

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Национальный исследовательский Томский государственный университет
Томский государственный университет систем управления и радиоэлектроники
Болгарская Академия наук
ООО «Научно исследовательское предприятие «Лазерные технологии»

ИННОВАТИКА-2019

СБОРНИК МАТЕРИАЛОВ

**XV Международной школы-конференции студентов,
аспирантов и молодых ученых
25–27 апреля 2019 г.
г. Томск, Россия**

Под редакцией А.Н. Солдатов, С.Л. Минькова

Scientific & Technical Translations



ИЗДАТЕЛЬСТВО

Томск – 2019

МОДЕЛИРОВАНИЕ ПОЛНОСВЯЗНОЙ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ В СРЕДЕ MATLAB

В. Шатравин

*Национальный исследовательский Томский государственный университет
shatravin@stud.tsu.ru*

MODELING A FEED-FORWARD NEURAL NETWORK IN MATLAB

V. Shatravin

National Research Tomsk State University

The article contains description of reconfigurable neural networks and results of modeling of simple feed-forward neural network-based classifier in Matlab Simulink.

Keywords: reconfigurable artificial neural network, modeling neural network in Matlab Simulink, machine learning, classification.

В последние десятилетия в информационных системах разного характера всё большую роль играет машинное обучение. Это связано с тремя ключевыми факторами:

1) для широкого класса комплексных задач (распознавание образов, классификация, предсказание будущих значений параметра) поиск и реализация жесткого алгоритмического решения является трудоёмким процессом. Однако эти задачи успешно решаются алгоритмами машинного обучения и искусственными нейронными сетями (ИНС);

2) в XXI в. графические карты начали использоваться для решения задач общего назначения, было разработано аппаратное обеспечение, способное к эффективным параллельным вычислениям (FPGA, ASIC), что снизило время обучения ИНС до приемлемых значений;

3) для обучения ИНС необходим большой объём данных. С недавних пор низкая стоимость дискового пространства носителей и доступность сети Интернет позволяют собирать и хранить большое количество информации о протекающих в системах процессах.

Однако, существует класс задач, для которых применение ИНС по-прежнему представляет трудности. Мобильные роботы, в особенности беспилотные летательные аппараты (БПЛА), имеют жёсткие ограничения к габаритам, массе и энергопотреблению узлов. Применение энергозатратных модулей приводит к ухудшению автономности, что является нежелательным эффектом [1]. Это затрудняет использование в составе одного робототехнического комплекса нескольких ИНС.

Решением данной проблемы может быть реализация динамически перестраиваемых искусственных нейронных сетей. Особенность данного типа ИНС – способность динамически менять свою структуру и значения параметров под воздействием внешнего управляющего сигнала. В каждый момент времени подобная вычислительная система будет находиться в одном из нескольких предопределённых состояний, каждое из которых является полноценной ИНС со своими параметрами. Управляющий сигнал приводит к переходу в новое состояние с другими значениями параметров. Это позволяет использовать единое реконфигурируемое устройство для решения различных задач.

Подобное устройство может быть реализовано на основе концепции перестраиваемых вычислительных сред (ПВС) [2]. ПВС представляет собой структуру, состоящую из простых однотипных элементов и соединений между ними. Каждый элемент посредством внешнего управляющего сигнала может быть динамически сконфигурирован на выполнение одной операции из некоторого базиса [3]. Задав соответствующий базис и разрядность данных, можно реализовать на базе ПВС широкий класс цифровых устройств, в том числе ИНС.

ПВС может быть решением проблемы динамического обучения ИНС в процессе эксплуатации. Часть вычислителей ПВС может быть сконфигурирована в обучающий блок для текущей архитектуры ИНС, для её обучения. Преимущество ПВС в подобных задачах – возможность независимого функционирования отдельных элементов, что допускает высокое распараллеливание процесса обучения ИНС.

Одним из ключевых этапов исследования реконфигурируемых ИНС на основе ПВС является построение модели классической ИНС вместе с её блоком обучения. В работе рассматривается реализация простой полносвязной сети с двумя входными нейронами, четырьмя нейронами на скрытом слое и одним выходным нейроном (рис. 1) в среде Matlab Simulink. Данная среда удобна для проведения экспериментов над моделируемой системой. Наибольший интерес для исследований представляют модели, состоящие из простых атомарных элементов.

Моделируемая в данной работе ИНС предназначена для решения задач классификации объекта по двум признакам (обозначим их f_1 и f_2) между двумя классами (c_0 и c_1). Характер распределения экземпляров между классами в зависимости от значений признаков приведён на рисунке 2. Обучающая и валидационная выборки были сгенерированы при помощи программного обеспечения автора [4] и содержат 2000 уникаль-

ных записей, доля валидационных данных – 0.2. Также использовался тестовый набор из 360 записей. Для упрощения принято, что объекты, лежащие вне границ своего класса, отсутствуют.

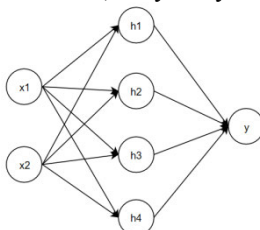


Рис. 1. Моделируемая ИНС

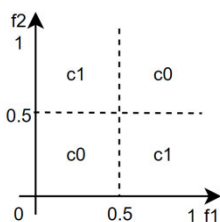


Рис. 2. Характер распределения классов на пространстве признаков

Первым этапом была реализована эталонная модель ИНС решения поставленной задачи. При этом использовались стандартные для машинного обучения инструменты – язык Python 3, приложения Jupyter Notebook, Keras, пакеты Pandas, NumPY, Scikit-learn. В связи с использованием ЭВМ компании AMD, TensorFlow заменён на PlaidML.

Испытания на эталонной модели показали, что оптимальные результаты (95-97% точности (accuracy, precision и recall)) достижимы при использовании в качестве функции активации ReLU для нейронов скрытого слоя и сигмоиды для выходного нейрона. Весовые коэффициенты инициализируются равномерно распределёнными случайными значениями (random uniform). Функция потерь – бинарная кросс-энтропия. Для приемлемой точности требуется 100 эпох обучения. Блок обучения реализует стохастический градиентный спуск (SGD).

Для достижения высокой степени атомарности разрабатываемой модели использовались преимущественно базовые элементы среды Simulink – константы, сложение, умножение, тактовая задержка. ReLU имитирует-

ся блоком \max и константой «0», производная от ReLU использует $\text{sign}(x)$. Сигмоида реализована функциональным блоком.

Для обучения сети с указанными параметрами необходимо произвести предварительную обработку обучающих и тестовых данных – приведение значений признаков к стандартному распределению. Данный этап является подготовительным и был реализован вне среды Simulink.

При обучении после каждой эпохи происходит перемешивание обучающей выборки. Для реализации подобного поведения после загрузки обучающей выборки (1600 объектов) при помощи окна команд Matlab был сгенерирован единый массив данных, включающий в себя случайно упорядоченные объекты, взятые в количестве эпох (1600×100 объектов). Первичная инициализация весов осуществлялась значениями, сгенерированными библиотекой *kegas* для эталонной модели ИНС.

Для оценки функционирования модели в среде Matlab было произведено сравнение результатов на выходе ИНС при подаче на вход тестовых данных с их исходными значениями. Абсолютная разность между этими величинами отражена на рисунке 3. Пунктиром обозначен уровень $y = 0.5$ – порог отнесения объекта к конкретному классу. Как видно из графика, лишь для 7 объектов из 360 (2%) классификация ошибочна. Анализ показал, что пики на графике соответствуют граничным точкам, представляющим сложность для классификации. Периодичность пиков связана с упорядоченностью объектов в наборе.

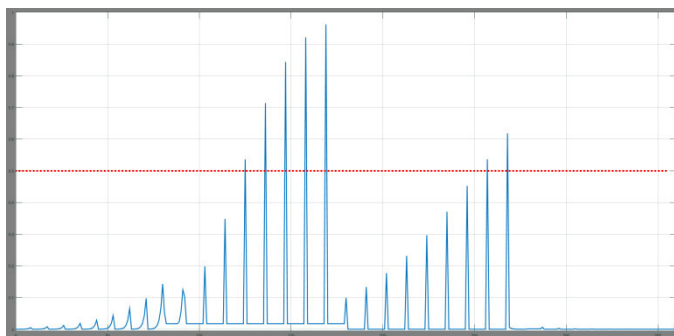


Рис. 3. Ошибка классификации модели сети

Литература

1. Carrio A. A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles / A. Carrio, C. Sampedro, A. Rodriguez-Ramos, P. Campoy // *Journal of Sensors*. – 2017. – Volume 2017. – P. 1–13.
2. Шидловский С.В. Логическая система с перестраиваемой структурой в задачах управления технологическими процессами // *Автометрия*. – 2005. – №4. – С. 104–113.
3. Шашев, Д.В. Морфологическая обработка бинарных изображений с использованием перестраиваемых вычислительных сред. / Д. В. Шашев, С. В. Шидловский // *Автометрия*. – 2015. – №3. – С. 19–26.
4. GitHub-репозиторий VladRaven5/NNDataGen [Электронный ресурс]. – URL: <https://github.com/VladRaven5/NNDataGen> (дата обращения 1.04.19).