

Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра суперкомпьютеров и квантовой информатики

Сравнительный анализ библиотек глубокого обучения Tensorflow и PyTorch

Курсовая работа
студента группы 323
Васильева Семён Михайловича

Научный руководитель
к. ф.-м. н. Попова Нина Николаевна

Москва, 2019



Актуальность и проблема бэнчмаркинга библиотек глубокого обучения

Нейронные сети решают такие задачи, классификация изображения, распознавание голоса, машинный перевод и др. значительно лучше традиционных алгоритмов.

Процесс обучения нейронных сетей требует больших вычислительных затрат из-за объема данных, который требуется обработать, и числа параметров современных нейронных сетей. В силу этого остро встает вопрос эффективности алгоритмов обучения.

Библиотеки обучения нейронных сетей по большей части не стандартизированы. Это создает препятствия при проведении сравнительного анализа.



Постановка задачи

Провести анализ производительности фреймворков построения и обучения нейронных сетей Tensorflow и Pytorch.

Провести вычислительный эксперимент и собрать данные о процессе обучения с различными наборами значений гиперпараметров на основе системы IBM PowerAI на кластерном вычислительном комплексе IBM Polus.

Разработать программное средство, позволяющий пользователю провести аналогичный эксперимент с возможностью выбора своего набора значений гиперпараметров.



Методология

Две нейронные сети - AlexNet и VGG16, адаптированные под задачи cifar-10 и cifar-100 соответственно.

Для каждой две реализации: PyTorch, Tensorflow.

Реализации моделей на разных фреймворках приведены к единообразному виду.

Скрипт, позволяющий запустить процесс обучения с выбором гиперпараметров: размер mini-batch, оптимизатор, learning rate.

Два эксперимента: зависимость пропускной способности модели от размера mini-batch, и зависимость скорости обучения в эпохах от выбора оптимизатора и значения learning rate.



```
python3 train.py [options]
```

Номер параметра	Возможные значения	Описание
1	cpu, cuda	cpu - вычисления на центральном процессоре, cuda - на GPU
2	tf, pytorch	Фреймворк, реализация на основе которого будет запущена
3	alexnet, vgg	Модель
4	Целое число	Число эпох обучения
5	Целое число	Размер mini-batch
6	sgd, adam, rmsprop	Оптимизатор
7	Вещественное число	Learning rate



Batch size - пропускная способность

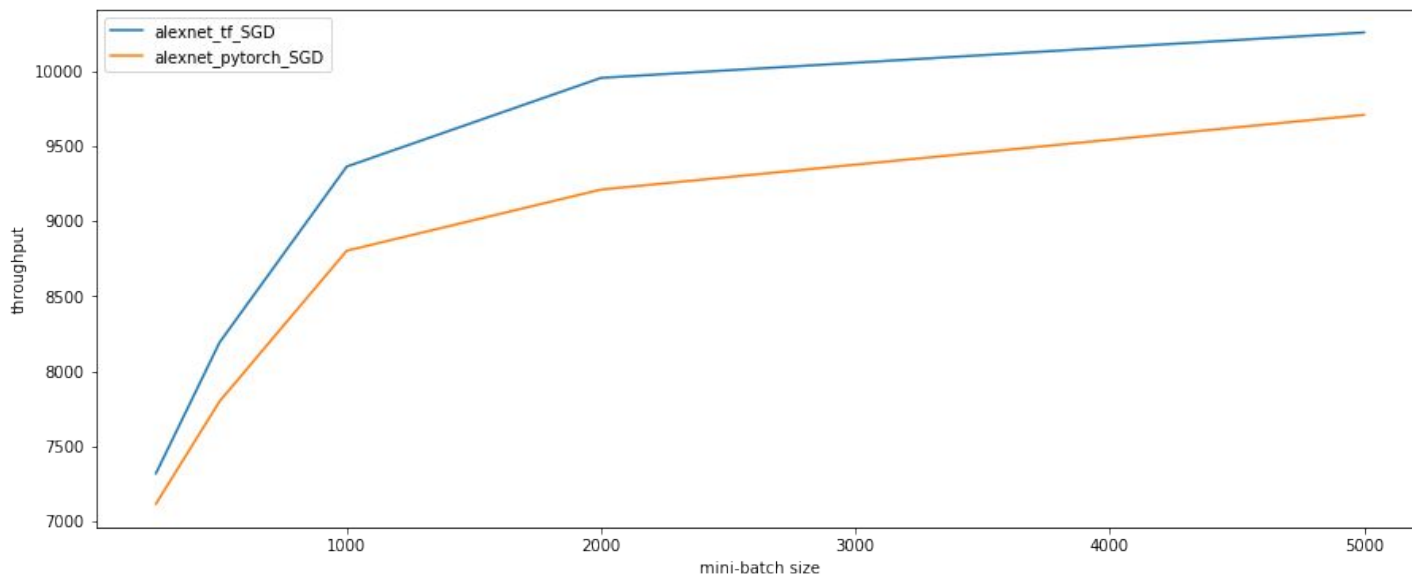
Model	Framework	Optimizer	Batch size	Epoch time	Throughput
AlexNet	Tensorflow	SGD	250	6,83	7322
			500	6,1	8193
			1000	5,29	9451
			2000	5,02	9960
			5000	4,88	10255



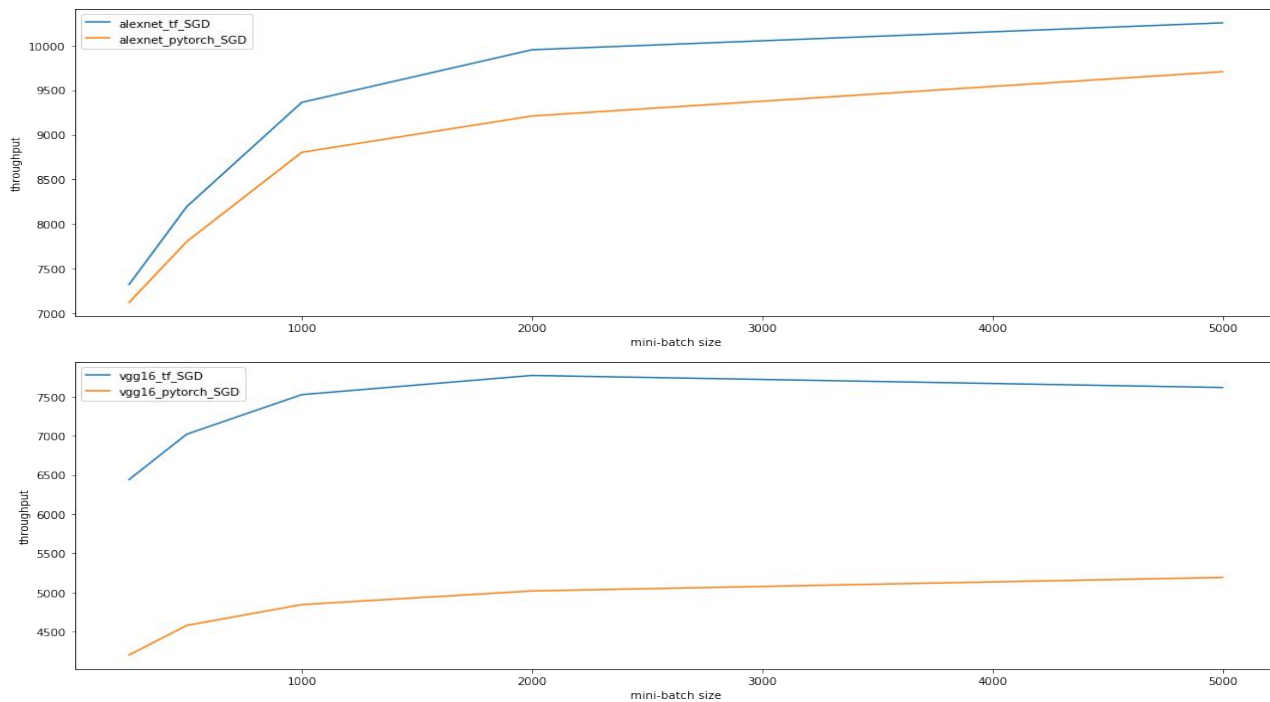
Оптимизатор / learning rate - скорость обучения

Model	Framework	Optimizer	L. r.	Ep. 5	Ep . 10	Ep.15	Ep.20	Ep.25	Ep.30
AlexNet	Tensorflow	SGD	0,01	0,517	0,657	0,755	0,851	0,917	0,977
			0,001	0,335	0,424	0,479	0,503	0,551	0,578
			0,0001	0,139	0,211	0,264	0,278	0,291	0,307

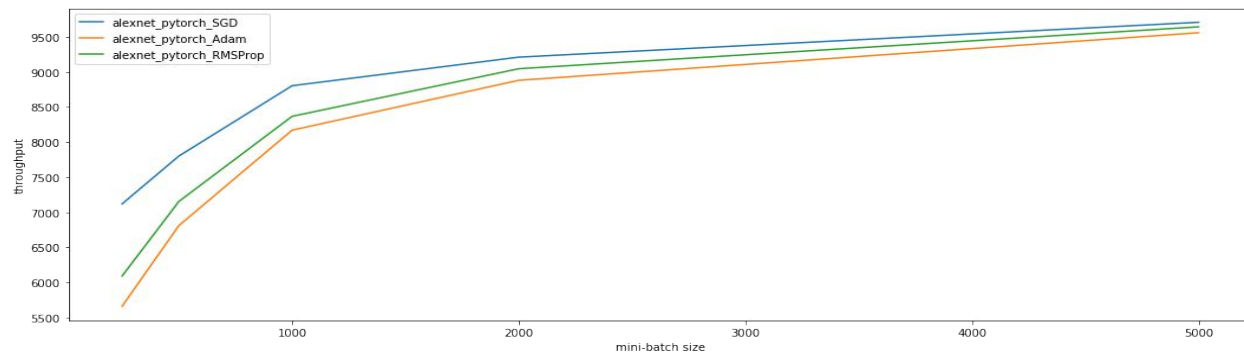
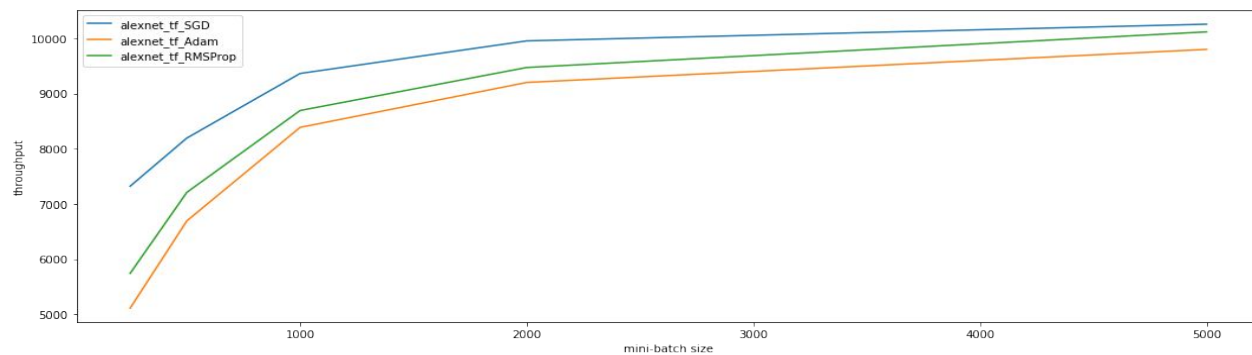
Пропускная способность зависит от размера mini-batch



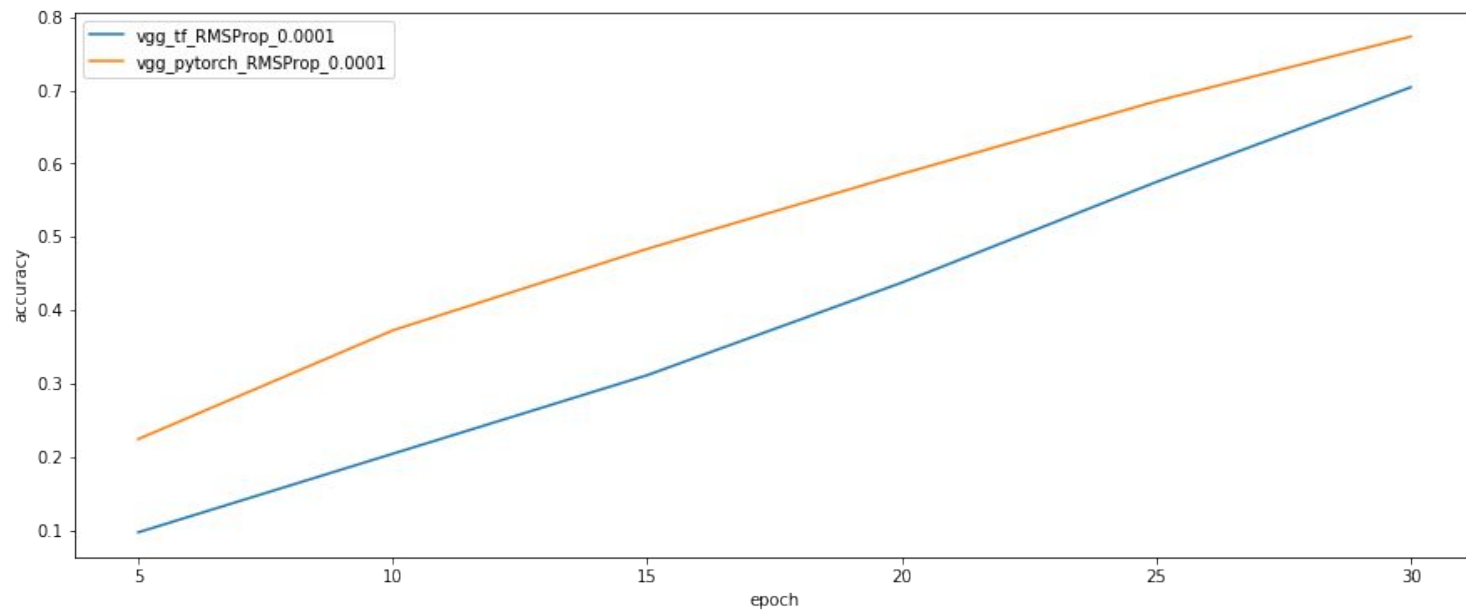
В Tensorflow операция свертки реализована эффективнее



SGD быстрее RMSProp, а RMSProp быстрее Adam



Модели, реализованные средствами PyTorch, сходятся за меньшее количество эпох





Заключение

В ходе выполнения курсовой работы:

- Реализовано программное средство для проведения эксперимента
- Проведен эксперимент, собраны данные
- Выполнен первичный анализ данных



Направление дальнейшей работы

- Расширить набор метрик
- Возможность подключать пользовательские модели
- Расширить набор поддерживаемых фреймворков