

# 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。

OC

TODO