Толеутаева Анита, группа 208. Домашнее задаие 3.

Содержание

1 Теоретическая часть

Определения 1. Введем следующие обозначения: $(x_1, y_1), ..., (x_N, y_N)$ обучающая выборка размера N, $x_i \in R^{1xM}$ - i-й отзыв из выборки, M = $s^{(0)}$ - размерность входных векторов, $y_i \in 0, 1, ..., K-1$, $s^{(l)}$ - количество нейронов в i-м слое, $W^{(l)}$ - матрица параметров l-го слоя размера $(s^{(l-1)}+1) * s^{(l)}$ (т.к. мы добавляем смещение — bias), где l = 1, 2, $\cdots$, L, L - количество слоев (число скрытых слоев равно L-1).

Задача. 1) Посчитайте производную функции tanh(z) и выразите ее через саму функцию tanh(z), считая что z — скаляр. Преобразуйте ответ, так, чтобы при вычислении tanh(z) и ее производной была только одна операция экспоненцирования.

$$tanh'(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}' = \frac{(e^z + e^{-z})(e^z + e^{-z}) - (e^z - e^{-z})(e^z - e^{-z})}{(e^z + e^{-z})^2} = \tag{1.1}$$

$$= \frac{-1 - e^{-2z} - e^{2z}}{(e^z + e^{-z})^2} = 1 - tanh^2(z) \tag{1.2}$$

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{1 - e^{-2z}}{1 + e^{-2z}} = -1 + \frac{2}{1 + e^{-2z}} \tag{1.3}$$

■

Задача. 2) Воспользовавшись обозначениями, введенными выше, выпишите формулы прямого прохода (forward pass) и вычисления оценочной функции кросс-энтропия $ce(W^{(1)}, ..., W^{(L)}, x, y)$ для одного примера для полносвязной нейронной сети с L-1 скрытым слоем для случая многоклассовой классификации (считаем, что есть K взаимоисключающих классов). В качестве активации для скрытого слоя используется tanh(z), для выходного слоя — softmax(z).
Вектор 1-го скрытого слоя:

$$z^{(1)} = [1, tanh(W^{(1)}[1, x])] \tag{1.4}$$

Вектор 2-го скрытого слоя:

$$z^{(2)} = [1, tanh(W^{(2)} z^{(1)})] = [1, tanh(W^2 * [1, tanh(W^{(1)}[1, x])])] \tag{1.5}$$

Вектор L-го скрытого слоя:

$$z^{(L)} = [1, tanh(W^{(L)} * z^{(L-1)})] \tag{1.6}$$

Значение $softmax([1, tanh(W^L * z^{(L-1)})]) = softmax(z^{(L)}) =$

$$= \frac{exp(z^{(L)})}{\sum_{i=0}^{K-1} exp(z_i^{(L)})} \tag{1.7}$$

Значение $P(w_i|w_{i-n}, ..., w_{i-1}, w_{i+1}, ..., w_{i+n}) = argmax_i softmax(z^{(L)})$
Оценочная функция кросс энтропия:
ce$(\mathrm{W}^{(1)}, ..., W^{(L)}, x, y) = -\sum_i y_i * log(P(w_i|w_{i-n}, ..., w_{i-1}, w_{i+1}, ..., w_{i+n}))$
■

Задача. 3) Выпишите формулы прямого прохода и вычисления оценочной функции $CE(W^{(1)}, ..., W^{(L)}, x_1, ..., x_N, y_1, ..., y_N)$ для батча из N примеров в векторизованном виде (без цикла по примерам). Рядом с каждой формулой укажите размеры всех матриц. Считайте, что батч представлен матрицей X $\in R^{N*M}$ и матрицей one-hot векторов правильных ответов Y $\in R^{N*K}$.
Матрица 1-го скрытого слоя:

$$Z^{(1)} = [1, tanh(W^{(1)}[1, X])] \tag{1.8}$$

Матрица 2-го скрытого слоя:

$$Z^{(2)} = [1, tanh(W^{(2)}[1, Z^{(1)}])] = [1, tanh(W^2 * tanh(W^{(1)}[1, X]))] \tag{1.9}$$

Матрица L-го скрытого слоя:

$$Z^{(L)} = [1, tanh(W^{(L)} * Z^{(L-1)})] \tag{1.10}$$

$softmax(Z^{(L)}) = softmax(tanh(W^L * tanh(W^{L-1} * (... * (tanh(W^{(1)}x)))) = exp(ZL)/exp(ZL).sum(axis = 1)$

likelyhood $\widehat{Y} \in R^{N*K} = argmax_i softmax(Z^{(L)})$

Оценочная функция ($X \in R^{N*M}; Y \in R^{N*K}$):

$$CE(W^{(1)}, ..., W^{(L)}, x_1, ..., x_N, y_1, ..., y_N) = -\frac{1}{N}\sum_{i=1}^{N} y_i * log(likelyhood\widehat{y}_i) = -\frac{1}{N} Y * log(likelyhood\widehat{Y}) \tag{1.11}$$

■

Задача. 4) Покажите, что softmax(z + c) = softmax(z), где c – вектор, все компоненты которого равны. Как можно воспользоваться этим свойством при реализации softmax, чтобы не экспоненцировать большие положительные числа (что может привести к переполнению числа с плавающей точкой)?

$$softmax(z+c) = \frac{e^{z+c}}{\sum_{j=1}^{K} e^{z_j+c}} = \frac{e^z * e^c}{\sum_{j=1}^{K} e^{z_j} * e^c} = \frac{e^z}{\sum_{j=1}^{K} e^{z_j}} = softmax(z) \quad (1.12)$$

Можно вычитать из вектора z вектор c, где $\forall i c_i = \frac{\sum_i^K z_i}{K}$ - среднее арифметическое компонент вектора z, или максимальную компоненту вектора z.
■

Задача. 5) Посчитайте, сколько всего параметров содержится в полносвязной нейронной сети с L-1 скрытым слоем, если входные вектора имеют размерность M, выходные вектора - K, а в каждом скрытом слое - H нейронов.

$W^{(l)}$ - матрица параметров l-го слоя размера $(s^{(l-1)} + 1) * s^{(l)}$. Каждая матрица содержит $(s^{(l-1)} + 1) * s^{(l)}$ параметров, значит всего параметров (в общем виде):
$\sum_{i=1}^{L}(s^{(i-1)} + 1) * s^{(i)} = \sum_{i=1}^{L} s^{(i)} * s^{(i-1)} + \sum_{i=0}^{L} s(i)$
$= (L) * H^2 + (L+1) * H + M * H + K * H = H * (L * H + L + 1 + M + K)$
■

Задача. 6) Для случая одного входного примера выведите формулу для $\delta^{(L)}$ — градиента оценочной функции по предактивациям в последнем слое $z^{(L)}$. Сначала выведите формулу для одной компоненты, затем для всего вектора.
Формула для частной производной оценочной функции $L = \sum_i y_i log(p_i)$, где $p_i = argmax_i softmax(z^{(L)})$:

$$\frac{\partial L}{\partial z_j^{(L)}} = -\sum_i y_i \frac{\partial log(p_i)}{\partial z_j^{(L)}} = -\sum_i y_i \frac{\partial log(p_i)}{\partial p_i} * \frac{\partial p_i}{\partial z_j^{(L)}} = \quad (1.13)$$

$$= -\sum y_i * \frac{1}{p_i} * \frac{\partial p_i}{\partial z_j^{(L)}} = -y_j(1-p_j) - \sum_{i \neq j} y_i * \frac{1}{p_i}(-p_i * p_j) = p_j * (y_j + \sum_{i \neq 1} y_i) - y_j = p_j - y_i \quad (1.14)$$

Градиент вектор оценочной функции для последнего L-го слоя:
$\delta^{(L)} = P - Y = argmax_i softmax(Z^{(L)}) - Y$
■

Задача. 7) Для случая одного входного примера выведите формулу для подсчета $\delta^{(l)}$ — градиента оценочной функции по $z^{(l)}$ — через $\delta^{(l-1)}$. Сначала выведите формулу для одной компоненты, затем для всего вектора.
tanh(x) $= -1 + \frac{2}{1+e^{-2x}}$ (используем п.1).

$$tanh'(x) = 1 - tanh^2(x) = 1 - (-1 + \frac{2}{1+e^{-2x}})^2 \quad (1.15)$$

$$\frac{\partial Z^l}{\partial z_i^{(l-1)}} = \frac{\partial [1, W^{(l)} * tanh(Z^{(l-1)})]}{\partial z_i^{(l-1)}} = w_i^{(l)} * tanh'(z_i^{(l)}) \quad (1.16)$$

Формула для частной производной оценочной функции кросс-энтропия:

$$\delta_i^{(l-1)} = \frac{\partial(ce)}{\partial z_i^{(l-1)}} = \frac{\partial(ce)}{\partial Z^l} * \frac{\partial Z^l}{\partial z_i^{(l-1)}} = \delta^{(l)} * w_i^{(l-1)} * tanh'(z_i^{(l-1)}) = \tag{1.17}$$

$$= \delta^{(l)} * w_i^{(l-1)} * (1 - tanh^2(z_i^{(l-2)})) \tag{1.18}$$

Градиент вектор оценочной функции для l-го слоя через $\delta^{(l-1)}$:

$$\delta^{(l-1)} = \frac{\partial(CE)}{\partial Z^{(l-1)}} = \delta^{(l)} * W^{(l-1)} * tanh'(Z^{(l-1)}) = \delta^{(l)} * W^{(l-1)} * (1 - tanh^2(Z^{(l-2)}) \tag{1.19}$$

■

Задача. 8) Для случая одного входного примера выведите формулу для $\nabla W^{(l)}$ce — градиента оценочной функции по весам $W^{(l)}$, используя $\delta^{(l)}$. Сначала выведите формулу для одной компоненты, затем для всего вектора.
$\nabla W^{(l)}$ce - матрица H*L

$$\frac{\partial L(w)}{\partial w_j^{(l)}} = \frac{\partial L(w)}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_j^{(l)}} \tag{1.20}$$

Т.к. из п.7 $\delta_i^{(l-1)} = \frac{\partial(ce)}{\partial z_i^{(l-1)}}$ и $\frac{\partial z_j^{(l)}}{\partial w_j^{(l)}} = \frac{\partial[1;a^{(l-1)}]*W^{(l-1)}}{\partial w_j^{(l)}} = (a^{(l-1)})^T$, (bias при подсчете градиента не учитывается), то градиент для одной компоненты можно записать в виде:

$$\frac{\partial L(w)}{\partial w_j^{(l)}} = \delta_i^{(l-1)} * (a^{(l-1)})^T \tag{1.21}$$

Аналогично для вектора компонент:

$$\nabla W^{(l)} ce = \delta^{(l-1)} * (a^{(l-1)})^T \tag{1.22}$$

■

Задача. 9) По аналогии с предыдущим пунктом, для батча примеров выведите формулу для DW[l] = $\nabla W^{(l)}$CE через DZ[l] = $\nabla Z^{(l)}$CE. Сначала выведите формулу для одной компоненты, затем для всей матрицы в векторизованном виде. Выпишите размеры всех матриц.
По условию, компонентами вектора DZ[l] являются векторы $\delta_i^{(l-1)}$ размером K => DZ[l] имеет размер NxK

$$DZ[l] = \nabla Z^{(l)} CE = [\delta_1^{(l-1)}, ..., \delta_N^{(l-1)}] \tag{1.23}$$

Из пункта п.8 (где нижний индекс - номер примера в батче, размер 1xK):

$$\nabla W^{(l)} ce_i = \delta_i^{(l-1)} * (a^{(l-1)})^T \tag{1.24}$$

Аналогично для всего вектора DW[l] с размером NxK:

$$DW[l] = \nabla W^{(l)} CE = [\nabla W^{(l)} ce_1, ..., \nabla W^{(l)} ce_N] = (a^{(l-1)})^T [\delta_1^{(l-1)}, ..., \delta_N^{(l-1)}] = \tag{1.25}$$

$$= (a^{(l-1)})^T DZ[l]$$

■

Задача. 10) Выпишите все формулы обратного прохода для батча в векторизованном виде. Для этого необходимо выразить DW[l] = $\nabla W^{(l)}$CE через матрицы X, Y, $\widehat{(Y)}$, A[l], Z[l], с помощью выведенных в предыдущих пунктах рекуррентных соотношений и вычисляя DZ[l] = $\nabla Z^{(l)}$CE в качестве промежуточных значений. Чтобы ускорить обратный проход, старайтесь переиспользовать матрицы выходов, активаций и предактиваций, вычисленные на прямом проходе. Рядом с каждой формулой выпишите размеры всех матриц. Чтобы упростить процесс реализации, рекомендуем выводить размерности всех вычисляемых матриц и сравнивать с выписанными.
Размерность DW[I] равна размерности матрицы весов между i-1'м и i'м слоями (то есть $R^{s^{(i-1)} * s^{(i)}}$).
Размерность DZ[I] равна размерности i'го слоя (то есть $R^{1 * s^{(i)}}$).

1. Из п.6 и п.9 т.к. $\delta_i^{(l-1)} = \widehat{(y)}_i - y_i$, то

$$DZ[L] = [\delta_1^{(l-1)}, ..., \delta_N^{(l-1)}] = \widehat{(Y)} - Y \tag{1.26}$$

2. DW[L] - вектор, компонентами которого являются векторы $DW[L]_i = (a_i^{(l-1)})^T DZ[l]$, i = 1, K. Длина каждого вектора - компоненты равна длине l-1-го слоя S(l-1), размер матрицы DW[L] равен S(l-1)xK.

$$DW[L] = (A^{(L-1)})^T DZ[L] \tag{1.27}$$

3. DZ[L-1] = $[\delta_1^{(l-2)}, ..., \delta_N^{(l-2)}]$
Из п.7. $\delta_i^{(l-2)} = \delta_i^{(l-1)} * w_i^{(l-2)} * tanh'(z_i^{(l-1)})$
Выражение для DZ[L-1] (где $\widehat{(W)}$ - матрица весов W без 1й строки):

$$DZ[L-1] = DZ[L] * \widehat{(W)}^{(L-1)} tanh'(Z^{(L-1)}) = DZ[L] * \widehat{(W)}^{(L-1)} (1 - tanh^2(Z^{(L-2)}) \tag{1.28}$$

4. По аналогии с DW[L] запишем выражение для DW[L-1]:

$$DW[L-1] = (A^{(L-2)})^T DZ[L-1] \tag{1.29}$$

5. Формулы в общем виде для i = [1, L-1]:
DZ[i] = DZ[i+1]*$\widehat{(W)}^{(i)}(1 - tanh^2(Z^{(i-1)}))$
$DW[i] = (A^{(i-1)})^T DZ[i]$
В частности, при i = 1: DZ[1] = DZ[2]*$\widehat{(W)}^{(1)}(1 - tanh^2(X)$
$DW[1] = (X)^T DZ[1]$
■

2 Практическая часть

Задача. А.1) Покажите, что косинус угла между векторами совпадает с их скалярным произведением, если вектора предварительно нормировать (поделить на евклидову норму). Выведите формулу, выражающую евклидово расстояние через косинус угла между 2 векторами для нормированных векторов.
x = $(x_1, ..., x_N)$
$y = (y_1, ..., y_N)$
$|x|^2 = x_1^2 + ... + x_N^2$
$|y|^2 = y_1^2 + ... + y_N^2$
скалярное произведение в Евклидовом пространстве:
$(x, y) = |x| * |y| * cos(\widehat{x, y}) = x_1 * y_1 + ... + x_N * y_N$
После нормирования (по свойству аддитивности векторов):
$x' = \frac{x}{|x|} = 1$
$y' = \frac{y}{|y|} = 1$
$(x', y') = |x'| * |y'| * cos(\widehat{x', y'}) = cos(\widehat{x', y'}) = x'_1 * y'_1 + ... + x'_N * y'_N$
Так как при умножении вектора на действительное число его направление не изменяется, то $cos(\widehat{x', y'}) = cos(\widehat{x, y})$
$(\frac{x}{|x|}, \frac{y}{|y|}) = cos(\widehat{x, y})$
■

Задача. А.2) Выберите 10 слов, начинающихся с первых двух букв (to) вашей фамилии в латинской транскрипции. Для выбранных вами слов найдите 15 ближайших слов. Сравните результаты при использовании в качестве меры близости скалярного произведения и косинуса угла между векторами – для этого разместите их в соседних столбцах таблицы. В каждой ячейке таблицы приведите исходное слово, ближайшие слова, отсортированные по убыванию меры близости, и значения меры близости.

1) toyota (3952)

01.	0.19188787	honda
02.	0.32948746	automaker
03.	0.33425330	nissan
04.	0.37386258	bmw
05.	0.37759106	auto
06.	0.38941322	motors
07.	0.39937827	ford
08.	0.41469539	motor
09.	0.42847858	renault
10.	0.43491645	mercedes
11.	0.43785128	mazda
12.	0.44769170	benz
13.	0.45048700	volkswagen
14.	0.45078808	chrysler
15.	0.48108041	daimlerchrysler
16.	0.48114043	prius

2) tomorrow (4003)

01. 0.29073957 wait
02. 0.35858200 wo
03. 0.37066445 ’ll
04. 0.37488164 happen
05. 0.38576409 go
06. 0.39120292 expect
07. 0.40014338 hopefully
08. 0.40554535 going
09. 0.41536511 tonight
10. 0.42795172 sooner
11. 0.43829401 ready
12. 0.43901382 anytime
13. 0.44933458 next
14. 0.46031001 happens
15. 0.47568245 we
16. 0.48288153 anyway

3) tools (4316)
01. 0.21921573 tool
02. 0.36978801 techniques
03. 0.40006313 methods
04. 0.41586617 hardware
05. 0.45092187 applications
06. 0.45885798 software
07. 0.46184336 using
08. 0.48825027 devices
09. 0.49035436 materials
10. 0.49986139 use
11. 0.50207173 computer
12. 0.50950980 uses
13. 0.51322453 processing
14. 0.51334177 utilizing
15. 0.51739933 technology
16. 0.52875206 utilize

4) totally (4367)
01. 0.17007319 completely
02. 0.23814503 utterly
03. 0.27396405 absolutely
04. 0.33604531 practically
05. 0.34195747 entirely
06. 0.35330127 basically
07. 0.37168648 useless
08. 0.38144227 otherwise
09. 0.40988821 virtually
10. 0.41110973 irrelevant
11. 0.43049251 unfortunately
12. 0.43142089 obviously
13. 0.43222930 morally

14. 0.44209680 terribly
15. 0.44978540 essentially
16. 0.46337807 quite

5) tonight (4386)

01. 0.37405679 night
02. 0.41536511 tomorrow
03. 0.49245519 watching
04. 0.49983370 happy
05. 0.50503348 'll
06. 0.50863290 finale
07. 0.51963209 everybody
08. 0.54893796 going
09. 0.55427217 nights
10. 0.55799631 maybe
11. 0.57399466 moment
12. 0.57609909 play
13. 0.58182758 starts
14. 0.58218137 go
15. 0.58627871 talk
16. 0.58714943 show

6) tower (2936)

01. 0.24580864 towers
02. 0.41316096 gate
03. 0.42543408 building
04. 0.43926433 built
05. 0.44383099 roof
06. 0.50249964 skyscraper
07. 0.50570814 constructed
08. 0.51693122 dome
09. 0.53805289 facade
10. 0.54395812 entrance
11. 0.56783493 buildings
12. 0.56947461 bridge
13. 0.57326248 erected
14. 0.57642795 walls
15. 0.58536249 lighthouse
16. 0.58856896 adjacent

7) tourists (2698)

01. 0.31399424 tourist
02. 0.33101523 visitors
03. 0.38698809 foreigners
04. 0.41789677 travellers
05. 0.46751792 locals
06. 0.47152904 vacationers
07. 0.47609301 stranded
08. 0.47950315 migrants

09. 0.49737219 destination
10. 0.4974854 traveling
11. 0.51169143 arrivals
12. 0.51692958 travelers
13. 0.51693684 travelling
14. 0.52024309 holidaymakers
15. 0.52238137 seekers
16. 0.54795278 expatriates

8) town (2485)
01. 0.17170352 villages
02. 0.30160073 town
03. 0.31650246 area
04. 0.32440404 neighborhoods
05. 0.33614868 cities
06. 0.34724622 northern
07. 0.35290148 suburbs
08. 0.35676446 areas
09. 0.37984657 populated
10. 0.38991531 southern
11. 0.40474140 village
12. 0.42203706 eastern
13. 0.42785037 roads
14. 0.44058102 communities
15. 0.44250837 residents
16. 0.44293711 neighbouring

9) to (5)
01. 0.13099856 take
02. 0.14494437 would
03. 0.16498682 instead
04. 0.16842757 could
05. 0.17962424 .
06. 0.20521863 for
07. 0.20726882 should
08. 0.20799549 while
09. 0.20805158 will
10. 0.21098084 taken
11. 0.21113584 but
12. 0.21179232 put
13. 0.21498517 they
14. 0.21680426 it
15. 0.21766360 move
16. 0.21842485 only

10) told(155)
01. 0.18017951 reporters
02. 0.19425428 said
03. 0.25387832 asked

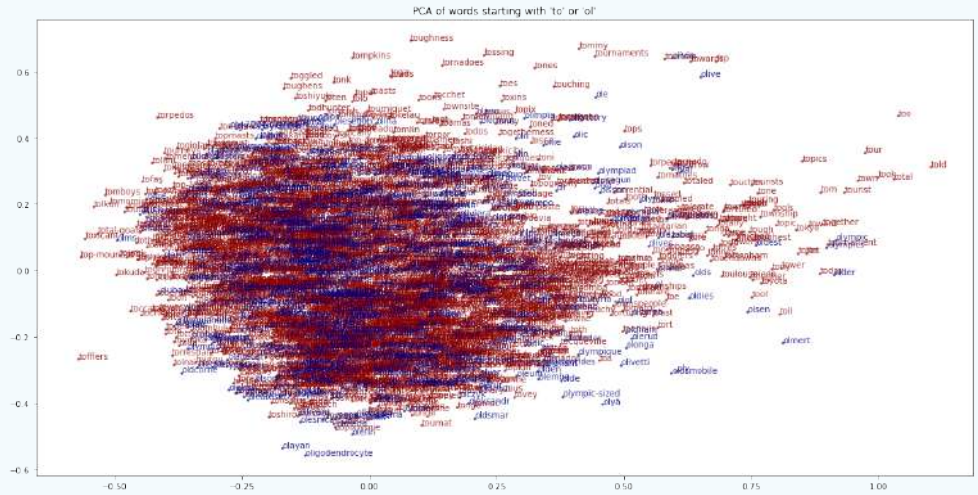
04. 0.26514014 saying
05. 0.27482457 quoted
06. 0.28773194 afp
07. 0.33985660 spokesman
08. 0.34779042 interview
09. 0.38802742 says
10. 0.40273428 telling
11. 0.40542880 statement
12. 0.42600188 informed
13. 0.43437441 spoke
14. 0.43476045 met
15. 0.43559644 insisted
16. 0.43643566 chief

Задача. A.3) Найдите 50 пар максимально близких друг к другу слов: $argmax_{w_i,w_j:i<j} sim(w_i, w_j)$ (1)Сравните результаты при использовании в качестве меры близости скалярного произведения и косинуса угла между векторами. Приведите сами пары, отсортированные по убыванию меры близости, и значения меры близости.

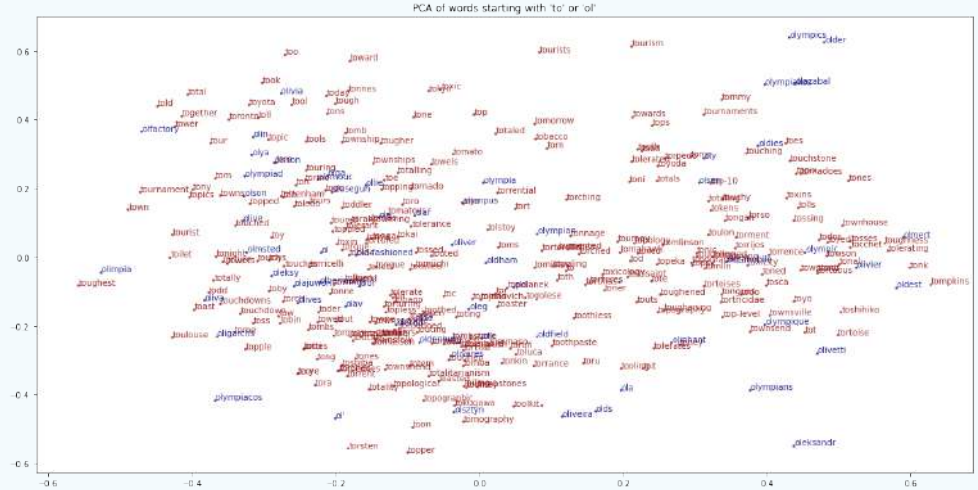
0.1234542199161225 tuesday thursday ———————— 0 0.1234542199161225 monday news ———————— 1 0.13655385795495997 tuesday friday ———————— 2 0.13655385795495997 thursday news ———————— 4 0.14479573897835998 wednesday news ———————— 6 tuesday [0. 0.12345422 0.13655386 0.14479574] ['tuesday', 'monday', 'thursday', 'wednesday']

Задача. A.4) Визуализируйте эмбеддинги всех слов, начинающихся с первых двух и со вторых двух букв вашей фамилии в латинской транскрипции (to, ol). Для визуализации воспользуйтесь методами снижения размерности и постройте диаграмму рассеивания (scatterplot). Для каждого эмбеддинга на графике добавьте подпись соответствующего слова. Для снижения размерности можно воспользоваться методом анализа главных компонент (Principal Component Analysis, PCA) из sklearn (см. пример). Выполняются лисвойства дистрибутивной семантики? Какие слова группируются в кластеры?

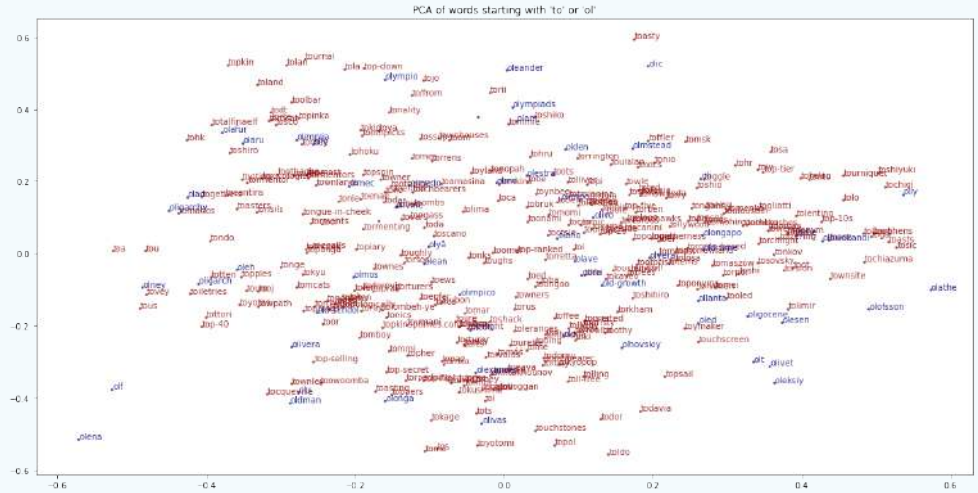
Т.к. слов, удовлетворяющих условию, оказалось много (порядка 3000), их невозможно различить:



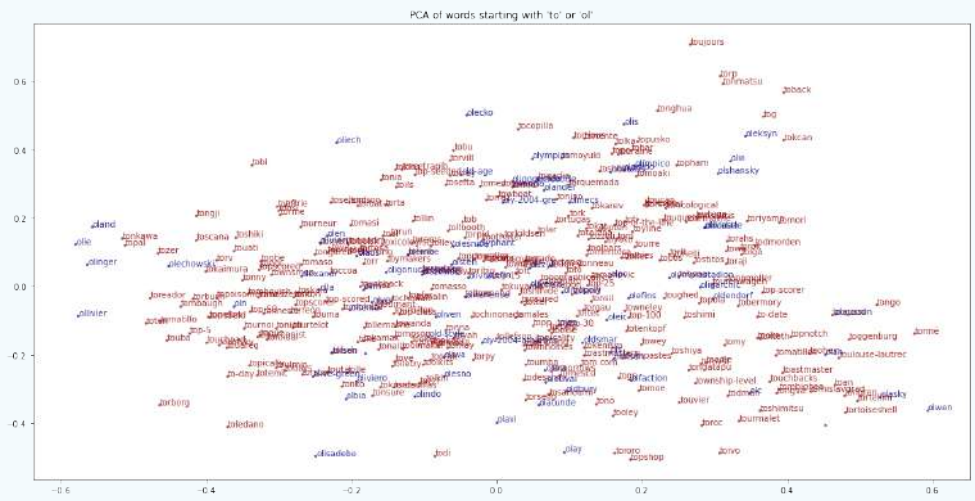
Поэтому было принято решение визуализировать несколько батчей слов из выборки: первые 50000:



вторые 50000:



и третьи 50000:



Из графиков видно, что в кластеры образуются, в первую очередь, слова из одной языковой группы (на 4 слайде около координат -0.4, 0.2 слова tongji, toshiki, touati, tokaimura).
Также рядом расположены родственные слова (на 2 слайде около координат 0.5, 0.6 слова olimpics и olimpiada, левее от них - пара слов tourists, tourism).

Задача. D) Для скольких токенов и какого количество уникальных слов из обучающей и тестовой выборки вы не нашли GloVe эмбеддинги? Приведите 20 примеров таких слов.
15341 ненайденных слов. В основном это слова не из английского языка, опечатки, имена собственные и сложные термины.

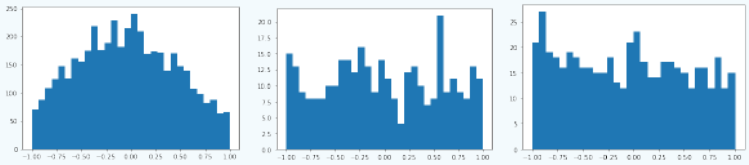
1. westfront
2. jáaccuse
3. overdramaticizing
4. bijomaru
5. uncapturable
6. kabuliwallah
7. 100min
8. unphilosophical
9. hollywoodized
10. maclachalan
11. dogmatists
12. yôko
13. jetée
14. sssssssssssoooooooooooo
15. einstien
16. inian
17. 60ish
18. trelkovsky
19. distiguished
20. waaaaay

Задача. G) Проведите gradient checking на каждом слое нейронной сети. С какой точностью сходятся градиенты, посчитанные численно, с градиентами, полученными из backpropagation модуля?

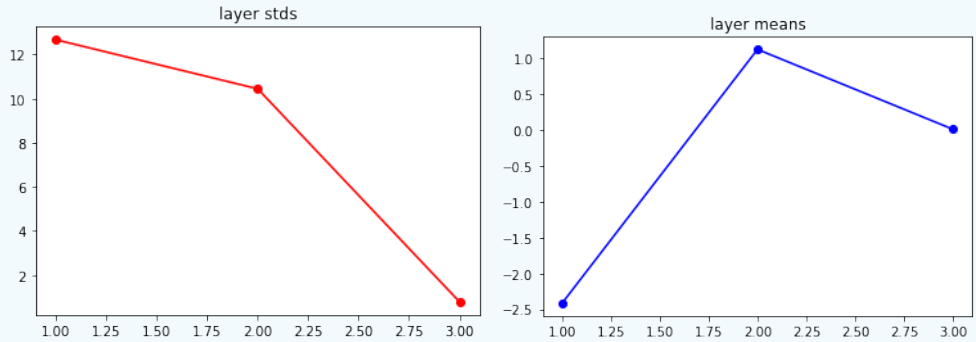
Отклонение градиента 1 и 2 матрицы весов, посчитанного численно, от градиента из backpropagation модуля, приблизительно равны:
42.27
83.44
и не меняются с течением обучения. Отклонение считается по формуле np.sum(abs(correct - grad).sum(axis = 0)/correct.shape[1])

Задача. E.1) Сразу после случайной инициализации чему равно матожидание $\widehat{(y)}$(x)?Чему в среднем равно матожидание оценочной функции без регуляризации? Вычисли-те значение оценочной функции без регуляризации на обучающей выборке, чему оно равно?
math expectation $\widehat{(y)}$(x) = 7521 примерно равно половине размера выборки
math expectation of loss function = -log(1/2) = 0,693
loss 0.015245313412279776

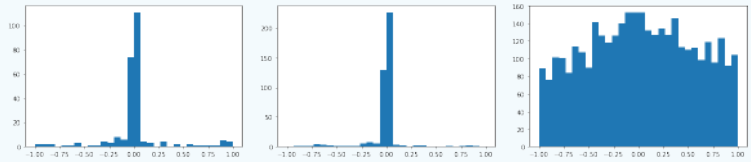
E.2) Постройте гистограммы, показывающие, как распределены компоненты входных векторов в зависимости от слоя нейросети, а также графики изменения среднего значения и дисперсии в зависимости от слоя. Для построения распределений воспользуйтесь функцией hist (см. пример), а для графиков зависимости среднего и дисперсииот номера слоя функцией – plot из библиотеки matplotlib (см. пример). Как меняются распределения с увеличением номера слоя? Попробуйте уменьшить и увеличить начальные случайные веса в 100 раз и постройте такие же графики. Как они изменились? Какой эффект оказывает на обучение сети инициализация весов слишком маленькимиили слишком большими значениями и почему?
Гистограммы 1 и 2 (скрытых слоев) и 3 (выходного) слоя, инициализация Xavier:



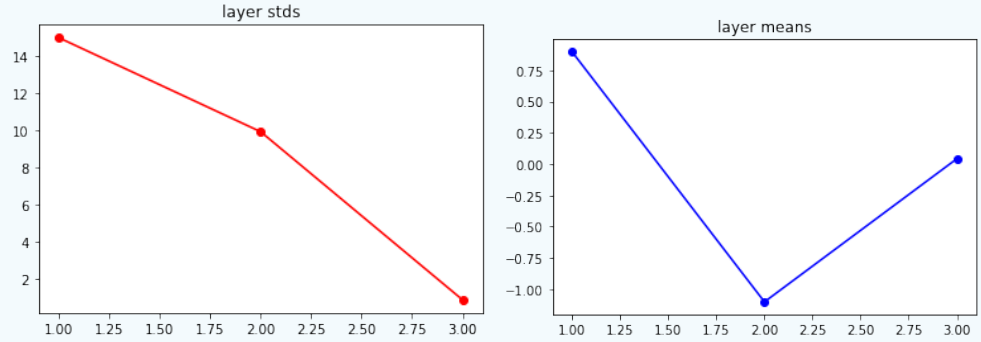
Графики изменения среднего значения и дисперсии в зависимости от слоя:



Гистограммы 1 и 2 (скрытых слоев) и 3 (выходного) слоя после увеличения начальных случайных весов в 100 раз:

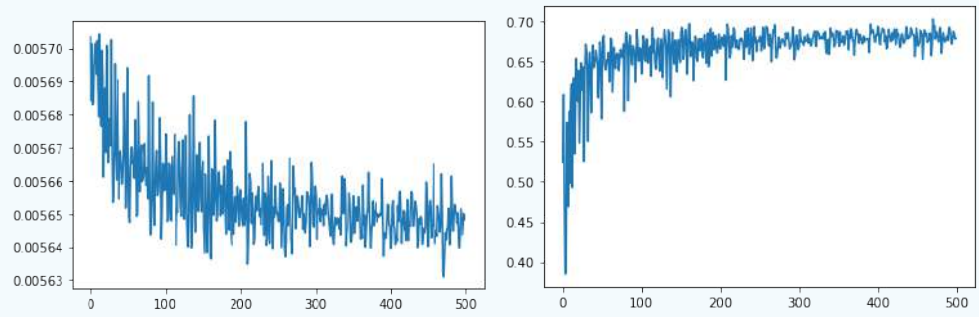


Графики изменения среднего значения и дисперсии:



Задача. J.1) Нарисуйте два графика, показывающих изменение оценочной функции и точности классификатора в процессе обучения(графики обучения), на каждом графике нарисуйте кривые для train (метрики на каждом батче) и dev (метрики на полном dev в начале каждой эпохи). Через сколько эпох обучение сходится (оценочная функция перестает меняться)?

График оценочной функции и точности при даных гиперпараметрах (learning rate 0.01, коэффициент регуляризации α 1e-5):

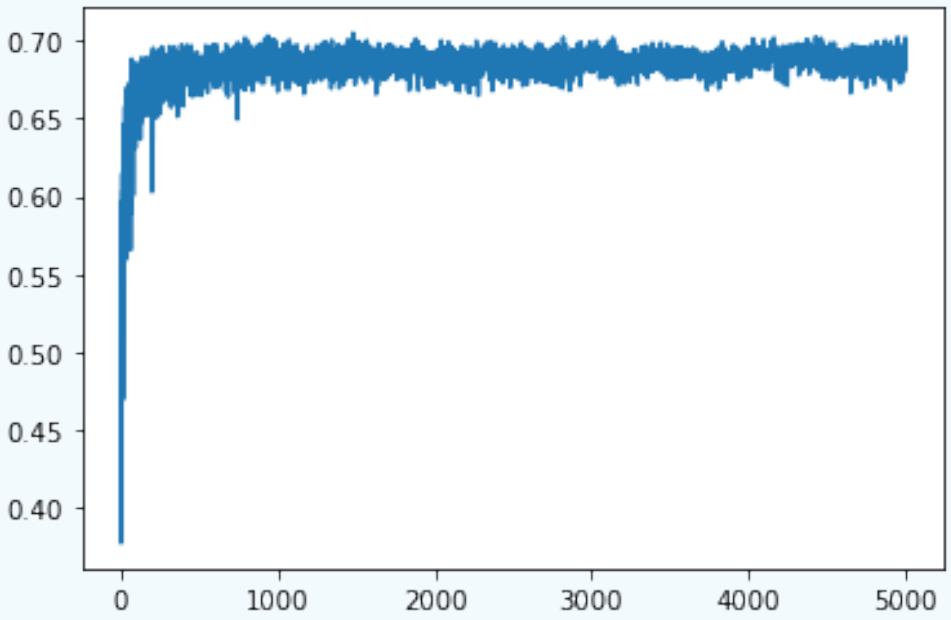


J.2) Какой точности классификатора вам удалось достичь на обучающей и тестовой выборке? Имеет ли место переобучение классификатора или недообучение? Что нужно сделать с α, чтобыулучшить результат?

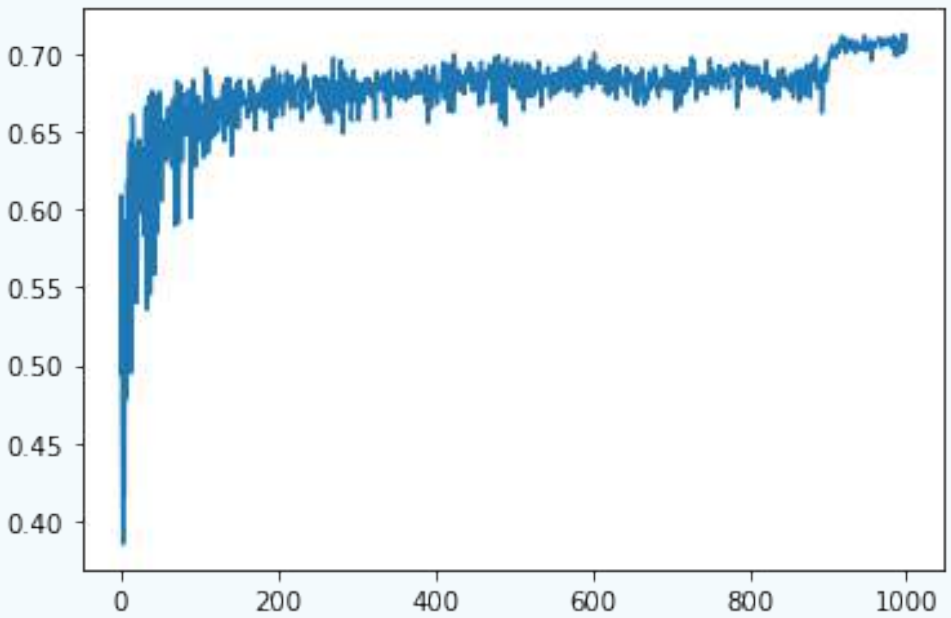
Точность на обучающей выборке: 0.75

Точность на тестовой выборке: 0.65

Нет, но без обновления параметров обучение стагнирует после 500-той эпохи, точность и loss колеблются вокруг одного значения (в частности, точность около 0.7).



Задача. K) Чему равно оптимальное значение α? Какое потребовалось число эпох и learning rate для обучения до сходимости?
α = 1e-5
learning rate в начале обучения 0.1, затем его стоит динамически изменять (уменьшать, например, в 2 раза) в процессе обучения.



Задача. L.1) Оцените чему равна точность классификатора на обучающей и тестовой выборке?
0.75 на обучающей выборке, 0.65 на тестовой.

L.2) Сколько времени занимает обучение классификатора и предсказание результатов для тестовой выборке?
5-7 минут длится обучение, приблизительно 3 минуты длится классификация.