Weight Initialization

Thursday, 27 May 2021 6:53 PM

Code → weight-initialization

Happens when a model is created and this randomness affects the model s performance

1. Constant weights.

1's

Extremely high initial loss

Similar problem with back propagation as 0's

0's

Loss doesn't decrease and is constant

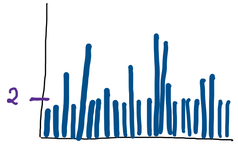
Because in back propagation it's difficult to know which weights are contributing the most to the error

Bad

Back propagation can not deal with consistency it fails

2. Random Uniform

np.random_uniform([-3,3])



most are at 2
little bit randomness

Good

Converges faster!

Rule for Best uniform range

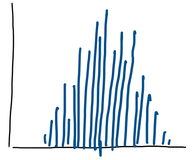
$[-y, y]$, $y = \frac{1}{\sqrt{n}}$ n = # of inputs in a neuron

3. Normal Distribution

np.random.normal(mean, std)

0 1

default



most close to mean

Random values will be closer to 0

Sometimes is better!

4. Automatic

pytorch uses Random uniform weight init.