卷积神经网络对恶意软件识别的应用

笪 海 周 丽

四川大学锦城学院 计算机与软件学院 四川 成都 611731

【摘 要】随着互联网和社会的现代化发展，人们经常会使用手机的一些应用来方便自己的生活，但由于这类应用大多数需要用户的敏感信息，所以有些不怀好意的人开发恶意软件对用户的手机进行感染，而在这一情况在安卓平台的尤为严重。为了能将下载完毕后的文件检测出是否为恶意软件，本文采取了一种静态检测的方法，通过对下载的 APK 文件进行反编译和解压缩等手段，从中提取出软件权限，将权限转化成特征图后，放入神经网络中进行检测，经过四折交叉检验后，得到的在验证集上的平均准确率为 88.205%。然后选取训练网络的数据集准确率最高的一组作为最终网络的训练集，最后在测试集上进行测试，结果表示，相较于传统的机器学习的方法，使用这种方式的效果更好。

【关键词】卷积神经网络;安卓恶意软件;交叉验证;权限

绪论

由于社会的现代化发展与互联网的应用普及，再加上人们越来越依赖于手机的使用。人们经常会使用手机的一些应用来方便自己的生活，但由于这类应用大多数需要用户的手机号、社交软件账号、身份证号码等敏感信息，所以有些不怀好意的人开发恶意软件对用户的手机进行感染。同时又因为安卓平台的开放性，许多用户会在不知情的情况下就安装了恶意软件，一旦我们手机上安装了恶意软件，这些人员就可以盗取数据，这将会对我们的信息安全产生巨大的隐患。

为了能将下载后的软件，可以在下载完毕后直接能检测出是否为恶意软件，本文采取了一种静态检测的方法，通过对下载的 APK 文件进行反编译和解压缩等手段，从 AndroidManifest.xml 中提取出软件权限，每一个权限视作一种特征，通过独热编码转化后，再转化为特征图片放入卷积神经网络中进行分类，实验证明使用这种方式，相比传统的机器学习的识别准确率要高。

1 相关理论概述

1.1 独热编码概述

独热编码也叫做 one-hot 编码，具体使用独热编码方式是将非数字的特征进行一次量化处理，把非数字的特征转化为 01，10 等二进制编码。这样将离散特征的值直接扩展到欧式空间，这时离散特征的取值就对应欧式空间的点,这样让特征之间的距离计算更加的合理，由此使模型的可用度增强。

1.2 卷积神经网络概述

卷积神经网络是一种前反馈神经网络，由输入层，输出层，卷积层，激活函数，池化层，全连接层所组成，经典的神经网络 LeNet-5 结构如图 1 所示：

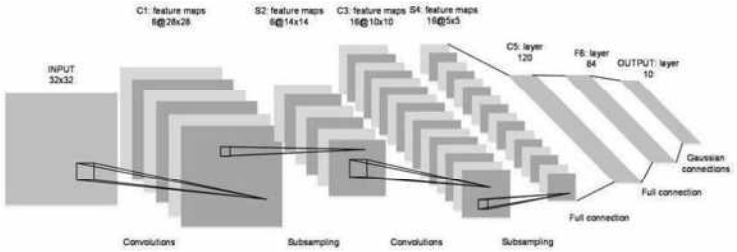

图 1：LeNet-5 结构图

其中，卷积层可通过卷积运算，提取特征图的特征信息，生成特征矩阵，由上图的 C1 为例，从输入层传入的 32*32*1 的图像首先进行卷积操作，处理成 28*28*6 的特征矩阵传入下一层。池化层可池化运算使特征矩阵减小，可有效的解决计算量过大的情况，由上图的 S2 为例，将上层传入的 28*28*6 的特征图经过池化操作，处理成一张 14*14*6 的特征图。将卷积 - 池化的运算结束以后，将多维特征矩阵转化为一维的特征向量，传入全连接层中，由上图 C5，F6 为例，最后交给分类器分类。

2 数据集的处理

2.1 数据来源

在数据的选择上，选择使用 CICDataset 上 MalDroid-2020 的公开数据集，有超过 17341 个 Android 示例，从中选择正样本 1150 个，负样本 1175 个，通过反编译和解压缩等手段从 APK 中提取出 AndroidManifest.xml 文件，并从该文件中取出该 APK 中的权限特征作为初始的数据集。

接着对每一个 APK 拥有的权限特征取并集，得到一共有 200 种不同的权限特征，再使用独热编码将每一个特征使用二进制向量进行转化，得到 1 * 400 的特征

向量，然后将这个特征向量转化为 20*20 的特征图，特征图中 1 个像素点的值就是软件对应 1 个权限的特征值[3]，最后对其特征图进行归一化后，得到训练、测试与验证集。正样本与负样本的特征图，如图 2 所示：

图 2：正样本（右）与负样本（左）的特征图

2.2 划分数据集

为了进行后面的四折检测验证，对总样本 2325 个数据进行划分，先将其中的 2094 个数据平均分为 4 份，每份数据集中的样本数约为 524，其中有 259 个正样本，265 个负样本，分别记为记为 A、B、C、D 数据集。最后把剩余的 231 个样本做测试集，其中 115 个正样本，116 个负样本记为 E 数据集。

3 模型训练

3.1 搭建卷积神经网络

由于安卓软件权限调用之间需要保持相互关联性的同时，需要考虑到图片大小的问题。经过多次实验后，在卷积神经网络中，只使用一层卷积 - 池化层，两个全连接层较为合理，由此所搭建的网络如图 3 所示：

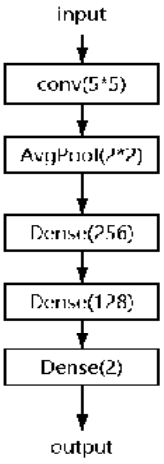

图 3： 网络结构

首先考虑将输入的图片大小，经过多次实验将图片调整至 32*32 的大小较为合理，同时为了保证权限调用之间的关联性，这里的输入图像会经过 5*5 的卷积运算，同时增加其维度，将原本的 1 个通道数增加到 6 个通道数。由于输入的原图像是一个二值图，考虑到使用最大池化操作可能会丢失一部分的特征，再次经过多次实验后，第二层采用 2*2 的平均池化操作，将卷积操作后得到的特征图片的大小减小至原来的一半。在其余的全连接层中，经过实验后选择了使运行速度快、输出结果更高的 256 与 128 个神经元的数量，最后其输出分类的结果。其他的参数经过多次实验后，得出合理的参数选择，由表 1 所示：

学习率	0.07	动量	0.9
权重衰减系数	0.005	分类器	Softmax
优化方式[6]	SGD	损失函数	Cross Entropy Loss
激活函数	Relu		

表 1：卷积神经网络中其他的参数

3.2 训练结果

训练时依次选择其中三份作为训练集，另外一份作为验证集，做四折交叉验证，结果由表 2 所示：

训练集和测试集	训练集损失值	训练集结果（准确率）	测试集结果（准确率）
A, B, C 为训练集，D 为验证集	0.2166	91.79%	88.89%
A, B, D 为训练集，C 为验证集	0.2012	92.80%	90.84%
A, C, D 为训练集，B 为验证集	0.2397	90.13%	90.46%
B, C, D 为训练集，A 为验证集	0.2536	90.76%	82.63%

表 2：四折交叉检验的结果

选取准确率最高的一组数据集进行训练与测试，所得出的训练集与测试集的损失图像如图 4 所示，由图可知大约在第 40 轮时，神经网络开始平滑下降，并收敛至最低。

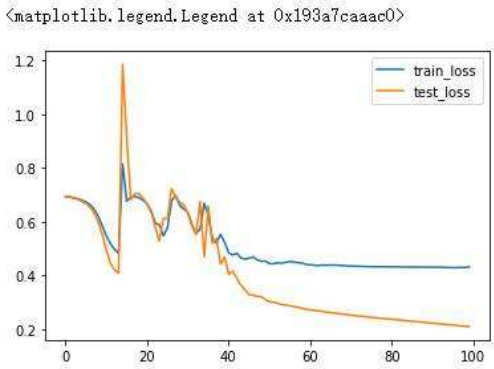

图 4 ： 训练集与验证集的损失图像

4 实验结果分析

4.1 模型的评价指标

传统评测一个模型的好坏使用的指标为真阳性率(TPR)、假阳性率(FPR)、准确率、Precision 和 F-Score。在这里记 TP 为预测出来是恶意软件的恶意软件的数量，TN 表示预测出来是良性软件的良性软件的数量，FN 表示预测出来是良性软件的恶意软件的数量，FP 表示预测出来是恶意软件的良性软件的数量[1]。这些指标的公式为：

$$TRP = \frac{TP}{TP+FN}, \quad (公式1)$$

$$FPR = \frac{FP}{FP+TN}, \quad (公式2)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP}, \quad (公式3)$$

$$Precision = \frac{TP}{TP+FP}, \quad (公式4)$$

$$F-Score = \frac{2*Precision*TPR}{Precision+TPR}, \quad (公式5)$$

4.2 对比不同的算法的实验结果与分析

使用传统的机器学习算法进行对比，结果由表3所示：

使用模型	TPR	FPR	Accuracy	Precision	F-Score
CNN	0.7328	0.0435	0.8442	0.9444	0.8252
MLP	0.7758	0.1217	0.8268	0.8653	0.8182
logistic 回归	0.7241	0.0957	0.8139	0.8842	0.7962

表3 不同算法的测试结果

由表3可以看出，在相同的测试集上，使用卷积神经网络可以在保证TPR、准确率（Accuracy）、Precision、F-Score的得分较高的同时，FPR也是最低的，因此在此数据集上，使用卷积神经网络明显的优于传统的机器学习的分类算法。

结论

在如今安卓系统的市场份额越来越大与互联网的应用方面越来越广的情况下，开发非法获取用户的信息和利益的恶意软件也越来越多．本文采用了一种静态分析的方法，将下载的APK文件进行反编译和解压缩等手段，从AndroidManifest.xml中提取出软件权限特征，再通过使用独热编码与改变特征向量转化为特征图的方式对特征进行预处理，并对经典的LeNet-5网络进行了一次特化改进。在此基础上实现了卷积神经网络的应用检测恶意软件的工具。

【参考文献】

[1]李靖平．一种基于权限的安卓恶意软件检测方法[J]．西北民族大学学报（自然科学版），2017,38(02):9-13.

[2]胡小春，朱成宇，陈燕．深度卷积神经网络模型的研究分析[J]．信息技术与信息化，2021(04):107-110.

[3]梁杰，陈嘉豪，张雪芹，周悦，林家骏．基于独热编码和卷积神经网络的异常检测[J]．清华大学学报（自然科学版），2019,59(07):523-529.

[4]傅依娴，芦天亮，马泽良．基于One-Hot的CNN恶意代码检测技术[J]．计算机应用与软件，2020,37(01):304-308,333.

[5]张雨薇，黄迎春．基于机器学习的恶意软件分类识别研究[J]．科技资讯，2018,16(30):9-11.

[6]张雪涛，孙蒙，王金双．基于操作码的安卓恶意代码多粒度快速检测方法[J]．网络与信息安全学报，2019,5(06):85-94.

[7]滕越，王天宝．安卓恶意软件检测模型研究[J]．科技风，2017(02):51,61.