**基于PyTorch平台的深度学习算法研究与应用**

软件项目计划书

**编写人：20TeamA**

**2020年3月16日星期一**

目 录

[软件项目计划书 1](#_Toc35284546)

[1. 引言 1](#_Toc35284547)

[1.1 背景 1](#_Toc35284548)

[1.2 参考资料 1](#_Toc35284549)

[1.3 标准、条约和约定 1](#_Toc35284550)

[2. 项目概述 2](#_Toc35284551)

[2.1 项目目标 2](#_Toc35284552)

[2.2 工作内容 2](#_Toc35284553)

[2.3 应交付成果 3](#_Toc35284554)

[2.4 项目验收方式和依据 3](#_Toc35284555)

[3. 项目团队组织 3](#_Toc35284556)

[3.1 组织结构 3](#_Toc35284557)

[3.2 人员组成 4](#_Toc35284558)

[3.3 协作与沟通 4](#_Toc35284559)

[4. 实施计划 5](#_Toc35284560)

[4.1 计划分析 5](#_Toc35284561)

[1) 深入了解PyTorch框架 5](#_Toc35284562)

[2) 需求分析 5](#_Toc35284563)

[3) 测试分析 5](#_Toc35284564)

[4) 功能实现 5](#_Toc35284565)

[5) 其他软件管理工具学习 6](#_Toc35284566)

[4.2 总体进度计划 6](#_Toc35284567)

[4.3 项目控制计划 7](#_Toc35284568)

[5. 关键问题 7](#_Toc35284569)

[6. 专题计划要点 7](#_Toc35284570)

# 引言

## 背景

深度学习(DL, Deep Learning)是机器学习(ML, Machine Learning)领域中一个新的研究方向，它被引入机器学习使其更接近于最初的目标——人工智能(AI, Artificial Intelligence)。

深度学习在数据挖掘，机器学习，计算机视觉，自然语言处理以及其他相关领域都取得了很多成果。深度学习使机器模仿视听和思考等人类的活动，解决了很多复杂的模式识别难题，使得人工智能相关技术取得了很大进步。

当前用于实现深度学习算法的主流框架主要包括加州大学伯克利分校的Caffe、谷歌的TensorFlow 、微软的Theano、亚马逊的MXNet以及Facebook的PyTorch。在此之中，基于Torch框架和Python语言的PyTorch，由于易于理解的抽象层次、灵活易用的接口设计以及对初学者友好的学习难度，使得其在全球学术界中逐渐成为最受欢迎的深度学习框架。

然而，由于人工智能领域的更新速度快，当前框架集成的高层算法及模型已经不能满足科研人员的实际需求。因此，对于深度学习多领域最新成果在平台上的实现和集成是必要且紧迫的，完成后可以有效地减少科研人员对过往算法和模型的复现工作，提高科研的效率和质量。

## 参考资料

[1] PyTorch官方文档[https://PyTorch.org/](https://pytorch.org/)

## 标准、条约和约定

参照如下标准：

GB/T 13702-1992 计算机软件分类与代码

GB/T 19003-2008 软件工程

GB/T 5538-1995 软件工程标准分类法

GB/T 9385-2008 计算机软件需求规格说明

GB/T 5532-2008 计算机软件测试规范

GB/T 18221-2000 信息技术程序设计语言

GB/T 8567-2006 计算机软件文档编制规范

# 项目概述

## 项目目标

从PyTorch平台出发，重新编写需求规格说明书、测试需求规格说明书。根据需求规格说明书与测试需求规格说明书对PyTorch进行模块化测试。在此基础上拓展应用，即前沿深度学习算法的实现与集成。

## 工作内容

1. 学习使用PyTorch平台使用规范。
2. 学习调研深度学习的发展现状，论证深度学习领域的必要性，细致分析项目需求，完成需求文档。
3. 在PyTorch平台下使用Python语言搭建模型，集成前沿深度学习算法，同时根据评审和工程中出现的新需求、新问题，进行项目开发和文档编写的迭代。
4. 使用主流数据集对算法的性能指标进行独立的测试，完成测试报告。
5. 根据课程的整体安排和每个实验的具体要求，依照项目时间计划完成其他各类文档和报告。

## 应交付成果

项目应交付的产品如表2所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 交付时间 | 要求 | 备注 |
| No.1 | 项目计划书 | 2020/3/17 | 完整、明确 | 暂无 |
| No.2 | 项目进度表 | 2020/3/17 | 完整、详细 | 暂无 |
| No.3 | 项目工作日志 | 待定 | 暂无 | 暂无 |
| No.4 | 软件需求规格说明书 | 待定 | 暂无 | 暂无 |
| No.5 | 需求检查单 | 待定 | 暂无 | 暂无 |
| No.6 | 测试需求规格说明书 | 待定 | 暂无 | 暂无 |
| No.7 | 软件测试分析报告 | 待定 | 暂无 | 暂无 |
| No.8 | 软件测试评审报告 | 待定 | 暂无 | 暂无 |
| No.9 | 项目工作量统计报告 | 待定 | 暂无 | 暂无 |
| No.10 | 项目方法总结报告 | 待定 | 暂无 | 暂无 |
| No.11 | 项目个人总结 | 待定 | 暂无 | 暂无 |

表 2 应交付成果列表

## 项目验收方式和依据

1. 验收方式

项目验收方式采用中期课堂评审和终期交付验收。

1. 验收依据

验收依据“应交付成果”清单进行验收。

# 项目团队组织

## 组织结构

此次项目有两个角色：组员和组长。组长与组员沟通协调，安排具体事宜、分配相应的任务。而组员则需要根据组长的安排要求，及时完成任务。

## 人员组成

人员组成如下表3所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **学号** | **姓名** | **角色** | **擅长** |
| 1 | SY1906423 | 张崇智 | 组员  组长 | 学习能力，代码能力，任务规划。熟悉PyTorch框架和计算机视觉-对抗攻防领域，对领域相关算法及模型有较深了解 |
| 2 | BY1906033 | 秦浩桐 | 组员 | 学习能力，文档写作，代码能力。熟悉PyTorch框架和计算机视觉-模型压缩领域，对领域相关算法及模型有较深了解 |
| 3 | SY1906120 | 高明骏 | 组员 | 学习能力，文档写作，代码能力。熟悉PyTorch框架和计算机视觉-模型压缩领域，对领域相关算法及模型有较深了解 |
| 4 | SY1906504 | 王茵迪 | 组员 | 学习能力，代码能力，文档写作。有使用PyTorch框架的经历，工作方向是自然语言处理领域的传统和深度学习方法。 |
| 5 | SY1906426 | 赵永驰 | 组员 | 研究过自然语言处理中短文本及长文本的多标签分类及细粒度分类，目前主要研究3D角色的动作生成，熟悉PyTorch框架 |
| 6 | BY1906010 | 黄 涵 | 组员 | 学习能力，代码能力。熟悉PyTorch框架和计算机视觉的检测分割领域，对领域相关算法及模型有深入了解 |
| 7 | SY1906420 | 吴振赫 | 组员 | 学习能力，代码能力。熟悉PyTorch框架和计算机视觉的检测分割领域，对领域相关算法及模型有深入了解 |

表 3 人员组成

## 协作与沟通

小组成员通过微信群、电话、邮件等方式进行及时沟通，在项目有重大决定时，可使用腾讯会议进行讨论。而且小组通过文档、Git工具协作完成相应的具体任务。

沟通和协作都要记录具体内容。

# 实施计划

## 计划分析

## 深入了解PyTorch框架

在组内多数成员能够熟练使用的条件下，尽可能全面地了解PyTorch框架的构建。深入分析源代码，了解各个模块的实现细节，理解其中关键的数据结构，并尽可能地体会开发者的意图。

## 需求分析

分析计算机视觉和自然语言处理研究人员的需求，搜集领域中经常使用以及经典的算法，将PyTorch库中没有封装集成的挑选出来。然后根据每个算法的特征，分析算法的实际使用模式，为每个算法需要提供的API和具体实现方式提供依据，以保证封装的算法模块的简单、易用、灵活、通用。另外，还需要了解一些评审指标和制作评审文档等。

## 测试分析

对其进行正确性以及性能测试，检查软件的实现有没有错误，运行是否高效。写出相应的测试规范和测试用例。此外，和分析需求时一样，我们还要了解测试评审指标及制定评审文档。

## 功能实现

根据需求分析中计算机视觉和自然语言处理算法模块的详细特征及其使用模式，对相关的算法分别进行实现，并提供简单、灵活、通用的API，方便研究人员使用。

## 其他软件管理工具学习

在研究过程中，我们会用到项目管理和项目配置工具，还有建模工具等，在整个的开发的过程中，我们都将伴随开发工具的学习，不再单独为研究工具而花费时间，可以通过其他阶段的时间剩余来合理安排。

## 总体进度计划

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工期 | 开始时间 | 结束时间 |
| **调研任务** | 2 | 3月6日 | 3月7日 |
| 上课 | 1 | 3月6日 | 3月6日 |
| 分配任务，调研项目 | 1 | 3月7日 | 3月7日 |
| **项目确定以及介绍** | 5 | 3月9日 | 3月13日 |
| 选题会议 | 1 | 3月9日 | 3月9日 |
| 调研PyTorch软件框架 | 1 | 3月9日 | 3月9日 |
| 调研进度管理工具 | 3 | 3月9日 | 3月11日 |
| 调研测试工具 | 3 | 3月9日 | 3月11日 |
| 学习软件需求分析文档撰写要求 | 2 | 3月12日 | 3月13日 |
| **项目计划书** | 3 | 3月13日 | 3月16日 |
| 上课 | 1 | 3月13日 | 3月13日 |
| 引言 | 1 | 3月13日 | 3月13日 |
| 项目概述 | 1 | 3月14日 | 3月14日 |
| 软件系统结构分析 | 1 | 3月15日 | 3月15日 |
| 预期结果 | 1 | 3月16日 | 3月16日 |
| 进度管理工具使用学习 | 1 | 3月16日 | 3月16日 |
| **软件需求分析** | 5 | 3月16日 | 3月21日 |
| 软件模块分析 | 4 | 3月16日 | 3月20日 |
| 测试需求分析 | 4 | 3月16日 | 3月20日 |
| 软工规范分析 | 4 | 3月16日 | 3月20日 |
| 扩展功能分析 | 4 | 3月16日 | 3月20日 |
| 需求分析会议 | 2 | 3月20日 | 3月21日 |
| **软件需求评审** | 3 | 3月22日 | 3月24日 |
| 修改软件模块需求 | 2 | 3月22日 | 3月23日 |
| 修改测试需求 | 2 | 3月22日 | 3月23日 |
| 修改软工规范需求 | 2 | 3月22日 | 3月23日 |
| 修改扩展功能需求 | 2 | 3月22日 | 3月23日 |
| 需求评审会议 | 1 | 3月24日 | 3月24日 |
| **软件需求复评审** | 2 | 3月25日 | 3月26日 |
| 确定软件模块需求 | 2 | 3月25日 | 3月26日 |
| 确定测试需求 | 2 | 3月25日 | 3月26日 |
| 确定软工规范需求 | 2 | 3月25日 | 3月26日 |
| 确定扩展功能需求 | 2 | 3月25日 | 3月26日 |
| **软件实现** | 15 | 3月27日 | 4月10日 |
| **软件产品改进与展示** | 15 | 4月10日 | 4月24日 |
| 软件改进 | 14 | 4月10日 | 4月23日 |
| 软件演示 | 1 | 4月24日 | 4月24日 |
| **软件测试分析** | 6 | 4月25日 | 4月30日 |
| 测试需求分析 | 6 | 4月25日 | 4月30日 |
| **软件测试评审** | 2 | 5月4日 | 4月5日 |
| 修改测试需求 | 2 | 5月4日 | 5月5日 |
| **软件测试复评审** | 2 | 5月6日 | 5月7日 |
| 确定测试需求 | 2 | 5月6日 | 5月7日 |
| **软件演示与测评** | 4 | 5月9日 | 5月12日 |
| **综合实验分析** | 4 | 5月13日 | 5月16日 |
| **综合实验总结** | 4 | 5月17日 | 5月20日 |
| **软件进度计划与控制** | 91 | 3月6日 | 6月4日 |
| **工作量估计与统计分析** | 91 | 3月6日 | 6月4日 |
| **配置管理** | 91 | 3月6日 | 6月4日 |

## 项目控制计划

暂无。

# 关键问题

暂无。

# 专题计划要点

暂无。