

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ РАЗЛИЧНЫХ АЛГОРИТМОВ ОБУЧЕНИЯ МНОГОСЛОЙНОГО ПЕРСЕПТРОНА ПРИ КОЛОРИЗАЦИИ ИЗОБРАЖЕНИЙ

© 2016 И.В. Лёзина, А.О. Новиков

Самарский национальный исследовательский университет имени академика С.П. Королёва

Статья поступила в редакцию 11.11.2016

В статье описана разработанная авторами автоматизированная система колоризации изображений в оттенках серого цвета. Разработка системы велась по методологии RAD (Rapid Application Development) – быстрая разработка приложений. Колоризация проводилась многослойным перцептроном, обученным с помощью алгоритма обратного распространения ошибки, алгоритма обратного распространения ошибки с использованием момента, алгоритма обратного распространения ошибки с использованием момента и динамического значения коэффициента обучения. В статье приводятся результаты исследования качества колоризации изображений.

Ключевые слова: автоматизированная система, колоризация изображений, машинное обучение, нейронные сети, искусственный интеллект, многослойный перцептрон, алгоритм обратного распространения ошибки, алгоритм обратного распространения ошибки с использованием момента,

Существует огромное количество полутоновых снимков помимо цветных изображений и фотографий. Новые технологии позволяют колоризировать изображения в оттенках серого цвета. Так, компании «Grading Dimension Pictures» и «Legend Films» в 2009 году колоризовали фильмы «В бой идут одни старики» и «Семнадцать мгновений весны» [1].

Анализ современных программных продуктов показывает, что колоризация изображений часто включает в себя и начинается с сегментации изображения [2] на зоны с неизменным уровнем яркости изображения. В данной работе применяется метод сегментации SWA (Segmentation by Weighted Aggregation), который использует пирамиды взвешенных графов, получаемые из предыдущего графа объединением схожих вершин, так как данный метод имеет сложность $O(n)$, где n – число точек изображения. В ходе алгоритма требуется построение минимальных остовных деревьев, для чего используется алгоритм Краскала, так он является одним из наиболее эффективных алгоритмов данного класса. Также, используется «pathcompression» для ускорения процесса – пикселям устанавливается указатель на представителя своего сегмента.

Для реализации процесса колоризации авторами была выбрана модель многослойного перцептрона. На вход многослойного перцептрона [3] поступает матрица значений яркости целевого пикселя и яркости окружающих его соседей. Количество соседей зависит от радиуса обзора. В случае радиуса обзора равного 1, коли-

чество элементов матрицы будет 9 – пиксель и 8 его соседей вокруг. Также, помимо матрицы, на вход нейронной сети подаются относительные координаты целевого пикселя. Относительные координаты вычисляются с помощью деления абсолютной величины на ширину и высоту изображения соответственно. Эти координаты должны помочь сети в определении пространственного отношения каждого пикселя.

Оптимальное число нейронов в скрытом слое определяется в ходе комплексного анализа сети. Перекрёстное влияние друг на друга оказывают следующие параметры сети: размер входной матрицы, число нейронов в скрытом слое, алгоритм обучения сети и значение коэффициента обучения. В рамках данной работы, многослойный перцептрон обучался при помощи двух градиентных алгоритмов: наискорейшего спуска и наискорейшего спуска с моментом и метода обратного распространения ошибки, а также модификации алгоритма наискорейшего спуска с моментом с адаптивным методом подбора коэффициента обучения [3, 4]. При обучении градиентными алгоритмами ставится задача минимизации целевой функции (формула 1), формируемой, как правило, в виде квадратичной суммы разностей между фактическими и ожидаемыми значениями выходных сигналов:

$$E(w) = \frac{1}{2} \sum_{t=1}^P \sum_{s=1}^M (y_s^{(t)} - d_s^{(t)})^2, \quad (1)$$

где P – количество обучающих примеров, M – число нейронов выходного слоя, $y_s^{(t)}$ – реальное значение, полученное на выходном нейроне s , а $d_s^{(t)}$ – эталонное значение для выходного нейрона s , при подаче на вход сети обучающего примера t .

1) алгоритм наискорейшего спуска находит локальный минимум целевой функции в направлении, обратном направлению вектора гради-

Лёзина Ирина Викторовна, кандидат технических наук, доцент кафедры «Информационные системы и технологии». E-mail: chuchyck@yandex.ru.

Новиков Артём Олегович, студент первого курса магистратуры факультета информатики.

E-mail: tetanovikov@gmail.com

ента, с шагом, равным коэффициенту обучения, значение которого подбирается в процессе обучения. Корректировка весов сети осуществляется по формуле:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot \frac{\partial E(t)}{\partial w_{ij}(t)}, \quad (2)$$

где w_{ij} – вес связи i -го нейрона последующего слоя и j -го нейрона предыдущего слоя сети, η – коэффициент обучения.

алгоритм наискорейшего спуска с моментом позволяет «вытолкнуть» целевую функцию из точки локального минимума и попробовать найти лучшее решение. Коэффициент момента не зависит от градиента целевой функции, а только учитывает изменение весов сети на предыдущей итерации обучения. Коэффициент момента также подбирается в процессе обучения. Корректировка весов сети осуществляется по формуле:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot \frac{\partial E(t)}{\partial w_{ij}(t)} + \alpha(w_{ij}(t) - w_{ij}(t-1)), \quad (3)$$

где α – коэффициент момента.

алгоритм обратного распространения ошибки используется для расчёта компонентов вектора градиента для выходного и скрытого слоёв нейронов сети. При этом сигналы ошибки распространяются от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы.

алгоритм наискорейшего спуска с моментом с адаптивным методом подбора коэффициента обучения. В случае использования этого метода коэффициент обучения будет изменяться в зависимости от полученной ошибки обучения пропорционально величине ошибки. Пользователь задает верхнюю и нижнюю границу коэффициента обучения.

Для вычисления погрешности обучения и тестирования была использована формула вычисления среднего квадратичного отклонения:

$$CKO = \sqrt{\frac{1}{P-1} \cdot \frac{1}{M} \cdot \sum_{t=1}^P \sum_{s=1}^M (y_s^{(t)} - d_s^{(t)})^2}. \quad (4)$$

Так как веб приложения набирают всё большую популярность, не требуют установки, а также веб приложения быстрее и удобнее обновлять, система [5] была реализована в виде веб приложения. Любой пользователь, имеющий доступ к интернету получает доступ к сервису, может загружать свои изображения для колоризации. Таким образом, пользователи освобождаются от необходимости установки приложения на свой персональный компьютер вместе со всеми необходимыми драйверами и программным обеспечением. Серверная реализация системы написана на языке Java в соответствии со спецификацией Java сервлетов [6].

В качестве сервера приложений был выбран Tomcat (в старых версиях – Catalina) – контейнер сервлетов с открытым исходным кодом, разрабатываемый Apache Software Foundation. Tomcat реализует спецификацию сервлетов, спецификацию JSP (Java Server Pages) и JSF (Java Server Faces).

Панель настройки колоризации в веб-приложении приведена на рис. 1.

При обучении вся выборка была разделена на обучающую и тестовую выборки. Качество ее работы, то есть погрешность тестирования, определялось по тестовой выборке на каждой итерации. Обучение останавливалось при начале роста погрешности обучения или достижении максимального количества итераций. Результаты исследований приведены в табл. 1.

Исходя из результатов исследования значений параметров системы, наиболее эффективными параметрами стали: третий порядок входной матрицы, 10 нейронов в скрытом слое, алгоритм наискорейшего спуска с моментом с адаптивным изменением коэффициента обучения с параметрами $h = 0.08$, $\alpha = 0.08$. Используя такие параметры, удалось добиться СКО тестирования 0.00091.

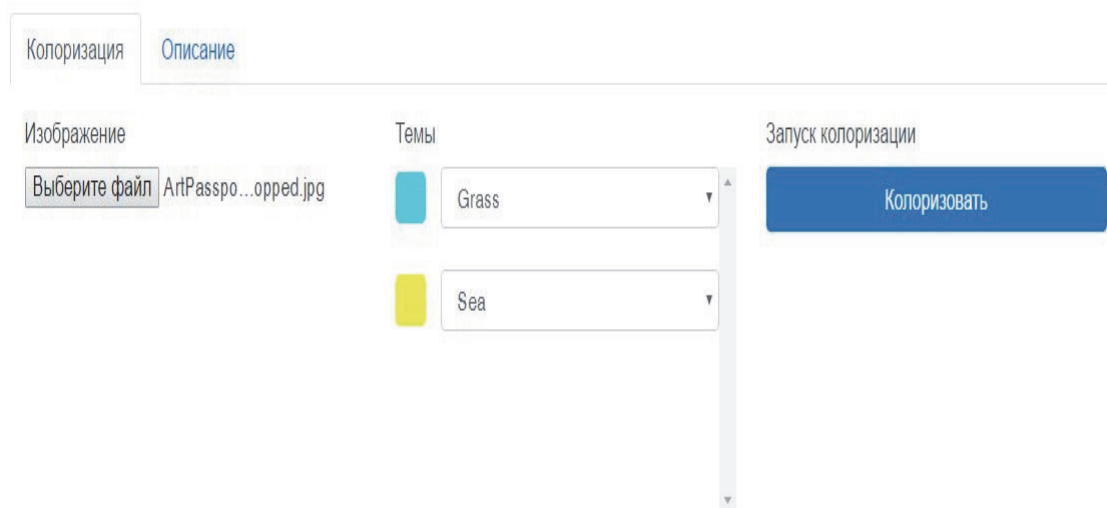


Рис. 1. Панель настройки колоризации

Таблица 1. Зависимость СКО тестирования от параметров сети и алгоритма обучения

Порядок матрицы	Количество нейронов	Алгоритм обучения	Параметры алгоритма обучения	СКО тестирования		
1	2	3	4	5		
3	10	Наискорейшего спуска	$\eta=0.01$	0.00129		
			$\eta = 0.04$	0.00114		
			$\eta=0.08$	0.00233		
		Наискорейшего спуска с моментом	$\eta=0.01, \alpha = 0.02$	0.00105		
			$\eta=0.04, \alpha = 0.04$	0.00101		
			$\eta= 0.08, \alpha = 0.08$	0.00101		
3	10	Наискорейшего спуска с моментом с адаптивным изменением коэффициента обучения	$\eta=0.01, \alpha = 0.02$	0.00096		
			$\eta=0.04, \alpha = 0.04$	0.00095		
			$\eta=0.08, \alpha = 0.08$	0.00091		
3	20	Наискорейшего спуска	$\eta=0.01$	0.00130		
			$\eta = 0.04$	0.00115		
			$\eta=0.08$	0.00234		
		Наискорейшего спуска с моментом	$\eta=0.01, \alpha = 0.02$	0.00106		
			$\eta=0.04, \alpha = 0.04$	0.00107		
			$\eta= 0.08, \alpha = 0.08$	0.00108		
		Наискорейшего спуска с моментом с адаптивным изменением коэффициента обучения	$\eta=0.01, \alpha = 0.02$	0.00097		
			$\eta=0.04, \alpha = 0.04$	0.00096		
			$\eta=0.08, \alpha = 0.08$	0.00093		
		4	10	Наискорейшего спуска	$\eta=0.01$	0.00131
					$\eta = 0.04$	0.00116
					$\eta=0.08$	0.00237
Наискорейшего спуска моментом	$\eta=0.01, \alpha = 0.02$			0.00107		
	$\eta=0.04, \alpha = 0.04$			0.00103		
	$\eta= 0.08, \alpha = 0.08$			0.00104		
Наискорейшего спуска моментом с адаптивным изменением коэффициента обучения	$\eta=0.01, \alpha = 0.02$			0.00097		
	$\eta=0.04, \alpha = 0.04$			0.00098		
	$\eta=0.08, \alpha = 0.08$			0.00099		
4	20			Наискорейшего спуска	$\eta=0.01$	0.00132
					$\eta = 0.04$	0.00117
					$\eta=0.08$	0.00236
		Наискорейшего спуска с моментом	$\eta=0.01, \alpha = 0.02$	0.00109		
			$\eta=0.04, \alpha = 0.04$	0.00107		
			$\eta= 0.08, \alpha = 0.08$	0.00108		
		Наискорейшего спуска моментом с адаптивным изменением коэффициента обучения	$\eta=0.01, \alpha = 0.02$	0.00097		
			$\eta=0.04, \alpha = 0.04$	0.00099		
			$\eta=0.08, \alpha = 0.08$	0.00098		

СПИСОК ЛИТЕРАТУРЫ

1. Википедия. Сегментация [Электронный ресурс]. URL: [https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%B3%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F_\(%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9\)\)](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%B3%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9))) (дата обращения 05.09.2016).
2. Компьютерная Графика и Мультимедиа. Методы сегментации изображений: автоматическая сегментация [Электронный ресурс]. URL: <http://cgm.computergraphics.ru/content/view/147> (дата обращения 05.09.2016).
3. Хайкин С. Нейронные сети: Полный курс [пер. с

- англ.]. 2-е изд. М.: Вильямс, 2006. 1104 с.
4. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с.
 5. Лёзина И.В., Новиков А.О. Колоризация изображений с помощью нейронных сетей // Перспективные информационные технологии (ПИТ 2015), том 1: труды Международной научно-технической конференции [под ред. С.А. Прохорова]. Самара: издательство Самарского научного центра РАН, 2015. С. 88-90.
 6. Хорстман К.С., Корнелл Г. Java 2 для профессионалов. М.: Вильямс, 2003. 848 с.

RESEARCHING A LEARNING ALGORITHMS EFFICIENCY FOR COLORIZATION BY USING A MULTILAYER PERCEPTRON

© 2016 I.V. Lyozina, A.O. Novikov

Samara National Research University named after Academician S.P. Korolyov

The article describes the software for grayscale images colorization by using a multilayer perceptron. The system has been developed by RAD (Rapid Application Development) methodology. The colorization is performed by multilayer perceptron learned by the back-propagation algorithm, the back-propagation algorithm with a moment, the back-propagation algorithm with a moment and dynamic learning coefficient. The article presents the results of the research representing the quality of image colorization. *Keywords:* automated system, image colorization, machine learning, neural networks, artificial intelligence, multilayer perceptron, back-propagation algorithm, back-propagation algorithm with a moment, back-propagation algorithm with a moment and dynamic learning coefficient.