From Data to Insights - Exercise sheet 11

discussed July 4 and 5

June 28, 2024

1 Deep learning inference of power spectrum amplitude

A scalar field $\delta(\boldsymbol{x})$ is defined on a spatial grid $\boldsymbol{x}$. It is a Gaussian random field, namely the outcome of a random experiment as follows:

- The Fourier space representation of $\delta(\boldsymbol{x})$, $\tilde{\delta}(\boldsymbol{k})$ is drawn as independent zero-mean Gaussians whose variance as a function of k is set by a function called the power spectrum.
- We do know the shape of the power spectrum a priori, but a parameter of interest is its amplitude A. Our prior probability for A is a uniform distribution between 0.1 and 100.

It turns out that there is an analytical solution to getting a minimum-variance estimate on A (in the limit that the prior is uninformative): You have to take the Fourier transform of $\delta(\boldsymbol{x})$ to arrive back at $\tilde{\delta}(\boldsymbol{k})$, square the Fourier amplitudes at each k, and average them with appropriate weighting and normalization according to the known shape of the power spectrum. Here we try instead to use neural networks to provide an estimate of A based on observed density fields.

Download the notebook `d2i_problemset_11.ipynb` from the cloud and run it / add some code or edit settings to answer the following questions, which are also mentioned as comments in the notebook.

1. Why does the structure of the fully connected NN we are setting up in the `PowNet` class match the problem at hand? What motivates the choice of activation function and cost function, and the initialization of weights?
2. What choices are made in the proposed training scheme (`first_attempt_at_training`)?
3. Plot a few entries in the test data and note their amplitudes in the plot.
4. Let's try to learn the optimal weights from the training data. Describe the degree of success in this first attempt of training, also including a comparison to the "optimal" result above this step in the notebook.
5. Perhaps we made some poor choices. Try to change some of them ("hyperparameters") and see how this perhaps improves the training outcome.
6. What happens when you continue to train the model whose weights are already set to the optimal configuration (FFT + square + matched filter) before training? Again, see whether changes to the training routine could improve the outcome.
7. Interpret your qualitative findings in the context of double descent.
8. Bonus: instead implement a deeper (i.e. multi-layer), convolutional neural network and apply it to the problem. How well does it perform? Better than the from-scratch trained fully connected network? Better than the algebraically initialized fully connected network? Why / why not?