Содержание

1 Теоретическая часть

Определения 1. Введем следующие обозначения:$(x_1, y_1), ..., (x_N, y_N)$ обучающая выборка размера N, $x_i \in R^M$ - вектор признаков i-ого примера, M - количество признаков, $y_i \in 0, 1$ -класс i-ого примера, w $\in R^{M+1}$ - вектор весов логистической регрессии. *Примечание.* Линейные преобразования над x_i имеют следующий общий вид :

$$w_0 + w^T x_i = w_0 + w_1 * x_{i,1} + ... + w_M * x_{i,M} \equiv w^T [1; x_i] \tag{1.1}$$

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:

$$w_0 * 1 + w^T x_i = w_0 + w_1 * x_{i,1} + ... + w_M * x_{i,M} \equiv w^T [1; x_i] \tag{1.2}$$

Задача. 1) Покажите, что нейрон с бинарной пороговой функцией активации может точно вычис-лять функции алгебры логики x_1 OR x_2, x_1 AND x_2, NOT(x_1 AND x_2): 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 x_1 XOR x_2: нарисуйте decision boundary, саму нейросеть, подпишите веса над связями между нейронами.
Preactivation function:

$$z = w^T x + b \tag{1.3}$$

$$h_w(x) = \begin{cases} 1 & z < t \\ 0 & z \geq t \end{cases}$$

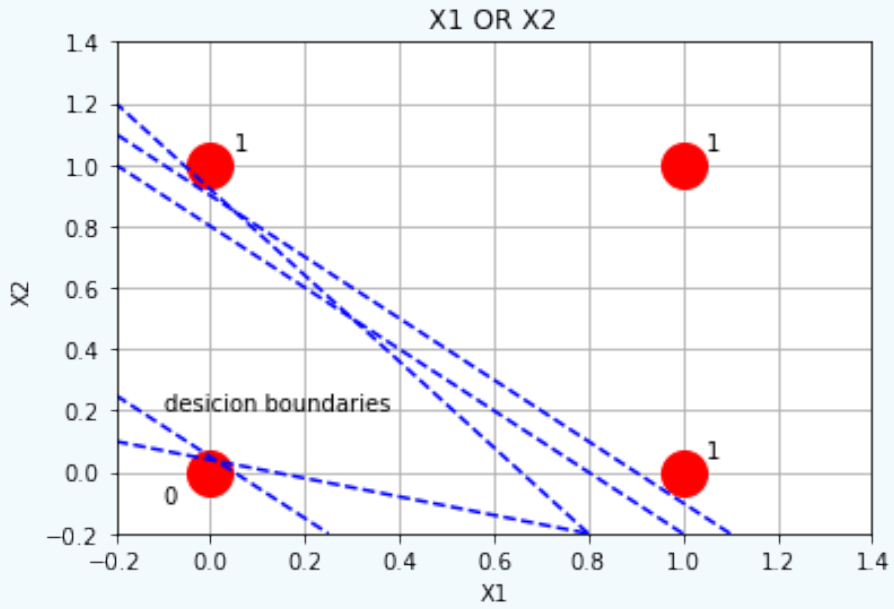
$$t = \frac{-b}{||w||} = \frac{-w_0}{\sqrt{w_1^2 + w_2^2}} \tag{1.4}$$

1) x_1 **OR** x_2. Выборка линейно разделима.

$$w_1 = w_2 = 1; t = \frac{\sqrt{2}}{2} * \frac{1}{2} = \frac{\sqrt{2}}{4};$$

$$w_0 = -t * ||w|| = -t * \sqrt{2} = -\frac{1}{2}$$

$$y = w_0 + w^T (x_1 + x_2) = -\frac{1}{2} + [1; 1]^T * (x_1 + x_2)$$

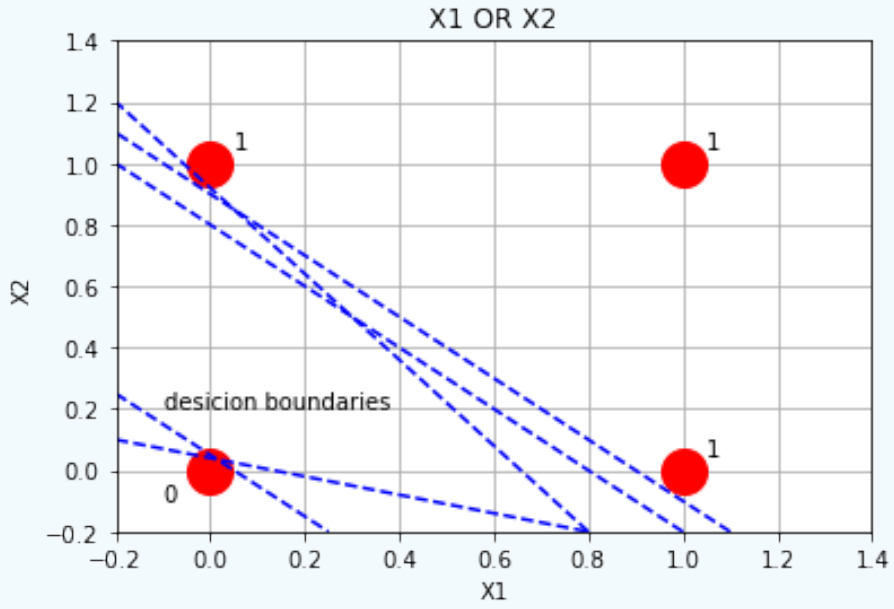


2) x_1 **AND** x_2. Выборка линейно разделима.

$$w_1 = w_2 = 1; t = \frac{\sqrt{2}}{2} * \frac{1}{2} + \frac{\sqrt{1}}{2} = \frac{3 * \sqrt{2}}{4};$$

$$w_0 = -t * ||w|| = -t * \sqrt{2} = -\frac{3}{2}$$

$$y = w_0 + w^T(x_1 + x_2) = -\frac{3}{2} + [1; 1]^T * (x_1 + x_2)$$

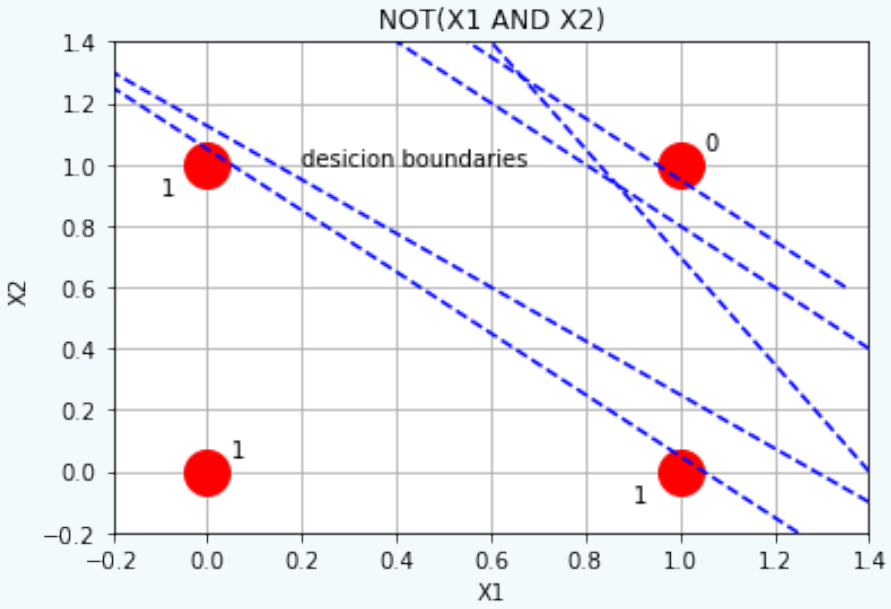


3) NOT(x_1 AND x_2). Выборка линейно разделима.

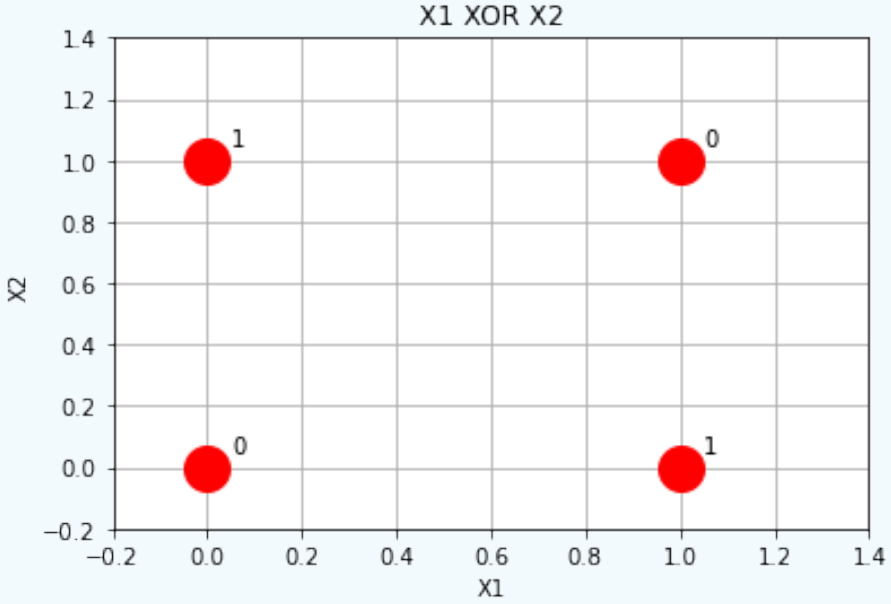
$$w_1 = w_2 = -1; t = -\frac{\sqrt{2}}{2} * \frac{1}{2} - \frac{\sqrt{1}}{2} = -\frac{3 * \sqrt{2}}{4};$$

$$w_0 = -t * ||w|| = -t * \sqrt{2} = -\frac{3}{2}$$

$$y = w_0 + w^T(x_1 + x_2) = \frac{3}{2} + [-1; -1]^T * (x_1 + x_2)$$



4) x_1 XOR x_2. Выборка линейно не разделима.



Отделение нижнего нуля:

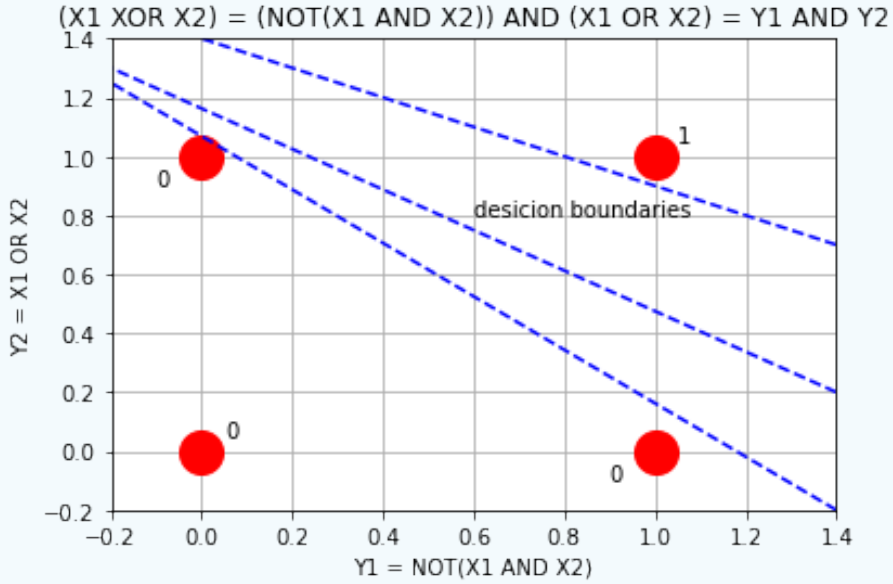
$$y_1 = -\frac{1}{2} + [1;1]^T * (x_1 + x_2)$$

Отделение верхнего нуля:

$$y_2 = \frac{3}{2} + [-1;-1]^T * (x_1 + x_2)$$

y_1 AND y_2

$$y = -\frac{3}{2} + [1;1]^T * (y_1 + y_2)$$



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

$$\sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2} * \frac{e^{-z}+1-1}{(1+e^{-z})^2} = \sigma(z)(1-\sigma(z)) \tag{1.5}$$

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

$$\sigma(-z) = \frac{1}{1+e^z} = \frac{1}{e^z * (1+e^{-z})} = \frac{e^{-z}}{1+e^{-z}} = \frac{1+e^{-z}-1}{1+e^{-z}} = 1 - \sigma(z) \tag{1.6}$$

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

$$z = [1;x]^T w \tag{1.7}$$

Activation function:

$$h_w(x) = \sigma(z) = \frac{1}{1 + e^{-[1;x]^T w}} \tag{1.8}$$

Задача. 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 $\widehat{y} = h_w(x)$. Чему равно значение оценочной функциипри нулевых весах (сразу после инициализации)?

$$bce(y, \widehat{y}) = -y * log\widehat{y} - (1 - y) * log(1 - \widehat{y}) \tag{1.9}$$

Для позитивного класса

$$bce(y, \widehat{y}) = -log(1 - \widehat{y}) \tag{1.10}$$

Для негативного класса

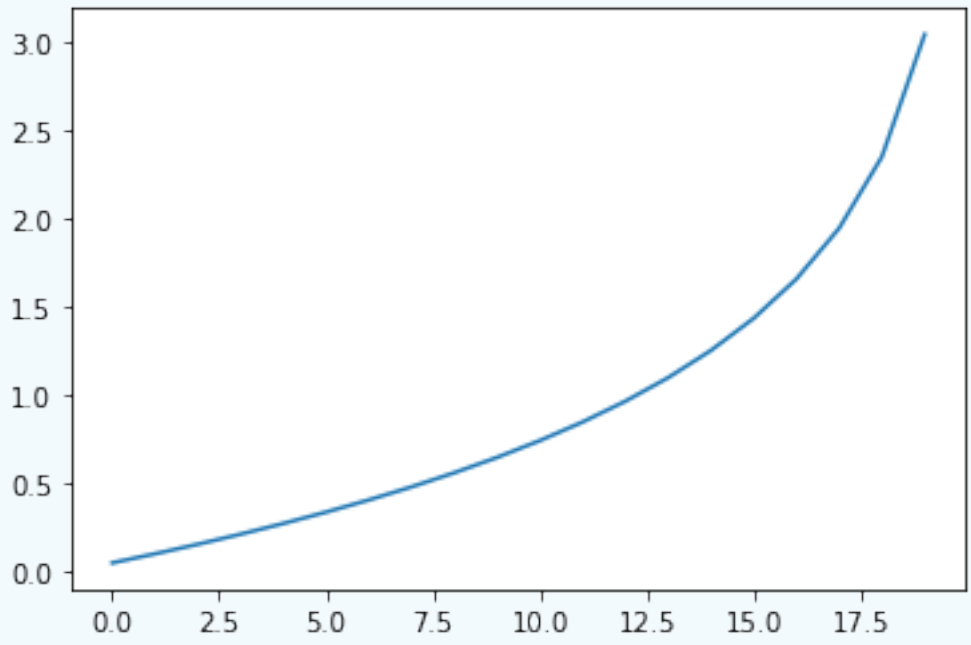
$$bce(y, \widehat{y}) = -log\widehat{y} \tag{1.11}$$

Оценочная функция при нулевых весах:

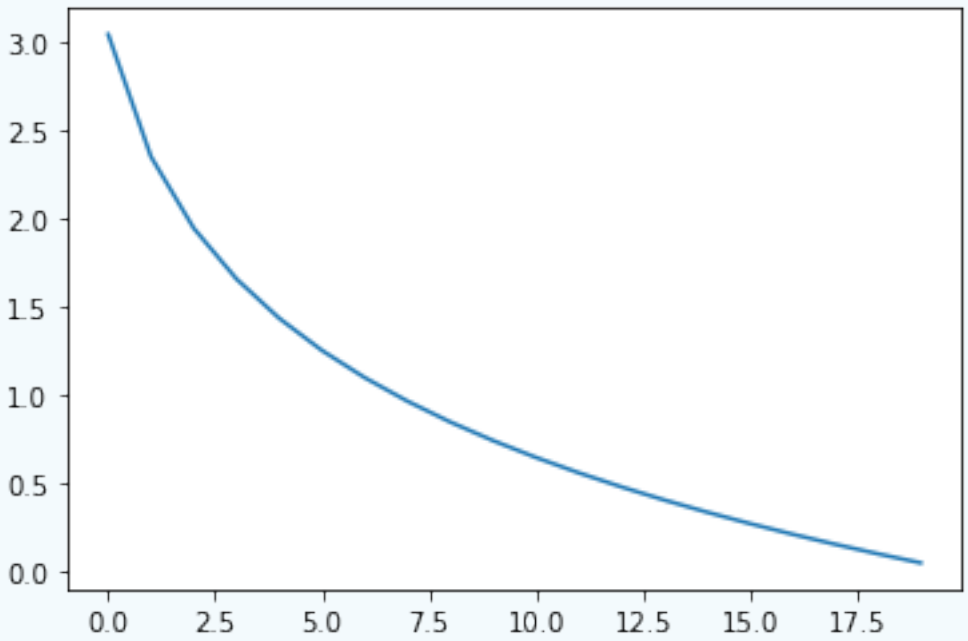
$$\widehat{y} = \sigma(w^T * x) = \frac{1}{1 + e^{-w^T * x}} = \frac{1}{1 + e^0} = \frac{1}{2} = 0.5 \tag{1.12}$$

$$bce(y, \widehat{y}) = ylog(0.5)(1y)log(0.5) = (-y - 1 + y)log(0.5) \approx 0.301 \tag{1.13}$$

pic 1. loss for positive class.



pic 1. loss for negative class.



Задача. 6) Рассмотрим следующую вероятностную модель:yBernoulli($h_w(x)$), 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.

$$h_w(x) = \frac{1}{1 + e^{-w*x}} \tag{1.14}$$

Истинный класс примера:

$$\begin{cases} P(y = 1|x; w) = 1\text{вероятность } h_w(x); P(y = 0|x; w) = 0\text{вероятность } 1 - h_w(x) \end{cases}$$

$$p(y|x; w) = h_w(x)^y (1 - h_w(x))^{1-y} \tag{1.15}$$

Функция правдоподобия:

$$L(w) = p(Y|X; w) = \prod_{i=1}^{n} h_w(x^{(i)})^{y^{(i)}} (1 - h_w(x^{(i)}))^{1-y^{(i)}} \tag{1.16}$$

Логарифмическая функция правдоподобия:

$$l(w) = logL(w) = \sum_{i=1}^{n} y^{(i)} * log(h_w(x^{(i)})) + (1 - y^{(i)}) * log(1 - h_w(x^{(i)})) \tag{1.17}$$

Метод максимального правдоподобия - метод оценивания неизвестного параметра путём максимизации функции правдоподобия. Частные производные логарифмической функции правдоподобия:

$$\eta * \frac{\partial}{\partial w_j} l(w) = \eta * \sum_{i=1}^{n} (y - h_w(x)) * x_j \tag{1.18}$$

Задача. 7) Посчитайте градиент оценочной функции $\nabla w L(w, x_1, ..., x_N)$ для бинарной (двух-классовой) логистической регрессии. В качестве оценочной функции использовать кросс-энтропию с L_2 регуляризацией:

$$L(w, x_1, ..., x_N) = -\frac{1}{N} \sum_{i=1}^{N} (y * log\widehat{y} + (1 - y) * log(1 - \widehat{y})) + \alpha \sum_{j=1}^{M} (w_j)^2 \tag{1.19}$$

Примечание: Обратите внимание, что в регуляризационный член $\alpha \sum_{j=1}^{M} (w_j)^2$ 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.
Сначала посчитаем частичные производные.

$$\frac{\partial}{\partial w_j} L(w, x_1, ..., x_N) = (y \frac{1}{\sigma(w^T x)} - (1 - y) \frac{1}{1 - \sigma(w^T x)}) \frac{\partial}{\partial w_j} \sigma(w^T x) + 2\alpha w_j = \tag{1.20}$$

$$= (y \frac{1}{\sigma(w^T x)} - (1 - y) \frac{1}{1 - \sigma(w^T x)}) \sigma(w^T x)(1 - \sigma(w^T x)) \frac{\partial}{\partial w_j} w^T x + 2\alpha w_j = \tag{1.21}$$

$$= (y(1 - \sigma(w^T x)) - (1 - y)\sigma(w^T x)) x_j + 2\alpha w_j = (y - \widehat{y}) x_j + 2\alpha w_j \tag{1.22}$$

Формула градиент вектора:

$$\nabla w L(w, x_1, ..., x_N) = \sum_{i=1}^{N} (y - \sigma(w^T x)) x_i + 2\alpha w = (y - \widehat{y}) X + 2\alpha w \tag{1.23}$$

■

Задача. 8) Запишите формулу для обновления вектора параметров w при обучении методом градиентного спуска.
J(w) - cost function

$$w_j = w_j + \eta * \frac{\partial}{\partial w_j} * J(w) = w_j + \eta * \sum_{i=1}^{n} (y^{(i)} - \widehat{y}^{(i)}) * x_j^{(i)} - 2\eta * \alpha w_j \tag{1.24}$$

$$w = w + \eta * \sum_{i=1}^{n} (y^{(i)} - \widehat{y}^{(i)}) * x^{(i)} - 2\eta * \alpha w \tag{1.25}$$

■

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

$$BCE(w, X) = -\frac{1}{N}\sum_{i=1}^{N}(y * log\widehat{y} + (1-y) * log(1-\widehat{y})) \tag{1.26}$$

Для минимизации BCE(w,X) нужно максимизировать $\sum_{i=1}^{N}(y * log\widehat{y} + (1-y) * log(1-\widehat{y}))$, то есть максимизировать $y_i * log\widehat{y_i} + (1-y_i) * log(1-\widehat{y_i})$ для $\forall x_i$.

$$y * log\widehat{y} + (1-y) * log(1-\widehat{y}) = log(\widehat{y}^y * (1-\widehat{y})^{1-y}) \tag{1.27}$$

Логарифм максимален при максимальной подлогарифмической функции

$$(1-\widehat{y}) * (\frac{\widehat{y}}{1-\widehat{y}})^y \tag{1.28}$$

При $y_i = 1$ выражение равно $\widehat{y_i}$ и максимально при единственном значении $\widehat{y_i} = 1$. При $y_i = 0$ выражение равно $1 - \widehat{y_i}$ и максимально при единственном значении $\widehat{y_i} = 0$.

Т.к. $\widehat{y} = w^T x$ то $\exists$! вектор весов w: $w^T x = \widehat{y}$, где $y_i = \widehat{y_i}$. ■

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

$$softplus(x) = log(1 + e^x), t = 2y - 1 \in [-1; 1] \tag{1.29}$$

$$softplus'(x) = \frac{1}{1+e^x} * e^x = \frac{1}{1+e^{-x}} = \sigma(x) > 0 \forall x \tag{1.30}$$

Следовательно, функция softplus(x) монотонно возрастает и минимальна при минимальном x. Для минимизации $softplus(tw^T x)$ нужно максимизировать $(tw^T x)$

$$\widehat{y} = w^T x \tag{1.31}$$

$$tw^T x = \begin{cases} \widehat{y}, t = 1, y = 1; \\ -\widehat{y}, t = -1, y = -1 \end{cases}$$

$(tw^T x)$ максимальна, если $(w^T x) = 1$ при y = 1 и $(w^T x) = 0$ при y = 0. ■

2 Практическая часть

Задача. A) Чему равен размер получившегося словаря?

46376 - словарь позитивных отзывов.
44583 - словарь негативных отзывов.
62311 - словарь из всех отзывов.

Задача. B) 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?

В формате dense-матрицы занимает 7477320000 байт = 7302070 КБ = 7131 МБ.
В формате csr matrix:
data.nbytes = 18254512 байт = 17826,7 КБ = 17,4 МБ.
indices.nbytes = 9127256 байт = 8913,3 КБ = 8,7 МБ.
indptr.nbytes = 60004 байт = 58,6 КБ.

В матрице выборки много нулей, хранение каждого нуля неэффективно и требует много памяти. Поэтому при переходе к формату csr matrix, в котором хранятся индексы и информация лишь о ненулевых элементах выборки, экономится очень много места (в нашем случае примерно 7050 МБ!).

Задача. C) 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 привычислении оценочной фунции? Как реализовать функцию сигмоида, чтобы этого непроисходило?

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-[1;x]^T w}} \tag{2.1}$$

$\sigma(z) \longrightarrow 1.0$ при $e^{-z} \longrightarrow 0 \Leftrightarrow z \longrightarrow \infty \Leftrightarrow w \longrightarrow \infty$.

Причина может быть в начальной инициализации слишком большими весами, или в выборе слишком большого гиперпараметра, который используется при градиентном спуске (делаем слишком большой шаг при обновлении весов).

Задача. E) Сколько времени занимает прямой проход на обучающей выборке, реализованный с/безцикла по примерам? Чему равно значение оценочной функции сразу после инициали-зации?
Time direct pass cycle 10.834149837493896 fs.
Time direct pass 0.03544139862060547 fs.
Значение оценочной функции сразу после инициализации равно ≈ 0.301

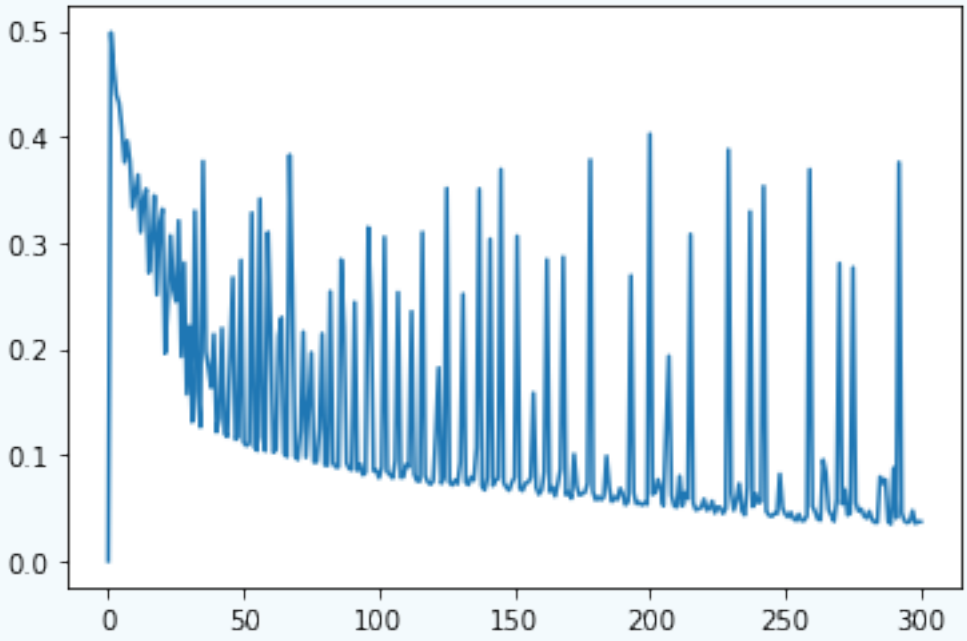
Задача. I) Приведите построенные графики обучения. Через сколько эпох обучение сходится? Какой точности классификатора вам удалось достичь на обучающей, валидационной, тестовой выборках? Имеет ли место переобучение классификатора или недообучение?

Достигается точность 0.999278 на 15000 эпохах, размер мини-батча 200. Стартовый learning rate = 0.1, если precision очень мало изменилось за очередные 500 эпох, то делим learning rate пополам.

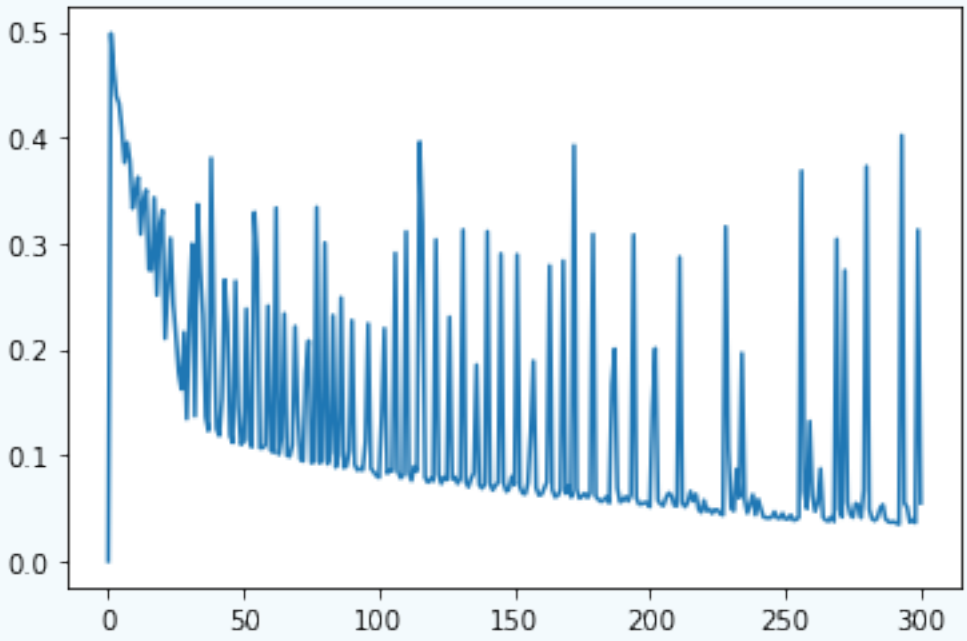
На тестовой выборке

Задача. J) Приведите графики обучения для нескольких различных значений learning rate. Какие выводы можно сделать?

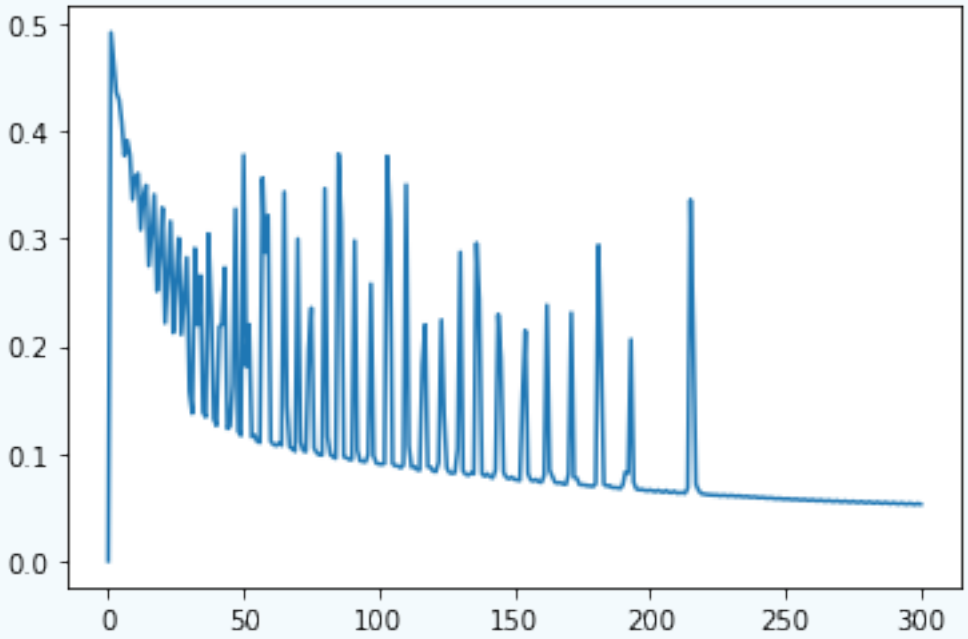
pic.1 - learning rate = 1



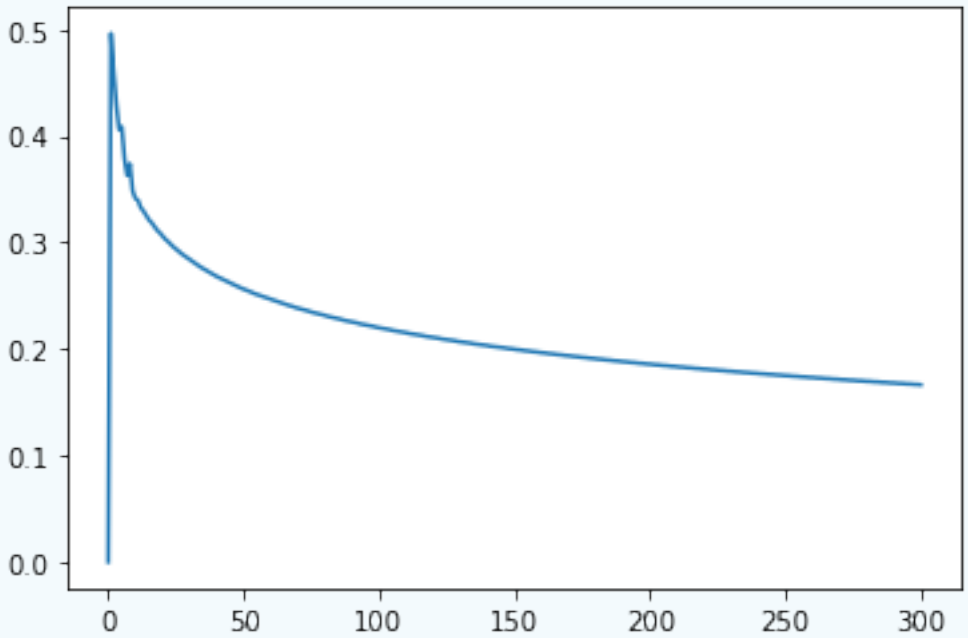
pic.2 - learning rate = 0.5



pic.3 - learning rate = 0.1



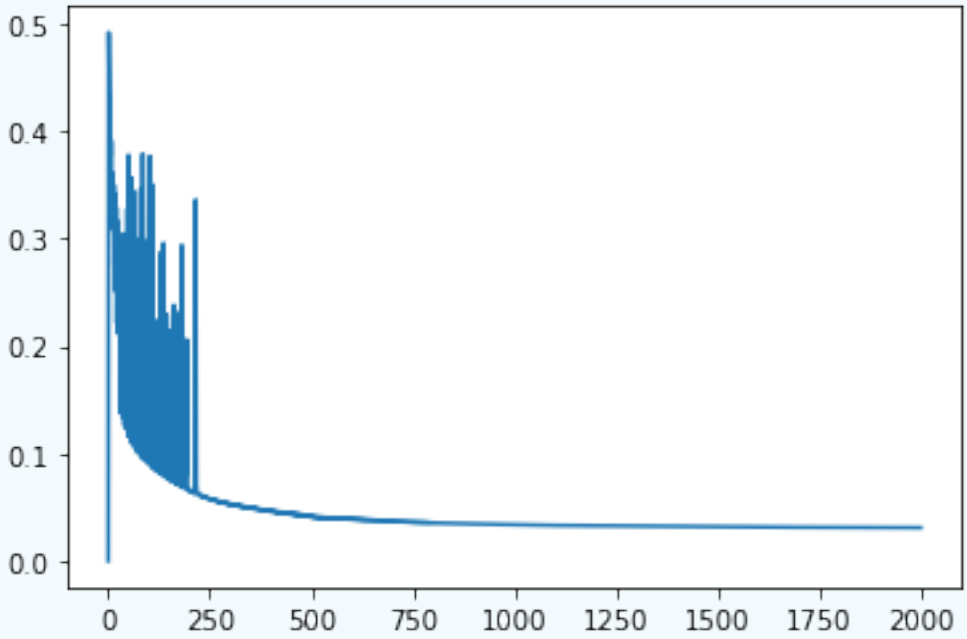
pic.4 - learning rate = 0.01



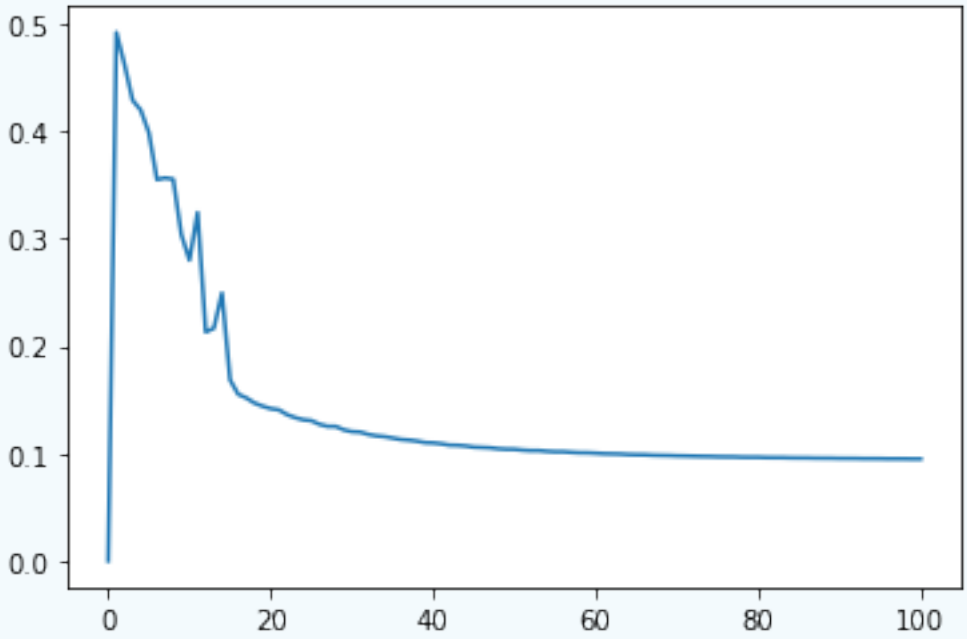
Выводы: learning rate не следует брать меньше 0.1, т.к. обучение останавливается. В начале обучение идет стабильно при любом learning rate, но все ранво придется его снижать. Неровности на графике обусловлены использованием mini-batch GD и выбором точек, по которым строится график (1 точка на 50 эпох): если для одной "порции"тестовых текстов веса подобраны, то для следующей порции, где встречаются другие признаки, обновляются уже другие веса.

Задача. К) 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?

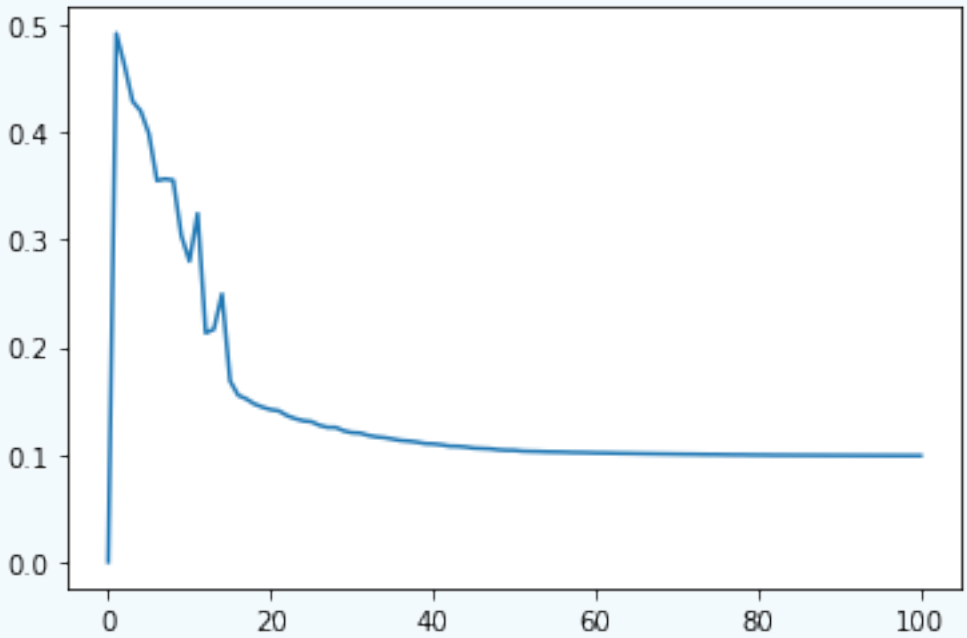
Learning rate для всех одинаковый = 0.1, снижается каждые 15000 эпох.
При $\alpha = 0$ достигается точность 0.97 на 90000-ых эпохах. При преодолении точности 0.95 даже при дальнейшем делении learning rate обучение заканчивается. Обучение длится приблизительно 5 минут.



При $\alpha = 0.5$ точность "застревает"на 0.64.
При $\alpha = 0.1$ точность "застревает"на 0.69.
При $\alpha = 0.01$ точность "застревает"на 0.82, набирая её очень быстро (на 300 эпохах!).



При $\alpha = 0.001$ точность "застревает"на 0.9, набирается на 3000 эпохах.

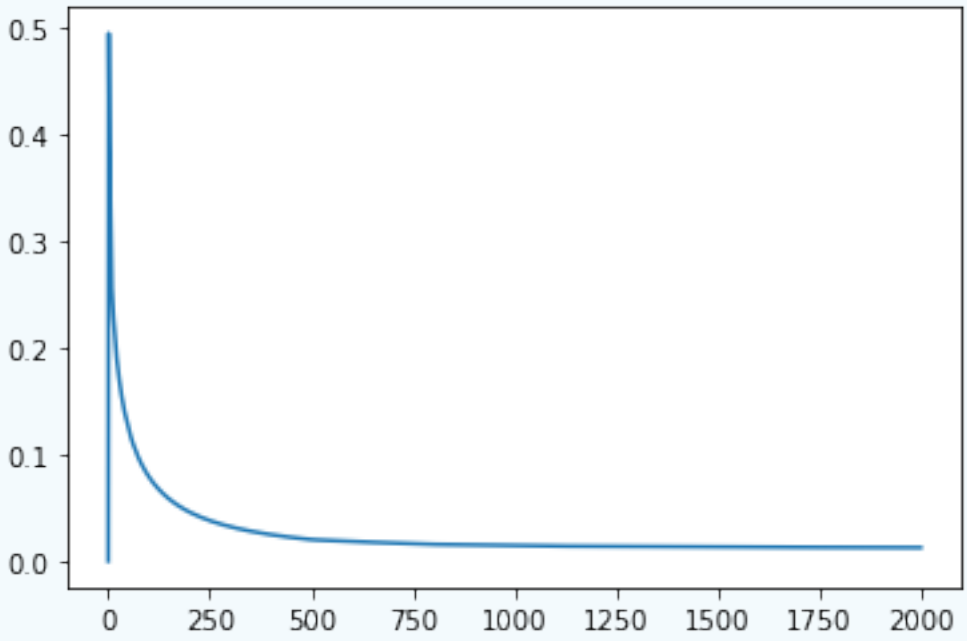


При $\alpha = 0.0001$ точность достигает 0.92 на 10000 эпохах и затем практически не меняется.
Вывод: стоит использовать небольшой коэффициент регуляризации в том случае, если мы хотим сильно увеличить скорость и не стремимся достигать высокой точности.

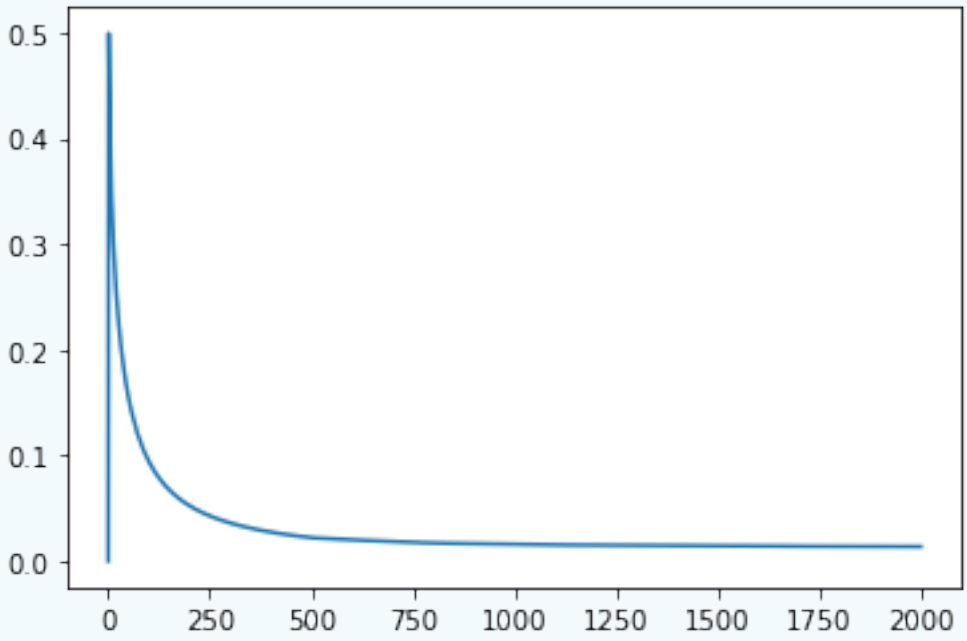
Задача. L) На сколько улучшается точность при добавлении n-грамм разного-порядка? Как потребовалось изменить гиперпараметры?

Одинаковый learning rate = 0.1.

При использовании 2-грамм обучение длится дольше, но график обучения плавнее. Точность 0.986 на 55000 эпохе, 0.987 на 79000 эпохе, затем обучение стагнирует даже при делении learning rate.



При использовании 3-грамм обучение длится еще дольше, график обучения еще плавнее. Точность 0.986 на 80000 эпохе, затем обучение тоже стагнирует.



Задача. M) Распечатайте 20 наиболее весомых признаков для позитивного и негативного класса.
For positive class:
10 2.296720875325847
enjoyed 2.3493946300026045

fantastic 2.360989922000899
especially 2.4367269148902593
reality 2.467372663309246
true 2.517198710013728
loved 2.593283221961914
performance 2.6443029170761734
quite 2.7722028231760754
strong 2.8086989628659955
job 2.826624111252118
music 2.8743733050164244
7 2.8758975113416794
best 3.1856650916429037
love 3.302227959080803
wonderful 3.3063230985704046
well 3.3305574873203296
fun 3.3683477585891923
perfect 4.277318399345455
excellent 5.163116460112307
great 5.959526180682657

For negative class:
poorly -2.597499878591322
looks -2.5800511220005244
horrible -2.5374856226600873
would -2.536074355401264
supposed -2.535400385521979
extremely -2.4885885161278316
stupid -2.454524445685155
could -2.4096960068423012
completely -2.3876856884350675
fails -2.3824570900552673
except -2.3622387987058326
wrong -2.344578150205949
interest -2.3329450839359405
badly -2.3192827752419207
none -2.3133013161325664
minutes -2.300133229237455
avoid -2.293135971513158
half -2.2844027846741266
pointless -2.221708554662847
even -2.1561158209134996
crap -2.1460337955604123