**Цель:**

Реализация системы автоматической генерации текстовых описаний (аннотаций) для изображений с использованием:

* Предобученной сверточной сети VGG16 для извлечения признаков изображений
* Рекуррентной сети GRU (Gated Recurrent Unit) для генерации текста
* Набора данных COCO (Common Objects in Context), содержащего изображения с несколькими текстовыми описаниями

**Основные этапы аннотации изображений:**

1. Подготовка данных

a. Загрузка данных COCO:

* Используется набор данных COCO, содержащий ~118k обучающих и ~5k проверочных изображений
* Каждое изображение имеет несколько текстовых описаний (аннотаций)
* Данные очень объемные (19GB обучающих и 816MB проверочных данных)

b. Предварительная обработка изображений:

* Изображения масштабируются и нормализуются
* Пропускаются через предобученную сеть VGG16 для извлечения признаков
* Векторы признаков (размером 4096) сохраняются для последующего использования

c. Обработка текстовых описаний:

* Токенизация текста (разбиение на слова/токены)
* Добавление специальных токенов:
  + «ssss» - начало предложения
  + «eeee» - конец предложения
* Создание словаря токенов
* Преобразование текста в последовательности числовых токенов

2. Архитектура модели

a. Векторизация изображений:

* Используется предобученная сеть VGG16 (без последнего слоя)
* Выход предпоследнего полносвязного слоя (4096 измерений) используется как векторное представление изображения
* Полносвязный слой преобразует 4096-мерный вектор в 512-мерный (для соответствия размерности GRU)

b. Генерация текста:

• Архитектура модели: Используется 3-слойная GRU сеть (Gated Recurrent Unit) с 512 нейронами в каждом слое. Такая глубокая архитектура позволяет модели эффективно обрабатывать последовательности и улавливать сложные языковые зависимости.

• Инициализация состояния: Начальное скрытое состояние всех трех слоев GRU устанавливается равным векторному представлению изображения (512-мерному вектору, полученному из VGG16). Это ключевой механизм, который «загружает» визуальную информацию в языковую модель перед началом генерации.

• Преобразование токенов: Каждый входной токен (слово) сначала преобразуется в плотный вектор фиксированной размерности (обычно 256-512) через слой Embedding. Эти эмбеддинги сохраняют семантические отношения между словами.

• Процесс генерации: На каждом шаге:

* Текущий токен (на первом шаге – специальный токен начала «ssss») проходит через слой Embedding
* Полученный вектор и предыдущее скрытое состояние подаются в GRU
* На выходе GRU получается распределение вероятностей по всему словарю
* Выбирается наиболее вероятное следующее слово (или используется стохастическая выборка)
* Выбранное слово становится входом для следующего шага

• Условия остановки: Генерация последовательности прекращается при:

* Появлении специального токена конца предложения «eeee»
* Достижении заранее заданного максимального количества слов (обычно 20-30)

с. Ключевая особенность: Вектор изображения влияет на генерацию не только через инициализацию, но и косвенно – через сохранение визуального контекста в скрытом состоянии GRU на протяжении всей последовательности.

3. Процесс обучения

a. Подготовка данных для обучения:

* Пары «вектор изображения – последовательность токенов описания»
* Вход модели: вектор изображения + последовательность токенов до текущего слова
* Цель: предсказать следующее слово в последовательности

b. Механизм обучения:

* Используется метод обучения с учителем (teacher forcing)
* Оптимизатор: RMSprop
* Функция потерь: категориальная кросс-энтропия
* Callbacks: ModelCheckpoint (сохранение лучших весов), TensorBoard (визуализация)

c. Особенности:

* Векторы изображений не переобучаются (заморожены)
* Обучаются только параметры GRU и слоя Embedding
* Используется padding для выравнивания длин последовательностей

4. Процесс генерации аннотаций

a. Инференс (генерация новых описаний):

1. Изображение пропускается через VGG16 для получения векторного представления
2. Вектор преобразуется до размерности GRU (512)
3. Начальное состояние GRU инициализируется этим вектором
4. Подается токен начала предложения «ssss»
5. На каждом шаге:
   * Предсказывается распределение вероятностей для следующего токена
   * Выбирается токен с максимальной вероятностью (или используется выборка)
   * Токен добавляется к выходной последовательности
   * Процесс повторяется с новым состоянием GRU
6. Генерация прекращается при получении токена «eeee» или достижении максимальной длины

b. Постобработка:

* Преобразование последовательности токенов обратно в текст
* Удаление служебных токенов («ssss», «eeee»)
* Возможна фильтрация и улучшение сгенерированного текста

5. Визуализация и оценка результатов

a. Пример работы:

* Показаны примеры изображений и их сгенерированные описания
* Сравнение с человеческими аннотациями из набора данных

b. Метрики оценки:

* BLEU (Bilingual Evaluation Understudy) – сравнение с эталонными описаниями
* Perplexity - оценка «уверенности» модели в предсказаниях
* Человеческая оценка качества описаний

**Вывод:**

Такой подход можно использовать в кибербезопасности для автоматического анализа графических угроз - например, для описания фишинговых страниц или подозрительных интерфейсов вредоносного ПО. Модель поможет выявлять скрытые данные в изображениях (стеганографию) и аномалии в графиках сетевого трафика, генерируя понятные текстовые отчеты. Кроме того, ее можно интегрировать в SIEM-системы для автоматического описания визуализаций атак (автоматическое преобразование графиков сетевой активности, логов и других данных в текстовые описания типа), что ускорит упростит работу аналитиков.

25.04.2025