数据科学与工程数学基础 作业1

2021年3月12日 上午

4k 字 34 分钟

一

> 卷积神经网络是一类典型的处理图像的模型，其中卷积是其中一种非常重要的函数操作。试计算下列输入和卷积核做卷积的结果。（注意：此处卷积操作无需旋转180度）
>
> $$\text{Input}=\begin{pmatrix}1&3&0&-1\\3&0&-1&2\\1&-1&2&0\end{pmatrix},\text{Kernel}=\begin{pmatrix}-1&1\\-1&1\end{pmatrix}$$

由卷积运算公式可知

$$\begin{aligned}output_{11}&=1\cdot(-1)+3\cdot1+3\cdot(-1)+0\cdot1=-1\\output_{12}&=3\cdot(-1)+0\cdot1+0\cdot(-1)+(-1)\cdot1=-4\\output_{13}&=0\cdot(-1)+(-1)\cdot1+(-1)\cdot(-1)+2\cdot1=2\\output_{21}&=3\cdot(-1)+0\cdot1+1\cdot(-1)+(-1)\cdot1=-5\\output_{22}&=0\cdot(-1)+(-1)\cdot1+(-1)\cdot(-1)+2\cdot1=2\\output_{23}&=(-1)\cdot(-1)+2\cdot1+2\cdot(-1)+0\cdot1=1\end{aligned}$$

故

$$\text{Output} = \begin{pmatrix} -1 & -4 & 2 \\ -5 & 2 & 1 \end{pmatrix}$$

二

> 现有一组图片数据集，任务目标是将这些图片分类。其中图片中包含的类别有：猫、狗、鹦鹉、人。
> 试用One-Hot向量将类别表示为向量。

$$\begin{aligned} \text{猫} &\xlongequal{def} [1,0,0,0]^T \\ \text{狗} &\xlongequal{def} [0,1,0,0]^T \\ \text{鹦鹉} &\xlongequal{def} [0,0,1,0]^T \\ \text{人} &\xlongequal{def} [0,0,0,1]^T \end{aligned}$$

三

> 现有文本集（一行为一个文本）如下。试计算，该文本集中各个单词（不区分大小写）在各文本中的TF-IDF值。

```
I know.
You know.
I know that you know.
I know that you know that I know.
```

首先计算整个文档中各个词语的IDF值

$$\begin{aligned} IDF_{(I)} &= \ln \frac{4}{3} \\ IDF_{(know)} &= \ln \frac{4}{4} = 0 \\ IDF_{(you)} &= \ln \frac{4}{3} \\ IDF_{(that)} &= \ln \frac{4}{2} = \ln 2 \end{aligned}$$

随后分别计算各个文档中各个词语的TF值

$$\begin{aligned} TF_{(I,1)} = \frac{1}{2}, TF_{(know,1)} = \frac{1}{2} \\ TF_{(you,2)} = \frac{1}{2}, TF_{(know,2)} = \frac{1}{2} \\ TF_{(I,3)} = \frac{1}{5}, TF_{(you,3)} = \frac{1}{5}, TF_{(know,3)} = \frac{2}{5}, TF_{(that,2)} = \frac{1}{5} \\ TF_{(I,4)} = \frac{1}{4}, TF_{(you,4)} = \frac{1}{8}, TF_{(know,4)} = \frac{3}{8}, TF_{(that,4)} = \frac{1}{4} \end{aligned}$$

由此可得各个单词在各文本中的TF-IDF值

$$TF-IDF_{(I,1)}=\frac{1}{2}\cdot\ln\frac{4}{3}\approx 0.1438$$
$$TF-IDF_{(know,1)}=\frac{1}{2}\cdot 0=0$$
$$TF-IDF_{(you,2)}=\frac{1}{2}\cdot\ln\frac{4}{3}\approx 0.1438$$
$$TF-IDF_{(know,2)}=\frac{1}{2}\cdot 0=0$$
$$TF-IDF_{(I,3)}=\frac{1}{5}\cdot\ln\frac{4}{3}\approx 0.0575$$
$$TF-IDF_{(you,3)}=\frac{1}{5}\cdot\ln\frac{4}{3}\approx 0.0575$$
$$TF-IDF_{(know,3)}=\frac{2}{5}\cdot 0=0$$
$$TF-IDF_{(that,3)}=\frac{1}{5}\cdot\ln 2\approx 0.1386$$
$$TF-IDF_{(I,4)}=\frac{1}{4}\cdot\ln\frac{4}{3}\approx 0.0719$$
$$TF-IDF_{(you,4)}=\frac{1}{8}\cdot\ln\frac{4}{3}\approx 0.0360$$
$$TF-IDF_{(know,4)}=\frac{3}{8}\cdot 0=0$$
$$TF-IDF_{(that,4)}=\frac{1}{4}\cdot\ln 2\approx 0.1733$$

四

> 现有一个数据集有5个数据，分别被分类在 $(0,1)^T,(0,1)^T,(0,1)^T,(1,0)^T,(1,0)^T$，而一个模型给出的评分分别为 $(2,8)^T,(1,9)^T,(3,2)^T,(1,5)^T,(2,0)^T$，试给出此时模型给各个数据的概率评分以及交叉熵损失的值。

由

$$Softmax(x_i)=\frac{e^{x_i}}{\sum\limits_{j=1}^{n}e^{x_j}}$$

可知各个数据的概率评分为

$$\begin{pmatrix}e^2/(e^2+e^8)\\e^8/(e^2+e^8)\end{pmatrix},\begin{pmatrix}e^1/(e^1+e^9)\\e^9/(e^1+e^9)\end{pmatrix},\begin{pmatrix}e^3/(e^3+e^2)\\e^2/(e^3+e^2)\end{pmatrix},\begin{pmatrix}e^1/(e^1+e^5)\\e^5/(e^1+e^5)\end{pmatrix},\begin{pmatrix}e^2/(e^2+e^0)\\e^0/(e^2+e^0)\end{pmatrix}$$
$$\approx\begin{pmatrix}0.0025\\0.9975\end{pmatrix},\begin{pmatrix}0.0003\\0.9997\end{pmatrix},\begin{pmatrix}0.7311\\0.2689\end{pmatrix},\begin{pmatrix}0.0180\\0.9820\end{pmatrix},\begin{pmatrix}0.8808\\0.1192\end{pmatrix}$$

又由交叉熵损失计算公式

$$L=-\sum_{c=1}^{K}y_c\log(p_c)$$

可知各个数据的交叉熵损失为

$$L_1=-(0\cdot\log 0.0025+1\cdot\log 0.9975)\approx 0.0036$$
$$L_2=-(0\cdot\log 0.0003+1\cdot\log 0.9997)\approx 0.0004$$
$$L_3=-(0\cdot\log 0.7311+1\cdot\log 0.2689)\approx 1.8949$$
$$L_4=-(1\cdot\log 0.0180+0\cdot\log 0.9820)\approx 5.7959$$
$$L_5=-(1\cdot\log 0.8808+0\cdot\log 0.1192)\approx 0.1831$$

五

> 设数据集为 $x_1,x_2,\cdots,x_n$ 其中被分为两类 y_1,y_2。如果使用线性分类器，试给出一个考虑结构风险的损失函数的公式。

线性分类器的评分函数可表示为（使用**Softmax**作为概率评分函数）

$$f(W,b,\mathbf{x}) = Softmax(W\mathbf{x}+b)$$

若使用交叉熵作为基础损失函数，使用 L_2 范数作为正则化项，则最终的损失函数可以定义为

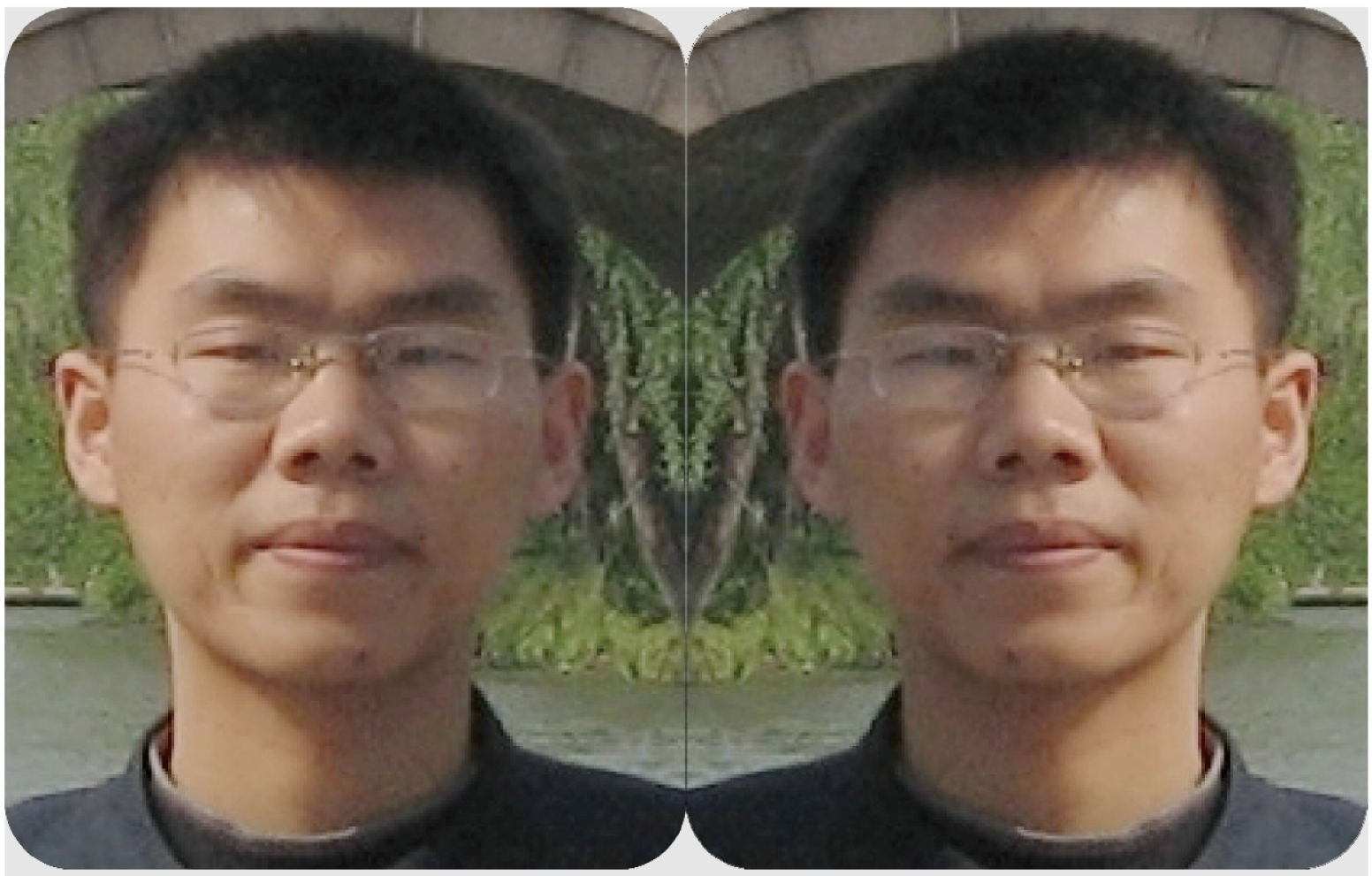
$$J = -\sum_{i=1}^{2} y_c \log(p_c) + \lambda||\mathbf{W}||_2^2$$

其中 y_c 为分类结果指示函数，p_c 为模型给出当前分类的概率评分

六

> 利用Python将一张黑白图片或彩色图片转化为矩阵或张量，并使图片水平翻转。

翻转结果：



merged

实现代码：

```
from PIL import Image
import numpy as np

origin_path = 'origin.jpg'
image = Image.open(origin_path)
tensor = np.asarray(image)
tensor_flipped = np.flip(tensor, 1)
image_flipped = Image.fromarray(tensor_flipped)
image_flipped = image_flipped.convert('RGB')
image_flipped.save('flipped.jpg')
```

数据科学数学基础　Mathematics　DataScience

数据科学与工程数学基础 作业2　　操作系统实验 Shell及系统调用