Artificial Neural Networks

Prof. Dr. Sen Cheng

Jan 27, 2020

Problem Set 14: Boltzmann Machines

Tutors: José R. Donoso (jose.donoso@rub.de), Nicolas Diekmann (nicolas.diekmann@rub.de)

Further Reading: A Practical Guide to Training Restricted Boltzmann Machines (Hinton; 2010)

1. **Contrastive Divergence.** You are provided with an unfinished implementation of a restricted Boltzmann machine (RBM) in *prob_rbm.py*. Implement a function *train* that trains the RBM using the 1-step contrastive divergence (CD) algorithm. Your function should make it possible to train the RBM using either the hidden units' activation probabilities or binary states during the positive phase.

2. **Training RBMs.** Load the *digits* data set provided by *sklearn*. Use the first 1200 examples as your training set and the rest as your test set (Note: Make sure to normalize the data). Do the following:

   (a) Train two RBMs, M_{binary} and M_{prob}, for 100 epochs, which use binary activations and activation probabilities during the positive phase, respectively. Use 16 hidden units and a learning rate of 0.01.

   (b) Save the trained RBMs.

   (c) Implement a function *reconstruction* that reconstructs the visible units for a given input (Note: During the positive phase do not use binary hidden units). Visualize the reconstructions of the first 32 examples of the test set for both RBMs. For comparison also visualize the ground truths.

   (d) Compute the mean squared error error (MSE) E_{rec} over all image pixels on the reconstructions of the test set. Plot E_{rec} as a function of training epoch for both RBMs. How do the error curves for the two RBMs compare?

3. **Pattern completion.** RBMs are generative models and are also able to repair or denoise corrupted input. Use the digits dataset and the two RBMs, M_{binary} and M_{prob}, from the previous problem. Load the networks you saved in problem 2b.

   (a) Apply increasing levels of Gaussian noise with $\sigma = \{0., 0.05, 0.1, ..., 0.45, 0.5\}$ to the images in the test set and let the RBMs reconstruct the input.

   (b) Compute E_{rec} for the reconstructions and plot E_{rec} as a function of noise level σ. How do the two RBMs differ in their ability to cope with increasing levels of noise?

4. **Data confabulation.** Generative models RBMs can confabulate new data. Use the two RBMs, M_{binary} and M_{prob}, from the previous problem and load the networks you saved in problem 2b.

   (a) Generate 16 random binary images. The images should be sparse with only 5%-10% units being active. Feed the random images to both RBMs and let them "reconstruct" the images for multiple iterations ($20-30$ iterations should suffice). Make sure to store the "reconstructed" images after each iteration.

   (b) Visualize the images confabulated by both RBMs at each iteration step. How do the images confabulated by the two RBMs, M_{binary} and M_{prob}, differ? How do they change over the course of iterations?