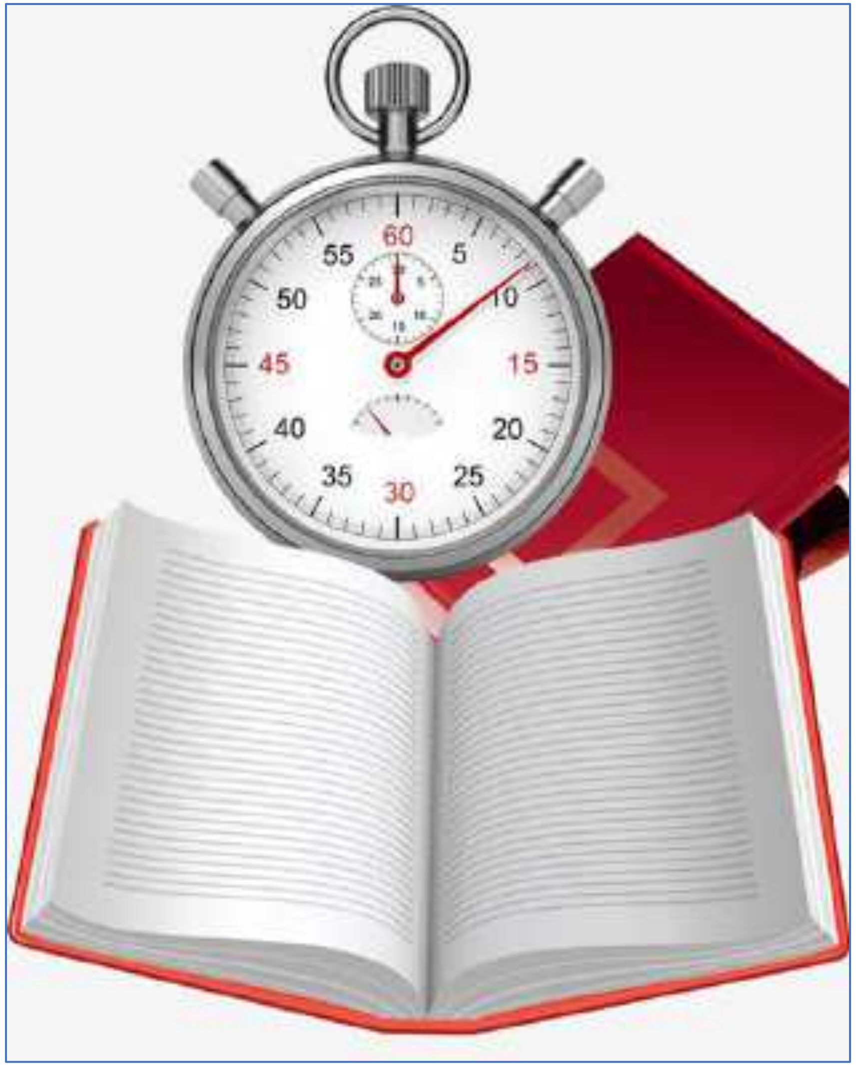
CS 103 -08
Perceptron Learning and ADALINE

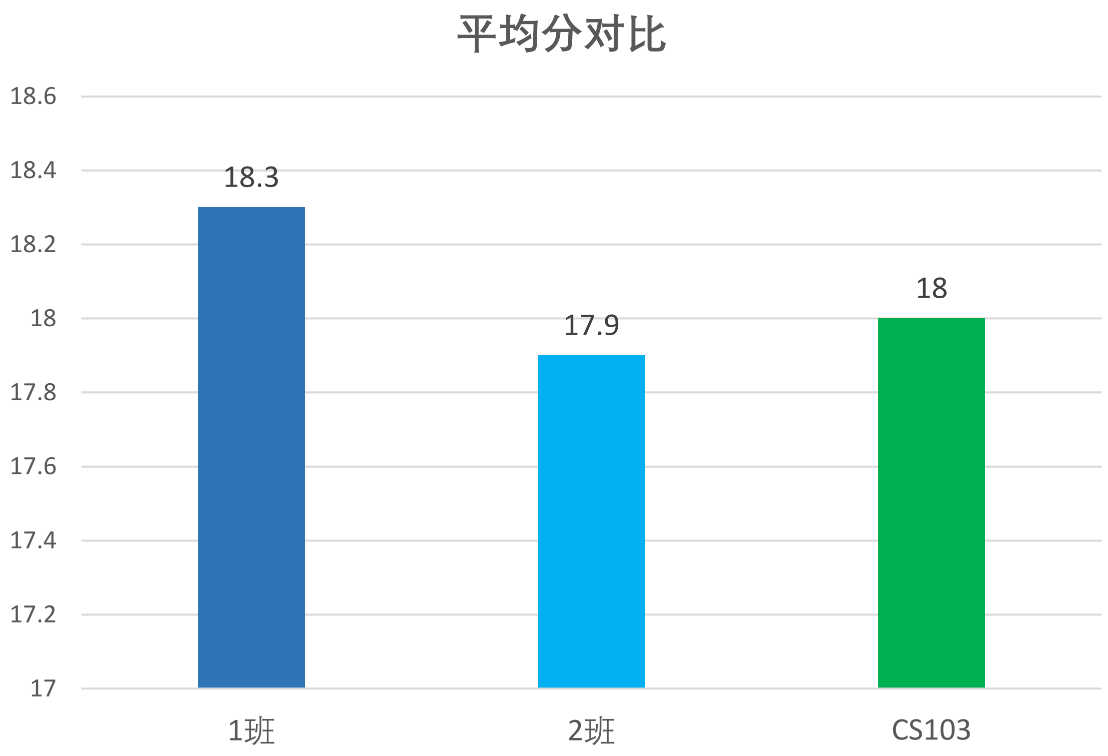
Jimmy Liu 刘江

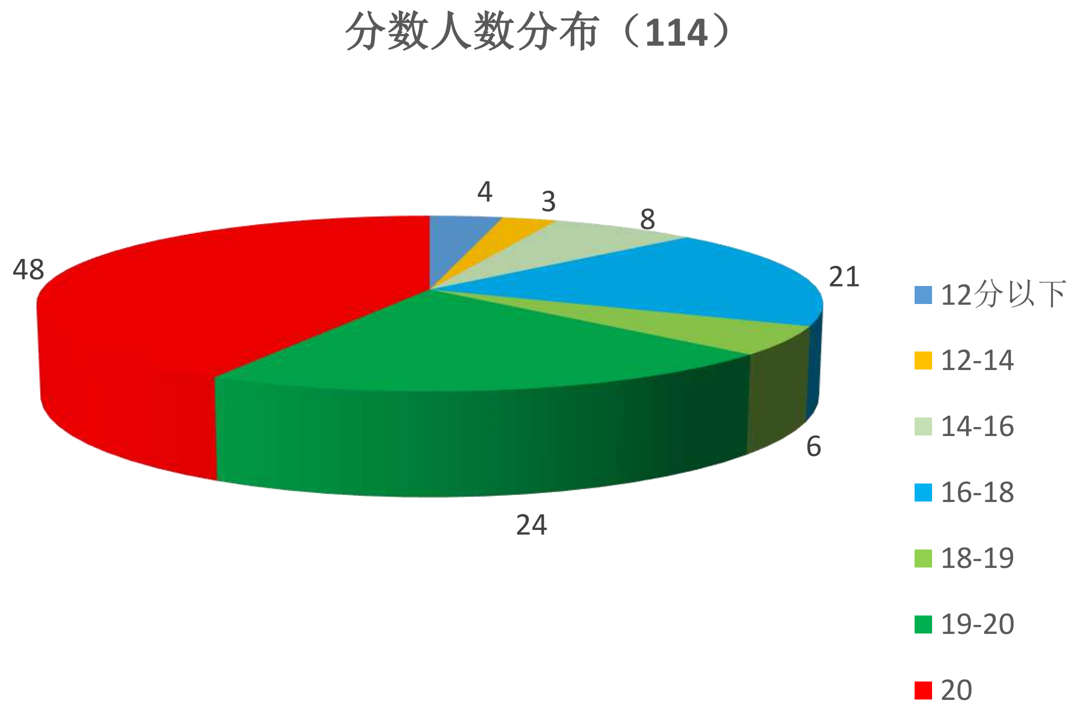
2020-11-6

Mid-Term Test Review

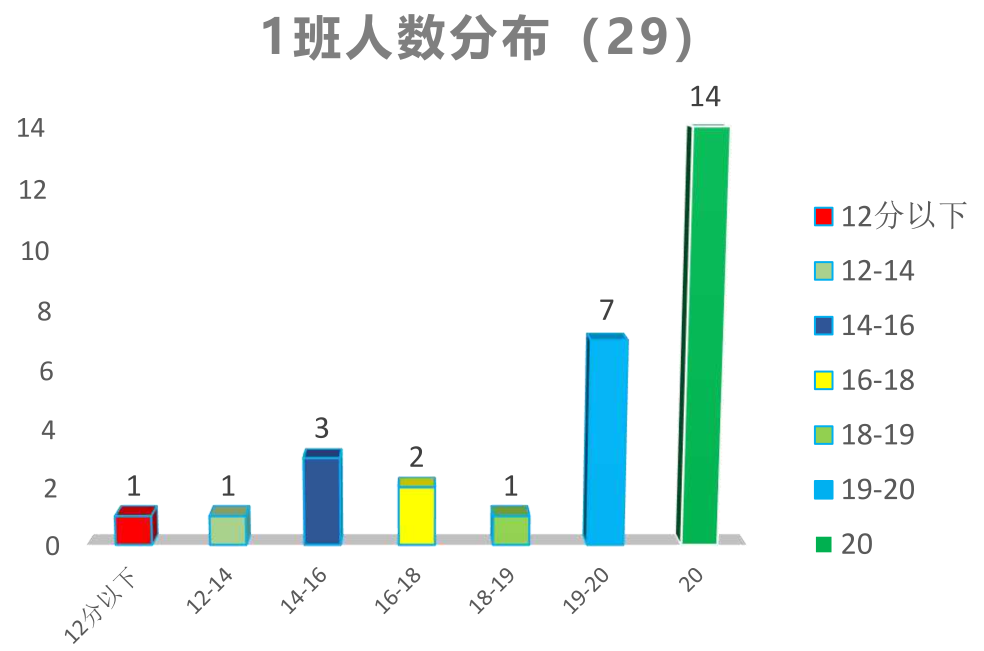


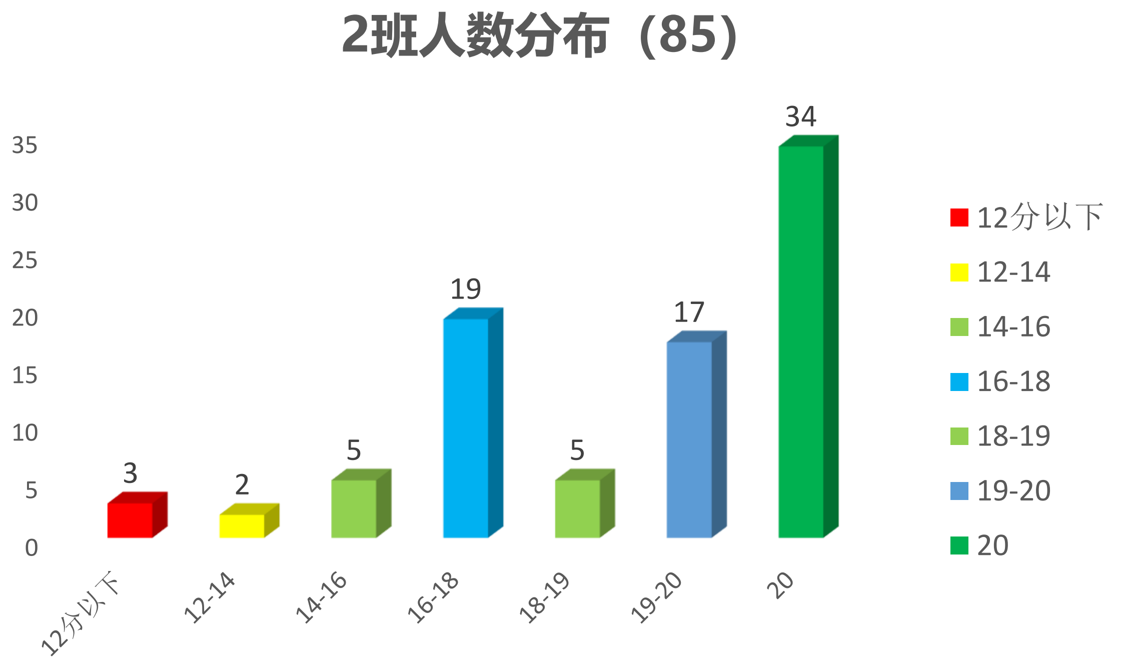
期中试卷总结


平均分对比
18.6
18.4
18.2
18
17.8
17.6
17.4
17.2
17
18.3
17.9
18
1班
2班
CS103


分数人数分布（114）
4
3
8
21
6
24
48
12分以下
12-14
14-16
16-18
18-19
19-20
20

期中试卷总结


1班人数分布（29）
14
12
10
8
6
4
2
0
1
1
3
2
1
7
14
12分以下
12-14
14-16
16-18
18-19
19-20
20
■ 12分以下
■ 12-14
■ 14-16
■ 16-18
■ 18-19
■ 19-20
■ 20


2班人数分布（85）
35
30
25
20
15
10
5
0
3
2
5
19
5
17
34
12分以下
12-14
14-16
16-18
18-19
19-20
20
■ 12分以下
■ 12-14
■ 14-16
■ 16-18
■ 18-19
■ 19-20
■ 20

期中试卷总结-上午

Course Name: Introduction of Artificial Intelligence 2020 Fall Semester Midterm Exam paper

南方科技大学

Course Name: Introduction of Artificial Intelligence

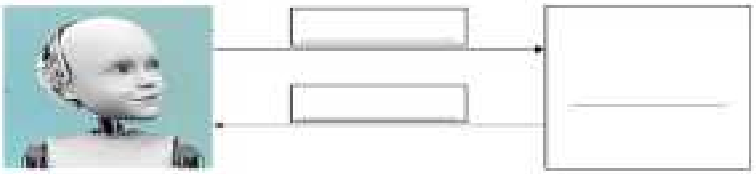
Dept.: Department of Computer Science and Engineering

Exam Duration: 50 minutes Exam Paper Setter: Liu Jiang

Question No.	1	2	3	4	5	6	7	8	9	10
Score	7	3	10							

This exam paper contains 3 questions and the score is 20 in total. (Please hand in your exam paper, answer sheet, and your scrap paper to the proctor when the exam ends.)

1. Please fill in the blanks with names of the three components for an agent, and use robot agent as example to describe how an agent functions. (7 marks)



2. Please describe what is a computer algorithm. (3 marks)

3. (10 marks)

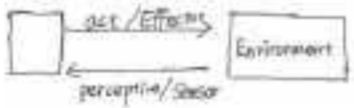
(1) Please list or draw the six components of the perceptron. (3 marks)

(2) Please write the formula of Perceptron Learning Rule (PLR) and explain the meaning of each parameter. (7 marks)

Mid-Term Test of CS103-Introduction to AI

Student ID: 11910858 Full Name: 刘俊成

1.

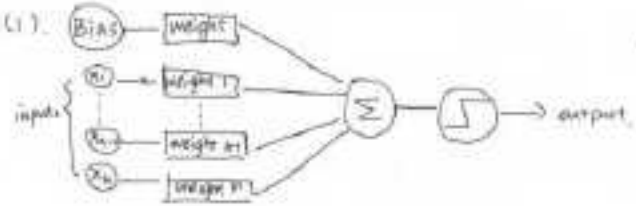


Robot agent { Sensor: Camera 来获取输入/图像等数据
Effector: 进行状态改变，执行操作

机器人通过摄像头等感知器获得周围环境相关的数据，经过处理后，将结果表现在机器人的行为/状态改变上和改变环境。

2. ① 有确定的步骤过程
② 在有限的时间/操作内完成
③ 获得的是正确的结果

3. (1)



(2) $W_i \leftarrow W_i + (\eta * x_i * E)$

New weight, weight, learning rate, input, Error

期中试卷总结-下午

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Course Name: Introduction of Artificial Intelligence

Dept.: Department of Computer Science and Engineering

Exam Duration: 50 minutes Exam Paper Setter: Liu Jiang

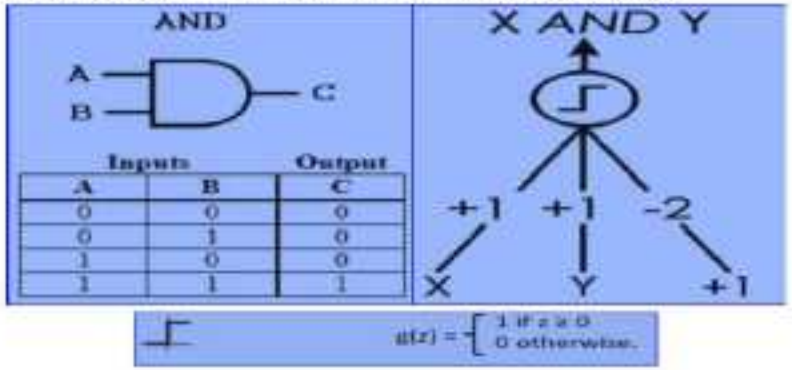
Question No.	1	2	3	4	5	6	7	8	9	10
Score	4	6	10							

This exam paper contains 3 questions and the score is 20 in total. (Please hand in your exam paper, answer sheet, and your scrap paper to the proctor when the exam ends.)

1. Please describe the difference between McCulloch and Pitts (MCP) Neuron and Perceptrons. **(4 marks)**

2. Please describe in details what is narrow AI, general AI and super AI, and please also give an example of each type and explain the reason why it belongs to that type. **(6 marks)**

3. Prove the following MCP neuron is an "AND" Artificial Neuron **(10 marks)**



期中试卷总结

Mid-Term Test of CS103-Introduction to AI

Student ID: 11810236 Full Name: [illegible]

1. Differences:

① MCP Neuron: weights are adjusted but not learned.
Perceptrons: weights are learned. And initially ~~weighted~~ weights are randomly selected among

② MCP Neuron: Its outputs are 0 or 1.
Perceptrons: Its outputs can be 0, 1, or -1.

③ MCP Neuron: It has only one output.
Perceptrons: It has many outputs and outputs don't share weights.

④ MCP Neuron: Weights can be negative.
Perceptrons: weights are ~~are~~ positive, among 0 to 1.

⑤ MCP Neuron is a single neuron model ~~for ...~~
Perceptrons are a simple layer feed-forward neuronal network, which is more complicated than MCP

2. ① Narrow AI definition: Narrow AI ~~is~~ the ability to assist with or take over specific tasks.
Example: Self-driving system can help driving a car with its intelligence without humans
Reason: Self-driving system can perceive traffic conditions and accordingly acts, make decisions to drive the car. Driving a car is a specific task which ~~the self~~ self-driving system can take

② General AI definition: General AI ~~is~~ is the ability to take knowledge from one domain and transfer another domain.
Example and reasons ~~Example~~: Alpha-Go, can take knowledge from mathematics and other human p... games and other appli...
And it can transfer these knowledge to ~~other~~

③ Super AI definition: Super AI ~~...~~ is an order of magnitude that is smarter than
Example: Not produced yet, maybe it can ~~be~~ a robot that can learn and compute by itself and even ~~feel people~~ has consciousness, ~~can~~ and can feel people
Reason: If something is smarter than human, then it ~~should~~ have a stronger ability in thinking and learning, and ~~...~~ can feel ~~your~~ people with its intelligence ~~...~~ Such a robot can learn, compute itself which is almost the same as human. And it can feel human which may indicate that it is smarter than human.

3.

Input A	Input B	Bias	W_1	W_2	W_3	Transfer function
0	0	1	1	1	-2	g(z)
0	1	1	1	1	-2	g(z)
1	0	1	1	1	-2	g(z)
1	1	1	1	1	-2	g(z)

$z = \sum = A \cdot W_1 + B \cdot W_2 + Bias \cdot W_3$

$0 \cdot 1 + 0 \cdot 1 + 1 \cdot (-2) = -2$

$0 \cdot 1 + 1 \cdot 1 + 1 \cdot (-2) = -1$

$1 \cdot 1 + 0 \cdot 1 + 1 \cdot (-2) = -1$

$1 \cdot 1 + 1 \cdot 1 + 1 \cdot (-2) = 0$

$g(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases}$

Output function: g(z)	A AND B
0	0
0	0
0	0
1	1

Group Project Update

小组项目-调研综述进展汇报

助教：张洋

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

助教：肖尊杰

6. 基于MRI图像的阿尔茨海默症分类：董廷臻、郑英炜（组长）、李博翙、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析：唐云龙、刘叶充、刘旭坤、马卓远、陈子蔚（组长）、江欣乐、陈浩然

助教：杨冰

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）、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

助教：章晓庆

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

小组项目-调研综述进展汇报

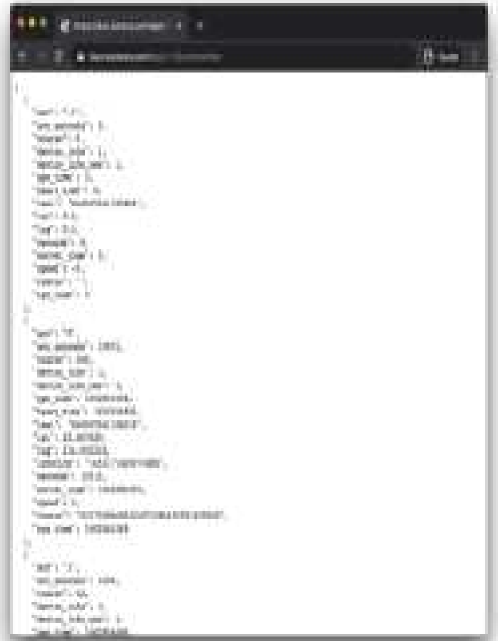
序号	题目	成员
1	AI+斗地主	孙永康、李怀武、胡鸿飞、吴一凡、杨光、张习之、金肇轩（组长）、于佳宁
2	AI+五子棋	周贤玮、韩梓辰（组长）、赵云龙、张坤龙、夏星晨
3	High Score Gamer	易辰朗、许天淇、黄北辰（组长）、赵思源、朱佳伟、宛清源
4	AI application on diabetes	周钰奇、李仪轩、董叔文、湛掌、胡钧淇（组长）、裴鸿婧
5	AI in lung cancer	夏瑞浩、李悦明、龚颖璇、吴云潇潇（组长）、姜欣瑜、王英豪
6	基于MRI图像的阿尔茨海默症分类	董廷臻、郑英炜（组长）、李博翱、朱嘉楠、李杨燊
7	AI Applications in 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	何忱、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

校园巴士线路优化小组

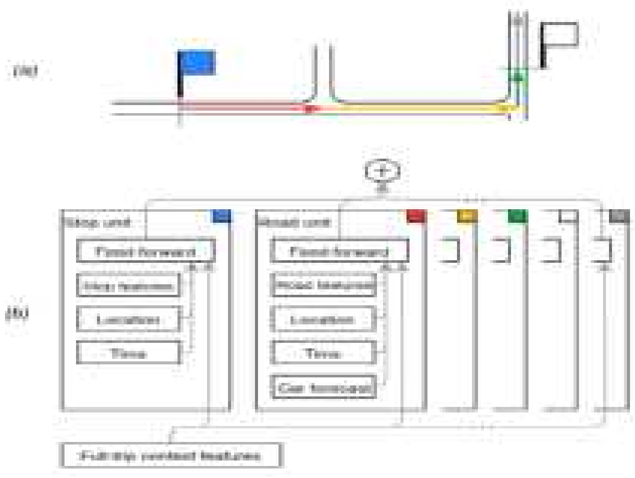
第19组：王祥辰、何鸿杰、吴子彧、樊青远（组长）、方琪涵、袁通

- 完成API server的搭建，开始收集数据（收集每日6:50-23:00的巴士运行数据。



阅读相关文章（bus prediction algorithm）

- E.g. https://ai.googleblog.com/2019/06/predicting-bus-delays-with-machine.html (这篇的 algorith 被用在了 google 地图中)
- Neural sequence models (RNN)

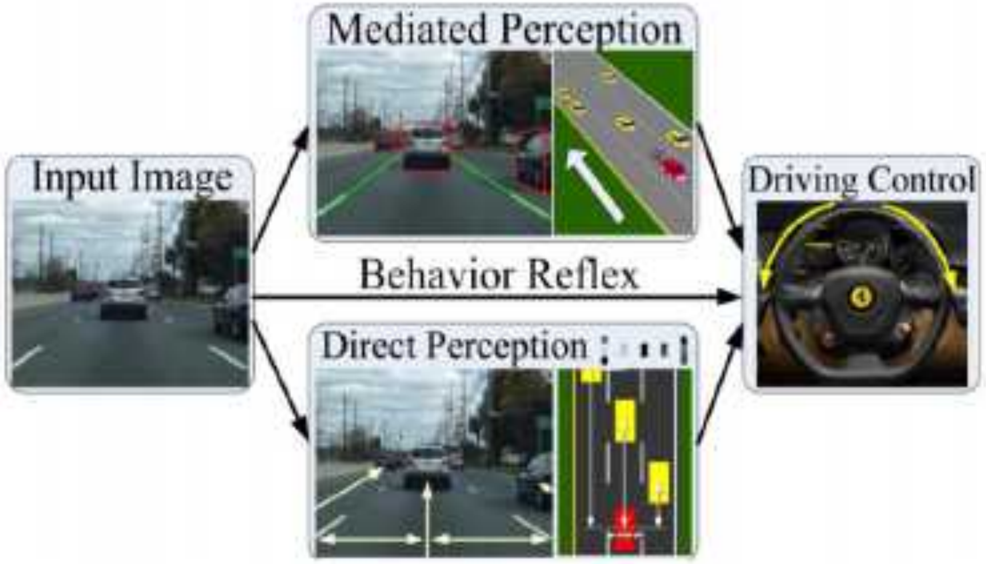


深度学习在自动驾驶中的应用综述

第22组：王晓轩

综述大纲

- 第一章：**绪论**
- 1.1研究背景
- 1.2国内外研究现状
- 1.3本文主要研究内容
- 第二章：自动驾驶中所设计的深度学习算法概述
- 2.1卷积神经网络
- 2.2卷积神经网络的训练
- 2.3深度强化学习
- 2.4递归神经网络
- 第三章：**深度学习在自动驾驶中的应用**
- 三种主要模型：
- 3.1间接感知模型
- 3.2直接感知模型
- 3.3端到端控制模型



第6组：AD分类

董廷臻、郑英炜（组长）、李博翱、朱嘉楠、李杨燊

基本完成对T2w和T1w的数据预处理（包括偏置场矫正，颅骨剔除和配准，T2w由于工具链支持不佳导致实际效果不好）
基本完成一键训练-评估框架（基于5-fold cross-validation，train:valid:test=3:1:1）
正在进行论文的阅读，复现和总结。5个人分4个通用方向，以及每人各一篇医学影像分类/分割相关论文。已经复现了ResNet、SENet、DenseNet和UNet。此外实验了两个自行设计的Net，性能差不多。
目前最好的结果（ResNet）：Accuracy 0.95，F1 0.8 MCC 0.85左右

目前遇到的瓶颈:
显存严重不足。块间致密连接导致跟踪的梯度急剧增多，尤其是DenseNet,Unet和自行设计的TriNet。这限制了batch size和网络的复杂度，从而导致网络性能下降。将来做Attention方向时问题会更加严重。对策：我们正在排查是否是程序本身的问题，同时寻找Memory-Efficient的解决方案。
网络设计能力和训练技巧缺乏。网络过拟合快，15-100 epoch后就出现泛化能力下降。对策：我们之前已经将YOLOv4等论文排入阅读列表中。
预处理使用的工具链问题频发。FSL仅支持对fMRI的头动矫正，现在预处理后的T1w和T2w图像均无法进行非线性配准。使用MNI152模板进行线性配准的效果极差。对策：若网络性能达到瓶颈，再尝试更换预处理工具链。

Applications of artificial intelligence in covid-19 patients

第八组 组长：罗岁岁 组员：尹子宜 周雅雯 肖雨馨 程旸

A. 背景介绍：covid-19 pandemic，AI，medical

B. 研究方向：

1. Detection & diagnosis：Image & Result analysis；Self-examination for symptoms；deep-learning model

2. Treatment：Drugs & Vaccines

- Drug delivery design and development；speed up drug testing in real-time
- Develop vaccines and treatments faster; be helpful for clinical trials
- Monitor the treatment of individuals; provide update information; provide solutions

3. Prognosis

- Predict the time and situation of recovery
- Give mental care and entertainment

C. 项目进展：

- Searching literature and gathering information

小组项目-调研综述进展汇报

助教：张洋

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

助教：肖尊杰

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：何忱、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析：唐云龙、刘叶充、刘旭坤、马卓远、陈子蔚（组长）、江欣乐、陈浩然

助教：杨冰

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）、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

助教：章晓庆

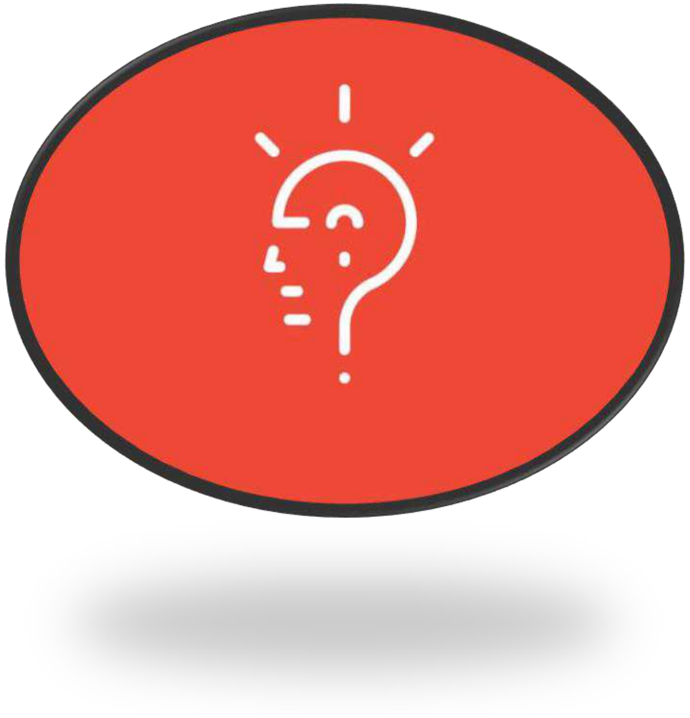
17. AI虚拟主播制作计划：王标、张倚凡（组长）、李康欣、何泽安、曾宇祺、 Zhang Kenneth
18. 人工智能技术在个性化推荐系统上的应用与研究：谭雅静、刘思岑、Ooi Yee Jing、孟宇阳、杨锦涛（组长）
19. 校园巴士路线优化：王祥辰、何鸿杰、吴子彧、樊青远（组长）、方琪涵、袁通
20. 给线稿上色的强大AI的算法研究：韩晗（组长）、刘思语、赵晓蕾、陈松斌
21. 人工智能应用于病理分析的前景与挑战：刘宇欣、李修治（组长）、沈睿琦
22. 深度学习在自动驾驶中的应用：王晓轩

小组项目-调研综述进展汇报

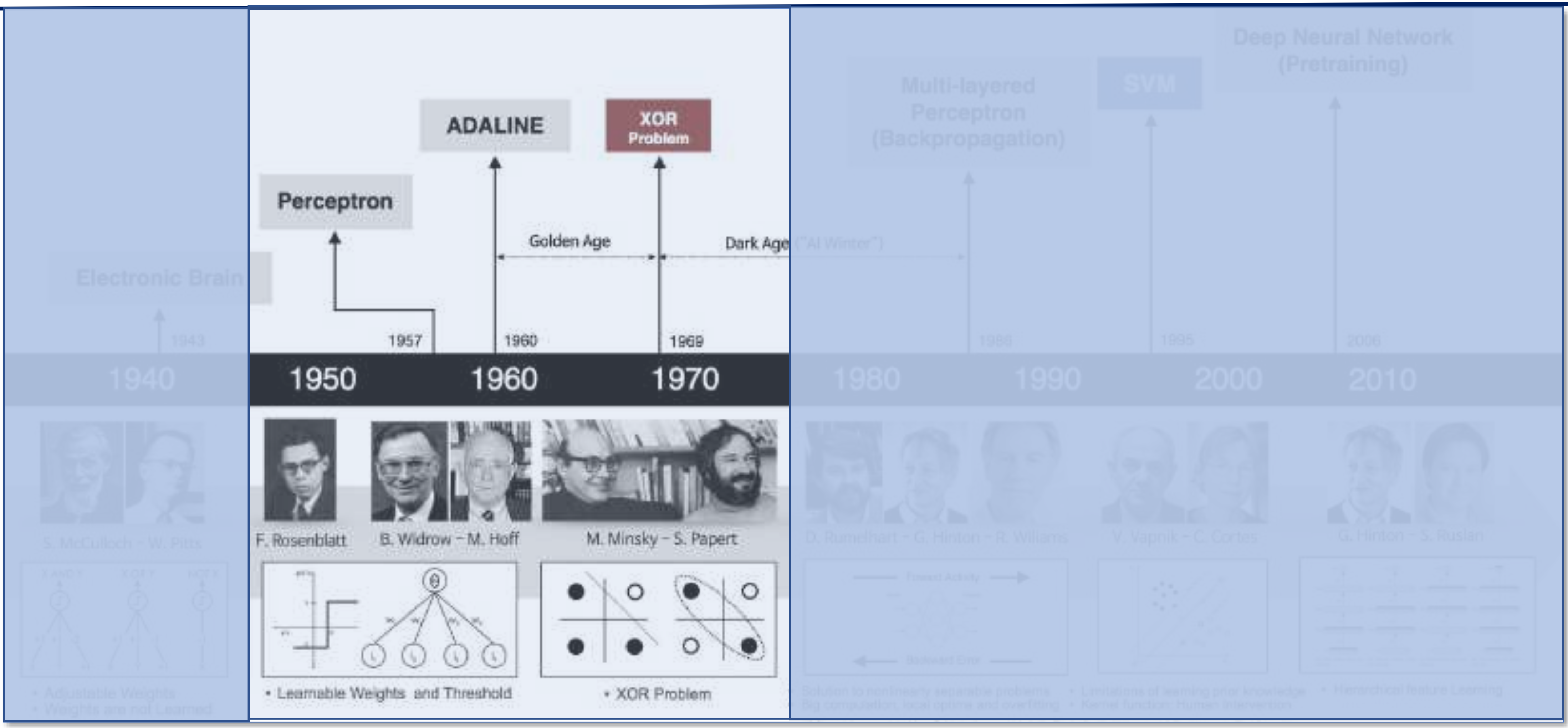
序号	题目	成员
1	AI+斗地主	孙永康、李怀武、胡鸿飞、吴一凡、杨光、张习之、金肇轩（组长）、于佳宁
2	AI+五子棋	周贤玮、韩梓辰（组长）、赵云龙、张坤龙、夏星晨
3	High Score Gamer	易辰朗、许天淇、黄北辰（组长）、赵思源、朱佳伟、宛清源
4	AI application on diabetes	周钰奇、李仪轩、董叔文、湛掌、胡钧淇（组长）、裴鸿婧
5	AI in lung cancer	夏瑞浩、李悦明、龚颖璇、吴云潇潇（组长）、姜欣瑜、王英豪
6	基于MRI图像的阿尔茨海默症分类	董廷臻、郑英炜（组长）、李博翱、朱嘉楠、李杨燊
7	AI Applications in 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

Any Question?



AI algorithm Developments - A Closer Look



Perceptron

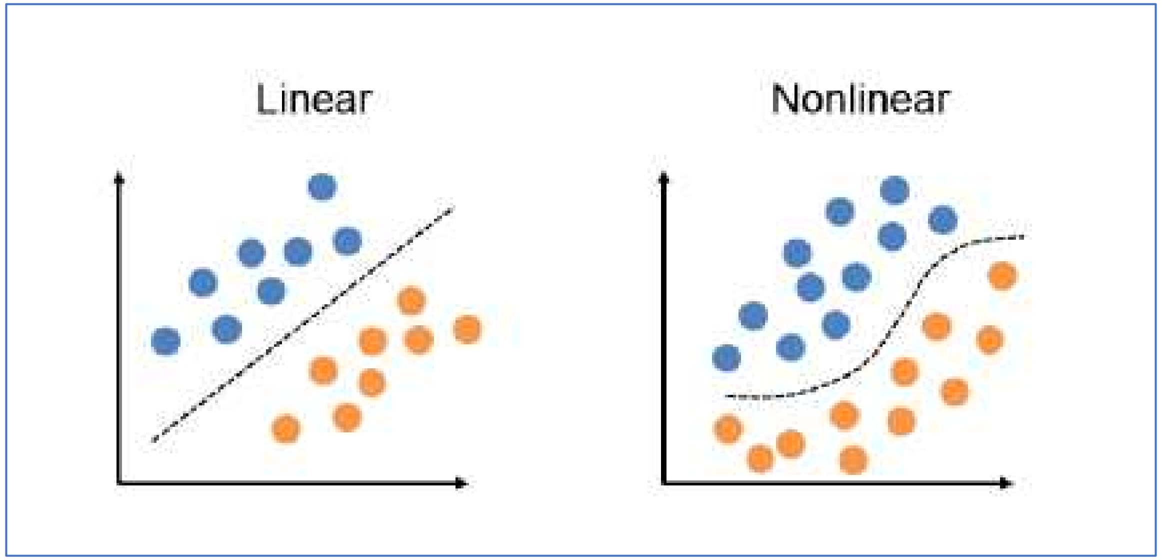
1 Perceptron

2 Perceptron Learning

3 ADALINE

4 Limitation of Perceptron

Linearly Separable



A function is said to be linearly separable when its outputs can be discriminated by a function which is a linear combination of features, that is we can discriminate its outputs by a line or a hyperplane.

Traditional Perceptron Decision Surface

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

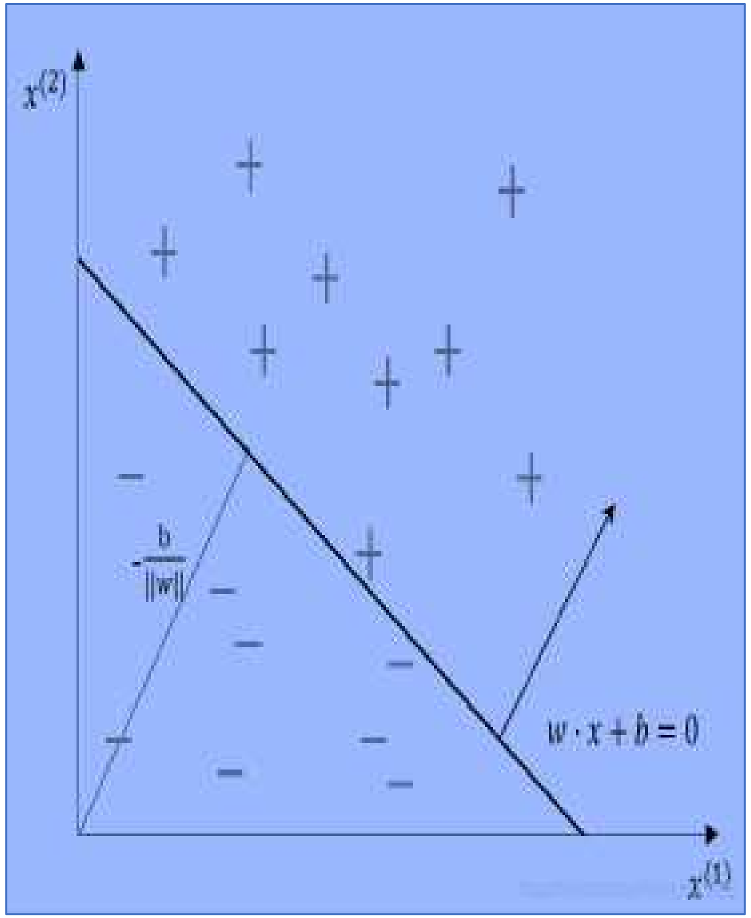
$$\sum_{j=0}^{n} W_j x_j > 0 \quad \text{or} \quad \mathbf{W} \cdot \mathbf{x} > 0$$

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

Perceptron → Linear Separator

Linear discriminant function or linear decision surface.

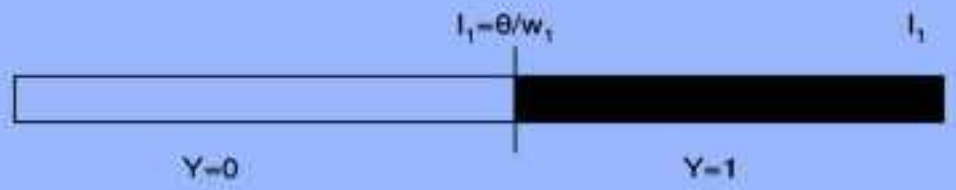
Weights determine slope and bias determines offset.



Decision Surface

Decision surface is the surface at which the output of the unit is precisely equal to the threshold, i.e. $\sum w_i I_i = \theta$

In **1-D** the surface is just a point:



In **2-D**, the surface is

$$I_1 \cdot w_1 + I_2 \cdot w_2 - \theta = 0$$

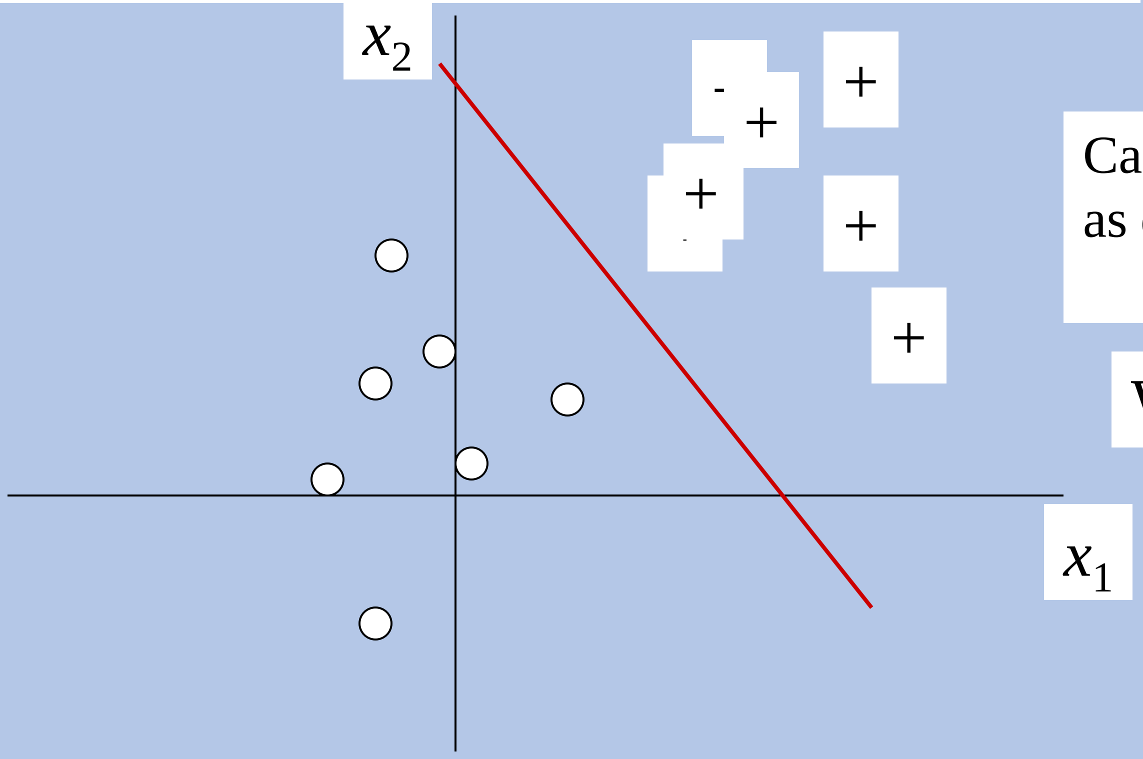
which we can re-write as

$$I_2 = \frac{\theta}{w_2} - \frac{w_1}{w_2} I_1$$

So, in 2-D the decision boundaries are **<u>always</u>** straight lines.

Exercise: Separation Line

Consider example with two inputs, x1, x2:



Can view trained network as defining a "separation line".

What is its equation?

$$-w_0 + w_1 x_1 + w_2 x_2 = 0$$

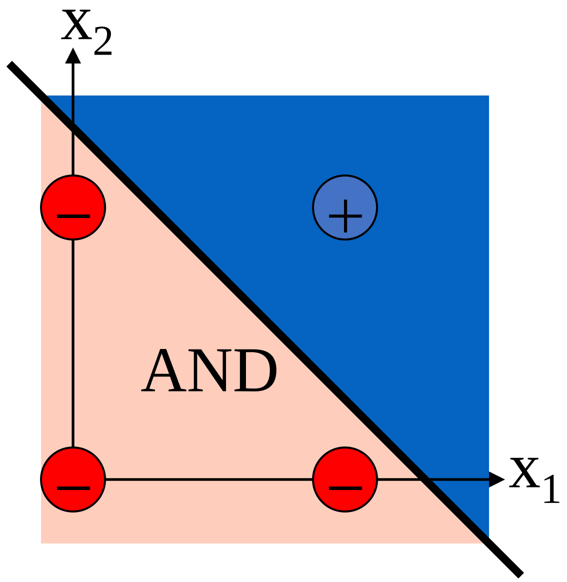
$$x_2 = -\frac{w_1}{w_2} x_1 + \frac{w_0}{w_2}$$

Percepton used for classification

Exercise: Plot the Separation Lines for AND

AND

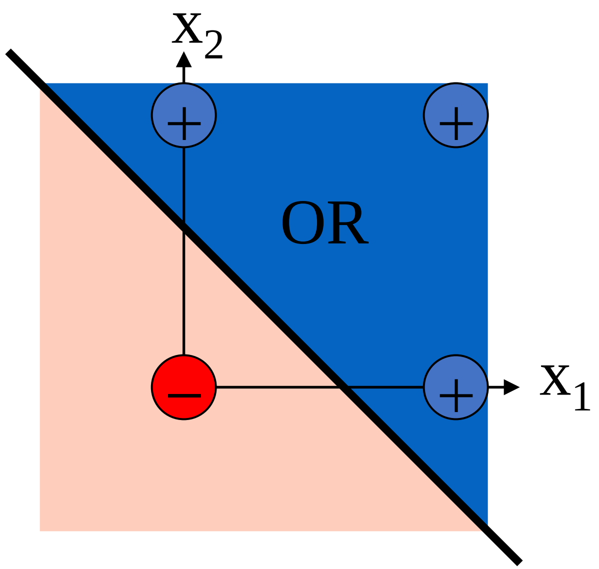
W1	= 1
W2	= 1
W0/θ	= 1.5



Exercise: Plot the Separation Lines for OR

OR

W1 = 1
W2 = 1
W0 / θ = 0.5

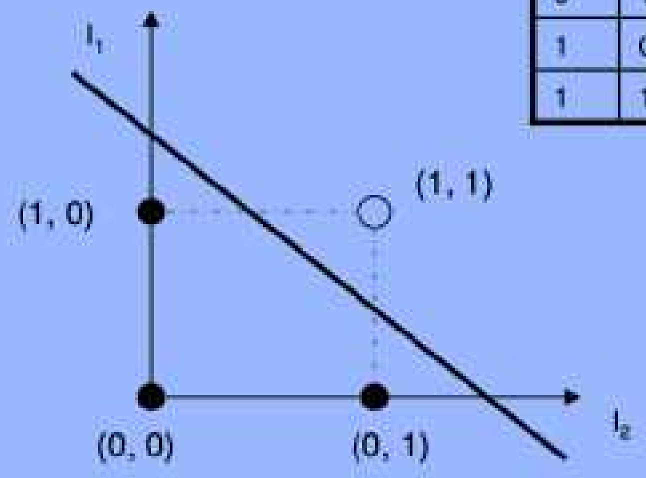


Decision Surfaces/Boundaries for AND and OR

We can now plot the decision boundaries of our logic gates

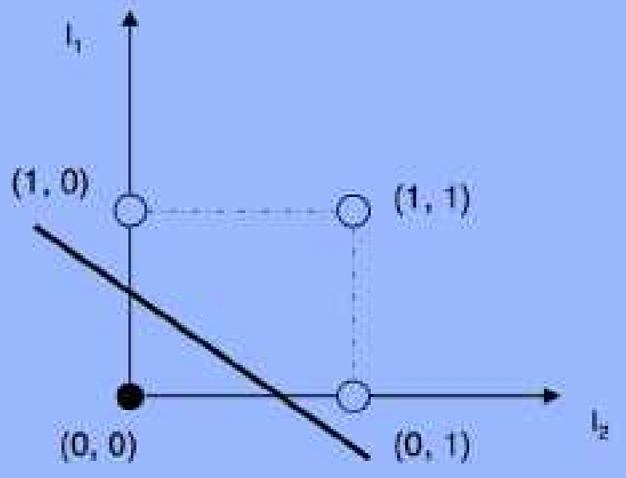
AND
w1=1, w2=1, θ=1.5

AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



OR
w1=1, w2=1, θ=0.5

OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1



“OR Perceptron” Training and Learning Example Input Samples

Consider learning the logical OR function.

Our examples are:

Sample	x0	x1	x2	label
1	1	0	0	0
2	1	0	1	1
3	1	1	0	1
4	1	1	1	1

Activation Function

$$S = \sum_{k=0}^{k=n} w_k x_k \quad S > 0 \; then \; O = 1 \quad else \quad O = 0$$

OR Perceptron Training and Learning Example Weight Update

$$S = \sum_{k=0}^{k=n} w_k x_k \quad S > 0 \; then \; O = 1 \quad else \quad O = 0$$

Weight Update (We set learning rate =1)

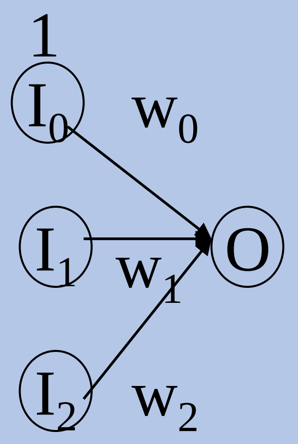
If perceptron is 0 while it should be 1,
add the input vector to the weight vector (if input = 1, you add 1)
(if input =0, you can assume that you add 0)
If perceptron is 1 while it should be 0,
subtract the input vector to the weight vector if input x is 1
(if input =0, you add 0)

Otherwise do nothing.

OR Perceptron Training and Learning Example
Learn First 2 Examples in Epoch 1

We'll use a single perceptron with three inputs.

We'll start with all weights 0 W= <0,0,0>



Example 1 I= < 1 0 0> label=0 W= <0,0,0>

Perceptron (1×0+ 0×0+ 0×0 =0, S=0) output → 0

→it classifies it as 0, so correct, do nothing

Example 2 I=<1 0 1> label=1 W= <0,0,0>

Perceptron (1×0+ 0×0+ 1×0 = 0) output →0

→it classifies it as 0, while it should be 1, so add input to weights

W = <0,0,0> + <1,0,1>= <1,0,1>

OR Perceptron Training and Learning Example
Learn Next 2 Samples in Epoch 1

Example 3 I=<1 1 0> label=1 W= <1,0,1>

Perceptron ($1\times1+ 1\times0+ 0\times1 > 0$) output = 1

→it classifies it as 1, while it should be 1, so do nothing

Example 4 I=<1 1 1> label=1 W= <1,0,1>

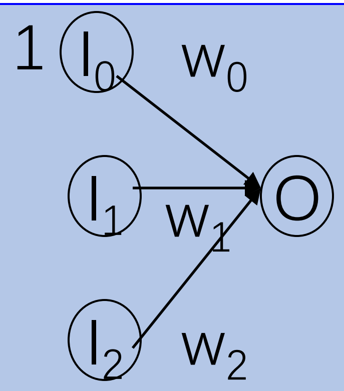
Perceptron ($1\times1+ 1\times0+ 1\times1 > 0$) output = 1

→it classifies it as 1, correct, do nothing

W = <1,0,1>

OR Perceptron Training and Learning Example
Learn First 2 Samples in Epoch 2

Epoch 2, through the examples, W = <1,0,1> .



Example 1 I = <1,0,0> label=0 W = <1,0,1>

Perceptron (1×1+ 0×0+ 0×1 >0) output → 1

→it classifies it as 1, while it should be 0,

so subtract input from weights

W = <1,0,1> - <1,0,0> = <0, 0, 1>

Example 2 I=<1 0 1> label=1 W= <0,0,1>

Perceptron (1×0+ 0×0+ 1×1 > 0) output →1

→it classifies it as 1, so correct, do nothing

OR Perceptron Training and Learning Example
Learn Next 2 Samples in Epoch 2

Example 3   I=<1 1 0>   label=1 W= <0,0,1>

Perceptron ($1\times0+ 1\times0+ 0\times1 > 0$) output = 0

→it classifies it as 0, while it should be 1, so

add input to weights

W = <0,0,1> + W = <1,1,0> = <1, 1, 1>

Example 4   I=<1 1 1>   label=1 W= <1,1,1>

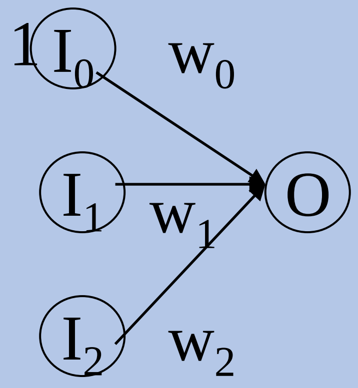
Perceptron ($1\times1+ 1\times1+ 1\times1 > 0$) output = 1

→it classifies it as 1, correct, do nothing

W = <1,1,1>

OR Perceptron Training and Learning Example Learn First 2 Samples in Epoch 3

Epoch 3, through the examples, W = <1,1,1> .



Example 1 I=<1,0,0> label=0 W = <1,1,1>

Perceptron (1×1+ 0×1+ 0×1 >0) output → 1

→it classifies it as 1, while it should be 0, so

subtract input from weights

W = <1,1,1> - W = <1,0,0> = <0, 1, 1>

Example 2 I=<1 0 1> label=1 W= <0, 1, 1>

Perceptron (1×0+ 0×1+ 1×1 > 0) output →1

→it classifies it as 1, so correct, do nothing

OR Perceptron Training and Learning Example
Learn Next 2 Samples in Epoch 3

Example 3 I=<1 1 0> label=1 W= <0, 1, 1>

Perceptron (1×0+ 1×1+ 0×1 > 0) output = 1

→it classifies it as 1, correct, do nothing

Example 4 I=<1 1 1> label=1 W= <0, 1, 1>

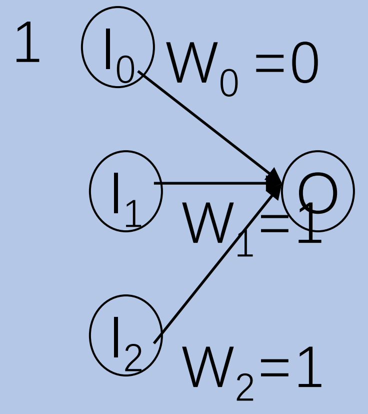
Perceptron (1×0+ 1×1+ 1×1 > 0) output = 1

→it classifies it as 1, correct, do nothing

W = <1,1,1>

OR Perceptron Training and Learning Example
Learn First Samples in Epoch 4

Epoch 4, through the examples, W= <0, 1, 1>.



OR

Example 1 I= <1,0,0> label=0 W = <0,1,1>

Perceptron (1×0+ 0×1+ 0×1 = 0) output → 0

→it classifies it as 0, so correct, do nothing

So the final weight vector W= <0, 1, 1> classifies all examples correctly, and the perceptron has learned the function!

Aside: in more realistic cases the bias (W0) will not be 0.
Also, in general, many more inputs (100 to 1000)

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1
example 2	1	0	1	1	0	0	1	1	0	0	0	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1
example 2	1	0	1	1	0	0	1	1	0	0	0	1
example 3	1	1	0	1	0	0	1	0	1	1	1	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1
example 2	1	0	1	1	0	0	1	1	0	0	0	1
example 3	1	1	0	1	0	0	1	0	1	1	1	1
example 4	1	1	1	1	1	1	1	1	0	1	1	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1
example 2	1	0	1	1	0	0	1	1	0	0	0	1
example 3	1	1	0	1	0	0	1	0	1	1	1	1
example 4	1	1	1	1	1	1	1	1	0	1	1	1
3 example 1	1	0	0	0	1	1	1	1	-1	0	1	1

Med

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1
example 2	1	0	1	1	0	0	1	1	0	0	0	1
example 3	1	1	0	1	0	0	1	0	1	1	1	1
example 4	1	1	1	1	1	1	1	1	0	1	1	1
3 example 1	1	0	0	0	1	1	1	1	-1	0	1	1
example 2	1	0	1	1	0	1	1	1	0	0	1	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1
example 2	1	0	1	1	0	0	1	1	0	0	0	1
example 3	1	1	0	1	0	0	1	0	1	1	1	1
example 4	1	1	1	1	1	1	1	1	0	1	1	1
3 example 1	1	0	0	0	1	1	1	1	-1	0	1	1
example 2	1	0	1	1	0	1	1	1	0	0	1	1.
example 3	1	1	0	1	0	1	1	1	0	0	1	1

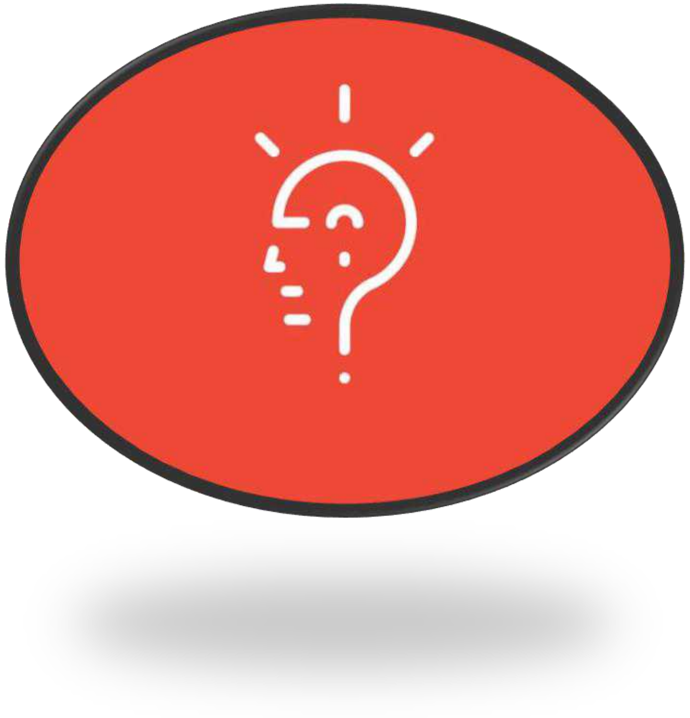
OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1
example 2	1	0	1	1	0	0	1	1	0	0	0	1
example 3	1	1	0	1	0	0	1	0	1	1	1	1
example 4	1	1	1	1	1	1	1	1	0	1	1	1
3 example 1	1	0	0	0	1	1	1	1	-1	0	1	1
example 2	1	0	1	1	0	1	1	1	0	0	1	1.
example 3	1	1	0	1	0	1	1	1	0	0	1	1
example 4	1	1	1	1	0	1	1	1	0	0	1	1

OR Perceptron Learning Summary

Epoch	x0	x1	x2	Desired Target	w0	w1	w2	Output	Error	New w0	New w1	New w2
1 example 1	1	0	0	0	0	0	0	0	0	0	0	0
example 2	1	0	1	1	0	0	0	0	1	1	0	1
example 3	1	1	0	1	1	0	1	1	0	1	0	1
example 4	1	1	1	1	1	0	1	1	0	1	0	1
2 example 1	1	0	0	0	1	0	1	1	-1	0	0	1
example 2	1	0	1	1	0	0	1	1	0	0	0	1
example 3	1	1	0	1	0	0	1	0	1	1	1	1
example 4	1	1	1	1	1	1	1	1	0	1	1	1
3 example 1	1	0	0	0	1	1	1	1	-1	0	1	1
example 2	1	0	1	1	0	1	1	1	0	0	1	1
example 3	1	1	0	1	0	1	1	1	0	0	1	1
example 4	1	1	1	1	0	1	1	1	0	0	1	1
4 example 1	1	0	0	0	0	1	1	0	0	0	1	1

Any Question?



Perceptron

1 Perceptron

2 Perceptron Learning

3 ADALINE

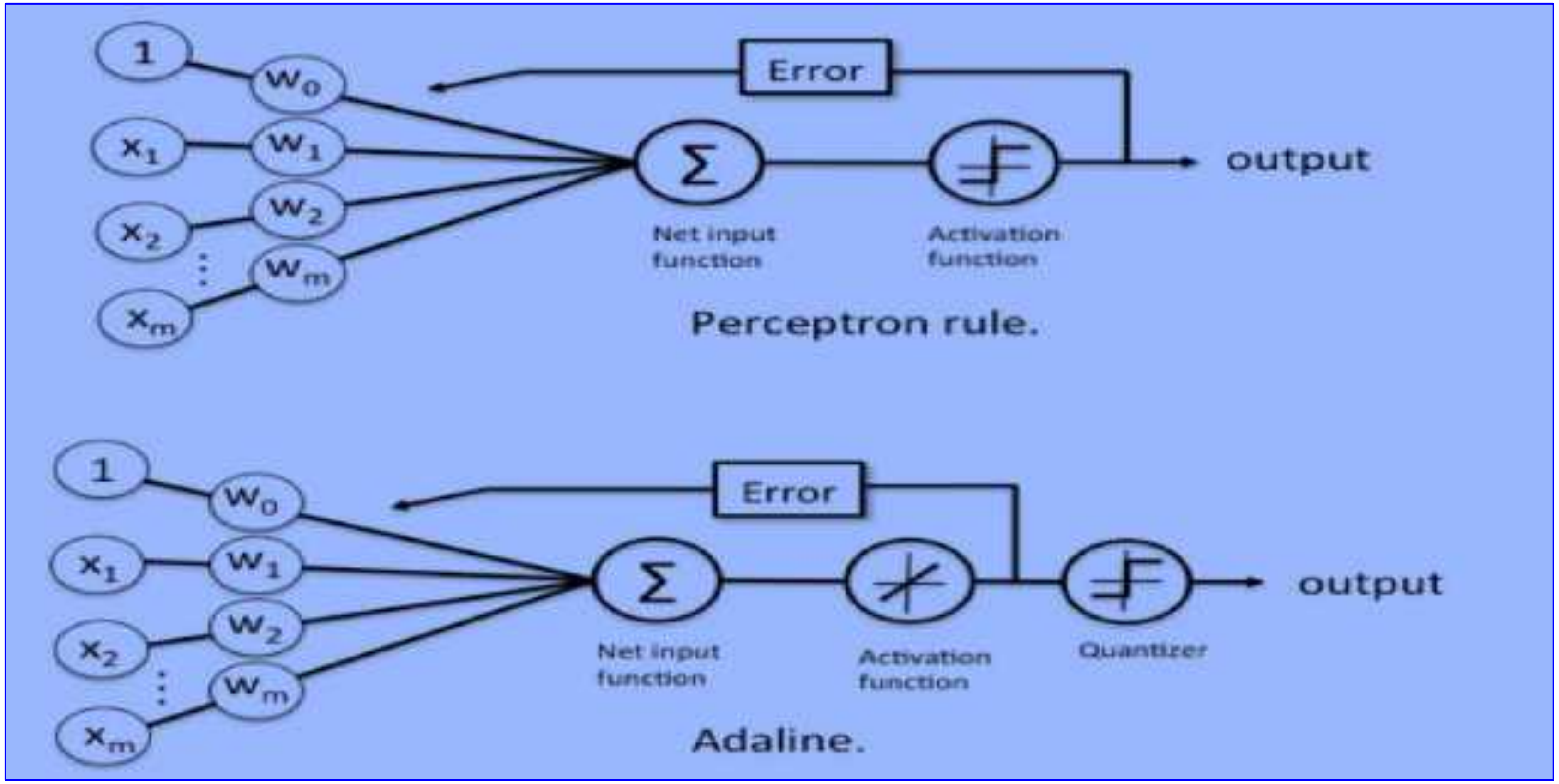
4 Limitation of Perceptron

ADALINE (Adaptive Linear Neuron)

ADALINE is an early single-layer artificial neural network based on Least Mean Squares (LMS) algorithms.

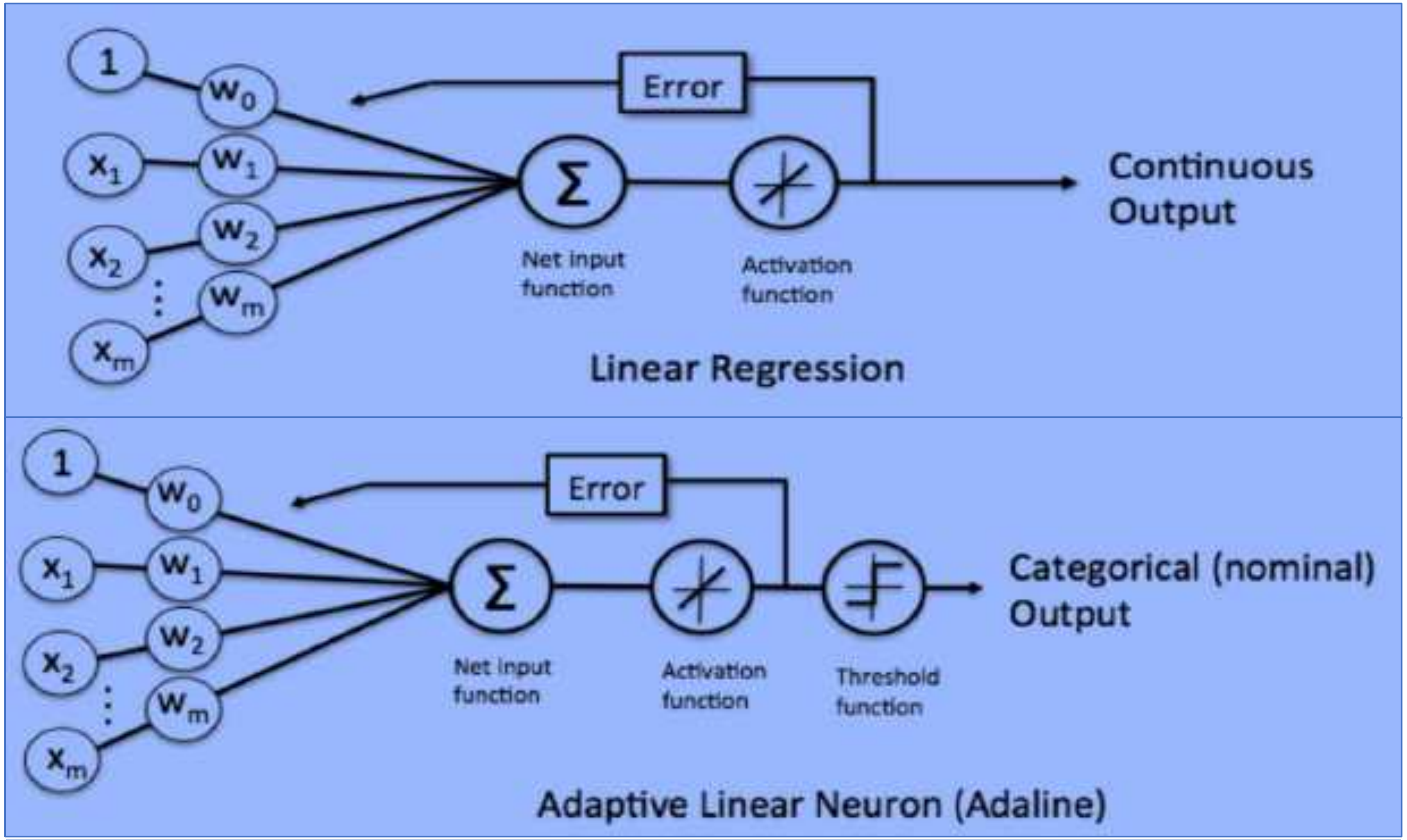
It was invented in 1960 by Stanford University science professor Bernard Widrow and his first Ph.D. student, Ted Hoff.

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

Linear Regression 线性回归 and ADALINE



Adaline algorithm is a identical to linear regression except for a threshold function that converts the continuous output into a categorical class label

Widrow Hoff Learning Algorithm

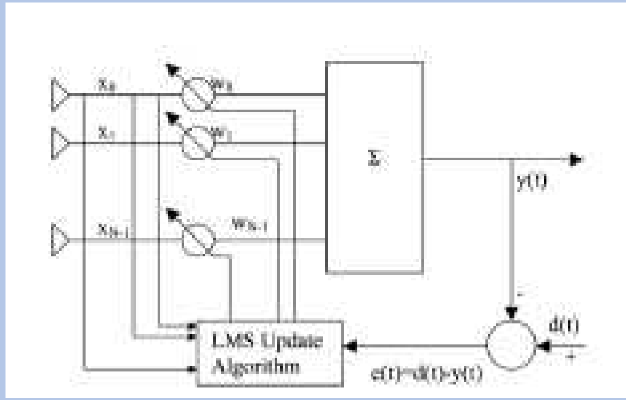
- Also known as **Delta Rule**. It follows gradient descent rule for linear regression.

- It updates the connection weights with the difference between the target and the output value. It is the least mean square learning algorithm falling under the category of the supervised learning algorithm.

- This rule is followed by ADALINE (ADAptive LInear NEuron or NEural Networks) and **MADALINE**. Unlike Perceptron, the iterations of Adaline networks do not stop, but it converges by reducing the least mean square error. MADALINE is a network of more than one ADALINE.

ADALINE (Adaptive Linear Neuron)

LMS algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean square of the error signal. It is a stochastic gradient descent method, which does not require gradient to be know and it is estimated at every iteration. In that way, the filter is only adapted based on the error at the current time.



* B.Widrow and M.E.Hoff, "Adaptive switching circuits," Proc. Of WESCON Conv. Rec., part 4, pp.96-140, 1960

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:

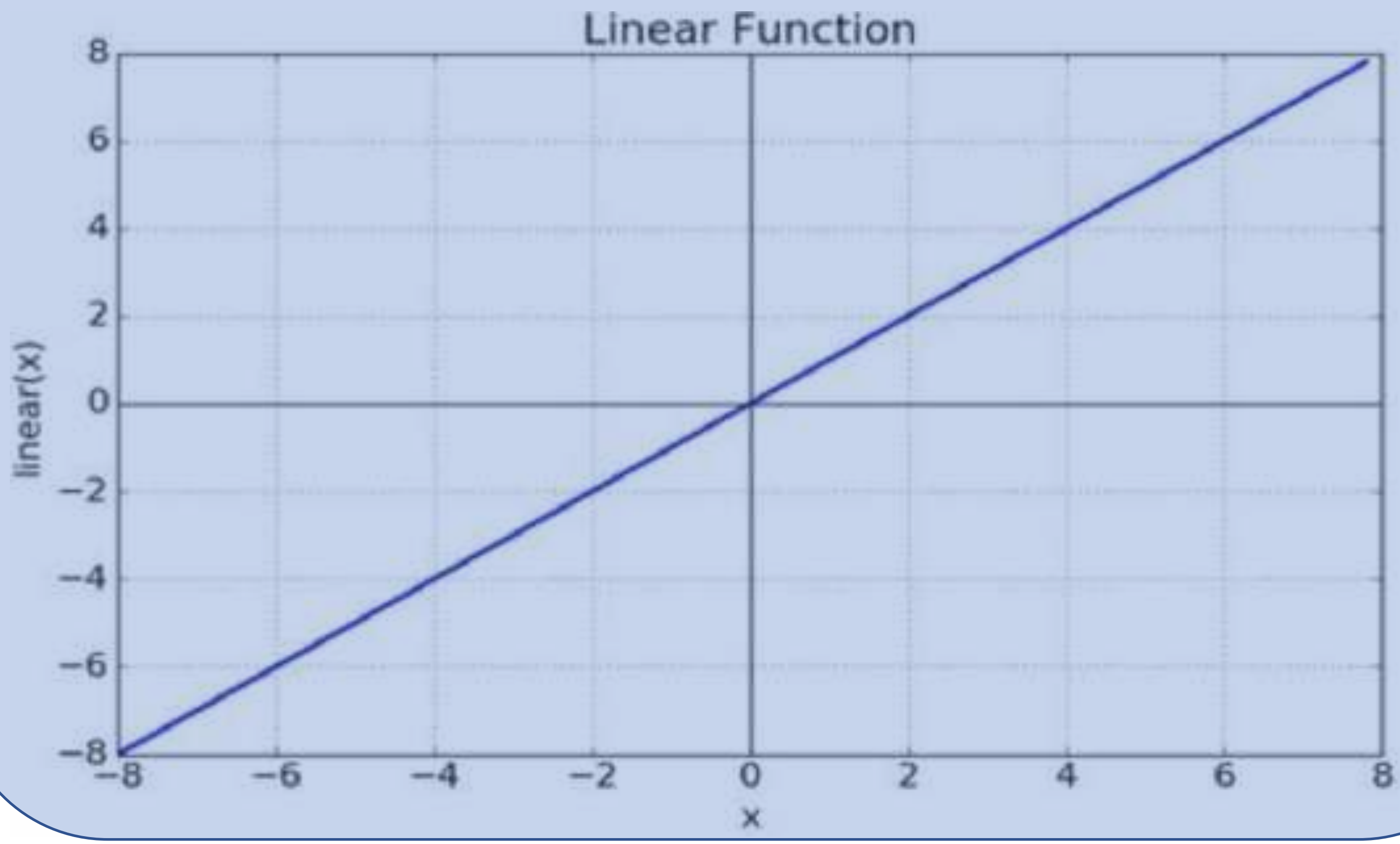
  Least Mean Square error (LMS) = $(t - y_{in})^2$,

  ADALINE converges when the least mean square error is reached.

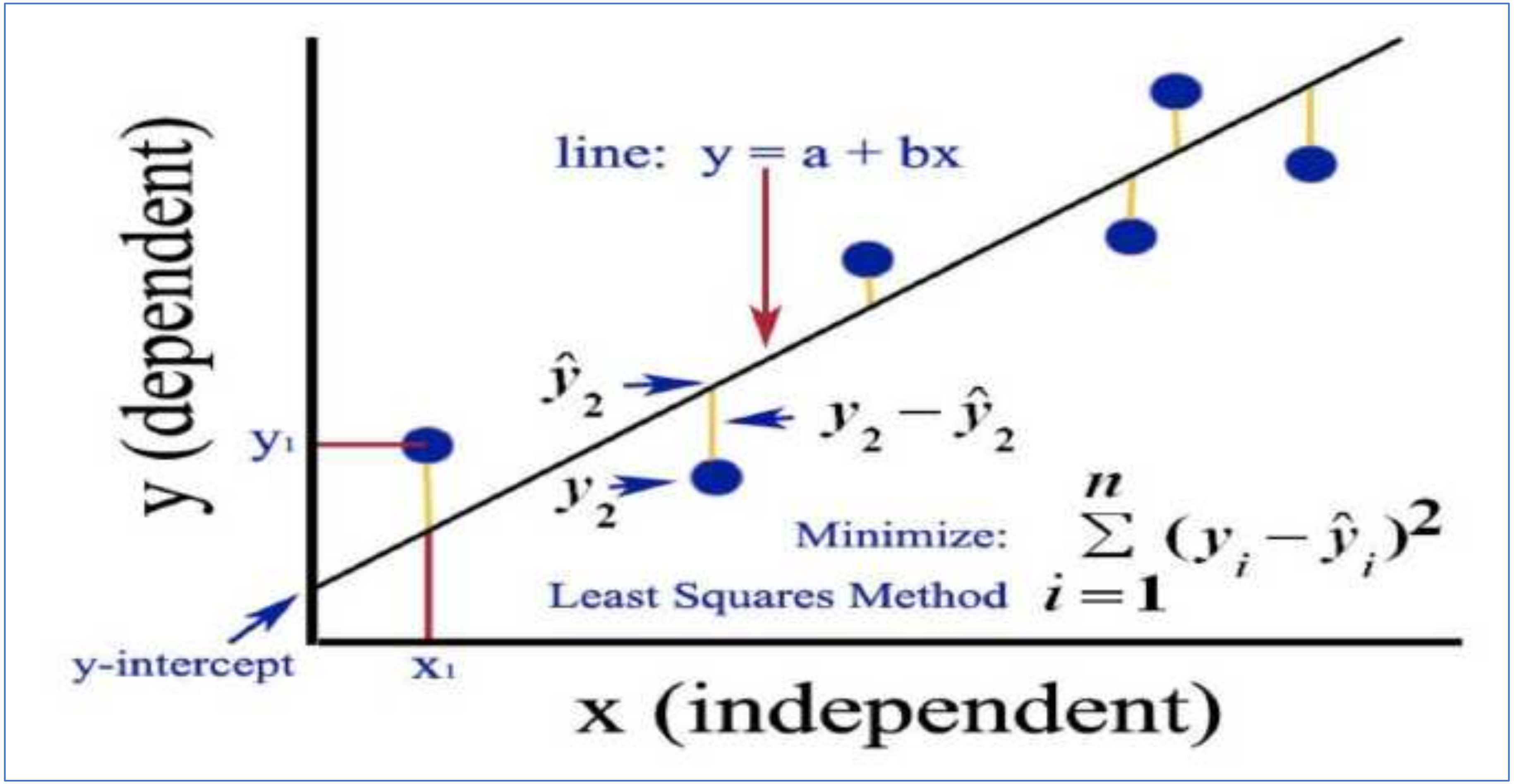
- Learning is an optimization search problem in weight space

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

“Artificial” Neuron
Linear Transfer (Activation) Function



Least Sum of Squared Errors (SSE) for MADALINE

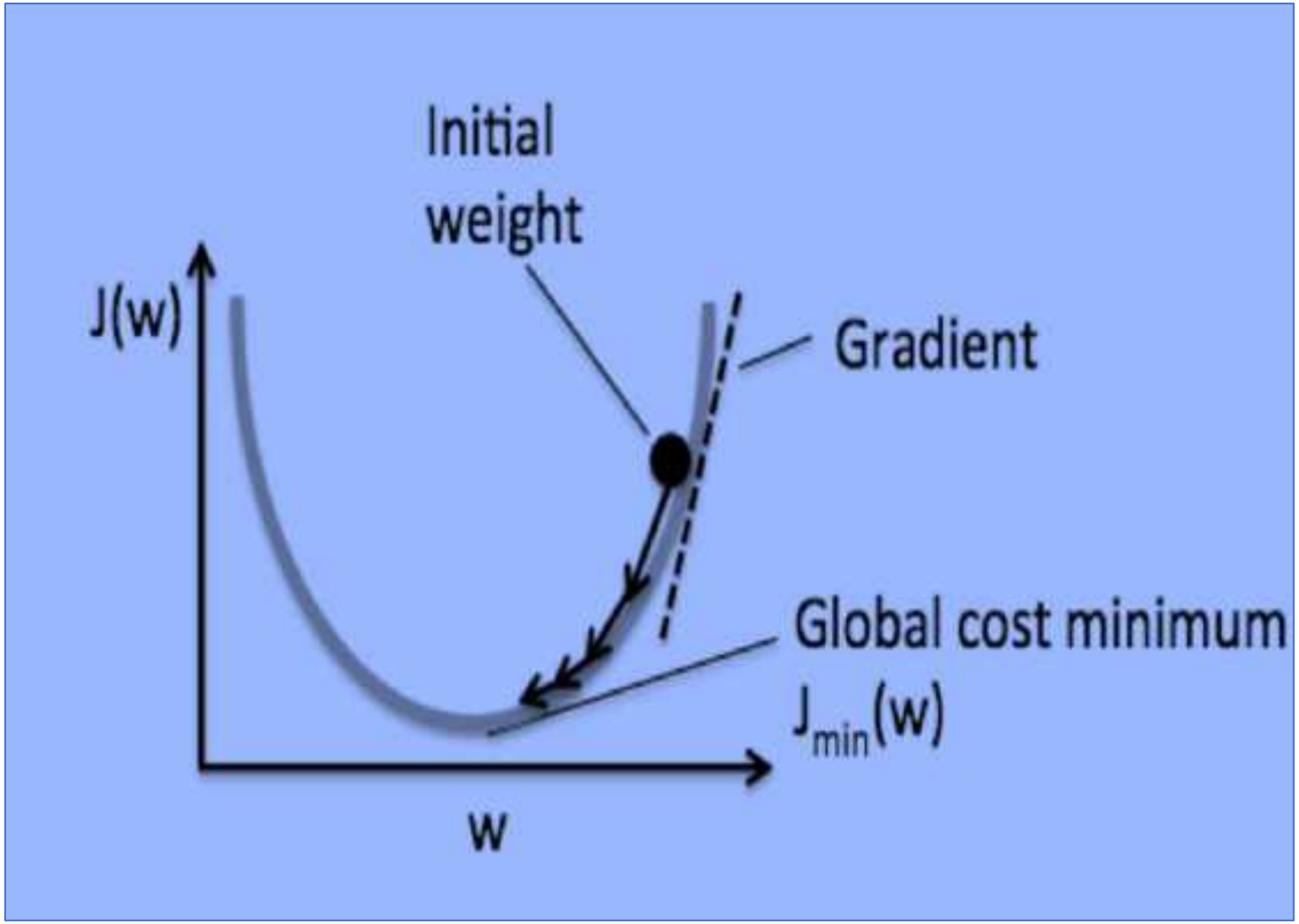


LMS Gradient Descent

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



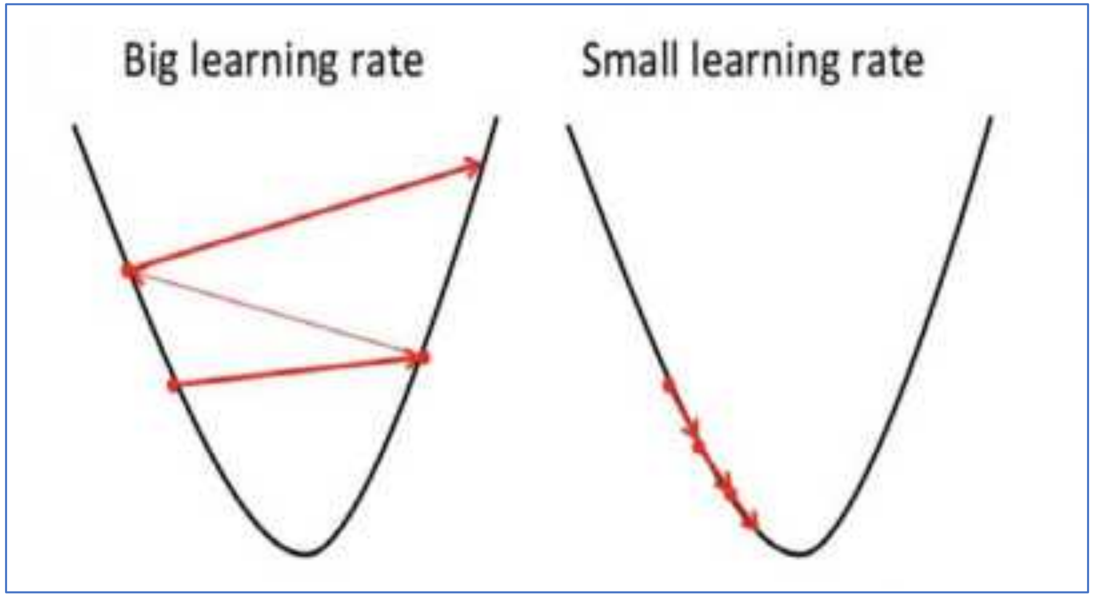
LMS Gradient Calculation

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

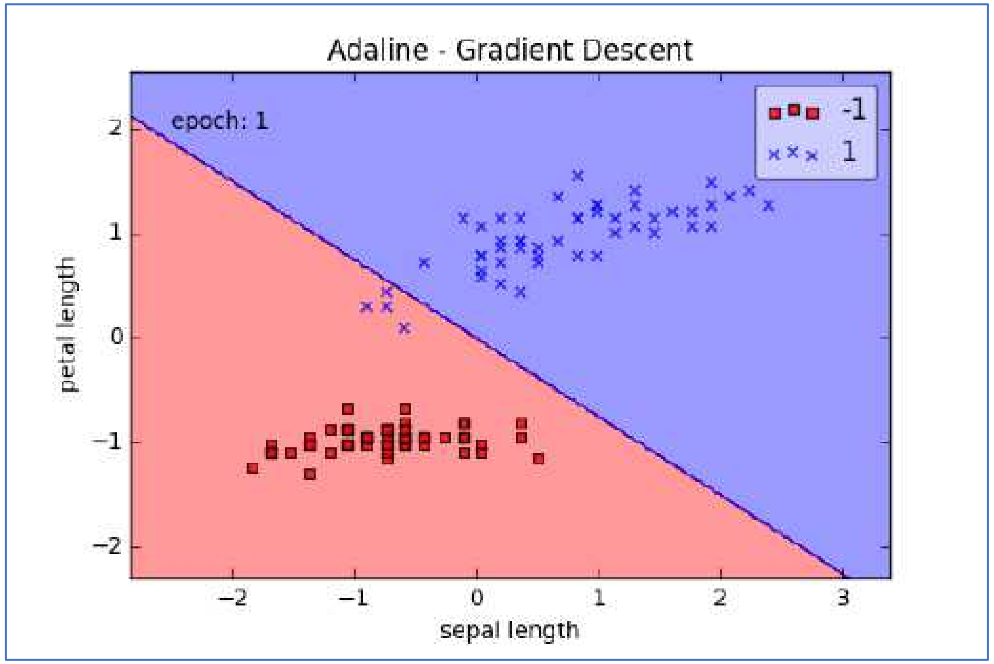
$$\begin{aligned}
&\frac{\partial J}{\partial w_j} \\
&= \frac{\partial}{\partial w_j} \frac{1}{2} \sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right)^2 \\
&= \frac{1}{2} \frac{\partial}{\partial w_j} \sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right)^2 \\
&= \frac{1}{2} \sum_i \left(y^{(i)} - \phi(z)_A^{(i)}\right)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

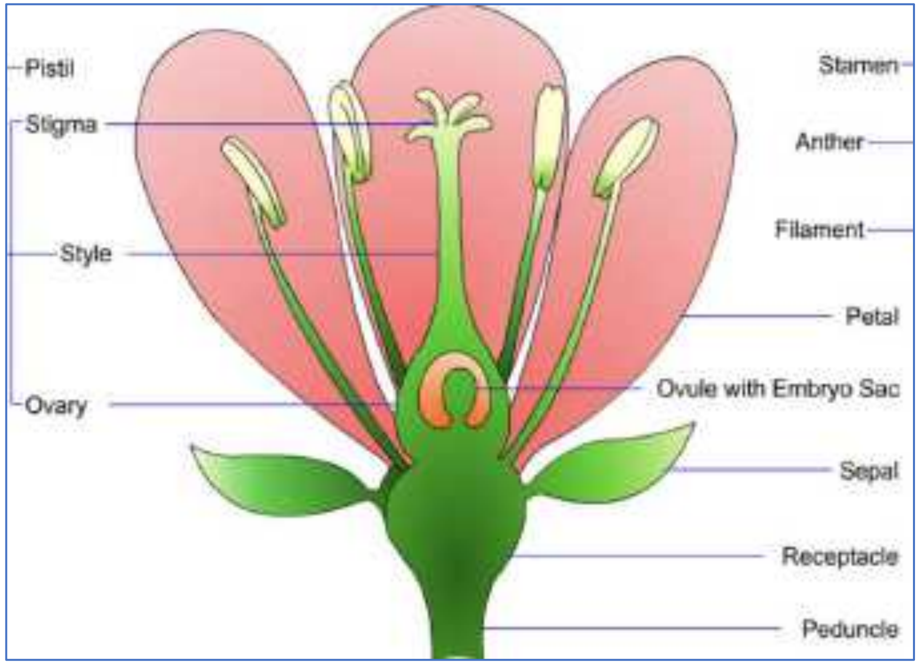
LMS Gradient Calculation

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

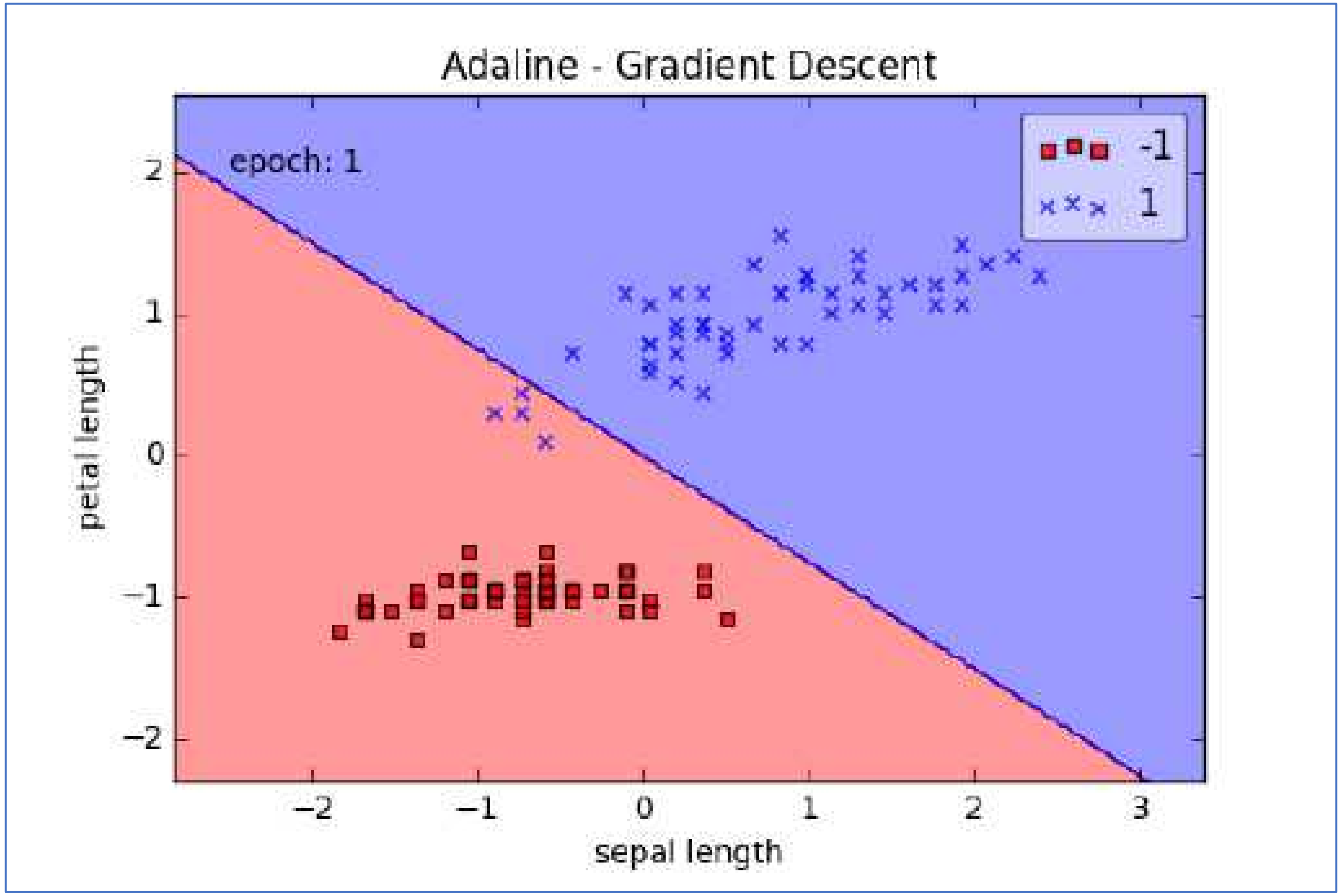


Flower Classification based on Sepal &Petal Length

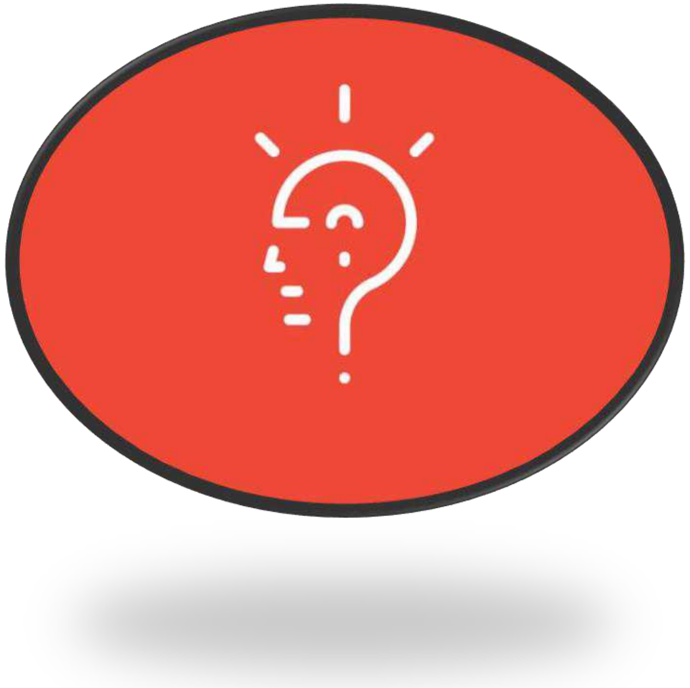




ADALINE Learning



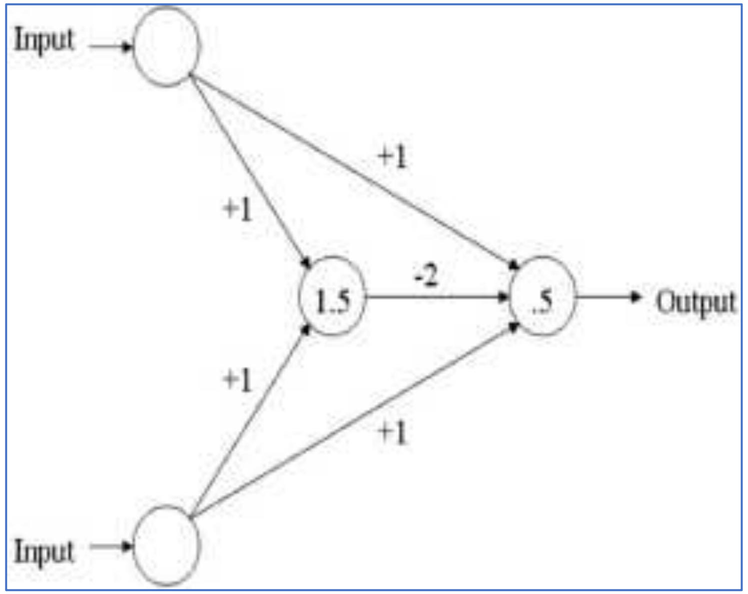
Any Question?



Med

Homework 08

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

CS 103 -08
Perceptron Learning and ADALINE

Jimmy Liu 刘江

2020-11-6