

# 计算机科学与技术学院神经网络与深度学习课程实验报告

实验题目: hw0		学号: 201605130098
日期: 2019. 4. 10	班级: 16 智能	姓名: 李云开
Email: <a href="mailto:liyunkai@foxmail.com">liyunkai@foxmail.com</a>		
实验目的: 了解并使用 numpy, pytorch		
实验软件和硬件环境: Python3.6, numpy, pytorch		
实验原理和方法: 分别实现相关激活函数与 slice, padding 方法		
实验步骤: (不要求罗列完整源代码)		
1、relu		
<pre>def vectorize_PrimeRelu(array):     """     Takes one 2-dimensional array and apply the derivative of relu function on all the values of the array.     :return:     """     temp_cal = torch.nn.Parameter(torch.from_numpy(array).float())     relu_result = torch.sum(torch.relu(temp_cal))     relu_result.backward()     return temp_cal.grad.numpy()</pre>		
直接用 pytorch 实现		
2, slice		
Slice 分三步, 分别是指定切片, 最后几位切片, 随机切片。 由于 numpy 不支持这种不定长向量, 所以用 List 实现。		

Dense input:

```
[[[-1.12583983 -1.1523602 -0.25057858]
  [-0.43387881 0.84871036 0.69200915]
  [-0.31601277 -2.11521935 0.32227492]]

 [[-1.26333475 0.34998319 0.26604941]
  [ 0.16645534 0.87438184 -0.14347385]
  [-0.11160933 -0.61358309 0.03159274]]

 [[-0.49267697 0.0537369 0.61805665]
  [-0.41280222 -0.84106481 -2.31604195]
  [-0.10230965 0.79244399 -0.2896677 ]]]
```

Sparse input:

```
$$[
  [
    [1., 1.],
    [2.],
    [1.2]
  ],
  [
    [2.3, 4.0],
    [6., 7.8., 9.]
  ]
]$$
```

```
[-0.10230965 0.79244399 -0.2896677 ]]]
[[[1.0, 1.0], [2.0]]]
[[[1.0], [2.0], [1.2]], [[4.0], [9]]]
[[[-1.12583983]
  [-0.43387881]]

 [[ 0.34998319]
  [ 0.87438184]]

 [[ 0.0537369 ]
  [-0.84106481]]]
```

三种结果展示: Process finished with `exit` code 0

```
sparseinput = [[[1., 1.], [2.], [1.2]], [[2.3, 4.0], [6., 7.8., 9. ]]]
input = torch.randn(3, 3, 3).numpy()
print(input)
print(slice_fixed_point(sparseinput, (0, 0, 0), (1, 2, 2)))
print(slice_last_point(sparseinput, 1))

print(slice_random_point(input, (2, 1)))
```

Pytorch 的随机种子有点复杂。

```
torch.manual_seed(0)
numpy.random.seed(0)
```

剩余可能的地方还需要对相关 GPU 随机种子初始化

3 padding

Padding 使用了 numpy 自带的 pad 函数:

包括 镜像以及在某个方向, 这里的输入用的是 dense input

结论分析与体会：

感觉 numpy 文档好乱= =，（用 pytorch 习惯了，感觉事少，省心。。

就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟 1—3 道问答题：

Pytorch 与 numpy 如何让随机生成的数据可以复现

初始化随机种子，

```
torch.manual_seed(0)
numpy.random.seed(0)
```