Первое испытание в Спарте № 399
или лабораторная + конкурс

Совсем недавно вы начали изучать новое боевое исскуство, которое называется машинным обучением. У нас в Спарте № 399 этим исскуством должен обладать каждый. В случае, если вы не справитесь с этим испытанием, то по правилам Спарты ~~вас сбрасывают со скалы~~ все будет плохо на экзамене. В отличии от других групп вам предстоит пройти более сложное испытание.

Описание и технические детали

Данные

Возможно, вы уже слышали про датасет MNIST — 60 000 рукописных цифр в train и 10 000 в test. В Спарте создали его hard-версию — 200 000 усложненных цифр в train и 20 000 в test.

Что нужно сделать

Для изображений из test научиться предсказывать цифры, которые на них изображены. Качество будет оцениваться метрикой accuracy — доля верно угаданных ответов на test.

Что можно использовать

Все, что было на лекциях 1-3 и на семинарах 1-5. Никакие методы реализовывать не нужно, если они уже есть реализованные где-либо.

А что если мало оперативки?

Не проблема! Специально для вас мы сделали доступ на кластер, на котором 80 Гб оперативки. Однако стоит учесть, что t-SNE требует очень много памяти, и уже на 30 000 объектов может требовать 20-30 Гб. Поэтому не стоит запускать t-SNE на выборке из более чем 30 000 объектов.

Про доступ на кластер читайте инструкцию, которая лежит на Диске. Для запусков t-SNE на большой выборке на кластере действует система расписаний. Подробнее в инструкции.

Откуда брать данные

Для работы за своим компьютером данные можно скачать с Диска. На кластере данные лежат в папке `/home/volkov/hard_mnist/` и называются `hard_train.txt`, `hard_test.txt`, `hard_train_labels.txt`. В каждой строке файлов `hard_train.txt`, `hard_test.txt` записано ч/б изображение размера 28x28, растянутое в строку. Для отрисовки картинки делайте `reshape((28, 28))` и устанавливайте `cmap='gray'` в функции `plt.imshow`.

Списывание

Поскольку в конкурсе несколько первых мест получат большие призы, делиться решением невыгодно. Лабораторные так же будут проверяться на списывание. Если вы думаете, что ваше списывание не заметно, то это может быть не так. По еженедельным домашним заданиям это видно. Но в данном случае списывание будет строго наказываться ~~сбрасыванием со скалы~~ нулем баллов за работу.

Подсказки на семинарах

Да, на семинарах 18 и 25 марта будут высказанны некоторые идеи, которые могут помочь при решении конкурса. Ну а могут и не помочь ;)

Дедлайн и куда присылать

На лабораторную дается две недели — **до 27 марта 23:59**. Стоит учесть, что традицию все делать в последнюю ночь стоит нарушить — например, применение t-SNE даже к небольшой выборке может занять пару часов. Конкурс доступен **до 30 марта**.

Задания нужно присылать на почту **probability.diht@yandex.ru**. В теме письма нужно сначала указать **[ML 399]** затем через пробел указать свое имя и фамилию.

Формат сдачи

Лабораторная. Нужно прислать файл ipynb со всеми графиками, подробными комментариями и выводами внутри него. За отсутсвие комментариев (не по коду, а по работе) и выводов баллы могут быть снижены. Ваш код запускаться не будет, но может быть прочитан.

Конкурс. **В течении суток** после окончания конкурса нужно прислать файл ipynb с кодом, с помощью которого у вас получился лучший результат. Это означает, что **код каждой посылки в систему kaggle стоит сохранять**. В начале этого файла должно быть краткое описание решения. Код должен быть рабочим — у некоторых он будет запущен и проверен на соответствие результатов.

Если вы хотите использовать непитоновские библиотеки, а что-то, что запускается из терминала, то пишите эти команды прямо в Jupyter'e, поставив в начале команды знак "!". Соответствущие файлы так же нужно прислать.

Баллы за задание

До 15 баллов за лабораторную и до 25 за конкурс (подробнее см. файл на Диске). Чтобы получить оценку выше 0 баллов, необходимо побить baseline-решение (см. ниже), то есть необходимо загрузить в kaggle файл с предсказаниями меток, на котором качетсво будет выше baseline-решения.

Небольшой совет

Поскольку некоторые методы могут работать очень долго, то в лабораторной лучше как можно чаще сохранять посчитанные результаты в файл, чтобы в случае чего не пришлось все пересчитывать.

Если есть вопросы
То лучше пишите мне в ВК. Так быстрее.

Конкурс
https://inclass.kaggle.com/c/hard-mnist-399
Инвайт: https://kaggle.com/join/i4ny87nw48newi

Формат файлов labels
В рамках конкурса такой файл имеет две колонки с названиями Id и label. Первая строка содержит эти названия через запятую. Все следующие строки содержат сначала номер тестового объекта, а после запятой его метку класса. В нашем случае колонка Id имеет вид range(20000). Пример файла можно посмотреть, открыв на странице конкурса вкладку Get the Data и скачать файл `hard_train_labels`.

Формат файлов с пикселями на странице конкурса так же отличается от от файлов на диске, но отличия только в заголовке и в колонке Id. Для обучения можно использовать файлы с диска.

Baseline
В качестве baseline взят простой 8-NN, обученный по первым 5000 объектов из файла `hard_train`. Код baseline не выдается — такие простые вещи должен написать каждый, кто хочет сдать курс. Тем более что на семинарах приводилось не мало такого кода.

Если вы не сможете построить классификатор, который будет не хуже baseline, то за работу (как за конкурс, так и за лабораторную) вам ставится 0 баллов.

Leaderboard
Результаты можете посмотреть во вкладке Leaderboard. Ваше имя для Leaderboard должно быть понятным, желательно имя и фамилию. До окончания конкурса результаты расчитаны только по трети тестовой выборки. После окончания конкурса результаты будут расчитаны по всей тестовой выборке.

Количество посылок в систему
Чтобы избежать сильной подгонки разрешается делать только **5 попыток в день**.

Лабораторная
Для начала визуализируйте несколько изображений цифр. Постарайтесь сделать это красиво в форме таблицы, как в примерах с семинаров. Не забывайте про `cmap='gray'` в функции `plt.imshow`. Далее спартанцы должны выполнить следующие задания.

Исследование kNN (4 балла)
В этой части задания возьмите в качестве пяти обучающих выборок части train размеров $\{2 \cdot 10^3, 4 \cdot 10^3, 6 \cdot 10^3, 8 \cdot 10^3, 10^4\}$, но можно и больше. В качестве тестовой выборки возьмите часть train размера 1000 но так, чтобы эта часть не пересекалась с выборками, которые вы взяли в качестве обучающих.

Для разных размеров обучающей выборки посчитайте accuracy (доля верно угаданных ответов) на тестовой выборке при классификации с помощью kNN для $k \in \{1, ..., 20\}$ и измерьте время работы. Постройте графики зависимости accuracy от числа k в kNN для каждой обучающей выборки. Для нескольких значений k постройте графики времени

работы в зависимости от размера выборки. Почему 2-NN в данном случае работает хуже всех?

Поскольку графики получатся шумными, проведите такие разбиения выборки несколько раз и усредните значения. Выберите лучшее k, обучите на нем kNN для самой большой обучающей выборки и постройте для него матрицу ошибок на тестовой выборке. В матрице ошибок на позиции (i, j) должно стоять число, равное доли объектов класса i, которые при классификации были отнесены к классу j. Визуализируйте эту матрицу при помощи `plt.imshow(error_matrix, interpolation='none')`.

Исследование PCA+kNN (4 балла)

В этой части задания возьмите в качестве обучающей выборки X_{tr} часть train размера 10000. В качетсве тестовой выборки X_{test} возьмите часть train размера 1000 но так, чтобы эта часть не пересекалась с выборкой X_{tr}, которую вы взяли в качестве обучающей.

Примените PCA для двух главных компонент к X_{tr} и визуализируйте выборку, как было показано на семинаре 5, но взяв в качестве цвета точки ее метку класса. Не забывайте устанавливать параметр `alpha` (прозрачность точек) со значением не больше 0.2, либо можно убрать границы точек `linewidths=0`.

Для всех $d \in \{1, ..., 20\}$ примените PCA к X_{tr}, получив отображение pca_d в сжатое пространство размерности d, соответствующее первым d главным компонентам. Для каждого d по сжатой выборке $pca_d(X_{tr})$ обучите kNN для $k \in \{1, ..., 20\}$, затем примените его на сжатой выборке $pca_d(X_{test})$ и посчитайте по полученным ответам accuracy. Для каких d и k получается лучший результат?

Изобразить матрицы качества и времени работы в зависимости от d и k. Лучше всего это делать с помощью `plt.imshow(accuracy, interpolation='none')`. Поскольку данные получатся шумными, провести разбиение выборки несколько раз и усреднить значения.

Исследование t-SNE+kNN (4 балла)

Исследование, аналогичное исследованию с PCA повторить для t-SNE с двумя отличиями. Поскольку t-SNE в отличии от PCA решает только E-problem, то применять его нужно сразу к $X_{tr} \sqcup X_{test}$, разделив эти выборки сразу после преобразования в сжатое пространство. Так же берите $d \in \{1, ..., 5\}$ для t-SNE. Почему при $d > 3$ качество получается хуже?

Дополнительно (3 балла)

a). Придумайте свой способ эффективного снижения размерности и реализуйте его. Может быть, стоит не учитывать некоторые пиксели.

b). Придумайте, какие дополнительные признаки можно посчитать по изображению. Улучшилось ли качество при их использовании?