Assignment 0. Sentiment analysis using logistic regression

26 февраля 2021 г.

Теоретическая часть

Введем следующие обозначения: $(x_{\{1\}}, y_{\{1\}}), \ldots, (x_{\{N\}}, y_{\{N\}})$ — обучающая выборка размера N, $x_{\{i\}} \in \mathbb{R}^M$ — вектор признаков i-ого примера, M — количество признаков, $y_{\{i\}} \in \{0, 1\}$ — класс i-ого примера, $w \in \mathbb{R}^{M+1}$ — вектор весов логистической регрессии.

Примечание. Линейные преобразования над $x_{\{i\}}$ имеют следующий общий вид: $w_0 + w^T x_{\{i\}} = w_0 + w_1 * x_{\{i\},1} + \ldots + w_M * x_{\{i\},M}$, где w_0 называется смещением (bias). Для удобства реализации здесь и далее мы всегда будем присоединять к входным векторам $x_{\{i\}}$ единицу, тогда линейные преобразования можно представить в следующем виде:

$$w_0 * 1 + w_1 * x_{\{i\},1} + \ldots + w_M * x_{\{i\},M} \equiv w^T [1; x_{\{i\}}]$$

1. Покажите, что нейрон с бинарной пороговой функцией активации может точно вычислять функции алгебры логики $x1\ OR\ x2$, $x1\ AND\ x2$, $NOT(x1\ AND\ x2)$: для каждой функции нарисуйте decision boundary, вычислите соответствующие ей веса. Предложите полносвязную нейронную сеть с одним скрытым слоем и пороговой бинарной функцией активации, которая может точно вычислять функиию $x1\ XOR\ x2$: нарисуйте decision boundary, саму нейросеть, подпишите веса над связями между нейронами.

2. Посчитайте производную сигмоиды $\sigma(z)$ и выразите его через саму сигмоиду, считая что z — скаляр.
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

3. Покажите, что для сигмоиды $\sigma(z)$ справедливо следующее выражение:
$$\sigma(-z) = 1 - \sigma(z)$$

4. Выпишите формулу гипотезы $h_w(x)$ для логистической регрессии.

5. Нарисуйте графики значения оценочной функции бинарная кросс-энтропия для одного примера из положительного и одного примера из отрицательного класса в зависимости от выхода логистической регрессии $\hat{y} = h_w(x)$. Чему равно значение оценочной функции при нулевых весах (сразу после инициализации)?
$$bce(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

6. Рассмотрим следующую вероятностную модель: $y \sim Bernoulli(h_w(x))$, т.е. истинный класс примера – случайная величина, принимающая значение 1 с вероятностью $h_w(x)$, предсказанной нашей логистической регрессией для данного примера. Выпишите функцию правдоподобия. С помощью принципа максимального правдоподобия обоснуйте вид оценочной функции бинарная кросс-энтропия.

7. Посчитайте градиент оценочной функции $\nabla_w L(w, x_{\{1\}}, \ldots, x_{\{N\}})$ для бинарной (двухклассовой) логистической регрессии. В качестве оценочной функции использовать кросс-энтропию с L_2 регуляризацией:
$$L(w, x_{\{1\}}, \ldots, x_{\{N\}}) = -\frac{1}{N} \sum_{i=1}^{N} (y_{\{i\}} \log h_w(x_{\{i\}}) + (1 - y_{\{i\}}) \log(1 - h_w(x_{\{i\}}))) + \alpha \sum_{j=1}^{M} (w_j)^2$$

Примечание: Обратите внимание, что в регуляризационный член $\sum_{j=1}^{M}(w_j)^2$ при суммировании обычно не включают w_0, поскольку он отвечает за общий сдвиг значений функции, который может быть произвольным, а L_2 регуляризация стремится минимизировать значения w_i.

8. Запишите формулу для обновления вектора параметров w при обучении методом градиентного спуска.

9. Докажите, что оценочная функция бинарная кросс-энтропия для бинарной логистической регрессии имеет единственный минимум в пространстве весов.

10. Покажите, что минимизация оценочной функции бинарная кросс-энтропия для логистической регрессии эквивалентна минимизации следующей функции (сумма по примерам и регуляризационный член опущены): $softplus(-tw^Tx)$, где

$$softplus(x) = log(1 + e^x), t = 2y - 1 \in \{-1 + 1\}$$

Практическая часть

Используя формулы из теоретической части, реализуйте логистическую регрессию. С ее помощью обучите анализатор тональности отзывыв о фильмах.

A Загрузите датасет, используя функцию $load\_dataset\_fast$ (https://github.com/nvanva/filimdb_evaluation/blob/master/score.py). Сделайте его предобработку и токенизацию. Постройте словарь, в котором будут содержаться все уникальные токены из обучающей выборки. **Вопрос:** Чему равен размер получившегося словаря?

B Для того, чтобы обучить логистическую регрессию необходимо представить обучающую и тестовую выборки в виде матриц размера $N * V$, где N – количество отзывов в выборке, а V – размер словаря. В такой матрице i-я строка содержит bag-of-words вектор i-ого отзыва из выборки, j-я компонента которого вычисляется как абсолютная частота (число вхождений) j-го токена из словаря в этот отзыв. Большинство элементов матрицы будут равны 0. В целях оптимальной работы с памятью воспользуйтесь scipy Compressed Sparse Row matrix. **Вопрос:** Сколько памяти занимает обучающая выборка (используйте поле nbytes каждого из массивов data, indices, indptr), почему? Сколько памяти она заняла бы в формате dense-матрицы (numpy.ndarray), почему?

C Напишите функцию сигмоида (покомпонентная), использую для вычисления экспоненты функцию numpy.exp(). Проверьте, что реализованная функция одинаково правильно работает со скалярами, векторами, матрицами - это потребуется для эффективной реализации модели. **Вопрос:** Хотя в теории значения сигмоиды находятся в интервале (0,1), из-за ограниченной точности вычислений с плавающей точкой ваша первая реализация сигмоиды может возвращать значения не из этого интервала (например, ровно 1.0 для больших положительных аргументов). Какая может возникнуть проблема при вычислении оценочной фунции? Как реализовать функцию сигмоида, чтобы этого не происходило?

D **Инициализация.** Напишите функцию инициализации весов (принимает число признаков, возвращает вектор нулей нужного типа и размера).

E **Прямой проход.** Напишите две функции, которые для заданной выборки (матрица векторов признаков X, вектор правильных ответов y), вектора весов w и значения гиперпараметра α вычисляют оценочную фунцию и точность: одну с использованием цикла по примерам, другую - без цикла, с помощью матричных операций csr_matrix.dot, numpy.sum и т.п. Вычислите аналитически значение оценочной функции при нулевых весах и проверьте, что оно совпадает с теми, которые возвращает ваш код. **Вопрос:** Сколько времени занимает прямой проход на обучающей выборке, реализованный с/без цикла по примерам? Чему равно значение оценочной функции сразу после инициализации?

F **Прямой+обратный проход.** Напишите функцию, которая для заданной выборки (матрица векторов признаков X, вектор правильных ответов y), вектора весов w и значения гиперпараметра α возвращает точность классификатора, значение оценочной функции L и ее градиенты $\frac{\partial L}{\partial w_j}$. Не используйте циклы.

G **Градиентный спуск.** Напишите функцию, которая используя посчитанные градиенты, делает шаг градиентного спуска (обновляет значение весов). Напишите функцию, которая принимает обучающую выборку, текущие значения весов, число эпох обучения, learning rate, обучает модель методом градиентного спуска и возвращает новые значения весов. *Примечание.* Рекомендуется также реализовать mini-batch gradient descent: разбейте обучающую выборку на мини-батчи по 100-500 примеров, на каждой итерации градиентного спуска считайте градиент не на всей выборке, а на очередном мини-батче. Это существенно ускорит время одной итерации, и позволит обучать модель гораздо быстрее!

H Для тестирования отключите регуляризацию ($\alpha = 0$) и проверьте, что классификатор может запомнить небольшую часть (1000 примеров) обучающей выборки (точность на ней должна быть равна 1.0, потребуется подобрать learning rate и число эпох обучения). *Примечание.* Если этого не происходит, ищите ошибки в реализации. В первую очередь рекомендуется проверить правильность вычисления градиентов, сравнив их с вычисленными по формуле второй разностной производной: см. gradient check.

I Установите learning rate равным 1e−3, коэффициент L_2 регуляризации α равным 1e−5. Обучите логистическую регрессию на обучающей выборке. Постройте графики изменения оценочной функции и точности на обучающей и валидационной выборках в процессе обучения (графики обучения). Для построения графиков можно воспользоваться библиотекой matplotlib (см. совсем краткое и чуть более подробное руководства по ней), рисовать графики можно в Jupyter Notebook (см. пример). **Вопрос:** Приведите построенные графики обучения. Через сколько эпох обучение сходится? Какой точности классификатора вам удалось достичь на обучающей, валидационной, тестовой выборках? Имеет ли место переобучение классификатора или недообучение?

J Попробуйте разные значения learning rate. **Вопрос:** Приведите графики обучения для нескольких различных значений learning rate. Какие выводы можно сделать?

K Сейчас в качестве коэффициента L_2 регуляризации α мы выбрали некоторое произвольное значение. При неправильном выборе α модель может как переобучиться (α слишком мало), так и остаться недообученной (α слишком велико). Подберите значение α, дающее максимальную точность на валидационной выборке. Учитывайте, что при изменении α меняется оценочная функция, поэтому число эпох обучения и learning rate может потребоваться подобрать заново. Используйте графики обучения для подбора! **Вопрос:** Приведите графики обучения для нескольких значений α. Какие выводы можно сделать? Какое потребовалось число эпох и learning rate для обучения до сходимости? Сколько времени заняло обучение, разметка тестовой выборки?

L Добавление n-грамм в качестве признаков позволяет существенно улучшить точность классификатора. Попробуйте подобрать подходящие значения n, чтобы достичь максимальной точности. Учитывайте, что после изменения числа признаков подобранные ранее гиперпараметры (learning rate, число эпох обучения и α) могут стать неоптимальными. **Вопрос:** На сколько улучшается точность при добавлении n-грамм разного порядка? Как потребовалось изменить гиперпараметры?

M Какие признаки оказались наиболее высомыми (получили максимальные по модулю веса) для классификации отзыва? **Вопрос:** Распечатайте 20 наиболее весомых признаков для позитивного и негативного класса.

Исследовательская часть

1. Помимо классического градиентного спуска, существуют различные его вариации, такие как Adam, Adagrad, RMSProp, которые активно используются в реальном мире (см.

краткий и более подробный обзоры этих методов). Реализуйте Momentum или Adagrad и используйте его для обучения логистической регрессии. Приведите графики обучения при различных значениях гиперпараметров. Какие наблюдения и выводы можно сделать?

2. Для улучшения работы классификатора важно попробовать понять, на каких именно примерах-отзывах он ошибается и почему. Приведите некоторое количество таких примеров, какие признаки приводят к ошибке (например, для положительных отзывов входят с большим отрицательным весом в $w^T x_{\{i\}}$, делая это скалярное произведение отрицательным, следовательно, приводя к неправильной классификации). Есть ли в неправильно классифицированных примерах какие-то характерные черты, особенности или схожие тендеции? Если да, то какие шаги можно было бы предпринять для дальнейшего улучшения классификатора?

3. В качестве компонент bag-of-words векторов можно использовать разные признаки: абсолютные частоты слов, относительные частоты, бинарные признаки (входит слово в пример или нет). Каждый признак можно прологорифмировать, привести к диапазону [0,1] или стандартизовать (вычесть среднее по обучающей выборке значение данного признака и поделить на стандартное отклонение. Попробуйте различные варианты и опишите результаты.