



Сеть готова, что дальше?

Лекция, Дмитрий Яценко.

14.04.2021

Создание алгоритма использующего ИНС. Цель и задачи.

- прикладная цель — решение конкретной технической проблемы с характеристиками решения не хуже существующих аналогов.
- демонстрация — например:
 - концепт демонстрирующий потенциально возможное продуктивное решение
 - концепт демонстрирующий скилы создателя, учебные цели
 - научная работа, лишь демонстрация идей или возможностей, концепт не обязателен.

Достижение

Вспомним, что работа с НС это learning/inference, соответственно и работа с сетью разделяется на две части. Во всех случаях кроме последнего (рассмотрим чуть позже) требуется реализация двух разных решений:

- (1) разработка алгоритма ИНС, решающего поставленную задачу со всеми сопутствующими условиями см. первую лекцию ,
- (2) разработка прикладного приложения, соответствующего требованиям к ПО.

1й этап

Состоит из:

- проектирования компонентов нейронной сети: Архитектура НН, Функция потерь, Метод оптимизации, Метрики
- реализации сети и ее обучении: выбор библиотек, создание пайплайна, выполнение процесса обучения на пайплайне, анализ полученных результатов, корректировка сети/рассмотрение новых гипотез [1]
- фиксации результатов в виде документации, в виде опубликованного проекта, экспорт сети для дальнейшего использования

Примечание 1: первая стадия при реализации научной работы должна иметь признак научной новизны, теоретической значимости исследования, и должна удовлетворять критериям научности.

Примечание 2: первая стадия при реализации образовательной работы (методичка, пособие, учебный курс, демонстрационные проекты) в основном служит для демонстраций алгоритмов НС.

2й этап

Состоит из:

- анализа постановки прикладной задачи и выбор технологической платформы для реализации
- проектирование и разработка ПО использующего сериализованную НС для инференса по фактическим данным.
- имплементация прикладной системы как ПО и/или библиотек/фреймворков и/или сервисов и/или программно аппаратных комплексов

Примечание 1: Вторая стадия для случая научной работы — публикация статьи в научном журнале, участие в конференции.

Примечание 2: Вторая стадия для случая образовательной работы — публикация и внедрение образовательного материала.

Пример

Пример реализации цели «Индивидуальный проект IT Академия Samsung трек ИИ определение пожара на фотографии»

Поскольку эта цель подпадает под (портфолио, учебные цели) задачи:

- получить датасет, спроектировать, реализовать, обучить НС, анализ и отчет,
- спроектировать, реализовать, опубликовать концепт.

Пример (1)

Датасет.

Получаем датасет со следующими параметрами:

- две части train(80%)/val(20%)
- два класса фотографий fire(91)/nofire(138)
- состав — фото из разных источников 3872x2592 — 300x214

Вывод:

- датасет слишком маленький для полноценного обучения,
- датасет скошенный,
- требуется предобработка фотографий — размер, обогащение.



Пример (1)

Сеть. Компоненты:

- Т.к. датасет маленький, используем transfer-learning - берем предобученную сеть Resnet18 (обучен на ImageNet ~15M) без fine-tuning
- К выходу сверточных слоев ResNet (признаковое описание ImageNet) подключаем полносвязный слой 512x2
- Некоторый трюк - CrossEntropyLoss
- SGD
- Метрики — Loss и Accuracy

Пример (1)

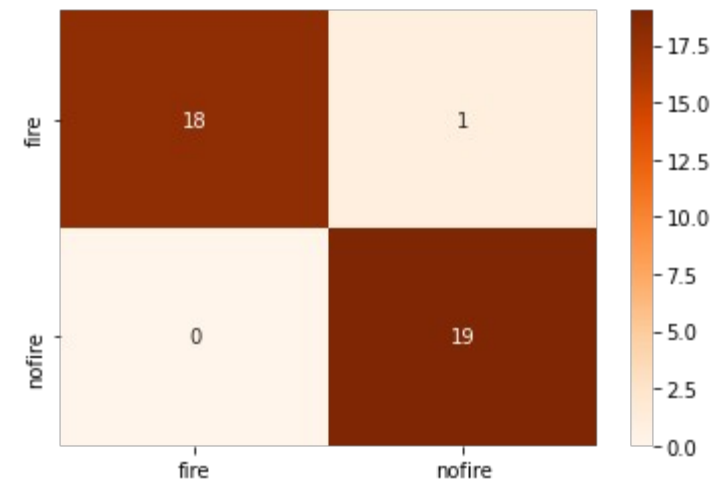
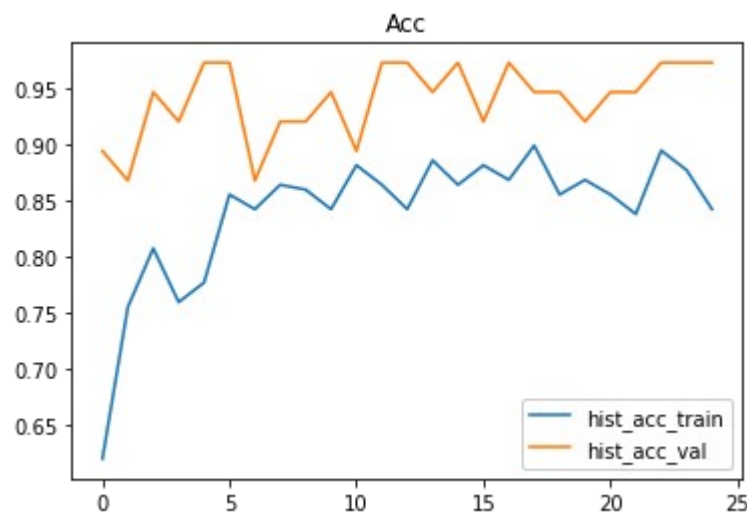
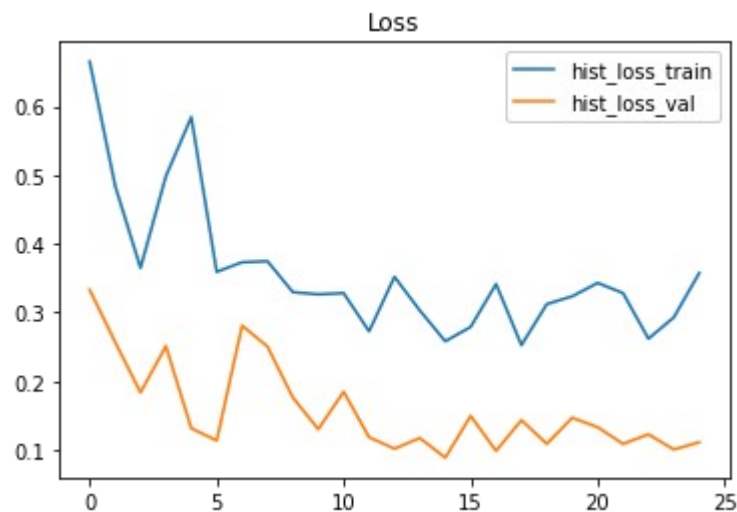
Реализация.

- пайплайна фактически нет, т.к. предполагаем что угадаем сеть с первого раза.
- среда исполнения ноутбук Google Colab,
- фреймворк pytorch
- предобученная модель — `torchvision.models.resnet18`

Демонстрация, приложение 1

Пример (1)

Training complete in 2m 0s
Best epoch:14 val Loss:0.088741 Acc: 0.973684



Второй этап

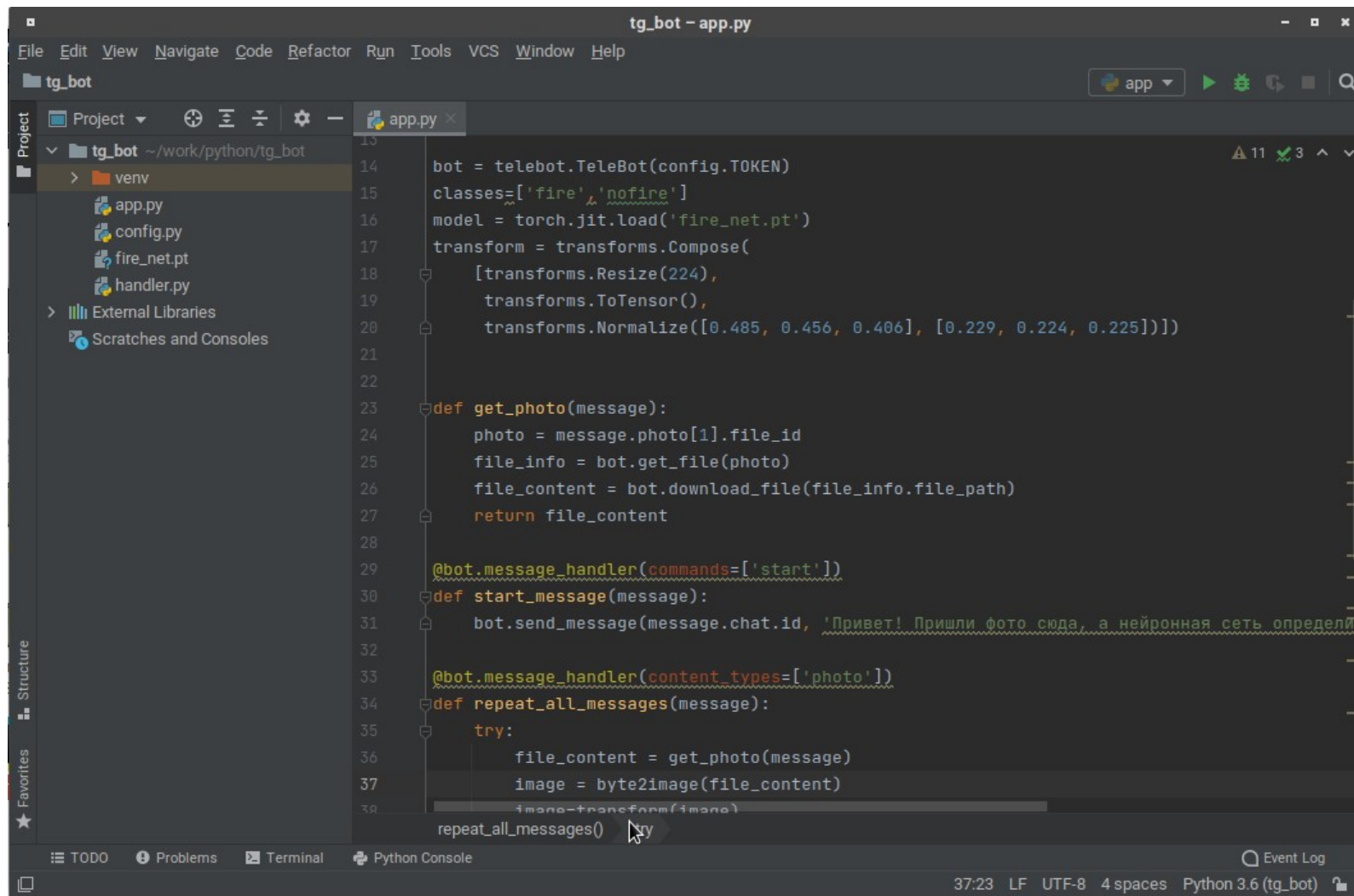
Анализ постановки прикладной задачи:

- Если задача научная или учебная, на этой стадии оформляют весь собранный на предыдущих этапах материал в виде статьи/метод.материала с учетом требований. В качестве демонстрации можно рассмотреть любую статью в научном журнале [3]
- Если речь о концепте необходимо продумать форму удобной для представления, но с учетом минимальной функциональности и минимальной трудоемкости. Например чат-бот телеграмм.
- Если требуется разработка прикладного решения нужно рассмотреть требования к решению и параметры результата, получившегося на первом этапе :
 - параметры инференса на полученной сети — время на один цикл,
 - среда исполнения,
 - требования по скорости инференса,
 - требования по оперативной памяти,
 - по системе вычислений (TOPS/FLOPS),
 - по программной платформе,
 - архитектура целевой системы,
 - характеристики потока данных для инференса
 - требования безопасности.

Синтез. По результатам анализа выбирается архитектура прикладного решения, используемые фреймворки и выполняется уже непосредственно проектирование АО и его разработка.

Пример (2)

Демонстрация, Приложение 2.



```
tg_bot - app.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help

Project
  tg_bot ~/work/python/tg_bot
    app.py
    config.py
    fire_net.pt
    handler.py
  External Libraries
  Scratches and Consoles

Structure
  app.py
    13
    14 bot = telebot.TeleBot(config.TOKEN)
    15 classes=['fire','nofire']
    16 model = torch.jit.load('fire_net.pt')
    17 transform = transforms.Compose(
    18     [transforms.Resize(224),
    19      transforms.ToTensor(),
    20      transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])])
    21
    22
    23 def get_photo(message):
    24     photo = message.photo[1].file_id
    25     file_info = bot.get_file(photo)
    26     file_content = bot.download_file(file_info.file_path)
    27     return file_content
    28
    29 @bot.message_handler(commands=['start'])
    30 def start_message(message):
    31     bot.send_message(message.chat.id, 'Привет! Пришли фото сюда, а нейронная сеть определит')
    32
    33 @bot.message_handler(content_types=['photo'])
    34 def repeat_all_messages(message):
    35     try:
    36         file_content = get_photo(message)
    37         image = byte2image(file_content)
    38         image = transform(image)
    39         repeat_all_messages()
```



Пример (2)

2) Приложение для конечного пользователя, оценка одежды на человеке.

Анализ:

- приложение не требует реального времени работы,
- скорее всего пользователь будет не рад отправки фотографий на сервер,
- поток данных для инференса низкий,
- предположительное место работы — сотовый телефон.

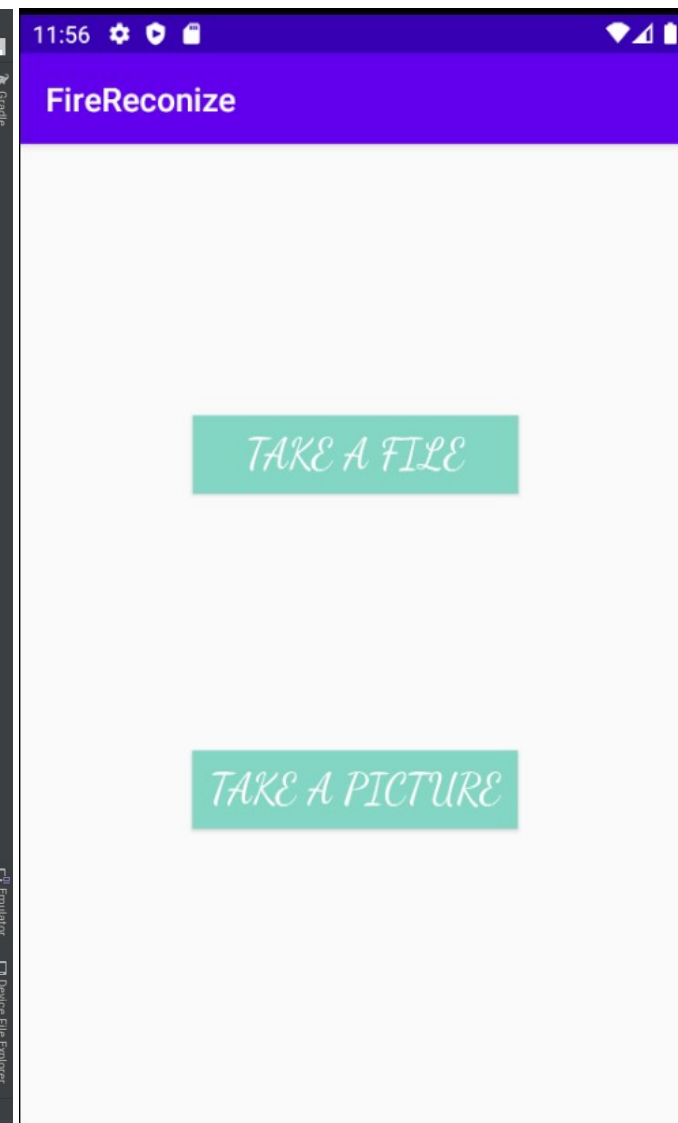
Предлагаемое решение:

- создать мобильное приложение,
- встроить полученную на первом этапе ИНС в приложение для инференса,
- поскольку время на инференс не ограничено, этот параметр скорее всего не критичен,
- если размер сети слишком велик (порядка 100M и выше), то применить квантизацию [2].

Пример (2)

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
recognize app src main java com samsung itschool firereconize Classifier predict app Pixel 2 API 30
1: Project
  app
    manifests
    java
      com.samsung.itschool
        Classifier
        MainActivity
        Utils
      com.samsung.itschool
      com.samsung.itschool
    java (generated)
    assets
    res
    res (generated)
  Gradle Scripts
    build.gradle (Project: FireRe
    build.gradle (Module: FireRe
    gradle-wrapper.properties
    proguard-rules.pro (ProGua
    gradle.properties (Project P
    settings.gradle (Project Set
    local.properties (SDK Local
  Resource Manager
  Build Variants
  Favorites
  Structure
  Logcat
  Database Inspector
  Profiler
  TODO
  Terminal
  Git

12 public class Classifier {
13     public static final int IMG_SIZE=224;
14     public static final String[] IMAGENET_CLASSES={"fire","nofire"};
15     public static final float[] mean = {0.485f, 0.456f, 0.406f};
16     public static final float[] std = {0.229f, 0.224f, 0.225f};
17     Module model;
18
19
20
21     public Classifier(String modelPath) { model = Module.load(modelPath); }
22
23     // приведение размера картинки и конвертация ее в тензор
24     public Tensor preprocess(Bitmap bitmap, int size){
25         bitmap = Bitmap.createScaledBitmap(bitmap,size,size,filter: false);
26         return TensorImageUtils.bitmapToFloat32Tensor(bitmap,this.mean,this.std);
27     }
28
29     // найти номер максимального элемента в массиве
30     public int argMax(float[] inputs){
31         int maxIndex = -1;
32         float maxvalue = 0.0f;
33         for (int i = 0; i < inputs.length; i++){
34             if(inputs[i] > maxvalue) {
35                 maxIndex = i;
36                 maxvalue = inputs[i];
37             }
38         }
39         return maxIndex;
40     }
41
42     // использование HC
43     public String predict(Bitmap bitmap){
44         Tensor tensor = preprocess(bitmap,IMG_SIZE);
45         IValue inputs = IValue.from(tensor);
46         Tensor outputs = model.forward(inputs).toTensor();
47         float[] scores = outputs.getDataAsFloatArray();
48         int classIndex = argMax(scores);
49         return IMAGENET_CLASSES[classIndex];
50     }
51 }
52
53
54
Android Studio 4.1.3 available
Update...
```



Пример (2)

3) Приложение для крупной организации, например распознавание fraudulent транзакций или фишинга.

Анализ:

- записей в крупной организации очень много — от 1млн и больше,
- данные обычно хранятся в DWH,
- требование по скорости инференса — ASAP,
- крупные организации обычно не сильно стеснены в объемах памяти и специализированных вычислителях,
- данные для инференса поставляются из БД,
- требования безопасности максимальные — закрытый информационный контур.

Предлагаемое решение:

- то время обработки, то скорость одного инференса критична, т.е. требуются все возможные ухищрения по повышению скорости обработки — дистилляция, квантизация, перевод на специализированные вычислители (миллионы записей!),
- можно получать данные от DWH через какой либо коннектор в сервер инференса в контуре безопасности, а результаты загружать обратно,
- а можно, если хранилище данных построено по архитектуре MPP (Greenplum/ADB) и имеет встроенную среду исполнения программ, выполнять инференс прямо на узлах MPP, что увеличит скорость еще в сотни раз.
- Кстати возможно использовать этот же MPP кластер и для быстрого обучения [4],[5].

Пример (2)

4) Распределенная система видеонаблюдения/охранная система.

Анализ:

- большое количество камер наблюдения направлены на территорию и предназначены для уведомления о проникновении,
- уведомление должно быть близким к реалтайму ($<1\text{сек}$)
- требования по безопасности не допускают передачу данных за пределы контура охраны,
- в местах размещения камер нельзя установить стационарное оборудование,
- поскольку камеры не обладают вычислительными возможностями, архитектура системы естественным образом будет клиент-серверной в рамках контура безопасности,
- серьезных ограничений на вычислительные ресурсы нет.

Предлагаемое решение:

- можно видеопоток от всех камер свести на один сервер и делать там инференс, для обнаружения проникновения нарушителей, правда нагрузка на сеть передачи данных и особенно на сервер будет большая,
- альтернативно, можно воспользоваться архитектурой EDGE и установить небольшие (и недорогие) вычислители например Rpi или Nvidia Jetson (квантизация!), на которых вести инференс, а результаты отправлять на сервер, тем самым снимая нагрузку [7].
- можно усовершенствовать решение выше разделив инференс на две стадии – на EDGE выделение карты признаков (например pretrained resnet) и передача на сервер, а на сервере сеть делает вторую фазу инференса. Почитать про двухстадийный инференс можно тут [6].

Пример (2)

5) Системы IoT — десятки датчиков, из тысяч местоположений передают информацию в аналитический центр, для построения статотчетов и проактивного анализа.

Анализ:

- Данные от датчиков по протоколу MQTT или по какому либо другому (иногда кастомному) через череду систем — брокеры (возможно EDGE Mosquito), стриминговые сервисы (NiFi/ADS), ELT/ETL загружаются в DWH, где над аккумулярованными данными производятся аналитические операции — нормализация, агрегация и подготовка операционного слоя данных и витрин данных,
- Витрины данных выводятся заказчикам информации (отделы аналитики, клиентский веб сайт) а в операционном слое работает проактивный анализ.

Предлагаемое решение:

- На операционном слое, как и в примере (2) можно применить ИНС, например как функцию на ЯП встроенном в СУБД,
- возможно в некоторых случаях проводить первую стадию инференса на EDGE, как в примере (3)

Вывод

Как и всегда разработка наукоемких технических решений требует опыт из различных областей деятельности. Ошибкой было бы думать, что требуется только навыки проектирования ИНС. Кроме этого требуются умение проектирования и разработки ПО, навыки системной интеграции, а также опыт внедрения и обслуживания ИС.

- Приложение 1.

Алгоритм ИНС. https://github.com/d-yacenko/NN_to_production_impl/blob/main/Annex1_fire_fc.ipynb

- Приложение 2.

Пример чат-бота. https://github.com/d-yacenko/NN_to_production_impl/tree/main/tg_bot

- Приложение 3.

Пример андроид приложения.

https://github.com/d-yacenko/NN_to_production_impl/tree/main/FireRecognize

1. Роман Суворов. Как эффективно проводить эксперименты: базовая структура проекта, процесс перебора гипотез, трюки для обучения нейросетей [Электронный ресурс]/Роман Суворов, 2020г - https://www.youtube.com/watch?v=RS_U6qodpsc&t=1s
2. Алексей Ивахненко. Мобильные архитектуры нейросетей и фреймворки для их запуска [Электронный ресурс]/ Алексей Ивахненко, 2020г - <https://www.youtube.com/watch?v=ASChrJhj-zY&t=11s>
3. TF-IDF vs Word Embeddings for Morbidity Identification in Clinical Notes: An Initial Study/
Danilo Dess, Rim Helaoui, Vivek Kumar¹, Diego Reforgiato Recupero and Daniele Riboni
University of Cagliari, Cagliari, Italy{danilo dessi, vivek.kumar, diego.reforgiato, riboni}@unica.it
Philips Research, Eindhoven, Netherlands rim.helaoui@philips.com
4. Michiel Shortt. AI/ML, Neural Networks & the future of analytics: Using Greenplum and PL/Python [Электронный ресурс]/Michiel Shortt , 2020г. - <https://www.youtube.com/watch?v=6ayC8eOamkw&list=PL4duir3J-8GW-P1RXN67IgXMS3LoHXG6u&index=2> , A42 Labs.
5. Bastiaan Sjardin. Large Scale Machine Learning with Python/Bastiaan Sjardin, Luca Massaron, Alberto Boschetti — Packt Publishing,2016
6. CATHERINE SANDOVAL, Two-Stage Deep Learning Approach to the Classification of Fine-Art Paintings / CATHERINE SANDOVAL , ELENA PIROGOVA, MARGARET LECH - School of Engineering, RMIT University, Melbourne, VIC 3000, Australia, 2019
7. Ультимативное сравнение embedded платформ для AI [Электронный ресурс]/ZlodeiBaal, 2019г - <https://habr.com/ru/company/recognitor/blog/468421/>



SAMSUNG

