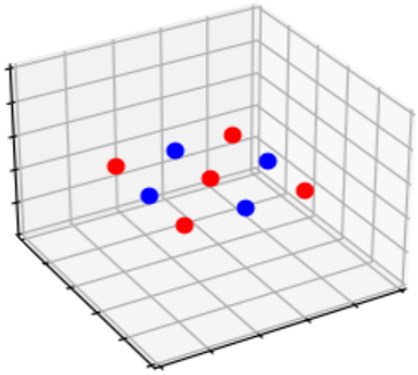
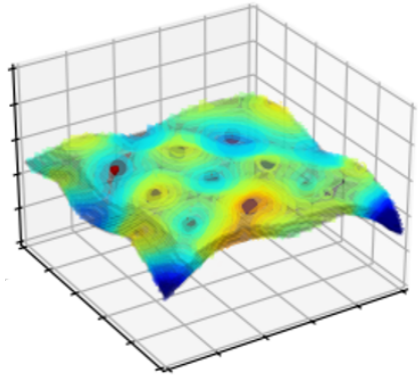
结论：

1. 输入数据一般分布在一个低维流形上，比如输入数据是3维的，那么流形可能分布在一个2维子空间里。如下图：



2. 训练DNN大概分为两个过程：①使决策边界由随机快速分布到数据的manifold附近 ②通过在决策边界中产生凹凸，来使决策边界移动到样本的正确一侧。如下图所示，五个凹坑（想让红点在它们之上），四个突起（想让蓝点在它们以下）



3. 为了辅助训练，DNN会在样本附近产生较大的梯度，使得决策边界向垂直于manifold的方向移动，从而使边界到样本的正确一侧。这个过程可以想象成是拿一个小锤在决策边界上砸，会产生许多凹凸不平的小坑。
4. 一个对泛化能力的可能的解释是，这种在样本点附近砸坑的行为会逐渐连成一个更大的坑，从而包住多个样本。这些被大坑包住的就是被泛化到的。
5. AT会使得决策边界的起伏更大。如下图所示：

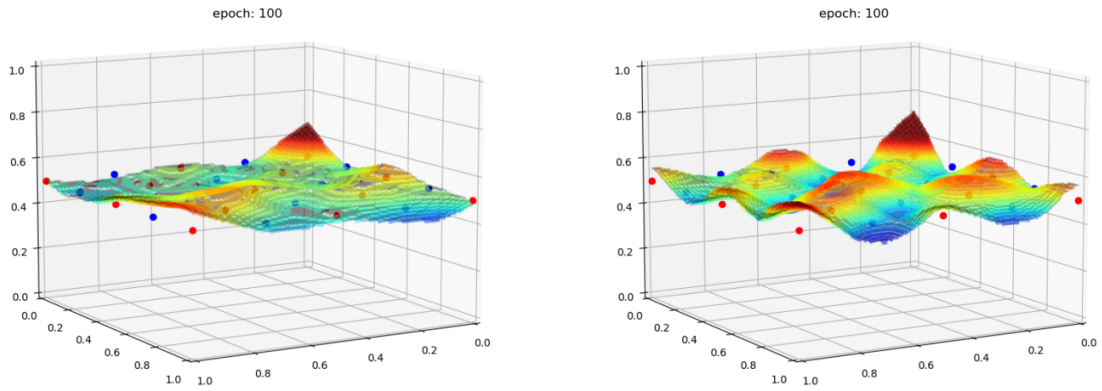


Figure 3: A 3D binary decision boundary of two NNs with the same architecture and same random initialization. The NN in the left image was trained with "clean" 2D data; the right NN was adversarially trained using a single-step L_2 PGD attack.