СПИСОК ЛИТЕРАТУРЫ

1. Каршиев З. А., Холод И. И.. Анализ распределенных данных: обзор методов объединения моделей // Сб. докл. XIV Междунар. конф. по мягким вычислениям и измерениям SCM`2011, СПб., 23–25 июня, 2011. Т. 1. С. 163–166.

2. Интеллектуальный анализ распределенных данных на базе облачных вычислений / М. С. Куприянов, И. И. Холод, З. А. Каршиев и др. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2011.

3. Интеллектуальный анализ данных в распределенных системах / М. С. Куприянов, И. И. Холод, З. А. Каршиев, И. А. Голубев. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2012.

4. Холод И. И. Способы параллелизации алгоритмов интеллектуального анализа // Изв. СПбГЭТУ «ЛЭТИ». 2012. Вып. 9. С. 39–47.

5. Куприянов М. С., Каршиев З. А. Формальная модель выполнения алгоритмов интеллектуального анализа // Изв. СПбГЭТУ «ЛЭТИ». 2012. Вып. 9. С. 60–68.

6. Холод И. И. Унифицированная модель Data Mining // Сб. докл. XIV Междунар. конф. по мягким вычислениям и измерениям SCM`2012, СПб., 25–27 июня, 2012. Т. 1. С. 237–240.

I. I. Kholod, Z. A. Karshiyev

METHODS PARALLELIZING DATA MINING ALGORITHMS

This article describes how to parallel data mining algorithms based on the block structure. This technique takes into account the characteristics of the environment of parallel algorithms, such as the distribution of the data being analyzed, as well as features of the algorithm is parallelized.

Mining, parallel algorithms, distributed data mining

УДК 004.93.12

И. В. Сорокин, А. В. Копыльцов

ИСПОЛЬЗОВАНИЕ НЕОКОГНИТРОНА ДЛЯ РАСПОЗНАВАНИЯ ВРЕДОНОСНЫХ ФАЙЛОВ

Построен алгоритм использования неокогнитрона для сравнения бинарных данных. Рассмотрены этапы обучения и тестирования нейросети с целью выявления в анализируемых файлах характерных участков, позволяющие судить о принадлежности файла к определенному классу вредоносных программ.

Распознавание образов, нейронные сети, неокогнитрон, вредоносные файлы

Оценивание похожести бинарных файлов является одной из проблем в системах обработки и хранения данных. Например, в теории информационной безопасности большую роль играют алгоритмы, позволяющие сравнивать исполняемые файлы между собой. В системах поиска одной из проблем является выделение схожих документов. Альтернативным способом для решения подобных проблем является применение методов обработки изображений и распознавание образов.

Нейронные сети нашли широкое применение во многих областях, где требуется автоматическая обработка данных, изображений и т. п. Например, использование нейросетевого подхода в медицине позволяет сегментировать медицинские снимки на участки для выявления отклонений и постановки точного диагноза [1]. Анализ геофизических данных, основанный на нейросетевых методах, используется при выявлении сейсмической активности [2], поиске залежей нефти и газа [3]. Теория нейронных сетей находит применение и во многих других областях, таких, как машиностроение, космическая отрасль, экономика и социология.

Неокогнитрон [4], [5] как тип нейронной сети, рассматриваемый в данной статье, используется, в основном, для распознавания печатных и рукописных символов [6], [7]. При этом символы не ограничиваются только латинскими или русскими буквами. Неокогнитрон можно использовать для распознавания древнеегипетских иероглифов [8]. Особый интерес представляют работы по исследованию нейросетей типа неокогнитрон для распознавания более сложных образов, например человеческого лица [9].

Предлагаемый метод использования неокогнитрона заключается в обучении нейросети распознаванию характерных участков в файлах компьютерных программ. Каждый участок обладает своей уникальностью и может характеризовать как отдельное семейство вредоносных программ, так и некоторую группу. Например, заголовок исполняемого файла операционной системы Microsoft Windows обладает характерной структурой и может быть рассмотрен как уникальный участок. Тем не менее, его наличие характерно для многих вредоносных файлов. С другой стороны, уникальные последовательности данных или кода могут однозначно характеризовать вредоносную часть программы. Следовательно, по наличию или отсутствию таких участков в файле можно судить о его принадлежности к определенному семейству вредоносных программ.

Прежде чем приступить к непосредственному использованию неокогнитрона, необходимо выполнить предварительную обработку анализируемого файла. Ее основной задачей является представление исходного файла в виде последовательности цифровых изображений с фиксированной шириной и высотой. При этом яркость каждого пикселя в изображении будет соответствовать значению конкретного байта из анализируемого файла. Для формирования подобных изображений предлагается воспользоваться одним из двух подходов.

Первый подход заключается в использовании метода скользящего окна. В этом случае длина окна будет определять параметры ширины и высоты конечного изображения. Например, в таблице показаны первые байты произвольного исполняемого файла. По вертикали – позиция через каждые 16 байт, по горизонтали – порядковый номер байта в каждой позиции. Значения байт и их позиций записаны в шестнадцатеричной системе. При длине окна 256 байт размер изображения можно задать как 16 × 16 пикселей. Таким образом, изменяя смещение окна, можно сформировать изображения по всему файлу.

Позиция в файле	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00
0010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	F0	00	00	00
0040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68
0050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F
0060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20
0070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00
0080	35	FF	4C	32	71	9E	22	61	71	9E	22	61	71	9E	22	61
0090	6F	CC	B1	61	72	9E	22	61	6F	CC	A1	61	73	9E	22	61
00A0	6F	CC	A6	61	76	9E	22	61	56	58	59	61	73	9E	22	61
00B0	71	CC	B6	61	70	9E	22	61	6F	CC	A1	61	67	9E	22	61
00C0	6F	69	63	68	71	9E	22	61	6F	CC	B3	61	70	9E	22	61
00D0	52	69	63	68	71	9E	22	61	00	00	00	00	00	00	00	00
00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00F0	50	45	00	00	4C	01	07	00	11	C2	60	4F	00	00	00	00

В качестве альтернативного подхода предлагается использовать предварительную сегментацию файла. Например, на основании исследования [10] исходный файл может быть представлен в виде последовательности участков разной длины с разным уровнем энтропии. В этом случае изображения можно составлять как по всему участку, так и по фиксированной его части. Это позволит сократить количество итоговых изображений и, как следствие, уменьшить количество операций распознавания при анализе всего файла. У данного подхода существует еще одно приемущество: использование дополнительных эвристик для уменьшения общего количества изображений. Появляется возможность не учитывать сегменты как с очень низким значением энтропии, так и с очень высоким. Дальнейшее распознавание таких участков не представляет интереса, так как по своему содержанию они однородны и не обладают уникальными свойствами.

Следует отметить, что если представить исходный файл в виде набора изображений с фиксированной шириной и высотой, то появляется зависимость между байтами, расположенными через равные промежутки, соответствующие ширине итогового изображения. Это может привести к тому, что одна и та же последовательность байт, смещенная на величину, не кратную ширине, может формировать разные искаженные изображения. К тому же оба предложенных подхода могут формировать смещенные изображения на файлах, принадлежащих к одному и тому же семейству вредоносных программ, использующих различные механизмы защиты от обнаружения вредоносных частей программы. Именно поэтому для распознавания подобных изображений авторами рассмотренно использование неокогнитрона.

Неокогнитрон – это иерархическая многослойная нейронная сеть, предназначенная для инвариантного распознавания образов. В дальнейшем планируется использовать архитектуру сети, описанную в [5]. Схематическое изображение слоев используемой архитектуры неокогнитрона представлено на рис. 1.

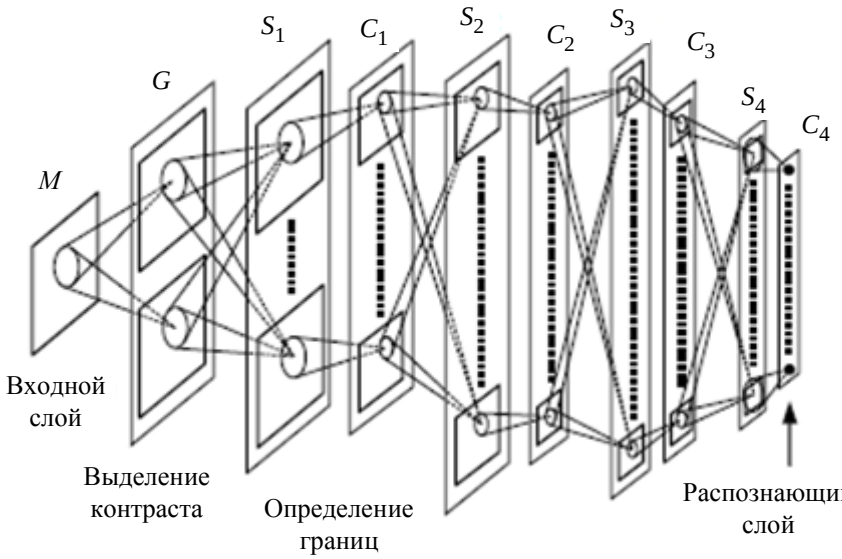

Рис. 1

На вход сети подается изображение в виде матрицы пикселей *M*. Далее следует слой *G*, выделяющий контраст [5]. Затем идут 4 слоя, состоящих из двух типов плоскостей: простые – *S* и сложные – *C*. Простые плоскости реагируют на появление характерных признаков в предыдущем слое. Например, плоскости S_1 в первом слое выделяют элементарные границы в исходном изображении, такие, как линии и углы. Предназначение сложных плоскостей – обобщение результатов предыдущих простых плоскостей. Тем самым достигается инвариантность распознавания к сдвигам, поворотам и другим искажениям входного образа. В результате последовательного прохода информации по всей обученной сети в конечном слое будет активизироваться только одна сложная плоскость, соответствующая определенному классу образов.

Для проведения эксперимента было сформировано 5 классов, каждый из которых характеризует определенные участки вредоносных программ семейства Trojan.Mayachok по классификации Dr.Web. На рис. 2 изображены 5 участков отдельно взятого файла. Участки выбирались в соответствии с форматом Portable Executable исполняемых файлов для операционной системы Microsoft Windows: *1* – заголовок, *2* – программный код, *3* – упакованные данные, *4* – секция импорта, *5* – секция реллоков. Каждое изображение сформировано на этапе предварительной обработки данных и представляет собой матрицу пикселей, интенсивность которых определяется значением соответствующего байта из файла. Чем больше значение байта, тем сильнее оттенок черного цвета соответствующего пикселя. Например, на третьем участке наблюдается однородность на всей площади изображения. Это объясняется равномерным распределением байт в участке упакованных данных. Напротив, на первом изображении заметно различие между областью заголовка и следующей за ней областью нулевых байт. Чтобы не вносить дополнительные изменения в обучающую и тестовую базы образов, в качестве ширины и высоты обрабатываемых изображений были выбраны такие же значения, как и в [5], размером 65 × 65 пикселей. При этом, учитывая структуру баз, одно изображение соответствовало 2016 байт с размером окна 63 × 32 байт.

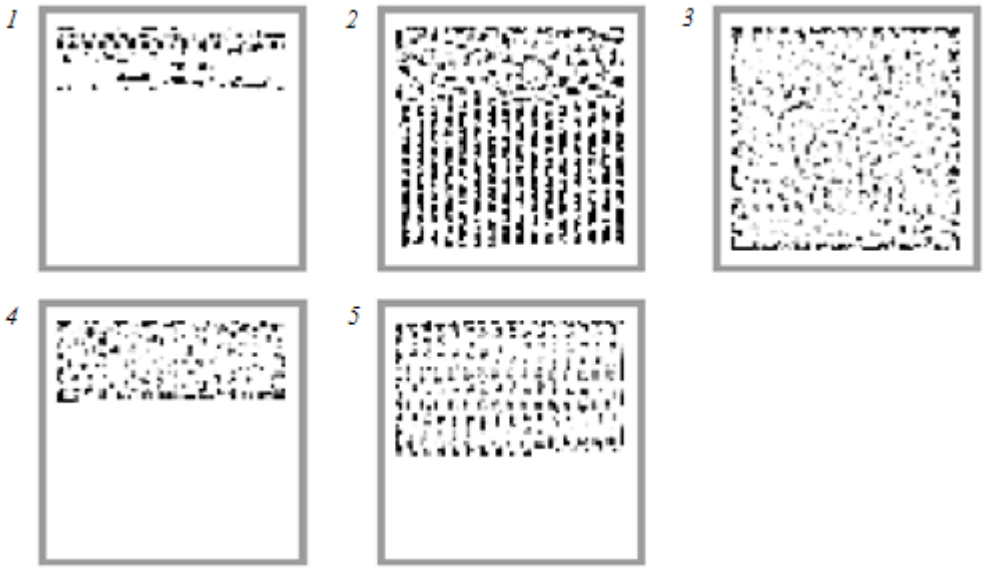

Рис. 2

Определившись с количеством классов, типом и размером участков, которые характеризуют выбранное семейство вредоносных программ, можно приступить к обучению неокогнитрона. Для обучения неокогнитрона был использован алгоритм, предложенный в [5]. Описание и реализация алгоритма обучения также доступны на сайте [5]. Настройка синаптических связей происходит слой за слоем, начиная с простых плоскостей первого слоя. При этом для каждого слоя авторы предлагают использовать разные методы обучения. Так, для простых плоскостей S_1 первого слоя может быть использовано обучение как с учителем, так и без него [4]. Для плоскостей S_2 и S_3 применяется обучение, основанное на принципе «победитель получает все», используемое в сетях, работающих на основе самоорганизации. В последнюю очередь обучаются плоскости S_4 из четвертого слоя. Здесь, по аналогии с двумя предыдущими слоями, используется конкурирующее обучение, но с дополнительным ограничением по количеству итоговых классов распознавания. Это связано с тем, что в последнем слое для одного и того же класса могут активизироваться несколько простых плоскостей S_4 . В том случае, когда для представленного обучающего примера победителем оказывается плоскость, ассоциированная с таким же классом, веса нейронов усиливаются. Если же победитель определяет неправильный класс, то добавляется еще одна простая плоскость S_4, ассоциированная с нужным классом. В итоге, в процессе распознавания окончательный результат будет зависеть от максимально активной плоскости S_4 в последнем слое.

Размер обучающей выборки составил 400 уникальных экземпляров. При этом на каждый класс приходилось по 80 экземпляров. В качестве настроечных параметров процесса обучения были заданы следующие значения: итоговое количество классов распознавания на четвертом слое (K_C), равное 5, начальное (K_S) и максимальное ($K_{S_{\max}}$) количество плоскостей на четырех основных слоях: 4, 8, 16, 5. Значения остальных параметров обучения были оставлены по умолчанию [5]. В итоге, общее время обучения составило 1 мин 40 с на компьютере с CPU 3 ГГц и оперативной памятью 4 Гбайт.

Аналогично процессу обучения была составлена выборка из 400 экземпляров для проведения тестовых сравнений. Учитывая все параметры, заданные при процессе обучения, сеть безошибочно смогла распознать все представленные тестовые образы. Время тестирования составило чуть больше 10 с. Следует отметить, что общее количество итоговых классов очень мало и, как следствие, размер сети незначителен. В реальных условиях с увеличением множества всевозможных семейств вредоносных программ количество классов будет расти, но, с учетом настроечных параметров используемого алгоритма [9], можно добиться удовлетворительных результатов.

Помимо использования исходных кодов [5] были изучены процедурные расширения для среды MatLab. Они позволяют наглядно представить весь путь активации нейронов в каждом отдельном слое неокогнитрона. Как и тестовая программа, они используют базу весов нейронных связей, полученную на предыдущем этапе обучения. На рис. 3 наглядно продемонстрирована активация нейронов в простых плоскостях основных слоев неокогнитрона.

В качестве примера взят образ из первого класса (рис. 2), ассоциирующийся с заголовком исполняемого файла. Из рис. 3 видно как происходит последовательное выделение и обобщение характеристик исходного образа. Первые две плоскости относятся к слою *G*, выделяющему контраст у представленного изображения. На них отчетливо видна разница между областью заголовка исполняемого файла и областью нулевых байт. Следующие 4 последовательности изображений относятся к простым *S*-плоскостям основных слоев неокогнитрона. В этих плоскостях, используя вычисленные коэффициенты связей между нейронами, происходит активация областей, обученных на реагирование определенным характеристикам. Например, в первых двух слоях заметно выявление локальных областей, соответствующих небольшим последовательностям байт. В следующем третьем слое размерность плоскостей гораздо меньше, чем у предыдущих. За счет этого происходит обобщение и выделение областей, характеризующих почти все байты заголовка исполняемого файла. В результате в последнем слое *S* наиболее активной становится единственная плоскость, соответствующая одному из пяти возможных классов.
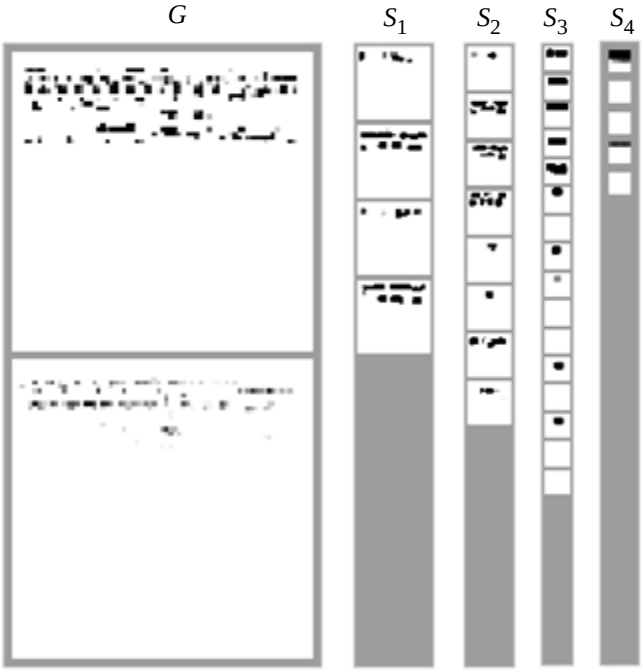
Рис. 3

Таким образом, описано и продемонстрировано использование универсальной сети «неокогнитрон» для распознавания характерных участков бинарных данных. Одной из возможных областей применения такого подхода может быть информационная безопасность. Предложенный подход имеет ряд ограничений и не претендует на эффективный механизм распознавания всех возможных вредоносных файлов. Тем не менее, он демонстрирует возможную альтернативу устоявшимся сигнатурно-байтовым способам обнаружения угроз.

СПИСОК ЛИТЕРАТУРЫ

1. Jabarouti-Moghaddam M., Soltanian-Zadeh H. Medical Image Segmentation Using Artificial Neural Networks. // Artificial Neural Networks - Methodological Advances and Biomedical Applications, In Tech, Rijeka, Croatia, 2011. P. 121–138.

2. Кислов К. В., Гравиров В .В. Нейросетевой метод детектирования землетрясения при повышенном уровне шума // Электронный журнал "Исследовано в России". 2010. № 070. С. 824–836. http://zhurnal.ape.relarn.ru/articles/2010/070.pdf.

3. Яневиц Р. Б., Беспечная Л. Ю., Беспечный В. Н. Построение сейсмологических моделей осадочных отложений на основании данных сейсморазведки и ПГИС // Геология, геофизика и разработка нефтяных и газовых месторождений. 2006. № 10. С. 105–114.

4. Fukushima K. Neocognitron: a hierarchical neural network capable of visual pattern recognition // Neural Networks. 1988. 1 (2). P. 119–130.

5. Fukushima K. Neocognitron for handwritten digit recognition. // Neurocomputing. 2003. Vol. 51. P. 161–180. http://visiome.neuroinf.jp/modules/xoonips/detail.php?item_id=375.

6. Садыхов Р. Х., Ваткин М. Е. Алгоритм обучения нейронной сети «неокогнитрон» для распознавания рукописных символов // Изв. НАН Беларуси. Сер. физ.-техн. наук. 2002. № 3. С. 81–86.

7. Федяев О. И., Махно Ю. С. Программный эмулятор нейросети типа неокогнитрон для распознавания графических образов // Научные работы Донецкого национального технического университета. Сер. «Информатика, кибернетика и вычислительная техника». 2008. № 9 (132). С. 265–269.

8. Ивашко А. Г., Кугаевских А. В. Возможность применения ИНС «Неокогнитрон» для распознавания древнеегипетских иероглифов // Вестник Тюменского государственного университета. 2009. № 6. С. 209–214.

9. Duffner S. Face Image Analysis With Convolutional Neural Networks. // PhD Thesis. Albert-Ludwigs-Universität Freiburg im Breisgau, Germany, 2007.

10. Копыльцов А. В., Сорокин И. В. Вейвлет-анализ структурной энтропии файлов // Изв. РГПУ им. А. И. Герцена. 2011. № 138. С. 7–15.

I. V. Sorokin, A. V. Kopyltsov

NEOCOGNITRON FOR MALWARE RECOGNITION

The algorithm for using neocognitron for compare binary data is developed. To identify specific areas of the analyzed files, in this paper we describe the stages of training and testing neural network. It can help to detect malicious programs.

Pattern recognition, neural networks, neocognitron, malware

УДК 004.421.6

А. Р. Лисс, И. А. Андрианов

ИСПОЛЬЗОВАНИЕ РАЗРЕЖЕННЫХ СУФФИКСНЫХ ДЕРЕВЬЕВ ДЛЯ ЗАДАЧ ОБРАБОТКИ ТЕКСТОВ

Потребность в эффективных алгоритмах обработки строк возникает во многих практических задачах. Одной из классических задач в этой области является задача поиска наибольшей общей подстроки. Для ее решения предлагается использовать разреженные суффиксные деревья, приводится описание алгоритма. Такой подход позволит обрабатывать данные существенно большего размера, чем в случае использования обычных суффиксных деревьев. Также в статье рассматривается решение задачи поиска максимальной симметричной подстроки.

Разреженные суффиксные деревья, поиск по сходству, алгоритм Укконена

В настоящее время наблюдается значительный интерес к алгоритмам обработки строк и последовательностей. Это объясняется постоянно растущей потребностью в эффективном решении такого рода задач в связи с развитием областей их применения – распознавание образов, вычислительная молекулярная биология, информационно-поисковые системы и др. В большинстве строковых алгоритмов используется общий подход, который заключается в проведении предварительной обработки (предобработки) одной или нескольких исходных строк – т. е. построении индекса.