Practical No. 2

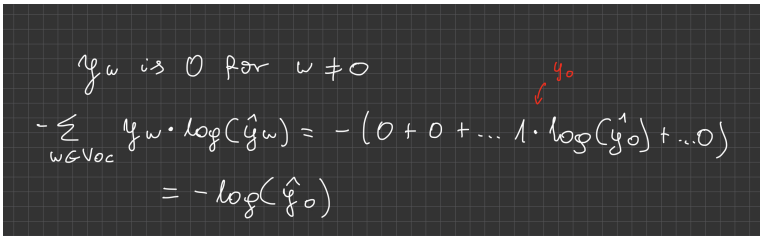
Karol Pustelnik
kp446518@students.mimuw.edu.pl

March 20, 2022

1 The reasoning behind word2vec

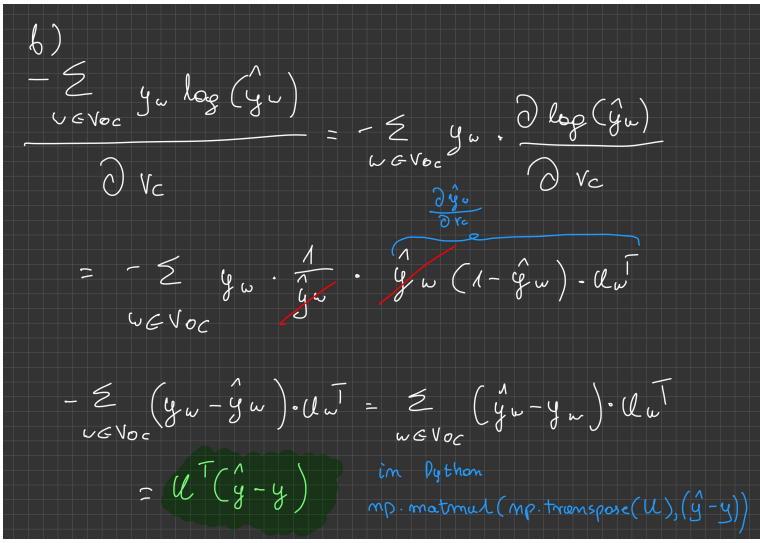
a)

Show that the naive-softmax loss given in Equation (2) is equivalent to the cross-entropy loss between y and $\hat{y}$.



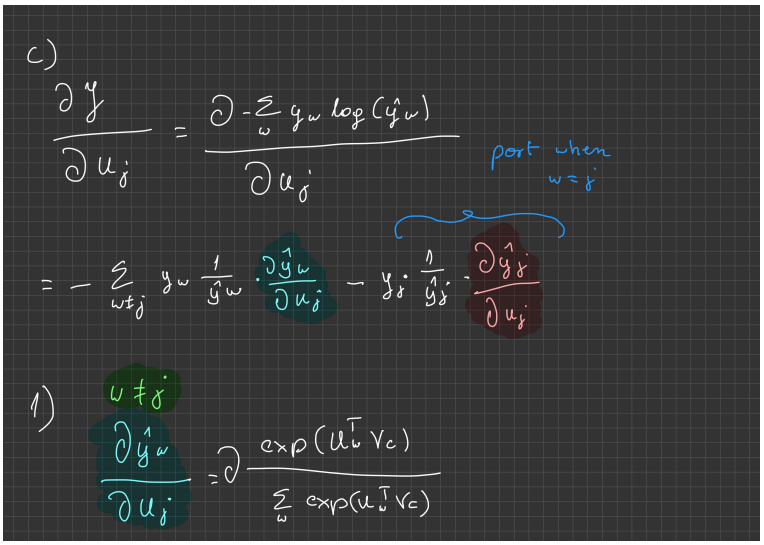
b)

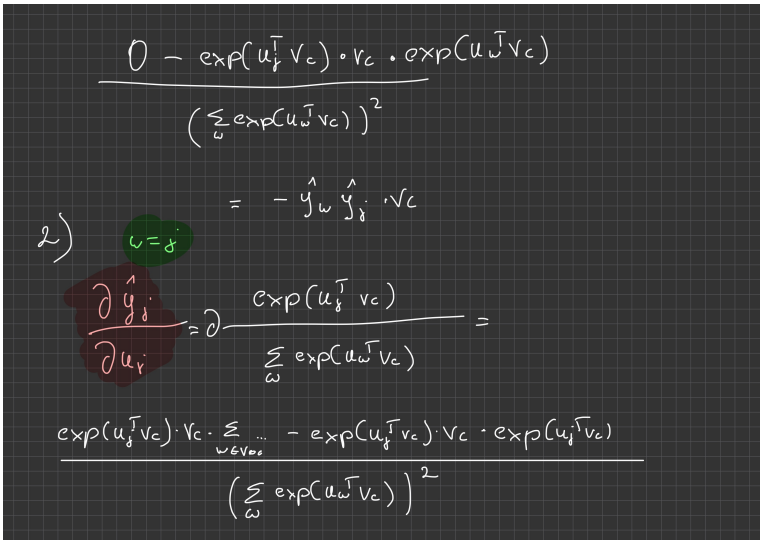
Compute the partial derivative of $J_{naive-softmax}(v_c, o, U)$ with respect to v_c. Write your answer in terms of of y, $\hat{y}$, and U.

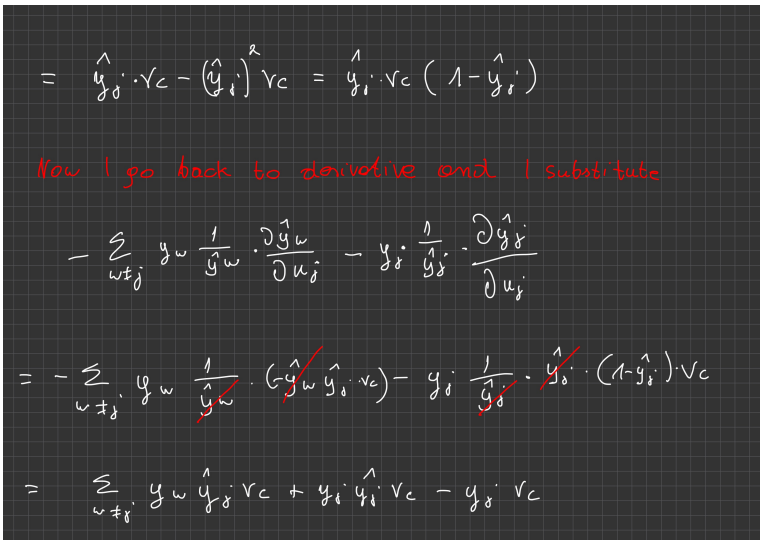


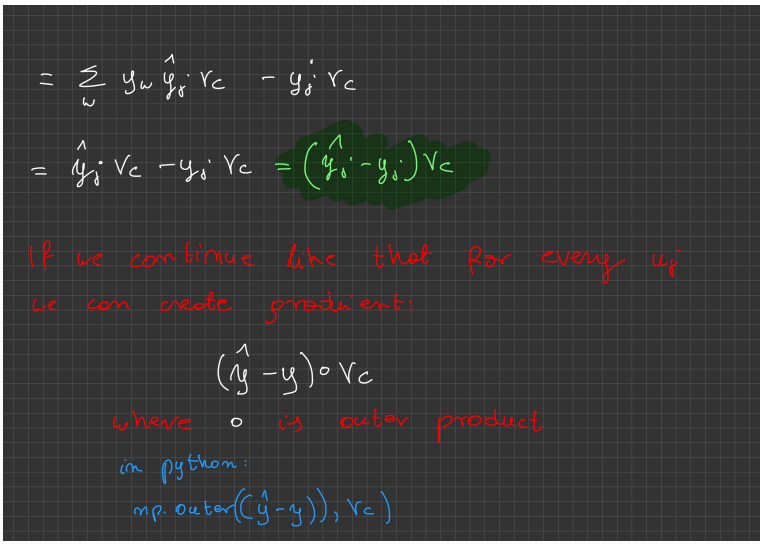
c)

Compute the partial derivatives of $J_{naive-softmax}(v_c, o, U)$ with respect to each of the outside word vectors, u_{w_s}. Write your answer in terms of y, $\hat{y}$, and v_c.



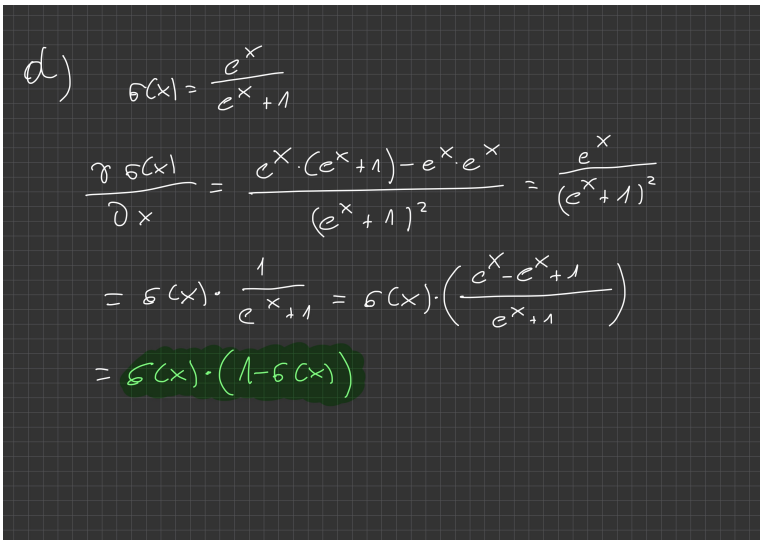






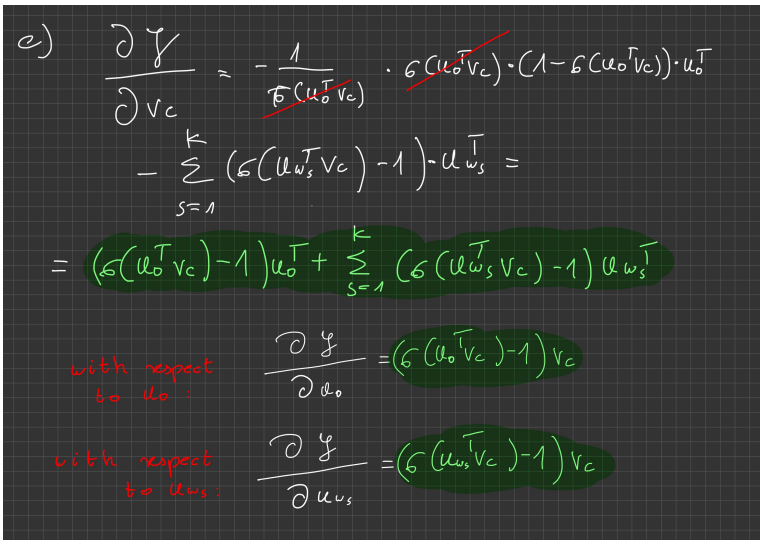
d)

Please compute the derivative of $\sigma(x)$ with respect to x.



e)

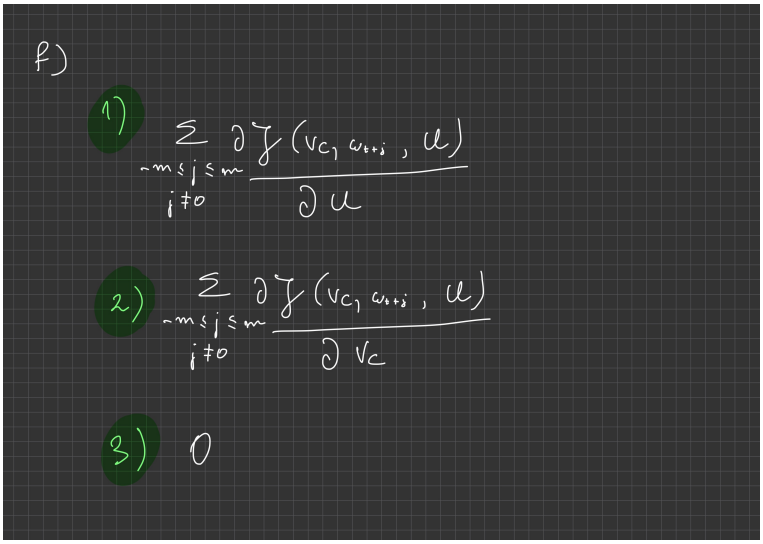
Please repeat parts (b) and (c), computing the partial derivatives of $J_{negsample}$ with respect to v_c, with respect to u_o, and with respect to a negative sample u_k. Please write your answers in terms of the vectors u_o, v_c, and u_k, where $k \in [1, K]$.



1. Why this loss function is much more efficient to compute than the naive-softmax loss?

   - Because the computations are based on much smaller sample.

f)



2 Building your version of word2vec

In the picture, I can see that there is almost no clustering of words. Everything is scattered. We also have words "swietny" and "zly" close to each other which is weird. I think that the selected seed was very unfortunate and led to weird results.

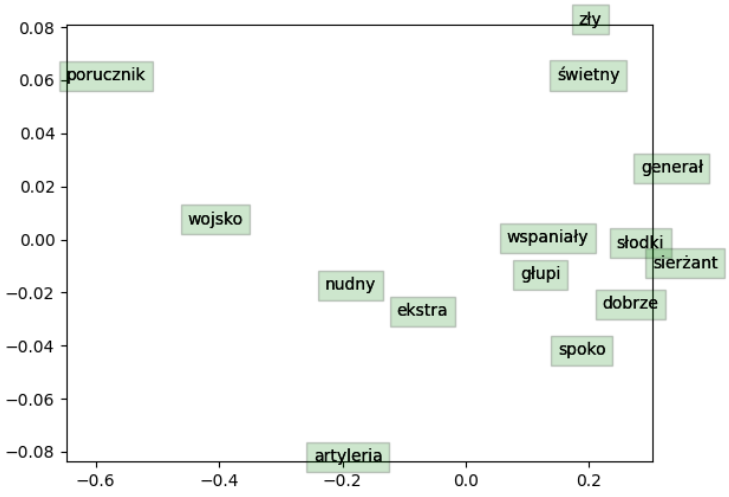


Figure 1: Visualization of 2D word embeddings

3 Optimizing Neural Networks

a)

1. Briefly explain (you can just give an intuition) how using m stops the updates from varying as much and why this low variance may be helpful to learning, overall.

    - Adding momentum to parameter updates is like increasing the speed of the ball that rolls down the hill. A faster ball is better because:
        * it oscillates less (because it has high momentum)
        * it reaches local (global) minima faster (because more momentum means more speed)
        * it is less likely to be stuck at bad local minima because it has enough speed to leave it

b)

1. Since Adam divides the update $\sqrt{v}$, which of the model parameters will get larger updates? Why might this help with by learning?

    - With $\sqrt{v}$, Adam normalizes the parameter updates, reducing high gradients parameters learning rate and increasing low gradients low gradients parameters learning rate. The normalization helps parameters learn at a similar pace and helps with faster convergence.

c) *

1. Why L2 regularization can be helpful in low-data setting?

    - Because if our dataset is small, we have a high chance of overfitting. L2 regularization ads noise so its harder for the model to learn the data.

2. What is their proposal for improving the Adam update scheme?

    - Authors suggest "decoupling the weight decay from the optimization steps taken w.r.t the loss function". It means that AdamW adjusts the weight decay term to appear in the gradient update.

d)

1. Why should we apply dropout during training but not during evaluation?

    - Dropout makes neurons output wrong values on purpose. Because neurons are disabled randomly the network will have different outputs every (sequences of) activation.

2. How dropout can be seen from the perspective of Bayesian theory?*

- Based on the paper, "a neural network with arbitrary depth and non-linearities, with dropout applied before every weight layer, is mathematically equivalent to an approximation to the probabilistic deep Gaussian process. This theory gives tools to model uncertainty with dropout NNs – extracting information from existing models that has been thrown away so far. This mitigates the problem of representing uncertainty in deep learning without sacrificing either computational complexity or test accuracy."