Assignment 4

November 10, 2020

All files referred to in this homework can be found on CourseWorks.

Your hand-in should be made on Gradescope. Your submission is going to be a jupyter notebook that contains answers to all the questions. Please use headings to separate your notebook into answers to each question.

To see examples of how to write latex code in jupyter, see this link.

You must submit *two* files:

1. An `.ipynb` file (i.e. jupyter notebook file) with your data analysis for answering questions
2. A pdf file with your jupyter notebook output. Please use the workflow described here for generating this pdf.

1 SVMs

1.1 Scaling the Inputs

True or false: in training an SVM it is generally a good idea to scale all input variables so that, for example, they all lie in some fixed interval or so that they all have the same mean, μ, and variance, σ^2, e.g $(\mu, \sigma^2) = (0, 1)$. Justify your answer.

1.2 Classifying Tumors

1. Load the breast cancer dataset using sklearn.datasets. Construct an SVM classifier for this data. You should randomly assign $t\%$ of your data to the training set and the remainder of your data to the test set. Then use cross-validation on your training set to build your classifier. You can take $t = 70\%$ initially.
2. Repeat part (1) $N = 50$ times to get N samples of the performance of the trained classifier on the test set. (Note that each of the N samples will have different training and test sets.) Compute the mean and standard deviation of the test- set-performance.
3. Repeat part (b) for values of $t = 50\%, 55\%, \ldots, 95\%$ and plot the mean test-set performance together with 95% confidence intervals for this performance against t. What conclusions can you draw?

1.3 SVMs and Cross-Validation

Suppose you have successfully trained an SVM with 10,000 training points and a Gaussian kernel where the values of C and σ were selected via cross-validation. Recall that the Gaussian kernel has the form

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

You are then given an additional 40,000 training points and so you wish to retrain your SVM using the entire 50,000 training points that you now have. However, you wish to avoid the heavy computational

expense associated with repeating the cross-validation exercise that you previously used to pick C and σ. Instead, you simply use the C and σ that you found using the first 10,000 training points, and then retrain your SVM using those hyperparameters, but on the new set of 50,000 data points. Do you see any potentially major problem with this? If so, what is it?

2 PyTorch Practice

1. Install PyTorch. I recommend using anaconda, in which case you can do it with the conda package manager: `conda install pytorch torchvision -c pytorch`
2. Do the PyTorch 60-minute blitz tutorial
3. Your jupyter notebook should have a section where you run each command from the PyTorch 60-minute blitz (this will only be lightly graded). You do not need to run the GPU commands.
4. Create a neural network with two hidden layers (the notebook shows how to create one with one hidden layer), both layers should be ReLU layers (you may simply take the one from the notebook and add a second layer with 256 output features, but feel free to get more creative)
5. Try SGD, Adam, and at least one other optimization algorithm from torch.optim. Try at least 3 different stepsizes for each algorithm (for Adam you should also try the default stepsize). Report on your experience with finding a reasonable stepsize for each algorithm (e.g. how sensitive is each algorithm to stepsize), and how the algorithms compare on loss minimization, training accuracy, and test accuracy.
6. If you pick the best setup from all your experiments above, based on either loss or training accuracy performance, do you get the best algorithm on test accuracy?

3 Function Approximation

1. Consider a ReLU network with a single hidden layer, with $W^{(1)} \in \mathbb{R}^{2\times 2}, x, b^{(1)} \in \mathbb{R}^2$ and $W^{(2)} \in \mathbb{R}^{1\times 2}, b^{(2)}, y \in \mathbb{R}$:
   - $h^1 = \sigma(W^{(1)}x + b^{(1)})$
   - $\hat{y} = W^{(1)}h^1 + b^{(2)}$

   Show that this network is a piecewise linear function. Specify the set of pieces and the value on each piece.
2. Consider a continuous piecewise-linear function

$$f(x) = \begin{cases} x + 3 & \text{if } x < 5 \\ 2x - 2 & \text{if } 5 \leq x < 10 \\ -1x + 28 & \text{if } 10 \leq x \end{cases}$$

   Show how to represent it with a ReLU network that uses a single hidden layer.