

# 目录

目录 .....	2
1 深度卷积神经网络 .....	1
1.1 引言 .....	1
1.2 基本分类 .....	2
1.3 传统人工神经网络结构 .....	2
1.4 深度卷积神经网络 .....	5
1.4.1 卷积 .....	6
1.4.2 共享权重和偏置 .....	8
1.4.3 池化 .....	8
1.4.4 激活函数 .....	9
1.4.5 网络的训练与学习 .....	11
1.5 小结 .....	13
2 基于深度学习的天波超视距雷达地海杂波识别 .....	15
2.1 引言 .....	15
2.2 地海杂波频谱数据分析 .....	16
2.2.1 数据集分组 .....	17
2.3 地海杂波识别算法 .....	18
2.3.1 数据预处理 .....	19
2.3.2 分类算法设计 .....	21
2.3.3 分类阈值设计 .....	26
2.4 仿真验证 .....	27
2.4.1 算法实现 .....	27
2.4.2 仿真结果分析 .....	29
2.4.3 特征可视化 .....	33
2.5 小结 .....	33
3 基于深度学习的辐射源识别 .....	35
3.1 引言 .....	35

3.2	辐射源信号分析 .....	35
3.2.1	模糊函数 .....	36
3.3	Open Set 分类器设计 .....	37
3.3.1	深度卷积神经网络分类器设计 .....	38
3.3.2	支持向量机Meta-Recognition 设计 .....	40
3.4	仿真实验与分析 .....	44
3.4.1	实验环境 .....	44
3.4.2	实验结果分析 .....	47
3.5	小结 .....	49
4	基于深度嵌入卷积聚类方法的地海杂波无监督分类 .....	51
4.1	引言 .....	51
4.2	卷积自编码网络 .....	53
4.2.1	自编码器 .....	53
4.2.2	卷积自编码器 .....	54
4.3	深度嵌入卷积聚类方法 .....	55
4.3.1	深度嵌入卷积聚类方法结构 .....	55
4.3.2	KL散度聚类方法 .....	56
4.4	仿真验证 .....	56
4.4.1	参数设置 .....	56
4.4.2	仿真结果分析 .....	57
4.5	小结 .....	57
5	总结 .....	59
5.1	本文的主要贡献 .....	59
5.2	后续的研究进展 .....	59
	参考文献 .....	61

# 1 深度卷积神经网络

## 1.1 引言

人工神经网络[1]是一种通过模仿大脑神经元行为进行信息处理的数学模型，但由于其无法承受大规模的参数和训练样本并且具有泛化能力差等问题。Fukushima[2]于1982年首次提出了卷积神经网络模型，Lecun 等人对神经网络传统算法在训练上面临的计算复杂度高等问题进行了改进，提出了基于梯度下降的优化算法[3]和BP算法[4]。2003年，Simard对卷积神经网络进行了简化[5]。Hinton 在2006年的两篇文章[6, 7]可以作为深度学习(Deep Learning,DL)正式提出的里程碑。Ciren等人[8]在2011年对神经网络进行改造使其可以通过GPU进行训练计算，并在大量的图像数据集对深度学习算法进行了测试，并且都取得了最好的成绩。其在工业界以及学术界掀起了巨大的浪潮，被应用于语音识别[9]、图像识别[10]和自然语音处理[11]等各种方面。专门针对于深度学习的芯片不断问世，例如谷歌公司的TPU[12]，大大提高了深度学习的应用场景。深度神经网络模型是一种非常强大的深度学习模型，他同时可以处理有监督和无监督学习任务。并且随着科技的发展，数据量越来越大，计算机并行能力也有了很大的提高。针对于海量数据，简单的线性模型由于无法充分利用计算能力，不再适用，可以预见在将来会有越来越多的工作应用到深度学习。目前，其内涵已经超出了传统的多层神经网络，甚至机器学习的范畴，逐渐朝着人工智能的方向快速发展[13]。

1982年，Kunihiko等人[2]首次将卷积神经网络模型的概念引入深度学习。后来许多学者在实践中对CNN的发展和理论分析作出了重大贡献。1989年，LeCun等人将基于梯度的学习方法[3]和BP算法[4]引入到CNN。2003年，Behnke写了一本总结CNN[14]的书。同年，Simard等人[5]对卷积神经网络进行了简化。2011年，Ciren等[8]进一步改进CNN并实现了GPU版本，使得CNN的训练识别速度有了巨大的提升，并使用CNN框架对多个图像数据库进行实验，并取得了最佳成果。

本章安排如下：2.2节对深度学习的基本分类进行了介绍，2.3节详细地阐述了传统人工神经网络结构的基本定义和神经元之间信息传递的过程，2.4节对本文主要利用的深度卷积神经网络的特性进行了介绍，2.5节进行本章总结。

## 1.2 基本分类

传统上可以把深度学习分为卷积神经网络 (Convolutional Neural Networks, CNN)、递归神经网络 (Recurrent Neural Networks, RNN)、长短时记忆网络 (Long short-term memory, LSTM)、深度信念网络 (Deep Belief NetWorks, DBN)、自编码器 (AutoEncoder)、稀疏编码 (Sparse Coding)、限制波尔兹曼机 (Restricted Boltzmann Machine, RBM) 等。

其中卷积神经网络是最流行的一种深度学习模型, 通过使用卷积层极大地减少了中间层的参数数目, 使学习效率更高并较少过拟合, 同时卷积操作独有的局部感受野 (local receptive fields)、共享权重 (shared weights) 和池化 (pooling) 三种特性也是处理序列元素分类识别的很重要的一点, 权重共享策略减少了需要训练的参数, 相同的权重可以让卷积核不受信号位置的影响来检测信号的特性, 使得训练出来的模型泛化能力更强; 池化运算可以降低网络的空间分辨率, 从而消除信号的微小偏移和扭曲。

递归神经网络是一种包含循环的, 允许信息持久化的神经网络模型。传统的前馈神经网络中, 单独的输入完全确定了余下层的神经元的激活值。而对于递归神经网络, 隐藏层和输出层的神经元的激活值不仅由当前的网络输入决定, 而且包含了前面的输入的影响。长短时记忆网络是一种特殊的递归神经网络, 主要用于解决递归神经网络前期模型难以训练的问题。其通过刻意设计的单元结构, 在递归神经网络的基础上添加了元胞状态 (cell state) 用来保存长期的状态, 然后通过门函数来控制此长期状态。

深度信念网络是一个概率生成模型, 是由多个限制波尔兹曼机组成, 这些网络被“限制”为一个可见层和一个隐藏层, 层间存在连接, 但是层内的单元间不存在连接。隐藏单元被训练来捕捉在可见层表现出来的高阶数据的相关性。

通过对于以上几种最常用的深度学习方法的介绍, 我们可以发现卷积神经网络是最适合处理本文这种静态类型的数据, 循环或者说是不同时刻的输入对于地海杂波类型的识别并没有提高, 故递归神经网络和长短时记忆网络显然不适合本问题。另一方面, 深度信念网络的生成模型并不关心不同类别之间的最优分类面的位置, 故其用于分类问题时, 分类精度没有判别模型高。且其学习的是数据的联合分布, 相比其他算法具有更高的复杂性。

## 1.3 传统人工神经网络结构

人工神经网络是一种模拟大脑神经系统的算法, 其可以从海量的训练样本中学习到一个权重函数, 用来进行模式识别、分类等。神经网络主要是利用将许多个单一神

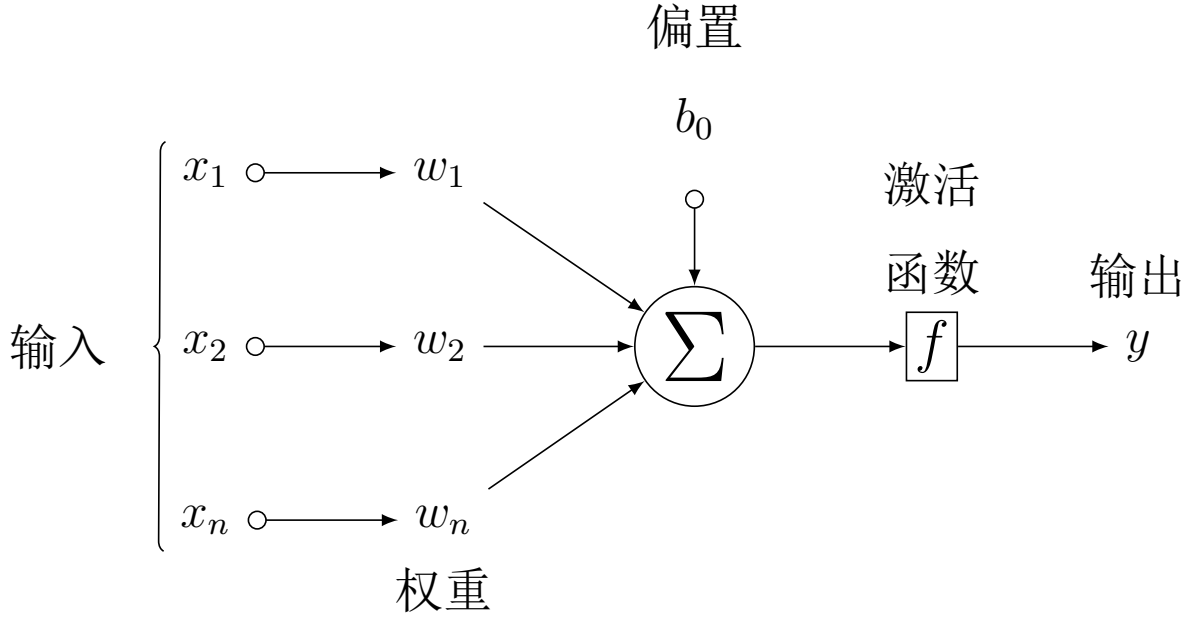


图 1-1: 神经元

神经元(也称作感知器)联结在一起，一个神经元的输出就可以是另一个神经元的输入，形成一个有向无环的网络结构。首先介绍单一神经元的结构，如图1-1所示，神经元一般具有多个输入 $x = [x_1, x_2, \dots, x_n]$ ，这些输入可以取0和1中的任意值。神经元对于每一个输入有权重 $w = [w_1, w_2, \dots, w_n]$  和一个总的偏置 $b_0$ ，其输出为：

$$y = f(w \cdot x + b) \quad (1-1)$$

这里的 $f$ 为该神经元的激活函数，我们在后面进行描述。

上述多个神经元分层互联即可以形成神经网络。图1-2是一个简单的神经网络，其中圆圈表示神经网络的节点，也即神经元。对于一个具有 $n_l$ 层的神经网络，如将第 $l$ 层用 $L_l$ 表示，则有最左边的一层 $L_1$ 为输入层，最右的一层 $L_{n_l}$ 为输出层。由于不能在训练样本集中观测到中间所有节点的值，故将这些层称做隐藏层。本例中网络的层数 $n_l = 3$ ，输入层具有4个输入单元，隐层具有5个隐藏单元，输出层只有一个输出单元。

根据输入 $x_i$ 求解输出 $y$ 的主要过程为前向传播。我们用 $w_{ij}^{(l)}$ 表示第 $l$ 层的第 $j$ 单元与第 $l+1$ 层的第 $i$ 单元之间的权重， $b_i^{(l)}$ 是第 $l+1$ 层中第 $i$ 单元的偏置项。因此， $w^{(l)}$ 为第 $l$ 层权重组成的向量， $w^{(1)} \in \mathbb{R}^{4 \times 3}$ ， $w^{(2)} \in \mathbb{R}^{1 \times 5}$ 。根据图1-1和公式1-1可以知道每个神经元节点的输出是经过激活函数的激活值，可以用 $y_i^{(l)}$ 表示第 $l$ 层的第 $i$ 单元的输出值。当 $l = 1$ 时， $y_i^{(1)}$ 为就是第 $i$ 个输入值，也即有 $y_i^{(1)} = x_i$ 。当权重已知的时候，我们就可以

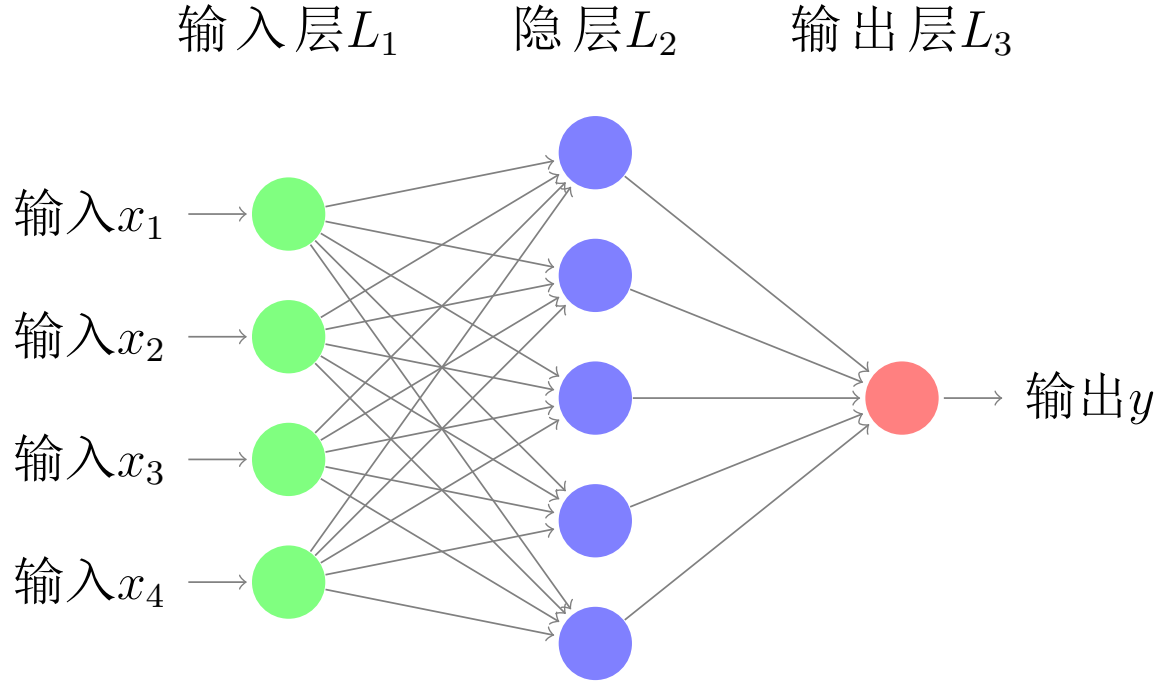


图 1-2: 神经网络

根据公式1-1求解此传播过程，进而求解最终的输出值 $y$ 。从 $L_1$ 层到 $L_2$ 层的过程为：

$$y_1^{(2)} = f(w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + w_{13}^{(1)}x_3 + w_{14}^{(1)}x_4 + b_1^{(1)}) \quad (1-2)$$

$$y_2^{(2)} = f(w_{21}^{(1)}x_1 + w_{22}^{(1)}x_2 + w_{23}^{(1)}x_3 + w_{24}^{(1)}x_4 + b_2^{(1)}) \quad (1-3)$$

$$y_3^{(2)} = f(w_{31}^{(1)}x_1 + w_{32}^{(1)}x_2 + w_{33}^{(1)}x_3 + w_{34}^{(1)}x_4 + b_3^{(1)}) \quad (1-4)$$

$$y_4^{(2)} = f(w_{41}^{(1)}x_1 + w_{42}^{(1)}x_2 + w_{43}^{(1)}x_3 + w_{44}^{(1)}x_4 + b_4^{(1)}) \quad (1-5)$$

$$y_5^{(2)} = f(w_{51}^{(1)}x_1 + w_{52}^{(1)}x_2 + w_{53}^{(1)}x_3 + w_{54}^{(1)}x_4 + b_5^{(1)}) \quad (1-6)$$

根据同样的计算过程可以得到：

$$y = y_1^{(3)} = f(w_{11}^{(2)}y_1^{(2)} + w_{12}^{(2)}y_2^{(2)} + w_{13}^{(2)}y_3^{(2)} + w_{14}^{(2)}y_4^{(2)} + w_{15}^{(2)}y_5^{(2)} + b_1^{(2)}) \quad (1-7)$$

我们只需要把公式1-6代入1-7就可以计算得到输入 $x_i$ 与输出 $y$ 之间的关系。为了使表达更清晰，我们用 $z_i^{(l)}$ 表示第 $l$ 层第 $i$ 单元输入加权和，则有：

$$z_i^{(l)} = \sum_{j=1}^n w_{ij}^{(l-1)}x_j + b_i^{(l)} \quad (1-8)$$

因此公式1-6变为

$$y_i^{(l)} = f(z_i^{(l)}) \quad (1-9)$$

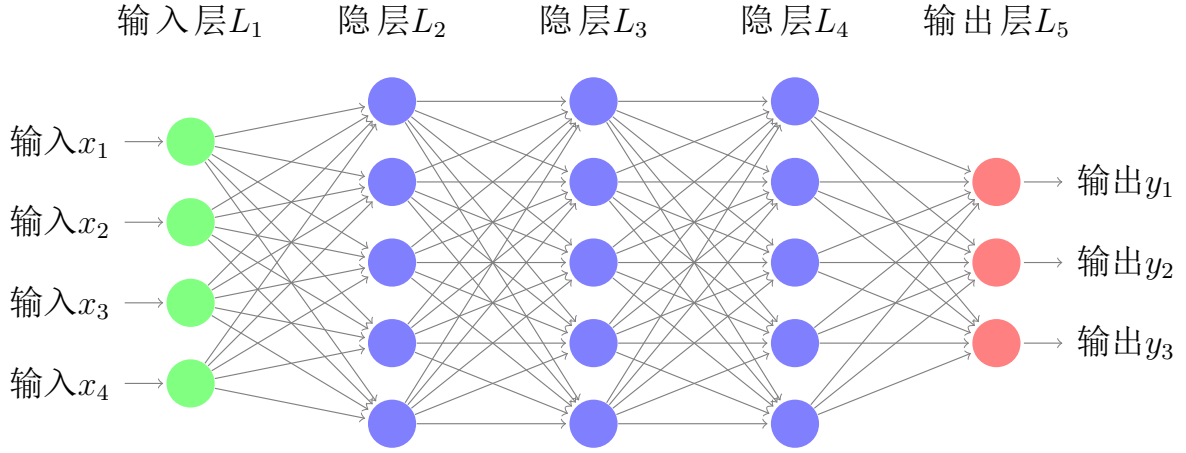


图 1-3: 具有多个输出单元的神经网络示意图

则整个前向传播过程可以简化为:

$$z^{(l+1)} = w^{(l)}y^{(l)} + b^{(l)} \quad (1-10)$$

$$h^{(l+1)} = f(z^{(l+1)}) \quad (1-11)$$

其中,  $b^{(l)}$  为第  $l+1$  层偏置项组成的向量,  $h^{(l)}$  为第  $l$  层各个单元输出值组成的向量。

根据上述计算过程, 对于一个具有  $n_l$  层的前馈神经网络 (在网络中没有闭环或者回路), 其第1层是输入层, 第  $n_l$  层是输出层, 中间的所有隐藏层  $L_l$  均与下一层  $L_{l+1}$  全连接。我们只需要逐个计算每一个层中的节点的输出值, 然后据此得到下一层的输出值。对于一个复杂的神经网络一般会具有多个隐藏层以及输出层中可以有多个输出单元。图1-3的神经网络有三层隐藏层:  $L_2$ 、 $L_3$  及  $L_4$ , 输出层  $L_5$  有三个输出单元。

## 1.4 深度卷积神经网络

卷积神经网络是深度学习中的一种重要算法, 在分类等很多领域具有很大的优势, 该方法经常被用于图像识别、语音识别等问题。卷积神经网络是一种特殊的人工神经网络, 由多对卷积层(Convolutional layer)和池化层(Pooling layer), 以及全连接层(Full-connected layer)组成。图1-4为一个典型的卷积神经网络, 最左边的为输入层, 其中的神经元为输入神经元。最右边的输出层包含有一个或者多个输出神经元, 其余的是多组卷积和池化层, 有限数量的完全连接的隐藏层。

网络的前向传播主要是卷积和池化操作, 其网络参数的更新采用误差反向传播(Error Backpropagation)算法。相比于传统人工网络, 卷积神经网络有以下几个特点:

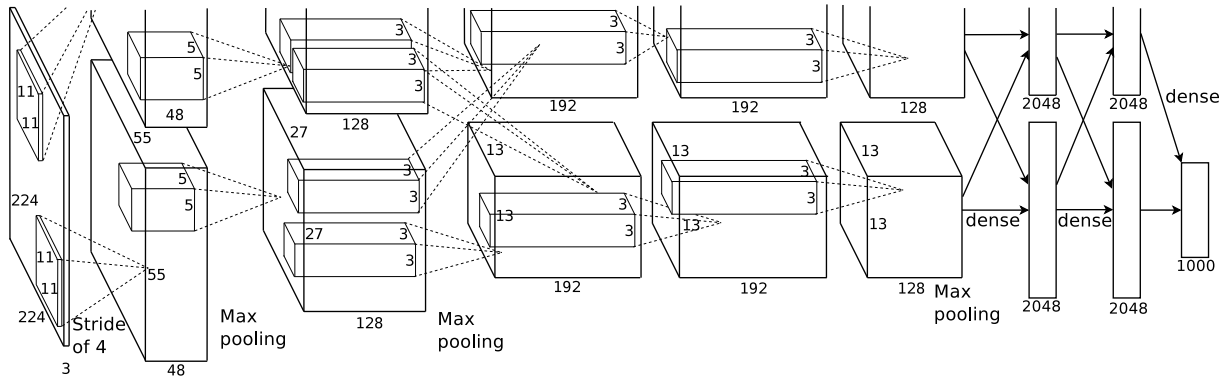


图 1-4: AlexNet 卷积神经网络模型[10]

- 输入为原始数据，对原始数据进行卷积运算，然后提取从最后一步生成的卷积数据的特征，丰富了识别中使用的特征。
- 多个卷积核可以利用输入数据中的局部结构，将整个输入空间划分成很小的隐藏单元。将各个隐藏单元的权重构建得到的卷积核作用于整个输入空间，从而得到特征向量。利用这种机制，我们不仅大大减少了参数数量同时提高了数据的平移不变性。
- 具有超过两个的隐藏层以及可以逐层初始化模型的参数，通过更多的隐藏层可以更好地对事物的特征进行学习表达。

#### 1.4.1 卷积

深度卷积神经网络在特征提取过程中一个主要操作为卷积，在前向计算过程中，对于输入的一定区域的隐藏层的特征向量 $x$ 和滤波器（权重） $w$ 点乘后得到新的特征向量，然后滑过一个个滤波器，以产生输出特征向量。权重实际上是一个四维张量，第一维 $F$ 是卷积输入层的特征向量的数量，另一维 $F_p$ 是卷积输出层的特征向量的数量，另外两维是在宽度和高度方向给出的卷积核的大小，也称作感受野。

对于传统的神经网络，每个输入元素会连接到每个隐藏神经元。然而，在卷积神经网络中，我们只是把输入数据进行小的、局部区域的连接，也即第一个隐藏层中的每个神经元会连接到一个输入神经元的一个区域。这个输入向量的区域被称为隐藏神经元的局部感受野。它是输入向量上的一个小窗口，对于每个连接学习一个权重而隐藏神经元同时也学习一个总的偏置。通过在整个输入频谱数据上交叉移动局部感受野，可以构建起第一个隐藏层。

感受野允许用一个子集来代替整个输入数据。它的目的是在输入数据中搜索相



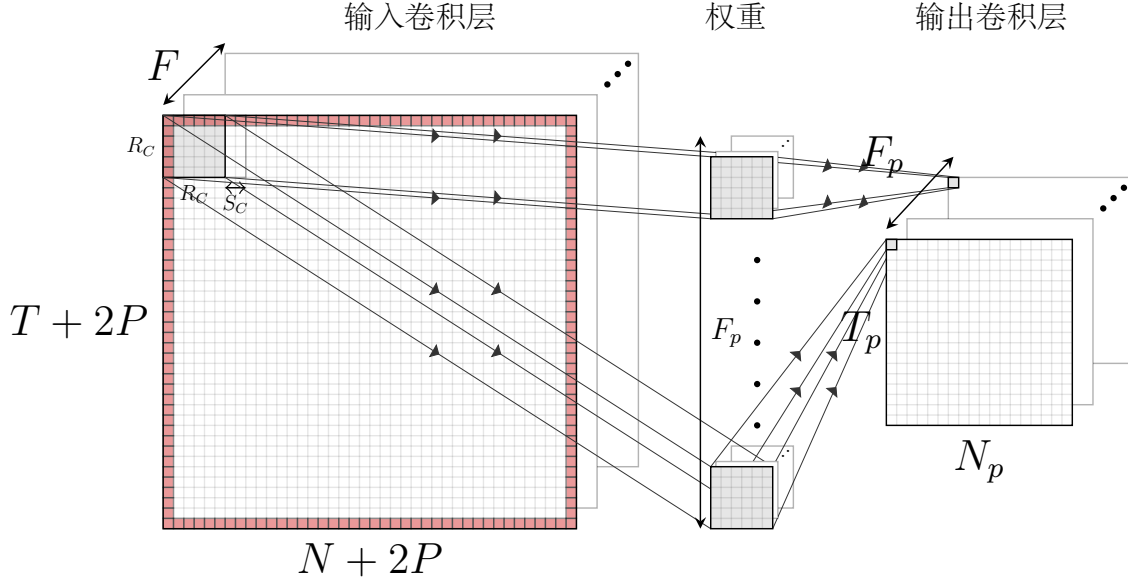


图 1-5: 卷积示意图

似的模式，而不管模式在哪里，也即平移不变性。输出图像的宽度和高度也由步长（stride）确定，步长为在进行再次应用卷积运算之前在垂直、水平方向上滑动的像素的数目。我们以图1-5这个对一个图像进行二维卷积的操作来介绍卷积运算。

其中 $R_C$  为卷积感受野的大小， $S_C$ 为卷积的步长。根据输入高度 $T$ 和宽度 $N$ ，可以计算输出图像的高度和宽度：

$$N_p = \frac{N + 2P - R_C}{S_C} + 1, \quad T_p = \frac{T + 2P - R_C}{S_C} + 1 \quad (1-12)$$

通常对输入数据进行边界扩充（padding）来使得输入特征图的宽度和高度满足 $N = N_p = T = T_p$ ，这样当前进步长 $S_C = 1$ 时，可以得到

$$P = \frac{R_C - 1}{2} \quad (1-13)$$

对于一个给定的层 $L_n$ ，卷积运算的计算过程如下：

$$y_{flm}^{(\nu)} = \sum_{f'=0}^{F_{\nu}-1} \sum_{j=0}^{R_C-1} \sum_{k=0}^{R_C-1} w_{f'jk}^{(o)f} h_{f'S_Cl+jS_Cm+k}^{(\nu)} \quad (1-14)$$

其中 $o$ 表示网络中的第 $o+1$ 个卷积核。 $\nu \in [0, N-1]$  表示网络中的第 $\nu$ 个隐层， $f \in [0, F_{\nu+1}-1]$ 为输入卷积层的第 $f$ 个特征图， $l \in [0, N_{\nu+1}-1]$ 为输入卷积层的第 $\nu$ 个特征图的宽度单位， $m \in [0, T_{\nu+1}-1]$ 为输入卷积层的第 $\nu$ 个特征图的高度单位。因此可以得到 $S_Cl+j \in [0, N_{\nu}-1]$ ， $S_Cl+m \in [0, T_{\nu}-1]$ 。于是可以通过添加激活函数计算隐藏单元的输出。考虑到边界填充可以得到下式

$$h_{fl+Pm+P}^{(\nu+1)} = f(y_{flm}^{(\nu)}) \quad (1-15)$$

每个滤波器只关心数据的部分特征，当出现它学习到的特征的时候，就会呈现激活态。

#### 1.4.2 共享权重和偏置

上面已经说过对于每个隐藏神经元具有一个偏置和连接到它的局部感受野的权重，同时对于该层的所有的隐藏神经元中每一个使用相同的权重和偏置。也即对第 $j$ 个隐藏神经元，输出为：

$$h_j = f(b + \sum_{m=1}^M w_m y_{j+m}) \quad (1-16)$$

这里 $f$ 是神经元的激活函数， $b$ 是偏置的共享值， $w_m$ 是一个共享权重的 $1 \times M$ 向量， $y_k$ 表示位置 $k$ 的输入激活值。这意味着第一个隐藏层的所有神经元检测完全相同的特征，只是在输入数据的不同位置，因此卷积网络可以很好地适应数据偏移情况。

#### 1.4.3 池化

我们在通过卷积获得了特征之后，下一步我们希望利用这些特征去做分类。理论上讲，人们可以用所有提取得到的特征去训练分类器，例如Softmax分类器（多分类的逻辑回归分类器），但这样做面临着计算量的挑战，除此以外过多的特征向量，也容易导致过拟合。由于本文的雷达信号具有一种“静态性”属性，在一个数据区域有用的特征极有可能在另一个区域同样适用。因此，为了描述数据量较多的数据，一个很自然的想法就是对不同位置的特征进行聚合统计，例如，可以计算频谱数据上一段频率范围内的某个特征的最大值(或平均值)。这些经过采样的统计特征相比使用所有提取得到的特征不仅具有低得多的维度，同时还不容易过拟合，在一定程度上会改善结果。这种聚合的操作称为池化。池化操作是卷积神经网络中一种常用的用来对数据进行降维处理的方法。一般有平均池化和最大池化，主要过程为选取输入特征向量的 $F$ 一个池化感受野 $R_p$ 和步长 $S_p$ 选取其最大或者平均元素来代表该区域，得到输出特征向量 $F_p = F$ ，其中宽度 $N_p < N$ ，高度 $T_p < T$ 。在池化操作过程中，我们一般不考虑输入隐藏层的边界扩充值，因此在公式1-17中存在 $+P$ 这一部分。

对于给定的第 $\nu$ 个平均池化操作的公式如下：

$$y_{flm}^{(\nu)} = \sum_{j,k=0}^{R_p-1} h_{f S_p l + j + P S_p m + k + P}^{(\nu)} \quad (1-17)$$

最大池化的公式为：

$$y_{flm}^{(\nu)} = \max_{j,k=0}^{R_p-1} h_{f S_p l + j + P S_p m + k + P}^{(\nu)} \quad (1-18)$$

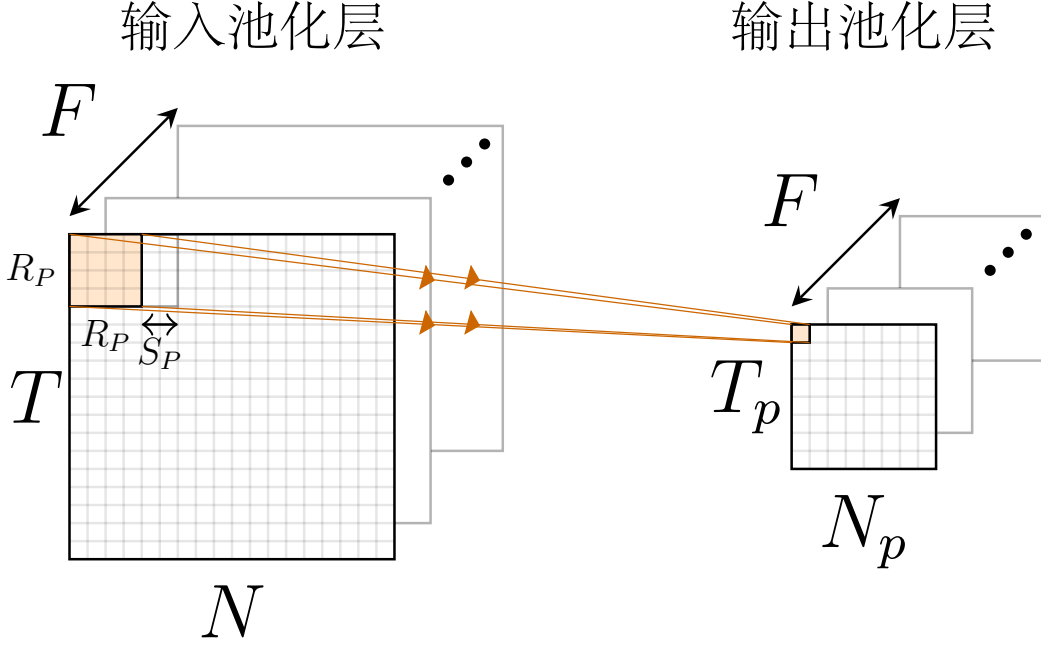


图 1-6: 池化示意图

其中,  $\nu \in [0, N - 1]$  表示网络中的第 $\nu$ 个隐层,  $f \in [0, F_{\nu+1} - 1]$ ,  $l \in [0, N_{\nu+1} - 1]$  以及  $m \in [0, T_{\nu+1} - 1]$ 。因此有  $S_P l + j \in [0, N_\nu - 1]$ ,  $S_P l + m \in [0, T_\nu - 1]$ 。最大池化是最常用的一种池化方式, 本文所用的神经网络结构均使用最大池化。对于第 $t$ 次批采样, 我们用  $j_{flm}^{(t)(p)}$ ,  $k_{flm}^{(t)(p)}$  来表示在  $l, m$  处特征向量图的最大值, 则公式1-18变为下式:

$$h_{fl+Pm+P}^{(t)(\nu+1)} = y_{flm}^{(t)(\nu)} = h_{fS_P l + j_{flm}^{(t)(p)} + P S_P m + k_{flm}^{(t)(p)} + P}^{(t)(\nu)} \quad (1-19)$$

深度卷积神经结构具有过拟合的自然趋势, 虽然可以通过权重共享来减少参数的数量。但是由于大多数情况下, 估计集的数量比训练集大一个数量级, 使得神经网络模型的泛化能力不足。在每个训练迭代中, 每个隐藏单元以预定概率被随机删除, 删除后学习过程继续。这些被称作dropout的随机扰动有效地防止了神经网络学习过程的依赖关系, 并在隐藏的单元之间创建了复杂的关系。这样增加了网络模型的复杂度, 从而提高深度神经网络模型的泛化能力。

#### 1.4.4 激活函数

在公式1-7中,  $f(x)$ 被用于每一个神经元节点, 因此激活函数的选择是构建卷积神经网络模型的一个非常重要的方面。我们在本节介绍几种常用的激活函数。

##### sigmoid 激活函数

在传统人工神经网络中常用的激活函数为sigmoid激活函数 (S型激活函数), 其取

值范围为 $(0, 1)$ ，其定义如下式：

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \quad (1-20)$$

对其求导可以得到：

$$\sigma'(x) = \sigma(x) (1 - \sigma(x)) \quad (1-21)$$

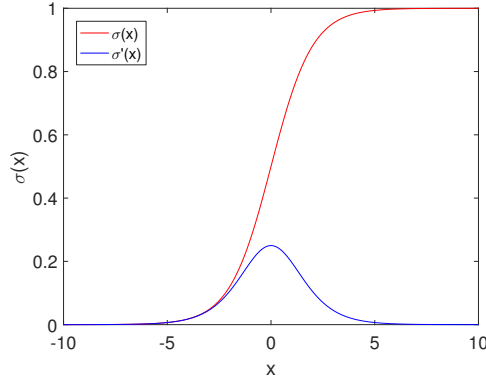


图 1-7: sigmoid 激活函数及其导数

### **tanh 激活函数**

tanh 激活函数的取值范围为 $(-1, 1)$ ，其定义如下：

$$f(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (1-22)$$

对其求导可以得到：

$$\tanh'(x) = 1 - \tanh^2(x) \quad (1-23)$$

### **ReLU 激活函数**

ReLU(Rectified Linear Unit)的取值范围为 $[0, +\infty)$ ，其定义如下式：

$$f(x) = \text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (1-24)$$

对其求导可以得到：

$$\text{ReLU}'(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (1-25)$$

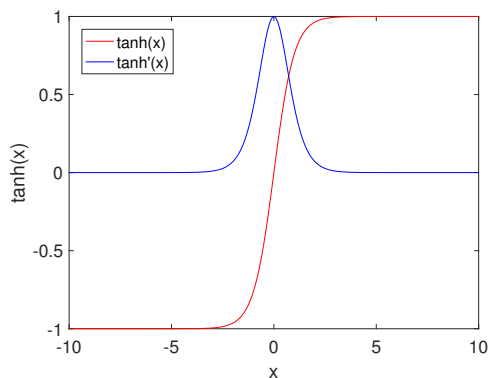


图 1-8: tanh2 激活函数及其导数

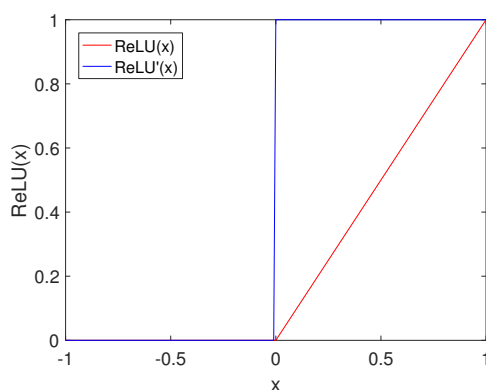


图 1-9: ReLU 激活函数及其导数

ReLU具有优于传统激活函数的几个优点：更快的计算速度和更有效的梯度传播（它们不像S形单元那样饱和），生物学可能性和稀疏激活结构。尽管它结构简单，但仍然保持足够的辨别性质。其缺点之一是随机权重的初始状态，多个单位可能过早地落入死区（零输出的恒定梯度）。因此，当与整个连接层进行全连接时，Sigmoid激活函数的效果更好。

#### 1.4.5 网络的训练与学习

参照上文的符号定义，用符号 $x$ 表示训练输入的向量集， $w$ 表示所有的网络中权重的集合， $b$ 是所有的偏置，用 $y = f(w \cdot x + b)$ 表示对应的期望输出。学习算法的主要目的是找到一个权重 $w$ 和偏置 $b$ 使得网络的输出 $y$ 可以拟合所有的训练输入 $x$ 。为此可以定义一个损失函数或代价函数(loss function) $C(w, b)$ 。针对于不同的问题可以选取不同的损失函数，公式1-26利用均方误差函数作为损失函数。

$$C(w, b) \equiv \frac{1}{2n} \sum_x ||y - a||^2 \quad (1-26)$$

其中， $n$ 是训练输入数据的个数， $a$ 表示当输入为 $x$ 时的输出向量。

从定义可以看出， $C(w, b)$ 越小说明分类越准确，那么训练神经网络的目的就是找到使得损失函数 $C(w, b)$ 最小的权重和偏置。为了求解该问题，将上述问题一般化，原问题变为最小化任意的具有 $m$ 个变量的多元实值函数 $C(v)$ ， $v = v_1, v_2, \dots, v_m$ 。这种具有大量变量的函数的解析解的求解是极其复杂的，一个比较合理的思路为利用数值计算的方法求取其极值点。每次对于 $C$ 中的自变量添加一个微小的变化 $\Delta v$ ，根据此变化反映出来的 $C$ 的变化 $\Delta C$ 更新下次的微小变化，从而使得 $C$ 可以持续减小。对 $C$ 中自变量的变化 $\Delta v = (\Delta v_1, \dots, \Delta v_m)^T$ ， $\Delta C$ 将会变为

$$\Delta C \approx \nabla C \cdot \Delta v \quad (1-27)$$

，这里的梯度 $\nabla C$ 定义如下：

$$\nabla C \equiv \left( \frac{\partial C}{\partial v_1}, \dots, \frac{\partial C}{\partial v_m} \right)^T \quad (1-28)$$

其把 $v$ 的变化关联为 $C$ 的变化，假设选取 $\Delta v = -\eta \nabla C$ ，其中的 $\eta$ 是学习率，一般取一个很小的正数，则公式1-27变为：

$$\Delta C \approx -\eta \nabla C \cdot \nabla C = -\eta \|\nabla C\|^2 \leq 0 \quad (1-29)$$

也即如果利用更新规则 $v \rightarrow v' = v - \eta \nabla C$ ， $C$ 会持续减小，此更新规则即为最基本的学习算法，梯度下降算法。图1-10展示了梯度下降的示意图，其跟随当前位置最陡的下降的方向也即负梯度，其中学习率 $\eta$ 描述在每个迭代步骤中下降的步长。可以根据选择不同的损失函数 $C$ 或者通过计算来完成学习率的选择等各种技术对学习算法进行优化。

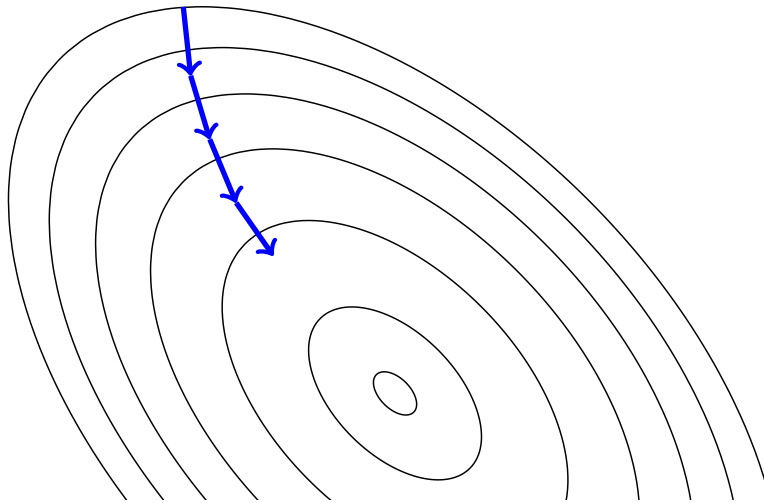


图 1-10: 梯度下降示意图

根据梯度下降的思想，我们可以结合卷积神经网络中的反向传播，获得对于神经网络训练流程图的训练流程图，如图1-11所示。

## 1.5 小结

本章介绍了深度学习的相关基础。首先是其基本分类，然后针对于本文利用的深度卷积神经网络，介绍了传统的神经元的结构和卷积神经网络基于此的改进。在最后讨论了深度卷积神经网络在雷达信号处理中的应用。

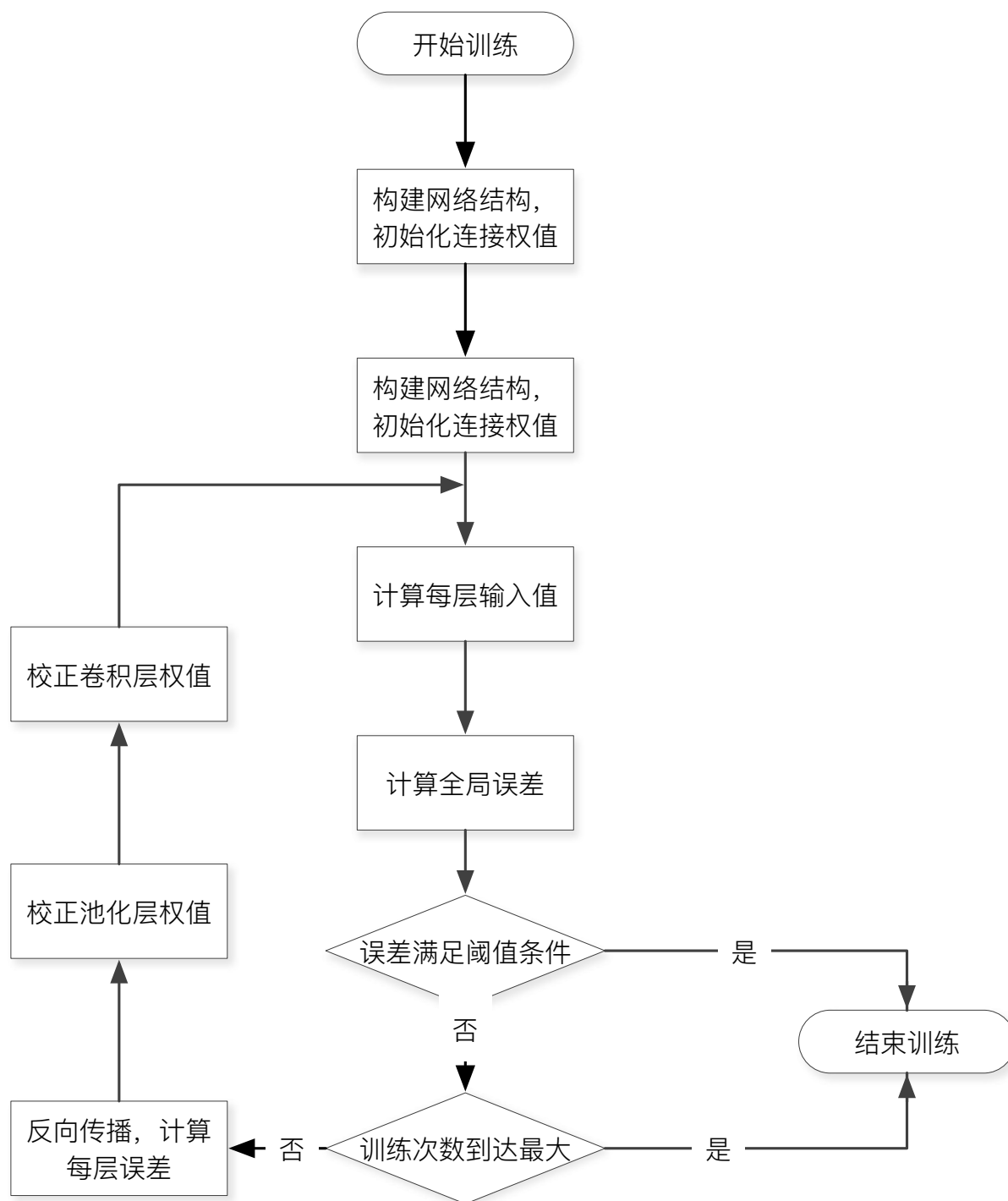


图 1-11: 深度卷积神经网络算法训练流程图——该图需要修改



## 2 基于深度学习的天波超视距雷达地海杂波识别

### 2.1 引言

天波超视距雷达目标定位精度依赖于传播模式的准确识别以及坐标配准系数的精确测量。电离层传播的复杂性使得传播模式很难精确确定，而电离层探测子系统独立于主雷达工作导致其提供的坐标配准参数与主雷达量测回波存在不一致性、误差大等问题，从而造成电离层传播模式识别正确率低、目标定位精度差等。天波超视距雷达地海杂波识别是一种基于无源信标（远海区域的岛屿等陆地）获得坐标配准系数的技术。鉴于远海地区有源设备的布置面临着较大的困难，通过区分识别地海杂波、构建地海边界轮廓、与先验地理轮廓信息匹配可同样提供坐标配准修正参数，改善周围航迹目标的定位精度。受分辨率低、定位精度差、系统偏差大、电离层多模、多路径传播等因素影响，天波雷达地海杂波识别技术存在很大挑战，主要体现在以下几个方面：

- 天波雷达的距离分辨率为7.5 – 30公里，方位分辨率为0.582 – 1.067度，低分辨率影响地海特性的判别以及匹配精度；
- 电离层状况变化情况十分复杂，导致地海杂波的特性并不稳定，区分地海杂波特性的Bragg峰会发生偏移甚至某个峰会消失，对地海杂波的建模影响很大，传统的建模方法一般根据实际数据选取一种分布来描述雷达杂波，如瑞利分布、威布尔分布、K分布等。但天波超视距雷达获得的杂波模型一直变化，这种建模方法不能得到很好的结果；
- 传统的地海杂波的分类特征很难定量描述。一个熟练的操作人员可能很熟练地区分出地海杂波，但是这部分却很难利用数学模型对其描述；

本章提供了一种新的在复杂的电离层环境下的基于深度神经网络的地海杂波识别技术，避免了对杂波进行建模选取特征的方法，从根本上避免了传统方法所面临的困难。构建了适用于天波超视距雷达地海杂波类型识别的卷积神经网络，利用大量训练数据对卷积神经网络进行训练，提取合理的特征；然后，利用提取的特征对实时雷达地海杂波回波进行在线分类识别。我们利用不同电离层条件、雷达工作条件、位置和时间下的频谱数据对我们所提出的算法进行了验证，实验结果证明我们的算法大大提高了天波超视距雷达地海杂波的识别正确率与实时性。

本章安排如下：3.2节对天波超视距雷达地海杂波信号进行了分析，并对数据集进行分组，3.3节构建了本章的基于深度学习的分类器，详细阐述了根据实际数据的特性构建分类器的过程，3.4节利用实际数据对于本章的分类器与已有方法进行对比，验证了本章提出方法的性能，3.5节进行本章总结。

## 2.2 地海杂波频谱数据分析

对于一般的进行天波雷达地海杂波识别的文章，其主要利用海杂波的Bragg特性，但是在实际情况中，受电离层条件、工作环境等的影响，会出现很多频谱很难分辨的情形，如图2-1所示。

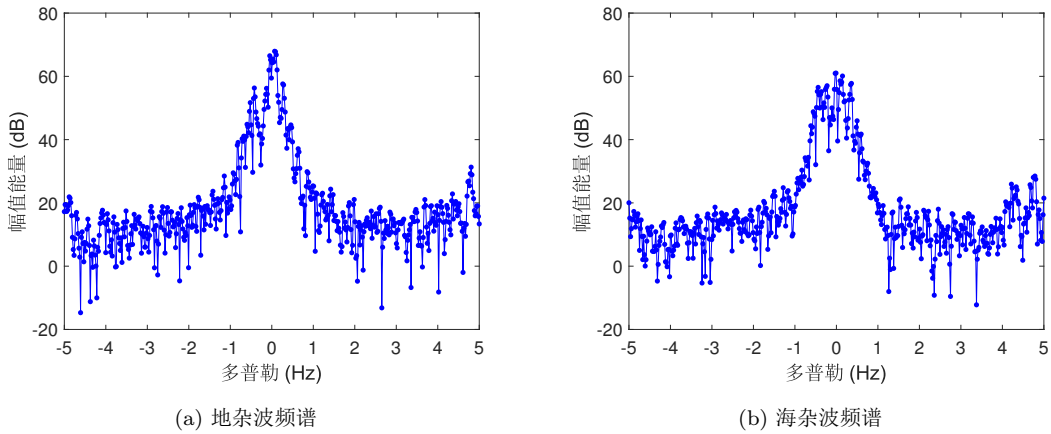


图 2-1: 两幅无法轻易区分的地海杂的频谱波示意图。图2-1a 的Bragg峰具有小的频偏，图2-1b 作为海杂波却无法容易的分辨出Bragg峰。

在天波超视距雷达的工作过程中，不同的时间和地理位置对应于不同的电离层条件，不同的电离层条件将严重影响频谱的状态。对于不同的雷达工作条件，由于发射频率变化，回波频谱密度和幅度也将随之变化。通过观察、学习大量不同条件的频谱数据，可以更全面地区分地海杂波。

在本章的问题中，我们选取了几种不同的雷达工作配置（工作频率、朝向等）下的数据。多普勒频率的范围可以是-5Hz至5Hz，也可以是-20Hz至20Hz。我们分别以不同的频率训练数据。此外，对于相同频率的数据，我们通过人工辨识的方法从中抽取可以准确判定为地或者海的样本数据用于实验的训练和测试。同时，为了保证样本的多样性，对于同一个雷达参数下的数据，我们根据季节、地理位置、一天的早中午进行了选择，确保可以覆盖尽可能多的情形。

我们所有的数据都来自不同时间，不同地点和不同雷达配置的频谱数据。我们分

析了所有频谱数据，并选择一些典型的频谱数据，如图2-2所示。

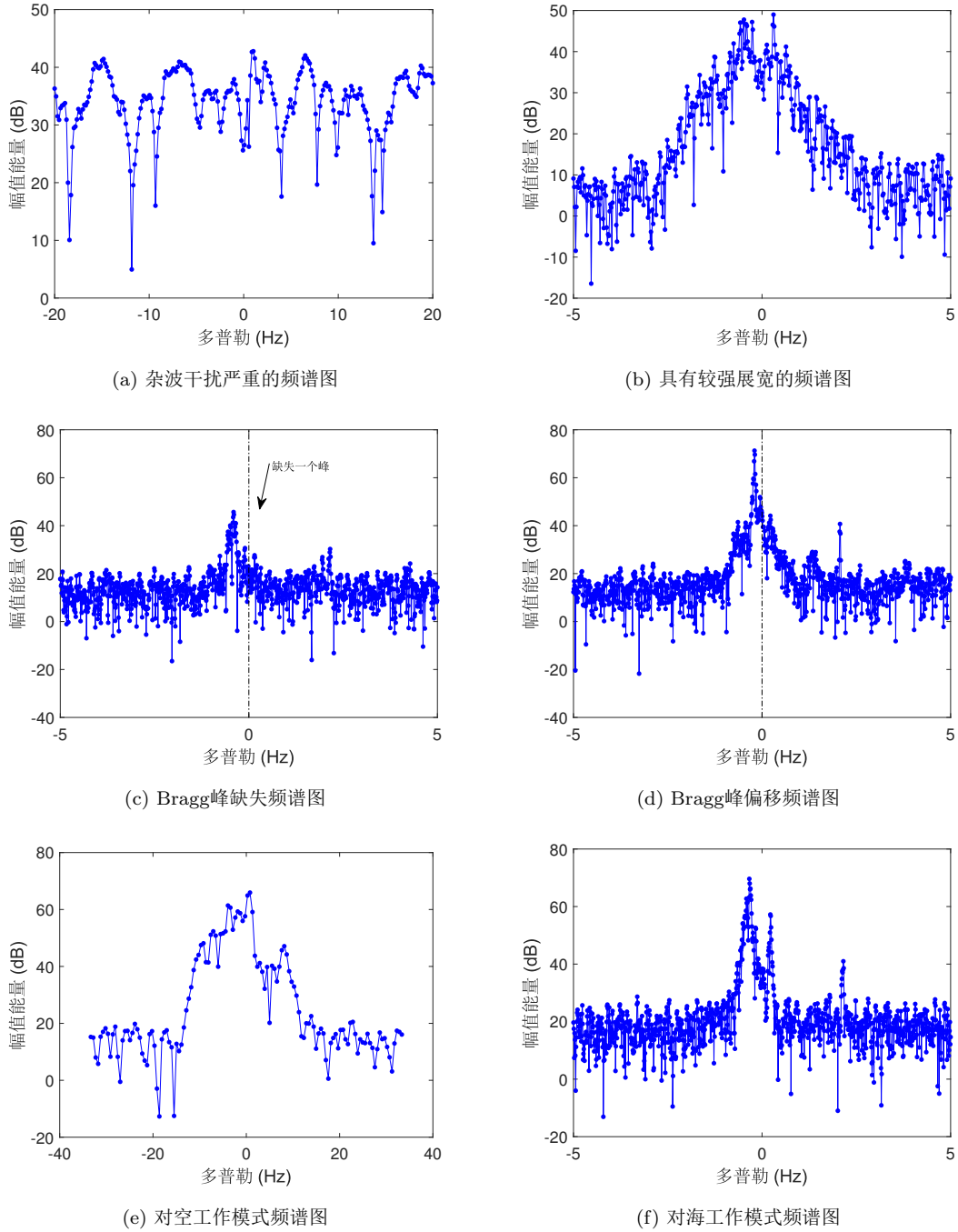


图 2-2: 复杂环境下频谱图

### 2.2.1 数据集分组

在本章的问题中，当雷达配置发生变化时，我们会获得不同的频率范围和精度频谱数据。例如，一些频谱数据的频率变化范围为-5Hz到5Hz、具有512个相干积累点，而另一些数据的频率变化范围为-10Hz到10Hz、相干积累点数为256个。因此，基于这

两个条件，我们将所有数据分为4组：

- 组A: 如图2-3a所示，具有128个相干积累点数，频率变化范围为-5Hz到5Hz;
- 组B: 如图2-3b所示，具有256个相干积累点数，频率变化范围为-5Hz到5Hz;
- 组C: 如图2-3c所示，具有512个相干积累点数，频率变化范围为-5Hz到5Hz;
- 组D: 如图2-3d所示，具有1024个相干积累点数，频率变化范围为-5Hz到5Hz;

我们只选择了上述四个具有典型意义的分组的数据来进行验证，舍弃了其余类型的与他们相似的数据，例如频率变化范围为-10Hz到10Hz的具有512个相干积累点的数据，这与组B的数据基本相同。

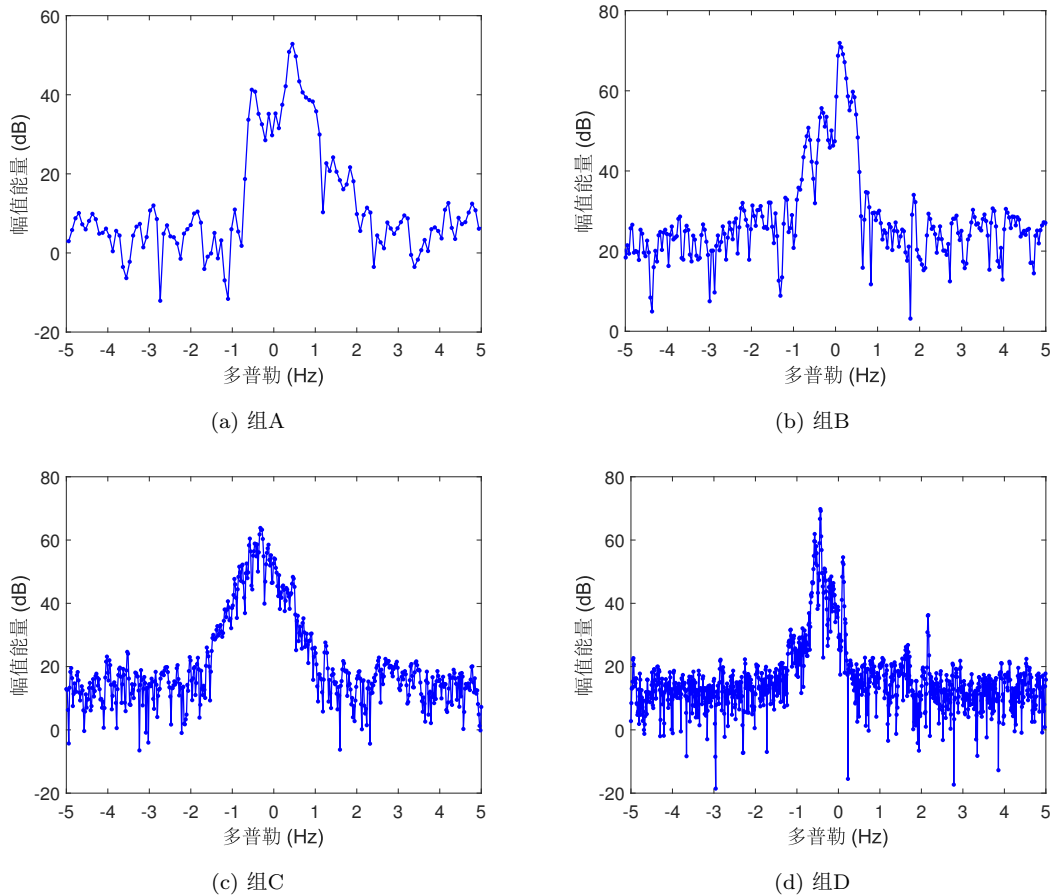


图 2-3: 不同组数据的频谱对比示意图

## 2.3 地海杂波识别算法

天波超视距雷达目标的坐标配准问题在很大程度上影响了其定位精度，特别是对于电离层参数无法准确及时获得的较远。识别地海分界线主要有两个优点：第一个是

我们可以使用获取的杂波地形图与实际的地图进行匹配，然后根据匹配结果计算偏移量，得到坐标修正系数，用来提高目标定位精度；另一方面，我们可以利用由识别结果获得的频谱上的偏移来校正频谱本身，以提高目标检测概率和准确度。

天波超视距雷达地海杂波识别技术的处理流程可分为信息预处理、地海杂波识别两个处理层，如图2-4所示。具体过程为：对频谱数据进行清洗、裁剪、融合等预处理操作，将预处理过的整体频谱数据输入到已经训练好的深度卷积神经网络分类器中进行识别。

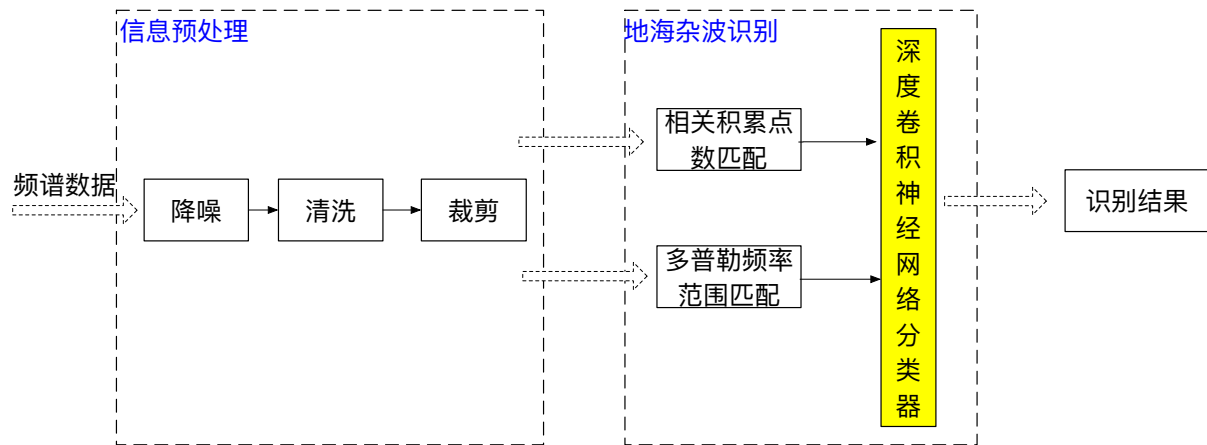


图 2-4: 系统结构图

### 2.3.1 数据预处理

传统的分类问题一般利用各种不同的由原始数据进行变换得到的特征进行分类，本章利用原始的杂波频谱数据来做地海杂波的识别。同时针对于地海杂波识别问题的数据形式，本章对输入数据做了二次处理。本章并没有选择某距离方位角单元的完整杂波频谱数据作为输入特征，而是考虑到用来区分地海杂波属性的特征主要集中于零频附近，因此对原始频谱数据做了剪切处理，只选择了零频附近一个区域的数据。通过减少大量无用数据，在一定程度上减少了计算量而且有助于防止过拟合现象的出现。

另一方面，我们最初从雷达得到的数据为对时域数据进行快速傅立叶变换后获得的频域中的数据，我们对这些数据进行平移，将快速傅立叶变换的直流分量移到频谱中心。虽然，表面上看利用平移前后的数据进行训练和测试区别不大。然而，在实际的情况中，在经过平移变换后，频域数据的特征更加集中，这样在执行卷积运算时学习得到的特征也更加准确。

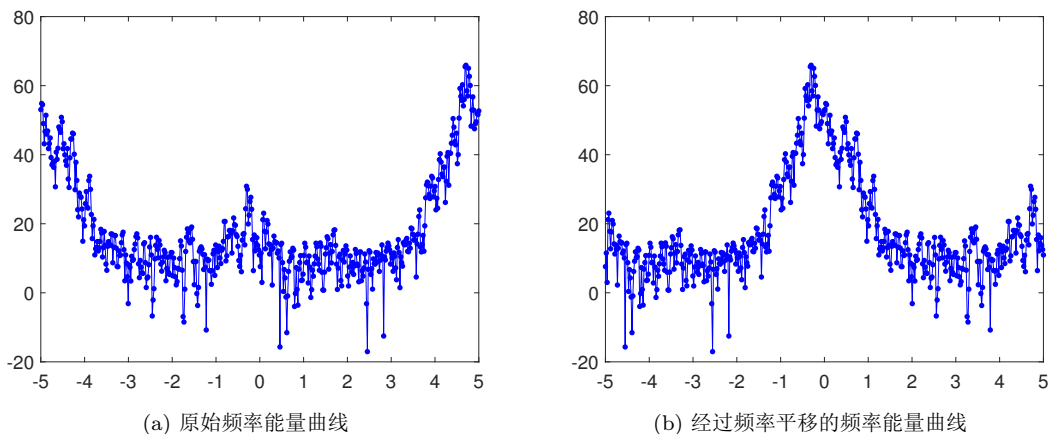


图 2-5: 频谱数据平移变换示意图

频谱数据预处理频谱数据是从天波超视距雷达获取的多普勒频率与幅度值对应的数据。在利用数据之前，我们首先对该数据进行清洗，把图2-6这种并非处于正常探测模式下的数据进行去除。

由于数据来自不同波位、不同时刻，具有不同的雷达工作频率，我们在利用这些数据数据进行训练或者识别前需要首先对其进行基本的处理。主要包括将数据按照积累点数、波位和多普勒频率范围进行分类，不同类别的数据分开处理。另一方面，由于地海杂波特征主要集中于多普勒频率较低的区域，我们可以将数据进行裁剪，只选取有效数据(本章中在权衡信息保留以及计算量的清洗下，保留了处于零频附近，且频率范围为 $[-5\text{Hz}, +5\text{Hz}]$ 的区域)，在一定程度上减小计算量。

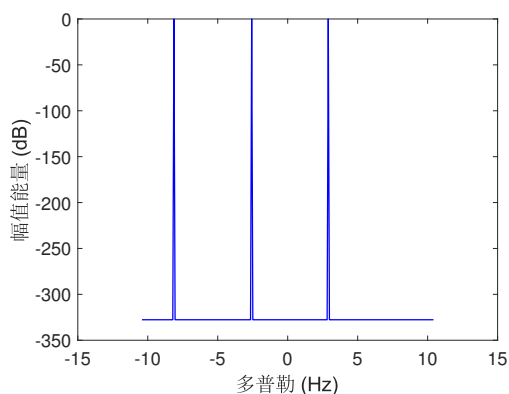


图 2-6: 需要被清洗掉的数据

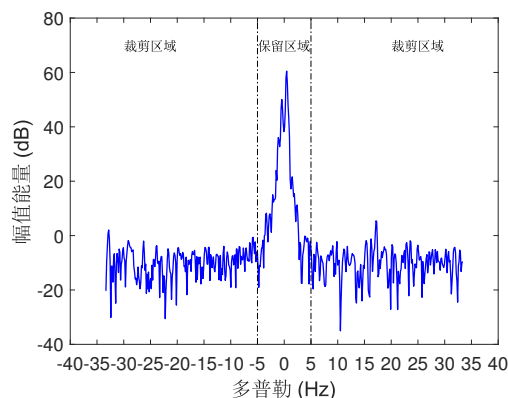


图 2-7: 数据裁剪示意图

受电离层非平稳、时变等特性影响，天波超视距雷达杂波数据可能会出现较大波动。对这种波动不加处理会导致地海杂波识别结果不准确。在一个相对短时间内，电离层会保持一个较平稳的状态，也即同一距离方位单元的真实的地海属性不会发生变

化。因此，本章采用滑窗融合的方法对输入数据进行预处理。其基本思想是将连续窗长时间 $N$  内的相邻杂波数据 $x_1, x_2, \dots, x_N$  进行加权融合得到新的频谱数据作为深度卷积神经网络分类器的输入

$$x = \sum_{i=1}^N w_i x_i \quad (2-1)$$

其中 $w_i$  为频谱数据 $x_i$  的权重，关于窗长及权重的选择会在后面仿真验证小节进行详细的讨论。从图2-8可以看出，融合后的毛刺明显变少，更利于最终的分类识别。

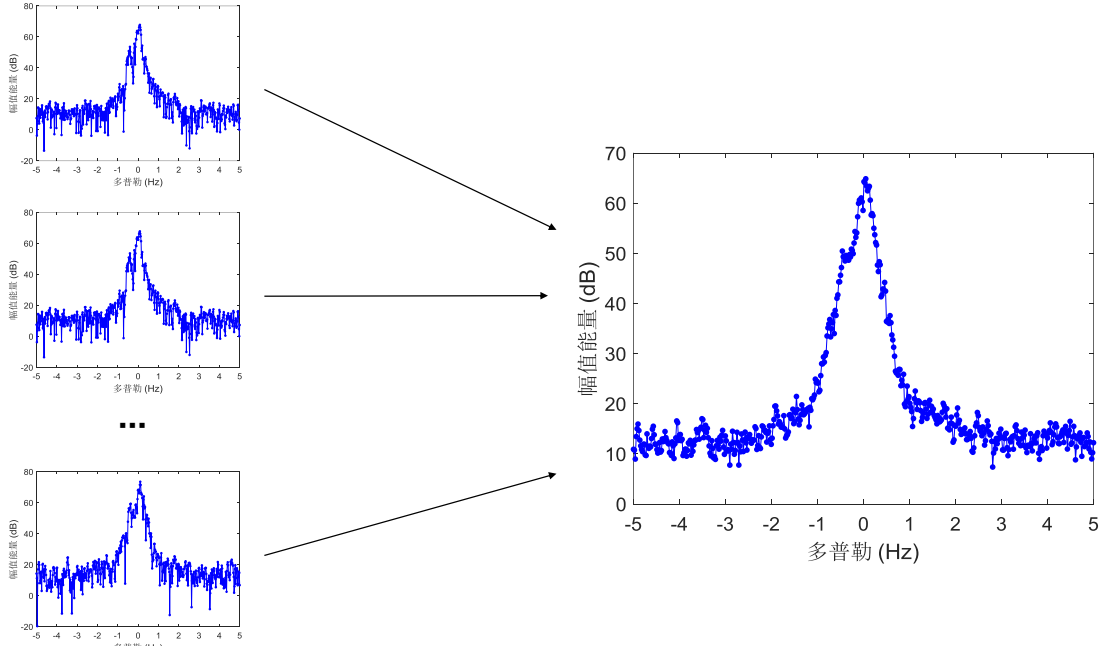


图 2-8: 数据融合示意图

### 2.3.2 分类算法设计

深度学习体系结构中有几大网络模型，其中的卷积神经网络可以直接将整个需要分类的数据作为网络的输入，避免了传统识别算法中复杂的特征提取和数据重建过程。基于此优点，使卷积神经网络在本章所需解决的天波超视距雷达地海杂波特征识别问题中有着巨大优势。典型的卷积神经网络由深层结构堆叠在一起的多个不同的层组成：输入层，多组卷积和池化层，有限数量的完全连接的隐藏层，以及输出层。其中最主要的部分为卷积层，其利用输入数据中的局部结构，将整个输入空间划分成很小的隐藏单元。将各个隐藏单元的权重构建得到的卷积核作用于整个输入空间，从而得到特征向量。利用这种机制，我们不仅大大减少了参数数量同时提高了数据的平移不变性。



天波超视距雷达地海杂波识别技术利用深度学习中的深度卷积神经网络算法，避免了对于地海杂波的建模，也即从根本上避免了传统方法所面对的困难。如图2-9所示，其主要可分为训练和识别两个步骤：利用大量已打好标签的样本通过深度卷积神经网络进行训练；然后对于新得到的雷达频谱数据利用模型进行识别，获得当前频谱数据的识别结果。

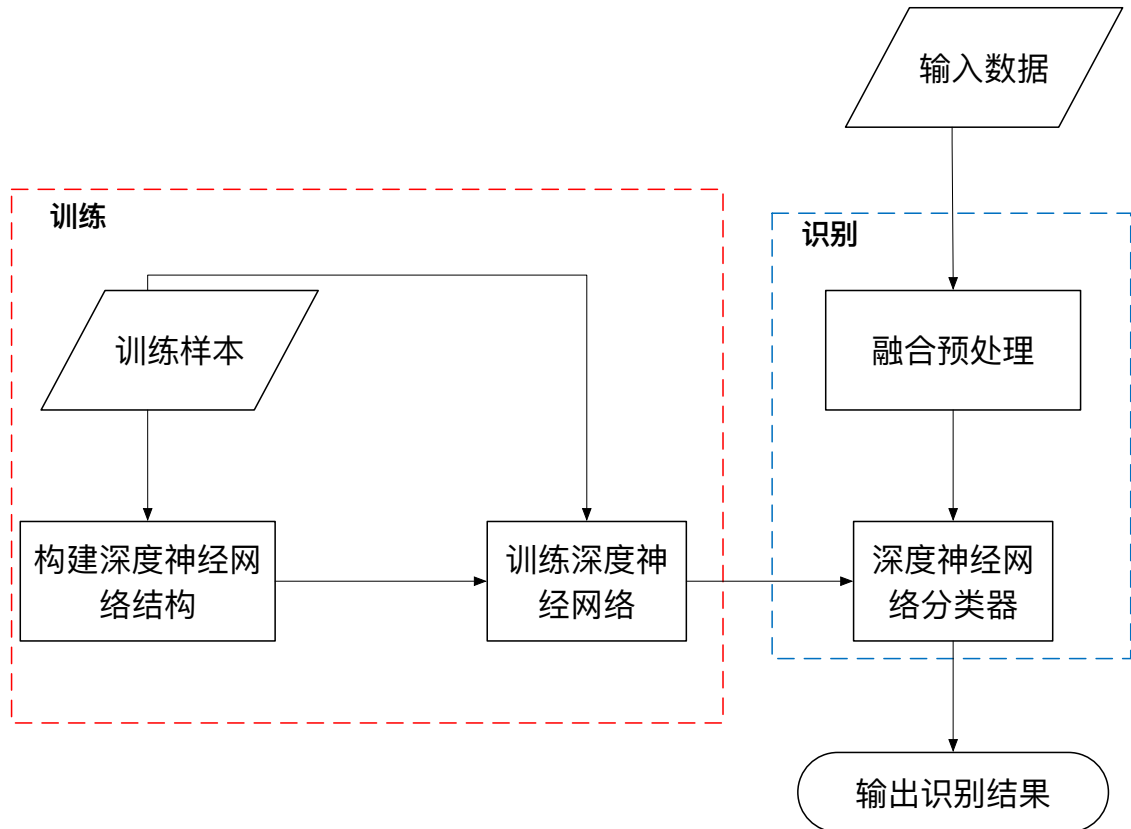


图 2-9: 地海杂波识别技术结构图

利用深度卷积神经网络进行天波超视距雷达地海杂波识别过程的主要挑战与难点在于网络模型的设计。如图2-10所示，本章设计的深度卷积神经网络的结构在功能上可以分为特征提取和全连接网络这两部分，特征提取层主要通过卷积操作和池化操作从输入的频谱数据中学习出最好的卷积核以及这些卷积核的组合方式，同时每一层的输出又作为下一层的输入，每层具有多个特征向量，每个特征向量具有多个神经元，并且每个特征向量来自于一种卷积核所提取输入的一种特征；全连接网络，主要是将任何一个神经元均和上一层的任何神经元之间建立管理，通过矩阵运算得到输出结果。



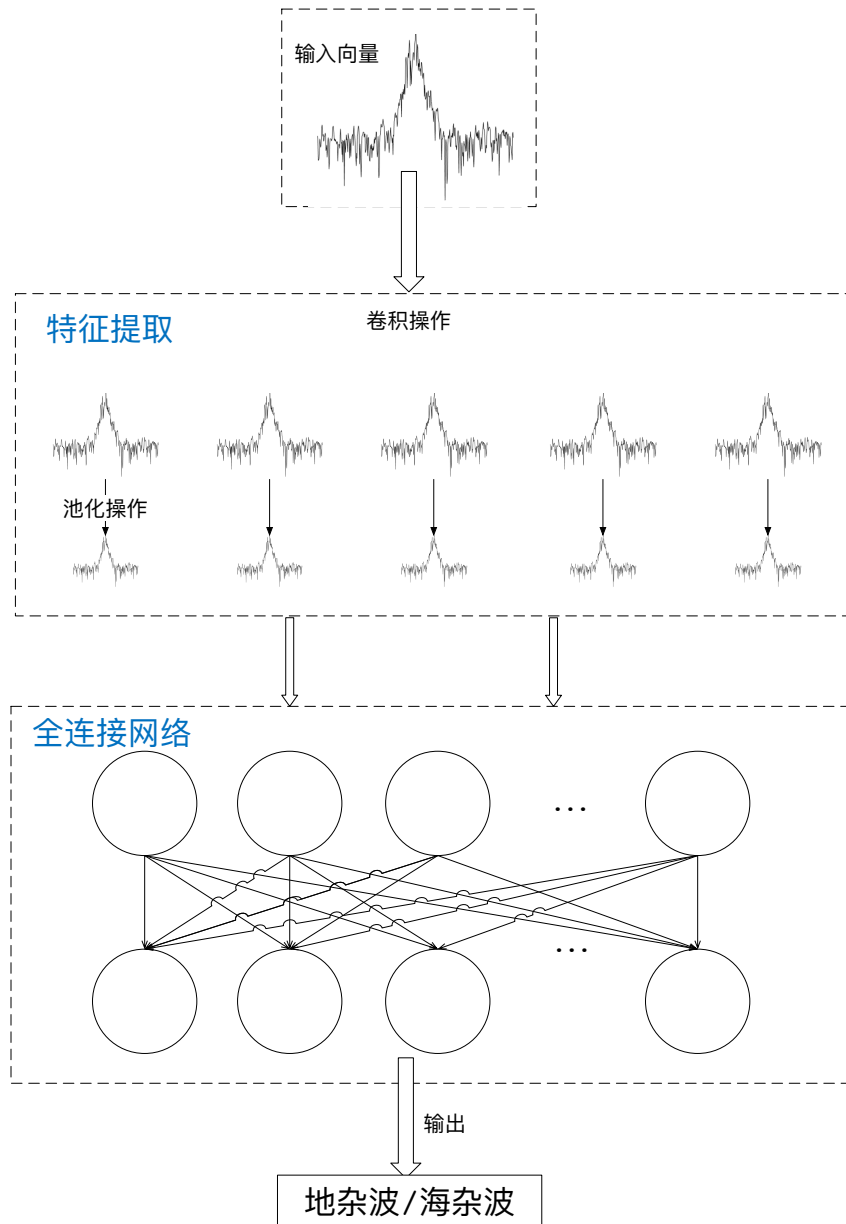


图 2-10: 特征提取和全连接网络

我们的地海杂波识别问题主要是基于频谱数据的特征来识别。人工识别主要基于海杂波中存在关于零频对称的Bragg峰，而地杂波只存在零频率附近的一个峰值。然而，还有一些无法直观的描述的特征，如整体幅度等等。此外，在一些频谱数据中仍然有一些无用的特征，例如，出现一个目标，这部分可以通过卷积特征提取和权重共享容易地去除对最终识别的影响。因此，本章基于CNN的方法很适合我们的问题。

通过利用CNN进行地海杂波识别，避免了对天波雷达回波的建模，从根本上避免了传统方法所面临的困难。本章根据地海杂波频谱的实际数据以及其反映出来的特性，构建了一个具有六层的基本卷积神经网络，如图2-11所示，每层具有多个特征向量，每

个特征向量具有多个神经元，并且每个特征向量从提取输入的卷积滤波器的特征导出。构建神经网络结构的其基本步骤为：

步骤1:输入地海杂波频谱数据（此处以大小为 $1 \times N$ 的输入序列为例），对其进行卷积运算，得到第一个卷积层。本章经过不同参数的试验对比结果，最终确定使用32个大小为 $1 \times 3$ 的卷积核，故特征向量中每个神经元与输入中的 $1 \times 3$ 的邻域相连，这样此卷积层中的特征大小就为 $1 \times (N - 3)$ 。又因为该卷积层有128个可训练参数(每个滤波器具有3个单元参数和一个偏置参数，一共32个滤波器，共 $(1 \times 3 + 1) = 128$ 个参数)，共 $128 \times (N - 3)$ 个连接，将连接通过ReLU激活函数。

步骤2:对卷积层进行最大池化处理，该操作将相邻的多个特征采用一个特征进行代替。通过降低特征向量的长度，在减小了计算量的同时也在一定程度上修正了过拟合情形。

步骤3:将经过上述两个步骤获得的特征向量作为新的输入，重复三次步骤1至2，可以得到一个三阶段的深度卷积神经网络结构。通过上述多阶段卷积操作，输入向量的特征获得了充分的提取。

步骤4: 构建输出层。压平步骤3获得的特征向量，把多维的输入一维化，以此作为卷积层到全连接层的一个过渡。在第一个全连接层的基础上添加dropout 参数，然后添加第二层全连接并通过激活函数Sigmoid，输出识别结果。

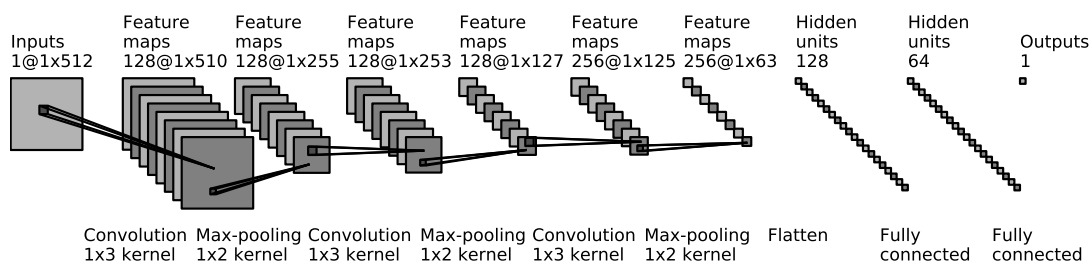


图 2-11: 卷积神经网络结构

## 训练算法

深度神经网络训练过程在搭建好合理的深度神经网络结构之后，下一步需要利用大量的数据对该网络进行训练。对于本章数据的训练的基本流程，对于地海杂波识别问题由于需要对不同相干积累点数和多普勒频率范围的数据进行分开训练，故首先需要对不同的数据进行分类处理并标注其地海杂波类型，通过此步骤完成训练样本的生

成。接下来就是利用训练样本对搭建好的网络结构进行训练，获得最终的分类器。训练过程或者说学习过程主要是利用了梯度下降算法，梯度反映了参数的移动方向。这其中很重要的问题就是学习率的选择，学习率过小则运行缓慢，过大则无法得到很好的结果。

传统的神经网络优化方法是mini-batch梯度下降。这个想法是计算每次迭代的mini-batch梯度，然后更新参数。然而，这种方法有两个缺点，一个是学习率的选择比较困难，因为它对所有参数使用相同的学习率，故很难选择一个很合适的初值，另一个是趋向于收敛到局部最优。

本章不仅考虑梯度，同时考虑包含梯度变化的信息，采用了一种具有自适应学习率的优化算法Adam(Adaptive Moment Estimation)。该算法利用梯度的一阶矩估计和二阶矩估计动态地调整每个参数的学习率。经过偏置校正后，每一次迭代学习率都有明确范围，使得参数更加平稳。传统的梯度下降更新规则为 $\theta \rightarrow \theta' = \theta - \eta \nabla C$ ，这里 $\theta$ 表示需要学习的参数， $\eta$ 为学习速率， $\nabla C$ 为损失函数的梯度，则变为：

$$v \leftarrow v' = \mu v - \eta \nabla C \quad (2-2)$$

$$\theta \leftarrow \theta' = \theta + v' \quad (2-3)$$

其中， $\mu$ 是用来控制梯度变化的超参数。训练神经网络模型。在搭建好神经网络模型后，利用训练样本对该模型做进一步训练，其基本的算法流程如下：

**算法 1 Adam 训练算法**

**输入:** 训练样本 $X$ , 样本标签 $Y$ , 最大迭代次数 $K$ , 步长 $\varepsilon$ , 初始目标参数 $\theta$

**输出:** 学习后的参数 $\theta$

- 1: 初始化矩估计的期望衰变率 $\rho_1 = 0.9, \rho_2 = 0.999$ , 一阶矩与二阶矩变量 $s = 0, r = 0$ , 迭代次数 $k = 0$
- 2: **while**  $k < K$  **do**
- 3:   从训练集中获取对应于目标 $y^{(i)}$  的 $m$  个采样样本 $\{x^{(1)}, \dots, x^{(m)}\}$ , 其中 $m$  为批处理中一批样本的个数。
- 4:   计算梯度 $g \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(x^{(i)}; \theta), y^{(i)})$ , 其中 $L$ 为对数损失函数, 其定义为 $L(P(Y|X), Y) = -\log P(Y|X)$ 。
- 5:   更新迭代次数 $k \leftarrow k + 1$
- 6:   更新有偏一阶矩估计 $s \leftarrow \rho_1 s + (1 - \rho_1)g$ , 有偏二阶矩估计 $r \leftarrow \rho_2 r + (1 - \rho_2)g \odot g$ , 其中 $g \odot g$ 表示对应元素的乘积。
- 7:   修正一阶矩估计 $\hat{s} \leftarrow \frac{s}{1 - \rho_1^k}$ , 二阶矩估计 $\hat{r} \leftarrow \frac{r}{1 - \rho_2^k}$ 。
- 8:   更新参数 $\theta \leftarrow \theta + \Delta\theta = \theta - \varepsilon \frac{\hat{s}}{\sqrt{\hat{r} + \delta}}$ , 其中 $\delta = 10^{-8}$  用来保持稳定性。
- 9: **end while**

由于本章是一个二分类的问题, 我们利用逻辑损失函数来训练本章的模型。

$$L(z_i) = \log(1 + e^{-z_i}) \quad (2-4)$$

其中,  $z_i = y_i(w^T x_i + b)$ ,  $y_i$  是采样样本 $x_i$ 的标签, 其为1代表地杂波, 0为海杂波。

### 2.3.3 分类阈值设计

由于我们的问题是一个二分类问题, 所以我们得到的是回波频谱数据是来自于海洋还是陆地的概率。对于一般的二分类问题, 我们可以使用0.5作为概率阈值来进行分类。但是, 在我们的问题上, 结合实际的情况, 我们需要考虑下面两个方面:

- 海洋的变化远远大于陆地, 海洋容易被误判为陆地。
- 海洋或者陆地均应该是连续的。

因此, 我们不能直接输出结果。我们需要找到合适的阈值来划分地海杂波, 这将在第2.4.2节通过仿真实验进行讨论。此外, 我们还需要根据该距离方位单元周围单元的识别结果进行进一步处理。也就是说, 如果我们得到初步结果 $y_{i,j}$ ,  $i$ 表示方位角的第 $i$

个单元,  $j$ 表示第 $j$ 个径向距单元。那么, 最终结果 $y_{i,j}$ 可以表示为:

$$y_{i,j} = \sum_{m=1}^M \sum_{n=1}^N w_{m,n} \cdot y_{m,n} \quad (2-5)$$

$w_{m,n}$ 是 $(m,n)$ 单元处识别结果 $y_{m,n}$ 的权重,  $M$ 为利用的方位上的区域单元个数,  $N$ 为利用的径向距上的区域单元个数。

## 2.4 仿真验证

在本节中, 我们利用实际数据评估了我们提出的基于深度学习的地海杂波识别算法的性能。将我们的基于CNN的算法与基于LMS[15]的根据多普勒频谱进行分类的算法和基于支持向量机进行分类的算法[16]进行了对比, 我们算法相对于其余两种算法在性能上的提高, 证明了我们算法的可用性。

### 2.4.1 算法实现

#### 基于LMS的算法

该方法主要基于文献[15], 其基本思想是利用加权最小均方算法来求解区域的模糊度, 其中区域模糊度的定义如下:

$$f_D = f_I + f_{cur} + f_B \begin{cases} 1/2 & \text{前向海} \\ 0 & \text{陆地} \\ -1/2 & \text{后向海} \end{cases} \quad (2-6)$$

其中,  $f_B$  是前向Bragg回波与后向之差,  $f_I$  和  $f_{cur}$  分别是电离层动力过程和海流产生的多普勒频偏。

#### 基于SVM的方法

这里我们参照文献[16], 选择了如下三个特征作为分类器的特征向量:

- 最大后向散射幅值
- 频谱中最大与次大幅值频率之差
- 频谱中最大与次大幅值幅度之差

同时, 我们利用了网格法对参数进行了调优。

为了确保我们有足够的数据来训练和测试我们的算法，我们选取了不同的雷达工作条件、天气、时间的多组数据（每组约有20000个频谱数据）。我们随机选择其中70%的数据作为训练数据，20%作为交叉验证数据，其他数据用作测试数据。

## 我们的算法

我们根据图2-11 所示的结构搭建了本章的深度卷积神经网络分类器。对于前两个卷积层具有32个滤波器，对于第三个卷积层有64个滤波器。全连接层添加到dropout参数为0.5。利用Adam优化算法来训练网络，初始的学习率为0.001，基本迭代次数为20次。我们使用相同的样本来分别训练和测试基于支持向量机的分类器和我们的算法。

## 评估方法

对于一个分类问题，最基本的算法性能评估方法是分类的准确度。然而，由于我们这里利用实际数据进行验证，这些数据没有准确的标签，尤其是地海交界处的杂波的类别更加难以确定。而在实际工程实践中，该部分的识别准确度影响着最终电离层参数的辨识。由于在实际的地图中可以得到地海交界处也即海岸线处的地理位置坐标，也即可以得到该区域的形状。因此，我们设计了与地图的匹配程度的评估方法来分析不同识别算法对于海岸线部分杂波识别的性能，其基本定义如下：

$$g_{C_1, C_2} = \frac{area(C_1 \cap C_2)}{\max(area(C_1), area(C_2))} \quad (2-7)$$

我们用0和1来二值化我们的区域，如图2-12所示。 $area(C)$ 表示区域 $C$ 的面积， $g_{C_1 \cap C_2}$ 表示区域 $C_1$ 和 $C_2$ 的匹配度。



图 2-12: 二值化地图

### 2.4.2 仿真结果分析

我们首先利用我们的四组数据对于三种算法进行了对比分析。实验结果如表2-1所示，可以明显的看到，我们的算法的识别正确率为最优。

表 2-1: 三种算法的识别精度对比

数据集	CNN	SVM	LMS
组A	96.97%	85.69%	81.85%
组B	97.14%	88.57%	81.93%
组C	99.04%	89.03%	88.57%
组D	99.69%	91.61%	90.33%

我们选取了CNN部分识别错误的结果进行分析，如图2-13 所示，发现该部分的数据属于特征十分不明显的内容，即使是人工判定也无法很准确的判定的结果。

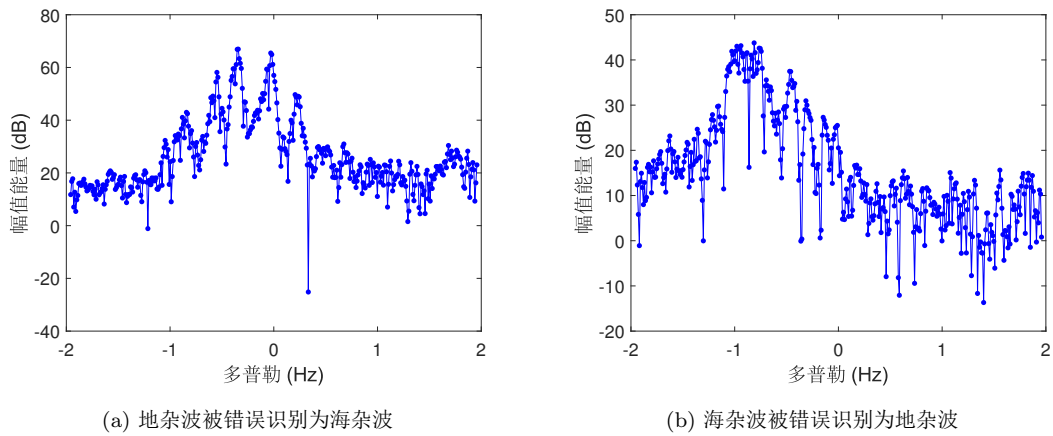


图 2-13: 识别错误的结果

由于数据集自身的一些限制条件，我们选择在组D所在的数据集进行了地图匹配度的比较实验。从表2-2可以看出，我们的算法在匹配度方面要远高于其余两种方法，证明了我们的算法在地海边界处仍然可以保持很高的精度。

表 2-2: 匹配度对比

	CNN	SVM	LMS
匹配度	0.92	0.23	0.21

为了验证我们的深度卷积网络在迭代次数方面的影响，其不同组数据的损失函数如图2-14所示，结果表明我们的算法可以在不同的数据集组中均可以较快的收敛。虽然，对于相同的神经网络结构，第一个数据集需要最多的迭代次数才能收敛，这是因为当频率和相干累计点的比例变小时信息或者说特征也随之减少，故需要较多的迭代次数。

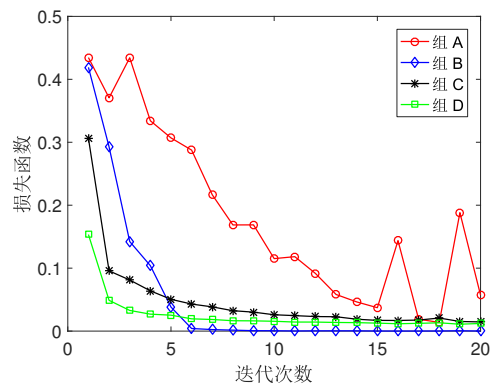


图 2-14: 不同数据集损失函数对比图

图2-15展示的是在样本集大小不同的情况下，我们的算法与基准算法的平均分类准确率的对比图。由于基准方法仅使用根据先验知识得到的阈值，因此随数据集增长其识别准确度变化不大，而我们的深度卷积神经网络的算法随着数据集内样本数量的增加，准确度有着显著提升。



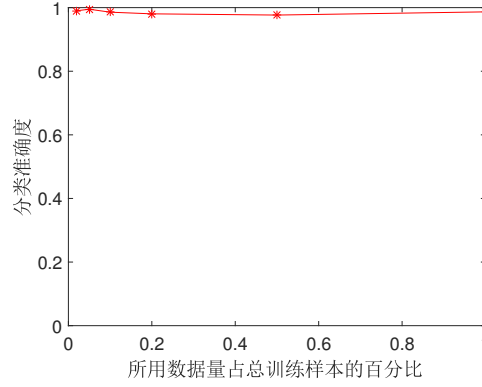


图 2-15: 不同数量数据集的分类准确度对比曲线图

众所周知，卷积神经网络的参数对于最终分类识别的准确率起着重要的作用。因此，我们需要对参数的选择进行一些分析。首先，我们分析在批大小（Batch Size）和迭代次数变化时，验证集数据的分类准确度。如图2-16所示，我们可以发现识别准确度随着迭代次数的增加而增长。而当批长度变大时，收敛速度加快。

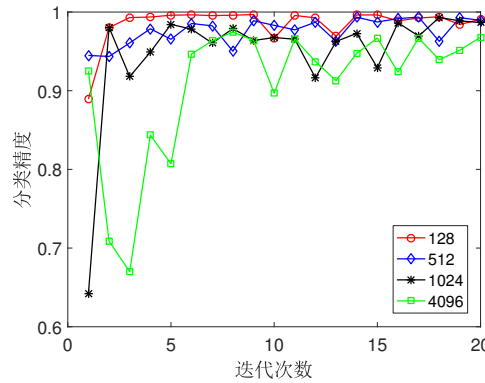


图 2-16: 不同批长度下，分类准确度与迭代次数曲线图

由于在预处理阶段，本章利用了滑窗算法对输入数据进行融合处理来减少由于某一帧数据中某距离方位单元由于出现的随机噪声对于我们识别结果的影响，此处对窗长参数进行设计比较。本章主要考虑到电离层会随时间发生变化且天波雷达的采样周期较长，过长的窗长对无法及时的响应电离层的变化，影响识别准确率以及与地图的匹配度。为了取得一个合适的窗长，我们首先利用不同窗长平均融合后的数据进行测试，得到图2-17 的结果，该结果证明了在窗长过长时候，匹配度会下降的结论，当窗长过大时匹配度会降到比窗长为1时还要低。为了进一步比较权重的变化对于识别结果的影响，我们选取公式2-1中的权重为 $w_N = \frac{1}{2}, w_k = \frac{1}{2(N-1)}, k = 1, 2, \dots, N-1$ ，得到如图2-18所示的结果。由于增加了主单元的权重，故其可以在窗长增加时，不会有过于明

显的匹配度下降。

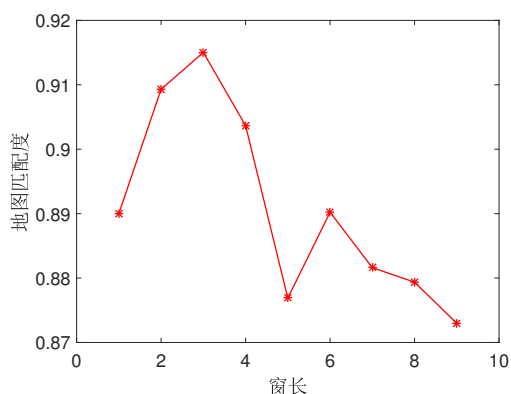


图 2-17: 匹配度与融合窗长对比曲线图

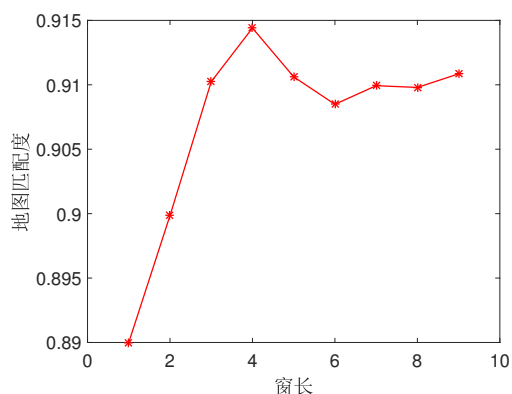


图 2-18: 更改权重后不同窗长匹配度结果对比曲线图

为了找出区分识别结果中海洋与陆地的最佳概率阈值，我们使用不同的概率阈值计算相同测试数据的正确率。图2-19显示，当阈值增大时，分类识别精度也随之提高，但精度的提高速度越来越慢，直至平稳。另一方面，当阈值仅为0.01时，识别率仍高于0.86。这说明我们的方法的分类结果的概率值均处于较高的水平，如2-20所示。

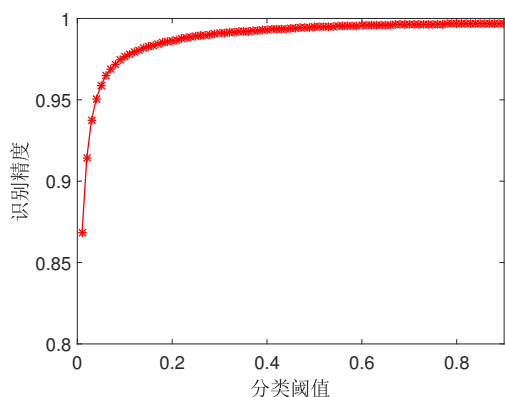


图 2-19: 识别率与概率阈值曲线图

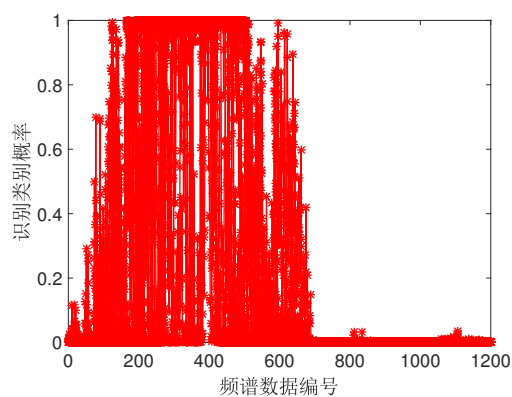


图 2-20: 不同帧数据识别结果概率值

为了验证本章提出的深度卷积神经网络地海杂波识别算法是否具有实时识别的能力，我们在CPU为i5-3.30GHz、运存为1GB的Linux虚拟机运行本章的算法程序，对整帧的实际雷达数据（共20400个分辨单元，每个分辨单元的相干积累点数为1024），其识别过程中占用内存峰值为178.5MB，识别时间为0.494141秒，其具备实时运行的能力。

经过上述实验在多个不同数据集上，对我们的算法和基于支持向量机的识别算法和基于LMS的算法的对比分析，可以得到下述结论：

- 我们基于卷积神经网络的方法中获得最好的结果，且在地图匹配结果方面显著优于其余两种方法。
- 我们的方法具有很强的鲁棒性，雷达参数或者自然变化对识别结果影响不大。

### 2.4.3 特征可视化

上面利用大量的测试数据的识别结果以及地图匹配结果对于我们基于卷积神经网络的算法进行了验证，但是卷积神经网络方法有一个问题是其为一个黑盒操作，我们无法直观地看到其用于分类的特征。因此，为了在理论上对于我们算法的有效性进行分析。在本节中，我们使用基于梯度变化的可视化方法，其思想为通过计算当输入数据的某个数据点发生变化时输出梯度的变化，得到每一个数据点对于输出梯度的影响，从而得到该频谱数据热力图。我们定义频谱数据序列为 $S = \{s_1, s_2, \dots, s_n\}$ ，其中 $n$ 是频谱序列中的点数，设输出概率为 $p(S)$ 。那么，我们可以得到式2-8：

$$p(S) = w^T S + b \quad (2-8)$$

其中 $w$ 和 $b$ 分别是我们的模型的权重和偏差。实际上，这里的权重 $w$ 表示对应点的重要性。在我们的模型中，类概率函数 $p(S)$ 是高度非线性函数，这里使用泰勒方法近似 $p(S)$ 。为了简化计算，我们使用一阶泰勒展开：

$$w = \frac{\partial p}{\partial S} \Big|_{s_i} \quad (2-9)$$

因此，我们以通过反向传播计算得到 $w$ 的方程式2-9。图2-21显示，特征点主要集中在我们预期的数据上。

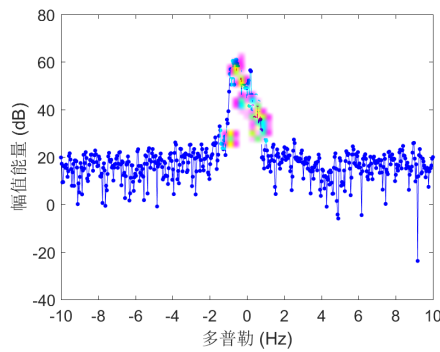


图 2-21: 特征重要程度热力图

## 2.5 小结

在本章中，我们提出了基于卷积神经网络天波雷达地海杂波识别的新算法。其主

要克服了传统的阈值识别方法或支持向量机算法根据经验从频谱数据中提取特征，导致操作复杂度高，分类精度低的缺点。同时，我们将我们的算法与传统算法和支持向量机算法进行了对比。实验结果表明，我们的方法在地海杂波识别问题上更加有效以及抗干扰性能更强，发现利用不同时间的同一区域的频谱数据进行融合可以很大程度上提高分类精度。在更高精度的识别结果的帮助下，我们可以得到更加精确的修正系数，可以为目标定位问题提供非常大的帮助。另一方面，如果我们针对于特定的问题对卷积神经网络的参数进行调整，可以进一步提高我们算法的性能。

### 3 基于深度学习的辐射源识别

#### 3.1 引言

辐射源识别算法需要可以准确区分出已知目标和未知目标，同时可以正确的对于已知目标进行分类。我们需要在未收集大量数据的前提下，可以迅速的识别出新的目标。与传统的利用已知类别的样本进行训练测试的机器学习算法不同，辐射源识别的问题是在Open Set的背景下，需要考虑输入未知分类样本的情况。由于复杂电磁环境下辐射源个体识别所面临的识别能力差等问题与挑战，传统辐射源识别方法具有很大的局限性，我们需要寻找一种新的方法解决该问题。

本章综合雷达信号处理、深度学习等多学科理论，详细分析不同辐射源雷达信号的差别，研究不同雷达的信号建模过程以及据此获取雷达信号的基本特征；综合考虑各种脉内细微特征，利用深度卷积神经网络与支持向量机进行结合，以雷达信号的模糊函数切片作为训练样本的特征向量，构建了一个可以对未知分类进行辨识的分类器。最后，利用实际数据进行验证，证明我们的分类器具有很高的准确性。

本章安排如下：4.2节对辐射源信号进行了分析，并对其进行预处理，求取其模糊函数切片，4.3节构建了本章的Open Set 分类器，详细地阐述了深度卷积神经网络这个主分类器与支持向量机Meta分类器的设计过程，4.4节利用实际数据对于分类器已知分类识别和未知分类辨别的性能进行了验证，4.5节进行本章总结。

#### 3.2 辐射源信号分析

对于辐射源信号的分析处理，本章主要考虑两方面：信号预处理、特征提取优化。

本章所获得的信号为雷达辐射源的I/Q两路数据，这是一种在雷达信号处理领域常见的用来描述信号的方法。其中I表示In-Phase，即同相，Q表示Quadrature，即正交，与I相位之差为90度。设需要表示的信号的峰值幅度为 $A$ 、相位角为 $\phi$ ，则有：

$$I = A \cos \phi \quad (3-1)$$

$$Q = A \sin \phi \quad (3-2)$$

也即，可以利用公式3-3表示信号：

$$Ae^{i\phi} = A(\cos(\phi) + i\sin(\phi)) = I + Qi \quad (3-3)$$

从而可以根据公式3-1和公式3-2得到根据I、Q数据求取信号的峰值幅度和相位角的下式：

$$A = \sqrt{I^2 + Q^2} \quad (3-4)$$

$$\phi = \tan^{-1}(Q/I) \quad (3-5)$$

在完成信号形式的转换后，我们首先需要对信号进行初步的预处理，剔除无用和错误的数据。

在特征提取优化方面，合理的特征是分类识别的基础。由于存在相同型号的辐射源，利用简单的参数特征无法很好的完成辐射源的个体识别，但是在实际中辐射源自身存在相位噪声以及各类杂散输出，此部分特征可以用来区分出型号、参数均相同的辐射源，因此我们需要选取一种可以提取雷达辐射源这种无意调制产生的信号脉内细微特征的方法。模糊函数不仅能描述辐射源信号的分辨特性与模糊度，还能描述由雷达辐射源信号所决定的测量精度、杂波抑制特性等，通过模糊函数在时延和频偏这两个维度上的变换，可以多角度的刻画出无意调制对于发射信号的影响，因此我们最终选取了利用雷达模糊函数挖掘辐射源的特征。

### 3.2.1 模糊函数

对于信号 $x(t)$ ，其瞬时自相关函数为 $R_x(t, \tau) = x(t + \tau/2)x^*(t - \tau/2)$ ，其中 $\tau$ 为时延，模糊函数的定义为，

$$A(\tau, \nu) = \int_{-\infty}^{+\infty} R_x(t, \tau) e^{j2\pi\nu t} dt \quad (3-6)$$

即 $R_x(t, \tau)$ 关于时间 $t$ 的傅里叶反变换。

为了方便在数字信号中使用，式3-6可以经过变换等价于下面的形式：

$$A(\tau, \nu) = \int_0^\tau x(t)x^*(t + \tau)e^{j2\pi\nu t} dt \quad (3-7)$$

对信号均匀采样，即对接收信号和参考信号离散化后，式3-7可以表示为：

$$A(\tau_l, \nu_m) = A(l, m) = \sum_{n=0}^{N-1} x(n)x^*(n+l)e^{\frac{j2\pi mn}{N}} \quad (3-8)$$

其中， $\tau_l = l/f_s$ 、 $\nu_m = mf_s/N$ 。

我们此处以一个简单的单载频矩形脉冲信号来展示模糊函数特征提取的作用，图3-1为模糊函数图，可以发现模糊函数存在一定的冗余，其主要变化均处于0时偏

和0频偏附近。为了减小计算量，本章在频偏为0附近取不同时间延迟的切片作为信号特征，即可以有效地提取信号的相位噪声和杂散输出等个体特征，并且此处受噪声的干扰较小，更加稳定，图3-2即为在频偏为0处的单载频矩形脉冲信号模糊函数切片。

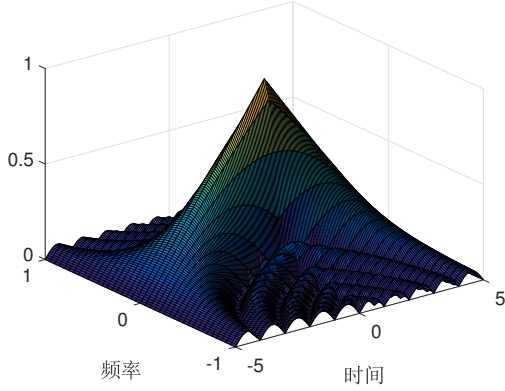


图 3-1: 单载频矩形脉冲信号模糊函数图

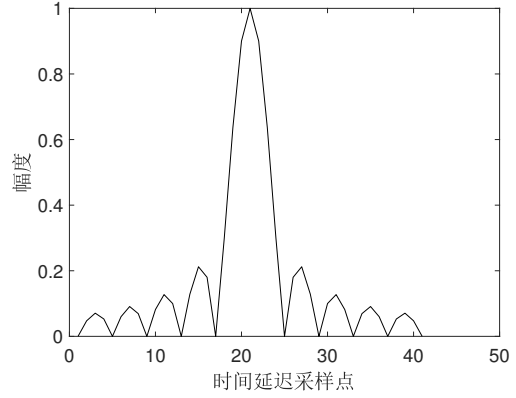


图 3-2: 单载频矩形脉冲信号模糊函数切片图

### 3.3 Open Set 分类器设计

通常的识别或者分类系统仅考虑的是一个闭集分类系统，然而在现实世界中，这种分类系统会遇到很大的问题。由于其最基本的假设是所有的类别均为先验已知，那么就会出现问題，例如在训练样本中不存在类别的样本就会被错误的分到某个类别中去。这种在训练的时候提供不完整的信息，而在测试的时候会添加未知分类的问题，称作Open Set 识别[17, 18]。这个问题还可以描述为需要在测试的时候拒绝未知样本。Open Set目标识别系统必须可以准确的处理下面三种类型的数据类：

- 已知的（目标）类，被标记为正训练样本的数据。
- 已知的未知（非目标）类，被标记为负训练样本的数据。
- 未知的未知（非目标）类，在训练样本中不存在的类别的数据。

传统的机器学习算法均为针对于闭集数据设计的，随着识别算法应用场景的增多和对精度要求的提高，许多学者开始了对Open Set识别的研究。Simonson[19]提出了一种称作*probabilistic fusion*(PF)的利用统计的方法来进行Open Set识别，其主要通过合并来自不同数据源的证据得到一个统计测试模型，根据此模型的分布来对于类别进行判断。Scheirer等人[20]提出了一种通过分析后验数据得分来进行类型判断的方法。

此部分主要解决的问题是当得到一个新的测试样本，如果该样本不属于已经经过训练的分类，那么传统的神经网络模型会将该样本指派给与其最相似的一个类别，此

种情况对于一个Open Set识别系统，也即类似于辐射源识别系统这种具有较多尚未经过训练的样本的一个数据集，首先这会导致其识别率下降，另一方面是由于对于未知的辐射源无法很好的确定，无法很好的完成预警等任务。目前，学者对于该问题的研究主要分为下面两个思路：

- 在训练集中添加一个“未知”类别，利用不同的来自非已知类别的数据作为训练样本对该类别进行训练，然后对于所有的输入数据进行类别的识别，对于识别结果为该类别的数据作为未知分类。
- 针对于多分类使用的softmax函数，可以设立阈值或者对于该识别结果进行一个评价（例如与已知类别数据的一个“距离”），通过这种方式分辨出未知分类。

第一个思路最大的问题是我们无法得到所有可能的未知类别的样本来进行训练，具有一定的局限性，不适用于我们这个具有大量来自未知分类数据的问题。针对于该问题，我们基于后一个思路设计了一个基于Meta-Recognition的可以识别未知辐射源的深度神经网络。首先是创建一个深度卷积神经网络分类器，该分类器的输出为该训练样本属于各个类别的概率，我们然后将此类别作为一个输入，输入到我们的Meta-Recognition中，这里我们设计一个支持向量机分类器作为Meta-Recognition，然后从该Meta-Recognition会进行判断该输入是否为一个未知分类。

### 3.3.1 深度卷积神经网络分类器设计

本章根据辐射源信号的实际数据以及其反映出来的特性，设计构建了一个具有10层的一维卷积神经网络。

该分类器作为主要分类器，且Meta-Recognition是以该分类器的输出作为输入，所以该分类器性能的好坏会直接影响到对已知类别的分类和对未知类别的判断。

第一层是输入层，由于模糊函数切片为一个 $1 \times 1000$ 的向量，因此输入层大小是 $1 \times 1000$ 。

第二层是卷积层 $C1$ ， $C1$ 对输入向量进行一维卷积运算提取特征，卷积运算可以最大程度的提取原始信号的特征。此处利用了256个大小为 $1 \times 3$ 的卷积滤波器，其窗口移动步长为1。

第三层是一个池化层 $S2$ ， $S2$ 层对上一层 $C1$ 做池化处理，池化的目的是在保留数据有用信息的同时，尽可能减少数据量。此处采用的是 $1 \times 2$ 的最大池化操作。

第四层是一个卷积层 $C3$ ， $C3$ 对 $S2$ 的特征进行卷积操作，此处利用了128个大小为 $1 \times 3$ 的卷积滤波器。



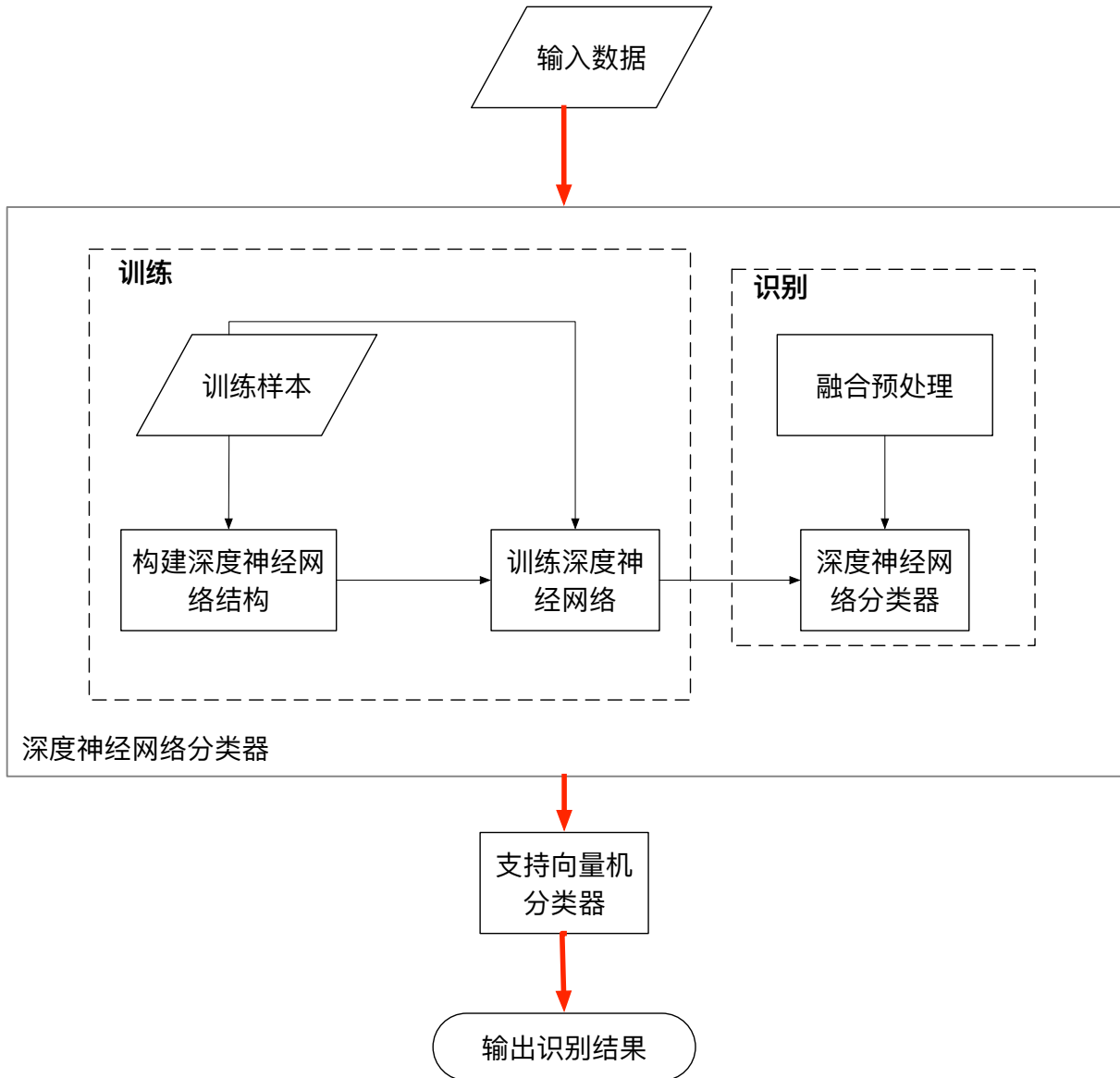


图 3-3: 分类器设计结构图

第五层是一个卷积层 $C4$ ，其结构与 $C3$ 相同，128个大小为 $1 \times 3$ 的卷积滤波器。

第六层是一个池化层 $S5$ ，其结构与 $S2$ 相同， $1 \times 2$ 的最大池化操作。

第七层是一个卷积层 $C6$ ，其结构与 $C3$ 相同，128个大小为 $1 \times 3$ 的卷积滤波器。

第八层是一个卷积层 $C7$ ，其结构与 $C3$ 相同，128个大小为 $1 \times 3$ 的卷积滤波器。

第九层是一个池化层 $S8$ ，其结构与 $S2$ 相同， $1 \times 2$ 的最大池化操作。

第十层是输出层，根据不同的类别个数选取相应的输出节点个数，首先将 $S8$ 的特征拉成一个一维向量，然后通过全连接网络与输出层进行连接，通过Softmax激活函数输出最终的结果。

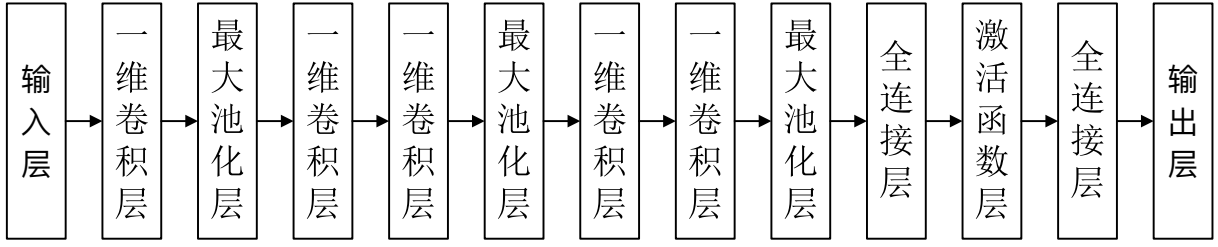


图 3-4: 深度卷积神经网络框架图

### 3.3.2 支持向量机Meta-Recognition 设计

#### 支持向量机原理

支持向量机是一种流行的分类方法，它可以在不需要大量数据的情况下产生良好的结果。对于一个二分类问题，设 $((x_1, y_1), \dots, (x_n, y_n))$ 为训练数据集，其中 $x_i$ 为某样本的特征向量， $y_i \in \{-1, +1\}$ 为该样本的标签。支持向量机的思想为找到一个超平面将这些样本划分为正类（标签为+1）和负类（标签为-1），并且使得正类和负类之间的距离最大。这个超平面的间隔被定义为正类与负类之间的最近距离。

对于一个线性分类问题，假设所有的数据满足下面的约束：

$$w \cdot x_i + b \geq +1 \quad y_i = +1 \quad (3-9)$$

$$w \cdot x_i + b \leq -1 \quad y_i = -1 \quad (3-10)$$

其中 $w$ 为超平面的法向量， $|b|/||w||$ 是从超平面到原点的垂直距离， $||w||$ 是向量 $w$ 的欧拉范数。将上述两个式子合并得到：

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i \quad (3-11)$$

公式3-11 中的训练样本构成了这个分类平面(图3-5 中的 $H_1$ 与 $H_2$ )。间隔 $\rho$  可以通过计算 $H_1$ 与 $H_2$ 的距离得到：

$$\rho = \frac{|1 - b|}{||w||} - \frac{|-1 - b|}{||w||} = \frac{2}{||w||} \quad (3-12)$$

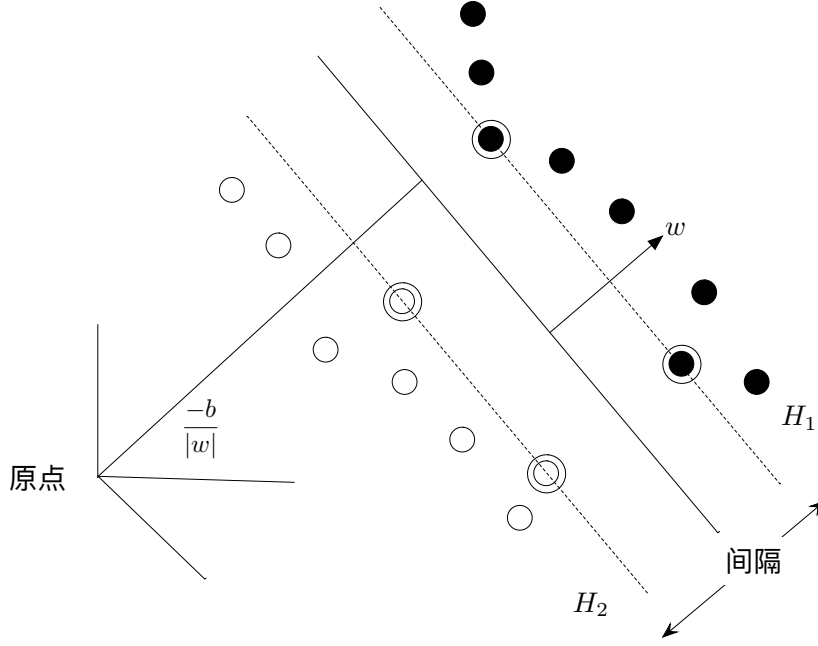


图 3-5: 标准分类平面，即具有最大间隔的超平面。画圆圈的样本组成了这个超平面，其被称作支持向量。

因此求解标准分类超平面的最大间隔的问题，就转变为下面的优化问题。

$$\min_{w \in \mathcal{H}} \tau(w) = \frac{1}{2} \|w\|^2 \quad s.t. \quad y_i(w \cdot x_i + b) \geq 1 \quad \forall i \quad (3-13)$$

为了使得约束更好表示，我们用拉格朗日优化算法对上式重新描述，

$$\min_{w, b} L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i y_i (x_i w + b) + \sum_{i=1}^l \alpha_i \quad (3-14)$$

其中  $\alpha_i \geq 0$  为约束条件。

在实际计算过程中，我们通过对偶定义求解优化方程3-14，通过最大化方程3-14相对于  $\alpha$  来求取其相对于  $w$  和  $b$  的最小值。利用Karush-Kuhn-Tucker 条件，则公式3-14变为下面对偶形式：

$$\max_{\alpha} L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad s.t. \quad \forall i \begin{cases} \sum_i \alpha_i y_i = 0 \\ \alpha_i \geq 0 \end{cases} \quad (3-15)$$

因此，通过求解这个对偶优化问题，可以得到系数  $\alpha_i$ 。其中满足  $\alpha_i > 0$  的解称作支持向量，他们位于标准分类平面  $H_1$  或者  $H_2$  上。注意到，仅有  $\alpha_i > 0$  的解影响最终的支持向量的选择。因此，可以得到决策函数：

$$f(x) = w^T x + b = \sum_{i=1}^M y_i \alpha_i (x_i^T x) + b \quad (3-16)$$

决策函数的符号取决于预测样本 $x$ 。

此处我们讨论的情形的一个假设是，我们可以把所有的样本完全分为不同的类别。但是显然在大多数情况下，这种假设是不成立的。另外，这种假设也会导致过拟合现象的出现。因此，文献[21] 提出了软间隔的支持向量机。其基本思想是，通过引入正的松弛变量 $\xi_i$ 来放宽公式3-9和3-10的约束。基于此，得到公式3-17

$$\forall i \begin{cases} w \cdot x_i + b \geq +1 - \xi_i & y_i = +1 \\ w \cdot x_i + b \leq -1 - \xi_i & y_i = -1 \\ \xi_i \geq 0 \end{cases} \quad (3-17)$$

这允许一些样本在边缘内部，甚至在相反类别的情况下进一步交叉（见图3-6）。虽然这种松弛使得支持向量机能够灵活地降低异常值的影响，但是从优化问题求解的角度来看，我们不希望有任意大的松弛变量 $\xi_i$ ，因为这会导致SVM获得平凡和次优的解。因此，我们通过使松弛变量成为目标函数3-13的一部分，来限制松弛度：

$$\min_{w \in \mathcal{H}, \xi \in \mathbb{R}^m} \tau(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad (3-18)$$

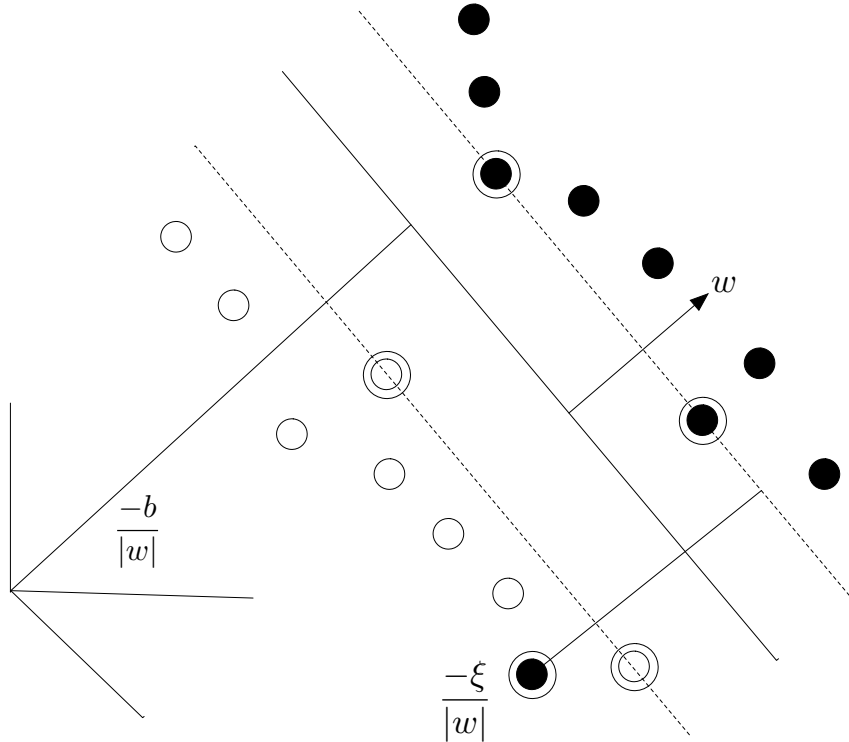


图 3-6: 软间隔支持向量机

其约束条件为公式3-17。超参数 $C > 0$ 是针对于误分类的惩罚系数，该系数需要根

据不同的分类任务和数据集进行调整。将其变为的对偶形式，则有

$$\max_{\alpha} L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad s.t. \quad \forall i \begin{cases} \sum_i \alpha_i y_i = 0 \\ C \leq \alpha_i \leq 0 \end{cases} \quad (3-19)$$

目前只是分析了线性支持向量机的问题，为了应对非线性分类问题，我们引入了核函数的概念。将训练数据通过某函数  $\Phi: \mathbb{R}^d \mapsto \mathcal{H}$ 。经过该变换后，我们只需要将原来计算  $\mathbb{R}^d$  的  $x_i \cdot x_j$  变换为计算在  $\mathcal{H}$  域的向量积  $\Phi(x_i) \cdot \Phi(x_j)$ 。为了降低计算量，可以引入核函数  $K$  来避免数据  $x_i$  和  $x_j$  从  $\mathbb{R}^d$  映射到  $\mathcal{H}$ 。

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (3-20)$$

因此，可以将公式3-19变为下式：

$$\max_{\alpha} L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad s.t. \quad \forall i \begin{cases} \sum_i \alpha_i y_i = 0 \\ C \leq \alpha_i \leq 0 \end{cases} \quad (3-21)$$

常用的核函数有下面几种：

- 线性核函数:  $K(x, y) = \langle x, y \rangle$
- 多项式核函数:  $K(x, y) = (\langle x, y \rangle)^d$
- RBF 核函数:  $K(x, y) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right)$
- Sigmoid 核函数:  $K(x, y) = \tanh(\gamma \langle x, y \rangle - \theta)$

## 支持向量机设计

我们可以利用所有的目标数据和未知目标的数据来作为训练样本对该SVM分类器进行训练，本部分我们以深度卷积神经网络的输出作为该分类器的输入，利用各类别的概率作为其特征进行训练识别。由于在类别的识别过程中，存在一定的波动性，这个会影响对于是否属于未知类别的分类判断，我们这里选取对于来自同一个辐射源的连续10拍的识别结果进行一个平均作为最终的输入。

下面是对于SVM分类器的设计，首先是核函数的选择。核函数将输入空间映射到高维特征空间，最终在高维特征空间中构造出最优分类超平面，从而把平面上本身不好分的非线性数据分开。常用的核函数为线性核函数和径向基核函数（Radial Basis Function, RBF）。对于核函数的选择，一般分为三种情况：

- 特征的数量很大，跟样本数量差不多，这时候选用逻辑回归算法或者是线性核的SVM
- 特征的数量比较小，样本数量一般，不算大也不算小，选用径向基核函数
- 特征的数量比较小，而样本数量很多，需要手工添加一些特征变成第一种情况

由于我们的问题符合第二种情况，故选择径向基核函数。

具有径向基核的支持向量机具有两个关键参数，惩罚参数 $C$ 的和核参数 $\sigma$ ，这两个参数的取值在很大程度上决定了SVM的性能的优劣。核函数的参数主要影响样本数据在高维特征空间中分布的复杂程度，即维数。特征子空间的维数越高，那么得到的最优分类超平面就会越复杂。反之亦然。因此只有选择合适的核参数得到合适的特征子空间，才能得到推广能力良好的SVM分类器。大量的实验数据表明，如果与样本点之间的距离很小， $\sigma \rightarrow 0$ ；如果与样本点之间的距离很大时， $\sigma \rightarrow \infty$ ；当 $\sigma$ 很小，径向基核函数支持向量机得到的判别函数差不多是一个常数，出现过拟合现象。当 $\sigma$ 很大时，样本的正确分类率也会比较低。

惩罚参数是影响SVM算法性能的另一个重要因素。它的作用主要是调节特征子空间中SVM模型的置信范围与经验风险的比例，使支持向量机的泛化能力达到最好。特征子空间不同时，最优参数值取值也会不同。惩罚参数与经验误差的惩罚和SVM的复杂度成正比，与经验风险值成反比，反之亦然。因此，选择合适的惩罚参数也是非常重要的。

从上面的分析可以看出，核参数影响着映射函数、进而影响样本子空间的复杂度。最后会影响分类器性能的好坏。惩罚参数作用是在数据子空间中调节支持向量机置信区间的范围。这些都说明了惩罚参数和核参数的选择非常重要。

### 3.4 仿真实验与分析

#### 3.4.1 实验环境

##### 关于IQ两路数据的一个叙述

本章利用了来自13架民航飞机的气象雷达辐射源的数据，飞机信息参见下表

表 3-1: 辐射源信号数据

飞机地址码	飞机航班号	飞机型号
7BF014	CCA1416	Airbus A330 (twin-jet)
780DB3	TBA9881	Airbus A330 (twin-jet)
780E06	CSN3438	Airbus A330 (twin-jet)
780EBB	CES293	Airbus A321 (twin-jet)
780EBF	CES2342	Airbus A320 (twin-jet)
7804F2	ZH9164	Airbus A320 (twin-jet)
7804F4	CES5856	Boeing 737
7806FC	CSN6402	Airbus A320 (twin-jet)
7808F0	CES5373	Airbus A319 (twin-jet)
78057F	CCA4103	Airbus A321 (twin-jet)
780063	CCA4377	Airbus A319 (twin-jet)
780375	CSC8253	Airbus A319 (twin-jet)
781022	EU2731	Airbus A319 (twin-jet)

对于我们的气象雷达辐射源信号，原始的I/Q数据的示意图如图3-7 所示

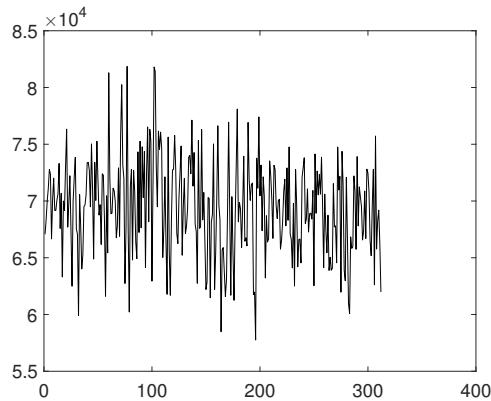


图 3-7: 原始I/Q信号

原始的雷达信号为I/Q信号，我们将他们利用快速傅里叶变换求取其模糊函数切片作为输入的特征向量。每架飞机原始信号为100次雷达扫描周期的信号，为了增加数据样本数目，我们对原始信号添加噪声生成了新的信号，最终每架飞机均有10000组信号。对于这些信号，我们选择其中的70%作为训练样本，20%用于交叉验证，10%作为测试，同时在测试样本中添加与其等量的来自未训练类别也即未知类别的样本共同作为测试样本用来衡量其未知分类的识别率。

由于我们原始获得的数据为I/Q两路数据，为了更好的捕捉到回波的特征信息，我们对数据进行了一个变换，求取其模糊函数，并做偏移为0附近的一个切片。由于深度学习需要大量的数据进行训练学习，而本身数据量偏少，故我们在已有数据的基础上在一定信噪比的前提下，生成部分仿真数据。

对于数据的选择方面，我们从数据中选择出2至8个类别分别进行实验，对于每一个类别，我们均选择大约10000组数据，其中70%作为训练样本，20%作为交叉验证样本，10%作为测试样本，同时在测试样本中又添加了与已知分类等数量的未知分类的数据进行测试。由于该变换后，数据之间的差距比较大，我们对数据进行了归一化。

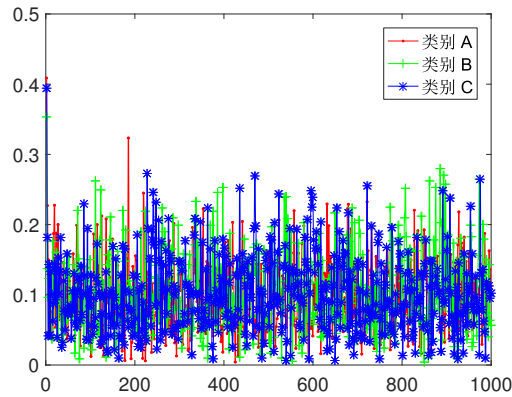


图 3-8: 不同类别样本特征图

我们采用的归一化方法为min-max标准化（Min-Max Normalization），也称为离差标准化，是对原始数据的线性变换，使结果值映射到[0, 1]之间。转换函数如下：

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3-22)$$

其中 $x_{max}$ 为样本数据的最大值， $x_{min}$ 为样本数据的最小值。

我们采用[网格搜索法](#)对SVM参数进行调优，最终选择参数惩罚参数为32，核参数 $\sigma$ 为0.0312。



### 3.4.2 实验结果分析

#### 深度卷积神经网络识别结果

该部分是否修改为表格，然后注意描述的时候准确表达所运用的数据，该部分主要利用某一个多类别的进行研究。

首先验证深度卷积算法的识别正确率，利用8个类别的数据进行训练和测试，迭代次数为100次。

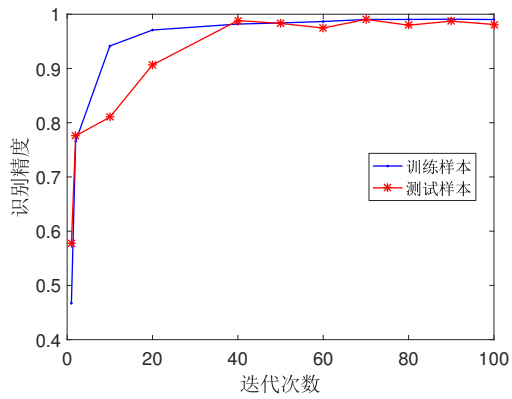


图 3-9: 迭代次数与识别准确率曲线图

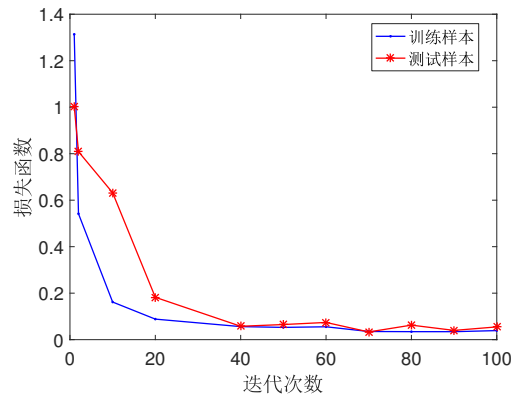


图 3-10: 迭代次数与损失函数曲线图

表 3-2: 深度卷积神经网络分类结果

	类别A	类别B	类别C	类别D	类别E	类别F	类别G	类别H
类别A	100%	0	0	0	0	0	0	0
类别B	7.42%	92.58%	0	0	0.13	0	0	0
类别C	0	0	97.83%	0	0	2.17%	0	0
类别D	0	0	3%	97%	0	0	0	0
类别E	0	0	0	0	100%	0	0	0
类别F	0	0	0.08%	0	0	99.92%	0	0
类别G	0	0	1.08%	0	0	0	98.92%	0
类别H	1.63%	0	0	0	0	0.63	0	98.37%

表 3-3: 深度卷积神经网络分类结果

迭代次数	训练样本集学习正 确率	测试样本集学习正 确率	学习时间 (s)
1	0.4670	0.5782	$1.0931 \times 10^3$
2	0.7658	0.7758	$2.3572 \times 10^3$
10	0.9414	0.8102	$1.1444 \times 10^4$
20	0.9709	0.9065	$2.1370 \times 10^4$
40	0.9819	0.9883	$4.3181 \times 10^4$
50	0.9839	0.9830	$5.3437 \times 10^4$
60	0.9866	0.9746	$6.4143 \times 10^4$
70	0.9903	0.9904	$7.4849 \times 10^4$
80	0.9904	0.9798	$8.5597 \times 10^4$
90	0.9908	0.9872	$9.6328 \times 10^4$
100	0.9902	0.9811	$1.0711 \times 10^5$

对于具有8个类别的数据，我们对于迭代次数与识别准确率进行了验证，其结果如图3-9 所示。

### Open Set 分类器识别结果

在上述样本的情形下，我们通过选取不同的类别数目，进行训练和测试得到表3-4的识别结果。从表中数据我们可以看出，随着样本类别数的增加，对于未知分类的识别准确率也随之有了大幅度的增加，而另一方面随着类别的增加，对于每个类别的识别准确率有一定的降低，但是仍然维持在比较高的水平。

表 3-4: 不同已知类别个数数据识别结果

已知类别数目	已知类别识别正确率	未知类别分辨正确率
2	99.55%	84.32%
3	98.50%	93.10%
4	98.56%	97.81%
5	98.48%	98.42%
6	96.32%	98.85%
7	96.26%	99.22%
8	96.08%	99.14%

### 3.5 小结

本章针对复杂电磁环境下辐射源的识别面临的电磁信号干扰大、雷达信号参数相近等问题与挑战，利用深度学习的思想与方法，深入研究辐射源脉内细微特征，设计合适的神经网络结构，并基于民航机载气象雷达数据进行初步验证。主要特色与创新点如下：

（1）利用深度学习方法进行辐射源识别前沿。通过对现有辐射源信号进行分析，利用其脉内细微特征作为训练样本，使得识别准确率有了较大的进步。虽然已有研究利用神经网络、支持向量机等机器学习算法进行识别，但是仍然需要基于雷达信号的基本参数，没有考虑信号的内部特征参数。

（2）本章采用方法具有较强的抗噪声、抗干扰能力以及较好的鲁棒性。传统方法进行辐射源个体识别前均需进行降噪、多径抑制和分选等复杂的信号预处理工作，这些操作会在一定程度上削弱雷达的个体特征。深度学习方法可以通过大量的样本，智能地判断各特征的权重，通过赋予不同的权重在保留雷达个体特征的情况下，避免干扰的影响。



## 4 基于深度嵌入卷积聚类方法的地海杂波无监督分类

### 4.1 引言

由于有监督学习需要对大量的数据进行打标签，这个过程通常需要花费大量的时间与精力。无监督学习也即聚类算法，由于其不需要数据标签，因此国内外学者对各个领域的聚类算法进行了广泛研究，产生了各种不同类型的聚类算法，例如层次聚类[22]，基于质心的聚类方法[23]、基于图的聚类方法[24]、基于回归模型[25]的聚类方法以及基于子空间[26]的聚类方法。另一方面，这些聚类方法一般可以分为两类，即生成式和判别式聚类算法。像K均值和高斯混合模型[27]这样的生成算法使用特征空间的几何特性来明确地表示簇，并且通过输入数据的统计分布来对分类进行建模。与生成式聚类算法不同，判别式方法直接使用分离超平面来识别类别，而不管数据的分布是怎样的。信息论[28]，支持向量机[29]和图谱论[30]算法是判别式聚类模型的例子。一般认为，判别式模型与其对应的生成式模型相比往往具有更好的结果，因为他们对数据分布的假设较少，直接将聚类分开，但是他们的训练可能会出现过拟合或陷入局部最优解的问题出现[31]。我们的DECC算法也是一种判别式聚类算法，但它受益于自编码器的辅助重构任务，以缓解我们判别式聚类算法的训练中的这个问题。

虽然，聚类问题已经在各种领域进行了应用研究，但是传统的聚类算法的性能收到数据维数以及数据量的影响，会产生维数灾难的问题。为了解决计算复杂度的问题，以往的研究往往首先将数据投影到一个低维流形中，然后将嵌入的数据聚类到这个新的子空间中[32]。处理大规模的数据集，也有一些研究只选择数据点的一个子集来加速聚类过程[33]。

自编码器(Autoencoder, AE)是简单的神经网络，旨在尽可能减少失真的情况下将输入转换成输出。虽然在概念上简单，但其在机器学习中起着重要的作用。自编码器是20世纪80年代由Hinton和PDP小组[34]首次提出的，用输入数据作为标签来解决无监督反向传播的问题。自编码器与Hebbian学习规则[1]一起为无监督学习提供了一个基本的范例，并开始解决如何以自组织的方式协调局部事件引起的变化去产生全局学习和智能表现。

深度聚类算法中使用最广泛的神经网络是堆栈自编码器[35]。SAE需要逐层预训练，然后以端对端(end-to-end)的方式进行微调。当层次更深时，预训练过程是十分耗

时的。此外,SAE是建立在全连接的层次上,对于处理局部信息是无效的。文献[36]的工作是第一个在不需要预训练的情况下直接以端到端的方式直接训练CAE的试验。

就要在特征空间中保存的数据的属性而言,基本的思路通过向目标函数添加先验知识来考虑稀疏性或图形约束[37]。这类算法可以分为两个阶段:特征学习和聚类。后来,文献[35]提出了共同完成特征学习和聚类的算法。深度嵌入聚类算法[35]以自学习的方式定义了一个有效的目标函数。定义的聚类损失用于同时更新网络和聚类中心的变换参数。但是,他们忽略了数据属性的保存,这可能导致特征空间的破坏。我们通过保留数据生成分布的局部结构并结合卷积层来改进DEC算法。

特征学习指的是一类试图利用一组基本向量或特征来描述一个数据集的学习方法,并且通常这个表示是稀疏的。在实践中,有很多不同的算法可以进行特征学习,包括自编码器、k-均值聚类、高斯混合模型和受限玻尔兹曼机(RBM)[38]。这些方法都倾向于学习类似的局部滤波器字典[38],例如用于自然图像的Gabor类边缘滤波器,或者用于MNIST数字数据集的书写笔划。虽然RBM是可以从数据生成分布中抽样的生成模型[39],但是自编码器被训练来优化它们对输入数据的重构。

最近,自编码器在深度架构方法[6, 7, 40, 41]中再次占据了中心位置,,特别是形式上类似受限玻尔兹曼机的自编码器,以堆栈形式组织,然后以无监督的方式自下而上地进行训练,随后是以有监督学习的方式训练顶层并微调整个架构。考虑到自下而上训练阶段对于最终的任务是不可知的,因此显然可以用于迁移学习方法。并且实际的实验已经证明这些深层次的体系结构可以在一些具有挑战性的分类和回归问题上得到最优的结果。

尽管广大学者已经产生了很大的兴趣,但除了文献[42, 43, 44]等,对自编码器和深层架构的理论认识的文章还很少。除此之外,使用深度一词可能会造成更多混淆。从计算机科学的角度来看,深层结构应该对于一些小的 $\alpha > 0$ 有 $n^\alpha$ 个多项式大小的层,其中 $n$ 是输入向量的大小[45]。但是在Hinton等人[6, 7]描述的架构中并不是这种情况,似乎有恒定或最好的对数深度。对于计算机视觉、语音识别和其他典型问题,有限和对数的深度之间的区别很小,几乎可以忽略。文献[46]对自编码器的理论进行了描述,提出了一个用自编码器解决线性与非线性问题的通用的数学框架。

如果存在关于深度架构的一般理论结果,则这些结果不太可能取决于特定的硬件实现,如RBM。相似的结果对于替代计算或更一般的计算形式也是如此。因此,这里提出的策略是引入一个通用框架,并研究不同类型的自编码器电路,特别是可以被看作是非线性自编码器的最极端形式的布尔自编码器。期望的是,在更简单的硬件实施例中,自编码器和深构架的某些特性可以更容易地在数学上识别和理解,并且对不同

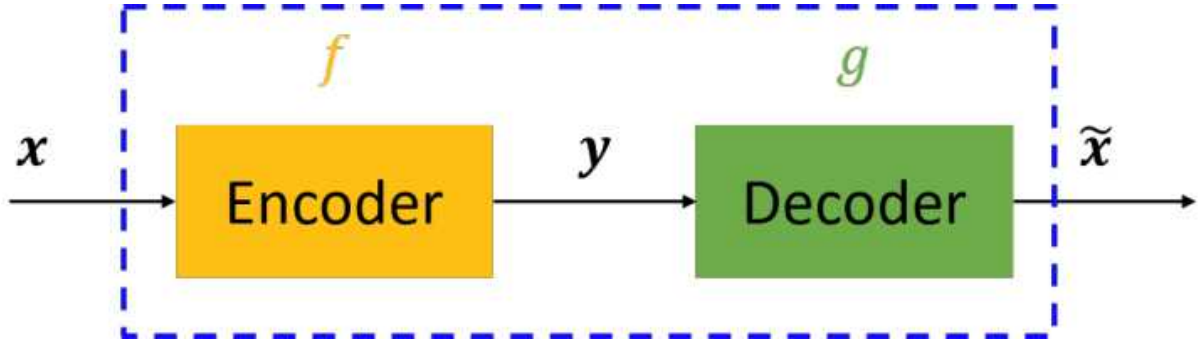


图 4-1: 自编码器模型

种类的自编码器的研究可以通过共同特性的标识来促进抽象和概括。

## 4.2 卷积自编码网络

### 4.2.1 自编码器

自编码器是一种隐层用来重构输入的神经网络。其模型如图4-1所示，它由编码器(Encoder)和解码器(Decoder)两部分组成，本质上是对输入信号的某种变换。编码器将输入信号  $x \in [0, 1]^{n_x}$  编码到某个隐层的表示  $h \in [0, 1]^{n_h}$ ，然后解码器再将  $h$  解码为输出  $x' \in [0, 1]^{n_x}$ 。通过最小化  $x$  与  $x'$  之间的差距，可以训练得到一个特征表示的映射  $h$ ，其可以用来重构输入。编码和解码公式如下：

$$h_j = \text{sigm}(b_j + \sum_{i=1}^{n_x} w_{ij}x_i) \quad (4-1)$$

$$x'_i = \text{sigm}(a_i + \sum_{j=1}^{n_h} w'_{ij}h_j) \quad (4-2)$$

其中， $n_x$  和  $n_h$  分别为输入表示和隐层表示的维数， $\text{sigm}(x) = \frac{1}{1+e^{-x}}$  是logistic损失函数， $\mathbf{W} = [w_{ij}]$  和  $\mathbf{W}' = [w'_{ij}]$  为权重矩阵， $\mathbf{a} = [a_i]$  和  $\mathbf{b} = [b_j]$  为偏差向量。

对于一组具有  $N$  个输入向量的给定训练数据集，设每个训练向量为  $x^{(n)}$  可以被映射到一个隐藏层表示  $h^{(n)}$ ，其重构表示为  $x^{2(n)}$ 。模型参数  $\Theta = \mathbf{W}, \mathbf{a}, \mathbf{b}$  通过最小化其损失函数进行求取，一般利用均方误差损失函数(mean squared error, mse)：

$$J(\theta) = \frac{1}{N} \sum_{n=1}^N \|x^{(n)} - x'^{(n)}\|_2^2 \quad (4-3)$$

$$\theta^* = \arg \min_{\theta} J(\theta) \quad (4-4)$$

通常，使用随机梯度下降 (SGD) 或其余基于梯度的优化算法来对公式4-4求解。

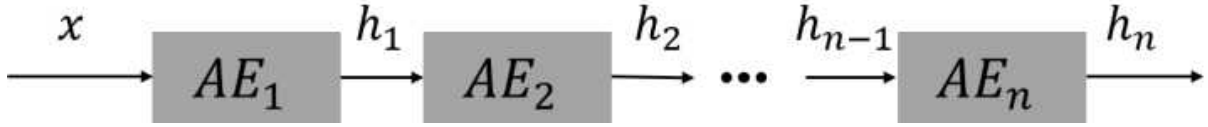


图 4-2: 堆栈自编码器模型

在此基础上，学者对自编码器进行了各种改进，提出了可以学习输入空间中扰动不变的特征的压缩自编码器[47]、提供有效的变分方法进行训练和生成推理的变分自编码器[48] 以及可以基于连续的数据流在运行中添加或合并隐藏单元的在线增量自编码器[49]。

#### 4.2.2 卷积自编码器

传统的自编码网络只有一个隐藏层，只能学习出一种特征变化，于是Bengio[50]等人在2007年的仿照stacked RBM构成的DBN，提出堆栈自编码器（Stacked Auto-Encoder, SAE），为非监督学习在深度网络的应用又添了猛将。堆栈自编码器的基本模型可以参看图4-2，其并没有这里就不得不提“逐层初始化”（Layer-wise Pre-training），目的是通过逐层非监督学习的预训练，

来初始化深度网络的参数，替代传统的随机小值方法。预训练完毕后，利用训练参数，再进行监督学习训练。

基于此，文献[51]提出了一种

完全连接的自编码器和深度自编码器都忽略了数据的局部结构信息。这在处理实际大小的输入时不仅是一个问题，而且在参数中引入冗余，迫使每个特征是全局的。然而，最成功的模型[52]采用的视觉和对象识别的趋势是发现局部特征，在整个输入中重复。卷积自编码器不同于传统的自编码器，因为它们的权重在输入中的所有位置之间共享，从而保持空间局部性。因此重建是由于基于数据的隐藏局部特征的线性组合。

CAE架构直观地类似于AE的架构，除了权重是共享的。对于输入x，第k个特征映射的潜在表示由下式给出

$$h^k = \sigma(x * W^k + b^k) \quad (4-5)$$

其中偏差也是全局共享的， $\sigma$  是一个激活函数（我们在我们所有的实验中使用缩放的双曲正切）， $*$ 表示二维卷积。因为我们希望每个滤波器专注于整个输入的特征（每个像素一个偏差将引入太多的自由度），所以使用每个潜在图的单个偏差。重建是使用获



得的

$$y = \sigma\left(\sum_{k \in H} h^k * \tilde{W}^k + c\right) \quad (4-6)$$

同样的对于每一个输入通道只有一个偏差 $c$ 。 $H$ 表示潜在特征图的组。 $\tilde{W}$ 标识权重的两个维度上的翻转操作。公式4-5和4-6中的2D卷积操作由上下文决定。一个 $m \times m$ 矩阵和一个 $n \times n$ 矩阵的卷积结果可能是 $(m + n - 1) \times (m + n - 1)$ (full convolution)也可能是 $(m - n + 1) \times (m - n + 1)$ (valid convolution)。其损失函数为最小均方误差(mean squared error, MSE)

$$E(\theta) = \frac{1}{2n} \sum_{i=1}^n (x_i - y_i)^2 \quad (4-7)$$

就像标准的神经网络一样，反向传播算法被用来计算相对于参数的误差函数的梯度。这可以通过使用以下公式的卷积运算容易地得到：

$$\frac{\partial E(\theta)}{\partial W^k} = x * \delta h^k + \tilde{h}^k * \delta y \quad (4-8)$$

$\delta h$  和  $\delta y$  分别是隐藏层状态和重构结果的变化量。然后使用随机梯度下降来更新权重。

我们的主要思想是，CAE有利于保留数据的局部结构，避免特征空间的失真。

我们介绍了一种用于分层特征提取的无监督方法,卷积自编码器。它学习生物学上似是而非的过滤器。一个CNN可以由一个CAE堆栈来初始化。虽然CAE的完整隐藏表示使得学习比标准自编码器更难，但是如果我们使用最大池化层，好的过滤器就会出现，这是一种执行稀疏代码的优雅方式，不需要通过反复试验来设置正则化参数。

### 4.3 深度嵌入卷积聚类方法

深层嵌入式聚类（Deep Embedding Clustering, DEC）[35]算法提供了一种无监督的方式学习特征表示和聚类的思路，通过联合优化神经网络和聚类中心提高了聚类算法的性能和鲁棒性。但是，该算法由于利用了堆栈自编码器，无法利用数据的局部特征信息，因此本章在此基础上提出了深度嵌入卷积聚类方法(Deep Embedding Convolution Clustering, DECC)，利用卷积自编码器代替传统的堆栈自编码器。

#### 4.3.1 深度嵌入卷积聚类方法结构

首先对于问题进行形式化描述，假设需要将 $N$ 个样本 $X = [x_1, \dots, x_N]$ 聚为 $K$ 个类别，其聚类中心点为 $\mu_1, \dots, \mu_K$ ，其中每一个样本 $x_i \in \mathbb{R}^{d_x}$ ，由于对于维数较大的样本会产生维数灾难的问题，因此需要引入一个嵌入函数 $\varphi_W : X \rightarrow Z$ ，可以将原始样本

映射到嵌入子空间  $Z = [z_1, \dots, z_N]$ ，其中  $z_i \in \mathbb{R}^{d_z}$  的维数远小于原始样本的维数，也即  $d_z \ll d_x$ 。

本章利用卷积自编码器作为嵌入函数对原始样本进行降维，其基本结构如图xxx所示。通过三次卷积操作充分提取输入样本的特征，我们在所有层上均利用ReLU激活函数，并且在编码器后以相反的顺序放置解码器，由于编码器为卷积操作，故解码器为逆卷积(Deconvolution)。在编码器与解码器中间添加嵌入层，该层的结果被用于进行聚类操作。DECC的结构如图xxx 所示。

#### 4.3.2 KL散度聚类方法

文献[53]利用学生t分布来衡量嵌入点  $z_i$  与聚类中心点  $\mu_k$  的相似度：

$$p_{ik} = \frac{(1 + \|z_i - \mu_k\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'} (1 + \|z_i - \mu_{k'}\|^2/\alpha)^{-\frac{\alpha+1}{2}}} \quad (4-9)$$

其中， $\alpha$  为学生t分布的自由度， $p_{ik}$  可以看作样本  $i$  到类别  $k$  的概率。

为了定义我们的聚类目标函数，我们使用辅助目标变量(auxiliary target variable) $Q$ 来迭代地改进模型预测。为此，我们首先使用Kullback-Leibler (KL) 散度来减小模型预测  $P$  和目标分布  $Q$  之间的距离。

$$\mathcal{L}_{kld} = KL(Q||P) = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K q_{ik} \log \frac{q_{ik}}{p_{ik}} \quad (4-10)$$

根据文献[35]，目标分布  $Q$  的定义如下：

$$q_{ij} = \frac{p_{ij}^2 / f_k}{\sum_k (p_{ik}^2 / f_k)} \quad (4-11)$$

其中  $f_k$  的定义为

$$f_k = \sum_i q_{ik} \quad (4-12)$$

公式4-4给出了自编码器重构的损失函数，故最终的损失函数为：

$$\mathcal{L} = \mathcal{L}_{mse} + \beta \mathcal{L}_{kld} \quad (4-13)$$

其中， $\beta > 0$  为调节两部分损失函数的权重。

### 4.4 仿真验证

#### 4.4.1 参数设置

本章根据如图设计的卷积自编码器进行特征的降维，该部分进行30次迭代，学习方法为Adam。然后利用随机20次k均值算法然后从中选取最优结果，作为起始的聚类

中心点，此也作为我们的一个对比算法。损失函数的权重系数 $\beta = 0.1$ ，学生t的自由度 $\alpha = 1$ 。

#### 4.4.2 仿真结果分析

我们利用第三章中组C的训练数据用于本章的实验。其实验结果如表4-1所示。

表 4-1: 辐射源信号数据

方法	无监督聚类精度	时间
CAE + k均值	66%	12
DECC ( $\beta = 0$ )	23%	12
DECC	453%	12

此处无监督聚类精度（unsupervised clustering accuracy, ACC）的定义为：

$$ACC = \max_m \frac{\sum_{i=1}^n g(l_i, m(c_i))}{n} \quad (4-14)$$

$$g(l_i, m(c_i)) = \begin{cases} 1 & l_i = m(c_i) \\ 0 & l_i \neq m(c_i) \end{cases}$$

其中， $l_i$ 为真实的标签， $c_i$ 为算法的聚类结果， $m$ 是聚类结果与实际标签类别之间的一一对应，其遍历所有可能假设。

#### 4.5 小结

本章结合卷积神经网络与自编码器提出了深度嵌入卷积聚类方法DECC。其通过在损失函数中将KL散度与重构损失两部分结合，使得本章的算法对于两部分可能都进行了考虑。

虽然与有监督学习算法相比，本章的算法精度仍然不够高，但是相比于传统的聚类算法精度有了一定的提高。在一定程度上可以减少打标签的过程。

本章的贡献是：

- 可以以端对端方式训练的卷积自编码器（CAE）被设计用于从未标记的图像学习特征。通过在图像中包含像素之间的空间关系，设计的CAE优于堆叠的自编码器。我们证明卷积层，卷积转置层和完全连通层足以构建一个有效的CAE。

- 根据聚类导向损失函数调整网络参数时考虑局部结构保留。我们证明，保持局部结构有助于稳定训练过程，避免特征空间的腐败。
- 我们提出了深度嵌入卷积聚类算法来自动聚类图像。DECC利用CAE本地结构保存的优势。最终的优化问题可以通过小批量随机梯度下降和反向传播得到有效解决。
- 对基准图像数据集进行了大量的实验。结果验证了CAE的有效性和局部结构的保存。

## 5 总结

### 5.1 本文的主要贡献

雷达信号分类一直是雷达领域一个很重要的方面，通过对接受到的信号的分类识别，可以更好的了解战场形势。在电子科技迅猛发展的今天，电磁环境急剧变化，雷达信号分类在军事侦查、目标识别、电子对抗等领域有着广阔的应用场景。而深度学习方法由于其在识别分类等领域的结果，引起了国内外越来越多学者的重视。本文对于雷达信号分类中的两个方面进行了研究，通过对雷达信号数据的分析，构建合理的卷积神经网络模型，实现了对于雷达信号类别的识别。同时，针对于识别过程中存在未知类别的情况，构建了一个支持向量机分类器作为Meta-Recognition进行二次识别，实现了未知分类的辨别。本文的主要研究成果和最终的结论如下：

本文首先综述了深度学习的研究现状和发展，以及其与传统神经网络方法相比的优点所在。描述了深度学习的几种常用方法和基本结构，研究了深度卷积神经网络的原理和其训练过程。

针对于天波超视距雷达中由于电离层的变化无法确定其坐标配准系数的问题，本文通过地海杂波识别进行地理位置匹配来获取该变换系数，用于提高目标定位精度。在地海杂波识别过程中，通过对于海量频谱数据的分析，构建了一个一维卷积神经网络分类器。同时将该算法的识别结果与传统的基于信号频谱分析和支持向量机算法进行对比，证明了本文算法的优越性。

对于辐射源识别中未知分类的辨别问题，首先概述了辐射源识别中雷达信号处理的过程，介绍了辐射源特征提取和辐射源识别的常用方法及存在的问题。本文将深度学习和支持向量机相结合，构建了用于辐射源分类及未知分类辨别的模型，通过实际数据进行实验验证，并对不同辐射源类别数、卷积神经网络层数、节点数、不同支持向量机参数与正确率进行了比较，讨论了相关参数对结果的影响。

### 5.2 后续的研究进展

在本文中，我们利用实际数据进行测试并验证了我们的深度学习方法在地海杂波识别方法和辐射源识别这两个方面的有效性。但是由于雷达数据处理方面信号种类多，在实际工程中涉及环节多，因此，对于利用深度学习进行雷达信号处理还有很多工作

需要完成。结合本文所研究的问题，我们希望未来可以在以下方面做进一步研究：

- 针对于雷达信号特征，对本文算法进一步调整。在天波雷达地海杂波识别中，我们现在把已有数据分成几组，分别进行训练。我们计划尝试使用一些方法对这些数据进行融合分析，以获得一个可以适应各种数据情况的模型。对于辐射源识别，利用更多的数据对算法进行进一步的验证，目前类别较少的情况下，部分结果会对选择的类别具有一定的依赖性，另一方面是选取更多更合适的特征进行训练学习。
- 进一步优化算法，提高计算效率。虽然在数据量比较大的情况下，深度学习算法具有准确率高的优势，但训练过程计算量较大。本文拟将深度学习方法同其他方法进行结合，进一步完善网络结构，降低计算量、提高训练速度。
- 进一步优化网络结构、相关参数的选取、训练方法等。深度学习的理论研究仍然存在一些不足，可以通过进一步的理论研究，选择更加优秀的参数和训练方法。除了本文的卷积神经网络结构之外，我们计划尝试其他深度学习的方法或者思想来解决我们的问题或者构建更优的神经网络结构。
- 对数据进行无监督或者半监督学习的方式进行训练。由于雷达信号量大，人为进行标记困难较大，本文计划进一步尝试无监督等减少人为标记的工作量。
- 本文目前的工作更偏重于工程应用，对于深度学习相关理论知识的研究不足，后续需要增强理论知识的研究。

## 参考文献

- [1] Hebb DO. The organization of behavior: A neuropsychological theory[M]. Psychology Press. 2005.
- [2] Fukushima K, Miyake S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition[C]. In Competition and cooperation in neural netsSpringer. 1982, :267–285.
- [3] LeCun Y et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE. 1998, 86(11):2278–2324.
- [4] LeCun Y et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation. 1989, 1(4):541–551.
- [5] Simard PY et al. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis.[C]. In ICDAR. Citeseervol. 32003:958–962.
- [6] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks[J]. science. 2006, 313(5786):504–507.
- [7] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets[J]. Neural computation. 2006, 18(7):1527–1554.
- [8] Ciresan DC et al. Flexible, high performance convolutional neural networks for image classification[C]. In Twenty-Second International Joint Conference on Artificial Intelligence. 2011.
- [9] Hinton G et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. Signal Processing Magazine, IEEE. 2012, 29(6):82–97.
- [10] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks[C]. In Advances in neural information processing systems. 2012:1097–1105.

- 
- [11] Collobert R et al. Natural language processing (almost) from scratch[J]. Journal of Machine Learning Research. 2011, 12(Aug):2493–2537.
- [12] Jouppi NP et al. In-datacenter performance analysis of a tensor processing unit[J]. arXiv preprint arXiv:1704.04760. 2017.
- [13] Silver D et al. Mastering the game of go without human knowledge[J]. Nature. 2017, 550(7676):354–359.
- [14] Behnke S. Hierarchical neural networks for image interpretation[M]vol. 2766. Springer Science & Business Media. 2003.
- [15] Turley M, Gardiner-Garden R, Holdsworth D. High-resolution wide area remote sensing for HF radar track registration[C]. In Radar (Radar), 2013 International Conference on. IEEE2013:128–133.
- [16] Jin ZL et al. SVM Based Land/Sea Clutter Classification Algorithm[C]. In Applied Mechanics and Materials. Trans Tech Publvol. 2362012:1156–1162.
- [17] Scheirer WJ et al. Toward open set recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013, 35(7):1757–1772.
- [18] Jain LP, Scheirer WJ, Boulton TE. Multi-class open set recognition using probability of inclusion[C]. In European Conference on Computer Vision. Springer2014:393–409.
- [19] Simonson K. Probabilistic fusion of ATR results[R]. Tech. rep.Sandia National Laboratories (SNL-NM), Albuquerque, NM1998.
- [20] Scheirer WJ et al. Meta-recognition: The theory and practice of recognition score analysis[J]. IEEE transactions on pattern analysis and machine intelligence. 2011, 33(8):1689–1695.
- [21] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning. 1995, 20(3):273–297.
- [22] Heller KA, Ghahramani Z. Bayesian hierarchical clustering[C]. In Proceedings of the 22nd international conference on Machine learning. ACM2005:297–304.
- [23] Lloyd S. Least squares quantization in PCM[J]. IEEE transactions on information theory. 1982, 28(2):129–137.



- 
- [24] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Transactions on pattern analysis and machine intelligence. 2000, 22(8):888–905.
- [25] Wang H, Nie F, Huang H. Multi-view clustering and feature learning via structured sparsity[C]. In International Conference on Machine Learning. 2013:352–360.
- [26] Agrawal R et al. Automatic subspace clustering of high dimensional data for data mining applications[M]vol. 27. ACM. 1998.
- [27] Biernacki C, Celeux G, Govaert G. Assessing a mixture model for clustering with the integrated completed likelihood[J]. IEEE transactions on pattern analysis and machine intelligence. 2000, 22(7):719–725.
- [28] Li H, Zhang K, Jiang T. Minimum entropy clustering and applications to gene expression analysis[C]. In Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE. IEEE2004:142–151.
- [29] Xu L et al. Maximum margin clustering[C]. In Advances in neural information processing systems. 2005:1537–1544.
- [30] Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm[C]. In Advances in neural information processing systems. 2002:849–856.
- [31] Raina R et al. Classification with hybrid generative/discriminative models[C]. In Advances in neural information processing systems. 2004:545–552.
- [32] Roth V, Lange T. Feature selection in clustering problems[C]. In Advances in neural information processing systems. 2004:473–480.
- [33] Shinnou H, Sasaki M. Spectral Clustering for a Large Data Set by Reducing the Similarity Matrix Size.[C]. In LREC. 2008.
- [34] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation[R]. Tech. rep.California Univ San Diego La Jolla Inst for Cognitive Science1985.
- [35] Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis[C]. In International Conference on Machine Learning. 2016:478–487.
- [36] Dundar A, Jin J, Culurciello E. Convolutional clustering for unsupervised learning[J]. arXiv preprint arXiv:151106241. 2015.

- 
- [37] Tian F et al. Learning Deep Representations for Graph Clustering.[C]. In AAAI. 2014:1293–1299.
- [38] Coates A, Ng A, Lee H. An analysis of single-layer networks in unsupervised feature learning[C]. In Proceedings of the fourteenth international conference on artificial intelligence and statistics. 2011:215–223.
- [39] Hinton G. A practical guide to training restricted Boltzmann machines[J]. Momentum. 2010, 9(1):926.
- [40] Bengio Y, LeCun Y et al. Scaling learning algorithms towards AI[J]. Large-scale kernel machines. 2007, 34(5):1–41.
- [41] Erhan D et al. Why does unsupervised pre-training help deep learning?[J]. Journal of Machine Learning Research. 2010, 11(Feb):625–660.
- [42] Montufar G, Ay N. Refinements of universal approximation results for deep belief networks and restricted Boltzmann machines[J]. Neural Computation. 2011, 23(5):1306–1319.
- [43] Sutskever I, Hinton GE. Deep, narrow sigmoid belief networks are universal approximators[J]. Neural computation. 2008, 20(11):2629–2636.
- [44] Baldi P, Hornik K. Neural networks and principal component analysis: Learning from examples without local minima[J]. Neural networks. 1989, 2(1):53–58.
- [45] Clote P, Kranakis E. Boolean functions and computation models[M]. Springer Science & Business Media. 2013.
- [46] Baldi P. Autoencoders, unsupervised learning, and deep architectures[C]. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning. 2012:37–49.
- [47] Rifai S et al. Contractive auto-encoders: Explicit invariance during feature extraction[C]. In Proceedings of the 28th International Conference on International Conference on Machine Learning. Omnipress2011:833–840.
- [48] Kingma DP, Welling M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114. 2013.
- [49] Zhou G, Sohn K, Lee H. Online incremental feature learning with denoising autoencoders[C]. In Artificial Intelligence and Statistics. 2012:1453–1461.

- [50] Bengio Y et al. Greedy layer-wise training of deep networks[C]. In Advances in neural information processing systems. 2007:153–160.
- [51] Masci J et al. Stacked convolutional auto-encoders for hierarchical feature extraction[J]. Artificial Neural Networks and Machine Learning–ICANN 2011. 2011, :52–59.
- [52] Lowe DG. Object recognition from local scale-invariant features[C]. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Ieee vol. 21999:1150–1157.
- [53] Maaten Lvd, Hinton G. Visualizing data using t-SNE[J]. Journal of Machine Learning Research. 2008, 9(Nov):2579–2605.