Implementing **M**oment**U**m **O**rthogonalized by **N**ewton-Schultz (MUON)

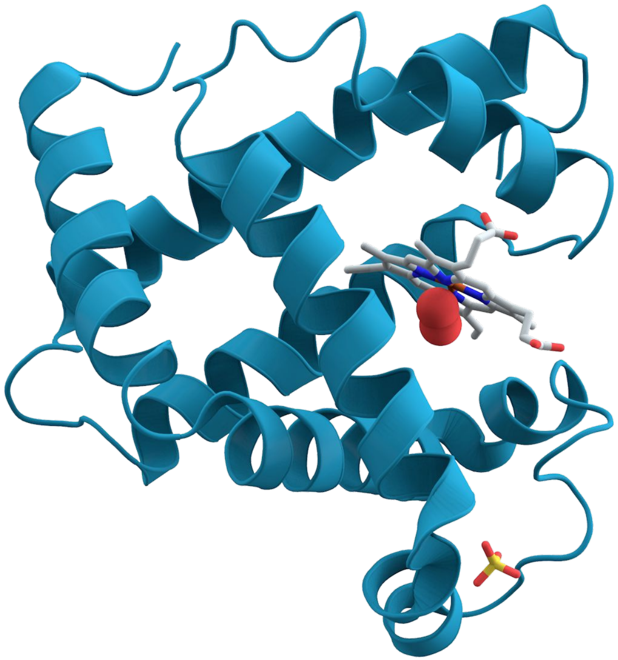
Erik Lidman Hillbom, Elias Lindstenz, Alve Carr, Tatsuya Hongka
eriklh@kth.se, elialin@kth.se, alvec@kth.se, hongka@kth.se

SF1672 Linear Programming Group 13

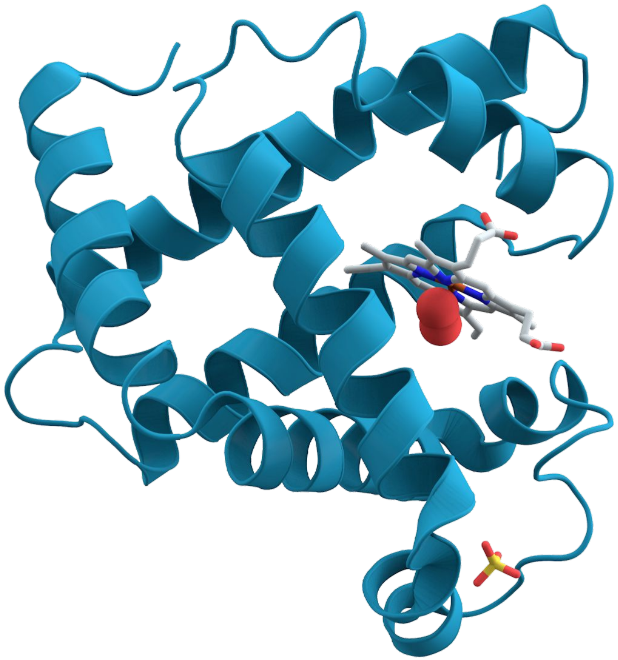
28 November 2025

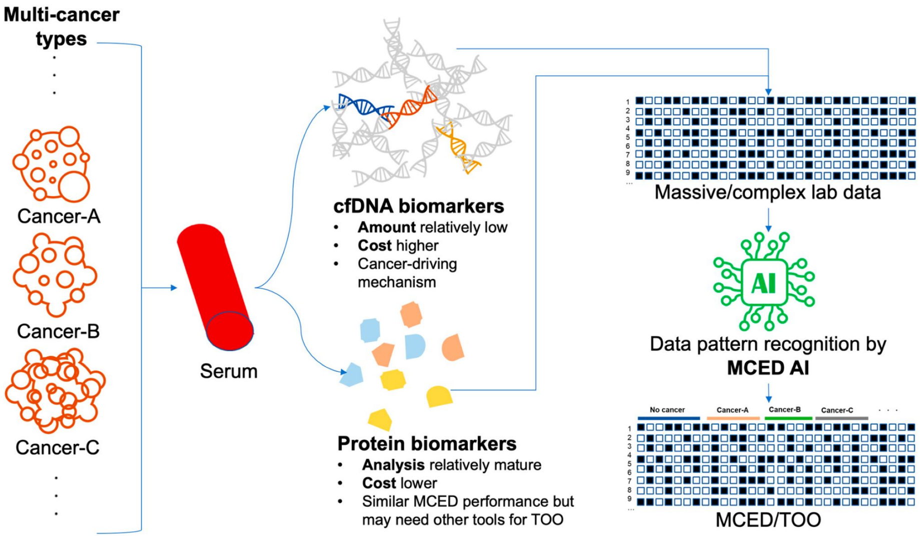
loss - gör rätt

byt pl

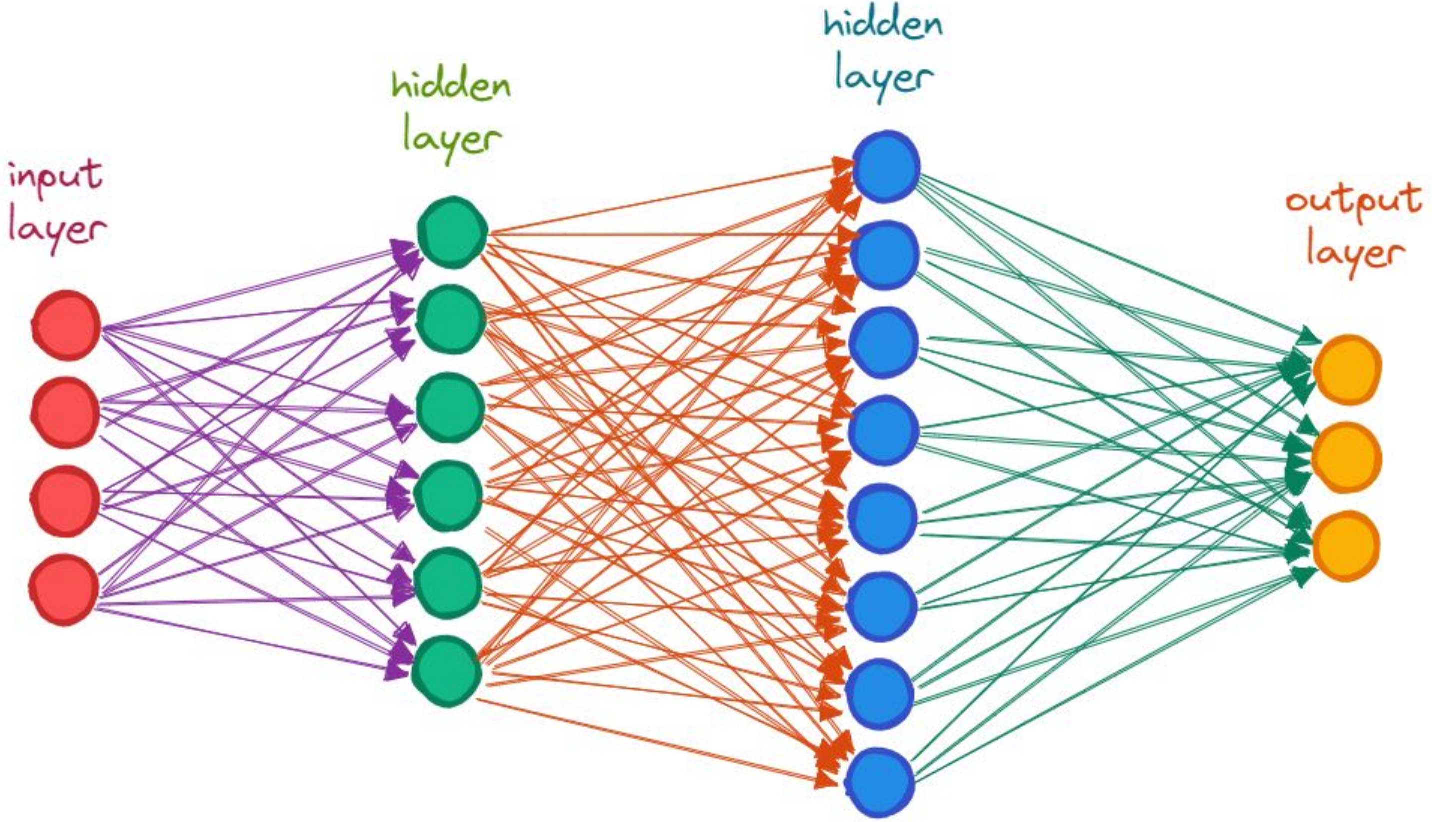


An introduction…

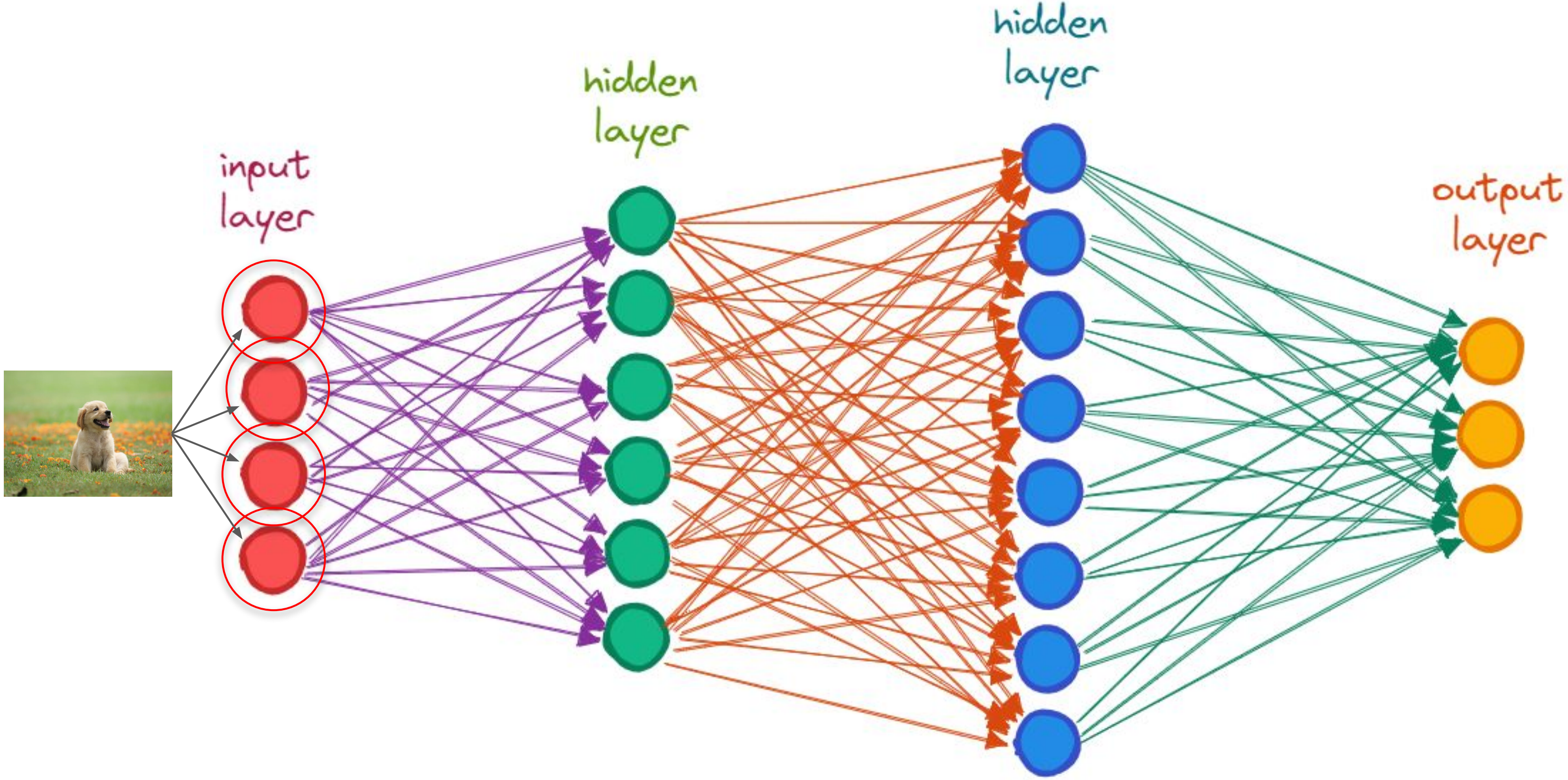




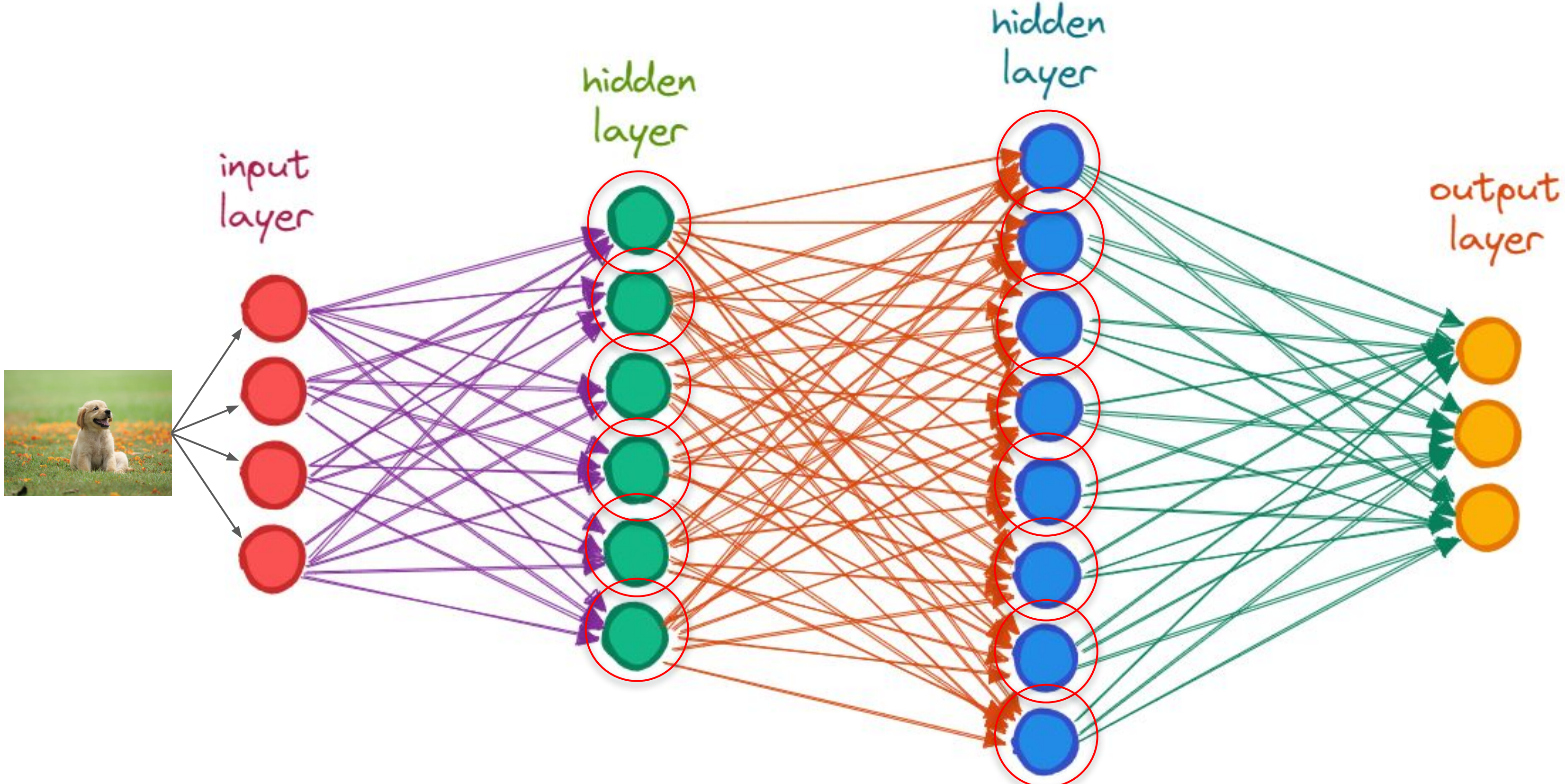
Neural Network for dummies



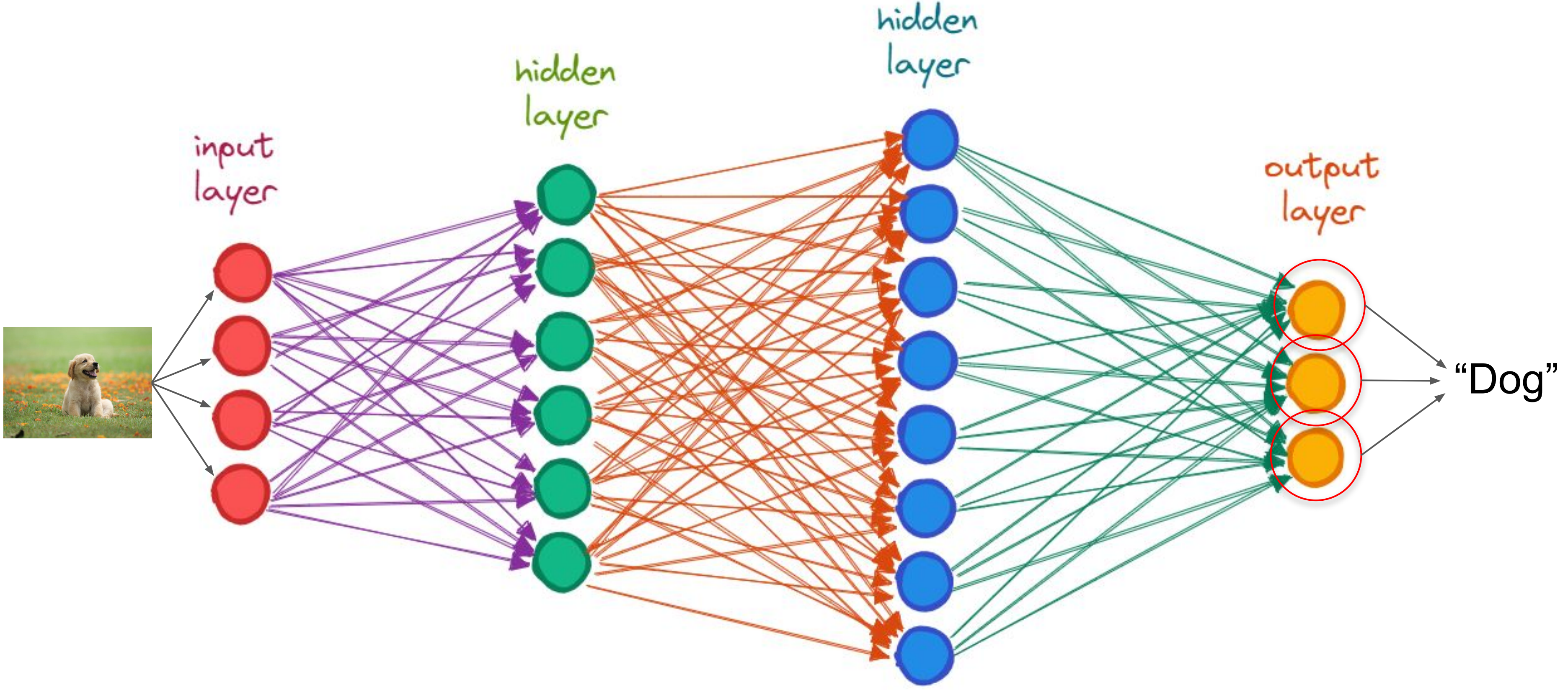
Neural Network



Neural Network

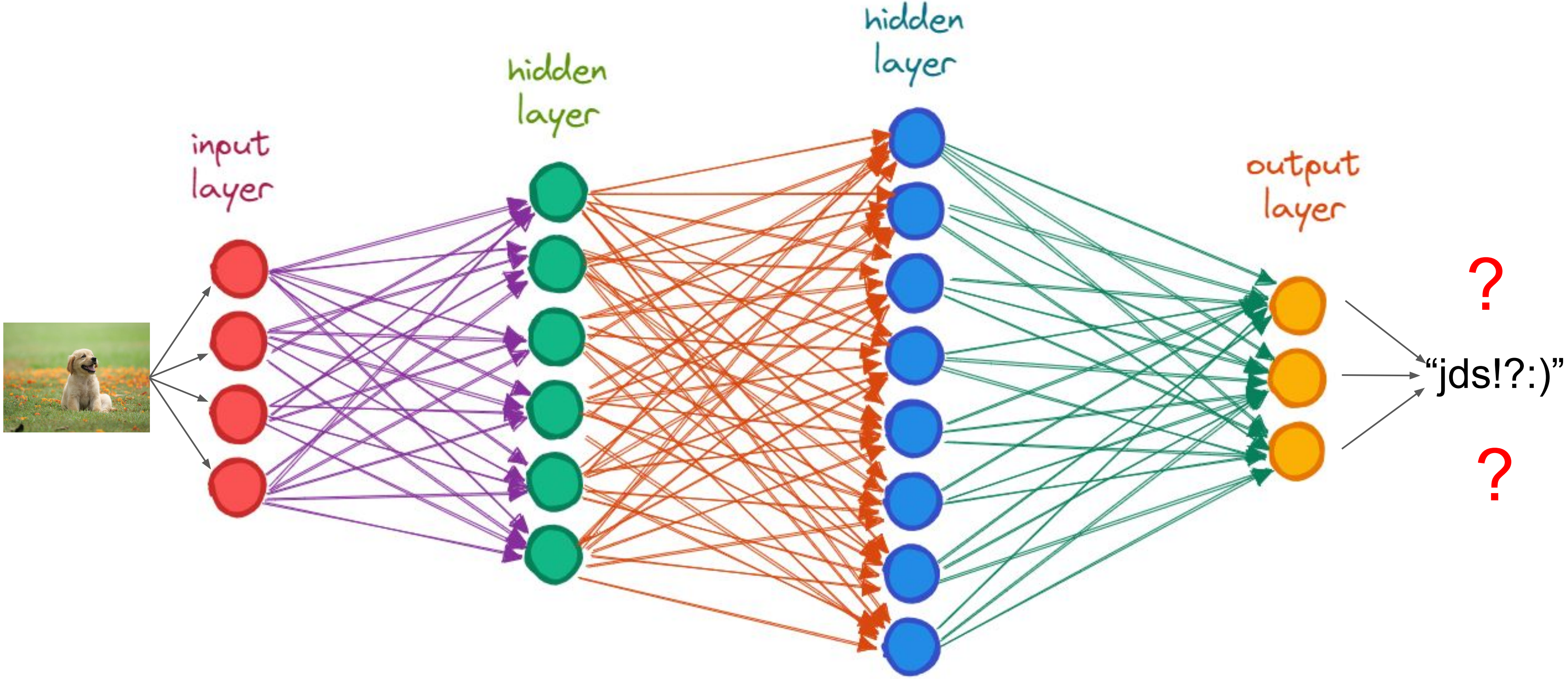


Neural Network



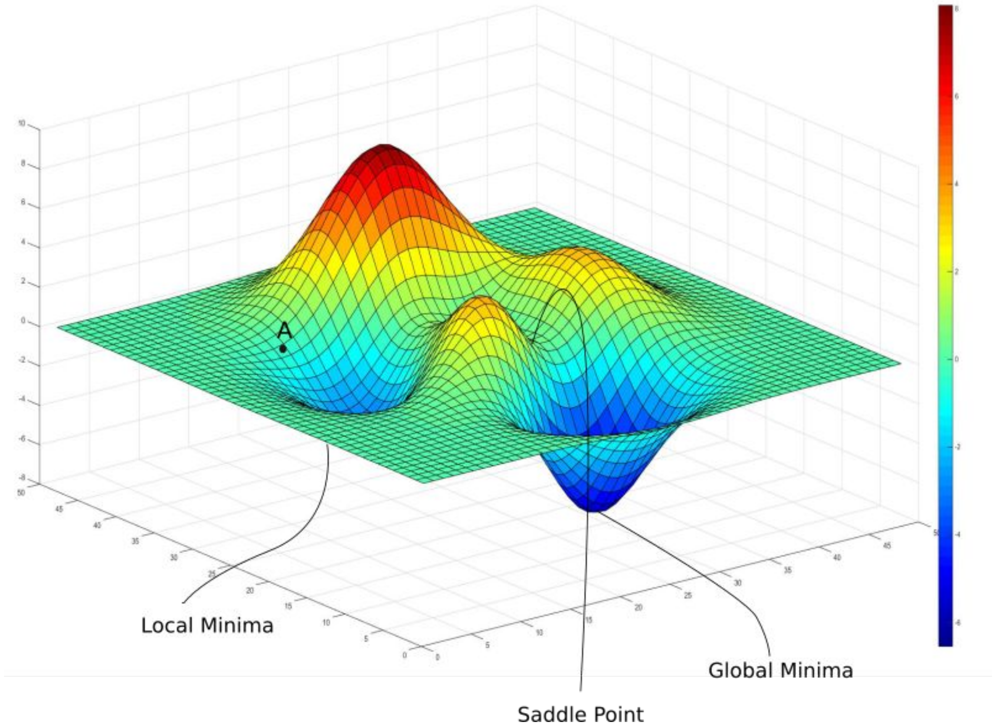
Neural Network for dummies

A Wiley Brand



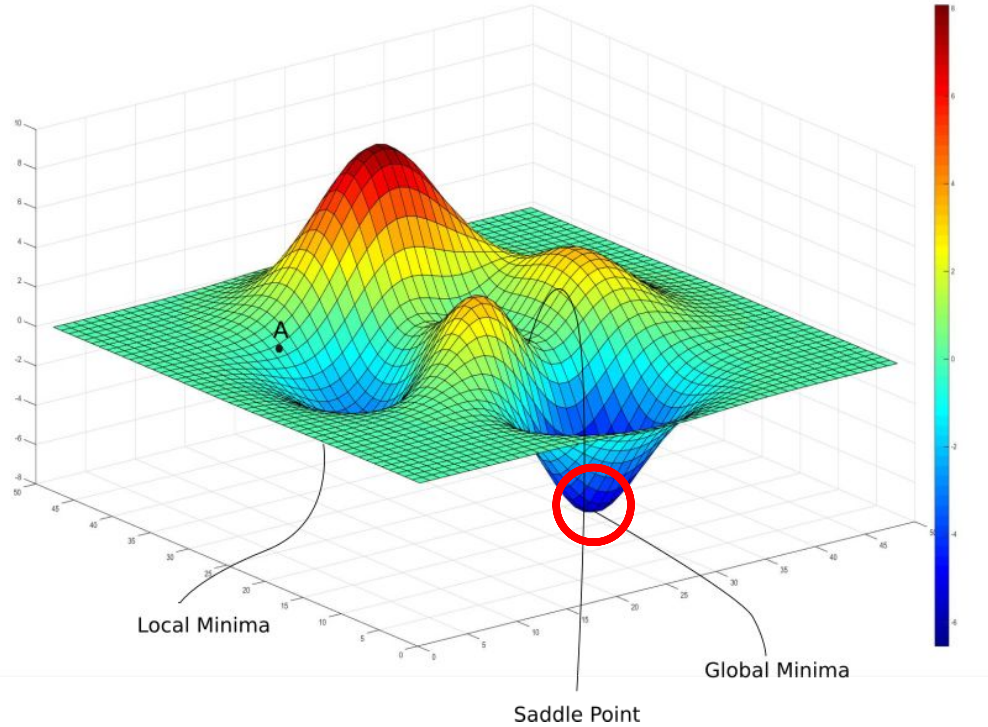
Loss function

$$\text{Loss} = \text{Desired Output - Actuall Output}$$



Loss function

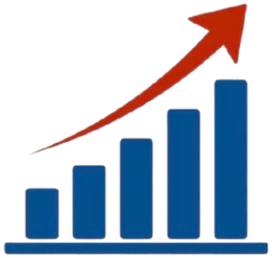
$$\text{Loss} = \text{Desired Output - Actuall Output}$$



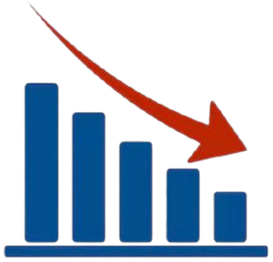
An introduction…

MUON

Better Results



Less Computation



Different Types of Optimizer

Gradient Descent

$$\theta_{t+1} \leftarrow \theta_t - \eta\, g(\theta_t).$$

Updated parameters

Previous parameters

Gradient

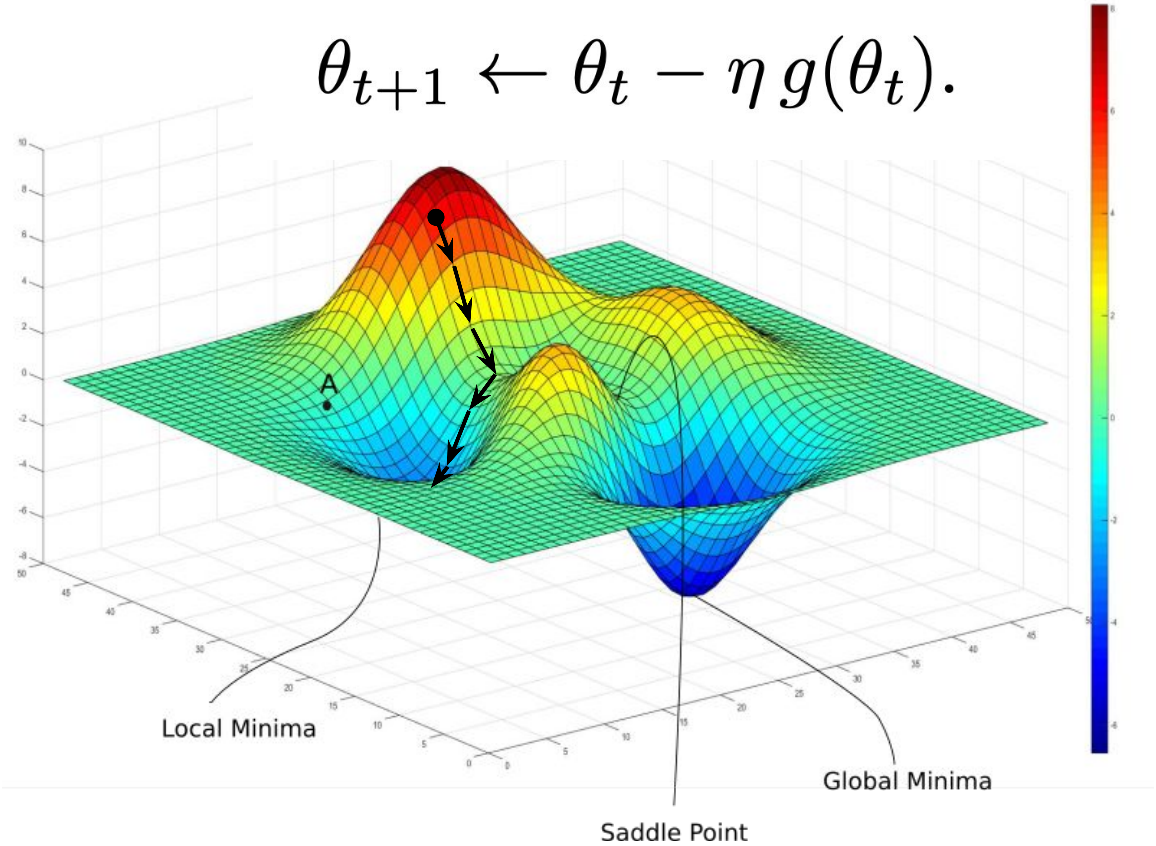
$$\begin{bmatrix} \frac{\partial L(\theta_t)}{\partial \theta_{1,1}} & \cdots & \frac{\partial L(\theta_t)}{\partial \theta_{1,n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L(\theta_t)}{\partial \theta_{m,1}} & \cdots & \frac{\partial L(\theta_t)}{\partial \theta_{m,n}} \end{bmatrix}$$

Adam

Muon

Gradient Descent

$$\theta_{t+1} \leftarrow \theta_t - \eta\, g(\theta_t).$$



The Defacto Standard Optimizer

Adam

$$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g(\theta_t),$$

$$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g(\theta_t)^2.$$

$$\theta_{t+1} \leftarrow \theta_t - \eta \, \frac{m_t}{\sqrt{v_t} + \epsilon},$$

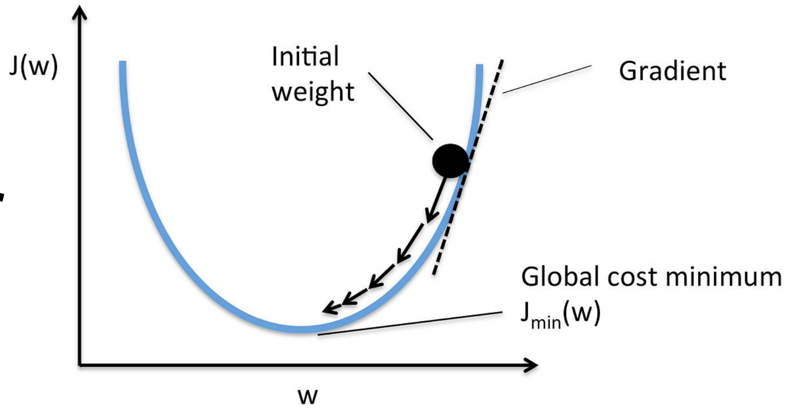
The Defacto Standard Optimizer

Adam

$$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g(\theta_t),$$

$$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g(\theta_t)^2.$$

$$\theta_{t+1} \leftarrow \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon},$$



The Defacto Standard Optimizer

Adam

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g(\theta_t),$$

$$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g(\theta_t)^2$$

$$\theta_{t+1} \leftarrow \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon},$$

Adaptive Scaling Factor

The Defacto Standard Optimizer

Adam

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g(\theta_t),$$
$$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g(\theta_t)^2$$
$$\theta_{t+1} \leftarrow \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon},$$

Adaptive
Scaling Factor
(Expensive to Have)

MomentUm Orthogonalized by Newton-Schulz (MUON)

Adam

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g(\theta_t),$$

$$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g(\theta_t)^2.$$

$$\theta_{t+1} \leftarrow \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon},$$

MUON

$$M_t \leftarrow \beta M_{t-1} + g(\theta_t)$$

$$N_t \leftarrow \frac{M_t}{||M_t||_F}$$

$$O_t \leftarrow NewtonSchulz5(N_t)$$

$$\theta_t \leftarrow \theta_t + \eta O_t$$

MomentUm Orthogonalized by Newton-Schulz (MUON)

Adam

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g(\theta_t),$$

$$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g(\theta_t)^2.$$

$$\theta_{t+1} \leftarrow \theta_t - \eta \, \frac{m_t}{\sqrt{v_t} + \epsilon},$$

MUON

$$M_t \leftarrow \beta M_{t-1} + g(\theta_t)$$

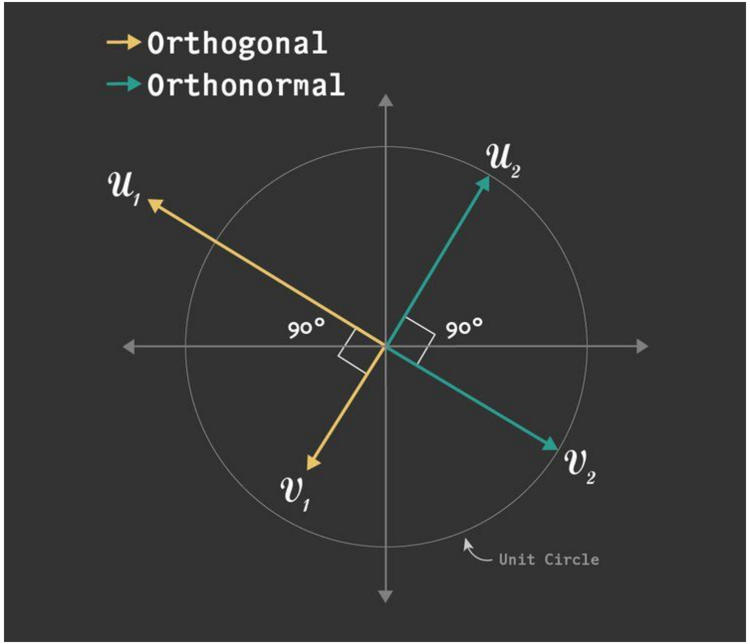
$$\left\{ \begin{aligned} N_t &\leftarrow \frac{M_t}{||M_t||_F} \\ O_t &\leftarrow NewtonSchulz5(N_t) \end{aligned} \right.$$

$$\theta_t \leftarrow \theta_t + \eta O_t$$

MomentUm Orthogonalized by Newton-Schulz (MUON)

$\approx$

Scaling achieved by making the **Momentum Matrix orthonormal** (perpendicular and unit length). →



One way to do this efficiently is through the **Newton-Schulz algorithm**, which MUON uses.

Why This Works

Without it, it has been shown that the momentum matrix tends to become low rank, so a few directions dominate.

The authors hypothesize that orthonormalization balances the effect of smaller directions in updates.

Lastly, update using the Orthonormal Momentum Matrix

$$\theta_{t+1} \leftarrow \theta_t + \eta O_t$$

Newton-Schulz Algorithm

$$\text{Gradient Estimate } g(\theta_t)$$

$$M_t \leftarrow \beta M_{t-1} + g(\theta_t)$$

$$N_t \leftarrow \frac{M_t}{||M_t||_F}$$

$$O_t \leftarrow NewtonSchulz5(N_t)$$

$$\theta_t \leftarrow \theta_t + \eta O_t$$

$$M = U\Sigma V^\top$$

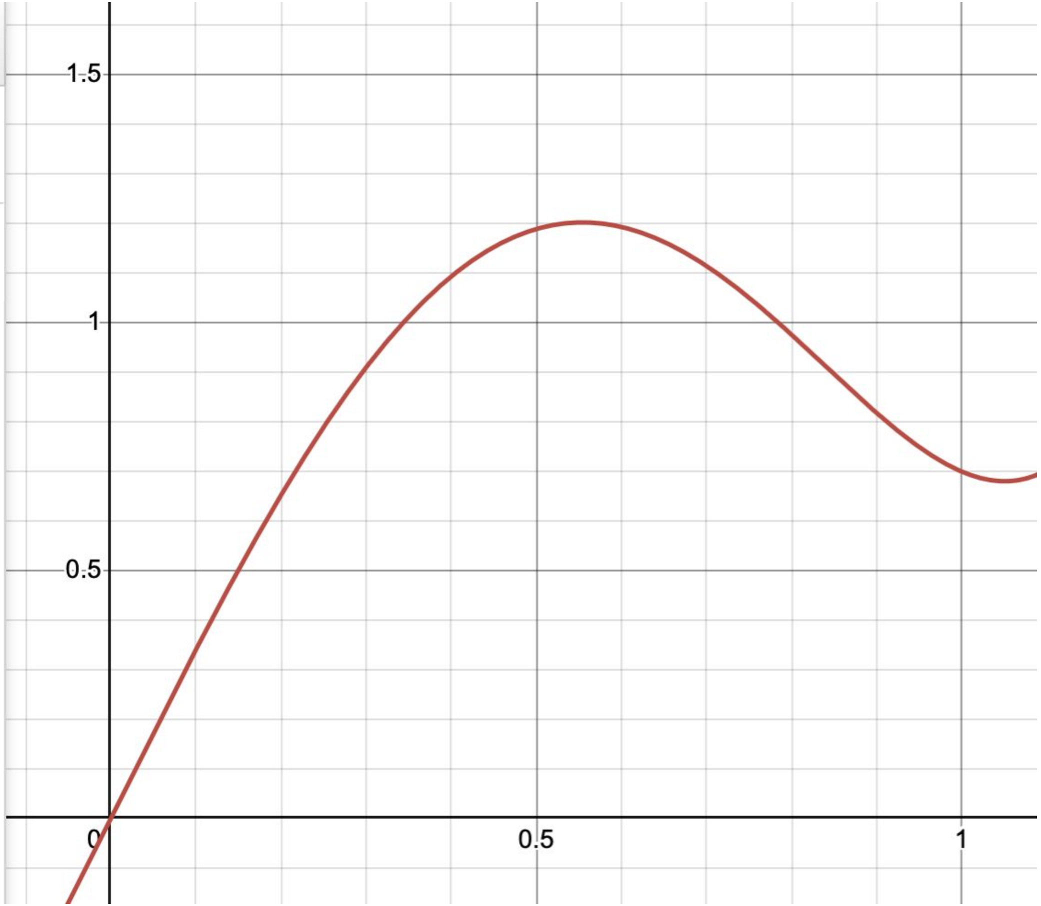
$$\begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_{n-1} & \\ & & & & \sigma_n \end{bmatrix}$$

$$M = U\boxed{\Sigma}V^\top$$

$$p(x) = ax + bx^3 + cx^5$$

$$p(M) = p(U\Sigma V^\top) = Up(\Sigma)V^\top$$

$$p(x) = 3.4445x - 4.777x^3 + 2.0315x^5$$



$$a = 3.4445$$

$$b = 4.7770$$

$$c = 2.0315$$

1

$p(x) = 3.4445x - 4.777x^3 + 2.0315x^5$

2

$p\left(p\left(p\left(p\left(p(x)\right)\right)\right)\right)$

3

$y = 1.3$

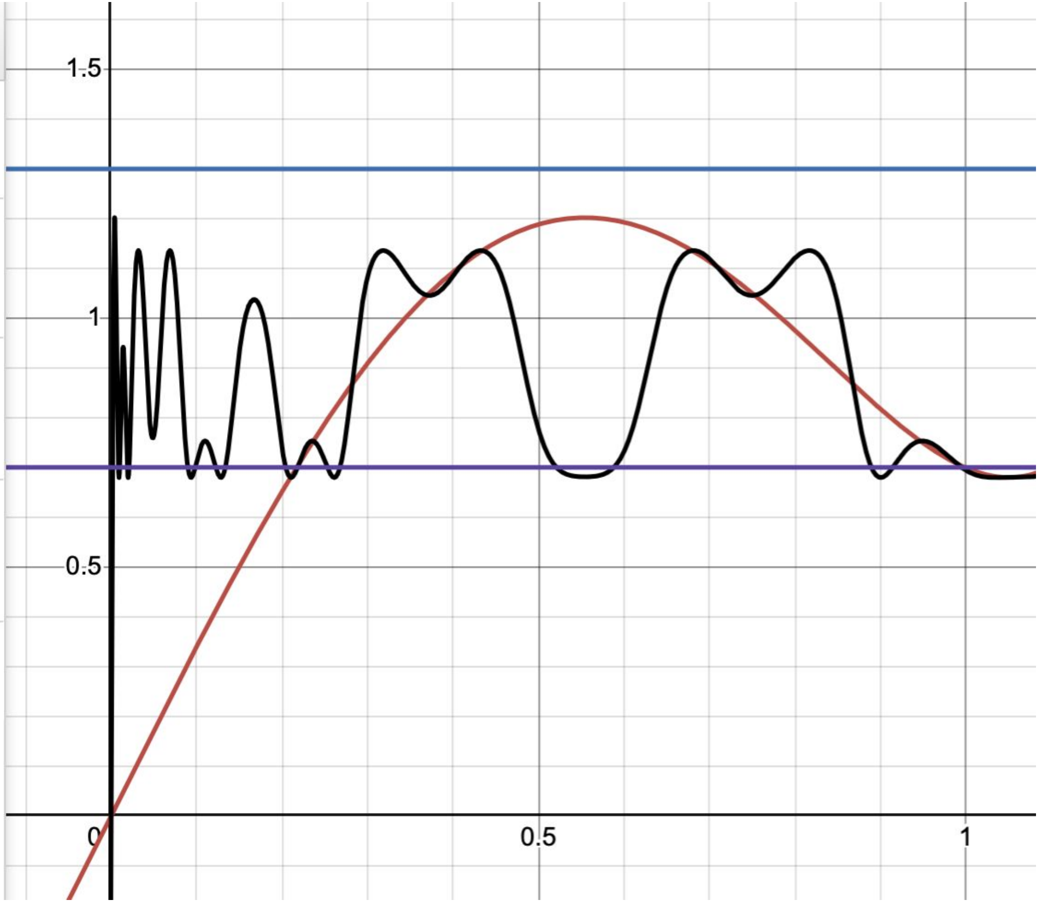
−10 10

4

$y = 0.7$

−10 10

5



Update Rule

$$\text{Gradient Estimate } g(\theta_t)$$

$$M_t \leftarrow \beta M_{t-1} + g(\theta_t)$$

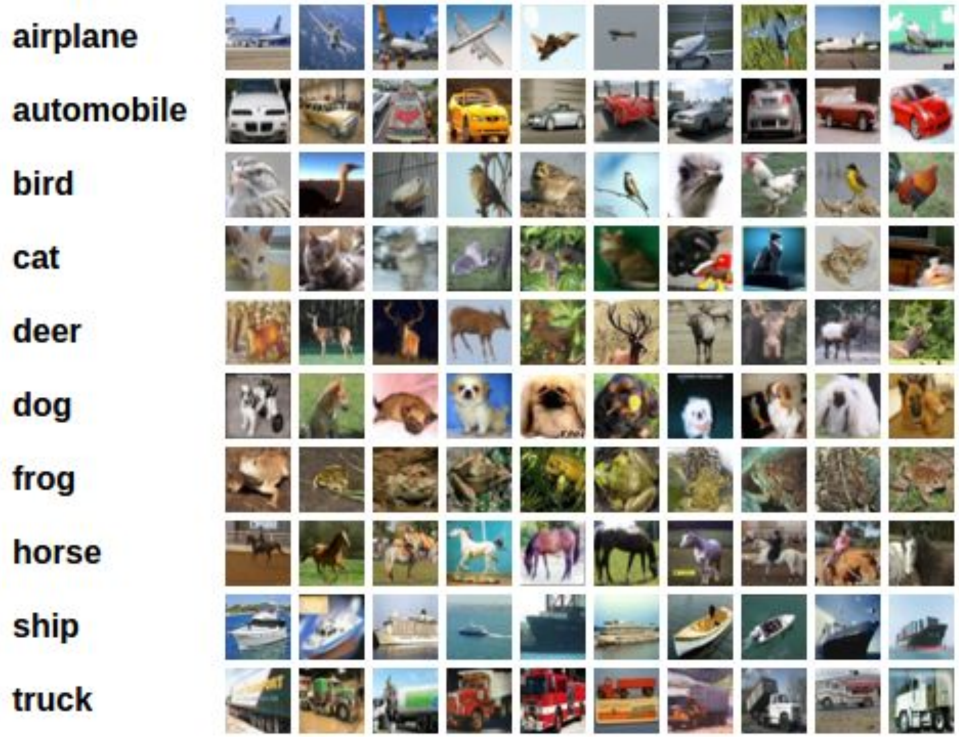
$$N_t \leftarrow \frac{M_t}{||M_t||_F}$$

Normalize to [0, 1] first

$$O_t \leftarrow NewtonSchulz5(N_t)$$

$$\theta_t \leftarrow \theta_t + \eta O_t$$

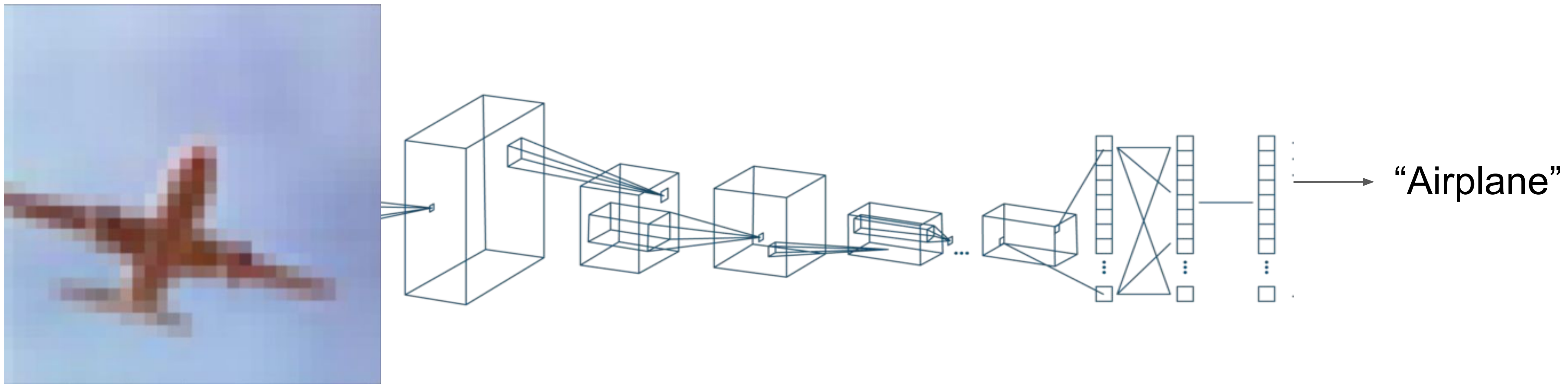
Test MUON Optimizer on CIFAR-10 Data



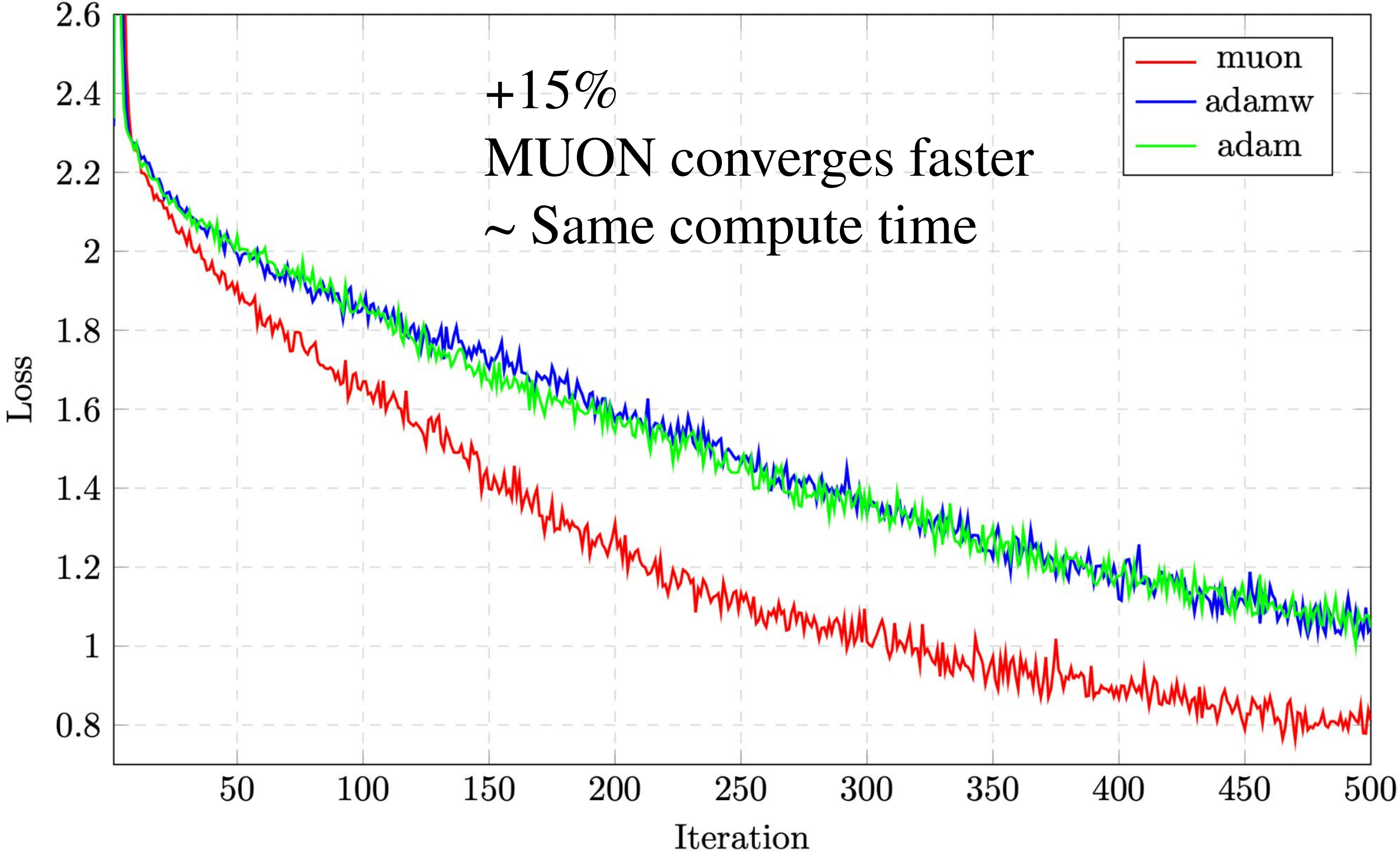
Total of 60k Images

10 classes

Goal



Given Image (Input) → Model → Predict Class (Output)



+15%
MUON converges faster
~ Same compute time
muon
adamw
adam
Loss
Iteration
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
50
100
150
200
250
300
350
400
450
500

Conclusion

Works (very) well.

Worth researching more.

Sources

Code

Bernstein, J. (2024) Newton-Schulz. Available at: https://docs.modula.systems/algorithms/newton-schulz/ (Accessed: 18 November 2025).

Google DeepMind (no date) AlphaFold. Available at: https://deepmind.google/science/alphafold/ (Accessed: 18 November 2025).

Huang, J.-B. (2025) This Simple Optimizer Is Revolutionizing How We Train AI [Muon]. [Video] Available at: https://www.youtube.com/watch?v=bO5nvE289ec (Accessed: 18 November 2025).

Jordan, K., Jin, Y., Boza, V., You, J., Cesista, F., Newhouse, L. and Bernstein, J. (2024) Muon: An optimizer for hidden layers in neural networks. Available at: https://kellerjordan.github.io/posts/muon/ (Accessed: 18 November 2025).

Wang, X. et al. (2024) ‘A pathology foundation model for cancer diagnosis and prognosis prediction’, Nature, 634, pp. 970–978. Available at: https://doi.org/10.1038/s41586-024-07894-z.