

《人工智能及其应用》

实验指导书

青岛大学计算机科学与技术学院

2018 年 4 月

前 言

本实验是为了配合《人工智能及其应用》课程的理论学习而专门设置的。本实验的目的是巩固和加强人工智能的基本原理和方法,并为今后进一步学习更高级课程和信息智能化技术的研究与系统开发奠定良好的基础。

全书共分为六个实验: 1.A*算法求解 8 数码问题; 2. 利用 $\alpha - \beta$ 搜索过程的博弈树搜索算法编写一字棋游戏; 3.线性分类器的设计实现; 4.感知器算法的设计实现; 5.SVM 线性分类器的设计实现; 6.卷积神经网络 CNN 框架的设计实现。每个实验包括有: 实验目的、实验内容、实验条件、实验要求、实验步骤和实验报告等六个项目。

本实验指导书包括两个部分。第一个部分是介绍实验的教学大纲; 第二部分是介绍六个实验的内容。

由于编者水平有限, 本实验指导书的错误和不足在所难免, 欢迎批评指正。

人工智能课程组
2018 年 4 月

目录

实验教学大纲.....	1
实验一 A*算法求解 8 数码问题.....	3
实验二 利用 α - β 搜索过程的博弈树搜索算法编写一字棋游戏	5
实验三 Fisher 线性分类器的设计实现.....	7
实验四 感知器算法的设计实现.....	10
实验五 SVM 分类器的设计实现.....	12
实验六 卷积神经网络 CNN 框架的设计实现	16

实验教学大纲

一、学时：16 学时，第 8 周至第 15 周。

二、主要仪器设备及运行环境：PC 机、Visual C++ 6.0、Matlab 7.0。

三、实验项目及教学安排

序号	实验名称	实验平台	实验内容	学时	类型	教学要求
1	A*算法求解 8 数码问题	VC++	设计与实现求解 N 数码问题的 A*算法	2	设计	课内
2	利用 $\alpha - \beta$ 搜索过程的博弈树搜索算法编写一字棋游戏	VC++	设计与实现一字棋的博弈树搜索算法。	2	设计	课内
3	Fisher 线性分类器的设计实现	Matlab	设计与实现线性可分的两类分类问题。	2	设计	课内
4	感知器算法的设计实现	Matlab	利用感知器算法实现多类分类问题的设计与实现。	2	设计	课内
5	SVM 分类器的设计实现	Matlab	1) SVM 分类器设计框架； 2) SVM 分类器的应用。	4	综合	课内
6	卷积神经网络 CNN 框架的设计实现	Matlab	1) 卷积神经网络 CNN 的设计框架； 2) 基于 CNN 网络的数字识别设计	4	综合	课内

四、实验成绩评定

实验课成绩单独按五分制评定。凡实验成绩不及格者，该门课程就不及格。学生的实验成绩应以平时考查为主，应占课程总成绩的 20%，其平时成绩又要以实验实际操作的优劣作为主要考核依据。对于实验课成绩，无论采取何种方式进行考核，都必须按实验课的目的要求，以实际实验工作能力的强弱作为评定成绩的主要依据。

评定各级成绩时，可参考以下标准：

（一）优秀

能正确理解实验的目的要求，能独立、顺利而正确地完成各项实验操作，会分析和处理实验中遇到的问题，能掌握所学的各项实验技能，能较好地完成实验

报告及其它各项实验作业，有一定创造精神和能力。有良好的实验室工作作风和习惯。

（二）良好

能理解实验的目的和要求，能认真而正确地完成各项实验操作，能分析和处理实验中遇到的一些问题。能掌握所学实验技能的绝大部分，对难点较大的操作完成有困难。能一般完成实验报告和其它实验作业。有较好的实验习惯和工作作风。

（三）中等

能粗浅理解实验目的要求，能认真努力进行各项实验操作，但技巧较差。能分析和处理实验中一些较容易的问题，掌握实验技能的大部分。有 30%掌握得不好。能一般完成各项实验作业和报告。处理问题缺乏条理。工作作风较好。能认真遵守各项规章制度。学习努力。

（四）及格

只能机械地了解实验内容，能一般按图、或按实验步骤“照方抓药”完成实验操作，能完成 60%所学的实验技能，有些虽作但不准确。遇到问题常常缺乏解决的办法，在别人启发下能作些简单处理，但效果不理想。能一般完成实验报告，能认真遵守实验室各项规章制度，工作中有小的习惯性毛病（如工作无计划，处理问题缺乏条理）。

（五）不及格

盲目地“照方抓药”，只掌握 50%的所学实验技能。有些实验虽能作，但一般效果不好，操作不正确。工作忙乱无条理。一般能遵守实验室规章制度，但常有小的错误。实验报告较多的时候有结果，遇到问题时说不明原因，在教师指导下也较难完成各项实验作业。或有些小聪明但不努力，不求上进。

五. 实验报告

实验报告独立成册，包括封页及所有实验报告内容，具体要求见附件 0. 最终实验报告提交至邮箱：cathy_huanyang@hotmail.com.

实验一 A*算法求解 8 数码问题

一、实验目的

熟悉和掌握启发式搜索的定义、估价函数和算法过程，并利用 A*算法求解 N 数码难题，理解求解流程和搜索顺序。

二、实验原理

A*算法是一种启发式图搜索算法，其特点在于对估价函数的定义上。对于一般的启发式图搜索，总是选择估价函数 f 值最小的节点作为扩展节点。因此， f 是根据需要找到一条最小代价路径的观点来估算节点的，所以，可考虑每个节点 n 的估价函数值为两个分量：从起始节点到节点 n 的实际代价 $g(n)$ 以及从节点 n 到达目标节点的估价代价 $h(n)$ ，且 $h(n) \leq h^*(n)$ ， $h^*(n)$ 为 n 节点到目的节点的最优路径的代价。

八数码问题是在 3×3 的九宫格棋盘上，摆有 8 个刻有 1~8 数码的将牌。棋盘有一个空格，允许紧邻空格的某一将牌可以移到空格中，这样通过平移将牌可以将某一将牌布局变换为另一布局。针对给定的一种初始布局或结构（目标状态），问如何移动将牌，实现从初始状态到目标状态的转变。如下图 1 表示了一个具体的八数码问题求解。



图 1 八数码问题的求解

三、实验内容

1. 参考 A*算法核心代码(见附件 1)，以 8 数码问题为例实现 A*算法的求解程序（编程语言不限），要求设计两种不同的估价函数。
2. 设置相同的初始状态和目标状态，针对不同的估价函数，求得问题的解，并比较它们对搜索算法性能的影响，包括扩展节点数、生成节点数等。
3. 设置与上述 2 相同的初始状态和目标状态，用宽度优先搜索算法（即令估计代价 $h(n)=0$ 的 A*算法）求得问题的解，以及搜索过程中的扩展节点数、生成节点数。
- *4. （选做）参考 A*算法核心代码，实现 A*算法求解 15 数码问题的程序，设计两种不同的估价函数，然后重复上述 2 和 3 的实验内容。
5. 提交实验报告和源程序。

四、实验报告要求

1. 分析不同的估价函数对 A*算法性能的影响。
2. 根据宽度优先搜索算法和 A*算法求解 8、15 数码问题的结果，分析启发式搜索的特点。

下面是实验报告的基本内容和书写格式。

实验名称

一、实验目的

二、实验原理

三、实验结果

按照实验内容，把结果填入表 1。

表 1 不同启发函数 $h(n)$ 求解 8 数码问题的结果比较

	启发函数 $h(n)$		
	不在位数		0
初始状态			
目标状态	123804765	123804765	123804765
最优解			
扩展节点数			
生成节点数			
运行时间			

***表 2 不同启发函数 $h(n)$ 求解 15 数码问题的结果比较**

	启发函数 $h(n)$		
	不在位数		0
初始状态			
目标状态			
最优解			
扩展节点数			
生成节点数			
运行时间			

四、实验总结

1. 画出 A* 算法求解 N 数码问题的流程图
2. 完成实验报告要求 1 和 2。
3. 总结实验心得体会
4. 提交完整实验程序

实验二 利用 $\alpha - \beta$ 搜索的博弈树算法编写一字棋游戏

一、实验目的

理解和掌握博弈树的启发式搜索过程，能够用选定的编程语言实现简单的博弈游戏。

- (1) 学习极大极小搜索及 $\alpha - \beta$ 剪枝。
- (2) 利用学到的算法实现一字棋

二、实验原理

2.1 游戏规则

“一字棋”游戏（又叫“三子棋”或“井字棋”），是一款十分经典的益智小游戏。“井字棋”

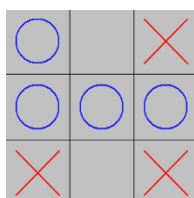
的棋盘很简单，是一个 3×3 的格子，很像中国文字中的“井”字，所以得名“井字棋”。“井字棋”游戏的规则与“五子棋”十分类似，“五子棋”的规则是一方首先五子连成一线就胜利；“井字棋”是一方首先三子连成一线就胜利。

井字棋（英文名 Tic-Tac-Toe）

井字棋的出现年代估计已不可考，西方人认为这是由古罗马人发明的；但我们中国人认为，既然咱们都发明了围棋、五子棋，那发明个把井字棋自然是不在话下。这些纯粹是口舌之争了，暂且不提。

2.2 极小极大分析法

设有九个空格，由 MAX，MIN 二人对弈，轮到谁走棋谁就往空格上放一只自己的棋子，谁先使自己的棋子构成“三子成一线”（同一行或列或对角线全是某人的棋子），谁就取得了胜利。



用圆圈表示 MAX，用叉号代表 MIN。

比如左图中就是 MAX 取胜的棋局。

估价函数定义如下：

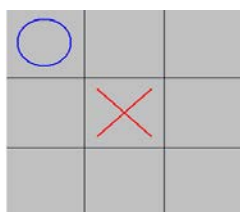
设棋局为 P ，估价函数为 $e(P)$ 。

(1) 若 P 对任何一方来说都不是获胜的位置，则 $e(P) = e(\text{那些仍为 MAX 空着的完全的行、列或对角线的总数}) - e(\text{那些仍为 MIN 空着的完全的行、列或对角线的总数})$

(2) 若 P 是 MAX 必胜的棋局，则 $e(P) = +\infty$ （实际上赋了 60）。

(3) 若 P 是 MIN 必胜的棋局，则 $e(P) = -\infty$ （实际上赋了 -20）。

比如 P 如下图示，则 $e(P) = 5 - 4 = 1$



需要说明的是， $+\infty$ 赋 60， $-\infty$ 赋 -20 的原因是机器若赢了，则不论玩家下一步是否会赢，都会走这步必赢棋。

2.3 α - β 剪枝算法

上述的极小极大分析法，实际是先生成一棵博弈树，然后再计算其倒推值，至使极小极大分析法效率较低。于是在极小极大分析法的基础上提出了 α - β 剪枝技术。

α - β 剪枝技术的基本思想或算法是，边生成博弈树边计算评估各节点的倒推值，并且根据评估出的倒推值范围，及时停止扩展那些已无必要再扩展的子节点，即相当于剪去了博弈树上的一些分枝，从而节约了机器开销，提高了搜索效率。

具体的剪枝方法如下：

(1) 对于一个与节点 MIN，若能估计出其倒推值的上确界 β ，并且这个 β 值不大于 MIN 的父节点(一定是或节点)的估计倒推值的下确界 α ，即 $\alpha \geq \beta$ ，则就不必再扩展该 MIN 节点的其余子节点了(因为这些节点的估值对 MIN 父节点的倒推值已无任何影响了)。这一过程称为 α 剪枝。

(2) 对于一个或节点 MAX，若能估计出其倒推值的下确界 α ，并且这个 α 值不小于 MAX 的父节点(一定是与节点)的估计倒推值的上确界 β ，即 $\alpha \geq \beta$ ，则就不必再扩展该 MAX 节点的其余子节点了(因为这些节点的估值对 MAX 父节点的倒推值已无任何影响了)。这一过程称为 β 剪枝。

从算法中看到：

- (1) MAX 节点(包括起始节点)的 α 值永不减少；
- (2) MIN 节点(包括起始节点)的 β 值永不增加。

在搜索期间， α 和 β 值的计算如下：

- (1) 一个 MAX 节点的 α 值等于其后继节点当前最大的最终倒推值。
- (2) 一个 MIN 节点的 β 值等于其后继节点当前最小的最终倒推值。

2.4 输赢判断算法设计

因为每次导致输赢的只会是当前放置的棋子，输赢算法中只需从当前点开始扫描判断是否已经形成三子。对于这个子的八个方向判断是否已经形成三子。如果有，则说明有一方胜利，如果没有则继续搜索，直到有一方胜利或者搜索完整个棋盘。

三、实验内容

利用极大极小搜索及 α - β 剪枝实现一字棋设计

程序实现参考：

<https://github.com/yqtaowhu/DataStructureAndAlgorithm/tree/master/Algorithm/TicTacToe>

四、实验报告要求

- 1. 画出设计程序的流程图
- 2. 给出核心函数的设计思路

实验三 Fisher 线性分类器的设计与实现

一、实验目的

- 1) 掌握 Fisher 线性判别的基本原理
- 2) 利用 Fisher 线性判别解决基本的两类线性分类问题

二、实验原理

2.1 基本思想

若把样本的多维特征空间的点投影到一条直线上，就能把特征空间压缩成一维。那么关键就是找到这条直线的方向，找得好，分得好，找不好，就混在一起。因此 fisher 方法目标就是找到这个最好的直线方向以及如何实现向最好方向投影的变换。这个投影变换恰是我们所寻求的解向量，这是 fisher 算法的基本问题。

样本训练集以及待测样本的特征数目为 n 。为了找到最佳投影方向，需要计算出各类均值、样本类内离散度矩阵和总类间离散度矩阵、样本类间离散度矩阵，根据 Fisher 准则，找到最佳投影准则，将训练集内所有样本进行投影，投影到一维 Y 空间，由于 Y 空间是一维的，则要求出 Y 空间的划分边界点，找到边界点后，就可以对待测样本进行一维 Y 空间的投影，判断它的投影点与分界点的关系，将其归类。

2.2 实现步骤

- (1) 计算各类样本均值向量 m_i ， m_i 是各个类的均值， N_i 是 ω_i 类的样本个数。

$$m_i = \frac{1}{N_i} \sum_{X \in \omega_i} X \quad i = 1, 2, \dots, n \quad (1)$$

- (2) 计算样本类内离散度矩阵 S_i 和总类内离散度矩阵 S_w

$$\begin{aligned} S_i &= \sum_{X \in \omega_i} (X - m_i)(X - m_i)^T \quad i = 1, 2, \dots, n \\ S_w &= \sum_{i=1} S_i \end{aligned} \quad (2)$$

- (3) 计算样本类间离散度矩阵 S_b

$$S_b = (m_1 - m_2)(m_1 - m_2)^T \quad (3)$$

- (4) 求向量 W^*

我们希望投影后，在一维 Y 空间各类样本尽可能地分开，也就是说我们希望两类

样本均值之差 ($m_1 - m_2$) 越大越好, 同时希望各类样本内部尽量密集, 即希望类内

离散度越小越好, 因此, 我们可以定义 Fisher 准则函数: $J_F(W) = \frac{W^T S_b W}{W^T S_w W}$

(4)

使得 $J_F(W)$ 取得最大值的 W^* 为 $W^* = S_w^{-1}(m_1 - m_2)$ (5)

(5) 将训练集内所有样本进行投影

$$y = (W^*)^T X \quad (6)$$

如果 S_w 是非奇异的, 则要获得类间离散度与类内离散度的比值最大的投影方向

$J_F(W)$ 的满足下式:

$$J_F(W) = \arg \max \left| \frac{W^T S_b W}{W^T S_w W} \right| = [w_1, w_2, \dots, w_m] \quad (7)$$

其中 $[w_1, w_2, \dots, w_m]$ 是满足下式的 S_b 和 S_w 对应的 m 个最大特征值所对应的特征向量。

注意到该矩阵最多只有 $C-1$ 个非零特征值, C 是类别数。

三、实验内容

实验 1: 利用 LDA 进行一个分类的问题: 假设一个产品有两个参数柔软性 A 和刚性 B 来衡量它是否合格, 如下图所示:

参数 A	参数 B	是否合格
2.95	6.63	合格
2.53	7.79	合格
3.57	5.65	合格
3.16	5.47	合格
2.58	4.46	不合格
2.16	6.22	不合格
3.27	3.52	不合格

根据上图, 我们可以把样本分为两类, 一类是合格的产品, 一类是不合格的产品。通过 LDA 算法对训练样本的投影获得判别函数, 然后判断测试样本的类别, 即输入一个样本的参数, 判断该产品是否合格。

实验 2: 基于 ORL 人脸库, 实验样本主要来自于两个人, 每人 45 张图片, 共有 90 个样本, 其中的 80 个样本作为训练样本, 10 个作为测试样本。通过 LDA 实现两类问题的线

性判别。

实验 3：（选做）基于 ORL 人脸库，基于 Fisher 线性分类器实现多类人脸的识别问题。

四、实验结果及分析

要求：

- 1) 写清楚具体的实验步骤及结果
- 2) 对实验中遇到的问题有何解决方法

实验四 感知器算法的设计实现

一、实验目的

1. 熟悉感知器算法。
2. 掌握感知准则函数分类器设计方法。
3. 掌握感知器算法，利用它对输入的数据进行多类分类。

二、实验原理

1. 感知器基本原理

感知准则函数是五十年代由 Rosenblatt 提出的一种自学习判别函数生成方法，由于 Rosenblatt 企图将其用于脑模型感知器，因此被称为感知准则函数。其特点是随意确定的判别函数初始值，在对样本分类训练过程中逐步修正直至最终确定。

用对所有错分样本的求和来表示对错分样本的惩罚，代价函数定义如下：

如果样本 y_k 被错分，则有 $\alpha^T y_k < 0$ ，因此可定义如下的感知准则函数：

$$J_p(\alpha) = \sum_{y_j \in \mathcal{Y}^k} (-\alpha^T y_j)$$

其中 \mathcal{Y}^k 是被 α 错分样本的集合。

显然， $J \geq 0$ 。当代价函数 J 达到最小值 0 时，所有的训练向量分类都全部正确。为了计算代价函数的最小迭代值，可以采用梯度下降法设计迭代算法，即：

$$\begin{aligned} \alpha(k+1) &= \alpha(k) - \rho_k \nabla J \\ \nabla J &= \partial J_p(\alpha) / \partial \alpha = \sum_{y_j \in \mathcal{Y}^k} (-y_j) \\ \therefore \quad \alpha(k+1) &= \alpha(k) + \rho_k \sum_{y_j \in \mathcal{Y}^k} y_j \end{aligned}$$

下一时刻的权向量是把当前时刻的权向量向目标函数的负梯度方向调整一个修正量，其中 ρ_k 为步长。通过不断更新 w ，这种算法就称为感知器算法（perceptron algorithm）。可以证明，这种算法在经过有限次迭代之后是收敛的，也就是说，根据此规则修正权向量，可以让所有的特征向量都正确分类。

2. 单样本修正法

基础感知器算法在每一步更新系数时，是把所有错分的样本按照某个系数加到权向量上。但这样计算量大，且推广性不强。因此可以使用单样本修正法或小批量样本修正法来训练感知器。如单样本修正法如下：

单样本修正：

- 固定增量法：

(1) 初值 $\alpha(0)$ 任意

(2) 对样本 y_j ，若 $\alpha(k)^T y_j < 0$ ，则 $\alpha(k+1) = \alpha(k) + y_j$ ($\rho_k = 1$)

(3) 对所有样本重复 (2)，直至 $J_p = 0$

- 显然，如果错分条件改为 $\alpha(k)^T y_j < b$ ，则考虑了余量 b

- 变增量，如绝对修正法：
$$\rho_k = \frac{|\alpha(k)^T y_j|}{\|y_j\|^2}$$

收敛：对线性可分样本集，经过有限次修正后一定可以找到一个解 α^* 。

3. 利用感知器实现多类判别

基本算法如下：

(1) 样本规范化

(2) 每一类设定一个初始权向量 $w_i (i = 1, \dots, M)$

(3) 对第 i 类的样本 y_j ，若

$$w_i^T y_j \leq w_t^T y_j \quad t = 1, \dots, M, \text{ 但 } t \neq i$$

则：

$$w_i(k+1) = w_i(k) + y_j$$

$$w_t(k+1) = w_t(k) - y_j \quad t \neq i$$

(4) 对所有样本重复 (3)，直到满足

$$w_i^T y_j > w_t^T y_j \quad t = 1, \dots, M, \text{ 但 } t \neq i$$

三、实验内容

1. 数据生成及规范化处理

利用高斯模型，生成 N 类 ($N > 5$) 数据 (2D or 3D)，并对生成样本进行规范化处理

2. 基于生成数据，利用感知器准则实现多类分类，得到最终分界面的表达式。

3. 生成测试数据列，并对测试数据进行分类判别。

程序实现参考：感知器算法 Matlab 源码

四、实验结果及分析

1. 完成实验内容并对遇到的问题进行分析

2. 附完整实验程序代码

实验五 SVM 分类器的设计与应用

一、实验目的

1. 熟悉 SVM 的基本设计原理。
2. 掌握 SVM 的使用方法。
3. 利用 SVM 实现人脸识别。

二、实验原理

1. 人脸识别系统的基本框架如图 1 所示

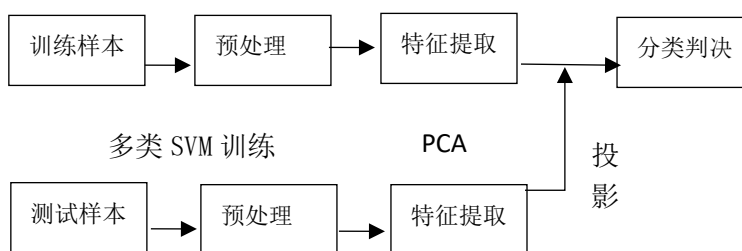


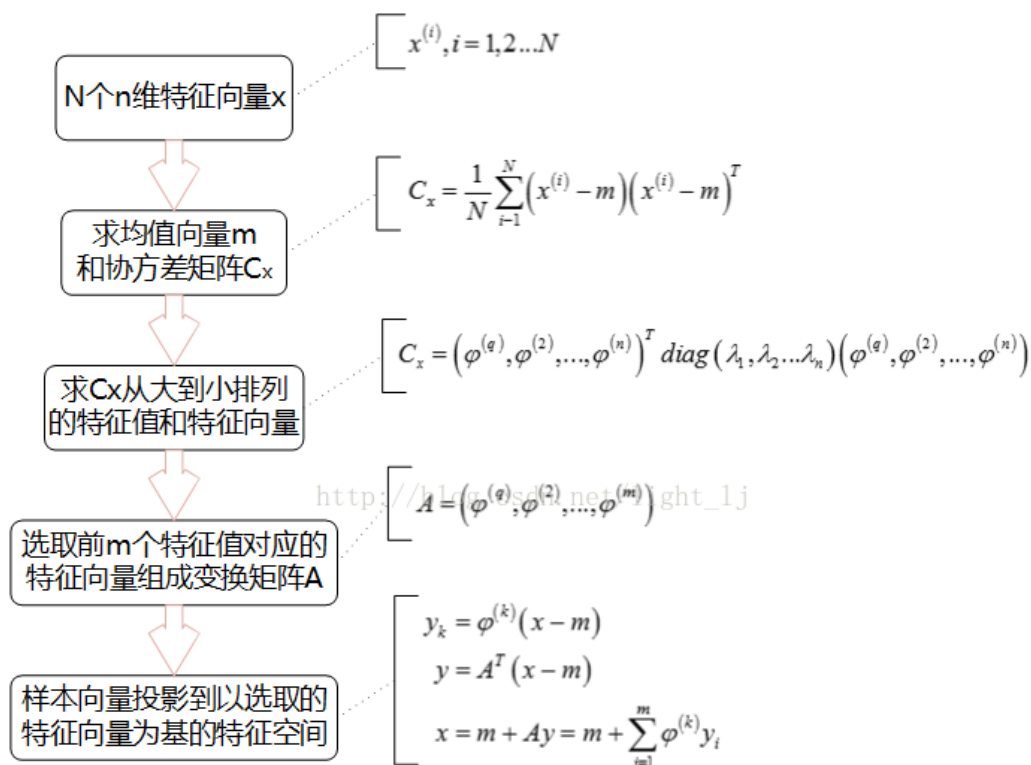
图 1 人脸识别系统的基本框架

2. 利用主成分分析 PCA 实现特征提取

图像识别中，常常用矩阵来表示人脸图像。然而，高维图像数据对整个识别系统的识别速度有限的，也不利于实时识别系统的实现。降维技术是解决这一问题的常用方法，使数据从原始图像高维空间转化为维数大大减小的特征空间，同时，又保留原始图像数据的绝大部分信息。

主成分分析利用 K-L 变换得到高维图像空间的一组正交基，保留其中较大的特征值对应的正交基，组成特征脸空间。将 PCA 用于人脸图像的最优表示，应用主分量重构人脸，提出特征脸（Eigenface）的概念，用 PCA 实现人脸图像的紧致表示，认为任何一幅图像都可以用一组特征脸的线性加权和来近似重构，其权重系数可以通过将人脸图像在本征脸空间投影得到，然后用投影到低维空间中基函数上的系数来表示人脸并进行识别，并采用奇异值分解（SVD）算法简化了特征脸的计算。

对于一幅 $M \times N$ 的人脸图像，将其每列相连构成一个 $D=M \times N$ 维的列向量。 D 就是人脸图像即图像空间的维数，设 n 是训练样本的数目， x_i 为第 i 幅人脸图像形成的人脸向量



将A的列向量 φ 看作低维空间的基，常量 m 不影响分布的特性，故将基向量的系数看作主成分 $y = (y_1, y_2, \dots, y_m)$ 就是原样本 x 在低维空间的表示

为了减少维数，把特征值从大到小排列，选取前 m 个，构成新的特征空间。由于这些特征向量对应的图像很象人脸，所以被称为“特征脸”。任何一幅图像都可以向特征脸子空间投影并获得一组坐标系数，这组坐标系数表明了该图像在特征脸子空间的位置，从而作为人脸识别的依据。

PCA 可去除原成分各维之间的相关性并进行降维，对降维后维数的选择要看对最小平方误差的要求，被舍去的协方差矩阵的特征根的平方和与所有特征根的平方和就是最小平方误差。

3. SVM 分类器设计

支持向量机 (Support Vector Machine, SVM) 是在统计学理论的基础上发展起来的新一代学习算法，它在文本分类、手写识别、图像分类、生物信息学等领域中获得较好的应用。

SVM 是一个二分类器，只能用于 2 类样本的分类，现在我们将它推广到多类问题。本文是对 SVM 进行推广到能够处理多类问题。可采用一对一的投票策略。将 A、B、C、D 4 类样本分成多个两类分类训练集，即 (A,B)、(A,C)、(A,D)、(B,C)、(B,D)、(C,D)，得到 6 个（对于 n 类问题，为 $n(n-1)/2$ 个）SVM 二分类器。在测试的时候，把测试样本 x 依次送入这 6 个二分

器，采取投票形式，最后得到一组结果。投票是以如下方式进行。

初始化： $\text{vote}(A)=\text{vote}(B)=\text{vote}(C)=\text{vote}(D)=0$ 。

投票过程：如果使用训练集(A,B)得到的分类器将 x 判定为 A 类，则 $\text{vote}(A)=\text{vote}(A)+1$ ，否则 $\text{vote}(B)=\text{vote}(B)+1$ ；如果使用(A,C)训练的分类器将 x 判定为 A 类，则 $\text{vote}(A)=\text{vote}(A)+1$ ，否则 $\text{vote}(C)=\text{vote}(C)+1$ ；...；如果使用(C,D)训练的分类器将 x 判定为 C 类，则 $\text{vote}(C)=\text{vote}(C)+1$ ，否则 $\text{vote}(D)=\text{vote}(D)+1$ 。

最终判决： $\text{Max}(\text{vote}(A), \text{vote}(B), \text{vote}(C), \text{vote}(D))$ 。如有两个以上的最大值，则一般可以简单地取第一个最大值所对应的类别。

三、 实验内容

1. 数据库的选择

可选取 ORL 人脸数据库作为实验样本，总共 40 个人，每人 10 幅图像，图像大小为 112*92 像素。图像本身已经经过处理，不需要进行归一化和校准等工作。实验样本分为训练样本和测试样本。首先设置训练样本集，选择 40 个人前 5 张图片作为训练样本，进行训练。然后设置测试样本集，将 40 个人后 5 张图片作为测试样本，进行选取识别。

2. 实验基本步骤

人脸识别算法步骤概述：

a) 读取训练数据集：

若 $\text{flag}=0$ ，表述读取原文件的前五幅图作为训练数据，若 $\text{flag}=1$ ，表述读取原文件的后五幅图作为测试数据，数据存入 `f_matrix` 中，每一行为一个文件，每行为 112*92 列。参见：`ReadFace.m`

b) 主成分分析法降维并去除数据之间的相关性；参见：`fastPCA.m`

c) 数据规格化；参见 `scaling.m`

d) SVM 训练（选取径向基和函数）得到分类函数；参见：`multiSVMtrain.m`

e) 读取测试数据、降维、规格化；参见：`multiSVM.m`

f) 用步骤 d 产生的分类函数进行分类（多分类问题，采用一对一投票策略，归位得票最多的一类）；参见：`main.m`

g) 计算正确率。

具体代码实现参加百度网盘：`facerecog_SVM`

3. 实验要求

1) 分别使用 PCA 降维到 20,50,100,200，然后训练分类器，对比分类结果，画出对比曲

线；

- 2) 变换 SVM 的 kernel 函数，如分别使用径向基函数和多项式核函数训练分类器，对比分类结果，画出对比曲线；
- 3) 使用交叉验证方法，变换训练集及测试集，分析分类结果。

四、 实验结果分析

- 1) 完成实验内容
- 2) 提交实验报告

实验六 卷积神经网络 CNN 框架的实现与应用

一、实验目的

1. 掌握卷积神经网络 CNN 的基本原理
2. 利用 CNN 实现手写数字识别

二、实验原理

利用 LeNet-5 CNN 框架，实现手写数字识别。

1. 网络层级结构概述如图 1：7 层神经网络

Input layer: 输入数据为原始训练图像

Conv1: 6 个 5*5 的卷积核，步长 Stride 为 1

Pooling1: 卷积核 size 为 2*2，步长 Stride 为 2

Conv2: 12 个 5*5 的卷积核，步长 Stride 为 1

Pooling2: 卷积核 size 为 2*2，步长 Stride 为 2

Output layer: 输出为 10 维向量

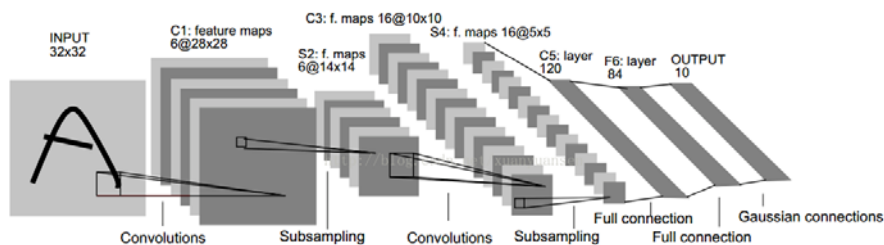


图 1: CNN 模型基本框架图

2. 实验基本流程

- (1) 获取训练数据和测试数据;
- (2) 定义网络层级结构;
- (3) 初始设置网络参数 (权重 W , 偏向 b) `cnnsetup(cnn, train_x, train_y)`
- (4) 训练超参数 `opts` 定义 (学习率, `batchsize`, `epoch`)
- (5) 网络训练之前向运算 `cnnff(net, batch_x)`
- (6) 网络训练之反向传播 `cnnbp(net, batch_y)`
- (7) 网络训练之参数更新 `cnnapplygrads(net, opts)`
- (8) 重复 (5) (6) (7), 直至满足 `epoch`
- (9) 网络测试 `cnntest(cnn, test_x, test_y)`

各函数实现参加百度网盘: CNN

三、实验内容

1. 数据集: MNIST 数据集, 60000 张训练图像, 10000 张测试图像, 每张图像 size 为 28*28
2. 利用 LeNet-5 CNN 框架, 实现手写数字识别。

四、实验结果及分析

1. 利用交叉验证方法, 分析识别结果
2. 分析网络参数 `opts` 设置对最终识别结果的影响, 画出相应的结果分析图