Homework 6

1. *Fitting neural networks with Julia.* In `nn_regression.json`, you will find a 4000×30 matrix `X_train` and a 4000-vector `y_train` consisting of training input and output data, and a 1000×30 matrix `X_test` and a 1000-vector `y_test` consisting of test input and output data, respectively.

   In `nn_regression.jl` we have also provided you with a function

   `nn_regression(X, Y, lambda).`

   This function takes in input/output data `X` and `Y`, and a local regularization hyper-parameter `lambda`. It outputs a neural network with parameters and activation functions defined in the code.

   In the file `neural_net.jl` in the starter code, the data from `nn_regression.json` has been loaded and standardized. No constant feature has been added. Do not add one.

   (a) In `neural_net.jl`, use ridge regression to fit a linear predictor to the training data. Plot the training and test RMS errors for regularization parameters using the values $10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}$ for λ.

   (b) Inspect the `nn_regression` function in `nn_regression.jl`. What is the form of the neural network model in the code? Specify the number of layers; for each layer, give the activation function and dimension of model parameters. In total, how many scalar model parameters are there in this neural network? (For example, $A \in \mathbf{R}^{13\times 10}$ has $13 \times 10 = 130$ scalar entries.)

   (c) Using `nn_regression.jl`, fit a neural network to the training data for the same regularization parameters from part (a). Plot the training and testing RMS error versus the regularization parameter.

   Remark. It is normal for training this neural network to take several minutes.

   Hint: The parameters of a neural net consist of a weights matrix and a bias vector for each layer of the neural net.

   Hint: Call `predictall(model,U)` to compute a vector of $\hat{y}$.

2. *Embedding words.* We have embedded the words “cat”, “dog” “house”, “school”, and “shovel” as vectors using two different word-to-vector embeddings. In `embedding.json`, you will find two matrices, `X1` and `X2`, corresponding to the two embeddings. The rows of `X1` and `X2` are the *representatives* of the words “cat”, “dog” “house”, “school”, and “shovel”, in this order.

   (a) Compute and report the 5×5 matrix of pairwise distances between the representatives for the first embedding (`X1`). That is, compute the 5×5 matrix where the entry in the ith row and jth column is the Euclidean distance between the ith and jth row of `X1`.

(b) Compute and report the 5×5 matrix of pairwise distances between the representatives for the second embedding (`X2`).

(c) Which embedding do you think is more faithful? Give a very short explanation for your choice. (Of course in a practical setting, you would not choose an embedding based on the pairwise distances between five items.)

Hint. As a sanity check, the diagonal entries of the pairwise distance matrices should be zero.

3. *One hot embedding pairwise distances.* You have categorical data that takes on K distinct values, $v_1, \ldots, v_K$.

   (a) Suppose you one-hot encode your categorical data. What is the Euclidean distance between any two distinct representatives?

   (b) Suppose you use a reduced one-hot encoding to your categorical data, with the first category v_1 the default. What is the Euclidean distance between the default representative and a non-default representative? What is the Euclidean distance between two distinct non-default representatives?

   (c) Give a very brief statement about the results of parts (a) and (b). The statement should involve ideas like how similar the different values $v_1, \ldots, v_K$ are.