题。**小数据问题**和**个性化问题。Zero-shot learning 和 One-shot learning 是迁移学习的两种极端形式。**

**迁移学习的实现方法：**

* **样本迁移**

* **特征迁移**

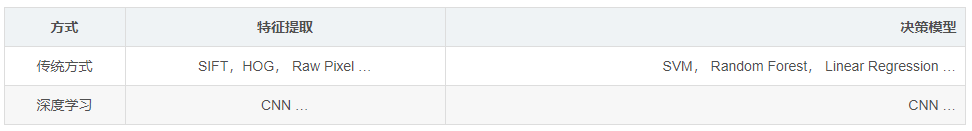
* **模型迁移**

* **关系迁移**

1. 计算机视觉

* **简介：**

计算机视觉（Computer Vision）又称为机器视觉（Machine Vision），顾名思义是一门**“教”会计算机如何去“看”世界的学科**。在机器学习大热的前景之下，**计算机视觉与自然语言处理（Natural Language Process， NLP）及语音识别（Speech Recognition）**并列为机器学习方向的三大热点方向。而计算机视觉也由诸如梯度方向直方图（Histogram of Gradient， HOG）以及尺度不变特征变换（Scale-Invariant Feature Transform， SIFT）等传统的手办特征（Hand-Crafted Feature）与浅层模型的组合逐渐转向了以卷积神经网络（Convolutional Neural Network， CNN）为代表的深度学习模型。



**方式特征提取 决策模型**

**传统方式 SIFT，HOG， Raw Pixel … SVM， Random Forest， Linear Regression …**

**深度学习 CNN … CNN …**

**svm（Support Vector Machine） ： 支持向量机**

**Random Forest ： 随机森林**

**Linear Regression ： 线性回归**

**Raw Pixel ： 原始像素**

**传统的计算机视觉对待问题的解决方案基本上都是遵循： 图像预处理 → 提取特征 → 建立模型（分类器/回归器） → 输出 的流程。 而在深度学习中，大多问题都会采用端到端（End to End）的解决思路，即从输入到输出一气呵成。本次计算机视觉的入门系列，将会从浅层学习入手，由浅入深过渡到深度学习方面。**

**语义感知：**

**几何属性：**

* 1. 传统方式

图像预处理 → 提取特征 → 建立模型（分类器/回归器） → 输出

* + 1. 梯度方向直方图（Histogram of Gradient， HOG）
    2. 尺度不变特征变换（Scale-Invariant Feature Transform， SIFT）
    3. 原始像素（Raw Pixel ）
  1. 深度学习方向
     1. 物体识别和检测（Object Detection）

物体检测一直是计算机视觉中非常基础且重要的一个研究方向，大多数新的算法或深度学习网络结构都首先在物体检测中得以应用如VGG-net， GoogLeNet， ResNet等等，每年在imagenet数据集上面都不断有新的算法涌现，一次次突破历史，创下新的记录，而这些新的算法或网络结构很快就会成为这一年的热点，并被改进应用到计算机视觉中的其它应用中去，可以说很多灌水的文章也应运而生。

物体识别和检测，顾名思义，即给定一张输入图片，算法能够自动找出图片中的常见物体，并将其所属**类别及位置**输出出来。当然也就衍生出了诸如**人脸检测（Face Detection），车辆检测（Viechle Detection）**等细分类的检测算法。

* **近年代表论文**

He, Kaiming, et al. “Deep residual learning for image recognition.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

Liu, Wei, et al. “SSD: Single shot multibox detector.” European Conference on Computer Vision. Springer International Publishing, 2016.

Szegedy, Christian, et al. “Going deeper with convolutions.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

Ren, Shaoqing, et al. “Faster r-cnn: Towards real-time object detection with region proposal networks.” Advances in neural information processing systems. 2015.

Simonyan, Karen, and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

* **数据集**

IMAGENET

PASCAL VOC

MS COCO

Caltech

人脸检测（Face Detection）

车辆检测（Viechle Detection）

* + 1. 语义分割（Semantic Segmentation）

语义分割是近年来非常热门的方向，简单来说，它其实可以看做一种特殊的分类，**将输入图像的每一个像素点进行归类**，用一张图就可以很清晰地描述出来。

很清楚地就可以看出，**物体检测和识别通常是将物体在原图像上框出**，可以说是“宏观”上的物体，而**语义分割是从每一个像素上进行分类，图像中的每一个像素都有属于自己的类别。**

* **近年代表论文**

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

Chen, Liang-Chieh, et al. “Semantic image segmentation with deep convolutional nets and fully connected crfs.” arXiv preprint arXiv:1412.7062 (2014).

Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation.” Proceedings of the IEEE International Conference on Computer Vision. 2015.

Zheng, Shuai, et al. “Conditional random fields as recurrent neural networks.” Proceedings of the IEEE International Conference on Computer Vision. 2015.

* **数据集**

PASCAL VOC

MS COCO

* + 1.  运动和跟踪（Motion & Tracking）
    2. 三维重建（3D Reconstruction）
    3. 视觉问答（Visual Question & Answering）
    4. 动作识别（Action Recognition）
  1. 计算机视觉基础
     1. 语义感知

**语义感知**在人工智能中是主角。人工智能目标是**解决“像素值”与“语义”之间的差距**（Gap）。

**场景理解（最终形态）**

* 视觉描述（Visual captioning）
* 视觉问答（Visual Question Answering）

**基础理解（根本）**

* 分类、识别、检测、分割、显著性等
* 得益于深度学习的发展，性能大幅提升

分类 Classification

物体、属性、场景等

检测 Detection

物体、行人、人脸等

识别 Recognition

物体：车牌、文本

人：人脸、指纹、虹膜、步态、行为

分割 Segmentation

检索 Retrieval

以文搜文、以图搜图、图文联搜

语言 Language

图片描述、图片问答

* + 1. 几何属性

3D建模

双目视觉

增强现实

* + 1.  主要研究的问题

数据图像处理

空域分析及变换

Sobel

拉普拉斯

高斯

中值

频域分析及变化

傅里叶变化（Fourier）

小波变换（Wavelet）

模板匹配、金字塔、滤波器组

特征数据操作

主成分分析（PCA）

奇异值分解（SVD）

聚类（Cluster）

图像特征集描述

颜色特征

RGB

HSV

Lab

直方图

几何特征

Edge

Corner

Blob

纹理特征

HOG

LBP

Cabor

局部特征

SIFT

SURF

FAST

* 1. 深度学习在计算机视觉中的应用
     1. 图像分类（卷积神经网络 CNN）
* **计算机视觉中的基础网络**
* **有监督深度模型时代的起点**
* **AlexNet --> VGG --> GoogleNet --> ResNet --> ResNext**
* **GoogleNet：**

Inception **V1 --> V2 --> V3 --> V4**

Inception ResNet **V1 --> V2**

**结构趋势：**

**更深（）**

**更宽（）**

**更多基数（）**

* + 1. 图像检测（区域卷积神经网络 R-CNN）
    2. 图像分割（全卷积神经网络 F-CNN）
    3. 图像描述（递归神经网络 RNN）
    4. 图像问答（递归神经网络 RNN）
    5. 图像生成（生成对抗网络 GAN）
  1. 图像检索