



КУМПАН ВИКТОР ВИКТОРОВИЧ

Junior Deep learning / Data science / Software engineering

ЦЕЛЬ

Совершенствование, развитие профессиональных компетенций и навыков командной разработки продукта. Внести вклад в развитие вашей компании!

ОБРАЗОВАНИЕ

НИУ ИТМО Г. САНКТ-ПЕТЕРБУРГ • 2019-2023 БАКАЛАВРИАТ

Факультет информационных технологий и программирования ФИТиП, направление подготовки прикладная математика и программирование. Средний балл 4.9. В рейтинге студентов в 5 лучших (за весь период обучения).

НАВЫКИ

- Владение английским B2 (Upper Intermediate)
- Фундаментальные знания математического и функционального анализа, а также дискретной математики.
- Знание методологии DevOps - CI/CD. Опыт работы с Docker, GIT.
- Уверенные знания Алгоритмов и STL, умение оценить асимптотику решения.
- Хорошие знания Python, C++, ООП и Функционального программирования.
- Хорошие знания PyTorch, Transformers, NumPy, Scikit-learn.
- Опыт работы с данными: Pandas, Matplotlib, Seaborn
- Хорошие знания SQL и опыт проектирование БД

ОПЫТ ПРОЕКТНОЙ РАБОТЫ

YANDEXCUP COMPUTER VISION • СОПОСТАВЛЕНИЕ ИЗОБРАЖЕНИЙ К НАИБОЛЕЕ РЕЛЕВАНТНОМУ ТЕКСТОВОМУ ОПИСАНИЮ / ZERO-SHOT.

В рамках турнира предоставлен датасет из ссылок на изображения (5 млн) и соответствующим им поисковым запросам (>20 млн). Написано простое MLOps решение для:

- Многопоточного скачивания, обработки (аугментации) данных, анализа текстовых данных, кластеризации текстовых описаний и визуализации embeddings
- Обучения моделей I2T zero-shot classification, создание моделей кластеризации (Train: BoW -> SentenceTransformer -> AgglomerativeClustering -> KNeighborsClassifier; Inf: SentenceTransformer -> KNeighborsClassifier)
- Оценка качества моделей, легирование (tensorboard)
- Inference

За baseline взят CLIP by Sber (visual_encoder - VIT-B32; text_encoder - GPT3Small) так как он влез в ограничения по весу модели. С коробки accuracy на private 42%, также пробовал заменить GPT3Small -> RuBert; GPT3Small -> BPE embeddings (предоставленные организаторами) + 4 transformers block. Финальным решением было приделать к baseline параллельный классификатор [1, N], класс которого соответствовал номеру projections стоявшего на выходе text_encoder (тренировка End-to-End). Это позволило поднять точность на семантически близких описаниях. В итоге accuracy на private возросло до 58%. Более подробно на github.



@SKYLARPRO



SKYLARPRO



+79508998389



ИТМО

- Опыт работы с облачными системами AWS, Azure, OpenStack.
- Умение стремительно развиваться в высококвалифицированном окружении
- Способность свободно коммуницировать с людьми и продуктивно работать в команде
- Понимание принципов ведения проектов Kanban, Scrum, Agile.

AI JOURNEY 16 МЕСТО (БРОНЗА) • AI TRAIN ДЕТЕКЦИЯ, СЕГМЕНТАЦИЯ ОБЪЕКТОВ НА ЖД-ИНФРАСТРУКТУРЕ

В рамках турнира предоставлен датасет из 11 классов детекции и 3-х классов для сегментации. Проведен статистический анализ данных. Для детекции объектов был взят YOLOV5 (bb – EfficientNet; Neck – Bi-FPN; Head - Conv), выполнена легкая аугментация данных, так как некоторые классы оказались чувствительны. Выполнена тонкая настройка NN (freeze bb) и подбор гиперпараметров. В начале тренировок loss взял CE, далее перешел к FL, это помогло поднять f-score для отстающих классов. Для сегментации использовал в качестве encoder ResNet-18, decoder – Unet тренировал с помощью CE, но после оптимизации перешел на JaccardLoss. Качество оценивалось как $\text{competition_metric} = 0.7 * \text{mAP@.5} + 0.3 * \text{meanIoU}$. CM на Baseline было 0.41, удалось повысить до 0.47.

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА • ВЛИЯНИЕ ADVERSARIAL ATTACKS НА СНИМКИ МРТ МОЗГА

Работа была проведена на датасете Brats (3d снимки мозга), выполнен анализ данных, за основу была взята CNN ResNet-18. Произведена кастомизация архитектуры под 3d изображения с применением transfer learning. Выполнена тренировка CNN с различными функциями активации ReLu и BReLU и ее тонкая настройка. Для проверки гипотез была произведена аугментация данных, тренировка CNN на FGSM атаках. Удалось достичь F-меру порядка 94%.

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА • ДЕТЕКЦИЯ ДЕФЕКТОВ СТАЛИ

Работа была проведена на датасете “Северсталь”, задача сегментации. За основу был взят U-net decoder и Res-Net 18 encoder, выполнена тонкая настройка NN. Произведен анализ датасета на возможные корреляции и дисперсию в данных. Выполнена аугментация данных и их предобработка. Качество модели проверял на IoU – удалось достичь порядка 65 % на валидации. Было применены методы борьбы с overfitting. Выполнена визуализация работы NN.



НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА • АНАЛИЗ ТОНАЛЬНОСТИ ПОСТОВ В СОЦИАЛЬНЫХ СЕТЯХ

Обучение модели было выполнено на собственном датасете и с более “продвинутыми” лэйблами. Произведен preprocessing текста заточенный под конкретно мою задачу, также проанализирован датасет для более глубокого понимания параметров текста. На самой начальной стадии были применены классический алгоритмы ML: LogRegCV, AdaBoost, SVM с различными vectorizers TF-IDF, CountVect, HashingVec. На бинарной классификации алгоритмы с тонкой настройкой и адаптивными stopwords показатели f-меру порядка 75 %. В качестве BaseLine для DL была взята архитектура Bert. Проблема небольшого датасета (2500/2500) была решена методом transfer learning, были взяты веса и embeddings натренированные на похожих задачах и тонкой настройкой определенных слоев и параметров тренировки NN. Удалось добиться f-меры порядка 85% без переобучения.

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ • C ++

1. Реализация шаблонного вектора со всеми основными операциями и с динамическим освобождением памяти.
2. Реализация нагруженного дерева (trie, бор) с упором на время выполнения и потребляемую память.
3. Реализация игры морской бой на основе принципа ООП и написание умного бота для игры. Виртуализация.
4. Реализация умной телефонной книги на основе STL, оптимизированной по времени работы и по памяти.
5. Реализация решателя судоки (количество решений, решение) оптимизировано по времени и по памяти.

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ • PYTHON

1. **Первое место на Всероссийском конгрессе молодых ученых.** Софт для автоматизированного поиска в открытых научных источниках информации для исследования предметной области.
2. Реализация игры жизнь, на основе ООП, работа с GUI и пользовательским интерфейсом.
3. Реализация системы контроля версии с основными командами и принципами работы.



4. Реализация пула процессов с управлением памятью и рабочими, если память во время вычислений выходит за пределы выделенной, то этот расчет отправляется и обрабатывается во втором раунде.
5. Реализация веб-сервиса “записной книжки” на Django, с реализацией REST API и упаковкой в Docker контейнер и покрытием Unit tests .

МАТЕМАТИЧЕСКИЕ МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ • PYTHON

1. Реализация **DecisionTree** с возможностью выбора основных гиперпараметров, на основе этого дерева реализация **RandomForest** с возможностью выбора основных гиперпараметров.
2. Реализация **Polynomial regression, Linear regression** с возможностью как аналитического решения, так и итерационного. Визуализация решений и применение на реальных данных.
3. Реализация нейрона с различными функциями активации Sigmoid(LogLoss), ReLu, LeakyReLu, ELU (MSELoss).
4. Реализация **Naive Bayes Classifier**.
5. Работа с основными библиотеками Pandas, NumPy, PyTorch.

