Shanghai Jiao Tong University

类脑智能大作业

Student name: *520030910388* 李熠星 *520030910342* 柳纪宇 *520030910341* 周骏东

Course: 类脑智能 – Professor: 叶南阳

Date: *January 6, 2022*

1 基础部分

1.1 ColoredMNIST 简介

ColoredMNIST 数据集是一种常用的检验模型分布外泛化 (Out-of-Distribution) 能力的数据集，通过对 MNIST 数据集添加不同的颜色得到。该数据引入了颜色对 label 的虚假因果性，在训练集和测试集中，不同 label 的样本的颜色分布有很大的差距，从而对模型学到从形状到 label 到真正的因果关系提出了较大挑战。

值得注意的是，数据集对 25% 的样本的 label 实施了翻转，因此理论上理想模型的预测正确率为 75%，若模型预测正确率超过 75%，则必有原数据集标错的样本预测结果与标错的结果相同，即模型性能不够优秀。

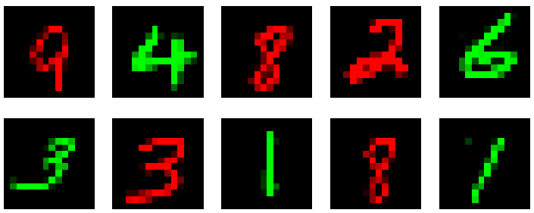


图 1: colored mnist 数据集

1.2 后门准则与数据集变换

在本实验中，我们抽象出如下图4所示的因果图，其中 C 为字体的颜色，也是带来虚假因果性的干扰项，X 为字体的形状，Y 为 label。

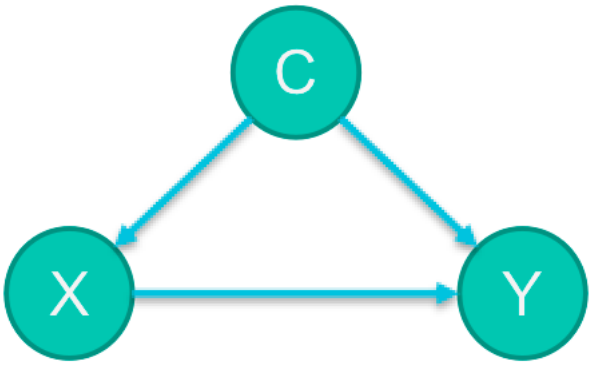


图 2: 因果图

我们通过对 X 施加干预来消除 C 对 Y 的影响，根据后门准则，我们有

$$P(y|do(x)) = \sum_c P(y|x,c)P(c) \tag{1}$$

使用数据集 (X,Y) 直接训练一个神经网络，我们会得到对 $P(y|x)$ 的拟合，在同样的因果图中，我们有

$$P(y|x) = \sum_c P(y|x,c)P(c|x) \tag{2}$$

比较上述两式，唯一的差别在于最后一项。在原数据集中，$P(c)$ 为一半概率为红色一半概率为绿色的均匀分布，因此，我们需要改造数据集，使得 $P(c|x)$ 也为平均分布，即对每种 label，保证其中红色样本和绿色样本数目相同。

1.3 实验结果

表格1的前两列展示了直接使用原数据集进行训练和使用调整后的数据集进行训练的结果。可以看到，直接把原数据集放入 CNN 中训练效果很差，几乎没有分类能力。

加入了后门调整后，我们的模型达到了 **64.32%** 的正确率。结合图像归一化等处理操作后，我们的基础模型在训练 5 轮后达到了 **70%** 以上的正确率，而进一步增加训练轮数则使得模型出现过拟合问题，正确率下降。过拟合问题在分布外泛化条件下尤为明显。

考虑到1.1中的分析，理论上理想模型的预测正确率为 75%，在基础部分我们的工作达到了不错的正确率。**我们在表格1中给出了模型的正确率结果**。

2 提高部分

2.1 SNN

在这个部分我们将使用 snnTorch 工具包实现脉冲神经网络的模拟，并探究提高鲁棒性的方法。

2.1.1 原理

脉冲神经网络 SNN 实现了生物神经模拟水平，除了神经元和突触状态之外，SNN 还将时间概念纳入了其操作之中，是一种模拟大脑神经元动力学的一类很有用的模型。

SNN 使用最拟合生物神经元机制的模型来进行计算，更接近生物神经元机制。脉冲神经网络与目前流行的神经网络和机器学习方法有着根本上的不同。SNN 使用脉冲而非连续值，这是一种发生在时间点上的离散事件。每个峰值由代表生物过程的微分方程表示出来，其中最重要的是神经元的膜电位。本质上，一旦神经元达到了某一电位，脉冲就会出现，随后达到电位的神经元会被重置。SNN 通常是稀疏连接的，并会利用特殊的网络拓扑。

2.1.2 实验

在编写 SNN 的训练代码时，需要注意的是以下三点 (1)：

1. 脉冲神经元的输出是二值的，而直接将单次运行的结果用于分类极易受到干扰。因此一般认为脉冲网络的输出是输出层一段时间内的发放频率，发放率的高低表示该类别的响应大小。因此网络需要运行一段时间后的平均发放率作为分类依据。

2. 我们希望的理想结果是除了正确的神经元以最高频率发放，其他神经元保持静默。常常采用交叉熵损失或者 MSE 损失，这里我们使用实际效果更好的 MSE 损失。

3. 每次网络仿真结束后，需要利用 utils.reset() 重置网络状态;

在表格1中的 SNN 一列展示了我们使用 SNN 作为网络 backbone 进行实验的结果。在使用 SNN 网络进行实验的过程中，我们发现学习率、优化器的动量参数等超参对于实验结果有较大的影响。我们在超参空间进行调参，使用效果较好的参数作为最终结果。

SNN 网络相较于我们组的其他实验，其收敛较缓。在 epoch 为 30 时正确率为 **66.39%**，但仍未收敛。网络在第 90 个 epoch 左右收敛，收敛后的最高正确率为 **68.90%**.

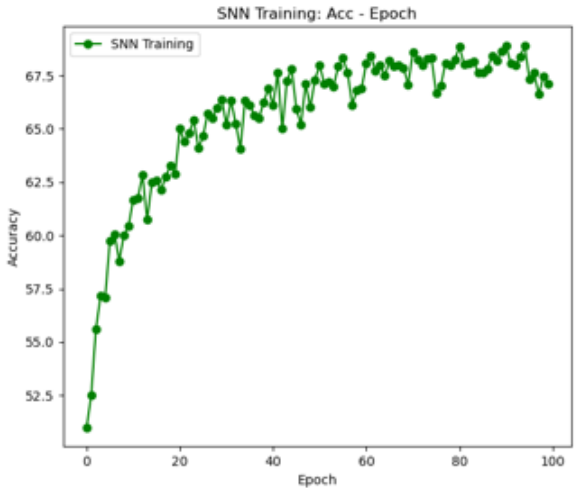


图 3: SNN 训练过程

2.2 MNN

在这个部分我们将使用 MemTorch 工具包实现忆阻器神经网络的模拟仿真，并探究提高鲁棒性的方法。

2.2.1 原理

忆阻器是一种能够表征电荷和磁通量之间关系的电路元件，它能够通过电阻值的变化记录流经的电荷或磁通。近年来，忆阻器在促进深度学习系统的加速和提高能源效率的仿麦呢表现出了巨大前景，通过 crossbar 结构我们能够高效地实现各种内存计算操作，如乘积累加和卷积等，从而能够被用于人工神经网络搭建。然而，忆阻器也面临着元件老化和非理想性的问题，这限制了忆阻器神经网络的准确性、可靠性和鲁棒性。因此，在实验仿真的过程中我们需要将忆阻器元件的非理想性考虑进去。

2.2.2 实验

使用 MemTorch 库 (2) 将 CNN 转化为 MNN，**表格1中的 MNN 一列展示了将原 CNN 转化为 MNN 后的预测准确率**。由于在转化中加入了对硬件非理想化的仿真，正确率相比原模型出现了较大程度的下降。为增加模型鲁棒性，我们尝试了在训练过程中向图片添加噪声，使得模型对抗干扰的能力有了一定增强，但相比我们实现的其它网络结果还是有一定差距，可见要消除元件非理想的干扰不能仅靠简单的数据拓展（结果见 MNN_noise 一列）。

2.3 DANN

2.3.1 原理

DANN (Domain-Adversarial Neural Network) (3) 是一种用于域迁移学习的神经网络模型。它的主要思想是将来自不同领域的样本输入到神经网络中，然后使用一个对抗网络来学习区分不同领域的特征。这个对抗网络的目的是使得不同领域的数据在特征空间中的映射尽可能相近，从而提高模型在不同领域之间的泛化能力。这种方法的优点在于，我们在训练前无需预先对数据集进行处理，也即无需获取关于数据集颜色的信息。

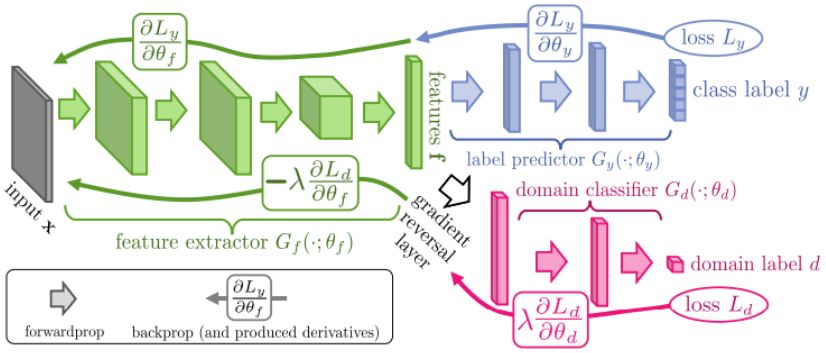


图 4: DANN 结构示意图

DANN(3) 一共由三部分组成：一个特征提取器，一个分类器和一个域分类器。特征提取器用于从输入图像中提取特征，分类器用于对输入图像进行分类，域分类器用于区分来自不同领域的特征。特征提取器和分类器构成了一个分类网络，而特征提取器和域分类器构成了一个对抗网络。在对抗网络中，目标域的样本替代了生成器所做的生成样本的工作，而域分类器则起到了类似鉴别器的作用。

在训练过程中，除了从源域中取样本通过分类网络计算分类误差之外，我们还从源域和目标域中随机取样本，将其通过特征提取器后输入域分类器，将得到的结果与域标签计算域分类误差。模型的总训练误差为分类误差、源域样本域分类误差和目标域样本的加权和。公式表达如下：

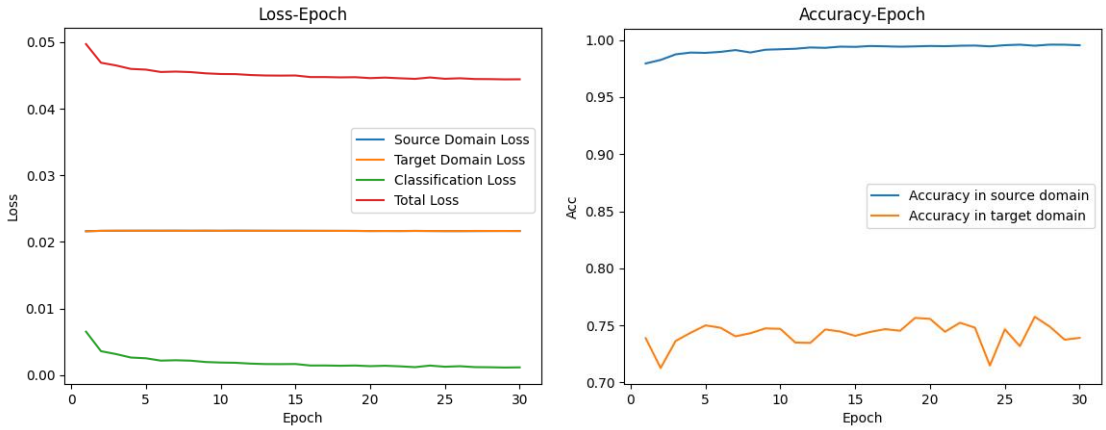
$$L_t = w_1 * l_c + w_2 * l_{d;s} + w_3 * l_{d;t}$$

其中 L_t 表示总误差，l_c 表示分类误差，$l_{d;s}$ 表示源域样本的域分类误差，$l_{d;t}$ 表示目标域样本的域分类误差。

在训练过程中我们的目标分别为（1）实现源域样本的正确分类，使得分类误差最小化（2）实现源域和目标域样本在经过特征提取器后的混淆，使得域分类误差最大化。为了同时实现这两个目标，DANN 在特征提取器和域分类器之间加入了梯度反转层 (Gradient Reversal Layer, GRL)，从而能够在后向传播的过程中同时在分类器中实现梯度下降并在域分类器中实现梯度上升。

2.3.2 实验

在实验过程中，我们首先尝试了使用 colored mnist 中的训练集作为源域，将测试集作为目标域进行训练。可以看到，我们的 DANN 神经网络在测试集上的准确率仅 0.1 左右，远远低于预期效果。观察到模型域分类误差一直维持在 0.7 左右，因此我们猜想 DANN 分类效果差的原因可能是在 DANN 的初始训练过程中，颜色相关的信息在特征学习过程中占了主导地位。由于初始模型学习到的表征方式几乎完全基于颜色，导致在后续的训练过程中无论梯度回传多少次模型都无法学习到一种形状主导的特征表示方法。**实验结果可见1中 DANN1 一栏。**



(a) 训练过程误差变化曲线图　　(b) 训练过程正确率变化曲线图

图 5: DANN 训练曲线（注：图 (a) 中源域样本域分类误差与目标域样本域分类误差曲线基本重合）

在这之后，我们还尝试了将标准 mnist 作为源域，将 colored mnist 作为目标域进行迁移学习。经过 30 个 epoch，我们的模型在测试集上达到了 74.83% 的最高准确率，该数值已经非常接近 colored mnist 数据集的理论最高准确率。这说明模型在 mnist 数据集中学习到的表征方式被很好地迁移到了 colored mnist 数据集上。实验结果可见1中 DANN2 一栏。

训练过程中的误差变化曲线和正确率变化曲线见图 5。

2.4 贡献

柳纪宇：MNN 与 SNN 接口初步搭建，DANN 调研与实现；
李熠星：基础版本实验，SNN 调试与后续实验，ppt 制作；
周骏东：基础版本代码实现，MNN 调试与实验，成果展示。

3 实验结果

Epoch	Origin	Balanced	SNN	MNN	MNN_noise	DANN1	DANN2
1	16.26	68.52	50.99	50.01	52.82	9.67	69.48
5	23.34	**70.32**	57.11	50.44	53.15	10.24	73.68
10	22.97	63.90	60.43	54.19	55.21	**12.32**	74.14
20	30.57	59.08	62.87	54.01	**55.34**	10.81	71.72
30	**37.04**	58.38	**66.39**	**54.58**	55.31	11.67	**74.61**

表 1: 不同方法下、不同 Epoch 下测试集准确率对比

参考文献

[1] https://snntorch.readthedocs.io/en/latest/tutorials/tutorial-7.html

[2] https://github.com/coreylammie/MemTorch

[3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Victor Lempitsky, Domain-Adversarial Training of Neural Networks, Journal of Machine Learning Research 2016, vol. 17, p. 1-35