Kernel ridge regression and neural networks

November 26, 2018

1 Problem 1: Time complexity of kernel ridge regression training (20 points)

Consider two variants of the kernel ridge regression algorithm for a given kernel $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. The first one uses exact kernel values and the second one applies random feature map trick, i.e. applies transformation $\phi_K : \mathbb{R}^d \rightarrow \mathbb{R}^m$ to all datapoints (where ϕ_K is a mapping defining random feature maps for a particular kernel K) and then applies linear regression on the transformed data. You can assume that $m \gg d$. Denote by N the number of all datapoints in the training set. Which method is more computationally efficient (training time) if $N \ll m$ and which when $N \gg m$? Explain your answer.

2 Problem 2: Autoencoders (40 points)

Consider a dataset $\mathcal{X} \subseteq \mathbb{R}^d$. Assume that there exists an l-dimensional linear subspace $\mathcal{L}$ $(l < d)$ spanned by the independent set of vectors $\{\mathbf{v}_1, ..., \mathbf{v}_l\}$ such that for each datapoint $x \in \mathcal{X}$ we have: $x \in \mathcal{L}$. Design an autoencoder minimizing error on that dataset and retrieving the lower-dimensional structure of that data. What is the error obtained by this autoencoder? Describe in detail the architecture of your autoencoder (e.g. define connection matrices, etc.).

3 Problem 3: Neural networks for MNIST (50 points)

Propose and implement neural network algorithm for MNIST images classification. What is training and test accuracy that you get ? Send the code, detailed description of the algorithm and all results (training curves, etc.). This time use entire training and test dataset.