

Deep Learning

Tema 2 - Autograd, MLP con iris



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID



Máster
Deep Learning

1. ¿Qué es la diferenciación automática?
2. Autograd en PyTorch
3. Uso de gradientes para optimizar un modelo
4. Modelo MLP
5. Cómo entrenar un MLP

1. ¿Qué es la diferenciación automática?



Diferenciación automática

- La **diferenciación automática** es un método computacional para calcular derivadas de forma eficiente y precisa. Es fundamental en aprendizaje profundo porque permite actualizar los pesos de una red neuronal mediante gradientes.

$$y = x^3$$

$$\frac{dy}{dx} = 3x^2$$

2. Autograd en Pytorch

Autograd en Pytorch

- PyTorch utiliza un *grafo computacional dinámico* que registra las operaciones aplicadas a los tensores.
- Para calcular la derivada automáticamente se utiliza ***.backward()***

*¿Pero se puede calcular el gradiente
de cualquier tensor?*

No, tenemos que definirlo al crear el tensor





Variable única

$$y = x^3$$

$$\frac{dy}{dx} = 3x^2$$

Múltiples variables

$$f = x^3 + 4z^2$$

$$\frac{dy}{dx} = 3x^2 + 0; \frac{dy}{dx} = 3x^2$$

$$\frac{dy}{dz} = 0 + 8x; \frac{dy}{dz} = 8x$$



Autograd en Pytorch: **reinicio de gradientes**

- Durante el entrenamiento, los gradientes se acumulan (suman) en cada iteración del bucle de entrenamiento.
- Para evitar usar gradientes acumulados de iteraciones anteriores, **debemos reiniciar los gradientes** después de cada paso.

```
grad.zero_()
```

```
zero_grad()
```

Autograd en Pytorch: **reinicio de gradientes**

	Uso principal	Alcance	Ventajas
<code>grad.zero_()</code>	Para reiniciar el gradiente de un tensor específico.	Solo afecta al tensor dado.	Puedes reiniciar específicamente los gradientes de un tensor sin afectar al resto.
<code>zero_grad()</code>	Para reiniciar gradientes de todos los parámetros de un modelo.	Afecta a todos los parámetros asociados al optimizador.	Más eficiente y común con modelos complejos, ya que reinicia automáticamente todos los gradientes del optimizador.

3. Uso de gradientes para optimizar un modelo

Descenso de gradientes durante el entrenamiento de modelos



Bucle de entrenamiento

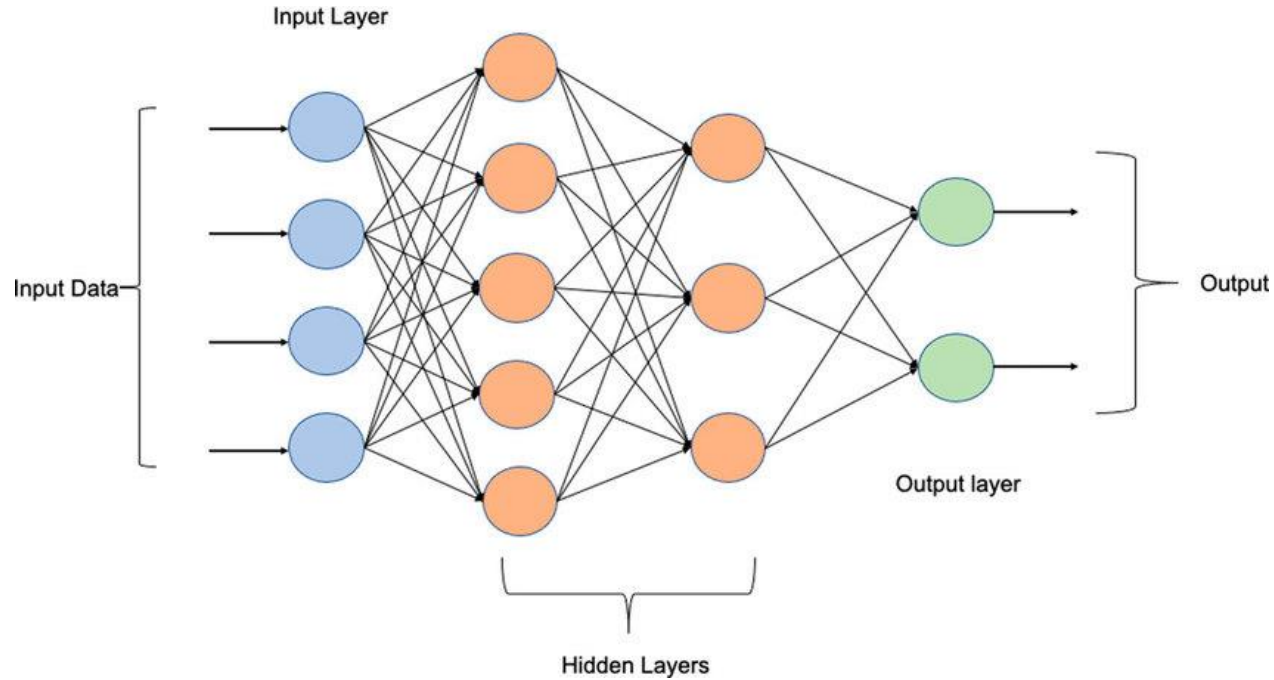
1 Calcular la
función de
pérdida

2 Calcular el
gradiente

3 Modificar el
valor del tensor

4 Reiniciar el
gradiente

4. Modelo MLP

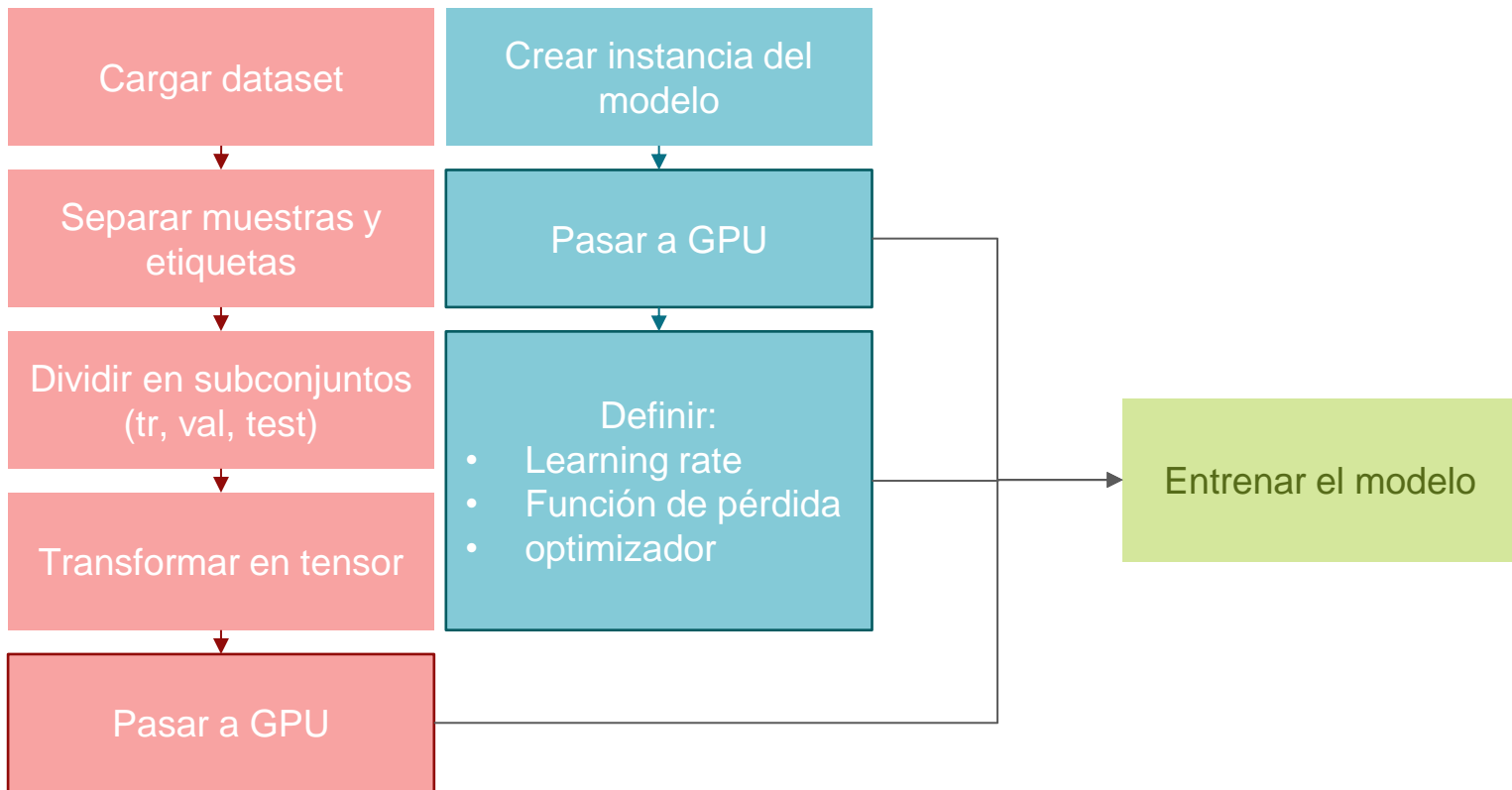


Afan, Haitham & Ibrahim Ahmed Osman, Ahmedbahaaldin & Essam, Yusuf & Ali Najah Ahmed, Al-Mahfoodh & Huang, Yuk & Kisi, Ozgur & Sherif, Mohsen & Sefelnasr, Ahmed & Chau, Kwok & El-Shafie, Ahmed. (2021). Modeling the fluctuations of groundwater level by employing ensemble deep learning techniques. *Engineering Applications of Computational Fluid Mechanics*. 15. 1420-1439. 10.1080/19942060.2021.1974093.

5. Cómo entrenar un MLP



Pasos para entrenar un MLP



6.EXTRA RESOURCES

Documentación

- https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html
- <https://pytorch.org/docs/stable/generated/torch.nn.Linear.html>

Deep Learning

Tema 2 - Autograd, MLP con iris



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID



Máster
Deep Learning