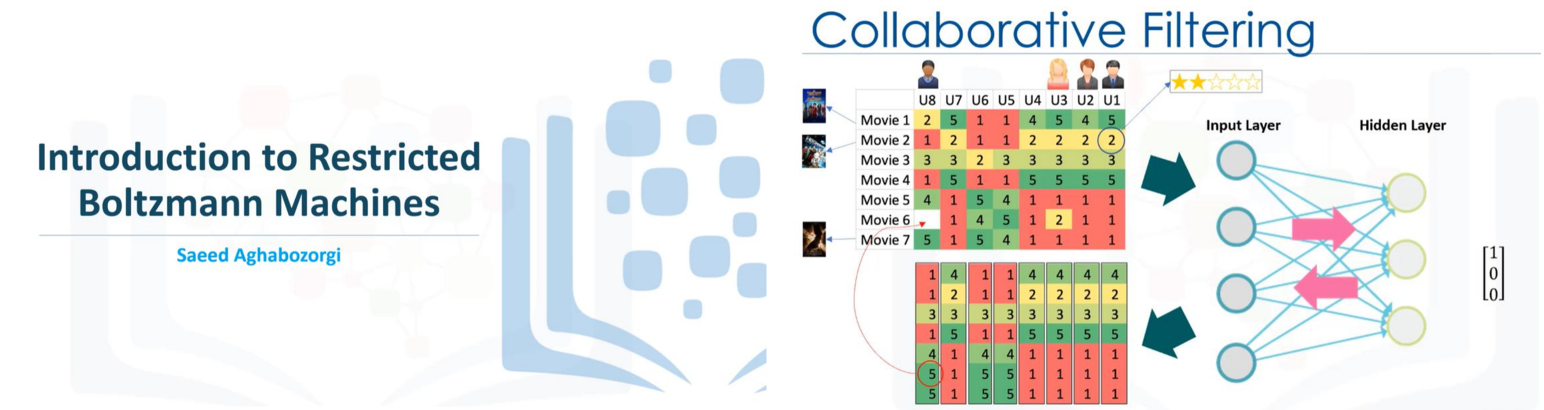
# Module 4 - Restricted Boltzmann Machines (RBMs)

## Learning Objectives

In this lesson you will learn about:

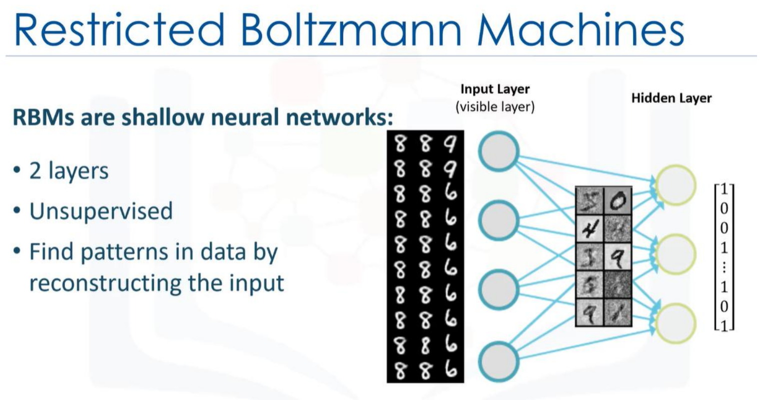
- The Applications of Unsupervised Learning
- Restricted Boltzmann Machine
- Training a Restricted Boltzmann Machine
- Recommendation System with a Restricted Boltzmann Machine

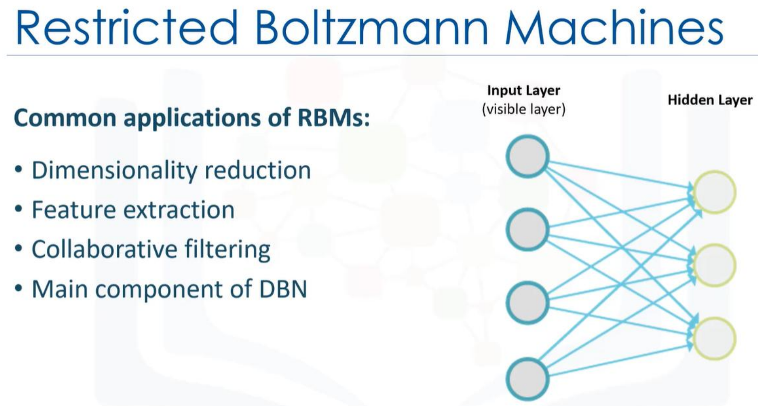
## 4.1 Intro to RBMs




Hello, and welcome! In this video we'll be providing an introductionto **Restricted Boltzmann Machines**. So let's get started! Imagine that we have access to a matrix of the viewer ratings of a certain number of movies on Netflix, where each row corresponds to a movie, and each column corresponds to a user's rating. For the sake of simplicity, let's say we're only examining 8 users and their ratings of 7 movies. The value in each cell shows the score that the user has given to the movie after watching it, and is based on a rating scale of 1 to 5. Also, imagine that we have a type of neural network, that has only 2 layers, the input layer and the hidden layer. Let's also assume that this network has learned in such a way that it can reconstruct the input vectors. For example, when you feed the first user vector into the network, it goes through the network, and finally fires up some units in the hidden layer. Then, the values of the hidden layer will be fed back into the network, and a vector, which is almost the same as the input vector, is reconstructed as output. We can think of it as making guesses about the input data. You feed the second user's ratings, which are not very different from first user, and thus, the same hidden units will be turned on, and the network output would be the same as the first reconstructed vector. We can repeat it for the third user. And for user number 4. Now, let's feed a user that has a completely different idea about these movies. As you can see, this particular user was not a fan of the first movie. When we feed the respective rating values into the network, different hidden units get turned on, and a different vector is reconstructed, which is almost the same as User number 5's preferences. It is the same for user number 6. And the process can be repeated for the other users.
Now, let's look at User number 8. He hasn't watched movie 6, but does have some preferences that are almost the same as Users 5 and 6, right? Let's feed this vector into our network. It'll fire up the same hidden units as Users 5 and 6. And, feeding back these values, we'll reconstruct a new vector. Look at the expected value for movie 6 in the reconstructed vector. Using this value, it's not hard to imagine that user number 8 might be interested in this movie, even though he hasn't watched it yet. Maybe we can even recommend it to him, yes? In fact, it is a way of solving **collaborative filtering**, which is a type of recommender system engine. And the network that can make such a model is called a Restricted Boltzmann Machine.

**总结:**

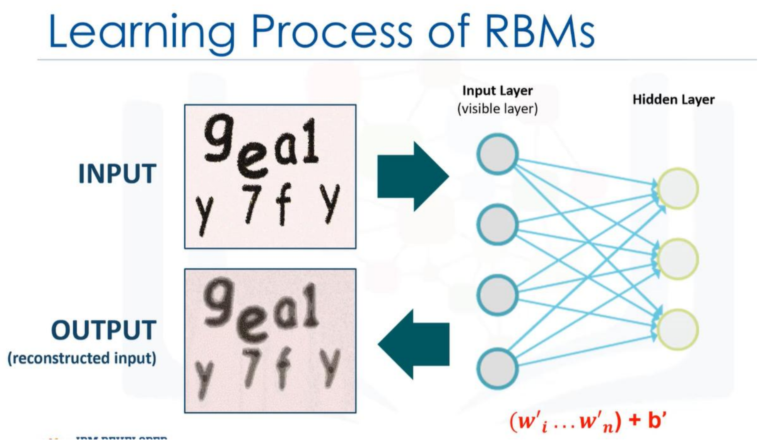


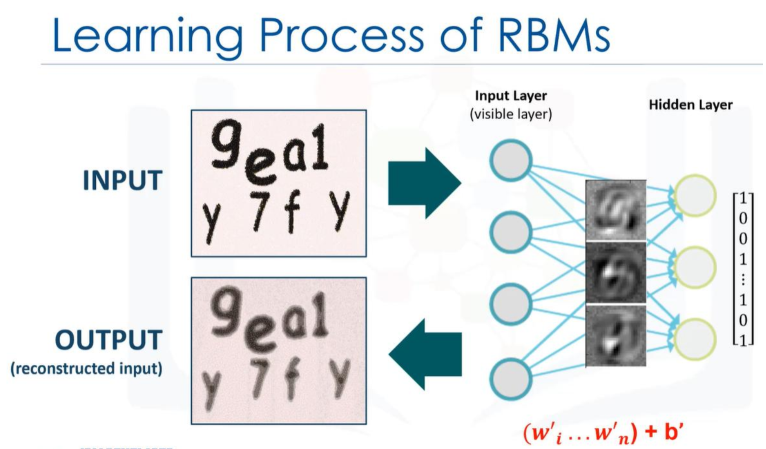




Restricted Boltzmann Machines (or **RBMs**, for short), are shallow neural networks that only have two layers. They are an unsupervised method used to find patterns in data by reconstructing the input. The first layer of the RBM is called the **visible layer**, and the second layer is the **hidden layer**. 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.

Restricted Boltzmann Machines are useful in many applications like dimensionality reduction,feature extraction, and collaborative filtering, just to name a few. On top of that, RBMs are used as the main block of another type of Deep Neural Network, which is called, Deep Belief Networks, which we'll be talking about later. By now, you should have basic knowledge about RBMs and their applications.

## 4.2 Training RBMs

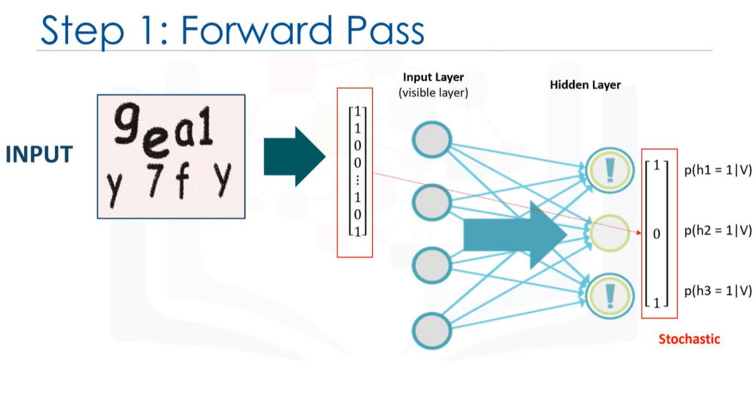


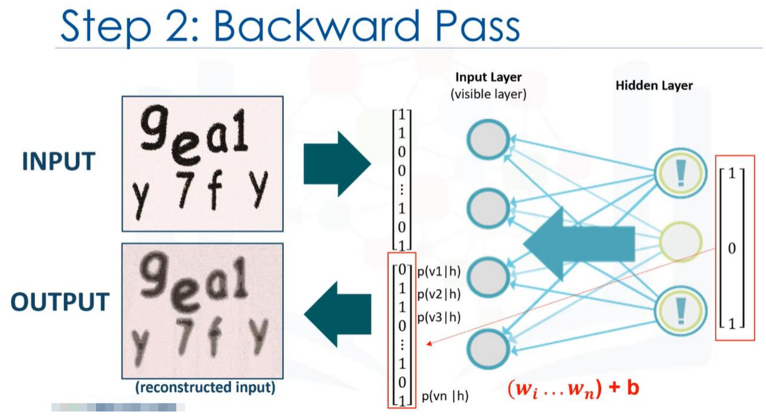




Hello, and welcome! In this video we will provide an overviewof RBMs, with a focus on understanding how they work. Let's get started Let's say that we provide an image as input to an RBM. The pixels are processed by the input layer, which is also known as the visible layer. RBMs learn patterns and extract important features in data by reconstructing the input. So, 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 reconstructed image, here, is only a representation of what happens. What's important to take from this example, though, is that the RBM can automatically extract meaningful features from a given input in the training process.

In fact, a trained RBM can reveal which featuresare the most important ones when detecting patterns. It can also represent each image with some hidden values, also referred to, as latent values.

**总结:**

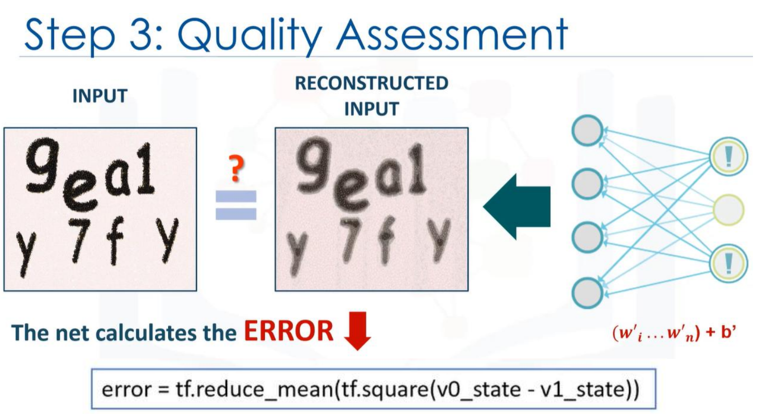


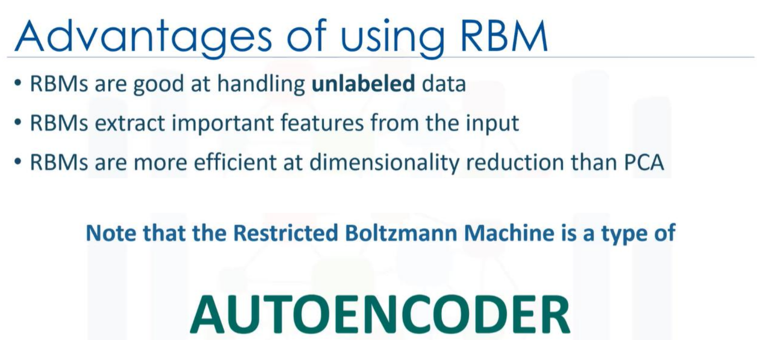




Now let's look at an RBM's training process,in which, three major steps are repeated.

- 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.

- 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.




Now, let's look at step 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.

Now, let's touch on a few reasons why RBMs are such a great tool.

- A big advantage of RBMS is that they excel when working with unlabeled data. Many important real-world datasets are unlabeled, like videos, photos, and audio files, so RBMs provide a lot of benefit in these types of unsupervised learning problems.
- Another advantage is that 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.
- Finally, RBMs learn from the data, they actually encode their own structure. This 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.

**总结：**

What is the main application of RBM?

- ○ Data dimensionality reduction
- ○ Feature extraction
- ○ Collaborative filtering
- ◉ All of the above

✔

How does an RBM compare to a PCA?

- ○ RBM cannot reduce dimensionality
- ○ PCA cannot generate original data
- ○ PCA is another type of Neural Network
- ◉ Both can regenerate input data
- ○ All of the above

✔

Which statement is TRUE about RBM?

- ○ It is a Boltzmann machine, but with no connections between nodes in the same layer
- ○ Each node in the first layer has a bias
- ○ The RBM reconstructs data by making several forward and backward passes between the visible and hidden layers
- ○ At the hidden layer's nodes, X is multiplied by a W (weight matrix) and added to h_bias
- ◉ All of the above

✔

Which statement is TRUE statement about an RBM?

- ○ The objective function is to maximize the likelihood of our data being drawn from the reconstructed data distribution
- ○ The Negative phase of an RBM decreases the probability of samples generated by the model
- ○ Contrastive Divergence (CD) is used to approximate the negative phase of an RBM
- ○ The Positive phase of an RBM increases the probability of training data
- ◉ All of the above

✔

**总结:**