Mila

Université de Montréal

Negative Momentum for Improved Game Dynamics

Gauthier Gidel*, Reyhane Askari Hemmat*, Mohammad Pezeshki, Gabriel Huang, Remi Lepriol, Simon Lacoste-Julien, Ioannis Mitliagkas

**equal contribution*

Simple Min-max smooth game:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}} \max_{\boldsymbol{\phi}\in\mathbb{R}} \boldsymbol{\theta}\cdot\boldsymbol{\phi}$$

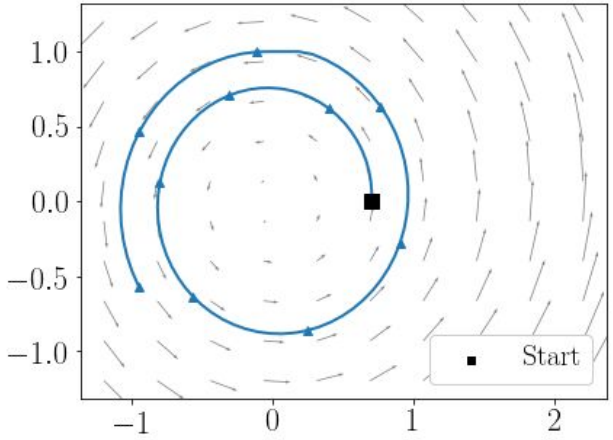
Gradient dynamic:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\boldsymbol{\phi}_t$$
$$\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta\boldsymbol{\theta}_t$$

Simple Min-max smooth game:

$$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \theta \cdot \phi$$

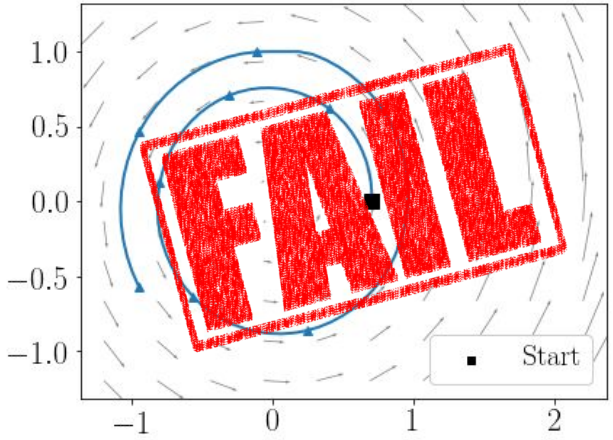
Gradient dynamic:



Simple Min-max smooth game:

$$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \theta \cdot \phi$$

Gradient dynamic:



Way to optimize bilinear games

Method	update rule	Convergence (iterates)
Simultaneous Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta\boldsymbol{\theta}_t$	✗

Way to optimize bilinear games

Method	update rule	Convergence (iterates)
Simultaneous Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta\boldsymbol{\theta}_t$	✗
Alternated Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1\boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta_2\boldsymbol{\theta}_{t+1}$	✗

Way to optimize bilinear games

Method	update rule	Convergence (iterates)
Simultaneous Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta\boldsymbol{\theta}_t$	✗
Alternated Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1\boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta_2\boldsymbol{\theta}_{t+1}$	✗
Alternated Gradient + negative momentum	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1\boldsymbol{\phi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta_2\boldsymbol{\theta}_{t+1} + \beta_2(\boldsymbol{\phi}_t - \boldsymbol{\phi}_{t-1})$	✓

Way to optimize bilinear games

Method	update rule	Convergence (iterates)
Simultaneous Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta M \boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta M^\top \boldsymbol{\theta}_t$	✗
Simultaneous Gradient + negative momentum	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta M \boldsymbol{\phi}_t + \beta(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta M^\top \boldsymbol{\theta}_t + \beta(\boldsymbol{\phi}_t - \boldsymbol{\phi}_{t-1})$	✗ (but less worse)
Alternated Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1 M \boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta_2 M^\top \boldsymbol{\theta}_{t+1}$	✗
Alternated Gradient + negative momentum	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1 M \boldsymbol{\phi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta_2 M^\top \boldsymbol{\theta}_{t+1} + \beta_2(\boldsymbol{\phi}_t - \boldsymbol{\phi}_{t-1})$	✓

Way to optimize bilinear games

Method	update rule	Convergence (iterates)
Simultaneous Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta\boldsymbol{\theta}_t$	✗
Simultaneous Gradient + negative momentum	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta M\boldsymbol{\phi}_t + \beta(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta M^\top\boldsymbol{\theta}_t + \beta(\boldsymbol{\phi}_t - \boldsymbol{\phi}_{t-1})$	✗ (but less bad)
Alternated Gradient	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1\boldsymbol{\phi}_t$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta_2\boldsymbol{\theta}_{t+1}$	✗
Alternated Gradient + negative momentum	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1\boldsymbol{\phi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$ $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta_2\boldsymbol{\theta}_{t+1} + \beta_2(\boldsymbol{\phi}_t - \boldsymbol{\phi}_{t-1})$	✓

(Improvements)

(Improvements)

This talk

General 2 player games:

Two players aim to minimize their respective cost functions:

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta} \in \boldsymbol{\theta}}{\arg\min}\, \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\phi}^*) \quad \text{and} \quad \boldsymbol{\phi}^* \in \underset{\boldsymbol{\phi} \in \boldsymbol{\phi}}{\arg\min}\, \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\theta}^*, \boldsymbol{\phi})$$

General 2 player games:

Two players aim to minimize their respective cost functions:

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}\in\boldsymbol{\theta}}{\arg\min}\, \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\phi}^*) \quad \text{and} \quad \boldsymbol{\phi}^* \in \underset{\boldsymbol{\phi}\in\boldsymbol{\phi}}{\arg\min}\, \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\theta}^*, \boldsymbol{\phi})$$

Examples:

- Simple class of zero-sum games: ($\mathcal{L}^{(\theta)} = -\mathcal{L}^{(\phi)}$)

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \max_{\boldsymbol{\phi}\in\mathbb{R}^p} \alpha\|\boldsymbol{\theta}\|_2^2 + (1-\alpha)\boldsymbol{\theta}^\top \boldsymbol{M}\boldsymbol{\phi} - \alpha\|\boldsymbol{\phi}\|_2^2, \quad \alpha \in [0,1],\ \boldsymbol{M} \in \mathbb{R}^{d\times p}$$

General 2 player games:

Two players aim to minimize their respective cost functions:

$$\boldsymbol{\theta}^* \in \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\phi}^*) \quad \text{and} \quad \boldsymbol{\phi}^* \in \operatorname*{arg\,min}_{\boldsymbol{\phi}\in\boldsymbol{\phi}} \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\theta}^*, \boldsymbol{\phi})$$

Examples:

- Simple class of zero-sum games: ($\mathcal{L}^{(\theta)} = -\mathcal{L}^{(\phi)}$)

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \max_{\boldsymbol{\phi}\in\mathbb{R}^p} \alpha\|\boldsymbol{\theta}\|_2^2 + (1-\alpha)\boldsymbol{\theta}^\top \boldsymbol{M}\boldsymbol{\phi} - \alpha\|\boldsymbol{\phi}\|_2^2, \quad \alpha \in [0,1],\ \boldsymbol{M} \in \mathbb{R}^{d\times p}$$

- Generative Adversarial Networks:

$$\mathcal{L}^{(\theta)}(\theta,\phi) \stackrel{\text{def}}{=} -\mathop{\mathbb{E}}_{\boldsymbol{x}'\sim q_\theta} \log f_\phi(\boldsymbol{x}') \quad \text{and} \quad \mathcal{L}^{(\phi)}(\theta,\phi) \stackrel{\text{def}}{=} -\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p} \log f_\phi(\boldsymbol{x}) - \mathop{\mathbb{E}}_{\boldsymbol{x}'\sim q_\theta} \log(1 - f_\phi(\boldsymbol{x}'))$$

(non-saturating GAN from Goodfellow et al. 2014)

General 2 player games:

Two players aim to minimize their respective cost functions:

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta} \in \boldsymbol{\theta}}{\arg\min}\, \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\phi}^*) \quad \text{and} \quad \boldsymbol{\phi}^* \in \underset{\boldsymbol{\phi} \in \boldsymbol{\phi}}{\arg\min}\, \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\theta}^*, \boldsymbol{\phi})$$

Dynamics of gradient based method depends on the gradient vector fields:

$$\boldsymbol{v}(\boldsymbol{\phi}, \boldsymbol{\theta}) := \begin{bmatrix} \nabla_{\boldsymbol{\phi}} \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\phi}, \boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\phi}, \boldsymbol{\theta}) \end{bmatrix}$$

General 2 player games:

Two players aim to minimize their respective cost functions:

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}\in\boldsymbol{\theta}}{\arg\min}\, \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\phi}^*) \quad \text{and} \quad \boldsymbol{\phi}^* \in \underset{\boldsymbol{\phi}\in\boldsymbol{\phi}}{\arg\min}\, \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\theta}^*, \boldsymbol{\phi})$$

Dynamics of gradient based method depends on the gradient vector fields:

$$\boldsymbol{v}(\boldsymbol{\phi}, \boldsymbol{\theta}) := \begin{bmatrix} \nabla_{\boldsymbol{\phi}} \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\phi}, \boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\phi}, \boldsymbol{\theta}) \end{bmatrix}$$

And its associated Jacobian,

$$\nabla \boldsymbol{v}(\boldsymbol{\phi}, \boldsymbol{\theta}) := \begin{bmatrix} \nabla^2_{\boldsymbol{\phi}} \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\phi}, \boldsymbol{\theta}) & \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\phi})}(\boldsymbol{\phi}, \boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\phi}, \boldsymbol{\theta})^T & \nabla^2_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\phi}, \boldsymbol{\theta}) \end{bmatrix}$$

Fixed point dynamics

Gradient method is defined as the repetition of the operator:

$$F_\eta(\boldsymbol{\phi}, \boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{\phi} & \boldsymbol{\theta} \end{bmatrix}^\top - \eta \, \boldsymbol{v}(\boldsymbol{\phi}, \boldsymbol{\theta})$$

Thus, the sequence computed is

$$(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t) = \underbrace{F_\eta \circ \ldots \circ F_\eta}_{t}(\boldsymbol{\phi}_0, \boldsymbol{\theta}_0) = F_\eta^{(t)}(\boldsymbol{\phi}_0, \boldsymbol{\theta}_0)$$

Fixed point dynamics

Gradient method is defined as the repetition of the operator:

$$F_\eta(\boldsymbol{\phi}, \boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{\phi} & \boldsymbol{\theta} \end{bmatrix}^\top - \eta \, \boldsymbol{v}(\boldsymbol{\phi}, \boldsymbol{\theta})$$

Thus, the sequence computed is

$$(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t) = \underbrace{F_\eta \circ \ldots \circ F_\eta}_{t}(\boldsymbol{\phi}_0, \boldsymbol{\theta}_0) = F_\eta^{(t)}(\boldsymbol{\phi}_0, \boldsymbol{\theta}_0)$$

We aim to converge to a *Nash Equilibrium*:

Tuning the step size

Jacobian of our fixed point operator:

$$\nabla F_\eta(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*) = I_n - \eta \nabla \boldsymbol{v}(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$$

- To have fixed point we need $\nabla \boldsymbol{v}(\boldsymbol{\phi}^*, \boldsymbol{\theta}^*)$ to be definite positive.
- Thus, small enough step-size $\Longrightarrow$ Eigenvalues in the unit disk.
- Want to find optimal step-size.

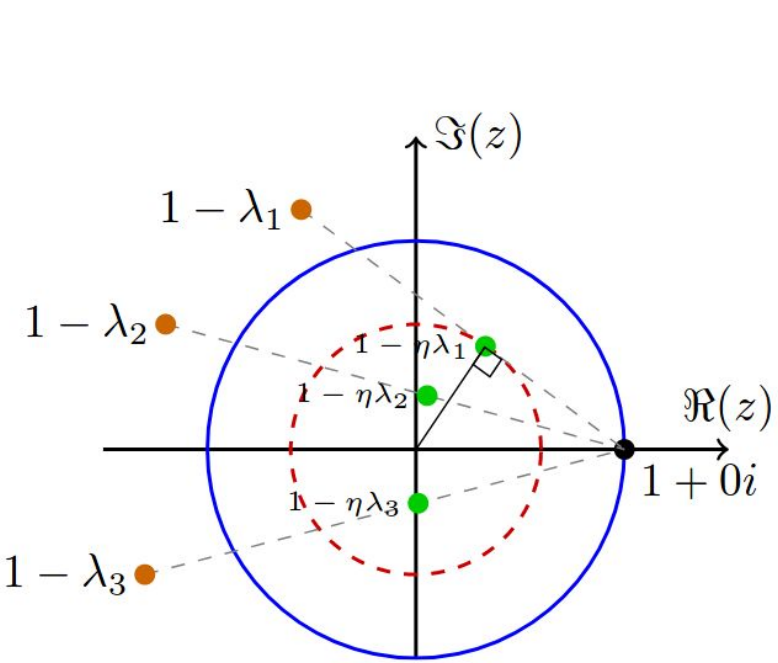
Fixed point dynamics

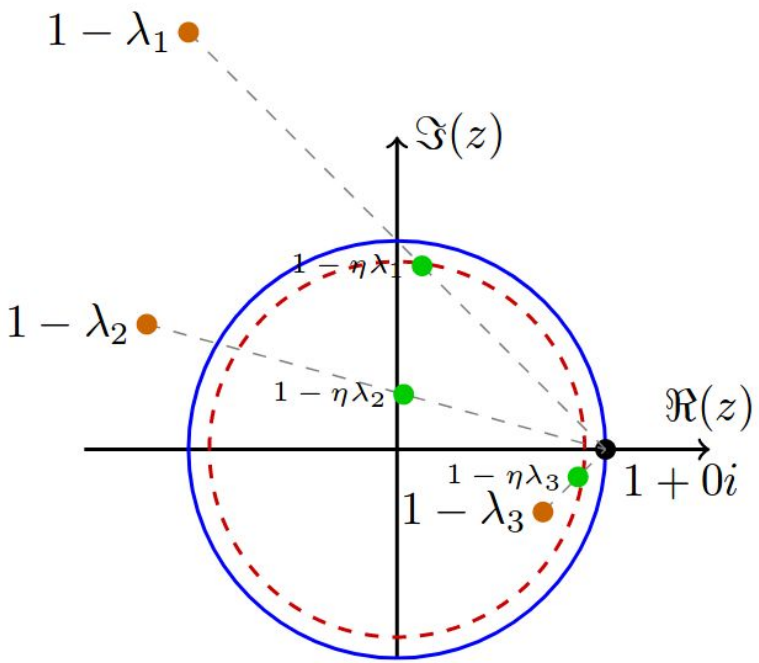
Jacobian of our fixed point operator: $\nabla F_\eta(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*) = I_n - \eta \nabla \boldsymbol{v}(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$

Theorem 1 (Proposition 4.4.1 Bertsekas (1999)). *Let μ_{max} be the largest eigenvalue (in magnitude) of $\nabla F_\eta(\boldsymbol{\phi}^*, \boldsymbol{\theta}^*)$. If $\rho_{\max} := |\mu_{max}| < 1$ then, for $(\boldsymbol{\phi_0}, \boldsymbol{\theta_0})$ in a neighborhood of $(\boldsymbol{\phi}^*, \boldsymbol{\theta}^*)$, the sequence $(\boldsymbol{\phi}_t, \boldsymbol{\theta}_t)$ converges to the local Nash equilibrium at a rate of $\mathcal{O}\left((\rho_{\max} + \epsilon)^t\right)$, $\forall \epsilon > 0$.*

- Local convergence.
- Stationary point may not be a Nash equilibrium. (See Adolphs et al. 2018)
- But any Nash equilibrium is an stationary point.
- In this talk: local results on stationary points.

Tuning the step size





Negative Momentum

Recall Polyak's momentum :

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{v}(\boldsymbol{x}_t) + \beta(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}), \quad \boldsymbol{x}_t = (\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$$

Fixed point operator requires a **state augmentation** : (because need previous iterates)

$$F_{\eta,\beta}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) := \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n \\ \boldsymbol{I}_n & \boldsymbol{0}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix} - \eta \begin{bmatrix} \boldsymbol{v}(\boldsymbol{x}_t) \\ \boldsymbol{0}_n \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix}$$

Negative Momentum

Theorem 3. *The eigenvalues of* $\nabla F_{\eta,\beta}(\boldsymbol{\phi}^*, \boldsymbol{\theta}^*)$ *are*

$$\mu_{\pm}(\beta, \eta, \lambda) := (1 - \eta\lambda + \beta)\frac{1 \pm \Delta^{\frac{1}{2}}}{2}, \tag{9}$$

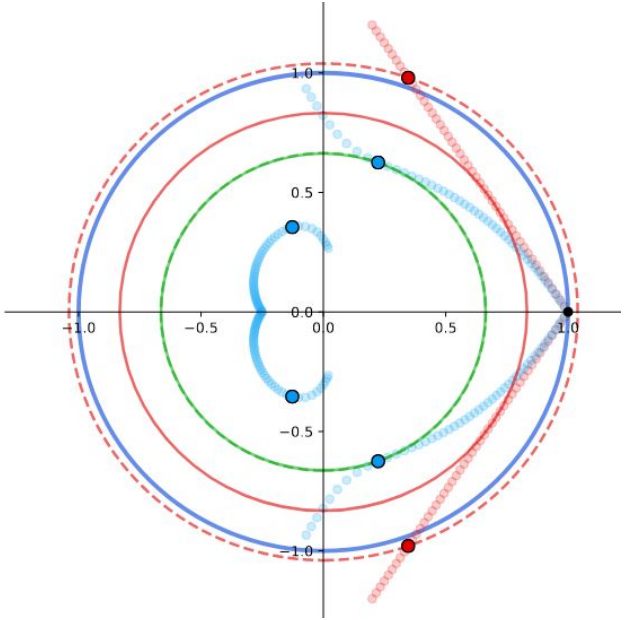
where $\Delta := 1 - \frac{4\beta}{(1-\eta\lambda+\beta)^2}$, $\lambda \in Sp(\nabla \boldsymbol{v}(\boldsymbol{\phi}^*, \boldsymbol{\theta}^*))$ *and* $\Delta^{\frac{1}{2}}$ *is the* complex *square root of* Δ *with positive real part*[3]*. Moreover we have the following Taylor approximation,*

$$\mu_+(\beta, \eta, \lambda) = 1 - \eta\lambda - \beta\frac{\eta\lambda}{1-\eta\lambda} + O(\beta^2) \quad \textit{and} \quad \mu_-(\beta, \eta, \lambda) = \frac{\beta}{1-\eta\lambda} + O(\beta^2) \tag{10}$$

[3] If Δ is a negative real number we set $\Delta^{\frac{1}{2}} := i\sqrt{-\Delta}$

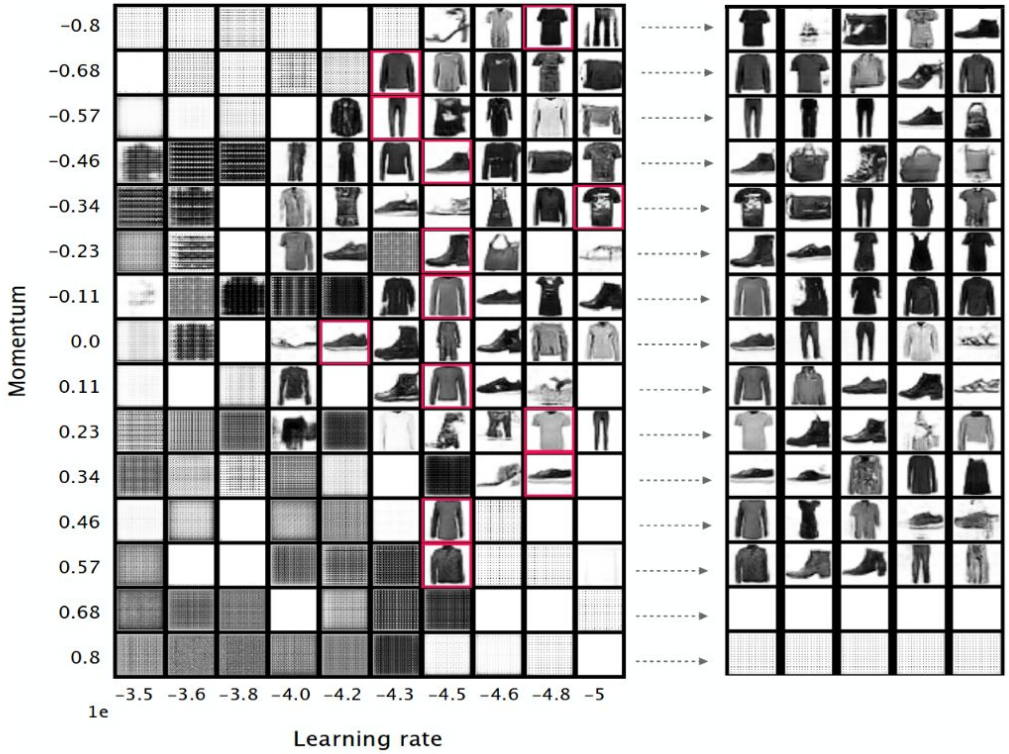
Negative Momentum

- **Fixed** momentum.
  (- 0.25)
- Step-size is **not** fixed.
- Helps when the eigenvalue has large imaginary part.



What happens in practice ?

Fashion MNIST:



What happen in practice ?

CIFAR-10:

Negative Momentum

To sum up:

- Negative momentum seems to improve the behaviour of the “bad” eigenvalues.
- If small enough seems to always help.
- It also allows larger step-size.

Thank you !

If you are interested in that topic:

- **NIPS Workshop : Smooth Games Optimization and Machine Learning**

Co-organized with:

Simon Lacoste-Julien · Ioannis Mitliagkas · Vasilis Syrgkanis · Eva Tardos

· Leon Bottou · Sebastian Nowozin

Soon : Call for contributions !!!