

М. П. Романчук

ОБҐРУНТУВАННЯ ТИПУ ФРЕЙМВОРКІВ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ОБРОБЛЕННЯ ДАНИХ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ

Важливим завданням у ході обробки даних дистанційного зондування Землі є автоматизація процесу дешифрування аерокосмічних знімків, зокрема виявлення та розпізнавання об'єктів у військовому дешифруванні. У статті розглянуто напрями автоматизації дешифрування знімків та виділено з них перспективний, що ґрунтується на використанні нейронних мереж глибокого навчання. Також проаналізовано технічні завдання, які виникають у ході створення алгоритмів та розгортання навчених моделей на різноманітних мобільних пристроях.

З'ясовано важливу роль програмних каркасів глибокого навчання в процесі тренування моделей нейромереж, метою яких є полегшення розробки та розгортання. Проаналізовано зміни популярності програмних каркасів протягом останніх років та акцентовано на потребі аналізу їх можливостей, що динамічно змінюються. Досліджено найпоширеніші програмні фреймворки для втілення підходів глибокого навчання, їх переваги та недоліки щодо розв'язання завдань тематичного дешифрування на доступних обчислювальних ресурсах. Розглянуто типи обчислювального графа, які використовують програмні каркаси глибокого навчання, та мови програмування, за допомогою яких можна створювати та розгортати моделі нейромереж. Здійснено аналіз фреймворків за обраними критеріями: розподілене виконання, оптимізація архітектури, відображення процесу навчання, спільна підтримка та портативність. У результаті виділено програмний каркас, який слід використовувати для проведення досліджень, та зроблено висновок про переважний фреймворк у промисловому використанні в ході глибокого навчання нейронної мережі для оброблення даних дистанційного зондування Землі.

Ключові слова: машинне навчання; нейронні мережі глибокого навчання; обчислювальний граф; автоматизація дешифрування; аерокосмічні знімки; фреймворк; виявлення об'єктів; дистанційне зондування Землі.

Постановка проблеми в загальному вигляді. Досвід локальних війн та збройних конфліктів останніх десятиліть доводить, що однією з основних тенденцій сучасного розвитку військової справи є інформатизація боротьби. Підвищення рівня інформатизації в державних органах, які здійснюють повноваження у сфері національної безпеки та оборони, передбачається досягти, зокрема, за рахунок моніторингу земної поверхні за результатами дистанційного зондування.

Прийняття рішень посадовими особами залежить від оперативності й достовірності оброблення даних дистанційного зондування Землі (ДЗЗ), а саме дешифрування знімків. Усі завдання дешифрування аерокосмічних знімків можна розділити на дві групи:

- щодо одержання узагальненої інформації про земну поверхню;
- щодо визначення наявності об'єктів на ній та їх характеристик.

© М. П. Романчук, 2019

Залежно від призначення і завдань, розв'язуваних у процесі дешифрування знімків, розрізняють два види дешифрування: загальногеографічне і галузеве. Найважливішим різновидом останнього є військове дешифрування знімків.

Військовим дешифруванням називається процес виявлення, розпізнавання й інтерпретації розташованих на місцевості об'єктів за їх зображеннями на знімках, а також визначення їх кількісних і якісних характеристик [1], що зазвичай здійснюється в ручному режимі дешифрувальником і має низку недоліків, а саме: низьку оперативність дешифрування, високі вимоги до підготовки спеціалістів, а звідси і вартість дешифрування, суб'єктивність тощо. Тому актуальним завданням є здійснення автоматизації процесу дешифрування аерокосмічних знімків.

Аналіз останніх досліджень і публікацій. Об'єкти на аерокосмічних знімках вирізняються за дешифрувальними ознаками, які поділяються на прямі та непрямі [1]. До прямих належить форма, розмір, колір, тон та тінь, а також складно об'єднувальна ознака – малюнок зображення. Непрямими ознаками є розміщення об'єкта, його географічне сусідство, сліди взаємодії з навколишнім середовищем.

Автоматизоване дешифрування знімків передбачає оброблення даних у цифровому вигляді з використанням обчислювальних засобів із відповідним програмним забезпеченням. Таке програмне забезпечення створюється на основі вирішення завдання «розпізнавання образів», що потребує своєрідного «банка пам'яті», де зібрані характеристики природних та штучних об'єктів.

Перспективні автоматичні способи дешифрування аерокосмічних знімків використовують машинне навчання (англ. ML) і нейронні мережі. Застосування останніх ґрунтується на самонавчанні з використанням типових образів об'єктів розпізнавання та принципів багат шаровості. У базі даних знаходяться еталони об'єктів у всіх можливих ситуаціях, а кожен шар нейронної мережі працює із зображенням на своєму рівні абстракції, деякі шари визначають межі об'єктів, що базується на перепадах контрасту, інші – форму, або колір об'єктів тощо. Процес самонавчання мережі формує машинне уявлення об'єкта.

Чим більше вхідних даних – тим з більшою ймовірністю буде розпізнаний об'єкт на знімку, однак важливим питанням є організація процесу навчання нейронної мережі: реалізація алгоритмів на обчислювальних засобах.

Глибоке навчання (англ. DL) [2] – це набір здатних навчатися алгоритмів, що моделюють абстракції високого рівня в даних з використанням архітектури та складаються з декількох нелінійних перетворень. Створення алгоритмів та розгортання навчених моделей є достатньо важким завданням для вчених та інженерів з обробки даних. Для стандартизації процесу розробки програмних продуктів створено низку бібліотек або фреймворків DL, метою яких є полегшення розробки складних систем. Проте існують обмеження, що задають правила створення структури проекту та написання коду. Наявні фреймворки глибокого навчання посилили загальну масштабність можливості досліджень у рамках цієї сфери, здійснили перехід до розробки моделей ML, що вже працюють на мобільних пристроях. Нові фреймворки та методи дозволяють розробити інструменти, що можуть запропонувати кращий рівень абстракції, а також спростити програмування. Можливості фреймворків динамічно змінюються, що потребує їх аналізу.

У табл. 1 наведено порівняння наукових статей в електронному репозиторії arXiv (cs.AI, cs.LG, cs.CV, cs.CL, cs.NE, stat.ML), у яких згадуються фреймворки, використовувані в сфері DL. Порівняння було проведено в повному тексті із загальною кількістю поданих документів, розміщених у січні – вересні 2018 року.

Таблиця 1

Загальна кількість розміщених статей із машинного навчання в arXiv

Фреймворк DL	Розміщено за період	% статей
Tensorflow	131	44,41
Keras	27	9,15
PyTorch	51	17,29
MxNet	27	9,15
Caffe	19	6,44
Caffe2	3	1,02
CNTK	10	3,39
Theano	10	3,39
Torch	10	3,39

Так, 44% усіх документів, поданих у січні – вересні 2018 року, згадують TensorFlow.

На рис. 1 показано, як із часом змінювалася популярність фреймворків (у відсотках) у пошукових запитах Google.

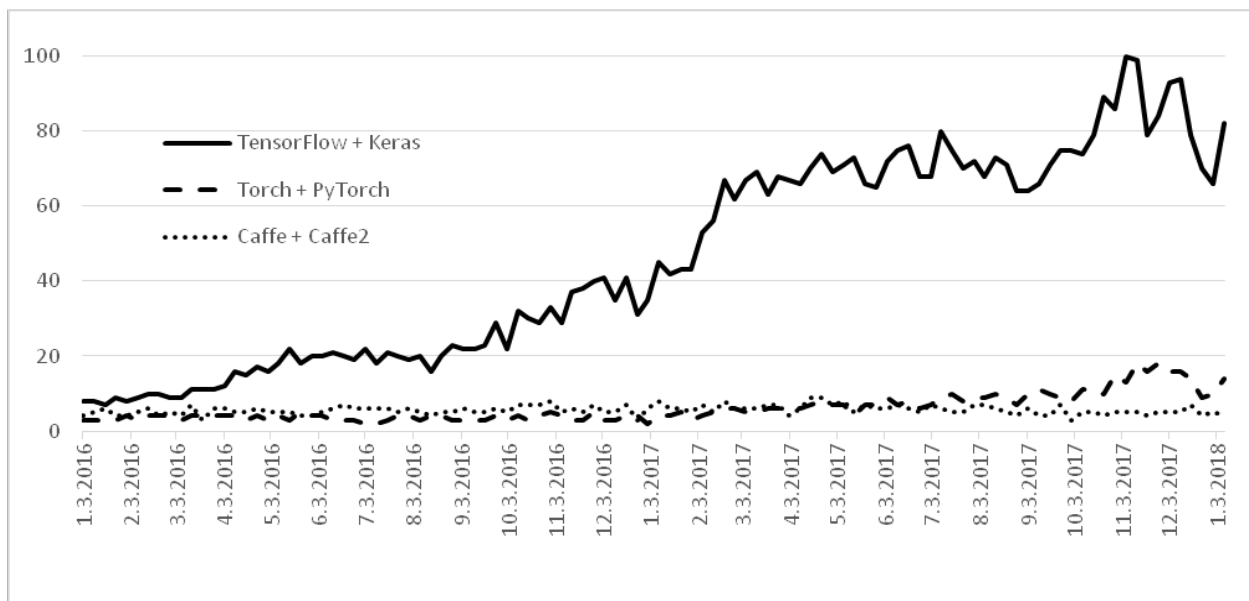


Рис. 1. Динаміка зміни популярності фреймворків

З рис. 1 видно, що Theano широко використовувався великий проміжок часу, але його зростання дещо сповільнилося. Caffe швидко зріс в 2014 році, але останні роки його випередив TensorFlow, який є продовженням Theano. Torch (з недавнього часу його продовженням є PyTorch) також зростає повільно і стабільно.

Формулювання завдання дослідження. Досить складно навіть для досвідчених дослідників та розробників визначити з усього набору фреймворків DL найбільш придатні для вирішення завдань на доступних обчислювальних ресурсах. Кожен фреймворк

побудований іншим способом для різних цілей. Розглянемо основні, щоб дати уявлення про те, які з них будуть кращими для автоматизованого розпізнавання зображень на аерокосмічних знімках.

Отже, метою статті є аналіз переваг та недоліків фреймворків глибокого навчання з метою визначення переважного у завданні виявлення і розпізнавання об'єктів за даними ДЗЗ.

Виклад основного матеріалу. Перш за все необхідно розглянути загальні риси найпоширеніших програмних каркасів.

Фреймворк **TensorFlow** розроблений компанією Google Brain для здійснення досліджень ML та DL. Він використовує статичний обчислювальний граф, який створюється перш ніж модель буде працювати [3–8]. TensorFlow доступний як на стаціонарних комп'ютерах, так і на мобільних пристроях, він дозволяє розробляти моделі за допомогою мов програмування Python, C++ та R, також надає прикладний програмний інтерфейс (англ. API) для їх використання в .NET мовами програмування Java, C, Go, JavaScript. TensorFlow – фреймворк із низьким рівнем програмування, що дозволило на його основі створити низку інших фреймворків, які мають на меті забезпечити шари абстракції високого рівня (Keras, Sonnet, TFLearn тощо). У TensorFlow можна визначити: планувальники, які є вхідними вузлами в обчислювальному графі, наприклад, вхідні дані; змінні – це значення, що існують в обчислювальному графі, наприклад, ваги та зміщення. На сьогодні TensorFlow має обмежену підтримку динамічних входів через Tensorflow Fold.

Фреймворк **Keras**, написаний на Python, є API для побудови нейронних мереж на високому рівні [9]. Його можна використовувати додатково до TensorFlow, Theano або CNTK. Keras призначений для прискорення створення моделі нейромережі. Він здійснює модульне виконання: модель розглядається як послідовність чи як граф автономних, повністю сконфігурованих модулів, які можна з'єднувати разом із мінімальними обмеженнями. Нові модулі легко додаються у вигляді нових класів та функцій.

Фреймворк **PyTorch**, розроблений компанією Facebook, використовує динамічний обчислювальний граф [10], який передбачає побудову нового графа на кожному прохідному етапі. PyTorch має об'єктно-орієнтований підхід, що також дозволяє легко розширювати його функціональні можливості, визначаючи власні класи.

Три рівні абстракції PyTorch полегшують його використання. Тензор у PyTorch – це імперативний nd-масив, подібний до numpy, що має здатність працювати на графічному процесорі (англ. GPU). Змінна є вузлом в обчислювальному графі, який дуже схожий на тензор, змінну і заповнювач у TensorFlow. Модуль – нейронний шар, який може зберігати навчені ваги, що можна використовувати для створення власних класів нейронних мереж.

Фреймворк **MXNet**, розроблений компанією Apache Software Foundation, підтримується компаніями Amazon, Microsoft. Він використовує мови програмування Python та C++, а також JavaScript, R і Go. MXNet здатний масштабувати і працювати з безліччю GPU за допомогою постпроцесорного компілятора, написаного на C++ і Compute Unified Device Architecture (CUDA), що робить його незамінним для промислового використання. Він забезпечує високу продуктивність і відмінну масштабованість, має гарні навички класифікації. Незважаючи на ці переваги, MXNet має невелику сферу застосування та дуже обмежену до використання документацію.

MXNet підтримує мережі з довгою короткотерміною пам'яттю (англ. LSTM) разом із рекурентними (англ. RNN) та згортковими (англ. CNN) нейронними мережами.

У жовтні 2017 року Amazon Web Services та Microsoft випустили новий фреймворк DL, який називається Gluon та є обгорткою MXNet, незабаром він також включатиме CNTK у Microsoft. Розробники стверджують, що інтерфейс Gluon спрощує розробку моделей DL, не відмовляючись від швидкості навчання. Він пропонує простоту кодування та більшу гнучкість, крім динамічних графів і високої продуктивності.

Фреймворк **Caffe** підтримується інтерфейсами C, C++, Python та MATLAB, а також інтерфейсом командного рядка. Він добре відомий своєю швидкістю, масштабованістю і його застосуванням у моделюванні CNN. Перевага від використання C++ у Caffe (постачається з інтерфейсом Python) – це можливість доступу до відкритих мереж DL мережного сховища Caffe Model Zoo, які попередньо підготовлені та можуть бути використані одразу.

Caffe є популярним фреймворком глибокого навчання для обробки зображень. Проте він не підтримує дрібнозернисті мережеві шари, як TensorFlow або CNTK. Інтерфейс Python мало документований, вимагає написання власного коду C++/CUDA для GPU, що не дуже добре для RNN, і є занадто громіздким для великих нейронних мереж. Caffe в даний час широко не застосовується для наукових досліджень, але може бути затребуваним для промислового використання.

Фреймворк **Caffe2**, розроблений Facebook, є наступником Caffe. Це легкий модульний фреймворк, побудований для досягнення успіху на мобільних пристроях й у виробництві. Він використовує статичні графи, подібні до TensorFlow, і має кращий інтерфейс Python порівняно з Caffe. Моделі можна навчати в Python, серіалізувати та потім розгортати без нього. Caffe2 може бути використаний з іншими системами через ONNX, наприклад PyTorch, для ефективного розгортання навчених моделей.

Фреймворк **CNTK**, розроблений компанією Microsoft, використовує статичний обчислювальний граф, подібний до TensorFlow. Він включає в себе CNN та RNN моделі. CNTK пропонує Python API над C++ кодом. Основна перевага фреймворка полягає в можливості легко створювати моделі для програмних продуктів, пов'язаних із вирішенням завдань обробки зображень. За підтримки Microsoft це також дозволяє легко інтегрувати отримані реалізації з Azure Cloud-сервісами.

Є критичні зауваження до CNTK щодо ліцензування, оскільки Microsoft не прийняла такі ліцензії відкритого коду, як: GPL, ASF або MIT. Також відомо, що він забезпечує більш високу продуктивність та масштабованість порівняно з наборами інструментів Theano або TensorFlow під час роботи на декількох обчислювальних машинах.

У порівнянні з Caffe, коли мова йде про розробку нових складних типів шарів, у разі використання CNTK користувачам не потрібно їх реалізовувати мовою низького рівня через відмінну деталізацію будівельних блоків.

Фреймворк **Theano** розроблений у 2007 році Йошуа Бенжіо та дослідницькою групою в Монреальському інституті алгоритмів навчання (MILA), у якому TensorFlow запозичив багато ідей [10, 11]. Theano є бібліотекою Python, надзвичайно швидкою та потужною, але його критикують за те, що це глибоке навчальне середовище низького рівня. Активна розробка в рамках програми була припинена з версії 1.0. Група компаній Theano оголосила в 2017 році, що після оприлюднення останньої версії подальшого розвитку не буде.

Фреймворк **Torch** пропонує широку підтримку алгоритмів ML. До введення TensorFlow і PyTorch він був головним конкурентом Theano. Даний фреймворк має основні переваги, характерні PyTorch: використання динамічних графів та розширення можливостей. Хоча він краще розвинутий порівняно з PyTorch, один з основних недоліків його полягає у використанні мови програмування Lua. Torch використовує CUDA разом із бібліотеками C/C++ і в основному був створений для масштабування розроблених моделей та забезпечення загальної гнучкості.

Розглянемо критерії відбору, за якими буде здійснено порівняння та аналіз основних фреймворків.

1. Розподілене виконання як спосіб розв'язання трудомістких обчислювальних завдань із використанням двох і більше комп'ютерів, об'єднаних у мережу.

Розподілене виконання підтримується більшістю фреймворків: TensorFlow та PyTorch побудовані з його використанням, обчислювальний граф може бути розподілений у кластері. CNTK має розподілений модуль для використання декількох графічних процесорів та машин, однак, як згадувалося раніше, існують деякі застереження щодо ліцензування. MXNet також може розподіляти навчання, проте відстає від інших фреймворків, у документації відсутній достатній опис. Caffe2 включає в себе вбудоване розподілене навчання, яке використовує Gloo. Torch, Caffe і Theano не мають вбудованої підтримки для даного виконання завдань.

2. Оптимізація архітектури фреймворка. Усі фреймворки оптимізовані для роботи на центральному процесорі (англ. CPU або GPU), більшість з них мають можливість переключатися між процесорами.

У TensorFlow існують окремі пакети для версій, що підтримуються центральним і графічним процесорами. TensorFlow намагається з'ясувати, який пристрій використовувати залежно від операції та наявних ресурсів, а також у якому об'ємі.

У PyTorch обидві версії, що підтримуються центральним і графічним процесорами, знаходяться в одному пакеті, тому необхідно чітко визначити, на якому процесорі він повинен працювати. Існує можливість легко переносити змінні на пристрої та з них. Наприклад, якщо в GPU визначено тензор, тоді будь-які наступні операції на ньому будуть відбуватися на GPU, поки тензор не буде переданий назад CPU.

Більшість фреймворків використовують GPU Nvidia і покладаються на бібліотеки CUDA і cudNN для виконання таких обчислень, як згортки та операції над матрицями. Продуктивність може відрізнятися залежно від інших чинників, наприклад розміру мережі.

Через проблеми з продуктивністю OpenGL більшість фреймворків не мають жодної підтримки для GPU за межами Nvidia. Спостерігається постійний прогрес для підтримки більшості фреймворків для обчислень на CPU, зокрема HIP від компанії AMD і MKL-DNN від компанії Intel.

3. Візуалізація. TensorFlow постачається з комплектом інструментів візуалізації, який називається TensorBoard, що дозволяє відображати тренувальний процес, конвергенцію тощо. Він використовує результати обчислень, які записуються на диск. Коли проводиться експеримент, то зберігаються підсумкові дані для певного запуску з диска. TensorBoard

може потім візуалізувати ці дані, наприклад, ви можете виконати кілька тестів для пошуку гіперпараметрів, запустити TensorBoard і одночасно порівнювати всі моделі. Це можливо робити це під час процесу навчання, оскільки TensorBoard періодично оновлюється. Отже, можна повторно візуалізувати будь-який процес будь-коли, якщо зберігати дані. TensorFlow та Keras також надається можливість використовувати метод history та стандартну бібліотеку matplotlib для аналізу результатів обчислень.

PyTorch і Torch можуть використовувати пакет visdom із Facebook для візуалізації. Інші фреймворки забезпечують мінімальні інструменти візуалізації або покладаються на бібліотеки з відкритими кодами, наприклад, graphviz та matplotlib.

Оскільки TensorBoard є бібліотекою з відкритим вихідним кодом, вона дозволяє використовувати будь-які фреймворки, у тому числі PyTorch і CNTK.

4. Спільна підтримка. TensorFlow переважає всі фреймворки, на думку розробників, оскільки він підтримується компанією Google, що полегшує доступ до його використання через сервіс Google Cloud TPU, Colaboratory. Навколо TensorFlow утворилася велика спільнота розробників із різними фреймворками, наприклад, Keras, що полегшують розробку в TensorFlow. Він використовується компанією Google у своїх продуктах та її дослідницьких групах: Google Brain і DeepMind, а також Airbnb, Dropbox, SAP, eBay тощо. Даний фреймворк також має найбільш повну документацію, підручники та книги.

PyTorch в основному застосовується дослідниками завдяки згаданим вище перевагам. Він розробляється та використовується Facebook, Twitter, NVIDIA, Salesforce, Uber, Stanford, CMU, NYU та іншими. Хоча в нього менша спільнота порівняно з TensorFlow, він має активну дошку обговорень Slack. PyTorch також має детальну документацію, офіційні підручники та книги.

CNTK підтримує переважно Microsoft. Загалом його використовують для розгортання розробниками додатків у середовищі Windows, які хотіли б включити моделі машинного навчання в прикладні та мобільні додатки.

MXNet має дошку для обговорень та список розсилки для спільноти. На сьогоднішній день Amazon значно підтримав його та інтегрував в AWS. Існує також деяка підтримка від Microsoft із введенням надбудови високого рівня Gluon, оскільки вона підтримуватиме CNTK як серверний додаток. На сьогодні покращується ситуація з документацією та підручниками.

5. Портативність. Усі фреймворки мають можливість зберігати та завантажувати конфігурацію моделі, а також вивчені параметри. Це можна зробити за допомогою Open Neural Network Exchange (ONNX), де збережена модель може бути використана в іншій структурі для висновку. Наприклад, моделі PyTorch можуть бути збережені за допомогою ONNX для розгортання на Caffe2 або CNTK. Це також дозволяє інтегрувати їх на мобільні пристрої. ONNX використовується за замовчуванням у PyTorch, Caffe2, CNTK та MXNet. Хоча TensorFlow ще не підтримує ONNX, конвертери з відкритим кодом доступні для того, щоб застосувати деяку частину цієї функції до моделі TensorFlow.

Проведене порівняння фреймворків глибокого навчання відображено в табл. 2, де наявність і кількість хрестиків позначає більш високий рівень визначеного критерію у фреймворка.

Порівняння фреймворків глибокого навчання

Критерій Фреймворк	Розподілене виконання	Оптимізація архітектури	Візуалізація	Спільна підтримка	Портативність
TensorFlow	XX	XX	XX	XX	XX
Keras	X	XX	XX	XX	XX
PyTorch	XX	XX	XX	XX	XX
CNTK	XX	XX	X	-	XX
MXNet	X	XX	X	-	XX
Torch	-	XX	X	X	X
Caffe2	XX	XX	-	-	XX
Caffe	-	XX	X	X	X
Theano	-	XX	X	X	X

Результати аналізу характеристик найпоширеніших фреймворків показали, що недоцільно розпочинати нові проекти на базі Torch, Theano та Caffe, тому що вони вже не оновлюються. Це стосується не функціонала та продуктивності фреймворків, а лише відсутності активного розвитку й підтримки. У зв'язку з чим розробники використовують TensorFlow, PyTorch, MXNet, CNTK і Caffe2, хоча існують застереження щодо CNTK.

Включення сумісності дає змогу швидше отримувати реалізації у виробництві. ONNX дозволяє моделювати навчання в одній системі та переносити на іншу для проведення аналізу. Моделі ONNX у даний час підтримуються в Caffe2, CNTK, MXNet та PyTorch, а також є можливість їх підключення для багатьох інших загальних структур і бібліотек. Це дозволяє користувачам більш легко переміщувати моделі між різними фреймворками.

PyTorch і TensorFlow найкраще підходять для процесу розробки та загального дослідження на низькому рівні. Вони є основою глибокого навчання із застосуванням Python. PyTorch краще підходить для проведення досліджень, оскільки простий у використанні та ґрунтується на динамічному обчислювальному графі. Переваги TensorFlow, в основному, пов'язані з його широким застосуванням, підтримкою спільнотою розробників, що дозволяє створювати програмний код для промислового використання.

Тому в системі автоматизованого дешифрування аерокосмічних знімків з метою виявлення та розпізнавання об'єктів доцільно використовувати TensorFlow як фреймворк глибокого навчання.

Висновки. Розвиток інформаційних технологій дозволяє значно збільшити кількість і якість інформації, яка використовується в інтересах сектора безпеки держави. Однак оперативність її надходження потребує автоматизації оброблення даних моніторингу, у тому числі й військового дешифрування даних ДЗЗ.

У процесі автоматизованого дешифрування аерознімків з метою виявлення та розпізнавання точкових об'єктів застосовують елементи штучного інтелекту

з використанням моделей нейромереж глибокого навчання. У процесі тренування моделей важливу роль відіграють фреймворки (програмні каркаси) глибокого навчання, метою яких є полегшення розробки та розгортання. Для побудови ефективної автоматизованої системи дешифрування необхідно обґрунтувати вибір типу фреймворка. Порівняння їх властивостей здійснювалося за такими критеріями: розподілене виконання, оптимізація архітектури, візуалізація, спільна підтримка та портативність.

Результати аналізу характеристик програмних каркасів за визначеними критеріями показали, що переважним є використання фреймворка TensorFlow для навчання нейронної мережі з метою подальшого оброблення даних ДЗЗ.

СПИСОК ЛІТЕРАТУРИ

1. Карпович И. Н. Военное дешифрирование аэроснимков : Учебник. Москва : Воениздат, 1990. 544 с.
2. Гудфеллоу Я., Бенджио Й., Курвиль А. Глубокое обучение. Москва : ДМК Пресс, 2018. 652 с.
3. TensorFlow: Large-scale machine learning on heterogeneous systems / Martin Abadi, Ashish Agarwal, Paul Barham et al. URL: <https://arxiv.org/abs/1603.04467> (last accessed: 20.01.2019).
4. Hope T., Yehezkel S., Lieder I. Learning TensorFlow. Sebastopol : O'Reilly Media, 2017. 242 p.
5. McClure N. TensorFlow Machine Learning Cookbook. Birmingham : Packt Publishing Ltd, 2017. 370 p.
6. Warden P. Building Mobile Applications with TensorFlow. Birmingham : Packt Publishing Ltd, 2017. 238 p.
7. Géron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow. Sebastopol : O'Reilly Media, 2017. 751 p.
8. Karim R. TensorFlow: Powerful Predictive Analytics with TensorFlow. Birmingham : Packt Publishing Ltd, 2018. 165 p.
9. Джулли А., Пал С. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow. Москва : ДМК Пресс, 2018. 294 с.
10. Deep learning with dynamic computation graphs / Moshe Looks, Marcello Herreshoff, DeLesley Hutchins & Peter Norvig Google Inc. URL: <https://arxiv.org/abs/1702.02181> (last accessed: 20.01.2019).
11. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. URL: <https://arxiv.org/abs/1605.02688> (last accessed: 20.01.2019).
12. Рашка С. Python и машинное обучение. Москва : ДМК Пресс, 2017. 418 с.
13. Bengio Y. Learning deep architectures for AI. Foundations and trends in Machine Learning. 2009. № 1. P. 1–127.

Подано 29.03.2019

Н. П. Романчук

ОБОСНОВАНИЕ ТИПА ФРЕЙМВОРКОВ ГЛУБОКОГО ОБУЧЕНИЯ ПРИ ОБРАБОТКЕ ДАННЫХ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ

Важной задачей в ходе обработки данных дистанционного зондирования Земли является автоматизация процесса дешифрирования аэрокосмических снимков, в том числе выявление и распознавание объектов в военном дешифрировании. В статье рассмотрены направления автоматизации дешифрирования снимков и выделено из них перспективное, основанное на использовании нейронных сетей глубокого обучения. Также

проанализированы технические задачи, возникающие при создании алгоритмов и развертывании обученных моделей на различных мобильных устройствах.

Отмечена важная роль программных каркасов глубокого обучения в процессе тренировки моделей нейросетей, целью которых является облегчение разработки и развертывания. Проанализированы изменения популярности программных каркасов в последние годы и акцентировано внимание на необходимости анализа их динамично изменяющихся возможностей. Исследованы распространенные программные фреймворки для воплощения подходов глубокого обучения, их преимущества и недостатки по решению задач тематического дешифрирования на доступных вычислительных ресурсах. Рассмотрены типы вычислительного графа, использующие программные каркасы глубокого обучения, и языки программирования, с помощью которых можно создавать и развертывать модели нейронных сетей. Осуществлен анализ фреймворков по выбранным критериям: распределенное выполнение, оптимизация архитектуры, отражение процесса обучения, совместная поддержка и портативность. В результате выделен фреймворк, который следует использовать при проведении исследований, и сделан вывод о преимущественном фреймворке в промышленном использовании в ходе глубокого обучения нейронной сети для обработки данных дистанционного зондирования Земли.

Ключевые слова: *машинное обучение; нейронные сети глубокого обучения; вычислительный граф; автоматизация дешифрирования; аэрокосмические снимки; фреймворк; выявление объектов; дистанционное зондирование Земли.*

M. P. Romanchuk

THE REASON OF THE FIREWORK TYPE OF DEPTH EDUCATION IN PROCESSING THE DATA OF REMOTE SURFACES OF THE EARTH

An important task in the processing of Earth remote sensing data is the automation of the decoding process of aerospace images, in particular the detection and recognition of objects in military decoding. In the article the directions of automation of decryption of photos are considered and promising from them is selected, which is based on the use of neural networks of deep learning, and also analyzed the technical problems that arise during the creation of algorithms and the deployment of trained models on a variety of mobile devices.

The important role of deep-instruction software frameworks in the process of training of neural network models is aimed at facilitating development and deployment. The changes in the popularity of software frameworks in recent years have been analyzed and the need to analyze their dynamically changing capabilities has been analyzed. The most widely used software frameworks for the implementation of deep learning approaches, their advantages and disadvantages for solving tasks of thematic decryption on accessible computational resources are explored. The types of computational graphs, which use the software of deep learning, and programming languages, with the help of which it is allowed to create and deploy models of neural networks are considered. The analysis of the frameworks according to selected criteria was performed: distributed execution, architecture optimization, reflection of the learning process, joint support and portability. As a result, the software framework to be used in conducting research is highlighted, and the conclusion is drawn about the predominant framework for industrial use in the course of in-depth training of the neural network for the processing of Earth remote sensing data.

Keywords: *machine learning; neural networks of deep learning; computing graph, decoding automation; aerospace images; frameworks; object detection; remote sensing of the Earth.*