УДК 004.057.5

ТЕХНОЛОГИИ ЛОКАЛИЗАЦИИ ГЛУБОКОГО ОБУЧЕНИЯ НА КЛИЕНТСКИХ УСТРОЙСТВАХ

Брыкин Г.С., бакалавр
МГТУ им. Н.Э. Баумана, факультет «Информатика, искусственный интеллект и системы управления», кафедра «Системы обработки информации и управления»
glebbrykin@colorfulsoft.ru
Научный руководитель: Гапанюк Ю.Е., к.т.н., доцент
МГТУ им. Н. Э. Баумана, факультет «Информатика, искусственный интеллект и системы управления», кафедра «Системы обработки информации и управления»
gapyu@bmstu.ru

Введение. В последние годы развитие технологий глубокого обучения привело к появлению множества алгоритмов, представляющих интерес для широкого круга пользователей, не владеющих специализированными аппаратными и программными средствами. Так, большой популярностью пользуются сервисы, позволяющие генерировать реалистичные изображения по текстовому описанию на естественном языке, реставрировать старые фотографии, подменять лицо и голос, стилизовать изображения под работы известных художников. Между тем возрастает потребность в переносе некоторых моделей глубокого обучения на клиентские устройства, обусловленная требованиями к конфиденциальности данных, высокому уровню доступности и надёжности. Существующие решения, предназначенные для локализации алгоритмов глубокого обучения на устройствах конечных пользователей, имеют ряд недостатков, затрудняющих, либо препятствующих, в некоторых случаях, их использованию. Так, например, существуют библиотеки для встраивания моделей глубокого обучения непосредственно в веб-страницы [1] [2] [3], при этом все вычисления производятся на стороне клиента. Недостатками данных решений на сегодняшний день являются повышенные требования к браузеру, невозможность встраивания в классические приложения, низкая эффективность в некоторых сценариях. Применение же классических фреймворков глубокого обучения [4] [5] в составе настольных и мобильных приложений недопустимо ввиду их требований к аппаратному и программному обеспечению. Существуют библиотеки, нацеленные на запуск предобученных моделей, например, ONNX [6]. Данные решения являются наиболее распространёнными на данный момент ввиду наивысшей эффективности, однако, они всё ещё являются аппаратно- и платформенно-зависимыми, а также не позволяют реализовать обучение моделей и запуск некоторых алгоритмов, например [7]. Разработка эффективных и удобных методов и технологий локализации глубокого обучения на клиентских устройствах является важной задачей, представляющей интерес, как для бизнеса, так и для некоммерческих проектов.

Основной идеей авторов работы является реализация всех компонентов нейросетевого приложения, включая вычислительный бэкэнд, в управляемом коде для платформ семейства .NET. Данное решение позволяет существенно упростить перенос на новые программные и аппаратные платформы, обеспечивая кроссплатформенность на двоичном уровне, а также позволяя реализовать автоматическую адаптацию программного продукта под аппаратные возможности текущей платформы, например, возможность автоматического выбора доступного и оптимального набора векторных инструкций процессора. Так, стек глубокого обучения и анализа данных System.AI [8] изначально предполагает и успешно реализует двоичную совместимость со всеми платформами, поддерживающими .NET Framework, Mono, .NET Core и .NET, независимо от операционной системы (Windows, Linux, Android, macOS, ReactOS) и архитектуры процессора (x86, x86-64, arm). Разработанный в рамках работы над проектом порт нейросети DeOldify [9] с графическим интерфейсом на двоичном уровне поддерживает .NET Framework и Mono на операционных системах Windows, Linux и ReactOS, при этом поддерживаются архитектуры процессора x86, x86-64 и ARM. DeOldify.NET может использовать векторизацию, если она доступна на текущей платформе,

либо же работать без неё. В ходе работы над проектом были созданы модификации алгоритмов свёртки Im2Col и Vinograd, требующие на несколько порядков меньше памяти, и, в некоторых случаях, работающие быстрее своих аналогов [10]. Помимо стека System.AI и набора приложений, был разработан новый высокоуровневый язык программирования общего назначения B#, предлагаемый в качестве замены языка Python. B# является C-подобным мультипарадигменным языком с динамической типизацией, поддерживающим низкоуровневые операции с памятью посредством указателей, а также современные возможности, такие как ООП. Поскольку B# является .NET-совместимым языком, программа, написанная на B#, может использовать все возможности BCL и сторонних управляемых сборок. Несмотря на то, что основной нишей для B# является написание модулей глубокого обучения и анализа данных, язык имеет все возможности для реализации других компонентов разрабатываемого продукта, например, графического интерфейса либо бэкэнда веб-приложения. В отличие от Python, B# является компилируемым языком, что позволяет отказаться от распространения исходных кодов вместе с продуктом, а также сделать продукт значительно более компактным за счёт отсутствия собственной среды выполнения и пакета стандартных библиотек. Более того, управляемая сборка, сгенерированная компилятором B#, совместима на двоичном уровне со всеми современными реализациями .NET, аппаратными и программными платформами, что в случае языка Python и C/C++ невозможно (данные языки обеспечивают кроссплатформенность на уровне исходного кода).

Заключение. В ходе работы над проектом был проанализирован рынок программных продуктов на базе глубокого обучения; рассмотрены существующие решения в области локализации моделей глубокого обучения на устройствах пользователей; предложено новое комплексное решение, позволяющее упростить разработку ПО на базе глубокого обучения для настольных, мобильных и встраиваемых устройств. Разработан детальный проект и функционирующие прототипы нового высокоуровневого языка программирования и портативной кроссплатформенной IDE для .NET. Изложенные в работе идеи успешно апробированы при реализации DeOldify.NET, доклады по разработанным алгоритмам были представлены на международной научно-технической конференции [11].

1. Daniel Smilkov, et. al: TensorFlow.js: Machine Learning for the Web and Beyond. arXiv:1901.05350 (2019)
2. Keras.js Homepage, https://github.com/transcranial/keras-js, last accessed 2024/02/06
3. ONNX.js Homepage, https://github.com/microsoft/onnxjs, last accessed 2024/02/06
4. Adam Paszke, et. al: PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 (2019)
5. Martín Abadi, et. al: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv:1603.04467 (2016)
6. ONNX Homepage, https://onnx.ai/, last accessed 2024/02/06
7. Leon A. Gatys, Alexander S. Ecker, Matthias Bethge: A Neural Algorithm of Artistic Style. arXiv:1508.06576 (2015)
8. Gleb S. Brykin: The System.AI Project: Fully Managed Cross-Platform Machine Learning and Data Analysis Stack for .NET Ecosystem. DOI: 10.14357/20790279230108
9. DeOldify.NET Homepage, https://github.com/ColorfulSoft/DeOldify.NET, last accessed 2024/02/06
10. Gleb S. Brykin: Patch2Vec: a simple and efficient convolution algorithm for mobile neural networks. In Intelligent Data Processing: Theory and Applications: Book of abstract of the 14th International Conference (Moscow, 2022), pp. 147-149, ISBN 978-5-907366-77-0
11. Gleb S. Brykin: DeOldify.NET: Cross-Platform Application For Coloring Black And White Photos. In XXIV International Scientific and Technical Conference "NEUROINFORMATICS-2022" : collection of scientific papers (Dolgoprudny, October 17–21, 2022), pp. 107-115, ISBN 978-5-7417-0823-1