Artificial Neural Networks: Exercise session 4

Stijn Staring (r0620003)

May 23, 2021

1 Restricted Boltzmann Machines

The Boltzmann machine is a parametrized generative model representing a probability distribution. It is a neural network that belong to energy based models. A Boltzmann Machine consist out of a layer of visible units $\boldsymbol{v}$ and hidden units $\boldsymbol{h}$. The energy function used by the model is shown in Eq. 1a and is part of the joint probability distribution. The standard type of Boltzmann Machines, uses binary-valued hidden and visible units and is therefore convenient to be used for black white images. The difference between a Restricted Boltzmann Machine and a conventional Boltzmann Machine, is that the units of one layer are not connected with each other. Because of this restriction, the partition function in the joint probability distribution seen in Eq. 1b becomes tractable. A RBM trains in an unsupervised manner using only the observations and calculating the gradient of the log-likelihood of the observations with respect to the weight matrix $\boldsymbol{W}$, and bias terms $\boldsymbol{a}$ and $\boldsymbol{b}$. Training a RBM means adjusting the RBM weight values and biases such that the probability distribution $P(\boldsymbol{v})$ fits the training data as well as possible. The gradient of the log-likelihood is calculated by the difference between the data expectation and model expectation and is used together with a learning rate and a gradient decent update strategy. However, because the calculation of the model expectation is intractable, the Contrastive Divergence algorithm is used. Gibbs sampling is applied to generate an observation from the model distribution and is used in the Contrastive Divergence algorithm. Gibbs sampling is initialized by a known observation

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{v}^T \boldsymbol{W} \boldsymbol{h} - \boldsymbol{b}^T \boldsymbol{v} - \boldsymbol{a}^T \boldsymbol{h}, \tag{1a}$$

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v}, \boldsymbol{h})}. \tag{1b}$$

1.1 Hyperparameter tuning

A small hyperparameter tuning is displayed in Table 1. It is seen that more iterations give better results. When looking at the column with 20 iterations, an U-shape is noticed in the pseudo-likelihood. For a small learning rate of 0.005 a larger error is found due to the slow convergence and a larger learning rate of 0.1 performs worse due to oscillations during the application of gradient descent. The use of more hidden units gives an higher likelihood of the observations. Using more iterations and hidden units increases the calculation load of the model. The final chosen values for the parameters are 20 iterations, 20 hidden units and a learning rate of 0.01. In Figure 2 the comparison on the reconstruction of the number 4 is given with the tuned parameters and the default ones. It shows that the tuned RBM is able to better reconstruct an image with deleted row than when using the default parameter values that uses 10 iterations, 10 hidden units and a learning rate of 0.01 during training.

Learning rate/Iterations	5	10	20
0,005	-141	-134	-127,41
0,01	-166	-147,66	-125,55
0,05	-141,15	-134,35	-127,41
0,1	-134,46	-130,24	-130,81

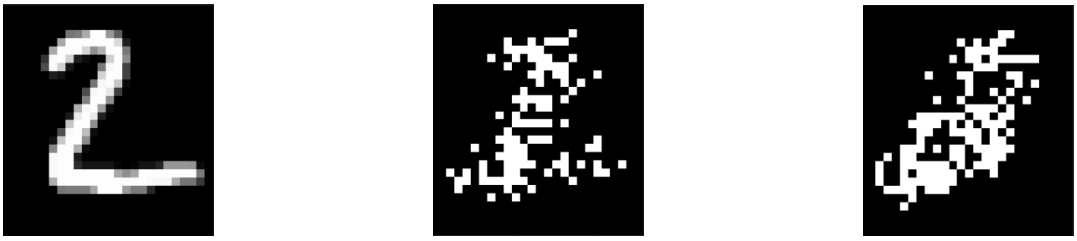
Iterations/Hidden units	5	10	20
5	-185	-155,08	-134,46
10	-186	-154	-130

Table 1: Pseudo-likelihood on the training data when training with different parameter values.

1.2 Gibbs sampling

In Gibbs sampling it is started from an initial known observation $\boldsymbol{v}^{(1)}$ and by the sequential use of $P(\boldsymbol{h}|\boldsymbol{v}^{(1)})$ and $P(\boldsymbol{v}|\boldsymbol{h}^{(1)})$, one iteration of the Gibbs sampling is completed and a new observation is obtained. A property of Gibbs sampling is that only after a infinite amount of iterations it can be guaranteed that the generated sample

originates from the model probability distribution that is tried to be sampled from. As can be seen in Figure 1 after one iteration the original number 2 is still visible but after 100 iterations it is not recognizable anymore. The influence of the initial observation is reduced after 100 iterations and the obtained observation will correspond better to one that originates from the model probability distribution.



(a) Original (b) 1 iteration (c) 100 iterations

Figure 1: The effect of iterations of Gibbs sampling.

1.3 Reconstructing images

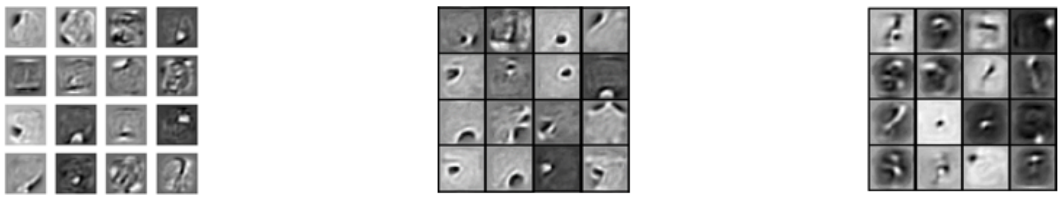
The general rule in reconstructing images is that when more information is missing from the original image, it is harder to reconstruct. Deleting more information can be done by deleting more rows or deleting crucial places that play a key role in distinguishing the number. Figure 2 shows the reconstruction of the number 4 using the default RBM and tuned RBM when rows 10 till 15 are removed. The tuned model shows better results. In order to reconstruct the image, the image with deleted rows is feeded as observation to the trained RBM and Gibbs sampling is applied. An observation is retrieved from the model probability distribution and the corresponding rows are used to fill the gap of the original image. For both images 10 Gibbs sampling steps are used.

(a) Default (b) Tuned

Figure 2: The effect of hyper parameter tuning.

2 Deep Boltzmann Machines

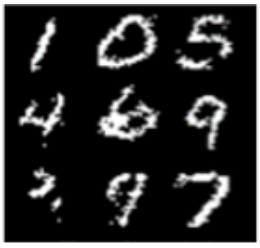
A Deep Boltzmann Machines is still restricted, which means that the nodes of the network of one layer are not connected. However, the difference with a shallow RBM is that it has multiple layers of hidden units stacked on top of each other. Due to this, a DBM can learn features that are non-linear combinations of features extracted in the lower layers. In comparison, the RBM can only extract features directly from the pixels. Therefore, more advanced features can be learned by a DBM. Figure 3 shows the weights of the hidden units of the RBM and two layers of the DBM. It is clear that both RBM and DBM show similar extracted features in layer 1. In the second layer more complex features are found that the RBM hasn't captured.



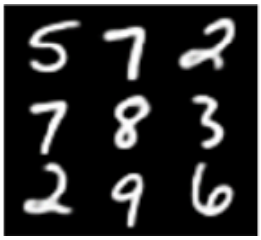
(a) RBM (b) DBM: L1 (c) DBM: L2

Figure 3: Weights of the units of a RBM and DBM.

That the DBM can better capture the important features of the digit is shown in Figure 4 where it seen that the images originating from a DBM are much more clear. For the generation of the images 1 Gibbs sampling iteration is used.



(a) RBM



(b) DBM

Figure 4: Generating digits with a RBM and DBM.

3 Generative Adversarial Networks