# RESTRICTED BOLTZMANN MACHINE (RBM)

## Module 4

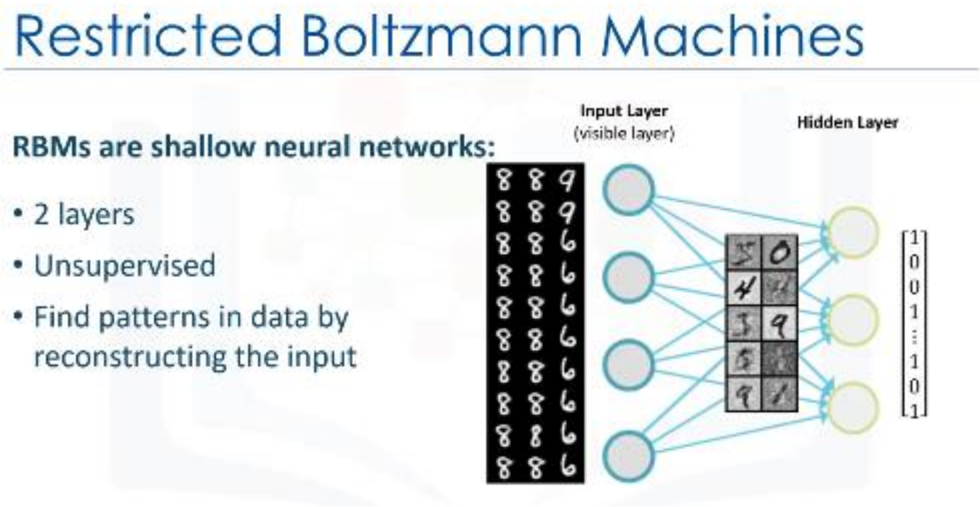
v-cardona
Deep Learning with Tensorflow

# Contenido

## Intro to RBMs

It is a model for solving collaborative filtering, which is a type of recommender system engine.

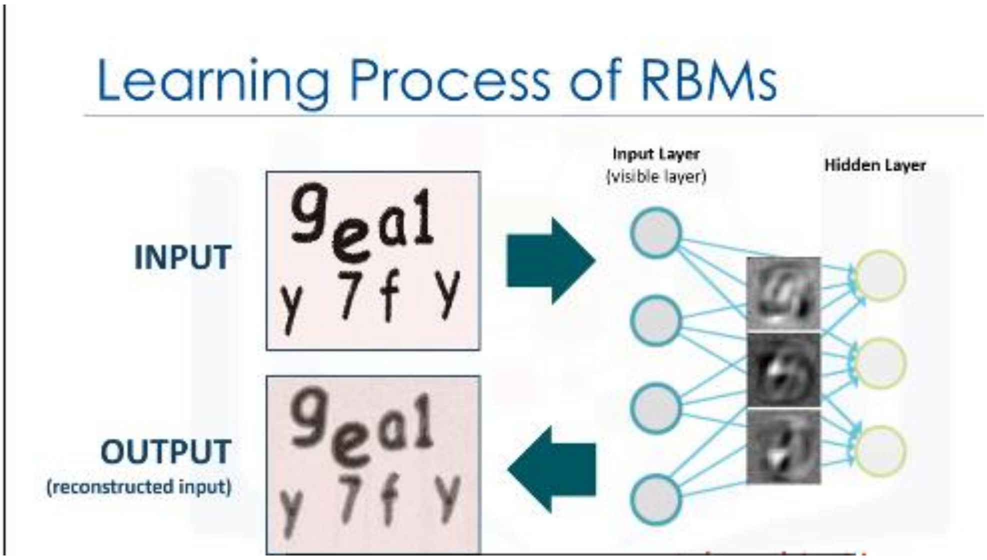
Restricted Boltzmann Machines (or RBMs, for short), are shallow neural networks that only have two layers. They are an unsupervised method used to find patterns in data by reconstructing the input. The first layer of the RBM is called the visible layer, and the second layer is the hidden layer. We say that they are "restricted" because neurons within the same layer are not connected. Feeding the input data, the network learns its weights. Then, feeding an input image, the values that appear in the hidden layer can be considered as features learned automatically from the input data. And, as there are a smaller number of units in the hidden units of an RBM, we can tell that the values in the hidden units are a good representation of data that are lower in dimensionality when compared to the original data.




Applications: dimensionality reduction, feature extraction, and collaborative filtering, RBM used as the main block of another type of Deep neural network (Deep belief networks), …

## Restricted Boltzmann machine

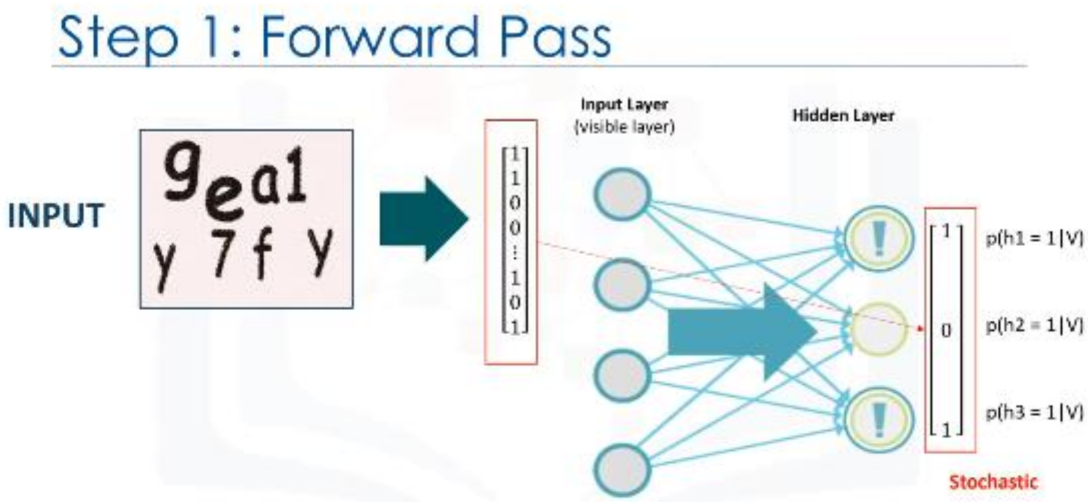
RBMs learn patterns and extract important features in data by reconstructing the input. The learning process consists of several forward and backward passes, where the RBM tries to reconstruct the input data. The weights of the neural net are adjusted in such a way that the RBM can find the relationships among input features, and then determines which features are relevant. After training is complete, the net is able to reconstruct the input based on what it learned.




The RBM can automatically extract meaningful features from a given input in the training process. In fact, a trained RBM can reveal which features are the most important ones when detecting patterns. It can also represent each image with some hidden values, also referred to, as latent values.

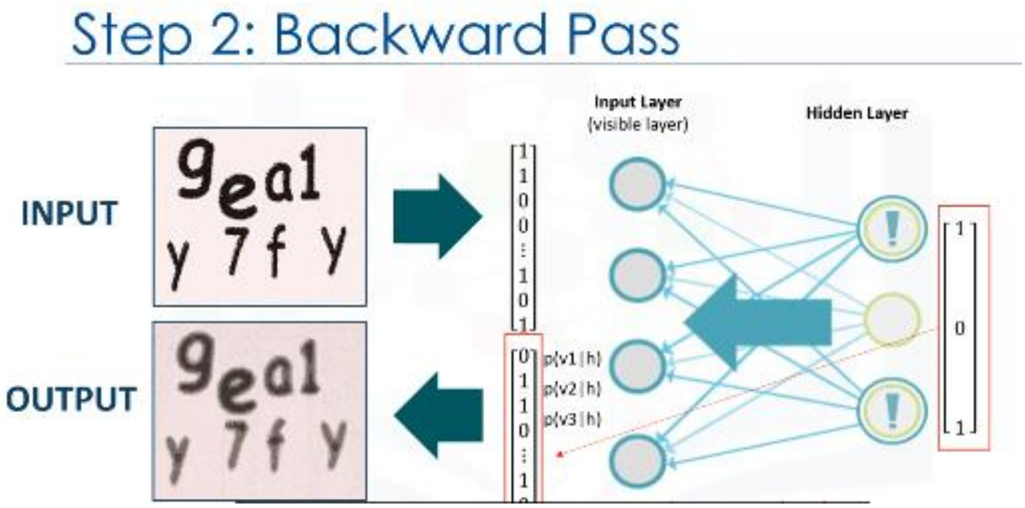
Training process:

1. The first step is the forward pass. In the forward pass, the input image is converted to binary values, and then, the vector input is fed into the network, where its values are multiplied by weights, and an overall bias, in each hidden unit. Then, the result goes to an activation function, such as the sigmoid function, which represents the probability of turning each individual hidden-unit on, or in other words, the probability of the node activation. Then, a sample is drawn from this probability distribution, and it finds which neurons may or may not activate. This means, it makes stochastic decisions about whether or not to transmit that hidden data.

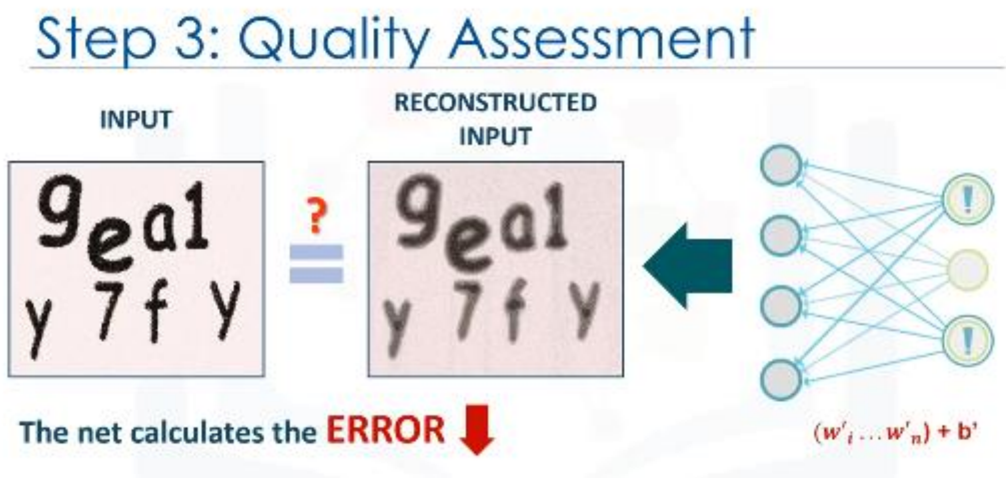



The intuition behind the sampling, is that there are some random hidden variables, and by sampling from the hidden layer, you can reproduce sample variants encountered during training. So, as you can see, the forward pass translates the inputs into a set of binary values that get represented in the hidden layer.

2. Then we get to step 2: the backward pass. In the backward pass, the activated neurons in the hidden layer send the results back to the visible layer, where the input will be reconstructed. During this step, the data that is passed backwards is also combined with the same weights and overall bias that were used in the forward pass. So, once the information gets to the visible layer, it is in the shape of the probability distribution of the input values, given the hidden values. And sampling the distribution, the input is reconstructed. So, as you can see, the Backward pass is about making guesses about the probability distribution of the original input.




3. Step 3 consists of assessing the quality of the reconstruction by comparing it to the original data. The RBM then calculates the error and adjusts the weights and bias in order to minimize it. That is, in each epoch, we compute the "error" as a sum of the squared difference between step 1 and the next step. These 3 steps are repeated until the error is deemed sufficiently low.



Advantages:

- They excel when working with unlabelled data. Many important real-world datasets are unlabelled, like videos, photos, and audio files, so RBMs provide a lot of benefit in these types of unsupervised learning problems.
- During the learning process, the RBM extracts features from the input data, decides which features are relevant, and how to best combine them to form patterns.
- RBMs are also generally more efficient at dimensionality reduction when compared to principal component analysis, which is considered a popular alternative.
- RBMs learn from the data, they actually encode their own structure.

Last pro is why they're grouped into a larger family of models known as the Autoencoders. However, Restricted Boltzmann Machines differ from Autoencoders in that they use a stochastic approach, rather than a deterministic approach.