

Aplicaciones prácticas de las redes neuronales no supervisadas

David Charte

21 jun 2019 – esLibre (Informática y Matemáticas)

CC BY-SA



LibreIM

esLibre



Introducción

Introducción

Autoencoders:

- aprendizaje **de características** no supervisado
- con **redes neuronales**

En esta charla veremos:

- 4 aplicaciones prácticas con código

Código fuente:

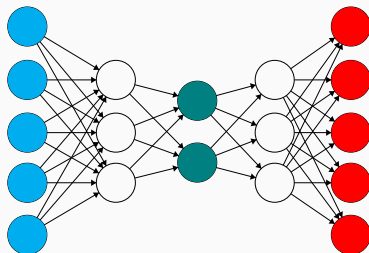
- github.com/ari-dasci/autoencoder-showcase
- usando el paquete **ruta** para R
cran.r-project.org/package=ruta

Charte, D. et al.: A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. Information Fusion 44, 78–96 (2018).

Charte, D., Herrera, F., Charte, F.: Ruta: Implementations of neural autoencoders in R. Knowledge-Based Systems 174, 4–8 (2019)

Fundamentos

Atributos \xrightarrow{f} Codificación \xrightarrow{g} Atributos



Optimizar para que $g(f(x)) \approx x$
(además, otras restricciones o propiedades)

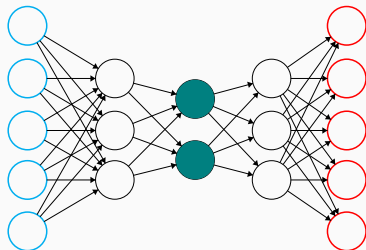
Ejemplos

Visualizar datos de alta dimensión

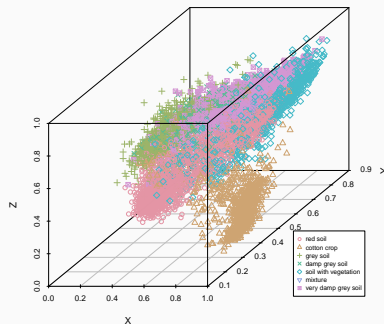
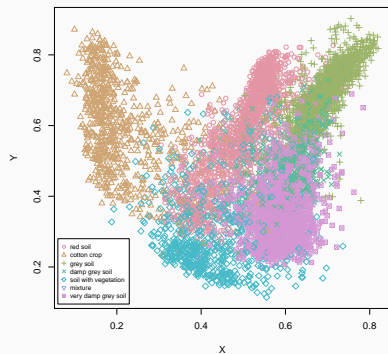
Usar una capa de codificación en **2 o 3 dimensiones**



Dibujar los nuevos **códigos** en gráficas de dispersión



Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006).



```

network <- input() + dense(12, activation = "relu") +
  dense(3, activation = "sigmoid") +
  dense(12, activation = "relu") + output("linear")
model <- autoencoder_sparse(network) %>% train(x_train, epochs = 40)
codes <- model %>% encode(x_train)
scatterplot3d(codes, color = rainbow_hcl(7)[class], (1:7)[class])

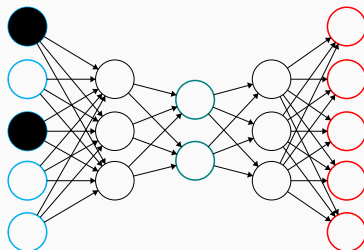
```

Reducción de ruido (en imágenes)

Entrenar el autoencoder con **datos ruidosos**

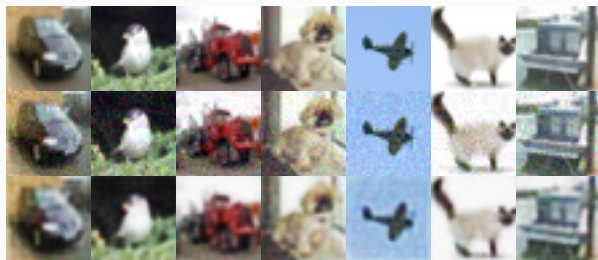


Sus **reconstrucciones eliminan ruido** de los **datos de entrada**



Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in neural information processing systems. pp. 341–349 (2012)

Muestras de test



Ruidosas

Limpias

```
network <- input() + conv(16, 3, upsampling = 2, activation = "relu") +
  conv(16, 3, max_pooling = 2, activation = "relu") +
  conv(3, 3, activation = "sigmoid")
```

```
model <- autoencoder_denoising(network, loss = "binary_crossentropy",
  noise_type = "gaussian", sd = .05) %>%
  train(x_train, epochs = 30, batch_size = 500, optimizer = "adam")
```

```
noisy <- noise_gaussian(sd = .05) %>% apply_filter(x_test)
recovered <- model %>% reconstruct(noisy)
```

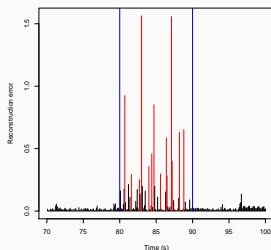
Detección de anomalías

Entrenar con **datos normales**



Los datos anómalos provocarán alto **error de reconstrucción**

Sakurada, M., Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis. pp. 4–11. ACM (2014).



Error de reconstrucción
en test (entre 80 y 90 s
hay anomalías)

```
reconstructions <- autoencoder_denoising(
  input() +
  dense(16, activation = "sigmoid") +
  output(),
  loss = "mean_squared_error",
  noise_type = "saltpepper", p = 0.1) %>%
  train(x_train, epochs = 50,
        batch_size = 32) %>% reconstruct(x_test)

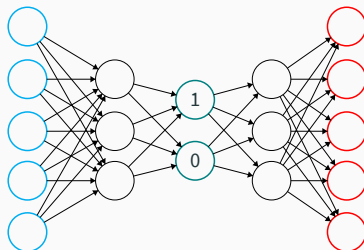
errors <- rowMeans(
  (reconstructions - x_test) ** 2)
```

Hashing semántico

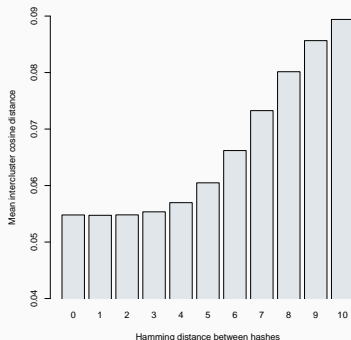
Forzar **codificación binaria**



Instancias similares obtienen **hashes similares**



Salakhutdinov, R., & Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7), 969-978.



```
network <- input()+dense(256)+layer_keras("gaussian_noise", stddev = 16)+
  dense(10, activation = "sigmoid")+dense(256)+output("sigmoid")
model <- autoencoder(network, "binary_crossentropy") %>%
  train(x_train, epochs = 50)
encodings <- model %>% encode(x_test) %>%
  apply(1, function(r) as.integer(r > 0.5))
```

Otras

- Mejoras en rendimiento de clasificación
- Mejora de calidad de voz
- Mezclar poses (humanas) 2 y 3 dimensionales
- Recomendación de etiquetas
- Traducción (incluso con reordenaciones)
- Predicción de movimiento en imágenes

Conclusiones

Conclusiones

El aprendizaje de características tiene **muchas aplicaciones**

Los autoencoders son **versátiles** y se pueden adaptar

¡Gracias!

¿Preguntas?

Aplicaciones prácticas de las redes neuronales no supervisadas

David Charte

21 jun 2019 – esLibre (Informática y Matemáticas)

CC BY-SA



LibreIM

esLibre

