

## ПРЕДСТАВЯНЕ И СРАВНЕНИЕ НА СОФТУЕРНИ ПЛАТФОРМИ ЗА МАШИННО ОБУЧЕНИЕ

Александър Любенов<sup>1</sup>, Георги Михалев<sup>2</sup>

<sup>1</sup>Технически университет - Габрово

<sup>2</sup>Технически университет - Габрово

## INTRODUCTION AND COMPARISON OF SOFTWARE PLATFORMS FOR MACHINE LEARNING

Aleksandar Lubenov<sup>1</sup>, Georgi Mihalev<sup>2</sup>

<sup>1</sup>Technical university of Gabrovo

<sup>2</sup>Technical university of Gabrovo

### Abstract

*The paper reviews some of the most commonly used machine learning software platforms that have the ability to run on superclusters built by SBC. An overview of the application of machine learning in production for solving problems of different nature is made. The Keras, TensorFlow, PyTorch and Theano deep learning platforms are discussed in detail. An analytical comparison of the capabilities based on the advantages, disadvantages and functional capabilities for each of them, is made.*

**Keywords:** machine learning, software platforms, Keras, TensorFlow, PyTorch, Theano;

### ВЪВЕДЕНИЕ

Трансформацията на съвременните производства по пътя към Индустрия 4.0, става все по-разпространено. Чрез автоматизация и внедряване на т.н. машинното обучение се повишава капацитета на производството и ефективността, при същевременно поддържане на високи стандарти за качество.

Машинното обучение и индустриалната автоматизация в производството се считат за способи за преодоляване на много от най-належащите предизвикателства в производството[1,2], като например намаляване на маржовете на приноса и очаквания недостиг на квалифицирана работна ръка. С непрекъснатия напредък в алгоритмите, изчислителната мощност и наличността на данни, бързо се появяват и първите случаи на използване на машинно обучение в производството[3,4,5].

Тъй като индустриалната автоматизация играе все по-голяма роля в производството, дълбоките прозрения, които машинното

обучение може да предложи, са от решаващо значение за оптимизацията му. Но преди производителите да могат да въведат платформа за машинно обучение, те първо трябва да разберат как тези решения работят в производствена среда и как да изберат най-подходящото за своите нужди.

Машинното обучение е процес в който компютрите се обучават като човек, чрез различните видове рамки за обучение. Това означава, че може им се дадат входните данни - т.е. огромни количества реални данни-за да развият свои собствени автономни „мисловни процеси“ с течение на времето. Машинното обучение обикновено се класифицира като контролирано, неконтролирано или полу-контролирано подкрепящо обучение[6]. Обикновено в производството се използват два модела:

- Регресионен модел - който анализира историческите набори от данни, за да предскаже неща, като например продължителността на даден машинен

компонент, въз основа на предишен опит. Това е известно като оставаща полезна продължителност на живота - или RUL на използвания компонент.

- Класификационен модел - този тип модел може да предвиди, като вероятност да се повреди машина или компонент в рамките на предварително определен период от време.

В статията се обръща внимание на използването на платформи за дълбоко обучение с цел създаване на приложения работещи върху суперкълъстери изградени от единно бордови компютри.

## ИЗЛОЖЕНИЕ

Потенциалните ползи от Машинното обучение в рамките на различни сектори са огромни. Някои от най-убедителните причини за използване на машинно обучение и изкуствен интелект в производството са: предсказуемо качество, оптимизиране на производителността, намаление на разхода за материали, прогнозна поддръжка, удължен живот на машини и оборудване, чрез прогнозиране на оставащия полезен живот (RUL) и много други.

От друга гледна точка машинното обучение може да проникне в съвременните фабрики и да добави известна интелигентност към наличните машини чрез:

- Визия - чрез бързо и стабилно разпознаване и наблюдение на обекти при различни условия без сложни и специфични процедури за фина настройка.
- Поддръжка - след като се знае как нормално се държат машините, можете да предскажете бъдещото им състояние и да се предупреди за възможни неизправности, преди да настъпят повреди.
- Оптимизация - чрез подобряване на съществуващите параметри за управление и максимизиране на резултата, ефективността или производителността на машината.
- Контрол - чрез откриване на цели нови стратегии за управление на процеси, които към момента не могат да се овладеят.

Дългосрочната цялостна цифрова интеграция и усъвършенстваната автоматизация на целия процес на проектиране и произ-

водство биха могли да отворят допълнителни възможности. Персонализирането например е рядко и скъпо, докато масовите стоки с голям обем са доминиращият модел в производството, тъй като понастоящем разходите за препроектиране на фабрична линия за нови продукти често са прекомерни.

Потребителите в по-голямата си част са готови да направят компромис, защото масово произвежданите стоки са много поевтини. Ако технологията, която прави производството по-гъвкаво, е широко разпространена, което кара персонализирането да стане достатъчно евтино, това може да създаде реална промяна на много пазари. Това е само част от ползите, които дълбокото обучение, чрез различни софтуерни платформи и съответен хардуер, може да донесе на индустрията.

Дълбокото обучение (Deep learning) е подгрупа на изкуствения интелект (AI), област, която набира популярност през последните няколко десетилетия. Подобно на всяка нова концепция, някои въпроси и подробности трябва да бъдат изгладени, преди да се използват в реални приложения[7].

## СОФТУЕРНИ ПЛАТФОРМИ ЗА ДЪЛБОКО ОБУЧЕНИЕ

Обикновено термините „дълбоко обучение“, „машинно обучение“ и „изкуствен интелект“ се използват взаимнозаменяемо и това води до потенциално объркване. Дълбокото обучение и машинното обучение са част от семейството на изкуствения интелект, въпреки че дълбокото обучение също е подмножество на машинното обучение.

Дълбокото обучение имитира невронните пътища на човешкия мозък при обработката на данни, използвайки ги за вземане на решения, откриване на обекти, разпознаване или превод на езици например. Обучението е без човешки надзор или намеса, извличайки информация от неструктурирани и немаркирани данни, чрез използване на йерархично ниво на изкуствени невронни мрежи, изградени като човешкия мозък. Докато традиционните програми за машинно обучение работят с анализ на данни линейно, йерархичната функция на дълбокото

обучение позволява на машините да обработват данни, използвайки нелинеен подход[8]. Разбирането на нюансите на тези концепции е от съществено значение за всяко обсъждане на някои от основните платформи за дълбоко обучение и AI приложения: Keras, TensorFlow, PyTorch и Theano.

### **Keras**

Keras е ефективен интерфейс за програмиране на невронни мрежи на високо ниво (API), написан на Python. Това е библиотека с невронни мрежи с отворен код и е проектирана да осигурява бързи експерименти с дълбоки невронни мрежи и може да работи на върха на TensorFlow, Theano и Microsoft Cognitive Toolkit, познат по-рано като CNTK.

При Keras се фокусира върху това, че е модулен, лесен за употреба и разширяем. Не обработва изчисления на ниско ниво, а вместо това ги предава на друга библиотека, наречена Backend. Keras беше приет и интегриран в TensorFlow в средата на 2017 г. Потребителите имат достъп до него чрез модула tf.keras. Библиотеката Keras обаче все още може да работи отделно и независимо.

### **PyTorch**

PyTorch е сравнително нова рамка за дълбоко обучение, базирана на Torch. Разработен е от изследователската група на Facebook за ИИ, с отворен код на GitHub през 2017 г., той се използва за обработка на приложения за естествени езици. PyTorch има репутация за простота, лекота на използване, гъвкавост, ефективно използване на паметта и динамични изчислителни графики. Всичко това което прави писането на сорскод по-лесно управляемо и увеличава скоростта на обработка.

### **TensorFlow**

TensorFlow е цялостна рамка за дълбоко обучение с отворен код, разработена от Google и пусната през 2015 г. Известна е с поддръжка на документация и обучение, мащабируеми възможности за производство и внедряване, множество нива на абстракция и поддръжка за различни платформи, като Android.

TensorFlow е символична математическа библиотека, използвана за невронни мрежи

и е най-подходяща за програмиране на потока от данни в редица задачи. Той предлага множество нива на абстракция за изграждане на модели за обучение.

При така обещаващото и бързо развиващо се навлизане в света на дълбокото обучение, TensorFlow предлага гъвкава, всеобхватна система от общностни ресурси, библиотеки и инструменти, които улесняват изграждането и внедряването на приложения за машинно обучение. Също така, както бе споменато по-горе, TensorFlow прие Keras, което прави сравняването на двете трудно на пръв поглед. Независимо от това, все пак ще сравним двете рамки с оглед на пълнотата, особено след като потребителите на Keras не е задължително да използват TensorFlow.

### **Theano**

Theano беше една от най-популярните библиотеки за дълбоко обучение, проект с отворен код, който позволява на програмистите да дефинират, оценяват и оптимизират математически изрази, включително многомерни масиви и изрази с матрична стойност.

Theano е разработен от Университета в Монреал през 2007 г. и е ключова основополагаща библиотека, използвана за дълбоко обучение в Python. Theano е една от по-старите платформи за дълбоко обучение и неговата популярност намалява, въпреки, че все още се ползва в академичните среди.

## **СРАВНЕНИЕ НА ПЛАТФОРМИ ЗА ДЪЛБОКО ОБУЧЕНИЕ**

Сравнението на представените по-горе софтуерни платформи и библиотеки за дълбоко обучение е аналитично, базирано на значими литературни източници в областта на изкуствения интелект с приложен характер, като [9-28].

### **Разлики между PyTorch и TensorFlow**

PyTorch се използва за много проекти за дълбоко обучение днес и популярността му се увеличава сред изследователите на AI, въпреки че от трите основни рамки е най-малко популярен. Тенденциите обаче показват, че това може да се промени скоро. Когато изследователите искат гъвкавост, възможности за отстраняване на грешки и кратка продължителност на обучението, те

избират PyTorch. Той работи на Linux, macOS и Windows.

Благодарение на добрата му документация и изобилието от обучени модели и уроци, TensorFlow е любимият инструмент на много професионалисти и изследователи в индустрията. TensorFlow предлага по-добра визуализация, която позволява на разработчиците да отстраняват грешки по-добре и да проследяват процеса на обучение, докато PyTorch предоставя само ограничена визуализация.

TensorFlow побеждава PyTorch при внедряването на обучителни модели в производството, благодарение на рамката за обслужване (TensorFlow Serving framework). PyTorch не предлага такава рамка, така че разработчиците трябва да използват Django или Flask като бекенд сървър.

В областта на паралелизма на данните PyTorch печели оптимална производителност, като разчита на естествена поддръжка за асинхронно изпълнение чрез Python. С TensorFlow обаче трябва ръчно да кодирате и оптимизирате всяка операция, изпълнена на конкретно устройство, за да разрешите разпределеното обучение.

### **Разлики между PyTorch и Keras**

И двата избора са добри, ако тепърва се започва да се работи с платформи за дълбоко обучение. Математиците и опитни изследователи ще намерят PyTorch повече по свой вкус. Keras е по-подходящ за разработчици, които искат plug-and-play рамка, която им позволява бързо да изграждат, обучават и оценяват своите модели. Keras също предлага повече възможности за внедряване и по-лесен експорт на модел. PyTorch е по-бърз от Keras и има по-добри възможности за отстраняване на грешки.

И двете платформи се радват на достатъчно добра популярност, предлагат много учебни ресурси. Keras има отличен достъп до код за многократна употреба и уроци, докато PyTorch има изключителна поддръжка на общността и активно развитие.

Keras се явява най-добрият при работа с малки набори от данни, бързо прототипиране и многобройна поддръжка. Това е най-популярната платформа благодарение на сравнителната си простота. Той работи на

Linux, MacOS и Windows.

### **Разлики между TensorFlow и Keras**

TensorFlow е платформа с отворен код от край до край, библиотека за множество задачи за машинно обучение, докато Keras е библиотека от невронни мрежи на високо ниво, която работи върху TensorFlow. И двата предоставят API на високо ниво, използвани за лесно изграждане и обучение на модели, но Keras е по-лесен за употреба, защото е вграден в Python.

Изследователите се обръщат към TensorFlow, когато работят с големи набори от данни и засичане на обекти нуждайки се от отлична функционалност и висока производителност. TensorFlow работи на Linux, MacOS, Windows и Android. Рамката е разработена от Google Brain и в момента се използва за нуждите на Google за научни изследвания и производство.

Трябва да се има предвид, че сравняването на TensorFlow и Keras не е най-добрият начин да се подходи към въпроса, тъй като Keras функционира като обвивка към рамката на TensorFlow. По този начин може да се дефинира модел с интерфейс на Keras, който е по-лесен за използване, след което да отиде в TensorFlow, когато трябва да се използва функция, която Keras няма, или се търси специфична функционалност на TensorFlow. За да се постигне тази цел, може да се постави код на TensorFlow директно в тренировъчния модел на Keras.

### **Разлики между Theano и TensorFlow**

Theano предлага бързи изчисления и е специализирана в обучението на алгоритми за дълбока невронна мрежа. Той е между-платформен и може да работи както с централни процесори (CPU), така и с графични процесори (GPU).

TensorFlow работи и на CPU и GPU. Той се основава на изчисляване на графики, което позволява на разработчика да визуализира по-добре изграждането на невронната мрежа, използвайки TensorBoard, което улеснява отстраняването на грешки.

Изборът на това коя платформа да се избере за работа, често е субективен и зависи от приложението, нуждите, необходимостта от функции за AI проекта. Таблица 1

представя обобщение на сравнението на трите основни платформи, описано по-горе.

**Таблица 1.** Сравнение на характеристиките на Keras, PyTorch и TensorFlow

	Keras	PyTorch	TensorFlow
<b>API Level</b>	Високо	Ниско	Високо и ниско
<b>Архитектура</b>	Проста, стегната, ясна	Комплексна, по-малко ясна	Не е лесна за използване
<b>Работа с набор от данни</b>	Малък набор от данни	Голям набор от данни, висока производителност	Голям набор от данни, висока производителност
<b>Отстраняване на грешки</b>	Проста мрежа, рядка необходимост от отстраняване на грешки	Добри възможности при отстраняване на грешки	Отстраняването на грешки е по-трудно
<b>Наличие на обучени модели</b>	Да	Да	Да
<b>Популярност</b>	Най-популярна	Трета по популярност	Втора по популярност
<b>Скорост</b>	Бавна, ниска производителност	Бърза, висока производителност	Бърза, висока производителност
<b>Написана на</b>	Python	Lua	C++, CUDA, Python

## ЗАКЛЮЧЕНИЕ

В статията са представени някои от най-използваните платформи за разработване на приложения за дълбоко обучение. Дадени са някои от проблемите в индустрията и как AI разработките могат да помогнат за решаването им. Разгледани са софтуерните платформи Keras, TensorFlow, PyTorch и Theano. Описани са някои от техните предимства и недостатъци. Направено е и аналитично сравнение, разглежданите платформи, като обобщението е дадено таблично.

## REFERENCE

[1] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.  
 [2] Bengio, Yoshua, Ian Goodfellow, and Aaron Courville. Deep learning. Vol. 1. Massachusetts, USA: MIT press, 2017.  
 [3] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.

[4] Yan, Le Cun, B. Yoshua, and H. Geoffrey. "Deep learning." nature 521.7553 (2015): 436-444.  
 [5] Deng, Li, and Dong Yu. "Deep learning: methods and applications." Foundations and trends in signal processing 7.3-4 (2014): 197-387.  
 [6] Kussul, Nataliia, et al. "Deep learning classification of land cover and crop types using remote sensing data." IEEE Geoscience and Remote Sensing Letters 14.5 (2017): 778-782.  
 [7] Wang, Jinjiang, et al. "Deep learning for smart manufacturing: Methods and applications." Journal of manufacturing systems 48 (2018): 144-156.  
 [8] Fawaz, Hassan Ismail, et al. "Deep learning for time series classification: a review." Data mining and knowledge discovery 33.4 (2019): 917-963.  
 [9] Reiser, Patrick, Andre Eberhard, and Pascal Friederich. "Graph neural networks in TensorFlow-Keras with RaggedTensor representation (kgcnn)." Software Impacts (2021): 100095.  
 [10] Kapur, Aayush. "Concepts, Methods and Applications of Neural Style Transfer: A review Article." (2019).  
 [11] Mathew, Amitha, P. Amudha, and S. Sivakumari. "Deep Learning Techniques: An Overview." International Conference on Advanced Machine Learning Technologies and Applications. Springer, Singapore, 2020.  
 [12] Benk, Michaela, and Andrea Ferrario. "Explaining Interpretable Machine Learning: Theory, Methods and Applications." Methods and Applications (December 11, 2020) (2020).  
 [13] Kastrati, Muhamet, and Marenglen Biba. "A State-Of-The-Art Survey on Deep Learning Methods and Applications." International Journal of Computer Science and Information Security (IJCSIS) 19.7 (2021).  
 [14] da Silveira Bohrer, Jonas, Bruno Iochins Grisci, and Marcio Dorn. "Neuroevolution of Neural Network Architectures Using CoDeepNEAT and Keras." arXiv preprint arXiv:2002.04634 (2020).  
 [15] Parisi, Luca, et al. "hyper-sinh: An accurate and reliable function from shallow to deep learning in TensorFlow and Keras." Machine Learning with Applications 6 (2021): 100112.  
 [16] Jo, Jun-Mo. "Performance Comparison Analysis of AI Supervised Learning Methods of Tensorflow and Scikit-Learn in the Writing Digit Data." The Journal of the Korea institute of electronic communication sciences 14.4 (2019): 701-706.  
 [17] Kapur, Aayush. "Concepts, Methods and Applications of Neural Style Transfer: A review Article." (2019).  
 [18] Glauner, Patrick. "Deep learning on big data sets in the cloud with Apache Spark and Google TensorFlow." (2016).  
 [19] Paszke, Adam, et al. "Pytorch: An imperative style, high-performance deep learning library." Advances

- in neural information processing systems 32 (2019): 8026-8037.
- [20] Howard, Jeremy, and Sylvain Gugger. Deep Learning for Coders with fastai and PyTorch. O'Reilly Media, 2020.
- [21] Jatavallabhula, Krishna Murthy, et al. "Kaolin: A pytorch library for accelerating 3d deep learning research." arXiv preprint arXiv:1911.05063 (2019).
- [22] Gao, Xiang, et al. "TorchANI: A free and open source PyTorch-based deep learning implementation of the ANI neural network potentials." Journal of chemical information and modeling 60.7 (2020): 3408-3415.
- [23] Bergstra, James, et al. "Theano: Deep learning on gpus with python." NIPS 2011, BigLearning Workshop, Granada, Spain. Vol. 3. Microtome Publishing., 2011.
- [24] Bahrampour, Soheil, et al. "Comparative study of caffe, neon, theano, and torch for deep learning." (2016).
- [25] Bahrampour, Soheil, et al. "Comparative study of deep learning software frameworks." arXiv preprint arXiv:1511.06435 (2015).
- [26] Kovalev, Vassili, Alexander Kalinovsky, and Sergey Kovalev. "Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?." (2016).
- [27] Bastien, Frédéric, et al. "Theano: new features and speed improvements." arXiv preprint arXiv:1211.5590 (2012).
- [28] Bergstra, James, et al. "Deep learning on GPUs with Theano." The Learning Workshop. 2010.