

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ФРЕЙМВОРКОВ ГЛУБИННОГО ОБУЧЕНИЯ



В. А. Ковалев¹

Заведующий лабораторией
Анализа биомедицинских
изображений ОИПИ НАН,
кандидат технических
наук



А. А. Калиновский¹

Научный сотрудник
лаборатории Анализа
биомедицинских
изображений ОИПИ
НАН



С. Л. Ковалев²

Ведущий инженер в
ООО “Альторос Девелопмент”

¹Объединенный институт проблем информатики Национальная академия наук Беларуси, Республика Беларусь

²ООО “Альторос Девелопмент, Республика Беларусь

E-mail: vassili.kovalev@gmail.com

Abstract. Целью данной работы является проведение сравнительного анализа основных, на сегодняшний день, фреймворков глубинного обучения. Перечень выбранных для сравнения фреймворков включает в себя: Theano, Torch, Caffe, TensorFlow и Deeplearning4J. В качестве основных параметров оценки выбраны: точность классификации при заданном числе эпох обучения, скорость обучения и скорость классификации.

Введение. На начальном этапе в качестве базовой архитектуры сети для сравнения фреймворков глубинного обучения была выбрана Полносвязная Нейронная Сеть (*Fully Connected Neural Network - FCNN*). В общем случае под этой архитектурой в литературе понимается Многослойная Нейросеть Прямого Распространения (*Feedforward MultiLayer Perceptron, MLP*). Прямое распространение подчеркивает то, что данные между слоями преобразуются последовательно от входа к выходу без обратных связей, т.е. выход какого-либо слоя не поступает на вход предыдущих слоев. В этом заключается основное отличие от рекуррентной сети, где наличие обратной связи позволяет внести в сеть динамику, т.е. зависимость решения от “времени”.

Основное назначение FCNN – классификация данных. Согласно Универсальной Теореме об Аппроксимации [2], для представления произвольной функции, а соответственно и построения классификатора, достаточно 3-слойной нейросети с одним скрытым слоем. Основная проблема такого подхода заключается в том, что размер промежуточного слоя не ограничивается теоремой сверху. Поэтому на практике используются FCNN с одним и более промежуточными слоями.

В последнее время наблюдается возрастающий интерес к новым подходам

и программным решениям, основанным на алгоритмах глубинного обучения, которые являются, по своей сути, нейронными сетями нового поколения. Популярность этих методов может быть объяснена тем фактом, что, начиная с 2009 года, решения на основе алгоритмов глубинного обучения выиграли большое количество международных соревнований в области распознавания образов.

На данный момент существует несколько фреймворков, которые в своей основе построены на базе алгоритмов глубинного обучения. В основном эти фреймворки распространяются в виде открытого исходного кода и могут быть использованы для разработки широкого спектра программных продуктов в разных областях. Исходя из этого, задача выбора фреймворка является актуальной, а ее решение осложняется отсутствием объективной информации о качественных характеристиках фреймворков глубинного обучения.

В связи с вышеизложенным, целью данной работы является оценка основных фреймворков глубинного обучения, включая Theano[4], Torch [5], Caffe [6], TensorFlow [7] и Deeplearning4J [8], по таким характеристикам: точность классификации при заданном числе эпох обучения, время обучения, время предсказания, а также другие важные параметры. В данном исследовании мы ограничились только архитектурой Полносвязной Нейронной Сети, т.к. ожидается, что данная статья является первой в серии публикаций по сравнительному анализу фреймворков глубинного обучения.

Тестовые данные. В качестве тестовых данных для оценки фреймворков используется база рукописных цифр MNIST (Mixed National Institute of Standards and Technology)[9]. Это эталонный архив, который на сегодняшний день чаще всего используется для оценки алгоритмов классификации изображений. Данная база была получена путем перемешивания данных из разных источников, нормализации к размеру 28 на 28 пикселей и сглаживания. Также все изображения были преобразованы в оттенки серого цвета. Примеры изображений показаны на рисунке 1.



Рис.1. Примеры из тестовой базы MNIST

Цифровые изображения поступают на вход фреймворка в виде вектора размерностью 784 элемента, каждый из которых варьируется в диапазоне от 0 до 255. Задача классификации состоит в том, чтобы корректно соотнести цифру на изображении со значением от 0 до 9. На этапе обучения сети используется 60 000 изображений, другие 10 000 изображений используются для тестирования обученной модели и оценки точности классификации.

Методы. Как упоминалось выше, в качестве основных метрик для сравнения фреймворков были выбраны:

- точность классификации;
- время обучения;
- время классификации;
- размер исходного кода, необходимый для описания сети, оцененный в количестве строк.

Необходимо отметить, что в данном исследовании эксперименты проводились на CPU. Планируется, что на следующих этапах идентичные измерения будут проведены и на GPU.

Для оценки характеристик фреймворков при масштабировании сети перечисленные выше метрики проверены на Полносвязной Нейронной Сети с разными параметрами масштабирования:

- в первом случае изменялась «глубина» сети, т.е. количество слоев (рис. 2);
- во втором случае изменялась «ширина» сети, т.е. количество нейронов в одном скрытом слое (рис. 3).

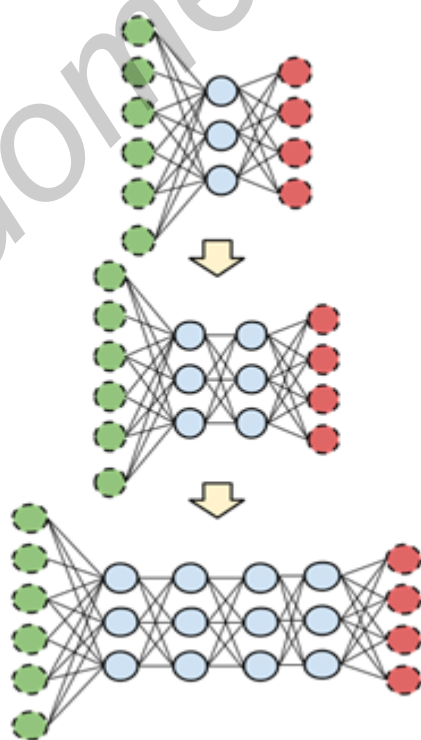


Рис. 2. Изменение параметров архитектуры в «глубину».

В экспериментах с изменением архитектуры сети в “глубину” применялись следующие значения параметров:

- количество скрытых слоев: от 1 до 4;
- количество нейронов в скрытом слое: 100.

В экспериментах с изменением архитектуры сети в “ширину” применялись следующие значения параметров:

- количество нейронов в скрытом слое: 64, 128, 512, 1024;
- количество скрытых слоев: 1.

В качестве функции активации применялись следующие функции:

- гиперболический тангенс (Tanh);
- положительная часть (Rectified-Linear Unit, ReLU) [10].

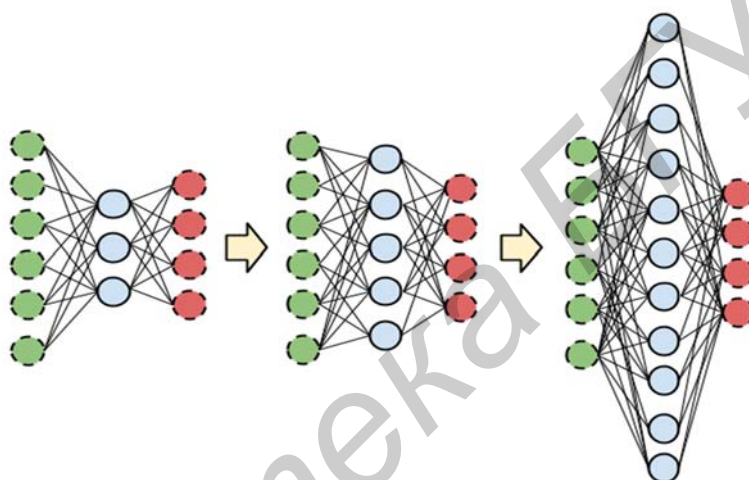


Рис. 3. Изменение параметров архитектуры в “ширину”

Для каждого типа активационных функций были проведены измерения со всем набором тестовых параметров.

Результаты

На рисунках 4 - 6 представлены результаты сравнения пяти вышеназванных фреймворков на полносвязной нейросетевой архитектуре по таким метрикам как: время обучения, время классификации, точность классификации для функции активации Гиперболический тангенс. Для всех экспериментов число эпох обучения равнялось 10.

На рисунке 4 можно наблюдать, что время обучения находится в районе отметки 30 секунд для всех четырех фреймворков кроме DeepLearning4J. Для этого фреймворка скорость обучения, в зависимости от количества слоев, варьируется от 140 для одного скрытого слоя, до 210 в случае 4 скрытых слоев.

Схожее поведение наблюдается для времени классификации (рис. 5), которое для Theano-Keras, Torch, Caffe и TensorFlow находится на уровне 0.4 секунд при запуске тестов на процессоре Intel i7. В то же время, для DeepLearning4J время классификации варьируется, начиная с 0.75 секунды для одного скрытого слоя, и постепенно увеличивается до 1.1 секунды для 4 слоев.

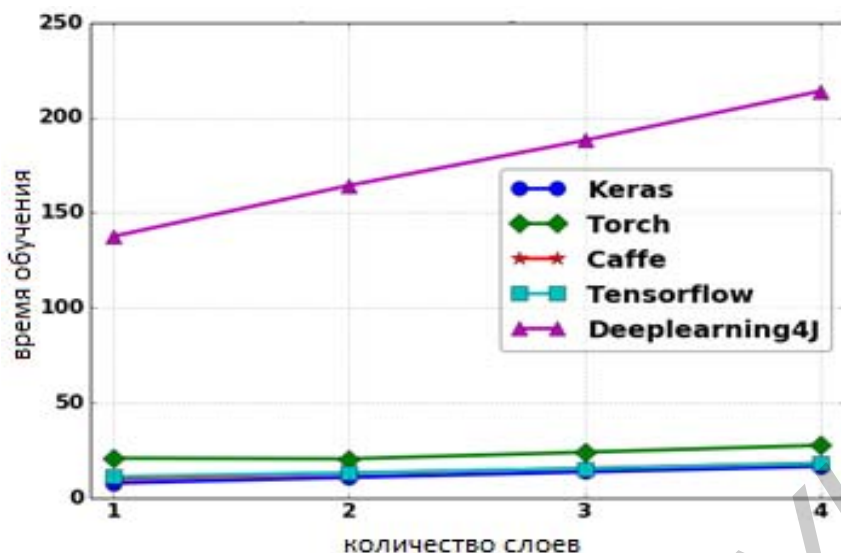


Рис. 4. Время обучения для случая масштабирования в глубину и функции активации гиперболический тангенс

В отличие от довольно когерентного поведения такого параметра, как времени обучения (рис. 4), точность классификации (рис. 6) показывает не столь однозначное поведение, делая выбор фреймворка более сложной задачей. В данном случае Theano-Keras, Deeplearning4J и Caffe показывают довольно высокую точность, начинающуюся с 94% (Caffe) для одного слоя и до 98% (Theano-Keras) для 4 слоев. В то время как Torch и TensorFlow значительно теряют в точности классификации с увеличением числа слоев сети.

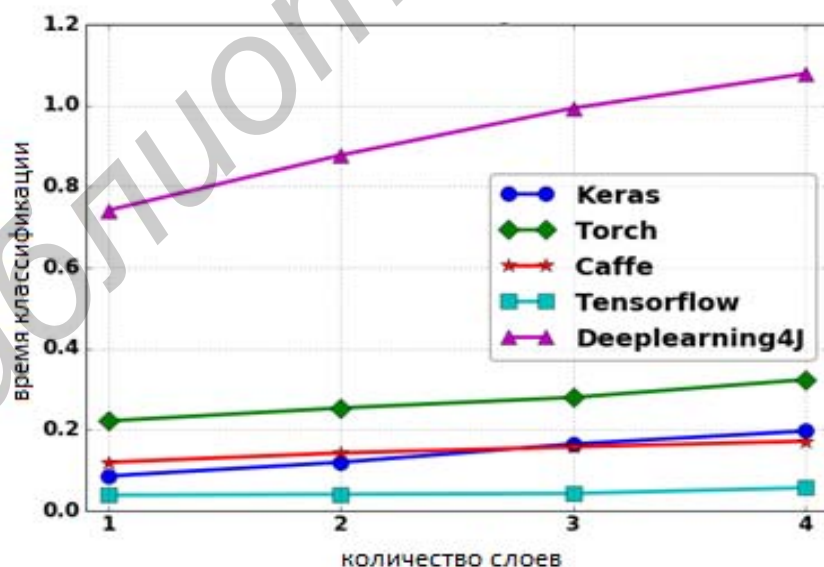


Рис. 5. Время классификации сети для случая масштабирования в глубину и функции активации гиперболический тангенс

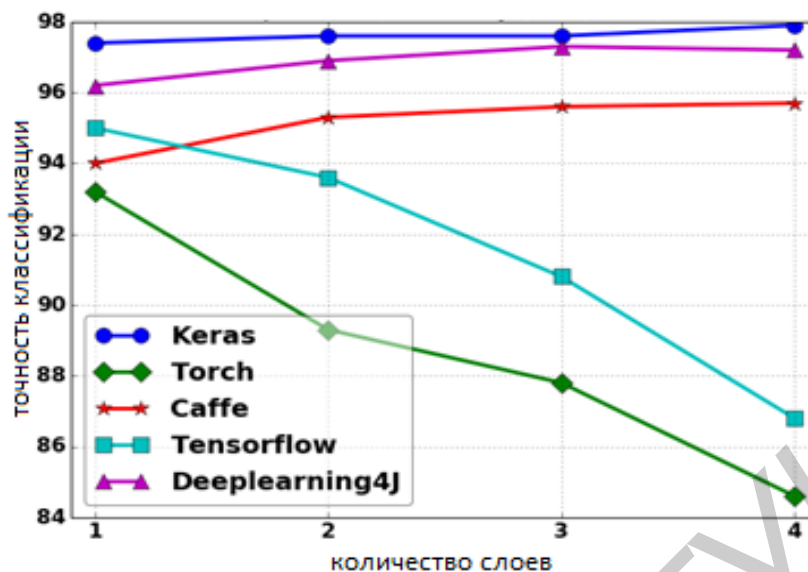


Рис. 6. Точность классификации сети для случая масштабирования в глубину и функции активации гиперболический тангенс

Также измерения времени обучения, времени классификации, точности классификации были проведены и для функции активации ReLU.

Сравнивая графики на рисунках 4 и 7, можно сделать вывод, что шаблон поведения фреймворков при разных функциях активации очень схож.

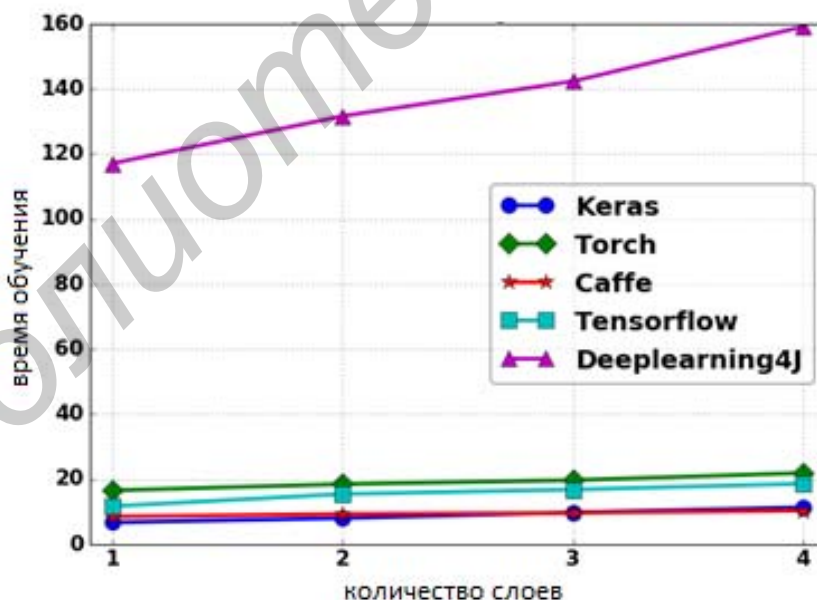


Рис. 7. Время обучения для случая масштабирования в глубину и функции активации ReLU.

Однако абсолютное значение времени обучения в случае функции активации ReLU примерно в два раза ниже, что является лучшим результатом по сравнению с функцией активации гиперболический тангенс. С другой стороны,

время классификации для функции ReLU (рис. 8), несмотря на некоторые флуктуации, очень схоже со временем классификации для функции активации гиперболический тангенс (рис. 5).

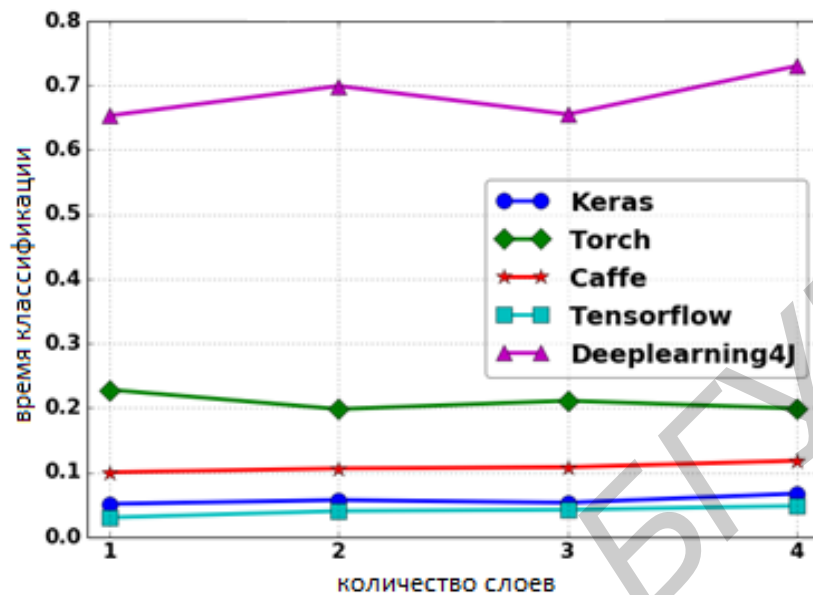


Рис. 8. Время классификации для случая масштабирования в глубину и функции активации ReLU

Далее приведены данные сравнительного анализа точности классификации для функции активации гиперболический тангенс (рис. 6) и ReLU (рис. 9). Из них следует, что Torch изменяет свое поведение значительно: точность классификации заметно возрастает, в то время как у других фреймворков этот параметр остается примерно на том же уровне.

Так как использование функции активации ReLU показало лучший результат в предыдущих тестах, оставшаяся часть эксперимента, оценивающая влияние увеличения размера скрытого слоя сети, проведена с ReLU функцией.

Как ранее было сказано, финальная часть исследования посвящена измерению времени обучения и точности классификации при изменении размера скрытого слоя сети, как показано на рисунке 3 в начале документа. Результаты эксперимента представлены на рисунках 10, 11 и 12 в том же порядке, как были представлены для предыдущих измерений.

Как можно видеть на рисунке 10, Deeplearning4J показывает существенно более длительное время обучения по сравнению с Theano - Keras, Torch, Caffe и TensorFlow. Кроме того, время обучения Deeplearning4J значительно возрастает с увеличением числа нейронов в слое. В то время как у других фреймворков время обучения остается практически постоянным при увеличении числа нейронов в слое.

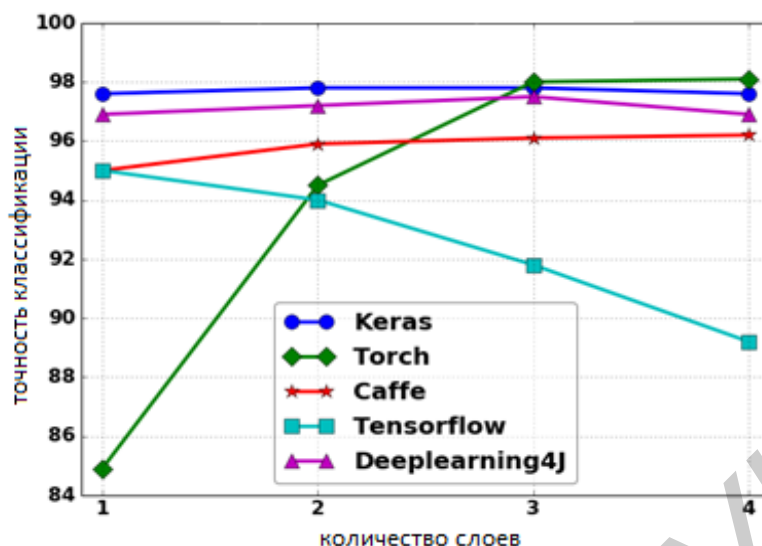


Рис. 9. Точность классификации сети для случая масштабирования в глубину и функции активации ReLU

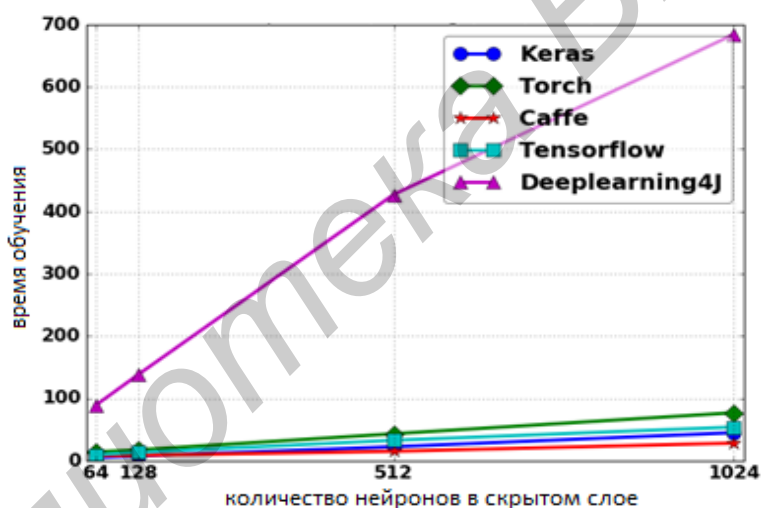


Рис. 10. Время обучения для случая масштабирования в ширину и функции активации ReLU

Что касается времени классификации (рис. 11), то все фреймворки показывают практически линейную зависимость от “ширины” скрытого слоя, хотя абсолютные значения этого параметра отличаются существенно.

Любопытно, что для большинства фреймворков, включая Theano – Keras, Caffe, Deeplearning4J и TensorFlow, точность классификации вне зависимости от изменения “ширины” слоя с 64 до 1024 нейронов остается практически постоянной (рис. 12). Однако в случае с Torch, точность классификации явно возрастает с ростом числа нейронов в слое.

Еще одним параметром фреймворка является сложность его интерфейса, которая измерена в строках кода, необходимых для реализации данного алгоритма (рис. 13).

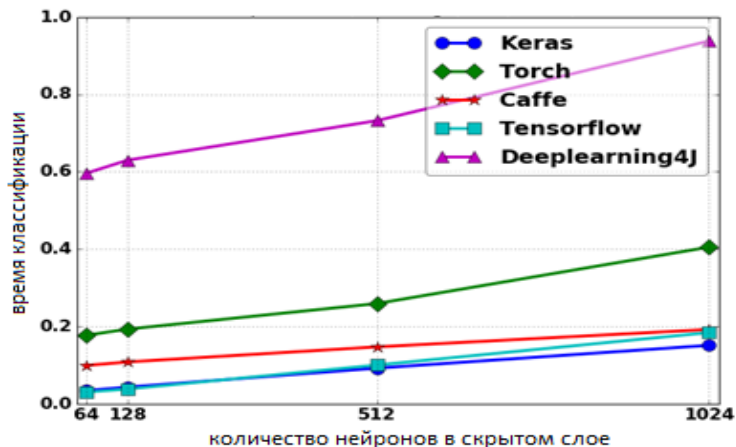


Рис. 11. Время классификации для случая масштабирования в ширину и функции активации ReLU

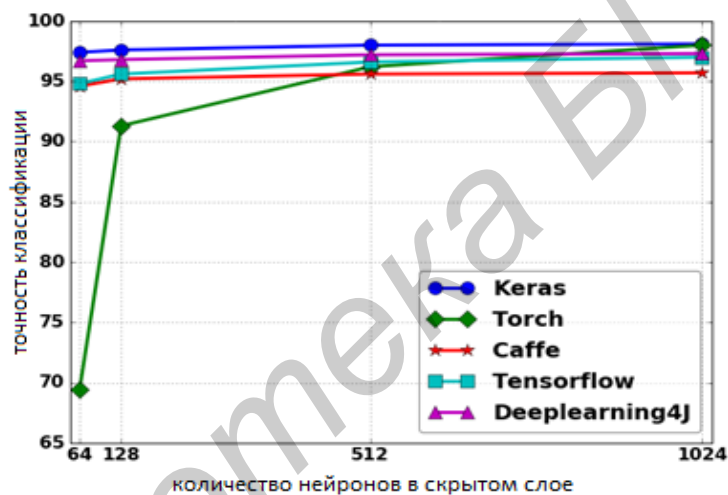


Рис. 12. Точность классификации сети для случая масштабирования в ширину и функции активации ReLU

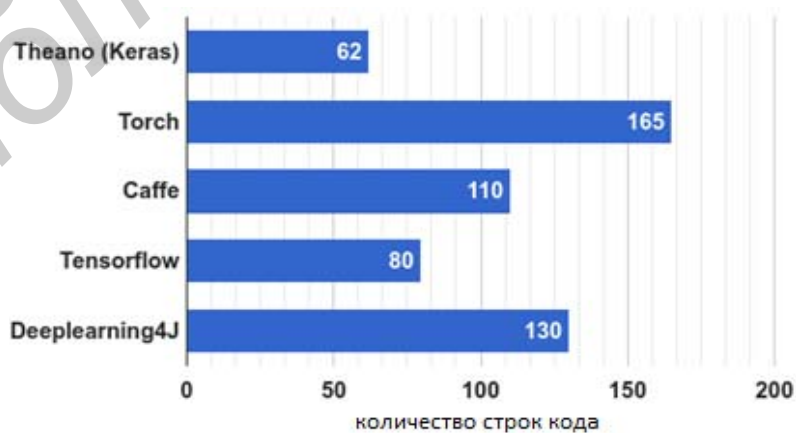


Рис. 13. Сложность интерфейса фреймворка, выраженная в строках кода, необходимых для реализации алгоритма

Выводы. Результаты, приведенные в данном исследовании, позволяют сделать выводы, сформулированные ниже в оценочном стиле. Количественные измерения приведены на графиках выше.

1 Как и ожидалось, увеличение количества слоев FCNN сети ведет к определенному росту вычислительных затрат, что приводит к увеличению времени обучения.

2 При увеличении количества нейронов в скрытом слое увеличивается время обучения, необходимое для достижения заданной точности классификации.

3 Использование ReLU в качестве нелинейной функции активации вместо гиперболического тангенса ведет к увеличению времени обучения и точности классификации.

4 При тестировании зависимости точности классификации от глубины нейронной сети (количества слоев) Torch и TensorFlow показали неожиданный результат: при увеличении глубины их точность классификации падает. Такой результат может быть объяснен либо некорректным вычислением градиента в процессе обучения, что маловероятно, либо необходимостью более тонкой настройки весовых коэффициентов, что требует более глубоких навыков работы с Torch.

5 В соответствии с результатами сравнения зависимости точности классификации от количества нейронов в скрытом слое, пять фреймворков разделились на 3 группы:

- для фреймворков Caffe и Deeplearning4J точность падает с увеличением количества нейронов;
- в отличие от предыдущих фреймворков TensorFlow и Torch, точность классификации увеличивается с увеличением числа нейронов в скрытом слое;
- Keras показывает стабильную точность классификации даже с относительно малым количеством нейронов в скрытом слое.

6 Deeplearning4J является самым медленным с точки зрения времени тренировки и классификации. Необходимо принять во внимание, что фреймворк находится в активной фазе разработки, и параметры могут измениться в будущем.

7 В итоге, принимая во внимание результаты тестирования скорости обучения, точности классификации, сложности построения алгоритма (количество строк кода), список фреймворков выглядит следующим образом: Theano - Keras,

Кроме численных параметров производительности, характеризующих фреймворки, можно выделить некоторые субъективные факторы, влияющие на решение о выборе фреймворка. Все фреймворки, приведенные в данном обзоре, являются инструментами с открытым исходным кодом. Наибольшее сообщество разработчиков поддерживает проекты у таких фреймворков, как TensorFlow и Theano - Keras. Такие фреймворки как Caffe и Theano - Keras на сегодняшний день являются более популярными в научном сообществе, в то

время как Torch и TensorFlow ориентированы на коммерческое использование. DeepLearning4J, в свою очередь, позиционируется, как основной фреймворк глубинного обучения в среде Java разработчиков. Наибольшее количество готовых моделей на сегодняшний день реализовано для таких фреймворков, как Caffe и Theano - Keras. Некоторые важные характеристики фреймворков сведены в таблицу 1.

Таблица 2. Характеристики фреймворков.

Фреймворк	Лицензия	Язык	CPU	GPU	Открытый код	Распределенный
TensorFlow	Apache 2.0	Python, C++	+	+	+	+
Theano	BSD	Python	+	+	+	-
Caffe	BSD	C++	+	+	+	-
Torch	BSD	Lua	+	+	+	-
DeepLearning4J	Apache 2.0	Java	+	+	+	+

Литература

- [1]. D.E. Rumelhart, G.E. Hinton, R.J. Williams. Learning representations by back-propagating errors. Cognitive modeling, vol. 5(3), 1988, p. 1.
- [2]. G. Cybenko. Approximations by superpositions of sigmoidal functions, Mathematics of Control, Signals, and Systems, vol. 2 (4), 1989, pp. 303-314.
- [3]. J. Schmidhuber. Deep Learning in Neural Networks: An Overview, Neural Networks, vol. 61, 2015, pp. 85–117.
- [4]. F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra I. Goodfellow, A. Bergeron, Y. Bengio. Theano: new features and speed improvements. arXiv preprint arXiv:1211.5590, 2012.
- [5]. R. Collobert, K. Kavukcuoglu, C. Farabet. Torch7: A matlab-like environment for machine learning. In: BigLearn, NIPS Workshop (No. EPFL-CONF-192376), 2011.
- [6]. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, ACM, 2014, pp. 675-678.
- [7]. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, S. Ghemawat. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv preprint arXiv:1603.04467, 2016.
- [8]. <http://deeplearning4j.org/> Last visited 16.04.2016.
- [9]. <http://yann.lecun.com/exdb/mnist/> Last visited 16.04.2016.
- [10]. X.Glorot, A.Bordes, Y.Bengio. Deep sparse rectifier neural networks. In: International Conference on Artificial Intelligence and Statistics, 2011, pp.315-323.