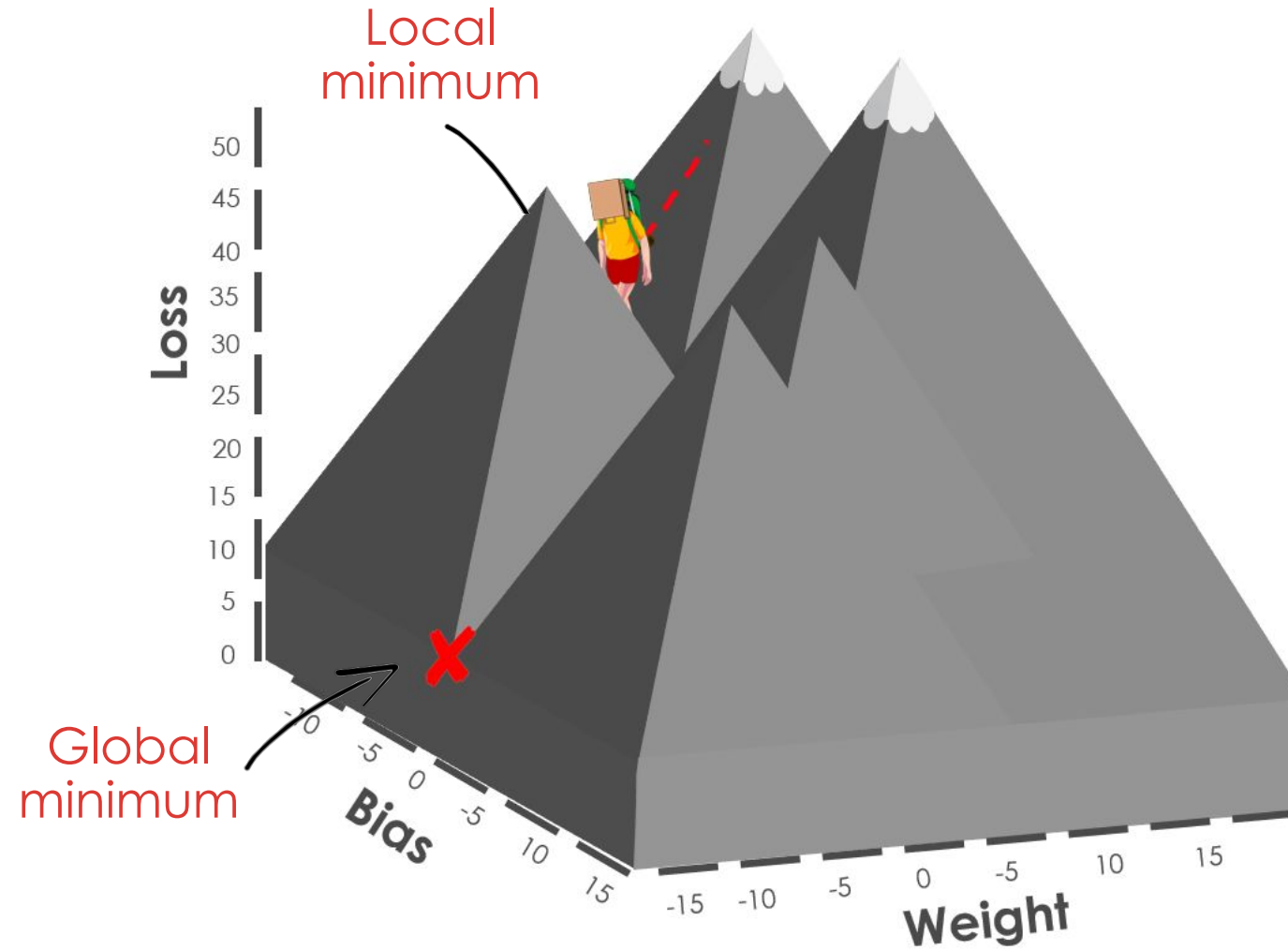




Gradient Descent Optimization

Deep Learning Pre-Work

Local Minimums



Agenda

- Epochs
- Batch size
- Learning rate
- Optimizers

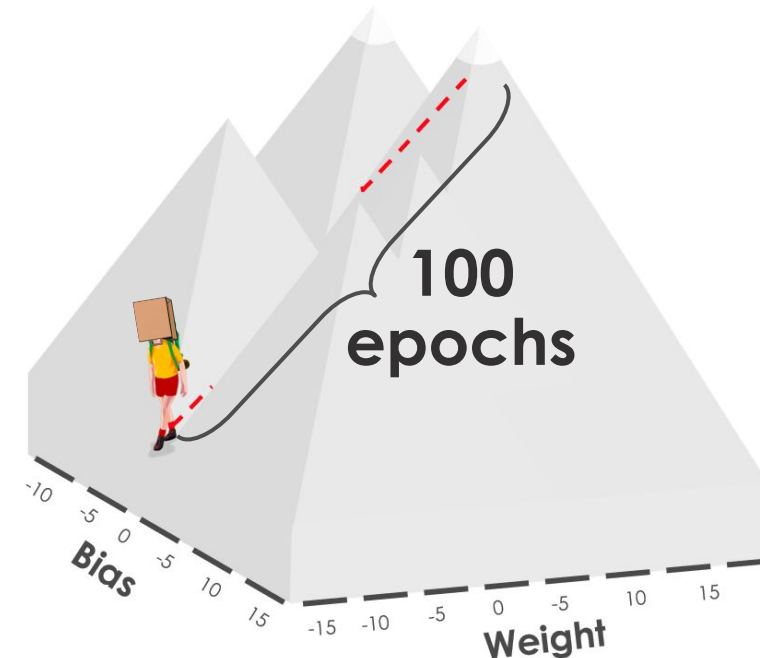
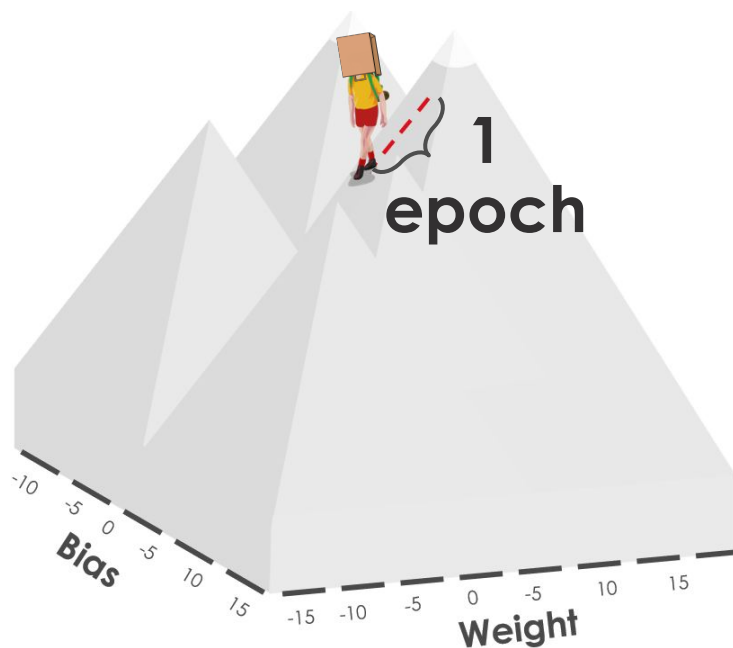
Epochs

Epochs

Training Data

0	0.0	0.0	1.00000
1	1.0	0.0	1.00000
2	1.0	0.0	0.67065
3	1.0	0.0	0.97467
4	1.0	0.0	0.28409
...
240	0.0	0.0	1.00000
241	1.0	0.0	0.00000
242	1.0	0.0	1.00000
243	1.0	0.0	0.94701
244	0.0	0.0	0.00000

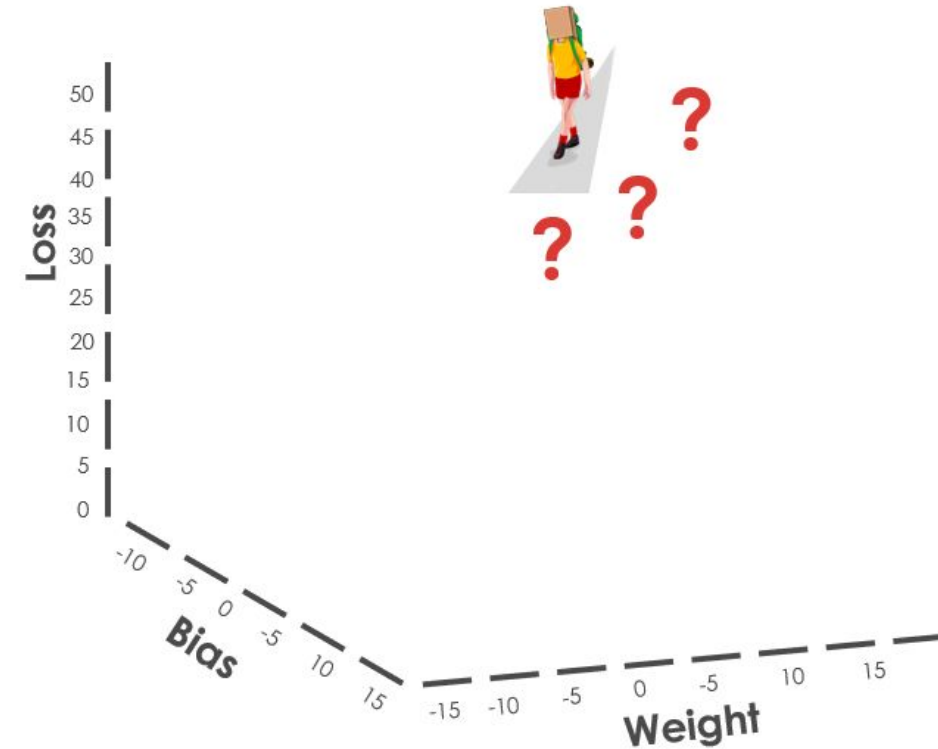
245 rows × 35 columns



Batch Size

Gradient Descent Steps

- Step 1 Start at a random bias and weight and calculate the loss
- Step 2 Take a step in the direction with the steepest gradient**
- Step 3 Calculate the new loss
- Step 4 Repeat steps 2 and 3



Batch size

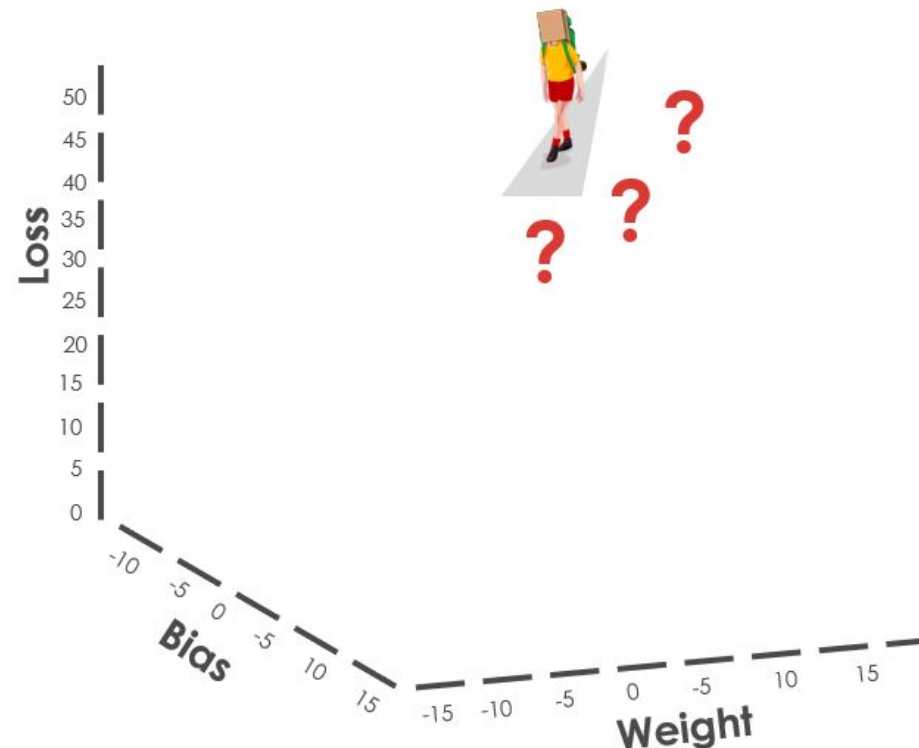
Batch
Gradient Descent

Mini-Batch
Gradient Descent

Stochastic
Gradient Descent

0	0.0	0.0	1.00000
1	1.0	0.0	1.00000
2	1.0	0.0	0.67065
3	1.0	0.0	0.97467
4	1.0	0.0	0.28409
...
240	0.0	0.0	1.00000
241	1.0	0.0	0.00000
242	1.0	0.0	1.00000
243	1.0	0.0	0.94701
244	0.0	0.0	0.00000

245 rows × 35 columns



Batch size

Batch Gradient Descent

Batch size: 245



Mini-Batch Gradient Descent

Batch size: 35



Stochastic Gradient Descent

Batch size: 1

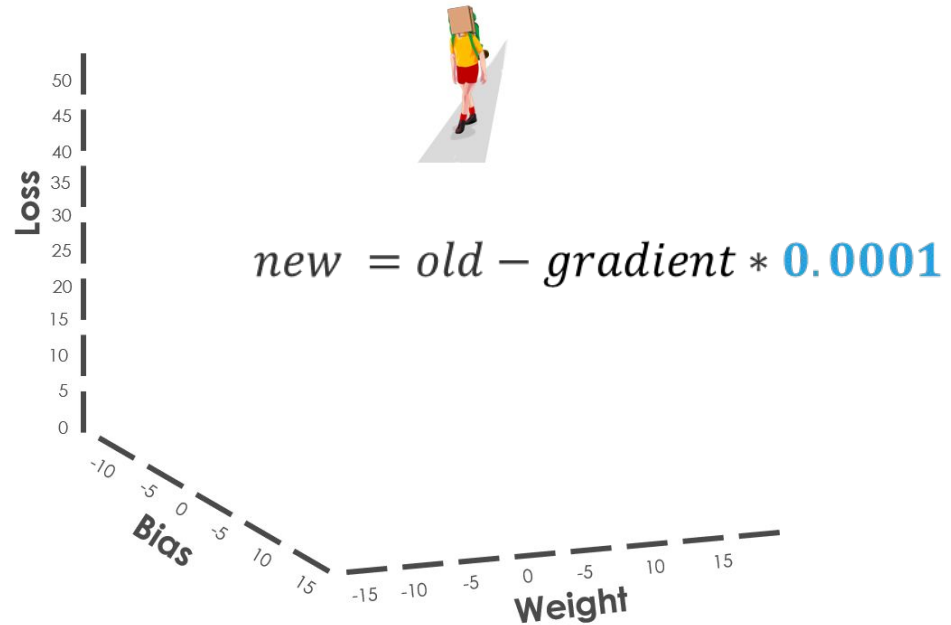


Training sample size: 245
Epochs:10

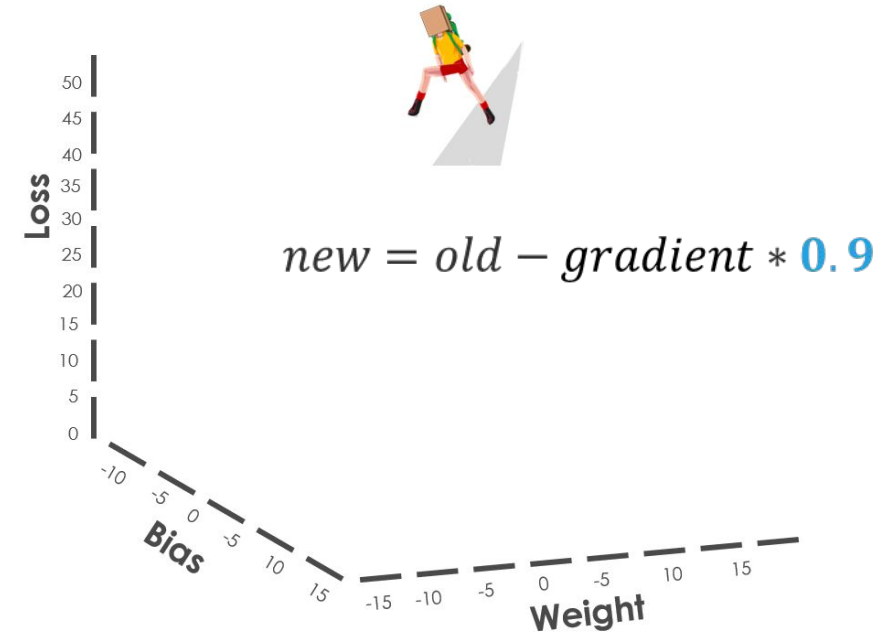
Learning Rate

Learning Rate

Little step



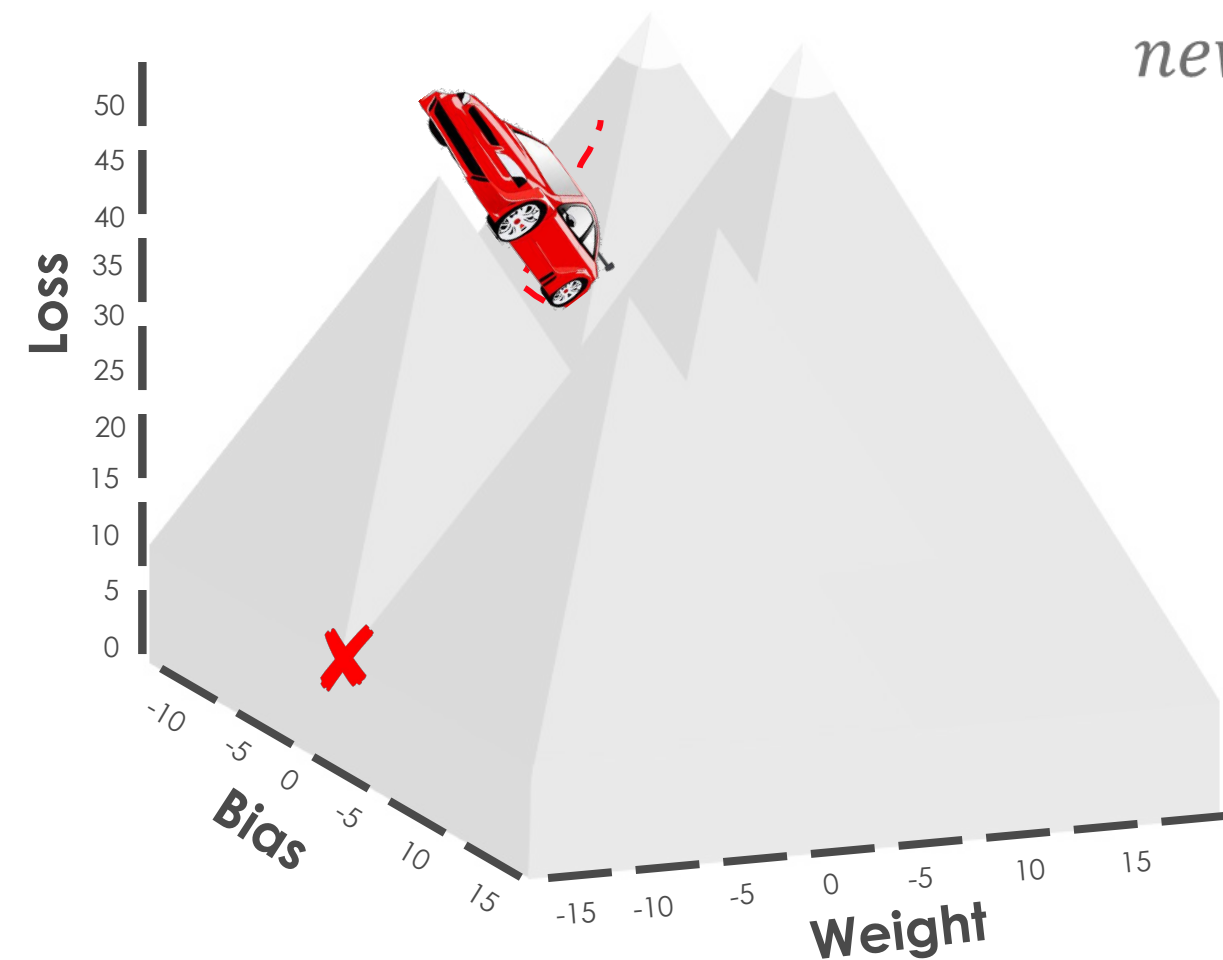
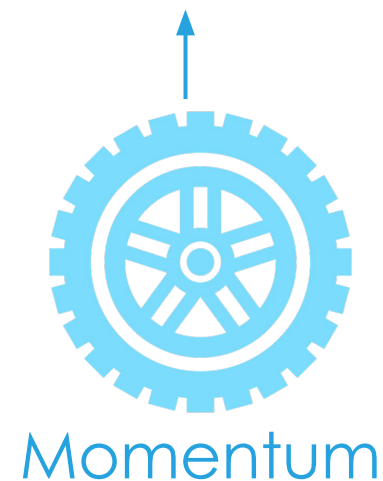
Big step



Optimizers

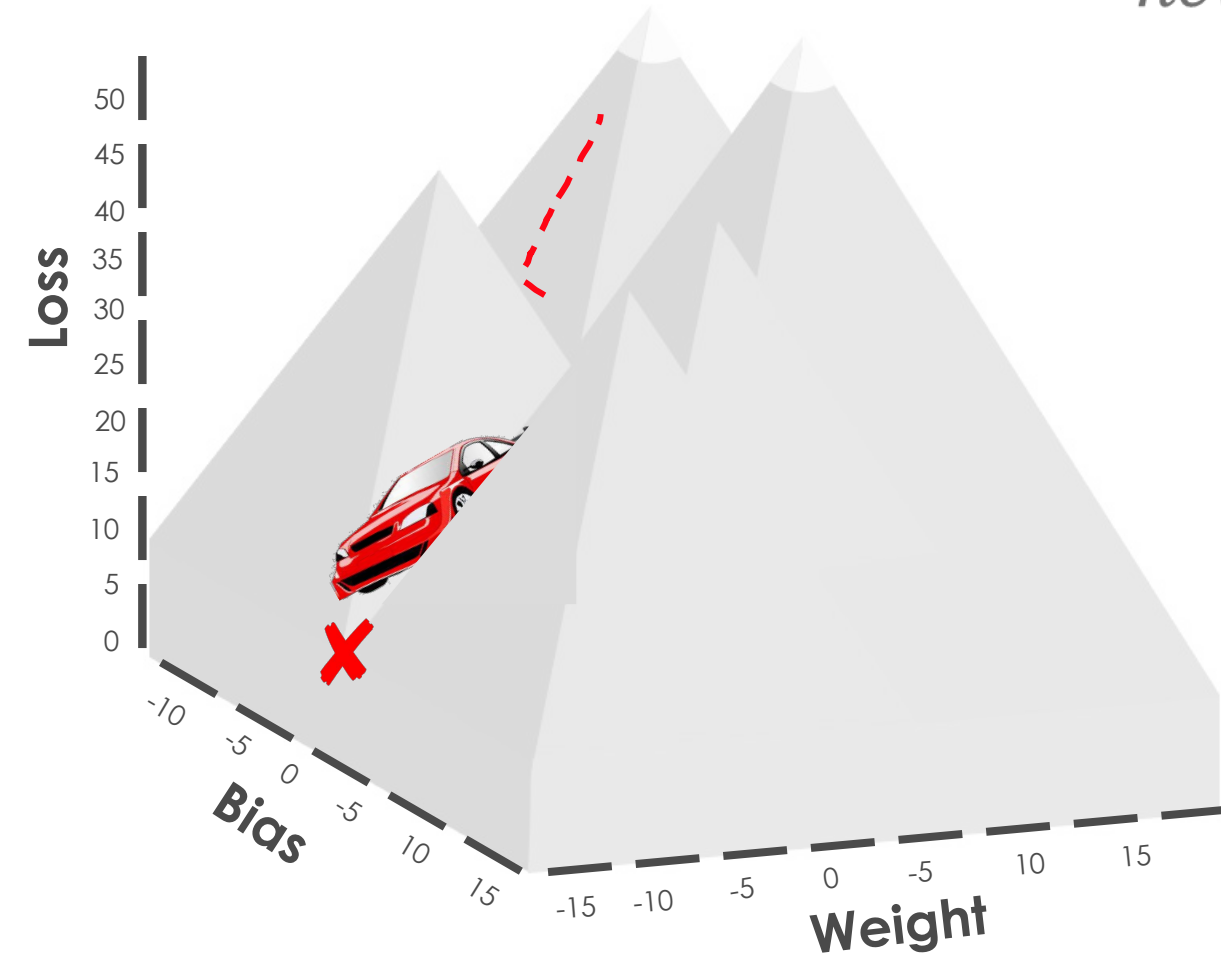
Momentum

$$\text{new} = \text{old} - \text{gradient} * \text{learning rate}$$



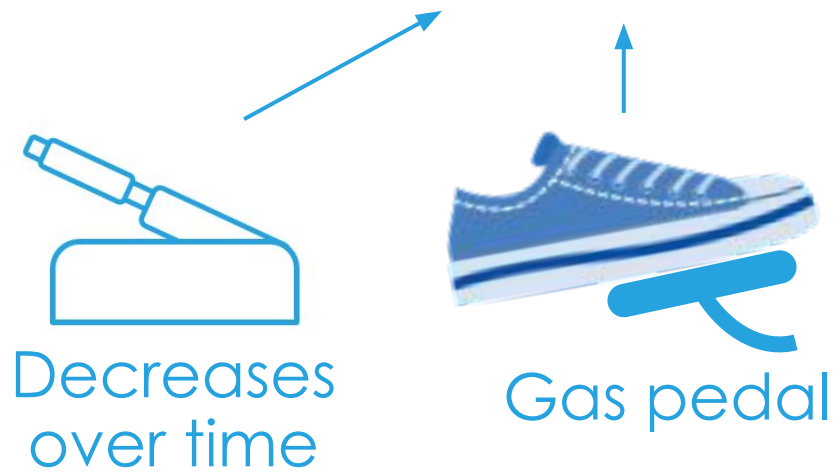
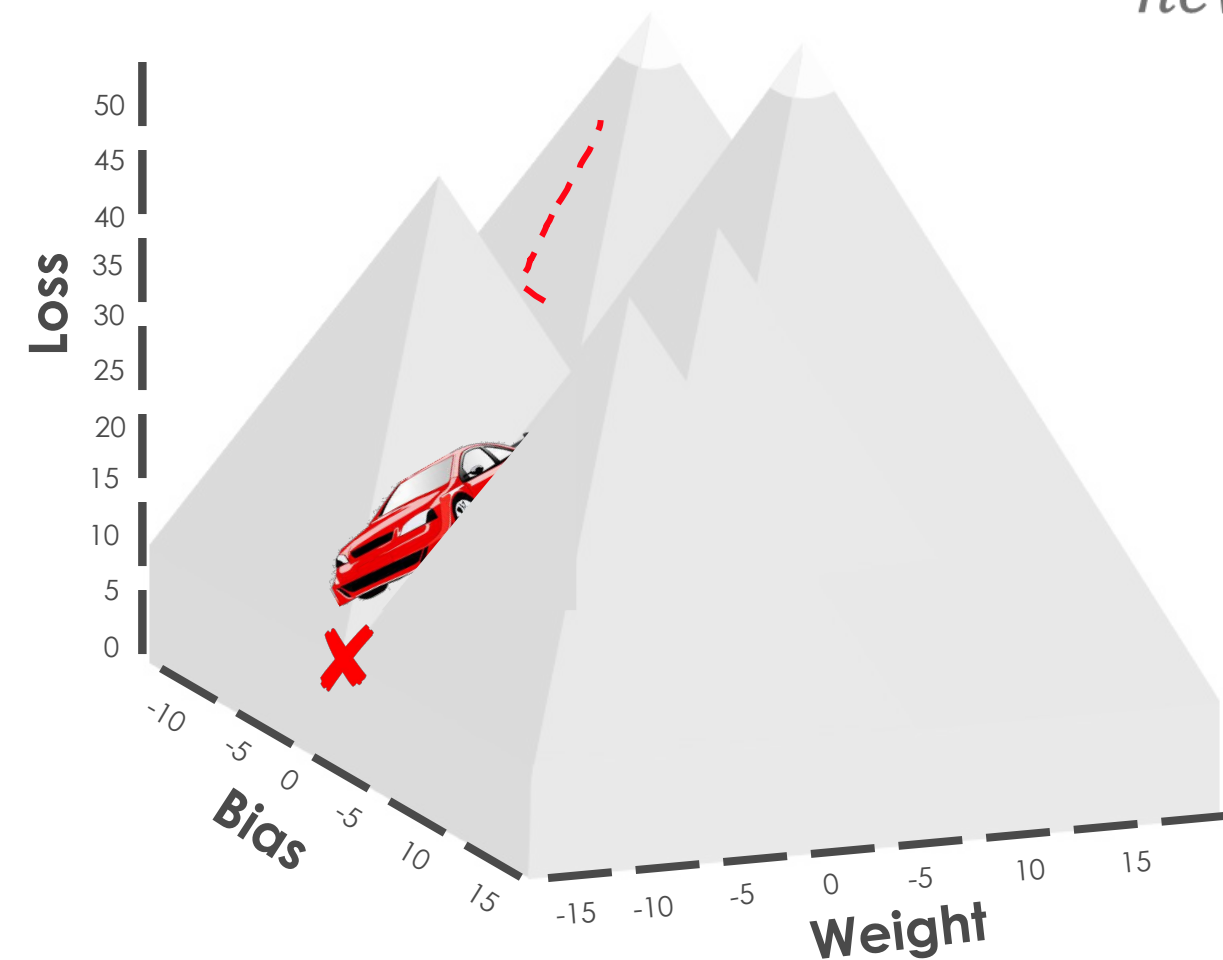
Adagrad

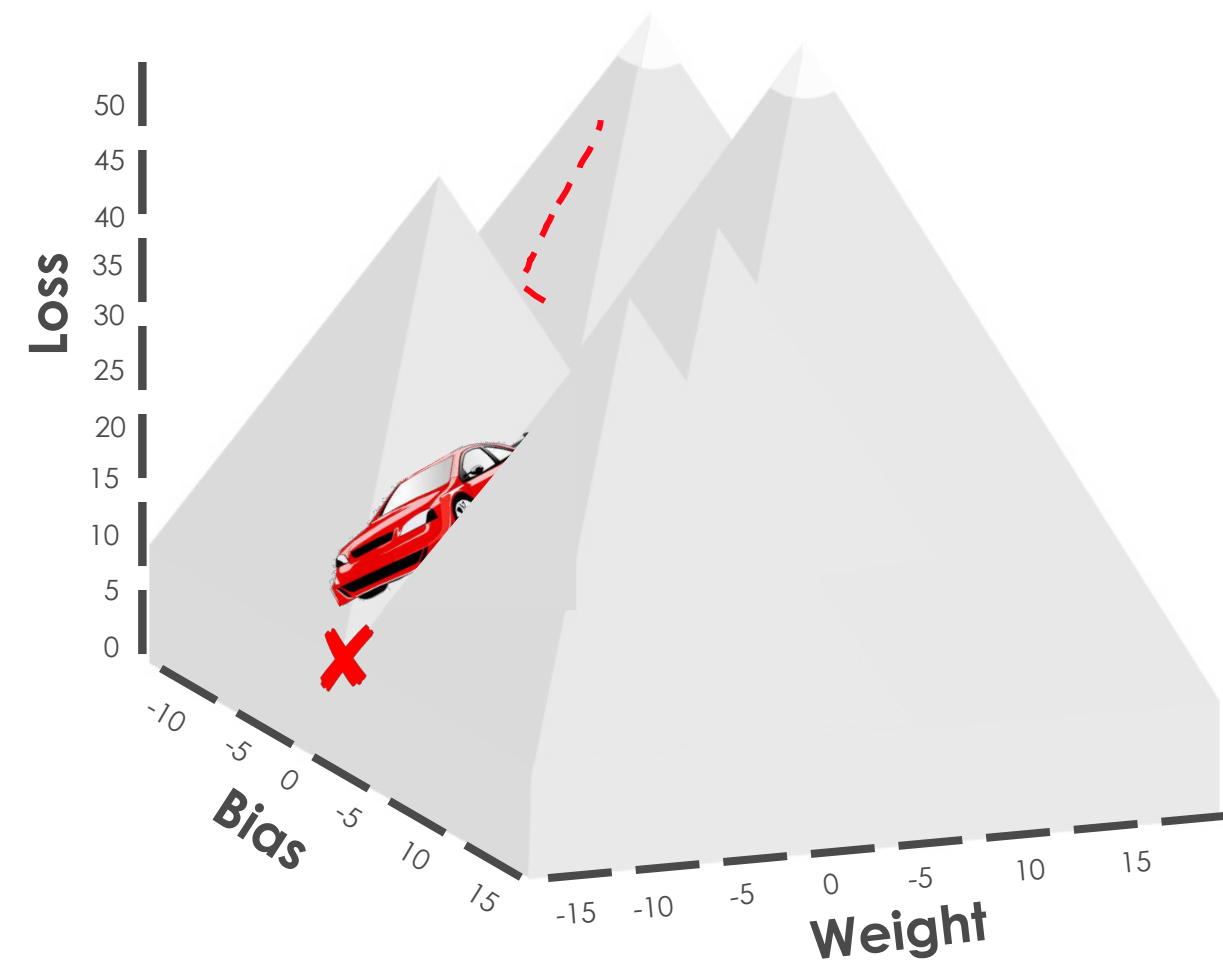
$$\text{new} = \text{old} - \text{gradient} * \text{learning rate}$$



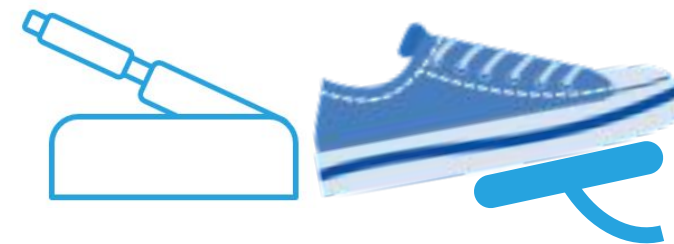
RMSprop

$$\text{new} = \text{old} - \text{gradient} * \text{learning rate}$$





Momentum



RMSprop