OPTIMIZER

Stochastic Gradient Descent (with Momentum)

```
for (x, y) in dataset:
    J = ∇ l(θ | x, y)
    m = J + momentum * m
    θ = θ − ε * m.mT
```

init momentum = 0

- Pros: Works in most cases
- Cons: Need to tune your learning rate

What happen if learning rate too high ?

→ Loss spikes, because:

- Magnitude of weights are much larger
  $\| \theta \| = \| w \| \nearrow$
- Magnitude of gradient are much larger
  $\| \nabla l_\theta \| = \| \nabla_w l \| \nearrow$

⇒ And since $w = w - lr * \text{gradient}$. Larger learning rate leads to larger "fluctuation" in weight, which leads to loss spike.

How to prevent loss spike ?

1. RProp

Basic idea: scale gradient by its magnitude $\| \nabla_\theta l \|$

```
...
m = J / J.norm() + momentum * m
...
```

2. RMS Prop:

Compute v: a running average of $\| \nabla_\theta l \|^2$

scale by $1/\sqrt{v}$

```
....
v = β2 * v + (1 − β2) * J.square()
m = J / v.sqrt() + momentum * m
....
```