УДК 003.26+004.032.26

ПРИМЕНЕНИЕ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ СТЕГАНОГРАФИИ

О.Д. ЮШКЕВИЧ, М.В. МАЛЬЦЕВ
Белорусский государственный университет, г. Минск, Республика Беларусь

Введение. Стеганография – направление криптологии (в широком смысле), цель которого – разработка средств передачи и хранения информации, скрывающих сам факт ее передачи или хранения [1]. Для достижения этой цели секретное сообщение встраивается в контейнер – данные, которые передаются или хранятся в открытом виде. В качестве контейнера как правило используются изображения, аудио и видео файлы, при этом встроенное секретное сообщение не обнаруживается человеческими органами чувств. Таким образом, секретные данные скрываются от пассивного наблюдателя и для их извлечения требуются специальные методы. Существуют различные подходы к построению таких методов: так называемый «слепой» стегоанализ используется, если математическая модель данных неизвестна [2], значимые для теории и практики результаты получены с помощью вероятностно-статистического моделирования [3, 4]. В настоящей работе для решения задач стеганографии применяются математические модели, доказавшие свою эффективность для решения многих практических задач, – нейронные сети.

1. Постановка задачи. Путь имеется контейнер C, представляющий собой изображение в формате RGB с разрешением $W \times H$ пикселей, состоящее из трех цветовых 8-битных каналов. Рассматривается задача встраивания в контейнер C секретного сообщения $M \in \{0,1\}^{d \cdot W \cdot H}$ длиной $d \cdot W \cdot H$ бит (короткие сообщения дополняются незначащими нулевыми битами), где $d \in \{1, 2, ..., 24\}$ – параметр, характеризующий число бит, встроенных в один пиксель контейнера C, который будем называть глубиной кодирования (максимальное значение $d = 24$ означает, что исходное изображение C полностью заменяется другим изображением, которое соответствует секретному сообщению M). Имеются два алгоритма: кодировщик E и декодировщик D. Кодировщик получает на вход контейнер C и сообщение M и выдает на выходе контейнер C' со встроенным в него сообщением M: $E(C, M) = C'$. Декодировщик получает на вход C' и выдает на выходе сообщение M: $D(C') = M$. Рассматриваемая в настоящей статье задача стеганографии состоит в разработке использующего нейронные сети алгоритма встраивания секретного сообщения M в контейнер C.

2. Архитектура генеративно-состязательной нейронной стеганографической сети. Для решения задачи стеганографии в настоящей статье используются генеративно-состязательные нейронные сети (generative adversarial network, GAN), предложенные сотрудником компании Google Яном Гудфеллоу (Ian Goodfellow) [5]. В базовом варианте данной модели имеются две нейронные сети: генератор и дискриминатор. Генератор порождает объекты, принадлежащие различным классам, а дискриминатор пытается отличать объекты из разных классов. В настоящей статье рассматривается более сложная модель, состоящая из трех нейронных сетей: Кодировщика, Декодировщика и Критика, структура которых приведена далее.

Основными строительными блоками используемых нейронных сетей являются сверточные слои. Свертка изображения $I = (I_{ij})$ (матрица размерности $W \times H$) по ядру $K = (K_{ij})$ (матрица размерности $w \times h$, $w \le W, h \le H$) представляет собой линейный фильтр, который проходит по изображению и ставит в соответствие ($w \times h$)-матрице пикселей число по следующему правилу:

$$(I * K)_{xy} = \sum_{i=1}^{w} \sum_{j=1}^{h} K_{ij} \cdot I_{x+i-1, y+j-1}, \; 1 \le x \le W - w + 1, \; 1 \le y \le H - h + 1. \qquad (1)$$

В процессе обучения нейросеть определяет оптимальное значение *K*, минимизируя заданную функцию потерь. На практике это приводит к тому, что сверточный слой учится выделять определенные признаки, и чем глубже находится слой, тем более абстрактные признаки он выделяет. Первые слои как правило выделяют общие признаки, такие как изменения контраста, формы, резкость изображения, границы изображения. Комбинируя эти слои, становиться возможным понижать размерность изображения и извлекать из него полезные признаки.

Архитектура Кодировщика. Кодировщик *E* получает на вход исходное изображение с и сообщение $M \in \{0,1\}^{d \cdot W \cdot H}$. Задача Кодировщика – встроить *M* в *C* таким образом, чтобы Критик не обнаружил факта встраивания. Важно отметить, что значение параметра *d* может регулироваться на этапе инициализации сети. Рассмотрим три варианта архитектуры Кодировщика. Обозначим: $Conv_{i \to j}(\mathrm{X})$ – операция применения к изображению *X*, состоящему из *i* каналов, *j* операций свертки (1), в результате чего формируется *j*-канальное изображение. Параметры ядер свертки вычисляются в ходе обучения нейронной сети, о котором сказано далее (начальные значения ядер генерируются случайным образом).

Для всех вариантов архитектуры Кодировщика общими являются следующие начальные операции:

1. Обработка изображения с сверточным блоком:

$$\alpha = Conv_{3 \to 32}(C).$$

2. Конкатенация секретного сообщения *M* и полученной свертки (операция конкатенации обозначена ||), обработка полученного тензора сверточным блоком:

$$\beta = Conv_{32 \to 32}(\alpha \,||\, \mathrm{M}).$$

3. Последовательное применение сверточных блоков к тензору β:

$$C' = E(C, M) = Conv_{32 \to 3}(Conv_{32 \to 32}(\beta)).$$

На последнем (третьем) шаге формируется изображение C' со встроенным секретным сообщением *M*.

Второй вариант архитектуры кодировщика, которого обозначим E_r (от англ. residual – остаток) основан на работе [6], в которой показано, что остаточные связи как правило улучшают сходимость и устойчивость алгоритма, поэтому предполагается, что их использование улучшит качество изображения со встроенным секретным сообщением: C' формируется следующим образом (знак + здесь означает попиксельное сложение изображений):

$$E_r(C, M) = C + Conv_{32 \to 3}(Conv_{32 \to 32}(\beta)).$$

В третьем варианте архитектуры кодировщика, которого обозначим E_d (от англ. Dense – плотный), используются дополнительные связи между всеми слоями, что позволяет слоям на разных уровнях использовать признаки других уровней. Формально это записывается следующим образом:

$$\gamma = Conv_{64d \to 32}(\alpha \,||\, \beta \,||\, M),$$

$$\delta = Conv_{96d \to 3}(\alpha \,||\, \beta \,||\, \gamma \,||\, M),$$

$$E_d(C, M) = C + \delta.$$

Результатом каждого варианта является изображение C', содержащее секретное сообщение *M* и имеющее те же характеристики разрешения и глубины, что и исходное изображение (контейнер) *C*.

Архитектура Декодировщика. Декодировщик *D* получает на вход изображение C' и возвращает некоторое сообщение $\hat{M}$. Задача обучения нейронной сети состоит в том, чтобы добиться совпадения $\hat{M}$ и *M*, т. е. того, что Декодировщик верно восстанавливает секретное сообщение *M*. В ходе работы Декодировщика выполняются следующие операции:

$$\alpha = Conv_{3\to 32}(C'),$$

$$\beta = Conv_{32\to 32}(\alpha),$$

$$\gamma = Conv_{64\to 32}(\alpha \| \beta),$$

$$D(C') = Conv_{96\to dWH}(\alpha \| \beta \| \gamma).$$

Архитектура Критика. Задача Критика *Cr* – оценить разность между исходным изображением *C* и изображением, полученным Кодировщиком, т. е. определить изменения, вносимые в *C* встраиваемым секретным сообщением *M*. В ходе работы Критика выполняются следующие операции:

$$\alpha = Conv_{32\to 32}(Conv_{32\to 32}(Conv_{32\to 32}(C'))),$$

$$Cr(C') = Mean(Conv_{32\to 1}(\alpha)),$$

где $Mean(X)$ означает среднее арифметическое всех значений матрицы *X*.

Процесс Обучения. Будем использовать стохастический градиентный спуск для минимизации следующих функций потерь:

$$L_d = E_{XP_C}\left(\sum y_i \log y_i\right),$$

$$L_s = E_{XP_C}\frac{1}{3WH}\left\|(C - E(C,M))^2\right\|_2,$$

$$L_r = E_{XP_C}\left(Cr(E(C,M))\right),$$

где E_{XP_C} означает усреднение по всем элементам из используемого множества изображений (датасета), y_i – *i*-я компонента *M*. Целью обучения является минимизация $L_d + L_s + L_r$. Сеть Критик оптимизируем дополнительно, для чего минимизируем функцию:

$$L_c = E_{XP_C}\left(Cr(C)\right) - E_{XP_C}\left(Cr(E(C,M))\right).$$

В приведенных далее компьютерных экспериментах во время каждой итерации сопоставляется изображение *C* с секретным сообщением *M*, которое генерировалось случайным образом. Использовался оптимизационный алгоритм Adam [7], норма градиента уменьшалась до 0,25.

3. Результаты вычислительных экспериментов

3.1. Масштабирование данных. Первоначально для обучения использовался датасет, содержащий 1 500 изображений размерности 64×64 пикселя: 1 200 изображений использовались для обучения, 300 – для валидации. Далее датасет был увеличен до 75 000 изображений с целью проверки гипотезы о росте метрик при увеличении датасета. Из 75 000 изображений 55 000 использовались для обучения, 20 000 – для валидации. Обучение 30 эпох (шагов градиентного спуска) на таком датасете заняло около 150 часов на видеокарте NVidia Tesla K80. В результате обнаружено, что увеличение количества данных не приводит к увеличению метрик, и последующие эксперименты проводились с первоначальным датасетом – установлено, что его достаточно для обеспечения 100%-ной точности восстановления секретного сообщения для *M* для глубины кодирования *d*=1.

Масштабирование слоев. Сети Кодировщик и Декодировщик были расширены до 7 сверточных слоев, содержащих по 32 фильтра, обучение проводилось на 1 500 изображениях. В результате экспериментов установлено, что количество эпох, необходимых для сходимости сети – достижения Декодировщиком 100%-й точности восстановления *M* – выросло с 30 до 65, при этом прироста в метриках не наблюдалось. Более того, при добавлении обычных слоев без остаточных связей в сеть Критик наблюдается регресс сети и ухудшение метрик. Это объясняется тем, что добавление слоев смещает акцент Критика на более высокоуровневые признаки, в то время как два слоя выделяют более низкоуровневые признаки, такие как контраст, резкость и т. д., но именно по их изменению можно отследить,

что в изображение встроено секретное сообщение. Исходя из того, что масштабирование слоев не дает прироста, или даже вызывает регресс, полагаем, что количество слоев в исходной архитектуре достаточно для поставленной задачи стеганографии.

Масштабирование глубины кодирования. Одним из главных гиперпараметров рассматриваемой архитектуры является глубина кодирования d. Этот параметр определяет количество битов, встраиваемых в один пиксель изображения. Разумно предположить, что чем больше битов встраивается, тем сильнее меняется изображение, и тем легче Критик обнаруживает секретное сообщение M, и тем дольше работает Декодировщик, восстанавливая M из C', что проиллюстрировано в результатах вычислительных экспериментов, представленных в таблице 1. Точность Декодировщика вычислялась как среднее число правильно извлеченных бит сообщения M по всему датасету.

Таблица 1

Масштабирование глубины кодирования

Глубина кодирования	Количество эпох	Точность Декодировщика	Количество эффективных встроенных битов
1	35	100	0,99
4	80	0,9978	2,52

Заключение. Таким образом, в статье разработана генеративно-состязательная нейронная сеть, реализующая алгоритм встраивания секретного сообщения в изображение формата RGB. Проведены вычислительные эксперименты, иллюстрирующие работоспособность данной сети, в ходе которых наилучшие результаты получены при сравнительно небольшом (3–4) количестве сверточных слоев и размере датасета (1 500 изображений); установлено, что масштабирование глубины кодирования приводит к увеличению количества эффективно встраиваемой информации, но значительно увеличивает время работы Декодировщика и увеличивает вероятность обнаружения секрета Критиком.

Список литературы

1. Словарь основных терминов по криптологии / Ю. С. Харин [и др.].– Минск : БГУ, 2014. – 92 с.
2. Pevny, T. Steganalysis by subtractive pixel adjacency matrix / T. Pevny, P. Bas, J. Fridrich // Proc. of the 11th ACM Multimedia & Security Workshop. – Princeton, 2009. – P. 75–84.
3. Харин, Ю. С. Распознавание вкраплений в двоичную цепь Маркова / Ю. С. Харин, Е.В. Вечерко // Дискретная математика. – 2015. – Т. 27, В. 3. – С. 123–144.
4. Волошко, В. А. Стеганографическая емкость одномерного марковского контейнера / В. А. Волшоко // Дискретная математика. – 2016. – Т. 28, В. 1. – С. 19–43.
5. Goodfellow, I. J. Generative Adversarial Networks / I. J. Goodfellow [et al/]. – 2014.
6. He, K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 770–778.
7. Kingma, D. Adam: A Method for Stochastic optimization / D. Kingma, J. Ba. – 2014.