基于 GitHub 公开神经网络模型库的调研

孟令涵　DZ1933020

1 背景

随着深度神经网络（DNN）模型在软件系统中越来越多的应用，它们已经逐渐成为了软件系统中不可或缺的组成部分。深度学习在一些对于人类来说难以准确描述的任务上取得了巨大的成功，比如图像分类或者语音识别；包括在一些安全领域的软件系统也充斥着深度神经网络，如医疗诊断和自动驾驶等。与传统软件组件的共享和重用相似，工程师们也会共享模型，以便在各种应用中重用，如 python 的人脸识别包在很多应用中都有使用。

但是，就像软件系统不可避免地有缺陷一样，模型也是有缺陷的，从而会导致出现分类或者预测的精度偏差。然而与软件不同的是，模型缺陷并不能通过直接修改模型轻易修复，因此对于 AI 模型的测试和修复也成为了新的研究热点，AI 模型的训练，调优和维护也逐渐成为软件工程中的一个重要步骤。

根据目前已经了解的关于 AI 模型测试的研究，大致有两种研究方向——effectiveness 和 efficiency。对于 AI 模型测试来说，effectiveness 可以理解为提升模型精度，在《MODE: Automated Neural Network Model Debugging via State Differential Analysis and Input Selection》一文中，将模型的精度损失称之为模型缺陷，模型缺陷可以分为两种，一种是由次优模型结构引起的，成为结构缺陷；另一种是由错误的训练过程引起的，称之为训练缺陷，这篇文章中主要侧重于训练缺陷。论文中首先进行模型的状态差异分析来识别导致模型错误的内部特征，生成热图，热图显示了重新训练的目标特征，根据热图选择现有的或者新的输入，以生成新的训练数据集，然后用于重新训练模型并修复错误。论文在三个数据集上（MNIST,FASHION-MNIST,CIFAR-10）上进行了实验，并选取了六个已训练好的模型，将其与以 MODE 方法选取的训练样本训练出的模型进行对比，模型精度都有显著的提升。

Table 3: Real-world Models Bug Fix

DataSet	Model	Original Acc.	# Samples	MODE Acc.	Random Acc.
MNIST	MNIST-10 [23]	95.2%	2000	97.4%	94.8%
	MNIST-11 [23]	93.4%	2000	96.8%	94.3%
Fashion MNIST	FM-7 [15]	87.6%	2000	92.3%	88.9%
	FM-8 [15]	91.6%	2000	92.6%	88.5%
CIFAR	CIFAR-6 [5]	87.3%	4000	93.2%	87.3%
	CIFAR-7 [5]	88.4%	4000	92.8%	88.2%

还有一个角度就是 efficiency，即以更小的采样成本来达到测试的目的。

在真实的场景下，对于采样的标记成本可能是巨大的，比如在医疗中对某些样本进行标记可能需要侵入性活检，那么这样的标记应该尽可能少，所以我们应该在保证测试准确的情况下，尽可能减少需要进行标记的样本（减少标记成本）。在《Boosting Operational DNN Testing Efficiency through Conditioning》一文中，作者利用最后一个隐藏层的神经元分布来指导采样，采样在最后一个隐藏层中神经元的空间分布中具有代表性，基于此提出了一种有效的神经网络操作测试方法，文中基于交叉熵最小化来实现这种方法的采样，这种方法与随机采样相比只需要一半的标记就可以达到相同的精度水平。

对于 AI 模型测试的研究是与神经网络模型分不开的，因此希望通过本次报告对目前已存在的神经网络测试数据集及相关的模型进行调研。

2 问题定义

本次报告希望达到以下的目的：

1）通过本次报告总结在神经网络模型测试相关研究中常用的数据集，了解数据集的相关信息。

2）通过本次报告总结近年来深度学习模型研究的相关内容，并基于 GitHub 收集相关模型。

3）通过本次报告对于 python 的 Keras 机器学习框架有初步的了解。

3 神经网络公开数据集介绍

在本节中，将对在神经网络模型测试相关研究中常用的数据集进行总结和介绍。

3.1 MNIST 数据集

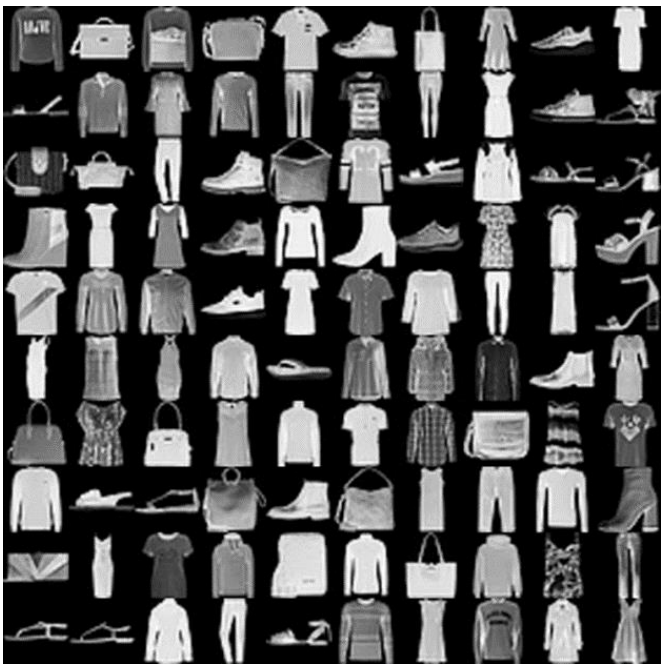
MNIST 数据集是机器学习的最经典数据集之一，最早于 1998 年在论文《Gradient-based learning applied to document recognition.》中提出，这个数据集是一个大型手写数字数据库，由 250 个不同人手写的数字构成，其中包含了 0-9 共 10 类手写数字图片，且每张图片都进行了归一化的处理，都是 28*28 像素大小的灰度图，每张图片中像素值的大小在 0-255 之间，0 为黑色，255 为白色。

MNIST 数据集中共包含 70000 张手写数字图片，其中有 60000 张用作训练集，10000 张用作测试集，原始数据可以在 http://yann.lecun.com/exdb/mnist/下载。

3.2 Fashion-MNIST 数据集

2017 年 8 月 27 日，Fashion-MNIST 图片库在 GitHub 上开源，Fashion-MNIST 几乎克隆了 MNIST 的所有外在特征：

- 60000 张训练图像和对应 Label；
- 10000 张测试图像和对应 Label；
- 10 个类别；
- 每张图像 28x28 的分辨率；

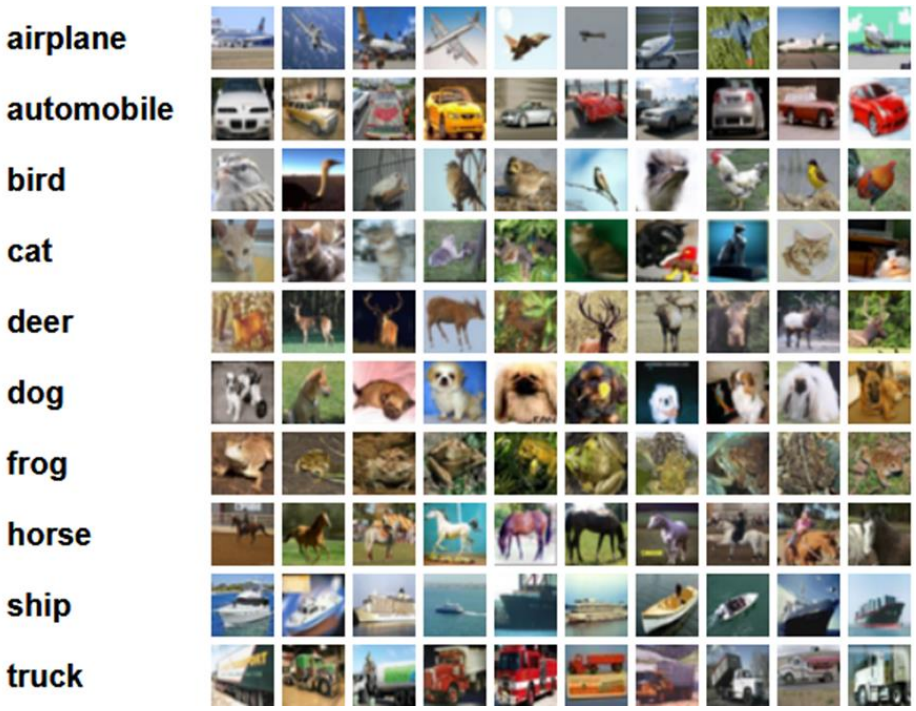


Fashion-MNIST 中的图片同样是灰度图，但是与 MNIST 不同的是，Fashion-MNIST 不再是抽象符号的图片集合，而是更加具象化的人类必需品——服装，共十大类：

Label	Description
0	T恤（T-shirt/top）
1	裤子（Trouser）
2	套头衫（Pullover）
3	连衣裙（Dress）
4	外套（Coat）
5	凉鞋（Sandal）
6	衬衫（Shirt）
7	运动鞋（Sneaker）
8	包（Bag）
9	靴子（Ankle boot）

3.3 CIFAR-10 数据集

CIFAR-10 是由 Hinton 的学生 Alex Krizhevsky 和 Ilya Sutskever 整理的一个用于识别普适物体的小型数据集。其中包含 10 个类别的 RGB 彩色图片：飞机（ airplane ）、汽车（ automobile ）、鸟类（ bird ）、猫（ cat ）、鹿（ deer ）、狗（ dog ）、蛙类（ frog ）、马（ horse ）、船（ ship ）和卡车（ truck ）。



其中每张图片的尺寸是 32*32 像素，每个类别有 6000 张图像，数据集一共有 5 万张训练图片和一万张测试图片。

与 MNIST 数据集相比，CIFAR-10 有以下不同：1）CIFAR-10 是 3 通道的彩色 RGB 图片，而 MNIST 是灰度图像；2）CIFAR-10 的图片尺寸比 MNIST 稍大；3）相比于手写字符，CIFAR-10 中的图片是现实世界中的真实物体，噪声更大，识别起来更加困难。

3.4 CIFAR-100 数据集

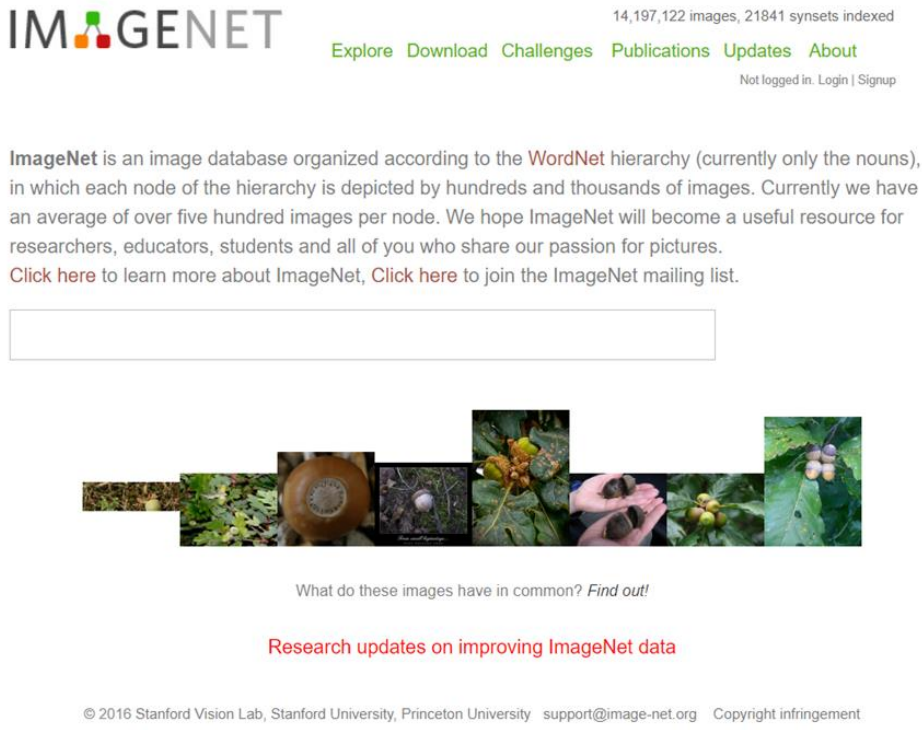
CIFAR-100 与 CIFAR-10 类似，同样是由 60000 张彩色图片组成，但是它的类别达到了 100 个，从属于 20 个超类。每个类别中包含 600 张图像，每类有 500 张训练图像和 100 张测试图像。同样的，每张图像的尺寸仍然是 32*32 像素，CIFAR-100 的类别列表如下：

超类	类别
水生哺乳动物	海狸，海豚，水獭，海豹，鲸鱼
鱼	水族馆的鱼，比目鱼，射线，鲨鱼，鳟鱼
花卉	兰花，罂粟花，玫瑰，向日葵，郁金香
食品容器	瓶子，碗，罐子，杯子，盘子
水果和蔬菜	苹果，蘑菇，橘子，梨，甜椒
家用电器	时钟，电脑键盘，台灯，电话机，电视机
家用家具	床，椅子，沙发，桌子，衣柜
昆虫	蜜蜂，甲虫，蝴蝶，毛虫，蟑螂
大型食肉动物	熊，豹，狮子，老虎，狼
大型人造户外用品	桥，城堡，房子，路，摩天大楼
大自然的户外场景	云，森林，山，平原，海
大杂食动物和食草动物	骆驼，牛，黑猩猩，大象，袋鼠
中型哺乳动物	狐狸，豪猪，负鼠，浣熊，臭鼬
非昆虫无脊椎动物	螃蟹，龙虾，蜗牛，蜘蛛，蠕虫
人	宝贝，男孩，女孩，男人，女人
爬行动物	鳄鱼，恐龙，蜥蜴，蛇，乌龟
小型哺乳动物	仓鼠，老鼠，兔子，母老虎，松鼠
树木	枫树，橡树，棕榈，松树，柳树
车辆1	自行车，公共汽车，摩托车，皮卡车，火车
车辆2	割草机，火箭，有轨电车，坦克，拖拉机

3.5 ImageNet 数据集

与前面所述的数据集相比，ImageNet 更加庞大，甚至不在一个数量级上。目前其官网上列出 ImageNet 共有 14197122 幅图像，总共分为 21841 个类别。

ImageNet 数据集最早由 2009 年提出，曾于 2010 年开始每年举办 ImageNet 大规模视觉识别挑战赛（ILSVRC），共持续 7 届。ImageNet 是根据 WordNet 层次结构组织的图像数据集，WordNet 是一个由普林斯顿大学认识科学实验室在心理学教授乔治·A·米勒的指导下建立和维护的英语字典。WordNet 根据词条的意义将它们分组，每一个具有相同意义的字条组称为一个 synset(同义词集合)。WordNet 为每一个 synset 提供了简短，概要的定义，并记录不同 synset 之间的语义关系。ImageNet 数据集每个节点含有至少 500 个对应物体的可供训练的图片/图像。



4 深度学习模型调研

在本节中，将对深度学习模型进行相应的介绍。

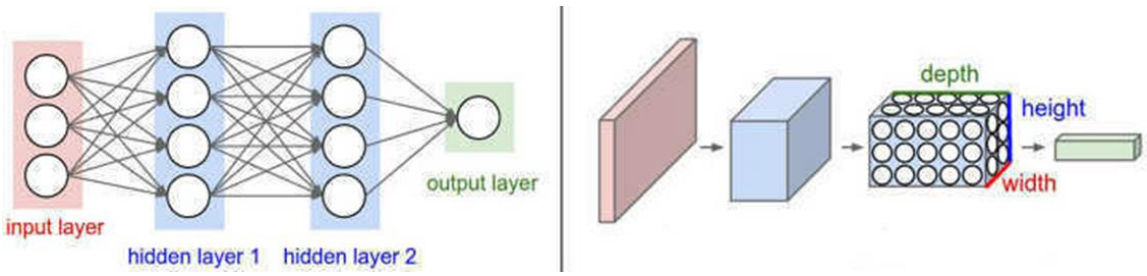
4.1 卷积神经网络

在介绍模型之前，首先介绍一下卷积神经网络。

在处理图像问题上，卷积神经网络有很大的优势。如果用全连接神经网络处理大尺寸图像会有明显的缺点：

- 将图像展开为向量会丢失空间信息；
- 参数过多效率低下，训练困难；
- 大量的参数可能会导致网络过拟合。

而卷积网络与常规神经网络不同，它各层的神经元是三维排列的：宽度，高度，深度。其中深度指的是激活数据体的第三个维度，而不是整个网络的深度，整个网络的深度指的是网络层数。举个例子，以 CIFAR-10 中的图像作为卷积神经网络的输入，该输入数据体的维度是 32*32*3（宽度，高度和深度）。



上图是常规神经网络和卷积神经网络的对比，我们可以看出，层中神经元只与前一层中的一小块区域连接，而不是像常规神经网络中的全连接方式。也因此，卷积神经网络在视觉处理中更具优势，目前常用的模型也都是基于卷积神经网络。

4.2 VGG 模型

VGG 是 Oxford 的 Visual Geometry Group 的组提出的（《Very Deep Convolutional Networks for Large-Scale Image Recognition》）。该网络是在 ILSVRC 2014 上的相关工作，主要工作是证明了增加网络的深度能够在一定程度上影响网络最终的性能。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers

论文中测试了在模型其他参数尽可能一致的情况下，增加模型深度对于模型性能的影响，从 11 层至 19 层进行了 6 次实验，其中效果较好的是 VGG16 和 VGG19.

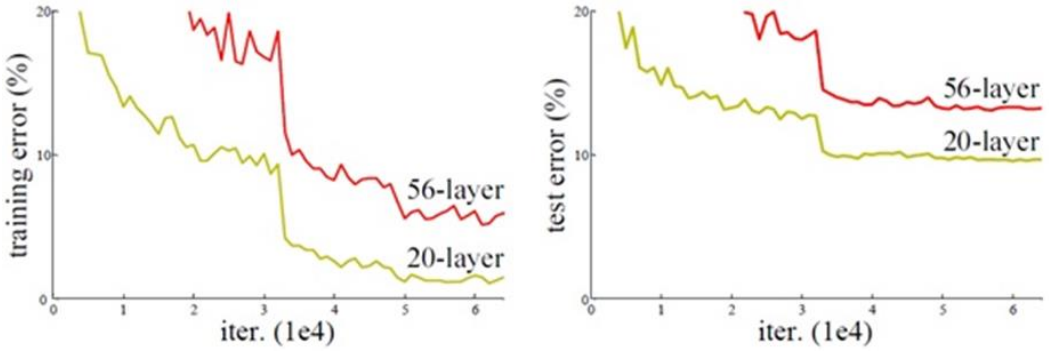
VGG 的结构非常简洁，整个网络都使用了同样大小的卷积核尺寸（3*3）和最大池化尺寸（2*2）；实验中验证几个小滤波器（3*3）卷积层的组合要比一个大滤波器（5*5 和 7*7）卷积层好，通过不断加深网络结构可以提升性能。

但是 VGG 模型消耗了更多的计算资源，使用了大量的参数，参数达到 140M，引起了更多的内存占用。

4.3 ResNet 模型

网络的深度对于模型的性能至关重要，当网络层数增加后，网络可以进行更加复杂的特征提取，从而使模型有更好的预测结果。但是，也有实验表明，网络深度增加时，网络的准确度会出现饱和甚至出现下降情况，这就是深度网络的退化问题。

如下图，56 层的网络比 20 层的效果还要差。而且这不是由过拟合导致的，因为训练导致的精度缺失同样很高。



深度网络的退化问题说明深度网络不容易训练，为了解决这个问题，论文中提出了残差学习的方法。ResNet 网络是参考了 VGG19 网络，在其基础上进行了修改，并通过短路机制加入了残差单元。最终在 ImageNet 数据集上进行测试时，ResNet 的误差为 3.57，且层数达到了 152 层，与 VGG 的层数已不在一个量级上，在 2015 年的 ImageNet 挑战赛上获得了 5 项第一。

5 Keras 框架介绍

目前 GitHub 上关于深度学习的模型，很多都与 Keras 密不可分。Keras 是基于 Tensorflow 的一个深度学习库，由 python 语言编写而成的高层神经网络 API，可以进行深度学习模型的设计、调试、评估、应用和可视化。

Tensorflow 是 Google 开源的基于数据流图的机器学习框架，支持 python 和 c++程序开发语言。tensor 意为张量（即多维数组），flow 意为流动。即多维数组从数据流图一端流动到另一端（如图中输入的数据经过模型的各个隐藏层流入到输出层）。

Keras 是为了支持快速实践而对 Tensorflow 的再次封装，让我们可以不用关注过多的底层细节，能够把想法快速转换为结果。

在 Keras 包中包含了之前所述的 MNIST，Fashion-MNIST，CIFAR-10，CIFAR-100 四个图像类的数据集，方便在训练和测试模型时调用。同时，Keras 建立模型也十分便捷，如下：

```
#最简单的序贯模型，序贯模型是多个网络层的线性堆叠
simple_model=Sequential()
#dense层为全连接层
#第一层隐含层为全连接层 5个神经元 输入数据的维度为3
simple_model.add(Dense(5,input_dim=3,activation='relu'))
#第二个隐含层 4个神经元
simple_model.add(Dense(4,activation='relu'))
#输出层为1个神经元
simple_model.add(Dense(1,activation='sigmoid'))
```

建立序贯模型，通过两行代码即可建立两个隐藏层，加上输出层即可完成

一个简单神经网络模型的构建，在 GitHub 中也有大量基于 Keras 构建的深度学习模型。

参考文献

1. Zenan Li, Xiaoxing Ma, Chang Xu, Chun Cao, Jingwei Xu, Jian Lü: Boosting operational DNN testing efficiency through conditioning. ESEC/SIGSOFT FSE 2019: 499-509
2. Xuanyi Dong, Yi Yang:NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search. CoRR abs/2001.00326 (2020)
3. Shiqing Ma, Yingqi Liu, Wen-Chuan Lee, Xiangyu Zhang, Ananth Grama:MODE: automated neural network model debugging via state differential analysis and input selection. ESEC/SIGSOFT FSE 2018: 175-186
4. Han Xiao, Kashif Rasul, Roland Vollgraf:Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. CoRR abs/1708.07747 (2017)
5. Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009
6. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database.InComputer Vision and Pattern Recognition, 2009. CVPR2009. IEEE Conference on, pages 248–255, June 2009.
7. Karen Simonyan, Andrew Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015
8. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Deep Residual Learning for Image Recognition. CoRR abs/1512.03385 (2015)