# 1 Theoretical part

1. Tanh derivative.

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

2. Forward pass.

$$a^{(0)} = x_{\{i\}}$$
$$z^{(k)} = \left[1, a^{(k-1)}\right] W^{(k)}, k = 1, \ldots, L$$
$$a^{(k)} = \tanh(z^{(k)}), k = 1, \ldots, L-1$$
$$a^{(L)} = \mathrm{softmax}(z^{(L)})$$
$$ce = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\left[a^{(L)}\right]_t\right), t = \mathrm{argmax}(y_{\{i\}}), \text{ given y is one-hot.}$$

3. Now with matrices.

$$X \in \mathbb{R}^{N\times M};$$
$$A^{(0)} = X;$$
$$A^{(k)} = \tanh(Z^{(k)}), k = 1, \ldots, L-1;$$
$$A^{(L)} = \mathrm{softmax}(Z^{(L)}).$$

$$W^{(1)} \in \mathbb{R}^{(M+1)\times S_1};$$
$$W^{(k)} \in \mathbb{R}^{(S_{k-1}+1)\times S_k}, k = 2, \ldots, L-1;$$
$$W^{(L)} \in \mathbb{R}^{(S_{L-1}+1)\times K}.$$

$$Z^{(k)} = [I_1, A^{(k-1)}]W^{(k)}, k = 1, \ldots, L;$$
$$I_1 = \{1\} \in \mathbb{R}^{N\times 1};$$
$$Z^{(k)} \in \mathbb{R}^{N\times S_k}, k = 1, \ldots, L-1;$$
$$Z^{(L)} \in \mathbb{R}^{N\times K}.$$

$$Y \in \mathbb{R}^{N\times K}, Y \text{ is a one-hot matrix;}$$
$$CE = -\frac{1}{N}\sum_{i=1}^{N} \log(A^{(L)}(Y)).$$

4. Softmax

$$[\text{softmax}(z+c)]_k = \frac{e^{z_k+c}}{\sum_{i=1}^K e^{z_i+c}} = \frac{e^{z_k}}{\sum_{i=1}^K e^{z_i}} = [\text{softmax}(z)]_k$$

5. Amount of parameters is how many weights there are in all $W^{(i)}$ matrices.

$$Amount = (M+1)H + (H+1)H(L-2) + (H+1)K$$

6. Cross entropy gradient for $z^{(L)}$.

$$CE = -\frac{1}{N}\sum_{i=1}^N \log\left(\frac{< e^{z_i^{(L)}}, y_i >}{\sum_{j=1}^K e^{[z_i^{(L)}]_j}}\right) = -\frac{1}{N}\sum_{i=1}^N \left( < z_i^{(L)}, y_i > - \log\left(\sum_{j=1}^K e^{[z_i^{(L)}]_j}\right)\right),$$

where y is a one-hot vector.

$$\delta_{ij}^{(L)} = \frac{\partial CE}{\partial [z_i^{(L)}]_j} = -\frac{1}{N}\left(y_{ij} - \frac{[e^{z_i^{(L)}}]_j}{\sum_{t=1}^K e^{[z_i^{(L)}]_t}}\right)$$

$$\delta^{(L)} = \frac{1}{N}(\text{softmax}(z^{(L)}) - y)$$

7. Cross entropy gradient for $z^{(l)}$, knowing $\delta^{(l+1)}$.

$$z^{(l+1)}(z_i^{(l)}) = [1, \tanh(z_i^{(l)})]W^{(l+1)}$$

$$\delta_{ij}^{(l)} = \frac{\partial CE}{\partial [z_i^{(l)}]_j} = \frac{\partial CE}{\partial [z_i^{(l+1)}]_1}\frac{\partial [z_i^{(l+1)}]_1}{\partial [z_i^{(l)}]_j} + \frac{\partial CE}{\partial [z_i^{(l+1)}]_2}\frac{\partial [z_i^{(l+1)}]_2}{\partial [z_i^{(l)}]_j} + \cdots + \frac{\partial CE}{\partial [z_i^{(l+1)}]_{S_{l+1}}}\frac{\partial [z_i^{(l+1)}]_{S_{l+1}}}{\partial [z_i^{(l)}]_j} =$$

$$= \delta_{i1}^{(l+1)}\frac{\partial [z_i^{(l+1)}]_1}{\partial [z_i^{(l)}]_j} + \delta_{i2}^{(l+1)}\frac{\partial [z_i^{(l+1)}]_2}{\partial [z_i^{(l)}]_j} + \cdots + \delta_{i,(S_{l+1})}^{(l+1)}\frac{\partial [z_i^{(l+1)}]_{S_{l+1}}}{\partial [z_i^{(l)}]_j}$$

$$[z_i^{(l+1)}]_t = w_{1t}^{(l+1)} + w_{2t}^{(l+1)}\tanh([z_i^{(l)}]_1) + w_{3t}^{(l+1)}\tanh([z_i^{(l)}]_2) + \\ + \cdots + w_{(j+1),t}^{(l+1)}\tanh([z_i^{(l)}]_j) + \cdots + w_{(S_l+1),t}^{(l+1)}\tanh([z_i^{(l)}]_{S_l})$$

$$\frac{\partial [z_i^{(l+1)}]_t}{\partial [z_i^{(l)}]_j} = w_{(j+1),t}^{(l+1)}(1 - \tanh^2([z_i^{(l)}]_j))$$

$$\delta_{ij}^{(l)} = \delta_{i1}^{(l+1)} w_{(j+1),1}^{(l+1)}(1 - \tanh^2([z_i^{(l)}]_j)) + \delta_{i2}^{(l+1)} w_{(j+1),2}^{(l+1)}(1 - \tanh^2([z_i^{(l)}]_j)) + \cdots + \\ + \delta_{i,(S_{l+1})}^{(l+1)} w_{(j+1),(S_{l+1})}^{(l+1)}(1 - \tanh^2([z_i^{(l)}]_j))$$

$$\delta^{(l)} = \delta^{(l+1)} W^{(l+1)^T} \odot (1 - \tanh^2(z^{(l)})).$$

$\odot$ — element-wise multiplication.

8. Cross entropy gradient for $W^{(l)}$, knowing $\delta^{(l)}$.
   Let $a_{i0}^{(l-1)} = 1, \quad i = \overline{1, S_{l-1}}$.

$$[z_k^{(l)}]_j = [a_{k0}^{(l-1)}, a_{k1}^{(l-1)}, \ldots, a_{k,(S_{l-1})}^{(l-1)}] \cdot [w_{1j}^{(l)}, w_{2j}^{(l)}, \ldots, w_{(S_{l-1}+1),j}^{(l)}]^T =$$

$$= a_{k0}^{(l-1)} w_{1j}^{(l)} + a_{k1}^{(l-1)} w_{2j}^{(l)} + \cdots + a_{k,(i-1)}^{(l-1)} w_{ij}^{(l)} + \cdots + a_{k,(S_{l-1})}^{(l-1)} w_{(S_{l-1}+1),j}^{(l)}$$

$$\frac{[z_k^{(l)}]_j}{\partial w_{ij}^{(l)}} = a_{k,(i-1)}^{(l-1)}$$

$$\frac{\partial CE}{\partial w_{ij}^{(l)}} = \frac{\partial CE}{\partial [z_1^{(l)}]_j} \frac{\partial [z_1^{(l)}]_j}{\partial w_{ij}^{(l)}} + \frac{\partial CE}{\partial [z_2^{(l)}]_j} \frac{\partial [z_2^{(l)}]_j}{\partial w_{ij}^{(l)}} + \cdots + \frac{\partial CE}{\partial [z_N^{(l)}]_j} \frac{\partial [z_N^{(l)}]_j}{\partial w_{ij}^{(l)}} =$$

$$= \delta_{1j}^{(l)} a_{1,(i-1)}^{(l-1)} + \delta_{2j}^{(l)} a_{2,(i-1)}^{(l-1)} + \cdots + \delta_{Nj}^{(l)} a_{N,(i-1)}^{(l-1)}$$
$$\nabla_{W^{(l)}} CE = [I, A^{(l-1)}]^T \delta^{(l)}$$

9. Backpropagation formulas

$$\delta^{(L)} = \frac{1}{N}(\mathrm{softmax}(z^{(L)}) - y)$$
$$\delta^{(l)} = \delta^{(l+1)} W^{(l+1)^T} \odot (1 - \tanh^2(z^{(l)})).$$
$\odot$ — element-wise multiplication.
$$\nabla_{W^{(l)}} CE = [I, A^{(l-1)}]^T \delta^{(l)}$$

# 2 Practical part

A. 1. Let $\alpha$ be the angle between vectors a and b.

$$\left\langle \frac{1}{\|a\|} a, \frac{1}{\|b\|} b \right\rangle = \frac{\|a\| \, \|b\|}{\|a\| \, \|b\|} \cos(\alpha) = \cos(\alpha)$$
$$\|a - b\| = \sqrt{\langle a - b, a - b \rangle} = \sqrt{\|a\|^2 - 2 \langle a, b \rangle + \|b\|^2} = \sqrt{2 - 2\cos(\alpha)}$$

2. 1) Dog → dog[0.0] cat[0.39547] dogs[0.54531] horse[0.6469] puppy[0.67009] pet[0.67458] rabbit[0.67516] pig[0.70851] snake[0.72122] baby[0.72172] bite[0.72278] boy[0.72349] cats[0.73488] animal[0.74132] monkey[0.742] rat[0.74218] mad[0.74238] crazy[0.75392] man[0.75869] elephant[0.75926] monster[0.76082] pack[0.76098] eating[0.76462] kid[0.77209] wolf[0.78182] ghost[0.78317].
   2) Web → web[0.0] internet[0.38198] online[0.38533] users[0.58913] websites[0.59214] google[0.60409] website[0.61329] facebook[0.62005] blog[0.62636] software[0.62719] user[0.6354] media[0.65163] networking[0.65479] network[0.66231] blogs[0.6626]

information[0.66357] interactive[0.6637] computer[0.66432] database[0.66728] sites[0.6682] messaging[0.67512] video[0.68306] page[0.68387] addresses[0.6889] search[0.69055] networks[0.70584].

3) Car $\rightarrow$ car[0.0] truck[0.39785] cars[0.47535] vehicle[0.48297] driver[0.55425] driving[0.56847] bus[0.59825] vehicles[0.60415] parked[0.64774] motorcycle[0.65322] taxi[0.65819] passenger[0.66509] pickup[0.66771] trucks[0.67368] cab[0.67606] suv[0.68054] train[0.68436] drivers[0.69728] bicycle[0.69853] jeep[0.71891] airplane[0.7190 wheel[0.72434] tractor[0.72443] driven[0.72759] mercedes[0.73894] bike[0.7407].
4) Work $\rightarrow$ work[0.0] working[0.40914] done[0.46678] well[0.49388] works[0.50206] own[0.51177] worked[0.54065] besides[0.55578] making[0.58018] doing[0.58077] and[0.58717] as[0.59768] for[0.59814] way[0.61589] addition[0.61594] writing[0.61884] instead[0.619] ways[0.6199] how[0.62176] idea[0.62345] focus[0.62601] life[0.6289] .[0.62948] this[0.6316] important[0.63485] full[0.63485].
5) Money $\rightarrow$ money[0.0] cash[0.44947] paying[0.49221] funds[0.4962] pay[0.50665] raise[0.55777] paid[0.56105] billions[0.59793] millions[0.60022] get[0.61172] fund[0.61918] keep[0.62246] tax[0.6324] savings[0.63534] credit[0.6409] make[0.64344] putting[0.644] taxes[0.64624] spend[0.64824] making[0.65026] giving[0.65113] proceeds[0.65276] cost[0.65432] expense[0.65707] raising[0.65884] taxpayers[0.65923].
6) Power $\rightarrow$ power[0.0] control[0.56904] powerful[0.68614] system[0.68734] turn[0.70106] pressure[0.70406] support[0.70679] bring[0.70692] its[0.71158] current[0.71864] to[0.71867] electricity[0.71945] creating[0.72502] energy[0.7261] controlling[0.73263] powers[0.73356] controlled[0.73368] build[0.73418] create[0.73464] controls[0.7369] effectively[0.73778] which[0.7398] as[0.7413] instead[0.74221] bringing[0.74574] own[0.74805].
7) War $\rightarrow$ war[0.0] occupation[0.54216] invasion[0.54997] wars[0.59207] conflict[0.60204] fighting[0.60611] military[0.60901] iraq[0.65123] forces[0.65501] wartime[0.65675] army[0.66075] battle[0.6713] civil[0.68253] during[0.68645] fought[0.68922] 1991[0.69052] soviet[0.69287] troops[0.69314] decades[0.69643] battles[0.70898] brought[0.71308] struggle[0.71314] continued[0.72624] force[0.72989] decade[0.73162] afghanistan[0.73407].
8) City $\rightarrow$ city[0.0] town[0.51227] downtown[0.54145] where[0.54313] cities[0.54683] area[0.57928] in[0.59527] outside[0.59596] near[0.60926] central[0.61106] nearby[0.64067] home[0.64277] capital[0.6451] neighborhood[0.65732] southern[0.6574] east[0.66709] southwest[0.66987] suburbs[0.67819] suburb[0.68071] metropolitan[0.68323] residents[0.6862] towns[0.68808] eastern[0.68933] west[0.68989] located[0.69009] opened[0.69153].
9) Student $\rightarrow$ student[0.0] teacher[0.45562] students[0.48505] teachers[0.53559] graduate[0.583] school[0.59342] teaching[0.61987] faculty[0.62998] education[0.63017] youth[0.66505] academic[0.66661] college[0.67763] undergraduate[0.69186] graduates[0.7071] university[0.70876] classes[0.70921] professors[0.71522] schools[0.71895] young[0.72787] working[0.73227] enrolled[0.73547] taught[0.73722] attending[0.74577] graduating[0.74935] harvard[0.75065] learning[0.75426].

10) Cool $\rightarrow$ cool[0.0] hot[0.5282] warm[0.61523] cold[0.61587] bit[0.64275] dry[0.68327] cooler[0.69001] little[0.69308] mix[0.70241] soft[0.71362] bright[0.71786] pretty[0.72061] chill[0.72808] looks[0.73225] wet[0.74029] sunny[0.74979] look[0.75038] too[0.75077] touch[0.75435] thin[0.75893] dark[0.76325] hard[0.77023] tends[0.77686] keeps[0.77689] smooth[0.77752] heat[0.77877].

3. 25 closest words.
Here are the words and their $\cos\alpha$:

(piyanart, srivalo | 0.999895918380112)
(artthielseattle, lauravecseyseattle | 0.9982480316676561)
(ba632, ba633 | 0.9981987474148983)
(tuesday, monday | 0.9981015491425101)
(tuesday, thursday | 0.997851731352666)
(tuesday, wednesday | 0.9977707917837518)
(monday, thursday | 0.9976342564249955)
(formula_4, formula_5 | 0.9976102258531189)
(formula_5, formula_6 | 0.9972320296874572)
(wednesday, thursday | 0.9971067630156977)
(gmathis, sksmith | 0.9968831788254773)
(june, july | 0.9965305312552686)
(formula_12, formula_13 | 0.9964950669677369)
(september, october | 0.9964256462412024)
(formula_7, formula_8 | 0.9963535860366887)
(wednesday, monday | 0.9963471209653657)
(october, february | 0.9963103829935064)
(22, 21 | 0.9962301545030089)
(formula_8, formula_9 | 0.996188001547275)
(26, 28 | 0.9961357287667479)
(sgushee, dgeorge | 0.9960865740672376)
(28, 27 | 0.9960239385995423)
(july, april | 0.9958918171212049)
(june, april | 0.995867670260325)
(14, 13 | 0.9957585582533608)

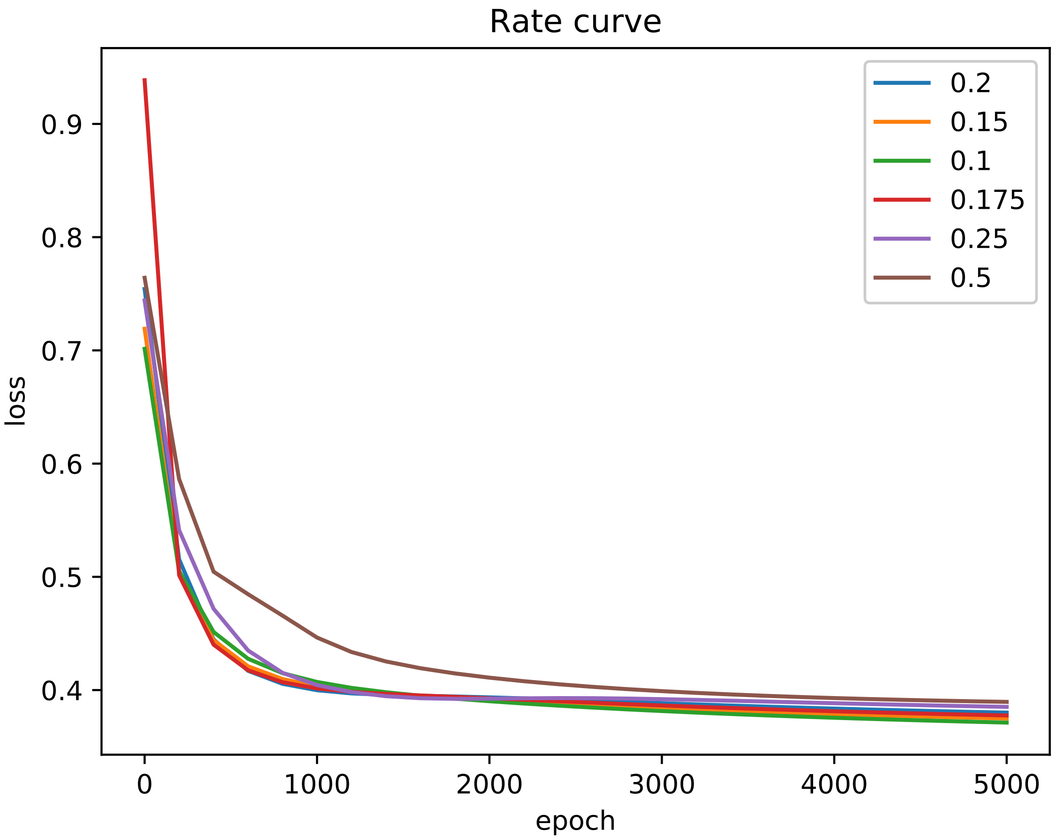
D. In training set there are **9477 words** that don't have embeddings. Here are 20 examples:
enging, moviestore, 20mn, grandeurs, anachronic, decaune, kitchy, kojac, lashelle, payaso, plinplin, blainsworth, blains, crackd, 1984ish, lonnrot, yidische, zaitung, discplines, buchfellner
In test set I found **5998 words** without embeddings.

E. Right after the initialization mean value of $\hat{y}(x)$ is **[0.5, 0.5]**. Mean value of loss function on training set is **0.9**.

G. Mean error of numerically calculated gradient comparing to backpropagation gradient is $10^{-9}$ (with $\varepsilon = 10^{-3}$).

J. With `learning_rate` = **0.01** it took **2000** epochs to converge.
Accuracy for train set reached **83.87**. On test set — **82.61**.
The network is underfitted. It means that we should lower the regularization parameter $\alpha$.

K. Here is the plot for different learning rates




The best learning rate here is **0.1**. It took more than **5000** epochs to converge.

L. Best achieved accuracy is **84.933** on train set and **83.830** on test set. Training takes near **10 minutes**. Classifying takes near **10 seconds**.

P.S. I also made **Adagrad** optimization.