R16/Ishank Juneja/16D070012

March 24, 2020

Tangram Puzzle and Connectionism

The *Tangram* is a puzzle where seven polygonal pieces have to be arranged in a manner so as to completely fill up a geometrical figure. These seven polygonal pieces consist of two large triangles, a square, a medium triangle, a parallelogram and two small triangles. The only restriction on the figure to be filled up is that there should exist at least one complete and exact covering of its region using the seven puzzle pieces. The simple act of overlaying a regular grid of points over the underlying geometrical figure modifies the problem to the *Grid Tangram*.
The paper by Oflazer presents an approach to solving the Grid Tangram using a *connectionist* method. Connectionism is an AI paradigm where computations are made using a network of neuron-like units. The paper in question came out in 1993 and in the decade leading up to its publication, artificial neural network models had been deployed to obtain approximate solutions to computationally hard problems. It is interesting to observe that present day deep neural network architectures had their foundations laid during this time.

Boltzmann Machine Architecture

Boltzmann machines (BM) are a type of stochastic neural network which are useful for solving constraint satisfaction problems in an approximate manner. The underlying design for the BM consists of a Hopfield Network, which is a boolean neural network consisting of memory elements and feedback connections. In present day neural network terminology, the Hopfield network would be called a recurrent neural network. Unlike a deep neural network, there is no notions of numbered hidden layers in BMs. In a BM, each unit computes a weighted sum of inputs connected to it. If the sum exceeds the units threshold then the unit outputs 1. If the sum does not exceed the threshold, the output is stochastic and is governed by a probability distribution generated using the precise value of the sum. The probability distribution used in the latter case is dependent on another parameter of the network known as its *temperature*.
The way the temperature, influences the probability distributions is that the probably of outputting a 1 in absence of the sum crossing the threshold increases with temperature. To achieve convergence onto a solution, the temperature is reduced gradually during the operation of a BM.

Model Applied to the Tangram Puzzle

To solve constraint satisfaction problems BMs are configured such that each neuron unit represents a hypothesis and the interconnections between the units represent constraints. Hypotheses that are consistent reinforce each other through positive weights while competing hypothesis are connected with negative weights. To solve the Tangram problem using BMs, the following quantities need to be represented using neuronal units-

- Each point on the problem grid must be represented using neuronal units. This is done by associating eight units with each grid point, one for each of the cardinal and ordinal directions. Each of the eight indicates whether or not the puzzle area extends beyond the grid point in its own direction.
- The shape, orientation and position of each piece is encoded using neuronal units. A different number of units are needed for every puzzle piece.
- Any unit with all its positive connections (described above) active can be the part of a solution if it does not conflict with another puzzle piece's attributes. In case a clash happens, the inhibitory constraints kick in. These inhibitory/correctness constraints are
  - Only one set of units representing a particular type of piece can be active since the same piece cannot be in two places at the same time.
  - Any grid-point can only be covered by a single puzzle piece

When neural networks are used for a pattern recognition task, a large collection of labelled data is required to train the network and learn its weights through backpropogation. In the Tangram setting, since the puzzle is understood completely, such training is not required and weights can be hand-coded. In the BM used in the paper, excitatory and inhibitory weights are selected in a manner such that if the sum of the excitatory units for a certain position and configuration of a puzzle piece is < 1, then it is rejected, if the sum is $= 1$ and there are no contradictions, then the output is accepted (set to 1) and if the sum is $= 1$ but there are inhibitory inputs then the output is set to 1 as per a probability distribution resembling the fermi-dirac distribution from statistical physics.
Following these rules, outputs of the units corresponding to the seven puzzle pieces are updated in an asynchronous manner and in a random order. Once a configuration in which all seven units are active, and no inhibitory constraints are violated, is achieved, a solution is obtained. Though a baseline for comparison of the effectiveness of this method, compared to say, a brute-force combinatorial approach has not been provided, the method seems quite promising.
A question I have is -

- Has a computational advantage been gained by employing Boltzman machines to the problem? Based on the number of links used, it seemed like we are still brute-forcing the solution albeit with a network architecture. This is also relevant since a proof of convergence for the BM method has not been provided, so there is already a trade-off there.