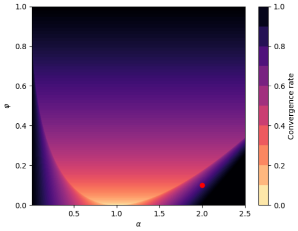
Optimization in Machine Learning

First order methods
Momentum on quadratic forms



Learning goals

- Momentum update in Eigenspace
- Effect of φ

MOMENTUM UPDATE

$$
\begin{aligned}
\boldsymbol{\nu}^{[t+1]} &= \varphi \boldsymbol{\nu}^{[t]} + \alpha \nabla f(\boldsymbol{x}^{[t]}) \\
\boldsymbol{x}^{[t+1]} &= \boldsymbol{x}^{[t]} - \boldsymbol{\nu}^{[t+1]}
\end{aligned}
$$

which simplifies to

$$\boldsymbol{\nu}^{[t+1]} = \varphi \boldsymbol{\nu}^{[t]} + \alpha(\mathbf{A}\boldsymbol{x}^{[t]} - \mathbf{b})$$

$$\boldsymbol{x}^{[t+1]} = \boldsymbol{x}^{[t]} - \boldsymbol{\nu}^{[t+1]}$$

for the quadratic form.

DYNAMICS OF MOMENTUM

Change basis as before with $\boldsymbol{w}^{[t]} = \mathbf{V}^\top(\boldsymbol{x}^{[t]} - \boldsymbol{x}^*)$ and $\boldsymbol{u}^{[t]} = \mathbf{V}\boldsymbol{\nu}^{[t]}$, again each component acts independently, but $w_i^{[t]}$ and $u_i^{[t]}$ are coupled:

$$u_i^{[t+1]} = \varphi u_i^{[t]} + \alpha\lambda_i w_i^{[t]}$$
$$w_i^{[t+1]} = w_i^{[t]} - u_i^{[t+1]}$$

We rewrite this:

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} u_i^{[t+1]} \\ w_i^{[t+1]} \end{pmatrix} = \begin{pmatrix} \varphi & \alpha\lambda_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_i^{[t]} \\ w_i^{[t]} \end{pmatrix}$$

inverting the matrix on the LHS, and unravel the recursion:

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \quad ; \quad \begin{pmatrix} u_i^{[t+1]} \\ w_i^{[t+1]} \end{pmatrix} = \begin{pmatrix} \varphi & \alpha\lambda_i \\ -\varphi & 1 - \alpha\lambda_i \end{pmatrix} \begin{pmatrix} u_i^{[t]} \\ w_i^{[t]} \end{pmatrix} = R^{t+1} \begin{pmatrix} u_i^0 \\ w_i^0 \end{pmatrix}$$

Taking a 2×2 matrix to the t^{th} power can be expressed via its eigenvalues, σ_1 and σ_2, where $R_j = \frac{R - \sigma_j I}{\sigma_1 - \sigma_2}$:

$$R^t = \begin{cases} \sigma_1^t R_1 - \sigma_2^t R_2, & \text{if } \sigma_1 \neq \sigma_2 \\ \sigma_1^t (tR/\sigma_1 - (t-1)I), & \text{if } \sigma_1 = \sigma_2 \end{cases}$$

- Careful, R is not symmetric, so the EVs can be complex
- In contrast to GD, where we got one geometric series, we have two coupled series with real or complex values

EIGENVALUES OF RECURSION MATRIX

The eigenvalues of an arbitrary 2×2 matrix are:

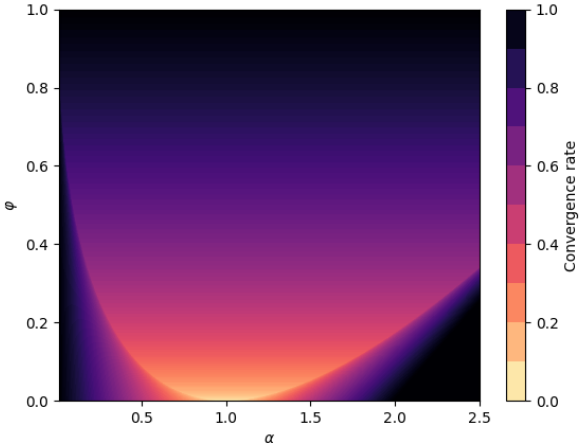
$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{and} \quad \lambda_{1,2} = \frac{\mathrm{tr}(A) \pm \sqrt{\mathrm{tr}(A)^2 - 4\det(A)}}{2}$$

For us this is:

$$\sigma_{1,2} = \frac{1 + \phi - \alpha\lambda_i \pm \sqrt{(1 + \phi - \alpha\lambda_i)^2 - 4\phi}}{2}$$

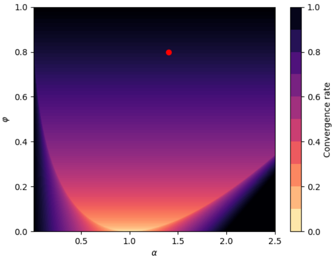
- We need both $|\sigma_1|, |\sigma_2| \leq 1$ for convergence
- For the complex case this reduces to $2\sqrt{\phi}$, which is surprisingly independent of α and λ_i
- For the real case we cannot simplify

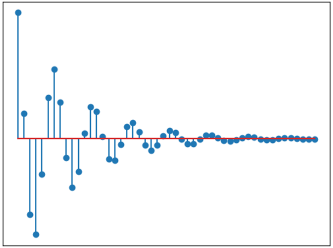
MOMENTUM CONVERGENCE REGIONS



Convergence rate is the slowest of $\max\{|\sigma_1|, |\sigma_2|\}$. Each region shows different convergence behavior.

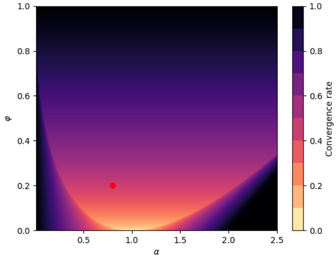
MOMENTUM WITH COMPLEX EIGENVALUES

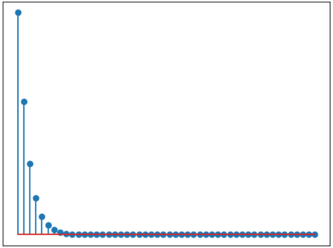




The eigenvalues of R are complex and we see low frequency ripples.

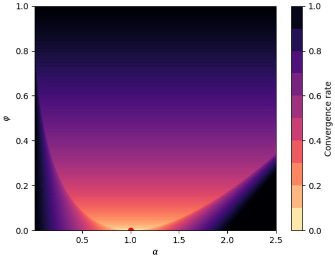
MOMENTUM WITH POSITIVE EIGENVALUES

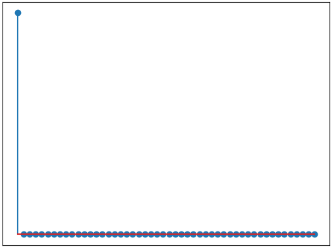




Here, both eigenvalues of R are positive with their norm < 1.
This behavior resembles gradient descent.

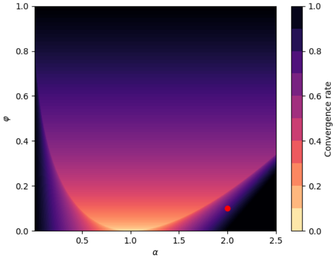
ONE-STEP CONVERGENCE

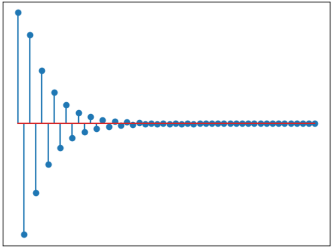




The step size is $\alpha = 1/\lambda_i$ and $\varphi = 0$ - we converge in one step.

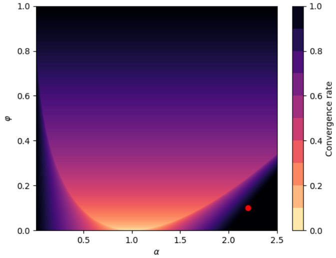
OSCILLATING ITERATES

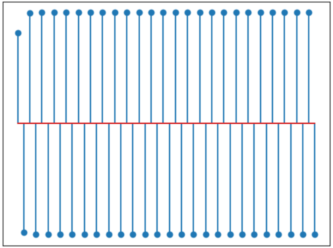




When $\alpha > 1/\lambda_i$, the iterates flip sign every iteration.

DIVERGING ITERATES





If $\max\{|\sigma_1|, |\sigma_2|\} > 1$, the iterates diverge.

CONVERGENCE CONDITIONS

- If we combine all conditions for convergence, we can see:

$$0 < \alpha\lambda_i < 2 + 2\phi \qquad \text{for} \qquad 0 \leq \phi < 1$$

- Comparing this with the results from before ($\phi = 0$), we see that we gain a stepsize factor of 2 before we diverge!
- Can obtain global convergence rate by optimizing over α and ϕ
- More involved, see blogpost for details
- We get $\alpha = \left(\frac{2}{\sqrt{\lambda_1}+\sqrt{\lambda_n}}\right)^2$ and $\phi = \left(\frac{\sqrt{\lambda_n}-\sqrt{\lambda_1}}{\sqrt{\lambda_n}+\sqrt{\lambda_1}}\right)^2$
- Results in convergence rate of $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$

PRACTICAL PARAMETER CHOICES

- Compared to GD with $\frac{\kappa-1}{\kappa+1}$ this is much better as the condition is rooted
- Of course, this would in principle require knowledge of the EVs λ_i
- But we can derive simple rule-of-thumb: for poorly conditioned problems, the stepsize is approximately twice that of GD and ϕ close to 1
- So we want to set ϕ to a high value and then still pick the highest α which still converges