Інформаційні технології

УДК 004.8

Матківська Наталія Михайлівна

студентка

Навчально-наукового комплексу

«Інститут прикладного системного аналізу»

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

ПРОГРАМНІ ІНСТРУМЕНТИ ПОБУДОВИ НЕЙРОННИХ МЕРЕЖ

Анотація. *В даній статті проаналізовано основні переваги та недоліки таких програмних інструментів для практичної побудови та навчання нейронних мереж, як: Theano, Torch, Caffe, Neon і Keras, Tensorflow.*

Ключові слова: *нейронні мережі, програмне забезпечення, багаторівневе навчання.*

В останні роки навчання нейронних мереж перетворилось у більшій мірі в інженерну дисципліну. З'явилося багато зручних бібліотек, які дозволяють побудувати модель нейронної мережі (будь-якої глибини), сформувати для неї граф обчислень, автоматично порахувати градієнти і виконати процес навчання. Причому зробити це можна як на процесорі, так і на відеокарті, що значно прискорює виконання. Є бібліотеки загального призначення, які дозволяють створити будь-який граф обчислень і містять готові імплементації компонент нейронних мереж: звичайні слої, згорткові, рекурентні, сучасні алгоритми оптимізації.

Сьогодні існує більше десятка бібліотек багаторівневого навчання, деяким з яких притаманна вузьконаправлена специфіка. Ґрунтуючись на рекомендаційній інформації з широкого кола джерел, для подальшого розгляду було вибрано такі бібліотеки, як Theano, Torch, Caffe, Neon і Keras [1]. Базуючись на двох простих критеріях, виразність і розмір активної спільноти розробників, виділимо 3 основні бібліотеки: Tensorflow, Theano і Torch.

Довгий час лідером залишалася бібліотека Theano, розроблена в університеті Монреаля. Мінусом даного фреймворку є наявність додаткової фази компіляції графу обчислень, що потребує велику кількість часу для налаштування будь-якої архітектури нейронної мережі. В 2015 році Google випустила бібліотеку з відкритим вихідним кодом - Tensorflow. Вона має більш виразний інтерфейс і, на відміну від Theano, дана бібліотека була розроблена для промислового використання, а не лише для науково-дослідницьких задач. Саме тому вона містить такі важливі функції, як можливість запуску в мобільному середовищі, можливість запуску на кластері з CPU та GPU та масштабування даного обчислювального кластеру.

Існують два типи бібліотек: символьні та імперативні. У символьних бібліотеках набагато більше можливостей багаторазового використання пам'яті, а оптимізація на основі графів залежностей здійснюється автоматично. Найпопулярнішими символьними (symbolic) бібліотеками на даний час є TensorFlow і Theano. На відміну від Theano, Tensorflow не орієнтована виключно на навчання нейронних мереж, тому можна використовувати колекції графів і черги в якості складових частин для високорівневих компонентів. Якщо необхідно навчати масштабні моделі і використовувати багато зовнішньої пам'яті, то Theano буде дуже повільно працювати через необхідність компіляції коду C / CUDA в бінарний код.

TensorFlow має прозору модульну архітектуру з користувацьким інтерфейсом для кожного з них.

Архітектура Theano досить непроста: весь код - це Python, де код C / CUDA упакований як рядок Python. В такому коді важко орієнтуватися, його непросто відлагоджувати і проводити рефакторинг. Більш того, візуалізація графів в TensorFlow реалізована значно ефективніше, ніж в Theano.

Основним недоліком бібліотеки Torch є те, що вона написана на Lua – скриптовій мові програмування, яка не набула масової поширеності на ринку. Відсутність необхідності вивчати нову мову програмування дає можливість залучення великої аудиторії Python розробників, тому вибір здійснюється між Tensorflow та Theano.

Огляд бібліотеки Torch для багаторівневого навчання

Бібліотека Torch розроблена як бібліотека для обчислень в наукових цілях і підтримує безліч технологій. Бібліотека дозволяє гнучко працювати з нейронними мережами на досить низькому рівні. Для реалізації нейронної мережі в Torch необхідно написати власний цикл навчання, в якому оголошується функція замикання, що обчислює відповідь мережі. Це замикання передається в функцію градієнтного спуску для поновлення ваг мережі. Використання Torch не вимагає серйозних витрат за часом написання програмного коду, крім того, мережі Torch мають перевагу порівняно з мережами інших фреймворків швидкістю класифікації і наявністю вичерпної документації.

Огляд бібліотеки Theano для багаторівневого навчання

Theano існує в першу чергу як розширення мови Python, що дозволяє ефективно обчислювати математичні вирази, що містять багатовимірні масиви. У бібліотеці реалізований базовий набір інструментів для побудови штучної нейронної мережі. Процес створення моделі і визначення її

параметрів вимагає написання об'ємного коду, що включає реалізацію класу моделі, самостійного визначення її параметрів, реалізацію методів, що визначають функцію помилки, правило обчислення градієнтів, спосіб зміни ваг. Theano є найбільш гнучким фреймворком для побудови штучної нейронної мережі.

Огляд бібліотеки Caffe для багаторівневого навчання

Caffe реалізована на C ++ і має обгортки для Python і Matlab. Топологія ІНС, вихідні дані і спосіб навчання задаються за допомогою конфігураційних protobuf-файлів (технологія і протокол серіалізації даних, що перевершує по ефективності xml і json) в форматі prototxt. Побудова структури мережі виконується з простотою, зручністю і наочністю. Із запропонованих бібліотек є найбільш зручною у використанні, крім того, перевершує інші розглянуті фреймворки в швидкості навчання. Бібліотека Caffe підтримується досить великим співтовариством розробників і користувачів і на сьогоднішній день є найпоширенішою бібліотекою глибокого навчання широкого кола застосування.

Огляд бібліотеки Tensorflow для багаторівневого навчання

TensorFlow - це бібліотека програмного забезпечення з відкритим вихідним кодом для задач машинного навчання, розроблена Google. Вона дозволяє створювати і навчати нейронні мережі різної архітектури для виявлення і розпізнавання зразків і пошуку взаємозв'язків. TensorFlow також включає в себе TensorBoard, який являє собою засіб візуалізації в браузері для оцінки ефективності навчання і мережевих параметрів моделі [2].

TensorFlow досягає своєї продуктивності завдяки паралелізації завдань між центральним і графічними процесорами. Ядро кожної операції реалізовано на C++ з використанням бібліотек Eigen і cuDNN для кращої продуктивності.

Кожне обчислення в TensorFlow представляється як граф потоку даних, він же граф обчислень. Граф обчислень є моделлю, яка описує як будуть виконуватися обчислення. Важливо зауважити, що побудова графа обчислень і виконання операцій в заданій структурі - два різних процеси. Граф складається з плейсхолдерів (tf.Placeholder), змінних (tf.Variable) і операцій. У ньому відбувається обчислення тензорів - багатовимірних масивів, які втім можуть бути числом або вектором.

Графи виконуються в сесіях (tf.Session). Існують два типи сесій - звичайні та інтерактивні (tf.InteractiveSession); інтерактивна сесія підходить для виконання в консолі. Сесія зберігає стан змінних (Variables) і черг (queues). Явне створення сесій і графів гарантує належне звільнення ресурсів пам'яті [3].

У графі кожна вершина має 0 або більше входів і 0 або більше виходів, і являє собою реалізацію операції. Тензори представлені ребрами графа, а саме масивами довільного розміру (тип масиву вказується під час побудови графа). Особливі вершини, управляючі залежності (control dependencies), також можуть бути в графі: вони вказують, що вихідний вузол для контрольної залежності повинен закінчити виконання до того, як вузол одержувача контрольної залежності почне виконуватися.

Кожна операція має назву і являє собою абстрактне обчислення (наприклад, підсумовування). У операції можуть бути атрибути: наприклад, можливість зробити операцію поліморфною для різних типів тензорів. Ядро - специфічна реалізація операції, яка може виконуватися на певному типі пристрою (центральний або графічний процесор).

Змінна – особливий вид операції, який повертає вказівник на постійно мінливий тензор: така змінна не зникає після одиничного використання графа. Вказівники на подібні тензори передаються численним операціями, які

потім змінюють вказаний тензор. У завданнях машинного навчання, параметри моделі зазвичай зберігають тензори в змінних, які оновлюються на кожному кроці навчання.

Дана робота виконана на єдиному пристрої з використанням CPU.

Висновки. В даній роботі було проаналізовано основні переваги та недоліки таких програмних інструментів для практичної побудови та навчання нейронних мереж, як: Theano, Torch, Caffe, Neon і Keras, Tensorflow.

Однією з найпоширеніших платформ є платформа Tensorflow, у якій кожна модель представлена у вигляді графу потоку даних (або графу обчислень). Серед основних його переваг є висока степінь паралелізації обчислень графу, можливість виконання обчислювальних операцій як на центральному процесорі, так і на потужних дискретних відеокартах на базі програмної платформи CUDA. Також Tensorflow має досить велику аудиторію користувачів у вигляді Python розробників та багатий пакет супроводжувальної документації. В перелік стандартної поставки входять допоміжні інструменти розробки, такі як Tensorboard для візуалізації графу обчислень та результатів навчання.

Також в реєстрі Docker Hub є готові образи, що містять всі пакети та їх залежності для імплементації сервісів з підтримкою програмної платформи Tensorflow.

Література

1. K. Simonyan and A. Zisserman. Very deep convolu- tional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.

2. A.MasoodandA.AliAl-Jumaily.Computeraideddiagnos-
   tic support system for skin cancer: A review of techniques and algorithms. International Journal of Biomedical Imag- ing, 2013, 2013.
3. Krizhevsky, A. ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, G. E. Hinton // Advances in Neural Information Processing Systems 25. — 2012. — P. 1106–1114.