Optimization

Seyoung Yun

References

- http://lear.inrialpes.fr/workshop/osl2013/slides/osl2013_bach.pdf
- https://github.com/abursuc/dldiy-practicals/blob/master/slides/Slide_October19.pdf
- http://cs231n.github.io/optimization-1/
- http://ruder.io/optimizing-gradient-descent/

Training Algorithm

- Loss Function : L2 ⇐ MSE | cross-entropy
  L1 ⇐ ABS | hinge loss

$$L(\beta) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(Y^{(i)}, \widehat{Y}^{(i)}(\beta)\right)$$

True label

obtained from your model

- How to optimize it?

  - Gradient Descent: $\beta(t+1) = \beta(t) - \gamma(t)\nabla L(\beta(t))$

  - Some variants of GD

  - Stochastic gradient descent and its variants

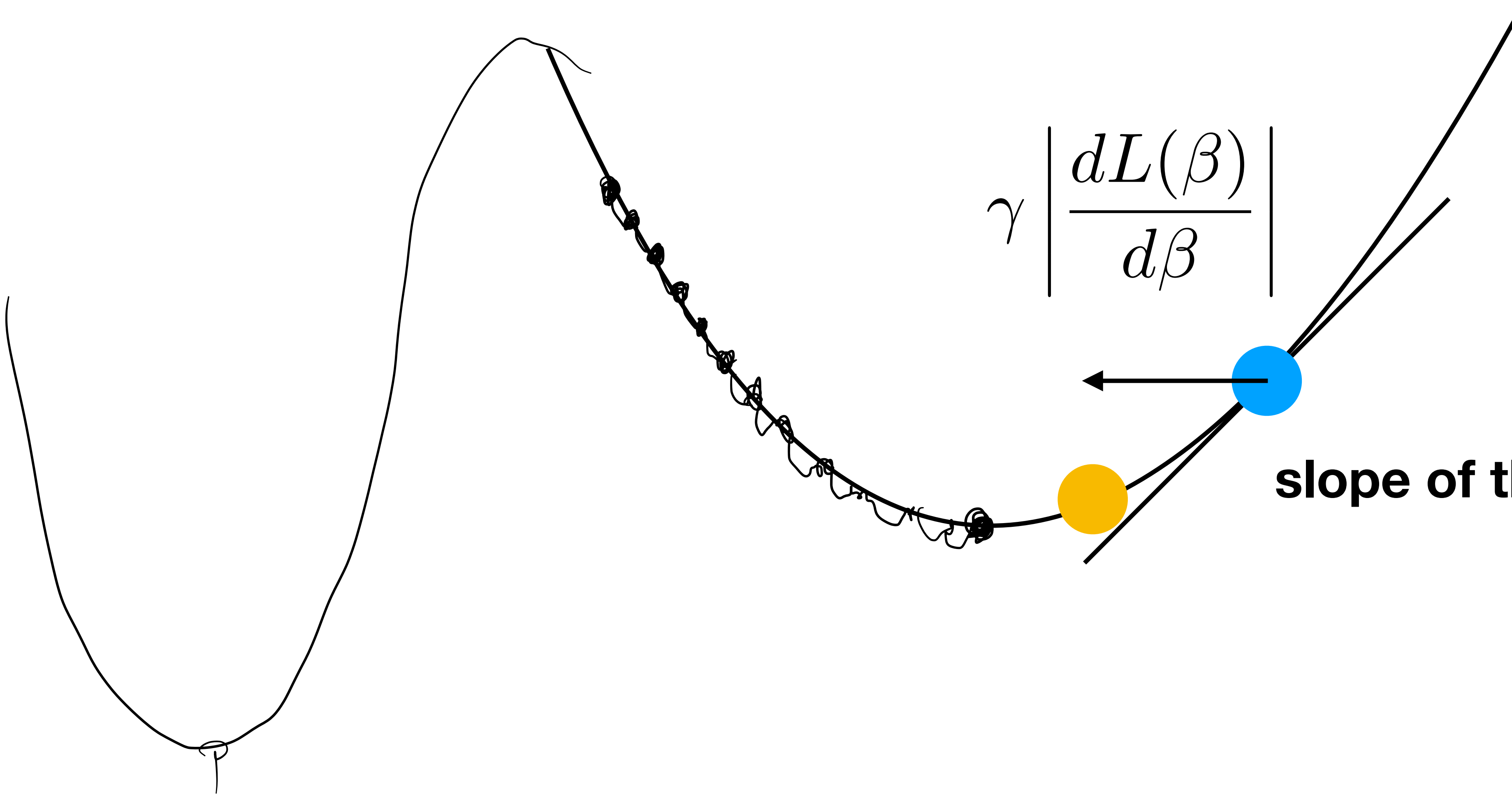
Gradient

$$\nabla L(\beta) = \begin{bmatrix} \frac{\partial L}{\partial \beta_1} \\ \frac{\partial L}{\partial \beta_2} \\ \vdots \\ \frac{\partial L}{\partial \beta_d} \end{bmatrix}$$

- 1-D example
  - Gradient: $$\frac{dL(\beta)}{d\beta} = \lim_{h \to 0} \frac{L(\beta + h) - L(\beta)}{h}$$

sign of gradient

$- \quad +$

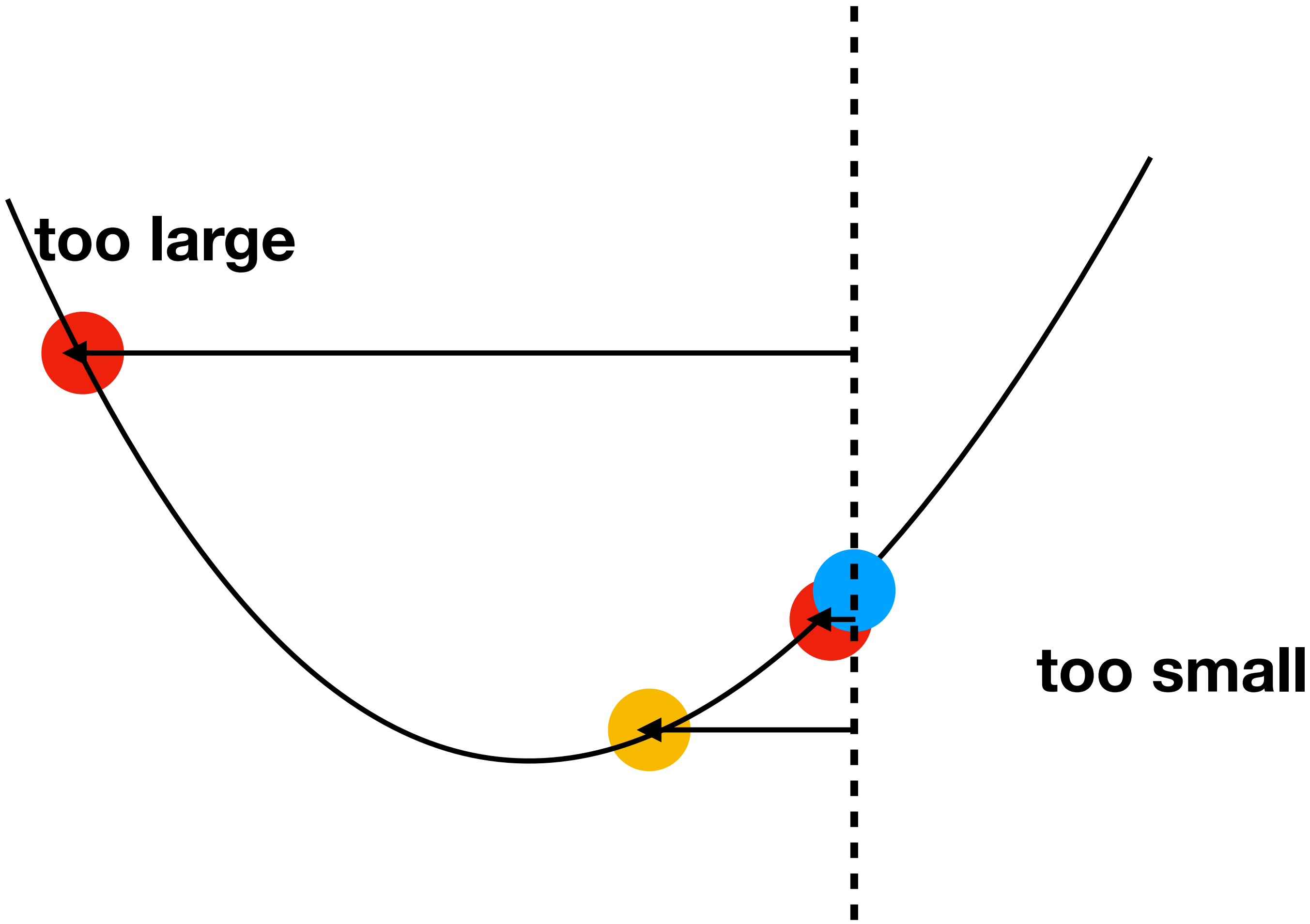


Step Size = learning rate

$$\beta(t+1) = \beta(t) - \underset{\substack{\uparrow \\ \text{step size}}}{\underline{\gamma}} \cdot \nabla L(\beta(t))$$

$\gamma \left| \frac{dL(\beta)}{d\beta} \right|$: the length of the update

$$\gamma_t = \frac{1}{\sqrt{t}} \text{ or } \frac{1}{t}$$



2D cases

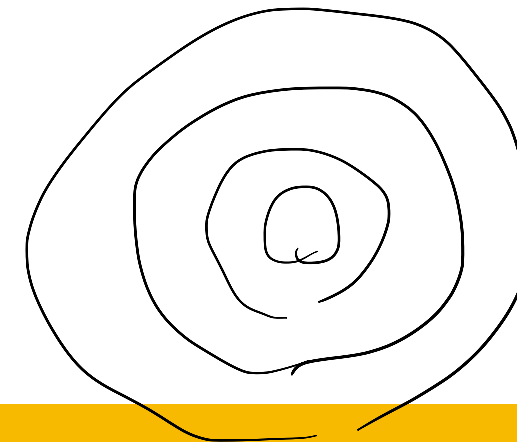
- Gradient : $\nabla L(\beta) = \begin{bmatrix} \frac{\partial L(\beta)}{\partial \beta_1} \\ \frac{\partial L(\beta)}{\partial \beta_2} \end{bmatrix}$ $-\gamma \nabla L(\beta)$

- Gradient Descent

- Zigzag: why? it is very hard to find a step size than is good for multi dimensional cases (vertical vs. horizontal)

Gradient descent vs Newton

$$\beta(t+1) = \beta(t) - \gamma(t)\nabla L(\beta(t))$$ **vs.**



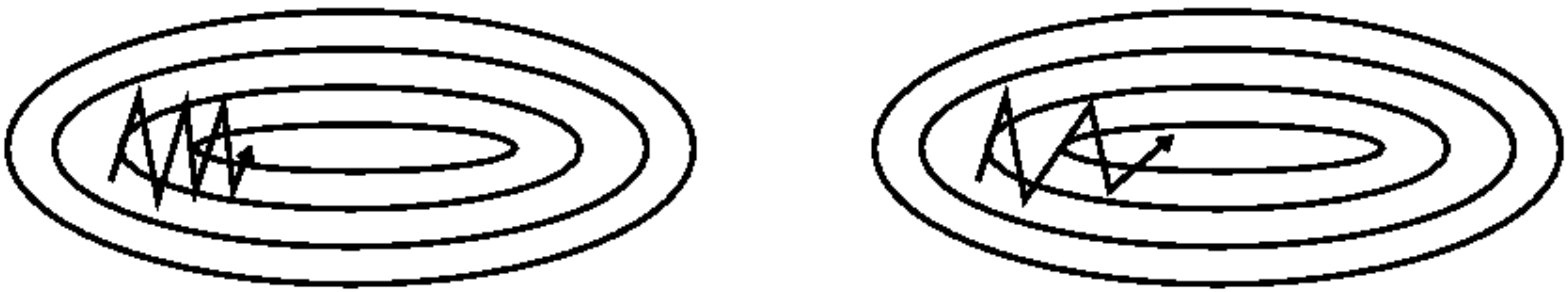
Newton

$$\beta(t+1) = \beta(t) - \gamma\nabla^2 L(\beta(t))^{-1}\nabla L(\beta(t))$$

10^6

- Newton converges faster to local minima memory 10^{12}
  computation 10^{18}
  - The inverse Hessian matrix control the step size adaptively
- No one want to compute a Hessian (or worst: inverse it)

Momentum



GD **with momentum**

momentum:
$$\Delta\beta(t+1) = \underline{\eta\Delta\beta(t)} + \gamma\nabla f(\beta)$$
$$\beta(t+1) = \beta(t) - \Delta\beta(t+1)$$

(t)

L

$$\Delta\beta(t+1) = (\eta+\gamma)\left(\underbrace{\frac{\eta}{\eta+\gamma}}_{\delta}\Delta\beta(t) + \underbrace{\left(1-\frac{\eta}{\eta+\gamma}\right)}_{(1-\delta)}\nabla L(\beta(t))\right)$$

AdaGrad

$$\frac{1}{c\sqrt{t}}$$

AdaGrad:

$$\beta(t+1) = \beta(t) - \frac{\gamma}{\sqrt{G_t + \varepsilon}} \nabla L(\beta(t))$$

$$G_{t+1} = G_t + (\nabla L(\beta(t)))^2$$

element-wise

$$\frac{1}{t} \cdot G_t = \left(\frac{1}{t} \sum_{i=1}^{t} (\nabla L(\beta(i)))^2 \right)$$

$$G = t \cdot \left(\text{Avg of } (\nabla L(\beta))^2 \right)$$

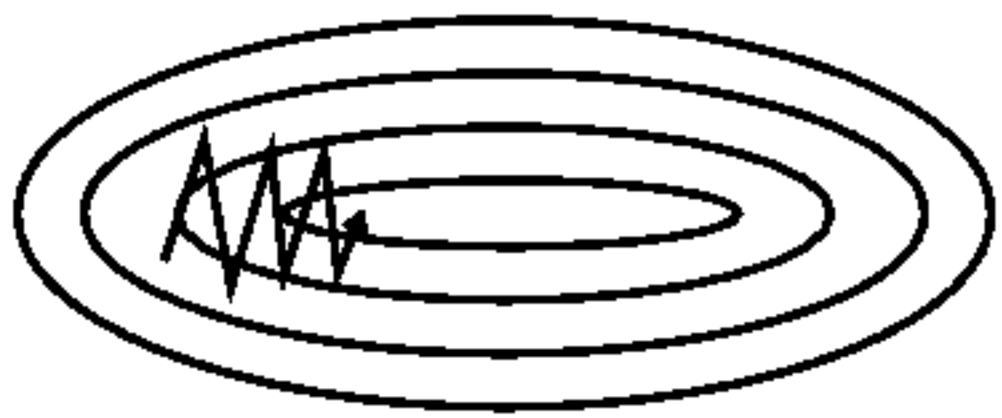
AdaGrad.

G_t

$= t \cdot \text{Avg}.$

RMSProp

$\Rightarrow G_t = \text{Avg}.$

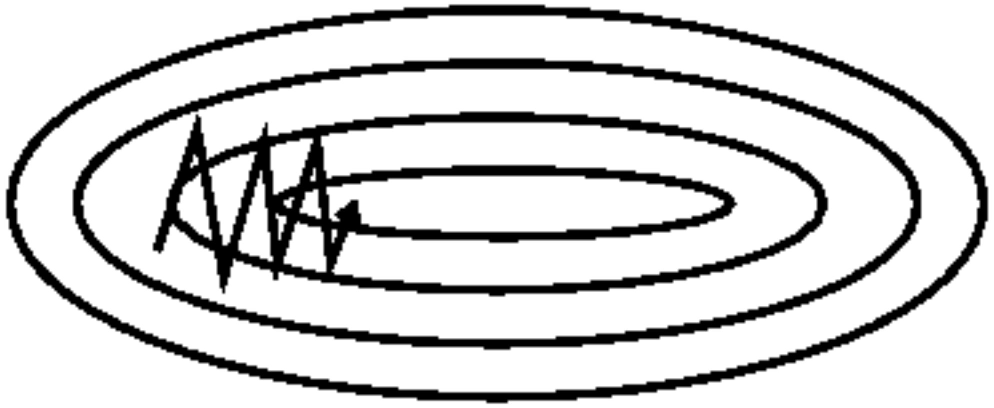


RMSProp: $$G_t = \eta G_{t-1} + (1-\eta)(\nabla L(\beta(t)))^2$$

$$\beta(t+1) = \beta(t) - \frac{\gamma}{\sqrt{G_t + \varepsilon}} \nabla L(\beta(t))$$

Adam

momentum
+
RMS prop.

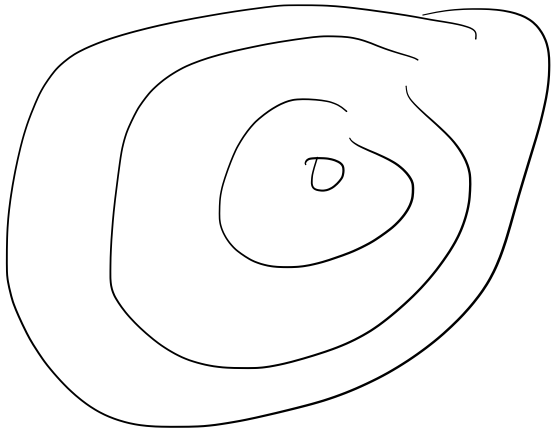


Adam:

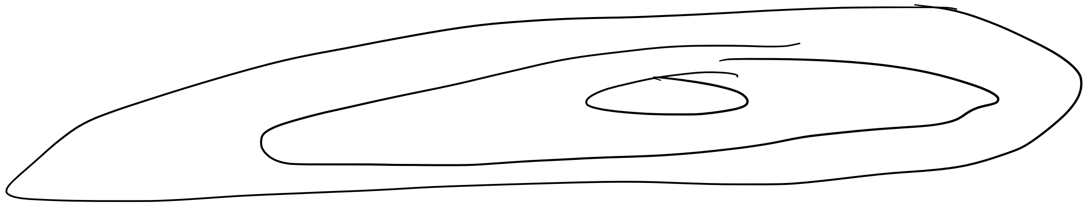
$$M_t = \eta_1 M_{t-1} + (1-\eta_1)\nabla L(\beta(t))$$

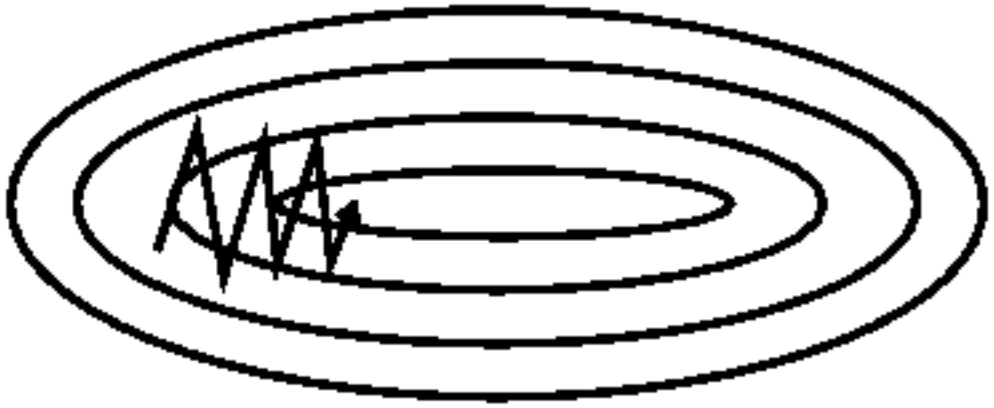
$$G_t = \eta_2 G_{t-1} + (1-\eta_2)(\nabla L(\beta(t)))^2$$

$$\beta(t+1) = \beta(t) - \frac{\gamma}{\sqrt{G_t + \varepsilon}} M_t$$



Adam





Adam:

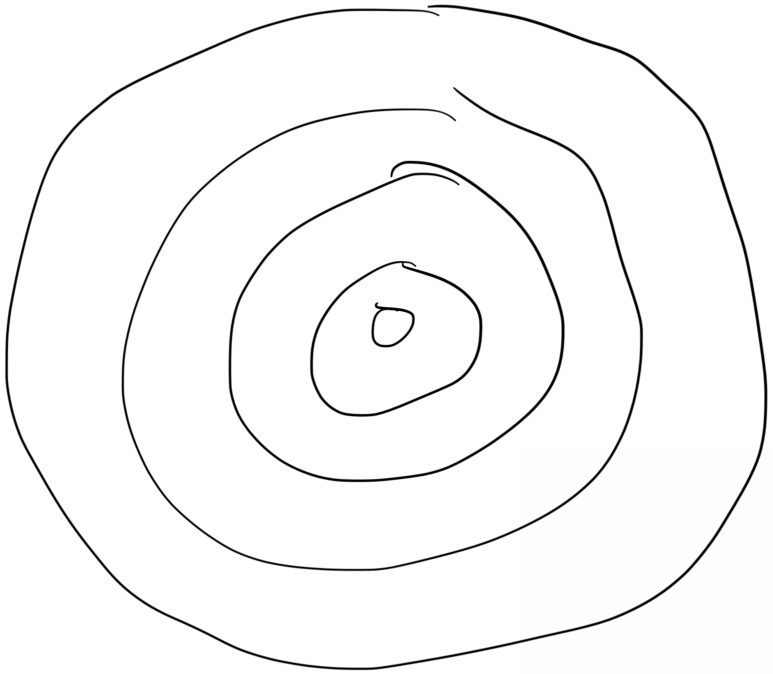
$$M_t = \eta_1 M_{t-1} + (1-\eta_1)\nabla L(\beta(t))$$

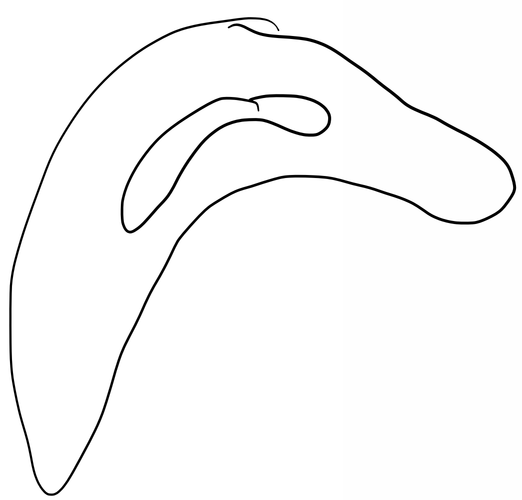
$$G_t = \eta_2 G_{t-1} + (1-\eta_2)(\nabla L(\beta(t)))^2$$

$$\hat{M}_t = \frac{M_t}{1-\eta_1^t} \quad \hat{G}_t = \frac{G_t}{1-\eta_2^t}$$

$$\beta(t+1) = \beta(t) - \frac{\gamma}{\sqrt{\hat{G}_t + \varepsilon}} \hat{M}_t$$

Optimization





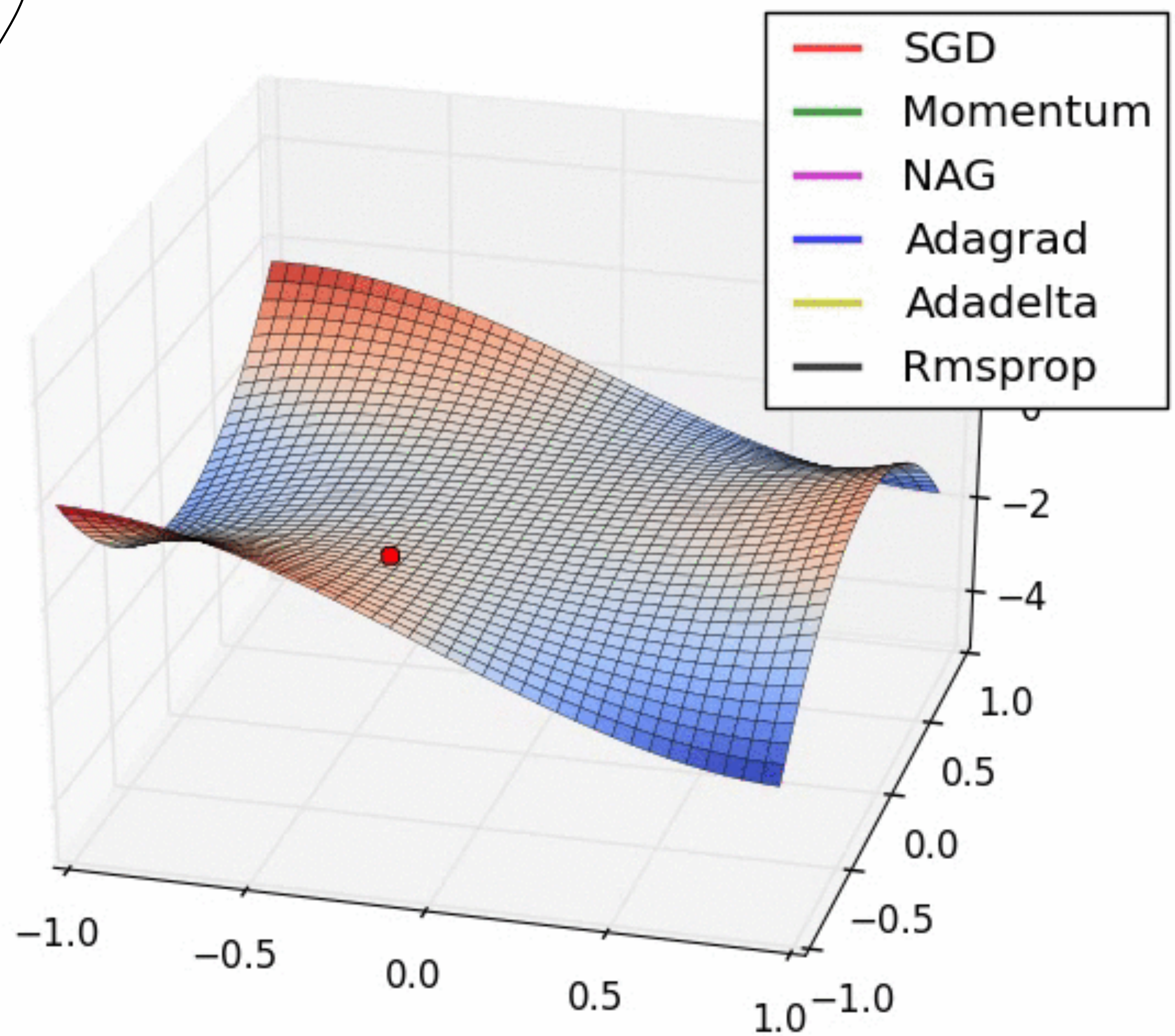


Image by Alec Radford

Optimization

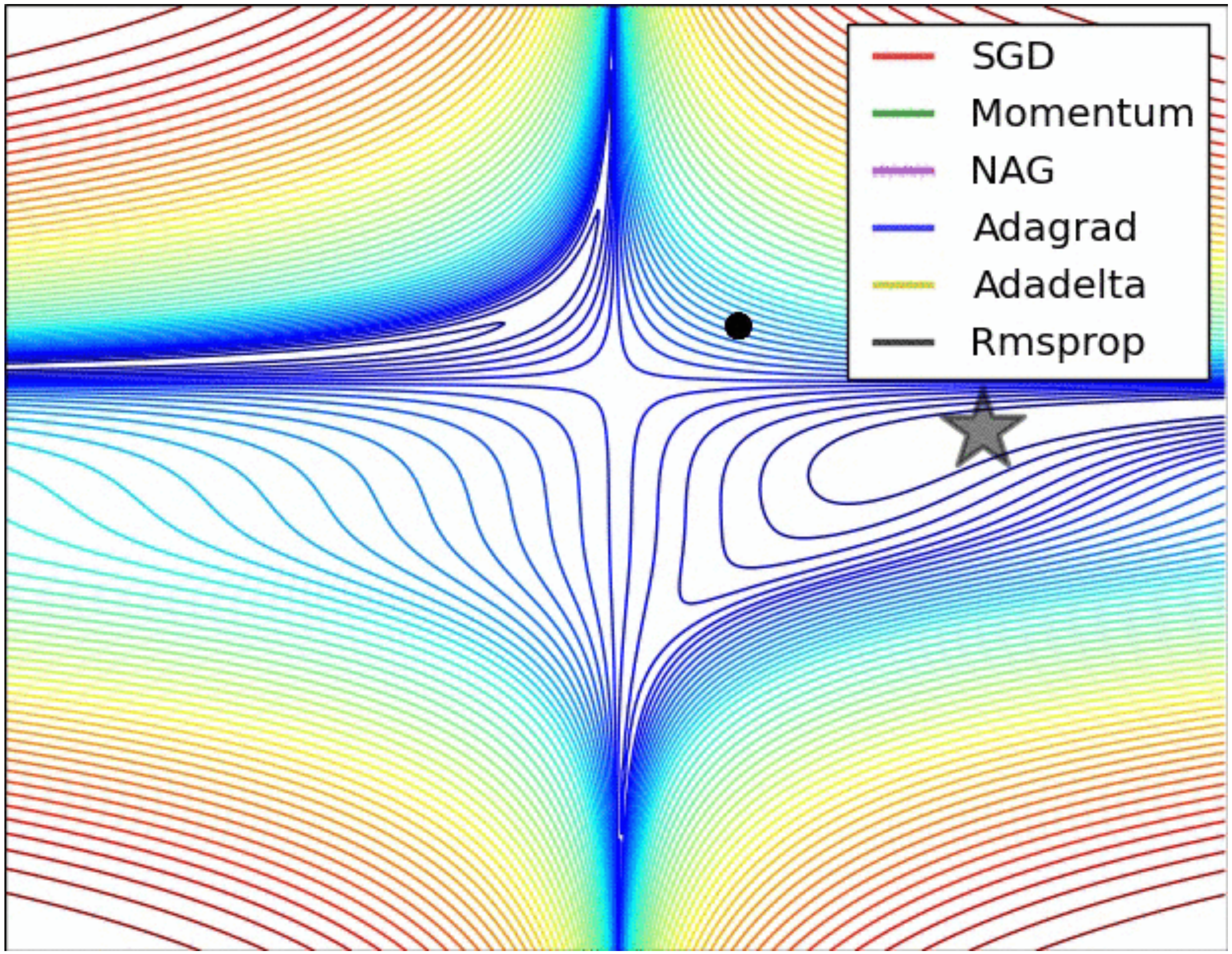


Image by Alec Radford

Batch vs mini-Batch vs SGD

(full)

$$L(\beta) = \frac{1}{n} \cdot \sum_{i=1}^{n} \ell(Y^{(i)}, \hat{Y}^{(i)}(\beta))$$

n #data points

- Batch Gradient Descent

$$\beta \leftarrow \beta - \eta \nabla L(\beta)$$

$$\nabla_\beta L(\beta) = \frac{1}{n} \sum_{i=1}^{n} \nabla_\beta \ell(Y^{(i)}, \hat{Y}^{(i)}(\beta))$$

- Mini-batch Gradient Descent

$$\beta \leftarrow \beta - \eta \nabla \left(\frac{1}{B} \sum_{\tau=1}^{B} \ell(Y^{(i_\tau)}, \widehat{Y}^{(i_\tau)}(\beta)) \right)$$

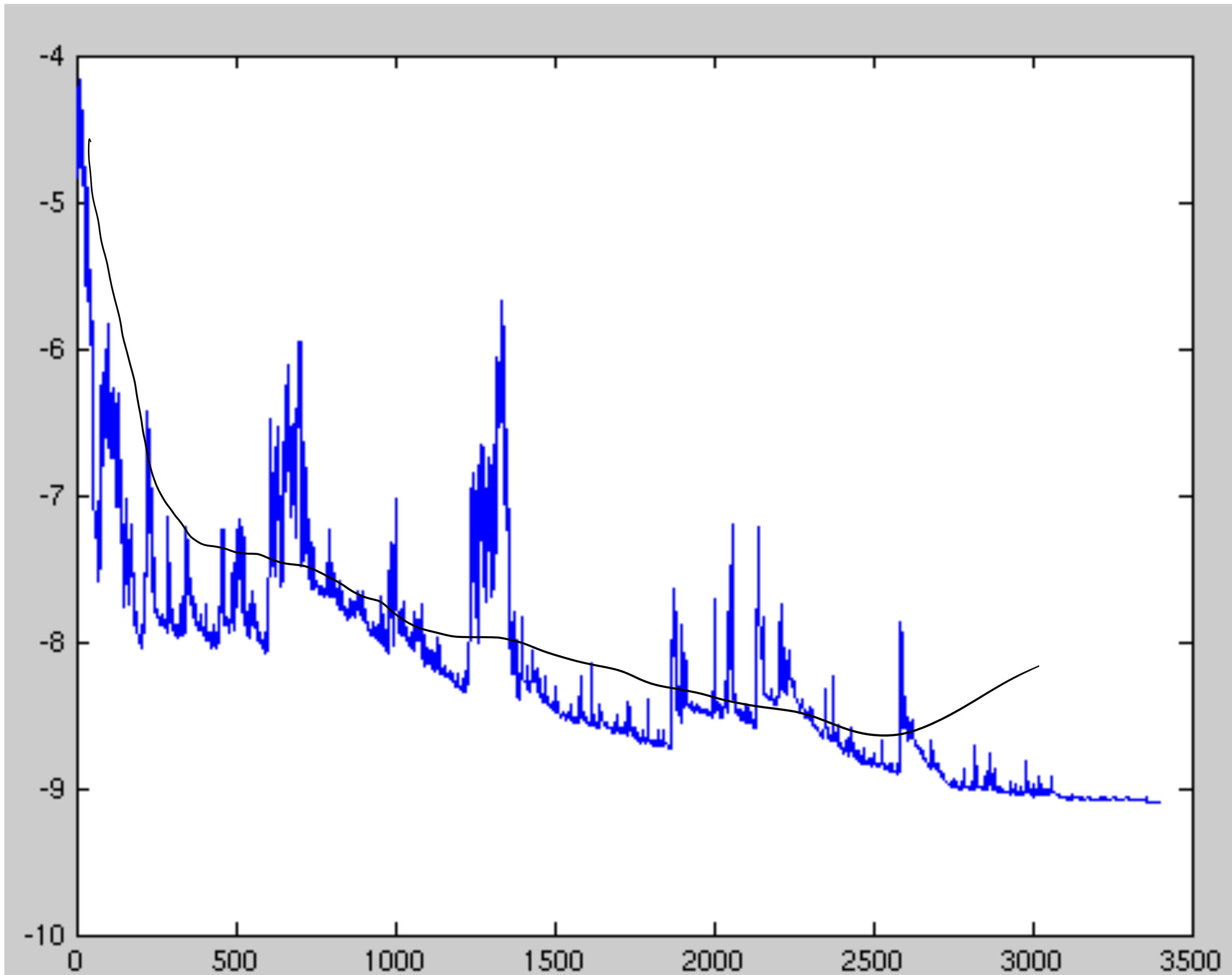
$B = 16, 8$
$32, 4$

- Stochastic Gradient Descent $(B = 1)$

$$\beta \leftarrow \beta - \eta \nabla \ell(Y^{(i)}, \widehat{Y}^{(i)}(\beta))$$

$$E\left[\frac{1}{B} \nabla \sum_{\tau=1}^{B} \ell(Y^{(i_\tau)}, \hat{Y}^{(i_\tau)}(\beta)) \right] = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(Y^{(i)}, \hat{Y}^{(i)}(\beta))$$

SGD fluctuation



$\beta \uparrow$

$\gamma \downarrow$

Learning rate decay

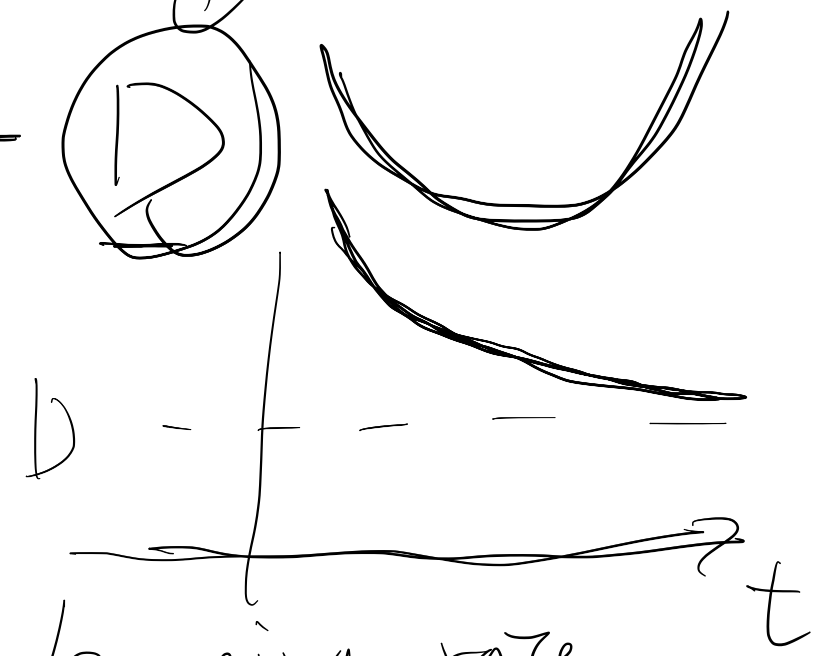
variance of each movement

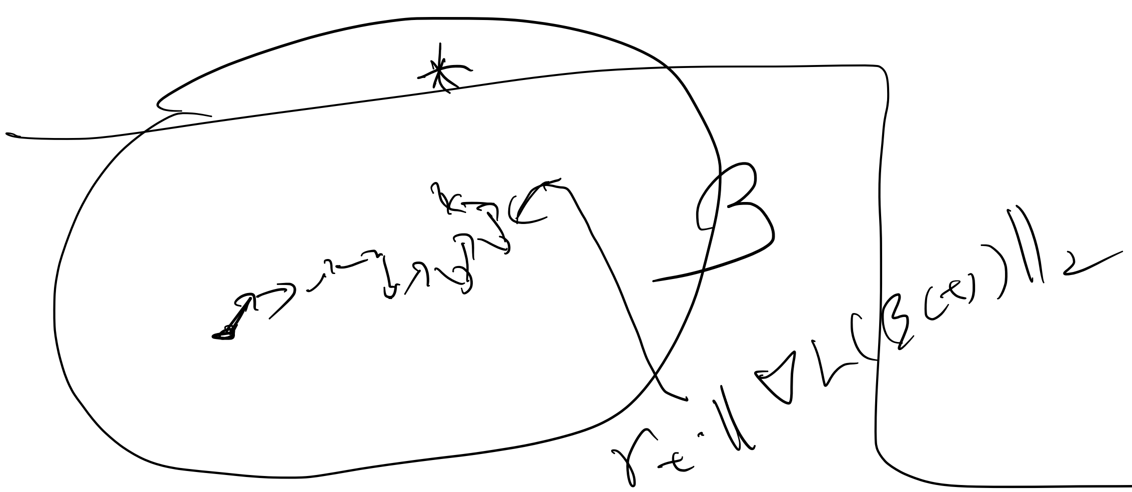
- Now, eta is a function of the number of iterations, epochs,

$$\mathbb{E}[f(x_t)] - f(x^*) \le \delta^t \cdot C + D$$

$0 < \delta < 1$

function of learning rate





$$\beta(t+1) = \beta(t) - r_t \nabla L(\beta(t))$$

$< M$

SGD

$$\sum_{t=1}^{\infty} r_t \to \infty \quad \& \quad \sum_{t=1}^{\infty} r_t^2 < \infty$$

$$r_t = \frac{1}{t}$$