

# Classification of Time-Series Images Using Deep Convolutional Neural Networks

中文解释: <https://blog.csdn.net/luolan9611/article/details/81018235>

论文下载链接: <https://arxiv.org/abs/1710.00886>

论文的源码下载链接: <https://sites.google.com/site/nimahatami/projects>

大多数时间序列分类 (TSC) 文献都针对于一维信号, 但论文使用递归图RP将时间序列转换为二维纹理图像, 然后利用深度CNN分类器识别。论文将序列分类问题转换为了图像识别分类问题, 充分利用了目前图像识别领域发展的优势, 针对UCR时间序列分类档案馆 (UCR time-series classification archive) 的数据集进行实验, 证明了所提出方法与现有的深层架构和最先进的TSC算法相比都有极大的准确性。

## Methodology

### 将时间序列编码成RP递归图

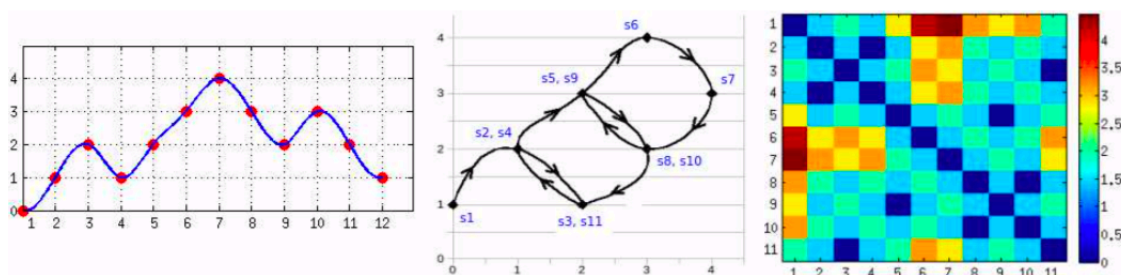


Figure 1: From time-series signal to recurrence plot. Left: A simple example of time-series signal ( $x$ ) with 12 data points. Middle: The 2D phase space trajectory is constructed from  $x$  by the time delay embedding ( $\tau = 1$ ). States in the phase space are shown with bold dots:  $s_1 : (x_1, x_2), s_2 : (x_2, x_3), \dots, s_{11} : (x_{11}, x_{12})$ . Right: The recurrence plot  $R$  is a  $11 \times 11$  square matrix with  $R_{i,j} = \text{dist}(s_i, s_j)$ .

三张小图中最左边的是包含12个数据点的时序信号示意图记为(1), 中间的是根据时序信号画的二维相空间轨迹示意图记为(2), 最右边的是递归图记为(3), 该递归图其实是个11\*11大小的矩阵。

从(1)到(2): (1)上有12个红点 $x$ , 点到点为一个状态 $s$ , 所以有11个状态。状态 $s_n:(x_n, x_{n+1})$ 。也就是说状态 $s_n$ 的位置是(第 $x$ 个数据点的纵坐标值, 第 $x+1$ 个数据点的纵坐标值)。

$$R_{i,j} = \theta(\epsilon - \|\vec{s}_i - \vec{s}_j\|), \quad \vec{s}(\cdot) \in \mathcal{R}^m, \quad i, j = 1, \dots, K$$

阶跃函数      阈值      欧几里德范数       $m$ 维欧几里德空间      状态 $s$ 的个数

从(2)到(3)：根据上图 $R_{i,j}$ 的那个公式计算R矩阵，用matlab。然后前面有一个像e的符号，那个代表的是阈值，用于将彩色的rp图进行阈值化的。 $R_{i,j}$ 是一个阶跃函数。K是状态s的个数。

## 使用深度CNN进行分类识别

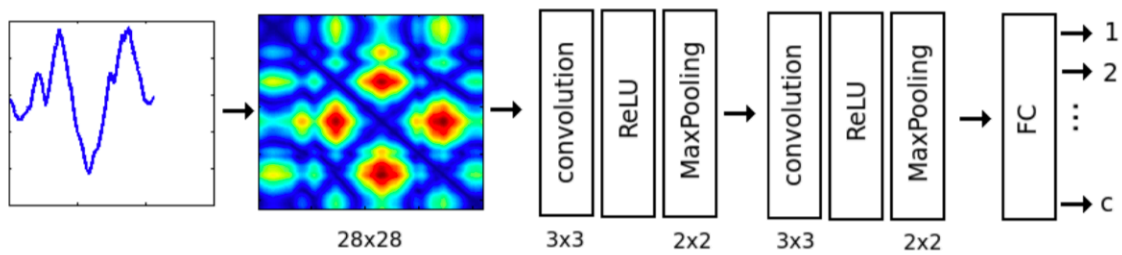


Figure 2: The proposed 2-stages CNN architecture for TSC. RP images are resized to 28x28, 56x56 or 64x64 (depend on the data) and fed into CNN model. This architecture  $32(5)-2-32(5)-2-125-c$  consists of 2 convolution, 2 pooling, and 2 fully-connected layers.

该深度网络的架构如上图所示。有两个卷积层，每个卷积层后的跟着一个ReLU修正线性单元，有两个池化层使用2\*2最大池化方法，有1个全连接层。其中C1(size)-S1-C2(size)-S2-H-O，C1和C2是两个卷积层filter的个数，size是kernel大小，H是隐藏层神经元的个数，c表示输出神经元的个数，其实就是要分多少类。

输入的图像大小是有要求的，28\*28，56\*56，64\*64。

## Experiment

UCR时序档案馆链接：[http://www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/)

作者从几十组数据中挑选了20组数据集，和现今的一些时序分类算法做了对比实验。

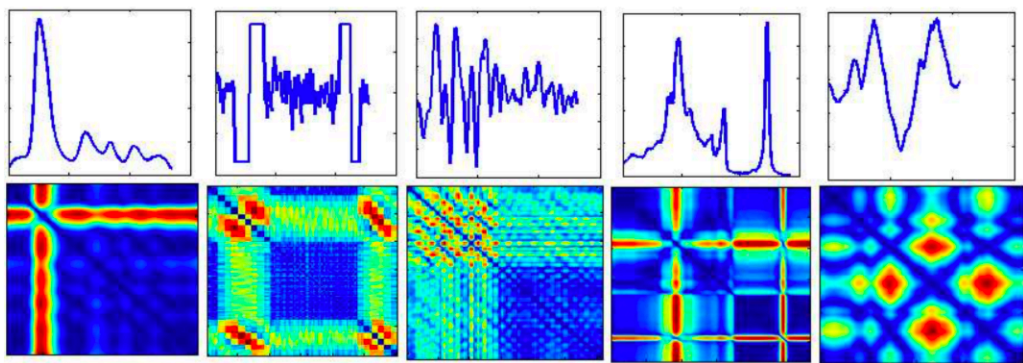


Figure 3: Application of RP ( $m = 3, \tau = 4$ ) time-series to image encoding on five different datasets from the UCR archive: 50words, TwoPatterns, FaceAll, OliveOil and Yoga data (from left to right, respectively).

- 50words

是由T.Rath从乔治华盛顿图书馆采集的含有50个类的字轮廓数据集，并在文章*Word image matching using dynamic time warping*中使用，每个序列都是一个单词，通过获取单词的高度轮廓来形成序列。

- TwoPatterns

由P.Geurts为2002年对决策树归纳的博士论文生成的模拟数据集，含有4个类。

- FaceAll

含有14名研究生的面孔数据集。每一个面部轮廓被映射到一维系列。

- OliveOil

该数据集中含有4个类，每一类都是来自其他国家的特级初榨橄榄油。多变量分析可以区分特级初榨橄榄油的地理来源。

- Yoga data

该数据集含有两个类，通过捕获两个在绿色屏幕前对瑜伽姿势之间的转换获得的。问题是区分一个男演员和女演员。通过找到轮廓并且测量轮廓导致中心的距离，将每个图像转换为一维时间序列。

## 把序列数据集制作成RP图（使用MatLab）

论文研究了基于CNN模型运用RP图进行时间序列分类的性能，递归图提供了一种通过相空间可视化轨迹的周期性特性的方法，并使我们能够通过2D表示研究m维相空间轨迹的某些方面。由于CNN近来在图像识别方面的出色成果，我们首先将时间序列信号编码为2D图，然后将TSC问题视为纹理识别任务。使用的CNN模型具有2个隐藏层，然后是一个完全连接的层。

## 用深度CNN训练分类

实验环境：Python(using Keras) on a PC with 2.4GHzX32 CPU and 32GB memory.

输入大小有要求：28\*28,56\*56,64\*64

2\*2最大池化，Dropout=0.25

全连接层包含128个隐藏的神经元和c个输出神经元，Dropout=0.5

### Architecture

这是个2级深度CNN模型，其大小为28×28的1通道输入和带有c神经元的输出层。每个特征学习阶段代表一个不同的特征级别，分别由卷积（过滤器），激活函数和池化组成。在两阶段特征提取结束时，将特征图展平并喂到全连接层中进行分类。

### Learning

训练论文提出的CNN架构类似于MLP。利用基于梯度的优化方法（误差反向传播算法）来估计模型的参数。为了加快收敛速度，可使用随机梯度下降（SGD）来更新参数。训练阶段有两个主要步骤：传播和权重更新。每个传播都涉及前馈和错误反向传播。前者通过逐层传递直到到达输出来确定输入矢量上的特征图。后一种方法根据预测输出的损失函数计算传播误差。

## Result

作者是挑选了20组数据和其它的方法做的实验。表头中c指的是分类类别，Ntr是训练样本数，Nte是测试样本数，l指的是序列长度。后面的都是别的方法名称。表中展示的是错误率。针对每种数据集，错误率最低的被加粗显示。作者的方法在10种数据集上都取得了第一名，然后计算了一个平均排位。作者使用的方法的平均排位是最高的。

Dataset	$c$	$N_{tr}$	$N_{te}$	$l$	1-NN DTW	Shapelet	BoP	SAX-VSM	TFRP	MCNN	GAF-MTF	ours
50words	50	450	455	270	0.31	0.44	0.46	-	0.43	<b>0.19</b>	0.30	0.26
Adiac	37	390	391	176	0.39	0.51	0.43	0.38	<b>0.20</b>	0.23	0.37	0.28
Beef	5	30	30	470	0.36	0.44	0.43	<b>0.033</b>	0.36	0.36	0.23	0.08
CBF	3	30	900	128	0.003	0.05	0.01	0.02	-	<b>0.002</b>	0.009	0.005
Coffee	2	28	28	286	<b>0</b>	0.06	0.03	<b>0</b>	0.03	0.036	<b>0</b>	<b>0</b>
ECG200	2	100	100	96	0.23	0.22	0.14	0.14	0.17	-	0.09	<b>0</b>
FaceAll	14	560	1690	131	<b>0.19</b>	0.40	0.21	0.20	0.29	0.23	0.23	<b>0.19</b>
Face4	4	24	88	350	0.17	0.09	0.023	<b>0</b>	0.21	<b>0</b>	0.06	<b>0</b>
Fish	7	175	175	463	0.17	0.19	0.074	<b>0.017</b>	0.12	0.05	0.114	0.085
GunPoint	2	50	150	150	0.093	0.061	0.027	0.007	0.02	<b>0</b>	0.08	<b>0</b>
Lightning2	2	60	61	637	0.13	0.29	0.16	0.19	0.04	0.16	0.11	<b>0</b>
Lightning7	7	70	73	319	0.27	0.40	0.46	0.30	0.31	<b>0.21</b>	0.26	0.26
OliveOil	4	30	30	570	0.16	0.21	0.13	<b>0.10</b>	0.13	0.13	0.2	0.11
OSULeaf	6	200	242	427	0.40	0.35	0.23	0.107	<b>0.07</b>	0.27	0.35	0.29
SwedishLeaf	15	500	625	128	0.20	0.27	0.19	0.25	<b>0.04</b>	0.066	0.06	0.06
SyntControl	6	300	300	60	0.007	0.08	0.03	0.25	-	0.003	0.007	<b>0</b>
Trace	4	100	100	275	<b>0</b>	0.002	<b>0</b>	<b>0</b>	-	<b>0</b>	<b>0</b>	<b>0</b>
TwoPattern	4	1000	4000	128	<b>0</b>	0.11	0.12	0.004	-	0.002	0.09	0.17
Wafer	2	1000	6174	152	0.02	0.004	0.003	0.0006	0.002	0.002	<b>0</b>	<b>0</b>
Yoga	2	300	3000	426	0.16	0.24	0.17	0.16	0.14	0.11	0.19	<b>0</b>
# wins	-	-	-	-	4	0	1	6	3	6	3	<b>10</b>
Ave.Rank	-	-	-	-	4.05	5.75	4.15	3.0	3.37	2.36	3.40	<b>2.15</b>

## Future Work

提出了一种新的TSC方法。利用CNN在图像分类方面的高性能，时间序列信号首先被转换成纹理图像(使用RP)，然后由深度CNN模型处理。这方法有以下优点：（1）RP使我们能够通过二维图像可视化m维相位空间轨迹的某些方面，以及（2）CNN以监督的方式自动联合学习不同级别的时间序列特征和分类。实验结果证明了这个方法的优越性。特别是，与使用RP的模型与传统分类框架(例如，SVM)和其他基于CNN的时间序列图像分类(例如，使用CNN的GAF、MTF图像表明)，在我们提出的模型中使用带有CNN的RP图像可以获得更好的结果。作为未来的工作，对于更困难的TSC任务需要训练更深层次的体系结构，这意味着需要更多可用的数据样本，因此，我们可以研究一种采用小样本量的时间序列分类。