Сравнение архитектур сиамской нейронной сети в рамках решения задача поиска похожих фрагментов кода

А.С. Юмаганов[1]

[1]*Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34а, Самара, Россия, 443086*

Аннотация

В работе рассматривается задача поиска похожих фрагментов кода (функций) в исполняемых файлах. В рамках предложенного ранее оригинального метода поиска, основанного на применении сиамской нейронной сети, исследуется влияние на его эффективность нескольких вариантов архитектуры используемой нейронной сети. Представлены результаты экспериментальных исследований эффективности решения задачи поиска похожих функций для каждого из них.

Ключевые слова

поиск, фрагменты кода, исполняемый файл, нейронная сеть

1. Введение

В настоящее время широкое развитие получила практика повторного использования кода программистами при разработке нового программного обеспечения. Наличие повторно используемого кода в программе может стать причиной нарушения авторских прав, возникновения ошибок и уязвимостей. Для обнаружения таких участков кода используются методы поиска похожих фрагментов кода, в которых код программы представляется в виде символьной последовательности команд и операндов [1,2] или в виде графа потока управления [3].

Настоящая работа является продолжением прежних исследований автора [2], посвященных разработке метода поиска похожих последовательностей кода в исполняемых файлах с помощью сиамской нейронной сети. Описанный ранее метод поиска использует следующие понятия: текущее множество объектов сравнения – множество объектов сравнения, для которых требуется найти похожие объекты сравнения; архивное множество объектов сравнения – множество известных объектов сравнения. С учетом представленных выше определений задача, решаемая рассматриваемым методом поиска, формулируется следующим образом: для заданной (или каждой) функции текущего множества найти наиболее похожую функцию из архивного множества. В рамках представленного метода поиска для формирования вектора окончательного описания функции исполняемого файла используется сиамская нейронная сеть, в основе которой лежит функция потерь contrastive loss.

В данной работе представлено сравнение двух вариантов архитектур сиамской нейронной сети (на основе двух функций потерь: contrastive loss и triplet loss), применяемых для решения задачи поиска похожих функций в исполняемых фалах.

2. Сиамская нейронная сеть

Сиамская нейронная сеть с функцией потерь contrastive loss [4] состоит из двух идентичных сетей с долгой краткосрочной памятью (LSTM), имеющих общие весовые коэффициенты. Векторное описание сравниваемых объектов (функций исполняемого файла) подаются на вход каждой из LSTM сетей. Выходы последних слоев подаются на вход функции *D*, оценивающей близость между ними. Функции потерь contrastive loss имеет следующий вид:

$$L(y,D) = 0,5*(1-y)D^2 + 0,5*y(\max(0,1-D))^2,$$

где y – выходное значение сиамской нейронной сети, принимающее значение 0, если объекты разные, и 1 – если объекты схожи.

В свою очередь сиамская нейронная сеть с функцией потерь triplet loss [5] состоит из трех идентичных LSTM сетей. На вход первой сети подается векторное описание опорного (anchor) объекта, на вход второй сети подается векторное описание объекта схожего с опорным, на вход третьей сети – векторное описание непохожего объекта. Функции потерь triplet loss имеет следующий вид:

$$L(D) = \max(D(a, p) - D(a, n) + m, 0),$$

где a – векторное описание опорного объекта, p – векторное описание объекта схожего с опорным; n – векторное описание объекта отличного от опорного; D – значение меры схожести двух объектов; m – параметр. Выделяют два основных подхода к выбору триплетов для обучения такой сети: hard mining и semi-hard mining. В первом случае для каждого опорного объекта в качестве похожего объекта выбирается наименее похожий объект (среди похожих), а в качестве непохожего – наиболее похожий (среди непохожих). Во втором случае выбираются триплеты, удовлетворяющие условию: $D(a, p) < D(a, n) < D(a, p) + m$.

В таблице ниже представлены результаты сравнения разных архитектур сиамской нейронной сети в рамках решения задачи поиска похожих функций в исполняемых файлах. Для обучения сетей использовалось по две версии двух исполняемых файлов, тестирование производилось на исполняемых файлах программной библиотеки curl.

Таблица 1
Сравнение архитектур сиамской нейронной сети

Текущее множество объектов сравнения	Средняя точность поиска P		
	Сиамская нейронная сеть с функцией потерь contrastive loss	Сиамская нейронная сеть с функцией потерь triplet loss	
		Hard mining	Semi-hard mining
curl 7.54.0	0,7501	0,7719	**0,7764**
curl 7.56.0	0,8403	**0,8540**	0,8538
curl 7.59.0	0,8745	0,8810	**0,8856**
curl 7.60.0	0,8859	0,8889	**0,8911**

3. Заключение

В работе рассмотрены две архитектуры сиамской нейронной сети, представлены результаты их сравнения в рамках представленного ранее метода поиска похожих функций в исполняемом файле. Лучший результат был продемонстрирован при использовании сиамской нейронной сети с функцией потерь triplet loss и способом выбора триплетов semi-hard mining.

4. Благодарности

Работа выполнена при частичной финансовой поддержке гранта РФФИ № 18-29-03135-мк.

5. Литература

[1] Юмаганов, А.С. Метод поиска похожих последовательностей кода в исполняемых бинарных файлах с использованием беспризнакового подхода / А.С. Юмаганов, В.В. Мясников // Компьютерная Оптика. – 2017. – Т. 41, № 5. – С. 756-764. DOI: 10.18287/2412-6179-2017-41-5-756-764.

[2] Yumaganov, A.S. Searching for similar code sequences in executable files using siamese neural network // CEUR Workshop Proceedings. – 2020. – Vol. 2667. – P. 203-206.

[3] Feng, Q. Scalable Graph-based Bug Search for Firmware Images / Q. Feng, R. Zhou, C. Xu, Y. Cheng, B. Testa, H. Yin // Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. – 2016. – P. 480-491.

[4] Hadsell, R. Dimensionality Reduction by Learning an Invariant Mapping / R. Hadsell, S. Chopra, Y. LeCun // IEEE Computer Society. – 2006. – Vol. 2. – P. 1735-1742.

[5] FaceNet: A Unified Embedding for Face Recognition and Clustering [Electronic resource]. – Access mode: https://arxiv.org/pdf/1503.03832 (29.12.2020).