

# Aprendizaje Automático

---

## Aprendizaje profundo, PCA y aplicaciones

Master Executive Big Data y Business Analytics

Edición 2015

# Temario

---

- ❖ Sesión 1: Introducción a aprendizaje automático
- ❖ Sesión 2: Modelos de clasificación y clustering
- ❖ **Sesión 3: Aprendizaje profundo, PCA y aplicaciones**



# Tabla de contenidos

---

- ❖ Peso de la Evidencia (WoE)
- ❖ Aprendizaje Profundo
- ❖ Análisis de Componentes Principales
- ❖ Motores de Recomendación

# Tabla de contenidos

---

- ❖ **Peso de la Evidencia (WoE)**
- ❖ Aprendizaje Profundo
- ❖ Análisis de Componentes Principales
- ❖ Motores de Recomendación



# Peso de la Evidencia (WoE)

---

El peso de la evidencia (WoE, Weight of Evidence) es un valor que indica la capacidad para compara la capacidad predictiva de cada una de los niveles de una variable:

$$WoE_i = \ln \left| \frac{R_i(T)}{R_i(F)} \right|$$

El valor del WoE se puede situar entre menos infinito, todos los registros del nivel son negativos, e infinito, todos los registros del nivel son positivos.

Con este indicador el IV se puede reescribir:

$$IV = \sum_{i=1}^N (R_i(T) - R_i(F)) \ln \left| \frac{R_i(T)}{R_i(F)} \right| = \sum_{i=1}^N (R_i(T) - R_i(F)) WoE_i$$

# Tabla de contenidos

---

- ❖ Peso de la Evidencia (WoE)
- ❖ **Aprendizaje Profundo**
- ❖ Análisis de Componentes Principales
- ❖ Motores de Recomendación



# Aprendizaje Profundo

---

El **Aprendizaje Profundo** (Deep Learning), o **Aprendizaje Jerárquico** (Hierarchical Learning), es una rama del aprendizaje automático basado en un conjunto de algoritmos que intentan modelar abstracciones de alto nivel en los datos mediante el uso de arquitecturas de modelo, con estructuras complejas o de otro tipo, compuestas por múltiples transformaciones no lineales.

Por ejemplo, una imagen puede ser representada como el valor de una matriz de píxeles o de una manera más abstracta mediante un conjunto de bordes, regiones de forma particular, etc.

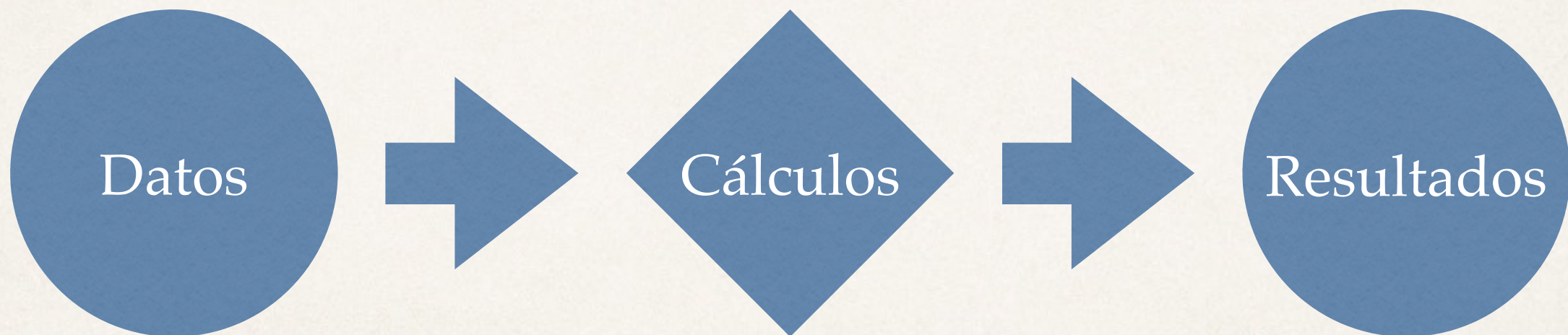


# Profundidad

---

En un diagrama de flujo la **profundidad** es: la longitud de la trayectoria más larga desde una entrada a una salida.

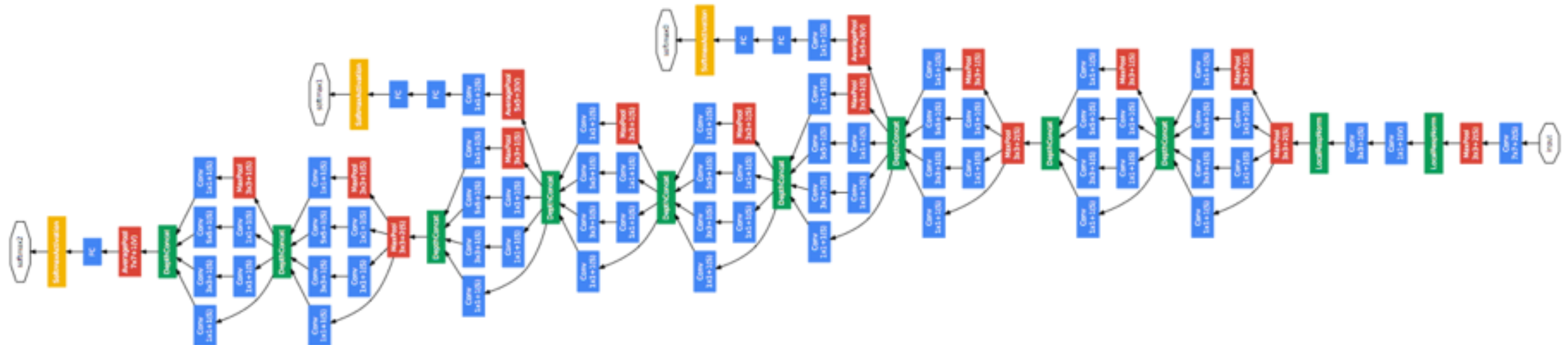
Los modelos de aprendizaje automático usan un modelo de computación simple:



Este es un modelo de profundidad 1



# Modelos de Aprendizaje Profundo



Fuente: Christian Szegedy et al. "Going deeper with convolutions" arXiv:1409.4842v1, 2014



# Niveles de abstracción

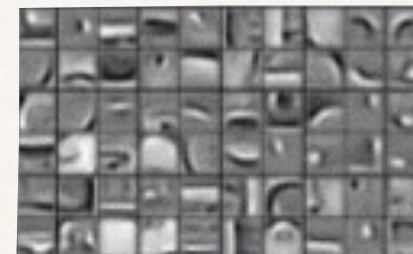
---



Caras



Objetos



Componentes de  
los objetos



Bordes



Pixels

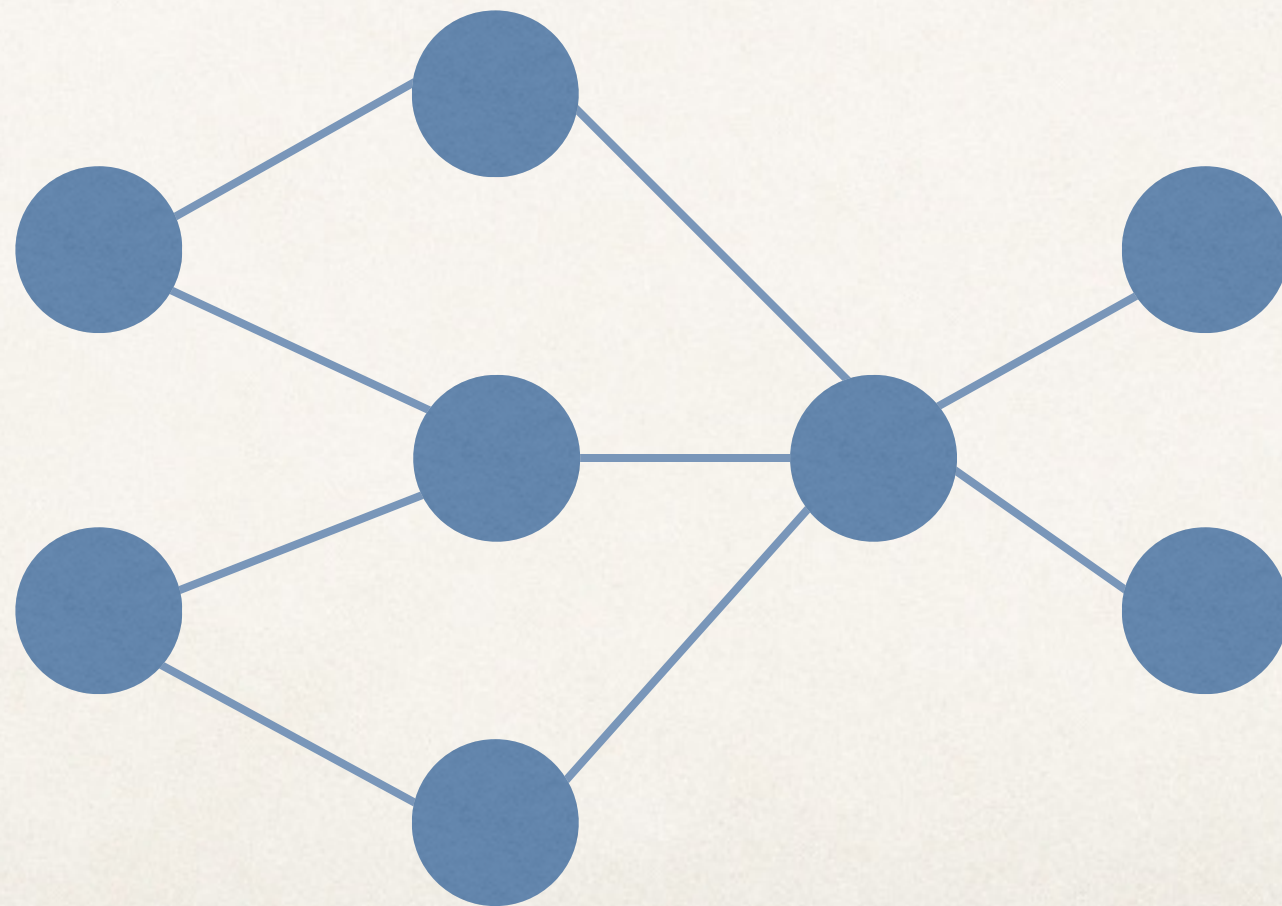


# Redes Neuronales

---

Las redes neuronales tradicionales se pueden considerar que tienen una profundidad igual al número de capas.

Las redes neuronales se suelen entrenar mediante propagación hacia atrás de los errores (Back Propagation).



# Entrenamiento de las redes profundas

---

La propagación hacia atrás de los errores presenta problemas en redes profundas:

- ❖ A medida que aumentan las capas las correcciones se van diluyendo
- ❖ Se quedan atascados en mínimos locales.
- ❖ Son necesarios datos previamente etiquetados.

En redes profundas se suele utilizar otras aproximaciones

- ❖ Uso inicial de aprendizaje no supervisado:
  - ❖ Permite abstraer el desarrollo de las capas.
  - ❖ Permite iniciar con mejores parámetros.
- ❖ Finalmente se inicia un proceso supervisado
  - ❖ Afina los valores de las capas intermedias.



# Instalación de Theano

---

En las sesiones de aprendizaje profundo se utilizará la librería Theano (<http://deeplearning.net/software/theano/index.html>), la cual no se incluye en la distribución estándar de Anaconda.

Theano es un paquete de software que le permite escribir código simbólico y compilarlo. Fue desarrollado por investigadores de aprendizaje automático en la Universidad de Montreal. Su uso no se limita a las aplicaciones de aprendizaje automático, pero fue diseñado con el aprendizaje automático en mente.

Para la instalación de Theano en Anaconda se han de ejecutar los siguientes comandos en la terminal:

```
pip install Theano
```

```
conda install pydot
```

A partir de este punto debería poderse importar Theano en un IPython Notebook utilizando el siguiente comando:

```
import theano
```



# Introducción a Theano: variables simbólicas

---

En Theano es necesario definir las variables de forma simbólica:

```
x0 = T.scalar() # 0-D  
x1 = T.vector() # 1-D  
x2 = T.matrix() # 2-D  
x3 = T.tensor() # 3-D  
x4 = T.tensor() # 4-D
```

Las variables definidas de esta forma no contienen ninguna valor, solamente se ha indicado el tipo de variable.



# Introducción a Theano: funciones

---

Posteriormente, las funciones se han de declarar para que sean compiladas:

```
y = x**2
f = theano.function(inputs = [x0], outputs = y)

print f(2)
```

Las Funciones pueden ser todo lo complejas que se desee:

```
def square(x):
    return x**2

y = square(x)
f = theano.function(inputs = [x0], outputs = y)

print f(2)
```

# Introducción a Theano: variables compartidas

---

Las variables simbólicas de Theano no tienen valores explícitos, solamente se utilizan para indicar las operaciones a realizar. Las variables compartidas son las que se han de utilizar cuando se desee indicar un valor explícito:

```
shared_var = theano.shared(np.array([[1, 2], [3, 4]],  
                               dtype = theano.config.floatX))
```

```
print shared_var.get_value()
```

```
f = theano.function([], shared_var)
```

```
print f()
```



# Introducción a Theano: gradientes

---

Una de las ventajas de utilizar Theano es su capacidad para calcular gradientes de funciones. Esto permite definir una función simbólica y calcular su derivada numérica.

```
sx = T.scalar()  
y  = x**2  
  
# y_grad = dy/dx  
y_grad = T.grad(y, x)  
  
# dy/dx = 2 * x  
y_grad.eval({x: 10})
```

# Tabla de contenidos

---

- ❖ Peso de la Evidencia (WoE)
- ❖ Aprendizaje Profundo
- ❖ **Análisis de Componentes Principales**
- ❖ Motores de Recomendación



# Análisis de componentes principales (PCA)

---

El análisis de componentes principales (PCA, Principal Component Analysis) es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos.



# Varianza y Covarianza

---

La varianza es la medida de la dispersión de un conjunto de una variable:

$$var(X) = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1} = \left( \frac{1}{n - 1} \sum_{i=1}^n X_i^2 \right) - \bar{X}^2$$

La covarianza es la medida del grado de variación conjunto de dos variable:

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X}) (Y_i - \bar{Y})}{n - 1}$$



# Matriz de Covarianza

---

La matriz de varianza contiene los valores de colarais entre cada par de dimensiones de un conjunto de datos. En numpy se puede calcular con el comando `np.cov`.

$$\begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) & cov(x_1, x_3) \\ cov(x_2, x_1) & cov(x_2, x_2) & cov(x_2, x_3) \\ cov(x_3, x_1) & cov(x_3, x_2) & cov(x_3, x_3) \end{bmatrix}$$

# Vectores propios y valores propios

---

Los vectores propios (o autovectores) de una matriz son los vectores no nulos que dan lugar a un múltiplo escalar de si mismos, con lo que no cambian de dirección. El múltiplo escalar es el valor propio (o autovalor).

$$A\vec{v} = \lambda\vec{v}$$

Los valores propios se obtienen mediante la expresión:

$$(A - \lambda I) \vec{v} = 0$$

lo que implica:

$$|A - \lambda I| = 0$$



# Reducción de la dimensionalidad con PCA

---

Las dimensiones de un conjunto de datos se puede reducir mediante PCA.

1. En cada una de las dimensiones se resta la media.
2. Se obtiene la matriz de covarianza de este conjunto de datos.
3. Se obtienen los vectores propios y valores propios de esta matrix.
4. Se proyectan los datos en los vectores propios para obtener los componentes.
5. Las componentes con mayores valores propios son las componentes principales.

# Tabla de contenidos

---

- ❖ Peso de la Evidencia (WoE)
- ❖ Aprendizaje Profundo
- ❖ Análisis de Componentes Principales
- ❖ Motores de Recomendación



# Algoritmos de recomendación

---

Existen dos tipos diferentes de sistemas de recomendación:

- ❖ Las recomendaciones basadas en usuarios: en estos sistemas se busca a clientes similares y sobre esta similitud se recomienda a cada uno los artículos que ha resultado de interés a los otros.
- ❖ Las recomendaciones basadas en artículos: en estos sistemas se buscan artículos similares a los que un usuario utiliza, sobre la base de puntuación los artículos y la similitud se presentan los que pueden ser de interés a otros usuarios.



# Filtrado colaborativo

---

Los sistemas de filtrado colaborativo se adaptan al siguiente funcionamiento:

- ❖ Los usuarios elaboran valoraciones sobre ítems.
- ❖ Se analizan las valoraciones realizadas por los usuarios con tal de establecer grupos o vecinos cercanos de usuarios con preferencias similares.
- ❖ Una vez obtenidos los grupos de vecinos para un usuario, se realizan las recomendaciones al usuario activo teniendo en cuenta sus vecinos más cercanos y sus correspondientes valoraciones.



# Filtrado colaborativo

	Red Delicious	Golden Delicious	Granny Smith	Winesap	Gala	Rome
Enrique	2	6		10	8	
Paco	7	4	9			2
Noelia	3	5	5	9	8	9
Pedro		9			2	6
María	9	8	2	2	1	5

¿Le gustara a Enrique la Granny Smith o Rome?

- ❖ Selecciona usuarios similares: Noelia
- ❖ Se utiliza las evaluaciones de los cercanos para estimar el la puntuación de Enrique.
- ❖ En base a las puntuaciones de Noelia es mas probable que valor mejor la Rome que la Granny Smith