南京大學

研 究 生 毕 业 论 文

（申请硕士学位）

论 文 题 目 基于动力系统的深度神经网络优化过程分析

作 者 姓 名 万俊

学科、专业名称 计算数学

研 究 方 向 深度神经网络

指 导 教 师 陆宏 副教授

2019 年 5 月 8 日

学　　号 ： **MG1621022**

论文答辩日期 ： **2019 年 5 月 11 日**

指 导 教 师 ：　　（签字）

Analysis of Deep Neural Network Optimization Process Based on Dynamic System

By

Jun Wan

Supervised by

Associate Professor **Hong Lu**

A Thesis
Submitted to the Faculty of Mathematics
and the Graduate School
of Nanjing University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science

Faculty of Mathematics

May 2019

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于动力系统的深度神经网络优化过程分析

计算数学 专业 2016 级硕士生姓名：万俊

指导教师（姓名、职称）：陆宏 副教授

摘 要

我们正在进入深度神经网络的时代。深度神经网络已经成功的应用到了非常多的领域，从图像识别到目标定位，从语音识别到机器翻译。虽然深度神经网络在很多的任务上已达到甚至超过了人类的水平，但直到现在，深度神经网络一直面临着黑盒智能、可解释性差等质疑，人们对深度神经网络的优化过程的理解仍然存在争议。目前深度神经网络的可解释性一直是学术界研究的重点。随着相关研究的不断深入，深度神经网络的优化过程的重要性已逐步显现出来，从数学角度解释深度神经网络优化过程是打开深度神经网络黑盒非常重要的一步。目前，人们通过与不同的数学方法相结合，提出了很多不同的相关理论。尽管如此，这些理论对某些深度神经网络优化过程的认识仍然不够深入。

本文主要从动力系统的角度来解释深度神经网络的优化过程。一方面在LandScape猜想的背景下，对深度神经网络的损失函数的梯度引入利普西斯连续性假设，通过动力系统流形稳定性定理建立起全量梯度下降法和动力系统方程之间的联系，以此分析全量梯度下降法的收敛性。另一方面结合随机动力系统方程的转移密度两歧性定理和动力系统方程的数值计算方法对随机梯度下降法和动量梯度下降法进行研究，并以此为基础，提出了一系列新的深度神经网络优化算法。在实例分析部分，本文搭建了三层的全连接神经网络模型和七层的卷积神经网络模型，基于Fashion-Mnist数据集对新的优化算法进行了实验分析，最后就实验结果与现有的一些传统优化算法的结果进行了对比。

关键词： 深度神经网络，动力系统，数值算法

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Analysis of Deep Neural Network Optimization Process Based on Dynamic System

SPECIALIZATION: Computational Mathematics

POSTGRADUATE: Jun Wan

MENTOR: Associate Professor Hong Lu

Abstract

We are entering the era of deep neural networks. Deep neural networks have been successfully applied in a wide range of fields, from image recognition to target location, from speech recognition to machine translation. Although deep neural networks have reached or exceeded human levels in many tasks, until now, deep neural networks have been faced with black box intelligence and poor interpretability. There are still many disputes in the understanding of the optimization process of deep neural networks. The interpretability of deep neural networks has always been the focus of academic research. With the deepening of relevant research, the importance of the optimization process of deep neural networks has gradually emerged. Explaining the optimization process of deep neural networks from a mathematical perspective is a very important step to open the black box of deep neural networks. At present, people have proposed many different related theories by combining with different mathematical methods. Nevertheless, these theories are still not deep enough to understand the optimization process of some deep neural networks.

This paper mainly explains the optimization process of deep neural network from the perspective of dynamic system. On the one hand, in the context of LandScape's conjecture, the gradient of the loss function of the deep neural network is introduced into the Lipschitz continuity hypothesis, and the relationship between the full gradient descent method and the dynamical system equation is established by the dynamic system manifold stability theorem. In this way, the convergence of the full gradient descent method is analyzed. On the other hand, combined with the transfer density two-discrimination theorem of stochastic dynamical equations and the numerical calculation analysis of dynamical system equations, the stochastic gradient descent method

and the momentum gradient descent method are studied. Based on these, a series of new optimization algorithm are proposed. In the experimental part, this paper builds a three-layer fully connected neural network model and a seven-layer convolutional neural network model, and then analyzes the new optimization algorithm based on the Fashion-Mnist data set. Finally, the results are compared with the results of some existing traditional optimization algorithms.

Keywords: Deep neural network, Dynamical system, Numerical algorithm

目　　录

插　　图

第一章　绪论

1.1　研究背景与意义

我们正在进入深度神经网络的时代，深度神经网络已在许多领域得到广泛应用。深度神经网络的著名应用包括图像分类 [He 等, 2016]、语音识别 [Xiong 等, 2016]、目标跟踪 [Liu 等, 2016]、自动驾驶 [Bojarski 等, 2016]等。虽然对于一些定义良好的任务，例如Alpha-Go [Silver 等, 2016]，深度神经网络已达到甚至超过了人类的水平，但直到现在，深度神经网络一直面临着黑盒智能、可解释性差等质疑，人们对深度神经网络的优化过程的理解仍然存在争议。目前深度神经网络的可解释性一直是学术界研究的重点。随着相关研究的不断深入，深度神经网络的优化过程的重要性已逐步显现出来，从数学角度解释深度神经网络的优化过程是打开深度神经网络黑盒非常重要的一步。

深度神经网络的优化过程是非常核心的步骤。构建一个深度神经网络的步骤可分为两步。第一步是根据具体的任务搭建网络结构，如图像处理任务中常用的卷积神经网络结构，自然语言处理任务中常用的递归神经网络结构。第二步则是在搭建好的网络结构上选择合适的优化算法对深度神经网络的参数进行优化更新。由此可见深度神经网络的优化过程是非常重要的，直接关系到模型结果的好坏。目前开发针对深度神经网络的新的优化算法仍然以实验为导向，如果能从理论上对深度神经网络的优化过程进行分析，那么就有可能基于理论来指导我们开发出新的优化算法。目前，人们通过将深度神经网络的优化过程与不同的数学方法相结合，也提出了很多不同的相关理论。尽管如此，这些理论对某些深度神经网络的优化过程的认识仍然不够深入。

深度神经网络的优化过程是非常复杂的过程。深度神经网络通常是由复合的高维的非线性函数组成，从纯优化的角度上来看，它要求我们处理的是棘手的高维优化问题，而且通常这种优化问题是一个非凸优化问题。传统的研究方法会降低深度神经网络的深度或是去掉其非线性的激活函数再进行研究，然而深度神经网络的成功却恰恰在于其深度以及激活函数的非线性，所以本文将在保留深度神经网络的深度以及非线性的前提上进行研究。为了能使用更多的数学工具对这种高维的非线性系统进行分析，本文将引入一些数学假设，赋予深度神经网络更多的数学性质，以方便使用更多的数学定理。本文将通过动力系统的相关理论来研究主流的深度神经网络优化算法。从动力系统角度重新审视主流的优化算法，以此提出新的深度神经网络优化算法。在实例分析中，本文

构建了三层的全连接神经网络和七层的卷积神经网络，并结合开源数据集对新的优化算法的有效性进行了检验。

1.2　相关研究现状

Baldi [Baldi 和 Hornik, 1989]于1989年提出LandScape猜想，LandScape猜想是指在深度神经网络的优化问题中，次优的临界点的海森矩阵非常可能有负的特征值。换句话说没有糟糕的局部极小值，而且几乎所有的鞍点都是严格的鞍点。Rong Ge，Ju Sun等人 [Sun 等, 2017] [Sun 等, 2018] [Ge 等, 2015]对LandScape猜想进行了实证性分析，并发现在某些特殊条件下，深度神经网络的确具有LandScape猜想的性质。Saxe等人 [Saxe 等, 2013] 从动力系统的角度对深度线性神经网络的优化过程进行了研究，通过研究表明，深度线性网络模型的优化表现出与深度非线性网络模型优化相似的性质，他们还进一步对随机正交初始化参数进行了研究。Su W等人 [Su 等, 2014]发现对于非凸优化问题中常用的nesterov算法实际上对应着一个动力系统方程的离散数值解，他们通过构造能量函数的方式，利用李雅普诺夫分析的方法证明了该优化算法的收敛性。Emmanuel J等人 [Candes 等, 2015]发现对于某些非凸优化问题，引入特殊的初始化条件后，使用梯度下降法能得到该问题的全局最小值。Weinan等人 [Weinan, 2017]阐述了使用连续动力系统来模拟机器学习中经常使用的高维非线性函数的想法，他们还特别着重讨论了连续动力系统与深度神经网络之间的联系。

在本文中，我们在LandScape [Baldi 和 Hornik, 1989]猜想的假设前提下，将深度神经网络的优化过程看作一个动力系统问题，引入一定的数学假设，通过相应的数学定理和数值计算方法对深度神经网络的优化过程进行分析和研究。

1.3　本文主要内容和创新

本文主要从动力系统的角度来解释深度神经网络的优化过程。一方面在LandScape猜想的背景下，对深度神经网络的损失函数的梯度引入利普西斯连续性假设，通过动力系统流形稳定性定理建立起全量梯度下降法和动力系统方程之间的联系，以此分析全量梯度下降法的收敛性。另一方面结合随机动力系统方程的转移密度两歧性定理和动力系统方程的数值计算方法对随机梯度下降法和动量梯度下降法进行了研究，并在此基础上提出了一系列新的深度神经网络优化算法。在实验部分，本文了搭建了三层的全连接神经网络模型和七层的卷积神经网络模型，然后基于Fashion-Mnist数据集对新的优化算法进行了实验

分析，最后就实验结果与现有的一些传统优化算法的结果进行了比较。本文的主要创新之处是建立起动力系统与深度神经网络优化算法的联系，通过动力系统的相关理论来对主流的深度神经网络优化算法进行研究分析，并以此提出新的优化算法。

具体结构内容安排如下：

第一章，绪论。主要阐述选题的研究意义和相关背景，并列出本文研究的内容和结构上的安排。

第二章，深度神经网络的概念与方法。主要介绍了深度神经网络的相关定义以及部分性质，为进一步的理论研究打好基础。

第三章，基于动力系统的梯度下降法。主要说明了动力系统与梯度下降法之间的关系，介绍了动力系统中的相关概念和定理，通过流形稳定性定理分析梯度下降法的收敛性。

第四章，基于随机动力系统的随机梯度下降法。主要介绍了随机动力系统的相关概念和定理。对随机梯度下降法进行数学建模，通过转移密度两歧性定理对随机梯度下降法进行理论分析。

第五章，基于动力系统的动量梯度下降法。主要介绍了主流的动量梯度下降法，通过动力系统方程的数值差分法研究动量梯度下降法与动力系统方程之间的关系。

第六章，RK与Adams神经网络优化算法。将动力系统的经典数值解法引入到深度神经网络的优化过程中，提出新的深度神经网络优化算法。

第七章，实例分析。搭建了三层的全连接神经网络模型和七层的卷积神经网络模型，针对前面几小节提出的新的优化算法，结合Fashion-Mnist数据集进行了对比实验。

第二章　深度神经网络的概念与方法

2.1　深度神经网络定义

深度前馈网络也叫做前馈神经网络或者多层感知机，是一种非常典型和常见的深度神经网络模型，这种模型被称为是前向的，是因为数据x通过网络的中间计算后得到最终输出值y，在模型的输出和模型本身之间没有反馈连接。如没有特殊说明，后面的深度神经网络均特指深度前馈神经网络。深度神经网络的目标是近似某个函数f^*，例如对于一个分类问题，$y = f^*(x)$将输入x映射到一个类别y。深度神经网络定义了一个映射$y = f(x;\theta)$，通过学习参数θ的值，使得它能够得到比较好的函数近似。深度神经网络的详细数学定义如下：

定义2.1.1：假设X为数据的输入空间，Y为数据的输出空间，函数族F是定义域在X上，值域在Y上的映射的集合：

$$F = \{f|Y = f_\theta(X)\}$$

$$f_\theta(x) = g_m(\theta_m \cdot (\cdots g_2(\theta_2 \cdot g_1(\theta_1 \cdot x))\cdots)$$

其中$x \in X$是输入数据（通常是高维数据），θ_i是第i层的参数，g_i是第i层的非线性的激活函数。对于形如F的函数族，称为深度神经网络。

非线性的激活函数g_i的引入使得函数族F的表征能力大大的增强。Cybenko等人 [Cybenko, 1989]已经证明对于任意的波莱尔可测函数g，总能找到一个函数族F，使得F中的某个函数f_{θ^*}和g足够的接近。这种逼近性质是由嵌套的非线性激活函数来保证的。在深度神经网络中，常用的非线性激活函数有sigmoid函数，tanh函数以及relu函数等。

sigmoid函数：

$$g_i(z) = \frac{1}{1 + e^{-z}}$$

tanh函数:

$$g_i(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

relu函数:

$$g_i(z) = \max(z, 0)$$

深度神经网络通常应用于监督学习中。假设对某事件进行统计，统计到数据集$T = \{x_i, f(x_i)\}_{i=1}^n$，其中$x_i$是第$i$次采样到的样本的特征，$f(x_i)$是对应样本的标签，$f$是真实的映射函数。我们期望从2.1.1 所定义的函数族中找到一个比较好的函数f'，使得$f' \approx f$，$f' \in F$。

定义2.1.2：给定一个采样到的数据集$T = \{x_i, f(x_i)\}_{i=1}^n$，假设$\{x_i, f(x_i)\}$采样自分布$P(X, f(X))$，称以下函数为期望风险函数：

$$L_{exp}(f') = \int_{X \times f(X)} l(f'(x), f(x)) P(x, f(x)) dx df(x)$$

其中l为损失函数，度量单个样本的损失。例如对于二分类问题中的01损失函数，如果该样本被预测正确，则该样本的损失函数为0，否则为1。$P(x, f(x))$表示样本对$(x, f(x))$存在的概率。

定义2.1.3：给定一个采样到的数据集$T = \{x_i, f(x_i)\}_{i=1}^n$，称以下函数为经验风险函数：

$$L(f') = \frac{1}{n} \sum_{i=1}^{n} l(f'(x_i), f(x_i))$$

经验风险$L(f')$是函数f'关于数据集的平均损失。当假设数据集T中的数据是独立同分布采样出来的时候，根据大数定律，经验风险函数趋近于期望风险函数。经验风险函数可以认为度量了函数f'在数据集T上相对于f的一个相似程度。当$f' \in F$时，$L(f')$简记为$L(\theta)$，$l(f'(x_i), f(x_i))$简记为$l(x_i, \theta)$。

根据上面定义，找到一个比较好的函数使得$f' \approx f$，$f' \in F$。等价于求解下述优化问题：

$$\min_{\theta} L(\theta)$$

2.2　深度神经网络基本性质

在讨论深度神经网络的一些基本性质之前，本文有如下定义:

定义2.2.1：对于$L(\theta)$中的某个点θ^*。

1. 称θ^*是函数$L(\theta)$的驻点，如果$\nabla L(\theta^*) = 0$。

2. 称驻点θ^*是局部极小值，如果存在以θ^*为中心的小邻域U_δ，使得$f(\theta^*) \leq f(\theta)$，$\forall \theta \in U_\delta$，反之称$\theta^*$为局部极大值，如果$f(\theta^*) \geq f(\theta)$，$\forall \theta \in U_\delta$。

3. 称驻点θ^*是鞍点，如果对任意的以θ^*为中心的小邻域U_δ，都$\exists \theta'$，$\theta'' \in U_\delta$，使得$f(\theta'') \leq f(\theta^*) \leq f(\theta')$。

定义2.2.2： 称驻点θ^*是$L(\theta)$的严格鞍点，如果$L(\theta)$在θ^* 处的海森矩阵至少有一个负的特征值，即$\lambda_{min}(\nabla^2 L(\theta^*)) < 0$。

命题2.2.3： 一般情况下

$$\min_{\theta} L(\theta)$$

是一个非凸优化问题。

实际上并不是所有的深度神经网络都是非凸优化问题，对于一些特殊结构的深度神经网络，他们可能是凸优化问题，但在大部分情况下$L(\theta)$的优化问题是一个非凸问题。Auer等人 [Auer 等, 1996]首先说明了在单神经元节点情况下，如果损失函数使用L_2作为损失函数，激活函数连续且有界的，那么对应的优化问题将有指数级的局部极小值，即一个非常复杂的非凸优化问题。

下面从一个直观的角度来观察为什么大部分情况下深度神经网络对应的优化问题是非凸优化问题。

假设有一组参数θ'使得$L(\theta)$达到最小值L_{min}。根据深度神经网络的定义，交换θ'中第k层参数中顺序和对应$k+1$层参数顺序得到θ''，如图2.1所示。

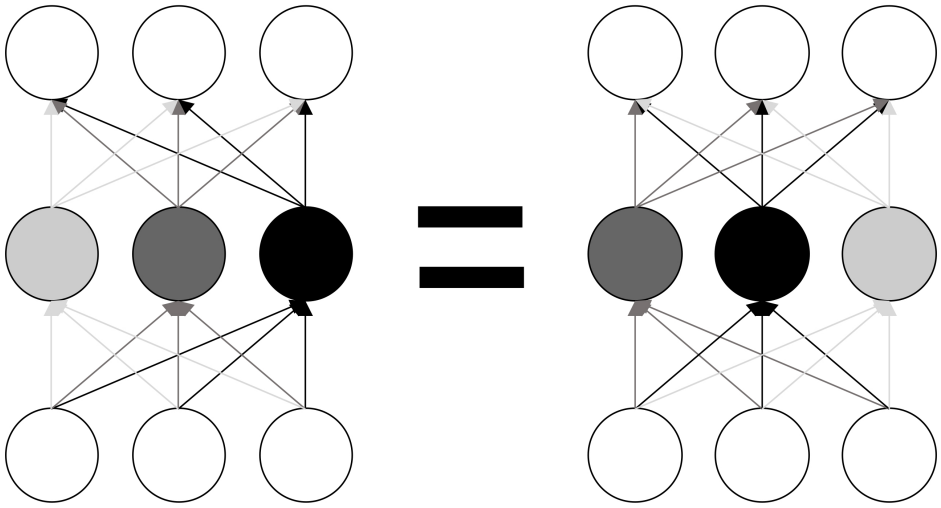

图 2.1: 神经网络参数交换

由于深度神经网络结构的对称性，显然$L(\theta') = L(\theta'')$，不断重复上述过程。对于$L(\theta)$，可以构造出指数级的集合T_θ，使得它们中的元素均达到$L(\theta)$ 的最小值。假设$L(\theta)$是凸函数，则由凸函数性质：

$$L(p\theta' + (1-p)\theta'') \leq pL(\theta') + (1-p)L(\theta'') = L_{min} \quad \forall p \in (0,1) \quad \forall \theta'\ \theta'' \in T_\theta$$

一般情况下很难使得$\forall p \in (0,1)$，$\forall \theta'\ \theta'' \in T_\theta$，$L(p\theta' + (1-p)\theta'') = L_{min}$，所以一般$L(\theta)$是非凸函数。

命题2.2.4：一般情况下，由于$L(\theta)$是非凸函数，所以函数$L(\theta)$有大量的驻点，这些驻点可能是局部极小值，可能是局部极大值，也可能是鞍点。

LandScape猜想：$L(\theta)$的次优临界点的海森矩阵非常有可能有负的特征值，换句话说，几乎所有的鞍点都是严格鞍点。

2.3　深度神经网络建模过程

深度神经网络建模过程包含深度神经网络结构设计、损失函数设计、优化方法选择、参数优化等几个步骤。针对监督学习任务，深度神经网络的建模过程可表示为算法1所示。

Algorithm 1: 深度神经网络建模过程

Input: 事件样本集T

Output: 深度神经网络模型f'

1: 根据任务设计深度神经网络结构，得到函数族$f_\theta(X)$；
2: 根据事件样本集T和具体任务，构造损失函数，得到$L(\theta)$；
3: 选择具体的优化算法opt；
4: 随机初始化参数θ_0；
5: 使用优化算法opt求解无约束优化问题$\min_\theta L(\theta)$；
6: 迭代求解出参数$\theta_1, \theta_2, \theta_3, \cdots, \theta_n$；
7: 返回最终模型$f' = f_{\theta_n}(X)$；

首先根据具体任务，选择合适的深度神经网络结构，得到确定的函数族$f_\theta(X)$。接着根据收集到的事件样本集T，构造对应的损失函数，得到$L(\theta)$。之后选择某个具体的优化算法opt，如梯度下降法、动量梯度下降法等。最后通过优化算法opt迭代求解无约束优化问题$\min_\theta L(\theta)$，得到最终的深度神经网络模型f'。

第三章　基于动力系统的梯度下降法

3.1　梯度下降算法

对于无约束优化问题:

$$\min_{\theta} L(\theta) \tag{3.1}$$

其中

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(x_i, \theta)$$

l为单个样本的损失函数，度量单个样本的损失，x_i为第i个训练样本特征，n表示样本数量。梯度下降法是一种解决无约束优化问题(3.1)的常用方法，假设$L(\theta)$的梯度存在，并记为$\nabla_\theta L(\theta)$。

梯度下降法的参数更新方式如下：

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} L(\theta_t)$$

其中θ_t表示第t步更新后的参数，α为每次更新的步长。将梯度下降法作如下变换：

$$\frac{\theta_{t+1} - \theta_t}{\alpha} = -\nabla_{\theta_t} L(\theta_t)$$

$$\lim_{\alpha \to 0} \frac{\theta_{t+1} - \theta_t}{\alpha} = \frac{d\theta_t}{dt}$$

所以梯度下降法等价于求解如下动力系统方程的欧拉向前法：

$$\frac{d\theta}{dt} = -\nabla_\theta L(\theta)$$

$$\theta(t_0) = \theta_0$$

θ_0对应梯度下降法的初始化参数。从上面的推导可以看出，对于一个多元函数的无约束优化问题，传统的梯度下降法等价于求解一个动力系统方程组的数值解法。后面将沿着这种思想建立起梯度下降法和动力系统之间的关系。

3.2　动力系统与流形稳定性定理

动力系统的概念，起源于常微分方程定性理论研究，考虑定义在R^m上的微分方程组：

$$\frac{d\theta}{dt} = f(\theta)$$
$$\theta(0) = x_0$$

$f \in C^r(R^m, R^m)$，$x_0 \in R^m$。我们知道满足方程组和初始条件的解$\phi(t, x_0)$总是局部存在的。如果f满足一定条件，那么解$\phi(t, x_0)$可以对一切$t \in R$和$x_0 \in R^m$有定义。把x_0写作x，这时，解$\phi(t, x)$满足如下条件:

1. $\phi(0, x) = x, \forall x \in R^m$

2. $\phi(s + t, x) = \phi(s, \phi(t, x)), \forall s, t \in R, x \in R^m$

这时称上述$\phi : R \times R^m \to R^m$为$R^m$中的动力系统，对于给定的$x \in R^m$:

$$orb_\phi(x) = \{\phi(t, x), t \in R\}$$

称为动力系统ϕ经过x点的轨道。

定义3.2.1：假设(X, d)是一个距离空间，f是X到X的同胚映射，对于$x \in X$

$$W^s(x, f) = \{y \in X | \lim_{k \to +\infty} d(f^k(y), f^k(x)) = 0\}$$

称为f在x点的稳定流形。这里f^k表示通过k次复合得到的函数。

定义3.2.2：假设(X, d)是一个距离空间，f是X到X的同胚映射，B表示开球，对于$x \in X$和$\epsilon > 0$

$$W^s_{loc}(x, f) = \bigcap_{j \geq 0} f^{-j}(B(f^j(x), \epsilon)) \bigcap W^s(x, f)$$

称为x点的局部稳定流形。$W^s_{loc}(x, f)$也可等价定义为：

$$W^s_{loc}(x, f) = \{y \in X | \begin{matrix} d(f^j(y), f^j(x)) \leq \epsilon, j = 1, 2, \cdots \\ \lim_{k \to \infty} d(f^k(x), f^k(y)) = 0 \end{matrix}\}$$

定义3.2.3：对于$m \in N$，我们把

$$D^m = \{u = (u_1, u_2, \cdots, u_m) \in R^m | \sum_{j=1}^{m} u_j^2 < 1\}$$

称为m维的圆盘。对于某个流形M，如果存在一个C^r同胚的映射$f: D^m \to M$，称M为C^r嵌入圆盘。

命题3.2.4(动力系统的流形稳定性定理 [Shub, 2013])：

假设0是C^r微分同胚$\phi: U \to E$的不动点，U是巴拿赫空间E中0点处的小邻域。假设$E = E_s \oplus E_u$，E_s是由$J\phi(0)$中小于等于1 的特征值对应的特征向量张成的空间，E_u是$J\phi(0)$中大于1 的特征值对应的特征向量张成的空间，$J\phi(0)$是ϕ在0点处的雅可比矩阵。则存在一个C^r嵌入圆盘且是局部稳定的流形$W^{cs}_{loc}(0,\phi)$，它在0 点处和E_s相切，我们称W^{cs}_{loc} 为局部稳定中心流形。此外，存在0 处的邻域B，$\phi(W^{cs}_{loc}) \cap B \subset W^{cs}_{loc}$，$\cap_{k=0}^{\infty}\phi^{-k}(B) \subset W^{cs}_{loc}$。

流形稳定性定理告诉我们，如果一个微分同胚映射作用在一个不动点的某邻域上，那么存在包含这个点的局部稳定中心流形W^{cs}_{loc}，它如同于和E_s 在该点处相切的圆盘，即$dim(E_s) = dim(W^{cs}_{loc})$。$W^{cs}_{loc}$ 包含了所有局部收敛于该点的点。

3.3　梯度下降法收敛性分析

定义3.3.1：$g_k(\theta)$表示以θ_0为初始值的第k 次梯度下降法迭代后的值：

$$g_k(\theta_0) = g_{k-1}(\theta_0) - \alpha\nabla L(g_{k-1}(\theta_0))$$
$$g_0 = \theta_0$$

我们将$g_k(\theta_0)$表示为复合函数的形式：

$$g_k(\theta_0) = g(g(g(\cdots g(\theta_0))\cdots)$$

其中：

$$g(\theta) = \theta - \alpha\nabla L(\theta) \tag{3.2}$$

易知$g: R^m \to R^m$。

定义3.3.2：$S(\theta^*)$称为驻点θ^*的全局稳定集：

$$S(\theta^*) = \{x : \lim_{k\to\infty} g^k(\theta) = \theta^*\}$$

定义3.3.3：向量函数$f(x)$称为利普西斯连续的，如果:

$$\|f(x) - f(y)\| \le \beta\|x - y\|$$

其中β称为利普西斯常数，$\|\cdot\|$是向量的二范数。

引理3.3.4：如果函数$L(\theta)$是C^2的且，

$$\|\nabla L(\theta') - \nabla L(\theta'')\|_2 \leq \beta\|\theta' - \theta''\|_2 \quad \forall\theta', \theta'' \tag{3.3}$$

则$\nabla^2 L(\theta)$的最大特征值不超过β。

证明：假设$\exists\theta_0$，S.T $\nabla^2 L(\theta_0)$ 最大特征值$\lambda_0 > \beta$。在λ_0的特征子空间中取一非零向量$\bar{v}$，$\|\bar{v}\|_2 = 1$，则$\nabla^2 L(\theta_0)\bar{v} = \lambda_0\bar{v}$

由泰勒公式：

$$\frac{\|\nabla L(\theta' + t\bar{v}) - \nabla L(\theta')\|_2}{t} = \frac{\|\nabla^2 L(\theta')t\bar{v} + o(\bar{t})\|_2}{t} > \lambda_0 - \frac{o(t)}{t}$$

当t充分小时，由于$\lambda_0 > \beta$，则$\lambda_0 - \frac{o(t)}{t} > \beta$ 根据假设：

$$\frac{\|\nabla L(\theta' + t\bar{v}) - \nabla L(\theta')\|_2}{t} = \frac{\|\nabla L(\theta' + t\bar{v}) - \nabla L(\theta')\|_2}{\|\theta' + t\bar{v} - \theta'\|_2} \leq \beta$$

矛盾，所以$\nabla^2 L(\theta)$的最大特征值不超过β。

定理3.3.5：若$L(\theta)$是C^2的且满足3.3式，$\alpha < \frac{1}{\beta}$，则3.2的函数g是微分同胚。

证明：假设$g(\theta') = g(\theta'')$，则$\theta' - \theta'' = \alpha(\nabla L(\theta') - \nabla L(\theta'')) \leq \alpha\beta\|\theta' - \theta''\|_2$

所以$\theta' = \theta''$，$g(\theta)$是单射。

下面证$g(\theta)$是满射：

$$g(\theta) = \theta - \alpha L(\theta)$$

$$g(\theta_x) = \theta_y$$

对$\forall\theta_y$，构造函数：

$$h(\theta_x) = \frac{1}{2}\|\theta_x - \theta_y\|^2 - \alpha L(\theta_x)$$

$$\nabla^2 h(\theta_x) = I - \alpha\nabla^2 L(\theta_x)$$

根据引理3.3.4，$\nabla^2 L(\theta_x)$的最大特征值不超过β，因为$\alpha < \frac{1}{\beta}$，所以$\nabla^2 h(\theta_x)$的特征值均大于0，所以$h(\theta_x)$是严格凸函数。$h(\theta_x)$有唯一的极小值点。

$$\theta_{x^*} = arg\min_{\theta_x} \frac{1}{2}\|\theta_x - \theta_y\|^2 - \alpha L(\theta_x)$$

由于θ_{x^*}是极小值点，所以：

$$g(\theta_{x^*}) = \theta_{x^*} - \alpha\nabla L(\theta_{x^*}) = \theta_y$$

所以$g(\theta)$是满射。由于$Jg(\theta) = I - \alpha\nabla^2 L(\theta)$，所以当$\alpha < \frac{1}{\beta}$时，$detJg(\theta) > 0$，这里$Jg(\theta)$表示向量函数$g$的雅可比矩阵。根据逆映射定理，$g^{-1}$也是连续可微的，综上所述，函数$g$是微分同胚。

定理3.3.6：假设$L(\theta)$是C^2的，且满足(3.3) 式的利普西斯条件，且θ^*是$L(\theta)$的严格鞍点，$0 < \alpha < \frac{1}{\beta}$，那么$\mathbf{m}(S(\theta^*)) = 0$，$\mathbf{m}$为定义在$R^m$上的勒贝格测度。

证明：因为$Jg(\theta) = I - \alpha\nabla^2 L(\theta)$，且$g$是微分同胚，根据动力系统的流形稳定性定理，$w_{loc}^{cs}$的维度等于$\nabla^2 L(\theta^*)$负特征值的个数，由于$\theta^*$是严格鞍，$Jg(\theta)$必有一个特征值大于1，$w_{loc}^{cs}$的维度小于$\theta$所在空间维度$m$。所以$\mathbf{m}(w_{loc}^{cs}) = 0$。

B是根据动力系统的流形稳定性定理得到的θ^*附近的小邻域。对$\forall\theta \in S(\theta^*)$，因为$\lim_{k\to\infty} g^k(\theta) = \theta^*$，$\exists K$，$g^k(\theta) \in B$，$k \geq K$。因为$\cap_{i=0}^{\infty} g^{-i}(B)$ 表示经过任意次迭代后均在B内的点的集合，所以$g^K(\theta) \in \cap_{i=0}^{\infty} g^{-i}(B)$，当有了$B$内收敛到$\theta*$的集合估计之后，我们可以通过$g^{-1}$逆向迭代出$S(\theta^*)$，所以有如下的等式：

$$S(\theta^*) \subset \bigcup_{l\geq 0}^{\infty} g^{-l}(\bigcap_{i=0}^{\infty} g^{-i}(B))$$

由流形稳定性定理，因为$\cap_{i=0}^{\infty} g^{-i}(B) \subset W_{loc}^{cs}$，$E(w_{loc}^{cs}) = 0$，所以$E(S(\theta^*)) = 0$。

这个定理告诉我们，在LandScape猜想的假设条件下，当$L(\theta)$和α满足一定的条件，我们对深度神经网络随机初始化参数，然后按照梯度下降法进行迭代，如果迭代的时间足够的长，那么恰好落到严格鞍的概率是0。这意味着如果深度神经网络满足LandScape假设且满足定理3.3.6的相关条件，在随机初始化参数后使用梯度下降法，收敛到某个局部极小值的概率为1。

第四章　基于随机动力系统的随机梯度下降法

4.1　随机微分方程与扩散过程

定义4.1.1：一般的随机微分方程组表示如下：

$$d\xi_t = b(t,\xi_t)dt + \sigma(t,\xi_t)dB_t \tag{4.1}$$

B_t为多维维纳过程，$b(t,\xi_t)$称为漂移系数，$a(t,\xi_t) \equiv \sigma(t,\xi_t)\sigma(t,\xi_t)^T$称为扩散系数。如果此随机微分方程组的解存在，那么其解作为随机过程称为扩散过程。

定义4.1.2：称$\sigma(t,x)$，$b(t,x)$满足Lipschitz条件，如果对$\forall T > 0$，存在$C_T > 0$，使得对于任意$t \le T$一致的满足如下的条件：

$$\|\sigma(t,x) - \sigma(t,y)\| + \|b(t,x) - b(t,y)\| \le C_T\|x - y\|$$

定义4.1.3：如果存在非负函数$\varphi(x)$，使时齐的Markov 过程的转移密度$p(t,x,y)$满足

$$\varphi(y) = \int p(t,x,y)\varphi(x)dx$$

那么我们将$\varphi(x)$称为转移密度$p(t,x,y)$的不变密度，也称为Markov 过程的不变密度。

定义4.1.4：我们称形如(4.1)式的随机微分方程组是良好的，如果其满足如下条件：

1. 该方程$\sigma(t,x)$，$b(t,x)$满足Lipschitz条件。
2. $b(t,x), a(t,x)$是光滑的。
3. $a(t,x)$有一个正的下界。

命题4.1.5(转移密度两歧性定理) [龚光鲁, 2008] 假设形如(4.1)式的随机微分方程是良好的，其解d维扩散过程$\{\xi_t, t \ge 0\}$的转移密度$p(t,x,y)$满足：或者$\lim_{t\to+\infty} p(t,x,y)$ =某个不变密度$\varphi(x)$，或者$\lim_{t\to+\infty} p(t,x,y) = 0$。并且有不变密度的充要条件为下述方程：

$$\frac{1}{2}\sum_{i,j=1}^{d}\frac{\partial^2}{\partial y_i \partial y_j}(a_{ij}(y)\varphi) - \nabla\cdot(b(y)\varphi) = 0 \tag{4.2}$$

有非负非零可积解φ。当条件成立时，$\varphi = \frac{\psi}{\int\psi}$就是扩散过程的不变密度。这时如果初值$\xi_0$ 的密度是$\varphi(x)$，那么$\{\xi_t, t \ge 0\}$是一个平稳过程。

4.2　随机梯度下降法收敛性分析

3.1节中介绍的梯度下降法虽然是解决无约束优化问题的通用方法，但是在解决具有大规模参数和样本数的机器学习问题的时候，此类全批量梯度下降法由于计算效率较慢显得力不从心。不妨假设真实问题的损失函数的梯度的方差为σ，每个样本的损失函数的梯度独立同分布的抽样于方差为σ的分布，则n个样本的损失函数的和的梯度的方差为$\frac{\sigma}{\sqrt{n}}$，分母$\sqrt{n}$表明使用更多的样本来估计梯度的方法的回报是低于线性的，所以可以使用随机抽取部分样本的方法来近似估计梯度，也就是随机梯度下降法。Léon Bottou等人 [Bottou, 2010]首次将随机梯度下降法引入到大规模的机器学习问题中。随机梯度下降法是从全量样本中随机抽取一部分样本，用这一部分样本的损失函数的和的梯度近似全批量样本损失函数的和的梯度。

$$L_r(\theta) = \frac{1}{|r|}\sum_{x_i \in r} l(x_i, \theta) \tag{4.3}$$

其中r为随机样本集合，$|r|$的大小称为BatchSize。

由于每次从全样本中随机选择固定大小的集合的梯度作为近似，且两次随机采样是近似独立的，所以我们不妨假设：

$$\nabla_\theta L_r(\theta) - \sqrt{2}\sigma\epsilon = \nabla_\theta L(\theta) \tag{4.4}$$

其中ϵ服从一个多元高斯分布$\epsilon \sim N(\mathbf{0}, I)$，类似3.1小节的讨论，随机梯度下降法可以近似转换为下述随机微分方程的欧拉向前法：

$$\frac{d\theta}{dt} = -\nabla_\theta L(\theta) + \sqrt{2}\sigma\epsilon$$

将上述随机微分方程转换为一般形式有：

$$d\theta_t = -\nabla_\theta L(\theta_t)dt + \sqrt{2}\sigma \cdot IdB_t \tag{4.5}$$

定理4.2.1：对于随机微分方程(4.5)，假设$\nabla_\theta L(\theta)$满足Lipschitz连续性，且$\nabla_\theta L(\theta)$光滑，如果θ能达到平稳解，则其平衡解表示如下：

$$\varphi(\theta) = \frac{e^{-\frac{L(\theta)}{\sigma^2}}}{\int e^{-\frac{L(\theta)}{\sigma^2}}}$$

证明： 由于随机微分微分方程组满足以上条件，根据命题4.1.5。我们只需证明$e^{-\frac{L(\theta)}{\sigma^2}}$满足命题4.1.5中的(4.2)。其中：

$$b(\theta) = -\nabla L(\theta)$$

$$(a_{ij}) = (\sqrt{2}\sigma \cdot I)(\sqrt{2}\sigma \cdot I)^T = 2\sigma^2 \cdot I$$

令$\varphi = e^{-\frac{L(\theta)}{\sigma^2}}$：

$$\Delta\varphi = \nabla \cdot (\nabla\varphi) = \nabla \cdot (-\frac{1}{\sigma} \cdot e^{-\frac{L(\theta)}{\sigma^2}} \cdot \nabla L(\theta)) = \frac{1}{\sigma^2}\nabla(\varphi \cdot (-\nabla L(\theta)))$$

$$\sigma^2\Delta\varphi - \nabla \cdot (-\nabla L(\theta)\varphi) = \nabla(\varphi \cdot -\nabla L(\theta)) - \nabla \cdot (-\nabla L(\theta)\varphi) = 0$$

由命题4.1.5有$\frac{e^{-\frac{L(\theta)}{\sigma^2}}}{\int e^{-\frac{L(\theta)}{\sigma^2}}}$为随机微分方程(4.5)的平衡解。

推论4.2.2：若随机微分方程(4.5)能达到平衡解，当$t \to +\infty$时，随机变量$\varphi(\theta)$ 能遍历所有可能的θ，且其密度函数满足$\frac{e^{-\frac{L(\theta)}{\sigma^2}}}{\int e^{-\frac{L(\theta)}{\sigma^2}}}$。

在深度神经网络的优化过程中，随机梯度下降法本质上等价于上述随机微分方程的求解。上述推论告诉我们，只要随机迭代的时间足够的长，那么最终得到的深度神经网络参数理论上可以遍历所有的θ值，而不会像传统梯度下降法那样陷入某一个局部极小值（或鞍点），而且如果$L(\theta')$的值越小，那么最后采样到θ'的概率会越大。所以随机梯度下降法倾向于去采样$L(\theta)$的值小的点。根据定理的结果，能发现概率的大小由σ^2决定，如果σ^2越大，那么这个分布越倾向于均匀分布，$L(\theta')$的值起到的作用就会越小。如果σ^2越小，那么$L(\theta')$的值起到的作用就会越大。换个角度来看，整个过程类似于贝叶斯优化中的勘探和开采过程，引入的噪声起到了勘探的作用，按照负梯度方向迭代起到了开采的作用。可以发现勘探过程是由随机噪声σ^2决定的，在有限的优化过程中，σ^2越大则勘探越频繁，σ^2越小则开采越频繁。所以在有限的优化过程中，我们需要合理的来设置σ^2的值，使得在有限的优化过程中尽量获取到更优的局部极小值点。

根据前面(4.4)式的假设，如果忽略掉浮点精度的舍入误差和离散格式的截断误差，σ^2的大小和BatchSize的大小是相关的，一般BatchSize越大，则σ^2越小，BatchSize越小，则σ^2越大。这启发我们可以通过变换BatchSize大小的方式，来指导随机梯度下降法的训练。实际上Samuel L. Smith等人 [Smith 等, 2017]通过实验发现，在学习率一定的情况下，在迭代过程中慢慢增加BatchSize的值不仅能提高训练的速度，而且能够达到更好的训练效果。根据上面的讨论，可以得到如算法2所示的BatchSize自适应优化算法。BatchSize自适应优化算法是从BatchSize的角度对原始的随机梯度下降法的一种优化。在随机梯度下降的过程中，根据事先设定好的条件逐步的增加BatchSize的大小。在第七章的实例分析中表明，在相同epoch的迭代次数下，BathSize自适应优化算法在测试集上的效果优于传统随机梯度下降法。

在程序实现过程，BatchSize自适应优化算法只需在原始的随机梯度下降法

Algorithm 2: BatchSize自适应优化算法

Input: 损失函数$L(\theta)$，迭代步长α，迭代epoch次n，增加$BatchSize$的条件

Output: epoch迭代n次后的参数θ_n;

```
随机初始化参数θ_0;
初始化BatchSize值bs;
for i = 1; i <= n; i++ do
    while 从全样本无放回抽样bs次得到L_r(θ) do
        θ_i = θ_i − ∇L_r(θ_i)
        if 全样本都被抽过一次 then
            跳出while循环
        end if
    end while
    if i达到指定增加BatchSize的条件 then
        增加BatchSize值bs
    end if
end for
return θ_n
```

基础上增加改变BatchSize的条件即可。在参数迭代的过程中，算法根据条件触发增加BatchSize的操作。

第五章　基于动力系统的动量梯度下降法

5.1　动量梯度下降法

如果使用迭代的方法求解形如(3.1)式的无约束优化题，核心的步骤有两个，一个是迭代方向的选择，另一个是迭代步长的选择。3.1小节的梯度下降法以及4.2小节的随机梯度下降法均选择负梯度方向为下降方向，他们都选择一个固定的参数α为迭代步长。

动量梯度下降法是在梯度下降法的基础上引入动量化方法。动量梯度下降法迭代的步长和梯度下降法一样保持为固定系数，但下降方向的选择则采取历史梯度信息的指数平均作为最终的下降方向。

$$m_{t+1} = \beta m_t - \nabla L(\theta_t)$$

$$\theta_{t+1} = \theta_t + \alpha m_{t+1}$$

其中m_{t+1}为历史梯度信息的指数平均。

定理5.1.1：对于动量梯度下降法，若$\beta = 1 - c\sqrt{\alpha}$，其中c为常数，$0 < \alpha < \frac{1}{c^2}$，其迭代格式近似等价于数值求解下面的动力系统方程：

$$\frac{d^2\theta}{dt^2} + c\frac{d\theta}{dt} = -\nabla L_\theta(\theta) \tag{5.1}$$

证明：由动量梯度下降法公式，我们有：

$$\theta_{t+1} - \theta_t = \alpha(\beta m_t - \nabla L(\theta_t)))$$

$$m_t = \frac{\theta_t - \theta_{t-1}}{\alpha}$$

$$\frac{\theta_{t+1} - \theta_t}{\alpha} - \beta\frac{\theta_t - \theta_{t-1}}{\alpha} = -\nabla L(\theta_t)$$

$$\frac{(\theta_{t+1} - \theta_t) - (\theta_t - \theta_{t-1})}{\alpha} + \frac{(1-\beta)}{\alpha}(\theta_t - \theta_{t-1}) = -\nabla L(\theta_t)$$

为了使得左边等式在$\alpha \to 0$时，能收敛到θ_t的相关导数，我们需要对左边的$\frac{(\theta_{t+1}-\theta_t)}{a}$进行拆分以及重新组合。

$$\frac{(1+\beta)}{2}\frac{(\theta_{t+1} - \theta_t) - (\theta_t - \theta_{t-1})}{\alpha} + (1-\beta)\frac{(\theta_{t+1} - \theta_t + \theta_t - \theta_{t-1})}{2\alpha} = -\nabla L(\theta_t)$$

$$\frac{(1+\beta)}{2}\frac{(\theta_{t+1}-\theta_t)-(\theta_t-\theta_{t-1})}{(\sqrt{\alpha})^2}+\frac{(1-\beta)}{\sqrt{\alpha}}\frac{(\theta_{t+1}-\theta_t+\theta_t-\theta_{t-1})}{2\sqrt{\alpha}}=-\nabla L(\theta_t)$$

令 $\sqrt{\alpha}\to 0^+$，则$\beta\to 1$，有：

$$\frac{d^2\theta}{dt^2}+c\frac{d\theta}{dt}=-\nabla L_\theta(\theta)$$

注意：当$\alpha\to 0^+$时，β也会动态的趋近于1，且α 不能等于0，否则会退化为梯度下降法对应的动力系统方程。

根据定理5.1.1我们可以看出，动量梯度下降法近似的动力系统方程某种程度上可以看作是梯度下降法近似的动力系统方程的某种“正则”表示，其中“正则”项为$\frac{d^2\theta}{dt^2}$。从微分方程数值方法的角度上来看，当加入“正则”后，使用 $\sqrt{\alpha}$进行数值迭代的时候其截断误差是$O(\alpha)$，而梯度下降法虽然截断误差也是$O(\alpha)$，但迭代的步长是α，由于α的取值都在0和1之间，所以相同迭代步长下动量梯度下降法具有更高的精度。

从推导过程中，我们可以发现需要假设$\beta=1-c\sqrt{\alpha}$。所以为了在使用动量梯度下降法的过程中，逼近的始终是同一个动力系统方程，这启发我们在调低学习率α的时候，可以同时调高β，其中$\beta=1-c\sqrt{\alpha}\quad\forall\alpha\in(0,1)$。

5.2　随机动量梯度下降法

和4.2小节的讨论类似我们不妨假设

$$\nabla_\theta L_r(\theta)-\sqrt{2}\sigma\epsilon=\nabla_\theta L(\theta)$$

其中ϵ服从一个多元高斯分布$\epsilon\sim N(0,I)$，动量梯度下降法的离散格式可以近似转换为下述随机微分方程：

$$\frac{d^2\theta}{dt^2}+c\frac{d\theta}{dt}=-\nabla_\theta L(\theta)+\sqrt{2}\sigma\epsilon$$

令$\gamma=\frac{d\theta}{dt}$，上述方程等价于：

$$\begin{bmatrix} I & 0\\ 0 & I\end{bmatrix}\begin{bmatrix}\frac{d\theta}{dt}\\ \frac{d\gamma}{dt}\end{bmatrix}+\begin{bmatrix}0 & -I\\ 0 & cI\end{bmatrix}\begin{bmatrix}\theta\\ \gamma\end{bmatrix}=\begin{bmatrix}0\\ -\nabla_\theta L(\theta)\end{bmatrix}+\begin{bmatrix}0\\ \sqrt{2}\sigma\end{bmatrix}$$

令$\xi=\begin{bmatrix}\theta\\ \gamma\end{bmatrix}$，将上述随机微分方程转换为一般形式有：

$$d\xi_t = \begin{bmatrix} \gamma \\ -\nabla_\theta L(\theta) - c\gamma \end{bmatrix} dt + \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{2}\sigma I \end{bmatrix} dB_t \tag{5.2}$$

定理5.2.1：对于随机微分方程(5.2)，当$c = 0$时，假设$\nabla_\theta L(\theta)$满足Lipschitz连续性，且$\nabla_\theta L(\theta)$光滑，如果θ能达到平稳解，则其平衡解表示如下：

$$\varphi(\theta, \gamma) = \frac{e^{-\frac{L(\theta)+\frac{\|\gamma\|_2^2}{2}}{\sigma^2}}}{\int e^{-\frac{L(\theta)+\frac{\|\gamma\|_2^2}{2}}{\sigma^2}}}$$

证明：由于随机微分微分方程组满足以上条件，根据命题4.1.5。我们只需证明$\varphi(\theta, \gamma)$满足命题4.1.5中的(4.2)。其中：$b = \begin{bmatrix} \gamma \\ -\nabla_\theta L(\theta) \end{bmatrix}$，$a = \begin{bmatrix} 0 & 0 \\ 0 & 2\sigma^2 I \end{bmatrix}$，$\gamma = (\frac{\partial\theta_1}{\partial t}, \frac{\partial\theta_2}{\partial t}, \cdots, \frac{\partial\theta_m}{\partial t})^T$。
因为：

$$\frac{1}{2} 2\sigma^2 \Delta\varphi = \sigma^2 \nabla \cdot (\nabla\varphi)$$

$$\nabla\varphi = -\frac{1}{\sigma^2}\varphi \begin{bmatrix} \gamma \\ \nabla_\theta L(\theta) \end{bmatrix} = \frac{1}{\sigma^2}\varphi b$$

所以：

$$\sigma^2 \nabla \cdot (\nabla\varphi) = \nabla \cdot (\varphi b)$$

所以命题4.1.5中的(4.2)成立，综上$\varphi(\theta, \gamma) = \frac{e^{-\frac{L(\theta)+\frac{\|\gamma\|_2^2}{2}}{\sigma^2}}}{\int e^{-\frac{L(\theta)+\frac{\|\gamma\|_2^2}{2}}{\sigma^2}}}$。

和前面一章的思路一样，在对动量梯度下降法和随机动量梯度下降法进行数学的理论分析之后，我们期望通过相关的定理和推论来指导我们对动量梯度下降法进行优化。Momentum自适应优化算法就是根据这种思想提出的新优化算法。Momentum自适应优化算法受启发于传统Momentum优化算法是动力系统方程5.1的数值近似。为了保证在优化的过程中始终是逼近的同一个动力系统方程，学习率α 和指数平均参数β就需要满足一定的等式。在传统Momentum优化算法中，我们常常需要通过调整学习率来获得比较好的优化效果。那么当调整学习率α的时候，就可以根据定理5.1.1中的等式来调整指数平均参数β。所以指数平均参数β是一个自适应调整的过程。综合前面几小节的讨论，得到如算法3所示的Momentum自适应优化算法。需要注意的是在c的选取中，要保证β的值在0 和1 之间。

Algorithm 3: Momentum自适应优化算法

Input: 损失函数$L(\theta)$，迭代epoch次n,超参数c,调低学习率α 的条件;

Output: epoch迭代n次后的参数θ_n;

```
随机初始化参数θ_0;
初始化学习率的值α;
初始化β;
初始化m_1
for i = 1; i <= n; i + + do
    while 从全样本无放回抽样bs次得到L_r(θ) do
        m_i = βm_i − ∇L(θ_i)
        θ_i = θ_i + αm_i
        if 全样本都被抽过一次 then
            跳出while循环
        end if
    end while
    if i达到指定降低α的条件 then
        降低α
        β = 1 − c √α
    end if
end for
return θ_n
```

该算法首先指定超参数c和调低学习率的条件，然后采用动量梯度下降法更新参数。在更新的过程中，如果触发调低学习率的条件，则调整学习率，并根据超参数c以及对应公式调整β的值。

第六章　RK与Adams神经网络优化算法

6.1　RK神经网络优化算法

前几小节启发我们，深度神经网络的优化过程可以看作是一个随机动力系统方程的求解过程。无论是梯度下降法，还是动量梯度下降法，都是求解动力系统方程的某种数值解法。动力系统方程的数值解法在计算数学中已经是个非常成熟的问题，除了欧拉向前法以外，我们还有精度更高的一些数值算法。本小节将Runge-Kutta方法引入到深度神经网络的优化过程中，从而得到RK神经网络优化算法。

Runge-Kutta方法最早是于19世纪末德国科学家C.Runge和M.W.Kutta提出的。后来又作了不同程度的改进和发展。Runge-Kutta方法至今仍然在动力系统方程中有着广泛的应用。

对于一阶常微分方程

$$\frac{dy}{dt} = f(t, y)$$

$$y(0) = \eta_0$$

四阶Runge-Kutta迭代格式如下：

$$y_{n+1} = y_n + \frac{\alpha}{6}(K_1 + K_2 + K_3 + K_4)$$
$$n = 0, 1, \cdots N - 1$$

其中

$$K_1 = f(t_n, y_n)$$
$$K_2 = f(t_n + \frac{1}{2}\alpha, y_n + \frac{1}{2}\alpha K_1)$$
$$K_3 = f(t_n + \frac{1}{2}\alpha, y_n + \frac{1}{2}\alpha K_2)$$
$$K_4 = f(t_n + \alpha, y_n + \alpha K_3)$$

$$y_0 = \eta_0$$

α为Rung-Kutta迭代格式离散的步长。上述方法的局部离散误差为$O(\alpha^5)$。我们将上述方法引入到深度神经网络的优化算法中。

根据5.1小节可知，对于深度神经网络，梯度下降法等价于求解如下动力系统方程组：

$$\frac{d\theta}{dt} = -\nabla_\theta L(\theta)$$

其中

$$\theta = (\theta_1, \theta_2, \theta_3, \cdots, \theta_m)$$

$$-\nabla L(\theta_1, \theta_2, \cdots, \theta_m) = (\frac{-\partial L_\theta(\theta)}{\partial \theta_1}, \frac{-\partial L_\theta(\theta)}{\partial \theta_2}, \cdots, \frac{-\partial L_\theta(\theta)}{\partial \theta_m})$$

m为神经网络参数的数量。

由此可以得到如算法4所示的RK神经网络优化算法。

Algorithm 4: RK神经网络优化方法

Input: 损失函数$L(\theta)$，迭代步长α;

Output: n次迭代后参数θ_n;

随机初始化参数$\theta_0 = (\theta_{0,1}, \theta_{0,2}, \cdots, \theta_{0,m})$;
for $i = 1; i < n; i++$ **do**
　for $j = 1; j <= m; j++$ **do**
　　$k_{1,j} = -\alpha \nabla L(\theta_{i,1}, \theta_{i,2}, \cdots, \theta_{i,m})$
　　$k_{2,j} = -\alpha \nabla L(\theta_{i,1} + \frac{1}{2}k_{1,1}, \theta_{i,2} + \frac{1}{2}k_{1,2}, \cdots, \theta_{i,m} + \frac{1}{2}k_{1,m})$
　　$k_{3,j} = -\alpha \nabla L(\theta_{i,1} + \frac{1}{2}k_{2,1}, \theta_{i,2} + \frac{1}{2}k_{2,2}, \cdots, \theta_{i,m} + \frac{1}{2}k_{2,m})$
　　$k_{4,j} = -\alpha \nabla L(\theta_{i,1} + k_{3,1}, \theta_{i,2} + k_{3,2}, \cdots, \theta_{i,m} + k_{3,m})$
　end for
　for $j = 1; j <= m; j++$ **do**
　　$\theta_{i+1,j} = \theta_{i,j} + \frac{1}{6}(k_{1,j} + 2k_{2,j} + 2k_{3,j} + k_{4,j})$
　end for
end for
return θ_n

需要注意的是，$\nabla L(\theta)$的计算将涉及到多次的深度神经网络的反向传播，所以相比于传统梯度下降法RK方法将更加的耗费时间。

6.2 Adams多步神经网络优化算法

和前小节一样，先讨论一阶常微分方程

$$\frac{dy}{dt} = f(t, y)$$

$$y(0) = \eta_0$$

RK方法在计算y_{n+1}时，只用到了前一个节点y_n的信息，而没有用到前几步的计算所得出的信息，只用前一个节点的信息的方法称为单步法。实际上经过多次单步法计算后，已经得出一系列$y_1, y_2, \cdots, y_n$的值，我们可以充分利用这些信息来计算y_{n+1}，在不增加太多计算量的同时还能获得更高的精度。

我们可以使用Adams显式多步法进行求解：

$$y_{n+1} = y_n + \frac{\alpha}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

$$f_i = f(t, y_i)$$

$$y_0 = \eta_0$$

为了进一步的使用前面几步的信息，我们可以把四步显式Adams法和三步隐式Adams法相联合得到Adams预测-矫正法，即PECE算法：

$$\begin{aligned}
P: &\quad y_{n+1}^{(0)} = y_n + \frac{\alpha}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \\
E: &\quad f_{n+1}^{(0)} = f(t_{n+1}, y_{n+1}^{(0)}) \\
C: &\quad y_{n+1} = y_n + \frac{h}{24}(9f_{n+1}^{(0)} + 19f_n - 5f_{n-1} + f_{n-2}) \\
E: &\quad y_{n+1} = f(t_{n+1}, y_{n+1})
\end{aligned}$$

将上述算法同样应用于向量场景下的深度神经网络中，就可以得到Adams多步神经网络优化算法，如算法5所示。在算法程序的实现过程中，本文将使用TensorFlow [Abadi 等, 2016]中的自动微分和反向传播来计算$\nabla L(\theta)$。当深度神经网络的参数量很大的时候，反向传播的效率是非常慢的。如果仅仅从截断误差的角度上来看，RK神经网络优化方法和Adams多步神经网络优化方法在近似动力系统方程上有着相同的精度，但前者前进一步需要四次反向传播，而后者却只需要两次反向传播。从这个角度上来看，Adams多步法是优于RK方法的。所以在第七章的实例分析中，我们将只讨论Adams多步神经网络优化算法，而并不讨论RK方法。

Algorithm 5: Adams多步神经网络优化方法

Input: 损失函数$L(\theta)$，迭代步长α;

Output: n次迭代后参数θ_n;

随机初始化参数$\theta_0 = (\theta_{0,1}, \theta_{0,2}, \cdots, \theta_{0,m})$;

使用梯度下降法，求得$\theta_1, \theta_2, \theta_3$

for $i = 1; i < n; i++$ **do**

  for $j = 1; j <= m; j++$ **do**

    $\theta^{(0)}_{i+1,j} = \theta_{i,j} - \frac{\alpha}{24}(55\nabla L(\theta_{i,j}) - 59\nabla L(\theta_{i-1,j}) + 37\nabla L(\theta_{i-2,j}) - 9\nabla L(\theta_{i-3,j})))$

    $\theta_{i+1,j} = \theta_{i,j} - \frac{\alpha}{24}(9\nabla L(\theta^{(0)}_{i+1,j}) + 19\nabla L(\theta_{i,j}) - 5\nabla L(\theta_{i-1,j}) + \nabla L(\theta_{i-2,j})$

  end for

end for

return θ_n

需要注意的是Adams多步神经网络优化算法相比于传统梯度下降法需要更多的内存空间，因为每次迭代都需要存储前四次迭代过程中变量的梯度值。

第七章　实例分析

前几章节中，通过研究动力系统方程来分析深度神经网络的优化过程。并以此提出了几种从理论上可能提高训练效率的新优化算法。

在本章节中，将对上述优化算法进行实例分析。我们选择使用TensorFlow自动微分框架，研究Fashion-Mnist数据集 [Xiao 等, 2017]的分类问题。

7.1　实验数据与实验模型

Fashion-Mnist是一个替代Mnist手写数字集的图像数据集。它是由Zalando（一家德国的时尚科技公司）旗下的研究部门提供。其涵盖了来自10 种类别的共7 万个不同商品的正面图片。Fashion-Mnist的大小、格式和训练集/测试集划分与原始的Mnist完全一致。60000/10000的训练测试数据划分，28x28的灰度图片。可以直接用它来测试深度神经网络优化算法的性能。这个数据集的样子大致如下（每个类别占三行）：

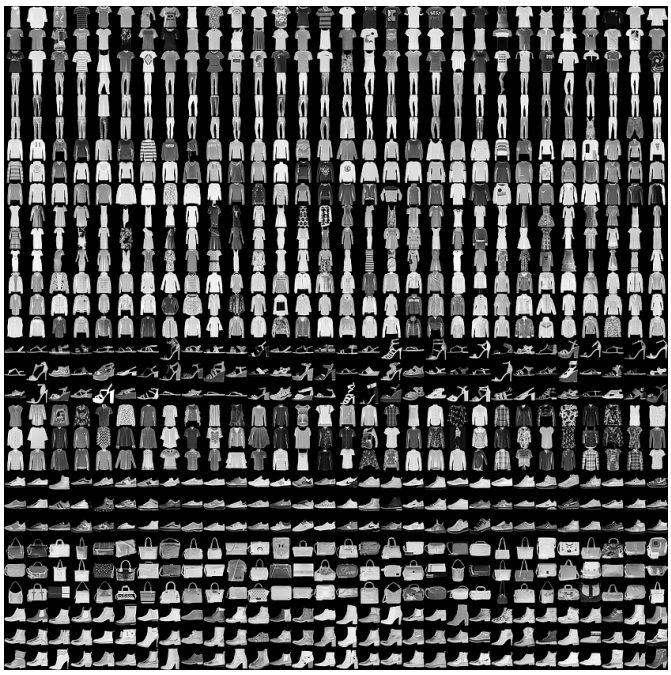

图 7.1: Fashion-Mnist数据集

本章节实验将使用三层全连接神经网络模型和七层卷积神经网络模型。

全连接神经网络模型的输入为$28 * 28$的灰度图，通过flatten操作展开成784维的向量，之后分别通过神经元个数为64和32 的隐藏层，激活函数为relu激活函数。最后通过神经元个数为10的softmax层输出预测值。损失函数选用交叉熵损失函数。整个网络结构如图7.2 所示。

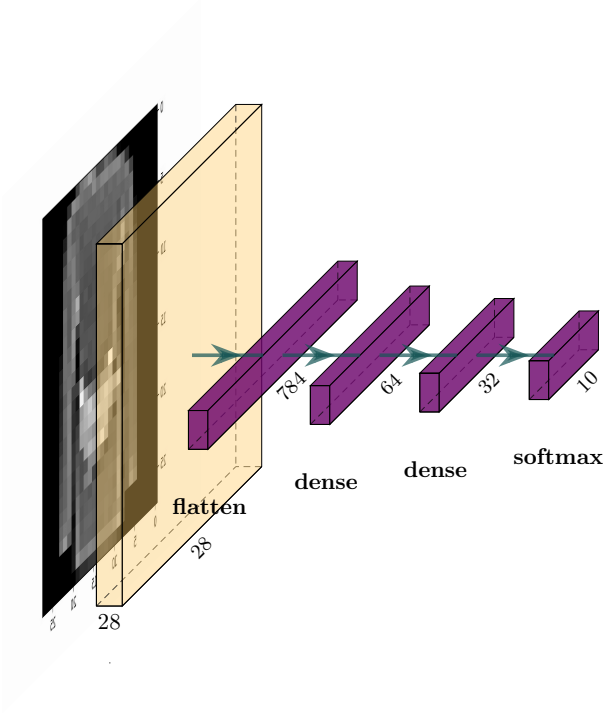

图 7.2: 全连接神经网络模型

卷积神经网络模型的输入为$28 * 28 * 1$的灰度图，首先通过大小为$5 * 5$卷积核数量为6的卷积层。之后通过poolsize为2的maxpool池化层。然后通过卷积核数量为16，大小为$5 * 5$的卷积层。接着通过poolsize为2的maxpool池化层。接着通过卷积核数量为120大小为$4 * 4$的卷积层。接着通过神经元个数为84的全连接层。最后通过神经元个数为10的softmax层输出预测值。损失函数选用交叉熵损失函数。所有的神经元的激活函数均使用relu 激活函数。整个网络结构的如图7.3所示。该模型的结构是根据**LeNet-5** [LeCun 等, 1998]改编得到。

7.2　实验设计与算法实现

本小节主要针对前面提出的三类优化算法进行实例分析：

1. BatchSize自适应的随机梯度下降法。

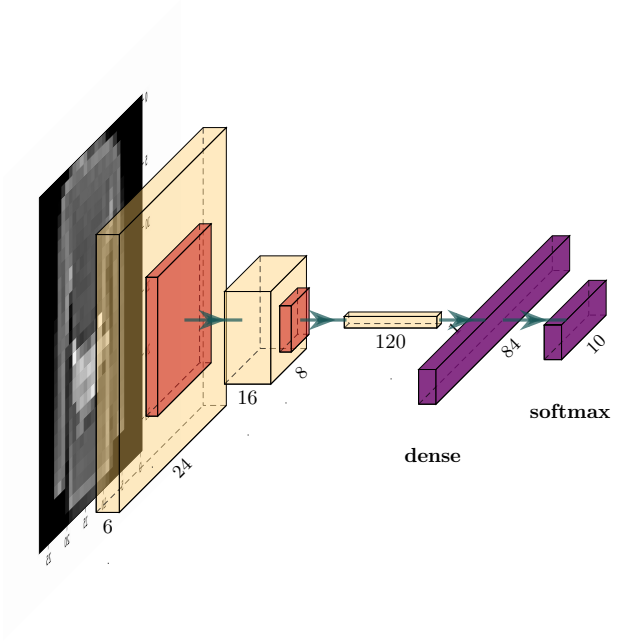

图 7.3: 卷积神经网络模型

2. Momentum自适应的动量梯度下降法。

3. Adams多步神经网络优化算法。

本文将以上三类算法分别与传统的梯度下降方法进行比较，为了避免随机数导致的偶然性，每次比较都将固定深度神经网络的初始化参数。我们对每个模型随机初始化了100套参数。

在BatchSize自适应随机梯度下降法实验中，对于全连接神经网络模型，我们固定学习率为0.1，初始化BatchSize大小为64，总共迭代的epoch次数为50次。在epoch 迭代到20次时修改BatchSize为1024，epoch迭代到30次时修改BatchSize为2048，epoch迭代到40次时修改BatchSize为4096。需要注意的是，由于我们对比对象的BatchSize不变，那么在相同迭代epoch的情况下，BatchSize不增加的实验组必然在训练集上收敛速度得更快，所以我们研究两者在测试集上的损失。我们对100套随机初始化参数进行了实验，记录每次实验中每一个epoch的损失，并取均值。对于卷积神经网络模型，和全连接神经网络模型设置基本相同，但将总共迭代的epoch次数设置为了18次，在epoch迭代到8次时修改BatchSize为1024，epoch迭代到12次时修改BatchSize为2048，epoch迭代到15次时修改BatchSize为4096。

在Momentum自适应动量梯度下降法的实验中，对于全连接神经网络模型，

我们设置初始的学习率为0.1，初始的动量参数为0.9，固定BatchSize大小为64，总共迭代的epoch次数为50次，在epoch迭代到20次时修改为为0.01，epoch迭代到35次时修改为0.001，动量参数根据自适应算法自动改变。分别记录传统动量梯度下降法与自适应动量梯度下降法在训练集上损失函数的下降情况。对于卷积神经网络模型，由于深度较深等原因，使用动量梯度下降法时，导致了梯度爆炸，所以我们对梯度进行了裁剪操作，并把初始的动量参数设置为0.5。其余的操作和全连接神经网络模型一样。

在Adams多步神经网络优化算法的实验中，我们设置初始的学习率为0.01，固定BatchSize大小为128，总共迭代的epoch次数为50次。前20次epoch迭代使用传统的梯度下降法，当epoch数大于20之后，自动切换为Adams多步法。重复实验，记录传统梯度梯度下降法和Adams多步神经网络优化算法在训练集上损失函数的下降情况。

7.3　实验结果分析与比较

对于全连接神经网络模型，传统BatchSize和自适应BatchSize结果如图7.4所示。

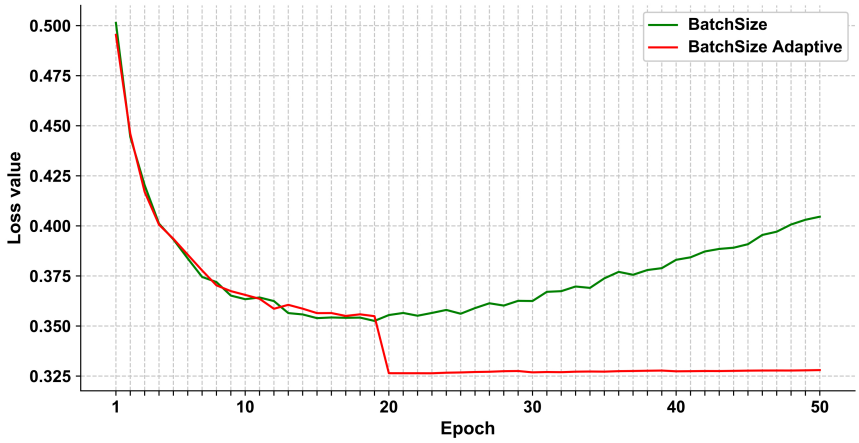

图 7.4: 全连接神经网络模型BatchSize自适应优化对比

对于卷积神经网络模型，传统BatchSize和自适应BatchSize结果如图7.5所示。

从图可以看出，对于全连接神经网络模型，在epoch等于20的时候，由于BatchSize的增加，损失函数有了进一步的降低。对于卷积神经网络模型在epoch等于8的时候，由于BatchSize的增加，损失函数有了进一步的降低。为

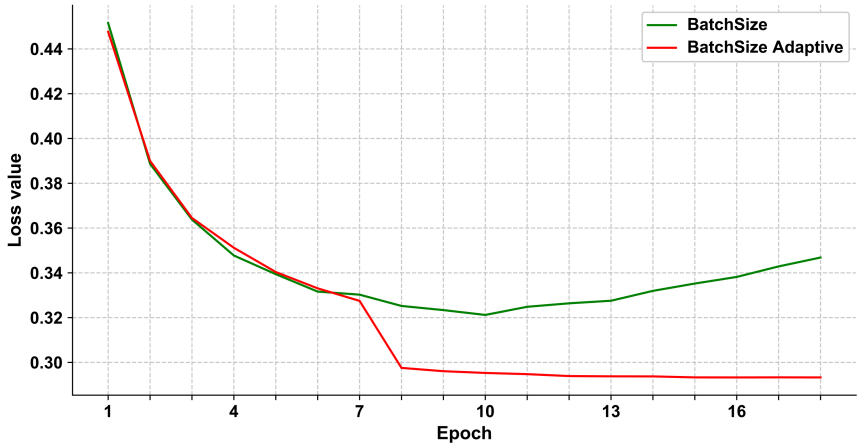

图 7.5: 卷积神经网络模型BatchSize自适应优化对比

了更好的看清后面BacthSize增加对损失的影响，我们放大卷积神经网络模型11至18这段epoch的图像7.6。

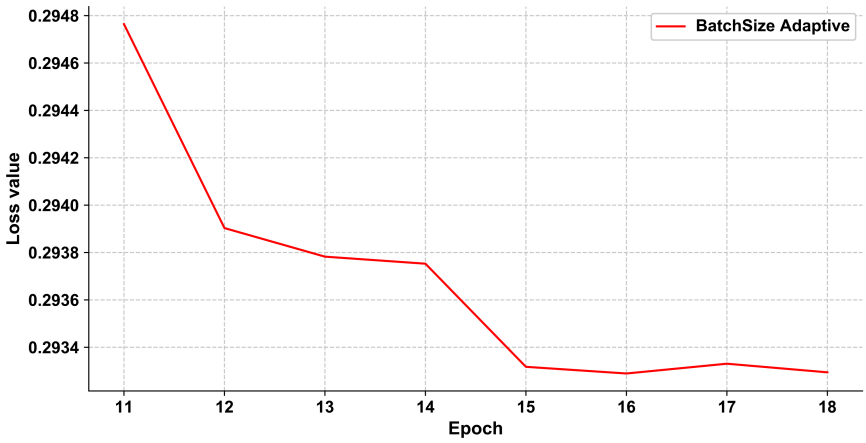

图 7.6: BatchSize 自适应优化对比2

从图可以看出，在epoch等于12，epoch等于15的时候，由于BatchSize的增加，带来了损失进一步的减少。

对于全连接神经网络模型Momentum自适应动量梯度下降法的结果如图7.7所示。 对于卷积神经网络模型Momentum自适应动量梯度下降法的结果如图7.8所示。

从图可以看出，对于全连接和卷积神经网络模型在epoch等于20，epoch等于35的时候，自适应Momentum动量法由于自动调整了动量参数β的值，使得损失有了进一步的下降。

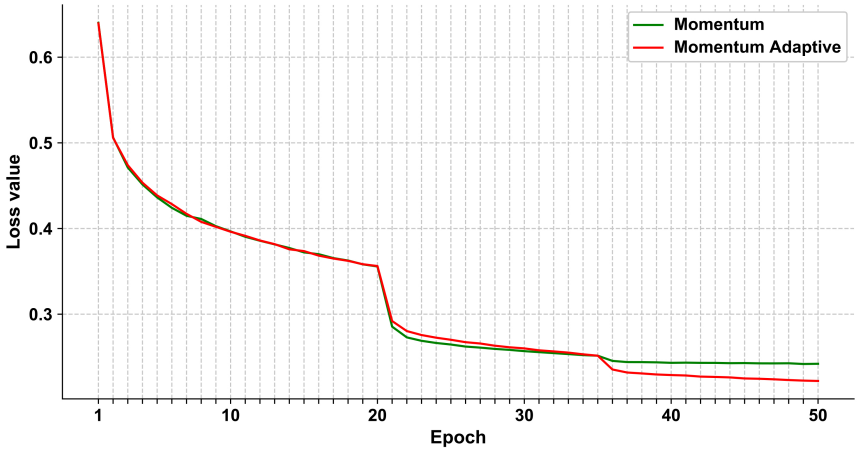

图 7.7: 全连接神经网络模型Momentum自适应优化对比

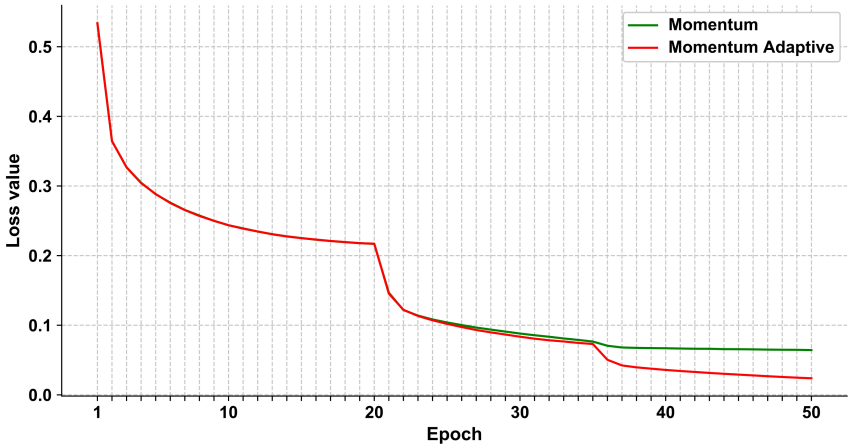

图 7.8: 卷积神经网络模型Momentum自适应优化对比

对于全连接神经网络模型Adams多步神经网络优化算法的结果如图7.9所示。 对于卷积神经网络模型Adams多步神经网络优化算法的结果如图7.10所示。 从图可以看出，在epoch等于20之前，由于两条曲线采用的优化算法都是梯度下降法，所以两条曲线几乎无差异，在epoch大于20之后，采用Adams多步法的曲线在收敛速度上有进一步的提升。

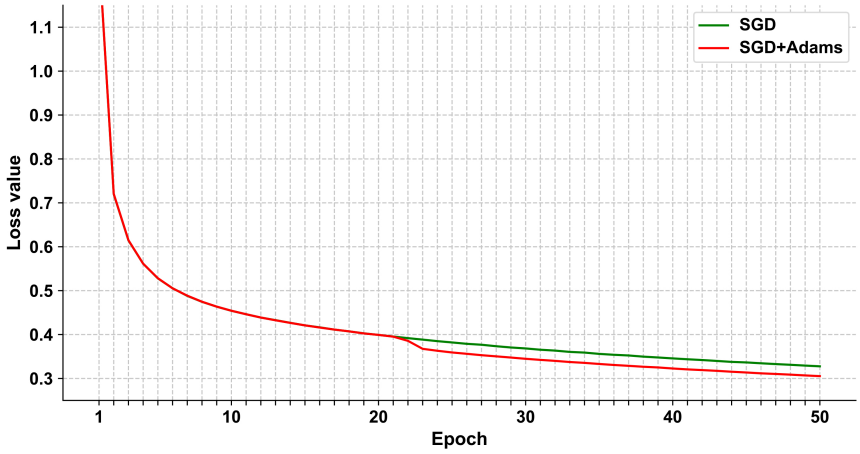

图 7.9: 全连接神经网络模型Adams多步法训练集对比

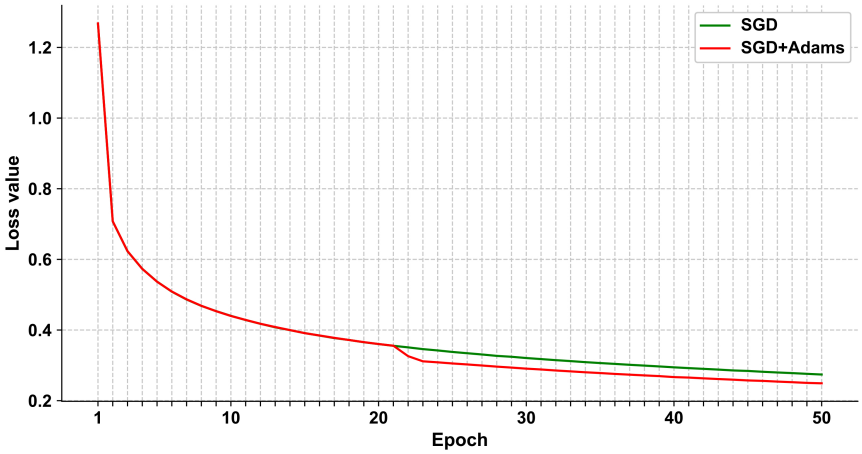

图 7.10: 卷积神经网络模型Adams多步法训练集对比

第八章　总结与展望

本文首先介绍了深度神经网络的相关知识，从基本的定义开始，分析了深度神经网络的一些数学性质，并从中抽象出需要解决的优化问题。之后介绍了动力系统的相关概念和深度神经网络常用的优化算法。在对深度神经网络引入一些数学假设后，利用动力系统中的流形稳定性定理和随机微分方程中的转移密度两歧性定理对全量梯度下降法和随机梯度下降法进行了理论分析，从数学上分析了他们的收敛性。之后从动力系统方程的数值计算的角度分析了部分深度神经网络优化算法的相关性质。最后在以上理论分析的基础上提出了一系列新的深度神经网络优化算法。在实例分析部分，使用全连接神经网络模型和卷积神经网络模型在Fashion-Mnist数据集上进行了实验，一定程度上说明了新的优化算法的有效性。

深度神经网络是一个新兴而复杂的问题，虽然本文的研究工作在某些方面有不错的进展，但是还有很大的提升空间，在以后的研究工作中，我准备从以下几个方面继续深入研究：

(1)在Adams多步法中，训练集损失虽然能稳定下降，但是测试集的损失在下降过程中会偶尔有抖动，但这种抖动并没影响测试集最终的收敛效果。我猜想Adams多步法可能引入了类似于贝叶斯优化的效果，除了开采过程以外，它也有勘探的功能。本文只是从优化角度讨论深度神经网络，但优化不等于学习，深度神经网络的成功在于它良好的泛化能力，所以下一步的工作可以进一步研究Adams多步法以及其他一些优化算法对泛化能力带来的影响。

(2)本文的核心思想是将深度神经网络的优化过程看作是一个动力系统，优化过程只是深度神经网络建模的一个子步骤。深度神经网络的结构设计相对于优化过程显得更加的重要和困难。实际上除了将优化过程看作是动力系统以外，深度神经网络的层级网络结构也可以看作是对一个动力系统的近似。最新的一些研究已经发现了两者之间的部分联系。在下一步的工作中可以进一步研究深度神经网络结构和动力系统之间的关系。

参考文献

龚光鲁, 2008. 随机微分方程及其应用概要. 清华大学出版社.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., 2016. Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265–283.

Auer, P., Herbster, M., Warmuth, M.K., 1996. Exponentially many local minima for single neurons, in: Advances in neural information processing systems, pp. 316–322.

Baldi, P., Hornik, K., 1989. Neural networks and principal component analysis: Learning from examples without local minima. Neural networks 2, 53–58.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., 2016. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 .

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010. Springer, pp. 177–186.

Candes, E.J., Li, X., Soltanolkotabi, M., 2015. Phase retrieval via wirtinger flow: Theory and algorithms. IEEE Transactions on Information Theory 61, 1985–2007.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems 2, 303–314.

Ge, R., Huang, F., Jin, C., Yuan, Y., 2015. Escaping from saddle points online stochastic gradient for tensor decomposition, in: Conference on Learning Theory, pp. 797–842.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, in: European conference on computer vision, Springer. pp. 21–37.

Saxe, A.M., McClelland, J.L., Ganguli, S., 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv preprint arXiv:1312.6120 .

Shub, M., 2013. Global stability of dynamical systems. Springer Science & Business Media.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., 2016. Mastering the game of go with deep neural networks and tree search. nature 529, 484.

Smith, S.L., Kindermans, P.J., Ying, C., Le, Q.V., 2017. Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489 .

Su, W., Boyd, S., Candes, E., 2014. A differential equation for modeling nesterov ‘s accelerated gradient method: Theory and insights, in: Advances in Neural Information Processing Systems, pp. 2510–2518.

Sun, J., Qu, Q., Wright, J., 2017. Complete dictionary recovery over the sphere ii: Recovery by riemannian trust-region method. IEEE Transactions on Information Theory 63, 885–914.

Sun, J., Qu, Q., Wright, J., 2018. A geometric analysis of phase retrieval. Foundations of Computational Mathematics 18, 1131–1198.

Weinan, E., 2017. A proposal on machine learning via dynamical systems. Communications in Mathematics and Statistics 5, 1–11.

Xiao, H., Rasul, K., Vollgraf, R., 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 .

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., Zweig, G., 2016. Achieving human parity in conversational speech recognition. arXiv preprint arXiv:1610.05256 .

简历与科研成果

基本情况 万俊，男，汉族，1991年8月出生，重庆渝中人。

教育背景

2016.9～2019.6 南京大学数学系 硕士

2010.9～2014.6 南京大学数学系 本科

这里是读研期间的成果（实例为受理的专利）

1. Weijun Shen, **Jun Wan**, "MuNN: Mutation Analysis of Neural Networks", in *2018 IEEE International Workshop on Automated Intelligent Software Testing*, May. 2018.
2. 陈振宇，沈维军，**万俊**，房春荣，"一种针对深度学习程序进行神经元变异的测试方法"，申请号：2018108229290，已受理。

致　　谢

时光荏苒，在南京大学的硕士研究生的学习生活即将结束，三年的时间转瞬即逝，这几年的经历必将成为我人生宝贵的财富。在此论文完成之际，谨向这几年来帮助我的老师和同学表达最衷心的感谢。

首先，我要深深的感谢我的导师陆宏老师和陈振宇老师。陆老师治学严谨，学识渊博，给予了我非常多关于人工智能的数学理论上的帮助。陈老师为人谦和，平易近人，每次组会讨论都带给我非常大的收获，在陈老师那增加了我的代码工程能力和思考问题的能力。借此机会，我谨向两位老师致以深深谢意。

其次，我还要感谢我的学长和同学：冯洋师兄、时新凯师兄、李龙同学，他们在学术上给予了我非常多的帮助，在此衷心的感谢你们，愿我们的友谊天长地久。

最后感谢我的父母，他们无私的关心和鼓励帮助我顺利完成学业。