CSC321: Introduction to Neural Networks and machine Learning

Lecture 16: Hopfield nets and simulated annealing

Geoffrey Hinton
Hopfield Nets

• A Hopfield net is composed of binary threshold units with recurrent connections between them. Recurrent networks of non-linear units are generally very hard to analyze. They can behave in many different ways:
  – Settle to a stable state
  – Oscillate
  – Follow chaotic trajectories that cannot be predicted far into the future.

• But Hopfield realized that if the connections are symmetric, there is a global energy function
  – Each “configuration” of the network has an energy.
  – The binary threshold decision rule causes the network to settle to an energy minimum.
The energy function

• The global energy is the sum of many contributions. Each contribution depends on one connection weight and the binary states of two neurons:

\[ E = - \sum_{i} s_i b_i - \sum_{i<j} s_i s_j w_{ij} \]

• The simple quadratic energy function makes it easy to compute how the state of one neuron affects the global energy:

\[ E(s_i = 0) - E(s_i = 1) = b_i + \sum_{j} s_j w_{ij} \]
Settling to an energy minimum

- Pick the units **one at a time** and flip their states if it reduces the global energy. Find the minima in this net.

- If units make **simultaneous** decisions the energy could go up.
How to make use of this type of computation

• Hopfield proposed that memories could be energy minima of a neural net.
• The binary threshold decision rule can then be used to “clean up” incomplete or corrupted memories.
  – This gives a content-addressable memory in which an item can be accessed by just knowing part of its content (like google)
  – It is robust against hardware damage.
Storing memories

• If we use activities of 1 and -1, we can store a state vector by incrementing the weight between any two units by the product of their activities.
  – Treat biases as weights from a permanently on unit
• With states of 0 and 1 the rule is slightly more complicated.

\[ \Delta w_{ij} = s_i s_j \]

\[ \Delta w_{ij} = 4 \left( s_i - \frac{1}{2} \right) \left( s_j - \frac{1}{2} \right) \]
Spurious minima

• Each time we memorize a configuration, we hope to create a new energy minimum.
  But what if two nearby minima merge to create a minimum at an intermediate location?
  This limits the capacity of a Hopfield net.

• Using Hopfield’s storage rule the capacity of a totally connected net with N units is only 0.15N memories.
  – This does not make efficient use of the bits required to store the weights in the network.
Avoiding spurious minima by unlearning

• Hopfield, Feinstein and Palmer suggested the following strategy:
  – Let the net settle from a random initial state and then do unlearning.
  – This will get rid of deep, spurious minima and increase memory capacity.
• Crick and Mitchison proposed unlearning as a model of what dreams are for.
  – That’s why you don’t remember them (Unless you wake up during the dream)
• But how much unlearning should we do?
  – And can we analyze what unlearning achieves?
Willshaw nets

- We can improve efficiency by using sparse vectors and only allowing one bit per weight.
  - Turn on a synapse when input and output units are both active.
- For retrieval, set the output threshold equal to the number of active input units
  - This makes false positives improbable
An iterative storage method

• Instead of trying to store vectors in one shot as Hopfield does, cycle through the training set many times.
  – use the perceptron convergence procedure to train each unit to have the correct state given the states of all the other units in that vector.
  – This uses the capacity of the weights efficiently.
Another computational role for Hopfield nets

- Instead of using the net to store memories, use it to construct interpretations of sensory input.
  - The input is represented by the visible units.
  - The interpretation is represented by the states of the hidden units.
  - The badness of the interpretation is represented by the energy.

- This raises two difficult issues:
  - How do we escape from poor local minima to get good interpretations?
  - How do we learn the weights on connections to the hidden units?
An example: Interpreting a line drawing

- Use one “2-D line” unit for each possible line in the picture.
  - Any particular picture will only activate a very small subset of the line units.
- Use one “3-D line” unit for each possible 3-D line in the scene.
  - Each 2-D line unit could be the projection of many possible 3-D lines. Make these 3-D lines compete.
- Make 3-D lines support each other if they join in 3-D. Make them strongly support each other if they join at right angles.
Noisy networks find better energy minima

• A Hopfield net always makes decisions that reduce the energy.
  – This makes it impossible to escape from local minima.
• We can use random noise to escape from poor minima.
  – Start with a lot of noise so it’s easy to cross energy barriers.
  – Slowly reduce the noise so that the system ends up in a deep minimum. This is “simulated annealing”.

![Energy landscape with local minima A, B, C]
Stochastic units

- Replace the binary threshold units by binary stochastic units that make biased random decisions.
  - The “temperature” controls the amount of noise
  - Decreasing all the energy gaps between configurations is equivalent to raising the noise level.

\[
p(s_i=1) = \frac{1}{1 + e^{-\sum_j s_j w_{ij} / T}} = \frac{1}{1 + e^{-\Delta E_i / T}}
\]

Energy gap \( = \Delta E_i \) \( = \) \( E(s_i=0) - E(s_i=1) \)
The annealing trade-off

- At high temperature the transition probabilities for uphill jumps are much greater.

\[ p(\text{pick higher energy state}) = \frac{1}{1 + e^{\Delta E/T}} \]

- At low temperature the equilibrium probabilities of good states are much better than the equilibrium probabilities of bad ones.

\[ \frac{P_A}{P_B} = e^{(E_A - E_B)/T} \]
How temperature affects transition probabilities

- High temperature transition probabilities:
  \[ p(A \rightarrow B) = 0.2 \]
  \[ p(A \leftarrow B) = 0.1 \]

- Low temperature transition probabilities:
  \[ p(A \rightarrow B) = 0.001 \]
  \[ p(A \leftarrow B) = 0.000001 \]
Thermal equilibrium

• Thermal equilibrium is a difficult concept!
  – It does not mean that the system has settled down into the lowest energy configuration.
  – The thing that settles down is the probability distribution over configurations.

• The best way to think about it is to imagine a huge ensemble of systems that all have exactly the same energy function.
  – The probability distribution is just the fraction of the systems that are in each possible configuration.
  – We could start with all the systems in the same configuration, or with an equal number of systems in each possible configuration.
  – After running the systems stochastically in the right way, we eventually reach a situation where the number of systems in each configuration remains constant even though any given system keeps moving between configurations.
An analogy

• Imagine a casino in Las Vegas that is full of card dealers (we need many more than 52! of them).

• We start with all the card packs in standard order and then the dealers all start shuffling their packs.
  – After a few time steps, the king of spades still has a good chance of being next to queen of spades. The packs have not been fully randomized.
  – After prolonged shuffling, the packs will have forgotten where they started. There will be an equal number of packs in each of the 52! possible orders.
  – Once equilibrium has been reached, the number of packs that leave a configuration at each time step will be equal to the number that enter the configuration.

• The only thing wrong with this analogy is that all the configurations have equal energy, so they all end up with the same probability.