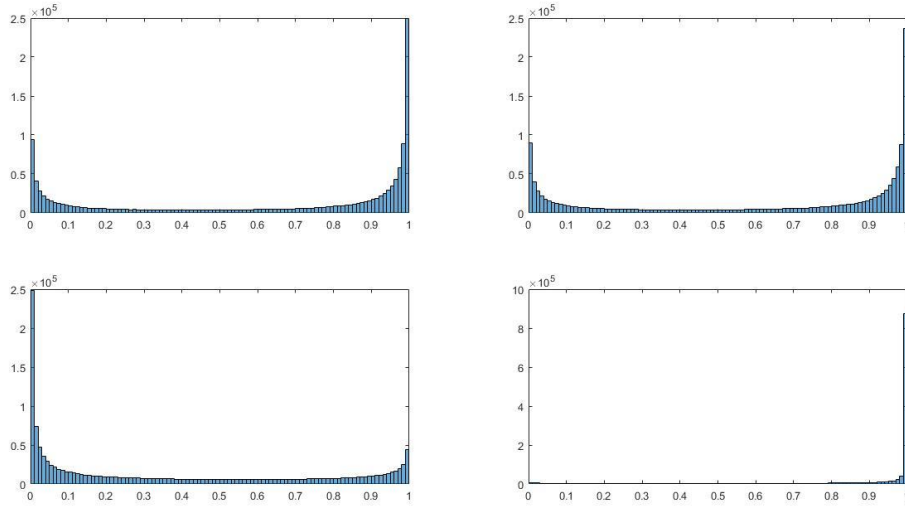


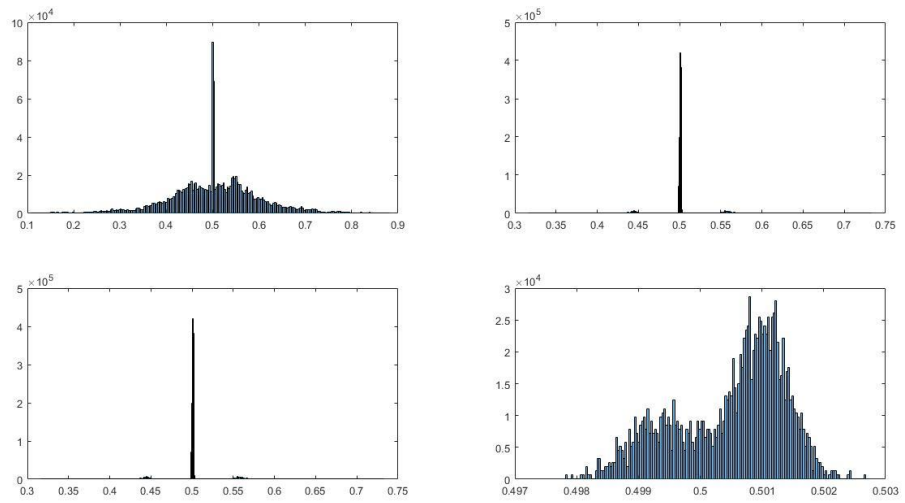
Sigmoid:

1. 三种 prune 方法对输出（最后一层激活函数的输出）的影响

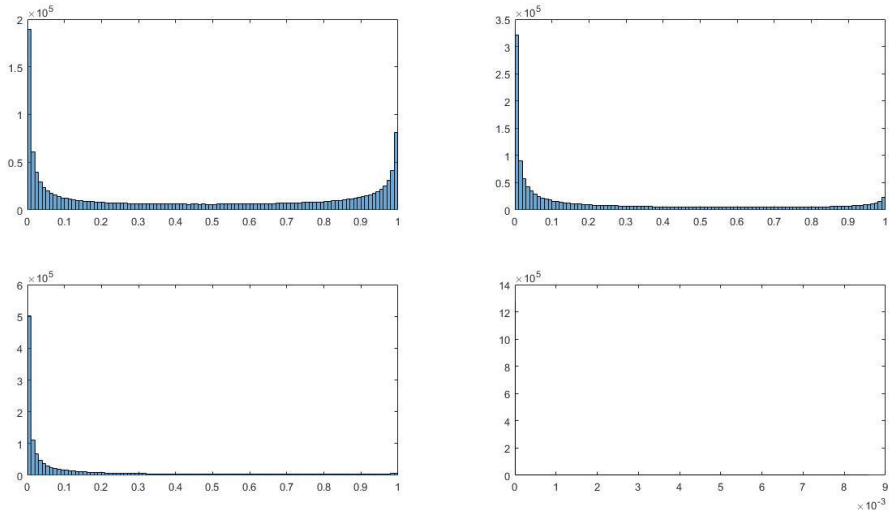
图像顺序: baseline - abs - positive - negative （三组实验为同一阈值）



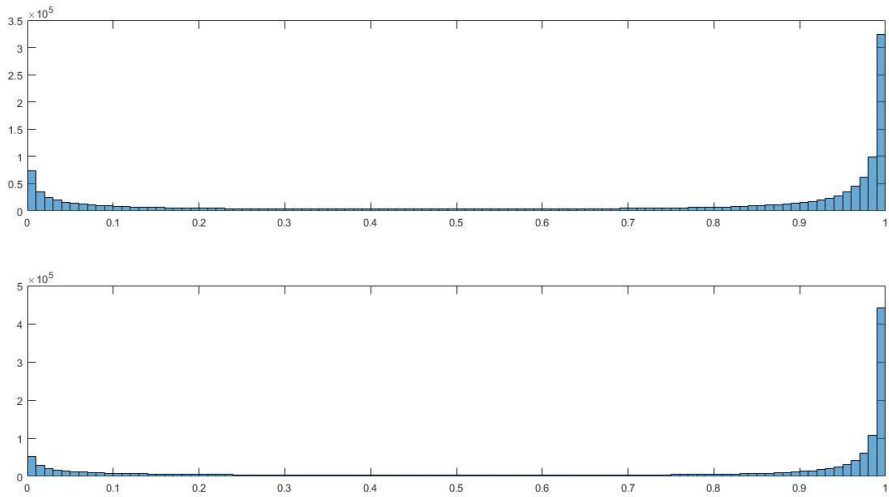
2. 使用 abs Prune —— 向中部集中



3. 使用 Positive Prune



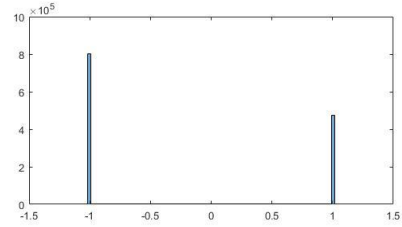
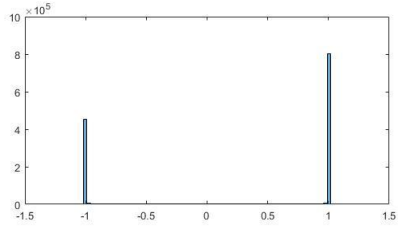
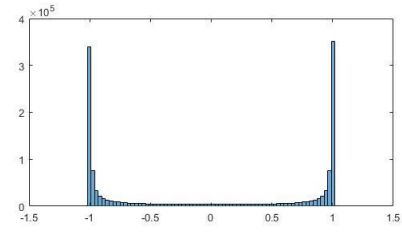
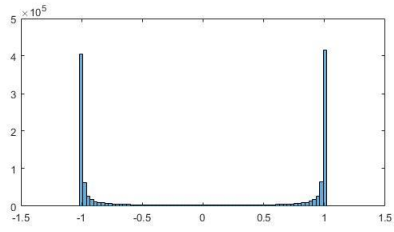
4. 使用 Negative Prune



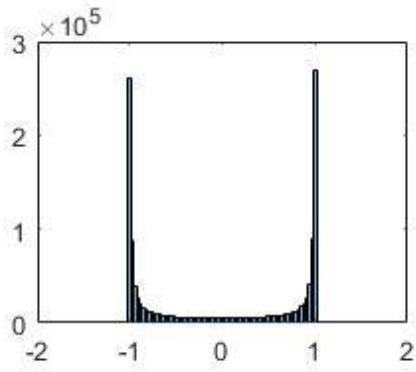
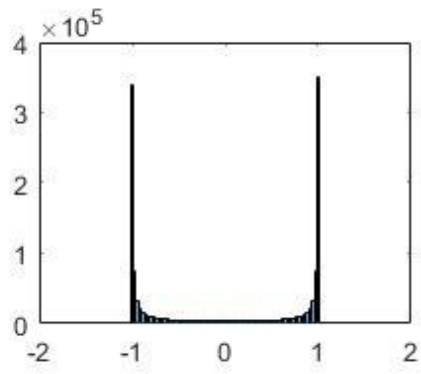
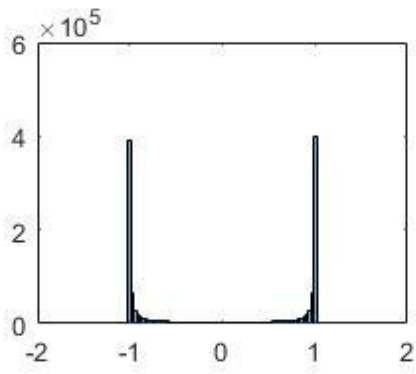
Tanh:

1. 三种 prune 方法对输出（最后一层激活函数的输出）的影响

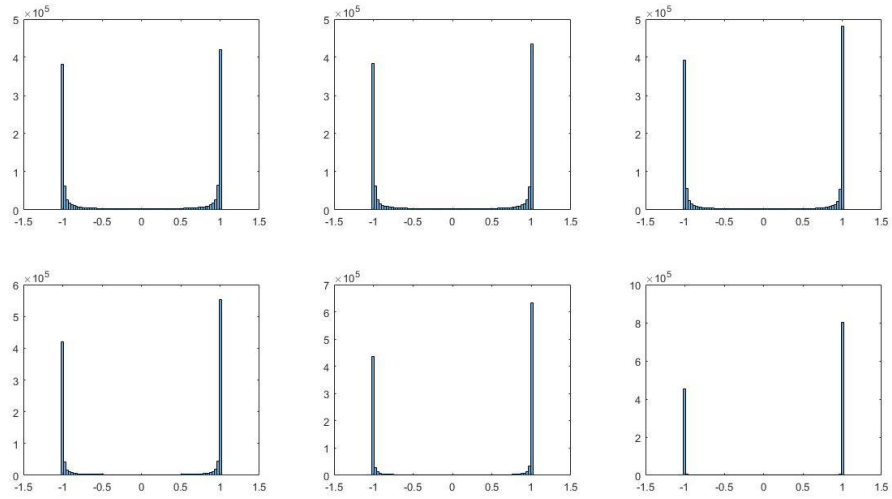
图像顺序：baseline - abs - positive - negative （三组实验为同一阈值）



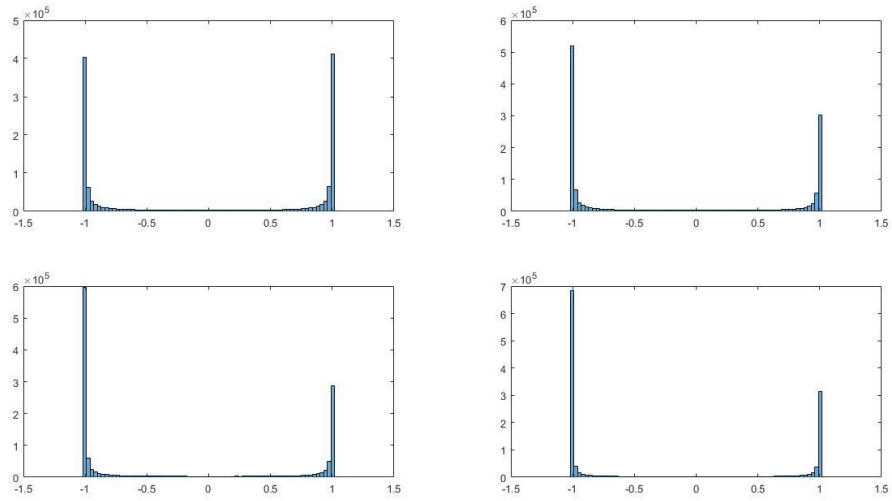
2. 使用 abs Prune —— 向中部集中



3. 使用 Positive Prune



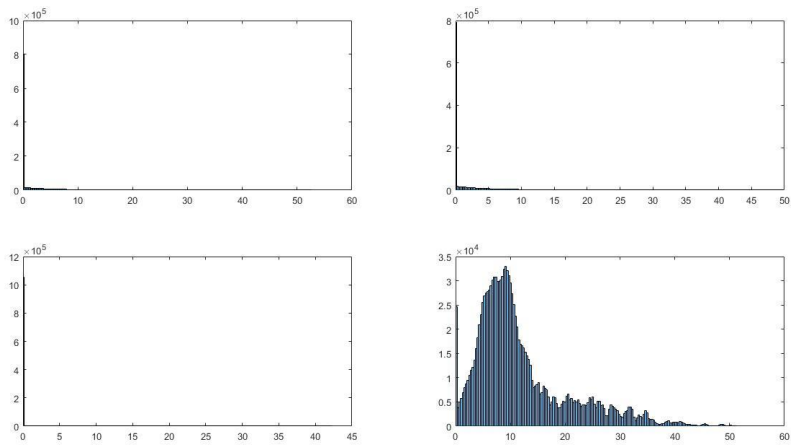
4. 使用 Negative Prune



ReLU:

三种 prune 方法对输出（最后一层激活函数的输出）的影响：

图像顺序：baseline - abs - positive - negative （三组实验为同一阈值）



Why ABS Prune is better than Positive/Negative Prune?

1.

使用绝对值阈值进行网络裁剪，只要阈值不过大，**不会造成输出层明显的偏移**；

而使用 Positive / Negative 则会**放大各层的定向偏移**，最终在输出层的输出明显不合理。

2.

在解码过程中，最关键的是各个输出值的大小关系（state 的后验概率），因为 Positive Prune

和 Negative Prune 的各层输出大多处于激活函数的饱和区，**所有值的大小都极为相近**。因此，

各输出值之间的大小关系与理想的输出相比，**完全被颠覆**。