1 传统机器学习

1.1 贝叶斯网络手写数字识别

1.1.1 实验内容

如图，有一张显示手写的屏幕，屏幕上的像素有亮/暗(1/0)两种状态。当字迹覆盖某对应位置的像素时，该像素为亮的状态，否则为暗的状态。本次实验中，你需要使用包含各像素信息和最终数字结果的训练数据搭建一个贝叶斯网络，并根据测试数据的像素亮暗预测对应的数字，与测试数据的真实标签进行比较并计算准确率。

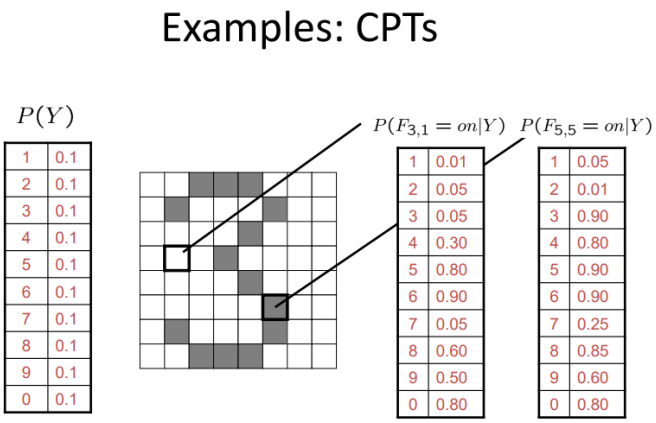


图 1: 示例贝叶斯网络

1.1.2 文件说明

`part_1/src/Bayesian-network.py` 为助教提供参考的代码框架，你可以在此基础上进行补全，或者自己从头实现。

文件尾部执行部分已经实现了数据的读取和像素和标签的分离。

本次实验使用的数据在 `data` 文件夹下，包含 `train.csv` 和 `test.csv` 两个格式一致的文件分别用于训练和测试。

文件的每一行为一份数据样本，包含 785 个数字，其中前 784 个数字为屏幕像素信息，每个像素值只可能为 0 或 1，最后一个数字为该数据对应的手写数字标签，取值范围是 0~9 的整数值。784 为 28×28 图片以从左到右、从上到下展平的结果。一份数据样本的例子：$\{0, 0, 0, 1, ..., 0, 3\}$ 表示一个数字 3 和它之前对应的像素值。

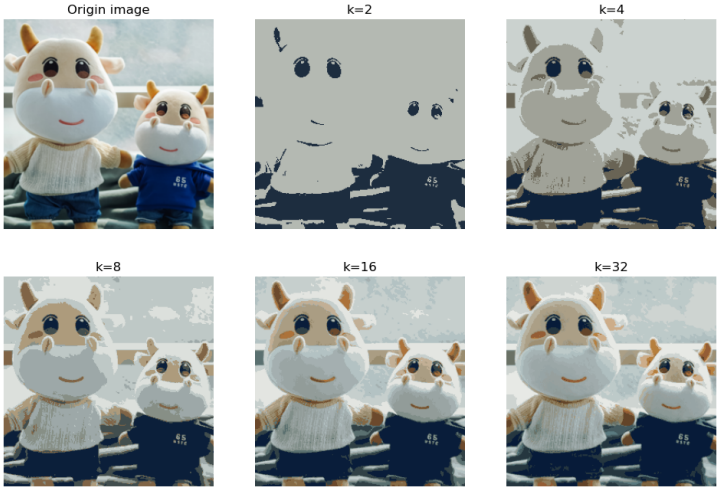
`train.csv` 包含 59999 行数据，`test.csv` 包含 9999 行数据。

1.2 利用 K-means 实现图片压缩

1.2.1 实验内容

本次实验，你需要实现KMeans算法，并使用KMeans算法来压缩一张图像。使用KMeans压缩图像的原理为，将图像上的每一个像素(R,G, B)视作一个数据点，并对所有点进行聚类。完成聚类之后，**将所有数据点的值替换成其聚类中心的值**，这样仅需保留几个聚类中心的数据点值以及其他点的类别索引，从而达到压缩图片的目的。

本次实验中你需要分别尝试 k=2,4,8,16,32 时的压缩结果。



1.2.2 文件说明

data 文件夹下的 ustc-cow.png 为本次实验使用到的图片，其大小为 256 × 256 像素，像素值为 0~255 的整数值。

KMeans.py 为助教提供参考的代码框架，你可以在此基础上进行补全，或者自己从头实现。

其中 KMeans 类中 initialize_centers() 为助教提供的初始化簇中心的函数，不建议修改。read_img() 读取对应路径的图片，返回值类似 numpy 的数组。如对助教提供的代码框架有所疑问，可以向助教询问。

1.3 实验要求

1. 在实验报告中，你需要配合相应的(伪)代码说明你实现两个实验的原理。
2. 对于贝叶斯手写数字识别实验，你需要在实验报告中贴出程序求得的准确率结果。
3. 对于 KMeans 图片压缩实验，你需要在实验报告中展示簇数分别取 2,4,8,16,32 时压缩后的图片。
4. 实验结束后，你需要提交两份程序的代码。
5. 鼓励使用numpy 的并行和索引技巧，禁止直接调用 sklearn 等机器学习库跳过贝叶斯网络和 KMeans 的原理实现。

2 深度学习

Transformer 主要用于自然语言处理(NLP)与计算机视觉(CV)领域。它也是 GPT 等生成式模型采用的架构。在本次实验中，我们将会实现一个简单的 Transformer.

2.1 实验内容

本次实验通过实现 transformer decoder 的架构，完成对莎士比亚文集的补全。实验的输入为 input.txt，其中是莎士比亚文集中的语句，需要模型根据这些语句根据 n-gram 作为语言模型进行自监督学习。n-gram 的具体形式如下：

$$P\left(x_i \mid x_{i-1}, x_{i-2}, ..., x_{i-(n-2)}, x_{i-(n-1)}\right)$$

其中待预测的词x_i，是由其前面的$n-1$个词 $x_{i-(n-1)}, x_{i-(n-2)}, ..., x_{i-1}$ 决定的。换言之，给定前 $n-1$ 个词，来预测下一个词是什么。

例子

给定文本：

```
To be, or not to be: that is the
```

让模型去预测下一个词 question

```
To be, or not to be: that is the question
```

在 transformer decoder 中，模型会根据当前的输入去预测下一个词，之后，模型会将之前的输入与预测的词串联起来作为新的输入，去预测下下个词，这个过程不断迭代，直到预测得到的词为特殊的结束词 EOS（End-of-Sequence token）或者达到预测词数上限时终止预测（本实验中只需要给定预测词数上限即可）。

2.2 实验说明与要求

2.2.1 文件说明

输入的数据在 part_2/data/input.txt中，为莎士比亚文集，训练数据与测试数据已经在代码中进行了拆分。

代码在 part_2/src/train.py 中，需要同学们进行补全。

2.2.2 实验要求

1. 请补全 train.py 中的代码，
2. 训练模型，记录其训练误差与测试误差，并画出二者随迭代次数的变化曲线
3. 使用训练好后的模型，自取文本片段，对其进行补全，在实验报告中给出你的输入以及补全的结果
4. 可以使用 torch，不许直接调用任何 Transformer 的实现库.

3 提交实验

3.1 截止日期

本实验的截止日期是：2023年7月10日 23:59:59，逾期扣分！

3.2 提交方式

压缩文件，命名格式为 LAB2_PBxxxxxxxx_王二.zip，上传到 bb.ustc.edu.cn 的作业区。

你的提交文件应按如下结构安排。如果你参照上面两个章节提交的文件与以下结构不同，以下面的文件结构为准。

```
-report.pdf
-part_1
  --data // original data here, you don't need any modification
  --src
    --Bayesian-network.py
    --KMeans.py
-part_2
  --data // original data here, you don't need any modification
  --src
    --train.py
```

注意：不需要提交 **part_2/tutorial**.

4 我们为你准备了一些可能有用的教程

4.1 传统机器学习章节

1. numpy 起步教程：NumPy quickstart — NumPy v2.0.dev0 Manual
2. numpy API 查询：NumPy Reference — NumPy v1.24 Manual
3. matplotlib 起步教程：Quick start guide — Matplotlib 3.7.1 documentation
4. matplotlib API查询：API Reference — Matplotlib 3.7.1 documentation

4.2 深度学习章节

如果你对于 Transformer 不甚了解，我们同样为你准备了一份详细的讲解教程，详见 part_2/tutorial/README.md.

请使用支持 markdown 的文本编辑器（如 Typora）来浏览具体内容。

4.3 配置环境有困难?

4.3.1 安装 Python

使用 Linux 系统的同学可以直接使用命令行安装 python 以及其他包管理器，如：conda。如果你尚未尝试过 Linux 系统并对其感兴趣，可以尝试这个教程：Linux 101 在线讲义。

使用其他系统的同学可以参照官方 tutorial: Python For Beginners 或 conda: Getting started with conda

你也可以自己寻找其他博客学习。善用搜索引擎!

4.3.2 安装 pytorch

PyTorch 官网

可以直接使用 pip 安装。终端内输入命令

```
pip install torch==1.8.2+cpu torchvision==0.9.2+cpu torchaudio===0.8.2 -
f https://download.pytorch.org/whl/lts/1.8/torch_lts.html
```

如果速度太慢可以考虑更换 conda/pip 软件源。

如果遇到任何问题欢迎在群里戳助教提问！如果涉及到非常具体的环境问题，请尽量按照提问的智慧在课程群组里参与讨论。记得贴上你的详细操作，报错段或不涉及到实验代码的代码段。