清华大学电子工程系

媒体与认知 课堂 2

2021-2022 学年春季学期

作业 1

郭中贺 2022 年 3 月 7 日

理论部分

1 单选题（15 分）

1.1 B

1.2 C

1.3 A

1.4 B

1.5 B

2 计算题（15 分）

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

AND

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

异或

图 1: AND,OR, 异或三种逻辑运算

2.1 基于如下单个人工神经元，设计实现两种逻辑门 AND、OR 运算。

$$z = w_1 x_1 + w_2 x_2 + b \tag{1}$$

$$y = f(z) = \begin{cases} 1, z > 0 \\ 0, z \leq 0 \end{cases} \tag{2}$$

2.1.1 AND 实现

只需取

$$w_1 = w_2 = 1, b = -1$$

即可实现与门。

验证：只有当 $x_1 = x_2 = 1$ 时，才能保证 $z = x_1 + x_2 - 1 = 1 > 0$，这样 $y = 1$；当 x_1, x_2 中一个为 1，另一个为 0 时，$z = x_1 + x_2 - 1 = 0$，则 $y = 0$；当 $x_1 = x_2 = 0$ 时，$z = x_1 + x_2 - 1 = -1$，则 $y = 0$。符合与门的设计。

2.1.2 OR 实现

只需取

$$w_1 = w_2 = 1, b = 0$$

即可实现或门。

验证：只有当 $x_1 = x_2 = 0$ 时，才能保证 $z = x_1 + x_2 + 0 = 0$，这样 $y = 0$；当 x_1, x_2 中一个为 1，另一个为 0 时，$z = x_1 + x_2 + 0 = 1$，则 $y = 1$；当 $x_1 = x_2 = 1$ 时，$z = x_1 + x_2 + 0 = 2$，则 $y = 1$。符合或门的设计。

2.2 上述形式的单个神经元是否可以实现逻辑门异或运算？如果是，请给出具体设计；若否，请解释理由。

不可能实现，只需要根据异或门的要求列四个简单的不等式即可证明。

$$b \leq 0, \quad w_1 + b > 0, \quad w_2 + b > 0, \quad w_1 + w_2 + b \leq 0$$

根据第二个和第三个式子，可得

$$w_1 + w_2 + 2b > 0$$

即

$$w_1 + w_2 + b > -b \geq 0$$

结合第四个式子可知

$$w_1 + w_2 + b = 0$$

再结合第二个和第三个式子可知，$w_1 < 0, w_2 < 0$，结合第一个式子则有 $w_1 + w_2 + b < 0$，导出矛盾。因此上述形式的单个神经元不能实现异或运算。

编程部分

3 编程作业报告

3.1 训练模型

运行 python classification.py train 训练模型后，控制台输出结果与 loss 变化曲线分别如图 2、3 所示。其中图 2 的 loss 为对应训练轮数后的验证 loss。

```
(meiren) D:\media-and-cognition\hw1>python classification.py train
epoch5 accuracy: 0.925
epoch5 loss: 0.27801621368154883
epoch10 accuracy: 0.92625
epoch10 loss: 0.26719941351097076
epoch15 accuracy: 0.92625
epoch15 loss: 0.2647063684742898
epoch20 accuracy: 0.9275
epoch20 loss: 0.2640828300733119
```

图 2: cmd 输出结果

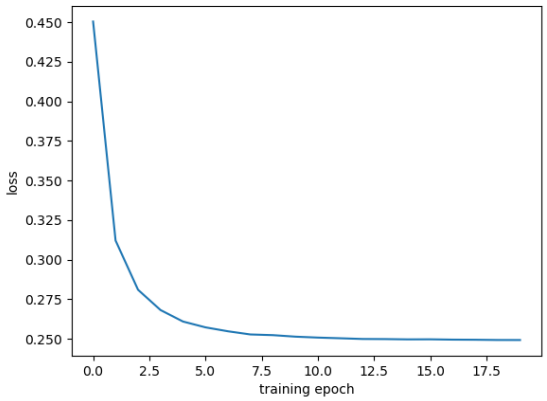

图 3: loss 曲线

结果分析：由图 3 曲线可知，一开始训练的 loss 较高，在 0.5 左右。随着训练轮数的增加，loss 逐渐下降，最终降到 0.25 左右。另一方面，由图 2 输出，可知模型在验证集上的准确率较高，为 92% - 93%，并且随着训练轮数增加准确率有小幅度上升。可见该模型训练效果较好。

3.2 测试模型

运行 python classification.py test 验证模型，控制台输出结果如图 4 所示。

```
(meiren) D:\media-and-cognition\hw1>python classification.py test
[Info] Load model from saved_models/model_epoch20.pth
[Info] Test accuracy = 91.9%
```

图 4: cmd 输出结果

可见使用训练 20 轮之后的模型，在测试集上进行测试，其准确率也较高达到了 91.8%。可见，该训练模型并不只能识别用于训练和验证的数据，具有一定的泛用性。

3.3 可视化

运行 python classification.py visual 进行可视化，输出结果如图 5 所示。

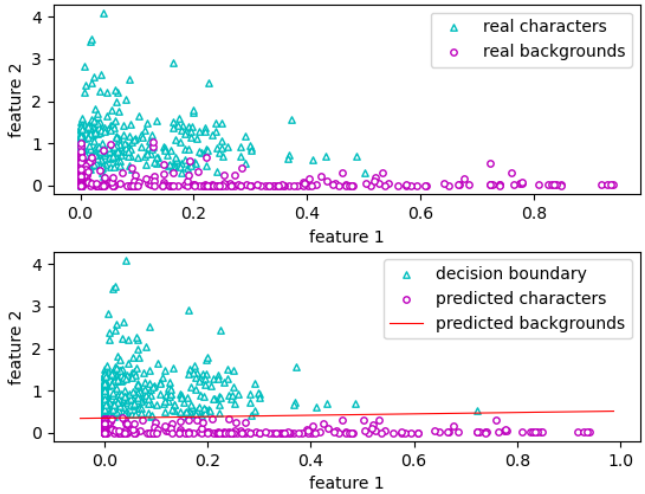

图 5: 可视化结果

通过分析，我们可以看出：大部分的 backgrounds 具有 feature2 数值较小的特点，且 feature1 分布较为分散。而训练模型的 decision boundary 利用这一特点将 characters 和 backgrounds 进行分类。大部分的数据能够分类正确。

4 问题与解决

本次编程作业较为简单，基本上按照助教习题课上的讲解即可搭建一个线性分类器。难点主要在于自己编写二分类交叉熵函数以及不使用 nn.Linear 实现线性层。二分类交叉熵函数由于其存在 log 运算，因此如果遇到 $\log 0$ 则会导致计算出的 loss 均为 inf 出错。我的解决方法是先将数据利用 torch.clip() 函数进行范围限制，防止出现 $\log 0$ 的情况，从而解决了这个问题。另一方面，我使用 nn.Parameter() 自定义训练参数 self.weight 和 self.bais，使得优化器能追踪这两个参数并自动计算参数，forward 函数利用 torch.matmul() 函数实现线性计算，实现了和 nn.Linear() 相同的功能。