

# 1 基于深度嵌入卷积聚类方法的地海杂波无监督分类

## 1.1 引言

聚类方法如今已经在各个领域的到了广泛应用,但是传统的聚类算法的性能受到数据维数以及数据量的影响,会产生维数灾难的问题。对于数据集规模较大的情形,有一些研究只选择数据集的一个子集来加速聚类过程<sup>[1]</sup>。为了解决计算复杂度的问题,以往的研究往往首先将高维度的数据映射到低维,然后将嵌入数据聚类到这个新的子空间中<sup>[2]</sup>。这类算法可以分为两个阶段:特征学习和聚类,其基本思路是向目标函数中添加表征稀疏性或者是图形约束的先验知识进行训练学习<sup>[3]</sup>。

特征学习指的是一类试图利用一组通常稀疏的基本向量或特征来描述一个数据集的学习方法。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(Auto-Encoder, AE)、K-均值聚类(K-Means)、高斯混合模型(Gaussian Mixture Model, GMM)和受限玻尔兹曼机(Restricted Boltzmann Machine, RBM)<sup>[4]</sup>。这些方法都倾向于学习一个近似局部滤波器字典<sup>[4]</sup>,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。

后来,文献 [5] 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(Deep Embedding Clustering, DEC)。该算法以自学习的方式定义了一个有效的目标函数。定义的聚类损失函数用于同时更新网络和聚类中心的变换参数。但是,深度嵌入聚类算法中使用最广泛的神经网络是堆栈自编码器。堆栈自编码器需要逐层预训练,然后以端到端(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,堆栈自编码器是建立在全连接的层次上,对于处理局部信息是无效的。

本章安排如下:5.2 节对卷积自编码网络进行了介绍,5.3 节构建了本章利用的深度嵌入卷积聚类器,通过利用卷积自编码器对输入地海杂波进行重构,然后利用本章提出的聚类损失函数进行训练,最终获得聚类结果,5.4 节利用实际数据对于本章提出的算法性能进行了分析讨论,5.5 节进行本章总结。

## 1.2 卷积自编码网络

### 1.2.1 自编码器

自编码器是一种利用隐层用来重构输入的神经网络。其模型如图 1-1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。对

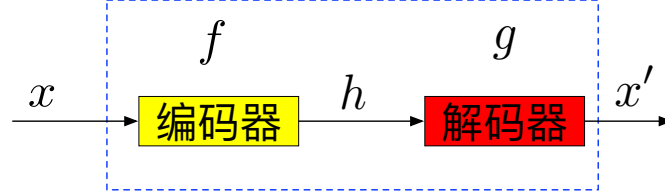


图 1-1 自编码器模型

于一组具有  $N$  个输入向量的给定输入数据集，编码器将第  $i$  个输入信号  $x^{(i)} \in \mathbb{R}^{d_x}$  编码到某个隐层的表示  $h^{(j)} \in \mathbb{R}^{d_h}$ ， $d_x$  和  $d_h$  分别为输入表示和隐层表示的维数，然后解码器再将  $h^{(j)}$  解码为输出  $x'^{(i)} \in \mathbb{R}^{d_x}$ 。通过最小化  $x^{(i)}$  与  $x'^{(i)}$  之间的差距，可以训练得到一个特征的映射  $\varphi$ ，其可以用来重构输入。

编码和解码公式如下：

$$h^{(j)} = f(b_j + \sum_{i=1}^{d_x} w_{ij} x_i^{(i)}) \quad (1-1)$$

$$x'^{(i)} = f(b'_i + \sum_{j=1}^{d_h} w'_{ij} h^{(j)}) \quad (1-2)$$

其中， $f$  是激活函数函数， $W = [w_{ij}]$  和  $W' = [w'_{ij}]$  分别为编码器和解码器的权重矩阵， $b = [b_j]$  和  $b' = [b'_i]$  分别为编码器和解码器的偏置向量。

在此基础上，国内外广大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 [6]、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 [7] 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 [8]。

### 1.2.2 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[9] 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增加了一种重要的思路方法。堆栈自编码器的基本模型可以参看图 1-2，其通过逐层非监督学习的预训练，也即逐层初始

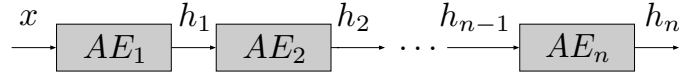


图 1-2 堆栈自编码器模型

化 (Layer-wise Pre-training)，来初始化深度网络的参数，替代传统的随机初始化的方法。预训练完毕后，利用训练参数，再进行监督学习训练。

全连接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，在视觉和对象识别领域最成功的模型 [10] 均主要利用了在整个输入空间发生重复的局部特征。基于此，文献 [11] 提出了一种卷积自编码器 (Convolutional Auto-Encoders, CAE)。卷积自编码器不同于传统的自编码器，它们的权重在输入向量中的所有位置之间共享，从而可以保持空间局部性。因此重构结果是隐藏局部特征的线性组合。从结构的角度看，CAE 类似于 SAE，但是将 SAE 中的全连接神经网络变为了卷积神经网络。参考第二章公式??，对于输入  $x$ ，第  $k$  个特征映射的隐藏表示由下式给出

$$h^k = f(x * W^k + b^k) \quad (1-3)$$

其中偏置  $b^k$  也是全局共享的， $f$  是激活函数， $*$  表示卷积操作。因为我们希望每个滤波器专注于整个输入空间的特征，另一方面如果每个数据点一个偏置将引入太多的自由度，所以对每一个隐藏层使用相同的偏置。公式 1-3 生成的激活图是输入  $x$  在低维空间中的编码，由于我们的目标是从所生成的特征映射重构输入  $x$ ，因此我们需要能够执行此操作的解码操作。卷积自编码器是完全卷积网络，因此解码操作也是一个卷积操作，一般称为反卷积操作 (Deconvolution) 或者转置卷积 (Transposed Convolution)。

另一方面，卷积减少了输出的空间范围，因此不可能使用卷积来重构具有相同空间范围的输入空间。我们可以使用输入填充来解决这个问题。如果用零填充输入  $x$ ，则第一卷积的结果可以具有比  $x$  更大的空间范围，因此第二卷积可以产生具有  $x$  的原始空间范围的体积。

因此，我们想要填充输入的零的数量是这样的：

$$\dim(x) = \dim(\text{decode}(\text{encode}(x))) \quad (1-4)$$

激活特征图  $h^k$  被用来作为解码器的输入，为了使得解码器的输出和原始的输入数据具有相同的结构，解码器反卷积的超参数由编码器的结构确定，因此，解码器的重构

过程由下式计算

$$x'^k = f(\sum_{k \in H} h^k * \tilde{W}^k + b'^k) \quad (1-5)$$

同样的，对于每一个输入通道只有一个偏置  $c$ ， $H$  表示潜在特征图的组。 $\tilde{W}$  表示权重的两个维度上的翻转操作。

正如第二章介绍，一般的分层网络特别是卷积神经网络，通常引入池化层层来获得平移不变性。因此本章的卷积自编码器在编码器阶段添加了池化层，相应的在解码器阶段需要添加上采样来进行处理，示意图如图 1-3 所示。

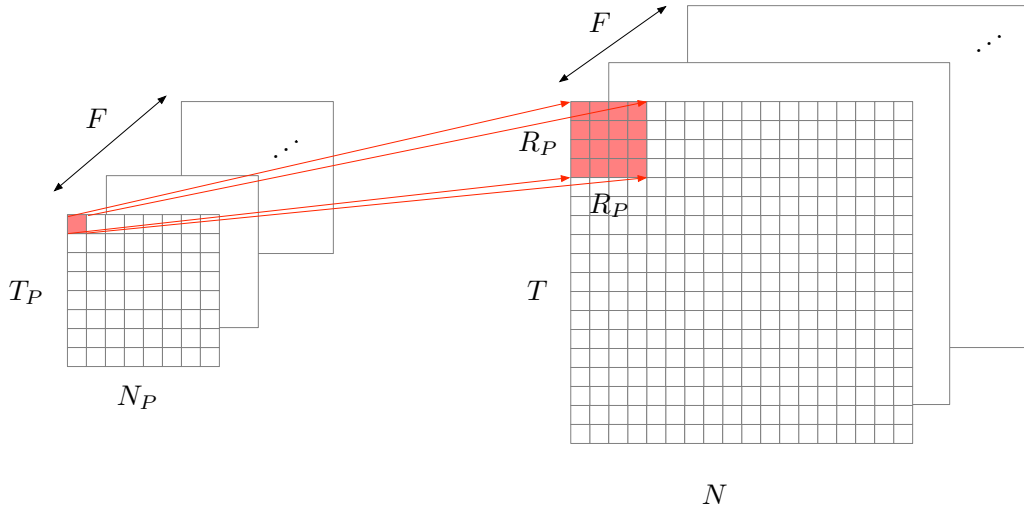


图 1-3 上采样示意图

设每个训练向量为  $x^{(i)}$  可以被映射到一个隐藏层表示  $h^{(i)}$ ，其重构表示为  $x'^{(i)}$ 。模型参数  $\Theta = W, b, b'$  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):

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \|x^{(i)} - x'^{(i)}\|_2^2 \quad (1-6)$$

$$\theta^* = \arg \min_{\theta} \mathcal{L} \quad (1-7)$$

一般情况下，可以使用随机梯度下降 (SGD) 等基于梯度的优化算法来对公式 1-7 求解。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，则可以得到下式：

$$\frac{\partial \mathcal{L}}{\partial W^k} = x * \Delta h^k + \tilde{h}^k * \Delta x' \quad (1-8)$$

$\Delta h$  和  $\Delta x'$  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。

本节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。相比于传统自编码器，其更有利于学习数据的局部结构，避免特征空间的失真。虽然卷积自编码器的过完备隐藏表示使得学习比标准自编码器更难，但是可以通过添加池化层加强其稀疏性而不需要通过反复试验来设置正则化参数。

### 1.3 深度嵌入卷积聚类方法

深层嵌入式聚类 (Deep Embedding Clustering, DEC) [5] 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。

#### 1.3.1 深度嵌入卷积聚类方法结构

首先对于问题进行描述，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  $d_z \ll d_x$ 。然后，利用 K-均值聚类算法，将子空间  $Z$  进行聚类，得到初始聚类中心  $\mu_k$ 。

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 1-4 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，故解码器为反卷积。在编码器与解码器中间添加嵌入层（图 1-4 中绿色标记的全连接层），该层的结果被用于进行聚类操作。DECC 的结构如图 1-5 所示。

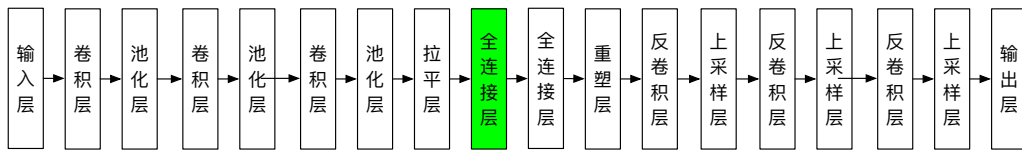


图 1-4 卷积自编码器结构图

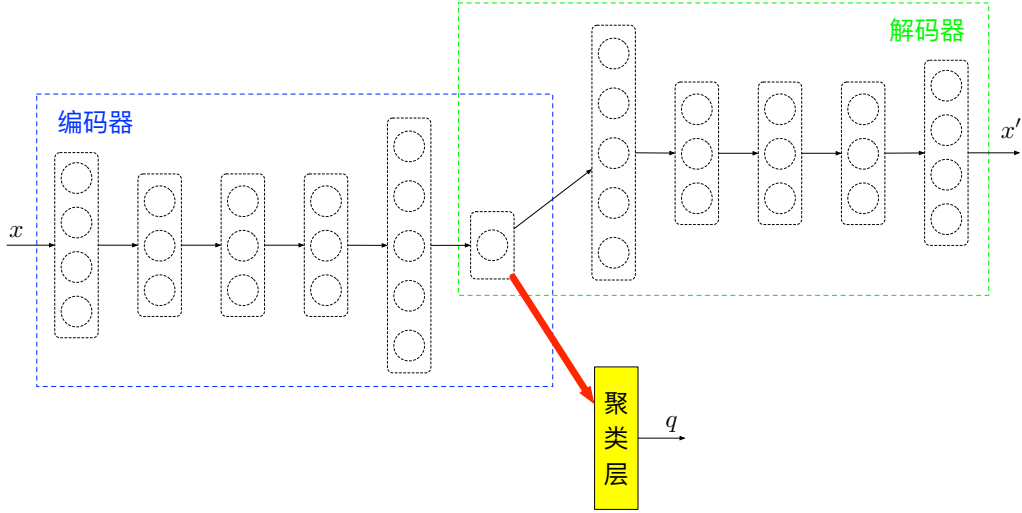


图 1-5 深度嵌入卷积自编码器结构图

### 1.3.2 KL 散度聚类方法

文献 [12] 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：

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

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。

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

根据文献 [5]，目标分布  $Q$  的定义如下：

$$q_{ik} = \frac{p_{ik}^2/f_k}{\sum_k (p_{ik}^2/f_k)} \quad (1-11)$$

其中  $f_k$  的定义为

$$f_k = \sum_i q_{ik} \quad (1-12)$$

公式 1-7 给出了自编码器重构的损失函数，故最终的损失函数为：

$$\mathcal{L} = \mathcal{L}_{mse} + \beta \mathcal{L}_{kld} \quad (1-13)$$

其中， $\beta > 0$  为调节两部分损失函数的权重。

### 1.3.3 学习算法

对于公式 1-13 中的损失函数，我们利用随机梯度下降和反向传播进行学习。根据上一节的描述可知，总共有三类参数需要学习：卷积自编码器的权重、聚类中心以及目标分布  $Q$ 。

#### 更新卷积自编码器的权重和聚类中心

对于一个给定的目标分布  $Q$ ，那么聚类损失  $\mathcal{L}_{kld}$  相对于嵌入点  $z_i$  和聚类中心点  $\mu_k$  的梯度计算公式如下：

$$\frac{\partial \mathcal{L}_{kld}}{\partial z_i} = 2 \sum_{k=1}^K (1 + \|z_i - \mu_k\|^2)^{-1} (q_{ik} - p_{ik}) (z_i - \mu_k) \quad (1-14)$$

$$\frac{\partial \mathcal{L}_{kld}}{\partial \mu_k} = 2 \sum_{i=1}^N (1 + \|z_i - \mu_k\|^2)^{-1} (p_{ik} - q_{ik}) (z_i - \mu_k) \quad (1-15)$$

$$(1-16)$$

那么对于一个小的批次  $m$  和学习率  $\lambda$ ， $\mu_k$  的更新公式为：

$$\mu_k = \mu_k - \frac{\lambda}{m} \sum_{i=1}^m \frac{\partial \mathcal{L}_{kld}}{\partial \mu_k} \quad (1-17)$$

编码器的权重更新公式为：

$$W_{de} = W_{de} - \frac{\lambda}{m} \sum_{i=1}^m \frac{\partial \mathcal{L}_{mse}}{\partial W_{de}} \quad (1-18)$$

解码器的权重更新公式为：

$$W_{en} = W_{en} - \frac{\lambda}{m} \sum_{i=1}^m \left( \frac{\partial \mathcal{L}_{mse}}{\partial W_{en}} + \beta \frac{\partial \mathcal{L}_{kld}}{\partial W_{en}} \right) \quad (1-19)$$

#### 更新目标分布

根据公式 1-11 可知，目标分布  $Q$  依赖于先验的软指派  $P$ 。因此，为了提升结果的鲁棒性，本章利用所有的嵌入点每隔  $T$  次迭代对目标分布进行更新，其更新主要利用公式 1-9 和 1-11。并且每次完成更新后，重新计算样本  $x_i$  的指派：

$$s_i = \arg \max_k p_{ik} \quad (1-20)$$

当相邻两次更新的指派变化小于一个阈值  $\delta$  时，停止迭代。整个算法流程图如下表。

**算法 1** DECC 算法学习过程

**输入:** 训练样本  $X$ , 聚类结果个数  $K$ , 最大迭代次数  $iter_{max}$ , 目标分布更新间隔  $T$ , 初始目标参数  $\theta$ , 学习率  $\eta$

**输出:** 卷积自编码器权重  $W_{en}$  和  $W_{de}$ , 聚类中心  $\mu$ , 指派标签  $s$

```

1: 初始化迭代次数  $iter = 1$ 
2: 只利用重构损失函数对卷积自编码器进行训练, 获得初始权重  $W_{en}$  和  $W_{de}$ 
3: 利用 K-均值聚类算法获得初始聚类中心  $\mu$ 
4: while  $iter < iter_{max}$  do
5:   if  $iter \bmod T = 0$  then
6:     计算所有的所有的嵌入点  $z_i = \varphi_W(x_i), i = 1, \dots, N$ 
7:     将公式 1-9代入公式 1-11更新目标分布  $Q$ 
8:     保存上一次指派  $s_{last} = s$ 
9:     计算新的指派  $s_i = \arg \max_k p_{ik}, i = 1, \dots, N$ 
10:    if  $sum(s_{last} \neq s)/N < \delta$  then
11:      停止学习
12:    end if
13:  end if
14:  从训练集中获取  $m$  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。
15:  更新  $\mu_k = \mu_k - \frac{\lambda}{m} \sum_{i=1}^m \frac{\partial \mathcal{L}_{kld}}{\partial \mu_k}$ 
16:  更新  $W_{de} = W_{de} - \frac{\lambda}{m} \sum_{i=1}^m \frac{\partial \mathcal{L}_{mse}}{\partial W_{de}}$ 
17:  更新  $W_{en} = W_{en} - \frac{\lambda}{m} \sum_{i=1}^m (\frac{\partial \mathcal{L}_{mse}}{\partial W_{en}} + \beta \frac{\partial \mathcal{L}_{kld}}{\partial W_{en}})$ 
18: end while

```

**1.4 仿真实验验证**

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  $\alpha = 1$ 。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。本章利用第三章中组 C 的训练数据用于本章的实验, 考虑到地海杂波在时间与空间上的连续性, 本章选取了相邻 3 帧, 相邻 9 个单元供 27 个单元的杂波信号将其进行组合, 并进行随机重排复制得到一个  $128 \times 128$  的二维数据, 其示意图参看图 1-6, 总共



有 47549 组训练样本。

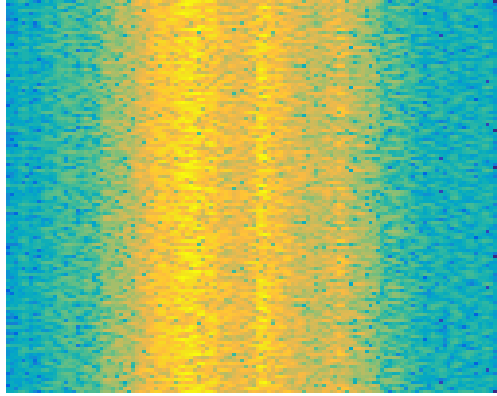


图 1-6 输入样本示意图

其实验结果如表 1-1所示。

表 1-1 无监督聚类精度对比表

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:

$$ACC = \max_m \frac{\sum_{i=1}^N g(l_i, m(c_i))}{n} \quad (1-21)$$

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

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。

## 1.5 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, 其通过利用卷积自编码器进行提取并在损失函数中将 KL 散度与重构损失两部分结合, 有效提高了聚类的精度。虽然与有监督学习算法相比, 本章的算法精度仍需进一步提高, 但是无监督学习的特性使得其避免了繁琐的标定过程。本章的主要贡献如下:

- 本章设计了可以以端到端方式训练的卷积自编码器来从未标记的数据中学习特征。通过在地海杂波数据中包含空间、时间信息，通过添加卷积层、卷积转置层和全连接层，设计了好于传统堆栈自编码器的卷积自编码器。
- 在根据聚类损失函数调整网络参数时考虑局部结构特征。实验结果证明，利用局部结构特征有助于稳定训练过程。
- 本章提出了深度嵌入卷积聚类算法来自动聚类地海杂波，DECC 利用了卷积自编码器提取局部特征的优势。
- 利用天波超视距雷达的实际地海杂波数据进行了实验，结果验证了深度嵌入聚类算法的有效性。

## 参考文献

- [1] Shinnou H, Sasaki M. Spectral Clustering for a Large Data Set by Reducing the Similarity Matrix Size.[C]. In LREC. 2008.
- [2] Roth V, Lange T. Feature selection in clustering problems[C]. In Advances in neural information processing systems. 2004:473–480.
- [3] Tian F et al. Learning Deep Representations for Graph Clustering.[C]. In AAAI. 2014:1293–1299.
- [4] 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.
- [5] Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis[C]. In International Conference on Machine Learning. 2016:478–487.
- [6] 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.
- [7] Kingma DP, Welling M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114. 2013.
- [8] Zhou G, Sohn K, Lee H. Online incremental feature learning with denoising autoencoders[C]. In Artificial Intelligence and Statistics. 2012:1453–1461.
- [9] Bengio Y et al. Greedy layer-wise training of deep networks[C]. In Advances in neural information processing systems. 2007:153–160.
- [10] 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.
- [11] Masci J et al. Stacked convolutional auto-encoders for hierarchical feature extraction[J]. Artificial Neural Networks and Machine Learning–ICANN 2011. 2011, :52–59.

- [12] Maaten Lvd, Hinton G. Visualizing data using t-SNE[J]. Journal of Machine Learning Research. 2008, 9(Nov):2579–2605.