Homework 4.

Due on Gradescope by 5 PM, Saturday, May 01

1. Given the input x_i and its label y, we use the following multi layer network and the loss function as a regression problem.

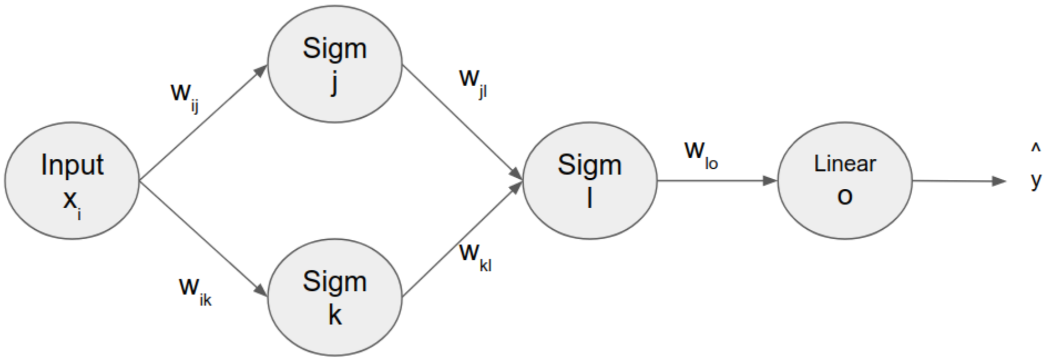

Figure 1: Multi layer network

$$L = \frac{1}{2}(\hat{y} - y)^2 \tag{1}$$

For the node labeled l, assume the output activation is applied to a linear combination of the inputs, i.e.,

$$z_l = w_{jl}z_j + w_{kl}z_k$$

where z_n is the output of the neuron n, $n = \{i, j, k, l\}$. The bias term is assumed to be zero in this case.

Derive the expressions

- $\frac{\partial L}{\partial w_{ij}}$,
- $\frac{\partial L}{\partial w_{ik}}$,
- $\frac{\partial L}{\partial w_{jl}}$,
- $\frac{\partial L}{\partial w_{kl}}$,
- $\frac{\partial L}{\partial w_{lo}}$

Sigm functions are sigmoid functions: $\sigma(x) = \frac{1}{1+e^{-x}}$.

Linear defines the linear function: $linear(x) = x$

2. **K-Nearest Neighbors (programming problem):** The purpose of this problem is to introduce you to the concepts of distance metrics (in this case the sum of square errors), sorting based on these distances – i.e., nearest neighbors to a query, training and testing data sets. In this particular case, the test data item is not in the ”training data”, though there is really no ”training” happening. This will become clear to you once you start working on the problem. However, we would like to keep this distinction between the training and test data since we will be re-using this data in future homework as well.

   For this homework, we will be using K-Nearest Neighbor to classify image from the CIFAR-10 dataset. Basic idea behind K-Nearest Neighbor algorithm is to use a distance measure in order to find the K-nearest neighbors of a test sample. The test sample is then assigned the label based on the labels of these neighbors.
   Dataset: We are going to use CIFAR-10 dataset in order to implement K-Nearest neighbor algorithm. Dataset can be found at https://www.cs.toronto.edu/~kriz/cifar.html Training and testing splits are also mentioned on the webpage. You may use only data_batch_1 for your training set, and the first 1000 samples of test_batch for the test set.
   Distance Measure: An example distance between 2 images can be the sum of squared difference between pixel intensities.
   Note: If you try to find difference between two 32x32 RGB (Red, Green, Blue) images, you will need to compute 32x32 differences for each of the three, R G B channels.
   Label assignment strategy: Once the K-Nearest Neighbors to a test sample are found, assign it the to the label which majority of its neighbors have. In cases of a tie, Consider the distance information in order to break the tie.

   (a) Apply K-Nearest Neighbor algorithm with k = 1 on the test samples. We define the classification error rate as

   $$P_e = \frac{\text{Number of Wrongly Classified Test Samples}}{\text{Total Number of Test Samples}} \tag{2}$$

   What is the error rate of your classification?

   (b) Repeat last step for k = 2, 5, 10, 20 and plot the error rate P_e against k. Is the error rate decreases with k? Should the error rate always decrease with k?

   (c) For each of the ten classes, pick a random image from test data and report its 10 nearest neighbors.

   Submission format. Please submit the problem number 2 through **a google colab notebook** as you have done in the last two homeworks. A skeleton colab notebook is *not* provided for this homework. Students are recommended to organize their code into modular cells for ease of readability and follow the same steps for submitting the colab notebook as done in previous homeworks. You will see a separate assignment on gradescope for problem 2 of this homework – be sure to submit each problem to the right assignment on gradescope.