**Definitions**

DBN: A DBN is a feed forward neural network with a deep architecture, i.e. with many hidden layers.

RBM: An RBM is a two layer recurrent neural network in which stochastic binary inputs are connected to stochastic binary outputs using symmetrical weighted connections.

Corpus: the entire collection tweets.

Vocabulary: The number of distinct words of the corpus.

**Feature representation DBN**

* Each message is represented as a vector with length equal to the number of distinct words in the corpus (i.e. |vocabulary|).
* The value of the vector’s component is the frequency of the corresponding word.

**Predicting Method:**

* We use DBN to detect spammers in twitter
* To initialize the weights of this deep neural network, we use RBM.

**Feature Representation RBM:**

* Trained using binary vectors, indicating the presence or absence of a word in a massage.

**RBM Training:**

* It is unsupervised.
* Given a training sample we propagate it stochastically through the RBM.
* The outputs of the hidden units follow the conditional distribution. We then sample from this distribution to produce a binary vector. This vector is propagated in the opposite direction through the RBM which results in a ‘confabulation’ (reconstruction) of the original input data.
* Finally, the state of the hidden units is updated by propagating this confabulation through the RBM.
* This above procedure is performed repeatedly for all the examples of the training set.
* We repeat this method for epoch=20
* We train as many RBMs as the number of hidden layers in the DBN. Which our case is 3 (L=3).

**DBN Construction:**

* The RBMs are placed one top of another resulting in a DBN without the output layer.
* The activations of these RBMs’ hidden units are driven by data, not confabulations.
* number of input units = |vocabulary|
* We use 3 hidden layers of 50-50-200 structure
* Output layer has two units, one for each class, whose weights are randomly initialized, and the DBN is fine tuned with respect to a typical supervise criterion ( such as mean square error or cross-entropy)

**Data Preprocessing:**

1. Filter non-english tweets
2. Filter users with less than two tweets
3. Remove URLs, hashtags and @user
4. remove words appearing in less than two tweets
5. remove stop words
6. Perform stemming for all the words.
7. We retain the top k words with respect to information gain score. (k=1500 and k=1000 used)

**Experiment**

* 10 fold cross validation.
* Randomly split the corpus to form 10 partitions in such a way that the spam ratio of the original corpus is retained in each partition.

**Performance measure**

* Accuracy
* Spam Recall
* Ham recall
* Spam Precision
* Ham precision