

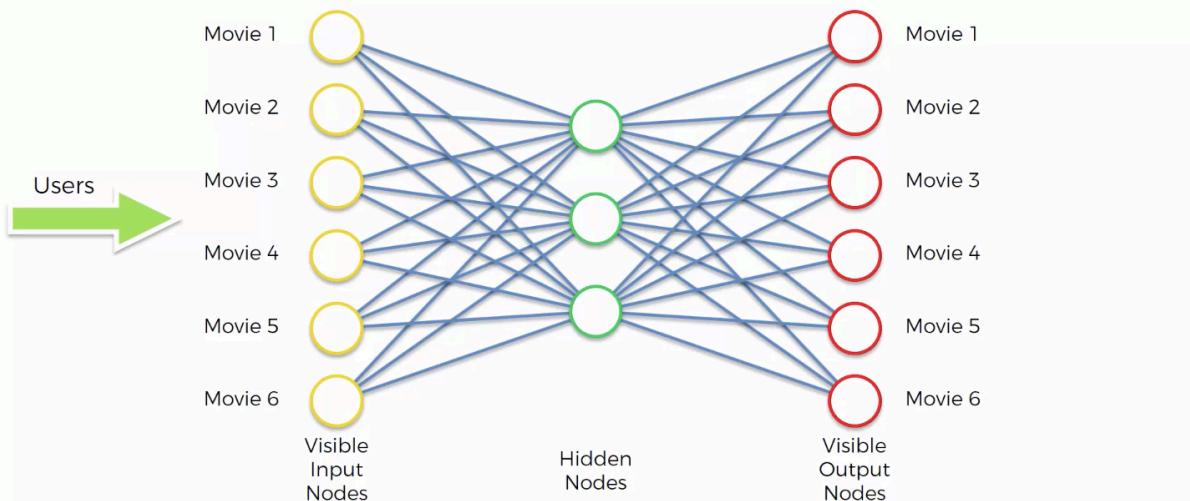
Training an auto encoder

Training an Auto Encoder

Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder



Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 1: We start with an array where the lines (the observations) correspond to the users and the columns (the features) correspond to the movies. Each cell (u, i) contains the rating (from 1 to 5, 0 if no rating) of the movie i by the user u .

Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 1: We start with an array where the lines (the observations) correspond to the users and the columns (the features) correspond to the movies. Each cell (u, i) contains the rating (from 1 to 5, 0 if no rating) of the movie i by the user u .



STEP 2: The first user goes into the network. The input vector $x = (r_1, r_2, \dots, r_m)$ contains all its ratings for all the movies.

Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 1: We start with an array where the lines (the observations) correspond to the users and the columns (the features) correspond to the movies. Each cell (u, i) contains the rating (from 1 to 5, 0 if no rating) of the movie i by the user u .



STEP 2: The first user goes into the network. The input vector $x = (r_1, r_2, \dots, r_m)$ contains all its ratings for all the movies.



STEP 3: The input vector x is encoded into a vector z of lower dimensions by a mapping function f (e.g: sigmoid function):

$$z = f(Wx + b) \text{ where } W \text{ is the vector of input weights and } b \text{ the bias}$$

Training an Auto Encoder

STEP 1: We start with an array where the lines (the observations) correspond to the users and the columns (the features) correspond to the movies. Each cell (u, i) contains the rating (from 1 to 5, 0 if no rating) of the movie i by the user u .



STEP 2: The first user goes into the network. The input vector $x = (r_1, r_2, \dots, r_m)$ contains all its ratings for all the movies.



STEP 3: The input vector x is encoded into a vector z of lower dimensions by a mapping function f (e.g: sigmoid function):

$$z = f(Wx + b) \text{ where } W \text{ is the vector of input weights and } b \text{ the bias}$$



STEP 4: z is then decoded into the output vector y of same dimensions as x , aiming to replicate the input vector x .

Training an Auto Encoder

STEP 1: We start with an array where the lines (the observations) correspond to the users and the columns (the features) correspond to the movies. Each cell (u, i) contains the rating (from 1 to 5, 0 if no rating) of the movie i by the user u .



STEP 2: The first user goes into the network. The input vector $x = (r_1, r_2, \dots, r_m)$ contains all its ratings for all the movies.



STEP 3: The input vector x is encoded into a vector z of lower dimensions by a mapping function f (e.g: sigmoid function):

$$z = f(Wx + b) \text{ where } W \text{ is the vector of input weights and } b \text{ the bias}$$



STEP 4: z is then decoded into the output vector y of same dimensions as x , aiming to replicate the input vector x .



STEP 5: The reconstruction error $d(x, y) = \|x-y\|$ is computed. The goal is to minimize it.

Training an Auto Encoder

STEP 1: We start with an array where the lines (the observations) correspond to the users and the columns (the features) correspond to the movies. Each cell (u, i) contains the rating (from 1 to 5, 0 if no rating) of the movie i by the user u .



STEP 2: The first user goes into the network. The input vector $x = (r_1, r_2, \dots, r_m)$ contains all its ratings for all the movies.



STEP 3: The input vector x is encoded into a vector z of lower dimensions by a mapping function f (e.g: sigmoid function):

$$z = f(Wx + b) \text{ where } W \text{ is the vector of input weights and } b \text{ the bias}$$



STEP 4: z is then decoded into the output vector y of same dimensions as x , aiming to replicate the input vector x .



STEP 5: The reconstruction error $d(x, y) = \|x-y\|$ is computed. The goal is to minimize it.



STEP 6: Back-Propagation: from right to left, the error is back-propagated. The weights are updated according to how much they are responsible for the error. The learning rate decides by how much we update the weights.

Training an Auto Encoder

STEP 1: We start with an array where the lines (the observations) correspond to the users and the columns (the features) correspond to the movies. Each cell (u, i) contains the rating (from 1 to 5, 0 if no rating) of the movie i by the user u .



STEP 2: The first user goes into the network. The input vector $x = (r_1, r_2, \dots, r_m)$ contains all its ratings for all the movies.



STEP 3: The input vector x is encoded into a vector z of lower dimensions by a mapping function f (e.g: sigmoid function):

$$z = f(Wx + b) \text{ where } W \text{ is the vector of input weights and } b \text{ the bias}$$



STEP 4: z is then decoded into the output vector y of same dimensions as x , aiming to replicate the input vector x .



STEP 5: The reconstruction error $d(x, y) = \|x-y\|$ is computed. The goal is to minimize it.



STEP 6: Back-Propagation: from right to left, the error is back-propagated. The weights are updated according to how much they are responsible for the error. The learning rate decides by how much we update the weights.



STEP 7: Repeat Steps 1 to 6 and update the weights after each observation (Reinforcement Learning). Or: Repeat Steps 1 to 6 but update the weights only after a batch of observations (Batch Learning).

Training an Auto Encoder

STEP 1: We start with an array where the lines (the observations) correspond to the users and the columns (the features) correspond to the movies. Each cell (u, i) contains the rating (from 1 to 5, 0 if no rating) of the movie i by the user u .



STEP 2: The first user goes into the network. The input vector $x = (r_1, r_2, \dots, r_m)$ contains all its ratings for all the movies.



STEP 3: The input vector x is encoded into a vector z of lower dimensions by a mapping function f (e.g: sigmoid function):

$$z = f(Wx + b) \text{ where } W \text{ is the vector of input weights and } b \text{ the bias}$$



STEP 4: z is then decoded into the output vector y of same dimensions as x , aiming to replicate the input vector x .



STEP 5: The reconstruction error $d(x, y) = \|x-y\|$ is computed. The goal is to minimize it.



STEP 6: Back-Propagation: from right to left, the error is back-propagated. The weights are updated according to how much they are responsible for the error. The learning rate decides by how much we update the weights.



STEP 7: Repeat Steps 1 to 6 and update the weights after each observation (Reinforcement Learning). Or: Repeat Steps 1 to 6 but update the weights only after a batch of observations (Batch Learning).



STEP 8: When the whole training set passed through the ANN, that makes an epoch. Redo more epochs.

Training an Auto Encoder

STEP 1

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0		1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	
User 4	1	0	1	1	0	1
User 5	0		1	1		1
User 6	0	0	0	0	1	
User 7	1	0	1	1	0	1
User 8	0	1	1		0	1
User 9	0	1	1	1	1	
User 10	1		0	0		0
User 11	0	1	1	1	0	1

Training an Auto Encoder

STEP 2

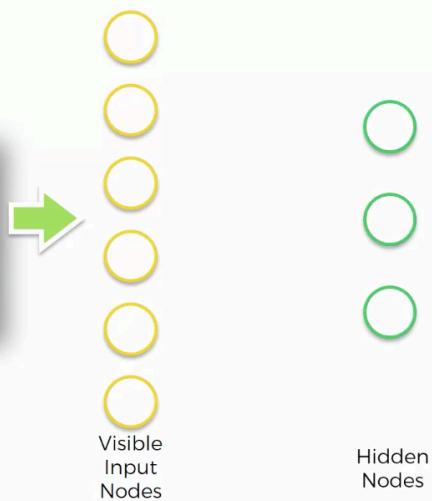
	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0		1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	
User 4	1	0	1	1	0	1
User 5	0		1	1		1
User 6	0	0	0	0	1	
User 7	1	0	1	1	0	1
User 8	0	1	1		0	1
User 9	0	1	1	1	1	
User 10	1		0	0		0
User 11	0	1	1	1	0	1



Training an Auto Encoder

STEP 3

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0		1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	
User 4	1	0	1	1	0	1
User 5	0		1	1		1
User 6	0	0	0	0	1	
User 7	1	0	1	1	0	1
User 8	0	1	1		0	1
User 9	0		1	1	1	1
User 10	1		0	0		0
User 11	0	1	1	1	0	1



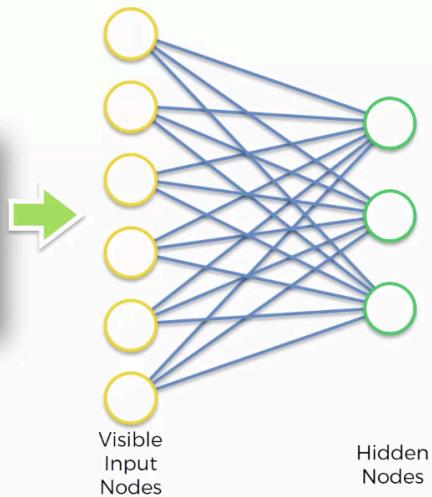
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 3

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0		1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	
User 4	1	0	1	1	0	1
User 5	0		1	1		1
User 6	0	0	0	0	1	
User 7	1	0	1	1	0	1
User 8	0	1	1		0	1
User 9	0		1	1	1	1
User 10	1		0	0		0
User 11	0	1	1	1	0	1



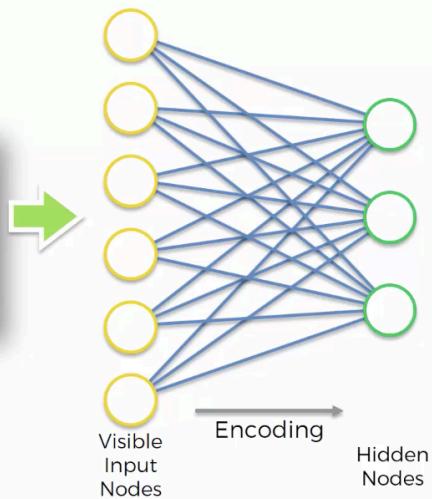
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 3

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0	0	1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	0
User 4	1	0	1	1	0	1
User 5	0	0	1	1	0	1
User 6	0	0	0	0	1	0
User 7	1	0	1	1	0	1
User 8	0	1	1	0	0	1
User 9	0	1	1	1	1	1
User 10	1	0	0	0	0	0
User 11	0	1	1	1	0	1



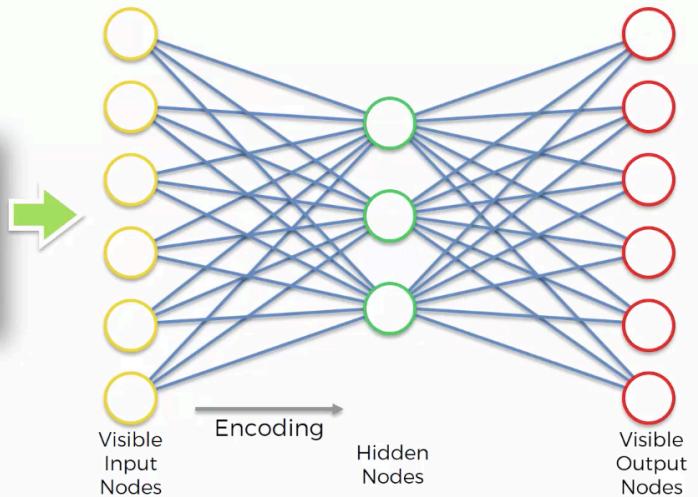
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 4

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0	0	1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	0
User 4	1	0	1	1	0	1
User 5	0	1	1	1	0	1
User 6	0	0	0	0	1	0
User 7	1	0	1	1	0	1
User 8	0	1	1	0	0	1
User 9	0	1	1	1	1	1
User 10	1	0	0	0	0	0
User 11	0	1	1	1	0	1



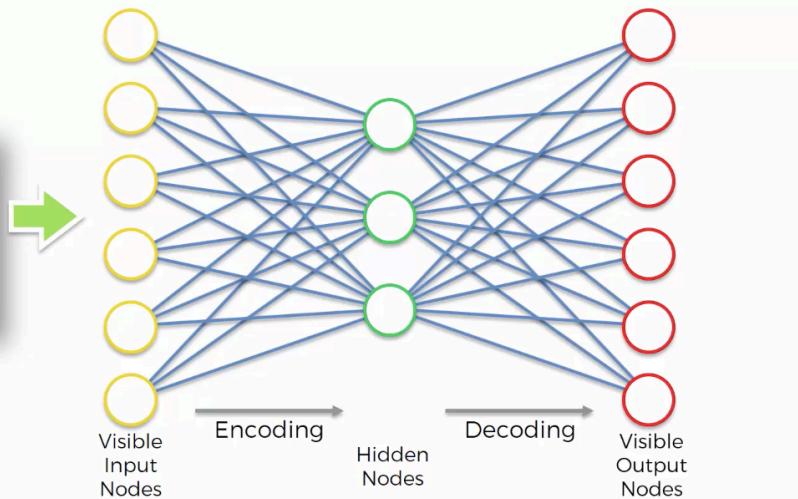
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 4

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0	0	1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	0
User 4	1	0	1	1	0	1
User 5	0	0	1	1	0	1
User 6	0	0	0	0	1	0
User 7	1	0	1	1	0	1
User 8	0	1	1	0	0	1
User 9	0	1	1	1	1	1
User 10	1	0	0	0	0	0
User 11	0	1	1	1	0	1



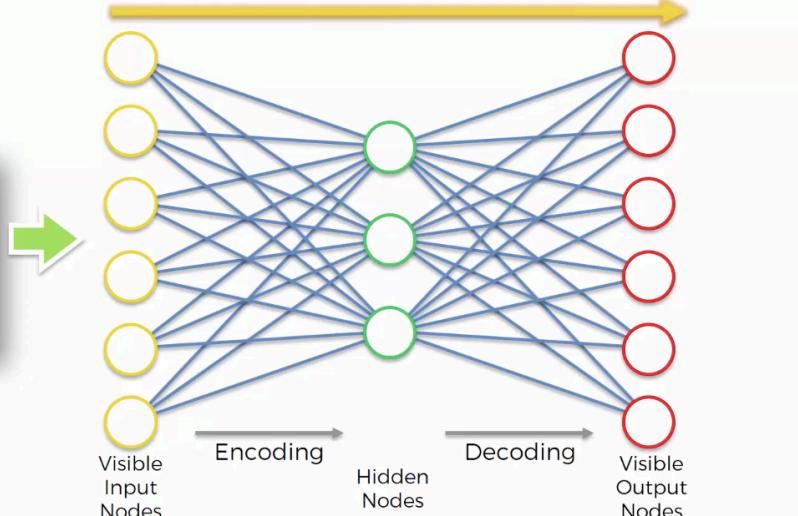
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 4

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0	0	1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	0
User 4	1	0	1	1	0	1
User 5	0	1	1	1	1	1
User 6	0	0	0	0	1	0
User 7	1	0	1	1	0	1
User 8	0	1	1	0	0	1
User 9	0	1	1	1	1	1
User 10	1	0	0	0	0	0
User 11	0	1	1	1	0	1



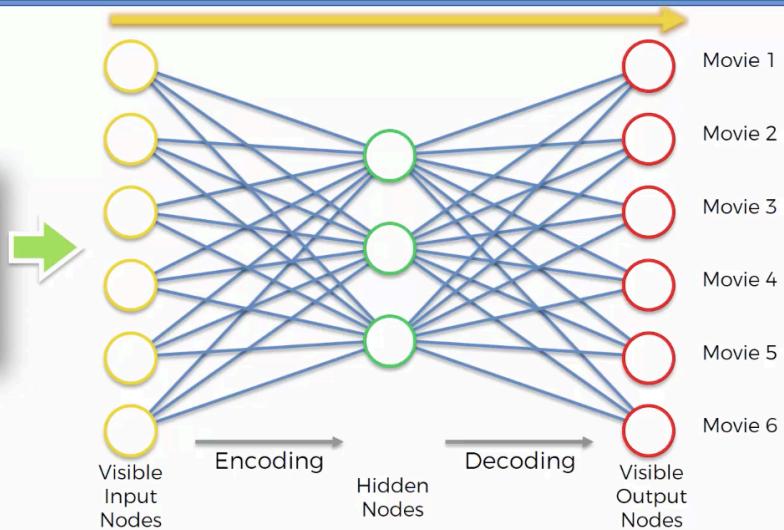
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 5

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0	0	1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	0
User 4	1	0	1	1	0	1
User 5	0	0	1	1	0	1
User 6	0	0	0	0	1	0
User 7	1	0	1	1	0	1
User 8	0	1	1	0	0	1
User 9	0	1	1	1	1	1
User 10	1	0	0	0	0	0
User 11	0	1	1	1	0	1



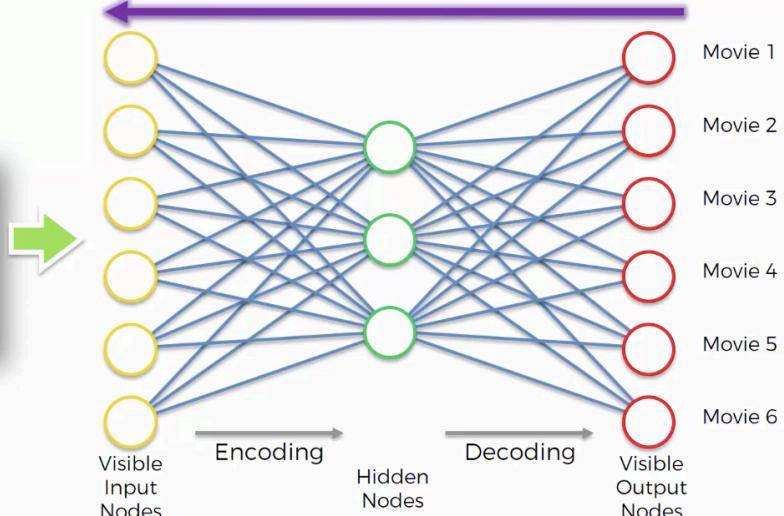
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 6

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0	0	1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	0
User 4	1	0	1	1	0	1
User 5	0	0	1	1	1	1
User 6	0	0	0	0	1	0
User 7	1	0	1	1	0	1
User 8	0	1	1	0	0	1
User 9	0	1	1	1	1	1
User 10	1	0	0	0	0	0
User 11	0	1	1	1	0	1



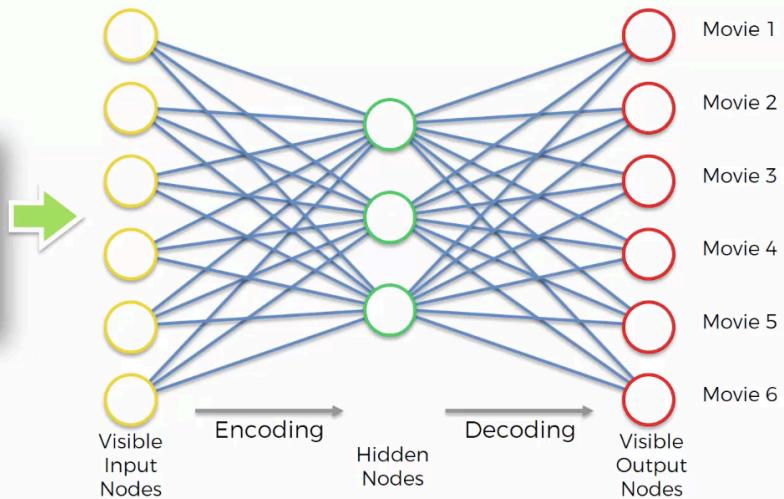
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 7

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0	0	1	1	1
User 2	0	1	0	0	1	0
User 3		1	1	0	0	
User 4	1	0	1	1	0	1
User 5	0		1	1		1
User 6	0	0	0	0	1	
User 7	1	0	1	1	0	1
User 8	0	1	1	0	0	1
User 9		0	1	1	1	
User 10	1		0	0	0	
User 11	0	1	1	1	0	1



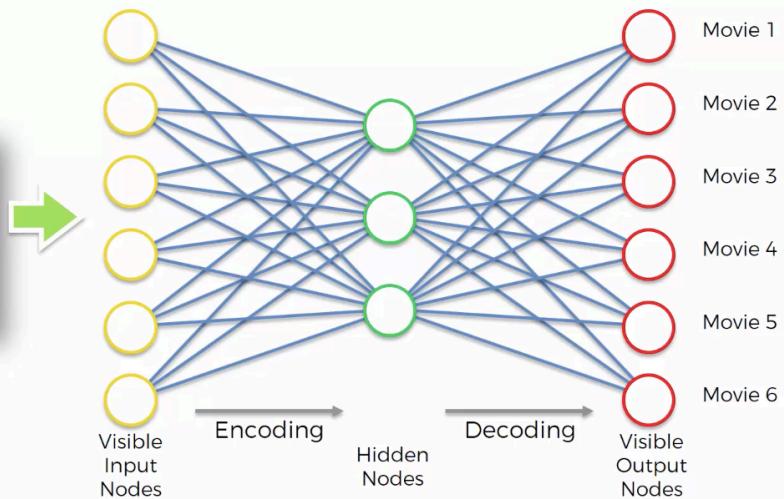
Machine Learning A-Z

© SuperDataScience

Training an Auto Encoder

STEP 7

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0	0	1	1	1
User 2	0	1	0	0	1	0
User 3		1	1	0	0	
User 4	1	0	1	1	0	1
User 5	0		1	1		1
User 6	0	0	0	0	1	
User 7	1	0	1	1	0	1
User 8	0	1	1	0	0	1
User 9		0	1	1	1	
User 10	1		0	0	0	
User 11	0	1	1	1	0	1



Machine Learning A-Z

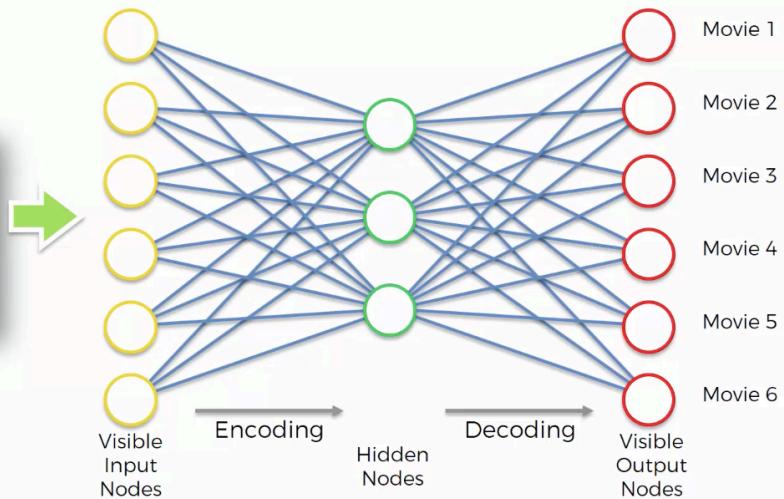
© SuperDataScience

Training an Auto Encoder

STEP 8

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	0		1	1	1
User 2	0	1	0	0	1	0
User 3	1	1	0	0	0	
User 4	1	0	1	1	0	1
User 5	0		1	1	0	1
User 6	0	0	0	0	1	
User 7	1	0	1	1	0	1
User 8	0	1	1		0	1
User 9	0	0	1	1	1	
User 10	1		0	0	0	
User 11	0	1	1	1	0	1

Repeat



Machine Learning A-Z

© SuperDataScience

In this step we go for the next epoch.

Training an Auto Encoder

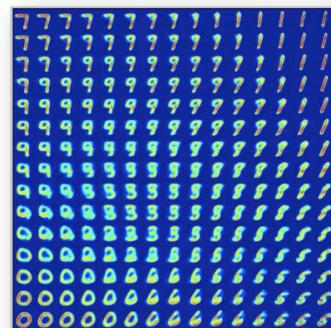
Additional Reading:

Building Autoencoders in Keras

By Francois Chollet (2016)

Link:

<https://blog.keras.io/building-autoencoders-in-keras.html>



Machine Learning A-Z

© SuperDataScience