HW1 講解

Couette flow(流體力學) and regression(回歸)

The most reliable way to anticipate the future is to understand the present.

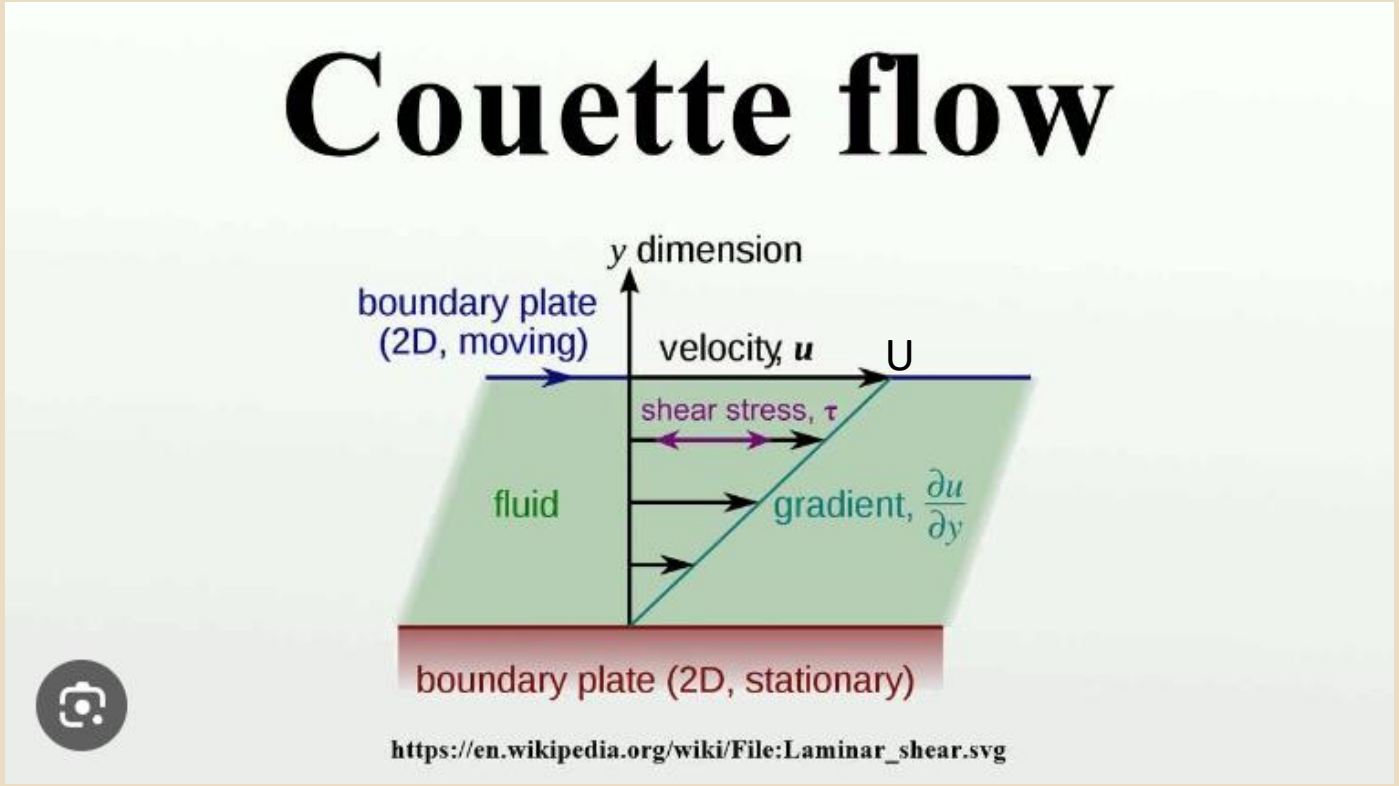
John Naisbitt-1989

Couette flow

Derive from fluid mechanics

Couette flow

Navier-Stokes equation: $\rho(\mathbf{r},t)[\frac{\partial \mathbf{u}(\mathbf{r},t)}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}(\mathbf{r},t)]=\rho(\mathbf{r},t)\,f(\mathbf{r},t)+\nabla\cdot\mathbf{T}$



$$u=\frac{z}{H}U$$

Regression

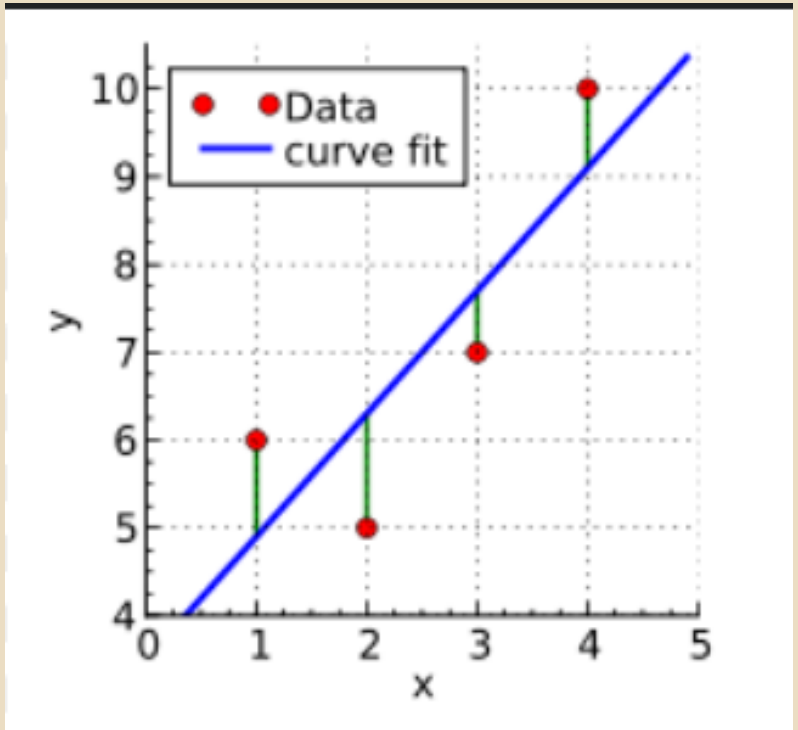
Curve fittung

最小平方法(線性)

$$y = wx + b$$

$$\min_{w,b} \sum_{i} (y - y_i)^2$$

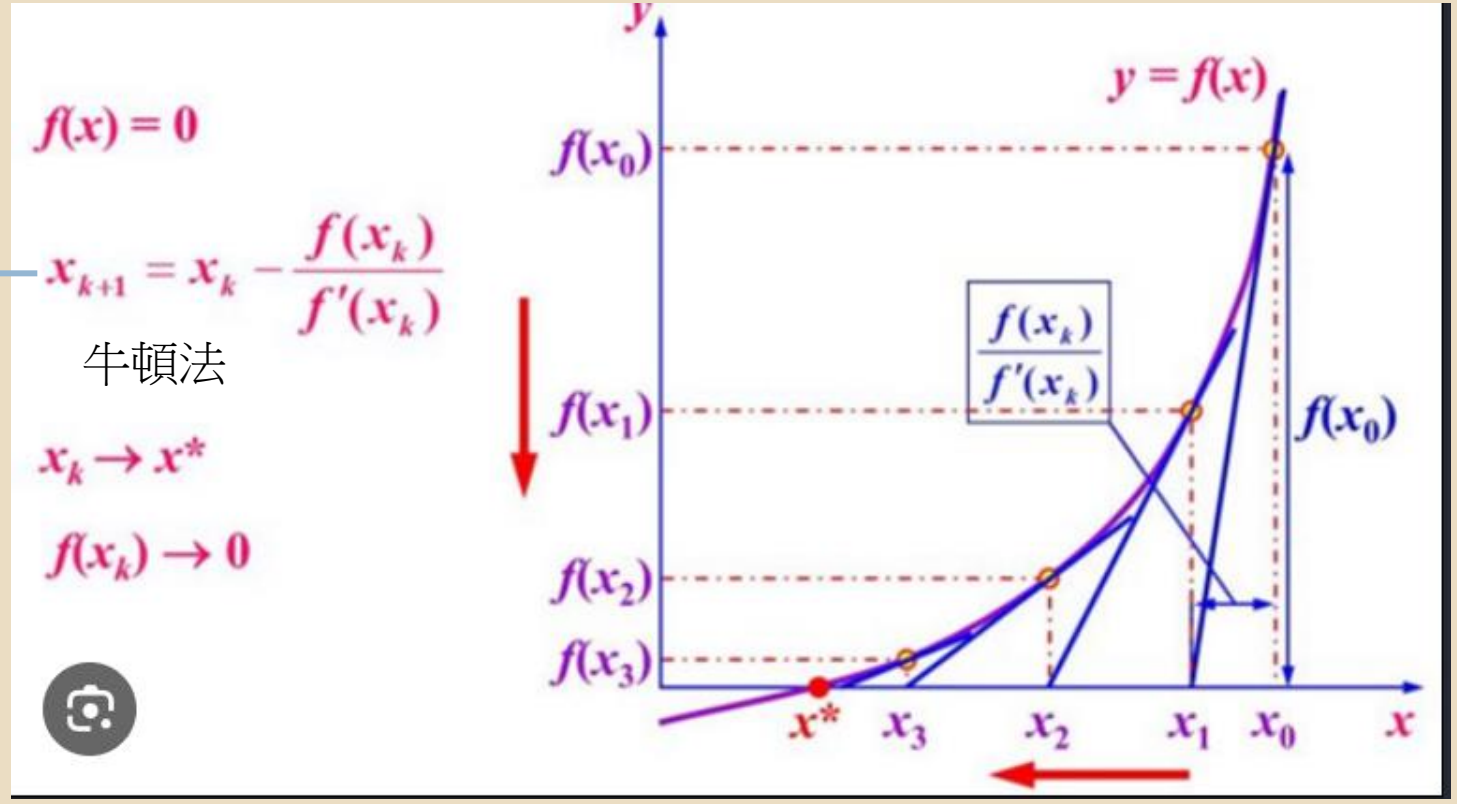
Loss function



牛頓法→梯度下降法

$$x_{k+1} = x_k - lr \times f^{'}(x_k)$$

梯度下降法



機器學習架構

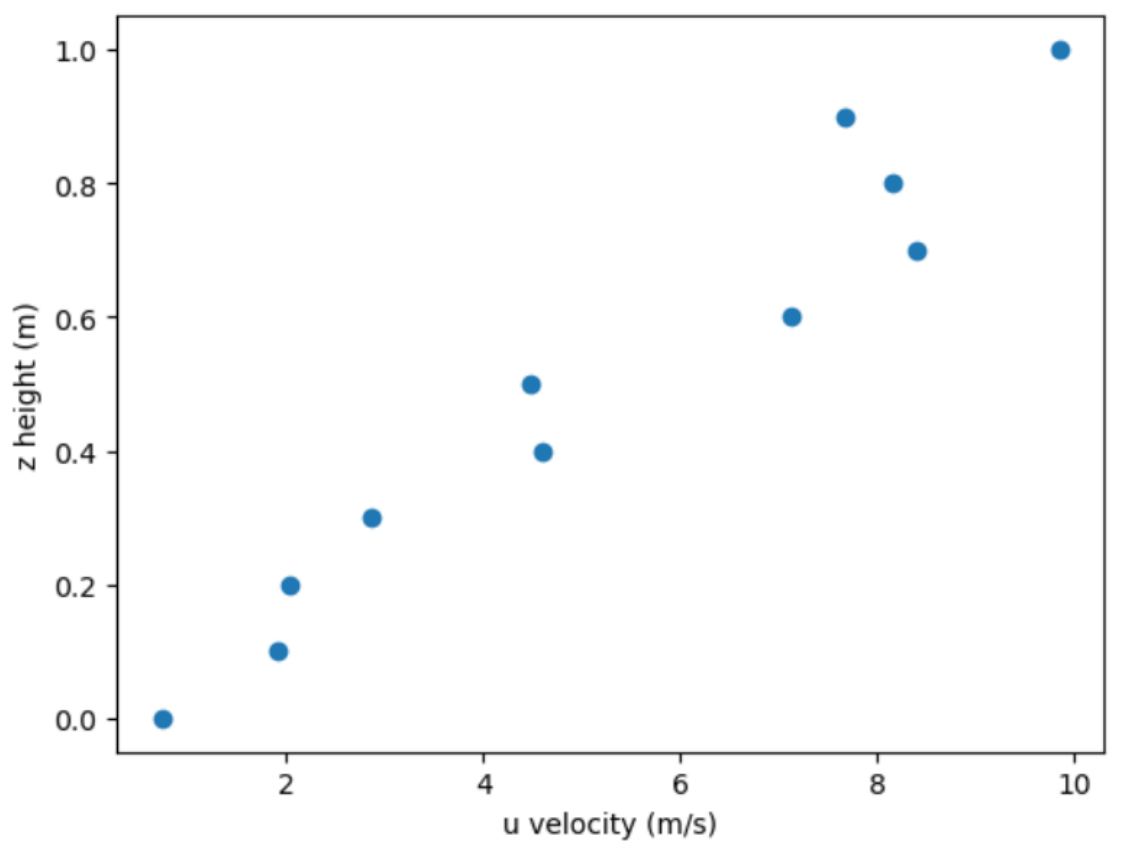
1. 使用線性模型: $u_p = z \times w + b$，其中up為預測速度、w為權重(weight)、b為偏位(bias)。
2. 使用最小平方法做為損失函數(loss function): loss_fn=(u_p-u)^2 (最小平方法)
3. 透過微分方式求得損失函數對w與b之梯度並結合學習率(learning rate)對w與b重複疊代進行最佳化(optimization) (梯度下降法)

HW1

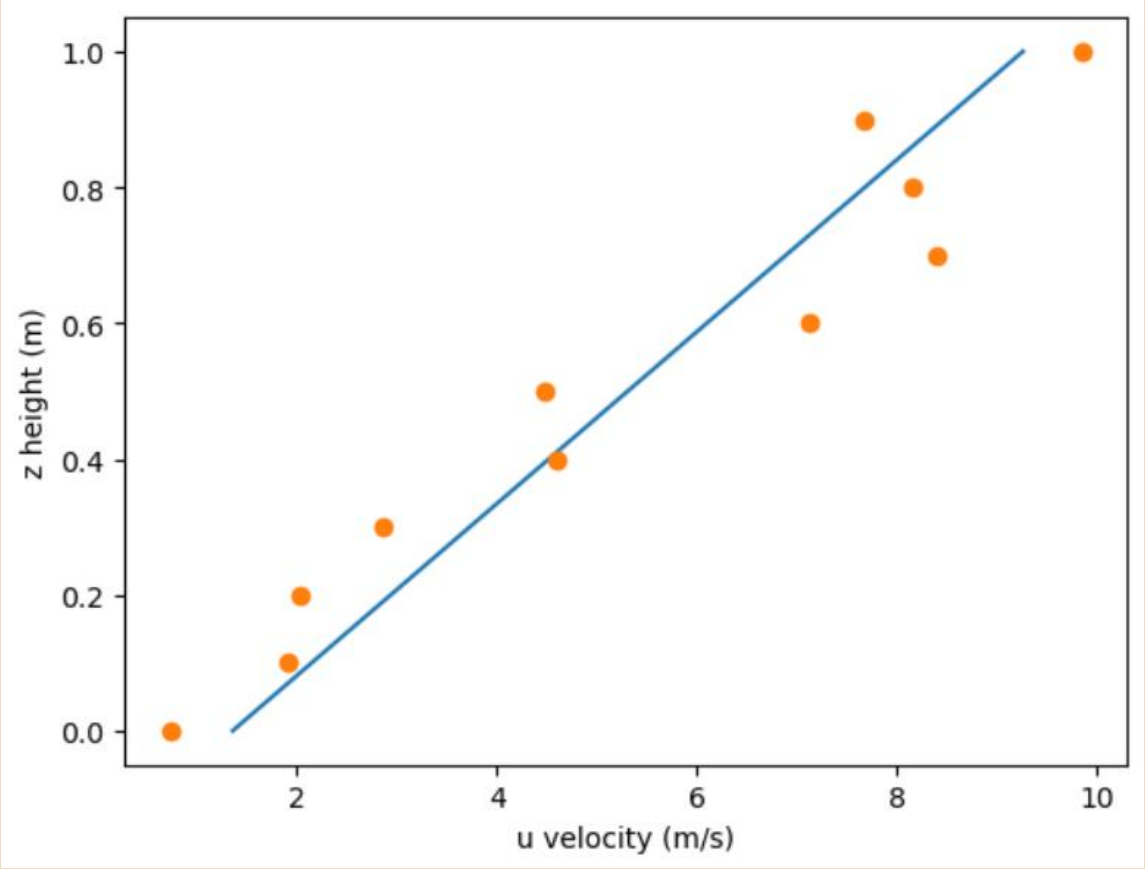
Curve fittung

Couette flow regression

Data



result



Basic level

Weight, bias, epoch, learning rate

```
''' Homework!!! You have to fill the number in this block to achieve basic grade '''
params=torch.tensor([XX, XX]) # weight and bias, arbitrary number
config={
    "number epoch": XXXX, # number of epoch, suggest from 100 to 50000
    "learning rate": XXXX, # learning rate, suggest from 1e-2 to 1e-5
    }
```

Intermediate level

(1)使用torch背向傳播自動計算梯度(backward)與SGD優化器進行更新參數。
(2)課堂上有提到使用L2 regularization可使weight之值降低，先將上述第一項任務完成，加入在其中加入L2 regularization。

```
'''Homework!!! Changing optimizer to this code,
Additionally, implement L2 regularization code tAo achieve intermediate level'''
#optimizer=optim.SGD([XXXX], XXXX, XXXX)
```

Training loop

```
#Parameters update
def training(x,y,params,n_epoch,lr):
    for epoch in range(1, n_epoch+1):
        w, b=params
        x_p=model(x,w,b)
        loss=loss_fn(x_p,y)
        grad=grad_fn(x_p,x,y)
        params=optim(params,lr,grad)
        '''Homework!!! Changing  optimize step to thiscode to achieve intermediate level'''
        #optimizer.zero_grad()
        #train_loss.backward()
        #optimizer.step()
        print('Epoch: %d, Loss: %f' %(epoch, loss))
    return params
```

Advanced level

(1)為了解模型是否有(overfitting)之情形出現，可將訓練資料分成training set與validation set以檢查損失函數之下降情況判斷訓練是否過擬和。將資料透過任何方法分割成training set與validation set，並印出validation loss。
(2)起透過增加一個變數early stop，讓validate loss若沒持續下降一定次數則提早停止訓練。

Hints:需使用函數或方法

shuffle_ind=torch.randperm(num) 回傳0至num-1之隨機排列

validate_ratio=0.2 為validation set佔所有dataset之比例

作業繳交

1. 將作業之code列印出紙本訂起來
2. 如果有做intermediate level/advanced level請特別標示出來
3. 請附上結果圖
4. 下周上課繳交

The most reliable way to anticipate the future is to create the future.

Lin-2024