

Прогнозирование аварий сетевого оборудования с использованием нейронных сетей

Д.С. Тайманов

Самарский национальный исследовательский университет имени академика С.П. Королёва, 443086, Московское шоссе, 34, Самара, Россия

Аннотация

Жизнь современного человека невозможно представить без доступа к средствам коммуникации. С ростом зависимости каждой сферы деятельности от надежных каналов связи растут и убытки, наносимые авариями сетевого оборудования. Можно сократить ущерб от аварии, минимизировав ее масштабы и уменьшив затраты на ремонт оборудования, или вовсе избежать этих проблем, если удастся заранее выявить отклонения в работе сети, приводящие к сбоям, и принять меры по их ликвидации. В данной работе будет исследовано использование нейронных сетей для решения описанной проблемы. Такое решение обладает способностью к выявлению закономерностей и хорошей робастностью, что особенно важно в условиях работы с разобщенными и частично отсутствующими данными.

Ключевые слова: интеллектуальный анализ данных; нейронные сети; прогнозирование; аварии сетевого оборудования; Big Data;

1. Введение

Из-за сбоев в работе сетевого оборудования провайдеры телекоммуникационных услуг несут ощутимые потери прибыли, а их клиенты терпят неудобства. В условиях роста влияния сетевых технологий на все сферы деятельности растет и урон, наносимый сбоями в работе связи как количественно, так и качественно.

С усложнением оборудования, развитием концепции Big Data и инструментов для работы с большим количеством разрозненной информации, поступающей с различных датчиков и систем, появляется возможность использовать эти данные для прогнозирования сбоев оборудования телекоммуникационной сети. Это позволит предотвращать аварии или хотя-бы минимизировать ущерб, предупреждая об инциденте и предоставляя информацию для эффективного решения проблемы. Таким образом, возникает задача прогнозирования аварий сетевого оборудования.

В настоящий момент сложно найти исследования в этой области. Среди областей с похожими проблемами можно выделить прогнозирование аварий механического оборудования [1] и прогнозирование ошибок в исходном коде программ [2]. В таких задачах входные данные имеют небольшую размерность и единый тип, в то время как в задаче прогнозирования аварий сетевого оборудования данные поступают с множества различных источников, разобщены и могут частично отсутствовать. Помимо этого, большая часть значений принадлежит перечислимому типу, а не числовому, как в [1] и [2]. Похожие данные встречаются в задаче прогнозирования заболевания костей. Авторы исследования [3] использовали глубокие сети доверия для выделения значимых факторов риска и предсказания развития болезни.

Применение нейронных сетей является интересным вариантом решения поставленной задачи. В качестве примера для построения моделей на начальных этапах можно использовать данные конкурса [4] по прогнозированию аварий сети компании Telstra. Среди различных решений, опробованных участниками соревнования, есть и несколько вариантов нейронных сетей, например, глубокие сети доверия. Однако такие подходы не пользовались особой популярностью. В данной работе будут исследованы различные архитектуры нейронных сетей в рамках задачи прогнозирования аварий сетевого оборудования.

2. Данные о состоянии сетевого оборудования

Производители современного сетевого оборудования встраивают в свои устройства все необходимые датчики для слежения за их состоянием. Помимо, а так же в комплексе с этими аппаратными средствами широко используются программные средства мониторинга здоровья систем. Таким образом, доступна информация о географическом расположении и типе оборудования, показаниях датчиков и допустимых пределах для них, режимах функционирования, показателях эффективности и нагрузки, различных событиях и т.д.

В качестве примера можно взять набор данных «Telstra Network Disruptions» [4]. Они представляют собой записи о событиях, каждое из которых характеризуется местонахождением оборудования *location*, к которому относится событие, типом события *event_type*, типом источника события *resource_type*, типом сообщения в журнале *severity_type*, различными признаками записей и степенью их выраженности *log_feature*, а так же серьезностью аварии *fault_severity* – именно это значение необходимо прогнозировать (подробнее – таблица 1).

Как можно увидеть, большинство значений представляют собой перечисления. То есть обозначают принадлежность конкретной записи какому-либо классу по какому-либо параметру, например «location 1», «event_type 15», «resource_type 8», «severity_type 2». Признаки *log_feature* характеризуются не только классом присущего событию признака, но и тем, насколько этот признак выражен. Таким образом, атрибут *log_feature* представляет собой кортеж «тип значения – уровень значения».

Таблица 1. Структура набора данных «Telstra Network Disruptions»

Название атрибута	Количество различных значений, N_i	Тип	Повторяемость, значений/событие	Описание
id	7381	целочисленный, неупорядоченный	-	Идентификатор для соединения данных. Уникален для каждого события.
<i>fault_severity</i>	3	целочисленный	-	Серьезность события - цель прогнозирования
location	929	перечисление	-	Местоположение оборудования
event_type	49	перечисление	12468/7381	Тип события.
resource_type	10	перечисление	8460/7381	Тип ресурса
severity_type	5	перечисление	-	Тип сообщения в журнале
log_feature	331	кортеж: <i>тип</i> – уровень <i>тип</i> : перечисление <i>уровень</i> : целочисленный	23851/7381	Признаки, извлеченные из сообщений в журнале

Атрибуты выборки распределены по различным таблицам, хранящимся в CSV-файлах. Это связано с тем, что некоторые атрибуты могут принимать сразу несколько значений для одного события. Например, в записи журнала могут присутствовать несколько признаков *log_feature*. Для связи между таблицами служит атрибут *id*. Структура таблиц представлена на рисунке 1 а.

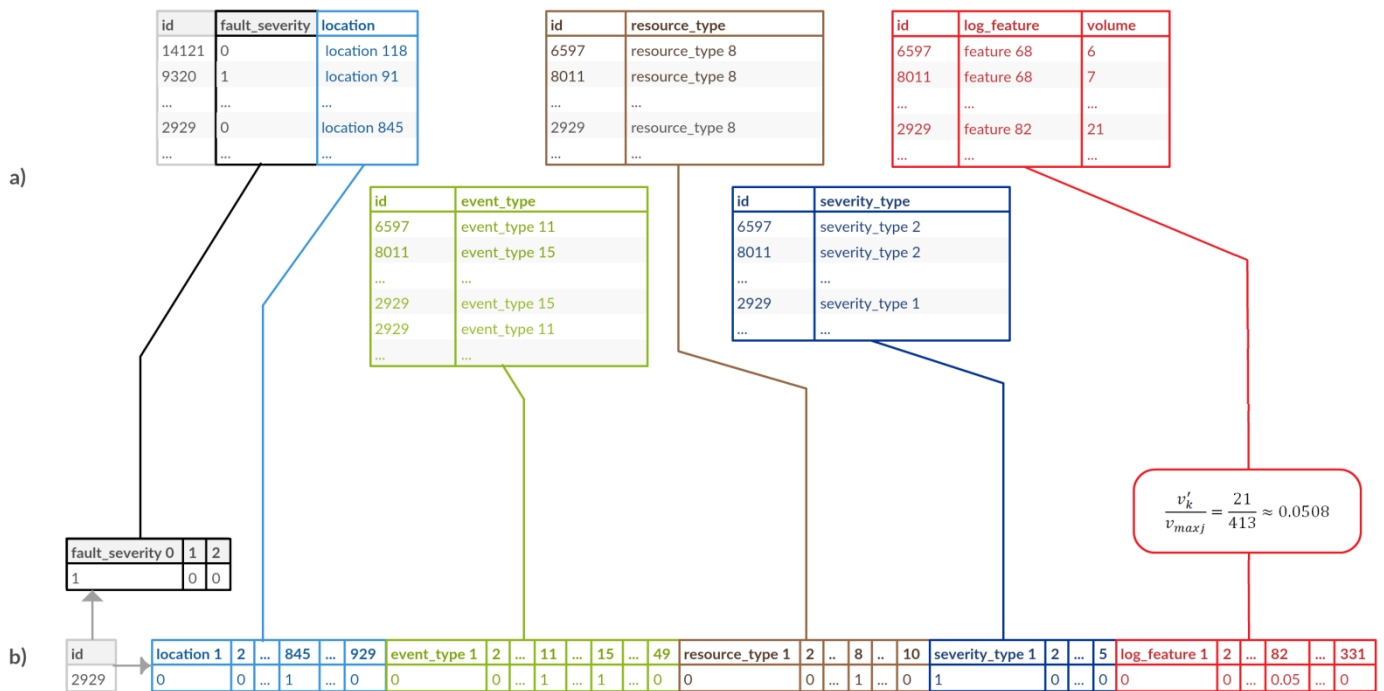


Рис. 1. Структура набора данных (а) и представление события для обучения (б).

Представим каждый атрибут в виде вектора, длина которого соответствует количеству различных значений атрибута, а значения для текущего события отмечаются единицами на соответствующих позициях (рисунок 1 б). Это наиболее простой вариант задания вектора для перечислимых неупорядоченных типов, принимающих по несколько значений на одно событие.

Таким образом, значения атрибутов *location*, *event_type*, *resource_type* и *severity_type* примут вид:

$$v_i = (x_1, \dots, x_{N_i}),$$

где N_i – количество значений i -го атрибута, $i = \overline{1, M}$, M – количество атрибутов типа «перечисление»; x_j подчиняется формуле:

$$x_j = \begin{cases} 1, & \text{если } X_{ij} \in f_i, \\ 0, & \text{иначе} \end{cases}$$

где f_i - значения i -го атрибута для текущего события, представляет собой множество всех значений, которые имеют место для события; X_{ij} - j -ое значение i -го атрибута; $\{X_{ij}\}_{i=1, \dots, M, j=1, \dots, N_i}$ - множество всех значений.

Точно так же формируется и вектор для прогнозируемого значения *fault_severity*.

Для кортежей (атрибут *log_feature*) последняя формула примет иной вид:

$$x_j = \begin{cases} v'_k, & \text{если } Y_j = f_k, \\ 0, & \text{иначе} \end{cases}$$

где v'_k - уровень k -го присутствующего признака, f_k - k -й присутствующий признак («тип» из кортежа), Y_j - j -й признак из множества всех возможных признаков, $j = 1, \dots, N_Y$, N_Y - количество возможных признаков *log_feature*.

Поскольку уровень для каждого из признаков изменяется в различных пределах, может иметь смысл попробовать так же и другой вариант формирования значений атрибута *log_feature*, чтобы «уравнять» диапазоны значений различных типов признаков в векторе:

$$x_j = \begin{cases} \frac{v'_k}{v_{maxj}}, & \text{если } Y_j = f_k, \\ 0, & \text{иначе} \end{cases}$$

где v_{maxj} - максимальное значение признака j -го типа. Поскольку все значения уровней признаков *log_feature* - натуральные числа в диапазоне от нуля до некоторого значения, нет необходимости сдвигать значение, а достаточно лишь масштабирования.

Таким образом, входной вектор значений V , соответствующий событию, будет представлять собой последовательность векторов v_i , нормирован (если выбран второй вариант представления значений *log_feature*) и иметь длину:

$$N = \sum_{i=1}^M N_i + N_Y.$$

В некоторых случаях нейронная сеть может состоять из нескольких подсетей, независимо обрабатывающих различные участки входного вектора. Выходные значения таких подсетей подаются на вход общей сети, на выходе которой образуется конечный результат. Для таких случаев необходимо сохранять информацию о сегментации результирующего вектора V на векторы v_i .

Для набора данных «Telstra Network Disruptions» длина составит 1324, что является достаточно большим значением. Поэтому в качестве общего подхода выбрано использование глубоких нейронных сетей.

3. Глубокие нейронные сети

Среди глубоких нейронных сетей, главным образом, следует выделить сети доверия [5], поскольку они позволяют выделять скрытые особенности для данных большой размерности и могут быть эффективно обучены и тонко настроены. Глубокие сети доверия представляют собой композицию ограниченных машин Больцмана (RBM – Restricted Boltzmann Machine) [6] или автокодировщиков. Каждый такой слой-модуль располагается последовательно (рис. 2).

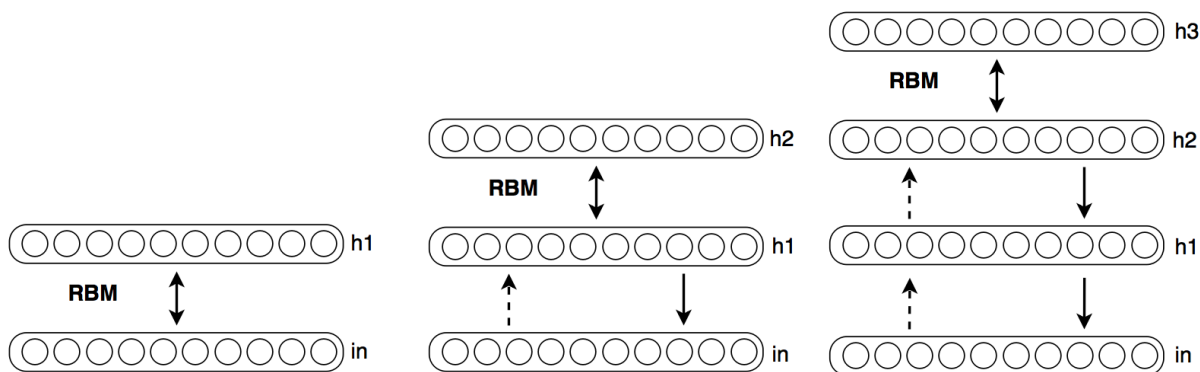


Рис. 2. Построение глубокой сети доверия.

Машина Больцмана (рис. 3 а) представляет собой стохастическую машину, компонентами которой являются стохастические нейроны [7]. Такая сеть полносвязна, а матрица весов W , которыми характеризуются связи между нейронами, симметрична и имеет нули на главной диагонали. Все нейроны делятся на видимые и скрытые. При обучении видимые нейроны фиксируются в своих состояниях, определяемых средой (значениями векторов из

обучающей выборки), а скрытые работают свободно, при этом состояние такой рекуррентной сети со временем стабилизируется при выполнении ограничений, наложенных на матрицу весов W . Скрытые нейроны при этом «извлекают статистические корреляции высокого порядка в ограничивающих векторах»[7].

В описанном случае происходит обучение без учителя и в режиме работы, в случае подачи на вход неполного вектора, сеть может осуществлять дополнение образов. Если же разделить видимые нейроны на входные и выходные и одновременно подавать обучающий вектор и требуемое выходное значение, можно осуществить обучение с учителем. Подробнее машины Больцмана описываются в источниках [6] и [7].

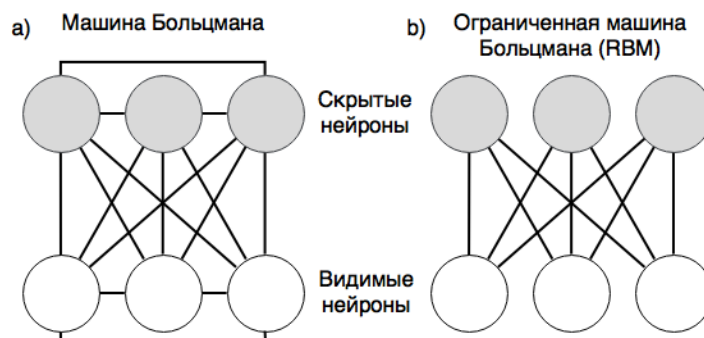


Рис. 3. Машина Больцмана (а) и ограниченная машина Больцмана (b).

Недостатком описанной модели является длительность достижения равновесного состояния для сетей больших размеров. Таким недостатком не обладают ограниченные машины Больцмана (рис. 3 б), что достигается удалением связей между однотипными нейронами и исключением, таким образом, многократной активации каждого из них в процессе установления равновесия. Такие сети могут быть хорошо обучены на практике [6].

Возвращаясь к глубоким сетям доверия (рис. 2), которые строятся из описанных выше ограниченных машин Больцмана, опишем алгоритм [8] формирования таких сетей:

- 1) Первые два слоя (in , $h1$) представляются как ограниченная машина Больцмана и обучаются на тренировочной выборке.
- 2) Через обученную машину Больцмана пропускаются входные данные, и берутся данные со скрытых нейронов слоя $h1$.
- 3) На полученных данных обучается следующая пара слоев ($h1$, $h2$), затем так же обучаются слои $h2$, $h3$, и так далее, пока не будут обучены последние два слоя.
- 4) Осуществляется тонкая настройка сети с использованием обучения с учителем или без учителя.

За счет такого построения сети и достигается эффективность обучения и точность результата.

При исследовании формируется набор возможных вариаций архитектур сетей:

- различные типы сетей (на основе ограниченных машин Больцмана, на основе автокодировщиков, другие глубокие сети);
- различное количество слоев сети;
- различные градации числа нейронов на каждом слое;
- выделение подсетей, осуществляющих предобработку различных сегментов вектора с последующим объединением в общую сеть. Подключение подсетей к общей сети так же может происходить на различных слоях.

В процессе исследования возможны модификации описанных вариантов нейронных сетей. Обучение всего многообразия топологий сетей будет производиться параллельно на одних и тех же данных.

Учитывая обилие различных вариаций, можно исследовать возможности применения генетических алгоритмов.

Для обучения на наборе «Telstra Network Disruptions» сети будут иметь 1324 входа на первом слое и 3 выходных нейрона. Входной вектор сегментируется на 5 частей, относящихся к различным атрибутам для обучения составных сетей.

Набор данных разбивается на обучающую и тестовую выборку тремя различными способами:

- случайным образом;
- сохраняя целостность наборов, относящихся к разным значениям атрибута *location*;
- включая в тренировочную/тестовую выборку значения из каждого *location* пропорционально выбранному соотношению.

4. Заключение

Одним из интересных подходов к решению задачи прогнозирования аварий сетевого оборудования является использование глубоких нейронных сетей, что оправдано размерностью данных о состоянии сетевого оборудования. Среди глубоких нейронных сетей следует обратить внимание на глубокие сети доверия.

Будет проведено сравнение различных вариантов представления данных и топологий сетей. Учитывая многообразие различных комбинаций параметров сетей и представлений данных, необходимо автоматизировать процесс поиска

наилучшего варианта. А именно, процессы выбора топологии, представления данных, обучения и оценки результатов работы сети. В этом может помочь применение генетических алгоритмов.

Литература

- [1] Dabrowski, D. Condition monitoring of planetary gearbox by hardware implementation of artificial neural networks / D. Dabrowski // *Measurement*. – 2016. – vol. 91. – P. 295–308.
- [2] Erturk, E. Iterative software fault prediction with a hybrid approach / E. Erturk, E. A. Sezer // *Appl. Soft Comput.* – 2016. – vol. 49. – P. 1020-1033.
- [3] Li, H. Identifying informative risk factors and predicting bone disease progression via deep belief networks / H. Li, X. Y. Li, M. Ramanathan // *Methods*. – 2014. – vol. 69(3). – P. 257–265.
- [4] Соревнование «Telstra Network Disruptions» [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/c/telstra-recruiting-network> (01.02.2017)
- [5] Hinton, G. E. Deep belief networks [Электронный ресурс]. – Режим доступа: http://www.scholarpedia.org/article/Deep_belief_networks (01.02.2017)
- [6] Hinton G E. Boltzmann_machine [Электронный ресурс]. – Режим доступа: http://www.scholarpedia.org/article/Boltzmann_machine (01.02.2017)
- [7] Хайкин, С. Машина Больцмана / С. Хайкин // *Нейронные сети: Полный курс* / С. Хайкин – 2-е изд., испр. – М., 2006. – Гл. 11. – С. 713-721.
- [8] Hinton, G. E A fast learning algorithm for deep belief nets. / G. E. Hinton, S. Osindero, Y. W. Teh // *Neural Computation*. – 2006. – vol. 18. – P. 1527-1554.