

УДК 004.8:623

А.В. ЕРМОЛЕНКО, кандидат технических наук

В.М. ПОЛУШКИН, кандидат технических наук

П.С. ВОРОНЦОВ

АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ОБУЧЕНИЯ СЕТИ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ В ЗАДАЧАХ ДЕТЕРМИНАЦИИ ГРАФИЧЕСКИХ ОБЪЕКТОВ

Статья посвящена исследованию методов детерминации объекта на графическом изображении посредством совершенствования алгоритмов обучения сетей обратного распространения ошибки. Кроме того, рассматриваются вопросы обработки изображения разностными фильтрами Прьюита, Собеля и др. для подачи данных на вход нейросети.

Ключевые слова: нейросети; распознавание образов; детерминация объекта; кластеризация; обратное распространение; градиентные алгоритмы.

Введение

Современный уровень развития информационных технологий позволяет обеспечить новый виток развития робототехнических комплексов военного назначения. Среди многообразия задач, решаемых современными системами технического зрения, важными представляются анализ объектов графической сцены, определение сегментов с заданными параметрами априорно известных классифицирующих признаков [1]. Реализация оперативного и качественного поиска объекта позволяет обеспечить поддержку принятия решений на всех уровнях управления в различных условиях оперативной обстановки, повысить скорость автоматического обнаружения, распознавания, классификации и ранжирования целей [2].

Применение нейронных сетей для решения задач классификации позволяет избежать сложных и громоздких расчетов при оперировании большими объемами многомерных данных, минимизировать размеры решающего устройства бортовой информационной управляющей системы образца военной техники. Необходимость поиска оптимального решения в условиях имеющихся ограничений требует определить подходящий тип сети и провести ее обучение с использованием динамически меняющегося набора обучающих данных. Не менее актуальным, в контексте оптимизации алгоритмической сложности, представляется снижение инвариантности входного потока данных [3].

1. Контурная предобработка изображения для формирования входного вектора нейросети

Методы распознавания и анализа образов могут быть применены к поиску определенного класса технических объектов с известными свойствами на сцене. Известной задачей систем технического зрения является идентификация колеса (рисунок 1) на динамическом («живом») кадре изображения.



Рисунок 1 – Детерминация объекта на «живом» кадре

Детерминация колеса на «живом» кадре представляет собой задачу поиска эллипса [4] с произвольными значениями радиусов и углов поворота относительно камеры (стереопары). Образ объекта в реальной двумерной (трехмерной) сцене сильно зашумлен, что обуславливает необходимость обращаться к алгоритмам слабо чувствительным к помехам. Это предполагает необходимость проведения предобработки изображения для последующей классификации.

Решение задачи выделения границ объекта состоит из последовательности операций (рисунок 2), основными из которых являются: сканирование сцены, прослеживание линии контура в условиях воздействия фоновых шумов, формирование кода контура и предварительная обработка контура.



Рисунок 2 – Определение контура объектов

Фундаментальным этапом распознавания образа на двумерной проекции является решение задачи выделения контуров. Эффективными на современном этапе развития информационных технологий представляются методы контурного анализа. В основе контурной фильтрации лежит оценка интенсивности пикселей, расположенных в некоторой дельта-окрестности искомого контура объекта. Основным механизмом контурного анализа изображений являются разностные фильтры [5]. Разностные графические фильтры представляют собой линейные фильтры, задаваемые дискретными аппроксимациями дифференциальных операторов (по методу конечных разностей).

Распространенными дифференциальными операторами для дискретных изображений являются фильтры Прьюита (Prewitt) и Собеля (Sobel)¹:

$$M_1^{prewitt} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, M_1^{sobel} = \frac{1}{4} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

с вычислением нормы градиента по трем смежным точкам:

$$G(x, y) = \sqrt{(A_{x+1,y} - A_{x,y})^2 + (A_{x,y+1} - A_{x,y})^2}. \quad (1)$$

В фильтре Собеля используется восемь отсчетов яркости в окрестностях центральной точки:

$$G(x, y) = \sqrt{(Gx_{x,y})^2 + (Gy_{x,y})^2}, G(x, y) \cong |Gx_{x,y} + Gy_{x,y}|. \quad (2)$$

Наряду с более точным определением нормы градиента алгоритм Собеля позволяет определять и направление вектора градиента в плоскости анализа изображения в виде угла между вектором градиента и направлением строк матрицы. В отличие от сглаживающих и контрастоповышающих фильтров, не меняющих среднюю интенсивность изображения, в результате применения разностных операторов получается, как правило, изображение со средним значением пикселя близким к нулю. Вертикальным перепадам (границам) исходного изображения соответствуют пиксели с большими по модулю значениями на результирующем изображении. Поэтому разностные фильтры являются также фильтрами выделения границы объектов.

Альтернативным к фильтрам Собеля и Прьюита представляется вычисление нормы градиента по четырем смежным точкам [6] (оператор Робертса):

$$G(x, y) = \sqrt{(A_{x+1,y+1} - A_{x,y})^2 + (A_{x+1,y} - A_{x,y+1})^2}. \quad (3)$$

Результаты компьютерного моделирования функционирования фильтров Собеля, Лапласа-Гаусса, Робертса, Канни, Робертса и Прьюита в условиях рассматриваемой задачи представлены на рисунке 3.

Проведенное компьютерное моделирование функционирования данных фильтров в среде Matlab подчеркивает более эффективные результаты контурной предобработки фильтрами Прьюита и Собеля. Именно данные фильтры предлагаются нами для последующего формирования входной признаковой базы нейронной сети.

¹ Горелик А.Л., Скрипкин В.А. Методы распознавания: учеб. пособие. 4-е изд. М.: Высшая школа, 2004. – 262 с.



оригинальный образ



фильтр Прьюита



фильтр Канни



фильтр Лапласа-Гаусса



фильтр Робертса



фильтр Собеля

Рисунок 3 – Предобработка изображения различными методами контурной фильтрации

2. Варианты обучения сети обратного распространения ошибки и эффективность их применения

В реальных условиях оперативный интерес представляет конкретный объект сцены, сведения о квалифицирующих признаках которого известны априорно. Таким образом, при выборе нейросетевой архитектуры возможно остановиться на сетях, для которых применяется алгоритм обучения с учителем, в частности, на сетях обратного распространения ошибки (*Back Propagation – BP*). Наличие обратной связи в процессе обучения может существенно повысить качество подстройки весовых коэффициентов синоптических связей.

Вне зависимости от модификации, методы обучения *Back Propagation* основаны на использовании градиента функции как основы для направления спуска. Наиболее распространенным алгоритмом определения направления обучения является наискорейший спуск (*gd*-алгоритм). Алгоритм предполагает последовательность итераций: решение текущего шага – вычисление градиента – коррекция и инкрементирование шага. При обучении сети на каждый нейрон первого слоя подаются все компоненты входного вектора признаковой базы. Далее все выходы скрытого слоя подаются на следующий слой, образуя полносвязную сеть. Минимизация ошибки сети достигается методом наименьших квадратов:

$$E(W) = \frac{1}{2} \sum_{j=1}^p (y_j - d_j)^2, \quad (4)$$

где y_j – значение j -го выхода нейросети; d_j – желаемое значение j -го выхода; p – число нейронов в выходном слое.

В ходе определения дальнейшего направления спуска на каждой итерации изменение веса производится по формуле:

$$\Delta w_{ij} = -h \frac{\partial E}{\partial w_{ij}}, \quad (5)$$

где h – параметр, определяющий скорость обучения.

В ходе исследований практических особенностей применения алгоритмов обучения для изображения, обработанного контурными фильтрами проведено компьютерное моделирование в среде Matlab. Моделирование обучения выявило случаи блокировки сети. Блокировка сети возникает, когда производные по различным весам различаются существенно, что приводит к остановке градиента в области локального экстремума целевой функции.

Во избежание блокировки сети предложено введение коэффициента инерционности μ для коррекции весов на предыдущем шаге итерации:

$$\Delta w_{ij}^n(t) = -h \delta_j^{(n)} x_i^n + \mu \Delta w_{ij}^n(t-1). \quad (6)$$

Для наискорейшего градиентного спуска доказана сходимость в пространстве весовых коэффициентов для бесконечно малых шагов [7]. Однако данное условие может оказаться выполнимым для времени обучения, стремящемся в бесконечность, что подтверждается и практическим моделированием работы сети. Это неприемлемо в условиях временных и ресурсных ограничений. Кроме того, высокая точность, получаемая на обучающей выборке, в ходе моделирования нередко приводит к неустойчивости результатов на тестовой выборке. Отметим, что чем лучше сеть адаптирована к конкретным условиям (изначально – к обучающей выборке), тем меньше она способна к обобщению и экстраполяции. Иными словами, сеть моделирует не функцию, а шум, присутствующий в обучающей выборке, особенно характерный для «живого» кадра графических объектов, как уже отмечалось выше. Констатируется переобучение сети [8].

Кроме того исследования показывают, что наискорейший градиентный спуск демонстрирует крайне медленную сходимость (рисунок 4) и имеет тенденцию к переобучению, обусловленную алгоритмической сложностью.

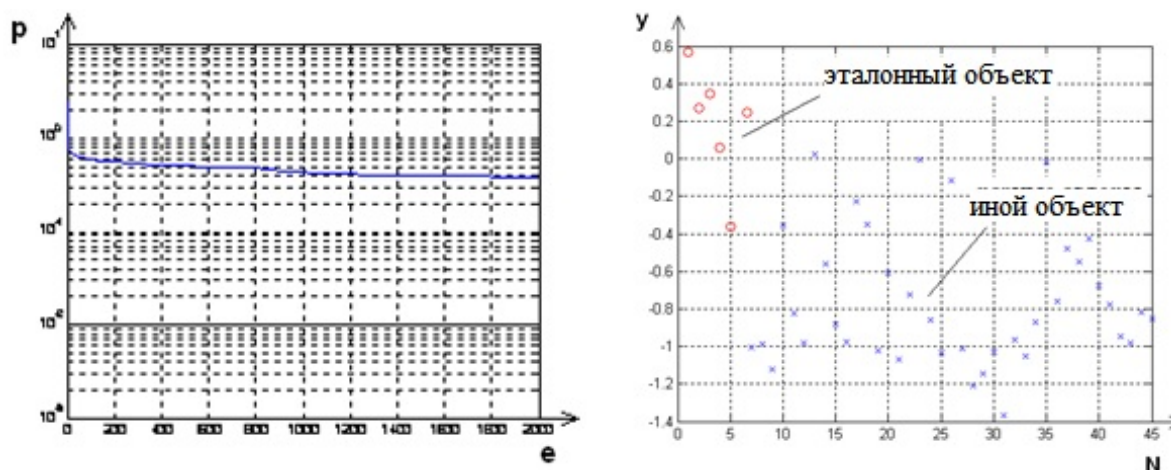


Рисунок 4 – Обучение сети наискорейшим градиентным спуском (e – количество эпох обучения, p – ошибка обучения, N – номер образца, y – выход сети)

Последовательность действий *gd*-алгоритмов: вычисление градиента – шаг вдоль антиградиента, не единственно возможная.

Ниже представлены результаты моделирования работы альтернативных, менее распространенных, но не менее эффективных алгоритмов Левенберга-Марквардта (*trainlm*), квазиньютоновского пересчета (*BFGS*) и Полека-Рибьера.

Алгоритм Левенберга-Марквардта основан на вычислении матрицы Якоби *A* для расчета промежуточной ошибки обучения *E(w)*. Основу обучения составляет изменение весовых коэффициентов нейронной сети на величину:

$$\Delta W = \begin{pmatrix} A \cdot E + I \cdot \mu \\ A \cdot A \end{pmatrix}. \quad (7)$$

Компьютерное моделирование функционирования *LM*-алгоритма показывает существенное снижение времени обучения в сравнении с наискорейшим градиентным спуском. Однако разброс ошибки обобщения остаётся высоким (рисунок 5).

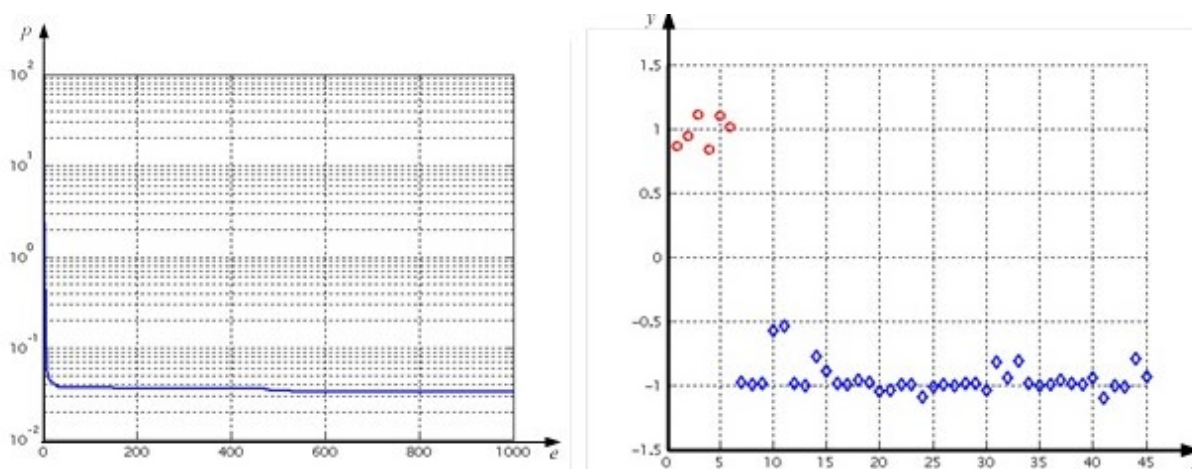


Рисунок 5 – Обучение сети алгоритмом Левенберга-Марквардта

В качестве альтернативы вычислению градиента (как в *GD*-алгоритме) и формированию матрицы первых частных производных (как в *LM*-алгоритмах) интересными представляются сведения о вторых частных производных квазиньютоновских *BFGS* (*Broyden C., Fletcher R., Goldfarb D., Shanno D.*) алгоритмов [9].

BFGS с ограниченной памятью оперирует информацией о кривизне оптимизируемой функции, содержащейся в матрице Гессе (*G*) (матрица вторых частных производных по управляемым параметрам). Информацию относительно кривизны функции оптимизации возможно накапливать на основе наблюдения за изменением градиента *g* во время итераций спуска. При этом аппроксимация нелинейной оптимизируемой функции производится без явного формирования матрицы Гессе [8].

К началу *k*-й итерации поиска известна аппроксимация матрицы Гессе минимизируемой функции, то есть информация о кривизне, накопленной на предыдущих итерациях. Очередное направление P_k на *k*-м шаге определяется как $B_k P_k = -g_k$, где B_k – матрица, аппроксимирующая гессиан; g_k – градиент функции оценки в начальной точке *k*-го шага.

После вычисления новой точки X_{k+1} и приближения B_k матрицы Гессе информация о кривизне функции обновляется, формируя матрицу B_{k+1} . *BFGS*-формула [7] и определяется как:

$$P_{k+1} = -g_{k+1} + \frac{(S_k, g_{k+1}) Y_k + (Y_k, g_{k+1}) S_k}{(Y_k, S_k)} - \frac{(S_k, g_{k+1})}{(Y_k, S_k)} \cdot \frac{1 + (Y_k, Y_k)}{(Y_k, S_k)} \cdot S_k, \tag{8}$$

где P_k – очередное направление квазиньютоновского поиска на *k*-м шаге;
 $S_k = (X_{k+1} - X_k)$ – вектор, показывающий изменение *X* на *k*-й итерации;
 $Y_k = (g_{k+1} - g_k)$ – изменение градиента.

В *BFGS* используются результаты предыдущего шага, поэтому первый шаг необходимо проводить иным способом, например, градиентным спуском.

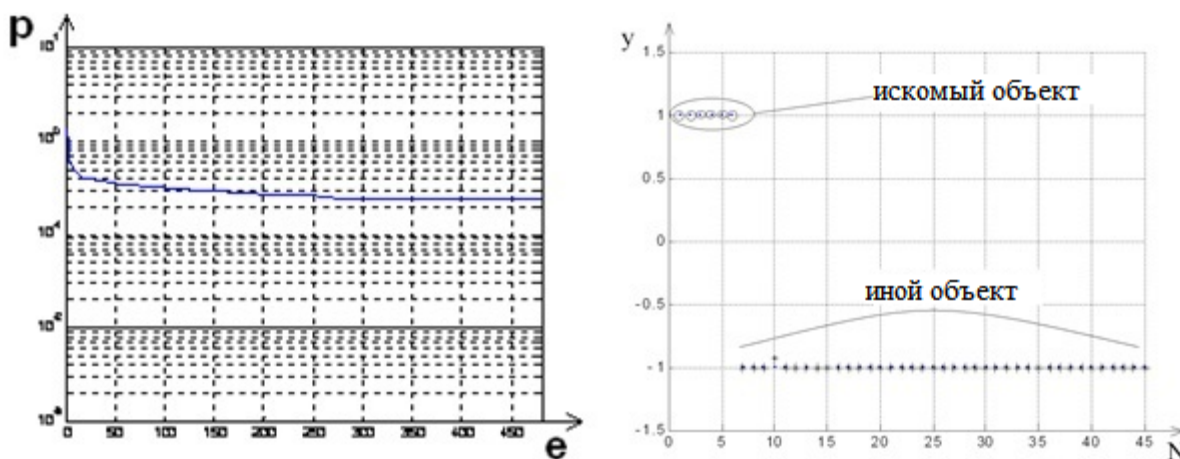


Рисунок 6 – Обучение сети квазиньютоновским (*BFGS*) алгоритмом

Из результатов тестирования (рисунки 4-6) нетрудно заметить, что нейронная сеть обратного распространения ошибки, обученная квазиньютоновским алгоритмом, является более эффективной в условиях поиска графического объекта на зашумленном образе. Обученная сеть демонстрирует хорошую сходимость и уверенную детерминацию объекта с минимизацией ошибок обучения и обобщения.

Временные и качественные показатели тестирования сети обратного распространения ошибки, обученной различными алгоритмами, представлены в таблице 1.

Таблица 1 – Моделирование сетей обратного распространения ошибки с различными алгоритмами обучения

№	Алгоритм обучения	Ошибка обучения	Ошибка обобщения	Число эпох обучения	Время обучения, с
1	Наискорейший спуск	0,0841	0,056	10000	259
2	Левенберга-Марквардта	0,0034	0,001	550	49
3	Квазиньютоновский	0,0007	0,00003	215	27
4	Полека-Рибьера	0,09	0,0008	328	35

Заключение

В статье предложен подход к повышению эффективности систем технического зрения робототехнических комплексов военного назначения. Рассмотрены различные варианты детерминации объекта на графической сцене посредством синтеза контурной предобработки графической сцены разностными фильтрами и нейросетового анализа. Предложенное решение позволяет повысить качество классификации в условиях временных и ресурсных ограничений. Контурное кодирование априорно известной информации о детерминируемом объекте позволяет синтезировать входную обучающую выборку и эффективно применить нейронные сети обратного распространения ошибки. Компьютерное моделирование различных алгоритмов обучения данной топологии сетей в условиях полученной признаковой базы показывает большую эффективность квазиньютоновских алгоритмов, основанных на *BFGS*-формуле пересчета весовых коэффициентов синоптических связей.

Список использованных источников

1. Виноградов А.Н., Калугин Ф.В., Недев М.Д., Погодин С.В., Талалаев А.А., Тищенко И.П., Фраленко В.П., Хачумов В.М. Выделение и распознавание локальных объектов на аэрокосмических снимках // *Авиакосмическое приборостроение*. 2007. №9. – С. 39-45.
2. Горский А.С., Никоноров В.И. Распознавание образов на поле боя на основе применения искусственных нейронных сетей // *Стратегическая стабильность*. 2021. №3(96). – С. 63-66.
3. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. 4-е изд. Т.3. Обучение, восприятие и действие. СПб.: Диалектика, 2022. – 640 с.
4. Ермоленко А.В., Корлякова М.О. Формирование нейросетевой модели геоинформационного датчика // *Известия института инженерной физики*. 2011. №1(19). – С. 25-30.
5. Bartlett P., Shawe-Taylor J. Generalization performance of support vector machines and other pattern classifiers // *Advances in kernel methods: support vector learning*. 1999. February. – P. 43-54.
6. Platt J. C. Fast training support vector machines using sequential minimal optimization // *Advances in kernel methods: support vector learning*. 1999. February. – P. 185-208.
7. Хайкин С. Нейронные сети. 2-е изд. М.: Вильямс, 2019. – 1104 с.
8. Казанович Я.Б. Сегментация изображений с помощью динамических нейронных сетей // *Научная сессия МИФИ-2008: материалы X Всероссийской науч.-технич. конф. «Нейроинформатика-2008»*. Ч.2. М.: МИФИ, 2008. – С. 37-96.
9. Белова И.К., Дерюгина Е.О., Ермоленко А.В. Методы контурного анализа при формировании пространства признаков в задачах нейросетевой идентификации // *Электромагнитные волны и электронные системы*. 2016. Т.21. №7. – С. 37-45.