

SGD - fit function to data

e.g.

too low: looks

Loop:

forever

find loss

too high: misses values

find gradient of loss function  $\nabla$   $\rightarrow$  adds diverges

update params (params  $\leftarrow$   $\text{params} - \frac{\eta}{K}(\text{gradient})$ )

minimize  
loss

learning  
rate  $\eta$

rectified linear function:  $y = mx + b$

but if  $y < 0$ ,  $y = 0$

def rectified\_linear( $m, b, x$ ):

$$y = mx + b$$

return torch.clip(y, 0.)

- Can be combined with other linear functions

e.g. rectified linear function + rectified linear function

- Can match any function if we combine lots of rectified linear functions

or use SGD to get those values

tip: try the fastest architecture  
first when optimizing data  
to save time

- train first model to see  
what tweaks are needed  
on DAY ONE

hyperparameter - a parameter used to  
calculate actual parameters

$$\frac{\sum (y - \hat{y})^2}{n}$$

$$\frac{\sum (ax^2 + bx + c - y)^2}{n}$$

$$\frac{\partial L}{\partial w} = \frac{\partial}{\partial w} \sum (2(ax^2 + bx + c - y)^2)$$

$$2(2ax^2 + 2bx + 2c - 2y)$$

can be combined