

## 测试文档

Yguo. 国越

2025 年 4 月 19 日

# 目录

|     |                |   |
|-----|----------------|---|
| 1   | 第一个段落。         | 1 |
| 1.1 | 数学公式 . . . . . | 2 |
|     | 正文内容:          |   |

$$\begin{matrix} max \\ \Sigma \\ min. \end{matrix}$$

1 第一个段落。

第二个段落。第二个段落。第二个段落。第二个段落。第二个段落。第二个  
段落。第二个段落。第二个段落。第二个段落。第二个段落。第二个段落。  
第二个段落。第二个段落。第二个段落。第二个段落。第二个段落。第二个  
段落。第二个段落。第二个段落。第二个段落。第二个段落。第二个段落。  
第二个段落。第二个段落。第二个段落。第二个段落。第二个段落。第二个  
段落。第二个段落。第二个段落。第二个段落。第二个段落。第二个段落。

第二个段落。第二个段落。第二个段落。第二个段落。

第三个段落. 第三个段落. 第三个段落. 第三个段落. 第三个段落. 第三个段落. 第三个段落. 第三个段落.

[illegible]

第三个段落. 第三个段落. 第三个段落. 第三个段落.

## 1.1 数学公式

$$\sigma = \epsilon \cdot \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

```
1 # 假设输入数据
2 batch_size = 4 # 假设有 4 个样本
3 channels = 512 # 从 Bottleneck 输出的特征通道
4 height = 7 # 特征图的高度
5 width = 7 # 特征图的宽度
6 spacial_dim = height # 假设输入特征图是正方形
7 embed_dim = channels # 嵌入维度与输入通道数一致
8 num_heads = 8 # 多头注意力头数
9 output_dim = 256 # 输出特征维度
10
11 # 生成模拟输入数据（模拟从 Bottleneck 输出的特征图）
12 def generate_input_data(batch_size, channels, height, width):
13     # 生成随机输入数据
14     input_data = torch.randn(batch_size, channels, height,
15                               ↪ width) # 随机生成输入数据
16     return input_data
17 input_data = torch.randn(batch_size, channels, height, width)
```

```
1 ./hfd_revised.sh bigcode/starcoderdata --dataset \
2 --include "python/*" \
3 --include "sql/*" \
4 --include "matlab/*" \
5 --include "javascript/*" \
6 --include "java/*" \
7 --include "json/*" \
8 --include "c/*" \
```

```
9      --include "rust/*" \  
10     --include "go/*" \  
11     --include "typescript/*" \  
12     --include "kotlin*" \  
13     --include "swift/*" \  
14     --include "julia/*" \  
15     --include "markdown/*" \  
16     --include "html/*" \  
17     --hf_username TsaiTsai0929 \  
18     --hf_token hf_xxxx \  
19     --tool aria2c \  
20     -x 4 -j 5 \  
21     --local-dir ~/autodl-tmp/dataprocess/data/starcoder
```

- 苹果
- 香蕉
- 橙子

1. LORA
2. PEFT
3. 3

**Step1** 低秩适配器

**Step2**