Использование глубокого обучения для поиска ошибок в программном коде

Богомолов Егор Олегович
научный руководитель: Т. А. Брыксин

JetBrains Research

27 сентября 2018 г.

Мотивация, актуальность

- Разных видов ошибок много
- Видов инструментов для их поиска много
- Концентрируемся на статическом анализе

Мотивация, современные решения

Современные баг-детекторы:

- Статический анализ кода
- Поиск известных паттернов
- Сложный анализ графов потока управления и данных

И их проблемы:

- Ручной разбор случаев и дальнейшее добавление эвристик
- Тяжело расширять на новые баги
- Практически не используются знания об именах переменных, функций и т.д.

Мотивирующий пример

Знания о названиях полезны

```
int getSquare(int xDim, int yDim) { ... }

int x = 3, y = 4;
int s = getSquare(y, x);
```

Мотивация

Баг-детекторы работают!

```
Dog dog = new Dog();
...
Tree tree = new Tree();
...
if (dog.equals(tree)) {
    ...
}
```

Мотивация

Правильный ли порядок операндов?

```
1 - i
```

Мотивация

Ошибка

```
for (int i = 2; i < n; i++) {
    a[i] = a[1 – i] + a[i – 2];
}
```

Корректный код

```
int bitReverse(int i) {
    return 1 – i;
}
```

Страшный пример

ID	Type	Parameter	Original argument		Correct argument	Days in repo	Comment
1	Duration	responseTTLDuration	frequencyCapDuration	⤫	responseTTLDuration	11	Discovered because of request to test an existing feature.
	Duration	frequencyCapDuration	responseTTLDuration	⤫	frequencyCapDuration		
	List<A>	slotResponse	slotResponse	→	slotResponse		
2	long	communityId	a.toDataObject().getId()	→	a.toDataObject().getId()	14	Many arguments with general-purpose types.
	long	senderId	e.getSenderId()	→	e.getSenderId()		
	long	recipientId	e.getRecipientId()	→	e.getRecipientId()		
	long	subject	subject	→	subject		
	String	textContent	htmlContent	⤫	textContent		
	String	htmlContent	textContent	⤫	htmlContent		
3	User	owner	owner	→	owner	792	Highest priority bug, fixed within 1 hour of filing. Called method is over-loaded in many different ways.
	long	objId	objId	→	objId		
	String	streamId	null	→	null		
	String	authKey	null	⤫	authKey		
	String	tag	authKey	⤫	null		
	int	offset	offset	→	offset		
	int	maxResults	maxResults	→	maxResults		
	Timer	timer	timer	→	timer		
	ComponentType...	components	components	→	components		
4	Builder	builder	builder	→	builder	163	Names are equal after removing the `get` prefix of method names
	boolean	isTransposed	isTransposed	→	isTransposed		
	int	startColumnIndex	a.getStartColumnIndex()	⤫	0		
	int	endColumnIndex	a.getEndColumnIndex()	⤫	rows.size()		
	int	startRow	0	⤫	a.getStartColumnIndex()		
	int	endRow	rows.size()	⤫	a.getEndColumnIndex()		
5	String	msgFormat	"Message"	⤫	e	139	Incorrect ordering resulted in wrong method being called because of over-loading
	Object...	args	e	⤫	"Message"		
6	Object	actual	Util.createThing(fromStuff)	⤫	someVariable	1504	Calls to JUnit's `assertEquals` method that swap expected and actual.[2]
	Object	expected	someVariable	⤫	Util.createThing(fromStuff)		
7	String	fileUser	stageName	⤓	fileUser	20	Needed a field instead of a local variable
	Collection<String>	writtenFilenames	customKeywordFiles	→	customKeywordFiles		
8	Object...	parts	Long.parseLong(getAccountId())	→	Long.parseLong(getAccountId())	53	Argument accidentally duplicated.
			Long.parseLong(getAccountId())	⤓	Long.parseLong(getAudienceId())		

Предпосылки к решению

Что хотим от баг-детектора:

- Легко добавлять новые баги
- Использовать больше информации из кода программы
- Сделать поддерживаемые классы ошибок более общими

Решение

Использовать машинное обучение!

Этапы решения

- Собрать много положительных примеров
- Собрать много отрицательных примеров
- Векторизовать фрагменты кода
- Обучить модель

Данные

- Положительные примеры — почти весь код
- Отрицательные примеры — тот же код, но испорченный
- Датасеты: JS 150K, Python 150K

Поясняющий пример

Исходный код

```
function setSize(width, height) {
    ...
}

var w = 5;
var h = 10;
setSize(w, h);
```

Положительный

```
setSize(w, h);
```

Отрицательный

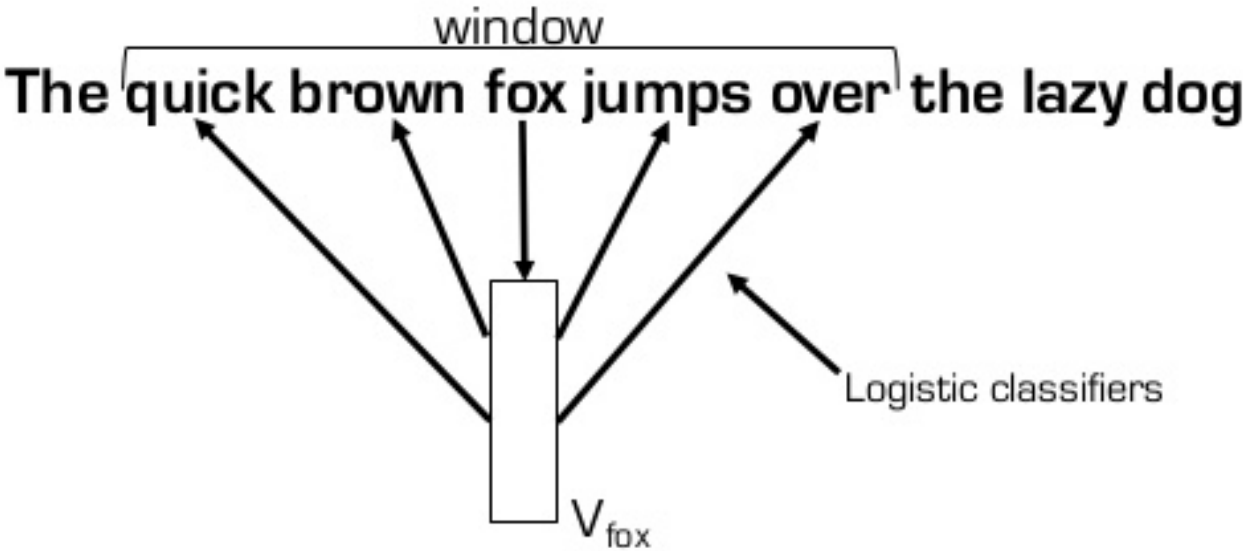
```
setSize(h, w);
```

Векторизация

- Вектор — кортеж из интересных нам токенов
- Основная проблема — векторизация отдельных токенов
- Решение — Word2Vec и развитие его идеи

Word2Vec: How it works?

- Map every word to an embedding
- Use a window around a selected word
- Use the embedding of the selected word to predict the context of the word



Применение Word2Vec

- Разбиение на токены:

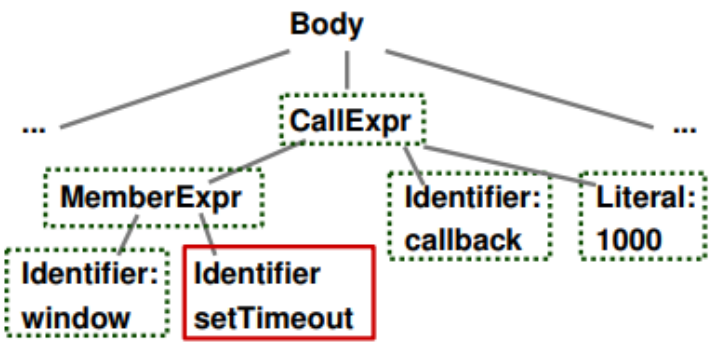
$$db.allNames() = db|.|allNames|(|)$$

- Код — это последовательность токенов
- По токену предсказываем окружение, получаем векторизацию

Усложнение Word2Vec

```
...
window.setTimeout(callback, 1000);
...
```

(a) JavaScript code.



Part of context	Value(s)
Parent p	MemberExpr
Position p_{pos} in parent	2
Grand-parent g	CallExpr
Position g_{pos} in grand-parent	1
Siblings S	{ ID:window }
Uncles U	{ ID:callback, LIT:1000 }
Cousins C	{}
Nephews N	{}

Полученное отображение

ID:length

Simil.	Identifier
0.59	ID:allowed_chars
0.56	ID:_length
0.52	ID:total_length
0.5	ID:minLength
0.5	ID:maxLength
0.47	ID:charSet
0.46	LIT:length
0.46	ID:get_length
0.45	ID:offset
0.45	ID:_position

ID:wrapper

Simil.	Identifier
0.69	ID:decorator
0.65	ID:wrapped
0.63	ID:update_wrapper
0.6	ID:new_func
0.58	ID:_wrapper
0.56	ID:_decorator
0.54	ID:func
0.53	ID:decorated
0.51	ID:WrappingFactory
0.51	ID:decorated_function

Векторизация на примерах

- Переставленные аргументы в вызове функции:

$$(n_{base}, n_{callee}, n_{arg1}, n_{arg2}, t_{arg1}, t_{arg2}, n_{param1}, n_{param2})$$

- Неправильный бинарный оператор или операнд:

$$(n_{left}, n_{right}, op, t_{left}, t_{right}, k_{parent}, k_{grandP})$$

Векторизация на примерах

- Переставленные аргументы в вызове функции:

$$(n_{base}, n_{callee}, n_{arg2}, n_{arg1}, t_{arg2}, t_{arg1}, n_{param1}, n_{param2})$$

- Неправильный бинарный оператор:

$$(n_{left}, n_{right}, op', t_{left}, t_{right}, k_{parent}, k_{grandP})$$

- Неправильный операнд:

$$(n'_{left}, n_{right}, op, t'_{left}, t_{right}, k_{parent}, k_{grandP})$$

$$(n_{left}, n'_{right}, op, t_{left}, t'_{right}, k_{parent}, k_{grandP})$$

Названия выражений

Expression	Extracted name
`list`	ID:list
`23`	LIT:23
`this`	LIT:this
`i++`	ID:i
`myObject.prop`	ID:prop
`myArray[5]`	ID:myArray
`nextElement()`	ID:nextElement
`db.allNames()[3]`	ID:allNames

Модель

- Двуслойный перцептрон
- 200 скрытых нейронов, ReLU
- Два слоя dropout=0.2
- Точность 85-95%

Результаты

Bug detector	Reported Bugs		Code quality problem	False positives
Swapped arguments	178	75	10	93
Wrong assignment	24	1	1	22
Wrong bin. operator	50	14	17	19
Wrong bin. operand	38	10	4	22
Total	290	100	32	156

Результаты, примеры

Обнаруженный баг #1

```
var p = new Promise();
if (promises === null || promises.length === 0) {
    p.done(error, result);
} else {
    promises[0](error, result).then(function(res, err) {
        p.done(res, err);
    });
}
```

Результаты, примеры

Обнаруженный баг #2

```
for (j = 0; j < param.replace; j++) {
    if (param.replace[j].from === paramVal)
        paramVal = param.replace[j].to;
}
```

Статистика

Bug detector	Examples		Training		Prediction	
	Training	Validation	Extract	Learn	Extract	Predict
Swapped arguments	1,450,932	739,188	7:46	20:29	2:56	5:19
Wrong assignment	2,274,256	1,090,452	2:40	22:45	1:29	4:03
Wrong bin. operator	4,901,356	2,322,190	2:44	51:16	1:28	12:16
Wrong bin. operand	4,899,206	2,321,586	2:44	51:13	1:28	10:09

Перенос на Python

- Получение информации из исходников в том же формате, что и для JS
- Простота добавления новых багов соблюдена
- Определение переставленных аргументов:
  - 90+% false-positive
- Неправильный операнд:
  - 20% false-positive
  - 20% переопределенные операторы
  - 60% ошибки

Плагин для PyCharm

Хорошие стороны:

- Модель быстро предсказывает
- Модель весит около 1Mb
- Можно реализовать испекцию и собирать данные

Плохие стороны:

- Для 50000 векторов файл с ними весит 250Mb
- Векторизация “из коробки” тормозит всю IDE
- Требуется оптимизация векторизации

Плагин для PyCharm, что дальше?

- Уже реализовано все для определения неверного оператора
  - Сбор данных в формате PyCharm
  - Инспекция, задействующая модель
- Добавить другие баг-детекторы
- Ускорить векторизацию
- Запустить плагин “в мир”

Ссылки на работу:

- Оригинальная статья
- Чуть более техническая версия статьи
- github.com/ml-in-programming/DeepBugs
- github.com/ml-in-programming/DeepBugsPlugin