

# ProblemList

---

1. 在对输入数据进行脉冲编码时需要把数据范围压缩到[0,1]之间，除了这个还需要对数据进行哪些预处理？标准化吗？这有什么意义？

网上说标准化能加速网络的训练，消除量纲的影响，不太懂。

2. 在对输入数据进行归一化时应该对所有输入数据采用同一归一标准还是对每条输入数据使用不同的归一标准？

3. 在使用测试集对网络进行验证的时候，应该使用整个测试集进行验证还是从测试集中选择一个batch进行验证？

前者似乎准确率更高一些，但是在大数据集中比较困难？

4. 1st-order model神经元中的参数beta有什么作用？beta大小会对神经元产生什么样的影响？

根据公式 $U[t+1] = \underbrace{\beta U[t]}_{\text{decay}} + \underbrace{WX[t+1]}_{\text{input}} - \underbrace{S[t]U_{\text{thr}}}_{\text{reset}}$ ，beta越大表示之前对当前的影响越多，这个神经元的记忆性会好点？

5. 他人经验：在数据集比较简单的情况下，让 $\alpha$ 尽可能小，而当数据集变复杂时，应使用更大的 $\alpha$ 。为什么？
6. alpha神经元到底是个啥？看不懂。

## # 实验部分

---

一个最简单的三层snn神经网络，使用128维的输入层、512维的隐层和6维输出层，时间步总共为32。脉冲编码使用速率编码，网络结构如下所示

```

1  # Network architecture
2  num_inputs = 128
3  num_hidden = 512
4  num_outputs = 6
5  num_steps = 32
6  net = SimpleLeakySNN(num_inputs, num_hidden, num_outputs,
    num_steps).to(device)

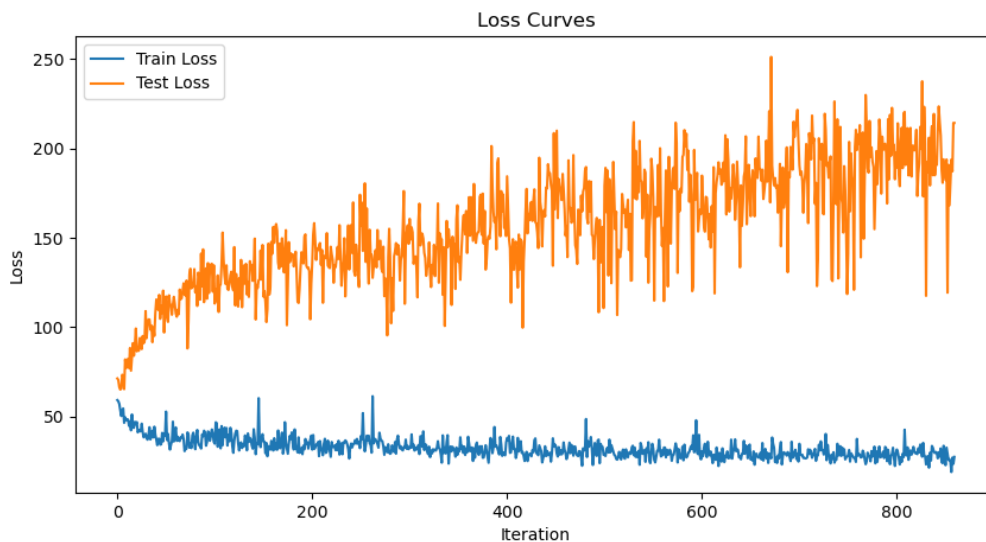
```

```

1  class SimpleLeakySNN(nn.Module):
2      def __init__(self, num_inputs, num_hidden, num_outputs,
    num_steps):
3          super().__init__()
4          self.num_steps = num_steps
5          beta = 0.9
6          # Initialize layers
7          self.fc1 = nn.Linear(num_inputs, num_hidden)
8          self.lif1 = snn.Leaky(beta=beta)
9          self.fc2 = nn.Linear(num_hidden, num_outputs)
10         self.lif2 = snn.Leaky(beta=beta)

```

训练结果惨不忍睹，一上来就进入了过拟合，以下为猜测的原因：



1. 数据集大小问题。训练集有11128条数据，测试集有2783条数据。

2. 网络太大了？再减少一点隐层神经元个数？

试过，效果一样烂。

3. 每次只抽取1个batch的测试集进行测试，loss测量不准确。可以考虑换成整个测试集进行测试。

可能的尝试方法：

1. 换个大点的数据集。。暂时找不到
2. 加入正则项。还在写
3. 换种神经元和编码方式。
4. k折交叉验证。