

Ю.В. Ветров (УО «Белорусский государственный университет информатики и радиоэлектроники», Минск)
Науч. рук. **Д.Н. Одинец**, канд. техн. наук, доцент

МОДЕЛЬ МНОГОЯДЕРНОГО НЕЙРОСЕТЕВОГО КЛАССИФИКАТОРА БЫСТРОГО ПЕРЕОБУЧЕНИЯ

Для современных мобильных робототехнических комплексов (МРК) широкое распространение интеллектуальные системы, выполняющие задачи классификации данных различного характера. Особенность данных систем состоит в том, что они адаптируются под конкретные условия в процессе своего жизненного цикла, формируя индивидуальную базу знаний. Использование одиночной нейронной сети как ядра классификатора в данном случае может привести к значительной потере производительности, связанной с длительным временем переобучения для каждого адаптационного цикла. Использование многоядерного нейросетевого классификатора, реализующего принцип быстрого переобучения, решает эту проблему.

Подобный классификатор основан на использовании нескольких распознающих ядер с распределёнными наборами знаний вместо одного ядра с полным набором. В общем случае, распознающим ядром является свёрточная нейронная сеть [1] с 4–5 скрытыми слоями, чередующими свёртку и субдискретизацию.

Ускорение достигается за счёт разбиения классификатора на отдельные независимые ядра, каждый из которых содержит индивидуальный набор знаний. Полный набор знаний классификатора, равномерно распределяется между всеми ядрами – таким образом, каждое ядро знает о существовании только своей выборки.

В ходе работы классификатора производится подача исходного образа на каждое из ядер с последующим выбором максимального выхода и его декодированием в блоке формирования результата. Такой подход определяет единственное требование к внутреннему строению ядер: увеличенную чувствительность к входным признакам, позволяющую выдать высокое значение выхода ядра в случае присутствия входного образа в базе знаний, либо же крайне низкое значение – в случае отсутствия.

Учитывая то, что в результате увеличения количества ядер и, как следствие, сокращения набора знаний для отдельного ядра уменьшается не только обучающая выборка, необходимая для переобучения, но и внутренняя структура ядра (нейронной сети), скорость обучения увеличивается экспоненциально:

$$N = \alpha \cdot \frac{1}{n^2}, \quad (1)$$

где N – итоговое количество операций;

α – коэффициент, зависящий от внутренней структуры ядра и влияющий на количество операций;

n – фактор кластеризации – количество ядер в классификаторе.

На рисунке 1 приведена структурная схема обобщённой модели многоядерного нейросетевого классификатора, использующего алгоритм обратного распространения ошибки для переобучения.

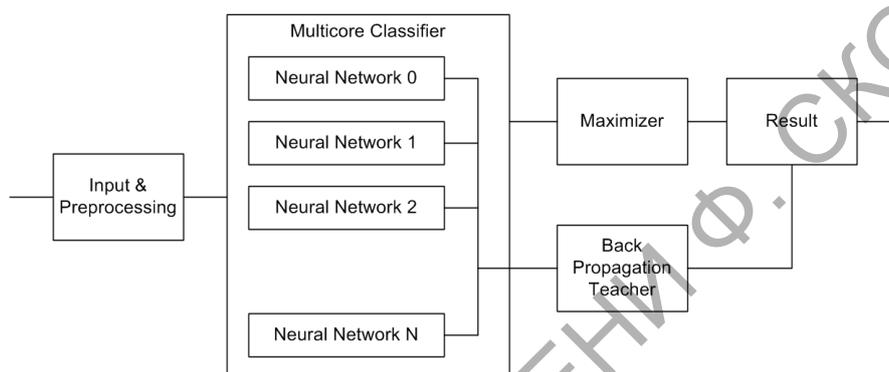


Рисунок 1 – Структурная схема многоядерного нейросетевого классификатора

Принимая во внимание логическую и архитектурную независимость ядер в классификаторе, можно реализовать преимущество экспоненциального роста в проектируемой системе за счёт использования средств параллельного программирования (CUDA, MPI) практически без потерь.

Следует отметить важный факт: переобучение данной модели в процессе её работы затрагивает только одно ядро. Одноядерный классификатор для переобучения потребует изменять пространство признаков для каждого элемента в наборе знаний, что увеличит число операций и, соответственно, время обработки экспоненциально (формула 1). Таким образом, данная модель пригодна к использованию в МРК, в которых требуется адаптация (переобучение) в режиме реального времени.

Перечислим преимущества предложенной модели:

1. Увеличение скорости начального обучения за счёт возможности к использованию средств параллельного программирования без изменения архитектуры.

2. Возможность быстрого переобучения, которое затрагивает только часть набора знаний классификатора. Принимая во внимания экспоненциальный рост скорости (формула 1) и современные высокопроизводительные средства программирования, можно добиться переобучения в фоновом режиме, которое не будет заметно на уровне пользователя.

3. Распределение набора знаний по отдельным ядрам, что позволяет исключать коллизии в классификации топологически схожих образов. Распределение знаний за счёт использования нескольких ядер позволяет разделить подобные образы по разным кластерам, что исключит коллизии и увеличит точность классификатора.

Программная модель системы была реализована на платформе Microsoft .NET Framework C# и оправдала все заявленные требования.

Таким образом, была разработана теоретическая и практическая модель многоядерного классификатора, использующего в качестве ядер нейронные сети. Данная модель обеспечивает высокую производительность. Программная реализация данной модели является переносимой и готова к встраиванию в любую интеллектуальную систему.

Работа выполнена в рамках ГПНИ «Механика, металлургия, диагностика в машиностроении» (Задание 1.13). Методическая помощь магистранту в исследовательской работе оказана ООО «Интеллектуальные процессоры» в рамках Европейской программы TEMPUS («Centers of Excellence for young RE-Searchers» № 544137-CERES).

Литература

1. Convolutional Neural Networks (LeNet) – Deep Learning 0.1 documentation. Deep Learning 0.1. LISA Lab [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://deeplearning.net/tutorial/lenet.html>.

В.Р. Власенко (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **А.В. Воружев**, канд. техн. наук, доцент

ТЕСТИРОВАНИЕ СЕТЕЙ ПРИ ПОМОЩИ УТИЛИТЫ SCAPY

Scapy – сетевая утилита, написанная на языке Python, которая позволяет посылать, просматривать и анализировать сетевые пакеты. В отличие от многих других утилит, утилита Scapy не ограничена только теми протоколами, пакеты которых она может генерировать. Фактически, она позволяет создавать любые пакеты и комбинировать атаки различных типов.

Нагрузочное тестирование (англ. load testing) – подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).