

基于手写体识别的等变卷积技术应用与理解

姓名：杨昊

学号：4124136025

电话：15735153435

邮箱：cvlab.yang@stu.xjtu.edu.cn

摘要：本次《深度学习与人工智能》大作业实现了普通卷积与等变卷积的对比，并基于简单深度神经网络完成了分类任务。实验使用了简化版的 LeNet 和 AlexNet，以及对应使用等变卷积的简化 LeNet 和 AlexNet 对 MNIST 数据集进行分类。实验设置包括四种情况：（1）原始训练集、验证集和测试集均不进行旋转；（2）仅训练集进行旋转增强；（3）验证集和测试集进行旋转；（4）所有数据均进行旋转。实验结果表明，使用等变卷积后，所有模型的训练效果均逊色于使用普通卷积的情况。数据处理代码、训练代码及实验结果已公开，详见 GitHub(<https://github.com/HaoYang-98/CLASS-Deep-Learning-and-Artificial-Intelligence>)。

1 介绍：

《Fourier Series Expansion Based Filter Parametrization for Equivariant Convolutions》这篇文章提出了一种基于傅里叶级数展开的滤波器参数化方法，旨在提升卷积神经网络（CNN）在处理具有对称性质（例如旋转、平移等）数据时的表现。文章的重点之一是提出了一种新的等变卷积技术，旨在通过有效的滤波器参数化来保持对称性和不变性，尤其是在图像处理、计算机视觉等领域中的应用。

等变卷积是一种卷积操作，其中卷积结果随着输入数据的对称变换而变化，但变换遵循特定的规律。例如，旋转等变卷积要求输入数据旋转时，卷积结果也应以相同的方式旋转。换句话说，卷积操作不仅对输入具有一定的特征提取能力，而且还能够保持对输入空间变换的“等变”特性，即输入经过某种变换后，输出的变换与输入的一致。

在此基础上，我的理解是：在使用等变卷积（Equivariant Convolutions）之后，如果输入的数据发生了旋转，模型依然可以输出正确的分类结果。这是因为等变卷积的设计目标之一就是确保卷积操作能够对输入数据的旋转（或其他对称变换）保持“等变”特性。具体来说，等变卷积确保卷积操作的输出随着输入数据的旋转（或其他变换）而进行相应的变换，而不是直接改变其类别标签。这意味着，旋转后的输入图像虽然在空间位置上有所变化，但它依然会通过等变卷积技术正确地识别出原始类别。

因此其直观地表述是：在分类任务中训练一个没有对训练数据进行旋转的模型，并且在测试中，输入的数据发生旋转，那么模型的判别性能会随之下降。但是在使用等变卷积之后，由于其具有旋转不变性，那么依然可以输出正确的分类结果。此次实验是基于这个想法去验证的。

2 网络模型：

此次实验共使用两个网络模型，一个是经过修改的LeNet，另一个是AlexNet。数据集使用MNIST。

在初期实验时候，使用了简化的LeNet以及对应修改的基于等变卷积的简化LeNet，如图1所示：

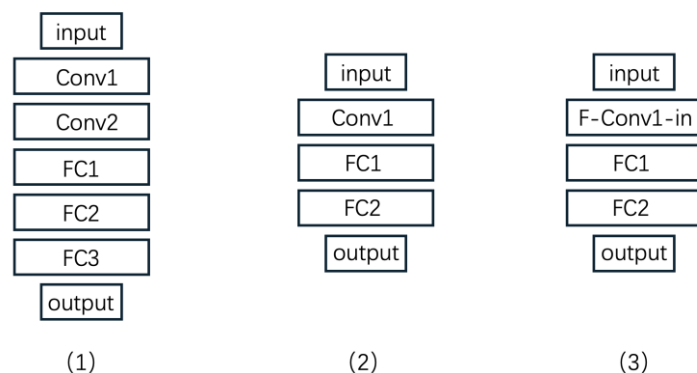


图1: LeNet, 简化版本LeNet以及基于等变卷积的简化LeNet

其中,左图为经典的LeNet,包含两个卷积层和三个全连接层。中间为经过简化后的LeNet,只包含一个卷积层和两个全连接层,右图为使用等变卷积的简化LeNet,将卷积层修改为等变卷积层。这样做的原因是在初期使用经典LeNet的时候发现,在MNIST数据集中分类准确率超过99%,我认为在这样的准确率下可能不能更好地展现等变卷积的优势,因此将网络进行缩减,以突出等变卷积对模型准确率识别的优势。

此外,根据我对等变卷积的理解,其可以有效应对目标旋转带来的性能下降的问题,因此我自然地将验证集和测试集进行了随机旋转而不对训练集进行旋转,在实验中发现识别性能并不好。另外在文章《Fourier Series Expansion Based Filter Parametrization for Equivariant Convolutions》的github (<https://github.com/XieQi2015/F-Conv>)提供的readme中提到三种对卷积层的不同设置(输入层、中间层、以及输出层)。然而,在简化的LeNet中,卷积层只有一层(具体的设置方式可以看实验部分)。我猜想可能是层数不够导致的性能差的问题(至少卷积层要有3层,来覆盖到github提到的所有等变卷积设置)。因此在此次实验中,又修改了一个网络: AlexNet。其中,原始的AlexNet有5个卷积层以及3个全连接层,如图2左图所示,那么,理所当然的,第一个卷积层就被替换为了等变卷积的输入层,中间3个卷积层替换成了等变卷积的中间层,最后一个卷积层替换为了等变卷积的输出层,如图2右图所示。

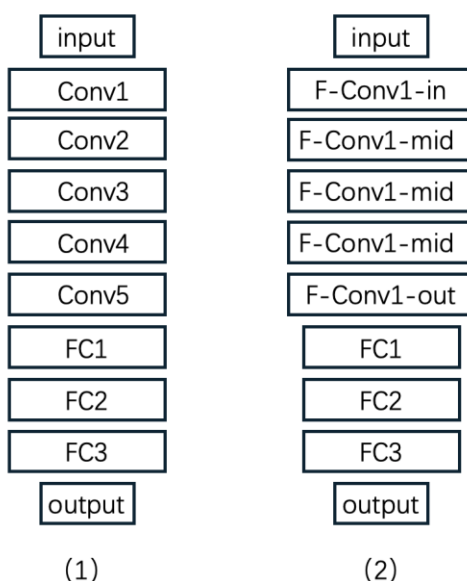


图2: AlexNet以及基于等变卷积的AlexNet

3 实验

实验设置：本次实验均在Nvidia RTX A6000上进行实验。为了保证实验结果的客观准确性，所有的实验均进行5次实验并取结果的均值。epoch设置为200，学习率都设置为lr=0.0002。具体的实验设置以及源代码已经公开在了我的github仓库中（<https://github.com/HaoYang-98/CLASS-Deep-Learning-and-Artificial-Intelligence>）。

实验结果：在LeNet中，网络模型使用简化LeNet（表2中用LeNet表示）以及使用等变卷积的简化LeNet（表2中用LeNet_F_Conv表示）。在AlexNet中，网络直接使用原始的AlexNet（表2中用AlexNet表示）以及将卷积层换成等变卷积的AlexNet（表2中用AlexNet_F_Conv表示）。实验一共分为四组，如表1所示，第一组数据集不对数据进行任何旋转，以测试模型的分类能力，第二组数据对训练集进行旋转数据增强，进一步看数据增强后的模型能力，第三组数据将验证集和测试集进行旋转数据扩充，但是不对训练集进行旋转，这样的话，训练好的模型会存在训练数据和测试数据的domain gap，其原因是数据发生了旋转，最重要的是，在第三组对比实验中，可以测试出等变卷积是否在对旋转数据能够进行有效识别。第四组实验将训练集、验证集和测试集都进行旋转，同样来表明模型在旋转手写数据集中的分类性能。

表1：MNIST数据集的不同设置

	训练集旋转	验证集旋转	测试集旋转
1	✗	✗	✗
2	✓	✗	✗
3	✗	✓	✓
4	✓	✓	✓

如表2所示，为所有网络模型在所有实验设置中的实验结果。表中部分异常数据已经通过红色标出，其原因可能是在训练随机初始化过程中失败导致。此外，LeNet和LeNet_F_Conv试验中，在通过大量实验可以看出，不管是在什么样的训练条件下，等变卷积网络似乎并没有普通的卷积操作好用，其实验结果都比普通的卷积要差。这可能是因为这个简化的网络只使用了等变卷积的输入层导致。

在此基础上，为了更好地应用等变卷积，我们使用AlexNet，增加了等变卷积的输入层、中间层以及输出层，实验结果同样如表2所示。

实验过程中的可视化结果：如图3所示，四种实验设置下，四个模型（LeNet、F_Conv_LeNet、AlexNet、F_Conv_AlexNet）的训练结果存在明显的差异。**实验1（无旋转增强）：**在没有旋转增强的情况下，所有模型的训练结果较为稳定，验证准确率迅速上升，但LeNet和AlexNet的训练曲线相对较平滑，而F_Conv_LeNet和F_Conv_AlexNet的曲线出现了波动，表明等变卷积对训练过程的影响较大。**实验2（仅训练集旋转增强）：**在只对训练集进行旋转增强的情况下，所有模型的训练损失下降速度较快，验证准确率逐渐提高。然而，使用等变卷积的模型（F_Conv_LeNet和F_Conv_AlexNet）相比普通卷积的模型（LeNet和AlexNet）在训练过程中的表现略逊色，尤其在验证集准确率的提升上。**实验3（验证集和测试集旋转）：**在验证集和测试集进行旋转增强时，模型训练过程中，LeNet和AlexNet仍然能够较好地收敛，而F_Conv_LeNet和F_Conv_AlexNet由于旋转增强带来的挑战，训练过程较为缓慢，且验证准确率较低，表明等变卷积在旋转不变性处理上可能存在一定的困难。**实验4（所有数据旋转）：**在所有数据进行旋转增强的情况下，训练过程的差异更加明显。LeNet和AlexNet在较短的时间内就取得了较高的验证准确率，而F_Conv_LeNet和

F_Conv_AlexNet 的训练曲线则表现为较慢的收敛速度和较差的验证性能，表明等变卷积在处理全面增强的情况下未能有效提升模型性能，反而在某些设置下导致了训练不稳定。

表2：所有的实验结果。其中红色字体为训练异常时候的结果，平均值没有将其放进计算。训练、验证和测试代表是否将训练集数据、验证集数据和测试集数据进行旋转。

		训练	验证	测试	1	2	3	4	5	平均
实验 1	LeNet				0.986	0.985	0.986	0.988	0.988	0.987
	LeNet_F_Conv	✗	✗	✗	0.970	0.982	0.981	0.975	0.984	0.978
	AlexNet				0.995	0.995	0.995	0.995	0.995	0.995
	AlexNet_F_Conv				0.992	0.990	0.989	0.990	0.989	0.990
实验 2	LeNet				0.984	0.984	0.984	0.983	0.985	0.984
	LeNet_F_Conv	✓	✗	✗	0.977	0.114	0.978	0.982	0.980	0.979
	AlexNet				0.994	0.994	0.994	0.994	0.996	0.994
	AlexNet_F_Conv				0.994	0.993	0.993	0.990	0.992	0.992
实验 3	LeNet				0.624	0.616	0.628	0.616	0.617	0.620
	LeNet_F_Conv	✗	✓	✓	0.610	0.614	0.615	0.609	0.114	0.612
	AlexNet				0.670	0.672	0.670	0.678	0.683	0.675
	AlexNet_F_Conv				0.677	0.666	0.662	0.671	0.681	0.671
实验 4	LeNet				0.980	0.977	0.979	0.979	0.978	0.979
	LeNet_F_Conv	✓	✓	✓	0.969	0.975	0.975	0.975	0.972	0.973
	AlexNet				0.994	0.993	0.993	0.993	0.993	0.993
	AlexNet_F_Conv				0.991	0.988	0.990	0.990	0.989	0.990

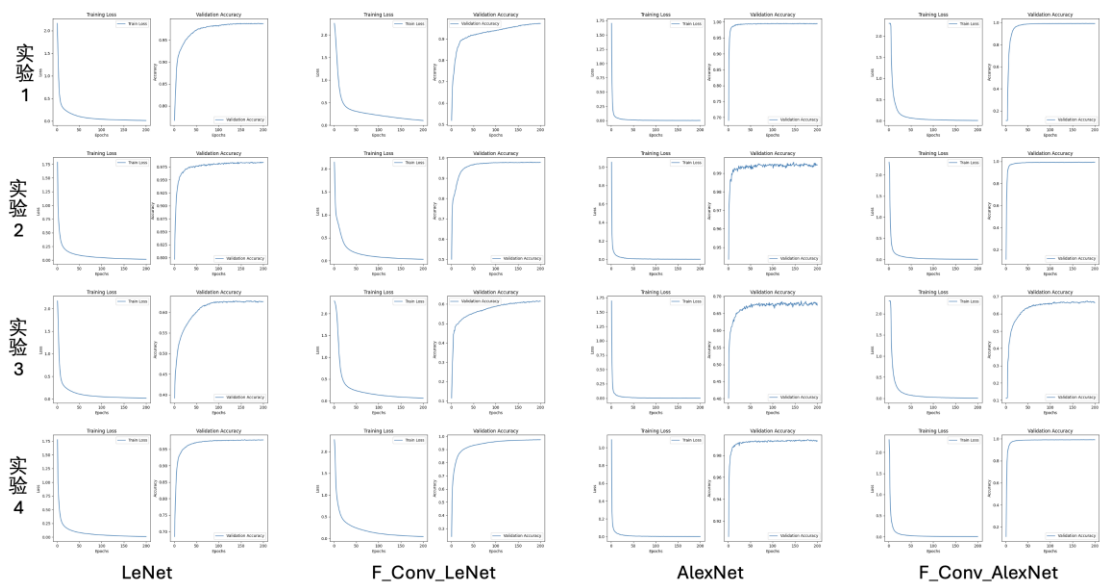


图3：实验过程中的训练损失和验证集准确率可视化。

参 考 文 献

- [1] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [2] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.
- [3] Xie Q, Zhao Q, Xu Z, et al. Fourier series expansion based filter parametrization for equivariant convolutions[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(4): 4537-4551.