

# 课后作业

手写或打印一页A4纸/题  
报告请于课堂提交

作业编号 4  
满分10分

# 使用Pytorch实现 ELU, Dropout层

1. 阅读 Pytorch 文档，理解 [ELU](#), [Dropout](#) 层的作用和参数
2. 下载 [示例代码](#)，阅读尝试运行。查看输出结果, 找到assert 报错的地方。
3. 理解注释代码，正确实现 backward() 函数，返回正确的反向传播输出。
4. 确保所有assert正确。打印查看所有 dx 等 TODO-Explain 的位置。
5. 回答问题 Q1, Q2。

# Q1 (5分) ELU 层反向传播实现与验证

Step 1: 复习课件，回顾求导的链式法则。

Step 2: 运行 `layer_elu.py`, 查看中间输出，理解注释。

Step 3: 实现 `MyELU` 类 `backward_manual()` 函数。

- 在 `forward` 中，`X_bottom` 为输入，`X_top` 为输出。
- 在 `backward` 中，传入 `X_top` 的导数 `delta_X_top`，计算 `X_bottom` 的导数 `delta_X_bottom` 作为回传输出。

Step 4: 理解 `main` 函数中，损失函数 `loss L` 的计算，理解 `Delta_X_top` 是  $DL/Dy$ ，即代码中的 `y_diff`. 理解 `y_diff` 作为 `Delta_X_top` 输入到 `backward` 函数进行回传。

Step 5: 完成 `backward_manual()` 函数。确保所有 `assert` 正确。

Step 6: 解释所有 `TODO-Explain` 行代码。

## Q2 (5分) Dropout 层反向传播实现与验证

Step 1: 运行layer\_dropout.py, 查看中间输出, 理解注释。

Step 2: 阅读pytorch [Dropout](#) 文档, 学习 Caffe 中的[cpp](#)实现, 理解伯努利分布的生成, 理解为什么drop之后需要scale, 理解mask函数的作用。

Step 3: 实现 MyDropout类 backward\_manual() 函数, 正确返回 X\_bottom 的导数 delta\_X\_bottom。

Step 4: 理解main函数中, 损失函数 loss L的计算, 理解 y\_diff, 理解 torch.autograd.grad 函数的含义, 并比较backward\_manual 和 torch.autograd 的输出是否一致。

Step 5: 完成 backward\_manual() 函数。确保所有assert正确。

Step 6: 详细解释所有 TODO-Explain 行代码。