

Other Architectures of Neural Networks

Dr. Mauricio Toledo-Acosta

Diplomado Ciencia de Datos con Python

Table of Contents

Other
Architectures
of Neural
Networks

Autoencoders

SOM

1 Autoencoders

2 SOM

Autoencoders

Other
Architectures
of Neural
Networks

Autoencoders

SOM

- La idea básica de un autoencoder es tener una capa de salida con la misma dimensionalidad que las entradas. La idea es tratar de reconstruir, de manera exacta, cada dimensión al pasarla por la red.

Autoencoders

Other
architectures
of Neural
Networks

Autoencoders

SOM

- La idea básica de un autoencoder es tener una capa de salida con la misma dimensionalidad que las entradas. La idea es tratar de reconstruir, de manera exacta, cada dimensión al pasarla por la red.
- La reconstrucción de los datos podría parecer una cuestión trivial, sin embargo, esto no es posible cuando el número de unidades en cada capa intermedia suele ser menor que el de la entrada (y salida).

Autoencoders

Other
architectures
of Neural
Networks

Autoencoders
SOM

- La idea básica de un autoencoder es tener una capa de salida con la misma dimensionalidad que las entradas. La idea es tratar de reconstruir, de manera exacta, cada dimensión al pasarla por la red.
- La reconstrucción de los datos podría parecer una cuestión trivial, sin embargo, esto no es posible cuando el número de unidades en cada capa intermedia suele ser menor que el de la entrada (y salida).
- Las neuronas de las capas ocultas aprenden una representación reducida de los datos que les permita reconstruir la entrada.

Autoencoders

Other
architectures
of Neural
Networks

Autoencoders
SOM

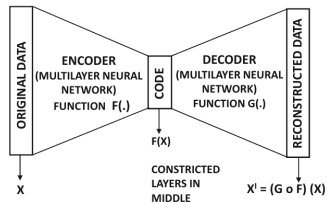
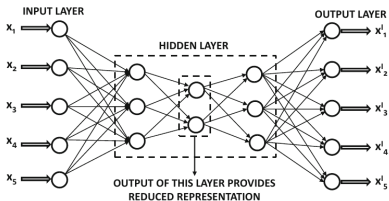
- La idea básica de un autoencoder es tener una capa de salida con la misma dimensionalidad que las entradas. La idea es tratar de reconstruir, de manera exacta, cada dimensión al pasarla por la red.
- La reconstrucción de los datos podría parecer una cuestión trivial, sin embargo, esto no es posible cuando el número de unidades en cada capa intermedia suele ser menor que el de la entrada (y salida).
- Las neuronas de las capas ocultas aprenden una representación reducida de los datos que les permita reconstruir la entrada.
- Esta reconstrucción, inherentemente, tiene pérdidas.

Autoencoders

Other
Architectures
of Neural
Networks

Autoencoders

SOM



Aplicaciones

Other
Architectures
of Neural
Networks

Autoencoders

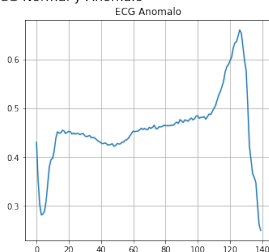
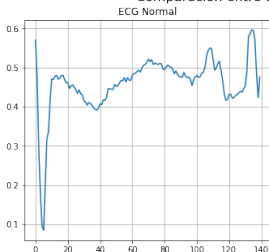
SOM

- Reducción de dimensionalidad.

Aplicaciones

- Reducción de dimensionalidad.
- Detección de Anomalías (outliers)

Comparación entre un ECG Normal y Anómalo



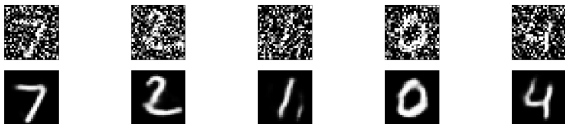
Aplicaciones

Other
Architectures
of Neural
Networks

Autoencoders

SOM

- Reducción de dimensionalidad.
- Detección de Anomalías (outliers)
- Denoising



Aplicaciones

Other
Architectures
of Neural
Networks

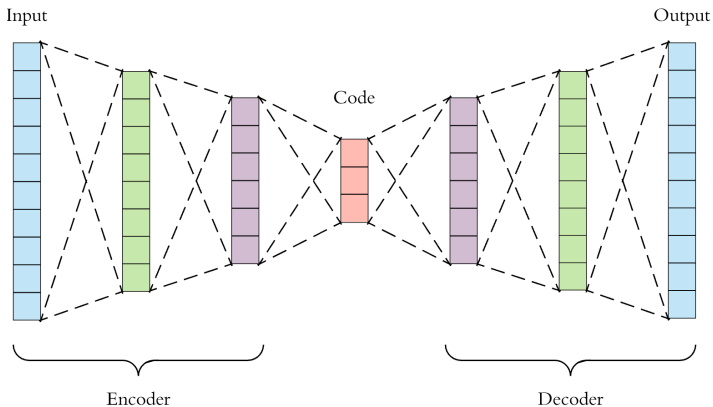
Autoencoders

SOM

- Reducción de dimensionalidad.
- Detección de Anomalías (outliers)
- Denoising
- Compresión de imágenes.

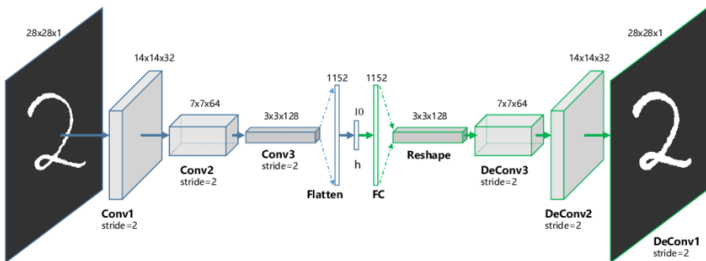
Autoencoders

- La arquitectura suele ser simétrica respecto a la capa oculta central.



Autoencoders

- La arquitectura suele ser simétrica respecto a la capa oculta central.



- De acuerdo a la tarea, puede ser que nos interese solamente la salida de la red, o la capa oculta latente.

Table of Contents

Other
Architectures
of Neural
Networks

Autoencoders

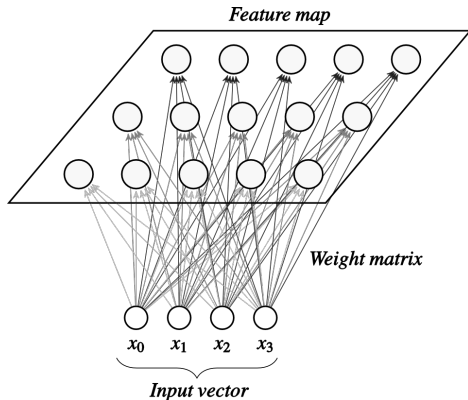
SOM

1 Autoencoders

2 SOM

SOM: Self-organizing map

The self-organizing map (SOM) is an algorithm for mapping from one (usually high-dimensional) space to another (usually low-dimensional) space.



Applications

Other
Architectures
of Neural
Networks

Autoencoders
SOM

- Classification
- Clustering
- Visualization
- Outlier Detection

Competitive Learning

Other
architectures
of Neural
Networks

Autoencoders

SOM

- 1 Randomize the node weight vectors in a map.
- 2 Randomly pick an input vector $D(t)$.
- 3 Traverse each node in the map.
- 4 Use the Euclidean distance formula to find the similarity between the input vector and the map's node's weight vector.
- 5 Track the node that produces the smallest distance (this node is the best matching unit, BMU) Update the weight vectors of the nodes in the neighborhood of the BMU (including the BMU itself) by pulling them closer to the input vector

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v(s)).$$

- 6 Increase s and repeat from step 2 while $s < \text{iter}$.