Exercises

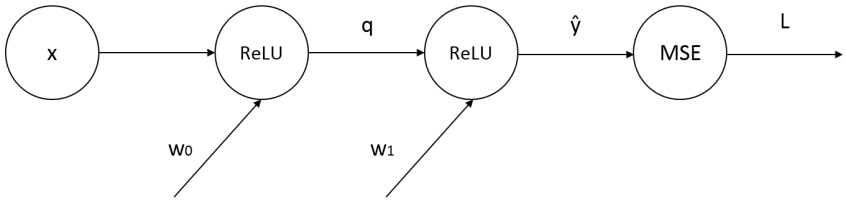

Figure 1: Graph

Exercise-1 We can use two approaches to compute new values for the weights w_0 and w_1. The first one is to update the weights after each observation. The second is to take the average of the computed weights and thus, only update once.

Observations: $(x_1, y_1) = (1, 2)$ and $(x_2, y_2) = (2, 3)$ Weights: $w_0 = 1$ and $w_1 = 2$

We use the MSE as loss function: $L = \frac{1}{2}(\hat{y} - y)^2$

The first approach:

$$\hat{y}_1 = w_1(w_0 x_1) \tag{1}$$
$$= 2 * 1 * 1 = 2 \tag{2}$$
$$L = \frac{1}{2}(2 - 2)^2 \tag{3}$$
$$= 0 \tag{4}$$
$$\tag{5}$$

As $L = 0$ the derivatives will all be zero and, therefore, the weights will not be updated. Thus, we continue to the next observation.

$$
\begin{aligned}
\hat{y}_2 &= w_1(w_0 x_2) \\
&= 2 * 1 * 2 = 4 \\
L &= \frac{1}{2}(4-3)^2 = \frac{1}{2} \\
\frac{\partial L}{\partial \hat{y_2}} &= [\frac{1}{2}(\hat{y_2} - y_2)^2]' \\
&= 2 * \frac{1}{2}(\hat{y_2} - y_2) * 1 \\
&= \hat{y_2} - y_2 \\
&= 1 \\
\frac{\partial L}{\partial w_1} &= \frac{\partial \hat{y_2}}{\partial w_1}\frac{\partial L}{\partial \hat{y_2}} \\
&= w_0 x_2 * \frac{\partial L}{\partial \hat{y_2}} \\
&= 2 * \frac{1}{2} \\
&= 1 \\
\frac{\partial L}{\partial q} &= \frac{\partial \hat{y_2}}{\partial q}\frac{\partial L}{\partial \hat{y_2}} \\
&= w_1 * 1 \\
&= 2 \\
\frac{\partial L}{\partial w_0} &= \frac{\partial q}{\partial w_0}\frac{\partial L}{\partial q} \\
&= x_2 * \frac{\partial L}{\partial q} \\
&= 2 * 2 = 4
\end{aligned}
$$

Thus, $w_1^* = w_1 - 0.1 * 2 = 1.8$ and $w_0^* = w_0 - 0.1 * 4 = 0.6$.

Exercise-2 Various decisions need to be made in a modeling process to address specific properties of the data and the modeling goal. In this task, you are given a description of a data structure and a goal for which you need to design a model.

Produce a figure depicting your model. Briefly explain the figure and justify all decisions made in the modeling process. In detail, describe at least: - Input data format - Number of layers - Type of layers (Dense, Recurrent, Convolutional - 1D, 2D, 3D) - Regularization - Model output —– caption? - Loss function —- 0/1 loss

The training and execution procedures for the model may differ, so you can use different descriptions for both.

Data and goal description:

The goal of this task is to generate captions for short video clips.

The video data is structured as sequences of color images. The model needs to be able to process a number of consecutive images that form a short video clip. The training data consists of video clips (few seconds) and a short caption (5-10 words).

For simplicity, the accuracy of the model is evaluated on the exact prediction of the caption. In other words, the model needs to produce correctly the specific words in a specific order for each video.

- Input data format: The data format will be a tensor of images. Each image is 3D, thus each sequence of images for each video will be 4D tensor.
- Number of layers: The model consists of two layers. A CNN and a RNN layer.
- Type of layers:
  1. CNN layer. The setup of the CNN is as we have seen in the lecture, type D. However, we end with with a singel vector after the two FC 4096. Hence, we do not have the FC 1000 and softmax at the end. Thus, for each image we run a CNN, each image becomes a FC 4096. Then we run a mean pooling over it to get one vector representing the total video. We choose to to a mean pooling afterwards, because we want the mean image of the video. For example, the video shows a boy playing football. However, the video shows the ball and the boy most of the time but there is also a lot of noise on the background clearly visible in some images. We then want to take the average such that every part of the video is captured by the FC 4096. Thus, the vector is a summary of the entire video. This single vector is then the input for the next layer.
     This can be seen in figure BLABLABLA
  2. RNN layer. This layer consists of the a recurrent neural network using the LSTM setup for each cell. The main idea here is to generate a word at each time stamp. This word comes from a dictionary on which the model is already pre-trained. The input at each cell is the output from the previous cell and the summary vector of the images. The output is a word that is choses with the highest probability: $\max \Pr[\text{word}_i|\text{previous words chosen}]$.
     This can be seen in figure BLABLABALA
- Regularization: ??????????????????
- Model output: The model outputs a sequence of words. This sequence is smaller than or equal to 10.
- Loss function: The loss function is the 0/1-loss function as we want the exact caption. Thus, if the caption that is ouputted is only slighly off, the loss will still equal one as it is not completely correct.