Math6373 spring2017 Deep learning and data mining Robert Azencott

Project 3 Due Sunday April 2 at midnight

Generic Boltzmann machines

Consider a generic Boltzmann machine having a total of N= 30 binary units denoted j=1, … ,30. The state of each unit only the values {+1 , -1}. For each pair of distict units (j, k), select at random the weight Wjk = Wkj in the interval [-1, +1]. Fix these weights.

Select a random initial binary configuration X_0 =[X0(1), X0(2), …, X0(30)]. Give the precise mathematical definition of the Boltzmann machine stochastic dynamics by a Gibbs sampler. Implement the Boltzmann dynamics to generate a random sequence of binary configurations X_0 X_1 X_2 , … , X_n ,... with n at least as large as 3000 (see below)

After each full sweep of the 30 units, and hence at each one of the steps 30 s , with s= 1,2,3,… , and for each unit "j" compute the empirical mean
$M_s(j) = (X_{30}\ (j) + X_{60}\ (j) + \ldots + X_{30s}) / s$

Fix a small threshold THR=1% and define the stabilization times

stab(j) = 1st value of s >100 such that | $M_{s+t+1}(j)$ - $M_{s+t}(j)$ | < THR for t=1,2,…,10

Define the stabilization time S of the machine as the smallest value of s such that more than 90% of the 30 units j verify stab(j) <= s

Define precisely the Boltzmann energy E(X) of any binary configuration X . At each time n = 30 s compute the energy En = E(X_n). Compute and display the histogram of the 200 energy values En where n= 30 S , 30 (S+1), …, 30 (S+200)

Repeat the preceding operations after replacing all your original weights Wjk by Wjk/10. Compare the results for S and for the energy histogram s

DATA BASE CAN BE THE SAME AS IN PROJECT 2 define an automatic classification task for this database : outline the characteristics of each feature F1 F2 …Fp ; present the distinct classes C1 C2 .. Cr ; select a training set and a test set

stochastic AUTO-ENCODERS based on the Restricted Boltzmann Machine (RBM)
select an RBM architecture with three layers L1 L2 L3 of dimensions **n1 , n2 = h , n3 = n1** ; the dimension h of the hidden layer will have to take many tentative values

- to determine **n1 = n3** indicate how you encode the value of each input feature Fk on specific "units" of layer 1

- the output layer will IDEALLY (after training) be able to generate an estimated output EstOut_j reproducing as well as possible the current input INP_j

- select 2 tentative values for h , namely 2 values h < n1=n3

Implement FAST RBM Learning algorithm to train the 4 auto encoders
explain clearly what is the algorithm for FAST RBM learning, and your choices for initialization of the weights, for batch sizes , for stopping the learning,

explain how you implement reading the outputs of the RBM

compute the Root Mean Squared Error RMSEn = sqrt(MSEn) at the end of each Batch BATn, plot the curve n➔ RMSEn and comment

after learning is stopped , call W* the terminal set of weights; the trained autoencoder is now parametrized by W*; compute the RMSE* of the trained autoencoder on the whole training set and on the whole test set; compare these performances;

Detailed analysis of hidden layer structure and efficiency (The training experiments generate 2 RBM Autoencoders)

PCA analysis: for each trained RBM autoencoder AUT, the set of all N training data generates a cloud of N configurations of the hidden layer H of AUT. Perform PCA for this cloud of N vectors in dimension h = dim(H). Compute the number of PCA eigenvalues needed to achieve 90% of the energy , and compute + display the projection of the p classes onto the first three eigenvectors Compare these results for your 4 autoencoders

Autoencoding efficiency : Define a new fixed output layer called OUTLAYER using exactly p nodes to encode classication outputs into p classes for the original classification problem . Let AUT be any one of the 2 RBM autoencoders just trained. Call L1 and H the first two layers of AUT , and FIX its already computed vector of weights W1 from L1 to H.
Define a new Boltzmann machine NewBM with 3 layers L1➔ H ➔ OUTLAYER, fixed weights vector W1 from L1 to H , and a new weight vector W2 from H to OUTLAYER

For this NewBM, implement Boltzmann Machine fast learning rule for classification in order to learn the adequate value of W2 but keep the W1 weights fixed during learning ; compute the percentage of correct answers on the training set and on the test set for this trained NewBM , and consider these numbers as evaluating the performance of the autoencoder AUT
Compute these two performances for your 2 RBM autoencoders and compare them