PROBLEM SET 3: APPLIED MATHEMATICS 216

Due: Sunday March 3 at 11:59pm

Problems.

(1) **The Ising Model, continued** We will continue our discussion of the Ising Model. You will recall the Ising class we gave you last week, that allowed simulations of the 2d Ising Model on a square lattice. Now we also provide an Ising class for doing simulations on a triangular (or hexagonal) lattice. The data is generated using a configuration as the left figure of Figure 1 in the link. The model here is the same but the number of nearest neighbors is different. It is well known that the transition temperature for the phase transition changes between the different lattices, but the basic physics of the phase transition remain the same.

We now are providing 2 different datasets: the first contains spin configurations for the square lattice of the Ising models at four different temperatures; the second contains spin configurations for the triangular lattice at four temperatures.

(a) Retrain your SVM (SVC) to train the classification model on both datasets two classify the 4 temperature classes.

(b) Train a fully connected neural network to do the classification, on both datasets.

(c) Train a convolutional neural network to do the classification, on both datasets. Make a table of your performance numbers for these three different models and upload these numbers. This, together with your code, should be uploaded to the course website when you turn in your homework.

(d) We have provided a test set of 10 spins configurations for each of the two problems. Each of the spin configurations is *not necessarily* at the temperatures of the training sets. Calculate your best estimate of the termperatures of these spin configuration. Upload your results to Kaggle.

(e) *Transfer Learning*. As we emphasize in class, one can freeze the training of the bottom of a network and retrain the top part of the network to adopt to a new situation. Use your CNN that you trained on the square lattice data to do transfer learning on the triangular lattice data. How does the performance compare to that of the direct methods? Add the performance numbers for transfer learning in your table from Part (c). Note that the training time for transfer learning is far below that for the direct optimization. For example, is 50 triangle example sufficient for the re-training process? Use your transfer learning result to predict the transition temperature of triangle lattice Ising model, as demonstrated in this Nature Physics publication.

(2) **Using Inception to Solve Scientific Problems** A state-of-the-art image classification neural network is called *Inception*. You can download the weights for inception from tfhub, as we demonstrated in the lab. Here we will add extra layers to the top of inception and retrain them on two different scientific problems:
   (a) The Ising Model – show that the square lattice data can be trained perfectly with inception.
   (b) Rayleigh-Bénard Convection, in which a flow is heated from below and cooled from top, is one of the paradigmatic system in fluid dynamics. When the temperature difference between the two plates (in dimensionless form Rayleigh number Ra) is beyond certain threshold, hot fluid tends to go up and cold fluid tends to go down, thus forming convection cells. What we supply here are the temperature snapshots from four different Ra, i.e. $Ra = 10^{14}$ as class0, $Ra = 10^{13}$ as class1, $Ra = 10^{12}$ as class2, and $Ra = 10^{11}$ as class3. The flow you see is highly turbulent; not only there are big convection cells but also lots of small vortices. The original dataset is around 4000*2000. We have already downsampled the data into the zip file "fluid_org".
      (i) Train the data in "fluid_org" with inception. Note that Inception takes pictures with 299×299 pixels. Shown that these images can be classified into different Ra nicely with inception. (You might find the python package PIL to be helpful here.)
      (ii) In the other dataset "fluid_crop" we have randomly choosen a region of 100×100 pixels from each original 4000*2000 pictures, i.e., just around 1% of the original picture! Again train the data with inception.
      (iii) Compare all the results with the trains without incepetion.
      (iv) Do you think that the network you trained on the 100×100 randomly cropped pictures produces a satisfactory prediction? If so, what critical points do your neural network capture? If not, what critical points do your neural network missed?