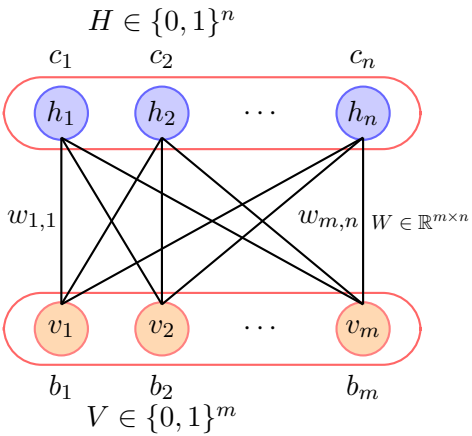
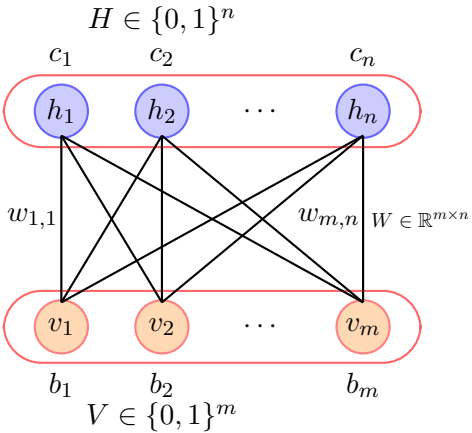
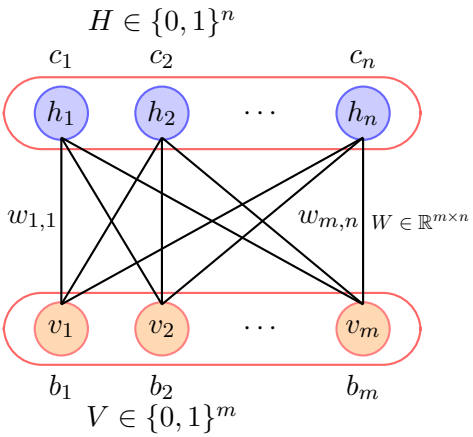
Module 19.5: Unsupervised Learning with RBMs



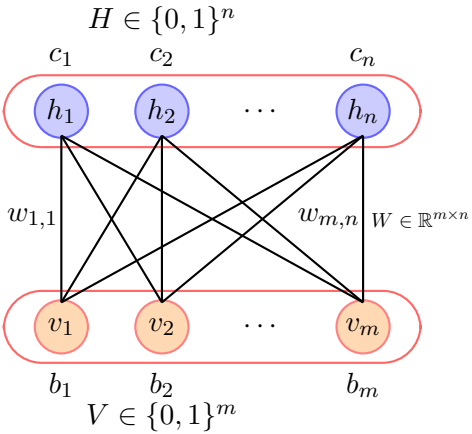
- So far, we have mainly dealt with supervised learning where we are given $\{x_i, y_i\}_{i=1}^n$ for training



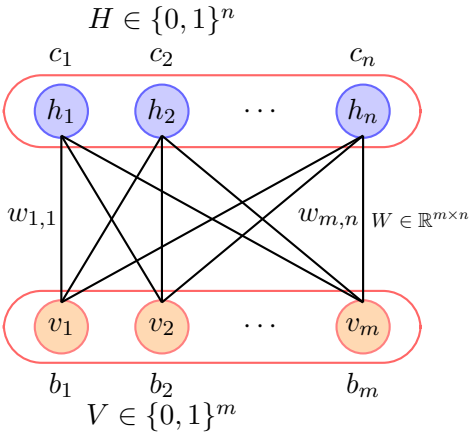
- So far, we have mainly dealt with supervised learning where we are given $\{x_i, y_i\}_{i=1}^n$ for training
- In other words, for every training example we are given a label (or class) associated with it



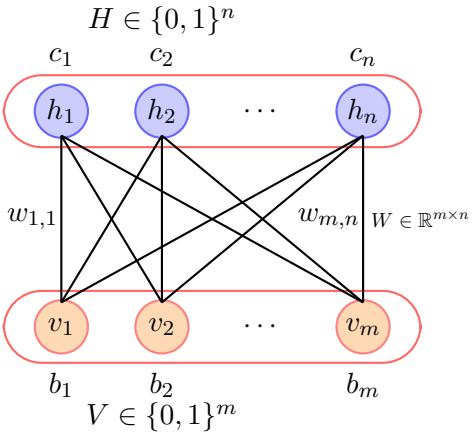
- So far, we have mainly dealt with supervised learning where we are given $\{x_i, y_i\}_{i=1}^n$ for training
- In other words, for every training example we are given a label (or class) associated with it
- Our job was then to learn a model which predicts $\hat{y}$ such that the difference between y and $\hat{y}$ is minimized



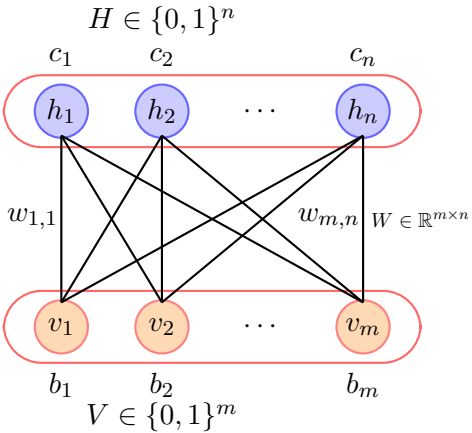
- But in the case of RBMs, our training data only contains x (for example, images)



- But in the case of RBMs, our training data only contains x (for example, images)
- There is no explicit label (y) associated with the input



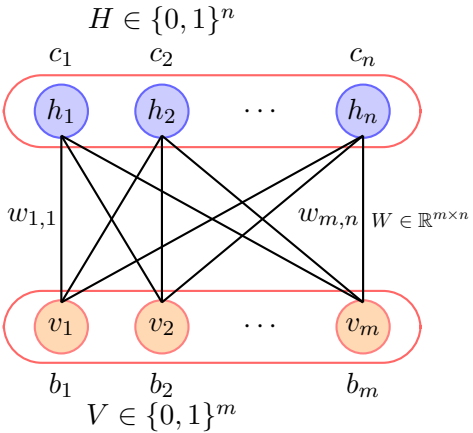
- But in the case of RBMs, our training data only contains x (for example, images)
- There is no explicit label (y) associated with the input
- Of course, in addition to x we have the latent variable h but we don't know what these h's are

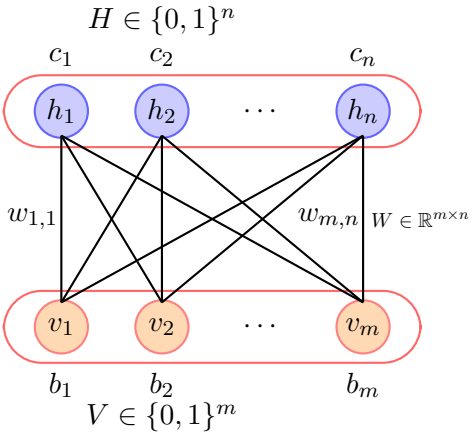


- But in the case of RBMs, our training data only contains x (for example, images)
- There is no explicit label (y) associated with the input
- Of course, in addition to x we have the latent variable h but we don't know what these h's are
- We are interested in learning $P(x, h)$ which we have parameterized as

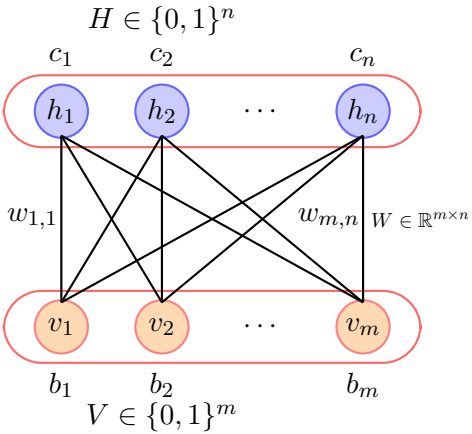
$$P(V, H) = \frac{1}{Z} e^{-(-\sum_i \sum_j w_{ij} v_i h_j - \sum_i b_i v_i - \sum_j c_j h_j)}$$

- What is the objective function that we should use?

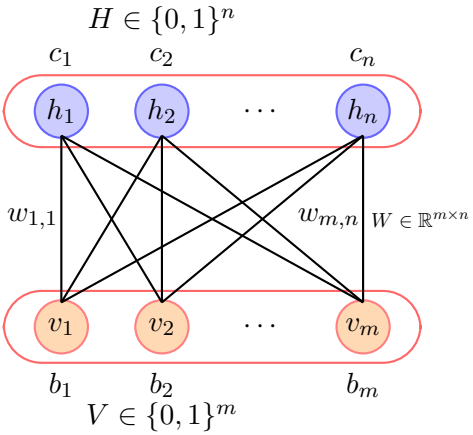




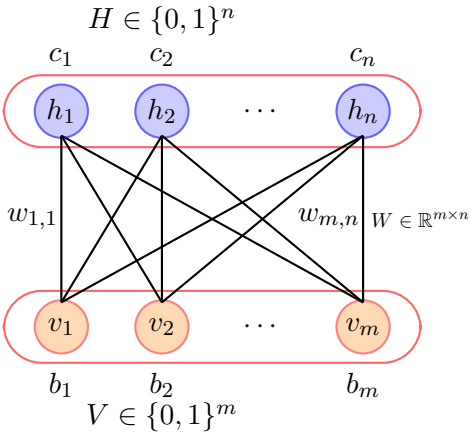
- What is the objective function that we should use?
- First note that if we have learnt $P(x, h)$ we can compute $P(x)$



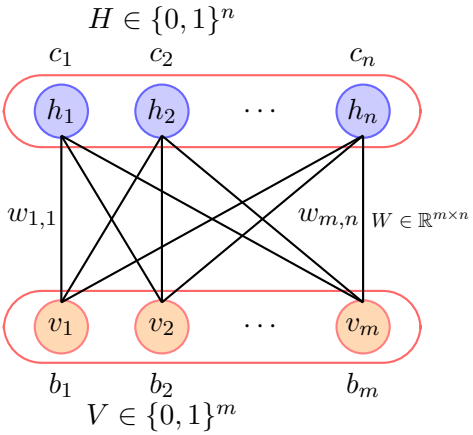
- What is the objective function that we should use?
- First note that if we have learnt $P(x, h)$ we can compute $P(x)$
- What would we want $P(X = x)$ to be for any x belonging to our training data?



- What is the objective function that we should use?
- First note that if we have learnt $P(x, h)$ we can compute $P(x)$
- What would we want $P(X = x)$ to be for any x belonging to our training data?
- We would want it to be high

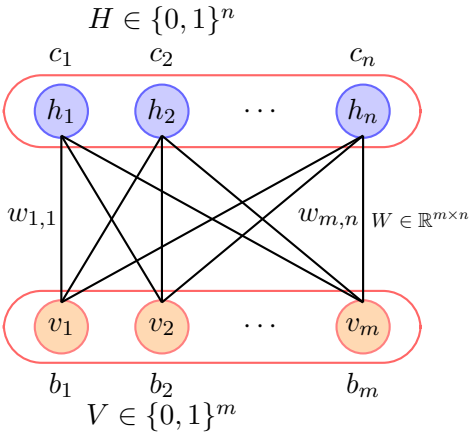


- What is the objective function that we should use?
- First note that if we have learnt $P(x, h)$ we can compute $P(x)$
- What would we want $P(X = x)$ to be for any x belonging to our training data?
- We would want it to be high
- So now can you think of an objective function



- What is the objective function that we should use?
- First note that if we have learnt $P(x, h)$ we can compute $P(x)$
- What would we want $P(X = x)$ to be for any x belonging to our training data?
- We would want it to be high
- So now can you think of an objective function

$$maximize \prod_{i=1}^{N} P(X = x_i)$$

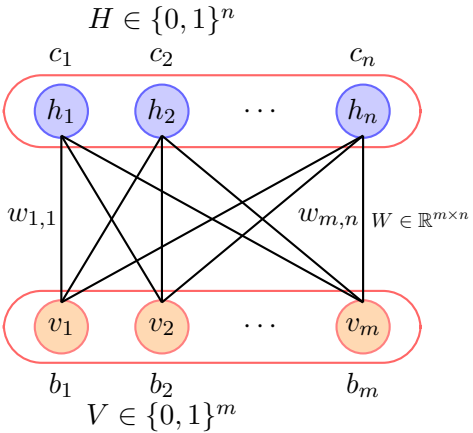


- What is the objective function that we should use?
- First note that if we have learnt $P(x, h)$ we can compute $P(x)$
- What would we want $P(X = x)$ to be for any x belonging to our training data?
- We would want it to be high
- So now can you think of an objective function

$$maximize \prod_{i=1}^{N} P(X = x_i)$$

- Or, log-likelihood

$$\ln \mathscr{L}(\theta) = \ln \prod_{i=1}^{l} p(x_i|\theta) = \sum_{i=1}^{l} \ln p(x_i|\theta)$$



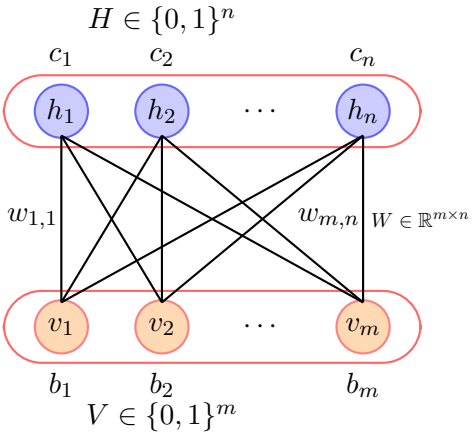
- What is the objective function that we should use?
- First note that if we have learnt $P(x, h)$ we can compute $P(x)$
- What would we want $P(X = x)$ to be for any x belonging to our training data?
- We would want it to be high
- So now can you think of an objective function

$$maximize \prod_{i=1}^{N} P(X = x_i)$$

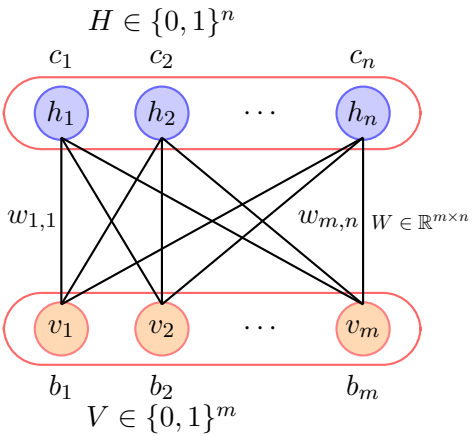
- Or, log-likelihood

$$\ln \mathscr{L}(\theta) = \ln \prod_{i=1}^{l} p(x_i|\theta) = \sum_{i=1}^{l} \ln p(x_i|\theta)$$

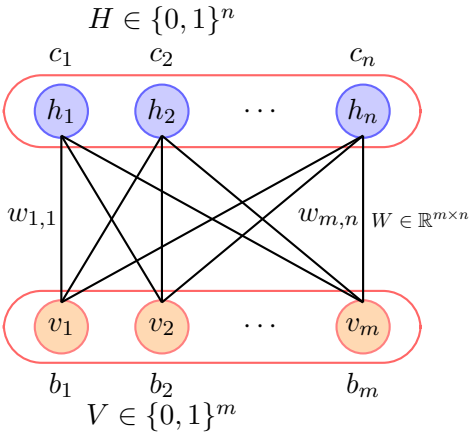
where θ are the parameters



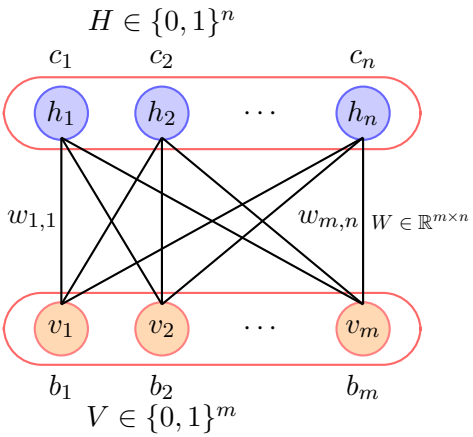
- Okay so we have the objective function now! What next?



- Okay so we have the objective function now! What next?
- We need a learning algorithm



- Okay so we have the objective function now! What next?
- We need a learning algorithm
- We can just use gradient descent if we are able to compute the gradient of the loss function w.r.t. the parameters



- Okay so we have the objective function now! What next?
- We need a learning algorithm
- We can just use gradient descent if we are able to compute the gradient of the loss function w.r.t. the parameters
- Let us see if we can do that