

АЛГОРИТМ КОЛОРИЗАЦИИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ НЕЙРОННЫХ СЕТЕЙ

Камзеев Н.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Калугина М.А. – канд. физ.-мат. наук, доцент

В работе рассмотрен один из подходов к решению задачи колоризации изображений. Описаны некоторые проблемы в этой области, реализована модификация модели с архитектурой на базе генеративно-состязательных нейронных сетей, позволяющая для обучения алгоритма использовать меньший объем данных, чем в базовой модели без потери качества результата.

Одним из наиболее впечатляющих приложений глубокого обучения является колоризация изображений. Колоризация представляет собой преобразование монохромного (черно-белого, сепия) изображения в цветное. Данный процесс применяется, например, для реставрации старых фотографий, киноплёнок и картин. Несколько лет назад задача колоризации требовала довольно много человеческих усилий: на воссоздание цвета старого черно-белого фильма мог уйти ни один год кропотливой работы, но сами идеи раскраски изображений были не очень сложными для понимания. Сейчас весь процесс от начала до конца может быть выполнен гораздо быстрее и эффективнее – с помощью методов искусственного интеллекта. Однако за счет сложного математического аппарата, использующегося в этих методах, алгоритм решения задачи колоризации кажется не таким простым, как при ручной раскраске. Применение математического аппарата обусловлено наличием того факта, что при преобразовании цветного изображения в его базовое одноканальное представление теряется очень много информации (два из трех цветовых измерений), а математический аппарат помогает ее частично восстанавливать, запоминая и затем воспроизводя семантику образов изображений: трава обычно имеет зеленый цвет, а небо – голубой. Таким образом, цель раскрашивания состоит не в том, чтобы восстановить фактический, истинный цвет, а скорее в том, чтобы произвести правдоподобную раскраску, которую пользователь может найти полезной, даже если она не соответствует оригинальному цвету изображения.

На данный момент особый успех в решении задач, связанных с обработкой изображений, имеют модели глубокого обучения. В одних статьях [1] к проблеме колоризации подходят как к задаче классификации и решают ее с помощью сверточных нейросетей (англ. CNN), уделяя внимание неопределенности задачи: автомобиль на изображении может принимать много различных и в то же время допустимых цветов, и без отсутствия дополнительных знаний нельзя отдать предпочтение какому-то определенному цвету. В других же исследованиях [2] подходят к колоризации как к задаче регрессии с некоторыми дополнительными настройками. У каждого подхода есть свои преимущества и недостатки, но в данной работе будет применена еще одна стратегия.

Генеративно-состязательные нейросети (англ. GANs) являются одной из самых интересных тем машинного обучения сегодня. Наиболее популярная сфера их применения – это генерация реалистичных фотографий, но они также дают потрясающие результаты в таких задачах, как изменение изображения с одного домена на другой (например лошадей на зебр) – CycleGAN, создание изображений с высоким разрешением из более низкого разрешения (англ. Super-resolution) – SRGAN, создание изображения по семантической окраске – Pix2Pix, создание изображений по заданному тексту – StackGAN, генерация музыки – GANSynth.

В проведенном исследовании за базовый алгоритм для изучения и последующего улучшения был взят алгоритм Pix2Pix [3]. Этот алгоритм предлагает общее решение для многих задач, связанных с преобразованием изображения (англ. Image-to-image translation), одной из которых как раз и является колоризация. В данном подходе используется функция потерь L1, которая делает задачу колоризации задачей регрессии, и функция потерь GAN, которая помогает решать задачу без размеченных данных. Но прежде, чем реализовывать алгоритм Pix2Pix и пытаться его модифицировать, нужно понимать архитектуру генеративно-состязательных сетей.

Генеративно-состязательная сеть состоит из двух нейросетей: генеративной (англ. Generator), которая пытается моделировать реальное распределение данных, и дискриминативной (англ. Discriminator), которая оценивает вероятность того, что выборка данных на входе данной модели принадлежит подлинному набору данных, а не сгенерированному.

В случае задачи колоризации генератор можно представить как модель, преобразующую одноканальные изображения в оттенках серого в двухканальные изображения с каналами *a (пространство зеленого и красного цвета) и *b (пространство желтого и голубого цвета). Дискриминатор принимает на вход два сгенерированных канала и объединяет их с одноканальным черно-белым изображением, тем самым получая цветное изображение в пространстве L^*a*b (L – отвечает за яркость), а затем оценивает вероятность того, что полученное изображение является

настоящим. Безусловно, дискриминатор нужно обучить на некотором наборе реальных изображений (трехканальные в цветовом пространстве L^*a^*b), которые не были произведены с помощью генератора.

Что касается оптимизационной функции, то она, как упоминалась ранее, состоит из двух функций потерь: cGAN-loss и L1-loss.

Первая помогает генерировать качественные и правдоподобные цветные изображения:

$$L_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_{x,z}[\log(1 - D(x, G(x, z)))], \quad (1)$$

где G – генератор, D – дискриминатор, x – черно-белое изображение, z – шум, подающийся на вход генератору, и y – двухканальное изображение, полученное генератором, или два канала реального изображения.

Вторая же минимизирует среднюю абсолютную ошибку предсказанных цветов:

$$L_{L1}(G) = E_{x,y,z}[|y - G(x, y)|_1]. \quad (2)$$

Если использовать только ее, то модель по-прежнему будет обучаться и решать поставленную задачу, но изображения будут получаться однотонными, с оттенками серого и коричневого.

Поэтому, объединив две функции потерь, можно получить следующую функцию для оптимизации:

$$G^* = \arg \min_G \max_D L_{cGAN}(G, D) + \lambda L_{L1}(G), \quad (3)$$

где λ — это коэффициент для регулирования влияния двух функций потерь на итоговую функцию.

Для получения хороших результатов с помощью данной архитектуры необходимо очень много данных. Если данных будет мало, то в результате окрашенные изображения будут иметь достаточно заметные аномалии. Однако, если обучить генератор на размеченных данных до того, как использовать его в итоговой модели, можно получить удивительные результаты. Предобучение состоит из двух ключевых частей. Первая – это использование уже обученной сети для классификации картинок – ResNet18 например. Вторая – весь генератор обучается, решая задачу колоризации, с функцией потерь L1.

В результате такой модификации для обучения всей состязательной сети было использовано всего 8000 изображений из набора данных COCO. Использование такого небольшого набора данных показывает достаточно хорошие результаты за счет того, что в скелете генератора используются сети, обученные на огромных наборах изображений из ImageNet, а исходный набор данных позволяет подстроить алгоритм под конкретную задачу.

Пример результатов работы итогового алгоритма приведен на рисунке 1.



Рисунок 1 – Результаты колоризации с помощью обученной модели: сверху – черно-белое входное изображение, снизу – сгенерированное цветное

Таким образом, разработанная модификация известного алгоритма Pix2Pix позволяет обучаться на меньшем наборе данных и в то же время дает отличные результаты. Идея модификации заключается в предварительном обучении генератора отдельно, используя для этого размеченные данные. Это позволяет устранить проблему, заключающуюся в том, что в начале обучения ни генератор, ни дискриминатор ничего не знает о задаче. В завершение отметим, что вся работа была реализована на языке программирования Python 3.8.

Список использованных источников:

1. Colorful Image Colorization / Richard Zhang, Phillip Isola, Alexei A. Efros // [arXiv:1603.08511v5](https://arxiv.org/abs/1603.08511v5) [cs.CV], 2016.
2. Old Image De-noising and Auto-Colorization Using Linear Regression, Multilayer Perceptron and Convolutional Neural Network / Junkyo Suh, Koosha Nassiri Nazif, Aravindh Kumar // <http://cs229.stanford.edu/proj2017/final-reports/5244272.pdf>, 2017.
3. Image-to-Image Translation with Conditional Adversarial Networks / Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros // [arXiv:1611.07004v3](https://arxiv.org/abs/1611.07004v3) [cs.CV], 2018.