Introduction to AI

lecture 1

- Active learning: It's about how much you think and learn.
- Collective study: Let us study together!
- Study Objectives and Assessment
  - Object 1: Agent
  - Object 2: 人工智能, 机器学习, 深度学习
  - Object 3: Engage AI to our life

lecture 2

- 什么是AI?
  - science & engineering of making intelligent machines & programs.
  - The study and design of intelligent agents.
  - AI is a developing concept. -> AI是不断发展改变的概念。
- 什么是Agent?
  - An intelligent agent is a system that

    $perceives\ its\ environment$ -> 感知环境

    $takes\ actions$ -> 采取行动

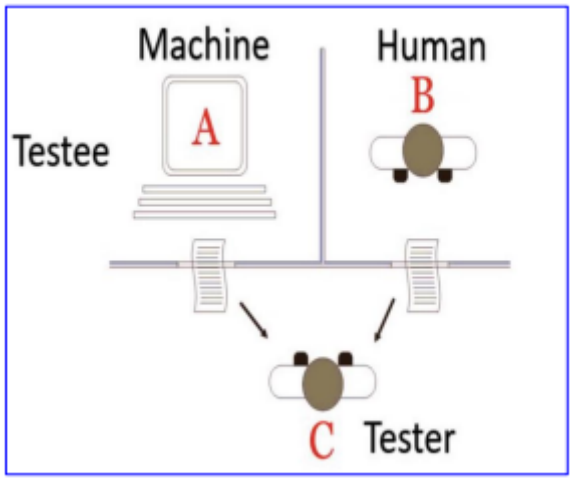
    $maximize\ its\ chances\ of\ success$ -> 最大化成功概率
  - e.g:
    - human agent
    - robotic agent
    - software agent
- 三类AI:
  - Narrow AI:
    - Dedicated to assist with or take over specific tasks.
    - 可以协助或接管一些特定的工作任务
  - General AI:
    - Takes knowledge from one domain and transfers to other domain.
    - 可以迁移知识，解决综合实际的问题。
  - Super AI:
    - Smarter than human.

lecture 3

- Computer Algorithm
  - It produces the correct output -> 产生正确的答案
  - It uses defined operations -> 采取特定的操作
  - It finishes in finite time -> 在有限的时间内完成
- AI, ML, DL
  - Artificial Intelligence: 1956年提出
  - Machine Learning: 1980s~2010s

- Deep Learning: 2010s~

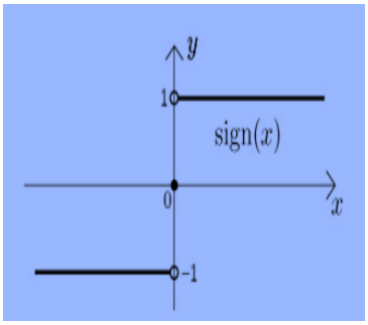
lecture 4

- AI algorithms
  - Electronic Brain -> 1943
    - A simple model of a neuron as computing machine 计算工具
    - $computes\ a\ weighted\ sum\ of\ its\ inputs$
    - $output\ 1\ if\ sum >= threshold (else\ output\ 0)$
    - $$y = f(\sum_m w_m x_m - b) \quad f(x) = \begin{cases} 1 & if\ x \geq 0; \\ 0 & otherwise \end{cases}$$
  - MCP Neuron
    - Weights are adjusted not learned.
    - 通过简单的计算细胞连接在一起试图理解大脑
    - 注意！如果$z = 0$认为应有$g(z) = 1$
  - 感知机：
    - 是一种最简单的人工神经网络（简单的前馈神经网络）
    - 一般认为感知机指单层神经网络，与多层感知机区分。
    - 缺点：不能处理线性不可分问题。
  - 图灵测试：1943
    - 用于测试机器是否具有智能
    - 2 people, 1 machine
    - 

    - 如果30%的情况下欺骗受试者 -> 具有智能
    - 2000年被通过
    - 应保持质疑态度

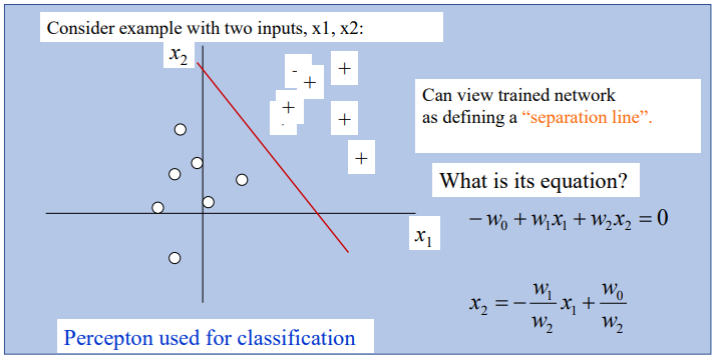
lecture 5

- 感知机：
  - 每个神经元：输入x_i，权重w_i，偏置b，激活函数$f(z), z = b + \sum_{i=1}^{n}(x_i w_i)$
  - 相比较于MCP神经元，感知机神经元的权重是通过**学习**得到的！
  - 使用阶跃函数，把输入分为两类：0和1（二元线性分类器）
- 如何学习？
  -



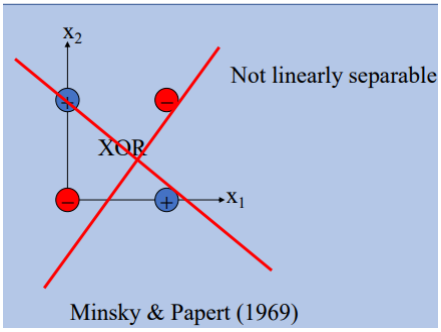
- Perception Learning Rule
  - $New\ w_i = w_i + \eta * x_i * E$
  - η：学习率，x_i：输入，E：$(ExpOut - CurOut)$
- Linear separable 线性可分
  - A function -> 其输出可以被线性函数分割

lecture 8

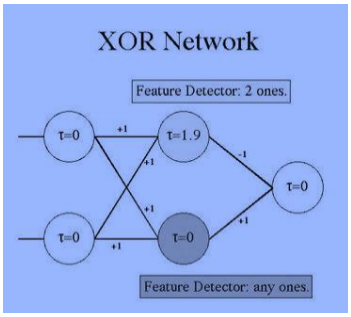
- perceptron -> Decision Surface（可以将训练的神经网络看作一个分割线/面）
  - 作为一条线：$x_2 = -\frac{w_1}{w_2}x_1 + \frac{\theta}{w_2}$
  - 

  - 不断训练，不断更新
  - 记清楚：$New\ w_i = w_i + \eta(t - o)x_i$
- ADALINE: Adaptive Linear Neuron
  - 相比感知机的用预测label来更新，我们在ADALINE中用output来更新，它可以告诉我们： “by how much we were right or wrong”
  - the iterations of ADALINE networks don't break, it converges by reducing the least mean square error（最小均方差）

lecture 10

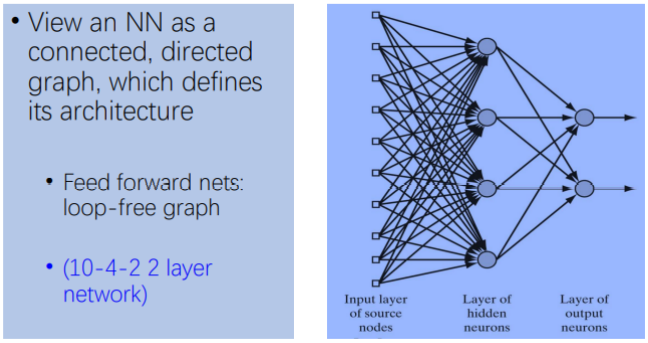
- Perceptron XOR problem
  - XOR问题不满足线性可分
  - 可以采用感知机的计算方程或者画图证明
  - 

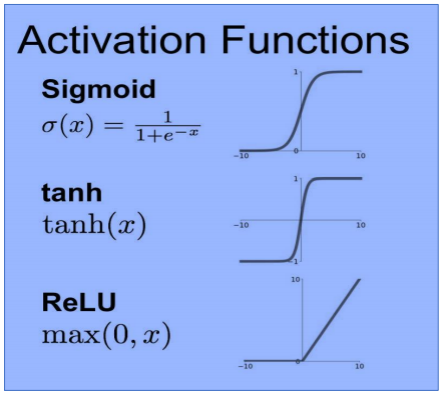
  - 但是XOR可以通过多层感知机来实现！

- 

- 常见神经网络构造
  - 单层前馈神经网络
  - 多层前馈神经网络(Multi Layer Feed Forward Network)

    General Network Architecture 2 – Multi Layer Feed Forward Network

    - 

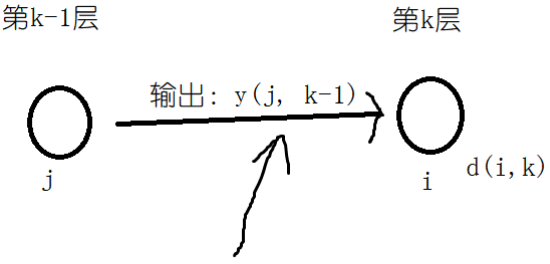
  - 循环神经网络
- 激活函数的扩增!
  - 

  - sigmoid：$sigmoid(z) = \frac{1}{1+e^{-z}}$

lecture 11

- 理论上来讲，有一个隐藏层的神经网络可以拟合一切函数。
- 2006年是$Deep\ Learning$的分界线。在此之前，浅层神经网络 is more favored，在此之后人们开始大量转向更深的网络。
- 关于$loss\ functions$
  - Quadratic Loss: 均方误差
    - MSE：$J = \frac{1}{2n}\sum_{i=1}^{n}(y_i - y_i^*)^2$，分母上的2便于求导。
  - Cross-Entropy：交叉熵误差
    - $Cross\ Entropy$：$J = y^* logy + (1-y^*)log(1-y)$
- 多分类问题：Softmax

- $y_k = \frac{e^{b_k}}{\sum_l (e^{b_l})}$

- RELU与sigmoid：
  - RELU可以避免梯度消失，Sigmoid相对平滑
  - 实践中RELU表现出了较好的效果
- BP如何实际操作?
  - 设置初始权值（权重和偏置）
  - 选择少量样本$X_{ans}\ and\ Y_{ans}$
  - 前向传播，得到Y_{calc}
  - 计算误差$e_i = Y_{ans,i} - Y_{calc,i}$
  - 更新权重：本题中以sigmoid求导反向传播为例
    - 对于第k层：$\Delta w_{i,j,k} = -\alpha d_{i,k} y_{j,k-1}$
    - $\Delta w_{i,j,k}$是第$k-1$层第j个到第k层第i个的权值变化量
    - $y_{j,k-1}$是第$k-1$层第j个的前向传播输出
    - $d_{i,k}$是 $\sum_t \frac{\delta L}{\delta d_{i+1,t}}$，也就是到损失函数值求导到目前节点的结果
      - （目前节点的变化对损失函数的贡献率）
    - 

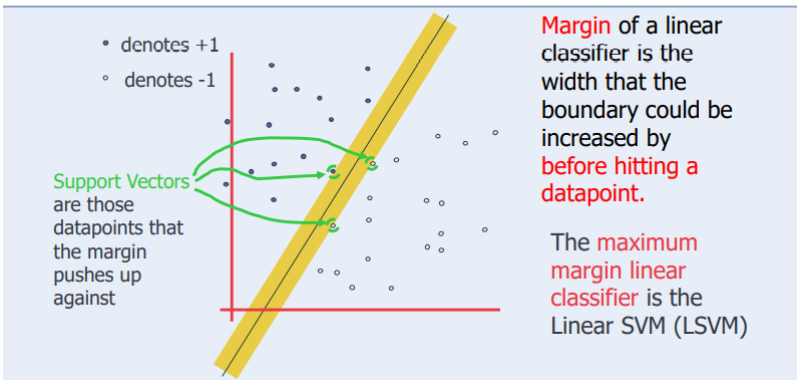
      delta_W(i,j,k)是这里的更新量。
      原先权值：W(i,j,k)
      更新后:W(i,j,k)+delta_W(i,j,k)

      α：学习率
    - 注意在反向传播中链式法则的应用
- 反向传播和感知机的区别：
  - 反向传播使用了非线性的激活函数
  - 反向传播使用求梯度进行更新来最小化误差
  - 反向传播不可以使用感知机的更新方法，因为对于隐藏层我们没有有效的更新方法（因为感知机是单层网络，它的更新方法是直接取结果的差值来更新，可是对有隐藏层的多层网络，隐藏层是没有这种“结果的差值”可以使用的。）
- GD/Mini-batch GD/SGD
  - GD：把所有数据一次喂进神经网络进行学习
  - Mini-batch GD：把数据分成小的块分次丢进神经网络学习
  - SGD：随机取一些小的块代替整体进行学习

lecture 12

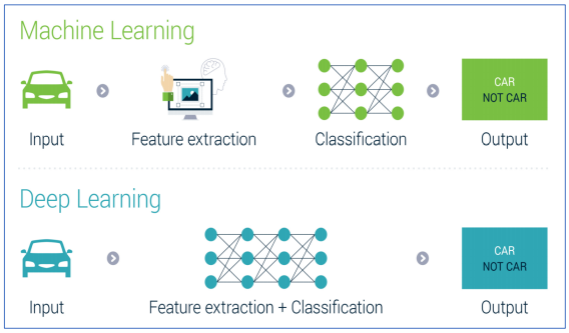
- SVM：支持向量机
  - 通过特定算法进行的监督学习
  - generally speaking：要把空间中的一些点分开，而且离分割线/面越远越好。

$$f(x,w,b) = sign(w.\ x + b)$$

- 

  - margin：分类线距离最近的分类点的宽度
  - support vectors：支持向量（顶在Margin上的点们）
  - 最大化这个margin的分类器就是Linear SVM
- 一个点（向量）$X(x_0, y_0)$到超平面$wx + b = 0$的几何距离怎么计算?
  - 对于二维的情况，我们有$l: Ax + By + C = 0, dis = \frac{|Ax_0+By_0+C|}{\sqrt{A^2+B^2}}$。
  - 而事实上，$w = (A, B)$就是l的法向量。（l对应方向向量：$v = (-B, A)$）
  - 原式因此可以推广改写：$dis = \frac{|X \cdot w+b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$，从而得到多维空间中点到超平面的距离公式。
  - 假定在两个类的支持向量上，与l平行的直线分别为$wX + b = 1$和$wX + b = -1$所以margin长度实际上为$m = \frac{2}{||w||}$
  - 所以事实上应当完成的事情就是$Minimize\ ||w||^2/2$
- $hinge\ loss$
  - 用于训练分类器的损失函数
  - $l(y) = max(0, 1 - ty)$：y是输出，t是目标结果(±1)
- Soft Margins：
  - 添加一个惩罚值$C \cdot \sum_{n=1}^{N} \xi_n$，让边缘“变软”
- Non-Linear SVM with Kernel Mapping：未知
- SVM和NN的关系：
  - Similarity：
    - 都把数据投射到高维空间进行划分
    - 都是非线性的
  - Differences：
    - data（SVM所需样本少于NN）
    - time（SVM快于NN）
    - mode（SVM一般使用GD，NN使用QP）
    - initialization（NN更依赖于好的初始参数）
- K-means clustering
  - 步骤：
    - 基于k的值，随机初始化k个中心点
    - 把n个点分别归类进入最近的中心点分为k类
    - 重新计算每一类中心点
    - 再次分类
  - P-Norm Distances
    - $= (\sum_{i=1}^{n} |x_i - y_i|)^{1/p}$

lecture 15

- *deep learning*
  - 是机器学习的一个分支，分为监督，半监督和无监督，使用特征学习
  - 特征学习是什么?
    - 让一个系统自主地发现数据中的特征，而非手动提取。
    - 

  - **三个要点** -> Key Drivers For Deep Learning
    - Big data
    - Better Algorithms
    - GPU Acceleration
- 卷积相关:
  - 具体原理和公式略
  - 卷积层的特征:
    - Local connectivity -> 局部点之间的连通性
    - Spatial arrangement -> 点的空间特征
    - Parameter sharing -> 不同的点在卷积过程中共享参数
  - $output\ size = \frac{N-F+2*Pad}{stride} + 1$
  - 池化层：提取，抽象，综合卷积层的信息
  - 全连接层：处理空间内所有点之间的关联（而非仅近点）
    - 主要计算量在全连接层
- LeNet -> 1998 开山鼻祖
- AlexNet -> 2006 改进LeNet
- ResNet -> 残差神经网络（跃层连接）
- GAN -> 生成数据的神经网络
- AlphaFold -> 处理蛋白质的神经网络（加入注意力机制后效果大大改善）
- 未来的可预见方向：无监督学习