Deep Learning for Speech and Language Processing
Exercise Sheet 3: Feed-Forward Neural Networks

Maximilian Schmidt, Pascal Tilli, Dirk Väth, Ngoc Thang Vu

November 16th, 2020

Deadline

Please submit your solutions until Monday, November 23rd, 2020, noon. You may also see the deadline at any time on the website under *Exercises* > *Upload*.

Submission

Calculation Tasks

Upload (only solutions!) to https://dlcourse.ims.uni-stuttgart.de/ using your account.

Theory Tasks

You don't have to submit this part, but it will help you with the contents of this course.

1 Warm-Up

Calculation Task Kevin and his friends are really fond of bananas. Therefore, they want to build a social media monitoring tool that constantly analyzes tweets and notifies them if a tweet is about bananas. Here are some example tweets:

1) A banana a day keeps the doctor away.
2) I would rather have a sweet chocolate cookie now.
3) Banana! Banana! Banana!

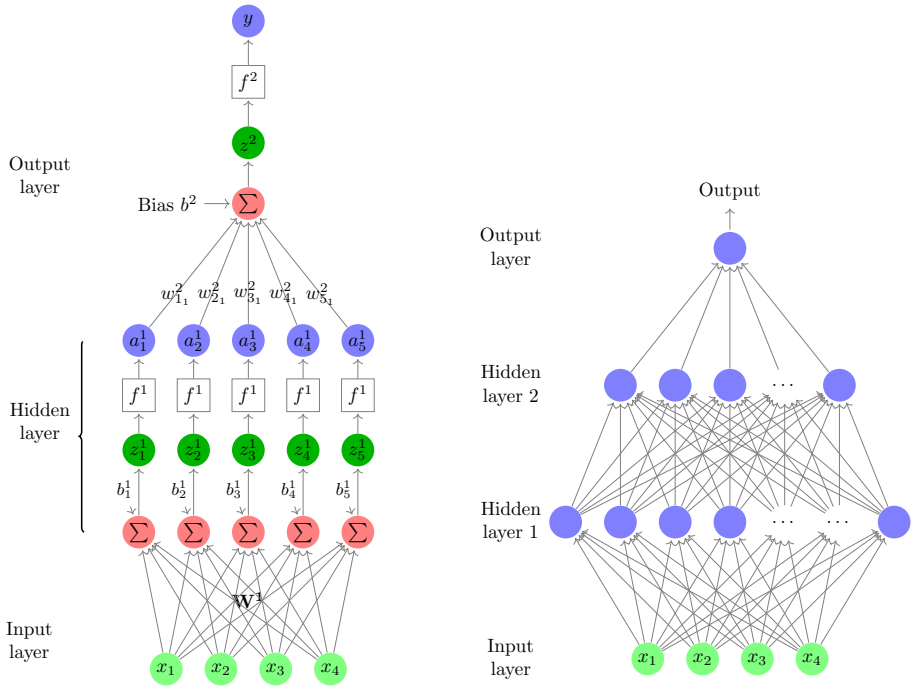
In order to provide the tweets as input to a statistical classifier, they need to map each tweet to a fixed-length vector. A very common approach is to represent each word w_i in the input as a unit vector of the size of the vocabulary, where $e_i = 1 \Leftrightarrow w_i$ is the i-th word in the vocabulary. This is called a **one-hot vector** representation. To obtain a representation for the complete input, i.e., the sequence of words in a tweet, we can simply sum the one-hot vectors for all words. This is often referred to as a **bag-of-words** representation.

Exercise 1.
Given the 4-word vocabulary
$V = \{v_0 : banana, v_1 : chocolate, v_2 : sweet, v_3 : cookie\}$,
compute the bag-of-words vector representations for the sample tweets (lowercase all words in the tweets, ignore the punctuation marks and use 0_4 as out-of-vocabulary vector):

(1) **Calculation Task** Tweet 1
(2) **Calculation Task** Tweet 2
(3) **Calculation Task** Tweet 3

2 Feed-forward Neural Network

Kevin has heard about deep learning and came up with the banana network depicted in the right figure below. Note that in the banana network, the hidden layers are displayed conflated. Each hidden layer l consists of first linear activation of the input a^{l-1} with $z^l = W^l a^{l-1} + b^l$ and then non-linear activation, such that $a^l = f(z^l)$ as shown in the left figure below.



(a) One-layer feed-forward network for binary classification.

(b) Banana network.

Exercise 2.

(1) **Calculation Task** How many parameters does the banana network have in total, if it has 200 units in the first hidden layer and 25 units in the second hidden layer?

(2) **Theory Task** Derive the function to compute the output y of this network, given an input vector x.

(3) **Calculation Task** Bob does not believe that the (untrained) network will really detect tweets about bananas. Therefore, you try to check the outputs of the network for the first tweet from Question 1. In order to make the computation by hand feasible, use a smaller network with the following parameters:

$$W^1 \in \mathbb{R}^{3\times 4} = \begin{bmatrix} -4 & -3 & 3 & 1 \\ -5 & 5 & -1 & -5 \\ 5 & -5 & 4 & 1 \end{bmatrix}, b^1 \in \mathbb{R}^3 = 0^3,$$

$$W^2 \in \mathbb{R}^{2\times 3} = \begin{bmatrix} -1 & -1 & 2 \\ 5 & -3 & -5 \end{bmatrix}, b^2 \in \mathbb{R}^2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, W^3 = \begin{bmatrix} 3 & -1 \end{bmatrix}, b^3 = \begin{bmatrix} 0 \end{bmatrix}$$

For the first two layers, we do not use an activation function, i.e. $f^1(z) = f^2(z) = z$. The last layer has non-linear activation with the logistic sigmoid: $f^3(z) = \sigma(z) = \frac{1}{1+e^{-z}}$. Give the resulting network output.

3 Multi-class Classification

Bob also wants to detect tweets about cookies, so he asks Kevin to build another network for this class. However, it is more efficient to extend the binary banana-classifier to a multi-class classifier, so they need to learn only one set of parameters to detect tweets about cookies and bananas at the same time.

Exercise 3.

(1) **Theory Task** Extend the network from Question 2 to the multi-class case with 3 outputs; y = $\begin{bmatrix} \text{p(banana)} \\ \text{p(cookie)} \\ \text{p(other)} \end{bmatrix}$.

Hint: You only have to change the output layer.

(2) **Calculation Task** Compute the outputs from the extended network for the first tweet of Question 1 using the bag-of-words representation with vocabulary V, the same weights for W^1 and W^2 and bias values b^1, b^2 as in Question 2 c) and $W^3 = \begin{bmatrix} -2 & 3 \\ -3 & 2 \\ -3 & 2 \end{bmatrix}$, $b^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$.

(3) **Calculation Task** Do the same for the second tweet of Question 1.

4 Error Computation

Bob and Kevin would like a numerical measure to assess how well their classifier does. They can do this by the means of a cost or error function C that compares the network's predicted probability distribution over N classes $\hat{y} \in \mathbb{R}^N$ with a ground truth $y \in \mathbb{R}^N$ and yields a scalar c, the cost. There are two commonly used choices for error functions in deep learning:

Mean Squared Error (MSE) $c_{\text{MSE}} = \sum_{i=0}^{N-1} (\hat{y}_i - y_i)^2$

Note: We omit the scaling factor here since it's not important for optimization.

Cross Entropy (CE) $c_{\text{CE}} = -\sum_{i=0}^{N-1} y_i \ln \hat{y}_i$

Note that for classification, primarily cross entropy is used.

Exercise 4.

(1) **Theory Task** In classification problems y is usually a unit vector with the entry at the position of the correct class set to 1. Simplify the cross entropy error function under this condition.

(2) **Calculation Task** Compute the loss for the output of the multiclass classification network assuming $\begin{bmatrix} 0.97 \\ 0.02 \\ 0.00001 \end{bmatrix}$ as output of the network and 'banana' as correct class

(1) using the mean squared error.

(2) using the cross entropy.