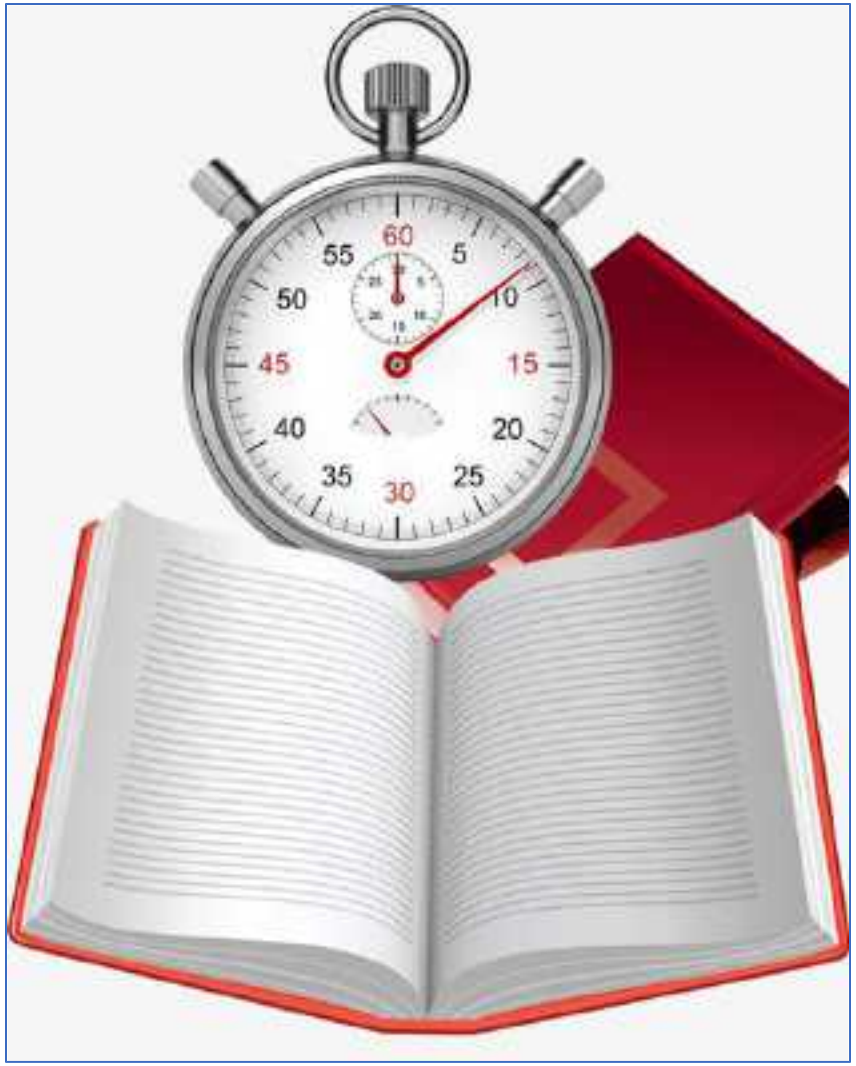
CS 103 -09
Machine Learning and BP

Jimmy Liu 刘江

2020-11-20

Review



Traditional Perceptron Decision Surface

A threshold perceptron returns 1 iff the weighted sum of its inputs (including the bias) is positive, i.e.,:

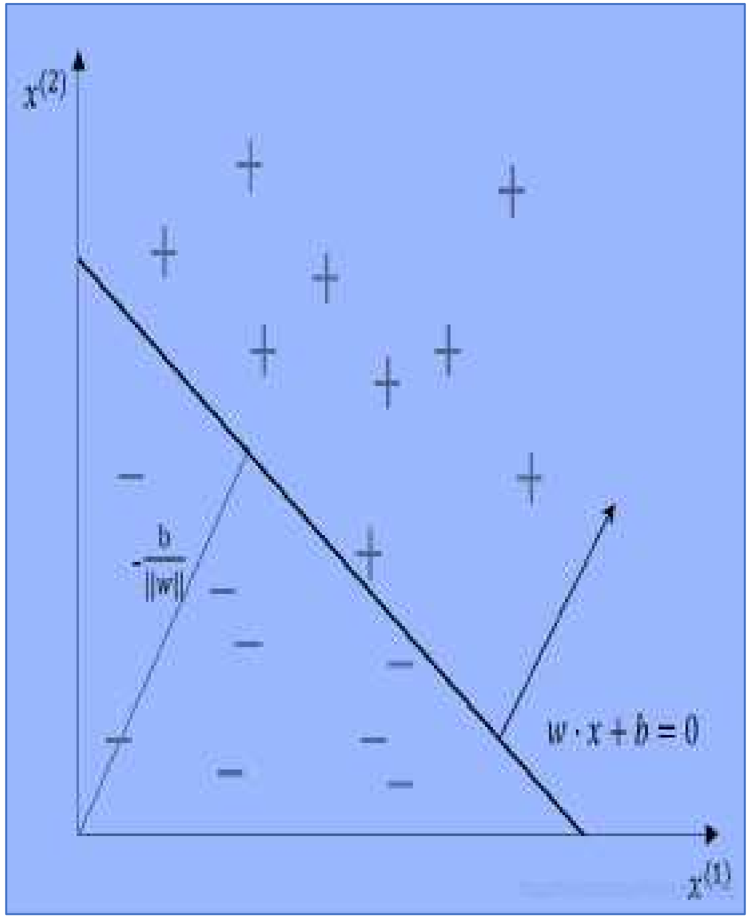
$$\sum_{j=0}^{n} W_j x_j > 0 \quad \text{or} \quad \mathbf{W} \cdot \mathbf{x} > 0$$

I.e., iff the input is on one side of the hyperplane it defines.

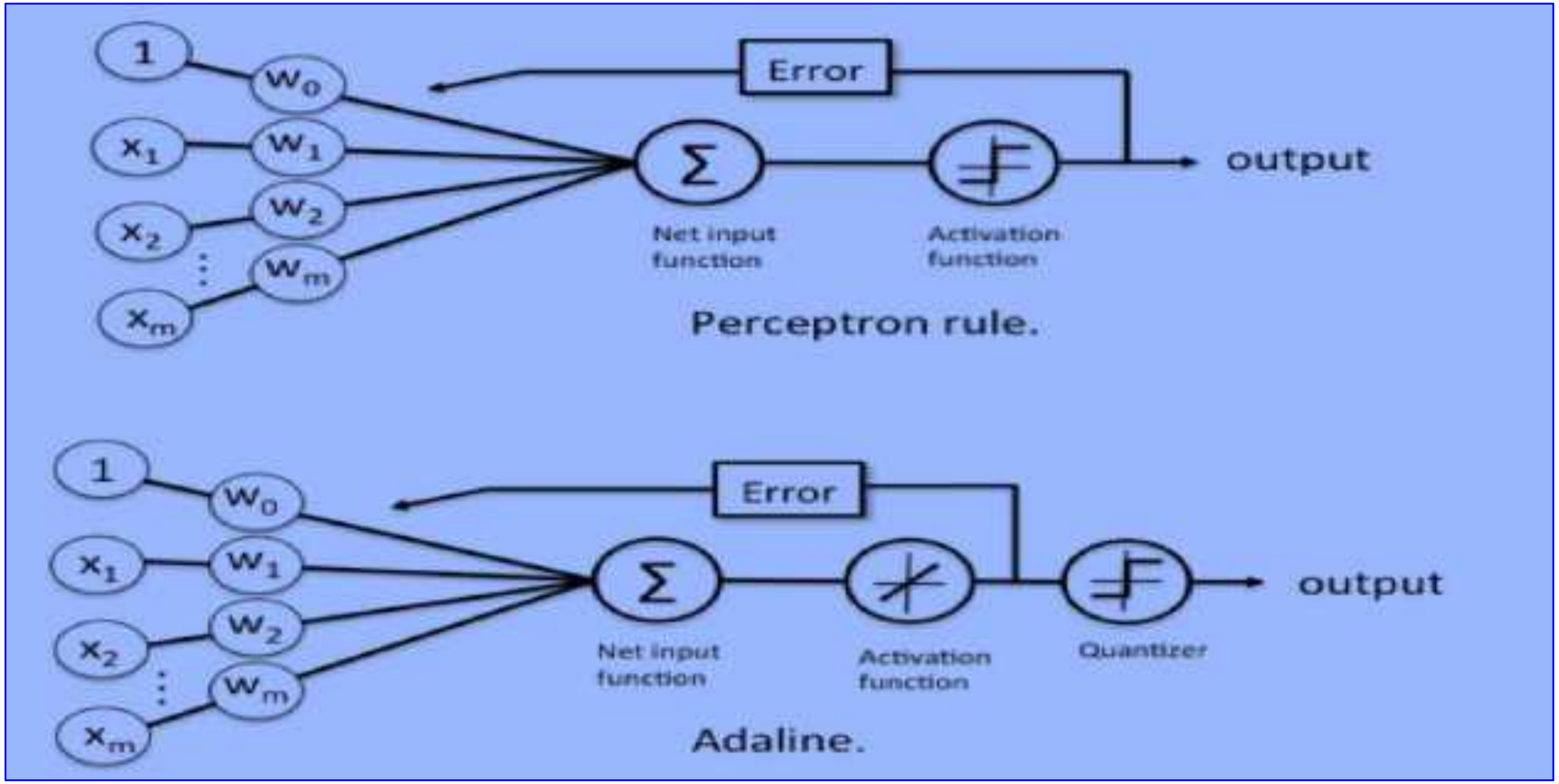
Perceptron → Linear Separator

Linear discriminant function or linear decision surface.

Weights determine slope and bias determines offset.



Perceptron and ADALINE



In the perceptron, we use the predicted class labels to update the weights, and in ADALINE, we use output to update, it tells us by “how much” we were right or wrong

Delta Learning Rule

- The motive of the delta learning rule is to minimize the error between the output and the target vector. The weights in ADALINE networks are updated by:

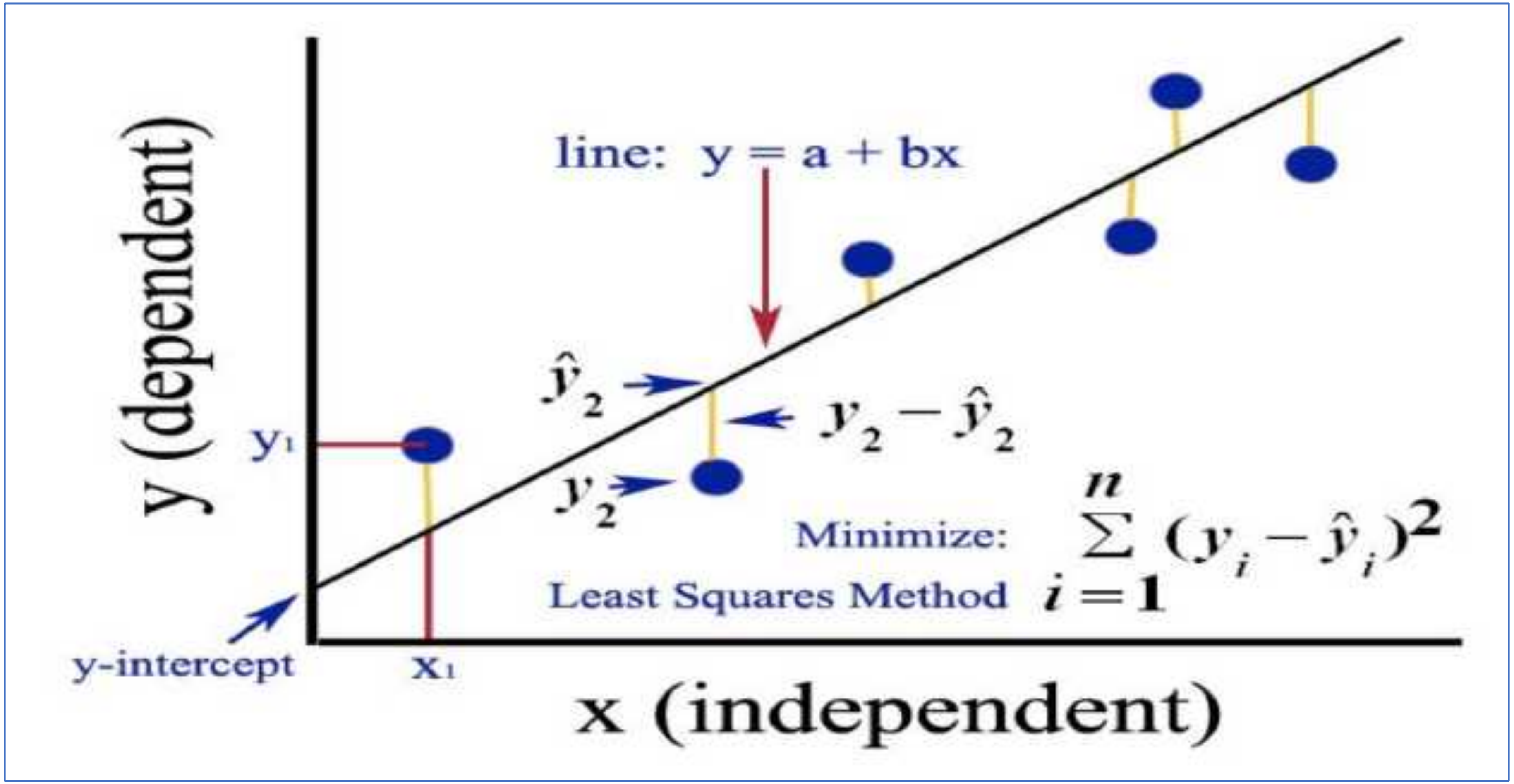
  Least Mean Square error (LMS) = $(t - y_{in})^2$,

  ADALINE converges when the least mean square error is reached.

- Learning is an optimization search problem in weight space

$\Delta W = \alpha . x_i . (t - y_{in})$, where α is the learning rate, x_i = input values and y_{in} = output, t = target value

Least Sum of Squared Errors (SSE) for MADALINE

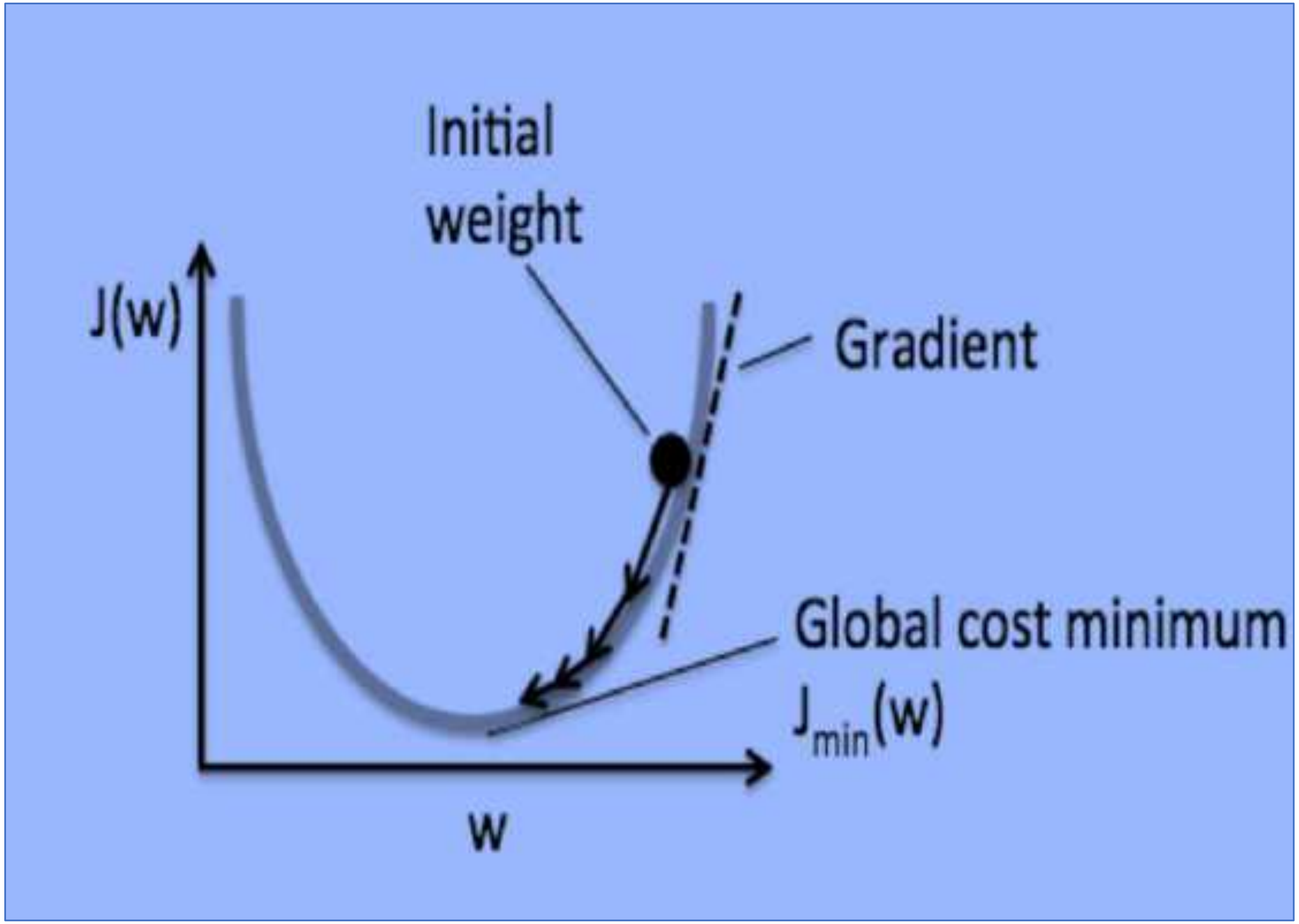


LMS Gradient Descent

Gradient Descent

Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. To find a local minimum of a function using gradient descent, we take steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point. But if we instead take steps proportional to the positive of the gradient, we approach a local maximum of that function; the procedure is then known as gradient ascent. Gradient descent is generally attributed to Cauchy, who first suggested it in 1847, but its convergence properties for non-linear optimization problems were first studied by Haskell Curry in 1944.

Wikipedia



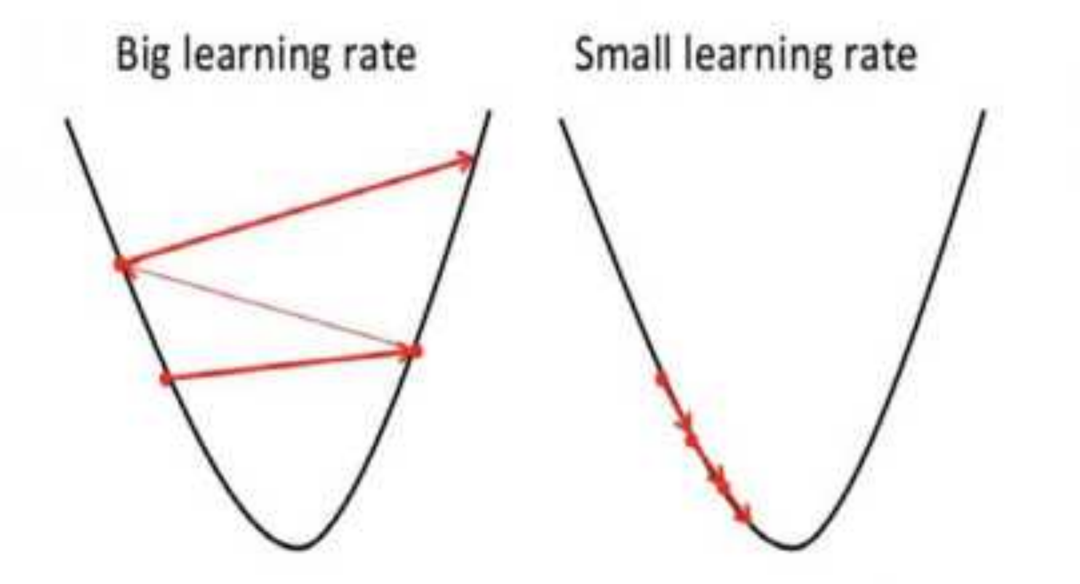
LMS Gradient Calculation = MADALINE learning

$$w_{t+1} = w_t - \alpha \frac{\partial J}{\partial w_t}$$

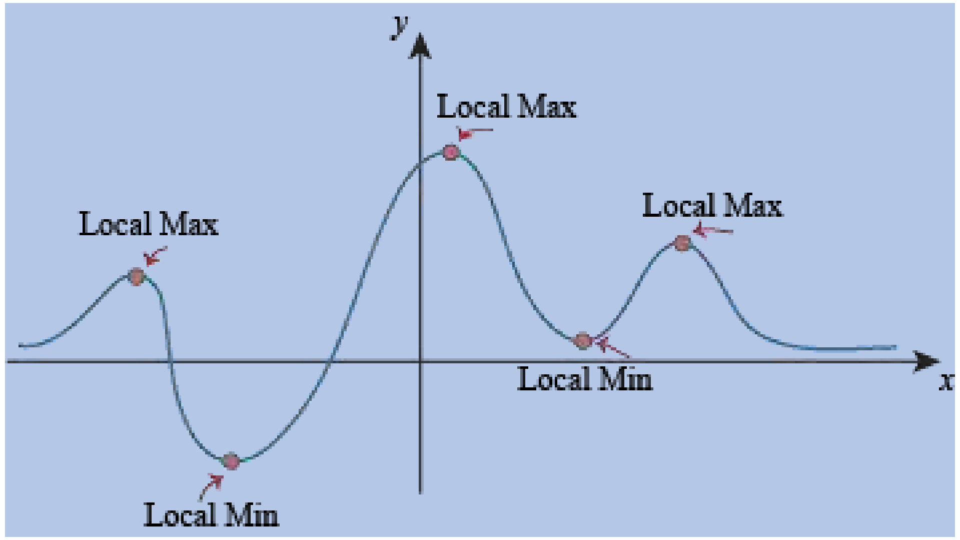
$$
\begin{aligned}
&\frac{\partial J}{\partial w_j} \\
&= \frac{\partial}{\partial w_j} \frac{1}{2} \sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right)^2 \\
&= \frac{1}{2} \frac{\partial}{\partial w_j} \sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right)^2 \\
&= \boxed{} \sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right) \frac{\partial}{\partial w_j} \left(y^{(i)} - \phi(z)_A^{(i)}\right) \\
&= \sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right) \frac{\partial}{\partial w_j} \left(y^{(i)} - \sum_i \left(w_j^{(i)} x_j^{(i)}\right)\right) \\
&= \sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right)\left(-x_j^{(i)}\right) \\
&= -\sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right) x_j^{(i)}
\end{aligned}
$$

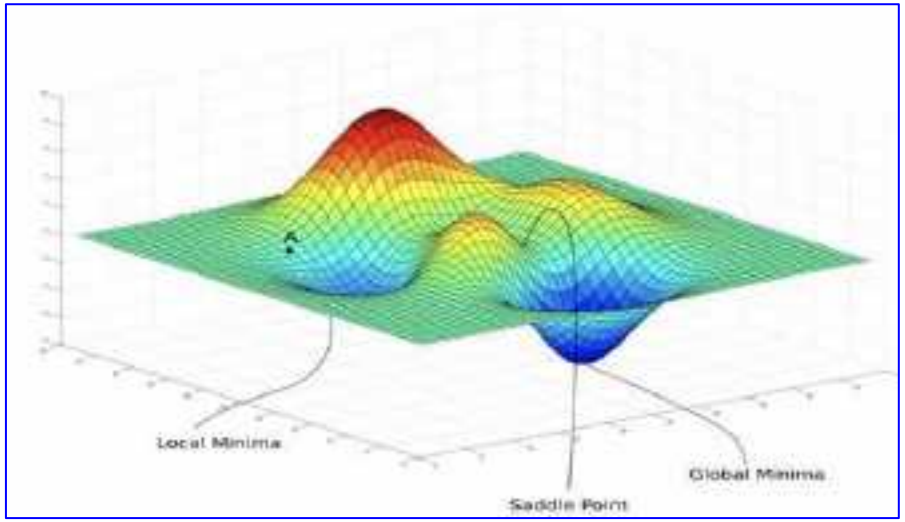
LMS Gradient Calculation

$$w_{t+1} = w_t - \alpha \frac{\partial J}{\partial w_t}$$

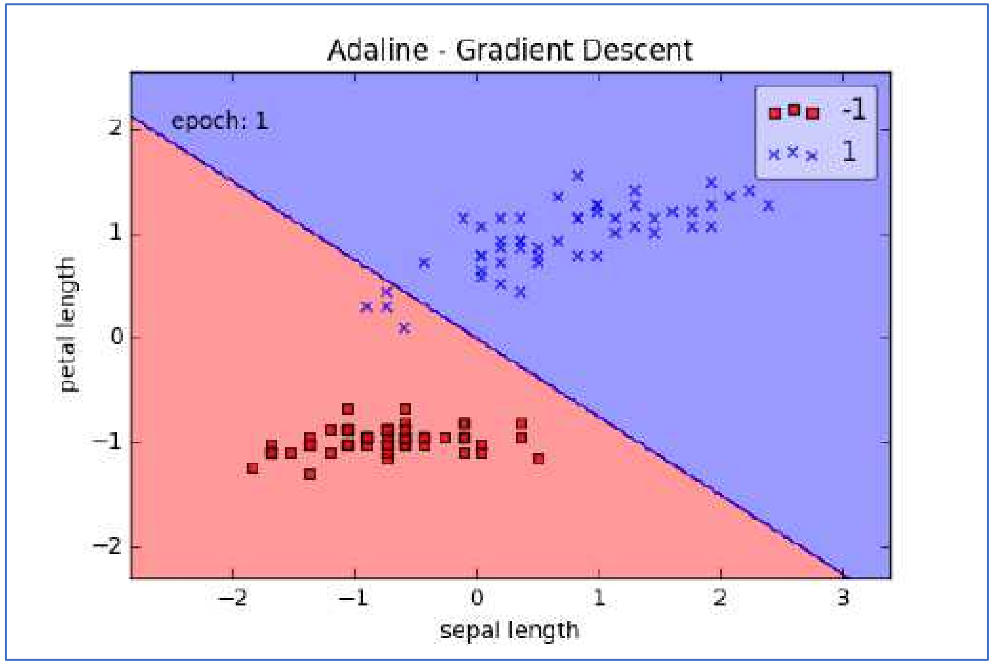


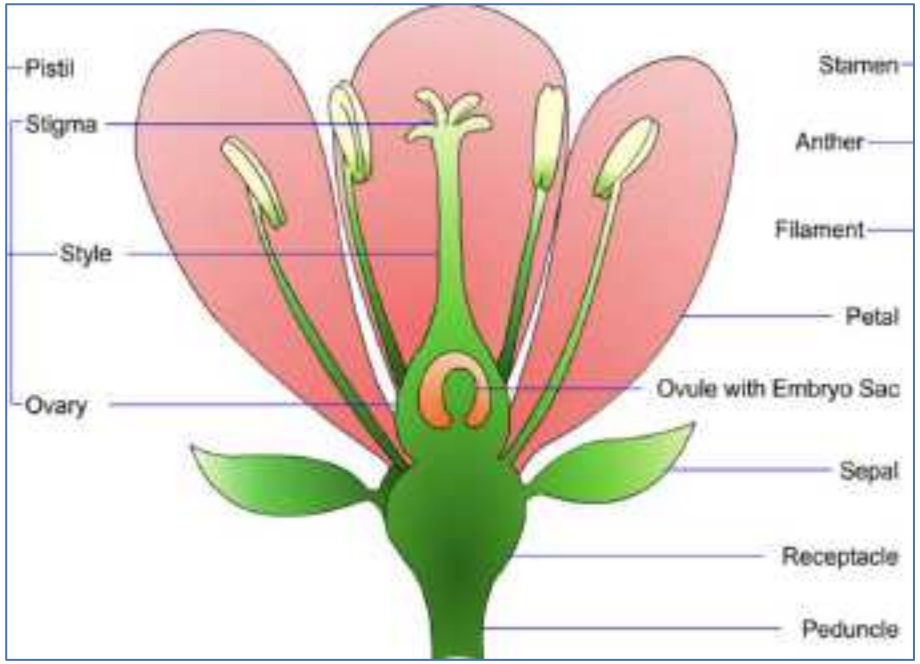
Local Minimum



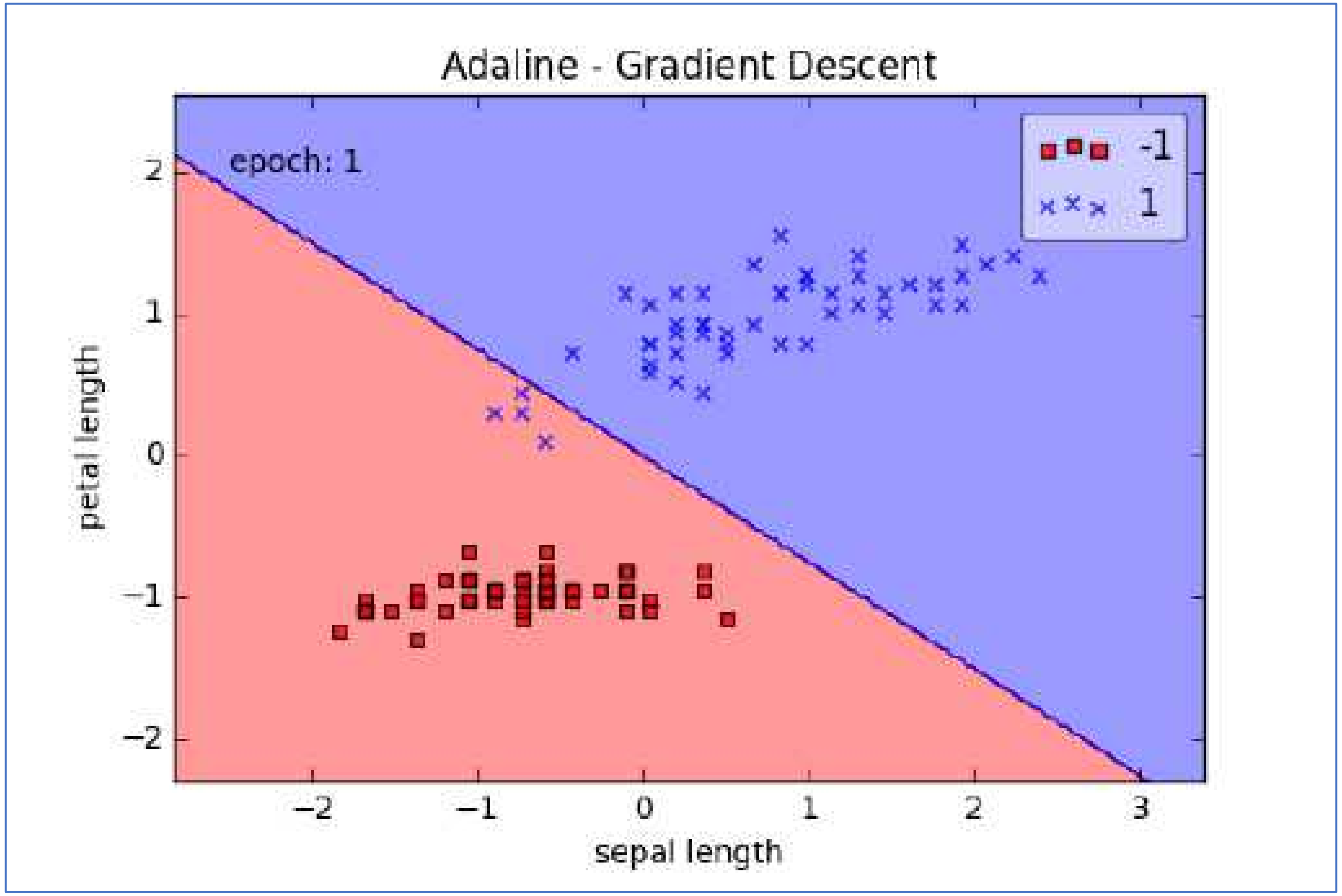


Flower Classification based on Sepal &Petal Length



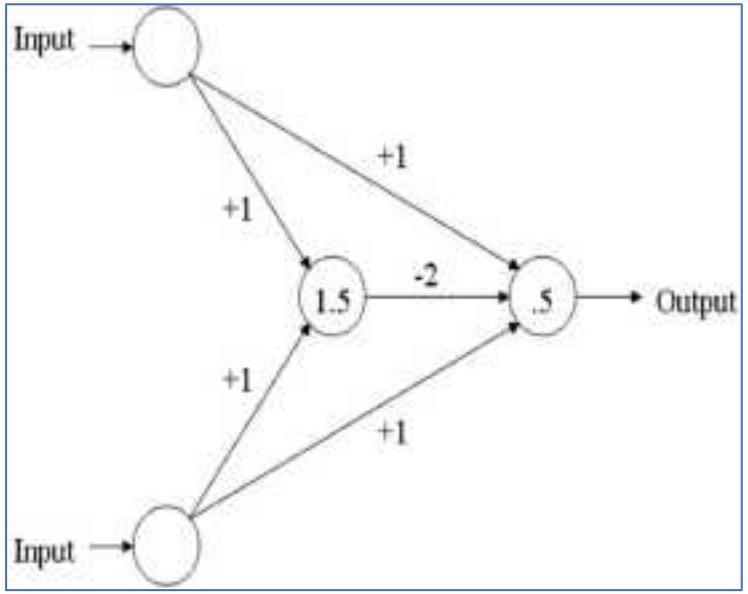


ADALINE Learning



Homework 06

1 Prove the network is an XOR network



Input		Output
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Homework 06 - Good

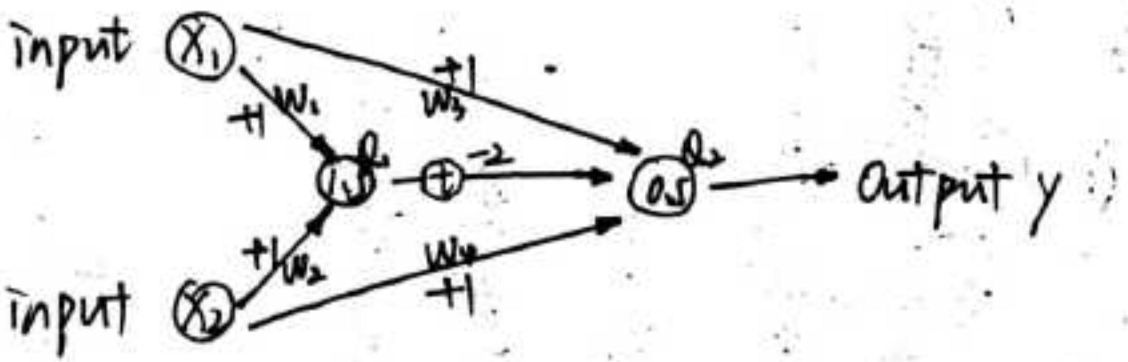
Input1 X1	Input2 X2	W1	W2	Z=sum(Xi*Wi)	Output(X3)F1=G(z)
0	0	1	1	0*1+0*1=0	0
0	1	1	1	0*1+1*1=1	0
1	0	1	1	1*1+0*1=1	0
1	1	1	1	1*1+1*1=2	1

W'1	W'2	W'3	Z'=sum(Xi*W'i)	OutputF2 =T(z')	X1 XOR X2
1	1	-2	0*1+0*1+0*(-2)=0	0	0
1	1	-2	0*1+1*1+0*(-2)=1	1	1
1	1	-2	1*1+0*1+0*(-2)=1	1	1
1	1	-2	1*1+1*1+1*(-2)=0	0	0

$$G(z)=\begin{cases}1, \text{ if } z>1.5\\ 0, \text{ else}\end{cases}$$

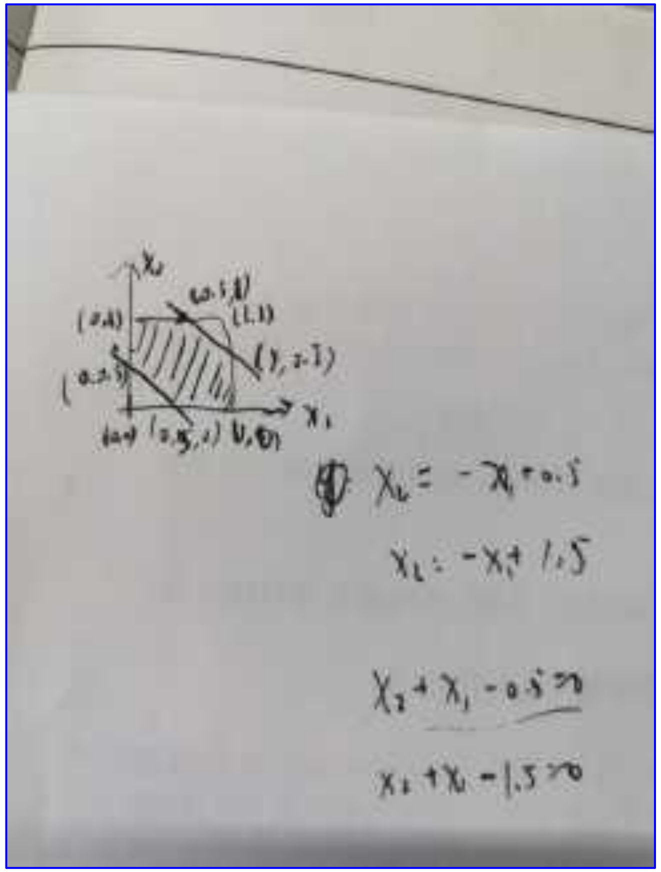
$$T(z')=\begin{cases}1, \text{ if } z'>0.5\\ 0, \text{ if } z'>0.5\end{cases}$$

Homework 06 - Good

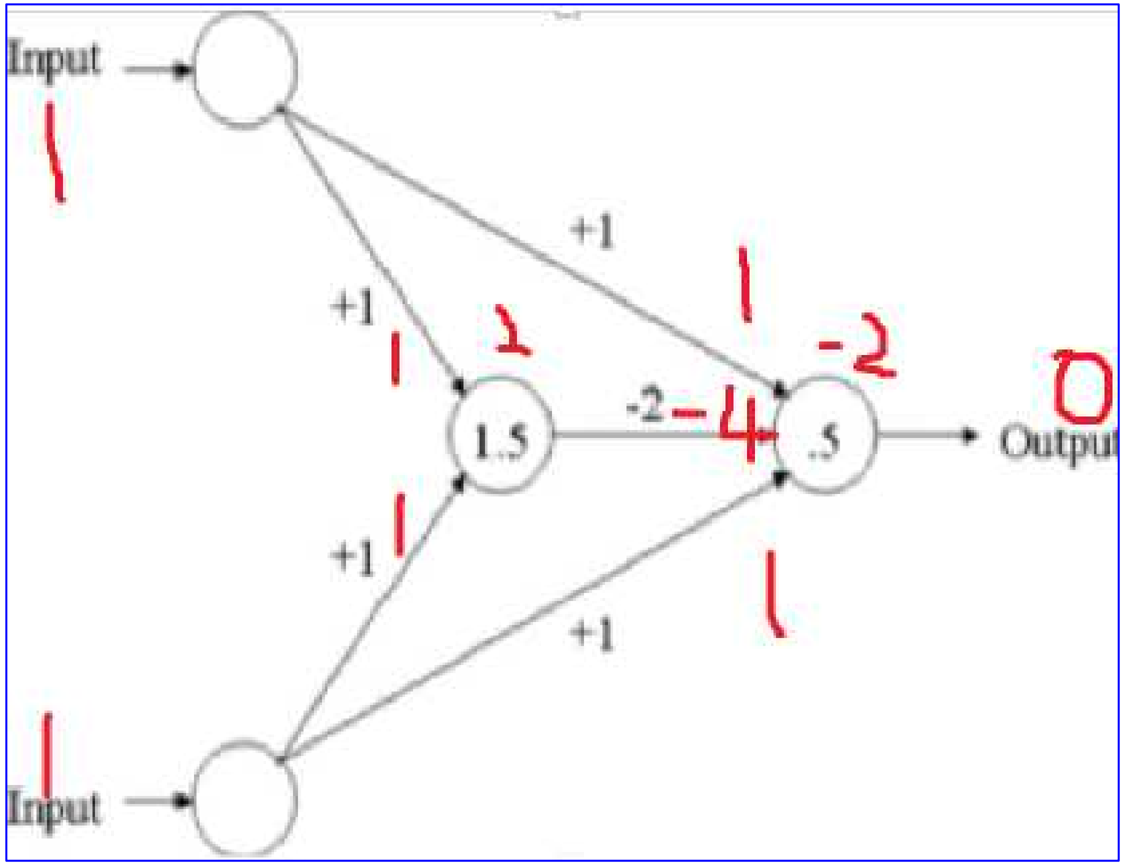


x_1	x_2	W_1	W_2	θ_1	$s_1=x_1W_1+x_2W_2$	t	W_3	W_4	W_5	θ_2	$s_2=x_1\cdot W_3+x_2\cdot W_4+t\cdot W_5$	y
0	0	1	1	1.5	$t=\begin{cases}1 & s_1>\theta_1\\ 0 & \text{else}\end{cases}$	0	1	1	-2	0.5	$y=\begin{cases}1 & s_2>\theta_2\\ 0 & \text{else}\end{cases}$	0
1	0	1	1	1.5		0	1	1	-2	0.5		1
0	1	1	1	1.5		0	1	1	-2	0.5		1
1	1	1	1	1.5		1	1	1	-2	0.5		0

Homework 06



$x_2 = -x_1 + 0.5$
$x_2 = -x_1 + 1.5$
$x_2 + x_1 - 0.5 > 0$
$x_2 + x_1 - 1.5 > 0$



Input
+1
+1
1.5
-2
.5
Output
+1
+1
Input
1
1
1
1
-4
-2
1
1
1
0

Homework 06

x1	x2	a1	h	a2	output
0	0	-1.5	0	-0.5	0
0	1	-0.5	0	0.5	1
1	0	-0.5	0	0.5	1
1	1	0.5	1	-0.5	0

Group Project Update

小组项目-调研综述进展汇报

序号	题目	成员
1	AI+斗地主	孙永康、李怀武、胡鸿飞、吴一凡、杨光、张习之、金肇轩（组长）、于佳宁
2	AI+五子棋	周贤玮、韩梓辰（组长）、赵云龙、张坤龙、夏星晨
3	High Score Gamer	易辰朗、许天淇、黄北辰（组长）、赵思源、朱佳伟、宛清源
4	AI application on diabetes	周钰奇、李仪轩、董叔文、湛掌、胡钧淇（组长）、裴鸿婧
5	AI in lung cancer	夏瑞浩、李悦明、龚颖璇、吴云潇潇（组长）、姜欣瑜、王英豪
6	基于MRI图像的阿尔茨海默症分类	董廷臻、郑英炜（组长）、李博翱、朱嘉楠、李杨燊
7	AI Applications in Breast Cancer Imaging	林文心、翟靖蕾（组长）、孙瀛、林宝月、陈帅名、冀鹏宇
8	Applications of artificial intelligence in covid-19 patients	罗岁岁（组长）、周雅雯、肖雨馨、程旸、尹子宜
9	基于OCT图像的眼部多种疾病诊断和分析的调研	何忧、郭煜煊、朱寒旭、赵子璇（组长）、王子杰、张晓新
10	人工智能对白内障分级的算法综述	赵宇航、徐格蕾、陈星宇、祖博瀛、黄弋骞（组长）
11	句子图片的文本情感分	唐云龙、刘叶充、刘旭坤、马卓远、陈子蔚（组

序号	题目	成员
12	gesture recognition	车文心、张静远、张骥霄（组长）、杜鹏辉
13	AI in Lab	孙含曦、于松琦、罗西（组长）、唐家豪、孙杰欣
14	人脸识别算法的发展与应用	易翔（组长）、陈俊滔、罗景南、胡泰玮、文颖潼、吴杰翰
15	人工智能在无障碍设施领域中的使用调查	马子晗（组长）、陈沐尧、林小璐、任艺伟、王增义
16	identification of handwriting elements	刘通、谈思序、赵伯航、张皓淇
17	AI虚拟主播制作计划	王标、张倚凡（组长）、李康欣、何泽安、曾宇祺、 Zhang Kenneth
18	人工智能技术在个性化推荐系统上的应用与研究	谭雅静、刘思岑、Ooi Yee Jing、孟宇阳、杨锦涛（组长）
19	校园巴士路线优化	王祥辰、何鸿杰、吴子彧、樊青远（组长）、方琪涵、袁通
20	给线稿上色的强大AI的算法研究	韩晗（组长）、刘思语、赵晓蕾、陈松斌
21	人工智能应用于病理分析的前景与挑战	刘宇欣、李修治（组长）、沈睿琦
22	深度学习在自动驾驶中的应用	王晓轩

Group 1: Fighting the Landlord With AI

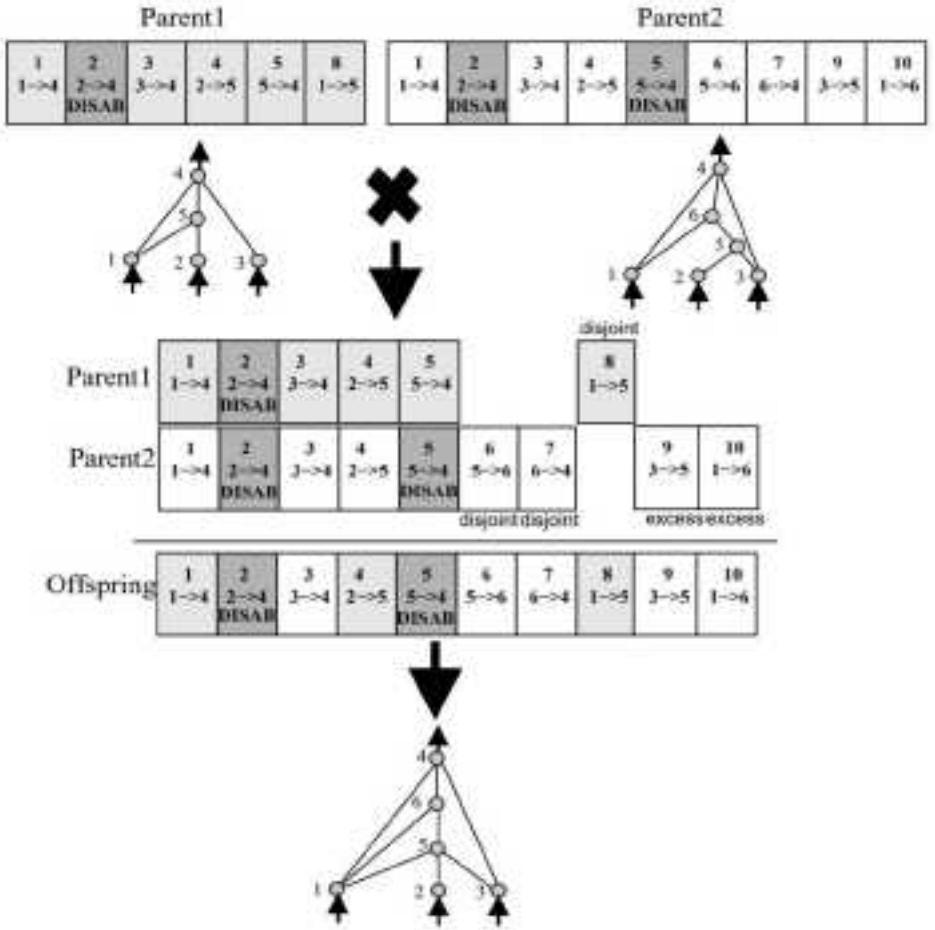
- Basic design
  - Implement the frame and basic game flow of the game·········FINISHED
  - Implement the rules of playing cards in detail···············ONGOING
  - Design the graphical interfaces·····························PLANNING

- To be finished in one week

- Adding AI
  - Learn basic AI and algorithm·····························ONGOING
  - Set up the AI structure··································PLANNING
  - Train AI···PLANNING

Group 3: High Score Gamer

- Written fully in Rust from scratch.
- A simple and naïve(for now) implementation.
- Solve cartpole problem with in seconds.



Group 4： AI application on diabetes

自动胰岛系统

整理了一些已经成功的AI＋ 糖尿病自动给药的案例。

主要算法是循环神经网络和对抗式生成网络。

探索到如何通过病例图像使用神经网络判断给药用量。

探究AI的在糖尿病自动给药方面的案例研究

围绕 1.病因及现状
2.困难及必要性
3.应用举例
4.未来展望
四方面入手，目前初稿大致完成

控病虚拟助手

尝试分析一些关于现有的糖尿病人虚拟助手app用到的人工智能算法以及app的运行原理。

个性化血糖管理

病症预期早筛

基于GA-Xgboost的糖尿病回归预测；

基于集成学习的糖尿病分类预测。

病症确诊特点：
1型糖尿病 “三多一少” ：
多尿、多饮、多食。
2型肥胖无力

Group 6

第6组（AD分类）第9-10周进度汇报

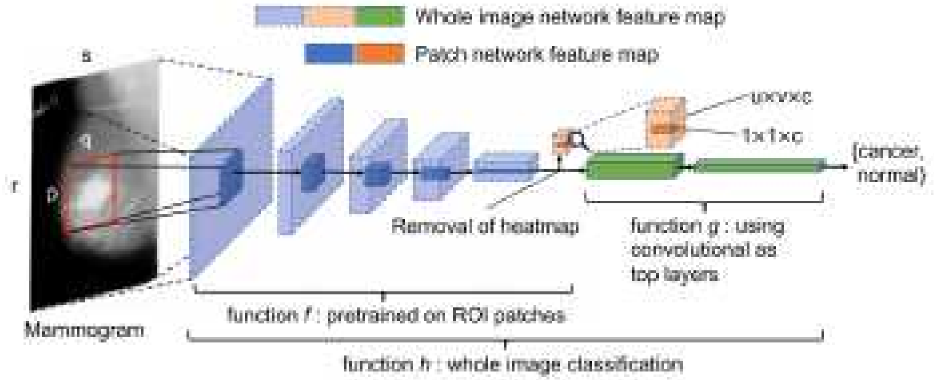
- 受High-frequency Component Helps Explain the Generalization of Convolutional Neural Networks 中的讨论的启发，去除了预处理过程中中值滤波的操作，实验表明这样做虽然没有提升性能，但提升了网络的稳定性。
- 目前取得的最好成绩（ResNet变种）：

id	TN	FP	FN	TP	Accuracy	F1	AUC	MCC
0	160	1	7	13	0.955801105	0.764705882	0.890062112	0.755525475
1	156	5	3	17	0.955801105	0.80952381	0.969875776	0.785728111
2	160	1	2	18	0.983425414	0.923076923	0.988509317	0.914173328
3	161	0	2	18	0.988950276	0.947368421	0.955590062	0.942845192
4	159	0	3	14	0.982954545	0.903225806	0.954125046	0.899043307

- ACC较之前提升0.02，F1较之前提升0.07左右
- 复现了Attention，ResNeXt，ACNet等网络，没有取得好的效果。
- 目前遇到的问题：
  - 目前复现的同类论文在我们的数据集上效果极差，F1一般在0.4-0.7左右。替换基本块（ACNet，SENet）也没有取得很好的效果。接下来打算测试新的训练方法以及其它数据增强方法。
  - 还有两个组员没有参与到项目中。

Group 7

AI Applications in Breast Cancer Imaging



AI in detection: CNN

Interpreting screening mammograms：

1. Calcification clusters
2. Soft tissue findings： masses， architectural distortions， asymmetries

Conventional computer-aided detection and diagnosis

1. Computer-aided detection (CADe) a

   aimed at locating suspicious lesions： soft tissue masses & calcification clusters.

2. Computer-aided diagnosis (CADx)

   estimate if a given, already detected lesion is benign or malignant

3. Use programmed-in features

   the algorithms are programmed to search for specific features (humans have identified)

   —— primary distinguishing feature between CADe/CADx algorithms and AI-based algorithms.

Deep learning convolutional neural networks(CNN)

1. Learn the feature by itself, not humans
2. Objectives: not only classification, but also location —— pre-training
3. Reduce the complexity of training —— transfer learning(using an already-trained CNN)

Group 8

- 细化了方向——AI在Covid-19制药方面的应用
- 找到了合适的文献
- 有了明确的目的

Model training using Bernoulli the Naïve Bayes classifier implemented in Scikit-learn

The Naïve Bayes algorithm was selected due to its basic implementation and demonstrated ability to perform in a variety of target prediction settings [28, 39, 40]. The specific classification algorithm of choice for this study is the Bernoulli Naïve Bayes algorithm, due to the ability of the algorithm to interpret the binary bit string features used to describe compound inputs. In comparison to other methods, this algorithm is also capable of maintaining its predictive power for highly imbalanced datasets, which is particularly beneficial when attempting to utilise large numbers of negative instances in training data [65, 66]. For example, it has been shown that enlarged negative training set sizes hinder the recall performance of the SMO, Random Forest, Ibk and J48 algorithms [67]. The preferable ratio of active to inactive compounds for these methods was found to be only around 1:9, a significant decrease from the 1:100 ratio envisaged for this study. Such algorithms would therefore require large scale undersampling of inactives data points to obtain acceptable performance, thus sacrificing the coverage of inactivity space.

In the target prediction context, an input molecule can be viewed as a vector of chemical features $\boldsymbol{x} = (F_1, \ldots, F_N)$, described by the fingerprints used to define a compound in relation to a target. Equation 1 shows the Bayes' theorem, underpinning Bayesian models. This equation returns the 'conditional probability' or $p(C|\boldsymbol{x})$, an evaluation of the certainty about a compound belong to class C [68].

$$p(C|\mathbf{x}) = \frac{p(C)p(\mathbf{x}|C)}{p(\mathbf{x})} \tag{1}$$

The likelihood function for Bernoulli Naive Bayes is based on Eq. 2, which represents how likely a query compound $\boldsymbol{x} = (F_1^{Test}, \ldots, F_N^{Test})$ exhibits activity against a given target C. The *BernoulliNB* class from the Scikit-learn [69] library was employed to implement the Bernoulli

Group 9

上周工作汇报

1. 11.9 开会：讨论新大纲内容和任务分工，每人分配到2篇文献（6位组员）
2. 参加高交会，学习人工智能新发展和新应用（2位组员）
3. 11.13开会：交流阅读文献之后的想法，修订和完善新大纲（5位组员）

这周进度情况

1. 根据上周讨论结果，每人继续阅读2篇文献
2. 11.20开会：一起填充和撰写文章内容

下周任务安排

1. 根据11.20晚上写作结果确定下一次开会时间
2. 考虑是否咨询和采访一些教授和医生
3. 继续阅读文献

大纲

background：

1. 青光眼介绍
2. 除基本知识（致病原因，现有诊断方式，治疗方法等）外集中介绍主要结构性病理变化以映照后文对于相关算法的分析
3. （采访）补充诊断青光眼的难点等

算法介绍

1. 简单介绍算法，最好针对不同病理变化明显之处进行分析
2. 对比算法优势和功能等，偏重分析
3. 待补充

Conclusion

Discussion

展望，阐述未来能够实现的功能及其意义等

Reference

Group 10

AI+医疗

- 1. 收集关于AI在白内障分级的应用的论文（不使用深度学习算法，论文时间应该比较久远）并简单写出概要
  2. 收集关于AI在白内障分级的应用的论文（使用深度学习算法）并简单写出概要
  3. 收集关于AI在白内障分级/其他眼部疾病的应用的论文/综述（杂项）并简单写出概要
  4. 收集前人做过的AI在白内障分级的应用的综述(至少2篇)
  5. 根据4的大概框架，确定下一步的方向以及概括123的基本内容

Group 11

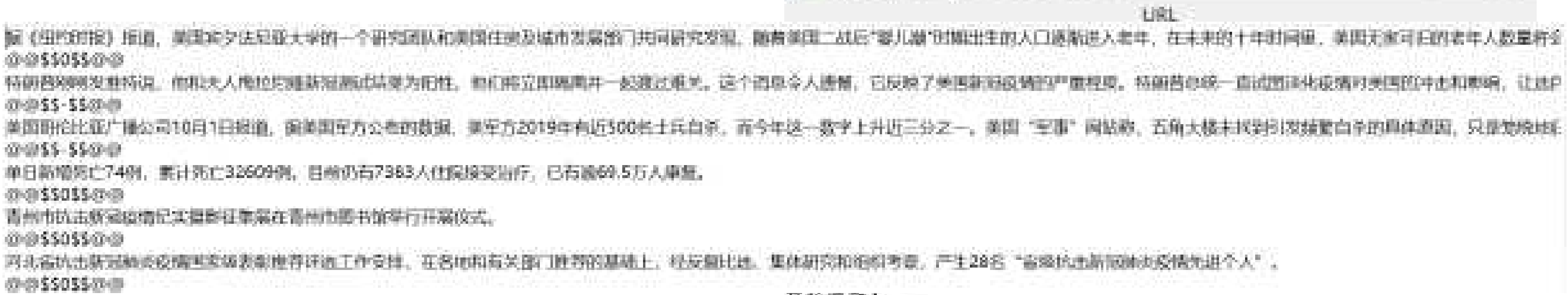
小组主题：图片文本识别与文本情感分析

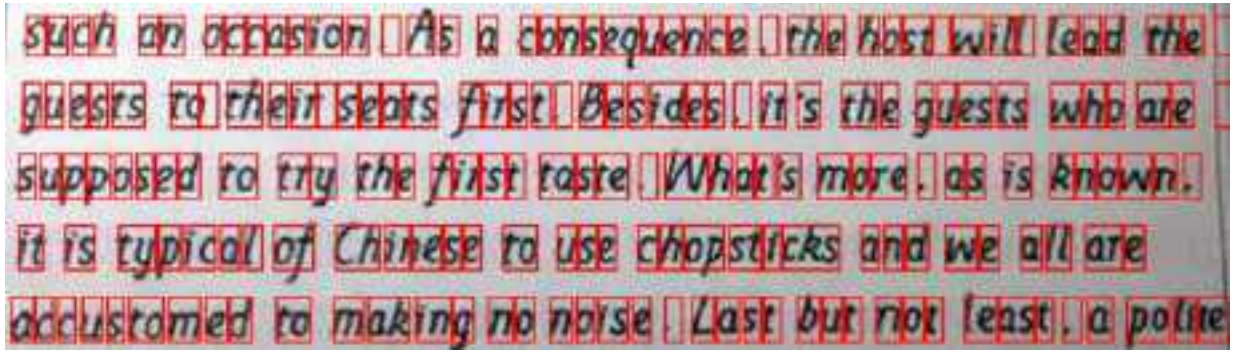
TODO LIST:

~~图像矫正~~
~~文字框选~~
~~汉字数据集获取~~
构建汉字识别模型
模型训练
~~爬虫~~
~~文本分析数据集获取~~
~~人工标注~~
模型构建与训练

上周进展：

1. 数据爬取与标注
2. 图像矫正，汉字框选部分的优化。
   对比直线探测法，轮廓提取法矫正图片的效果，并选择最优参数
   调整图像灰度化，二值化的参数。使其适用性更广
   调整腐蚀与膨胀的参数，增大框选的准确度





Group 12

- 组员：张骥霄，车文心，张静远，杜鹏辉
- 项目内容：手势识别及其应用
- 本周进展：
    - 学习CNN，GCN，TCN等项目所需算法
    - 学习pytorch等项目所需工具
    - 学习训练一些简单的神经网络，为搭建项目模型做准备
    - 提出了一些可能行之有效的模型想法

Group 13 AI in Lab

罗西：阅读A mobile robotic chemist 论文，收起其他ai in lab 的前沿技术与应用。把survey的结果在paper中阐释。

于松琦：阅读CNKI论文五篇，都是和ai lab前沿相关。Survey—位respondent。

总结:
在逐渐收集前沿资料和调查实验室研究员的过程中:
1.了解+ 理解 最新的 AI 应用在 不同实验室 的例子
2.结合上课所学的AI概念，体会在解决不同问题时候，AI 所发挥的作用。

第15组：人工智能在辅助视障残疾人方面的应用调查

已完成：

1.查找相关资料

2.分配任务

3.大体框架思路

4.找到相应的ai产品（基于飞桨的智能眼镜视觉辅助系统、盲人视觉辅助眼镜 II），开始对具体的产品资料进行分析

第16组： Handwriting identification

Write a Project Plan Describing Intended Algorithms and Application You Want to Research for Your Project and Project Milestones.

Week 5-6: python learning

Weak 7-11: basic algorism and math knowledge learning, and realization

Week 12-13: essay editing and modifying codes.

- 小组讨论确定文献综述分工，并确定在11.29前完成初稿。

- 发展背景，zbh
- 流程 tsx
- 算法1 刘通
- 算法2 张hq
- ddl 11.29

Python
pytorch

Mnist data
Retrieved from
https://www.kesci.com/mw/project/5dc99685d78b45002cd9bbbe/dataset

tools | resource | theory | implementation | report

Bp
visualization

Group 17: AI Vtuber making project

王标、张倚凡（组长）、李康欣、何泽安、曾宇祺、 Zhang Kenneth

Target:
an agent that receives conversations in Chinese and output the most proper reply

Progress:

1. Front end(a web for input and training sample)
2. Resources (introduction/background/development of NLP; previous NLP models; technique for realizing NLP via Python)
3. Try (梯度下降法，反向传播算法)

Group 18：个性化推荐系统

谭雅静、刘思岑、Ooi Yee Jing、孟宇阳、杨锦涛（组长）

Indian Journal of Science and Technology

Evolution of Recommender Systems from Ancient Times to Modern Era: A Survey

Richa Sharma* and Rahul Singh

University Institute of Engineering, Chandigarh University, Gharuan, Mohali - 140413, Punjab, India;
richas30@yahoo.com, rahulsinghcse25@gmail.com

Summary of recommendation system development

LIU Liling[1]

[1] Beijing, Haidian District, Zhong Guan Cun South Street, No.5 Yard, Beijing Institute of Technology, China

Abstract. A recommendation system is an information service system that connects users and projects: for one thing, it helps users discover potential projects of interest; for another thing, it helps project providers to deliver projects to users who are interested in it. The recommendation system is a powerful system that can add value to the company or business. In the future, it will continue to be researched and developed to bring a better experience to users.

计算机工程与科学

Computer Engineering & Science

基于知识图谱嵌入与多神经网络的序列推荐算法

沈冬东，汪海涛，姜　瑛，陈　星

（昆明理工大学信息工程与自动化学院，云南 昆明）

CSDN

基于内容推荐算法详解(比较全面的文章)

知乎

数据科学—线DSFrontier

图算法 & 图分析

知乎

R语言中文社区

协同过滤推荐算法总结

知乎

大数据与人工智能

协同过滤推荐算法

Goup19：校园巴士线路优化

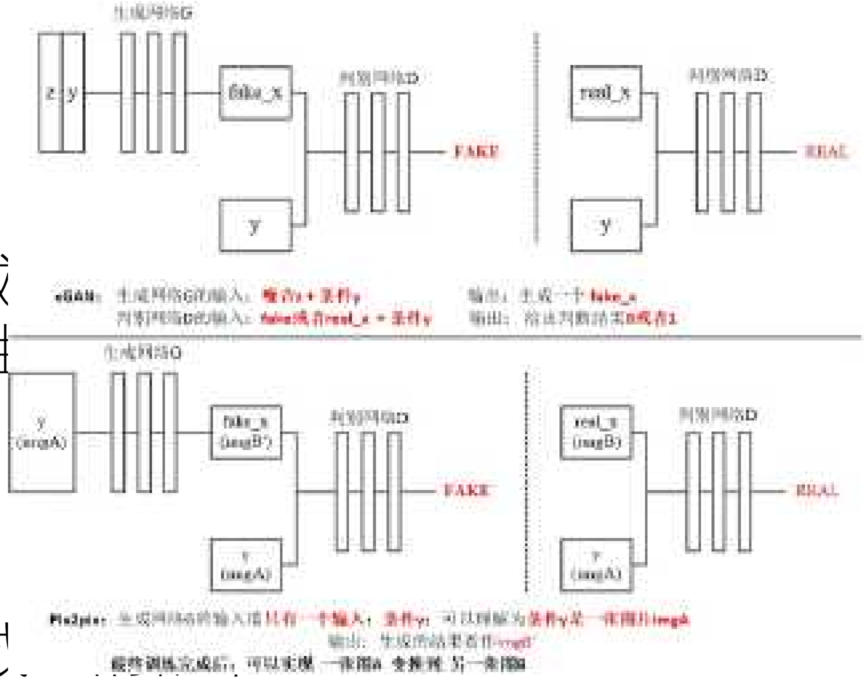
王祥辰、何鸿杰、吴子彧、樊青远（组长）、方琪涵、袁通

- 完成校园巴士位置实时显示的demo
- https://bus.sustcra.com/realtime-map-osm.html
- 完成对校园巴士车站，路径的标记。

给线稿上色的AI的算法研究

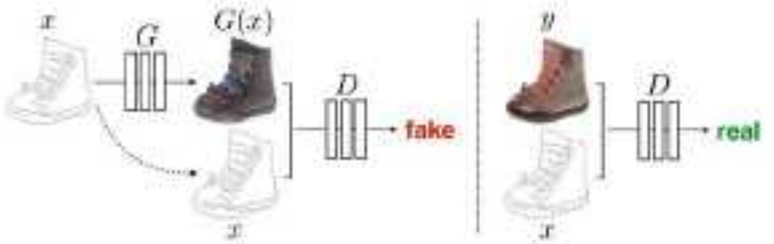
第20 组：韩晗（组长）、刘思语、赵晓蕾、陈松斌

- GAN（Generative Adversarial Nets）生成式对抗网络
  - 生成器G(Generator)：随机输入噪声，输出映射的图像即生成
  - 判别器D(Discriminator)：输入图像，输出判断输入图像来自生



- CGAN（Conditional GAN）条件生成式对抗网络
  - 生成器G：输入是噪声和标签，输出还是生成图；
  - 判别器D：输入是生成图、真实图以及标签，输出是来自生成

- Pix2pix模型
  - 生成器G： U-Net 结构
  - 判别器D： patchGAN 结构



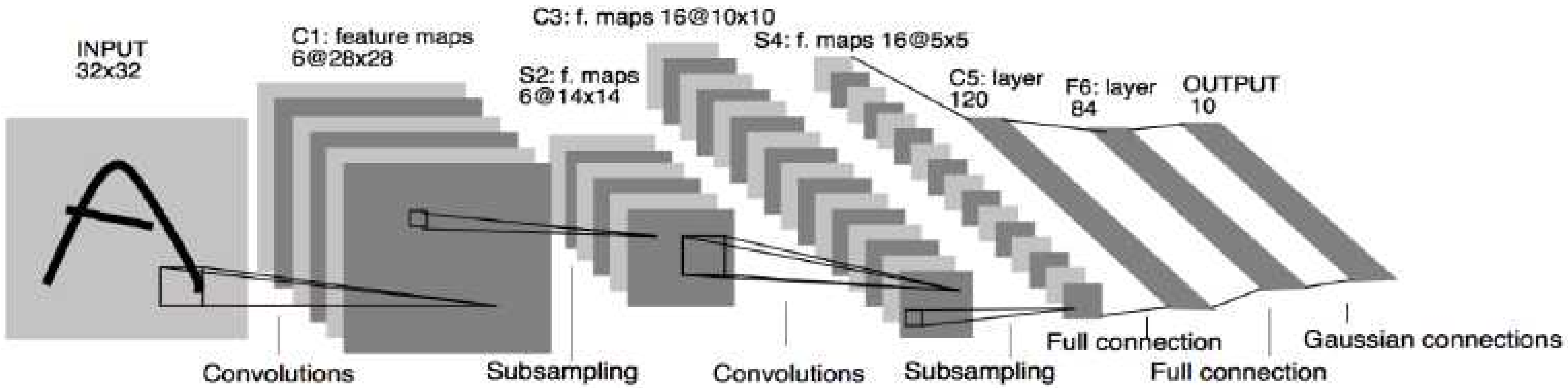
Group21：人工智能在皮肤癌诊断领域的可能性探索

刘宇欣、李修治（组长）、沈睿琦

	传统机器学习	支持向量机	卷积神经网络
准确率	+	++	+++
耗时（复杂性）	+	+++	+++
训练样本量	+	+	+++
图像特征提取能力	+	++	+++
解决的问题	简单线性问题	非线性问题（存在缺陷）	绝大部分问题

Group22：深度学习—卷积神经网络

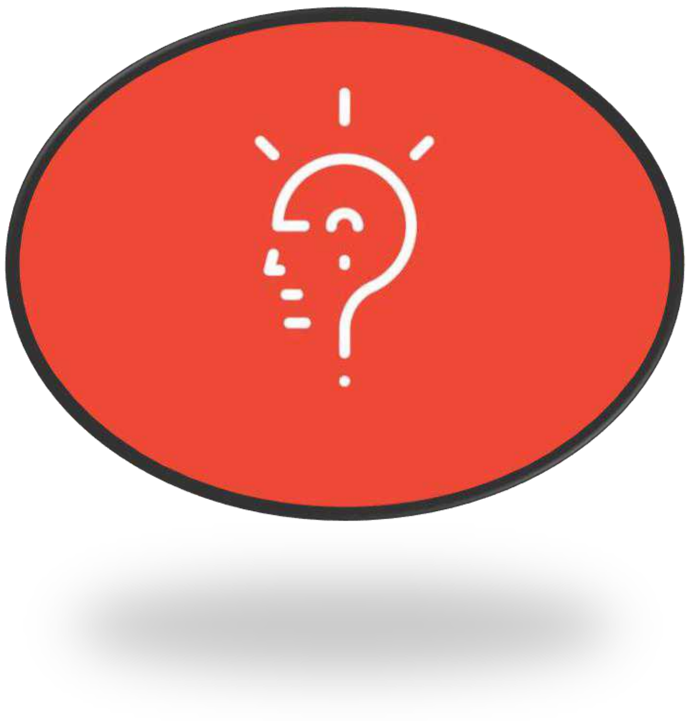
王晓轩



深度学习—卷积神经网络

- 深度学习：　深度学习是机器学习研究中的一个新的领域，其动机在于建立、模拟人脑进行分析学习的神经网络，它模仿人脑的机制来解释数据，例如图像，声音和文本。深度学习是无监督学习的一种。
- 卷积神经网络是一种基于生物视觉感知理论进行设计的神经网络
- 输入层 卷积层 池化层 全连接层 和分类器

Any Question?



Perceptron

3 4

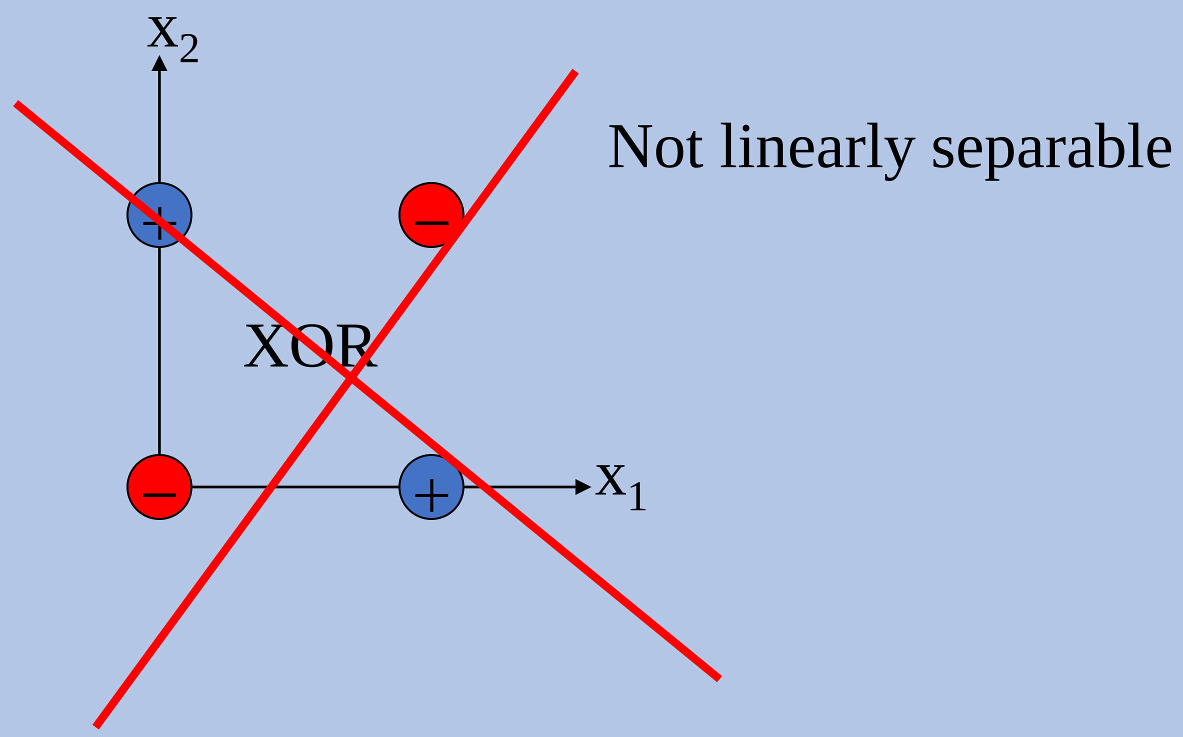
1 Perceptron

2 Perceptron Learning

3 ADALINE

4 Limitation of Perceptron

Perceptron XOR Problem



Minsky & Papert (1969)

Bad News: Perceptrons can only represent linearly separable functions.

Perceptron XOR Problem Proof

- Consider a threshold perceptron for the logical XOR function (two inputs):

$$w_1x_1 + w_2x_2 > T$$

- XOR can be represented as:

	x1	x2	label
1	0	0	0
2	1	0	1
3	0	1	1
4	1	1	0

Given our examples, we have the following inequalities for the perceptron:

From (1) $0 + 0 \le T$ → $T \ge 0$

From (2) $w_1 + 0 > T$ → $w_1 > T$

From (3) $0 + w2 > T$ → $w_2 > T$

From (4) $w1 + w2 \le T$

$w1 + w2 > 2T$

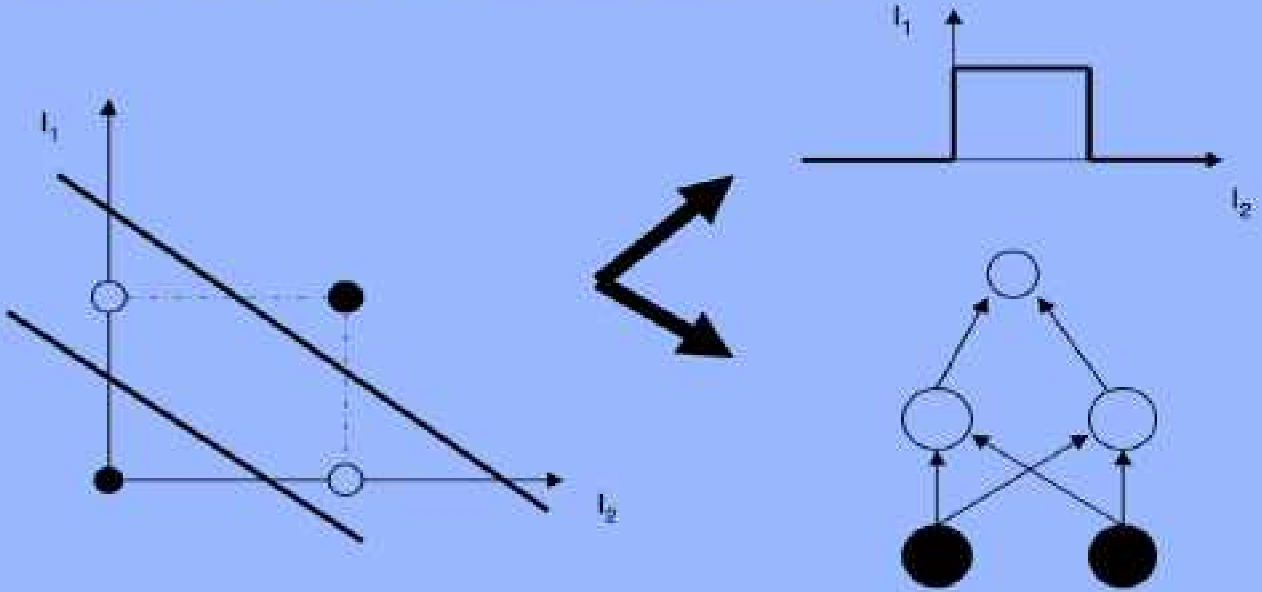
contradiction

So, XOR is not linearly separable

XOR Decision Surface (Boundary)

The difficulty in dealing with XOR is rather obvious. We need two straight lines to separate the different outputs/decisions:

XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



Solution: either change the transfer function so that it has more than one decision boundary, or use a more complex network that is able to generate more complex decision boundaries.

Marvin Minsky (1927-2016)

- In 1969, Minsky and Seymour Papert published “Perceptrons,” a book that assailed Rosenblatt’s work and, essentially, sealed its fate. The following year, Minsky won the A.M. Turing Award – computing’s highest honor.

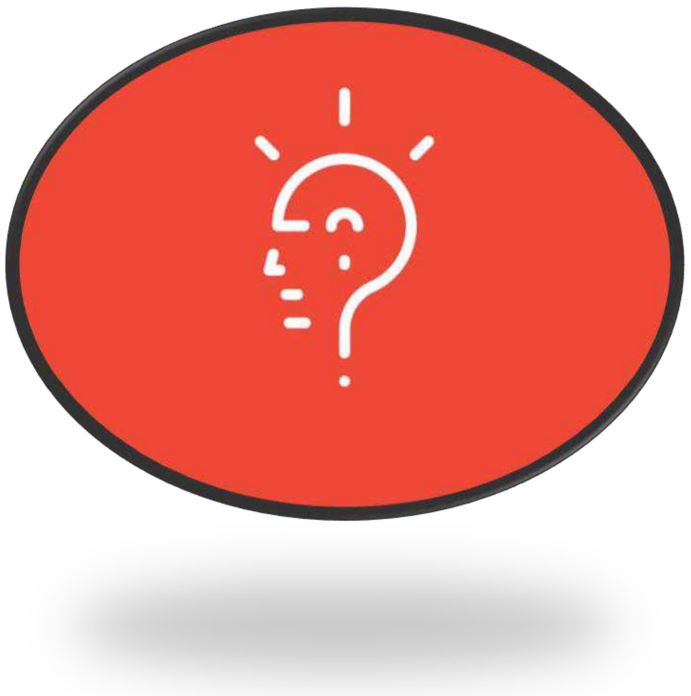
- The problem was, Rosenblatt’s perceptron had only one layer, while modern neural networks have hundreds.

- What Rosenblatt wanted was to show the machine can recognize objects. His algorithm is still fundamental to how we’re training deep networks today.

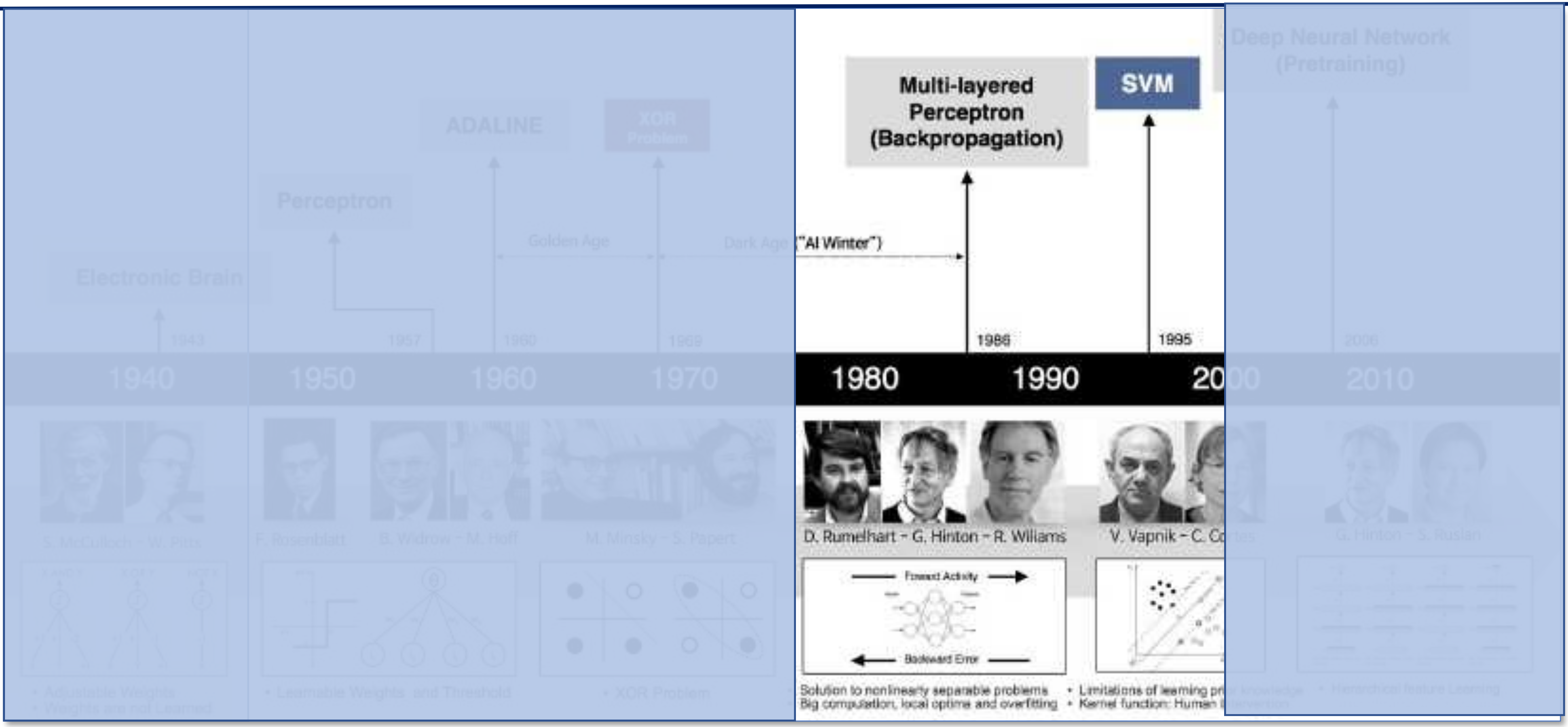
AI Algorithm Development – Further Close Look (in Early AI Days)

- Perceptron, ADALINE and their limitation
  - Algorithm Perceptron not able to implement basic none-linear function XOR
  - Machine translations: NRC, 1950s', translation from English to Russian
    - "The spirit is willing but the flesh is weak" produced
    - "The vodka is good but the meat is rotten"
    - Requires knowledge to establish content
- Funding for the industry was slashed

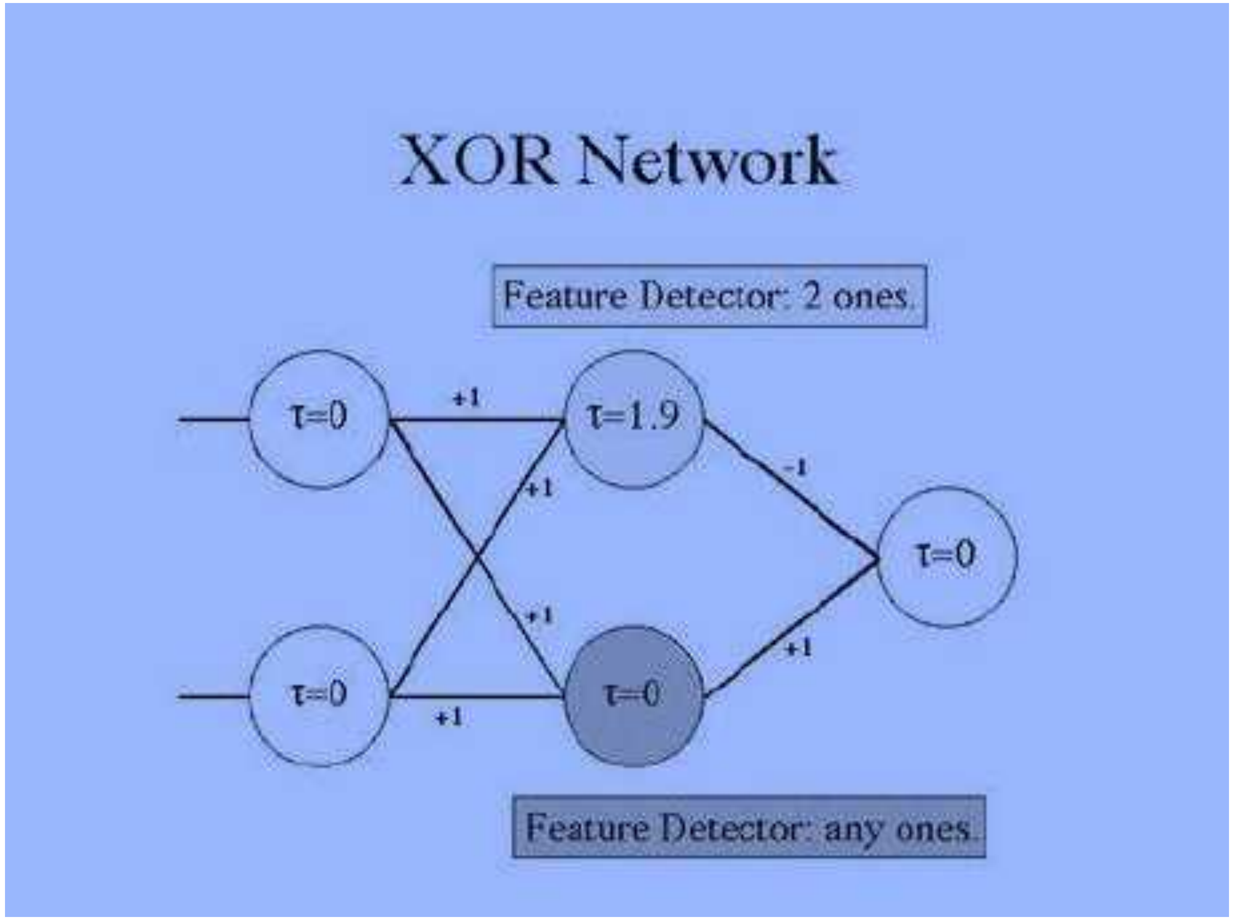
Any Question?



AI algorithm Developments – Machine Learning

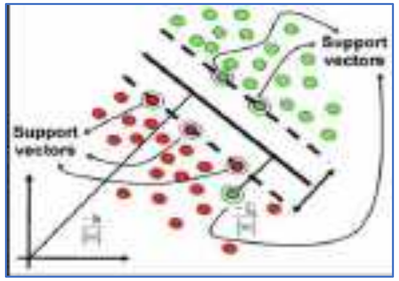


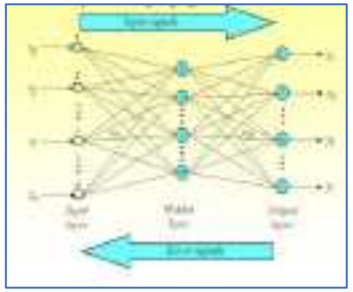
XOR in Multiple Layer Perceptron

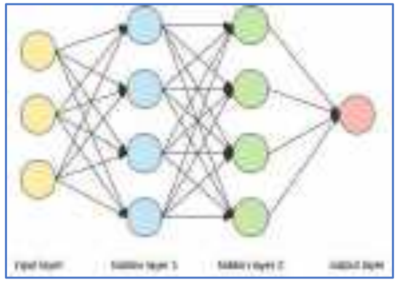


AI Algorithm Development –Machine Learning

- Back Propagation
- Support Vector Machine
- Machine Learning
- Knowledge
    - Knowledge Representation
    - Knowledge Graph
    - Knowledge Tree
    - Knowledge Search
    - Deep Blue







Machine Learning

3 5

1. Back Propagation
2. Supporter Vector Machine
3. Machine Learning
4. Knowledge

3 Key Network Components of Neural Network

Artificial neural networks (ANNs), usually simply called neural networks(NNs), are computing systems vaguely inspired by the biological neural networks that constitute animal brains.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

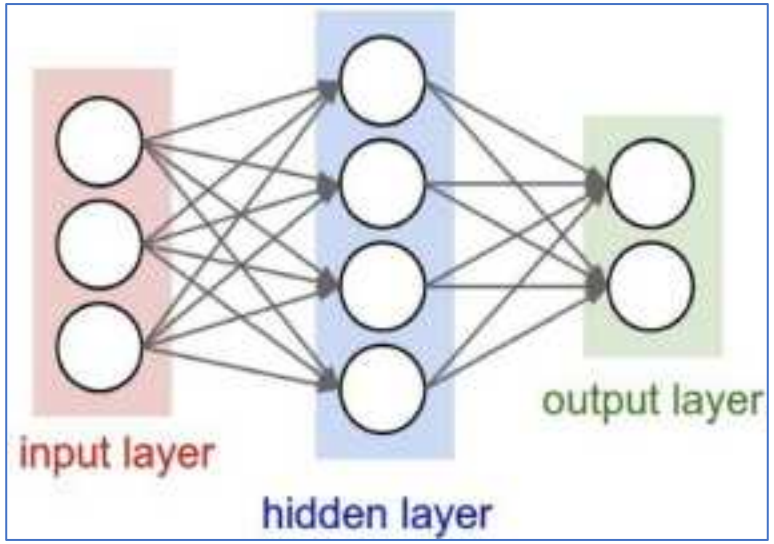
The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold.

Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

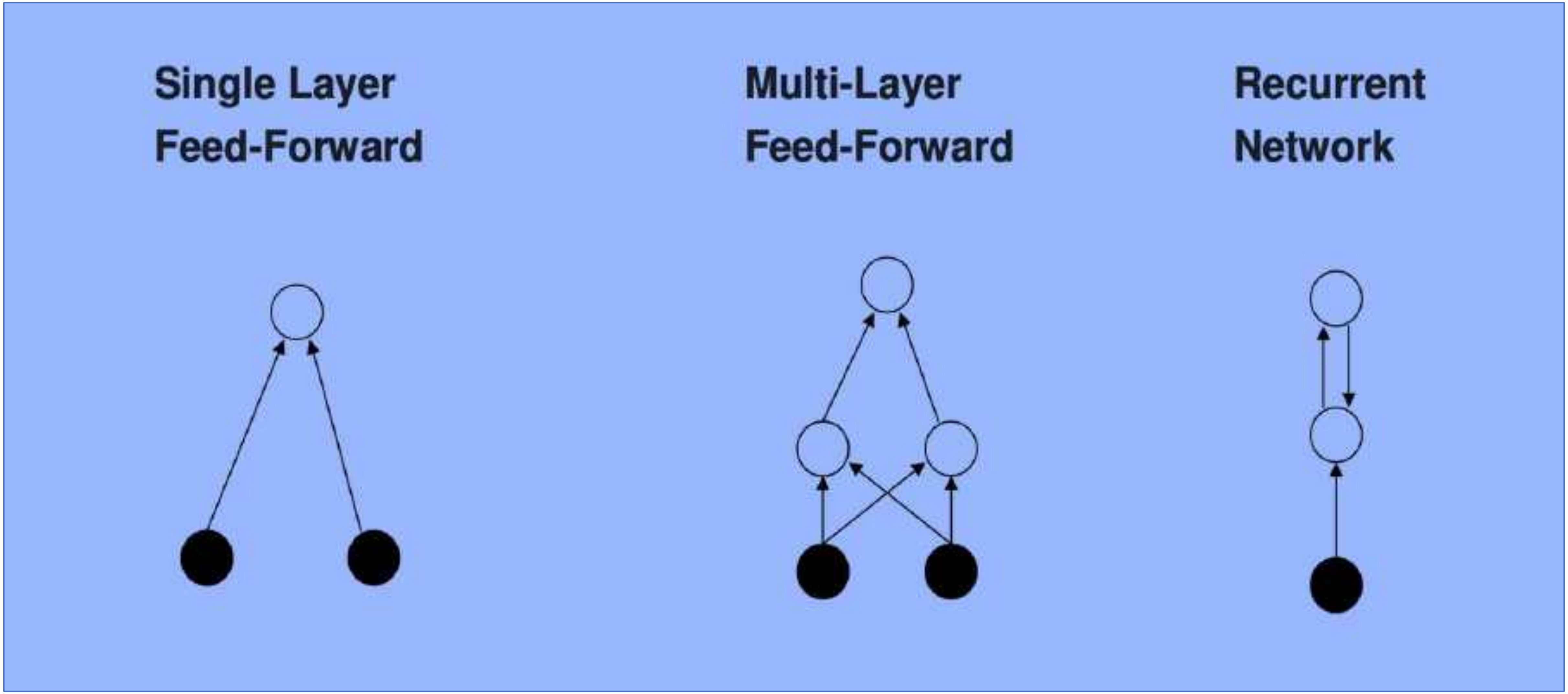
Wikipedia

3 Key Network Components of Neural Network

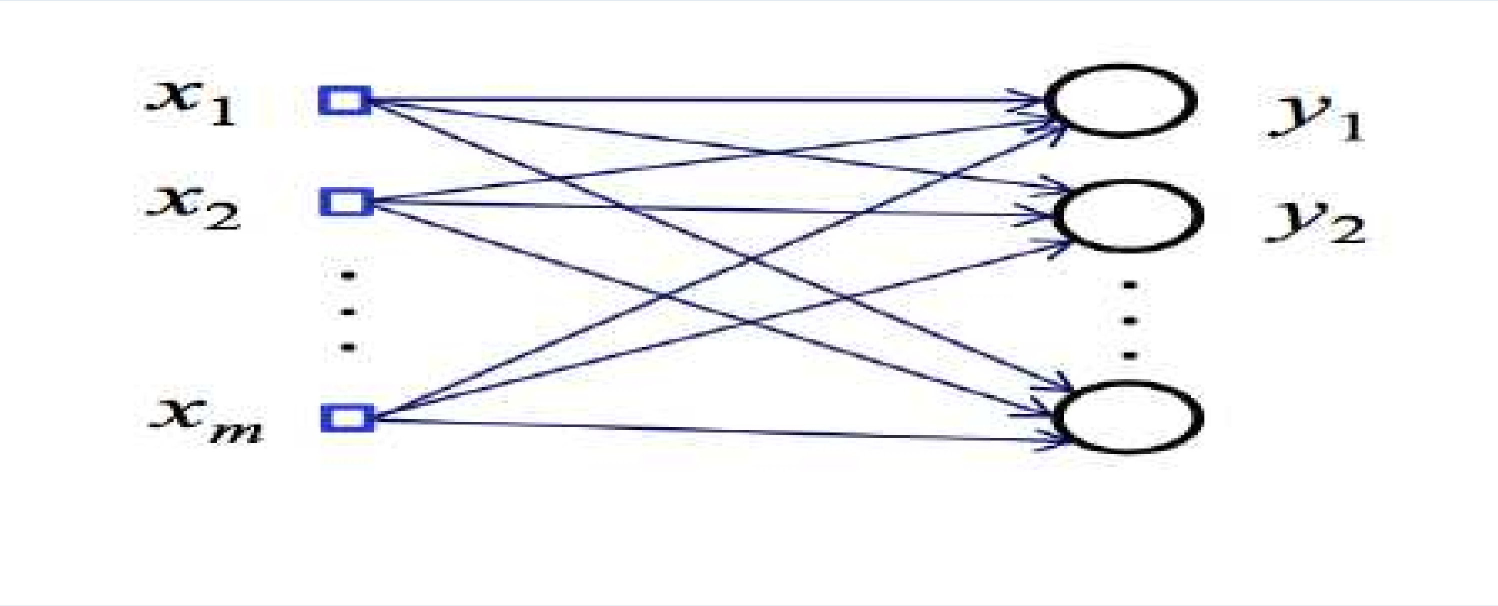
- Network Architecture
- Transfer Function
- Learning Rule



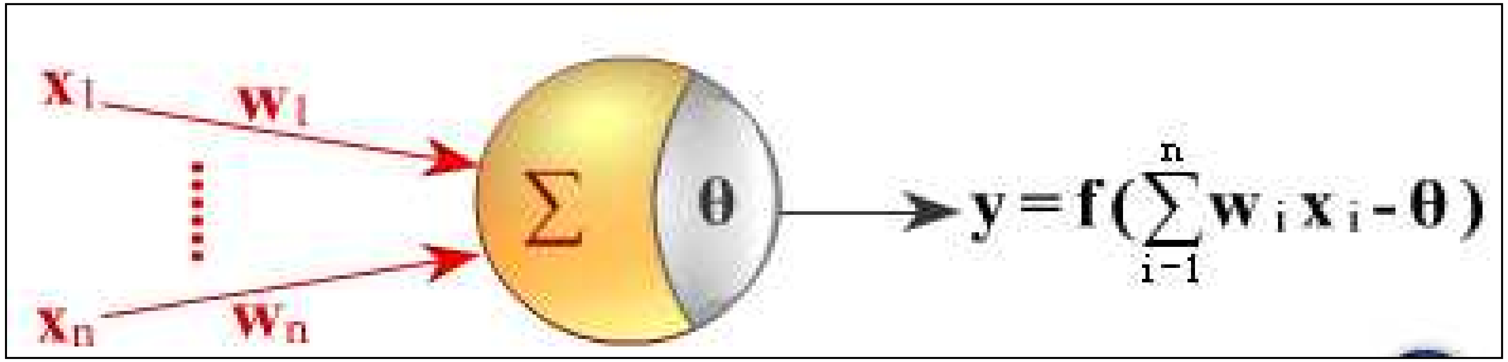
Network Structure Extension From Perceptron



Recall: General Network Architecture 1 – One Layer Feed Forward Network

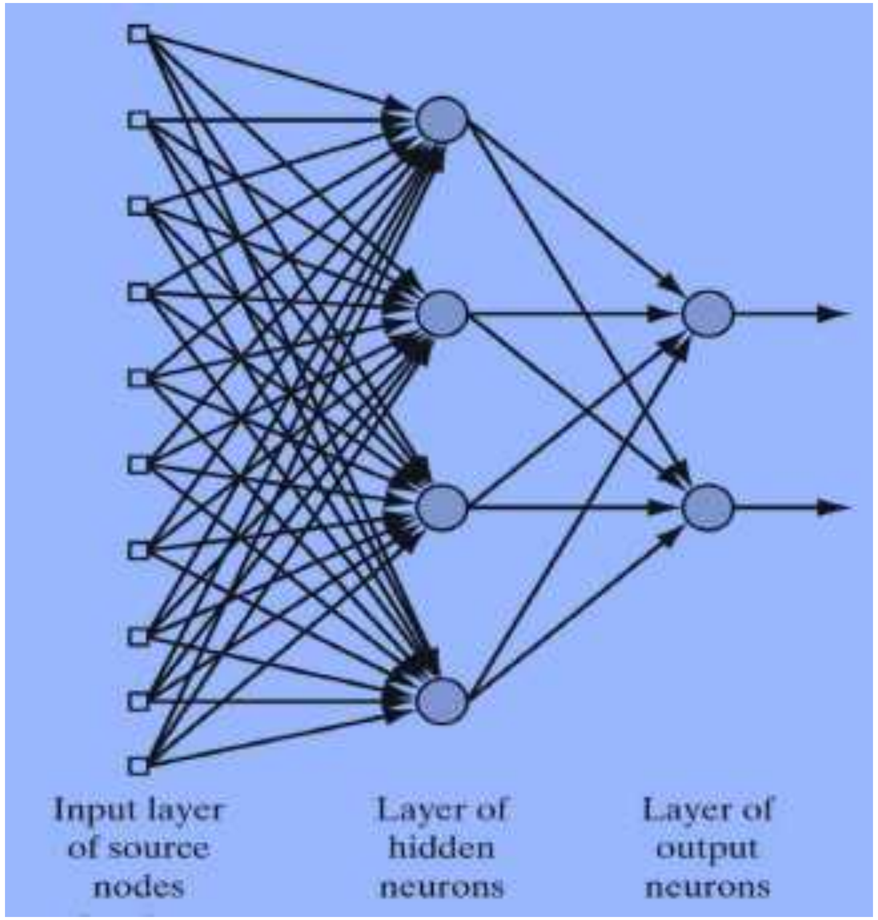


One Layer Feed Forward NN – A general form of single neuron Perceptron

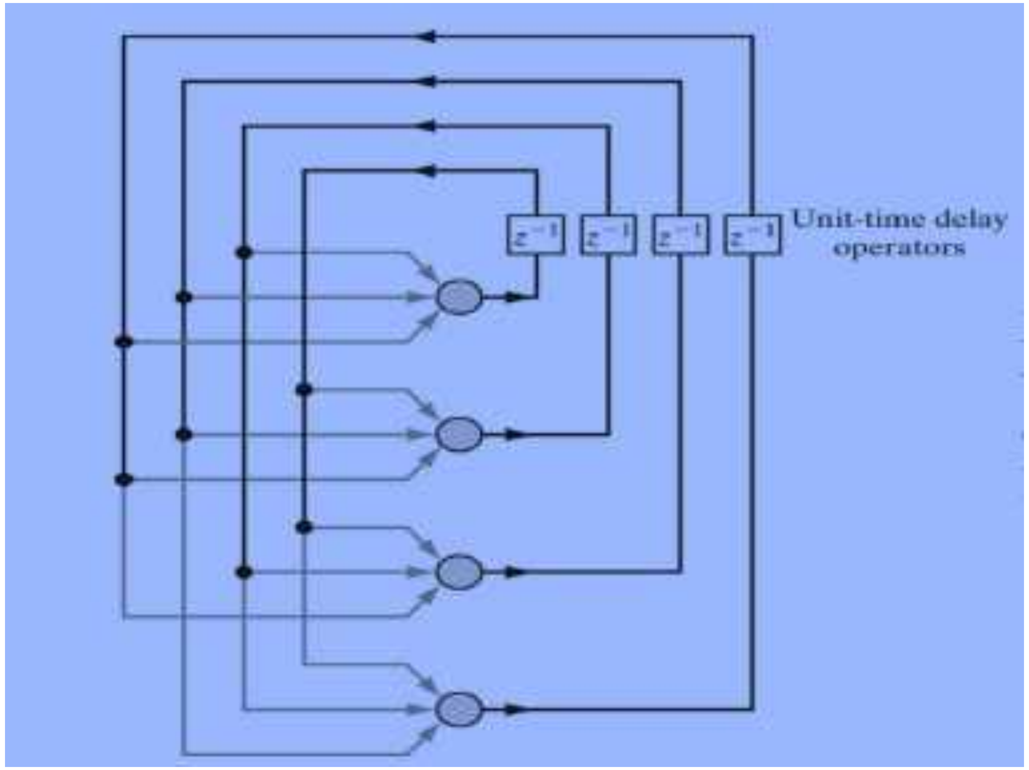


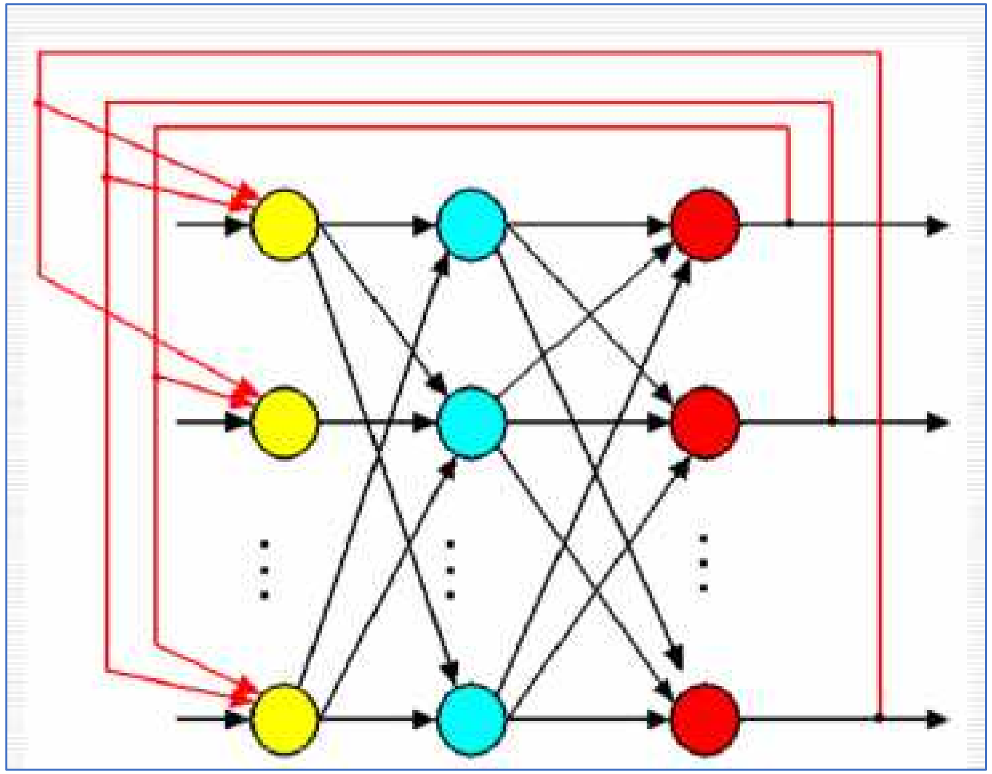
General Network Architecture 2 – Multi Layer Feed Forward Network

- View an NN as a connected, directed graph, which defines its architecture
  - Feed forward nets: loop-free graph
  - (10-4-2 2 layer network)



Network Architecture 3 – Recurrent Network

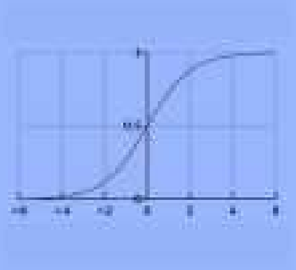




The network input typically sets the initial condition of the output layer

Activation (Transfer) Function Extension

Activation Function



In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs. A standard integrated circuit can be seen as a digital network of activation functions that can be "ON" (1) or "OFF" (0), depending on input. This is similar to the behavior of the linear perceptron in neural networks. However, only nonlinear activation functions allow such networks to compute nontrivial problems using only a small number of nodes, and such activation functions are called nonlinearities.

Wikipedia

Activation Functions

Sigmoid

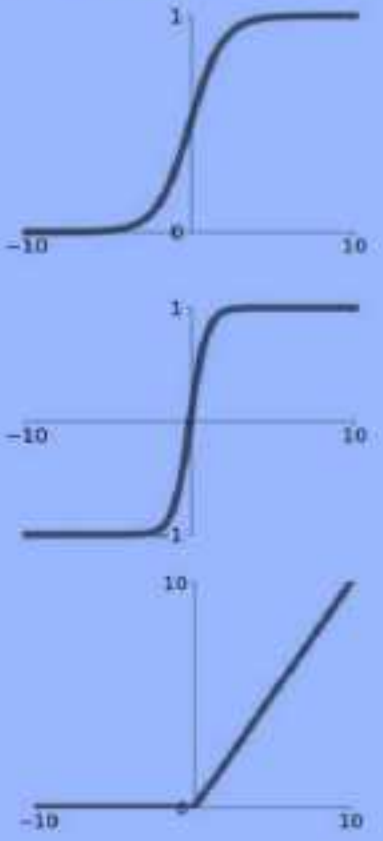
$\sigma(x) = \frac{1}{1+e^{-x}}$

tanh

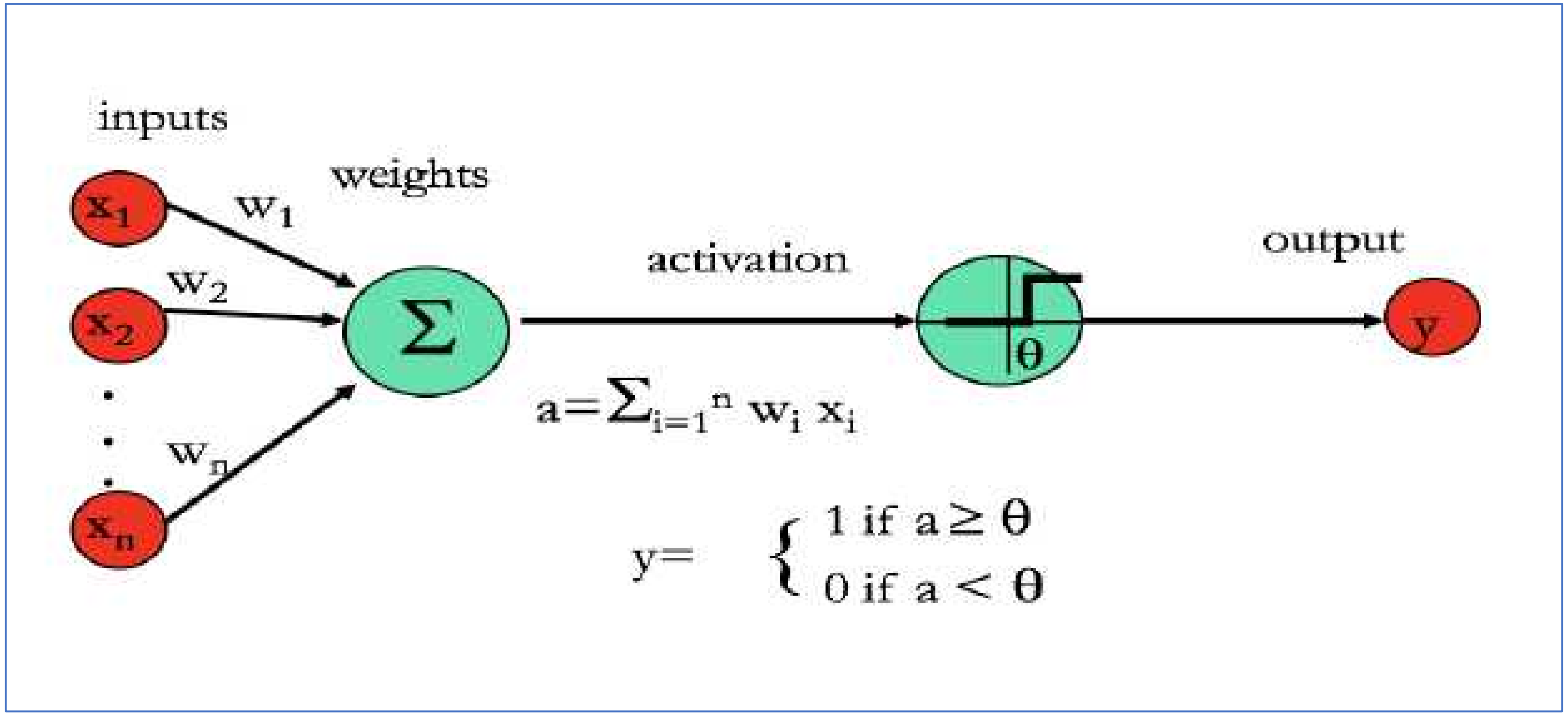
$\tanh(x)$

ReLU

$\max(0, x)$

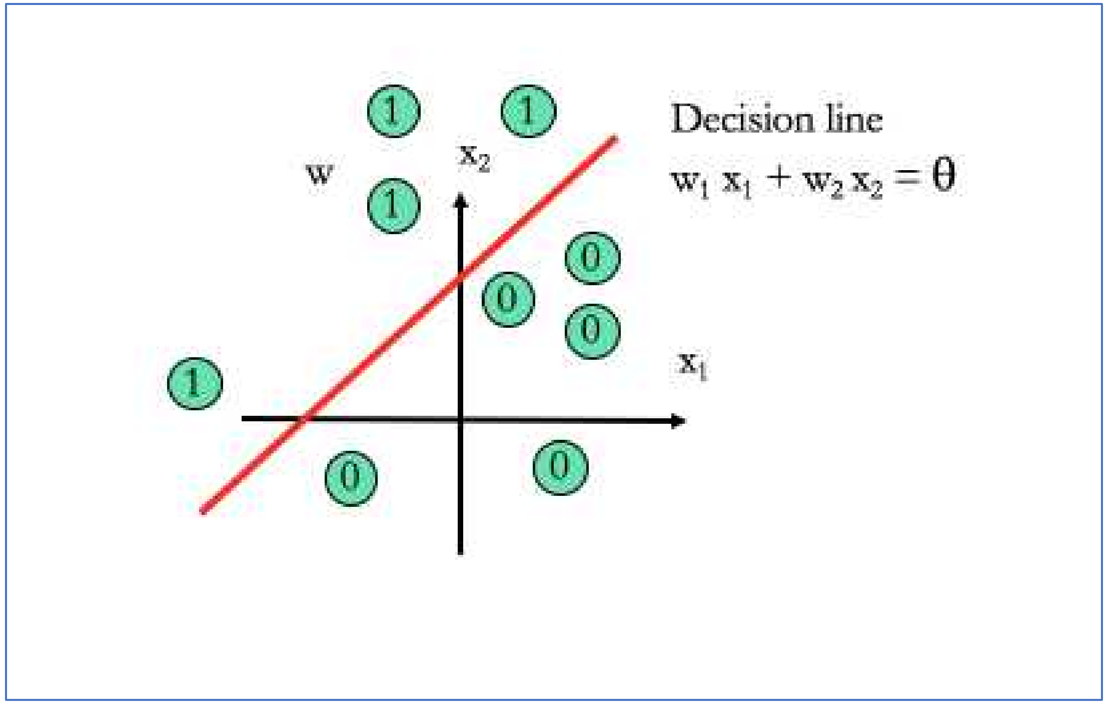


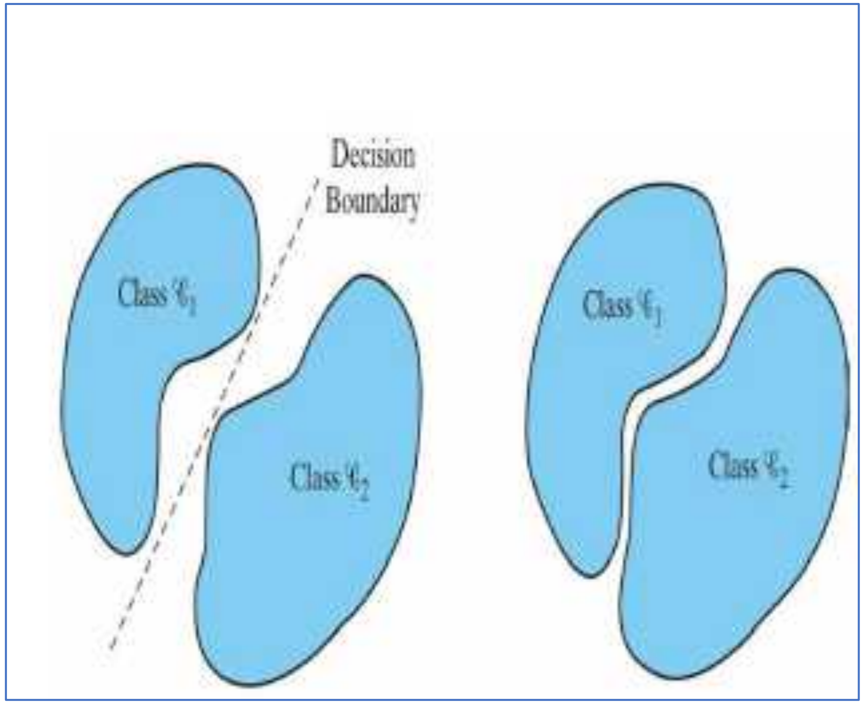
Recall: TLU Transfer Function



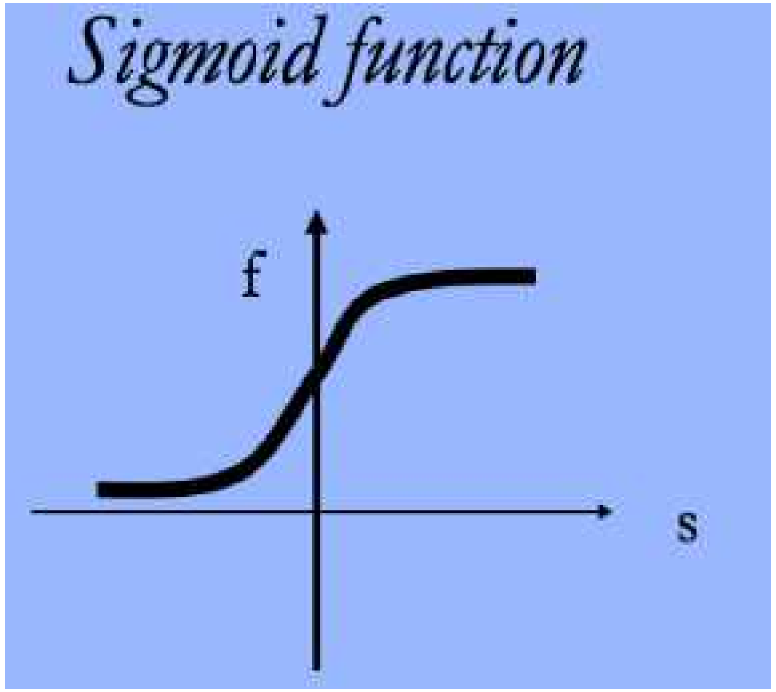
Threshold Logic Unit (TLU)

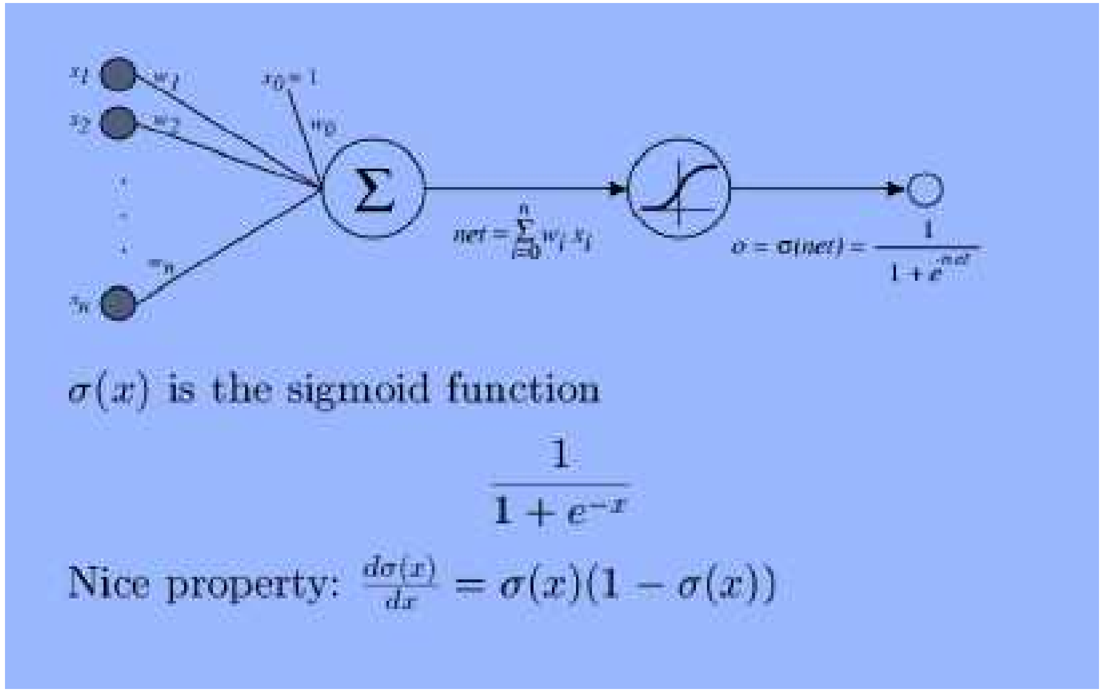
Recall: Decision Boundary





Sigmoid Transfer Function





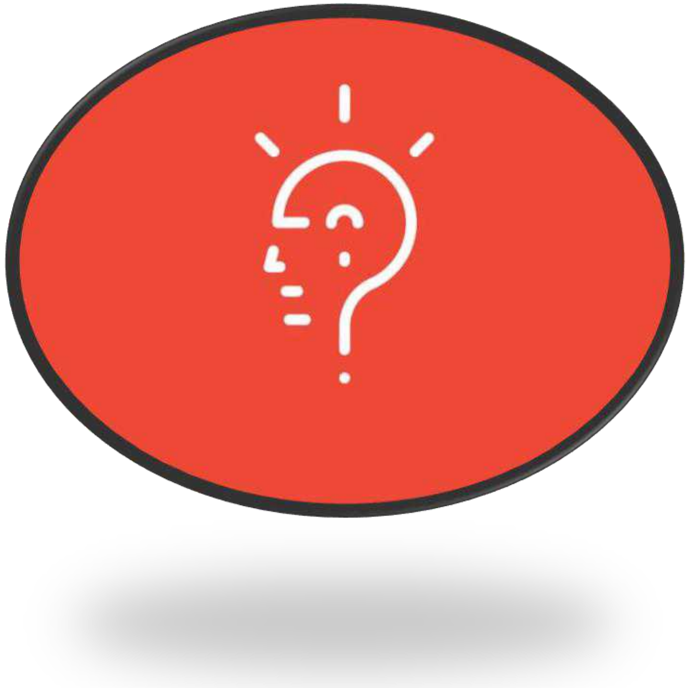
Sigmoid Unit

Learning Rule Extension – Back Propagation

In machine learning, backpropagation(backprop, BP) is a widely used algorithm in training feedforward neural networks for supervised learning. Generalizations of backpropagation exists for other artificial neural networks (ANNs), and for functions generally. These classes of algorithms are all referred to generically as "backpropagation". In fitting a neural network, backpropagation computes the gradient of the loss function with respect to the weights of the network for a single input–output example, and does so efficiently, unlike a naive direct computation of the gradient with respect to each weight individually. This efficiency makes it feasible to use gradient methods for training multilayer networks, updating weights to minimize loss; gradient descent, or variants such as stochastic gradient descent, are commonly used. The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule.

Wikipedia

Any Question?



No Homework Week

CS 103 -09
Machine Learning and BP

Jimmy Liu 刘江

2020-11-20