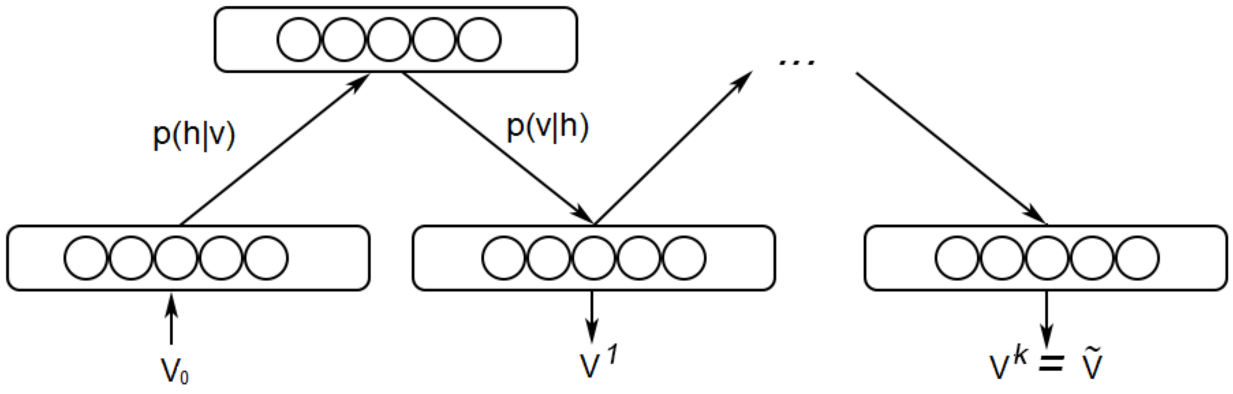
Problem Statement

As you already know, you have to perform Topic Modelling on a set of documents and print the distribution of words in each topic.

You have seen that the training process is contrastive divergence. In this assignment, we shall create a function to perform Gibbs sampling k times within contrastive divergence often referred to as CD-k. Let's understand the structure of the notebook and the CD-k training process:

- Importing the dataset.
- Create a bag of words model.
  Note that the shape of a bag of words model will be (data_size, vocablury_size)
  Please **note** that the above shape might vary with the way you perform bag of words model but the index *data_size* will be the same.
- Training using CD-k
  This involves performing Gibbs sampling k-times which can be better understood using the following image.



Contrastive Divergence

1. You start with the input batch of data, v_0.
2. You then calculate $p(h|v_0)=\sigma(c_j+\sum_{i=1}^{n} v_i w_{ij})$ as seen in the lecture.
   Vectorized implementation: $p(h|v_0)=\sigma(C+V.W)$
3. Using this $p(h|v_0)$, you sample h_0.
4. Now that you have got h_0, you calculate $p(v|h_0)=\sigma(b_j+\sum_{j=1}^{m} h_j w_{ij})$ as seen in the lecture.
   Vectorized implementation: $p(v|h_0)=\sigma(B+H.W_T)$
5. Using this $p(v|h_0)$, you sample v_1.

You repeat this k times till you get h_k and v_k.

Step 2 and 3 are performed in the function *sampleHiddenLayer()* while Step 4 and 5 are performed in *sampleVisibleLayer()*.

sampleHiddenLayer() and *sampleVisibleLayer()* are combined to create a function *gibbs()* which does one iteration of Gibbs sampling.

gibbs is repeated k-times in the function *cd_k()* to perform Contrastive Divergence k-times.

You have already learned that the training process in an RBM involves maximizing the joint probability distribution. Using the energy function as defined in the lectures and the above sampling process, the update matrices and vectors simplify as follows:

6. $\Delta W = v_0 \otimes p(h_0|v_0) - v_k \otimes p(h_k|v_k)$
7. $\Delta b = avg\_across\_batch(v_o - v_k)$
8. $\Delta c = avg\_across\_batch(p(h_0|v_o) - p(h_k|v_k))$

You do average across batch because you need a vector update for the bias vectors and v_o, v_k and $p(h_0|v_o), p(h_k|v_k)$ are matrices.

Note that the exact derivations are not covered as it is very complex and the Prof. has given the intuition about the training process in the previous segments.

Since you have to maximize the joint probability distribution, you use **gradient ascent** here with momentum. It is recommended that you understand the working of momentum from here.

The momentum equations are as follows:

9. $mW_t = \gamma\ mW_{t-1} - \Delta W$
10. $mb_t = \gamma\ mb_{t-1} - \Delta b$
11. $mc_t = \gamma\ mc_{t-1} - \Delta c$

γ is the momentum coefficient here.

Using these momentum terms, you update the weights and biases as follows:

12. $W_t = W_{t-1} + \alpha\ mW_t$

13. $b_t = b_{t-1} + \alpha \, mb_t$
14. $c_t = c_{t-1} + \alpha \, mc_t$

α is the learning rate.

The above equations are implemented in the function *train()*.

These equations should help you in implementing topic modelling using RBMs without any issue.

In the end, you'll be able to see the words that constitute the different topics.

All the best!