Some Demonstrations of the Effect of Number of Trainable Parameters on Performance of a ML Setup (Non-Assessed)

Anirbit *anirbit.mukherjee@manchester.ac.uk*
Department of Computer Science
The University of Manchester

Demonstration-1 Try understanding what this code is doing, (link)

The code is set to a choice of hyperparameters such that it shows a large net outperforming a small net.

You are strongly encouraged to understand this code towards the following goals,

1. It gives a template for how to write a "professional" neural net code using a very standard data in PyTorch – the most ubiquitous neural training software! (This is an essential skill for almost any ML job at this point.)
2. In particular, you will get to see the Adam algorithm in action - which is the baseline for all predominant methods of training nets.
3. You are urged to play around with the different hyperparameters like the minibatch size, the step-length and width of the net to explore how stable is the phenomenon shown in the demonstration.

   Upon doing the above, do plot any of the metrics of performance (at the end of training) vs variations of these key hyperparameters. Overlay these plots for the underparameterized and the overparameterized model and vividly see how the hierarchy of performance between them depends on these settings.

Demonstration-2 Try understanding what this code is doing,(link).

The above shows how for the same training data and the same range of number of trainable parameters, the performance can be non-monotonic with the size for the net but be monotonic for a linear model. This should vividly drive home the point that the effect of size on performance is significantly driven by the non-linearity of the predictor.

Try investigating how much of this demonstration is stable when the labeling function is changed or the architecture is changed.