基于LSTM递归神经网络的密码猜测

2017 IEEE计算科学与工程国际会议(CSE)和IEEE嵌入式与普适计算国际会议(EUC)

2017 IEEE计算科学与工程国际会议(CSE)和IEEE嵌入式与普适计算国际会议(EUC)

2017 IEEE计算科学与工程国际会议(CSE)和IEEE嵌入式与普适计算国际会议(EUC)

# 、灿哥、、、、连慧娟

上海交通大学信息安全工程学院

中国上海[qiuwd@sjtu.edu.cn](mailto:qiuwd@sjtu.edu.cn)

# 巩峥

中国广州华南师范大学计算机科学学院[cis.gong@gmail.com](mailto:cis.gong@gmail.com)

***摘要—密码经常用于数据加密和用户身份验证。由于人们倾向于选择有意义的单词或数字作为他们的密码，很多密码很容易被猜到。介绍了一种基于长短期记忆递归神经网络的密码猜测方法。在用来自泄露的Rockyou数据集的3000万个密码训练我们的LSTM神经网络之后，生成的33.5亿个密码可以覆盖剩余的Rockyou数据集的81.52%。与PCFG和马尔可夫方法相比，该方法具有更高的覆盖率。***

***关键词——密码猜测；递归神经网络；LSTM***

1. 介绍

密码猜测的目的是生成密码字典或数据集，以最小的规模覆盖尽可能多的用户密码。一方面，这些生成的密码可以检测用户密码的安全风险。另一方面，这些生成的密码可以促进字典攻击的性能。

近年来，密码生成方法的研究集中在:(1)使用更好的模板(如概率上下文无关文法(PCFG) [1，5，7，13，14])；(2)使用字符的相互关系(例如，马尔可夫链方法[2，8，10])。

与仅考虑几个相邻字符之间关系的基于马尔可夫的模型相比，递归神经网络(RNNs)能够揭示密码中所有字符之间的相互关系。因此，RNNs具有改进密码猜测的理论可行性。

我们的密码猜测模型是基于长短期记忆(LSTM) RNNs。我们用泄漏的密码训练我们的神经网络，并用搜索算法生成密码。本文的贡献在于实现了基于LSTM神经网络的密码猜测，并对不同层数/神经元数的密码猜测进行了比较。实验结果表明，与基于模板的方法和基于马尔可夫的方法相比，该方法在生成相同长度密码的情况下，通常能达到更高的测试密码覆盖率。

本文的结构如下。第二节简要概述了相关工作。第三节介绍了LSTM层和softmax回归。第四节给出了我们的实现。

第五部分给出了详细的实验结果和比较。我们的结论将在第六节中给出。

1. 相关作品

在Weir等人提出从训练密码生成模板并使用字典填充这些模板以进行密码猜测的PCFG[1]之后，Chou等人开发了TDT模型以生成更准确的模板和字典[7]；李等对中文网页密码进行了改进，增加了汉语拼音和规则[5]；Veras等人用语义和句法标签构建了PCFG[14]；Houshmand等人用键盘模式和多词模式优化了猜测效率[13]。

此外，Narayanan和Shmatikov首先在字典攻击中引入了Markov模型[10]。Castelluccia等人基于马尔可夫模型估计密码强度[2]。2015年，Dürmuth等人实现了基于马尔可夫模型的密码破解程序OMEN，通过生成100亿个密码，猜中了260万个Rockyou密码的80%以上[8]。

基于模板的模型和基于马尔可夫的模型已被应用于密码破解，如开膛手约翰

[11]和Hashcat [3]。这两个模型也被用于有针对性的在线密码猜测[12]。其他相关研究集中于这两种模型的组合[4]和评估[4，6]。根据[4]，基于马尔可夫的模型比基于模板的模型显示出更好的结果。

2016年，Melicher等人提出，神经网络可以模拟人类创建的密码，他们在密码强度测量中使用了神经网络[15]。他们用PGS和PGS+数据集训练数据，用MTurk和000webhost数据集测试。

1. 预赛
2. *LSTM递归神经网络*

递归神经网络能够从序列中学习。为了避免梯度消失问题，递归神经网络中的神经元可以由长短期记忆(LSTM)神经元代替[9]。

LSTM神经元被组织成层。LSTM层的外部输入是数据向量矩阵。矩阵的行长度代表序列的长度，等于时间步数。矩阵的列长度代表批次中序列的数量(批次大小)

786

978-1-5386-3221-5/17 31.00美元2017 IEEE DOI 10.1109/CSE-euc . 2017.155

788

978-1-5386-3221-5/17 31.00美元2017 IEEE DOI 10.1109/CSE-euc . 2017.155

785

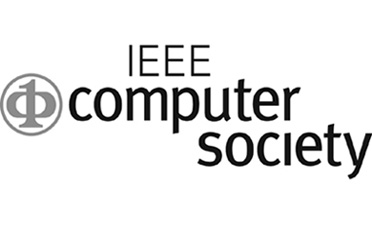
978-1-5386-3221-5/17 31.00美元2017 IEEE DOI 10.1109/CSE-euc . 2017.155



2017 IEEE计算科学与工程国际会议(CSE)和IEEE嵌入式与普适计算国际会议(EUC)



978-1-5386-3221-5/17 31.00美元2017 IEEE DOI 10.1109/CSE-euc . 2017.155

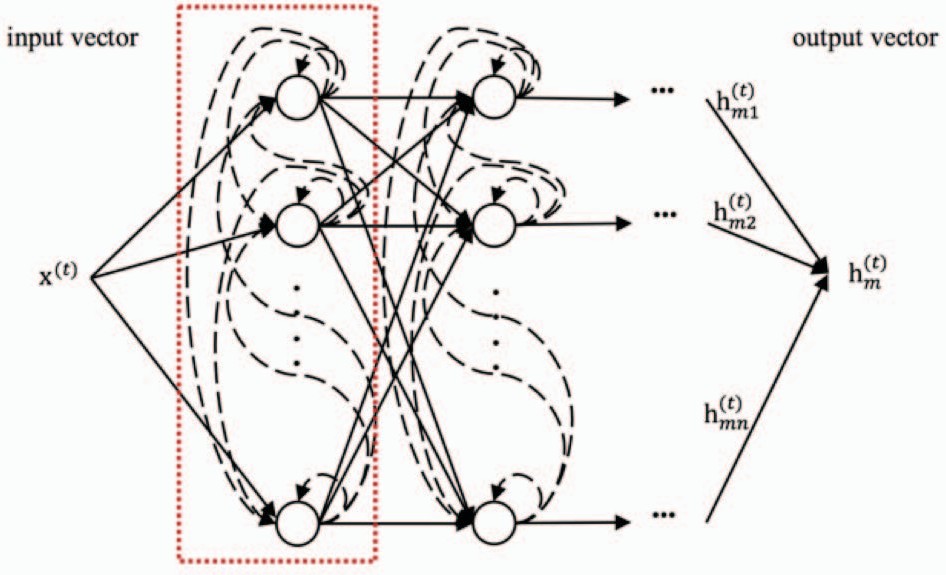
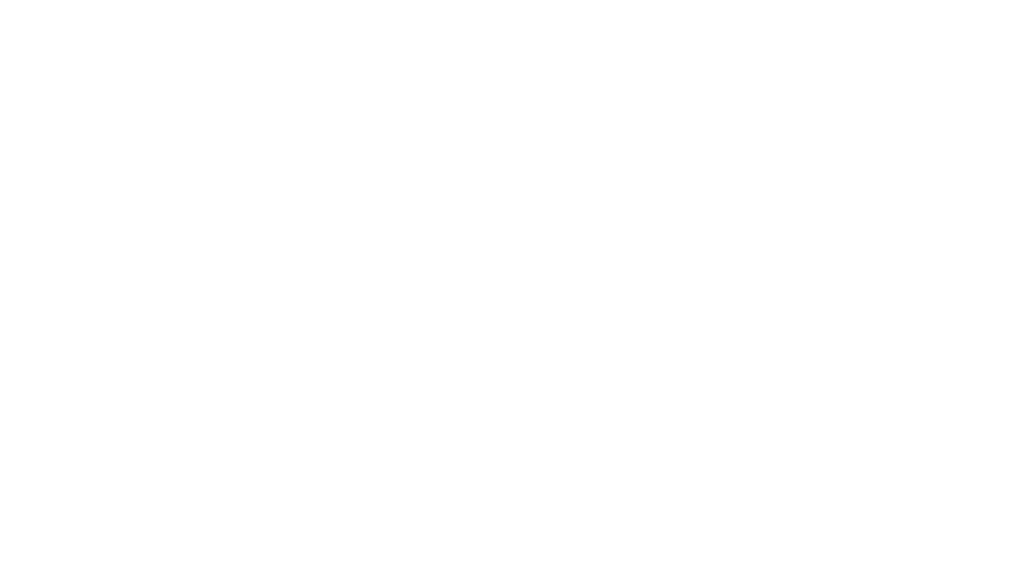
  



785

正在处理。在每个时间步长，LSTM层的外部输入是矩阵的列向量。

图1是n神经元m层LSTM结构。在每个时间步t，第I层的外部输入是第(i-1)层的输出向量(实线)。一层的输出向量将在下一时间步反馈给同一层的神经元(虚线)。

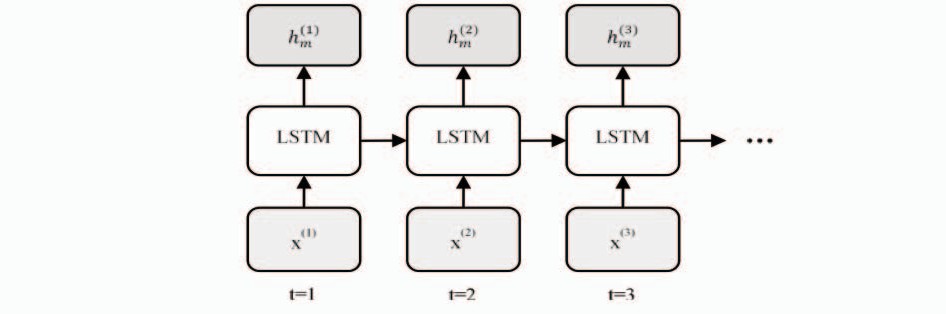


图一。LSTM层的结构。

图2是按时间步长扩展的m层LSTM的结构。在时间步长t，外部输入是矩阵的第t列向量。LSTM层接收x(t)以及最后的输出向量h (t-1)，h (t-1)，...，h (t-1)来计算电流输出向量h (t)。

*m*

1 2米



图二。LSTM层按时间步长的扩展结构。

1. *Softmax回归*

Softmax回归是为多分类问题设计的[16]。标签(类别)组成子集C，并且标签的数量是|C|。分类概率可以被估计并且假设被计算为等式(1)，x表示输入向量，y表示标签，θ表示softmax回归的参数。

𝛩𝑇𝑥

1. 履行
2. *密码猜测模型*

我们的神经网络包括一个输入层、几个LSTM层、一个softmax层和一个输出层。如图3所示，LSTM层数标记为m。δ代表所有LSTM层中的参数。θ表示softmax层中的参数。



x

投入

LSTM

层

h𝑚

h𝛩

softmax输出

图3。我们神经网络的结构。

我们的密码猜测模型的基本假设包括:(1)每个长度为len的密码“p1p2...plen”是一个序列x(1)=p1，x(2)=p2，...，x(len)= plen；(2)x(t)(2≤t≤len)的值与序列前缀x(1)，x(2)，

...，x(t-1)。

我们的密码猜测模型的基本思想包括:

(1)当使用x(1)，x(2)，...，x(t-1)作为序列输入；(2)我们的神经网络是由泄露的密码训练出来的；

(3)一旦在训练后给定前缀，下一个字符可以根据概率分布通过选择算法来决定；(4)密码中的每个字符(除了第一个字符)可以根据(3)顺序预测。

1. *密码预处理*

泄露的密码用来训练我们的神经网络。在我们的模型中，泄漏的密码分为三部分:训练集、验证集和测试集。训练集用于训练LSTM中的参数；验证集用于确保参数得到充分训练(尤其是防止过拟合)；测试集用于评估我们的模型的性能。

在使用密码进行训练之前，预处理包括:(1)数据清洗；(2)数据格式转换。

根据生成的要求，决定字符集和密码长度的范围。字符子集是C={c1，c2，c3...，ced}(ced=ed，ED代表密码结束)。数据清洗包括三个步骤:(1)删除含有不良字符的密码；(2)删除超过限制长度的密码，以及(3)打乱密码。

数据格式转换旨在转换监督神经网络的密码。数据格式转换包含两个步骤:(1)字符替换

(用标签L(ci) = i，ciC替换字符)；(2)

𝑝 𝑦 = 𝑐1 𝑥;𝜃

一

k j

𝜃 𝑥

𝑇

j=1

𝑒

𝑒 1

序列转换(转换“L(p )L(p)”...L(p)"

𝑝 𝑦 = 𝑐

𝑥;𝜃

𝛩𝑇𝑥

(1)

(2)

(伦)

1 2 len

ℎ𝛩

𝑥 =

⋮ 2 =

𝑒 2 ,

变成x =L(p1)，x =L(p2)，...，x =L(plen)，x(len+1)=L(ED))。

𝑝 𝑦 = 𝑐 𝐶 𝑥;𝜃

−𝜃𝑇 −

一

2

⋮

𝑇

𝛩 𝑥

𝑒 𝐶

1. *培训密码*

我们训练我们的模型来逼近密码中每个字符的真实概率分布。

例如，给定长度训练数据x(1)=L(p1)，

𝜃 =

−𝜃𝑇 −

 

x(2)=L(p2)，...，x

(伦)

=L(plen)，x

(len+1)

=L(ED)，被监督人

⋮

−𝜃𝑇 −

𝐶

学习标签是一个序列y(1)=x(2)，y(2)=x(3)，...，y(len)=x(len+1)。

在每个时间步长t(1≤t≤len ),提取LSTM层

序列输入x(1)，x(2)，...，x(t)，第m个LSTM层输出h (t)。softmax层接收h (t)和

α = 𝑛ℎ𝑖𝑡 ×100%.

𝑛𝑡𝑒𝑠𝑡

*m m* 动词 （verb的缩写）实验结果

计算概率分布。与监督标签y(t)相比，可以计算误差[16]。对于每个训练密码，softmax的成本函数回归如下。

𝐽 𝜃, ð1, ð2,… , ð𝑚

*A.RNN的关键参数*

我们的实验基于使用GTX970的Theano。为了找到更好的关键参数，我们设计了两个实验:1)比较不同LSTM层数的α曲线；2)比较每个LSTM层不同数量神经元的α曲线。最后，我们生成了一系列



𝑙𝑒𝑛 𝐶

𝑖=1 j=1

= −1/𝑙𝑒𝑛



1 𝑦 𝑖 = j对数

 

使用不同参数的不同大小的密码数据集。

1. *隐藏层数(HLN):在这一部分，三个实现不同数量的LSTM隐藏*

1{ }是指示函数，θ是softmax层的参数，δ是LSTM层的参数。|C|表示子集C中的字符数。

𝑒𝜃j ℎ𝑚

𝑇 𝑖

r=1

𝐶 𝑒𝜃𝑇ℎ 𝑖

𝑚

由于θ和δ代表我们模型的全部参数，成本函数计算误差，并且θ和δ将通过使用时间反向传播(BPTT)来调整。

*D.正在生成密码*

这种方法(算法1)避免了生成重复的密码。因为我们能够预测概率

层(HLN)的实施:一个，两个和三个隐藏层。每个LSTM层的神经元数量是256。实验在小规模数据集上进行，阈值分别为10-6，10-7，10-8，10-9。图4是一个对数图，x轴代表生成的对数，y轴代表α。随着阈值的降低，生成的密码数量增加。图4显示，HLN的增加带来了发电结果的轻微、不明显的增加(在相同阈值下仅约1%)。

字符的分布，每个密码前缀出现的可能性可以通过概率乘法公式来估计。

算法1。密码生成方法charset:字符子集C = {c1，c2，…，C | C | = ED } first \_ character \_ probabilities:第一个字符的统计结果

67%

57%

47%

37%

27%

1E-6

1E-7

1E-8

1E-9

一

2

3

*前缀:密码前缀数组*

*LUT:一个{关键字=前缀，值=概率}的查找表*

*阈值:最小概率*

//加载第一个字符的概率

//标签函数L(ci) = i，ciC

对于字符集中的c:

if(first \_ character \_ probabilities(L(c))> threshold):

*前缀. push(c)*

*LUT[c]= first \_ character \_ probabilities(L(c))endfor；*

//迭代所有可能密码的概率大于阈值

while(前缀不为空):

*current\_prefix = prefixes.pop()*

//通过NN next \_ char \_ prob = neural network(current \_ prefix)获取charset中c的下一个可能字符的概率:

if(LUT[当前前缀] \* next\_char\_prob(L(c))>阈值):if (c == ED):打印当前前缀

否则:

*prefixes.push(当前前缀+c)*

*LUT[当前前缀+c]= LUT[当前前缀]\*下一个字符概率(L(c))*

**endfor**

**endwhile**

*E.测试密码*

4.00年+04年

图4。不同数量的LSTM层。

1. *每层神经元数(NNPL):在这一部分中，实现了五个实现:每层神经元数(NNPL)是32，64，128，256，512。所有RNN实现包含两个LSTM隐藏层。实验在小规模数据集生成中进行，但足以识别更好的参数:10-6、10-7、10-8、10-9。图5也是一个对数图，x轴代表生成数的对数，y轴代表α。随着NNPL的增加，世代效应增加。表I .显示了NNPL的覆盖率增加，可以看出两点:(1)随着NNPL的增加，覆盖率α增加，尤其是NNPL较小时；(2)随着NNPL的增加，覆盖率的增幅逐渐减小。当NNPL达到256时，覆盖率没有显著提高。*

测试集的覆盖率α反映了密码猜测的准确性。公式如等式(3)所示，ntest是测试集的大小，nhit是测试集和生成的密码之间的交集的大小。在生成相同大小的时，α越大，生成效果越好

密码或α是相同的，但生成的密码较少。

46%

36%

26%

16%

1E-6

2.5E-7

1.56E-8

6.25E-8

512

256

128

64

32

3.20E+04 3.20E+05 3.20E+06

图5。每层都有不同的神经元。

随着HLN和NNPL的增加，相同阈值下覆盖率增加，但训练和生成速度降低。最后，在接下来的实验中，我们选择了一个具有2个隐藏LSTM层和每个LSTM 256个神经元的RNN。

表一.不同国家调查问卷之间的覆盖率增加

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1E-6** | **2.5E-7** | **6.25E-8** | **1.56E-8** |
| 32➺64 | 6.12% | 6.40% | 6.40% | 6.19% |
| 64➺128 | 4.18% | 4.53% | 4.49% | 4.16% |
| 128➺256 | 1.54% | 1.79% | 1.92% | 1.90% |
| 256➺512 | 0.45% | 0.64% | 0.67% | 0.74% |

*B.与先前研究的比较*

我们对一些大规模数据集进行了采样，并将结果与先前的研究进行了比较。我们的LSTM模型包含2个隐藏的LSTM层，每个LSTM层256个神经元。

LSTM模型由3000万个Rockyou (RY)密码训练，包括2700万个训练集和300万个验证集。除了Rockyou测试集(260万个密码)，Myspace数据集(MS)和脸书数据集(FB)也被用作测试集。

我们将我们的实现与OMEN-4Gram、概率上下文无关文法(PCFG)、约翰开膛手(JtR)马尔可夫模式和JtR增量模式进行了比较。

表二。不同算法的覆盖率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **算法** | **列车组** | **#猜猜** | **测试装置** | | |
| ***RY-***  ***260万*** | ***女士*** | ***运货单(freight bill)*** |
| 一 | 预兆-  4Gram | RY-  30米 | 1.0E+10 | 80.40% | 77.06% | 66.75% |
| 2 | PCFG [8] | RY- 30M | 1.0E+09 | 32.63% | 51.25% | 36.4% |
| 3 | JtR-  马尔科夫[8] | RY- 30M | 1.0E+10 | 64% | 53.19% | 61% |
| 四 | JtR公司[8] | RY- 30M | 1.0E+10 | 54% | 25.17% | 14.8% |
| 5 | LSTM | RY-  30米 | 3.4E+09 | 81.52% | 77.98% | 59.71% |
| 6 | LSTM | RY- 30M | 9.8E+08 | 73.77% | 69.30% | 50.35% |

表二。显示了不同算法的覆盖率。对比5与1、3、4，对于Rockyou测试集和MySpace数据集，我们在更少的猜测中击中了比Omen-4Gram、JtR马尔可夫模式和JtR增量模式更多的密码；但是脸书数据集的猜测效果似乎更差。在RY-e、Myspace和脸书数据集的测试中，我们的覆盖率远远高于PCFG。

不及物动词结论

LSTM递归神经网络可以应用于密码猜测。实验结果表明，我们的神经网络优于基于模板和基于马尔可夫的模型，因为我们生成的数据集在大多数测试集中命中更多的密码。

为了得到更好的结果，可以进一步调整神经网络的结构。未来的工作可能会集中在不同数据集之间的差异，研究共同或特定的神经网络。

承认

邱卫东受到教育部高校新世纪优秀人才基金NCET-12-0358和上海市科技领军人才项目16XD1424400的资助。

巩峥得到了国家自然科学基金项目61572028、广东省科学技术基金项目(2016B010125002)和广东省自然科学基金项目(2014A030313439)的资助。

参考

1. 米（meter的缩写））Weir，S. Aggarwal，B. d. Medeiros和B. Glodek，“使用概率上下文无关文法的密码破解”，第30届IEEE安全和隐私研讨会，2009年，第391-405页。
2. C.Castelluccia、M. Dürmuth和D. Perito，“来自马尔可夫模型的自适应密码强度计”，NDSS，2012年。
3. J.Steube，“hashcat -高级密码恢复”，[在线]。可用:[www.hashcat.net。](http://www.hashcat.net/)
4. J.马，杨文伟，罗，李，“概率密码模型研究”，2014年IEEE安全与隐私研讨会，2014年，第689-704页。
5. Z.李、韩文伟、徐文伟，“中国网络密码大规模实证分析”，第23届安全研讨会，2014年，第559-574页。
6. B.Ur、S. M. Segreti、L. Bauer、N. Christin、L. F. Cranor、S. Komanduri和R. Shay，“在密码可猜测性建模中测量真实世界的准确性和偏差”，2015年第24届USENIX安全研讨会，第463-481页。
7. H.Chou c、H. C. Lee、H. J. Yu、F. P. Lai、K. H. Huang和C. W. Hsueh，“基于从公开密码中学习到的模式的密码破解”，创新计算、信息和控制国际期刊，第9卷，2013年2月，第821-839页。
8. 米（meter的缩写））Dürmuth，F. Angelstorf，C. Castelluccia，D. Perito和A. Chaabane，“OMEN:使用有序马尔可夫枚举器进行更快的密码猜测”，载于工程安全软件和系统国际研讨会，2015年，第119-132页。
9. Z.C. Lipton，J. Berkowitz和C. Elkan，“对用于序列学习的递归神经网络的评论”。可用城市:arxiv.org/pdf/1506.00019.pdf.
10. A.Narayanan和V. Shmatikov，“使用时空折衷对密码的快速字典攻击”，载于第12届ACM计算机和通信安全会议论文集，2005年，第364- 372页。
11. 南设计师，“开膛手约翰密码破解”，[在线]。可用:[www.openwall.com/john.](http://www.openwall.com/john)
12. D.王，张，王平，严，黄，“有针对性的在线密码猜测:一个被低估的威胁”，载于2016年ACM SIGSAC计算机与通信安全会议论文集，2016年，第1242-1254页。
13. 南Houshmand，S. Aggarwal和R. Flood，“下一代PCFG密码破解”，IEEE信息取证和安全汇刊，第10卷，2015年8月，第1776-1791页。
14. R.Veras，C. Collins和J. Thorpe，“论密码的语义模式及其安全影响”，NDSS，2014年。
15. W.Melicher、B. Ur、S. M. Segreti、S. Komanduri、L. Bauer、N. Christin和L. F. Cranor，“快速、精简和准确:使用神经网络对密码猜测性建模”，载于《USENIX安全学报》，2016年。
16. 斯坦福UFLDL。《Softmax回归》【在线】。可用:[http://uf LDL . Stanford . edu/wiki/index . PHP/soft max \_ Regression。](http://ufldl.stanford.edu/wiki/index.php/Softmax_Regression)