

Lecture 01

Introducción a Inteligencia Artificial



Temas de la Lección

- 1. Panorama general del curso**
2. ¿Qué es el aprendizaje automático?
3. Las categorías generales del aprendizaje automático
4. Flujo de trabajo del aprendizaje supervisado
5. Notación y jerga necesarias en aprendizaje automático
6. Aspectos prácticos y herramientas



Temas Planeados

Módulo 1: Fundamentos de IA y DL (Semana 1-4)

- Modelos lineales y repaso de cálculo y álgebra lineal
- Funciones de costo e interpretación probabilística
- Back-propagation en forma tensorial
- Algoritmos de optimización para DL

Módulo 2: Visión por computador (Semana 5-8)

- Redes convolucionales
- Arquitecturas modernas y transfer learning
- Modelos generativos: Autoencoders, VAEs, GANs, Diffusion

Módulo 3: Procesamiento de lenguaje natural – NLP (Semana 9-12)

- Transformers (BERT, GPT)
- Fine-tuning de LLMs
- Evaluación de LLMs, seguridad y alineamiento
- Sistemas multi-modales (CLIP)



Temas Planeados

Módulo 4: Aplicaciones de IA moderna e Ingeniería de IA (Semana 13-16)

- DeepRL
- APIs de IA generativa e Ingeniería de Prompts
- RAG
- Agentes
- Despliegue de sistemas de IA



Evaluación:

Módulo 1: Implementación de módulo de entrenamiento de redes (20% - Individual)

Fecha de asignación: Semana 4

Fecha de entrega: Semana 6

Módulo 2: Implementación de sistema de visión por computador (20% - Individual)

Fecha de asignación: Semana 8

Fecha de entrega: Semana 10

Módulo 3: Implementación de sistema de NLP (20% - Individual)

Fecha de asignación: Semana 12

Fecha de entrega: Semana 14

Módulo 4: Implementación de sistema con GenAI (20% - Grupal)

Fecha de asignación: Semana 14

Fecha de entrega: Semana 16



Evaluación:

Proyecto: Aplicación de IA (20% - Grupal)

Anteproyecto: Idea de proyecto, justificación, objetivo general y específicos (5%)

Fecha de asignación: Semana 12

Fecha de entrega: Semana 13

Entrega final: Marco teórico, marco experimental, resultados, conclusiones (15%)

Fecha de asignación: Semana 14

Fecha de entrega: Semanas 16-18



¿Qué es el Aprendizaje Automático?

Un breve panorama antes de adentrarnos en el Aprendizaje Profundo:

1. Panorama general del curso
- 2. ¿Qué es el aprendizaje automático?**
3. Las categorías generales del aprendizaje automático
4. Flujo de trabajo del aprendizaje supervisado
5. Notación y jerga necesarias
6. Aspectos prácticos y herramientas

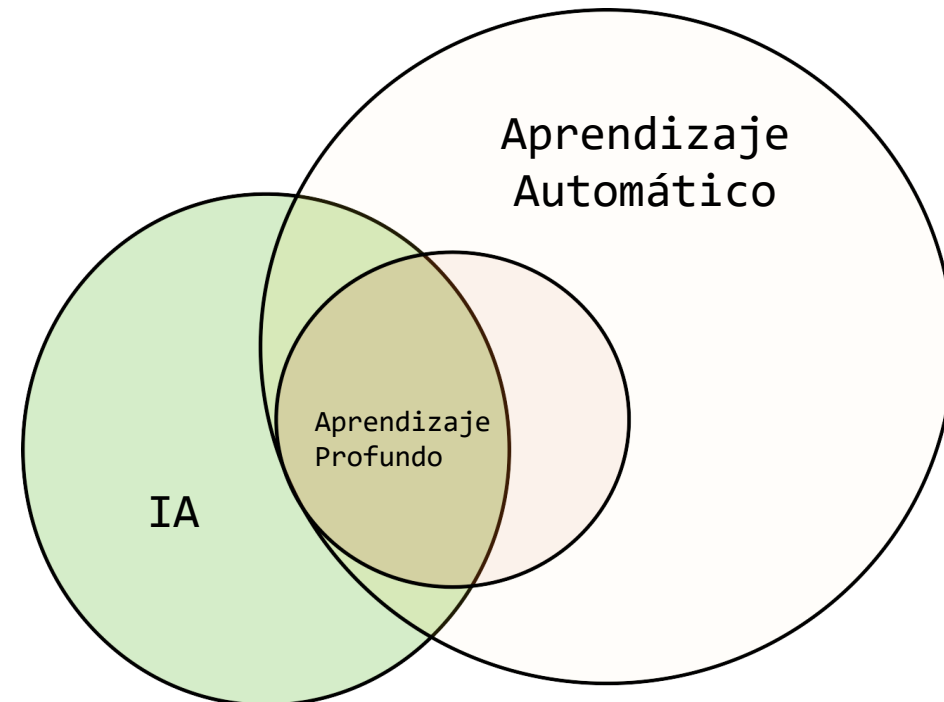


Tipos de Inteligencia Artificial

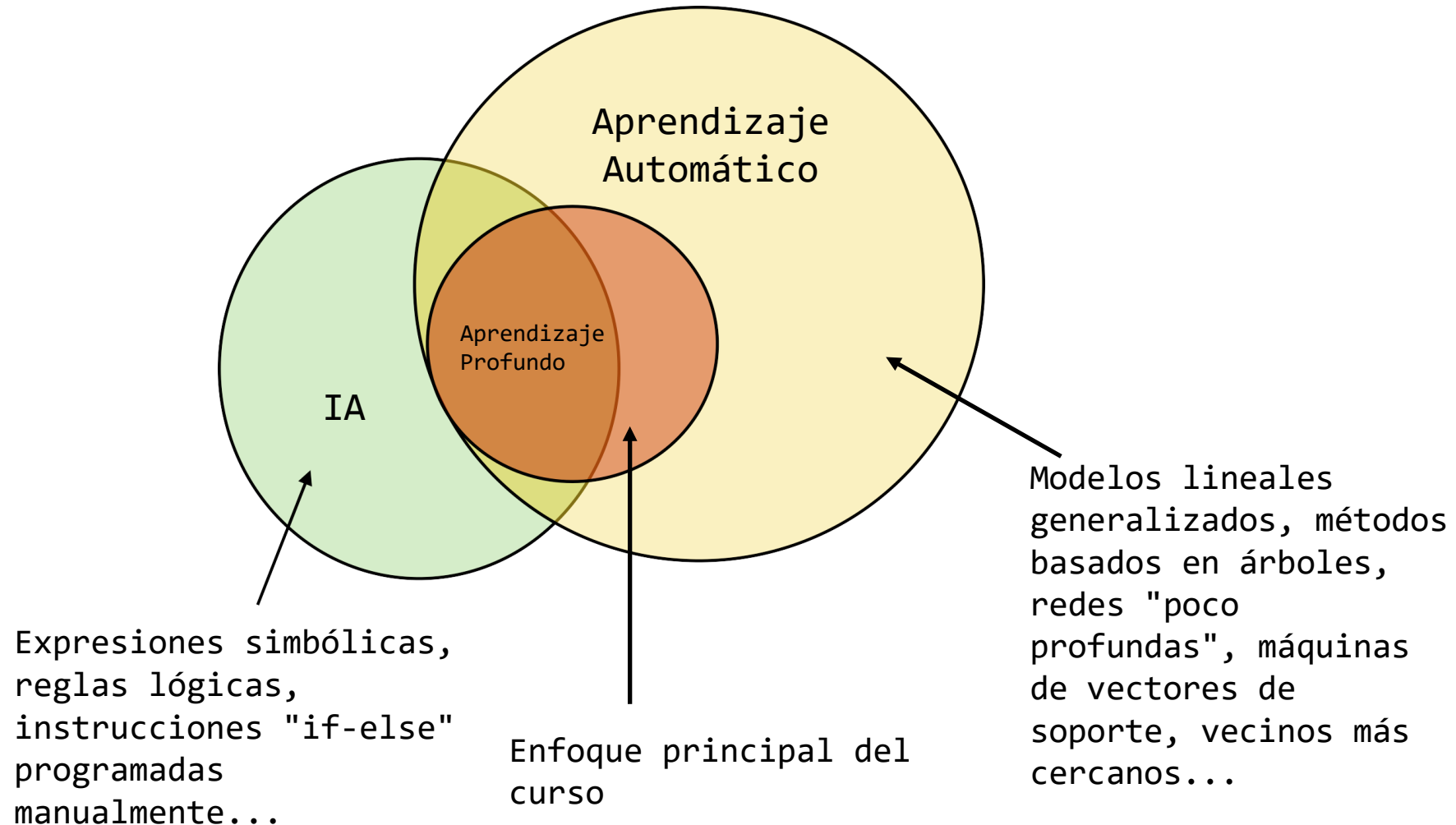
Inteligencia Artificial (IA): Subcampo de la informática centrado en resolver tareas que los humanos realizan con facilidad.

IA Estrecha (narrow): Diseñada para resolver tareas específicas de forma eficiente.

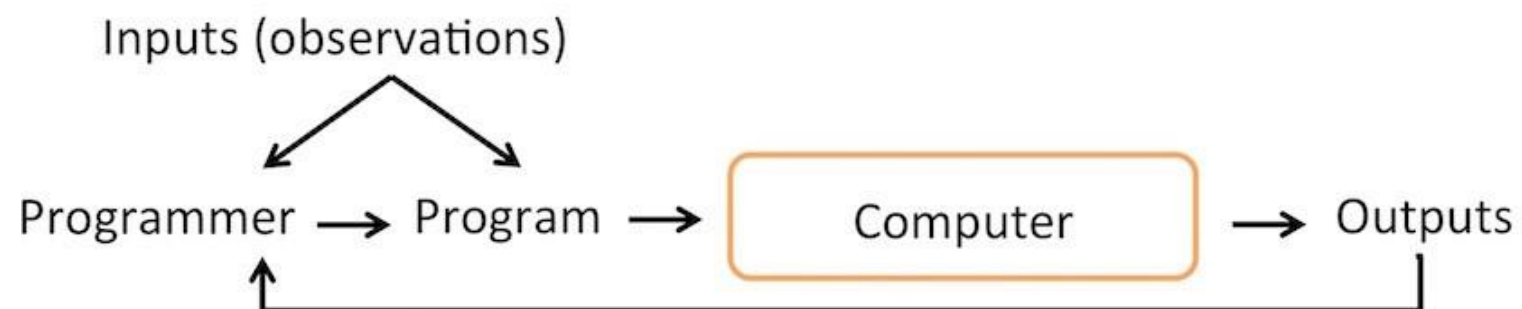
Inteligencia Artificial General (AGI): IA capaz de abordar múltiples tareas y comportarse como un humano en diversos contextos.



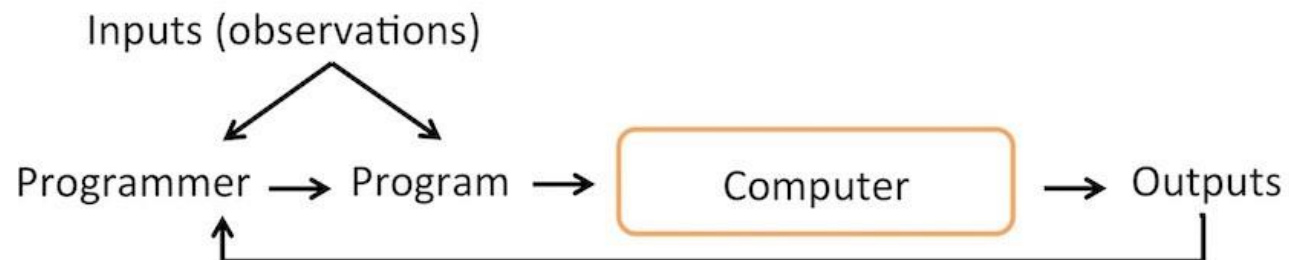
De qué trata este curso



The Traditional Programming Paradigm



The Traditional Programming Paradigm



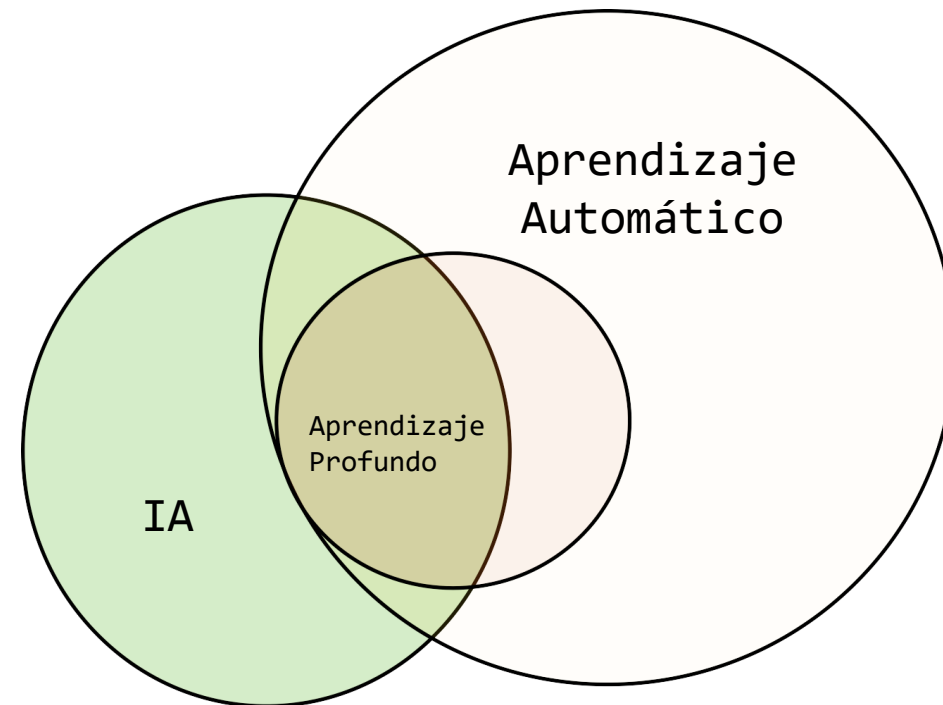
Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed
– Arthur Samuel (1959)

Machine Learning

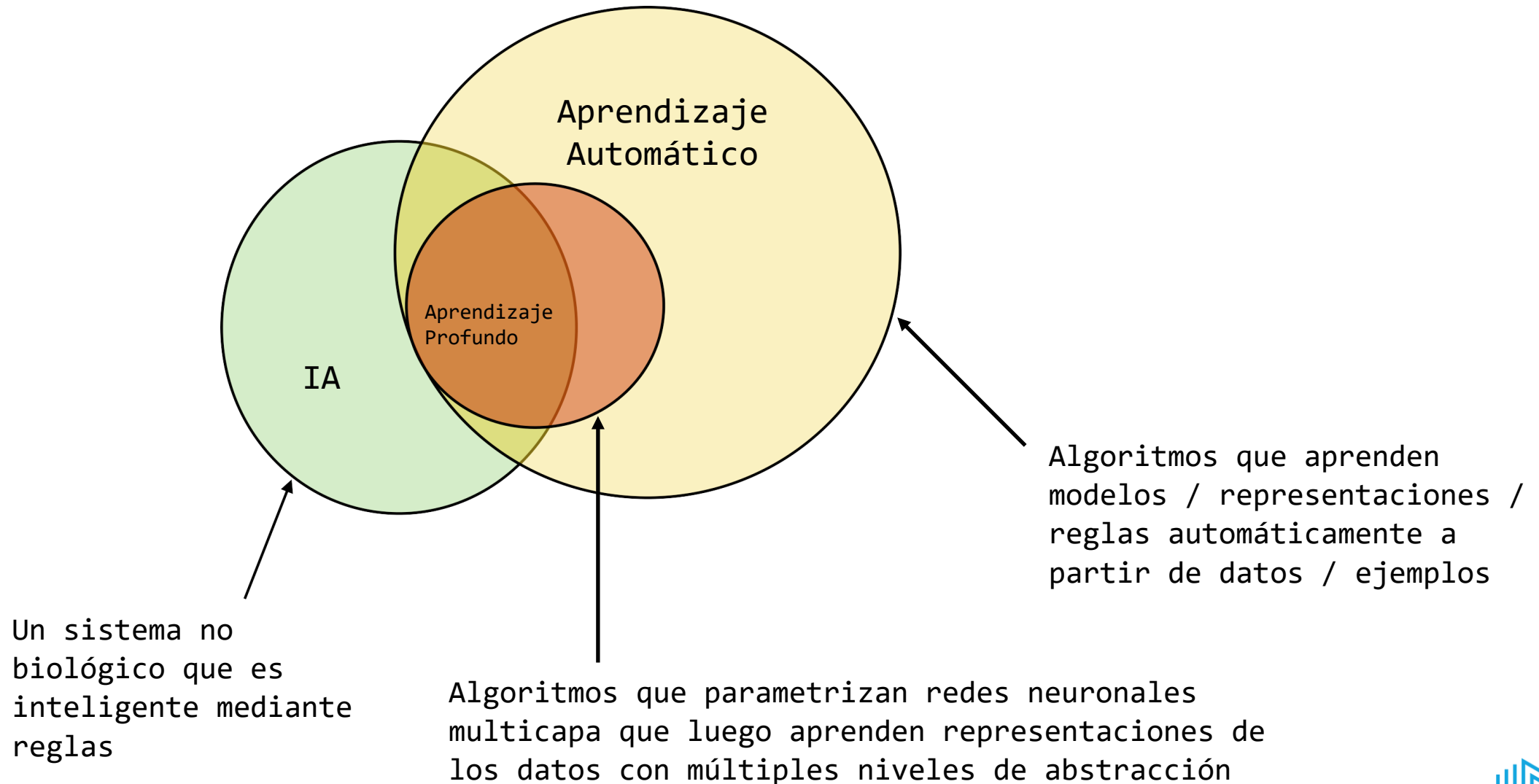


No todos los sistemas de IA utilizan Aprendizaje Automático

Deep Blue utilizaba chips VLSI personalizados para ejecutar el algoritmo de búsqueda alfa-beta en paralelo, un ejemplo de GOFAI (Good Old-Fashioned Artificial Intelligence - Inteligencia Artificial Tradicional).

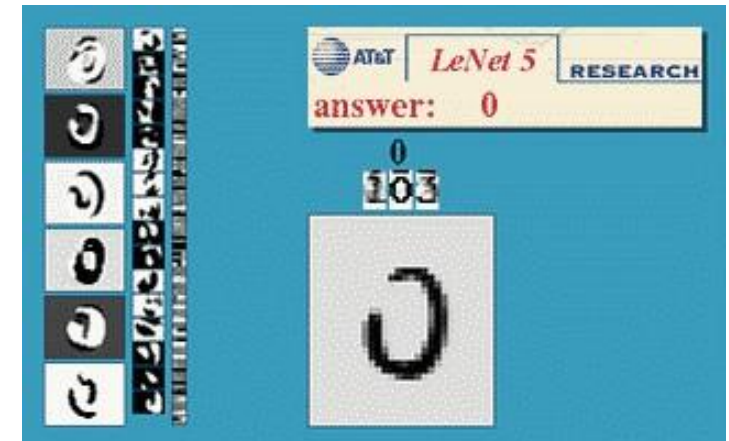


Ejemplos de las tres "áreas" relacionadas



Algunas aplicaciones del Aprendizaje Automático / Aprendizaje Profundo

- Detección de correo no deseado
- Detección y coincidencia de huellas dactilares / rostros (ej. teléfonos)
- Búsqueda web (DuckDuckGo, Bing, Google)
- Predicciones deportivas
- Oficina de correos (clasificación de cartas por códigos postales)
- Cajeros automáticos (lectura de cheques)
- Fraude con tarjetas de crédito
- Predicciones bursátiles)



Source: <http://yann.lecun.com/exdb/lenet/>



Más aplicaciones:

- Asistentes inteligentes (Apple Siri, Amazon Alexa, ...)
- Recomendaciones de productos (Netflix, Amazon)
- Autos autónomos (Uber, Tesla)
- Traducción automática (Google Translate)
- Análisis de sentimiento
- Diseño de fármacos
- Diagnósticos médicos
- ...



Las 3 Categorías Generales del Aprendizaje Automático

(Esto también aplica al Aprendizaje Profundo)

1. Panorama general del curso
2. ¿Qué es el aprendizaje automático?
- 3. Las categorías generales del aprendizaje automático**
4. Flujo de trabajo del aprendizaje supervisado
5. Notación y jerga necesarias
6. Aspectos prácticos y herramientas



Las 3 Categorías Generales del Aprendizaje Automático (y Aprendizaje Profundo)

Aprendizaje Supervisado

- Datos etiquetados
- Retroalimentación directa
- Predecir resultados / el futuro

Aprendizaje No Supervisado

- Sin etiquetas/objetivos
- Sin retroalimentación
- Encontrar estructura oculta en los datos

Aprendizaje por Refuerzo

- Proceso de toma de decisiones
- Sistema de recompensas
- Aprender una secuencia de acciones



Aprendizaje Supervisado es la subcategoría más grande del aprendizaje automático

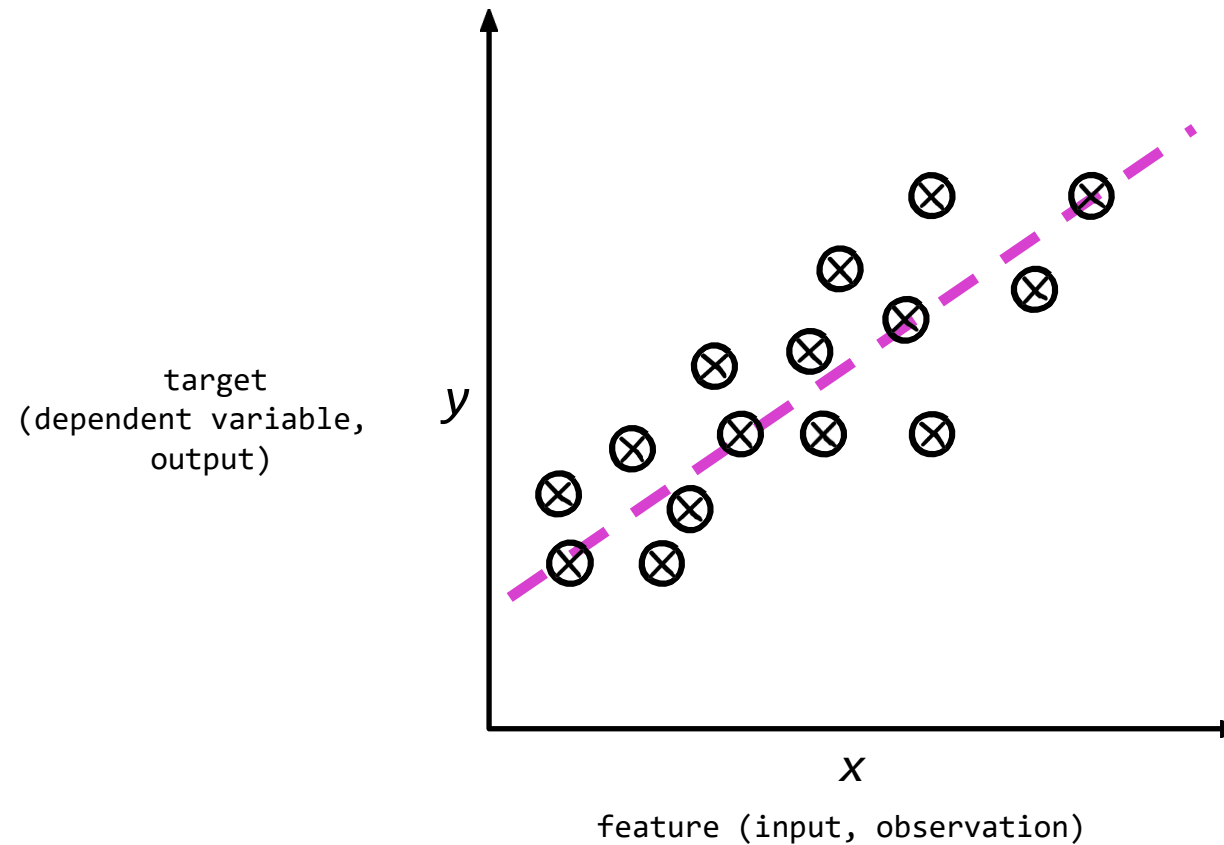
Aprendizaje Supervisado

- Utiliza Datos etiquetados
- Proporciona Retroalimentación directa
- Permite predecir resultados / el futuro

Source: Raschka and Mirjalily (2019). Python Machine Learning, 3rd Edition



Aprendizaje Supervisado 1: Regresión



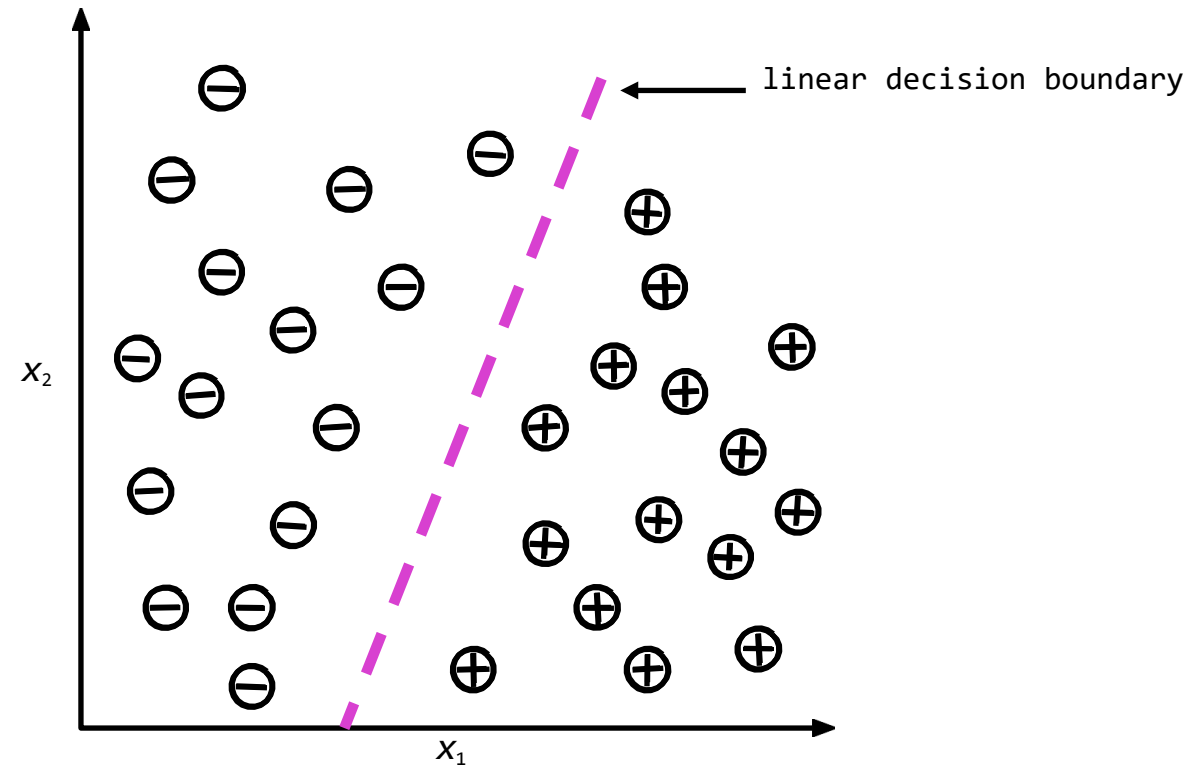
Source: Raschka and Mirjalili (2019). *Python Machine Learning, 3rd Edition*



Aprendizaje Supervisado 2: Clasificación

Ejemplo de clasificación binaria con dos características(variables independientes o predictores)

¿Cuáles son las
etiquetas de clase (y's)?



Aprendizaje Supervisado 3:

Regresión Ordinal

- La regresión ordinal también se conoce como clasificación ordinal o ranking (aunque el ranking es un poco diferente)

Dependencia de orden, como en la regresión métrica, pero sin una distancia métrica

Valores discretos, como en la clasificación, pero con dependencia de orden

$$r_K > r_{K-1} > \dots > r_1$$

E.g., movie ratings: *great > good > okay > for genre fans > bad*



Aprendizaje Supervisado 3: Regresión Ordinal?

- **Ranking:** Predecir el orden correcto (Pérdida = 0 si el orden es correcto; por ejemplo, clasificar una colección de películas según su "calidad")



- **Regresión Ordinal:** Predecir la etiqueta correcta (ordenada) (Por ejemplo, la edad de una persona en años; aquí se considera el envejecimiento como un proceso no estacionario)

Excerpt from the UTKFace dataset
<https://susanqq.github.io/UTKFace/>



Aprendizaje Supervisado 3:

Regresión Ordinal

- **Ranking:** Predecir el orden correcto (Pérdida = 0 si el orden es correcto; por ejemplo, clasificar una colección de productos según su "tamaño")



- **Regresión Ordinal:** Predecir la etiqueta correcta (ordenada) (Por ejemplo, la edad de una persona en años; aquí se considera el envejecimiento como un proceso no estacionario)

Excerpt from the UTKFace dataset
<https://susanqq.github.io/UTKFace/>



La Segunda Subcategoría del Aprendizaje Automático (y del Aprendizaje Profundo)

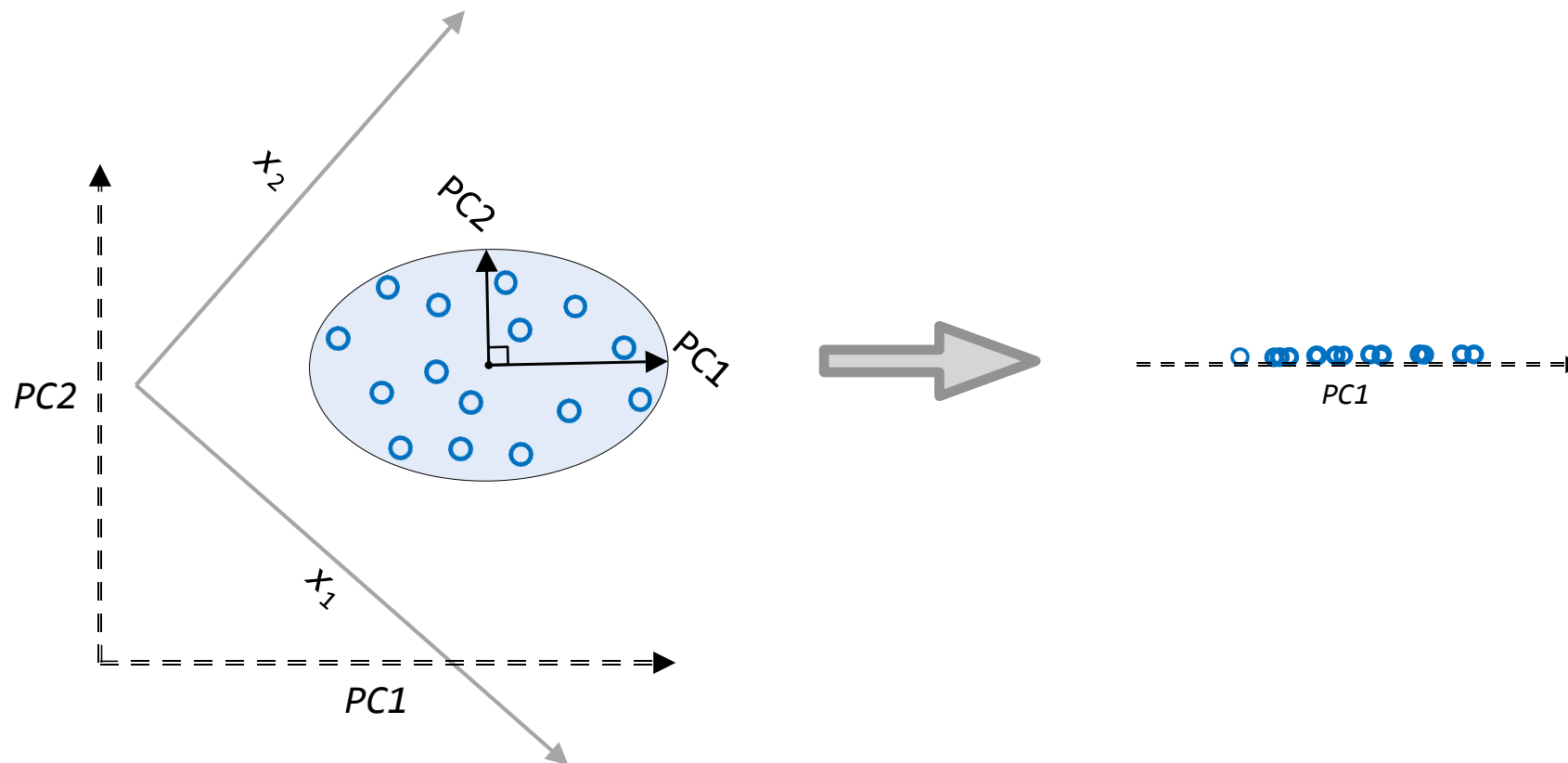
Aprendizaje No Supervisado

- Sin etiquetas/objetivos
- Sin retroalimentación
- Encontrar estructura oculta en los datos



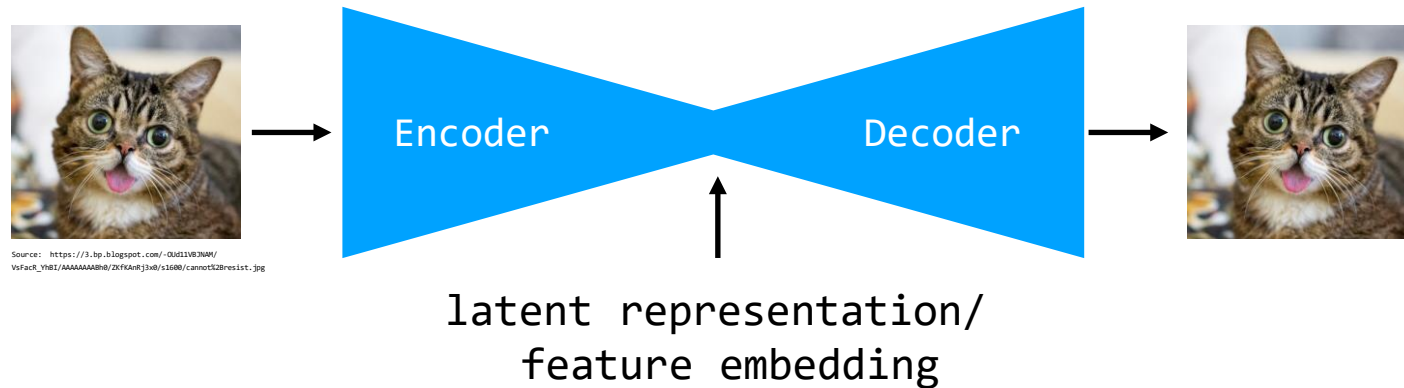
Aprendizaje No Supervisado 1: Aprendizaje de Representaciones / Reducción de Dimensionalidad

E.g., Principal Component Analysis (PCA)



Aprendizaje no supervisado 1: Aprendizaje de representaciones / Reducción de dimensionalidad

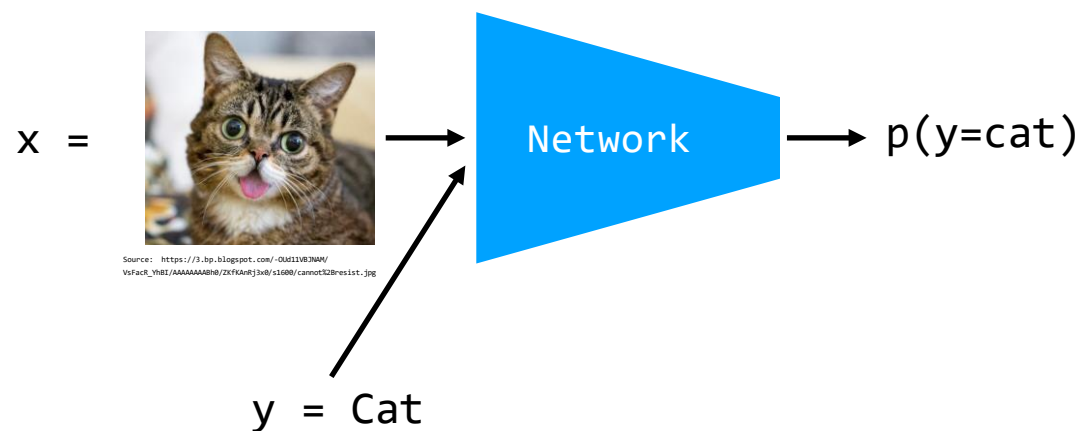
E.g.,
Autoencoders



(se cubrirán más adelante en este curso)

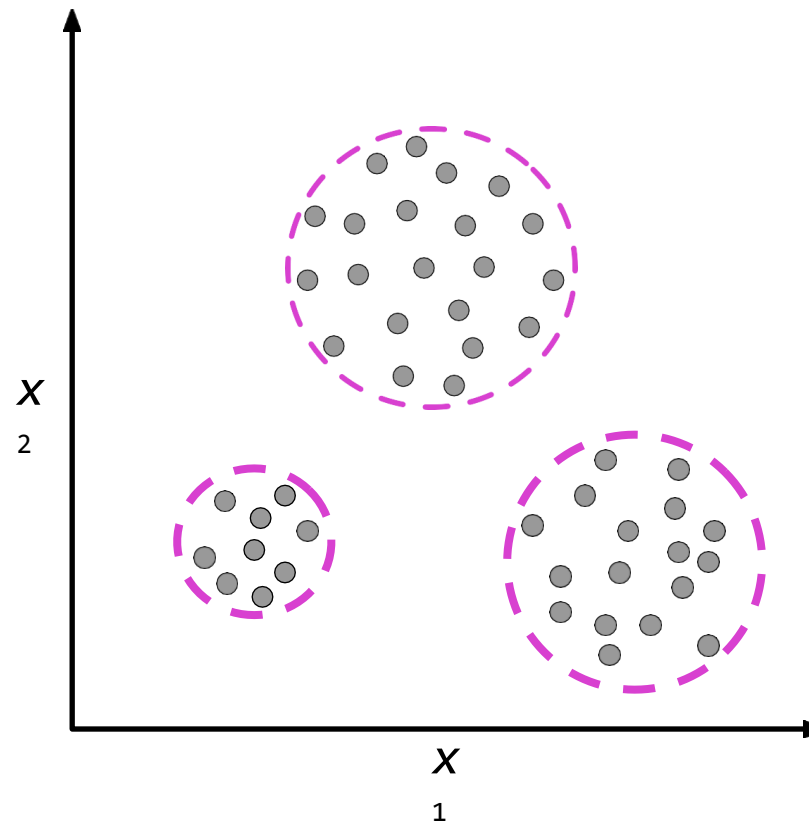


Recordatorio: La clasificación funciona así



Aprendizaje no supervisado 2: Agrupamiento (Clustering)

Asignación de pertenencias a grupos para ejemplos sin etiquetas
(instancias, puntos de datos)



Aprendizaje por refuerzo: La tercera subcategoría del aprendizaje automático (y del aprendizaje profundo)

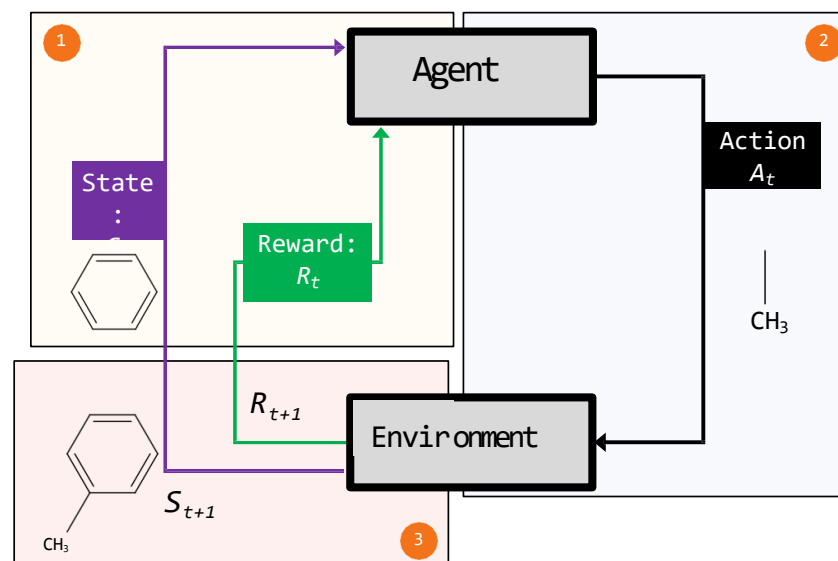


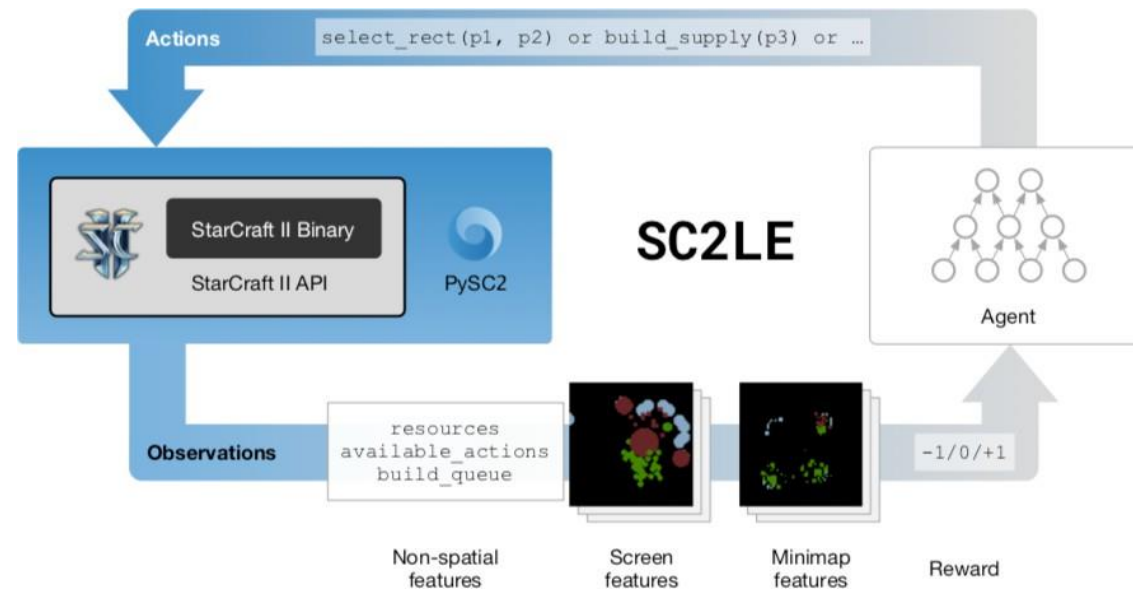
Figure 5: Representation of the basic reinforcement learning paradigm with a simple molecular example. (1) Given a benzene ring (state S_t at iteration t) and some reward value R_t at iteration t , (2) the agent selects an action A_t that adds a methyl group to the benzene ring. (3) The environment considers this information for producing the next state (S_{t+1}) and reward (R_{t+1}). This cycle repeats until the episode is terminated.

Source: Sebastian Raschka and Benjamin Kaufman (2020)

Machine Learning and AI-based approaches for bioactive ligand discovery and GPCR-ligand recognition



Aprendizaje por refuerzo: La tercera subcategoría del aprendizaje automático (y del aprendizaje profundo)



Vinyals, Oriol, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani et al. "Starcraft II: A new challenge for reinforcement learning." *arXiv preprint arXiv:1708.04782* (2017).



Aprendizaje semisupervisado

- Mezcla entre aprendizaje supervisado y no supervisado
- Algunos ejemplos de entrenamiento contienen salidas, pero otros no
- Se utiliza el subconjunto etiquetado del entrenamiento para etiquetar la parte no etiquetada del conjunto de entrenamiento, que luego también utilizamos para entrenar el modelo



Aprendizaje semisupervisado

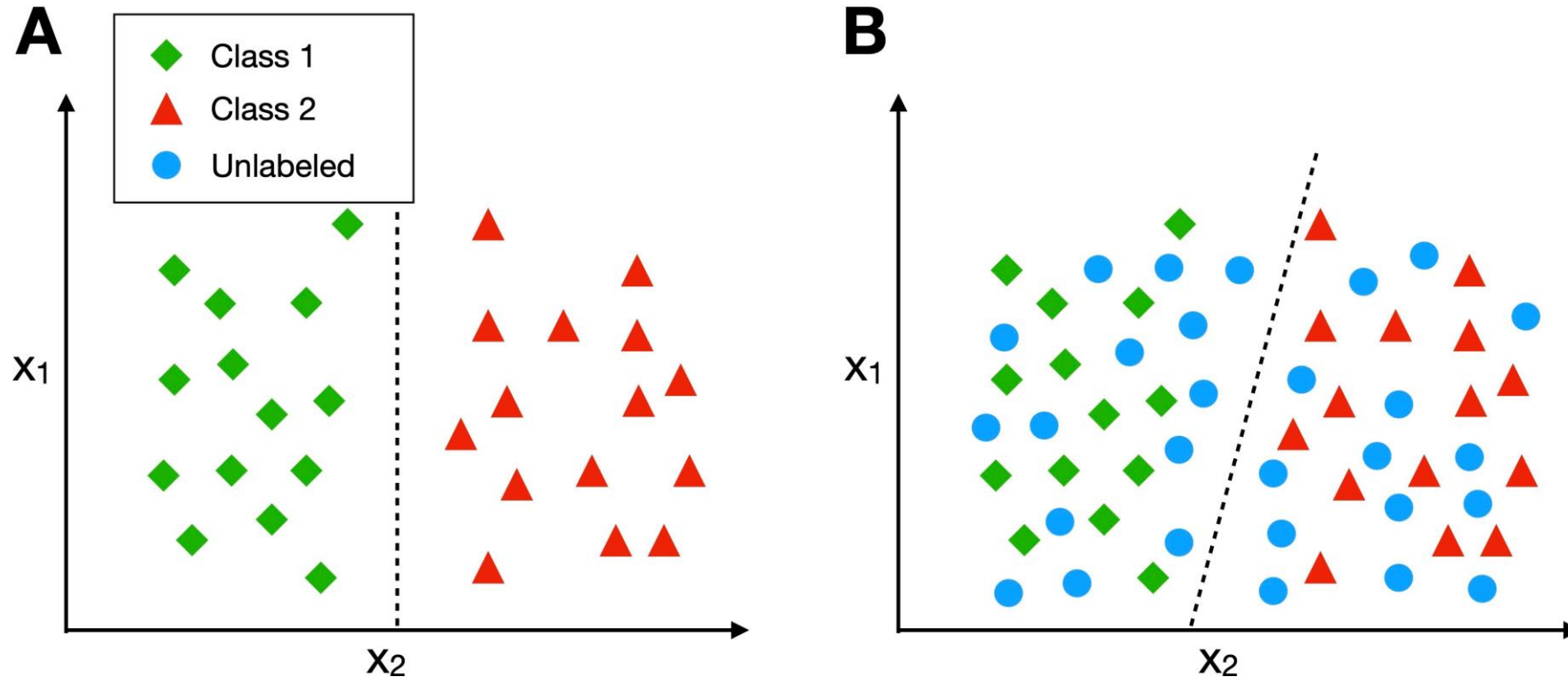


Ilustración del aprendizaje semisupervisado que incorpora ejemplos no etiquetados. (A) Una frontera de decisión derivada solo de los ejemplos de entrenamiento etiquetados. (B) Una frontera de decisión basada en ejemplos tanto etiquetados como no etiquetados.

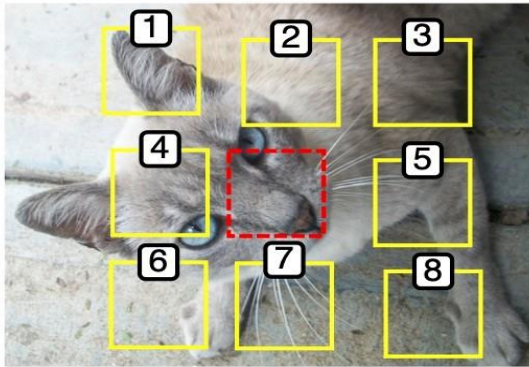


- Un desarrollo reciente y una tendencia prometedora en investigación en aprendizaje profundo
- Particularmente útil si no hay modelos preentrenados disponibles para transferencia en el dominio objetivo
- Es un proceso de derivar y utilizar información de etiquetas directamente de los datos mismos en lugar de que humanos las anoten

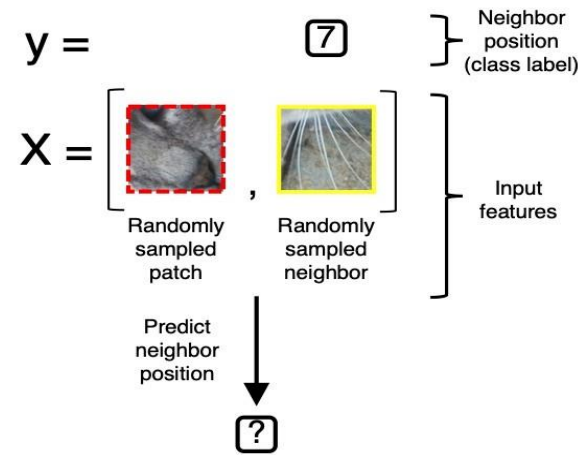


Aprendizaje auto-supervisado

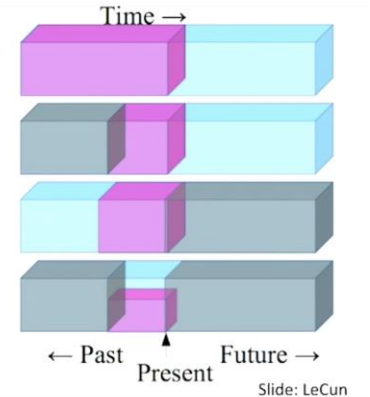
A



B



- Predict any part of the input from any other part.
- Predict the **future** from the **past**.
- Predict the **future** from the **recent past**.
- Predict the **past** from the **present**.
- Predict the **top** from the **bottom**.
- Predict the occluded from the visible
- **Pretend there is a part of the input you don't know and predict that.**



Slide: LeCun

Aprendizaje auto-supervisado mediante predicción de contexto. (A) Se selecciona un parche aleatorio (cuadro rojo) junto con 9 parches vecinos. (B) Dado el parche aleatorio y un parche vecino aleatorio, la tarea es predecir la posición del parche vecino relativo al parche central (cuadro rojo).

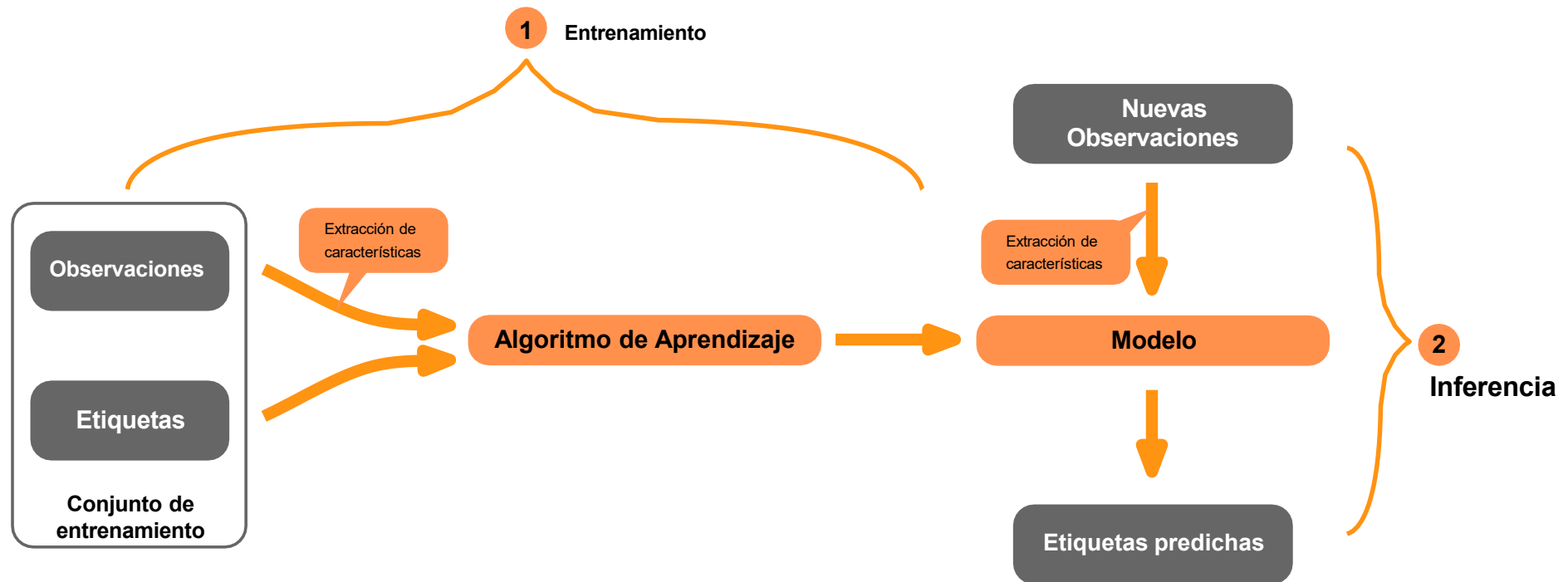


El flujo de trabajo del aprendizaje supervisado

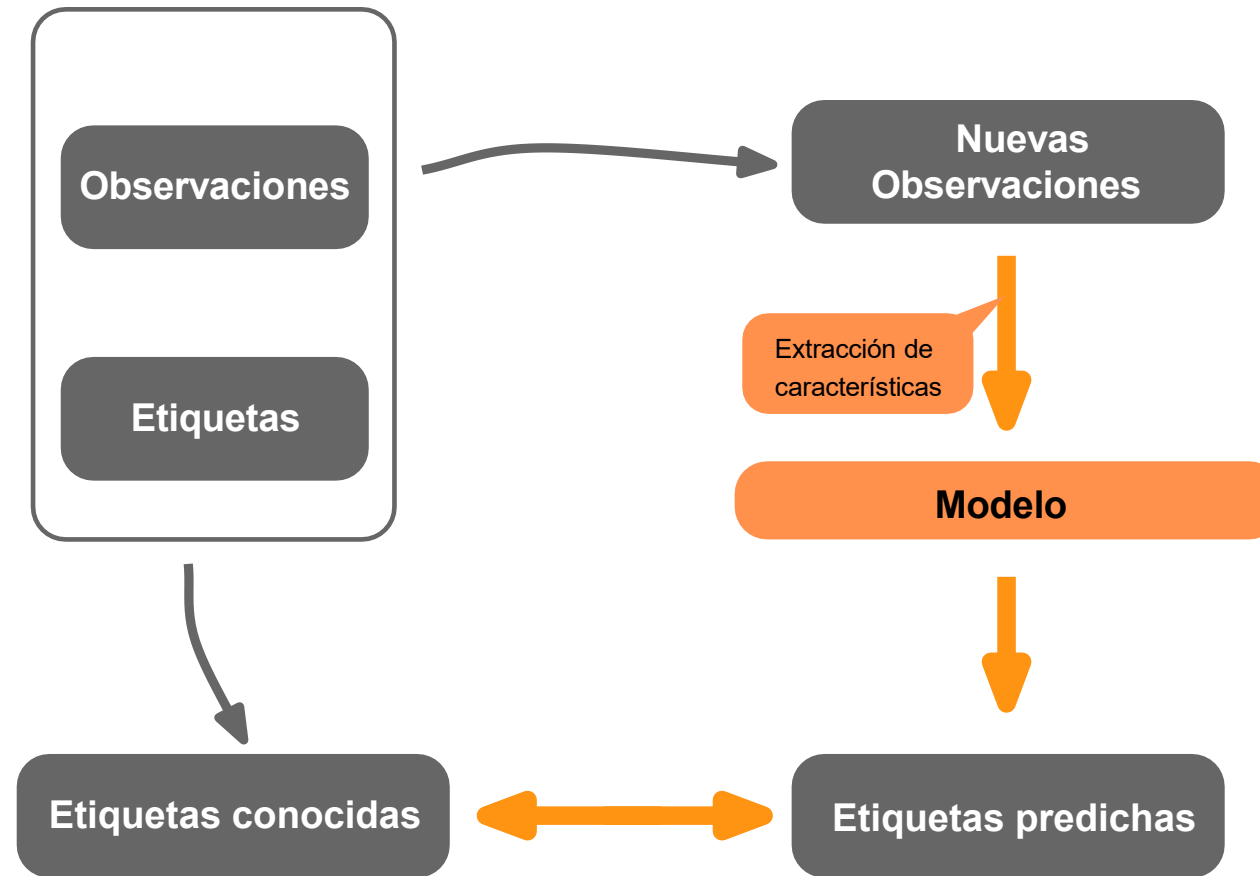
1. Panorama general del curso
2. ¿Qué es el aprendizaje automático?
3. Las categorías generales del aprendizaje automático
- 4. Flujo de trabajo del aprendizaje supervisado**
5. Notación y jerga necesarias
6. Aspectos prácticos y herramientas



Flujo de trabajo de aprendizaje supervisado



Uso de un conjunto de prueba para evaluar el rendimiento de un modelo predictivo



Datos estructurados vs no estructurados

A

Feature vector of the 1st training example

Class label

| Index | Sepal length | Sepal width | Petal length | Petal width | Species |
|-------|--------------|-------------|--------------|-------------|----------------|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 2 | 4.9 | 3 | 1.4 | 0.2 | Iris-setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| ... | ... | ... | ... | ... | ... |
| 150 | 5.9 | 3 | 5.1 | 1.8 | Iris-virginica |

B



Aprendizaje automático vs aprendizaje profundo

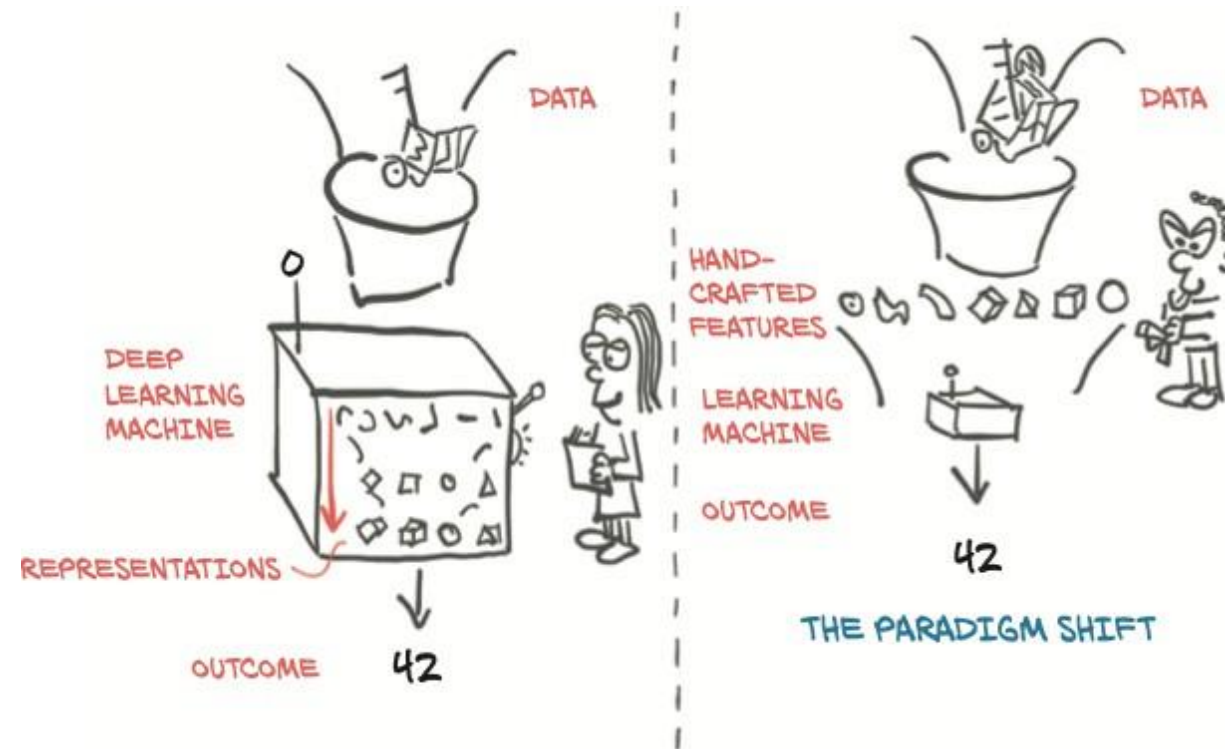


Image source: Stevens et al., *Deep Learning with PyTorch*. Manning, 2020



Aprendizaje automático vs aprendizaje profundo

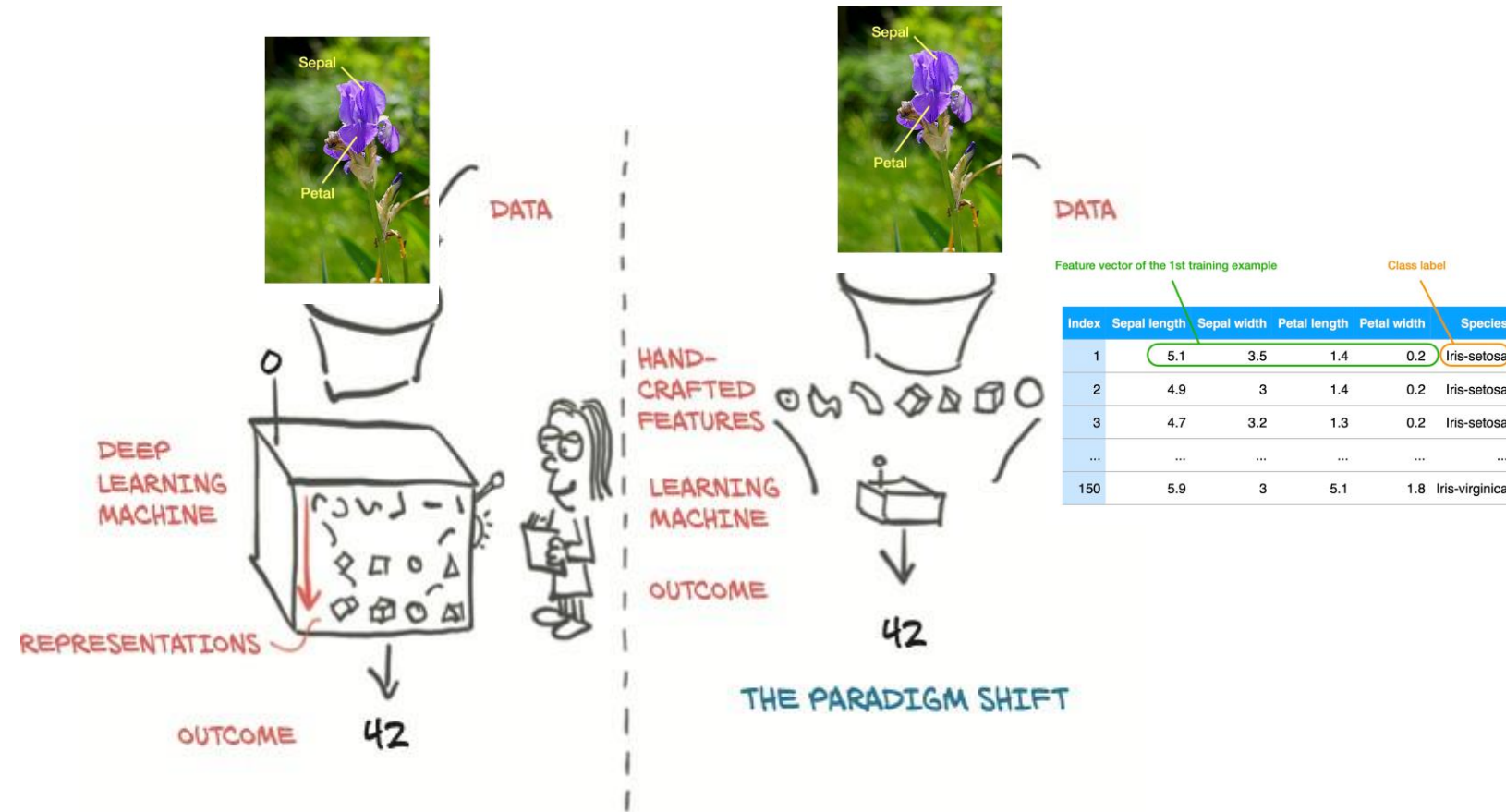


Image source: Stevens et al., *Deep Learning with PyTorch*. Manning, 2020



Terminología y notación del aprendizaje automático

(Nuevamente esto también aplica al Aprendizaje Profundo)

1. Panorama general del curso
2. ¿Qué es el aprendizaje automático?
3. Las categorías generales del aprendizaje automático
4. Flujo de trabajo del aprendizaje supervisado
- 5. Notación y jerga necesarias**
6. Aspectos prácticos y herramientas



Jerga del aprendizaje automático

1/2

Aprendizaje supervisado:

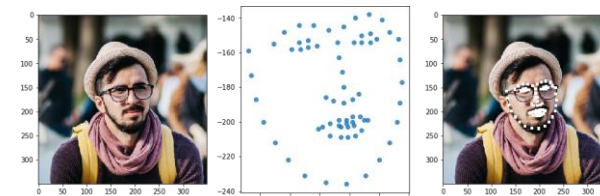
Aprender una función que mapee entrada x (características) a salida y (objetivos)

Datos estructurados:

Bases de datos, hojas de cálculo/archivos csv

Datos no estructurados:

Características como píxeles de imágenes, señales de audio, oraciones de texto (antes del aprendizaje profundo, se requería una amplia ingeniería de características)



Source: http://rasbt.github.io/mlxtend/user_guide/image/extract_face_landmarks/



Aprendizaje supervisado (notación más formal)

"training examples"
↙

Training set: $\mathcal{D} = \{\langle \mathbf{x}^{[i]}, y^{[i]} \rangle, i = 1, \dots, n\},$

Unknown function: $f(\mathbf{x}) = y$

Hypothesis: $h(\mathbf{x}) = \hat{y}$ ← sometimes t or o

Classification ↙

$$h : \mathbb{R}^m \rightarrow \mathcal{Y}, \quad \mathcal{Y} = \{1, \dots, k\}$$

Regression ↘

$$h : \mathbb{R}^m \rightarrow \mathbb{R}$$



Representación de datos

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Feature vector



Representación de datos

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Feature vector

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

Design Matrix

$$\mathbf{X} = \begin{bmatrix} x_1^{[1]} & x_2^{[1]} & \dots & x_m^{[1]} \\ x_1^{[2]} & x_2^{[2]} & \dots & x_m^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{[n]} & x_2^{[n]} & \dots & x_m^{[n]} \end{bmatrix}$$

Design Matrix

4
5

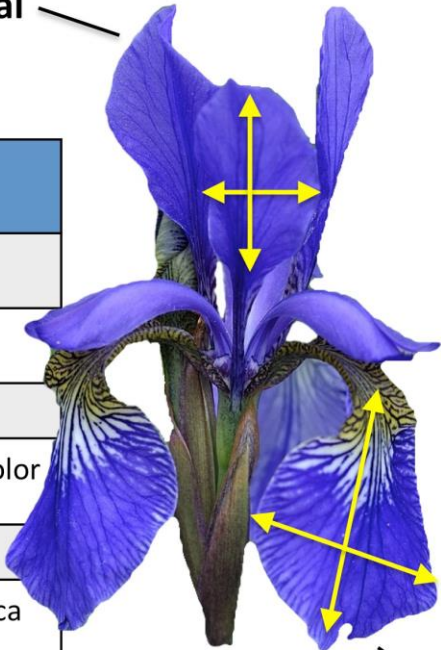


Representación de datos (datos estructurados)

$m =$ _____

$n =$ _____

| | | | | | | |
|-----|--------------|-------------|--------------|-------------|------------|--|
| | | | | | | |
| | Sepal length | Sepal width | Petal length | Petal width | | |
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | Setosa | |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | Setosa | |
| ... | | | | | | |
| 50 | 6.4 | 3.5 | 4.5 | 1.2 | Versicolor | |
| ... | | | | | | |
| 150 | 5.9 | 3.0 | 5.0 | 1.8 | Virginica | |
| | | | | | | |



Petal

Sepal



UNIVERSIDAD EAFIT



Representación de datos (datos no estructurados; imágenes)

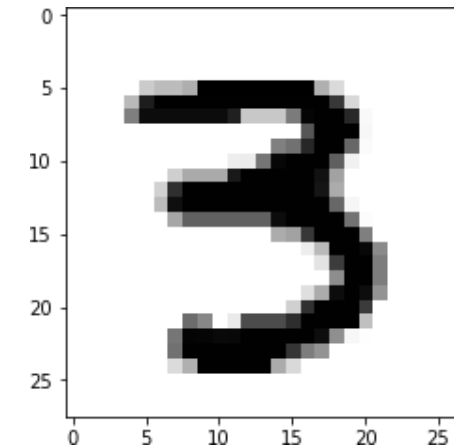
Convolutional Neural Networks

Image batch dimensions: torch.Size([128, 1, 28, 28]) ← "NCHW" representation (more on that later)

Image label dimensions: torch.Size([128])

```
print(images[0].size())  
  
torch.Size([1, 28, 28])
```

```
images[0]  
tensor([[[[0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000],  
        [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000],  
        [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000],  
        [0.0000, 0.0000, 0.0000, 0.0000, 0.5020, 0.9529, 0.9529, 0.9529,  
          0.9529, 0.9529, 0.9529, 0.8706, 0.2157, 0.2157, 0.2157, 0.5176,  
          0.9804, 0.9922, 0.9922, 0.8392, 0.0235, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000],  
        [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.6627, 0.9922, 0.9922, 0.9922, 0.0314, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000],  
        [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.4980, 0.5529,  
          0.8471, 0.9922, 0.9922, 0.5961, 0.0157, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0000],  
        [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,  
          0.0000, 0.0000, 0.0000, 0.0667, 0.0745, 0.5412, 0.9725, 0.9922,  
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000]]]])
```



Entrenar un modelo

Ajustar un modelo = parametrizar un modelo = aprender a partir de los datos

Ejemplo de entrenamiento

Sinónimo de registro de entrenamiento, instancia de entrenamiento, muestra de entrenamiento (en algunos contextos, muestra se refiere a una colección de ejemplos de entrenamiento)

Característica (feature)

Sinónimo de observación, predictor, variable, variable independiente, entrada, atributo, covariable

Objetivo (target)

Sinónimo de resultado, verdad base (ground truth), salida, variable de respuesta, variable dependiente, etiqueta (en clasificación)

Salida / Predicción

Usar esto para distinguirlo de los objetivos; aquí se refiere a la salida del modelo.

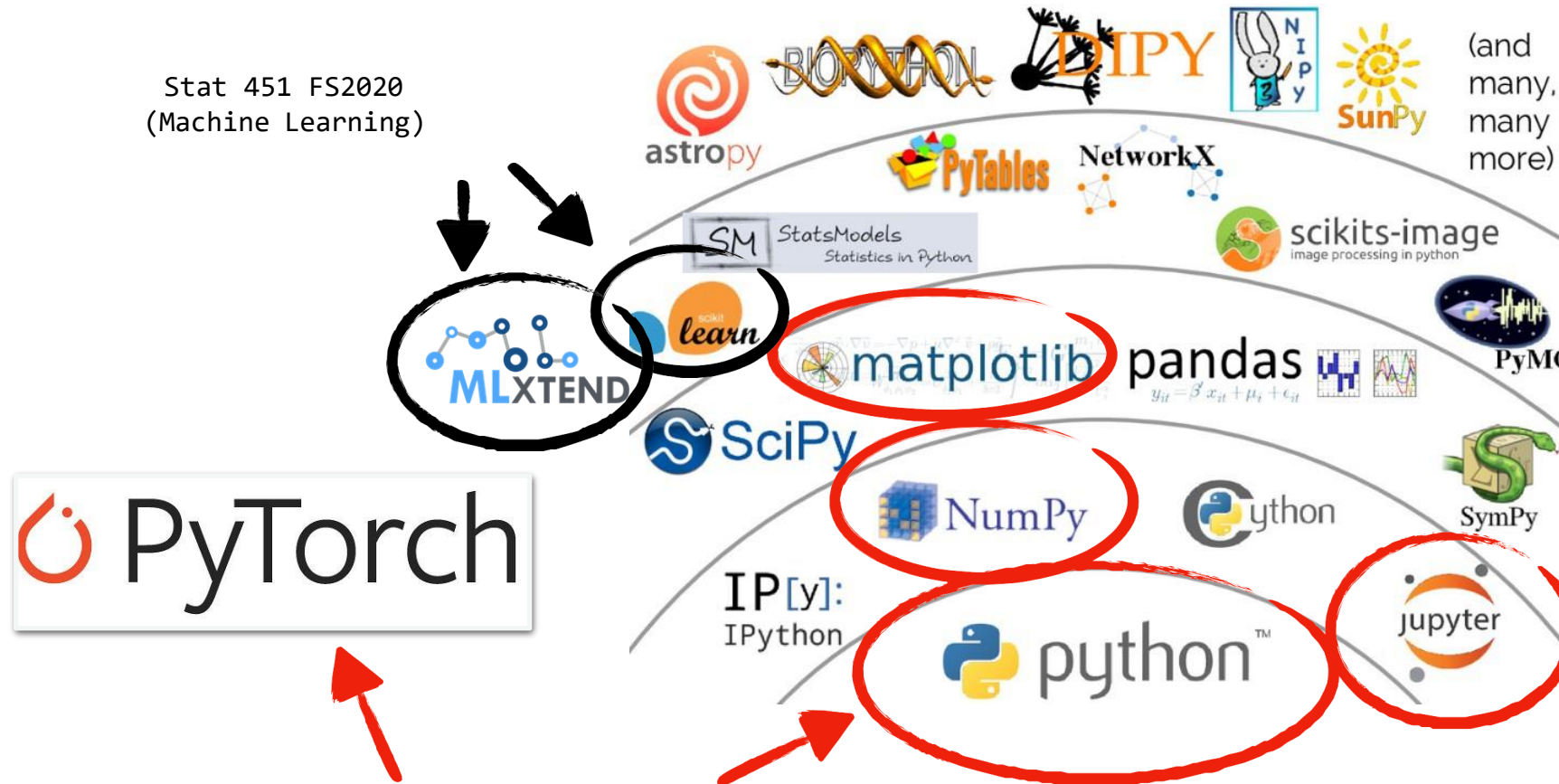


Aspectos Prácticos: ¡Nuestras Herramientas!

1. Panorama general del curso
2. ¿Qué es el aprendizaje automático?
3. Las categorías generales del aprendizaje automático
4. Flujo de trabajo del aprendizaje supervisado
5. Notación y jerga necesarias
- 6. Aspectos prácticos y herramientas**



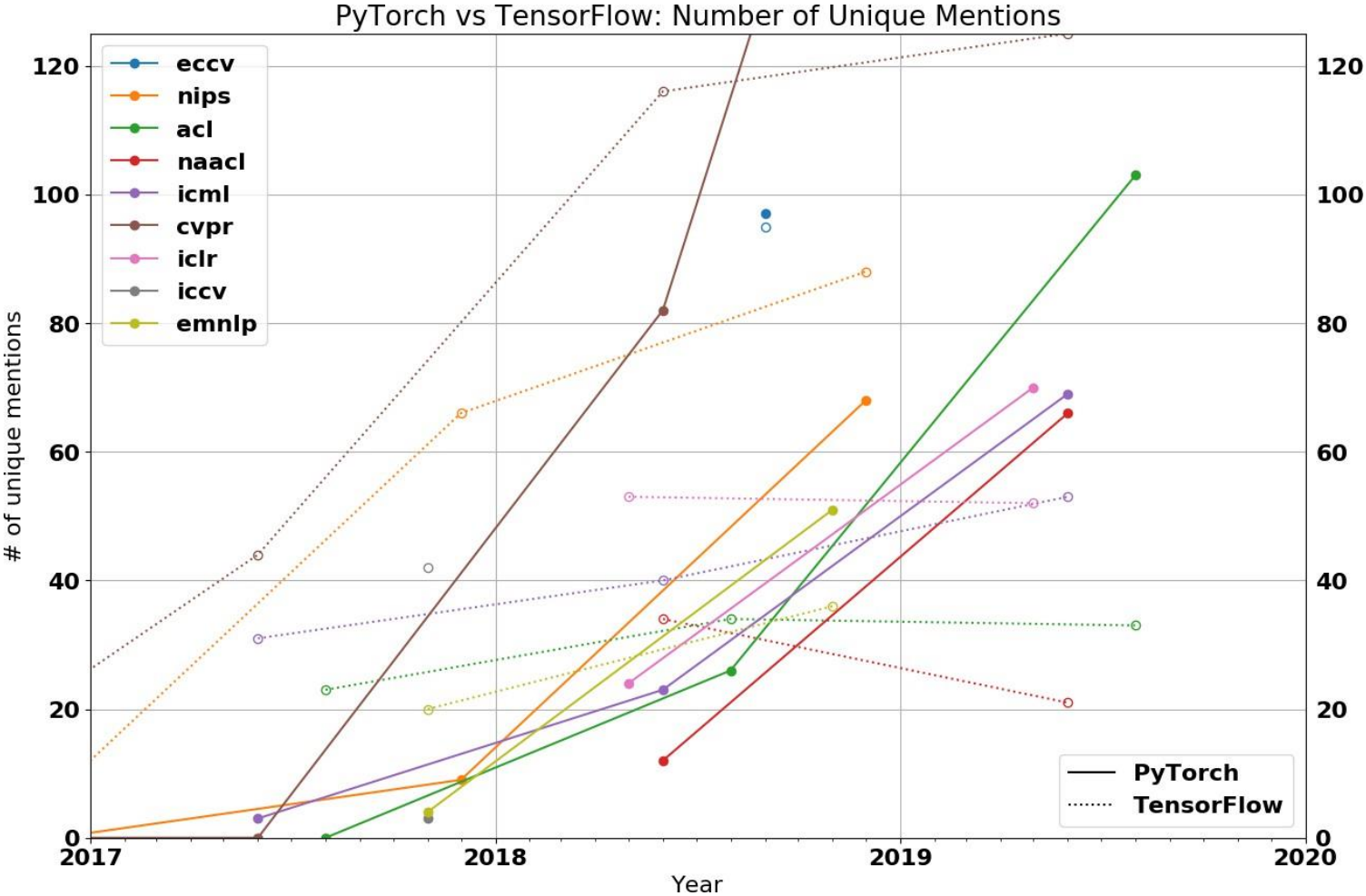
Principales Bibliotecas Científicas de Python



Principales herramientas para este curso

Image by Jake VanderPlas. Source:
<https://speakerdeck.com/jakevdp/the-state-of-the-stack-sciipy-2015-keynote?slide=8>





Source:
<https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry/>



| CONFERENCE | PT 2018 | PT 2019 | PT GROWTH | TF 2018 | TF 2019 | TF GROWTH |
|------------|---------|---------|-----------|---------|---------|-----------|
| CVPR | 82 | 280 | 240% | 116 | 125 | 7.7% |
| NAACL | 12 | 66 | 450% | 34 | 21 | -38.2% |
| ACL | 26 | 103 | 296% | 34 | 33 | -2.9% |
| ICLR | 24 | 70 | 192% | 54 | 53 | -1.9% |
| ICML | 23 | 69 | 200% | 40 | 53 | 32.5% |

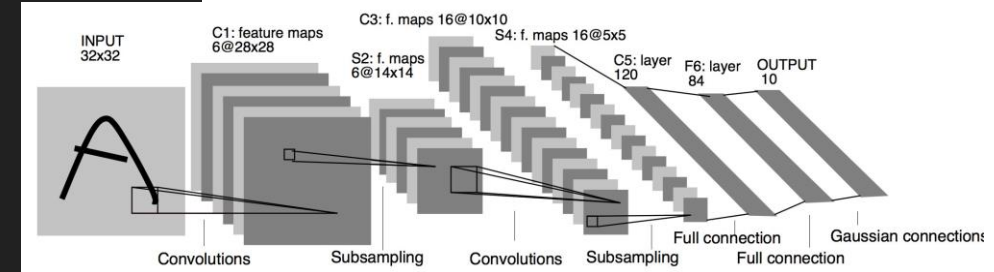
In 2018, PyTorch was a minority. Now, it is an overwhelming majority, with 69% of CVPR using PyTorch, 75+% of both NAACL and ACL, and 50+% of ICLR and ICML. While PyTorch's dominance is strongest at vision and language conferences (outnumbering TensorFlow by 2:1 and 3:1 respectively), PyTorch is also more popular than TensorFlow at general machine learning conferences like ICLR and ICML.



```

34 #####
35 # ## MODEL
36 #####
37
38 class LeNet5(torch.nn.Module):
39
40     def __init__(self, num_classes):
41         super().__init__()
42
43         self.features = torch.nn.Sequential(
44             torch.nn.Conv2d(1, 6, kernel_size=5),
45             torch.nn.Tanh(),
46             torch.nn.MaxPool2d(kernel_size=2),
47             torch.nn.Conv2d(6, 16, kernel_size=5),
48             torch.nn.Tanh(),
49             torch.nn.MaxPool2d(kernel_size=2)
50         )
51
52         self.classifier = torch.nn.Sequential(
53             torch.nn.Linear(16*5*5, 120),
54             torch.nn.Tanh(),
55             torch.nn.Linear(120, 84),
56             torch.nn.Tanh(),
57             torch.nn.Linear(84, num_classes),
58         )
59
60     def forward(self, x):
61         x = self.features(x)
62         x = torch.flatten(x, 1)
63         logits = self.classifier(x)
64         probas = torch.nn.functional.softmax(logits, dim=1)
65         return logits, probas
66
67

```



<https://code.visualstudio.com>

<https://github.com/rasbt/stat453-deep-learning-ss21/tree/main/L01/code>




```
(base) raschka@lambda-quad:~/code/stat453-ss21-exp$ python simple_cnn.py
PyTorch version: 1.7.0
Using cuda:0
[W Context.cpp:69] Warning: torch.set_deterministic is in beta, and its de
on operator())
Epoch: 001/010 | Batch 0000/0422 | Loss: 2.2935
Epoch: 001/010 | Batch 0050/0422 | Loss: 0.5462
Epoch: 001/010 | Batch 0100/0422 | Loss: 0.3154
Epoch: 001/010 | Batch 0150/0422 | Loss: 0.2551
Epoch: 001/010 | Batch 0200/0422 | Loss: 0.1792
Epoch: 001/010 | Batch 0250/0422 | Loss: 0.2210
Epoch: 001/010 | Batch 0300/0422 | Loss: 0.1551
Epoch: 001/010 | Batch 0350/0422 | Loss: 0.2155
Epoch: 001/010 | Batch 0400/0422 | Loss: 0.2306
Epoch: 001/010 | Train: 96.72% | Validation: 97.15%
Time elapsed: 0.09 min
Epoch: 002/010 | Batch 0000/0422 | Loss: 0.1028
Epoch: 002/010 | Batch 0050/0422 | Loss: 0.1167
Epoch: 002/010 | Batch 0100/0422 | Loss: 0.0660
Epoch: 002/010 | Batch 0150/0422 | Loss: 0.1024
Epoch: 002/010 | Batch 0200/0422 | Loss: 0.0847
Epoch: 002/010 | Batch 0250/0422 | Loss: 0.0905
Epoch: 002/010 | Batch 0300/0422 | Loss: 0.1024
Epoch: 002/010 | Batch 0350/0422 | Loss: 0.0719
Epoch: 002/010 | Batch 0400/0422 | Loss: 0.1302
Epoch: 002/010 | Train: 98.12% | Validation: 98.03%
Time elapsed: 0.18 min
Epoch: 003/010 | Batch 0000/0422 | Loss: 0.0720
Epoch: 003/010 | Batch 0050/0422 | Loss: 0.0984
Epoch: 003/010 | Batch 0100/0422 | Loss: 0.0373
Epoch: 003/010 | Batch 0150/0422 | Loss: 0.0685
Epoch: 003/010 | Batch 0200/0422 | Loss: 0.0511
Epoch: 003/010 | Batch 0250/0422 | Loss: 0.0617
Epoch: 003/010 | Batch 0300/0422 | Loss: 0.0768
```

```
Epoch: 010/010 | Batch 0000/0422 | Loss: 0.0095
Epoch: 010/010 | Batch 0050/0422 | Loss: 0.0113
Epoch: 010/010 | Batch 0100/0422 | Loss: 0.0135
Epoch: 010/010 | Batch 0150/0422 | Loss: 0.0028
Epoch: 010/010 | Batch 0200/0422 | Loss: 0.0019
Epoch: 010/010 | Batch 0250/0422 | Loss: 0.0049
Epoch: 010/010 | Batch 0300/0422 | Loss: 0.0132
Epoch: 010/010 | Batch 0350/0422 | Loss: 0.0114
Epoch: 010/010 | Batch 0400/0422 | Loss: 0.0270
Epoch: 010/010 | Train: 99.55% | Validation: 98.68%
Time elapsed: 0.88 min
Total Training Time: 0.88 min
Test accuracy 98.66%
```

