

Torch Week3 Report

`__repr__` output in grade_part1.py

```
Running module_repr...

Module repr output:
SimpleNet(
  (layer1): Linear(in_features=3, out_features=4, bias=True)
  (activation): Tanh()
  (layer2): Linear(in_features=4, out_features=2, bias=True)
)
✓ module_repr passed (10 points)
```

Grade summary from grade_all.py

```
=====
WEEK 3 TEST SUMMARY
=====
PART 1: Core Module System                ✓ PASSED
PART 2: Simplest Concrete Modules          ✓ PASSED
PART 3: Initialization Functions           ✓ PASSED
PART 4: Concrete Modules                   ✓ PASSED

OVERALL RESULT: 4/4 parts passed
🎉 ALL TESTS PASSED! Week 3 implementation is complete.
```

Challenges encountered and solutions

本周内容参考了不少 pytorch 源码，尤其是 module.py。前面主要是靠读代码和简化代码写的。不得不感叹工业级代码的异常处理和各种分支好多。后面基本就是读 tutorial 和面向测试点调试。中间又发现 C++ 层的 `Tensor::mean` 函数没处理负数下标，导致 `estate_value_predict` 中间的向量全都是 nan以及 week2 中 `Function` 的 `run` 函数有一个小问题。或许还是要加一些随机数轰炸的测试点才能测全。

一个写了比较久的地方是 `MultiHeadAttention`。一开始看到这个类的时候还有点小激动，因为小学期演讲中有提到过这个但是当时没咋听懂在干什么。然后在各大视频网站搜索 `MultiHeadAttention` 的原理和实现。感觉真正难的其实是建模和数学推导的过程，大概明白之后代码写还是比较好写的。

总的来说这周内容比前两周好实现一些，但收获也不少。从 0.5（毕竟框架不是自己写的）开始搭建一个能跑的简易 module 还是挺有意思的。btw 没想到这样一个简易的框架就能跑出接近 88% 的识别手写数字正确率。

Optional challenge: Conv2D

Semantic of unfold, it's backward and its use in Conv

unfold 也叫 im2col, 即 Image to Column。它的想法是将输入图像中所有重叠的、由卷积核定义的 "patch" (小块区域) 提取出来, 并将每个 patch 展平 (flatten) 成一个列向量。然后把这些列向量被堆叠起来, 形成一个新的二维矩阵作为输出。这个过程是为了将二维卷积操作 (Conv2D) 转化为矩阵乘法, 从而在一次矩阵乘法中就可以完成卷积。具体而言是转化成 $Output = Unfolded_Image @ Flattened_Kernel$ 。

unfold 操作的 backward 是 fold。fold 操作也叫 col2im, 即 column to image。它将 unfolded 矩阵中的列向量 (代表卷积核感受野的梯度或值) 累加回原始图像的对应位置。由于卷积操作通常有重叠区域, fold 操作在将展开的向量累加回原始图像时, 通常是对重叠区域的值进行求和或平均 (我采用求和)。

My implementation

Conv2D 没经过严格测试, 只是通过了AI生成的4个测试 (\tests\week3\my_conv_test.py)。先放这备份了。

(clownpiece 目录下)

- tensor/tensor_pybind.cc:
601-620 行添加了 unfold 和 fold 函数的绑定
- tensor/tensor.cc 和 tensor/tensor.h
在 tensor.cc 最后用 naive 的 for 循环方法实现了 unfold 和 fold 函数
- tensor.py
TensorBase 和 Tensor 类中加了支持 Unfold 和 Fold 函数的内容
- autograd/function.py
添加了 class UnFold(Function)
unfold 的 backward 是 fold, 直接调用
- nn/layers.py:
class Conv2D
实现了简单卷积核 module。
定义好卷积核参数之后, 有可被学习的卷积核 (weight) 和偏置 (bias)。
Forward 函数中具体定义了 Conv2D 的卷积操作, 主要由 UnFold 函数实现。