Extracción de Atributos

Dr. Jesús Ariel Carrasco Ochoa
ariel@inaoep.mx
Oficina 8311

Contenido

- Introducción
- PCA
- LDA
- Escalamiento multidimensional
- Programación genética
- Autoencoders

Extracción de atributos

- Objetivo
  - Preprocesamiento
    - Clasificación Supervisada
    - Regresión
    - Agrupamiento
  - Visualización

Extracción de atributos

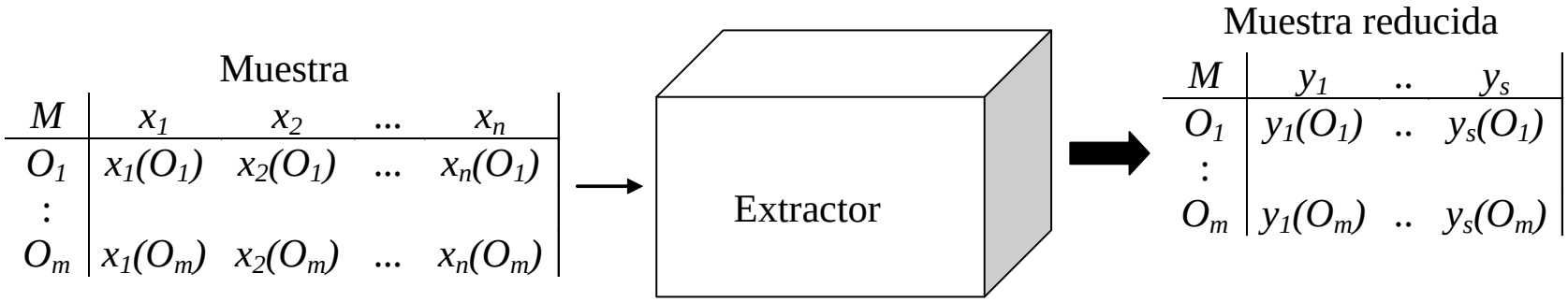
- Objetivo
  - Preprocesamiento
    - **Clasificación Supervisada**
    - Regresión
    - Agrupamiento
  - **Visualización**

Extracción de atributos

Por qué hacer Extracción de atributos en clasificación supervisada

- Mejorar los resultados de la clasificación
- Reducir el costo de la clasificación
- Acelerar el proceso de clasificación

Extracción de variables



Extracción de atributos

El objetivo es transformar el espacio de representación en otro de menor dimensión que conserve (o mejore) la información contenida en la representación original

- Directas
- Heurísticas

Estrategias de Extracción

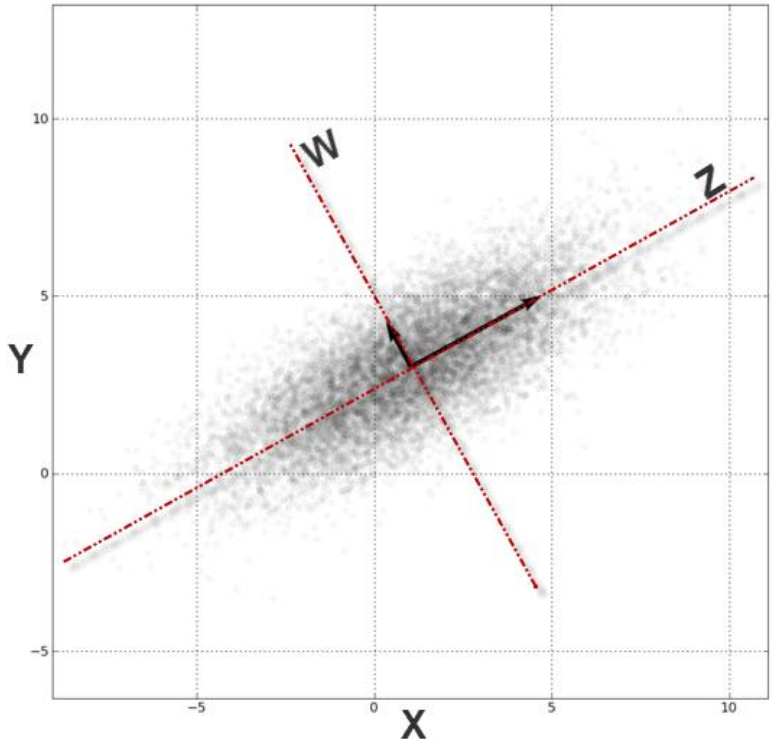
- Directas (tratan de conservar o mejorar alguna característica de los datos)
  - PCA
  - LDA
  - Escalamiento multidimensional
- Heurísticas
  - Programación genética
  - Autoencoders

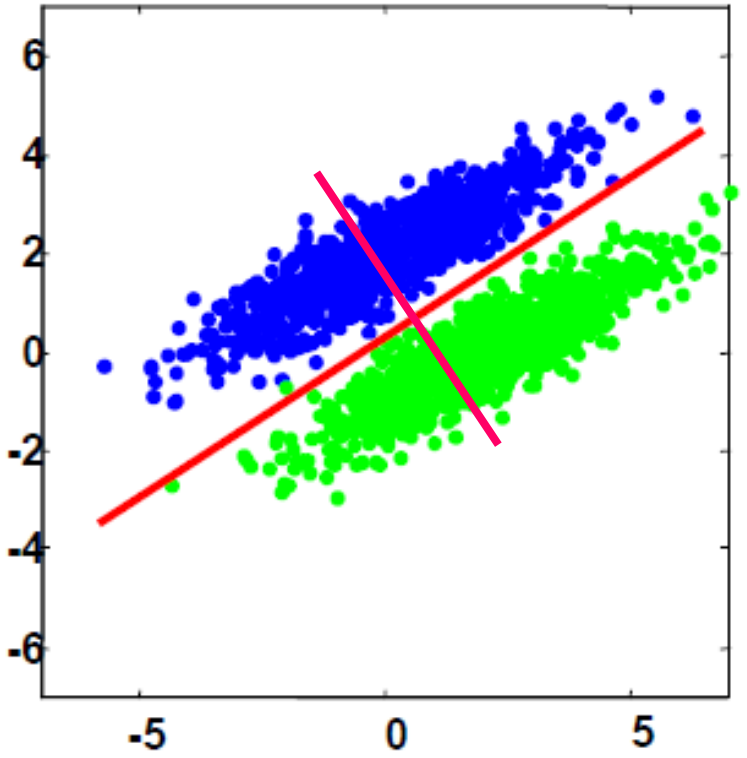
PCA

- Transformación lineal
- Busca un nuevo sistema de coordenadas ortogonales donde cada eje conserve tanta varianza (de los datos originales) como sea posible
- La primera componente preserva la mayor cantidad de varianza, la segunda preserva la mayor cantidad de la varianza restante, y así sucesivamente.
- No utiliza la información de las clases

PCA

- Busca las direcciones en los datos con mayor variación





PCA

Sea $X=\{x_i \mid i=1,\ldots,n\}$ con $x_i \in \Re^m$ un conjunto de n objetos almacenados en una matriz de nxm

1. Centrar los datos (A cada columna de X le restamos la media de la columna)
2. Calcular la matriz de covarianza de mxm como:

$$\Sigma_X=(1/m)X^TX$$

3. Calcular los eigenvectores y eigenvalores de Σ_X

$$\Sigma_X V=\lambda V$$

4. La matriz V de mxm contiene los eigenvectores y

PCA

Cada fila de V es una de las componentes principales

Si conservamos todas no se reduce la dimensión de los datos

5. Se seleccionan en V_k (de mxk) las k<m primeras que capturen el K% de la varianza (los primeras k tales que la suma de sus eigenvalores sea ≥ K/100)

6. La nueva representación Y (de nxk) se obtiene como:

$$Y=XV_k$$

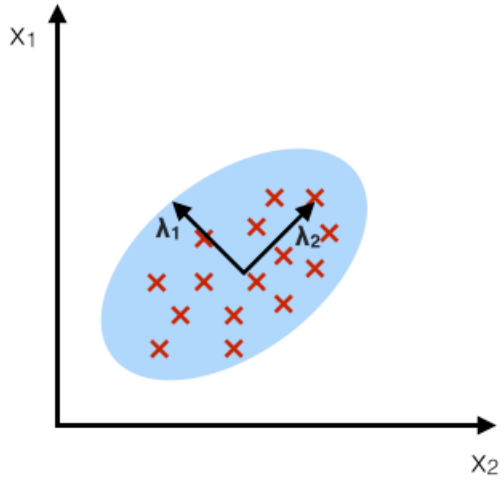
LDA

- Transformación lineal
- Busca un nuevo sistema de coordenadas que preserve o maximice la separación entre las clases
- Utiliza la información de las clases

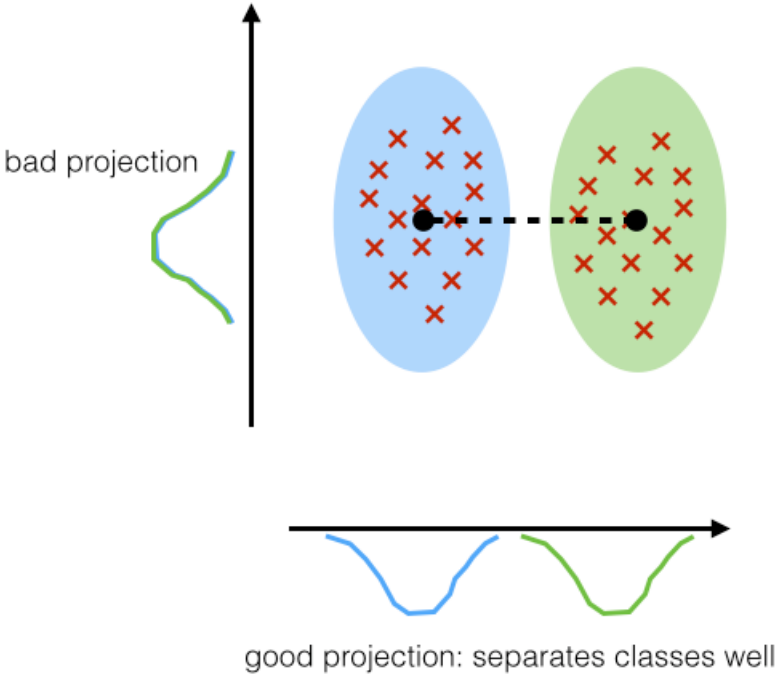
LDA

- Busca las direcciones en los datos con mayor separación entre las clases

PCA



LDA



LDA K clases ($K_1,\ldots,K_K$)

- Dispersión intra-clase

$$\mathbf{S}_W = \sum_{i=1}^{K} \mathbf{S}_i \qquad \mathbf{S}_i = \sum_{\mathbf{x}^t \in K_i} \left(\mathbf{x}^t - \mathbf{m}_i\right)\left(\mathbf{x}^t - \mathbf{m}_i\right)^T$$

- Dispersión entre-clases

$$\mathbf{S}_B = \sum_{i=1}^{K} |K_i| \left(\mathbf{m}_i - \mathbf{m}\right)\left(\mathbf{m}_i - \mathbf{m}\right)^T \qquad \mathbf{m} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{m}_i$$

- Buscar **W** tal que maximice

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}$$

Los eigenvectores más grandes de $\mathbf{S}_W^{-1}\mathbf{S}_B$

LDA

Sea $X=\{x_i \mid i=1,\ldots,n\}$ con $x_i \in \Re^m$ un conjunto de n objetos almacenados en una matriz de nxm

1. Calcular las matrices S_W y S_B de mxm
2. Calcular $\Sigma_X = S_W^{-1} S_B$
3. Calcular los eigenvectores y eigenvalores de Σ_X

$$\Sigma_X V = \lambda V$$

4. La matriz V de mxm contiene los eigenvectores y el vector λ contiene los eigenvalores ordenados de mayor a menor, además $\sum_i^m \lambda_i = 1$

LDA

Cada fila de V es una de las componentes principales

Si conservamos todas no se reduce la dimensión de los datos

5. Se seleccionan en V_k (de mxk) las k<m primeras que capturen el K% de la varianza (los primeras k tales que la suma de sus eigenvalores sea ≥ K/100)

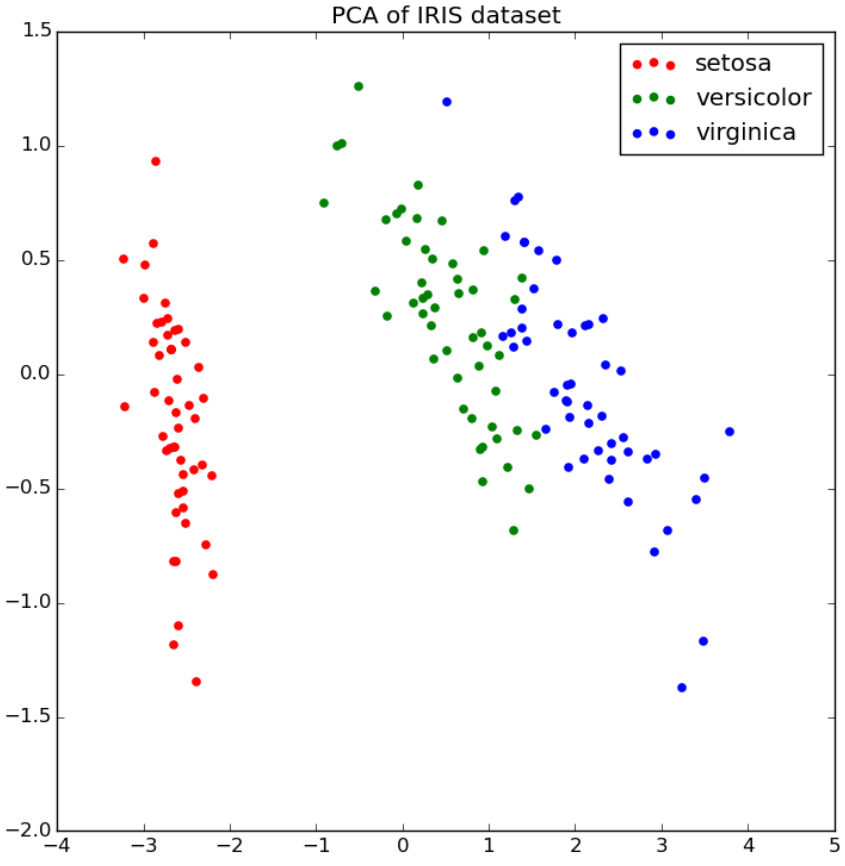
6. La nueva representación Y (de nxk) se obtiene como:

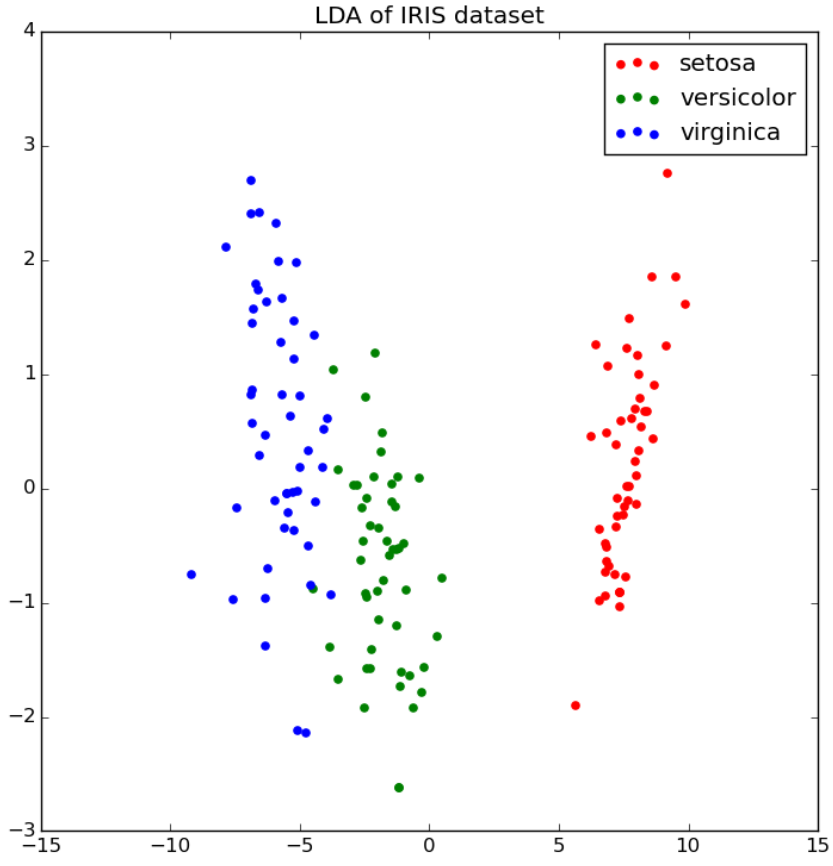
$$Y=XV_k$$

PCA vs LDA

Iris 150 objetos descritos por 4 atributos divididos en 3 clases

[Dua, D. and Graff, C. (2019). UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, Irvine, CA: University of California, School of Information and Computer Science.]





Escalamiento Multidimensional

- El objetivo es mapear los objetos n dimensionales originales a $\Re^d$, de modo que se preserven las distancias entre objetos

- Usualmente se toma d=2 para visualizar

Escalamiento Multidimensional

- Sea D es la distancia en el espacio original y d la distancia en $\Re^d$
- Sea o_i el mapeo en $\Re^d$ del objeto O_i
- El objetivo es encontrar un mapeo tal que:

$$\forall O_i, O_j : D(O_i, O_j) = d(o_i, o_j)$$

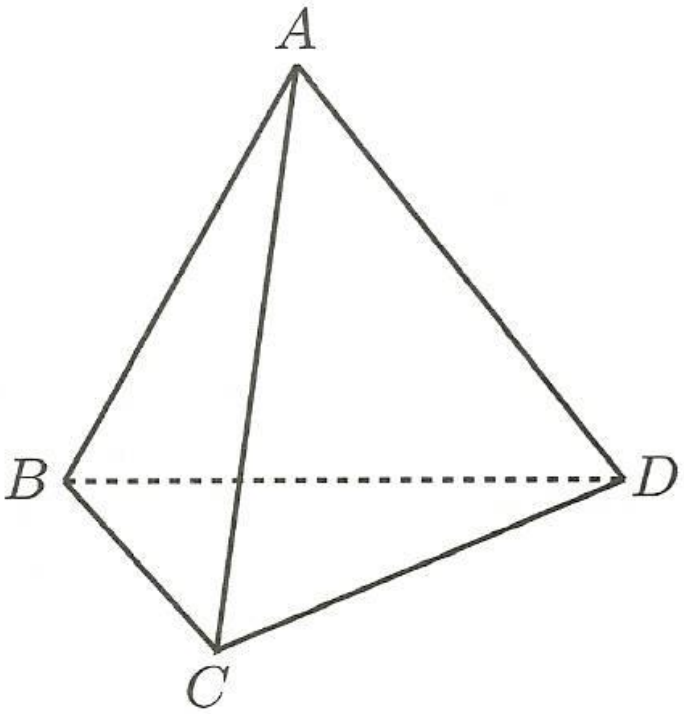
Escalamiento Multidimensional

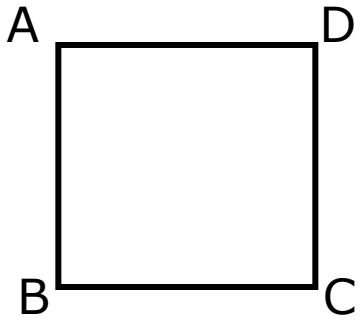
- En general el objetivo no se puede lograr
- Por lo tanto, lo que se busca es un mapeo tal que:

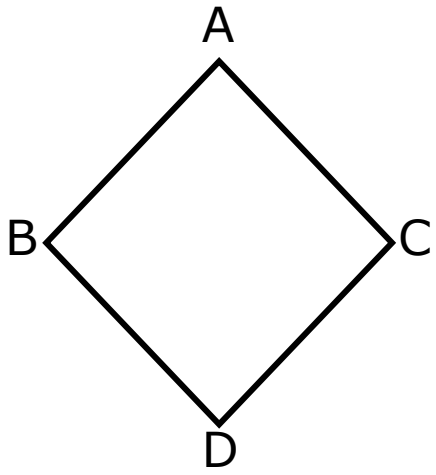
$$\forall O_i, O_j : D(O_i, O_j) \approx d(o_i, o_j)$$

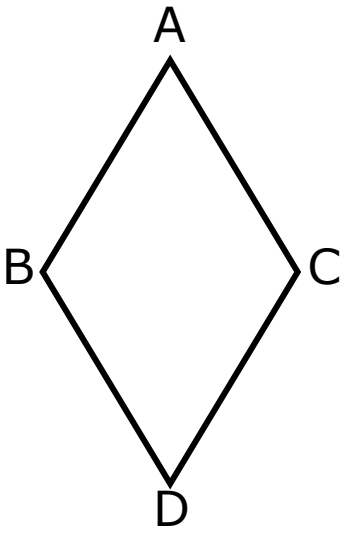
Escalamiento Multidimensional

Tetraedro









Escalamiento Multidimensional

¿Cómo decidir qué tan bueno es un mapeo ?

$$MSE = \frac{1}{m} \sum_{i,j} \left(D(O_i, O_j) - d(o_i, o_j) \right)^2$$

$$STRESS = \sqrt{\frac{\sum_{i,j} \left(D(O_i, O_j) - d(o_i, o_j) \right)^2}{\sum_{i,j} d(o_i, o_j)^2}}$$

Escalamiento Multidimensional

¿Cómo construir un mapeo?

- Utilizar un optimizador
  - Método del gradiente
  - Algoritmos genéticos
  - …

Programación genética

Idea

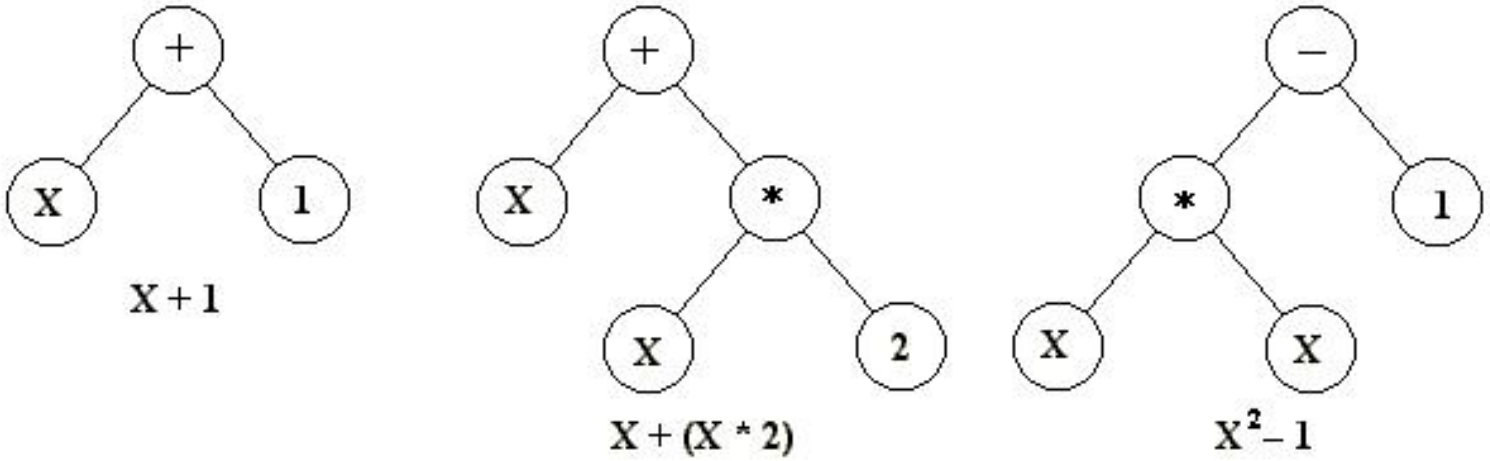
Utilizar algoritmos genéticos para generar programas que realicen una tarea

- Los individuos son programas
- La función de evaluación (fitness) debe evaluar qué tan bien un individuo (programa) resuelve la tarea objetivo

Programación genética

Representación de individuos

- Los individuos son programas usualmente expresiones matemáticas codificadas como árboles



Programación genética

Cruza (2 individuos)

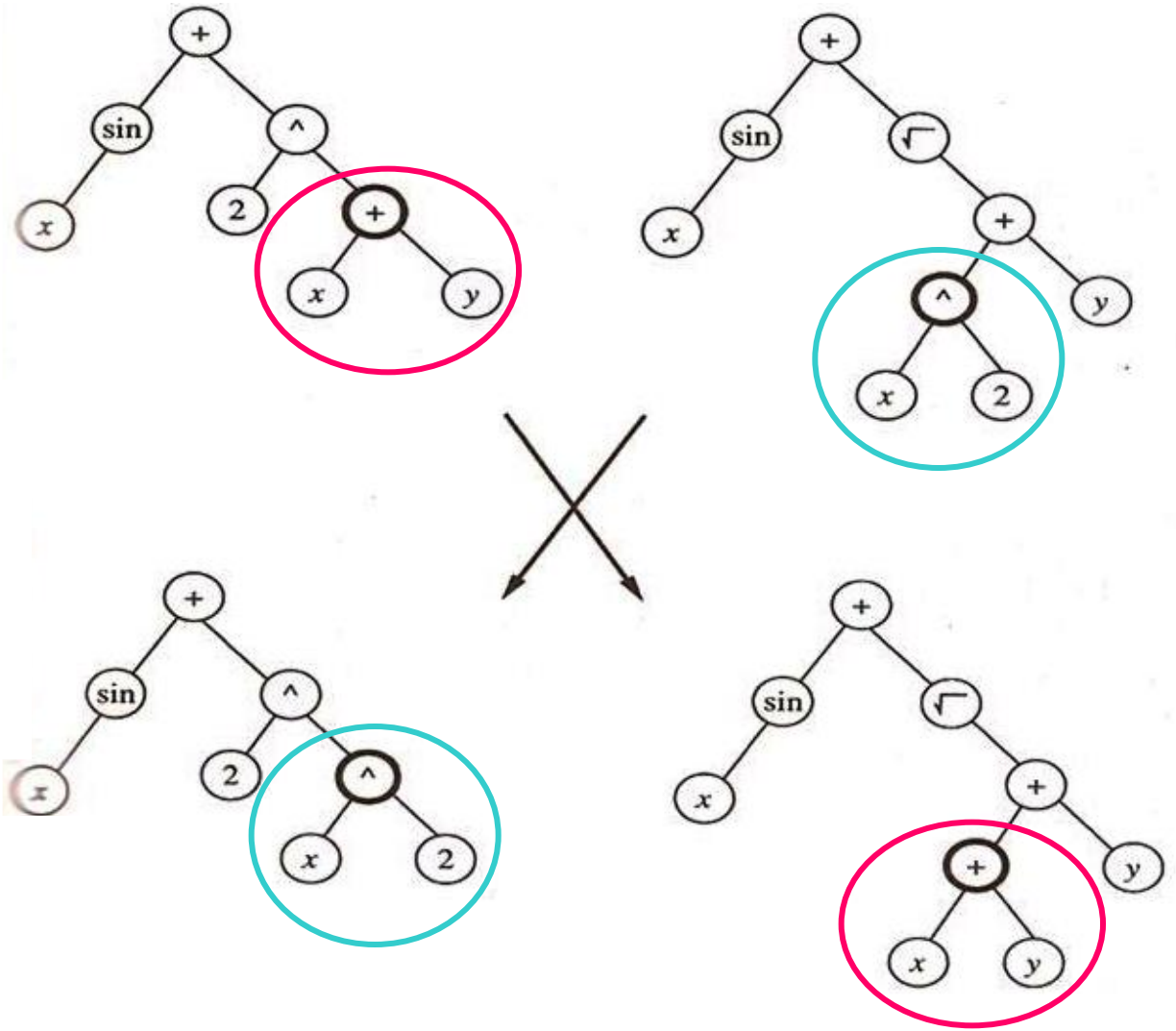
- Se selecciona un nodo de cada individuo y se intercambian

Mutación

- Se selecciona un nodo y se modifica

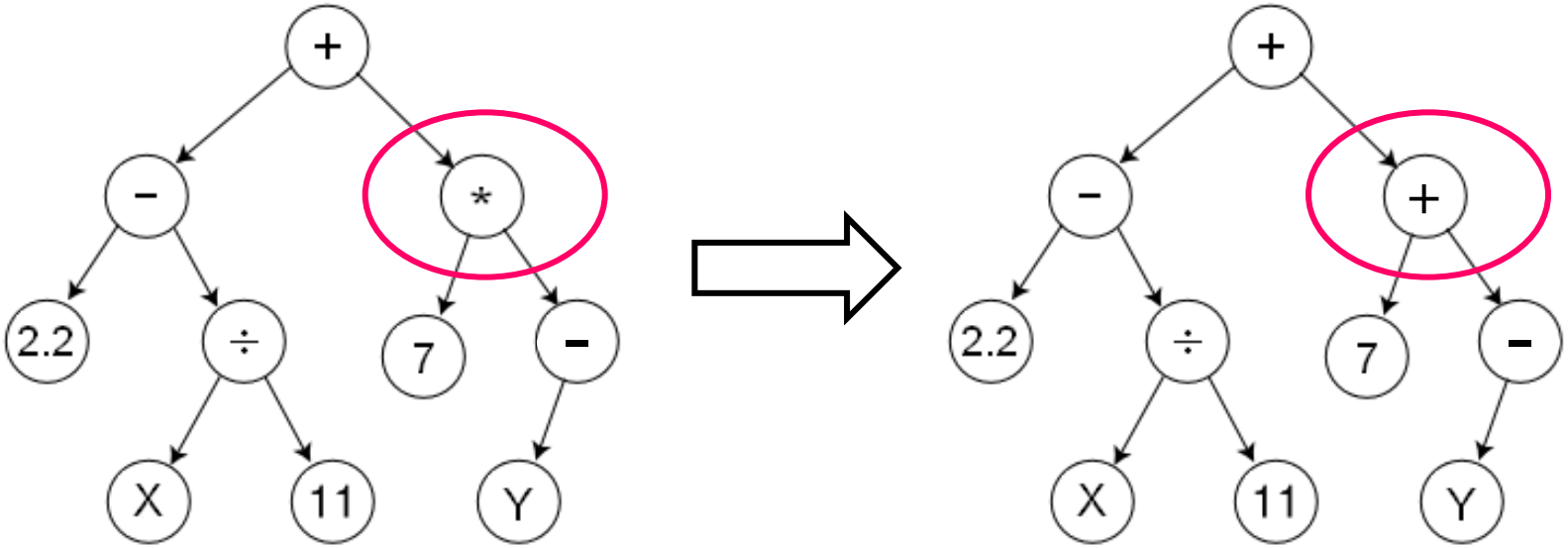
Programación genética

Cruza



Programación genética

Mutación



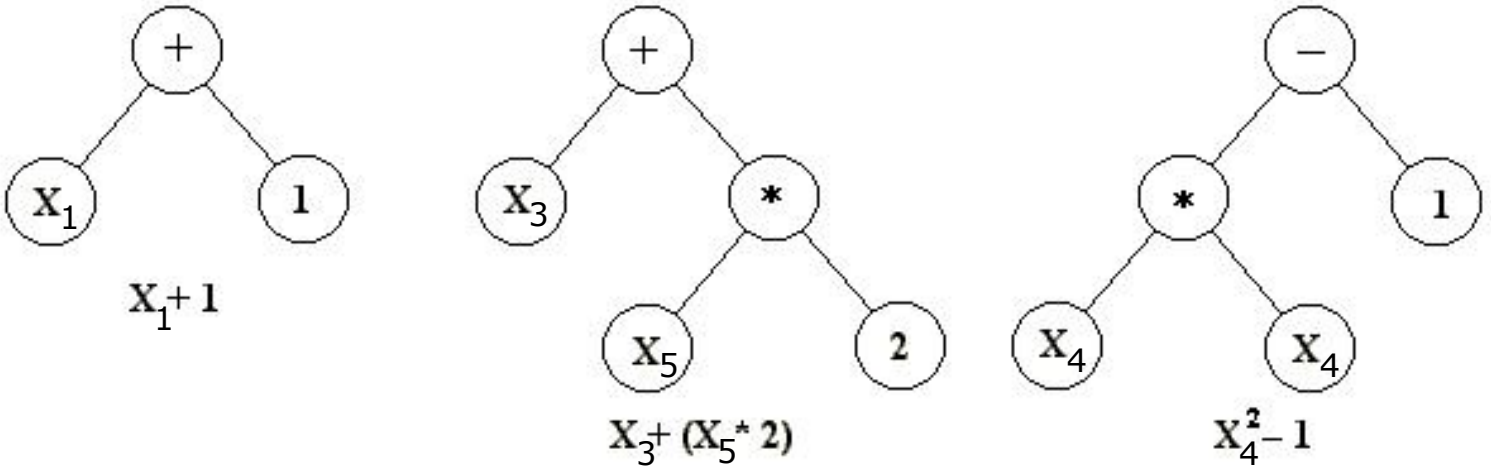
Programación genética

- Generar población inicial P
- Para i=1, ... , numGeneraciones
  - evalúa(P)
  - P2 = cruza(P)
  - evalúa(P2)
  - P3 = mutación (P∪P2)
  - evalúa(P3)
  - P = selecciona(P∪P2∪P3)
- salida = mejorElemento(P)

Programación genética

¿Cómo utilizar PG para extraer atributos?

- Cada individuo representa una transformación de los n atributos originales en k nuevos atributos
- Los individuos son colecciones de k árboles (funciones)
  - Sólo se utilizan operadores de +, -, *
  - Los operandos son constantes o alguna de las variables originales



Programación genética

¿Cómo utilizar PG para extraer atributos?

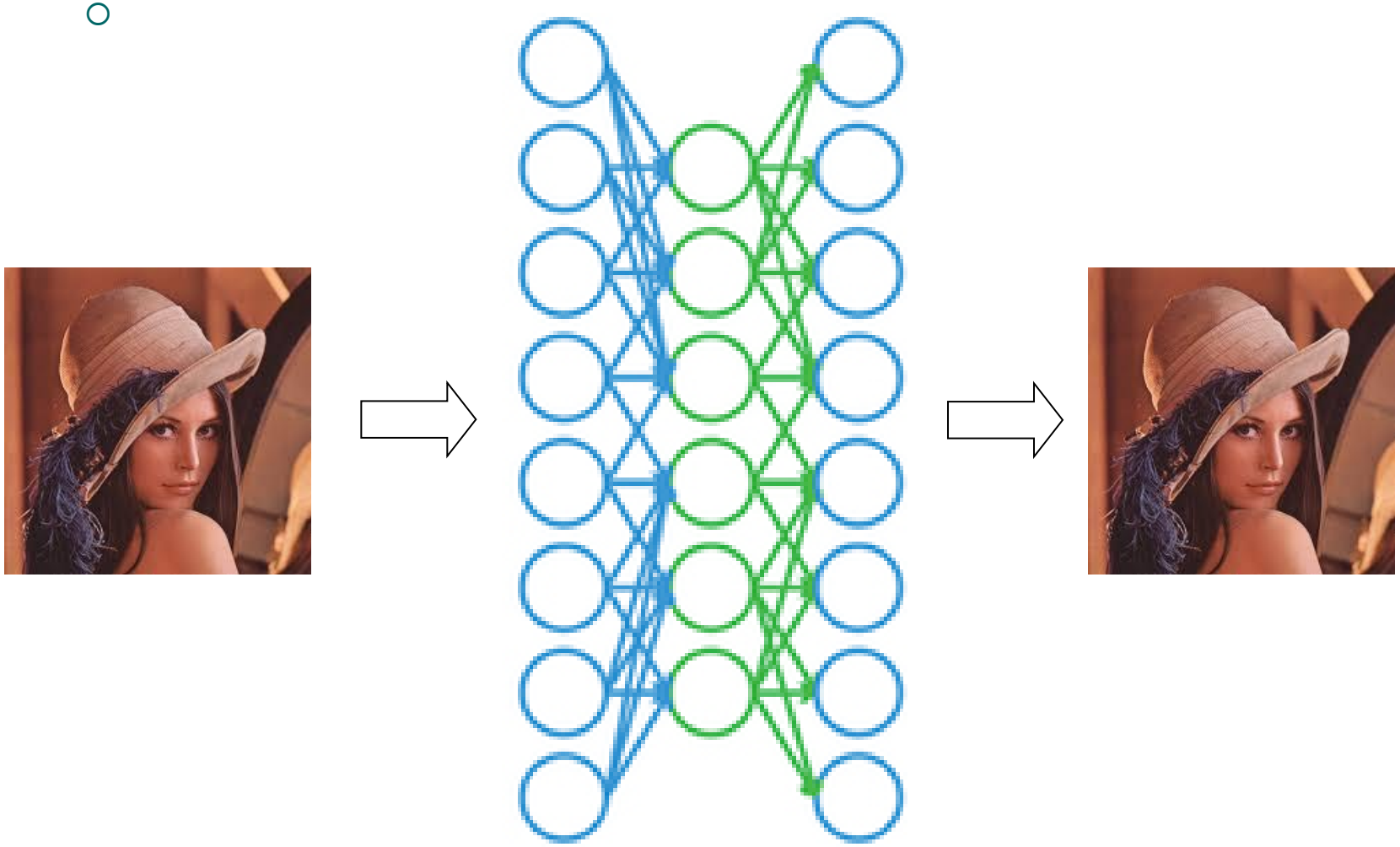
- La **cruza** consiste en intercambiar un árbol seleccionado al azar entre dos individuos
- La **mutación** consiste en eliminar un árbol seleccionado aleatoriamente y agregar otro generado aleatoriamente
- La **función de fitness** para un individuo consiste en aplicar la transformación indicada y utilizar un clasificador.
  - La evaluación de individuo será la calidad de los resultados de clasificación, evaluados con alguna medida de calidad de clasificación

Autoencoders

Idea

- Utilizar una red neuronal de 3 capas en la cual las capas de entrada y de salida son del mismo tamaño (tantas neuronas como variables originales) y la capa central es más pequeña (tantas neuronas como variables a generar)
- Se entrena la red con la muestra para que la salida sea igual a la entrada (usualmente con retropropagación)
- La salidas de la capa intermedia son los valores de las nuevas variables

Autoencoders



Autoencoders

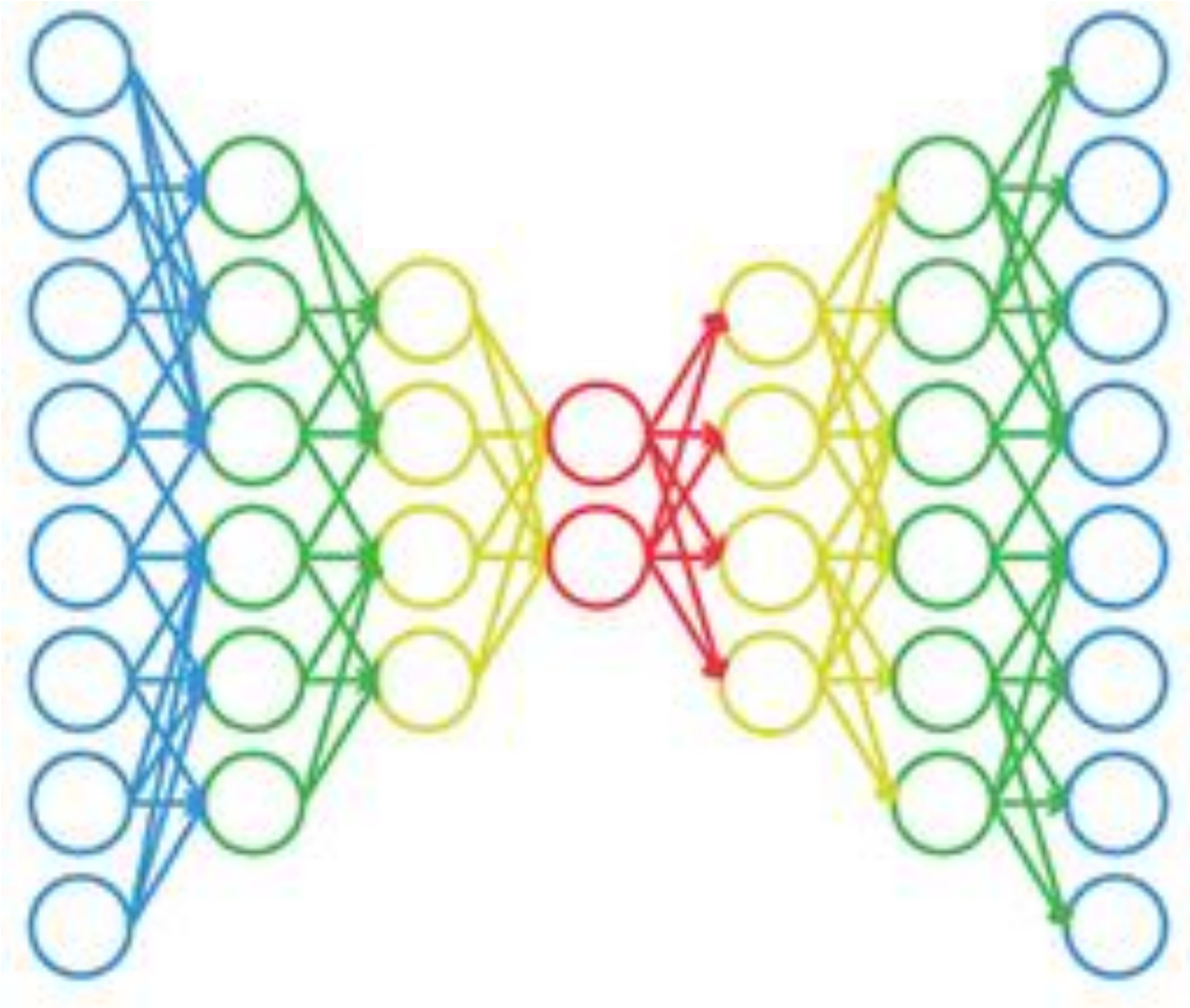
Problema

- La capa oculta no puede ser muy pequeña en relación con las de entrada y salida, pues en ese caso no es capaz de “concentrar” adecuadamente la información de la entrada

Solución

- Agregar múltiples capas ocultas que van disminuyendo paulatinamente de tamaño y después aumentando en la misma proporción (Deep Learning)

Autoencoders

- 

Autoencoders

Problema

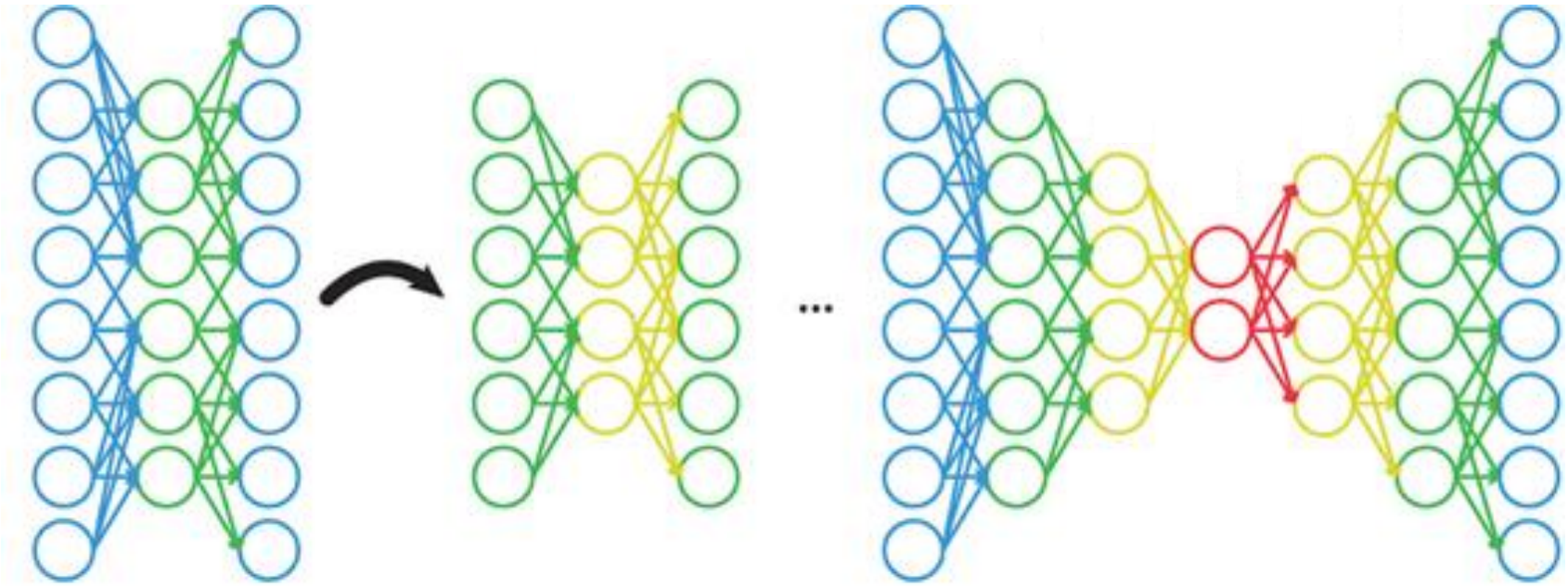
- Una red tan grande es muy costosa de entrenar y el ajuste de los pesos no es muy bueno.

Solución

- Entrenar una a una las capas ocultas

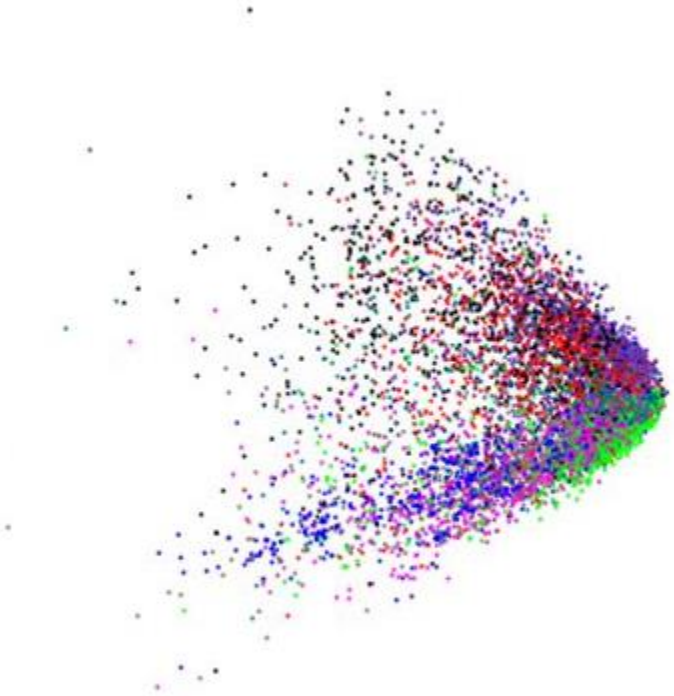
Autoencoders

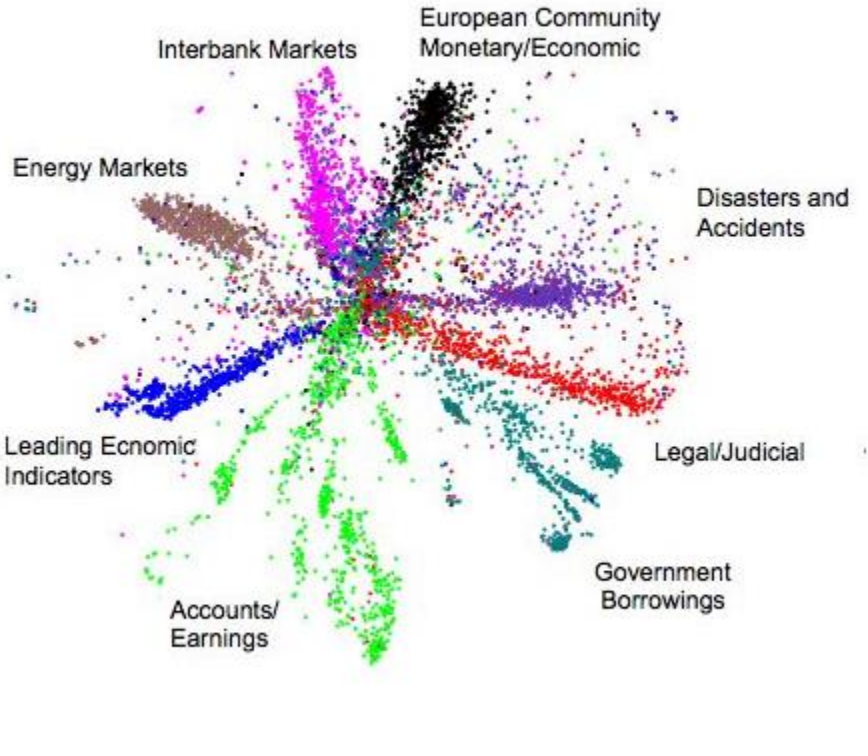
- Entrenar capa por capa



Autoencoders

- Ejemplo: PCA vs. Autoencoder en documentos





Evaluación de extractores de variables

- Utilizando un clasificador
  - Seleccionar un conjunto de bases de datos
  - Utilizar algún método de validación aplicando extracción+clasificación
  - Utilizar alguna medida de evaluación de calidad de clasificación

Extracción de Atributos

Dr. Jesús Ariel Carrasco Ochoa
ariel@inaoep.mx
Oficina 8311