VIME: Self- and Semi-supervised Learning for Tabular Data

项目概述

本项目基于论文 VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain (NeurIPS 2020)，使用现代 PyTorch 2.x 重新实现了 VIME 框架。

在原有 VIME 框架基础上，新增了 DAE (Denoising Autoencoder) 基线用于对比：

方法	描述	训练任务
Supervised	纯监督学习基线	直接用标注数据训练分类器
DAE	去噪自编码器预训练	无损重构：clean → clean
VIME-Self	VIME 自监督预训练	去噪 + mask 预测：corrupted → clean + mask
VIME	VIME 半监督学习	VIME-Self + 伪标签一致性正则化

关键区别：- **DAE**：只用重构损失，无 mask 预测任务 - **VIME-Self**：重构损失 + mask 预测损失（需要学习识别哪些特征被打乱）

为了更好地理解 VIME 的工作机制，我创建了一个可控的合成数据集生成器：

生成过程：

```
潜变量 z (10维) ~ N(0,1)
    ↓
有用特征 x_informative = min-max(z @ W_x + noise)  [连续值, [0,1]]
    ↓
噪声特征 x_noise ~ U(0,1)  [与y无关的随机噪声]
    ↓
最终特征 x = [x_informative, x_noise]  [固定200维]
    ↓
标签 y (10类) = softmax(z @ W_y + noise)
```

关键特性：- 总特征维度固定为 200 - `n_noise_features` 控制噪声特征数量 - 有用特征数 = 200 - n_noise_features

```
## 实验结果

### 实验设置
- **标注样本数**: 500（所有实验统一）
- **迭代次数**: 10次（报告均值±标准差）
- **随机种子**: 42-51（每次迭代不同seed保证可重复性）
- **默认参数**: `p_m=0.3`, `alpha=2.0`, `K=3`, `beta=1.0`（MNIST）或`beta=0.1`（合成数据）

### 合成数据实验结果

**运行命令**:
```bash
# 无噪声特征（0%无关变量比例，200个有用特征）
python train.py --dataset synthetic --n_noise_features 0

# 50个噪声特征（25%无关变量比例，150个有用特征）
python train.py --dataset synthetic --n_noise_features 50

# 100个噪声特征（50%无关变量比例，100个有用特征）
python train.py --dataset synthetic --n_noise_features 100
```
```

无关变量比例	方法	准确率 (Mean ± Std)	vs Baseline
0%	Supervised	0.7199 ± 0.0251	-
	DAE	**0.7656 ± 0.0044**	**+6.3%**
	VIME-Self	0.7647 ± 0.0064	+6.2%
	VIME	0.7553 ± 0.0054	+4.9%
25%	Supervised	0.6921 ± 0.0112	-
	DAE	0.7039 ± 0.0108	+1.7%
	VIME-Self	0.7306 ± 0.0067	+5.6%
	VIME	**0.7582 ± 0.0038**	**+9.6%**
50%	Supervised	0.5514 ± 0.0142	-
	DAE	0.5741 ± 0.0100	+4.1%
	VIME-Self	0.6280 ± 0.0071	+13.9%
	VIME	**0.6793 ± 0.0060**	**+23.2%**

代码架构

```
VIME/
  train.py              # 主训练脚本
  data.py               # MNIST数据加载
  synthetic_data.py     # 合成数据生成器（新增）
  autoencoder.py        # Autoencoder实现（新增）
  self_supervised.py    # VIME-Self实现
  semi_supervised.py    # VIME半监督实现
  baselines.py          # 监督学习基线（MLP, Logit, XGBoost）
  utils.py             # 工具函数（mask生成、评估指标）
```

使用方法

安装依赖

```
python -m venv venv
source venv/bin/activate  # Windows: venv\Scripts\activate
pip install -r requirements.txt
```

运行实验

MNIST 数据集:

```
python train.py --dataset mnist --iterations 10 --label_no 500 --seed 42
```

合成数据（无噪声）:

```
python train.py --dataset synthetic --n_noise_features 0
```

合成数据（50 个噪声特征，25% 无关变量比例）:

```
python train.py --dataset synthetic --n_noise_features 50
```

合成数据（100 个噪声特征，50% 无关变量比例）:

```
python train.py --dataset synthetic --n_noise_features 100
```

主要参数

参数	说明	默认值
`--dataset`	数据集选择 (mnist/synthetic)	synthetic

参数	说明	默认值
--iterations	实验重复次数	10
--label_no	标注样本数量	1000
--p_m	Corruption 概率	0.3
--alpha	特征损失权重	2.0
--K	增强样本数量	3
--beta	半监督损失权重	1.0
--n_noise_features	噪声特征数（仅合成数据）	0
--seed	随机种子	42

技术细节

为了与原始 Keras 实现保持一致，使用相同的 RMSprop 参数：

```
optimizer = torch.optim.RMSprop(
    model.parameters(),
    lr=0.001,       # Keras default
    alpha=0.9,      # Keras default (rho)
    eps=1e-7        # Keras default
)
```

致谢

本项目基于以下论文和代码：

```
@inproceedings{yoon2020vime,
  title={VIME: Extending the Success of Self-and Semi-supervised Learning to Tabular Domain},
  author={Yoon, Jinsung and Zhang, Yao and Jordon, James and Van Der Schaar, Mihaela},
  booktitle={Advances in Neural Information Processing Systems},
  volume={33},
  year={2020}
}
```

License

本项目基于原始 VIME 代码改编，遵循相同的开源协议。