

# AUTOENCODERS

teoría, herramientas y casos prácticos

David Charte



UNIVERSIDAD  
DE GRANADA



18 de diciembre de 2019

Fundamentos

Herramientas

Casos prácticos

Detección de anomalías

# FUNDAMENTOS

# Aprendizaje no supervisado

## DEFINICIÓN

Búsqueda de patrones en datos sin etiquetar

## MOTIVACIÓN

Escasez de datos etiquetados,  
búsqueda de estructuras implícitas

## MÉTODOS CLÁSICOS

K-medias (*clustering*), LOF (anomalías)

# Aprendizaje de representaciones

## MOTIVACIÓN

Los modelos dependen de las variables

Las representaciones mezclan factores que explican la variación

## TRANSFORMACIONES CLÁSICAS

Componentes principales, discriminantes lineales,  
escalado multidimensional, extracción de *manifolds*

## APLICACIONES

Procesamiento de señales, reconocimiento de objetos, NLP, *Transfer learning*

## LECTURA RECOMENDADA

*Representation Learning. A review and new perspectives* (Bengio et al.)

Capas de operaciones sencillas

+

Transformaciones no lineales

+

Se propagan errores hacia atrás

+

Se actualizan parámetros con SGD

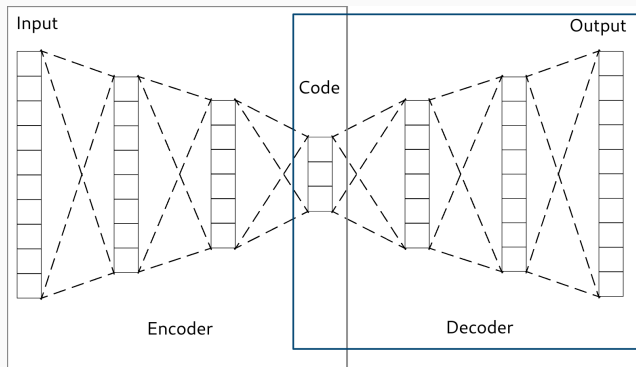
## CLAVES

Cada capa extrae una representación

Una arquitectura es una familia paramétrica de funciones

El objetivo es diferenciable (c.p.d.)

# Autoencoders



Composición de un **codificador** y un **decodificador**.

OBJETIVO

$$\min_{\theta} \sum_x d(c, g_{\theta}(f_{\theta}(x)))$$

LECTURA RECOMENDADA

*A practical tutorial on autoencoders for nonlinear feature fusion* (Charte et al.)

**HERRAMIENTAS**



# Creación de un autoencoder

## TENSORFLOW/KERAS

```
l_input      = Input(shape=(100,))
l_encoding   = Dense(10)(l_input)
l_code       = Input(shape=(10,))
l_decoding   = Dense(100)(l_code)
encoder      = Model(l_input, l_encoding)
decoder      = Model(l_code, l_decoding)
autoencoder  = Model(l_input, decoder(l_encoding))
autoencoder.compile(optimizer="rmsprop", loss="mean_squared_error")
```

[RUTA](https://cran.r-project.org/package=ruta) (cran.r-project.org/package=ruta)

```
model = autoencoder(input() + dense(10) + output())
```

# Implementaciones

**RUTA** `ruta.software`

Autoencoders denso, convolucional +  
*sparse, **contractive**, denoising, robust, **variational***

**AUTOENCODER** `pypi.org/project/autoencoder/`

Autoencoder convolucional personalizable

**KERAS** `github.com/keras-team/keras/blob/master/examples/variational_`  
`autoencoder_deconv.py`

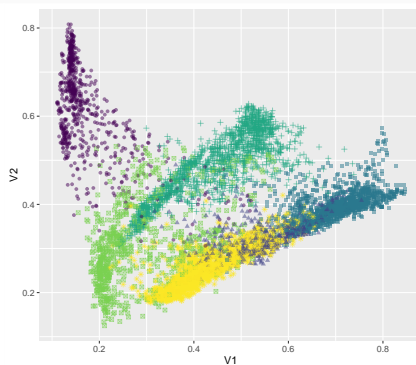
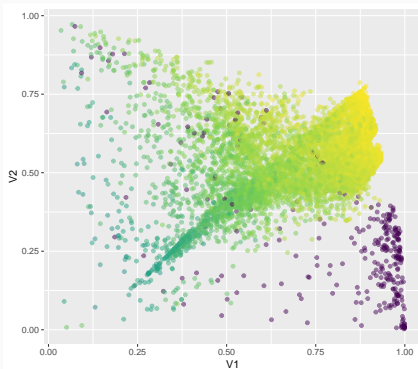
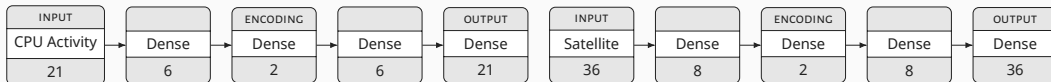
Autoencoder variacional convolucional (MNIST)

**SEQ2SEQ** `pypi.org/project/seq2seq`

Encoder-decoder para secuencias reales

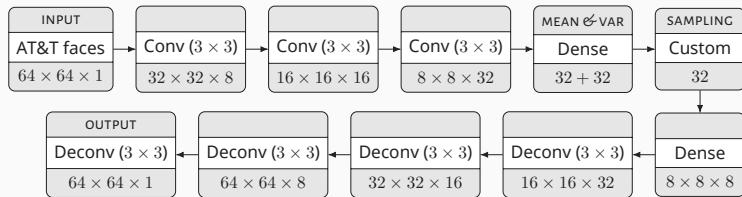
# CASOS PRÁCTICOS

# Embedding

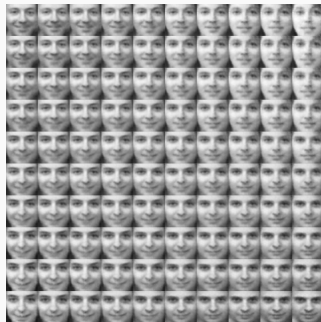


# Ajuste de distribuciones (I): instance generation

## Variational AE:

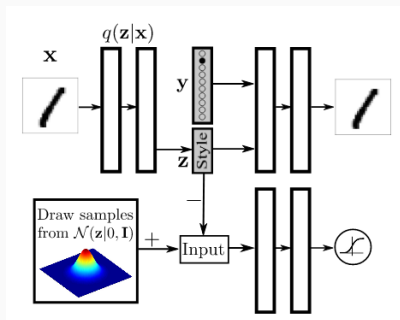


```
cross_ent = tf.nn.sigmoid_cross_entropy_with_logits(  
    logits=predictions, labels=inputs)  
logpx_z = -tf.reduce_sum(cross_ent, axis=[1, 2])  
logpz = log_normal_pdf(z, 0., 1.)  
logqz_x = log_normal_pdf(z, mean, logvar)  
variational_loss = -tf.reduce_mean(10*logpx_z + logpz - logqz_x)
```



# Ajuste de distribuciones (II): feature disentanglement

**Adversarial AE** = AE + GAN

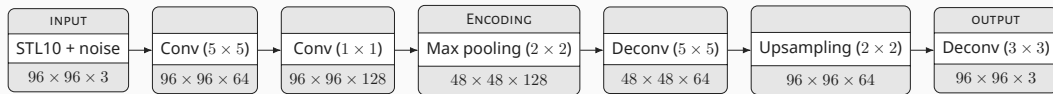


	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	0	1	2	3	4	5	6	7	8	9
2	0	1	2	3	4	5	6	7	8	9
3	0	1	2	3	4	5	6	7	8	9
4	0	1	2	3	4	5	6	7	8	9
5	0	1	2	3	4	5	6	7	8	9
6	0	1	2	3	4	5	6	7	8	9
7	0	1	2	3	4	5	6	7	8	9
8	0	1	2	3	4	5	6	7	8	9
9	0	1	2	3	4	5	6	7	8	9

LECTURA RECOMENDADA

*Adversarial Autoencoders* (Makhzani et al.)

# Limpieza



# Otras aplicaciones

**Superresolución** de imágenes

**Compresión** de imágenes y señales

**Hashing** semántico

**Transfer learning**

Recuperación de **poses** humanas

Aprendizaje de **formas 3D**

Sistemas de **recomendación** y etiquetado



# DETECCIÓN DE ANOMALÍAS

## CONJETURA

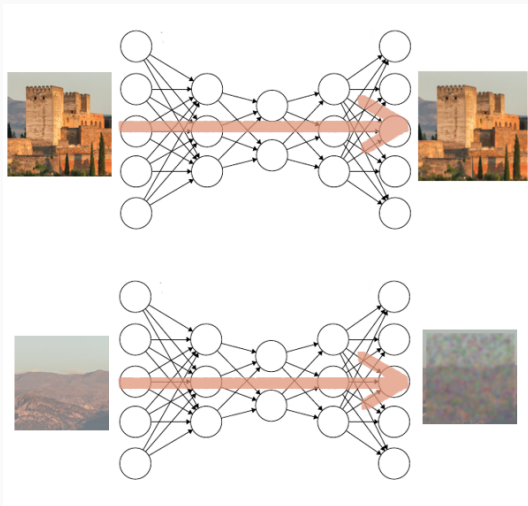
Un autoencoder entrenado en datos normales no puede reconstruir datos anómalos

## MÉTODO

Score = error de reconstrucción

## LECTURA RECOMENDADA

*Anomaly detection using autoencoders with nonlinear dimensionality reduction*  
(Sakurada et al.)



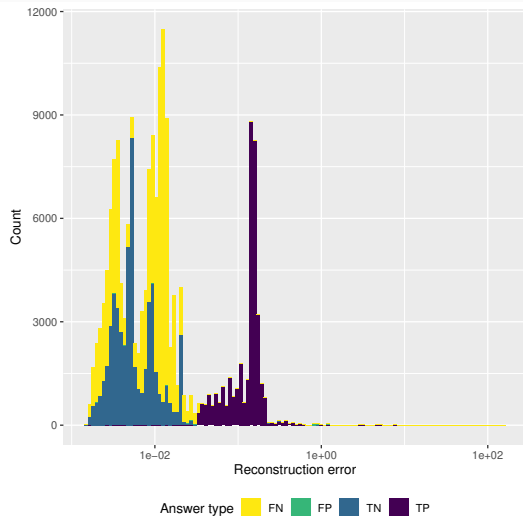
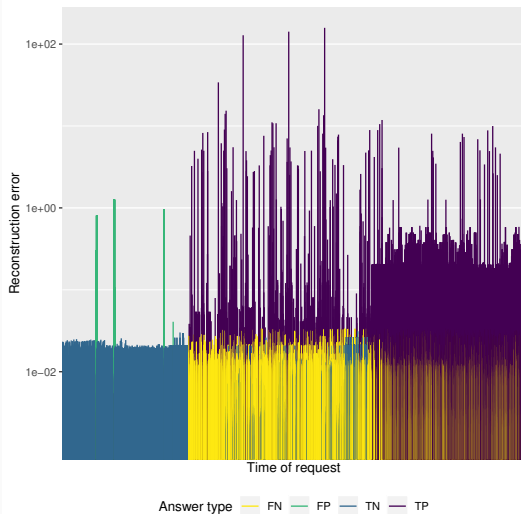
## Detección de ataques en redes: **UNSW-NB15**

```
autoencoder = Sequential([
    Dense(units=2, input_shape=187, activation="relu"),
    Dense(units=187)
])
autoencoder.compile(loss="mean_squared_error", optimizer="adam")
autoencoder.fit(train_x, train_x, epochs=5, batch_size=256)

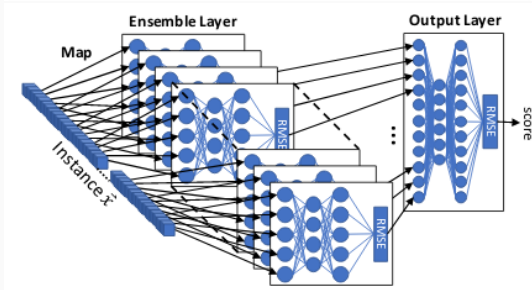
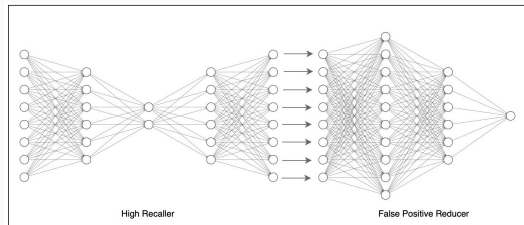
rec = autoencoder.predict(test_x)
train_rec = autoencoder.predict(train_x)

# cálculo de errores...
```

# Resultados



1. Variacional
2. FP reducer/  
clasificador
3. Ensemble



## Casos reales en mantenimiento predictivo

Chen, Z., & Li, W. (2017). Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. *IEEE Transactions on Instrumentation and Measurement*, 66(7), 1693-1702.

Jiang, G., He, H., Xie, P., & Tang, Y. (2017). Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 66(9), 2391-2402.

Sun, W., Shao, S., Zhao, R., Yan, R., Zhang, X., & Chen, X. (2016). A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement*, 89, 171-178.

Gugulothu, N., TV, V., Malhotra, P., Vig, L., Agarwal, P., & Shroff, G. (2017). Predicting remaining useful life using time series embeddings based on recurrent neural networks. *arXiv preprint arXiv:1709.01073*.

**EN RESUMEN...**

Un autoencoder encuentra representaciones de forma no supervisada

Se adapta a multitud de casos de representation learning

Detecta anomalías conociendo solo casos normales