Практическая работа №6 Долгинцев Степан КТСО-03-20

# Знакомство с алгоритмами.

Подключаем нужные библиотеки

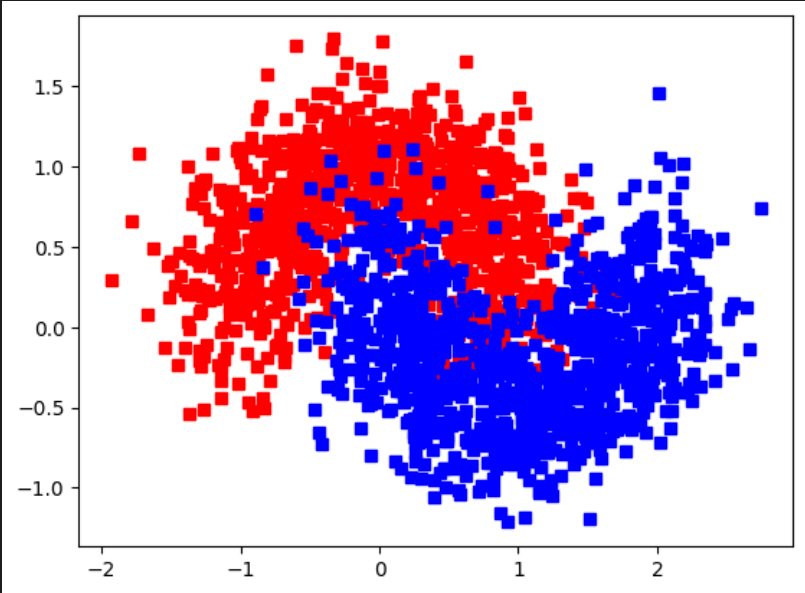
Изображение выглядит как текст

Автоматически созданное описание

Загружаем и сплитим данные в 3 стака – трейн, валидейшн и тест.

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание



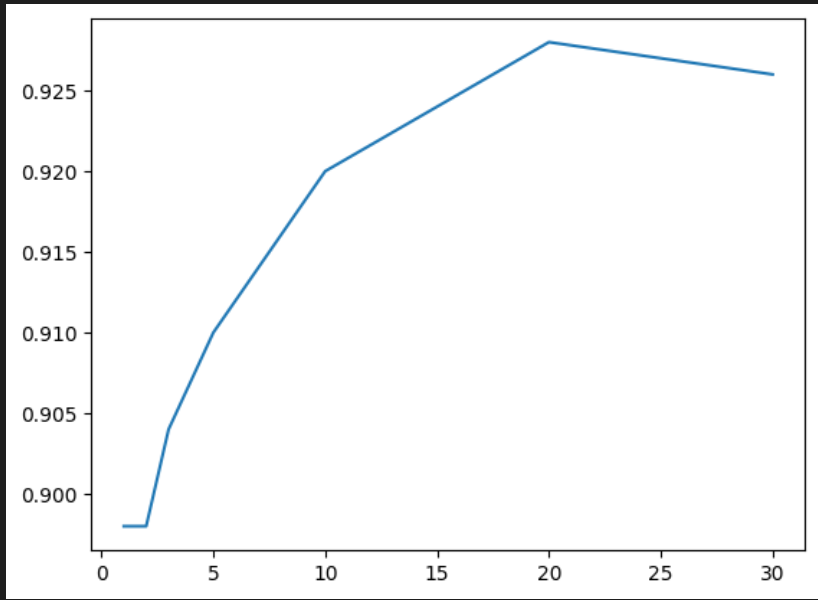
Выбираем лучшее K для алгоритма К-ближайших соседей простым перебором.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

График точности от выбора К.

Точность на тестовых данны:

Изображение выглядит как текст

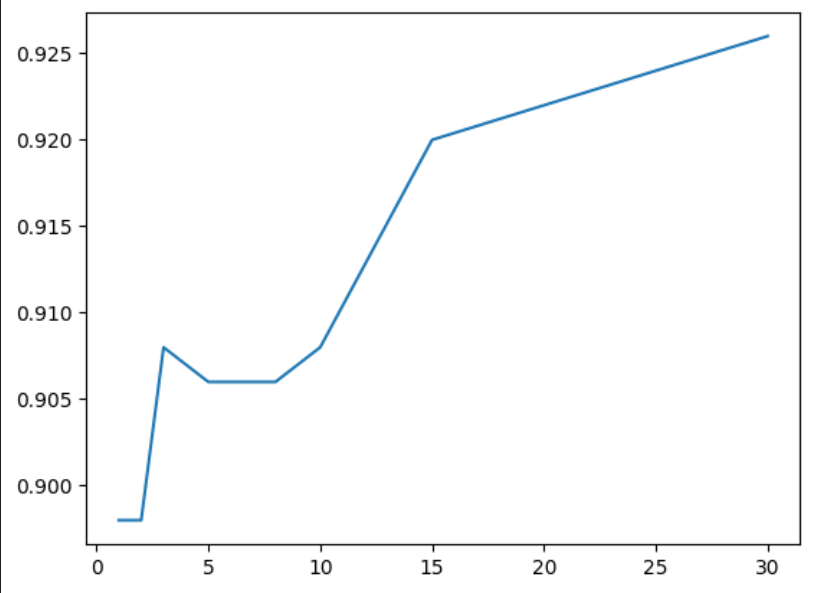
Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изменим параметр весов на distance и теперь более удаленные узлы будут оказывать меньшее влияние на результат нежели более близкие. По дефолту они имеют одинаковый вес вне зависимости от расстояния.

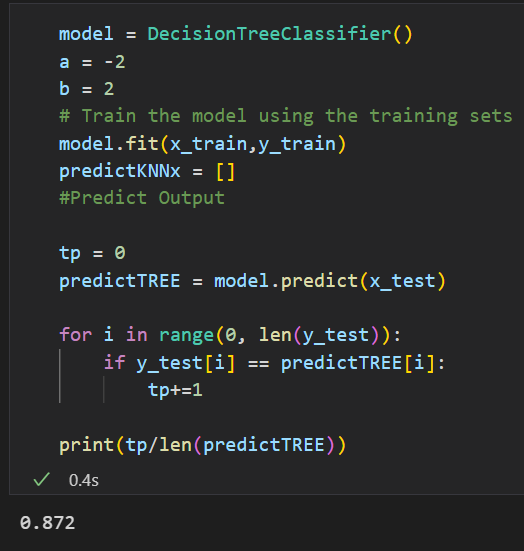
Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

На тестовых данных вот такая точность.



На этих же данных алгоритм деревьев решений показывает вот такой результат.

Изображение выглядит как текст

Автоматически созданное описание

Случайный лес такой.

Изображение выглядит как текст

Автоматически созданное описание

Адабуст – соответственно такой.

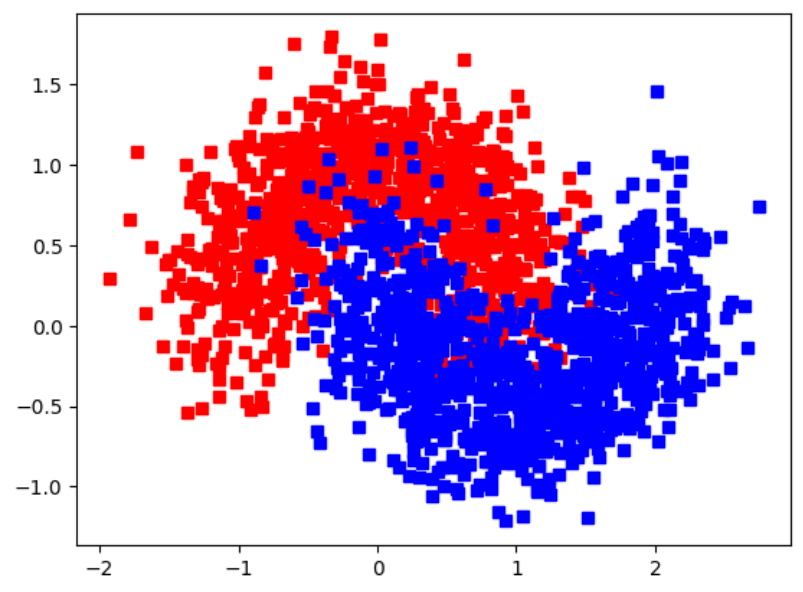
# Деревья решений

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание



Все тоже самое, что и в прошлый раз.

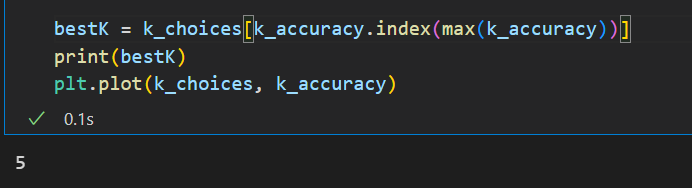
Изображение выглядит как текст

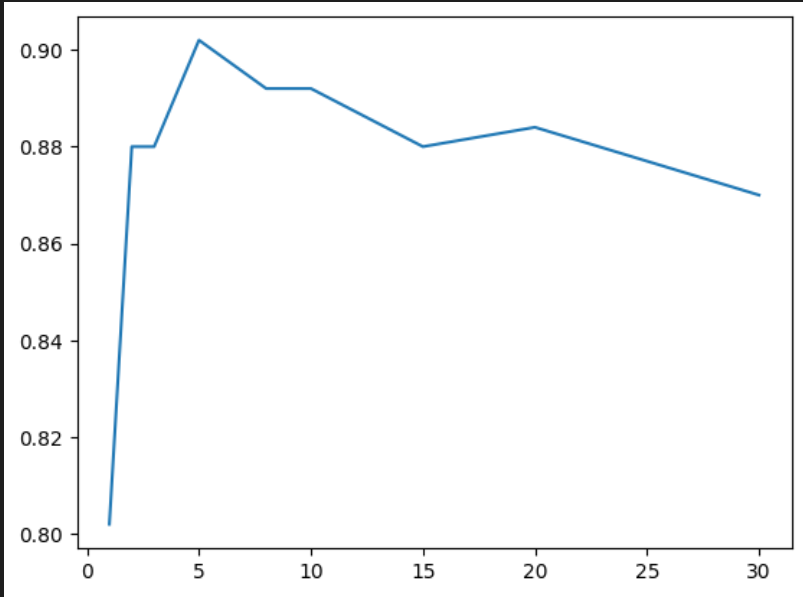
Автоматически созданное описание

Обычное решение без выбора опций.

Изображение выглядит как текст

Автоматически созданное описание

Точность относительно выбора глубины. 



Глубина 5 оказалась лучшей

Помимо точности можно выбрать критерий для нашего дерева.

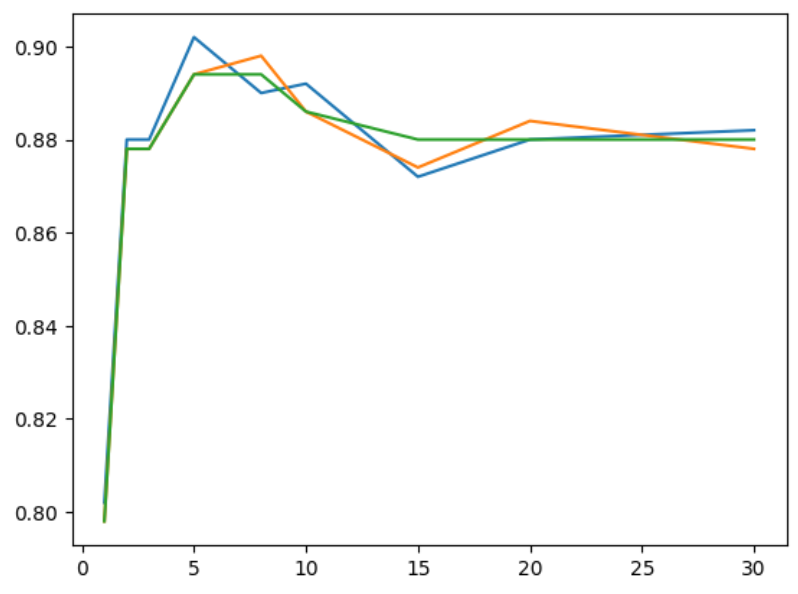
Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание



Изображение выглядит как текст

Автоматически созданное описание

Глубина 5 и критерий Джини оказались лучшими гиперпараметрами.

Изображение выглядит как текст

Автоматически созданное описание

Точность на тесте с новыми данными.

# Линейный классификатор

Вообще как сказано в методичке, что нужно использовать гиперболический тангенс, но у этой функции есть ряд недостатков по сравнению с такой функцией как ReLu. Например угасающий градиент на относительно больших иксах.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Подключаем нужные библиотеки, а также загружаем набор данных, в этот раз это будет новый набор данных, а именно: номера дома из The Street View House Numbers (SVHN) Dataset.

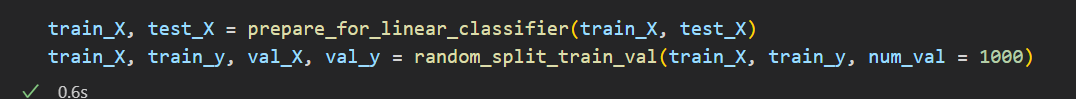
Посмотрим на них:

Изображение выглядит как текст

Автоматически созданное описание



Подготавливаем данные к нашему классификатору, а также сплитим трейн на трейн и валидацию.



Наш линейный классификатор включает в себя такие функции как softmax и cross-entropy-loss, а также L2 регуляризацию.Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

LinearSoftmax -это функция которая соединяет весь пайплайн алгоритма в себе и выдает loss и изменение весов при помощи вычисления производной

Также есть Класс:



У которого есть методы фит и предикт:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

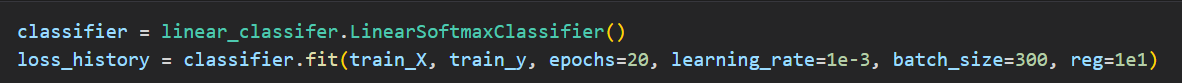
Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

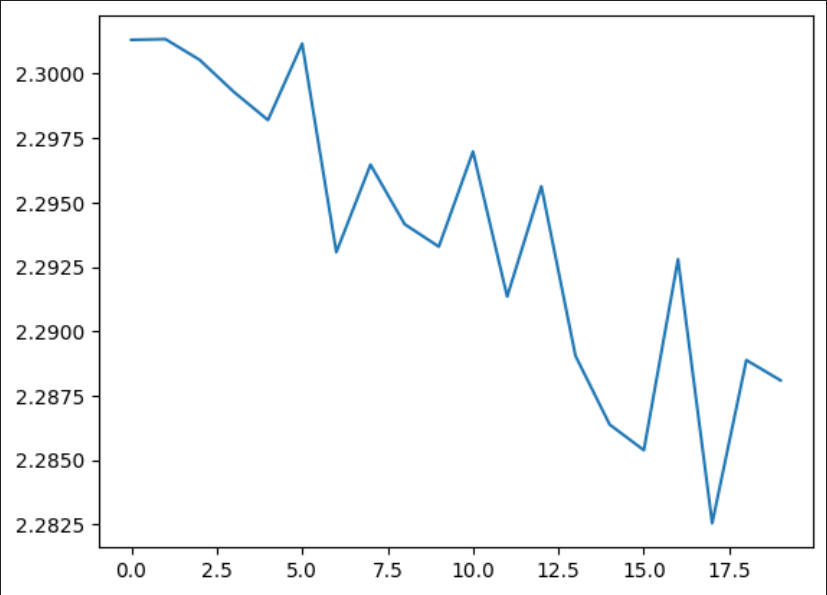
Конкретно здесь реализован для градиентного спуска стохастический метод(SGD), но для того, чтобы этот алгоритм стал не стохастическим размер батча нужно увеличить до размера данных.

Далее создаем этот класс и присваиваем ему наши данные, а также гиперпараметры: такие как кол-во эпох, скорость обучения, размер батча и сила регуляризации.



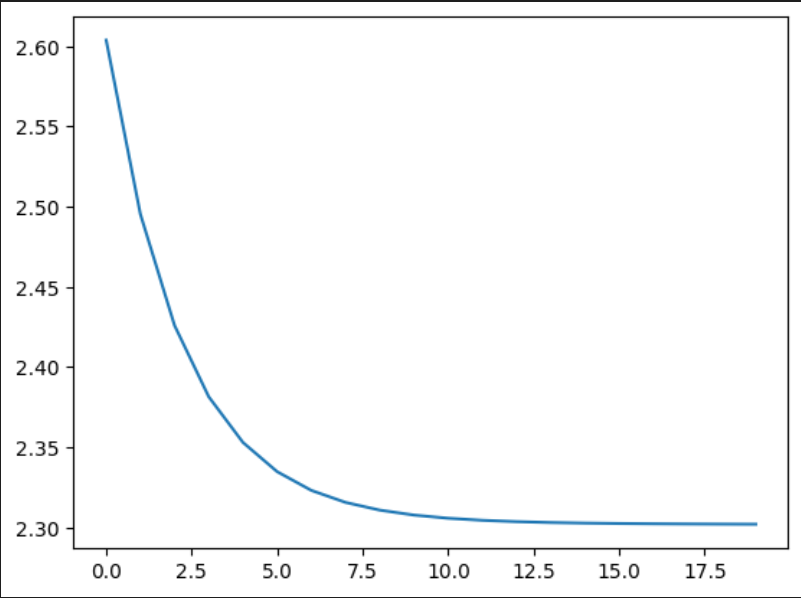
Построим график ошибки относительно эпох:





Не стохастический: Изображение выглядит как текст

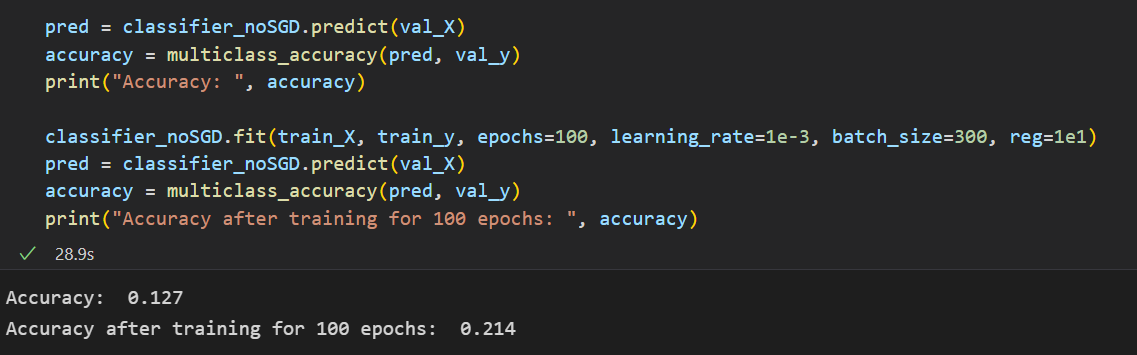
Автоматически созданное описание



Точность на валидации после задания других гиперпараметров относительно дефолтных.

Изображение выглядит как текст

Автоматически созданное описание



Ищу лучший лернин рейт и силу регуляризации.

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Точность на тестовых данных:Изображение выглядит как текст

Автоматически созданное описание

Вспомогательная функция для точности:Изображение выглядит как текст

Автоматически созданное описание