

## КРАТКИЕ СООБЩЕНИЯ

УДК 004.032.26

### ОЦЕНКА ВРЕМЕННЫХ ЗАТРАТ НА РАЗРАБОТКУ С ИСПОЛЬЗОВАНИЕМ МЕТРИК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ю.В. ЛЯХ

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию 21 марта 2017

**Аннотация.** Исследована возможность использования метрик программного обеспечения для оценки временных затрат на разработку существующего программного продукта, а также для прогнозирования временных затрат на разработку нового программного продукта на стадии проектирования. Для выявления закономерностей в эмпирических данных было использовано машинное обучение с использованием нейронных сетей.

**Ключевые слова:** машинное обучение, нейронные сети, оценка временных затрат, метрики программного обеспечения.

**Abstract.** The article describes investigation of the ability to use software metrics for development time estimation of existing software product as well as investigation of the ability to predict development time of new software product at the design stage. Machine learning was used to identify empirical data regularities with neural networks using.

**Keywords:** machine learning, neural networks, time estimation, software metrics.

**Doklady BGUIR. 2017, Vol. 106, No. 4, pp. 83–87**  
**Development time estimation using software metrics**  
**Y.V. Liah**

#### Введение

Согласно исследованиям [1–3], погрешность в оценке сроков выполнения проектов составляет от 20 до 120 %. Эти статистические данные – симптомы глуболежащей проблемы недостаточности данных для проведения более точной оценки.

Собственных данных одной компании, как правило, недостаточно для получения качественной оценки. Нежелание других компаний делиться данными своих проектов понятно. Если не имеется возможности получить документальные данные о проекте, можно использовать данные о конечном программном продукте. Такими данными могут быть различные метрики программного обеспечения: количество строк кода (КСК), цикломатическая сложность, степень покрытия кода тестированием, количество объектов предметной области (КОПО), количество классов и интерфейсов (ККИ) [4]. Преимущества использования данных такого рода:

1. Доступно больше данных. Один проект может соответствовать нескольким программным продуктам (в качестве модулей или отдельных программ с высокой степенью интеграции). Построение оценочных моделей на основе проектных данных затруднено из-за недостаточного уровня документации и нежелания делиться своими данными с другими организациями.

2. Автоматизировать сбор данных проще. Существует большое количество программного обеспечения (в том числе бесплатного) для извлечения метрических показателей из конечного продукта.

### Проведение экспериментов

Для выявления закономерностей в эмпирических данных было использовано машинное обучение с использованием нейронных сетей. Преимущества использования нейронных сетей включают в себя: способность обобщать, адаптивность, гибкость структуры, высокая потенциальная параллельность вычислений, возможность работы при наличии большого числа неинформативных или избыточных данных, потенциально сверхвысокое быстродействие [5, 6].

Контролируемую нейронную сеть можно рассматривать как ориентированный граф, состоящий из узлов и соединений (весов) между узлами [7]. Каждый вектор состоит из входных и выходных значений. В проведенных экспериментах по определению временных затрат на разработку готового продукта входными данными служили три метрики программного обеспечения: количество строк кода, количество объектов предметной области, количество классов и интерфейсов. В экспериментах по прогнозированию временных затрат на разработку нового продукта на стадии проектирования метрика «количество строк кода» не использовалась, так как она недоступна на данном этапе. Во всех экспериментах выходными данными являлись временные затраты на разработку (в часах).

Цель нейронной сети – описать отношение между входными и выходными данными на полном наборе векторов. В процессе обучения входные данные тренировочного вектора распространяются по всей сети. Во время перемещения по сети, они умножаются на соответствующие веса и результаты суммируются. Если полученная сумма превышает порог для узла, то выходные данные для этого узла служат входными данными для другого узла (рис. 1).

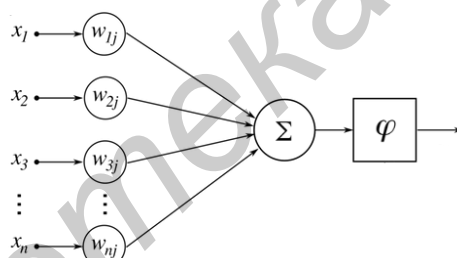


Рис. 1. Узел нейронной сети

Активационная (пороговая) функция нейронной сети – функция, вычисляющая выходной сигнал искусственного нейрона. В качестве пороговой функции во всех проведенных экспериментах использовалась сигмоидальная функция (рис. 2).

Описанный процесс повторяется до тех пор, пока нейронная сеть генерирует выходное значение для соответствующей входной части вектора. По окончании процесса рассчитанное значение сравнивается с известным эталонным результатом для определения отклонения (ошибки). Корректировка весов происходит после каждого полного прохода (называемого также эпохой). Конечная цель – минимизация отклонения. Если нет ограничения на количество эпох, то процесс останавливается при полном устранении отклонения. После корректного завершения тренировки, сформированная модель «замораживается» и проверяется на другом независимом наборе векторов (называемым тестовым набором).

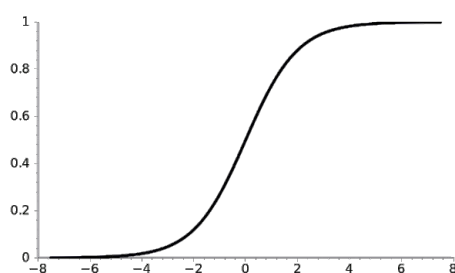


Рис. 2. Активационная сигмоидальная функция

## Результаты и их обсуждение

Эксперименты были проведены на двух наборах данных. Первый набор данных ( $X$ ) собран с программных продуктов предметной области «электронная коммерция», второй ( $Y$ ) – предметной области «автоматизированные системы управления проектами».

В каждом эксперименте нейронная сеть сначала обучается на одном из наборов данных  $X$  или  $Y$  (ограничение на количество эпох – 1000), затем на основе полученных знаний оценивает наборы соответственно  $Y$  и  $X$ .

В первом эксперименте производилась оценка временных затрат по трем метрикам программного обеспечения: КСК, КОПО, ККИ. Набор данных  $X$  выступал в качестве обучающего, набор  $Y$  – тестового.

Результаты, представленные в табл. 1, показывают, насколько точно нейронная сеть оценивает временные затраты. Реальное время выполнения показывает суммарное количество реальных часов, затраченное на выполнение проектов; рассчитанное время – суммарная оценка затрат системой.

Таблица 1. Оценка тестового набора  $Y$  по трем метрикам

Проба	Реальное время выполнения, ч	Рассчитанное время выполнения, ч	Отклонение, %
1	12196	12801	5
2	12196	12576	3
3	12196	12569	3
4	12196	12564	3
5	12196	12769	5
6	12196	12679	4
7	12196	12571	3
8	12196	12573	3
9	12196	12562	3
10	12196	12664	4

Повторим тот же эксперимент для обучающего набора  $Y$  и тестового набора  $X$ . Результаты представлены в табл. 2.

Таблица 2. Оценка тестового набора  $X$  по трем метрикам

Проба	Реальное время выполнения, ч	Рассчитанное время выполнения, ч	Отклонение, %
1	17624	16514	-6
2	17624	16337	-7
3	17624	16508	-6
4	17624	16573	-6
5	17624	16533	-6
6	17624	16516	-6
7	17624	16689	-5
8	17624	16584	-6
9	17624	16601	-6
10	17624	16540	-6

Точность оценок по трем метрикам программного обеспечения оказалась высокой. Это значит, что применяя данный подход возможно с высокой точностью оценить временные затраты на создание программного продукта (имея в распоряжении готовый продукт). Такая информация является ценной при разработке не уникальных продуктов. Оценив время разработки схожих продуктов, можно более качественно оценить временные затраты на собственный продукт на стадии проектирования.

В следующем эксперименте производилась оценка временных затрат по двум метрикам программного обеспечения: КОПО, ККИ. Целью данного эксперимента являлась оценка возможности прогнозирования длительности разработки уникального программного продукта. На стадии проектирования КОПО можно определить с очень высокой точностью, ККИ – с некоторой допустимой погрешностью. Получение информации о КСК не представляется возможным, поэтому данная метрика исключена из эксперимента.

Результаты эксперимента представлены в табл. 3, 4.

Таблица 3. Оценка тестового набора У по двум метрикам

Проба	Реальное время выполнения, ч	Рассчитанное время выполнения, ч	Отклонение, %
1	12196	5784	-111
2	12196	5890	-107
3	12196	5929	-106
4	12196	5701	-114
5	12196	5721	-113
6	12196	5708	-114
7	12196	5860	-108
8	12196	5894	-107
9	12196	5804	-110
10	12196	5797	-110

Таблица 4. Оценка тестового набора Х по двум метрикам

Проба	Реальное время выполнения, ч	Рассчитанное время выполнения, ч	Отклонение, %
1	17624	27164	54
2	17624	27121	54
3	17624	26799	52
4	17624	27348	55
5	17624	27246	55
6	17624	27122	54
7	17624	27035	53
8	17624	27147	54
9	17624	27146	54
10	17624	26911	53

Точность оценок по двум метрикам программного обеспечения оказалась крайне низкой. Следовательно, применение данного подхода не имеет смысла для прогнозирования сроков выполнения уникального продукта. Проведенный эксперимент также говорит о важности показателя КСК при оценке сроков создания программного продукта (в комбинации с метриками КОПО и ККИ).

### Заключение

Использование метрик программного обеспечения позволяет с высокой точностью оценить временные затраты на разработку типового программного продукта, что можно использовать при оценке сроков выполнения на стадии проектирования. Однако с ограниченным набором метрик, что имеет место при разработке уникального продукта, качество оценивания значительно падает и полученные результаты не несут практической пользы.

### Список литературы / References

1. Information Resources Management Association (IRMA). Project Management: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications. IGI Global, 2016. 2410 p.
2. Zwikael O., Smyrk J. Project Management for the Creation of Organisational Value. Springer Science & Business Media, 2011. 359 p.
3. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley, 2013. 1296 p.
4. Fenton N., Bieman J. Software Metrics: A Rigorous and Practical Approach. CRC Press, 2014. 617 p.
5. Roychowdhury V., Siu K., Orlitsky A. Theoretical Advances in Neural Computation and Learning. Springer Science & Business Media, 2012. 496 p.
6. Haykin S. Neural Networks and Learning Machines. Pearson Education, 2011. 936 p.
7. Artificial Neural Networks and Machine Learning / V. Mladenov [et al.]. Springer, 2013.

#### **Сведения об авторе**

Лях Ю.В., аспирант кафедры инженерной психологии и эргономики Белорусского государственного университета информатики и радиоэлектроники.

#### **Адрес для корреспонденции**

220013, Республика Беларусь,  
г. Минск, ул. П. Бровки, д. 6,  
Белорусский государственный университет  
информатики и радиоэлектроники  
тел. +375-33-656-04-10;  
e-mail: mat6560410@gmail.com;  
Лях Юрий Владимирович

#### **Information about the autor**

Liah Y.V., postgraduate student of department of engineering psychology and ergonomics of Belarusian state university of informatics and radioelectronics.

#### **Address for correspondence**

220013, Republic of Belarus,  
Minsk, P. Brovka st., 6,  
Belarusian state university of  
informatics and radioelectronics  
tel. +375-33-656-04-10;  
e-mail: mat6560410@gmail.com;  
Liah Yuri Vladimirovich

Библиотека БГУИР