COM503: Deep Learning

Term Paper

Question 1. You are given a set of the same data set as was for COM502: Machine Learning. You are now well aware that the Gaussian part is simply random with weak linear tendency, and the non-Gaussian part is closely related to AR(1), i.e. yesterday's stock returns very much foretell today's and tomorrow's returns. Now that you have learned deep-learning, it is your time to counterargue your deep-learning(DL) maniac boss with reasonable detail. Although you feel that simple comparison between statistical approach and most fitted DL model should be enough to convince your ignorant boss, you would like to build your argument concrete, in case your boss brings in external authority as ignorant as him.

1. From the earlier set of exercise, you would like to split the model into two groups. One with near Gaussian process and the other with non-Gaussian. Since each part is not concatenated, you have to build a piecewise model with state indicator. Prove or counter-prove whether non-linear terms give you any distinctive benefit in terms of R^2. If your model is not piecewisely determined, i.e. you pick 1. full Gaussian assumption, 2. full non-Gaussian assumption 3. no assumption of distribution, in comparison to your piecewise specification, what's the benefit of yours? What's the cost?

2. For state specification, is there any benefit if you turn your attention to 2nd- or higher-moments, including ARCH, GARCH? In the time-span of the data, when is the most ideal case? Where do you find it more apparent? Do you see any reason to focus on industry variables?

3. You have learned well from earlier math courses that all models must represent data generating process (DGP), and you indeed can see DGP-based model is costly to build from 1). You would like to minimize the cost. First, to keep the computational cost low, you have chosen Kalman filter. What will be your state specification? Does the Kalman approach help you build the model less costly? If so, in what sense? Does the model perform better? If so, in which part?

4. Though Kalman does give you one benefit, you also witness costs that you have to pay for. Your assumption is that the model is too rigid to linearized process, therefore inviting non-linearity can be a worth trade-off experiment. Instead of learning non-linear Kalman, you have chosen to adapt RNN (Recurrent Neural Network) models with some state variable. Loosely illustrate your RNN specification and argue the needs and functions of state variables. How many do you believe is needed? Do you need state variables across the time series? Where is your model's ’memory’?

5. Given universal approximation theorem, can you argue the model can create non-linearity without being influenced by data? If the DGP does affect, what's your estimation strategy with any Neural Network models?

6. With or without steps proposed in 5), can you argue RNN performs any better than your earlier piecewise specifications? What about LSTM or GRU? (All you are recommended to do is to change parameters that indicate RNN types in your DL library. No further modification is required.)

7. Why is stochastic process not fit to non-linear models? How is your approach in 5) related to stochasticity of the data? In the data's time line, when does that become more obvious than others? If you have found any additional state variables with higher moments, does this help your RNNs? Can E-M approach be any help? Why or Why not?

8. What is the advantage and disadvantage of RNN models in terms of back-propagation? Can you perform ensemble like estimation with RNNs? If so, how? Why can you not use other DL models for our case? Given your argument in 7), why is it difficult to invite E-M for our DGP? If there is any work around, outline it in words. If not, discuss why it is not.

9. Just like COM502's 9), you are now given to test your model only for momentum. Compare your piece-wise (logit) and RNN models with other models, only if possible. In your comparison, provide a confusion matrix of each model result.

10. Your boss is completely dissatisfied with your above arguments. He has two counterarguments. One is for accuracy that he claims your optimizer (supposedly Adam for all cases) is problematic. He claims that you should use RMSPrp or Momentum optimizers. What is your rebuttal? In addition to your illustration with a summary graph, you must provide logic in words. For the computational cost, he insists that pre-learning by autoencoder give decisive advantage to RNN models. Do you see any factors in your time series that can be argued for autoencoder?

Bonus. Assume that you have more data by simulation that reflects 1st, 2nd, 3rd, and 4th moments of each pieces. It should be randomly generated so that you can avoid data overlaps. The process is similar to GAN, but you do not have factors, therefore you have to play with moments. Are these reasonable factors? Why or why not? With more simulated data, can you argue that you have more information? Does your model supposedly perform any better?

In your answer, provide necessary arguments and graphs. For questions that mathematical derivation can sharpen your argument, precise intuitive reasoning can be sufficient. Good luck with the term paper.