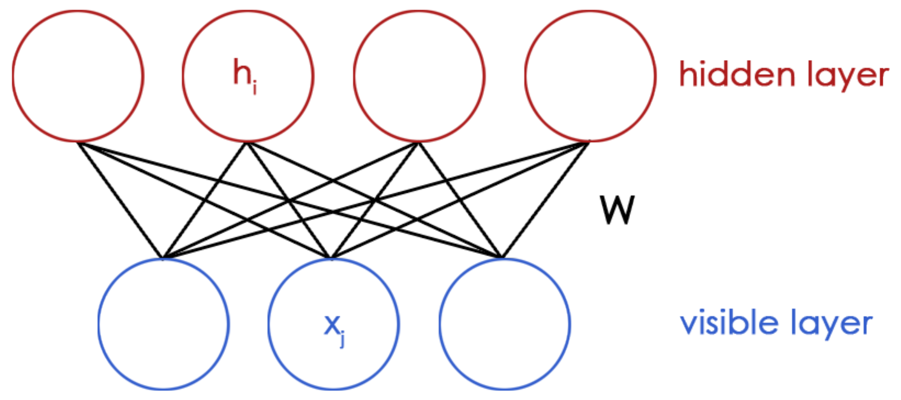
Restricted Boltzmann Machine

{ Chaimaa Kadaoui, Othman Sbai and Xi Shen } ENS Paris Saclay

Definition

A Restricted Boltzman Machine is an undirected graph which can be decomposed in two layers.



It is *restricted* because no connection between nodes of the same layer is supposed. The matrix W characterizes the connections between the two layers. For simplification, we suppose that the variables $\mathbf{x}$ and $\mathbf{h}$ are binary.

Energy and Probability

We introduce the two bias vectors c and b to define the energy of this graph:

$$E(x,h) = -h^\top Wx - c^\top x - b^\top h$$

We can write the joint probability of $\mathbf{x}$ and $\mathbf{h}$ as:

$$p(x,h) = \exp(-E(x,h))/Z$$

Z is the partition parameter which can be computed by calculating all the possibles values of $\mathbf{x}$ and $\mathbf{h}$. In practice, this parameter is intractable.

Inference We can prove that given $\mathbf{x}$, h_j follows a Bernoulli:

$$p(h_j = 1|x) = sigm(b_j + W_{j.}x)$$

$W_{j.}$ is the j-th row of W and $sigm$ defines the sigmoid function.

Contrastive Divergence

To train the RBM with our data $\mathbf{x}^{(t)}$, we would like to minimize the average loss: $\frac{1}{T}\sum_t l(f(\mathbf{x}^{(t)})) = \frac{1}{T}\sum_t -\log p(\mathbf{x}^{(t)})$

We will apply a stochastic gradient descent. We derive each term of the sum with respect to our model parameter θ as follow (where $\mathbb{E}_h$ and $\mathbb{E}_{x,h}$ are expectations of h and (x,h) respectively):

$$\frac{\partial -\log p(\mathbf{x}^{(t)})}{\partial\theta} = \mathbb{E}_h\left[\frac{\partial E(\mathbf{x}^{(t)},h)}{\partial\theta}|\mathbf{x}^{(t)}\right] - \mathbb{E}_{x,h}\left[\frac{\partial E(\mathbf{x},h)}{\partial\theta}\right]$$

The first term is called the *positive phase* and the second term the *negative phase*. Because of the difficulty to compute the second term, we will use an algorithm called *Contrastive Divergence*. The key points of the algorithm are:

- We estimate the expectation $\mathbb{E}_{x,h}$ by sampling a single point $\tilde{\mathbf{x}}$.
- To do so, we use Gibbs sampling in chain (we apply it k times).
- We initialize our Gibbs sampling with $\mathbf{x^{(t)}}$.
- Update parameters W, b, c

Conclusion

- Pellentesque eget orci eros. Fusce ultricies, tellus et pellentesque fringilla, ante massa luctus libero, quis tristique purus urna nec nibh. Phasellus fermentum rutrum elementum. Nam quis justo lectus.
- Vestibulum sem ante, hendrerit a gravida ac, blandit quis magna.
- Donec sem metus, facilisis at condimentum eget, vehicula ut massa. Morbi consequat, diam sed convallis tincidunt, arcu nunc.
- Nunc at convallis urna. isus ante. Pellentesque condimentum dui. Etiam sagittis purus non tellus tempor volutpat. Donec et dui non massa tristique adipiscing. [1]

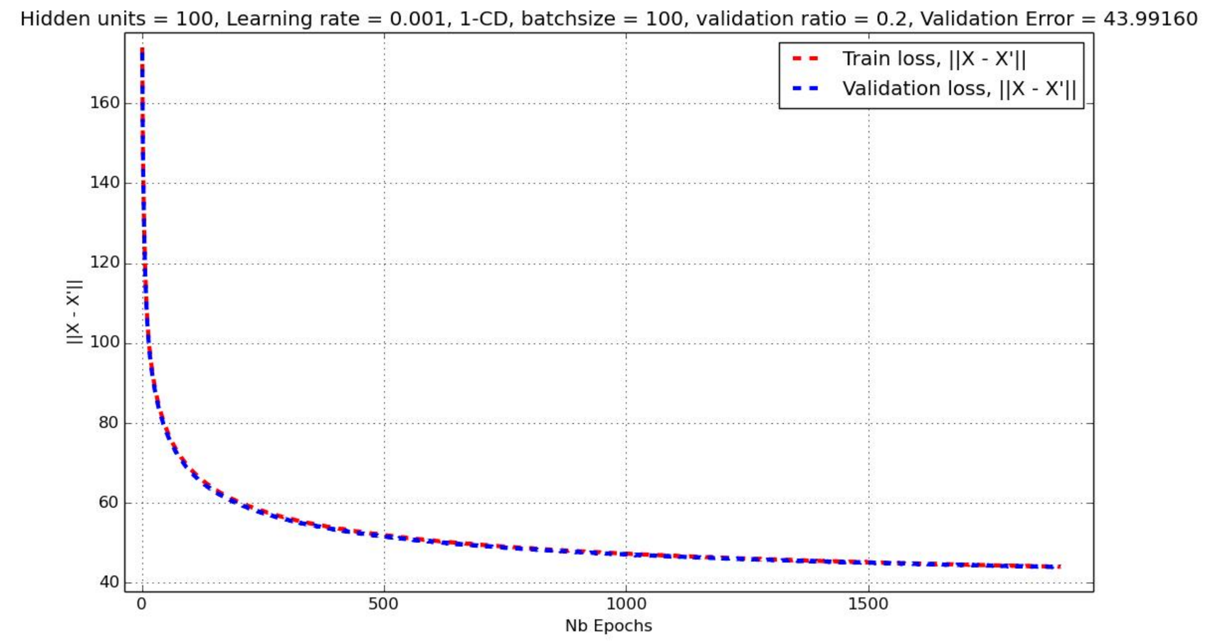
Future Research

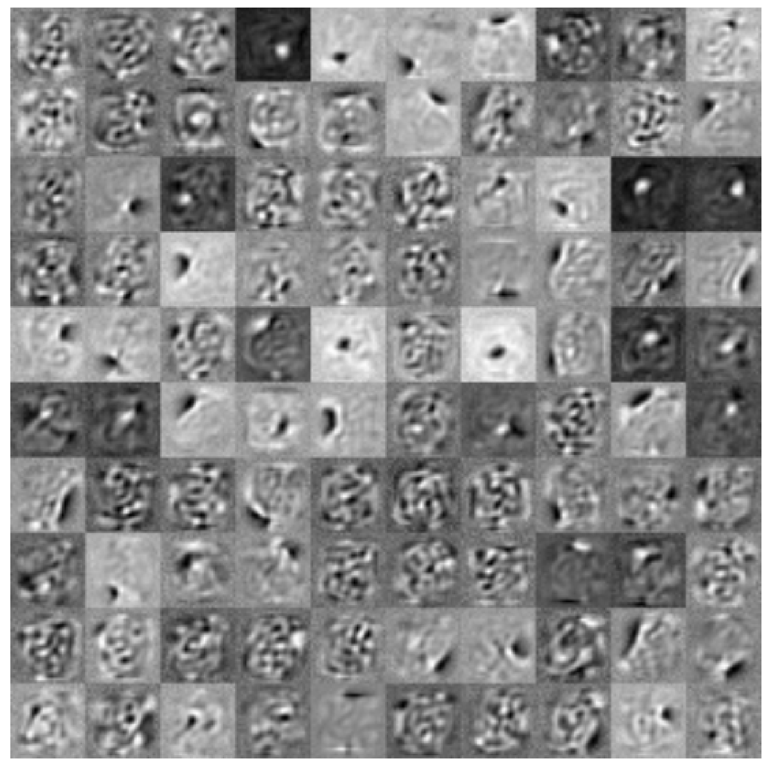
Integer sed lectus vel mauris euismod suscipit. Praesent a est a est ultricies pellentesque. Donec tincidunt, nunc in feugiat varius, lectus lectus auctor lorem, egestas molestie risus erat ut nibh.

Maecenas viverra ligula a risus blandit vel tincidunt est adipiscing. Suspendisse mollis iaculis sem, in *imperdiet* orci porta vitae. Quisque id dui sed ante sollicitudin sagittis.

Results On MNIST data

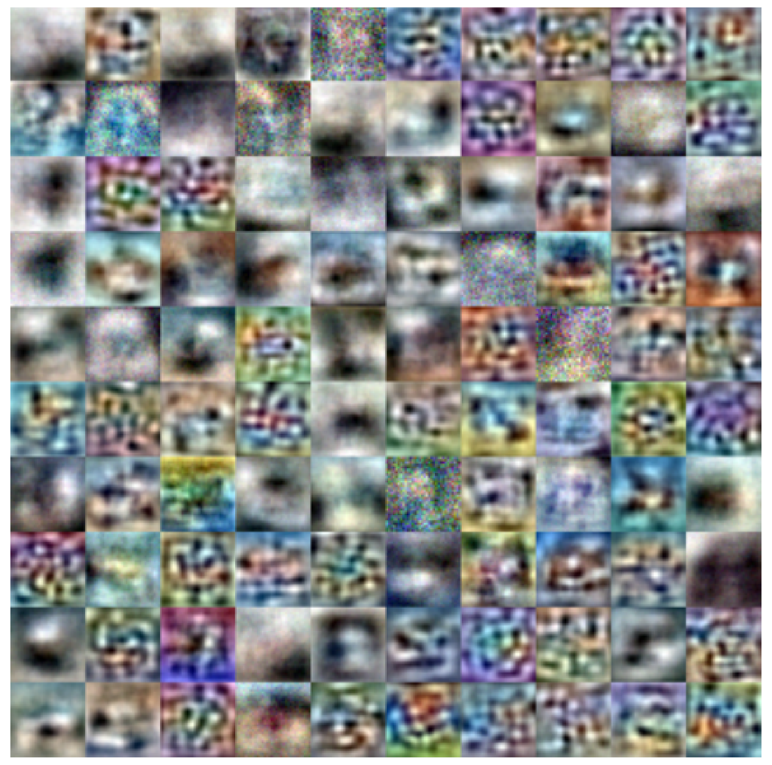
We implemented the RBM training algorithm both using numpy and using tensorflow, we applied it to the MNIST dataset.





Results On CIFAR data

We implemented the RBM training algorithm both using numpy and using tensorflow, we applied it to the MNIST dataset.



References

[1] Asja Fischer and Christian Igel. Training restricted boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39, 2014.

[2] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

[3] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM, 2008.