Template for Brain/Machine Learning

Hopfield Networks

Paste your graph of energy versus iteration for the Hopfield network's evolution.

Paste your original corrupted image, an intermediate point and the final image of the smiley face.

Paste your plot of how many memories you can remember using the Hamming distance.

Paste both your graphviz landscape picture and your energy landscape diagram.

(*Optional Extra Credit*) Using your more efficient code, paste your original corrupted thumbnail, an intermediate point, and the final image of your thumbnail

Restricted Boltzmann Machines

For a small RBM with 2 hidden spins and 5 visible spins, plot the theoretical probability distributions against the distributions obtained by sampling from your RBM using Gibbs sampling. **p(v,h)** **p(v)** **p(h)** **p(v|h)**

Train a small RBM with 3 visible spins to match a small toy probability distribution. Plot the toy distribution q(v) on top of the distribution p(v) obtained by sampling your RBM using Gibbs sampling. Include error bars.

p(v) versus q(v)

Train an RBM on a random quantum circuit. Plot the true probability distribution q(v) on top of the distribution p(v) obtained by sampling your RBM using Gibbs sampling. Include error bars.

Quantum Circuit Description (before compiling)

p(v) versus q(v)

(Optional extra credit): **Train an RBM on MNIST. Using your trained RBM, generate 20 new images of digits by Gibbs sampling. To do this, each time initialize your visible spins to a random configuration and perform k=10000 Gibbs sampling iterations. Paste the 20 final configurations here.**

Feed forward neural networks (extra credit)

Demonstrate with a simple test that your one-layer neural network's output is correct.

Demonstrate with a simple test that your two-layer neural network's output is correct.

Demonstrate with a simple test that your two-layer neural network's cost function and gradients outputs are correct.

For a one and two-layer neural network, show that your backpropagation gradients match your finite difference gradients.

Train a neural net on the Ising model dataset we give you.

Print the accuracy of your trained network:

Plot the (average) output of your network $z^{(L)}$ as a function of inverse temperature beta J.