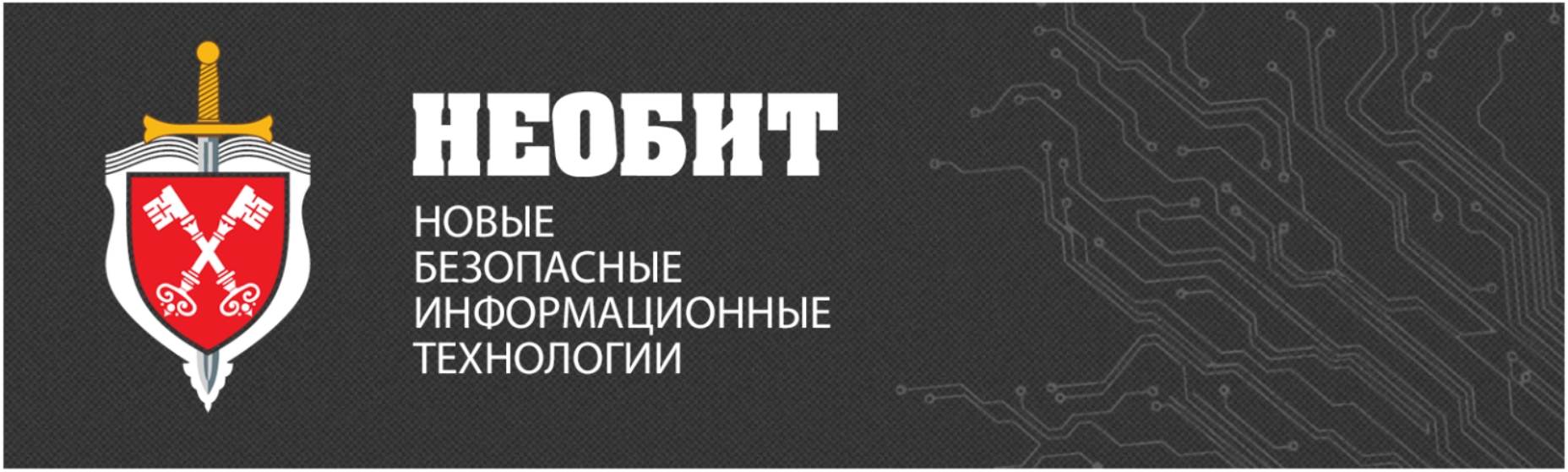


Поиск уязвимостей в программных компонентах киберфизических систем с помощью методов глубокого обучения

Демидов Р.А., аналитик

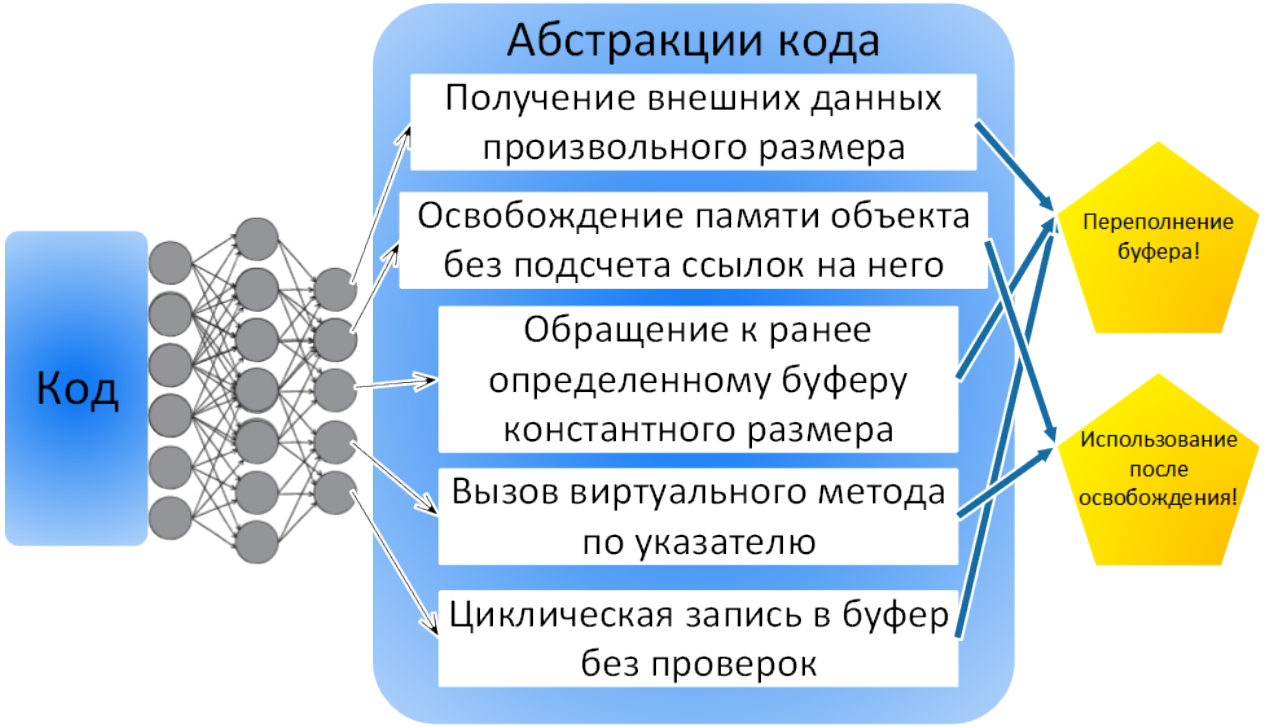
Как выявляются уязвимости?

- **Фаззинг** - косвенно: через перебор входов и падение программы на некоторых из них.
- **Символьные вычисления** - строго: через анализ путей к потенциально уязвимым местам в коде и проверку выполнимости системы ограничений на символьные переменные.
- **Аналитик**:
  а) “понимает” код без непосредственного исполнения.
  б) “видит” уязвимости, имея общее представление о работе программы.
  в) лучше понимает код высокого уровня, чем машинный код.

Разыскивается алгоритм:

- “Смотрит” на код так, как это делает аналитик.
- Строит представление кода высокой степени абстракции
- Сигнализирует о наличии уязвимостей во входном образце кода, основываясь на построенном представлении.

Уязвимость как свойство кода



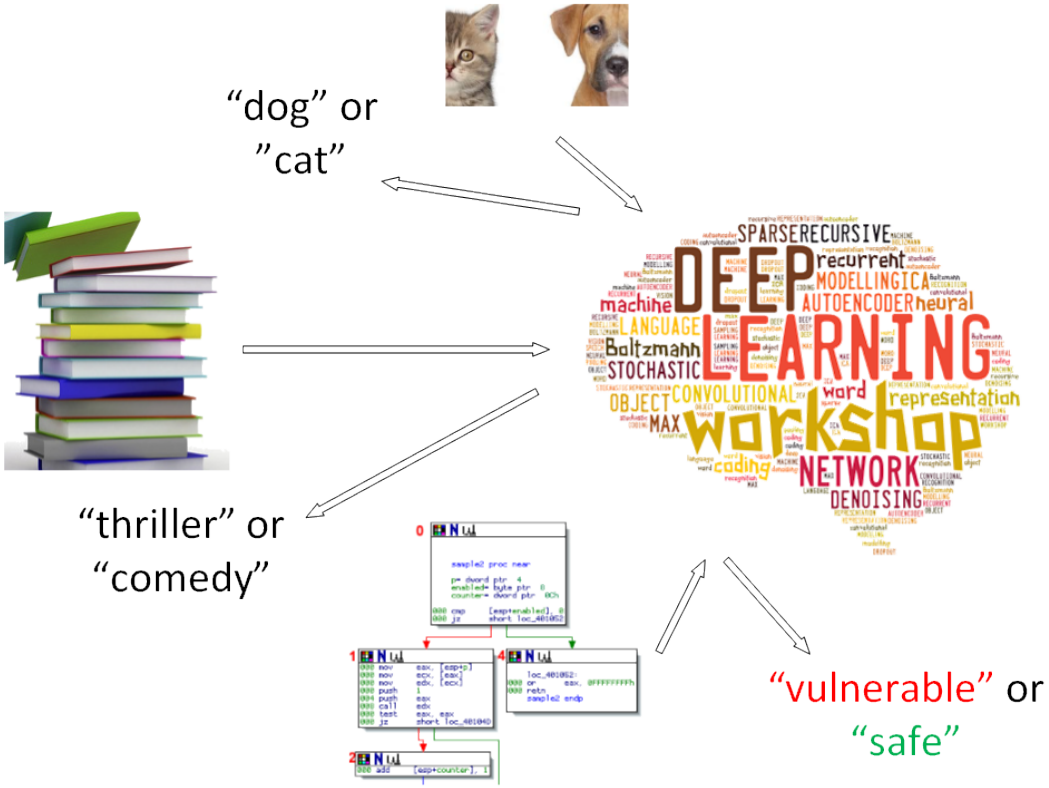
Уязвимости:

- это абстракции кода *как целого*.
- комбинации *менее абстрактных* свойств в том же коде.
- есть в разных программах, но имеют общие черты.

НЕОБИТ

Как построить такой алгоритм?

- **Глубокое обучение (Deep Learning) –** набор алгоритмов машинного обучения, моделирующих высокоуровневые абстракции в данных с использованием иерархии нелинейных преобразований данных.



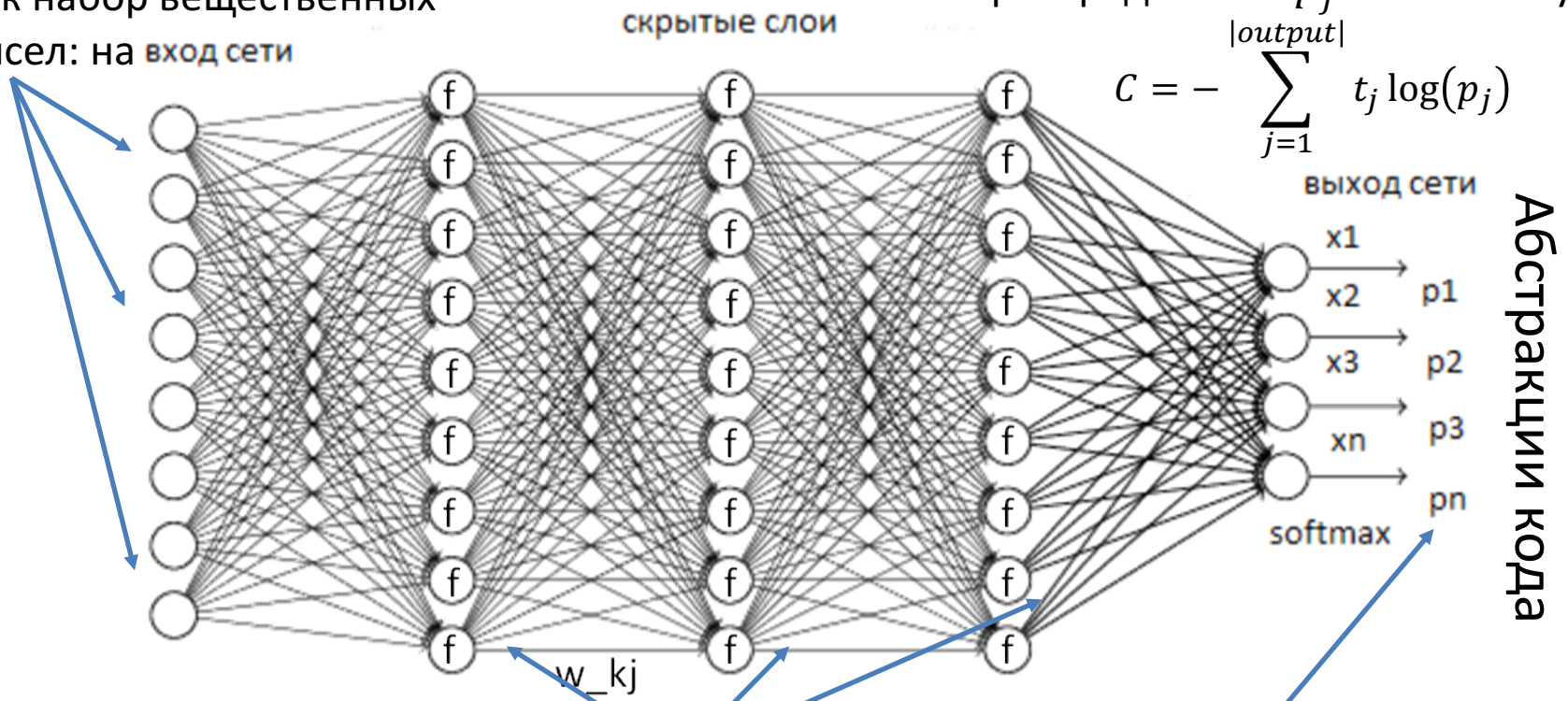
НЕОБИТ

Как построить такой алгоритм?

- На сегодня глубокое обучение успешно используется для:
  - классификации изображений и видео,
  - выявление эмоций в голосе,
  - выявления сложных свойств фраз в тексте (эмоций, акцента, ключевого слова, оттенка речи и т.д.).
- **Предлагается применить подходы глубокого обучения для задачи классификации машинного кода по типам программных уязвимостей.**

Глубокие сети в задачах классификации

Данные, закодированные как набор вещественных чисел: на вход сети



Абстракции кода

Кросс-энтропия C: похожесть распределения p_j эталонному t_j:

$$C = -\sum_{j=1}^{|output|} t_j \log(p_j)$$

Выход нейрона x_j слоя L: функция активации
- взвешенная сумма предыдущего слоя.

$$x_j = f(s_j) = f\left(\sum_{k}^{|L-1\ layer|} w_{kj} x_k\right)$$

Softmax: вероятности p_j принадлежности к классам через выходы всей сети.

$$p_j = \frac{\exp(x_j)}{\sum_{k=1}^{|output|} \exp(x_k)}, \sum p_j = 1$$

НЕОБИТ

Обучение семантике инструкций

- Текст/код – **данные дискретны**. Разреженное(sparse) кодирование:

$$\{\text{инструкция}\} \xrightarrow{\varphi} [0,..,1,0,..0] \in \mathbb{R}^{|dictionary|}$$

- **Векторное представление** дискретных объектов – плотное(dense) представление ϕ объектов в $\mathbb{R}^n$ с учетом их семантики.

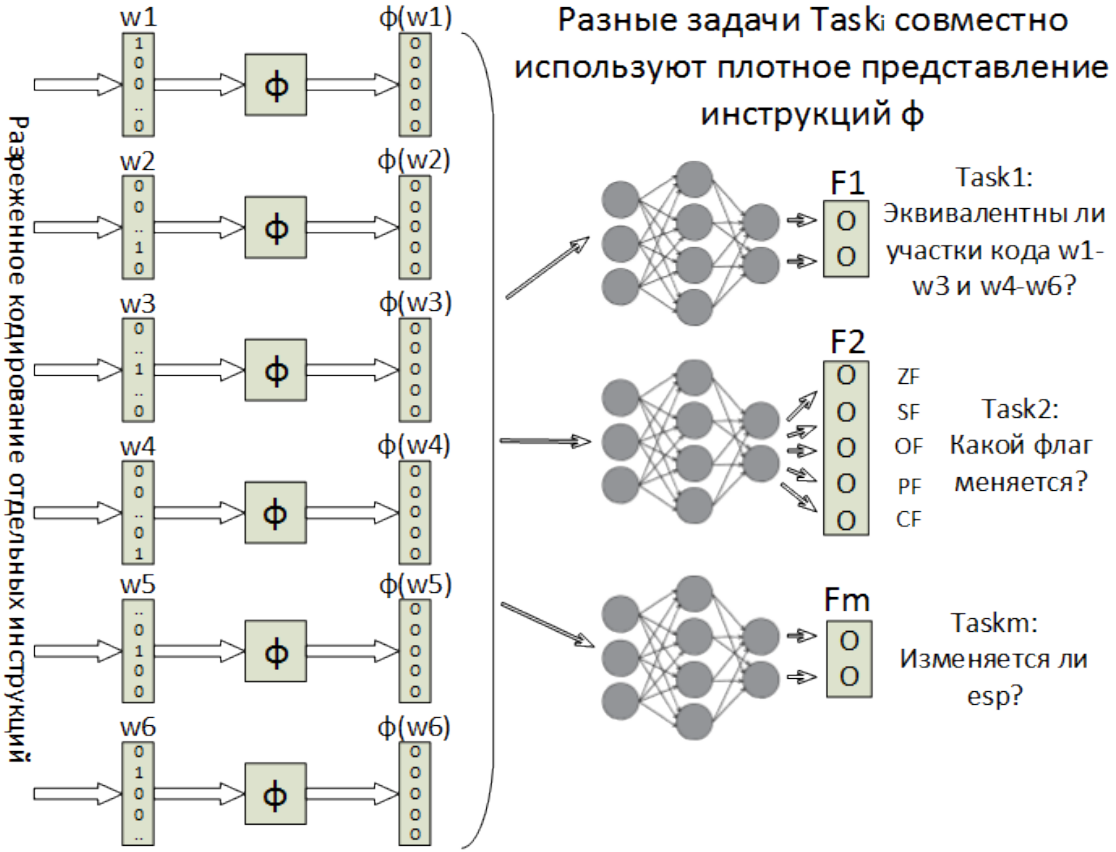
$$w \xrightarrow{\phi} \phi w, w \in \mathbb{R}^{|dictionary|}, \phi \in \mathbb{R}^{n*|dictionary|}, \phi w \in \mathbb{R}^n$$

- **Цель:** близкие по семантике объекты имеют близкие образы в $\mathbb{R}^n$.
- **Фаза предварительного обучения**: построение векторных представлений отдельных инструкций.

Как их обучать?

Обучение семантике инструкций

- **Shared representations**: неявно обучаемся единой семантике инструкций $w \xrightarrow{\phi} \phi w$ при последовательном обучении решения множества смежных задач $\{Task_i\}$:



НЕОБИТ

Алгоритм обратного распространения ошибки

- Метод послойного обучения нейронных сетей на помеченных данных $(\{x_i^0\}, \{t_j\})$.
- Градиентный спуск в пространстве весов:

$$\delta w_{ij} = \eta \frac{\partial C}{\partial w_{ij}}, 0 < \eta < 1$$

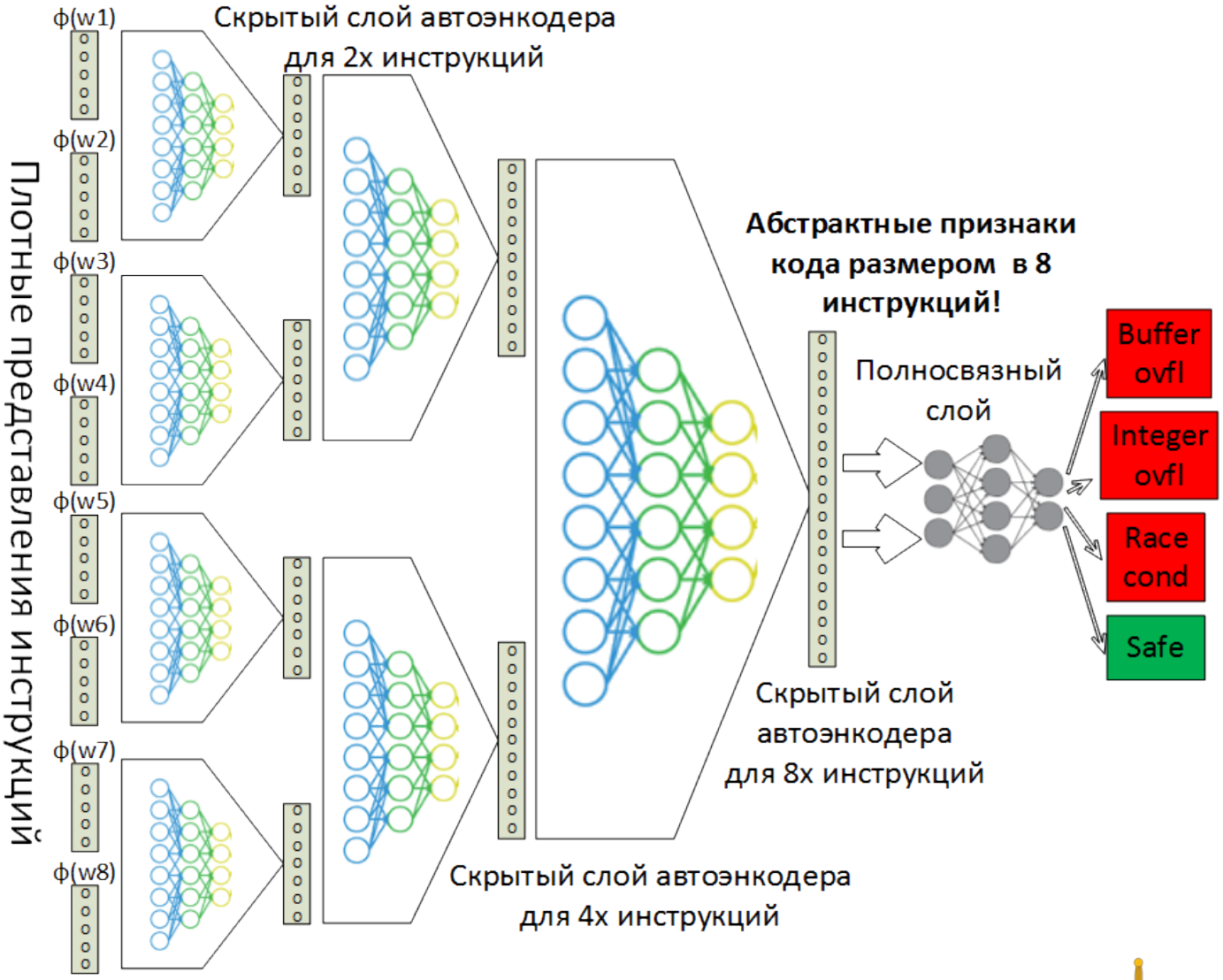
- Поправки весов последнего слоя:

$$\frac{\partial C}{\partial w_{ij}} = \sum_{k}^{|output|} \frac{\partial C}{\partial s_k} \frac{\partial s_k}{\partial w_{ij}} = \sum_{k}^{|output|} (x_k - t_k) \frac{\partial s_k}{\partial w_{ij}}$$

- Поправки весов нижних слоев через поправки верхних слоев:

$$\frac{\partial C}{\partial w_{ij}} = \sum_{k}^{|L|} \frac{\partial C}{\partial s_k} \frac{\partial s_k}{\partial w_{ij}} = \sum_{k}^{|L|} \left(\sum_{l}^{|L+1|} \frac{\partial C}{\partial s_l} \frac{\partial s_l}{\partial x_k} \frac{\partial x_k}{\partial s_k} \right) \frac{\partial s_k}{\partial w_{ij}}$$

Архитектура сети



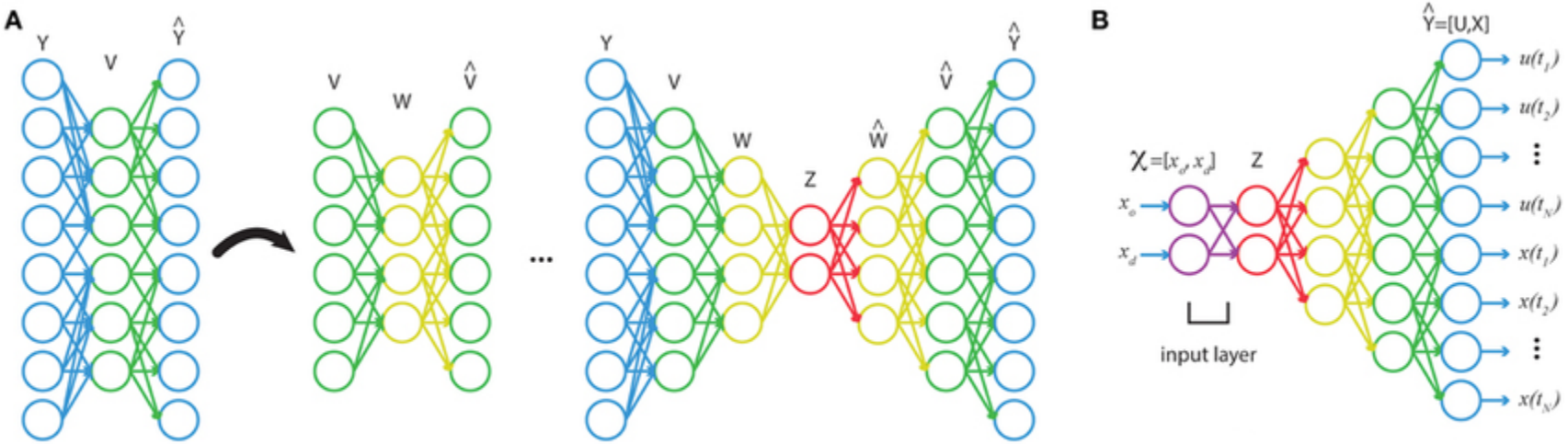
НЕОБИТ

Обучение сети

- **Вход сети** – вектор представлений подряд идущих инструкций.
- **Автоэнкодеры**: послойное обучение на непомеченных данных абстрактным признакам на признаках предыдущего слоя.
- **Предобучение промежуточных слоев**: последовательно строим представления для 2, 4, 8,… подряд идущих инструкций путем обучения автоэнкодеров.
- **Вход полносвязного слоя**: выделенный набор представлений всего кода**.**
- **Обучение на помеченных данных:**
  1) изменяются только веса в полносвязном слое.
  2) оптимизируется близость вычисленного распределения к тренировочному.
  3) тем самым выявляются комбинации высокоуровневых свойств кода, приводящие к уязвимостям разных классов.

Автоэнкодеры

- Размер входного слоя = размеру выходного: $|Y| = |\hat{Y}|$. **Критерий оптимальности: близость выхода сети ко входу.**
- **Средний слой Z является новым представлением данных**.
- Слою кодирования соответствует слой раскодирования: $|Y| \Leftrightarrow |\hat{Y}|, |V| \Leftrightarrow |\hat{W}|$ …
- Размеры слоев кодирования меньше размеров предыдущих: $|Y| > |V| > |W| > |Z| < |\hat{W}| < |\hat{V}| < |\hat{Y}|$
- Послойное обучение на непомеченных данных, начиная с крайних слоев: $(V, \hat{V}), (W, \hat{W}), Z$

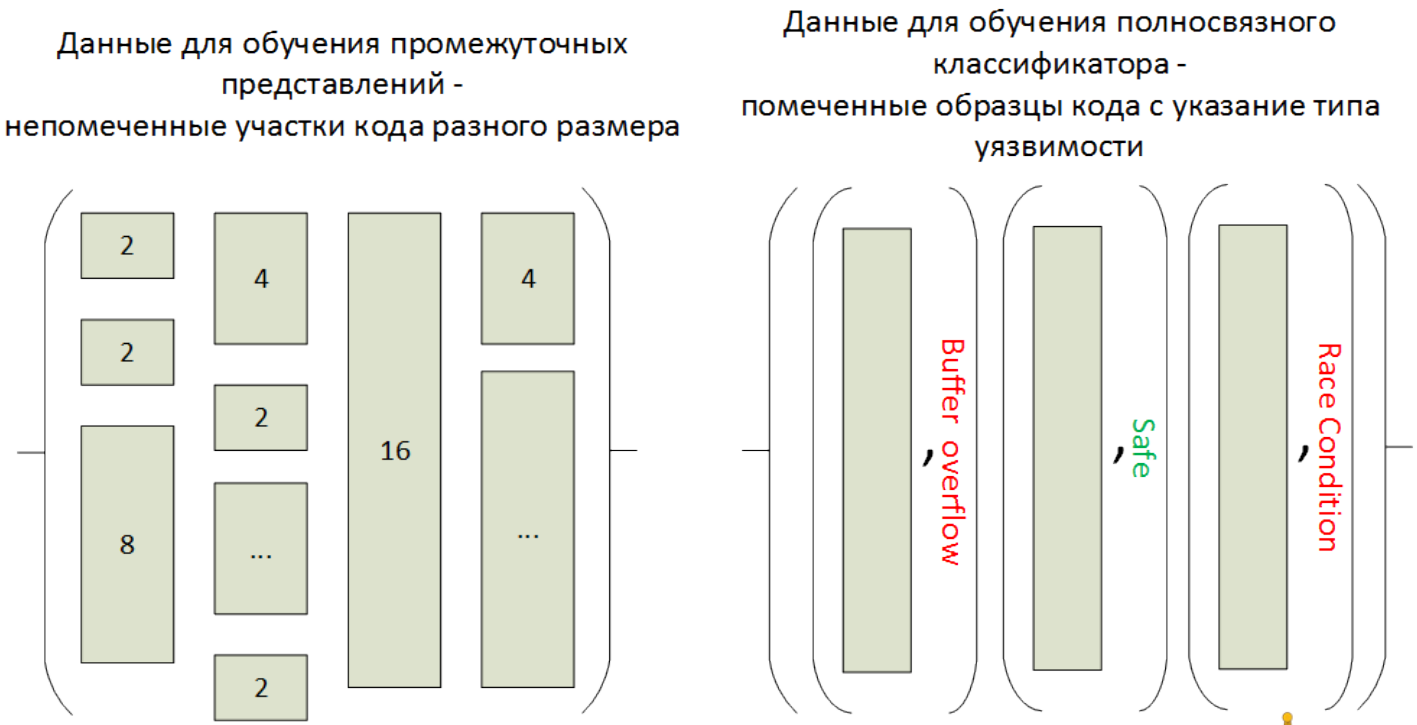


Послойное обучение | Архитектура сети | Раскодирующие слои

НЕОБИТ

Тренировочные данные

- Несколько фаз обучения – несколько видов данных.
- **Предобучение промежуточных слоев**: непомеченный набор фрагментов кода разных размеров.
- **Обучение классификатора уязвимостей**: данные помечены - $\{(code, \{t_j\})_i\}$



НЕОБИТ

Разработанный подход

- **Используемая архитектура**: глубокая нейронная сеть.
- **2 фазы обучения**:
  1) обучение векторным представлениям отдельных инструкций,
  2) обучение абстрактным представлениям о наличии уязвимостей разных классов в коде.
- **Данные:** образцы кода с пометками и без для соответствующих фаз обучения.
- **Результат обучения:** сеть выявляет уязвимости в новых образцах кода.

НЕОБИТ

Санкт-Петербург, ул. Гжатская, 21 литер Г
Тел./факс: (812) 535-28-06
Сайт: neo-bit.ru
Почта: info@neo-bit.ru