Variations of Gradient Descent

Heni Ben Amor, Ph.D.
Assistant Professor
Arizona State University

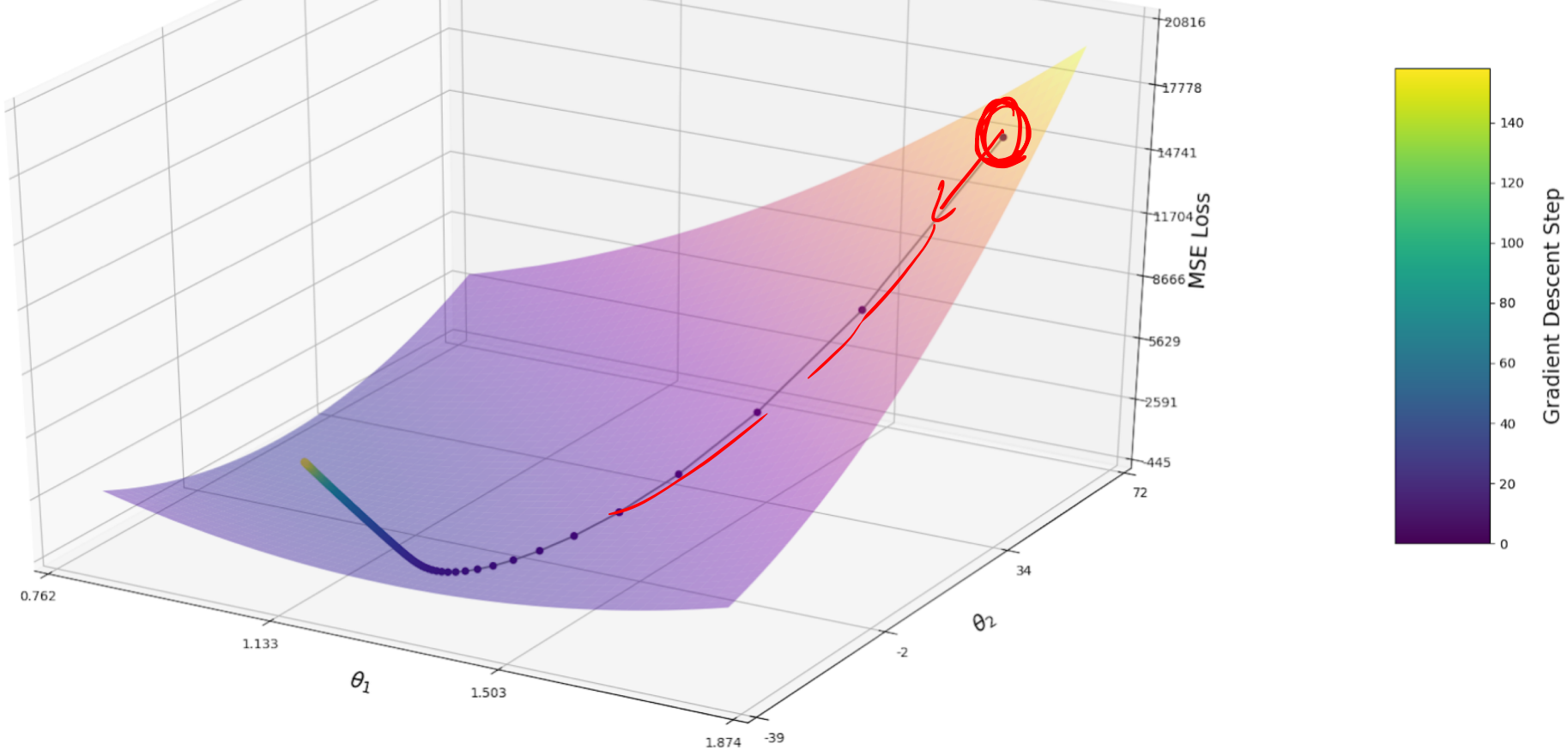
Ira A. Fulton Schools of Engineering
Arizona State University

Gradient Descent Variations

- Batch Gradient Descent
- Stochastic Gradient Descent
- Minibatch Gradient Descent
- Using Momentum
- Resilient Backpropagation (RProp)
- Adaptive Moment Estimation (Adam)
- Many more…

Batch Gradient Descent

- Single update over all training data.
- Gradient is the mean of each sample's gradient.



Batch Gradient Descent

- Single update over all training data.
- Gradient is the mean of each sample's gradient.

inputs outputs

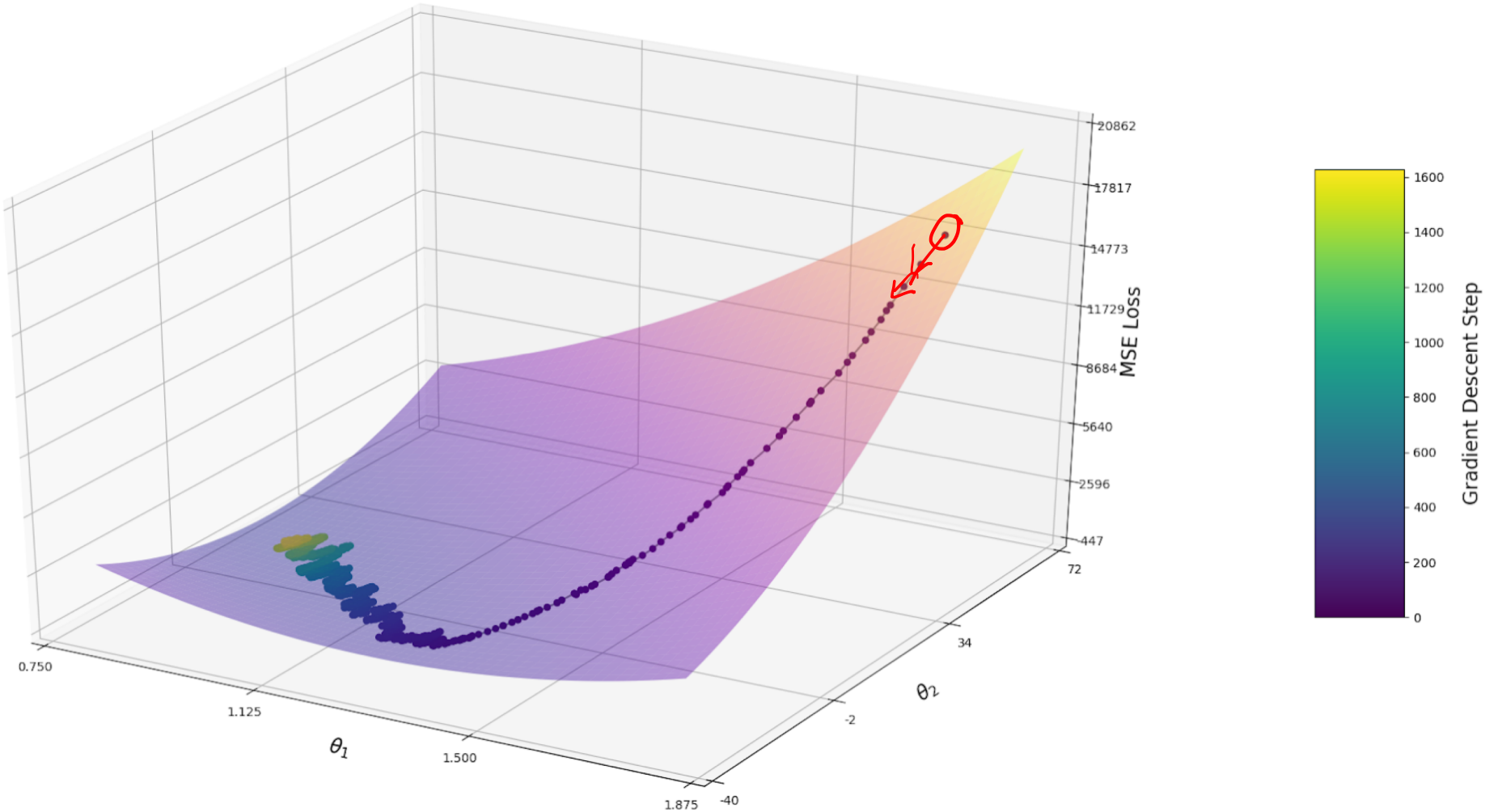
$$\boxed{\theta_{t+1}} = \theta_t - \alpha \nabla \mathcal{L}(\theta_t; \mathbf{X}, \mathbf{Y})$$

$$\nabla \mathcal{L}(\theta; \mathbf{X}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} \nabla \mathcal{L}(\theta; \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$$

data points

Stochastic Gradient Descent (SGD)

- Take a step for each shuffled training sample.



Stochastic Gradient Descent (SGD)

- **Take a step for each shuffled training sample.**

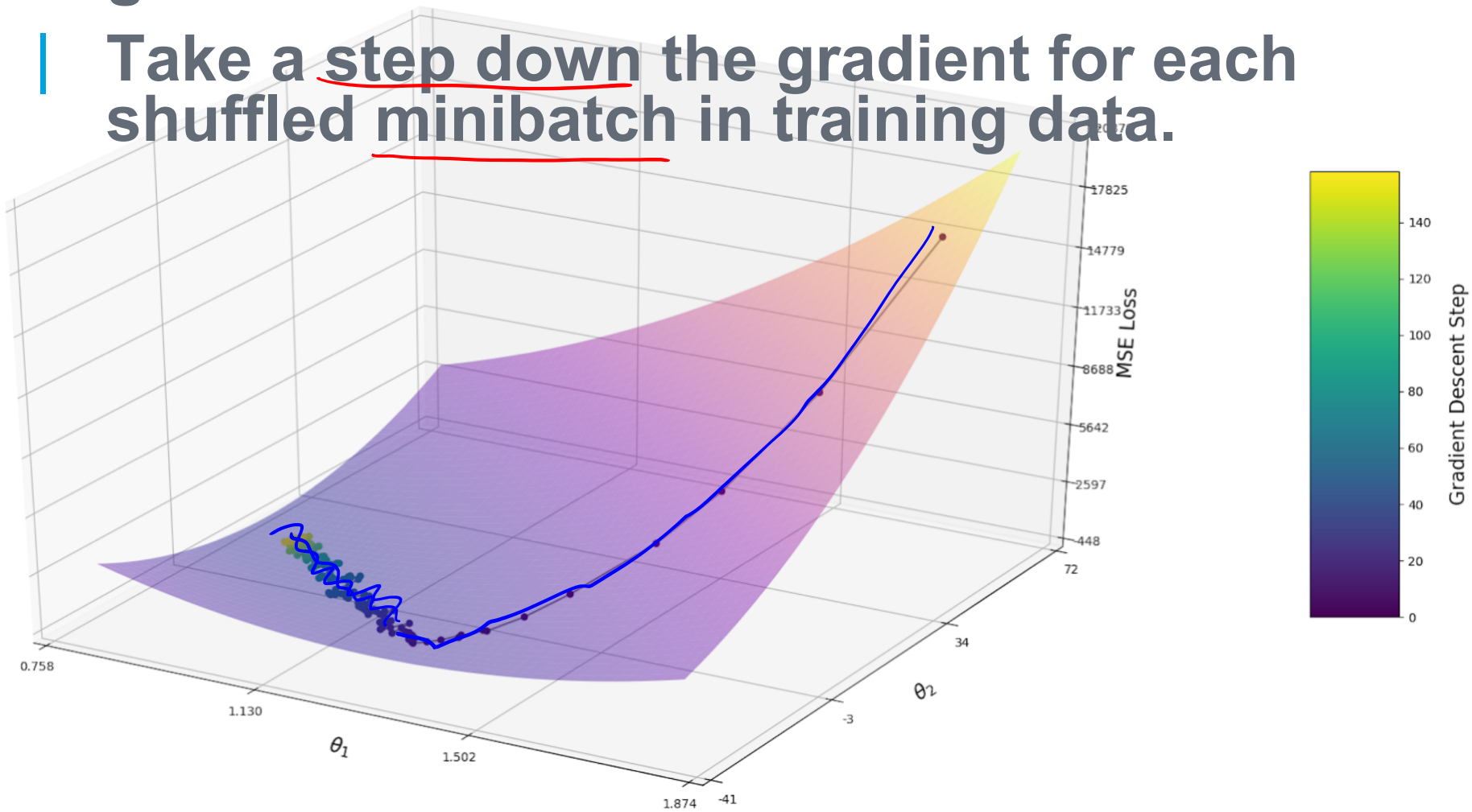
$$\theta_{t+1} = \theta_t - \alpha \nabla \underbrace{\mathcal{L}(\theta_t; \mathbf{x}^{(i)}, \mathbf{y}^{(i)})}$$

points : N

single point : i

Minibatch Gradient Descent

- Middle ground between SGD and batch gradient descent.
- Take a step down the gradient for each shuffled minibatch in training data.



Minibatch Gradient Descent

- Middle ground between SGD and batch gradient descent.
- Take a step down the gradient for each shuffled minibatch in training data.

100 samples minibatch = 10 samples

$$\theta_{t+1} = \theta_t - \alpha \nabla \mathcal{L}(\theta_t; \mathbf{x}^{(mb)}, \mathbf{y}^{(mb)})$$

$$\text{minibatch(mb)} = \text{range}(\text{i} \cdot batchsize, (i+1)batchsize)$$

Minibatch Gradient Descent

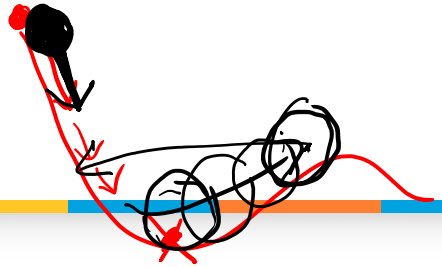
- Middle ground between SGD and batch gradient descent.
- Take a step down the gradient for each shuffled minibatch in training data.

$M \ll N$

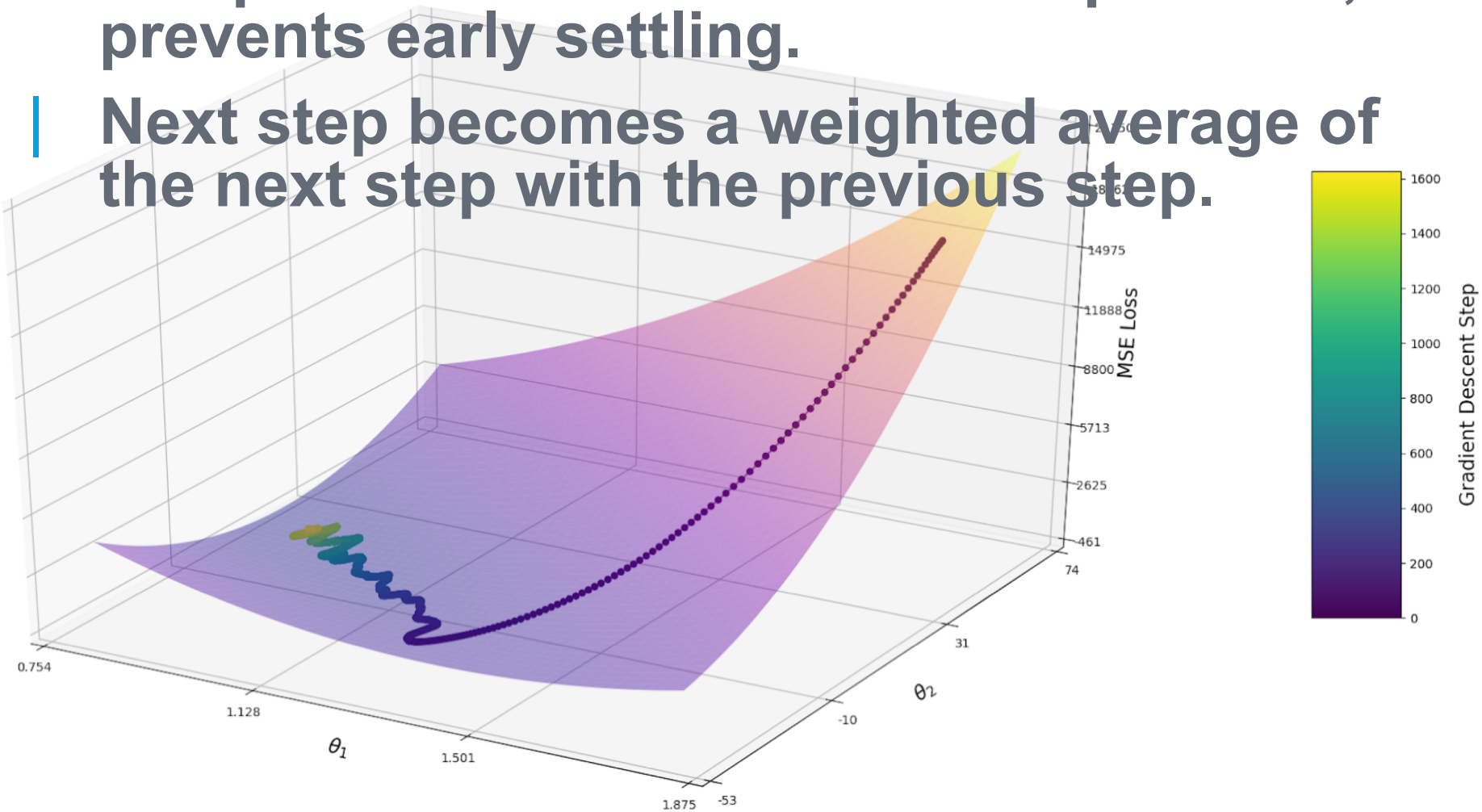
$$\nabla \mathcal{L}(\theta; \mathbf{x}^{(mb)}, \mathbf{y}^{(mb)}) = \frac{1}{M} \sum_{j=1}^{M} \nabla \underbrace{\mathcal{L}(\theta; \mathbf{x}^{(j)}, \mathbf{y}^{(j)})}$$

$$\mathbf{x}^{(j)}, \mathbf{y}^{(j)} \in \mathbf{x}^{(mb)}, \mathbf{y}^{(mb)}$$

SGD with Momentum



- A method to accelerate descent and dampen the oscillations SGD is prone to, prevents early settling.
- Next step becomes a weighted average of the next step with the previous step.



SGD with Momentum

- A method to accelerate descent and dampen the oscillations SGD is prone to, prevents early settling.
- Next step becomes a weighted average of the next step with the previous step.

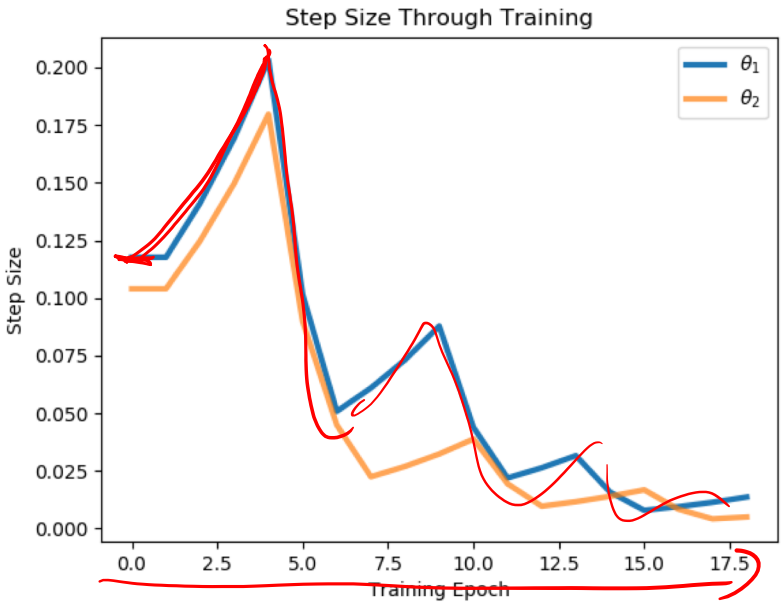
$$\theta_{t+1} = \boxed{\theta_t} - \boxed{g_t}$$

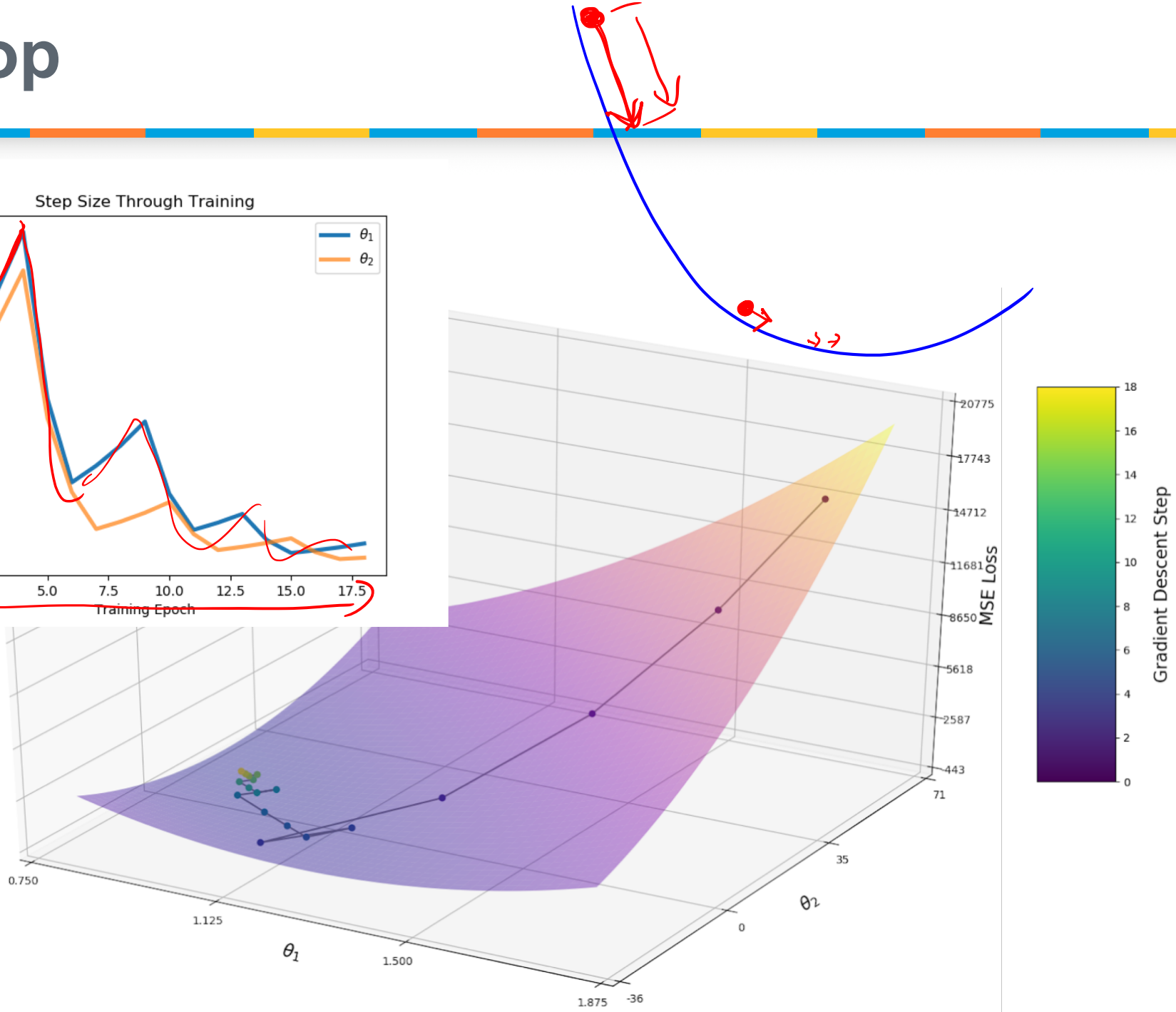
$$g_t = \boxed{\beta} \cdot g_{t-1} + \boxed{(1-\beta)} \underbrace{\alpha \nabla \mathcal{L}(\theta_t; \mathbf{x}^{(i)}, \mathbf{y}^{(i)})}_{g_t}$$

$$g_0 = \mathbf{0}$$

Often $\beta = .9$

RProp


Step Size Through Training
Step Size
Training Epoch
θ_1
θ_2
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


MSE Loss
20775
17743
14712
11681
8650
5618
2587
-443
θ_1
0.750
1.125
1.500
1.875
θ_2
-36
0
35
71
Gradient Descent Step
18
16
14
12
10
8
6
4
2
0

RProp

- Ignores the magnitude of the gradient
- Only looks at the gradient's sign for each parameter

$$sign\left(\frac{\partial \mathcal{L}}{\partial \theta_i}\right) =$$

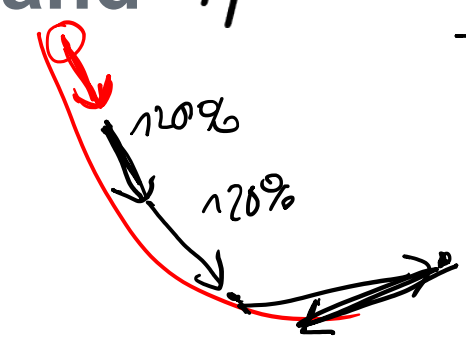
sign $\left(\frac{\partial \mathcal{L}}{\partial \theta_i}\right)_{t-1}$	+	−
+	+	−
−	−	+

RProp

$$\begin{bmatrix} w_1 \\ \vdots \\ w_{10} \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \\ \eta_{10} \end{bmatrix}$$

- Magnitude for each parameter is determined by an update value η_{θ_i}
- If the gradient sign of the parameter θ_i is the same as in the previous step taken, increase η_{θ_i} by scaling by η^+, otherwise scale by η^-.
- Often $\eta^+ = 1.2$ and $\eta^- = 0.5$

1.2

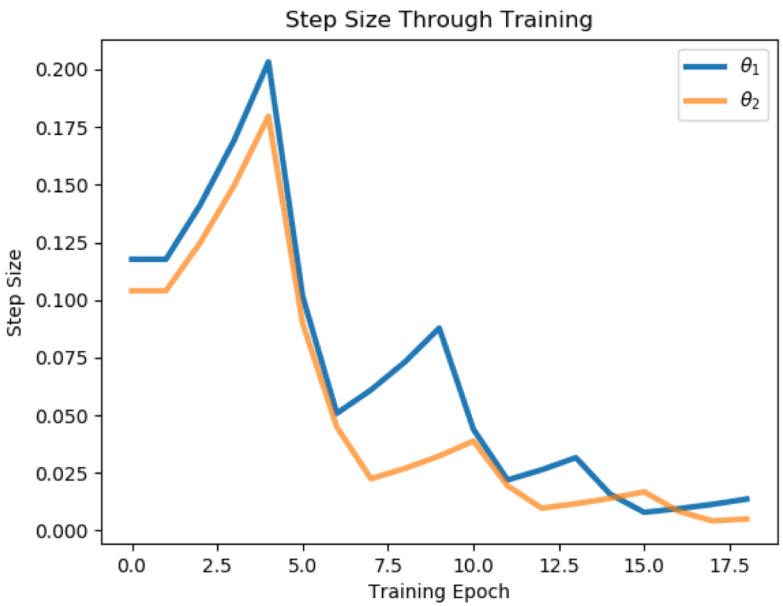


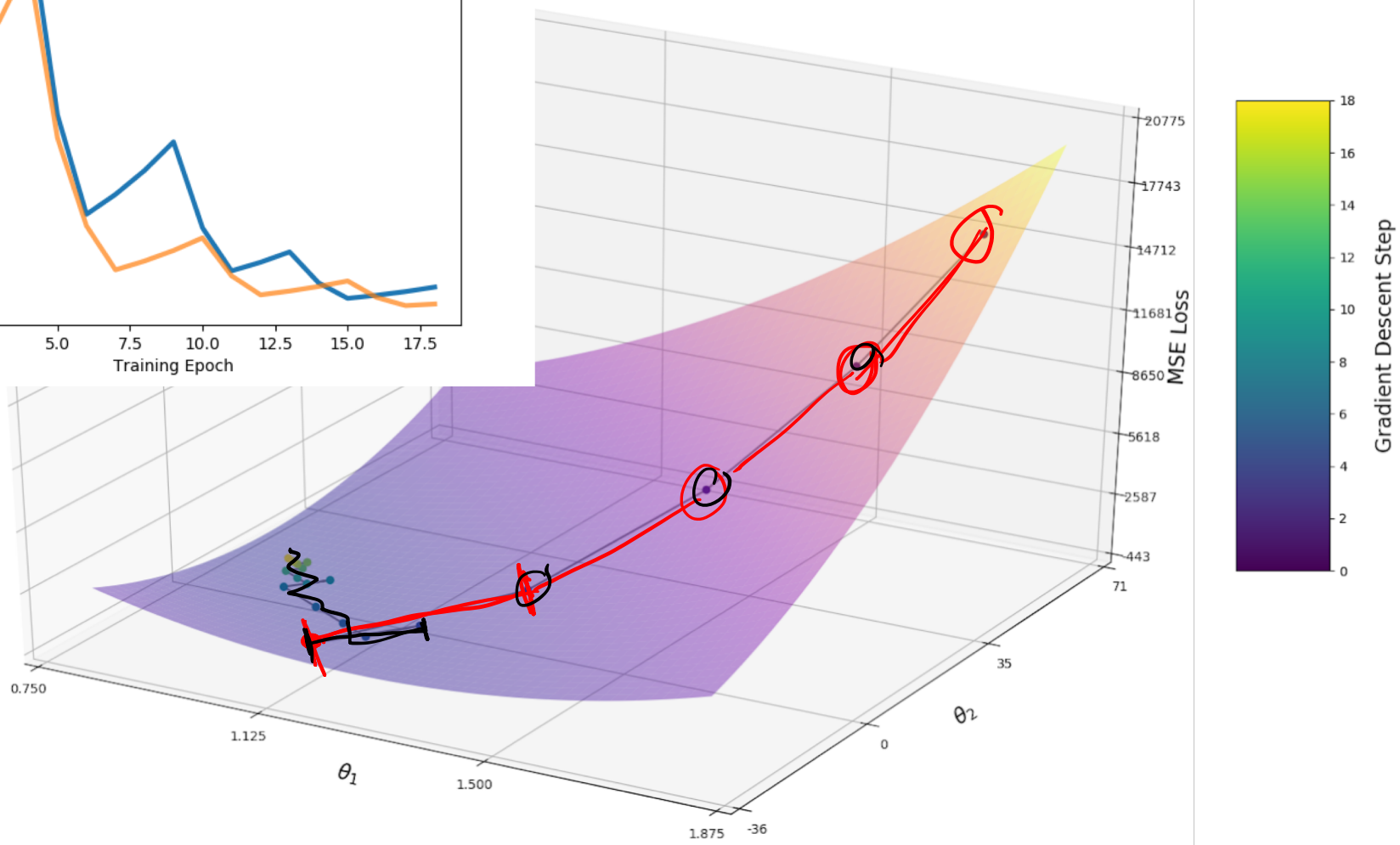
RProp

- Intuition: increase step size in direction where a local minima has not been crossed, decrease step size when crossing local minima.
- Step for each parameter becomes:

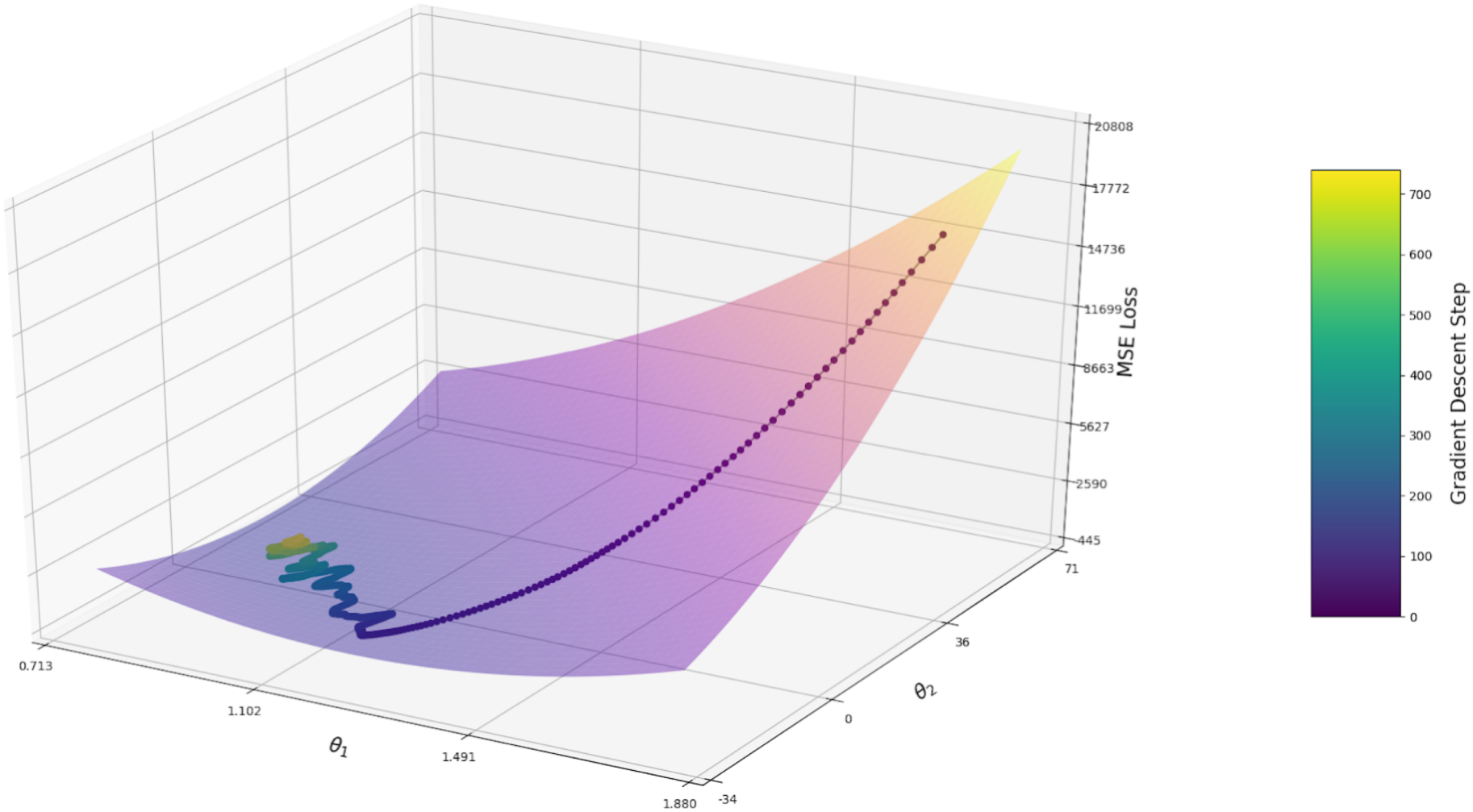
$$\text{step}_{\theta_i} = sign\left(\frac{\partial \mathcal{L}}{\partial \theta_i}\right) \cdot \eta_{\theta_i}$$

RProp


Step Size Through Training
Step Size
Training Epoch
θ_1
θ_2
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


MSE Loss
20775
17743
14712
11681
8650
5618
2587
443
θ_1
0.750
1.125
1.500
1.875
θ_2
-36
0
35
71
Gradient Descent Step
18
16
14
12
10
8
6
4
2
0

Adam


20808
17772
14736
11699
8663
5627
2590
445
MSE Loss
0.713
1.102
1.491
1.880
θ_1
-34
0
36
71
θ_2
700
600
500
400
300
200
100
0
Gradient Descent Step

Adam

- Adam uses estimates of the gradients' first and second moments, mean and uncentered variance respectively, to compute adaptive learning rates.
- First moment estimate:

$$\mathrm{m}_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla\mathcal{L}$$

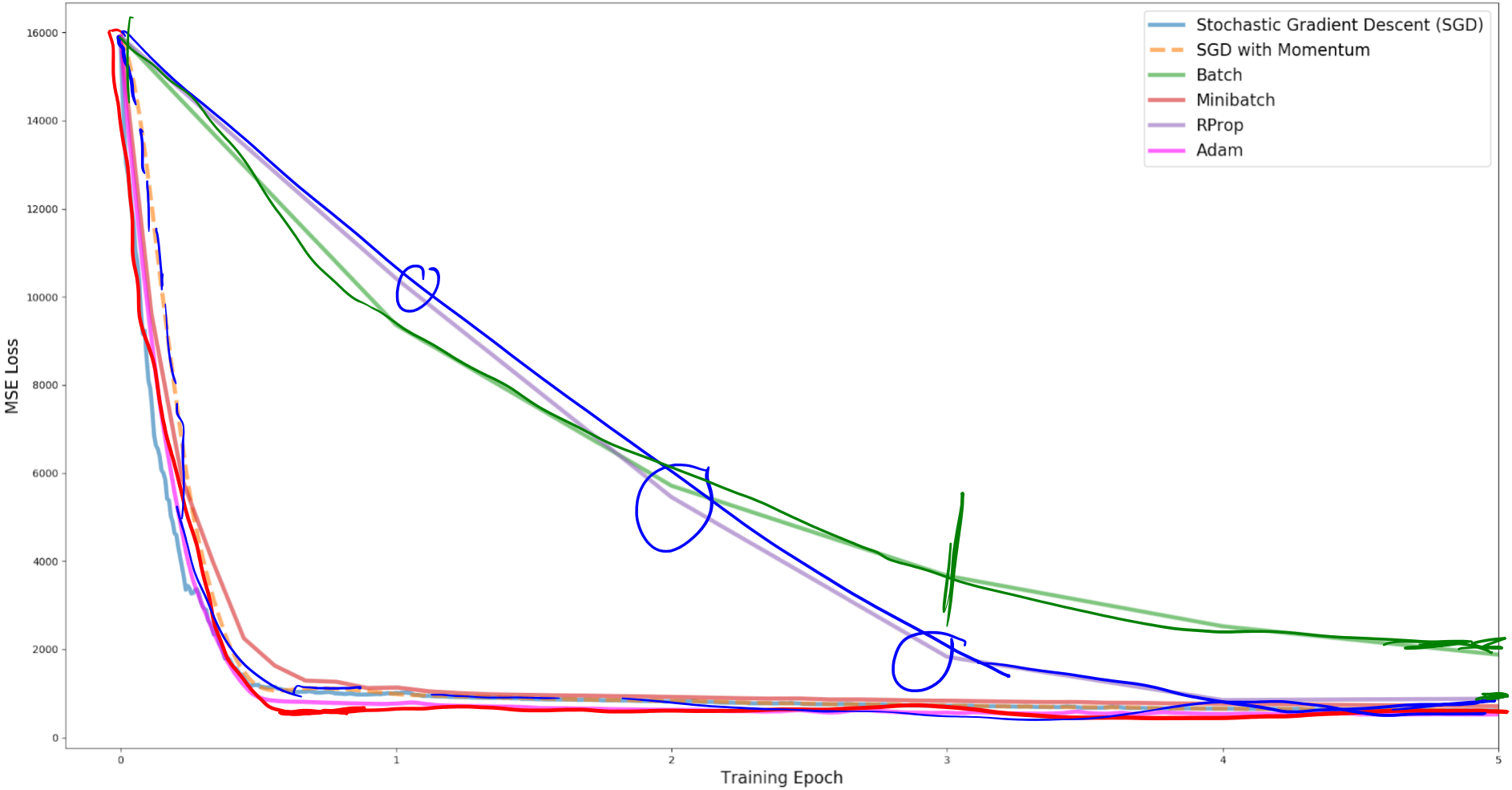
current time

- Second moment estimate:

$$\mathrm{v}_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla\mathcal{L})^2$$

$$\mathrm{m}_0 = \mathbf{0} \quad v_0 = \mathbf{0}$$

Training Loss Comparison



Summary

- **Gradient descent variations**
- **Different sample size methods:**
  - Batch
  - Stochastic
  - Minibatch
- **Adaptive learning rate methods:**
  - Momentum
  - RProp
  - Adam