1. General Background

I am a mathematician by training. So in order to work with me, in addition to being extremely independent, you should have a strong background in probability and analysis. Specifically, you should have received an A in ORF 523, ORF 525, ORF 526, and ORF 527, or their equivalents. You will also be expected at some point to learn high dimensional probability at the level of getting an A in ORF 550.

2. My Research

My research these days is focused on the theoretical study of neural networks. The main directions of my work are:

- Understanding the approximation capacity of neural networks [HS17, Han17, DDF+19, DHP21]

- Understanding the behavior of neural networks at the start of training [HR18, Han18, HN19a, HR19, HR, HN19a, Han21a, HP21, RYH21, HJR21]

- Understanding generalization for neural networks [RYH21, HS, HN19a, Han21b]

The questions I am interested in tend to be strongly rooted in understanding how neural networks work in practice and how to make them better. Mathematically, my work typically uses tools from stochastic processes, random matrix theory, high dimensional probability, combinatorics, random geometry, and functional analysis.

3. Advising Style

My advising style emphasizes your need to be broadly interested in theoretical machine learning and mathematics. This means that rather than giving you a specific problem to work on, I will help you choose a direction of study by providing the input Papers to the advising algorithm listed in the next section. This algorithm is meant to get you up to speed in an area. Once it is successfully executed, I’ll help you look for a problem to work on. In the meantime, if you’re potentially interested in working me, you should participate (i.e. attend and present at) my informal seminar on deep learning theory, which runs every week. You should also attend my ORF 543 course on deep learning theory.
4. Advising Algorithm

```python
def BorisAdvising(Papers):
    CurrentUnderstanding = 0
    TargetUnderstanding = 1

    While interested == True and CurrentUnderstanding < BroadlyKnowledgeable:
        Learning(CurrentUnderstanding, TargetUnderstanding, Papers)
        Result = Presentation(Time = 20min, Venue = AtBoard,
                                CurrentUnderstanding, TargetUnderstanding)

        If Result == True:
            CurrentUnderstanding += 1
            TargetUnderstanding += 1
            Presentation(Time = 50min, Venue = LearningSeminar,
                           CurrentUnderstanding, TargetUnderstanding)

    def Learning(CurrentUnderstanding, TargetUnderstanding, Papers):
        While CurrentUnderstanding < TargetUnderstanding:
            Read(Papers)
            Read(References)

    def Presentation(Time, Venue, CurrentUnderstanding, TargetUnderstanding):
        If Venue == AtBoard:
            While CurrentUnderstanding < TargetUnderstanding:
                Read(Papers)
                Read(References)

            return CurrentUnderstanding >= TargetUnderstanding
```


