

# Tema 4.3

## Proceso de entrenamiento

Miguel Ángel Martínez del Amor

Deep Learning

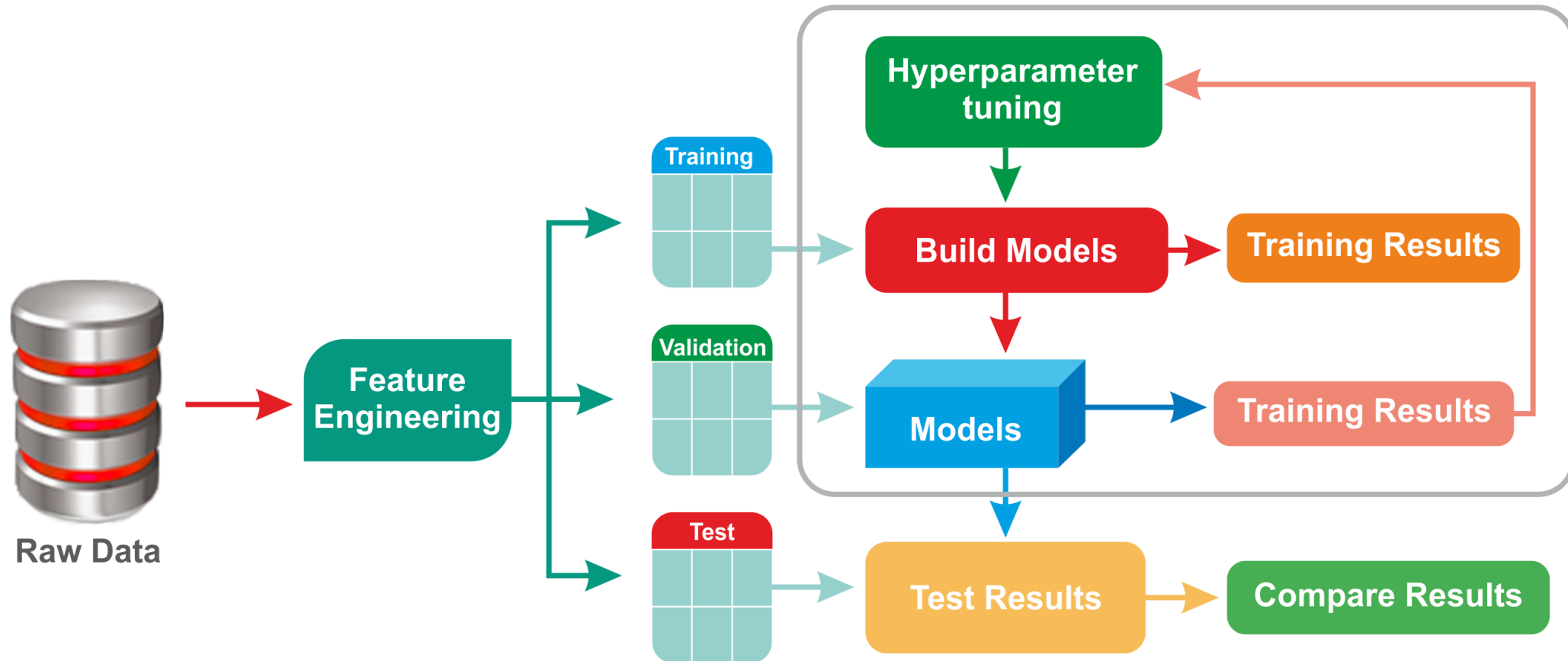
Departamento Ciencias de la Computación e Inteligencia Artificial

Universidad de Sevilla

# Contenido

- El proceso de aprendizaje
- Optimización de hiperparámetros

# El proceso de aprendizaje



# El proceso de aprendizaje

## 1. Primera configuración

- *Preprocesamiento de datos, aumento de datos*
- *Elección de una arquitectura, funciones de activación, inicialización de pesos*
- *Elección de método de regularización*
- *Elección de función de coste y método de optimización*

## 2. Dinámica de entrenamiento

- *Monitorización del entrenamiento (lo trabajaremos en las prácticas)*
- *Optimización de parámetros*
- *Optimización de hiperparámetros*

## 3. Evaluación

- *Ensamblado de modelos*

# Optimización de hiperparámetros

## Model Design

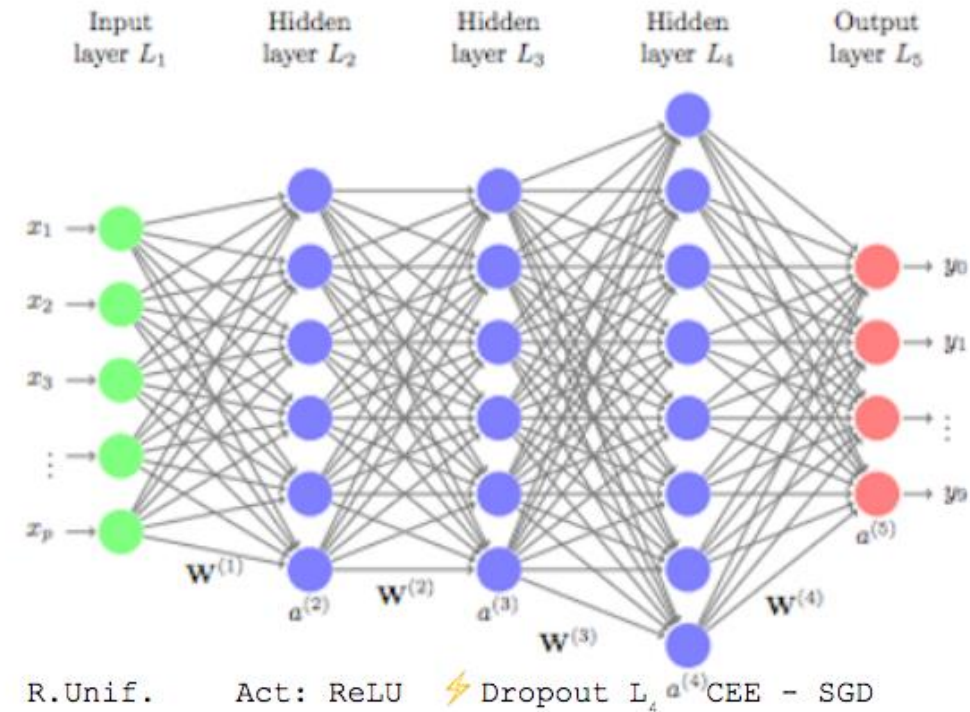
- Weight init.: Random Uniform
- Act.: ReLU
- Loss: CEE
- # Hidden Layers: 3
- # Units per layer  $\{p, p+1, p+1, p+3, 10\}$
- Optimizer: SGD
- Dropout layer:  $L_4$

## Hyperparameters

- Learning rate
- Dropout Rate
- Batch size

## Model Parameters

- $W^{(1)} \Rightarrow W^{(4)}$



# Optimización de hiperparámetros

- **Hiperparámetros** con los que jugar:
  - La **arquitectura** de la red
  - El **learning** rate, su decaimiento y tipo de actualización
  - El **momentum** (si se escoge este método con SGD)
  - **Regularización** (L2/Dropout)
  - Tamaño del **batch**
- **Métodos:**
  - Manual: Babysitting o “prueba y error”
  - Búsqueda paralela: Grid Search, Random Search, Bayesian Optimization

# Optimización de hiperparámetros: babysitting

- Comenzar **observando la pérdida** obtenida del modelo:
  - Al comienzo, desactivar la regularización
  - Observar que la pérdida incrementa al introducir regularización
  - Comprobar que podemos hacer sobreajuste sobre unos cuantos datos
- Comenzar con un **factor pequeño de regularización y encontrar el learning rate** que decrementa la pérdida:
  - Si la pérdida no decrece: learning rate muy bajo
  - Si la pérdida explota: learning rate demasiado alto
  - Probar valores entre  $10^{-3}$  y  $10^{-5}$ .

# Optimización de hiperparámetros: babysitting

- Hacer validación cruzada:
  - Primero con pocas épocas para ver rápidamente los valores que funcionan bien.
  - Probar valores en escala logarítmica! Es decir:  $lr = 10^{-3}, 10^{-4}, 10^{-5}$
  - Después de elegir
    - Probar más épocas.
    - Afinar los valores de los hiperparámetros (esta vez no logarítmicamente)

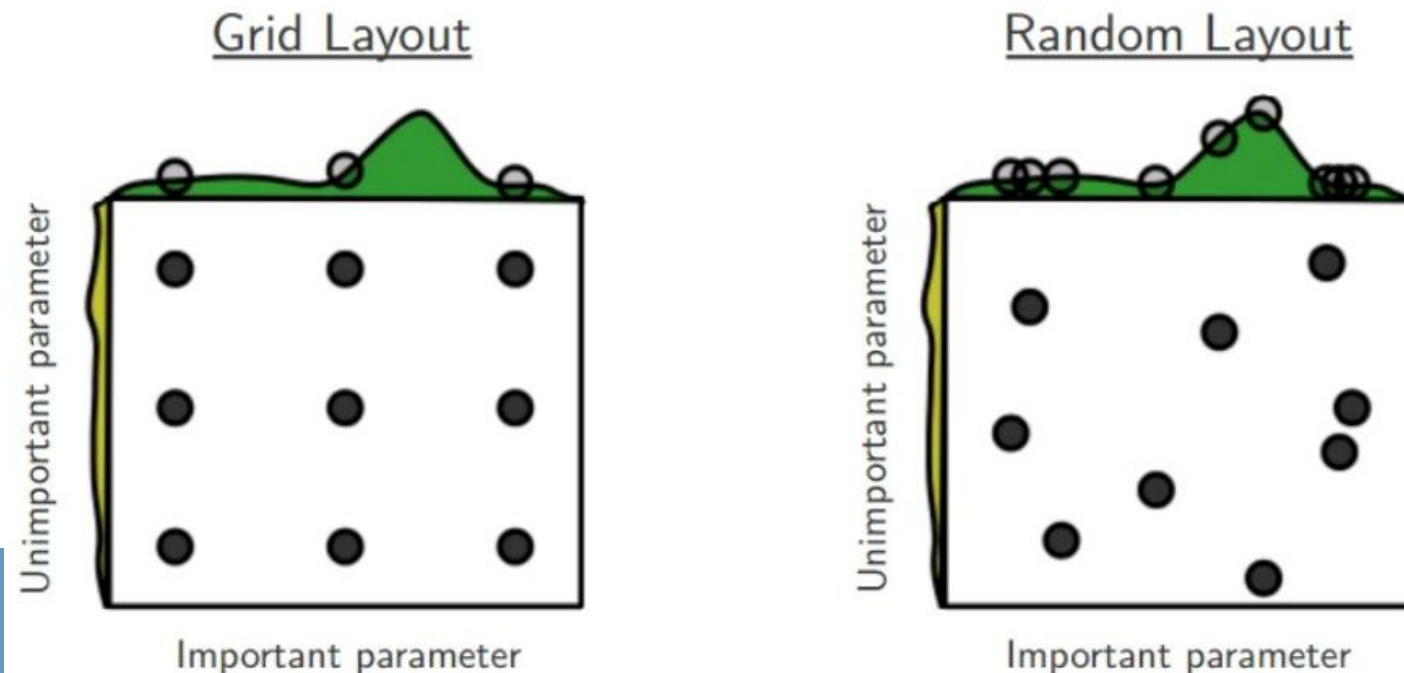


# Optimización de hiperparámetros: Grid search

- Probar configuraciones en paralelo mediante un grid:
  - Definir un grid de n dimensiones, donde cada dimensión es un hiperparámetro.
  - Para cada dimensión, definir un rango de posibles valores. Por ejemplo, batch size = 4, 6, 8, 16, 32, 64, 128; learning rate = 0,01; 0,001; 0,0001
  - Busca para todas las posibles configuraciones y espera los resultados para encontrar la mejor combinación.

# Optimización de hiperparámetros: random search

- Introducido en [[Bergstra y Bengio, 2012](#)]:
- La diferencia real está en el primer paso, random search escoge los puntos aleatoriamente del espacio de configuración



# Recapitulación

- Hemos repasado de nuevo el **proceso** a seguir para entrenar un modelo desde cero.
- Hemos visto algunos métodos para **optimizar** los **hiperparámetros**:
  - Prueba y error, o **babysitting**
  - Paralelos: en **grid** o **random**.