# 深度迁移学习在故障诊断中的应用

## 1. 简介

该存储库包含通过 PyTorch 实现的用于跨负载故障诊断传输任务的流行深度传输学习算法，包括：

一般监督学习分类任务：训练和测试采用相同的机器、工况和故障。

 域自适应：源域数据的分布可能与目标域数据不同，但目标域的标签集与源域相同，即德𝑠=（𝑋𝑠，是𝑠），德吨=（𝑋吨，是吨），𝑋𝑠≠𝑋吨，是𝑠=是吨。

* +  **DDC**：深度域混淆[[arXiv 2014]](https://arxiv.org/pdf/1412.3474.pdf)
  +  **Deep CORAL**：深度域适应的相关对齐[[ECCV 2016]](https://arxiv.org/abs/1607.01719)
  +  **DANN**：通过反向传播实现无监督领域自适应[[ICML 2015]](http://proceedings.mlr.press/v37/ganin15.pdf)
  + 去做

 开放集域自适应：源域数据的分布可能与目标域数据不同，而且目标标签集包含未知类别，即德𝑠=（𝑋𝑠，是𝑠），德吨=（𝑋吨，是吨），𝑋𝑠≠𝑋吨，是𝑠∈是吨。我们参考它们的共同类别是𝑠∩是吨作为已知类，并且是𝑠∖是吨（或者是吨∖是𝑠) 在目标域中作为未知类别。

* +  **OSDABP**：通过反向传播进行开放集域自适应[[ECCV 2018]](http://openaccess.thecvf.com/content_ECCV_2018/papers/Kuniaki_Saito_Adversarial_Open_Set_ECCV_2018_paper.pdf)
  + 去做

**基于小样本**学习的轴承故障诊断方法请参见：<https://github.com/Xiaohan-Chen/few-shot-fault-diagnosis>

## 🎈 2. 引用

有关轴承故障诊断中迁移学习的进一步介绍，请阅读我们的[论文](https://ieeexplore.ieee.org/document/10042467)。如果您发现此存储库有用并在您的工作中使用，请引用我们的论文，谢谢〜：

@ARTICLE{10042467,

author={Chen, Xiaohan and Yang, Rui and Xue, Yihao and Huang, Mengjie and Ferrero, Roberto and Wang, Zidong},

journal={IEEE Transactions on Instrumentation and Measurement},

title={Deep Transfer Learning for Bearing Fault Diagnosis: A Systematic Review Since 2016},

year={2023},

volume={72},

number={},

pages={1-21},

doi={10.1109/TIM.2023.3244237}}

## 🔧 3. 要求

* 蟒蛇3.9.12
* Numpy 1.23.1
* pytorch 1.12.0
* scikit-learn 1.1.1
* torchvision 0.13.0

## 👜 4.数据集

[从CWRU 方位数据集中心](https://engineering.case.edu/bearingdatacenter/48k-drive-end-bearing-fault-data)下载方位数据集，并按照以下结构将.mat文件放入文件夹中：./datasets

datasets/

└── CWRU/

├── Drive\_end\_0/

│ └── 97.mat 109.mat 122.mat 135.mat 174.mat 189.mat 201.mat 213.mat 226.mat 238.mat

├── Drive\_end\_1/

│ └── 98.mat 110.mat 123.mat 136.mat 175.mat 190.mat 202.mat 214.mat 227.mat 239.mat

├── Drive\_end\_2/

│ └── 99.mat 111.mat 124.mat 137.mat 176.mat 191.mat 203.mat 215.mat 228.mat 240.mat

└── Drive\_end\_3/

└── 100.mat 112.mat 125.mat 138.mat 177.mat 192.mat 204.mat 217.mat 229.mat 241.mat

## 📝 5. 使用方法

**注意**：在迁移学习任务中使用预训练模型初始化主干和分类器时，请先运行分类任务以生成相应的检查点。

该存储库实现了四种典型的神经网络，包括 MLP、1D CNN、1D ResNet18 和 2D ResNet18（torchvision 包）。更多详细信息可在文件夹中找到./Backbone。

**一般监督学习分类：**

* 在相同的机器、工作条件和故障上训练和测试模型。使用以下命令：

python3 classification.py --datadir './datasets' --max\_epoch 100

**迁移学习：**

* 如果使用DDC迁移学习方法，请使用以下命令：

python3 DDC.py --datadir './datasets' -backbone "CNN1D" --pretrained False --kernel 'Linear'

* 如果使用DeepCORAL迁移学习方法，请使用以下命令：

python3 DDC.py --datadir './datasets' -backbone "CNN1D" --pretrained False --kernel 'CORAL'

* 如果使用DANN迁移学习方法，请使用以下命令：

python3 DANN.py --backbone "CNN1D"

**开放集域适应：**

* 目标域包含未知类，请使用以下命令：

python3 OSDABP.py

## 🔦6.结果

以下结果并不代表最佳结果。

**一般分类任务：**  
数据集：CWRU  
负载：3  
标签集：[0,1,2,3,4,5,6,7,8,9]

|  | **多层感知网络** | **CNN1D** | **ResNet1D** | **ResNet2D** |
| --- | --- | --- | --- | --- |
| acc（时间域） | 93.95 | 97.70 | 99.58 | 98.02 |
| acc（频域） | 99.95 | 99.44 | 100.0 | 99.96 |

**迁移学习：**  
数据集：CWRU  
源负载：3  
目标负载：2  
标签集：[0,1,2,3,4,5,6,7,8,9]  
预训练模型：True

时间域：

|  | **多层感知网络** | **CNN1D** | **ResNet1D** | **ResNet2D** |
| --- | --- | --- | --- | --- |
| DDC（线性核） | 75.47 | 85.53 | 91.79 | 91.32 |
| 深海珊瑚 | 82.33 | 88.23 | 93.88 | 90.84 |
| 丹恩 | 87.68 | 94.77 | 98.88 | 93.95 |

频域

|  | **多层感知网络** | **CNN1D** | **ResNet1D** | **ResNet2D** |
| --- | --- | --- | --- | --- |
| 深海珊瑚 | 98.65 | 98.22 | 99.75 | 99.31 |
| 丹恩 | 99.38 | 98.74 | 99.89 | 99.47 |

**开放集域适应**

* OSDABP 数据集：CWRU  
  源负载：3  
  目标负载：2  
  源标签集：[0,1,2,3,4,5]  
  目标标签集：[0,1,2,3,4,5,6,7,8,9]  
  预训练模型：True

| **标签** | **0** | **1** | **2** | **3** | **4** | **5** | **不克** | **全部** | **仅知道** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 多层感知网络 | 99.83 | 95.96 | 59.76 | 76.10 | 19.85 | 96.58 | 59.21 | 70.21 | 75.99 |
| CNN1D | 100.0 | 94.95 | 94.47 | 99.08 | 47.31 | 74.32 | 26.36 | 61.75 | 85.35 |
| ResNet1D | 100.0 | 100.0 | 80.14 | 100.0 | 43.32 | 93.49 | 45.22 | 70.04 | 86.58 |
| ResNet2D | 100.0 | 100.0 | 94.82 | 100.0 | 18.55 | 98.12 | 53.42 | 72.95 | 85.96 |

## 🏕️7.另请参阅

* 多尺度 CNN 和 LSTM 轴承故障诊断 [[论文](https://link.springer.com/article/10.1007/s10845-020-01600-2)][ [GitHub](https://github.com/Xiaohan-Chen/baer_fault_diagnosis) ]
* TFPred 自监督学习用于少量标记故障诊断 [[论文](https://www.sciencedirect.com/science/article/pii/S0967066124000601)][ [GitHub](https://github.com/Xiaohan-Chen/TFPred) ]

## 🌐 8. 致谢

@article{zhao2021applications,

title={Applications of Unsupervised Deep Transfer Learning to Intelligent Fault Diagnosis: A Survey and Comparative Study},

author={Zhibin Zhao and Qiyang Zhang and Xiaolei Yu and Chuang Sun and Shibin Wang and Ruqiang Yan and Xuefeng Chen},

journal={IEEE Transactions on Instrumentation and Measurement},

year={2021}

}

我要感谢以下人员对此存储库的贡献：[@Wang-Dongdong](https://github.com/Wang-Dongdong)