ЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ ОБУЧЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ В РАМКАХ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ ЛИЦ

HEURISTIC ALGORITHMS FOR TRAINING CONVENTIONAL NEURAL NETWORKS IN THE FRAMEWORK OF FACE RECOGNITION TECHNOLOGY

УДК 004.932.2

Булыга Филипп Сергеевич, магистрант, Южный федеральный университет, г. Таганрог, e-mail: bulyga@sfedu.ru

Bulyga Philip Sergeevich, Master student, Southern Federal University, Taganrog e-mail: bulyga@sfedu.ru

Аннотация

В данной статье рассматриваются различные подходы к обучению сверточных нейронных сетей, в частности эвристические алгоритмы обучения. Также в статье приведены результаты исследования, связанного с выявлением наиболее быстрого алгоритма обучения сверточной нейронной сети при решении прикладных задач обнаружения лиц. В ходе данных исследований рассматривались такие характеристики процесса обучения нейронной сети как количество эпох обучения, скорость, а также виды ошибок, возникающих при обучении. В данной статье обучающим набором изображений для сверточных нейронных сетей служила база лиц FERET.

Annotation

This article discusses various approaches to training convolutional neural networks heuristic learning algorithms. The article also presents the results of a study

related to identifying the fastest learning algorithm for a convolutional neural network when solving applied problems of face detection. During these studies, such characteristics of the learning process of a neural network as the number of learning epochs, speed, as well as the types of errors that occur during training were considered. In this article, the FERET face database was used as a training set of images for convolutional neural networks.

Ключевые слова: искусственный интеллект, распознавание образов, распознавание лиц, нейронные сети, сверточные нейронные сети, алгоритмы обучения, эвристические алгоритмы обучения.

Keywords: artificial intelligence, pattern recognition, face recognition, neural networks, convolutional neural networks, learning algorithms, heuristic learning algorithms.

Введение

В случае возникновения необходимости решения прикладных задач детектирования объектов на изображении или фрагменте видеоряда, одним из наиболее эффективных подходов можно считать применение нейронных сетей различных архитектур. Классические архитектуры нейронных сетей при решении подобных задач не являются оптимальными по ряду причин:

1) Размерность нейронной сети напрямую зависит от размерности сигнала, поступающего на вход;
2) Возрастание количества ценных признаков влечет за собой увеличение размерности нейронной сети, как следствие увеличение вычислительной сложности, а также времени необходимого на обучение;
3) Игнорирование топологии входного сигнала/изображения;

В данной статье рассмотрены сверточные нейронные сети (СНС), предложенные в 1988 году французским ученым Яном Лекуном, в качестве альтернативы стандартным архитектурам нейронных сетей. Предложенная архитектура искусственной нейронной сети обладает рядом преимуществ:

- Возможность формирования признаков высокого уровня, основываясь на признаках низкого уровня в пределах одно класса;
- Временная эффективность в сравнении с классическими архитектурами;
- Эффективное извлечение отдельных элементов на входном сигнале/изображении;

Главная цель данной статьи – представить сравнительный анализ эффективности алгоритмов обучения сверточной нейронный сети при решении задач обнаружения лиц. Данное исследование включает в себя анализ скорости обучения, количество эпох обучения, а также показатели точности обнаружения.

Алгоритмы обучения

Алгоритм обучения «Back Propagation». Данный алгоритм впервые был предложен в 1974 году группой ученных П. Дж. Вербосом и А. И. Галушкиным. Впоследствии алгоритм «Back Propagation» был модернизирован в 1986 году ученым Д. И. Румельхартом. Основная идея данного алгоритма заключается в сведении к минимуму эмпирической ошибки обучения.

Функция ошибки – параметр, определяемый разницей значения получаемого в процессе обучения и значения желаемого результата. Процесс обучения – корректировка весовых коэффициентов таким образом, чтобы значение эмпирической ошибки было меньше заданного порога.

Значение функции ошибки определяется в соответствии с формулой (1.1):

$$P_q = \frac{1}{2} \times \sum_{i=0}^{M} (d_{qi} - x_{qi})^2 \qquad (1.1)$$

где, P_q – значение функции ошибки q-го образа; d_{qi} – желаемое выходной значение i-го нейрона q-го образа; x_{qi} – текущее выходное значение i-го нейрона q-го образа, полученное в ходе обучения.

В свою очередь корректирование весовых коэффициентов проводится по формуле (1.2):

$$\Delta\omega_{ji} = -\gamma \times \vartheta_{qi} \times x_{qi} \qquad (1.2)$$

где, ϑ_{qi} – величина ошибки i-го нейрона q-го образа; x_{qi} – текущее выходное значение i-го нейрона q-го образа; γ – коэффициент времени, затраченного на обучение, $\gamma \in (0; 1)$.

Величина ошибки нейронной сети определяется в соответствии с формулой (1.3):

$$\vartheta_j^{(q)} = (\mathrm{f}_j^{(q)}(S))` \sum_i \omega_{ji}\vartheta_i^{(q+1)} \quad (1.3)$$

где, $\vartheta_j^{(q)}$ – величина ошибки j-го нейрона q-го слоя; $\vartheta_i^{(q+1)}$– величина ошибки i-го нейрона $(q+1)$-го слоя; ω_{ji} – весовой коэффициент связи, соединяющий два нейрона; $(\mathrm{f}_j^{(q)}(S))`$ – значение итоговой производной функции активации j-го нейрона q-го слоя.

К недостатком данного подхода обучения можно отнести малый шаг корректировки весовых коэффициентов, что в совокупности увеличивает время, затрачиваемое на процесс обучения. При этом порождая новую задачу подбора оптимального шага корректировки.

Алгоритм обучения «Quick Propagation». Алгоритм «Quick Propagation» относится к группе эвристических алгоритмов обучения нейронных сетей. Впервые данный алгоритм был предложен Г. Брауном, М. Ридмиллером и С. Фалманом. «Quick Propagation» – модификация предыдущего алгоритма «Back Propagation», базирующегося на классическом алгоритме градиентного спуска.

Основное отличие данного алгоритма заключается в анализе знака частной переменной, а не ее значения, в свою очередь вариация шага корректировки весовых коэффициентов является индивидуальной и преобразовывается в течение всего процесса обучения.

Обучение сети при помощи «Quick Propagation» происходит благодаря вычислению вектора-градиента, а величины весов преобразовываются при этом в направлении антиградиента. Вычисление вектора-градиента производится по формуле (2.1):

(2.1)

$$s_n = \frac{\alpha\left(\omega_n + \Delta\omega_n^{(q-1)}\right) - \alpha\left(\omega_n^{(q)}\right)}{\Delta\omega_n^{(q-1)}}$$

где, s – n-й компонент вектора-градиента; ω – значение весового коэффициента;

После завершения этапа вычисления вектора-градиента следует этап преобразования величин весовых коэффициентов, вычисляемый в соответствии с формулой (2.2):

$$\omega_n^{(q)} = -\beta(s_n^{(q-1)} + \varphi_\omega \omega_n^{(q-1)}) + \theta_n^{(q)} \Delta\omega_n^{(q-1)} \quad (2.2)$$

где, $s_n^{(q)}$ – величина значения n-го коэффициента полученного на q-й итерации; s_n – компонент вектора-градиента; β – множитель обучения; θ_n – множитель фактора момента; φ_ω – параметр уменьшения величины весового коэффициента.

Отличительная особенность алгоритма «Quick Propagation» заключается в наличие двух дополнительных параметров: параметр сокращения величины весового коэффициента (зачастую данный параметр равен 10^{-4}) и параметр множителя момента, предназначенного для адаптирования алгоритма обучения.

Каждый весовой коэффициент обладает индивидуальным множителем момента, рассчитываемым по формуле (2.3):

$$\theta_n^{(q)} = \frac{s_n^{(q-1)}}{s_n^{(q-2)} - s_n^{(q-1)}} \quad (2.3)$$

После этого множитель момент определяется минимальной величиной из $\widehat{\theta_n}$ и q_{max}.

Алгоритм обучения «Resilient Propagation». Resilient Propagation – алгоритм обучения предложенный в 1992 году исследователями Г. Брауном и М. Ридмиллером. Для данного алгоритма, так же, как и для «Quick Propagation» важен знак производной, а нее ее значение.

Вычисление значения преобразования весов происходит в соответствии с формулой (3.1):

(3.1)

$$\Delta_{n,m}^{(r)} = \begin{cases} \eta^{+} \times \Delta_{n,m}^{(r)}, \text{если } \dfrac{\partial E^{(r-1)}}{\partial \omega_{n,m}} \times \dfrac{\partial E^{(r)}}{\partial \omega_{n,m}} > 0 \\ \eta^{-} \times \Delta_{n,m}^{(r)}, \text{если } \dfrac{\partial E^{(r-1)}}{\partial \omega_{n,m}} \times \dfrac{\partial E^{(r)}}{\partial \omega_{n,m}} < 0, \\ \Delta_{n,m}^{(r)}, \text{если } \dfrac{\partial E^{(r-1)}}{\partial \omega_{n,m}} \times \dfrac{\partial E^{(r)}}{\partial \omega_{n,m}} = 0 \end{cases}$$

при условии: $0 < \eta^{-} < 1 < \eta^{+}$

Если знак частной производной весового коэффициента изменится на противоположный, это является следствием слишком большого шага корректировки коэффициентов, при котором алгоритм пропустил локальный минимум. В случае возникновения подобной ситуации необходимо снизить величину параметра преобразования на η и вернуться к предыдущим значениям весовых коэффициентов (3.2):

$$\Delta\omega_{n,m}^{(r)} = \Delta\omega_{n,m}^{(r)} - \Delta_{n,m}^{(r-1)}, \quad \text{если } \frac{\partial E^{(r-1)}}{\partial \omega_{n,m}} \times \frac{\partial E^{(r)}}{\partial \omega_{n,m}} < 0 \tag{3.2}$$

Если знак частной производной остался неизменным, необходимо увеличить параметр преобразования на η^{+}, тем самым ускорив сходимость (3.3):

$$\Delta\omega_{n,m}^{(r)} = \begin{cases} -\Delta_{n,m}^{(r)}, \text{если } \dfrac{\partial E^{(r)}}{\partial \omega_{n,m}} > 0 \\ +\Delta_{n,m}^{(r)}, \text{если } \dfrac{\partial E^{(r)}}{\partial \omega_{n,m}} < 0 \\ 0, \text{если } \dfrac{\partial E^{(r)}}{\partial \omega_{n,m}} = 0 \end{cases} \tag{3.3}$$

В случае если величина итоговой производной отрицательна, весовой коэффициент при этом будет увеличиваться, иначе – уменьшаться. Затем будет осуществляться корректировка весовых коэффициентов.

Генетический алгоритм обучения. Генетические алгоритмы относятся к группе эвристических алгоритмов, главная задача которых заключается в решении задач оптимизации. Впервые, популярность генетическим алгоритмам пришла с публикацией трудов Д. Г. Холланда и К. Д. Джонга в 1970-х годах.

Данные алгоритмы основаны на поэтапной обработке множества альтернативных решения, при помощи методов случайной комбинации и отбора необходимых параметров. Принцип работы алгоритма представляет из себя аналогию механизмов естественного отбора.

Параметром оптимизации для данного типа алгоритмов выступает среднеквадратичная ошибка искусственной нейронной сети на обучающем наборе данных.

Первый этап работы алгоритма – популяция. На данном этапе происходит генерация начальной популяции особей. Каждая особь представляет из себя матрицу весовых коэффициентов нейронной сети (ω), элементы которой задаются при помощи генерации случайных чисел в заданном интервале значений.

Второй этап работы алгоритма – селекция. На данном этапе из сформированной популяции размерности M извлекается некоторое множество особей N. Затем из множества особей N формируется подмножество T экземплярами которого выступают наиболее приспособленные особи. Оценка уровня приспособленности особи определяется в соответствии с правилом (4.1):

$$f_i = f(X_i), \tag{4.1}$$

где, $X_i = \{x_{iq} : q = 1,2, \ldots, M\}$ – хромосома i-й особи; x_{iq} – значение генов текущей особи.

Вероятностное значение скрещивания i-ой особи рассчитывается в соответствии с формулой (4.2):

$$p_i = \frac{f_i}{\sum_j f_j}, \tag{4.2}$$

где, p_i – вероятностное значение скрещивания i-ой особи; f_i – величина приспособленности i-ой особи.

Третий этап работы алгоритма – скрещивание. На данном этапе определяются точки, в которых происходит деление хромосомы на некоторое количество фрагментов с последующим обменом фрагментами между особями. Процесс скрещивания определяется в соответствии с условием (4.3):

(4.3)

$$k_q = \lambda h_q + (1 - \lambda) h_q$$

$$\text{при условии } 0 \leq \lambda \leq 1$$

где, h_q – q-й ген родительской особи; k_q – q-й ген потомка.

Четвертый этап работы алгоритма – мутация. Данный процесс позволяет вносить случайные преобразования в хромосомы особей, тем самым предупреждая попадание популяции в локальный экстремум. Процесс мутации с некоторой долей вероятности преобразовывает значение гена, принадлежащего хромосоме на противоположное. Таким образом формируется новая популяция M_2.

По завершению мутации происходит вычисление величины функции приспособленности для каждой хромосомы принадлежащей популяции M_2. Условием завершения алгоритма может выступать преодоление порогового значения функции ошибки, либо преодоление порогового значения по заданному количеству эпох.

Применение генетических алгоритмов при обучении искусственных нейронных сетей позволяет избежать ситуации попадания в локальные минимумы функции.

Алгоритм обучения Левенберга-Марквардта. Данный алгоритм был разработан в 1944 году исследователем К. Левенбергом, в последствии был оптимизирован в 1963 году другим ученым Д. Марквардтом. Предложенный алгоритм является комбинацией двух методов: Гаусса-Ньютона и метода градиентного спуска, и выступает в качество одного из способов оптимизации параметров нелинейных регрессионных моделей. Данный алгоритм вобрал в себя лучшие качества из комбинируемых методов: скорость сходимости (метод Гаусса-Ньютона) и стабильный уровень работы (метод градиентного спуска).

Главная особенность рассматриваемого алгоритма заключается в том, что алгоритм в своей работе применяет матрицы, сформированные на основании вторых частных производных функций (матрицы Гессе). Данные матрицы позволяют описывать поведение функции во втором порядке. Оптимизируемым

параметром для данного алгоритма также выступает среднеквадратичная ошибка искусственной нейронной сети на обучающем наборе данных.

Оценивание величины значения $\Delta\omega$ происходит благодаря линейному приближению функции (5.1):

$$f(\omega + \Delta\omega, x) \approx f(\omega, x) + Jac\Delta\omega, \quad (5.1)$$

где, ω – весовые коэффициенты, преображенные в векторную форму; Jac – якобиан функции $f(\omega, x_m)$ в точке ω;

Матрица Якоби в общем виде можно представить следующим образом (5.2), (5.3):

$$Jac = \begin{bmatrix} \widetilde{Jac_1} \\ \dots \\ \widetilde{Jac_M} \end{bmatrix}, \quad (5.2)$$

$$\widetilde{Jac_q} = \begin{bmatrix} \frac{\partial f(\omega, x_1)}{\partial \omega_1} & \frac{\partial f(\omega, x_1)}{\partial \omega_2} & \dots & \frac{\partial f(\omega, x_1)}{\partial \omega_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f(\omega, x_M)}{\partial \omega_1} & \frac{\partial f(\omega, x_M)}{\partial \omega_2} & \dots & \frac{\partial f(\omega, x_M)}{\partial \omega_N} \end{bmatrix}, \quad (5.3)$$

Для нахождения величины значения $\Delta\omega$ необходимо решить системы линейных уравнений (5.4):

$$\Delta\omega = (Jat^Q \times Jat)^{-1} \times Jat^Q \times (y - f(\omega)), \quad (5.4)$$

Поскольку численное значение обусловленности матрицы $Jat^Q * Jat$ является квадратом числового значения обусловленности матрицы *Jat*, то матрица $Jat^Q * Jat$ может получиться существенно вырожденной. Исходя из этого, необходимо ввести новый регулирующий параметр $\varphi \geq 0$ (5.5):

$$\Delta\omega = (Jat^Q \times Jat + \varphi \times I)^{-1} \times Jat^Q \times (y - f(\omega)), \quad (5.5)$$

где, I – единичная матрица.

В свою очередь, регулирующий параметр φ назначается каждый раз с наступлением новой итерации. В случае выбора оптимального значения параметра φ возможно достичь монотонного убывания функции минимизации. В случае если значение ошибки E_q убывает слишком быстро, малая величина значения φ сводит данный алгоритм к простому алгоритму Гаусса-Ньютона.

Если же параметр φ будет иметь большую величину, то уровень влияния аппроксимированной матрицы Jat^Q * Jat практически будет сведено к нулю. Благодарю таким действиям, появляется возможность обеспечения увеличения величины шага вдоль пологих участков функции, а также уменьшение величины шага вдоль крутых спусков.

Алгоритм прекращает свою работу в трех случаях: вектор, сформированный на основе весовых коэффициентов сводит ошибку к минимальному значения; достигнуто пороговое значение количество эпох обучения нейронной сети; приращение $\Delta\omega$ в последующей итерации будет меньше порогового значения. Значение ω полученное на последней итерации считается искомым.

Результаты исследований алгоритмов обучения

В данной статье рассматривается влияние различных алгоритмов обучения на скорость обучение, а также точность обнаружения искомых объектов на изображении. Для тестирования алгоритмов была реализована сверточная нейронная сеть, состоящая из 7 слоев: один входной и выходной слой, два сверточных, два подвыборочных и один обычный слой. Условием завершения обучения алгоритма выступало несущественное изменение ошибки, преодолевшей минимальный порог в течение пяти эпох.

В качестве выборки обучения применялась база изображений лиц размерности 10500 экземпляров. Также выборка включает в себя изображения, содержащие сложный задний фон. Результаты обучения сверточной нейронной сети различными алгоритмами представлены на рисунках 1 – 4.

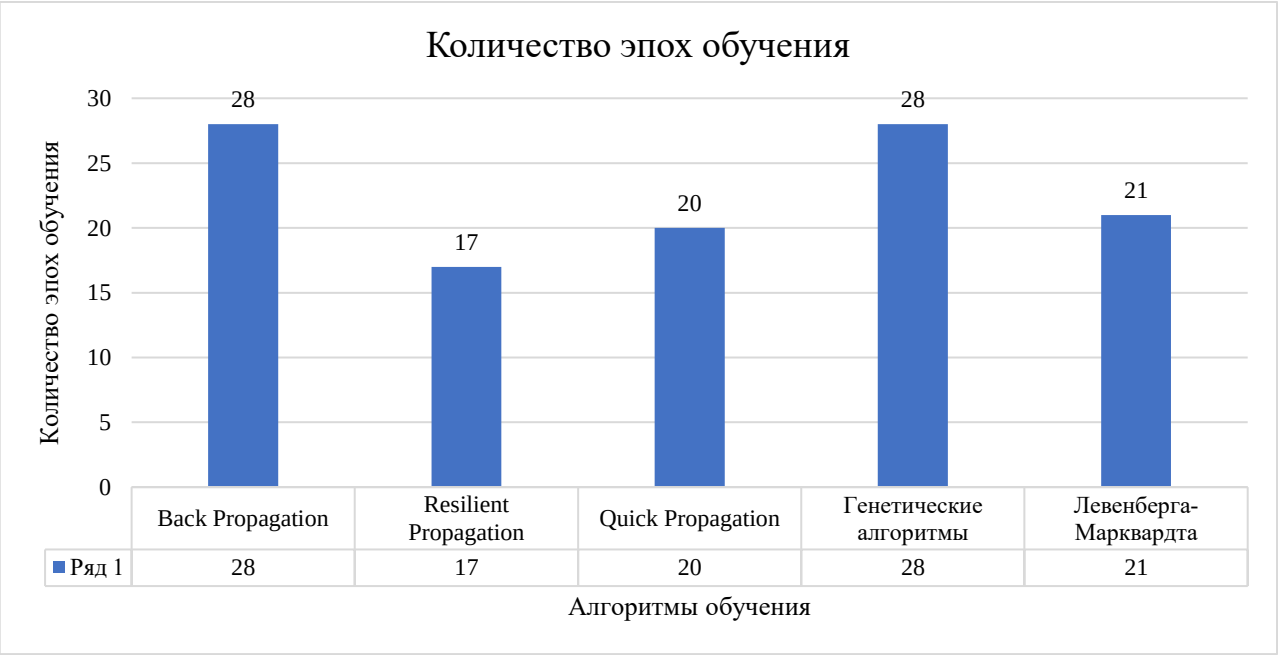

Рисунок 1 – Сравнительная диаграмма количества эпох обучения.

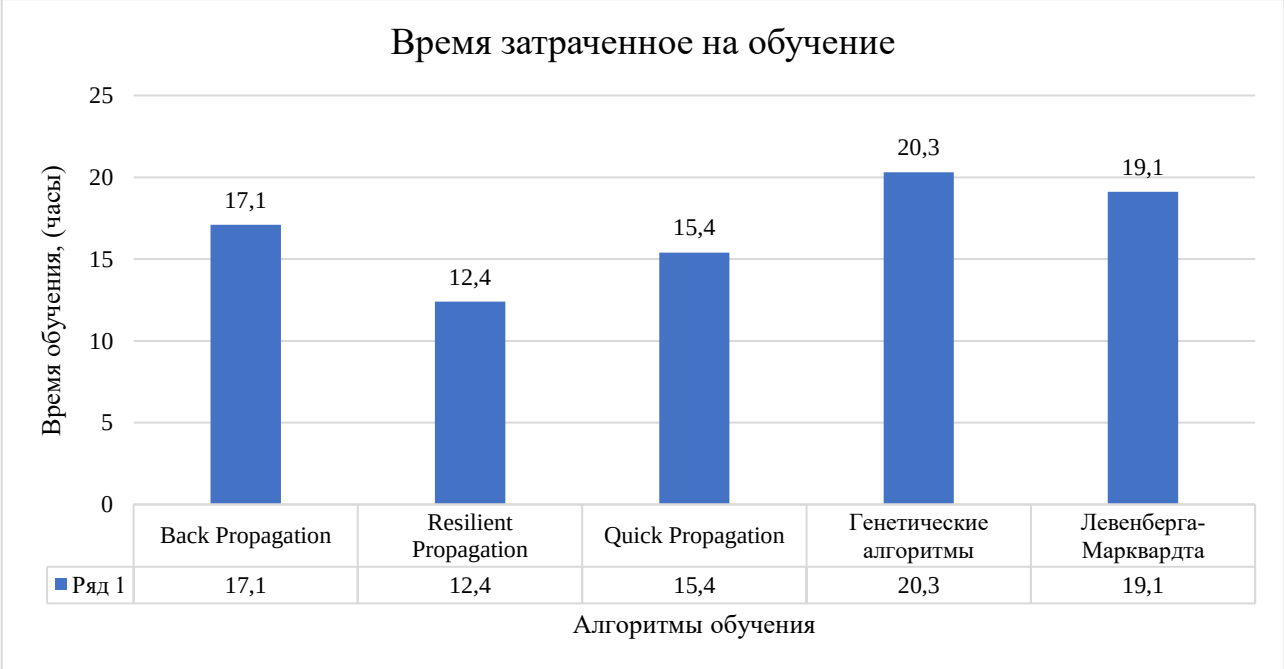

Рисунок 2 – Сравнительная диаграмма времени обучения алгоритмов.

Комментируя результаты касающиеся времени обучения алгоритмов, а также количество эпох, полученных в ходе обучения, стоит отметить два алгоритма показавшие наилучшие результаты: «Quick Propagation» – 20 эпох, 15,4 часов; «Resilient Propagation» – 17 эпох, 12,4 часа.

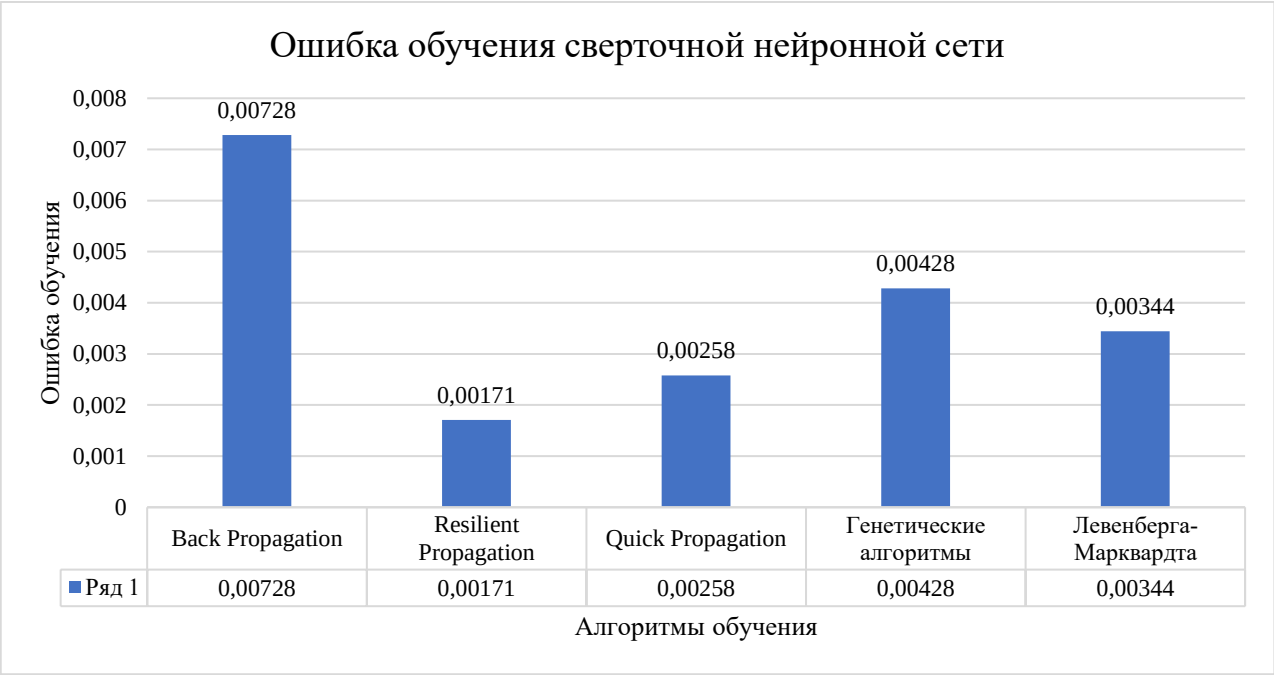

Рисунок 3 – Ошибка обучения сверточной нейронной сети.

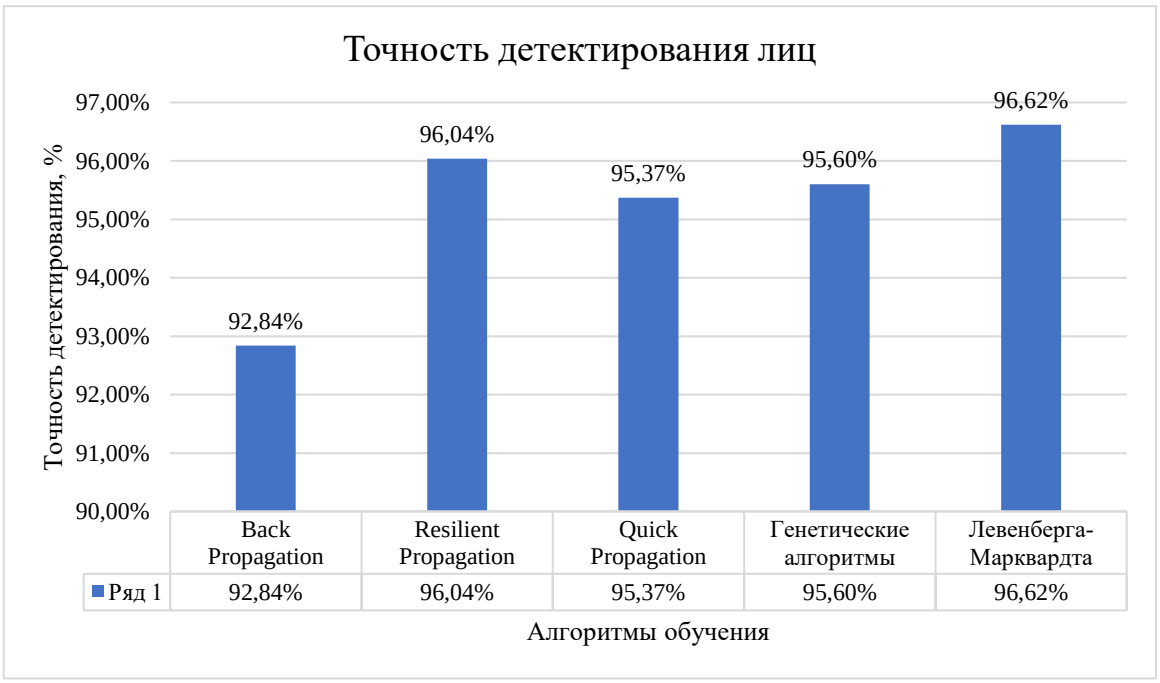

Рисунок 4 – Точность детектирования лиц.

На рисунках 3 – 4 представлены результаты, полученные в ходе проведения исследований по следующим категориям: значение ошибки в ходе обучения и показатели точности обнаружения лиц. По первому показателю наилучшие результаты были получены методами: «Quick Propagation» – 0,00258; «Resilient

Propagation» – 0,00171. По второму показателю: «Resilient Propagation» – 96,04% «Левенберга-Марквардта» – 96,62%.

Заключение

В данной статье приведен краткий обзор методов обучения нейронных сетей, рассмотрены классические методы обучения (метод обратной ошибки), а также рассмотрены эвристические алгоритмы обучения. Проведены исследования позволяющие оценить скорость обучения, значение ошибки обучения, а также показатель точности обнаружения искомых объектов. По итогам данного исследования можно сделать заключение, что в сравнении с классическим подходом обучения, любой из вышеперечисленных алгоритмов показывают преимущественные результаты.

Также, стоит оговориться, что показатели скорости обучения, могут разниться в зависимости от конфигурации вычислительных машин. В данном случаи все исследования проводились на единой конфигурации, в идентичных условиях.

Литература

1. Азарнова Т. В., Баркалов С. А., Полухин П. В. Разработка гибридного алгоритма обучения структуры динамической байесовской сети на основе метода Левенберга-Марквардта // Вестник ЮУрГУ. Серия: Компьютерные технологии, управление, радиоэлектроника. 2018. № 4. С. 16-24.
2. Бредихин А. И. Алгоритмы обучения сверточных нейронных сетей // Вестник ЮГУ. 2019. № 1. С. 41-54.
3. Каплунов Т. Г. Применение генетических алгоритмов для обучения нейронной сети // Вестник Таганрогского института имени А.П. Чехова. 2015. № 1. С. 63-67.
4. Пшеничкин Е. В., Цуканов М. В., Рыженков Д. В. Распознавание рукописных символов с помощью нейронных сетей методом с обратным распространением ошибки // Инновационная наука. 2018. № 2. С. 14–16.

5. Analysis resilient algorithm on artificial neural network backpropagation / W. Saputra, M. Zarlis, R. W. Sembiring, D. Hartama // Journal of Physics: Conferemce Series. Vol. 930, 2017. Art. no. 012035.

Literature

1. Azarnova TV, Barkalov SA, Polukhin PV Development of a hybrid algorithm for learning the structure of a dynamic Bayesian network based on the Levenberg-Marquardt method // Bulletin of SUSU. Series: Computer technology, control, electronics. 2018.No. 4.P. 16-24.
2. Bredikhin AI Learning algorithms for convolutional neural networks. Vestnik YuGU. 2019.No. 1.P. 41-54.
3. Kaplunov TG Application of genetic algorithms for training a neural network // Bulletin of the Taganrog Institute named after A.P. Chekhov. 2015. No. 1. S. 63-67.
4. Pshenichkin EV, Tsukanov MV, Ryzhenkov DV Recognition of handwritten characters using neural networks using the backpropagation method // Innovative Science. 2018. No. 2. P. 14-16.
5. Analysis resilient algorithm on artificial neural network backpropagation / W. Saputra, M. Zarlis, R. W. Sembiring, D. Hartama // Journal of Physics: Conferemce Series. Vol. 930, 2017. Art. no. 012035.