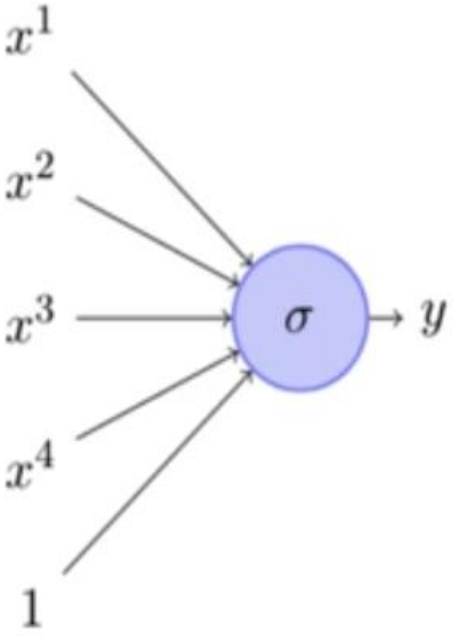
Deep Learning: Adagrad, RMSProp, Adam

राष्ट्रीय प्रौद्योगिकी संस्थान सिक्किम
NATIONAL INSTITUTE OF TECHNOLOGY SIKKIM

Course Instructor:
Dr. Bam Bahadur Sinha
Assistant Professor
Computer Science & Engineering
National Institute of Technology Sikkim

Why do we need a different learning rate for every feature?



$$y = f(x) = \frac{1}{1+e^{-(\mathbf{w}\cdot\mathbf{x}+b)}}$$

$$\mathbf{x} = \{x^1, x^2, x^3, x^4\}$$

$$\mathbf{w} = \{w^1, w^2, w^3, w^4\}$$

$$\nabla w^1 = (f(\mathbf{x}) - y) * f(\mathbf{x}) * (1 - f(\mathbf{x})) * x^1$$
$$\nabla w^2 = (f(\mathbf{x}) - y) * f(\mathbf{x}) * (1 - f(\mathbf{x})) * x^2$$

Can we have a different learning rate for each parameter which takes care of the frequency of features ?

How do we convert the above intuition into an equation?

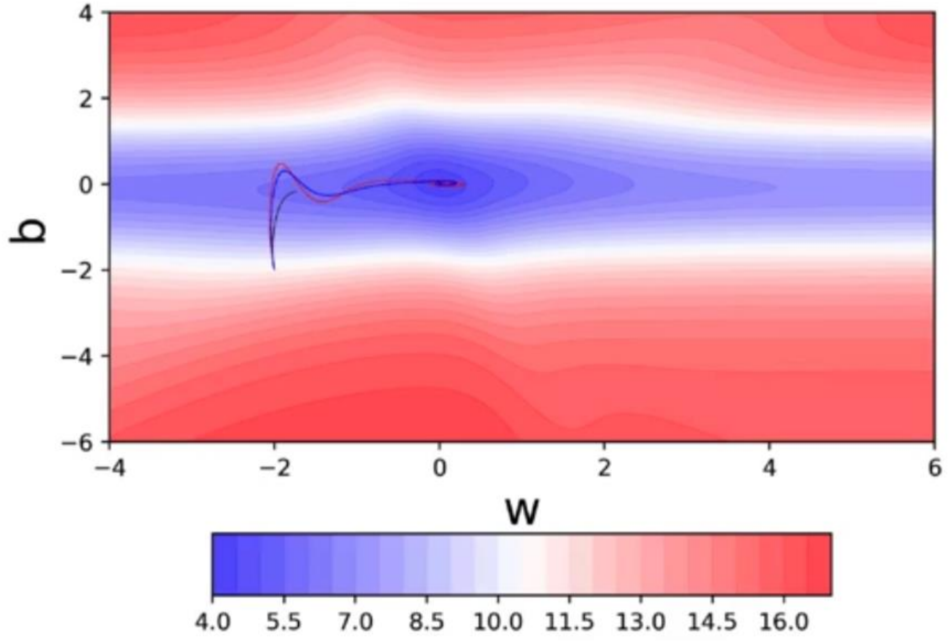
Adaptive Learning Rate

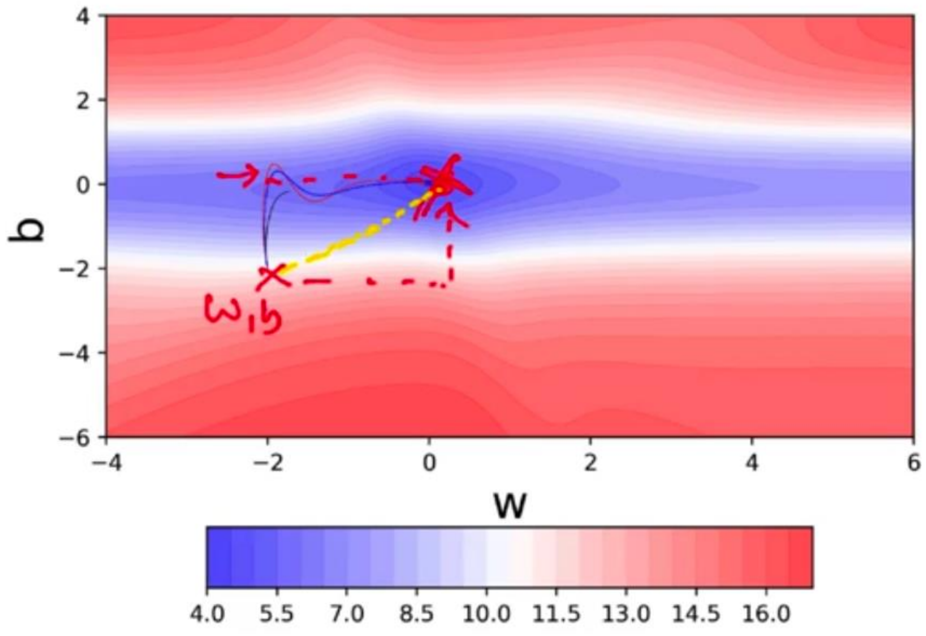
Intuition: Decay the learning rate for parameters in proportion to their update history (fewer updates, lesser decay)

Adagrad

$$v_t = v_{t-1} + (\nabla w_t)^2$$

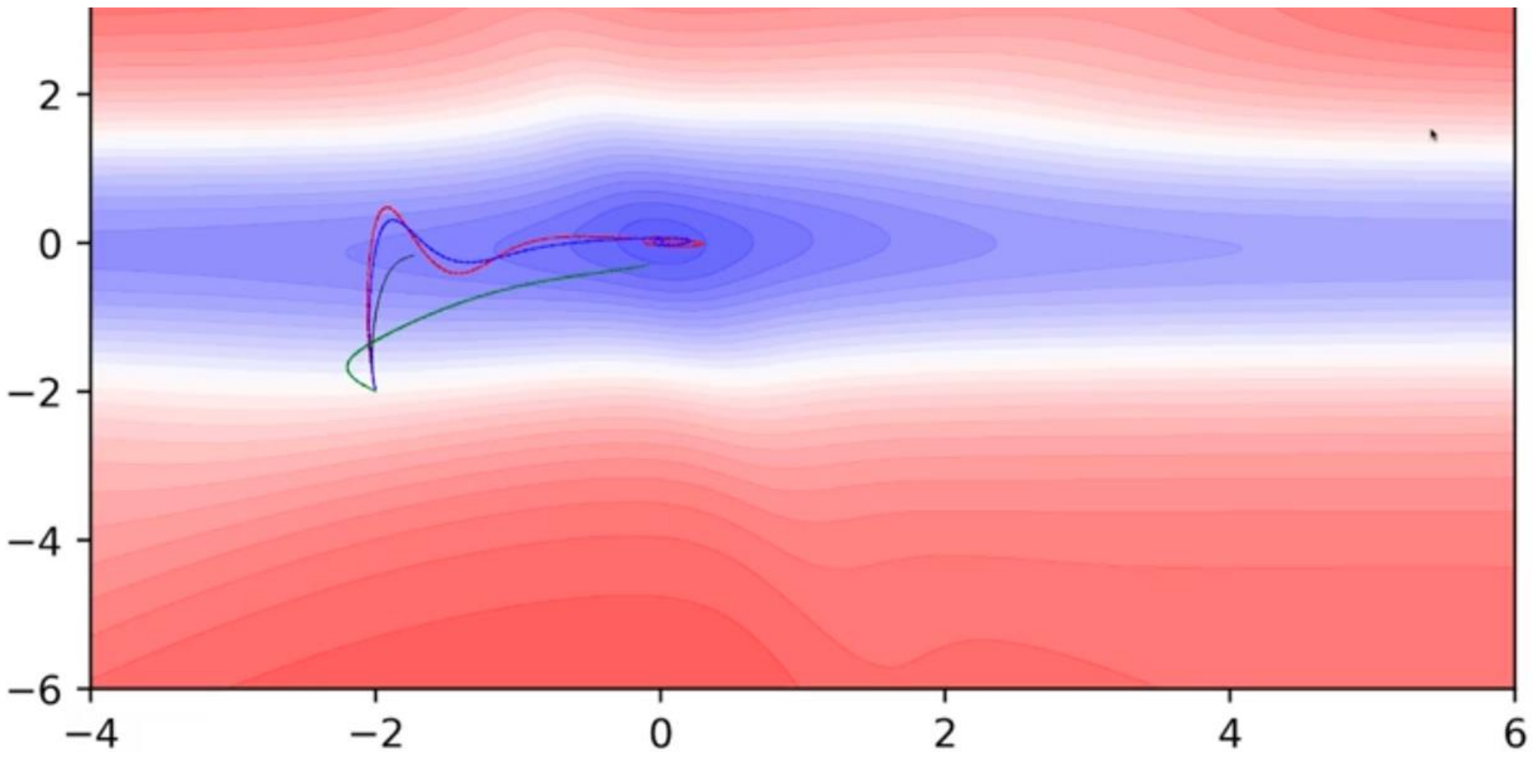
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{(v_t)} + \epsilon} \nabla w_t$$





Black: GD, Red: MGD, Blue: NAGD

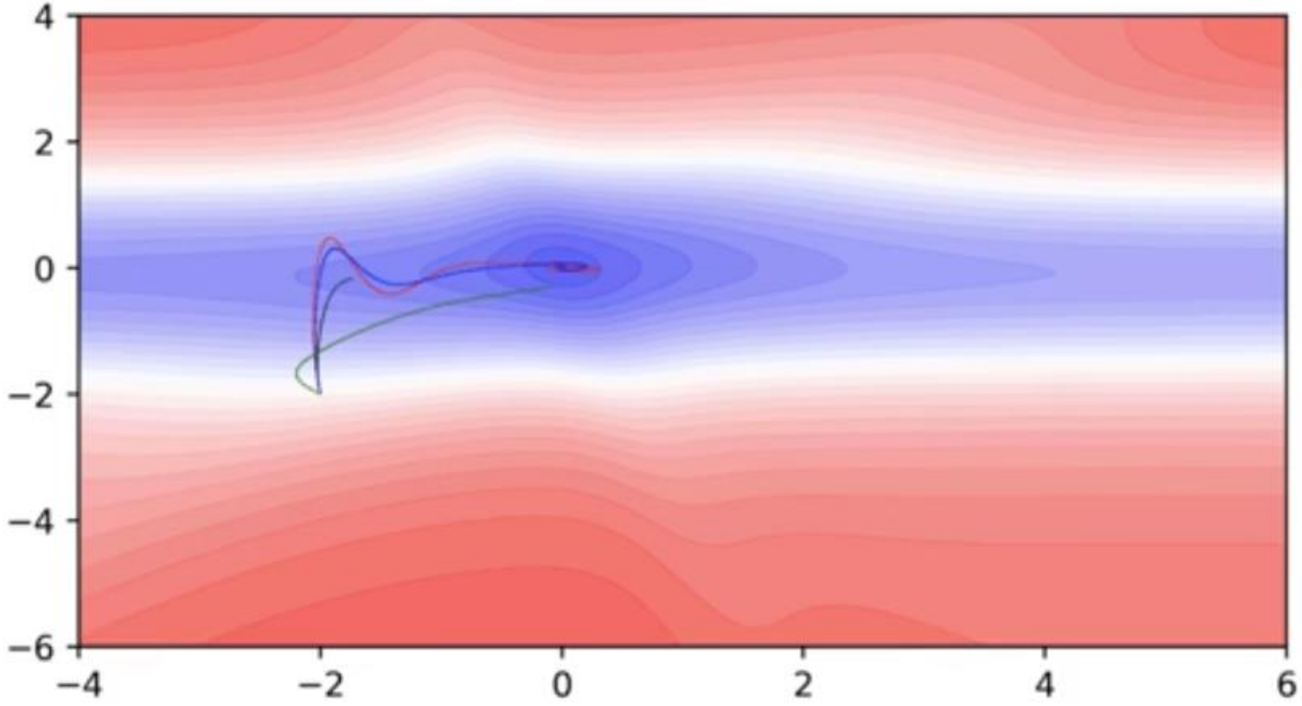
Vanilla Vs MGD Vs NAGD
(while dealing with sparse features)



Green: Adagrad

Adagrad
(20 epochs result)

Adagrad - Advantage & Disadvantage



Advantage

- Parameters corresponding to sparse features get better updates

Disadvantage: The learning rate decays very aggressively as the denominator grows (not good for parameters corresponding to dense features).

RMSProp

Intuition: Why not decay the denominator and prevent its rapid growth ?

Adagrad

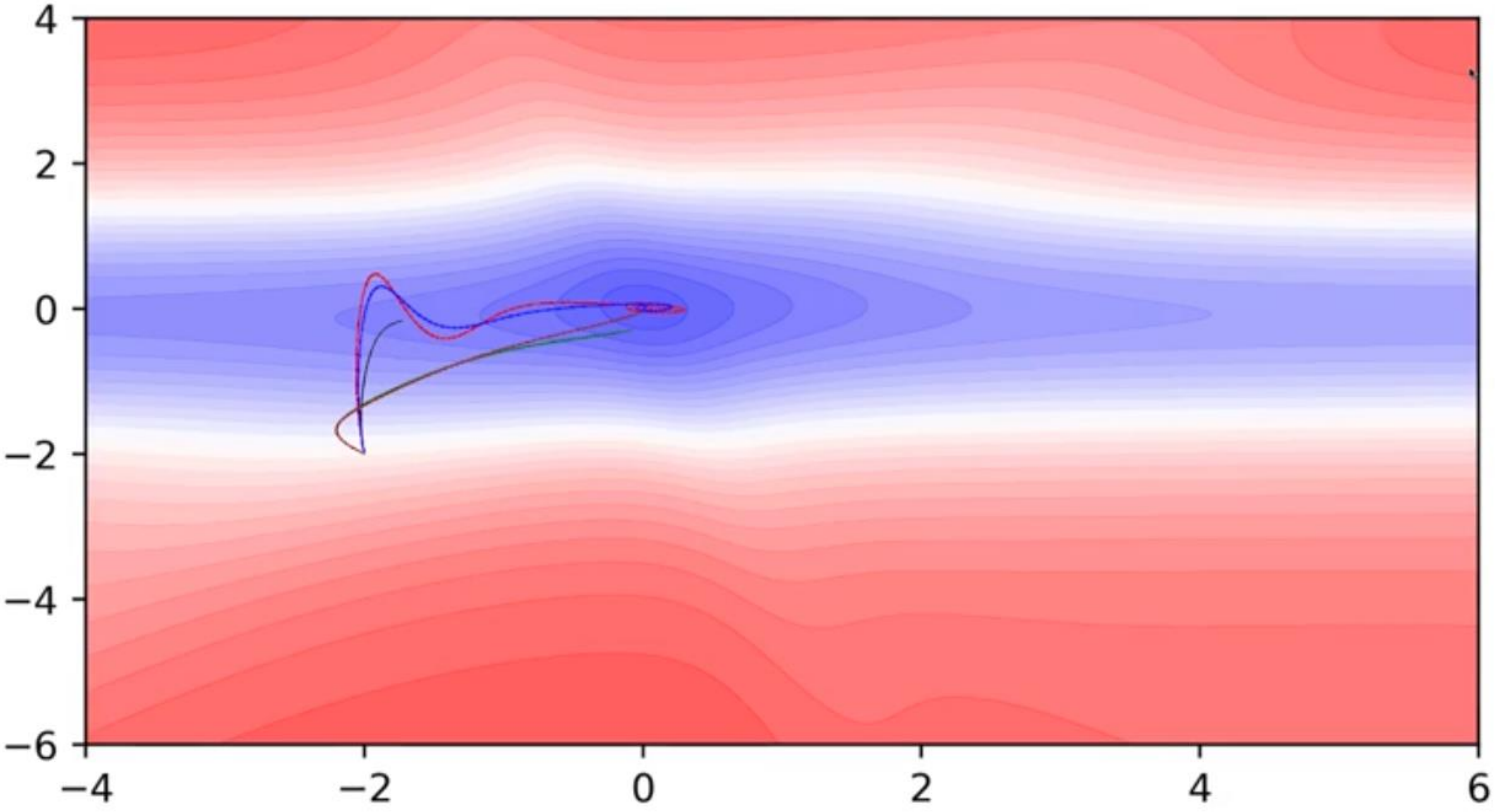
$$v_t = v_{t-1} + (\nabla w_t)^2$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{(v_t)} + \epsilon} \nabla w_t$$

RMSProp

$$v_t = \beta * v_{t-1} + (1 - \beta)(\nabla w_t)^2$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{(v_t)} + \epsilon} \nabla w_t$$



Brown: RMSProp

RMSProp
(converged in 25 epochs – Dense Features)

Observations

- Adagrad got stuck when it was close to convergence (it was no longer able to move in vertical (b) direction because of the decayed learning rate).

- RMSProp overcomes this problem by being less aggressive on the decay.

Adam

Momentum based Gradient Descent Update Rule

$$v_t = \gamma * v_{t-1} + \eta \nabla w_t$$
$$w_{t+1} = w_t - v_t$$

RMSProp

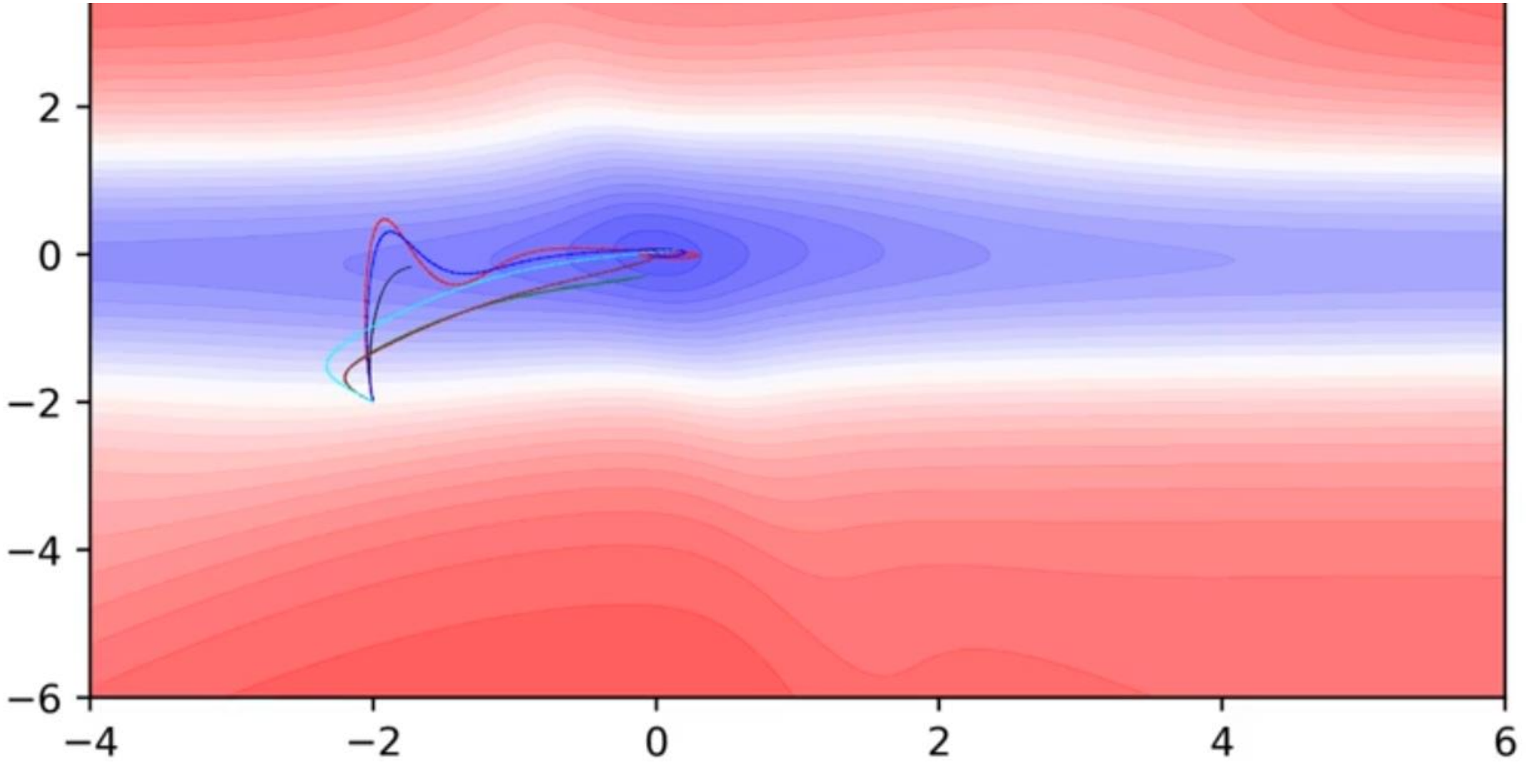
$$v_t = \beta * v_{t-1} + (1-\beta)(\nabla w_t)^2$$
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{(v_t)} + \epsilon} \nabla w_t$$

Adam

$$m_t = \beta_1 * v_{t-1} + (1-\beta_1)(\nabla w_t)$$
$$v_t = \beta_2 * v_{t-1} + (1-\beta_2)(\nabla w_t)^2$$
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{(v_t)} + \epsilon} m_t$$

$$m_t = \frac{m_t}{1-\beta_1^t}$$
$$v_t = \frac{v_t}{1-\beta_2^t}$$

Cyan: Adam



Summary

Initialise w, b

Iterate over data:

$compute\ \hat{y}$

$compute\ \mathscr{L}(w, b)$

$$\left.\begin{aligned} w_{111} &= w_{111} - \eta\Delta w_{111} \\ w_{112} &= w_{112} - \eta\Delta w_{112} \\ &\ldots. \\ w_{313} &= w_{313} - \eta\Delta w_{313} \end{aligned}\right\}$$

till satisfied

Algorithms

- GD
- Momentum based GD
- Nesterov Accelerated GD
- AdaGrad
- RMSProp
- Adam

Strategies

- Batch
- Mini-Batch (32, 64, 128)
- Stochastic