

硕 士 研 究 生 读 书 报 告



题目 关于continual learning的思考

作者姓名 陈文儒

作者学号 21951428

指导教师 张志猛

学科专业 软件工程

所在学院 软件学院

提交日期 2020年10月1日

The Implementation of Remote Procedure Call Service

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Erteng Liu

By

Ye Li

Zhejiang University, P.R. China

2019

摘要

本文重点探讨了软件开发中一个至关重要的问题——软件需求问题。软件开发人员及用户往往容易忽略沟通的重要性，导致软件开发出来后，不能很好地满足用户的需要，造成人力、物力和资源的浪费，还使软件性能深受影响。所以在软件开发前期对软件需求的分析就显得十分重要，本文以软件需求工程的角度并具体从软件需求开发、软件需求管理来阐述需求分析在系统设计中的影响和作用。

**关键词**：软件需求， 需求分析，系统设计

Abstract

The paper discuss the important problem in software development——requirements analysis. Developer and user always ignore the communication, it causes directly the software does not meet the good demands of the user ,and cost a lot of time and money. Moreover, it affects the performance of the software. So, the requirements analysis is important in the early time of the development. This paper mainly discuss the requirements analysis’s influence on the system design from requirements develop, requirement management, requirement program.

**Keywords：**software requirement, requirement analysis, system design

1引言

人在学习新知识的时候，能根据之前的知识很快的学习相似的知识，并且能不遗忘从前的知识。而机器，或者更准确一点说神经网络，在学习新任务的同时会出现一些问题——灾难性遗忘问题（catastrophic forgetting），意思是模型学习了新任务的b，而再回去预测旧任务a时发现预测不准确了。因此，针对此现象，需要提出解决方案，以解决灾难性问题，我们将这种方法称之为持续学习（continual learning）。

continual learning（也叫lifelong learning，incremental learning等等），中文一般称持续学习等等。而本文，在阅读了关于continual learning的相关文献，对continual learning有一个大致的脉络，能为实际应用中的项目落地后的持续学习提供更好的帮助。

continual learning其主要思想是约束梯度的方向，本文介绍的几种方法，都是基于梯度约束，而实现的效果也比较好。

**2 Elastic weight consolidation**

Elastic weight consolidation（EWC）的灵感来自哺乳动物的记忆，研究发现哺乳动物的大脑可能会通过大脑皮层回路来保护先前获得的知识，从而避免灾难性遗忘。实验中，一个小鼠需要记住一个行的技能时，大脑中一些突触就会被加强（单一神经元的树突棘数量的增加）。并且即使进行了后续的其他任务的学习，这些增加了的树突棘能够得到保持，以便几个月后相关能力仍然得到保留。但是当这些树突棘被选择性擦除后，相关的技能就会被遗忘。这表明对这些增强的突触的保护对于任务能力的保留至关重要。

而EWC，这个算法的主要思想是基于上述的发现。具体做法简单概述为：神经网络中并不是每个节点的都对结果有很大影响，在学习新任务时，减轻那些对旧任务影响过大的节点权重，即可达到继续学习的效果。

**2.1 具体方法**

假定目前有两个学习任务A，B。θA θB 是这两个任务的模型中的参数。A任务先学习，得到稳定的结果。这时再学习B任务，为了不让模型遗忘A任务，需要限制θA，使θA限制在一个比较低的错误范围。因此EWC能够在学习新任务的同时，将θA当作一个二次惩罚，就像图1所示。这个过程就像是压弹簧，对A来说，弹簧强度就应该增大，这样只有更大的惩罚才能改动θA，就能更好保留A任务的记忆，而对于B来说，弹簧强度就不变，这样也能更好地记住B任务，从而保留下两个任务的记忆。所有参数的强度都不一样，哪些对A任务影响大的参数，他们的强度应该更大。

那么如何为每个参数选择这个强度呢？

图示

描述已自动生成

**2.2 计算强度**

作者意图是通过概率来计算这个强度的概率。给定一个数据集D，通过θ的先验概率，计算θ关于D的条件概率：

|  |  |
| --- | --- |
|  | （1） |

该公式是基于贝叶斯公式推导出来的：

文本, 信件

描述已自动生成

上述公式的实际上的值是，简单的来说是这个问题的loss值的负数：-L(θ)。

上述只是针对一个任务参数进行推导，假定现在有两个任务A，B。那么这条公式就可以重新推导为：

|  |  |
| --- | --- |
|  | （2） |

表格

描述已自动生成

左边仍然是参数的后验概率（给定全部数据），右边是仅仅依赖于任务B的。而任务A必须要被后验概率吸收。

由于后验概率是很难得到的，因此，作者根据Laplace approximation，将后验概率近似为高斯分布，这个高斯分布是A任务的参数θA得到mean和对角线精度由Fisher information matrix（F）的对角线给出。F有被证明以下三点重要特性：a)F等价于loss函数的二阶导数的近似最小；b)他可以仅仅由loss的一阶导数求得，因此这也很容易能得到他;c)他保证了半正定。

|  |  |
| --- | --- |
|  | （2） |

Fisher information是一次观测值所能提供的关于未知参数θ的信息量期望值的一种度量。相当于是弹簧的强度的一个度量。

当B任务训练完时，来了一个C，此时又可以将A，B当成就任务，以此类推。

**2.3 有监督训练**

作者设置了一个多层的全连接的神经网络，用来训练多个有监督任务。将数据进行洗牌以及做小batch，每个训练的任务都有固定的训练次数，并且不可增加。

图A中，可以看到EWC表现的非常好，能记住之前的任务，但是SGD就是在每个任务中有遗忘之前任务的迹象，而L2正则化发生了灾难性遗忘（在训练任务C时，任务B发生的）。

作者又把任务SGD拿出来单独比较，增加任务数量以后，这个记忆直线下降，图B所示。

图C表示的是，任务相似度对fisher矩阵重叠的影响。

图形用户界面

描述已自动生成

**3 其他相关做法**

**3.1 LWF**

LWF其名Learning without Forgetting，其主要思想是通过知识蒸馏的方式处理灾难性遗忘问题。

如图所示，正常训练的模型如此。用θs表示前面特征提取的模型参数，用 θo表示用来分类的层的参数。

图表

描述已自动生成

文章先是列举了现有的方案：如下三种，以及自身的LWF。

图形用户界面

描述已自动生成

Fine-tuning和Feature Extraction实际上适用于相似任务，即数据集基本相似。而Joint Training必须要之前的老数据集，这在一些条件下是不被允许的，例如数据是需要隐私保护的，数据太大而不可能同时保存下来。

那么如何实现呢?

首先是预训练， 先让新增加的θn收敛，然后再使用知识蒸馏的方式联合学习。

|  |  |
| --- | --- |
|  | （3） |

这是新任务的Loss，也就是正常的交叉熵loss（MSE）。那么知识蒸馏是通过加入原始模型的loss做的，也就是：

文本

描述已自动生成

这里是指当前模型生成的label，是指原模型生成的label。这是一个对交叉熵loss改进的loss函数。其中：

文本

描述已自动生成

目的是为了增大样本数较少的weight。

下面用一个伪代码算法表示如下过程：

文本, 信件

描述已自动生成

过程很清晰，最后一行，就是本文的关键。R是指一些正则化，而两个Loss函数上面已经解释过了。最后有个λ参数，这个参数决定了新旧任务在训练过程中的重要性之比，一般为1，这样两头都能兼顾。

**4 小结**

参考文献

[1] Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the national academy of sciences, 2017, 114(13): 3521-3526.

[2] Li Z, Hoiem D. Learning without forgetting[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(12): 2935-2947.