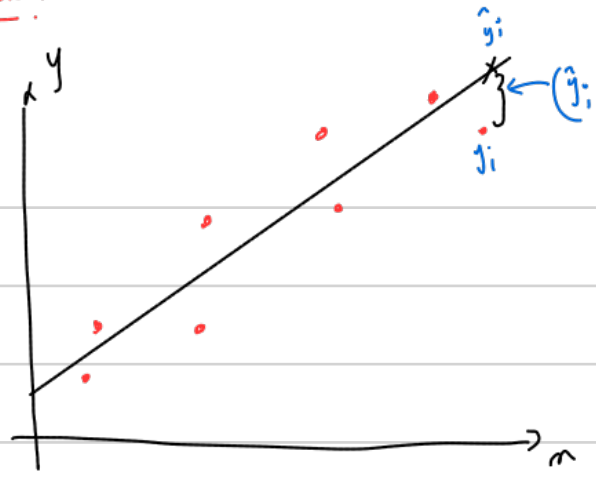
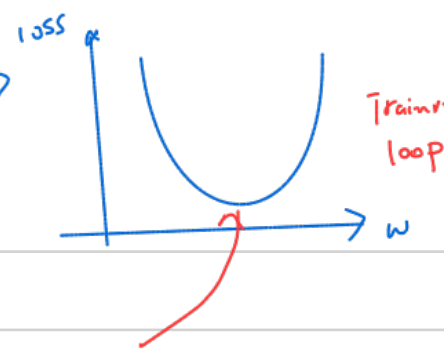
# Regression




$(\hat{y}_i - y_i)^2$ ← mean square Error ⟹ loss = $w^2$




Trainny loop

- Tences ← data containers
- Autograds ← Numerical diffi - -
- Optimum ← Optimizer

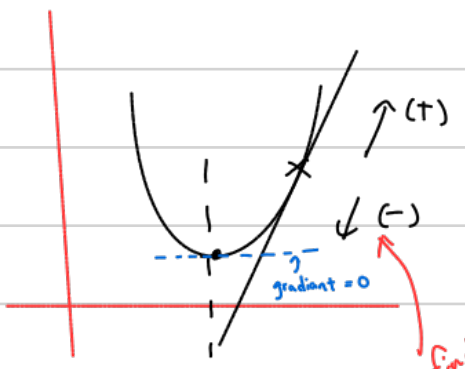
Find the optimal

$f(w) = w^2$

$f'(w) = 2w$

moving downhill ⇒ $w\text{-new} = w - \eta \cdot f(w)$

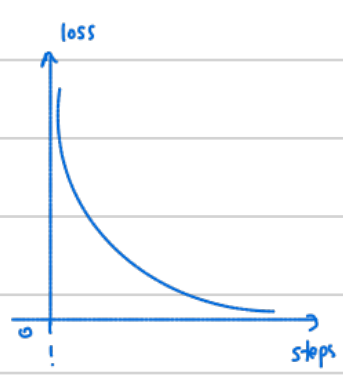
$= w - \eta(2w)$




to find the minimum we goes with the gradient descend.

$\eta$ ← learny rate (size of the step) (hyper parameter)

doesn't change the valu in the learning process -

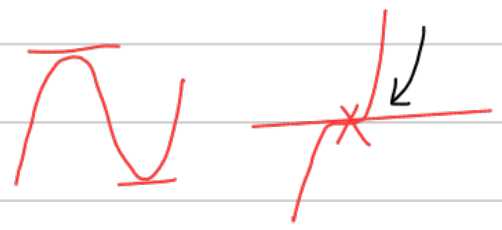



* use a treashhold to find the minimal poit.

why we not taking gradiant = 0 ? (points may be overfitted / There can be a fuction which goes through all points)

```python
# ------------------------------
w = -3.0             # starting point
lr = 0.3             # learning rate
steps = 25           # number of iterations
thresh= 0.001

for i in range(steps):

    gradient = grad_loss(w)
    w_next = w - lr * gradient

    # Plot the frame
    plot_step(w, w_next, gradient)

    # ================================================
    # 👉 STUDENT TASK: Add your stopping criteria here
    # ================================================
    # OPTION CORRECT
    if (abs(w_next-w)< thresh) and (abs(gradient) < thresh):
        print(i)
        print("Stopping criteria met!")
        break


    # Update w
    w = w_next

    time.sleep(0.4)   # smooth pacing
```

Both we have to check



## Importane of the learning Rate

"changing Learning rate we can find the global minima"

pytorch optimizers

SGD (Stochastic Gradient Descent)

Vanila gradiant Desent SGD with (batch size = full_dataset)

Mini batch SGD

Momentum.

Adam.

RMSprop.

Adagrad.

} automatical track all parameters update using gradients apply learning rate scaling & other tricks.