A discussion on

Eve

A Gradient Based Optimization Method with Locally and Globally Adaptive Learning Rates

Presented by David Ryan MacDonald
Toronto AI

Paper: https://arxiv.org/abs/1611.01505

Eve

authors

Jayanth Koushik

PhD student at CMU, H-index: 4

@jayanth_koushik

Graham Neubig

Assistant Professor of CS at CMU, H-index: 26

@gneubig

Hiroaki Hayashi

student at CMU

@hiroberri

Paper: https://arxiv.org/abs/1611.01505

Eve

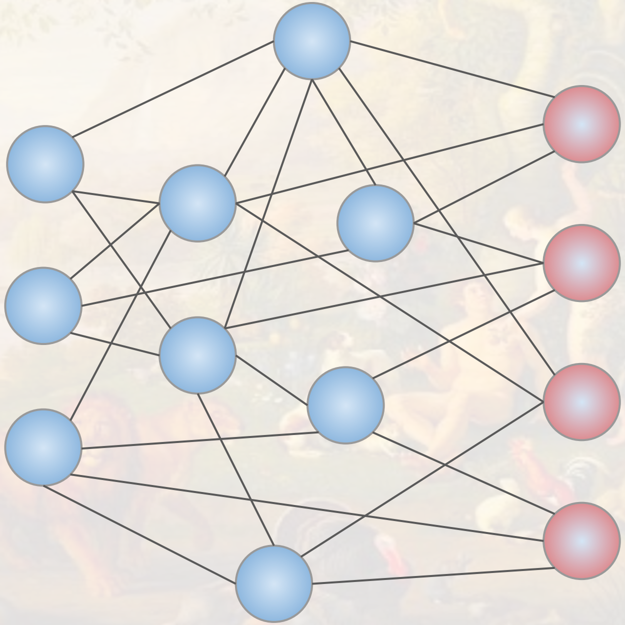
Adam and Eve in the Garden of Eden
Wenzel Peter
(Karlsbad 1745 - Rome 1829)

The basics

$$f_\theta(X)$$

data

$$X \rightarrow$$



prediction

$$\rightarrow \bar{y}_\theta$$

Model

$$f_\theta(X) \rightarrow \bar{y}_\theta$$

Goal: for predictions to match ground truth

$$\bar{y}_{\theta} \longrightarrow y$$

- we want our output predictions to converge to the ground truth
- The gap between our predictions and our target is the called the **loss**

Losses

poisson

mean squared error

huber

kullback-leibler divergence

cross entropy

endless variety of others...

cosine proximity

pairwise distance

hinge loss

Loss

$$\mathcal{L}_{\theta}(\bar{y}_{\theta}, y)$$

Rephrasing our goal:

- We want to update the parameters of our model, to find a parameterization that minimizes the measured loss (our distance from the target)
- By minimizing the loss, our predicted values approach the ground truth labels.

SGD
Stochastic Gradient Descent

(learning rate alpha)
step size

(i.e. jacobian/gradient of the loss)
step direction

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta \mathcal{L}_\theta(\bar{y}, y)$$

new position in parameter space

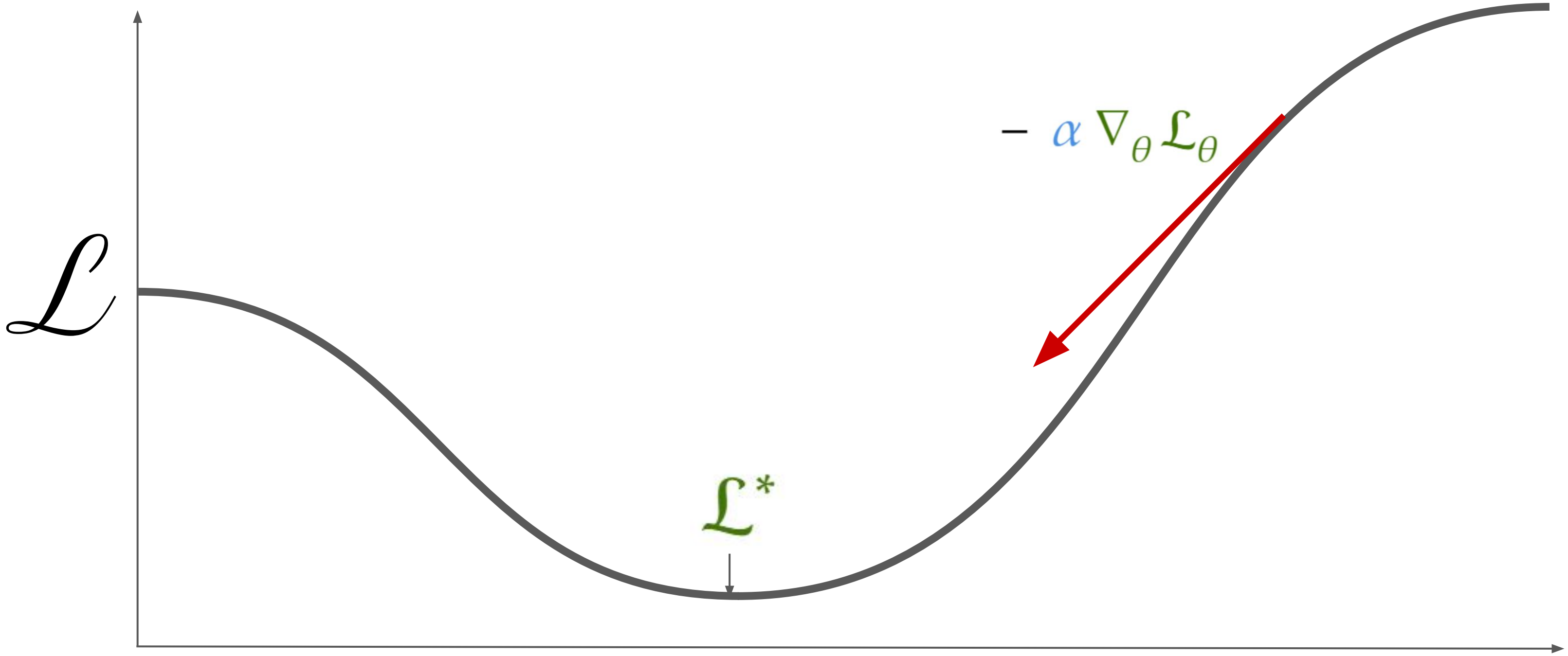
current position in parameter space
(i.e. weights & biases)

head downhill

(head to regions of lower loss)

Parameter update

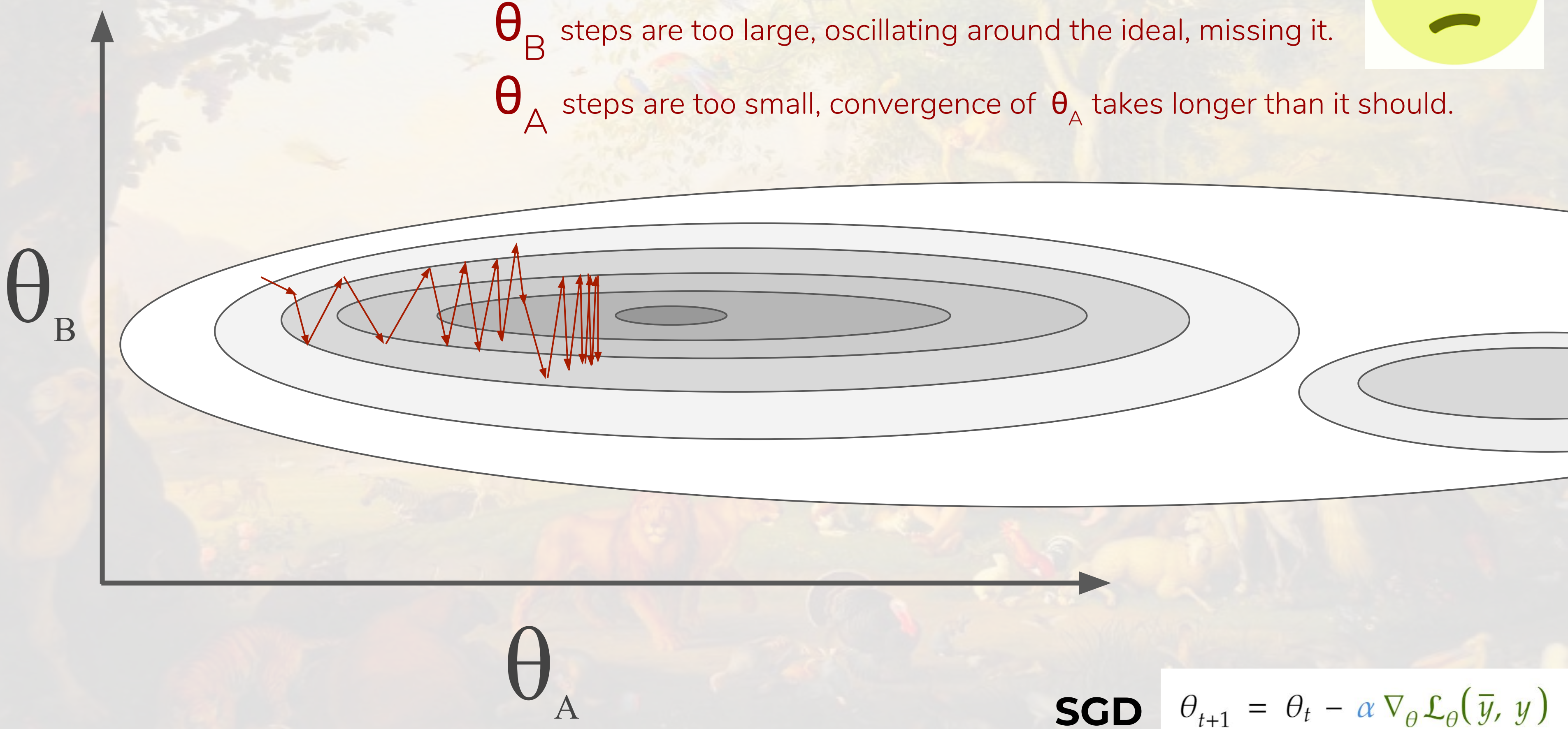
$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta \mathcal{L}_\theta(\bar{y}, y)$$



Parameter space

θ_B steps are too large, oscillating around the ideal, missing it.

θ_A steps are too small, convergence of θ_A takes longer than it should.

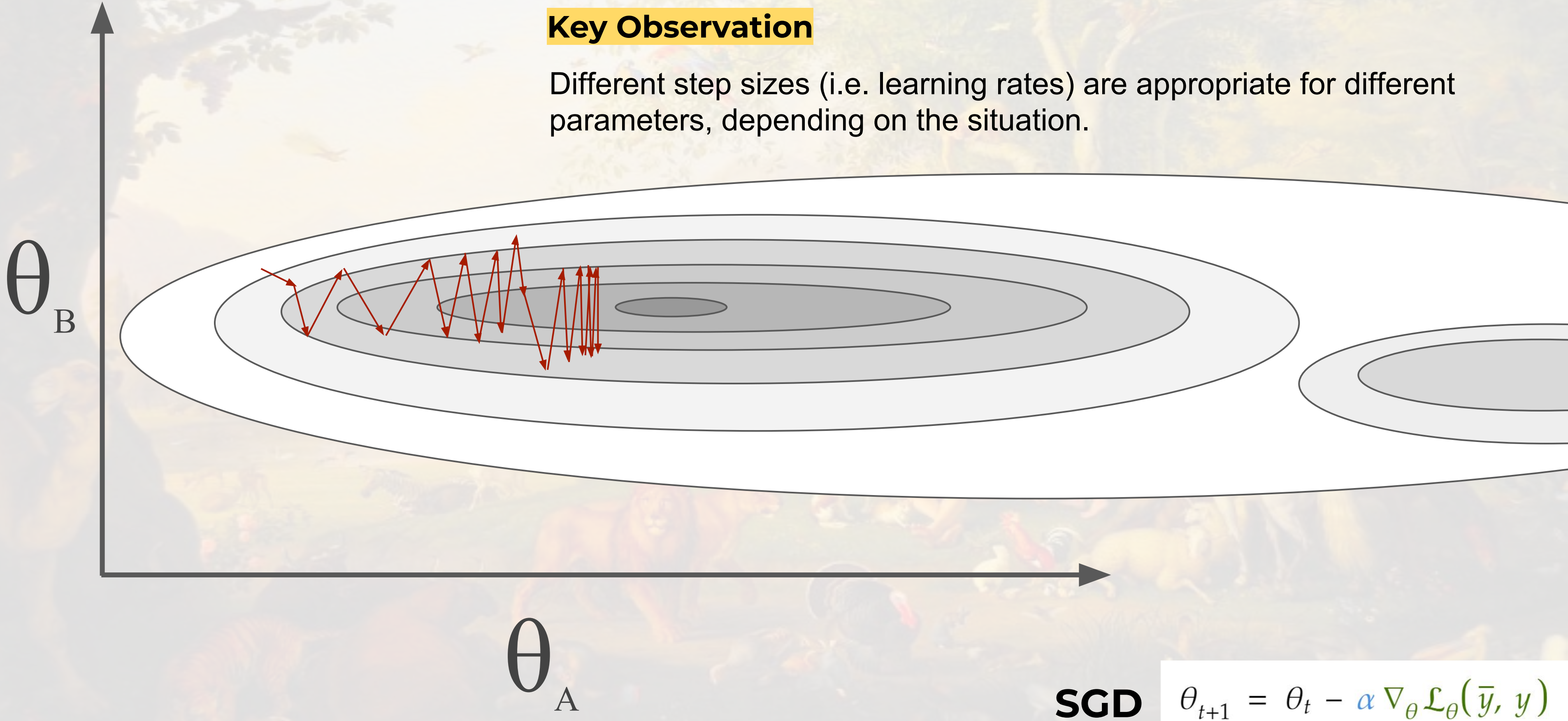


SGD $\theta_{t+1} = \theta_t - \alpha \nabla_\theta \mathcal{L}_\theta(\bar{y}, y)$

How do we improve this?

Key Observation

Different step sizes (i.e. learning rates) are appropriate for different parameters, depending on the situation.



SGD $\theta_{t+1} = \theta_t - \alpha \nabla_\theta \mathcal{L}_\theta(\bar{y}, y)$

Solution: Adaptive learning rates

- Give each parameter their own learning rate

- ADAGRAD family of algorithms:
    - RMSProp
    - Adadelta
    - Adam

- Adam is a very popular alternative to SGD

SGD:

$$\theta_{t+1} = \theta_t - \alpha \nabla_t$$

Adam:

$$\theta_{t+1} = \theta_t - \alpha \frac{\textit{average recent gradient}}{\textit{average recent deviation in the gradient}}$$

Adam

$$\theta_{t+1} = \theta_t - \alpha \frac{\text{average recent gradient}}{\text{average recent deviation in the gradient}}$$

Unpacking 'average recent gradient':

- SGD always takes steps in the 'downhill' direction, based on its immediate local position in the loss landscape.

- But the loss landscape is noisy! The loss landscape in parameter space that the optimizer must navigate through totally depends on the input data (loss -> predicted values -> training data). Since our input data changes with every batch we send into the network, so does our loss landscape!

- Adam attempts to cancel this noise by adding momentum to its trajectory through loss space.   It does this by heading in the direction based on the running average of recent gradients, instead of in the direction of the current (potentially noisy) gradient

Adam

$$\theta_{t+1} = \theta_t - \alpha \frac{\textit{momentum}}{\textit{average recent deviation in the gradient}}$$

Adam's momentum:
smoothes out the impact of noisy gradients

Adam's Momentum

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\text{average recent deviation in the gradient}}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_t$$

$$\textit{suppose } \beta_1 = 99\% :$$

$$m_t = 99\% \cdot m_{t-1} + 1\% \cdot \nabla_t$$

***note**: *in Adam, a small correction term is applied to* m_t *(not shown here) to give a better initial value for the momentum at the start of training.*

Adam

$$\theta_{t+1} = \theta_t - \alpha \frac{\text{momentum}}{\text{average recent deviation in the gradient}}$$

The denominator

- The denominator term allows the Adam optimizer to slow training after entering an area of the parameter space where the loss landscape becomes very steep. Taking big steps is risky in steep terrain.

- Similarly, when a parameter enters a ‘valley’ in the loss landscape, the denominator allows this parameter to speed up training, allowing for bigger steps to be taken.

Adam

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Denominator

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\nabla_t^2$$

- v_t is used to regulate the learning rate based on a moving average of the squared gradient for the parameter
- The moving average is controlled by our second hyperparameter, $\boldsymbol{\beta_2}$
- This moving average is usually an order of magnitude slower to change than the momentum. (i.e. by default $\boldsymbol{\beta_1}$ = *99% while* $\boldsymbol{\beta_2}$= *99.9%*)

- epislon is used to put an upper limit on the 'speed gain' in shallow regions

****note**: in Adam, a small correction term is applied to v_t (not shown here) to give a better initial value for the moving average of the squared gradients at the start of training.*

Adam

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Adam

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Eve

$$\theta_{t+1} = \theta_t - \frac{\alpha}{d_t} \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Eve

$$\theta_{t+1} = \theta_t - \frac{\alpha}{d_t} \frac{m_t}{\sqrt{v_t} + \epsilon}$$

New term

- With Eve, we dynamically adjust the learning rate based on how close we are to the objective.

Eve - Intuitions

$$\theta_{t+1} = \theta_t - \frac{\alpha}{d_t} \frac{m_t}{\sqrt{v_t} + \epsilon}$$

1. A large variation in the loss between timesteps should be given less weight, and so a smaller step should be taken.

2. If we are far from the minimum, take big steps.

$$\theta_{t+1} = \theta_t - \frac{\alpha}{d_t} \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Eve

1. A large variation in the loss between timesteps
   → smaller steps

Thus we want $d_t \propto |\mathcal{L}_t - \mathcal{L}_{t-1}|$

Eve

$$\theta_{t+1} = \theta_t - \frac{\alpha}{d_t} \frac{m_t}{\sqrt{v_t} + \epsilon}$$

2. If the current loss $\mathcal{L}$ is far from the minimum $\mathcal{L}^*$, take big steps.

Thus, $$d_t \propto \frac{1}{\mathfrak{L}_t - \mathfrak{L}^*}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{d_t} \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Eve Putting both intuitons together

A large variation in the loss between timesteps $\rightarrow$ smaller steps

$$\frac{\alpha}{d_t} \propto \alpha \frac{\mathcal{L}_t - \mathcal{L}^*}{|\mathcal{L}_t - \mathcal{L}_{t-1}|}$$

If the current loss $\mathscr{L}$ is far from the minimum $\mathscr{L}^*$, take big steps.

$\theta_{t+1} = \theta$

KABOOM

Unstable

$$\frac{\alpha}{d_t} \propto \alpha \frac{\mathcal{L}_t - \mathcal{L}^*}{|\mathcal{L}_t - \mathcal{L}_{t-1}|}$$

In the numerator, we want to take big steps when we are far from the minimum.

KABOOM

...wever, this i... at risk of a positive fe... L_t increases, then ...ses, which can ... minim... es the step size, wh... n further from the minim...m…

Hello? What was that?!
Is everyone still here?

What can we do now?

Well, it exploded, so let’s clip the new term so that it stays in a bounded range, between 1/c and c

Great idea!

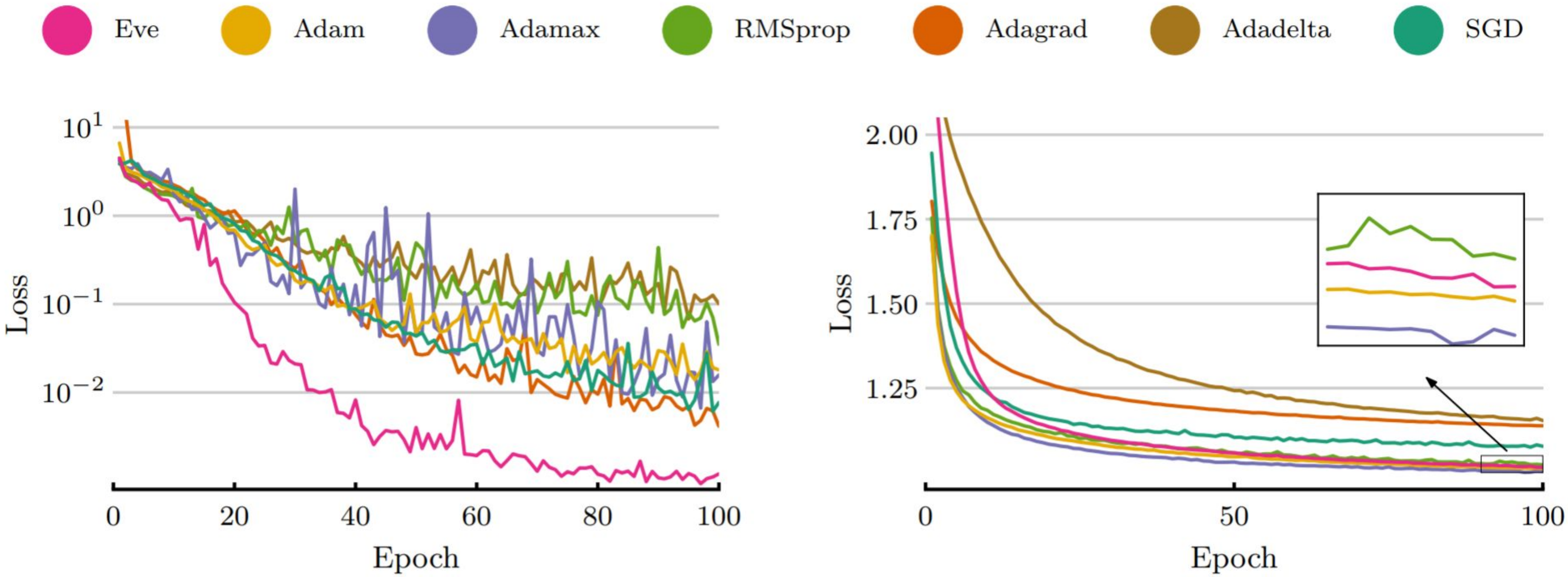
Let’s also add smoothness to d_t with another running average!

Yay!

Sounds great!

Eve Experiments

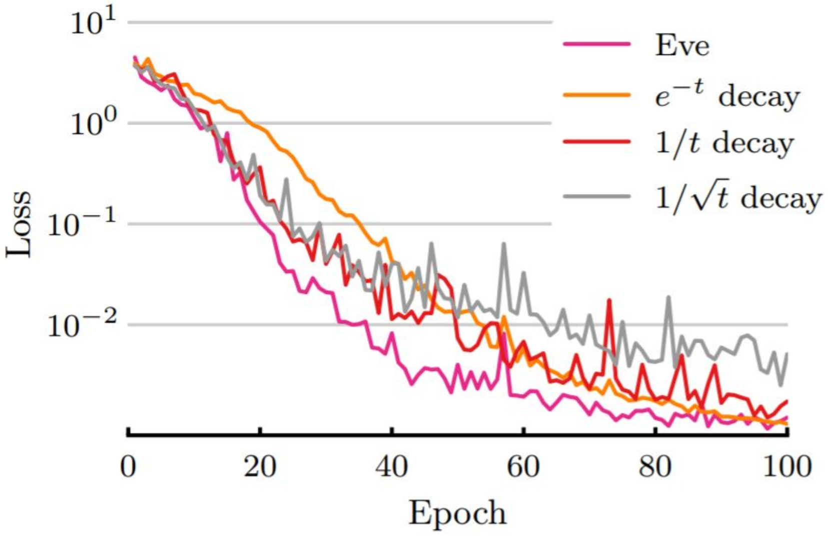
Eve Optimizer comparison



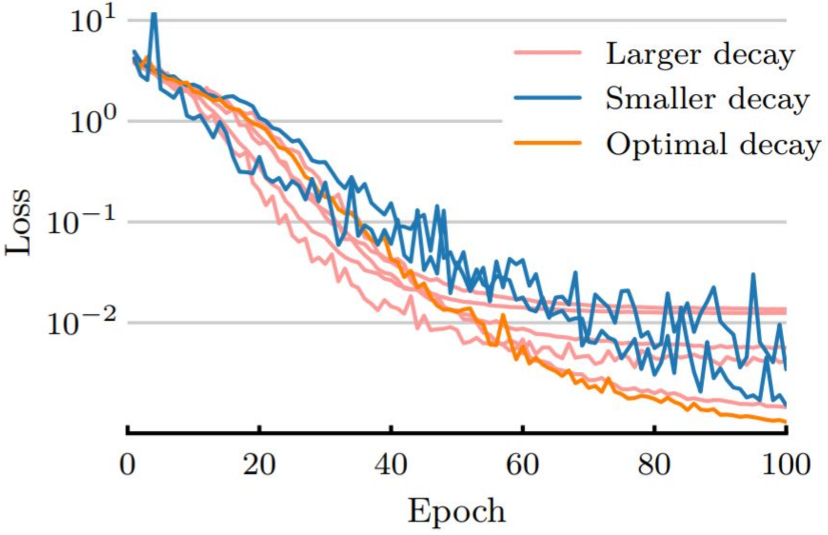
Training loss comparison. In both experiments, Eve achieves similar or lower loss than other optimizers.

Source:
Eve: A Gradient Based Optimization Method with Locally and Globally Adaptive Learning Rates
https://arxiv.org/pdf/1611.01505.pdf

Eve Comparison with decay strategies

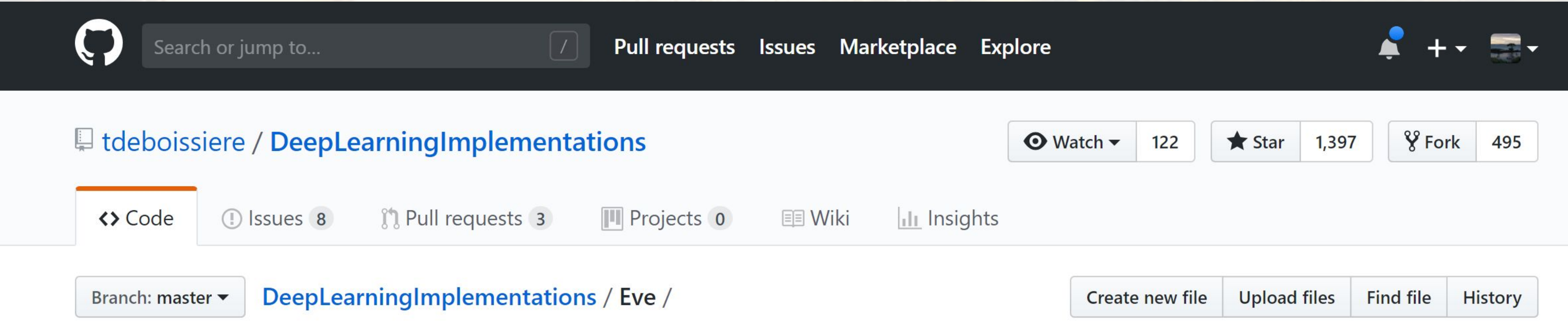


a. Eve compared with different decay strategies.



b. Adam with different exponential decay strengths.

Keras implementation



Eve

Thank you!

Adam and Eve in the Garden of Eden
Wenzel Peter
(Karlsbad 1745 - Rome 1829)