基于卷积神经网络的恶意 URL 检测模型

董青青 2018213688

网络空间安全学院 2018211805 班

摘要

在网络带给人快捷和便利的同时，也浮现出了各种各样的网络安全问题。恶意 URL 作为多种网络攻击的媒介，已经成为网络的公害。传统的黑名单检测方法准确率低，基于机器学习的算法需要人工提取特征值，存在时间资源开销大等问题。考虑到卷积神经网络在自然语言处理中取得的优异成绩，本文提出一种基于卷积神经网络的恶意 URL 检测方法。通过数据预处理、词嵌入和模型训练，自动提取出 URL 字符串中的隐含特征。实验结果表明，该模型初步完成了检测恶意 URL 的目的，但仍存在误报率高和召回率低的问题，需要未来进一步的学习和研究。

关键字：恶意 URL;卷积神经网络;网络空间安全;深度学习

1 绪论

1.1 研究背景和意义

随着互联网的迅速发展，网络已然渗透到了现代人生活的方方面面，在网络带给人快捷和便利的同时，也浮现出了各种各样的网络安全问题。根据瑞星发布的《2019 年中国网络安全报告》显示，2019 年瑞星“云安全”系统在全球内共截获恶意 URL 总量 1.45 亿个，挂马类网站有 1.2 亿个，钓鱼类网站共有 2,454 万个。恶意 URL 总数最多的国家是美国,约有 5,612 万个，其次是德国--848 万个和墨西哥--649 万个，分别为二、三位。防不胜防的恶意网站时刻威胁着用户的个人信息以及财产安全，给人们造成了极大的困扰。

恶意 URL 通过诱导不知情用户点击访问有害网址，达到窃取隐私信息，执行恶意代码等目的，因此在网络攻击的过程中扮演着十分重要的角色。卡巴斯基实验室 2019 年年度报告显示，2019 年，共有 2.73782113 亿个 URL 被 Web 防病毒组件识别为恶意 URL，而同一指标在去年为 5.54159621 亿。这个数据并不能说明恶意 URL 数量上较去年下降。相反，可以想到，恶意 URL 愈发具有伪装性，更难被安全软件识别出来。无独有偶，根据 Webroot 网络安全公司 2019 年的威胁报告，40%的恶意 URL 拥有合法的域名，而合法的网站也经常会被注入恶意的内容。因此，研究出一种能有效检测恶意 URL 的解决办法，对维护网络安全的健康发展有着十分重要的意义。

1.2 研究现状

目前恶意 URL 检测方法大致可以分为三类：黑名单技术、基于内容、机器学习、深度学习。

1.2.1 黑名单技术

黑名单(Blacklist)技术是一种较早的恶意 URL 检测方法，是许多安全软件和浏览器所采用的方法，例如 Chrome 浏览器的网页检查功能。当用户即将进入一个新的网站，浏览器会在黑名单中根据其 URL 的特征检测它是否为恶意 URL，若确定该网站有害，会立即提醒用户。黑名单记录并维护着大量已经被确认或者根据启发式规则建立的恶意 URL 列表，它可以定期更新，或者根据名单上已经存在的父 URL 产生新的子 URL。黑名单技术响应及时，是一种轻量级的检测工具。然而因为其只能检测已经存在的 URL，当检测样本是一个全新的 URL,黑名单技术将难以应对，因此漏报率很高。

1.2.2 内容检测

内容检测技术是根据网页内容，例如网页的标签、CSS 布局、图片等静态特征或者 javascrIPt 代码、网页跳转、界面响应等动态特征进行检测。检查网页是否有恶意代码，或者异常行为，以判断该网站是否为恶意网站。还有的研究者采用沙箱技术，动态地对 URL 进行监控和分析。它通过模拟真实环境下用户点击 URL 的过程，实时监控网页的行为，达到识别恶意网站的目的。内容检测的方法依赖于对于恶意 URL 内容特征的把握，需要检测的方面比较综合，因此运行起来开销较大，检测时间长，同时存在漏报率和误报率比较高的问题。

1.2.3 机器学习

随着人工智能的兴起，利用机器学习(Machine Learning)检测恶意 URL 引起了学者们的注意。机器学习分为有监督式学习和无监督式学习，监督式学习算法根据带有标签的数据训练模型，让模型能够预测待测样本的分类情况；无监督学习从不代标签的输入数据中找出其隐藏的特征信息,学习表示该类数据。基于机器学习的恶意 URL 检测方法首先需要收集大量 URL 数据集样本，手动提取出 URL 的域名、WHOIS 信息，

URL 长度、每个字符出现的频率，关键词、参数数量等特征值，利用自动向量机(SVM)，决策树、朴素贝叶斯(Naïve Bayes)等有监督算法或 KMeans 等无监督算法，训练出特征模型，最后将该模型运用于待检测 URL 进行预测，产生分类结果。经实验验证机器学习方法的准确率显著低高于黑名单方法和内容检测方法，然而该方法对特征的选取极为敏感，模型的好坏很大程度取决于人工提取特征的好坏，直接影响着测试结果的准确率，容易出现过拟合的现象。因此，机器学习算法还需要花费大量的时间提取特征和实验。

1.2.4 深度学习

深度学习(Deep Learning)是机器学习算法的一个分支，近年来深度学习已成为人工智能的热点研究领域。深度学习应用广泛，尤其是在图像识别、文本分类、自然语言处理等方面取得了较好的成绩。利用深度学习检测恶意 URL 仍在初始阶段，恶意 URL 样本的稀少成为研究的一大困难。相比于传统的机器学习算法，深度学习不需要依赖人工经验提取样本特征，只需要将原始的文本或图片数据集，转化成模型可以识别的矢量表达，通过多层神经网络模型，训练提取出特征并分析。使用时将待检测的 URL 输入模型中进行预测，就能判断该待测样本是否为恶意 URL。

1.3 论文组织结构

本文共分为五章，主要研究了基于卷积神经网络的恶意 URL 检测模型，各章的主要内容如下：

第一章绪论，主要介绍了本文的选题背景及意义，重点分析介绍了常见的几种恶意 URL 检测的技术。

第二章相关工作，展示了国内外研究者对于恶意 URL 检测做出的研究和工作，分析了各种方案的优势和不足，并基于此提出一种基于深度学习的检测方案。

第三章介绍了本文提出恶意 URL 检测方案的总体结构，分为模型训练和 URL 预测两部分。

第四章具体介绍了卷积神经网络模型的实现，模型由输入层、词嵌入层、卷积层、池化层、全连接层、Softmax 层组成，同时详细地介绍了每一层的设置和功能。

第五章呈现了该算法的实验结果和分析工作，数据集的来源和组成，实验参数的设置、给出了实验的准确率、漏报率等参数，并且基于实验结果说明了该模型的优点和不足。

2 相关工作

目前，已经有许多针对识别恶意 URL 进行的研究。Zhang 等(2007)[1]开发了一种基于网页内容检查钓鱼网站的方法：CANTINA，该方法使用 TF-IDF 技术，选出最能代表网站的五个关键词，根据网站的成立时间、访问量、和与正规网站的相似度等特征，同时配合启发式的域名、IP 地址等特征综合判定该网站是否为恶意链接。该方法实现简单，但依赖搜索引擎实现，且验证过程较为原始，误判率和漏判率都比较高；

Prakash 等(2010)[2] 则提出了 PhishNet:：一种基于启发式黑名单的检测方法，该方法通过将黑名单中的父 URL 按照一系列的启发式原则生成潜在的恶意子 URL，验证其有效性，并加入黑名单中。从 IP 地址、域名、目录结构、品牌名 4 个方面匹配待检测 URL，从而产生预测结果。启发式原则很好的解决了传统黑名单检测无法预测新产生 URL 的问题，但仍需要依赖原有黑名单，与日益多变复杂的恶意 URL 相比仍存在一定的局限性。

随着机器学习的流行，越来越多的研究者开始运用机器学习来预测 URL 是否为恶意的。Ma 等(2009)[3] 从 URL 中提取出 who-is 信息、域名、DNS 记录等多组特征，

使用有监督学习算法，同时比较了 Bayes，SVM，LR 等分类器对预测结果的影响。实验表明，机器学习弥补了黑名单方法难以识别新产生 URL 的不足，同时无须像基于行为式的检测方法那样点开有潜在威胁的网站，拥有更高的准确率和安全性。但是机器学习算法对特征的选取极为敏感，模型的好坏很大程度取决于人工提取特征的好坏，直接影响着测试结果的准确率。此外，机器学习算法还需要花费大量的时间提取特征和实验。

大数据时代的来临，使得基于深度学习的检测方法逐渐成为目前研究的主流。常用的深度学习算法有卷积神经网络(CNN)和循环神经网络(RNN)。潘司晨等(2019）[4] 提出一种基于卷积神经网络的检测模型算法，通过词嵌入(Word Embedding)对原始 URL 的字符进行词向量编码,从而转换为模型可识别的二维向量,再通过卷积神经网络模型,自动学习高层次的特征。Le 等(2018) [5]基于字符级卷积神经网络(char-CNN)和词语级神经网络模型(word-CNN)提出 URLNet:将 URL 数据集并行表示为字符级和词语级的向量，传入卷积神经网络中自动学习并提取出文本的特征，最后将待测 URL 向量放入模型中产生预测结果。张慧等(2019） [6]等优化了 CNN 模型结构，提出一种基于卷积神经网络的 URLs 特征自动提取方法。Luo 等(2019) [7]使用自编码器将 URL 预处理后转化为一张特征图像，再放入 CNN 中进行预测，使特征对研究者来说更为直观明显。同时，Peng 等(2019)[8] 采用注意力机制，用工具自动提取出文本、域名 URL 中等特征，并且结合 LSTM 和 CNN 两种模型预测结果。王欢欢等(2019） [9] 提出一种基于双向 Ind-RNN 的恶意 URL 分析与检测算法.通过对恶意 URL 分析与检测特点的研究,提取主机信息特征和 URL 信息特征.把主机信息特征与 URL 信息特征相融合,并利用 Bi-IndRNN 算法对恶意 URL 进行分析与检测。深度学习方法在 URL 检测中都表现出

了优秀的成果，且节省了手工提取特征的时间和精力，能有效识别出新生成的恶意网站。

综上，根据前人的研究，本文提出了一种基于卷积神经网络的恶意 URL 检测方法。

3 框架概述

本文所提出的基于卷积神经网络的恶意 URL 检测系统由两大部分组成：模型训练和样本测试。在训练阶段，系统将收集到的有标签的数据集进行预处理，送入神经网络模型中进行训练，得到训练好的卷积神经网络检测模型。在测试阶段，同样将待检测的 URL 样本经过预处理后送入训练好后的 CNN 模型中进行预测，测试结果将分为正样本和负样本两部分。框架结构如图 3-1 所示

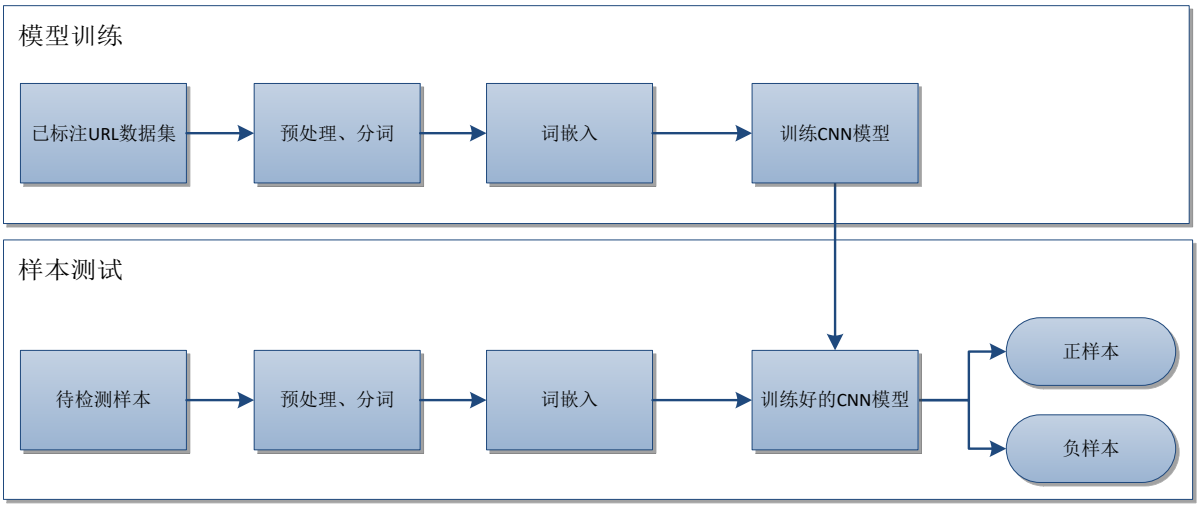

图 3-1 系统框架

4 实现恶意 URL 检测模型

4.1 数据预处理

URL 即统一资源定位符，由协议、域名、主机名和可选项构成。下面以 http://www.imailtone.com/WebApplcation1/WebForm1.aspx?name=tom&;age=20#resume 为例，其格式图 4.1 所示

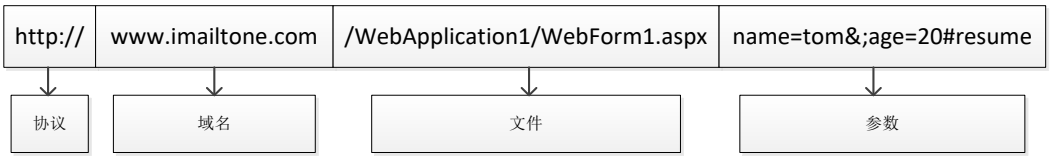

图 4.1 URL 格式

对于正常 URL 和恶意 URL，其协议部分大部分相同，为“http”或“https”，因此我们首先将收集到的 URL 协议字段去除。为了降低 URL 中大小写混用导致对训练的影响，我们将所有字符统一转换成小写，接着过滤 URL 中的特殊符号。我们把‘.’, ‘,’, ‘=’, ‘;’, ‘:’, ‘?’, ‘-’, ‘#’,等分隔符都替换为符号‘/’, 再根据‘/’符号将 URL 分隔成词语。分词之后，词语的数量大多小于 100。因此我们把最大词语的数量设置为 100，对于长度大于 100 的 URL 舍弃 100 以后部分，如果不足 100，则采取向后填充特殊符号<PADDING>的方式补齐。通过预处理后得到的 URL 为一串长度为 100 的词语，紧接将其送入词嵌入模型训练，生成 URL 的词向量表示。下面以 http://www.abc123.com?URL=http://www.def456.com?id=5&userName=Admin 为例，说明预处理的主要过程。操作过程如**错误!未找到引用源。**所示

表格 4. 1 样本预处理

操作	处理后的样本
删除协议字段	www.abc123.com?URL=www.def456.com?id=5&userName=Admin
转换成小写	www.abc123.com?URL=www.def456.com?id=5&username=admin
替换特殊符号	www/abc123/com/URL/www/def456/com/id/5/username/admin
分词填充	[‘www’, ‘abc123’, ‘com’, ‘URL’, ‘abc123’, ‘ www’, ‘def456’, ‘com’, ‘id’, ‘5’, ‘username’, ‘abc123’, ‘admin’, ‘<PADDING>’, ‘<PADDING>’,……, ‘<PADDING>’]

4.2 词嵌入(Word Embedding)

4.2.1 one-hot 编码

在自然语言处理中，经常要通过词嵌入(Word Embedding)将中文、单词、符号等处理成机器能够识别的向量形式，一般通过独热(one-hot)实现，或者通过 word2vec 词向量表示。one-hot 编码根据词在词典中所在的位置，向量的分量只用 0 或 1 表示。

假设一个词在词典中的位置为i，词典中不同词的数量（词典大小）为 N，那么它的 one-hot 编码是一个除第i位是 1,其他 N-1 位全是 0 的 n 维向量。通过 one-hot 编码,每一个词都可以被映射成长度为 N 的向量。下面为 one-hot 编码示例，如图 4.2.1 所示。

这种方法虽然实现简单，但无法准确地表示不同词之间的相似度，若语料库过

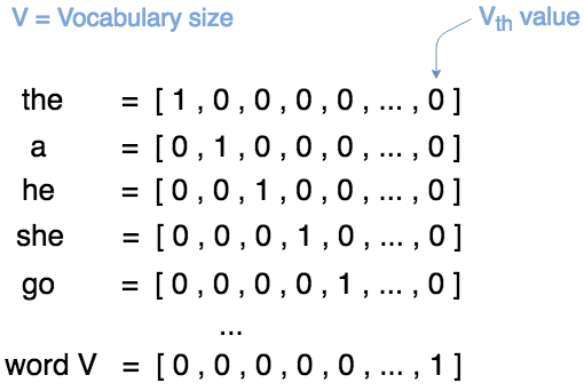

图 4.2.1 one-hot 编码

大，还会导致词向量过长的问题。所以本文采用第二种 word2vec 方式实现 URL 的词嵌入过程。

4.2.2 word2wec 模型

Word2vec 通过训练将每个词映射成 K 维稠密、连续的实数向量，通过空间距离的计算来判断两个词是否存在语义相关性。 它有两种训练模型，分别是跳字模型(skip-gram)和连续词袋模型(CBOW)。跳字模型是用一个词语作为输入，来预测它的上下文词语。相反，连续词袋模型则是拿一个词语的上下文作为输入来预测这个词语本身。图 4.2.2 为两种方式的模型示意图。本文采用连续词袋模型进行词嵌入。

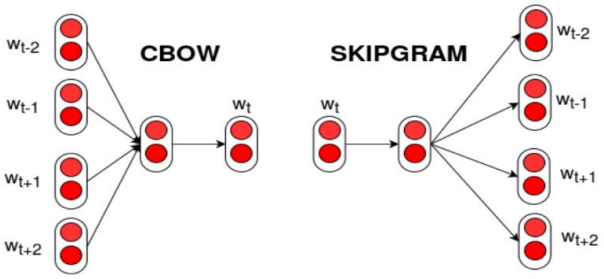

图 4.2.2 word2ec 模型

通过词嵌入层后，每一个单词都被表示为一个长度为 64 的实数向量，因此每一条 URL 都被映射为 100*64 的二维向量。接着将带有标签的词向量送入卷积神经网络中进行训练。

4. 3 卷积神经网络

基于卷积神经网络的恶意 URL 检测模型主要由词嵌入层、卷积层、池化层、全连接层和 Softmax 分类器六层组成，模型结构图如图 4.3 所示：

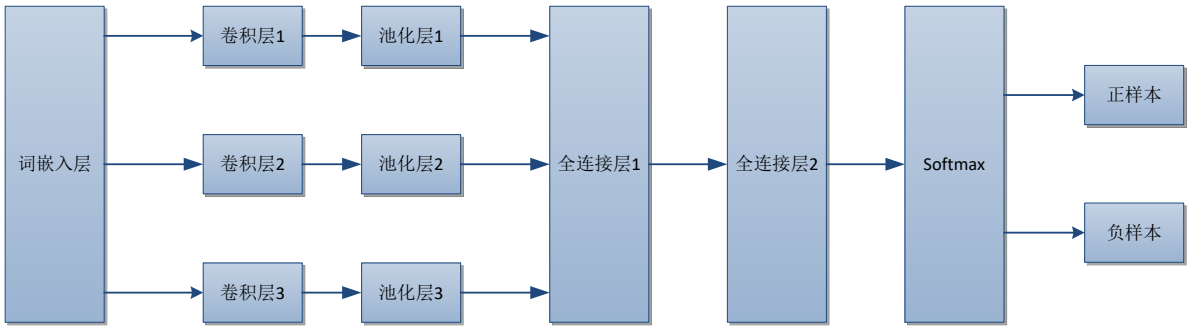

图 4.3 CNN 模型结构

该模型由三大部份组成：第一部分是模型的输入，利用 word2vec 词嵌入模型将原始 URL 转换为 100*64 的矢量矩阵。第二部分是自动特征提取过程，通过卷积池、池化层构成。第三部分是输出层，通过全连接层和 Softmax 分类器对样本 URL 进行分类。以下将分别对每一层的结构进行详细介绍。

4.3.1 词嵌入层

模型的第一层是词嵌入层，其作用是将一条 URL 作为输入数据传入模型中，经过词嵌入将其表示为向量矩阵，并与下一层的神经元相连接。经过该层后，每条 URL 数据转换成由单词向量组成的矩阵$u, u \in R^{100\times64}$,标签y转化成向量[0,1]（恶意 URL）或[1,0](正常 URL).

4.3.2 卷积层

第二层是卷积层，作用是将输入层传入的数据进行卷积操作，完成局部特征的自动提取功能。为了使模型看到不同数量的字符，更好地提取 URL 中词和上下文之间的关系，实验中卷积层使用的是 3 种不同大小的卷积核，宽度为 100，高度 h 分别为 3，4，5，每种类型的卷积核有 128 个，共有 384 个卷积核。对输入层长度为 h 的字符窗口进行卷积操作后再经过激活函数 f 得到新的特征$c_i, c_i = f(w \cdot x_{i:i+h-1} + b)$。

其中$w \in R^{h \times k}$ 是卷积核，$x_{i:i+h-1}$是第 i 个字符到第$i + h$ 个字符组成的矩阵，b 是偏差项 $bias$，激活函数 f 是 $ReLU$函数, $ReLU(x) = \begin{cases} x, x > 0 \\ 0, x \leq 0 \end{cases}$. 相比于其它激活函数来说，由于 ReLU 非负区间的梯度为 0，因此不存在梯度消失的问题，从而使得模型一直以正常的速度收敛。

通过卷积操作后得到一个列为 1，行为 $(L - h + 1)$的 feature map 特征，记为 c，$c = (c_1, c_2, \ldots c_{L-h+1})$,这里的 L=100，是输入矩阵的长度。接着将 feature map 送入下一层池化层做最大池化。

4.3.3 池化层

第三层是池化层，它的功能是对特征做进一步的提取，把最重要的特征提取出来，起到特征降维、减小模型过拟合和提高训练速度的作用。[11]池化层有最大池化和平均池化两种提取特征的方式。平均池化提取区域内特征向量的平均值，而这里使用的是最大池化，取出 feature map 向量中的最大值作为最重要的特征，即$c = max\{c\}$。最终对于每个 feature map 池化后都得到一个一维向量$m \in R^{1 \times 384}$，把这个向量送入全连接层。

4.3.4 全连接层

模型最后部分是两个全连接层。第一层大小是 384*128，第二层大小是 128*64，激活函数都为 RELU。它作用是对池化层得的特征进行一个加权和，将特征空间通过线性变换映射到样本标记空间。最后一层全连接层通过使用 Softmax 分类器得到各个分类的概率。

Softmax 函数能将一个任意一个K维实数向量映射到另一个K维实向量中，使得每个元素的范围落在(0,1)之间，并且元素和为 1。本文研究的问题属于二分类模型，Softmax 分类器的输出是一个向量元素介于(0,1)之间的 2 维向量，且元素累加和为 1 。隐藏层函数的输出值代表了 URL 属于正常 URL 分类和恶意 URL 分类的概率，概率最大的，为文本分类的结果。

为了防止出现过拟合的现象(overfitting),连接层设置了 Dropout 来降低模型的泛化误差。Dropout 是非常有用和成功的一种技术，能够有效控制过拟合的问题。在训练神经网络时，神经网络的一些神经元会按照$keep_prob$的概率随机被删除，而在测试时，所有的神经元都会被激活，每一个单元的参数要预乘以$keep_prob$。在我们的模型中，$keep_pro$b 值被设置为 0.5，以提高深度学习的性能。

4.4 模型训练

模型总体训练过程如图 4.4 所示。

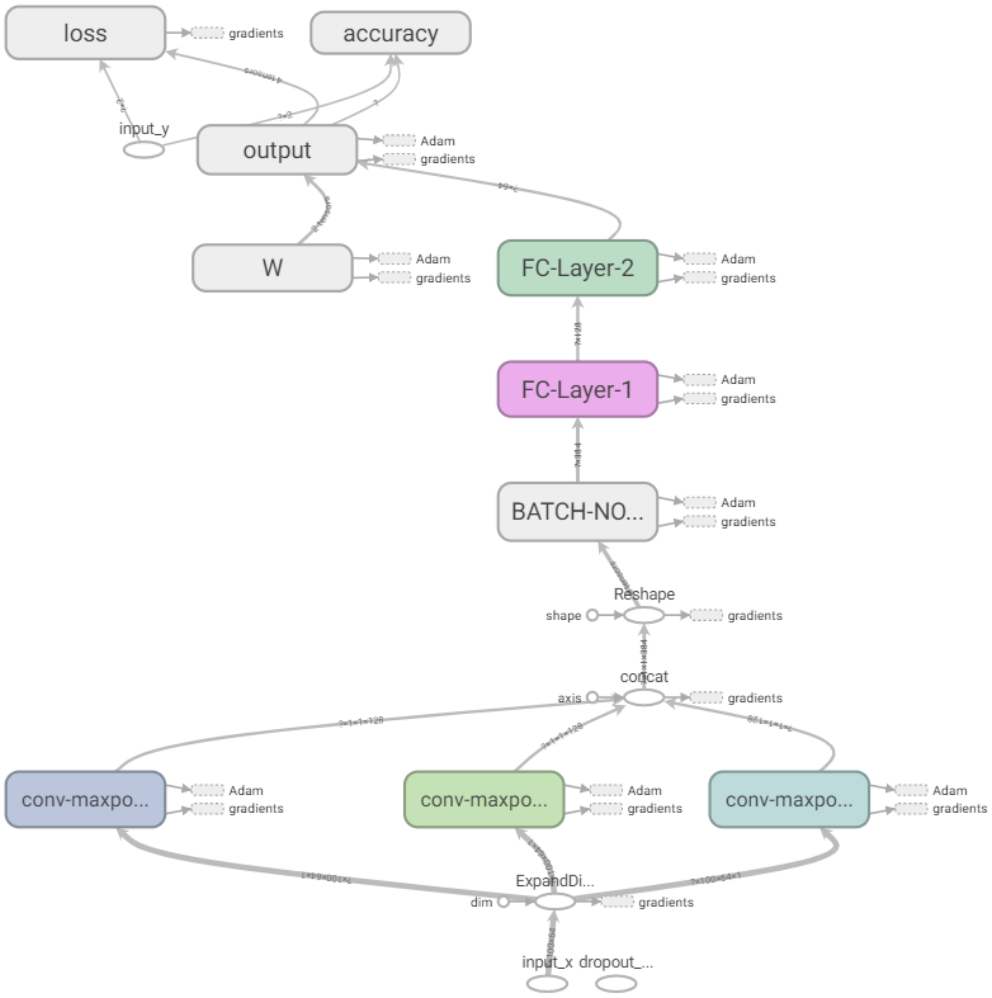

图 4.4 模型训练过程

4.4.1 训练方法

本实验数据集的样本数目较大,总样本数量约为 41 万条，若每次训练都使用所有训练集来计算损失函数，训练时间和计算量将非常大。因此，本文采用小批量随机梯度下降算法。每次更新参数时，仅根据当前的n个样本，能够提高训练的速度，同时减少内存空间的使用。常用的优化算法有 Adam 算法、Momentum 算法和 RMSprop 算法等等，本文采用 Adam 优化器来训练 CNN 模型，它是一种能够代替传统随机梯度下

降算法的一阶优化算法，它能依据训练数据迭代地更新神经网络权重，它有计算效率高，实现简单，内存占用少等优点。[12]

4.4.2 参数初始化

模型各参数初始采用 Xavier 初始化，该方法考虑了随机初始化的一般梯度分散问题，通过计算合适的随机初始化范围，在网络正向和反向传播时，每一层输出值的方差与前一层的方差一致，在一定程度上避免了网络深化引起的梯度分散问题。其特点是：在正向传播的过程下，激活值的方差保持不变。在反向传播过程中，状态值梯度的方差不变。Xavier 初始化实现的是下面的分布。

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right]$$

4.4.3 损失函数

损耗函数用于测量模型的实际输出与期望结果之间的差距，本文采用经典的交叉熵损失函数，$l = -\sum_i y_i' log(y_i)$其中，$y_i'$代表真实概率，$y_i$是预测的某个类别的概率。在训练结果接近真实值时，其仍然可以保持在高梯度状态，保持模型以正常速度收敛下去。

5 实验结果与分析

5.1 数据集获取

本文所使用的实验数据集来源于公开数据集 PhishTank、Kaggle 和 google，总共收集了 411185 条 URL，其中训练集正样本（恶意 URL） 66384 条，负样本（正常 URL） 344793 条，正负样本比为 1:5。测试集样本总量是 75656， 正样本（恶意 URL）18019 条，负样本（正常 URL） 57637 条，正负样本比为 3:10. 样本示例如表格 5. 1 所示。

表格 5.1 数据样本

Example	Label	number
diaryofagameaddict.com	bad	66384
linkedin.com/pub/lucpinard/11/ab5/880	good	344793

5.2 实验环境

本实验环境配置如表格 5.2 所示。

表格 5.2 实验环境

参数	数值
操作系统	Windows 10 64 位家庭版
CPU	Intel(R) Core(TM) i5-8300H CPU@ 2.30 GHZ 2.30GHZ
GPU	NVIDIA GTX 1050 4G
内存	8.00GB
硬盘	128G HDD
Python	3.6.1
Tensorflow	2.0.1

5.3 参数设置

模型参数的设置决定了算法的准确率、性能，对检测结果起到了至关重要的作用。本文在大量实验的基础上，测出了实验的最优参数，如表格 5.3 所示。

表格 5.3 实验参数

参数	数值
batch_size	64
num_epochs	200
test_size	0.25
Dropout	0.5
learning_rate	0.01
优化器	Adam

5.4 评估方法

评价分类模型的常见指标有正确率(Accuracy)、准确率(Precision)、召回率(Recall)、和 F1 值(F1-score)、假正率(False Positive Rate)和假负率(False Negative Rate)等.

- 正确率是最普遍的评价指标，是分类正确的数据占总体样本的比例，在恶意 URL 检测这样的不平衡的数据集上，单靠正确率很难全面评价模型的分类能力。
- 召回率是被分类正确的正样本数占总体正样本数的比例，用来评估模型对正样本的检测水平。
- F1 值 同时兼顾了分类模型的正准率和召回率，是模型正准率和召回率的加权平均值。
- 假正率也是误报率，指的是被分类错误的负样本占负样本总数的比例，用来评价模型对负样本的检测水平。
- 假负率也是漏报率，指的是被分类错误的正样本占正样本总体的比例，用来评价模型对负样本的敏感度。

因为恶意 URL 往往具有严重危害性，因此在比较模型好坏的过程中，优先考虑召回率和假正率，其余指标作为参考。上述各指标的含义和计算方法如表 5.4 所示。

表格 5.4 各评价指标的含义或计算方法

指标	含义或计算方法
TP	预测结果为正样本，实际也为正样本
FP	预测结果为正样本，实际为负样本
TN	预测结果为负样本，实际为负样本
FN	预测结果为负样本，史记为正样本

正确率	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
准确率	$Precision = \frac{TP}{TP + FP}$
召回率	$Recall = \frac{TP}{TP + FN}$
F1 值	$F1 = \frac{2 * Precision * Recall}{Prcision + Recall}$
假正率	$FPR = \frac{FP}{TN + FP}$
假负率	$FNR = \frac{FN}{TP + FN}$

5.5 实验结果与分析

5.5.1 训练结果

图 5.5.1-1 和图 5.5.1-2 分别是训练过程中损失函数和正确率变化曲线。

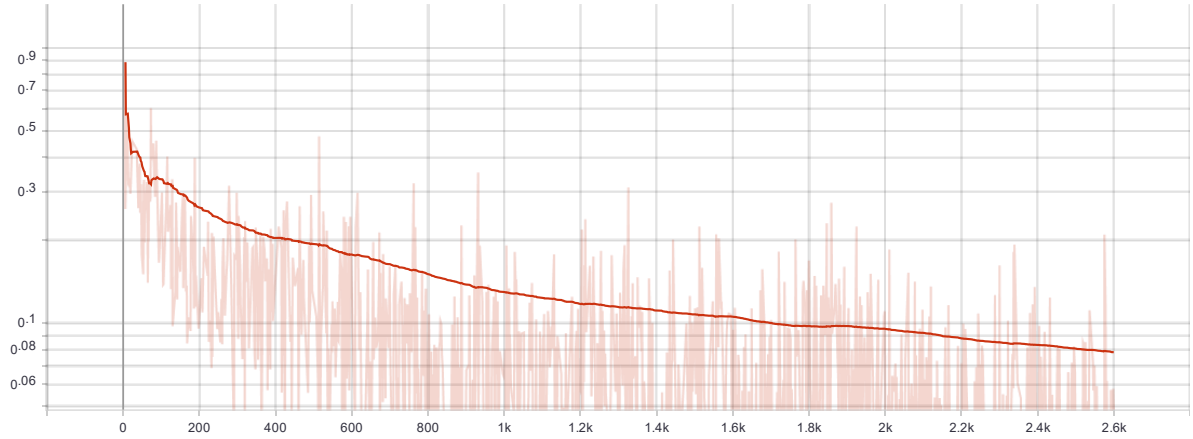

图 5.5.1-1 Loss 函数曲线

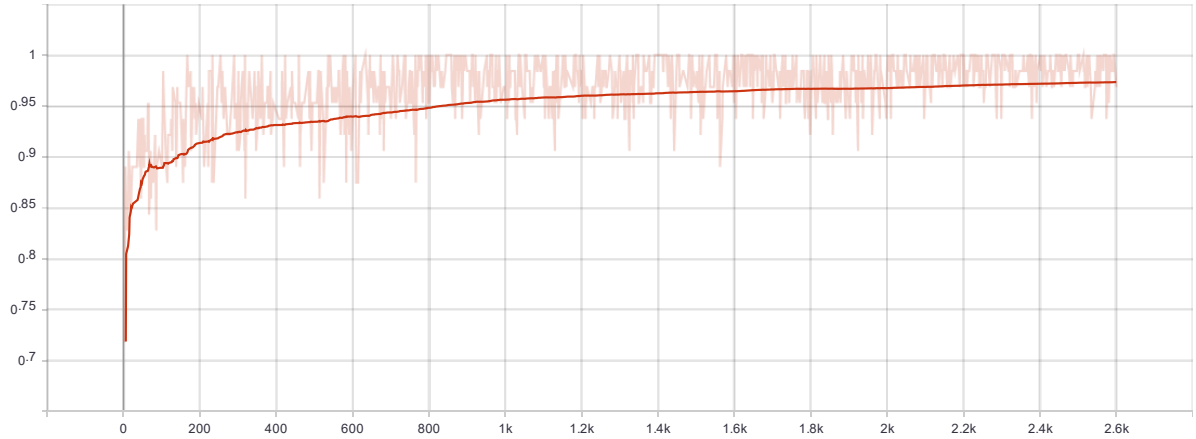

图 5.5.2-2 Accuracy 函数曲线

从图中可以看到，随着迭代过程的增加，loss 函数不断下降，最终趋于 0.08，说明训练过程中参数收敛正常。而正确率随着迭代的过程不断增加，最终趋于 0.96，表现出了较高的正确率。

5.5.2 测试结果

本实验测试数据共有 75656 条，其中正样本 18019 条，负样本 57637 条，通过实验测得该模型各评价指标如表格 5.5.2 所示。

表格 5.2.2 实验结果

评价指标	值
TP	8192
FP	9827
TN	52016
FN	11332
正确率	0.79579
准确率	0.59297
召回率	0.45463
F1 值	0.51467
假正率	0.09755
假负率	0.54536

从实验结果可以看出，在测试集上该模型的正确率约达到 80%，低于训练正确率 96%，说明该模型存在过拟合的问题。F1 值为 51.5%，误报率（假负率）偏高，漏报率（假正率）较低，初步符合了检测恶意 URL 的目的。但是该模型存在误报率高和召回率低的问题，需要进行更多更深入的研究和学习。

结束语

随着互联网的不断发展，恶意 URL 的检测引起了人们的重视，本文提出一种基于卷积神经网络的恶意 URL 检测方案，并利用 Python 和 Tensorflow 进行代码实现，充分利用了卷积神经网络在短文本分类方面的优点。实验表明，本文提出的方案在正确率、漏报率两个方面有比较好的表现，但是召回率较低，误报率高，模型存在过拟合，这也是未来的改进工作方向。

参考文献

[1]. Yue Zhang, Jason I. Hong, and Lorrie F. Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In Proceedings of the 16th international conference on World Wide Web (WWW ‘07). Association for Computing Machinery, New York, NY, USA, 639–648. DOI:https://doi.org/10.1145/1242572.1242659

[2]. P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," 2010 Proceedings IEEE INFOCOM, San Diego, CA, 2010, pp. 1-5.

[3]. Ma, Justin & Saul, Lawrence & Savage, Stefan & Voelker, Geoffrey. (2009). Beyond blacklists: learning to detect malicious Web sites from suspicious URLs. 1245-1254. 10.1145/1557019.1557153.

[4]. 潘司晨,薛质,施勇.基于卷积神经网络的恶意 URL 检测[J].通信技术,2018,51(08):1918-1923.

[5]. Pham, Quang & Sahoo, Doyen & Hoi, Steven. (2018). URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection.

[6]. 张慧,钱丽萍,汪立东,袁辰,张婷.基于 CNN 和多分类器的恶意 URLs 检测[J].计算机工程与设计,2019,40(10):2991-2995+3019.

[7]. Luo, Chaochao & Su, Shen & Sun, Yanbin & Tan, Qingji & Han, Meng & Tian, Zhihong. (2019). A Convolution-Based System for Malicious URLs Detection. Computers, Materials & Continua. 61. 399-411. 10.32604/cmc.2020.06507.

[8]. Peng, Yongfang & Tian, Shengwei & Yu, Long & Lv, Yalong & Wang, Ruijin. (2019). A Joint Approach to Detect Malicious URL Based on Attention Mechanism. International Journal of Computational Intelligence and Applications. 18. 1950021. 10.1142/S1469026819500214.

[9]. 王欢欢,田生伟,禹龙,彭咏芳,裴新军.基于 Bi-IndRNN 的恶意 URL 分析与检测[J].新疆大学学报(自然科学版),2019,36(02):174-181.

[10]. [1]陶文静. 基于卷积神经网络的新闻文本分类研究[D].北京交通大学,2019.

[11]. 白璐. 基于卷积神经网络的文本分类器的设计与实现[D].北京交通大学,2018.

[12]. Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations