1 Discussion of result, restricted Boltzmann machine

I will briefly explain the code/algorithm I have implemented in matlab and then I will discuss the result given from figure 1.
So the algorithm train a restricted Boltzmann machine to learn the XOR data set. I have implement the algorithm 3 from the course book to train the restricted Boltzmann machine. When the training is done the Kullback-Leibler divergence as a function of the number of neurons M is determined. The Kullback-Leibler divergence is determined by iterate the dynamics of the restricted Boltzmann machine, where all possible patterns for 3 bits are considered. Then the probabilities for which the different patterns occur is determined. After getting the probabilities for all the patterns for all runs over the different numbers of neurons M, the Kullback-Leibler divergence is determined by equation (4.18) in the course book.

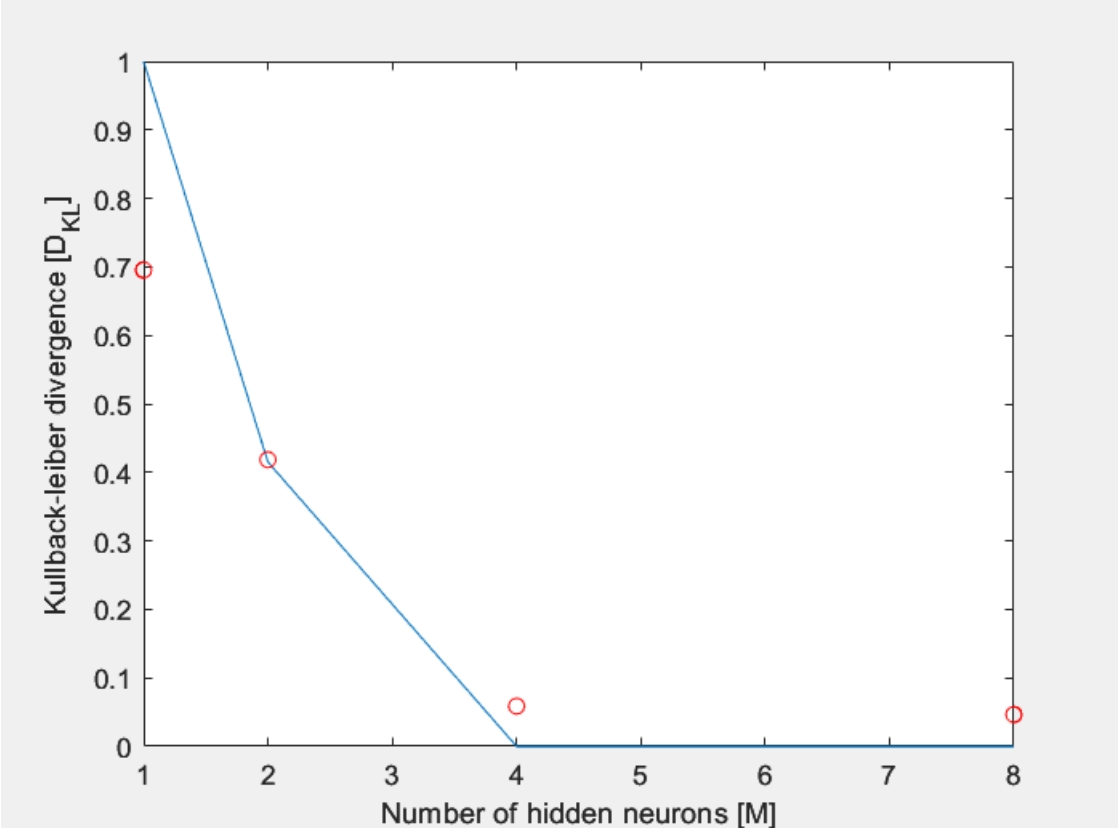

Figure 1: Red dots shows numerical estimates of the Kullback-Leibler divergence D_{KL} versus the number M of hidden neurons while the blue line shows the upper bound for D_{KL} determined as equation (4.40) in the course book

We can see in figure 1 that the restricted Boltzmann machine does approximate quite accurate but not to the optimal solution which corresponds to the blue line in the figure (equation 4.40 in the coursebook). The course book explains that the CD-k algorithm do not guaranteed to converge to the optimal solution and the CD-k algorithm is algorithm 3 that I have used to train the restricted Boltzmann machine so that's one reason why we don't get the optimal result. The result also depends a lot of the parameters, for example the learning rate, number of minibatches and so on. It also depends on how long we run the dynamics when we calculate the Kullback-Leibler divergence. I have an outer loop that runs 3000 times and an inner loop that runs 2000 times where frequencies

at which the different patterns occur is determined. I have tried different numbers on the outer and inner loop but, I found kind of good results for the numbers 3000 and 2000. The probabilities for both M = 4 and M = 8 gets kind of close to what we want which is 25% for the patterns in the XOR data set and the rest of the patterns to zero probability. For M = 8 some of the patterns in XOR gets a little bit higher than 25% and some gets a little bit smaller than 25% but not much. I also get that the patterns that are not in the XOR data set occure around 0.2% which is very low, almost to zero probability.