Introduction to Deep Learning for Speech and Text Processing
Exercise Sheet 6: Neural Networks

Thang Vu

21st November 2025

1 Warm-Up

We might want to build a social media monitoring tool that constantly analyzes data from Instagram and notifies them if a text is about bananas. Here are some example texts:

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 texts as input to a statistical classifier, they need to map each 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
$V = \{v_0 : \text{chocolate}, v_1 : \text{sweet}, v_2 : \text{banana}, v_3 : \text{cookie}\}$,
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) Text 1
(2) Text 2
(3) Text 3

2 Feed-forward Neural Network

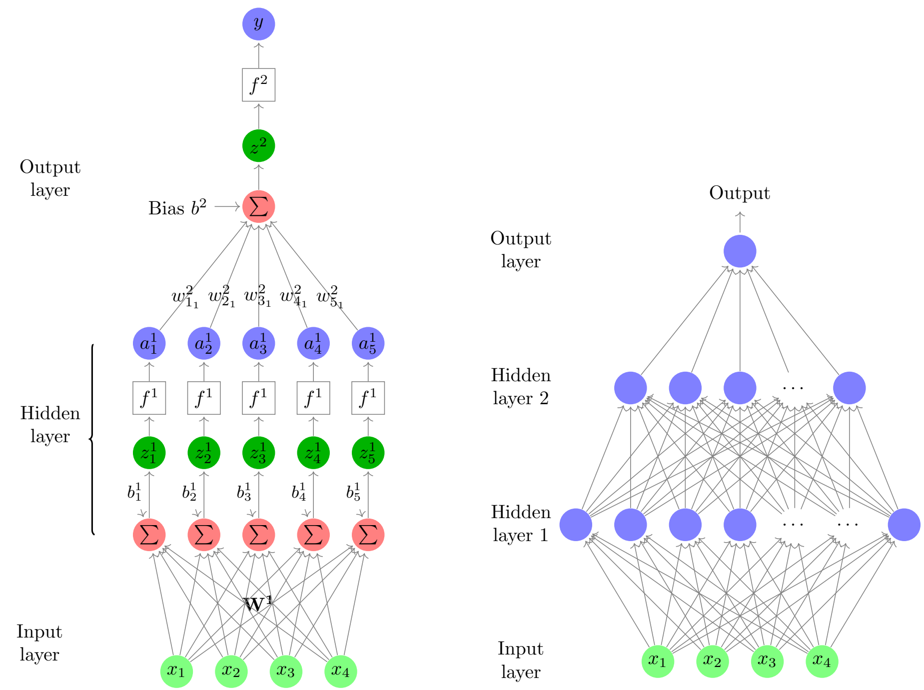
You might come up with the banana network depicted in figure 1b. Note that in the banana network, the hidden layers are displayed conflated. Each hidden layer l consists of first linear activation of the input a^{l-1} with $z^l = W^l a^{l-1} + b^l$ and then non-linear activation, such that $a^l = f^l(z^l)$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

Exercise 2.

(1) How many parameters does the banana network have in total, if it has 300 units in the first hidden layer and 200 units in the second hidden layer?

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

(3) Compute the output of the network for the first text from Exercise 1. In order to make the computation by hand feasible, use a smaller network with the following parameters:



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

(b) Banana network.

Figure 1: Visualization of a single layer neural network (a) and our banana network (b) consisting of multiple layers.

$$W^1 \in \mathbb{R}^{3\times 4} = \begin{bmatrix} 0 & 1 & -2 & 1 \\ 2 & 3 & 0 & -1 \\ 1 & 0 & -3 & 0 \end{bmatrix}, b^1 \in \mathbb{R}^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$
$$W^2 \in \mathbb{R}^{2\times 3} = \begin{bmatrix} 0 & -2 & 1 \\ 2 & 0 & -1 \end{bmatrix}, b^2 \in \mathbb{R}^2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, W^3 \in \mathbb{R}^{1\times 2} = \begin{bmatrix} -1 & 1 \end{bmatrix}, b^3 \in \mathbb{R} = 1$$

For the first two layers, we use the Rectified Linear Unit (ReLU) as activation function, i.e., $f^1(z) = f^2(z) = ReLU(z) = max(0, z)$. The last layer has non-linear activation with the sigmoid, i.e., $f^3(z) = \sigma(z) = \frac{1}{1+e^{-z}}$. Give the resulting network output.

3 Multi-class Classification

Now we want to extend the binary banana-classifier to a multi-class classifier that can detect texts about cookies and bananas at the same time.

Exercise 3.

(1) Extend the network from Exercise 2 to the multi-class case with a single correct class and 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) Compute the outputs from the extended network for the first text of Exercise 1 using the bag-of-words representation with vocabulary V, the same weights for W^1, W^2 and bias values b^1, b^2 as in Exercise 2.3 and

$$W^3 = \begin{bmatrix} 1 & 1 \\ -1 & 2 \\ 0 & -1 \end{bmatrix},\ b^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

4 Error Computation

Now we want a numerical measure to assess how well their classifier does. We 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. Here, we use cross-entropy as a common choice for classification:

Cross-Entropy (CE) $c_{\text{CE}} = -\sum_{i=0}^{N-1} y_i \ln \hat{y}_i = -\ln \hat{y}_t$, where t is the correct class

Exercise 4.

(1) Compute the loss for the output of the multiclass classification network for the first text from exercise 3.2 assuming 'banana' as correct class.

(2) For the backward pass, we successively compute the derivatives starting from the last layer. Compute δ^L for the previously computed loss.

5 Stochastic Gradient Descent

We will now perform the backward pass:

Exercise 5.

(1) For the parameters W^3

   (1) compute the gradient of W^3 with the result being the matrix $\begin{bmatrix} \frac{\partial c}{\partial w_{11}^3} & \frac{\partial c}{\partial w_{12}^3} \\ \frac{\partial c}{\partial w_{21}^3} & \frac{\partial c}{\partial w_{22}^3} \\ \frac{\partial c}{\partial w_{31}^3} & \frac{\partial c}{\partial w_{32}^3} \end{bmatrix}$

   (2) perform a gradient update step using these derivatives with $\eta = 0.1$.

(2) For the parameters W^2

   (1) compute the gradient of W^2 analog to the previous task.

   (2) perform a gradient update step using these derivatives with $\eta = 0.1$.

(3) For the parameters W^1

   (1) compute the gradient of W^1 analog to the previous task.

   (2) perform a gradient update step using these derivatives with $\eta = 0.1$.