DL Scaling rules

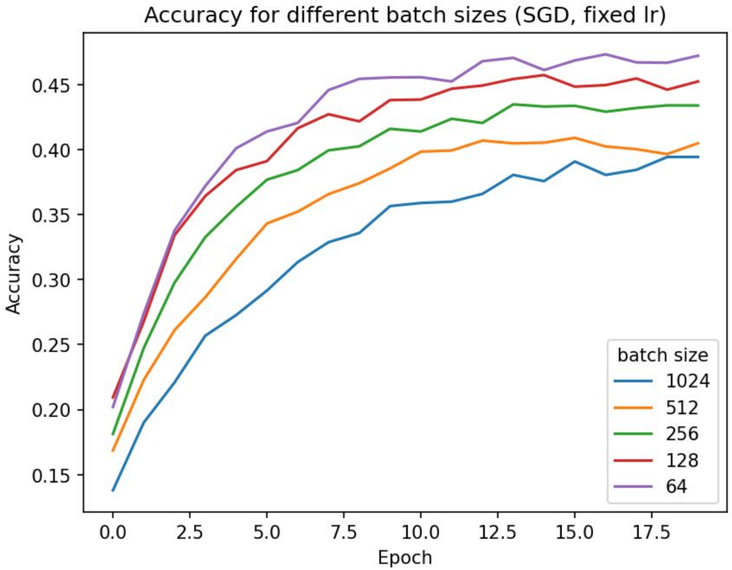
Denis Sapozhnikov, HSE AMI 202

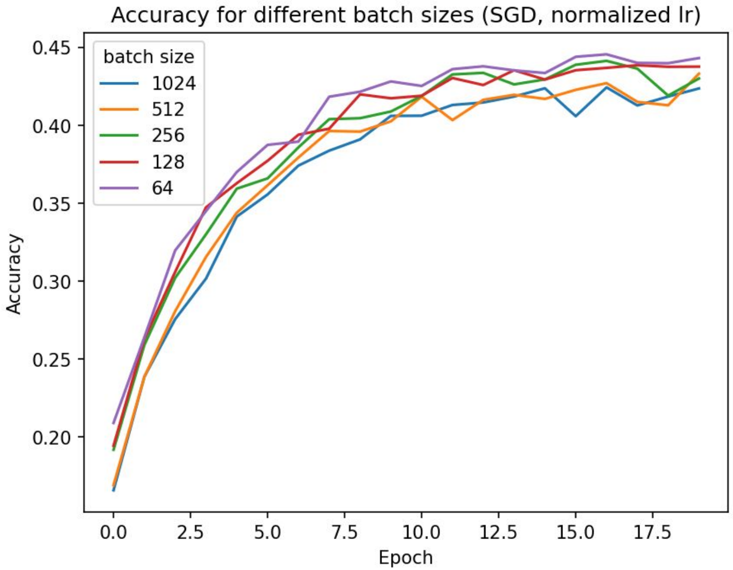
Motivation

- You want to reproduce Somebody et al. from AnyAI
- You have implemented the entire article
- You do not have 8192 TPU as Somebody et al
- You choose small batch size and keep the other parameters the same
- You sucks

SGD scaling rule (Goyal et al. 2017)

Linear Scaling Rule: *When the minibatch size is multiplied by k, multiply the learning rate by k.*

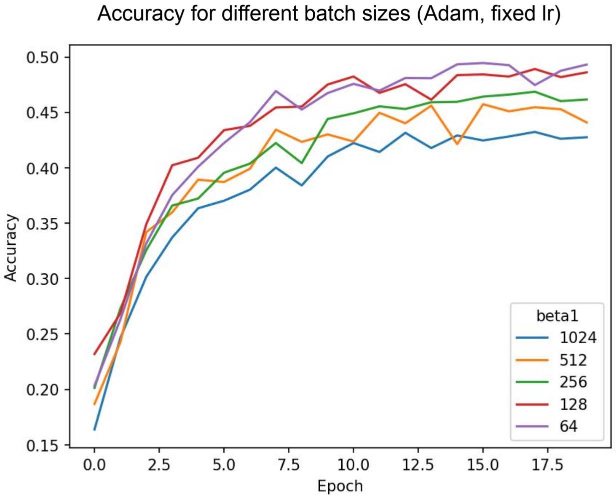


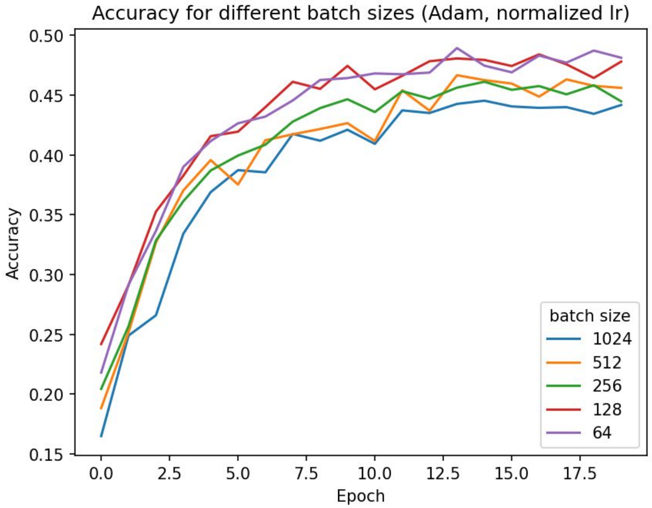


Adam scaling rule (Malladi et al. 2022)

Definition C.3 (Adam Scaling Rule). *When running Adam (Kingma & Ba, 2015) with batch size* $\hat{B} = \kappa B$, *use a learning rate* $\hat{\eta} = \sqrt{\kappa}\eta$, *beta coefficients* $\hat{\beta}_1 = 1 - \kappa \times (1 - \beta_1)$, $\hat{\beta}_2 = 1 - \kappa \times (1 - \beta_2)$, *and adaptivity parameter* $\hat{\epsilon} = \frac{\epsilon}{\sqrt{\kappa}}$ *(Malladi et al., 2022).*

- Beta1 is typically 0.9
- Formula breaks for k > 10 🤡
- Should we really scale beta's?
- Well, may be

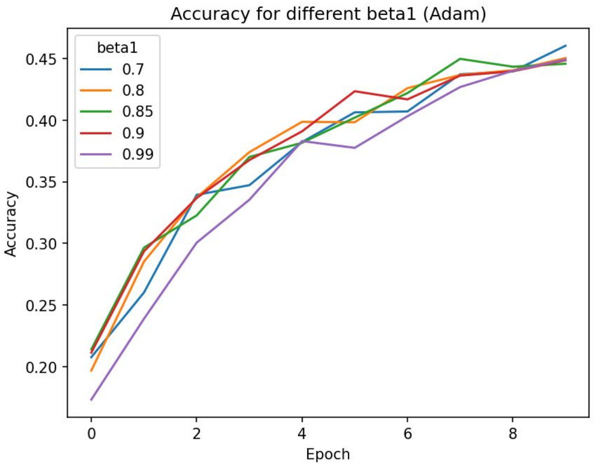


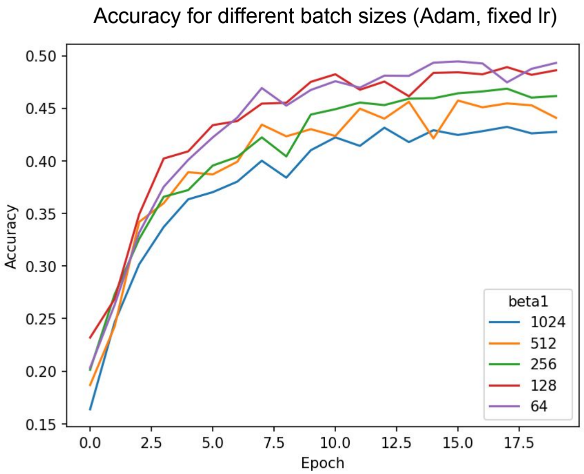


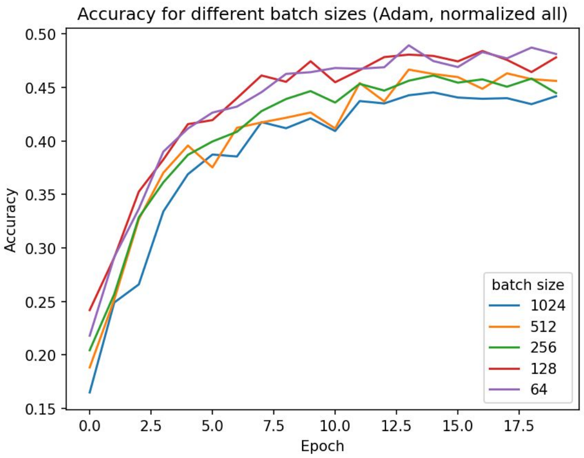
Adam scaling rule (Malladi et al. 2022): scale betas

Definition C.3 (Adam Scaling Rule). *When running Adam (Kingma & Ba, 2015) with batch size $\hat{B} = \kappa B$, use a learning rate $\hat{\eta} = \sqrt{\kappa}\eta$, beta coefficients $\hat{\beta}_1 = 1 - \kappa \times (1 - \beta_1)$, $\hat{\beta}_2 = 1 - \kappa \times (1 - \beta_2)$, and adaptivity parameter $\hat{\epsilon} = \frac{\epsilon}{\sqrt{\kappa}}$ (Malladi et al., 2022).*

- Cannot find any dependency on beta1



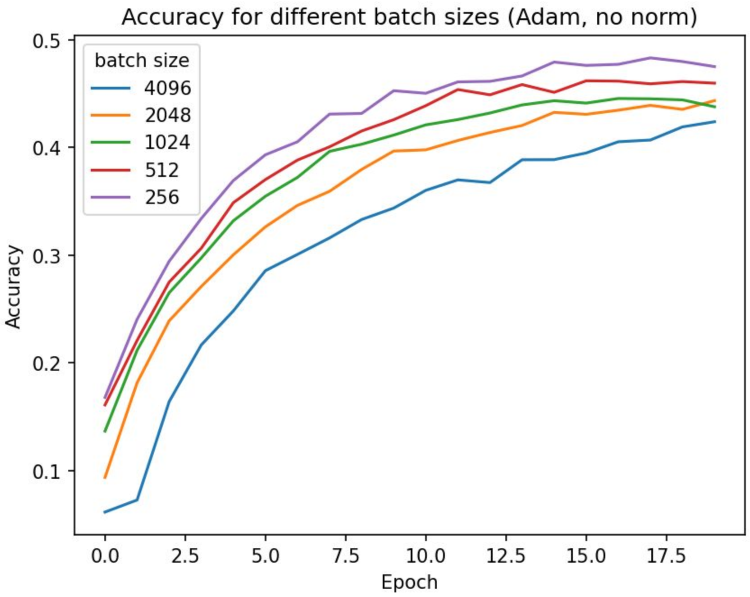


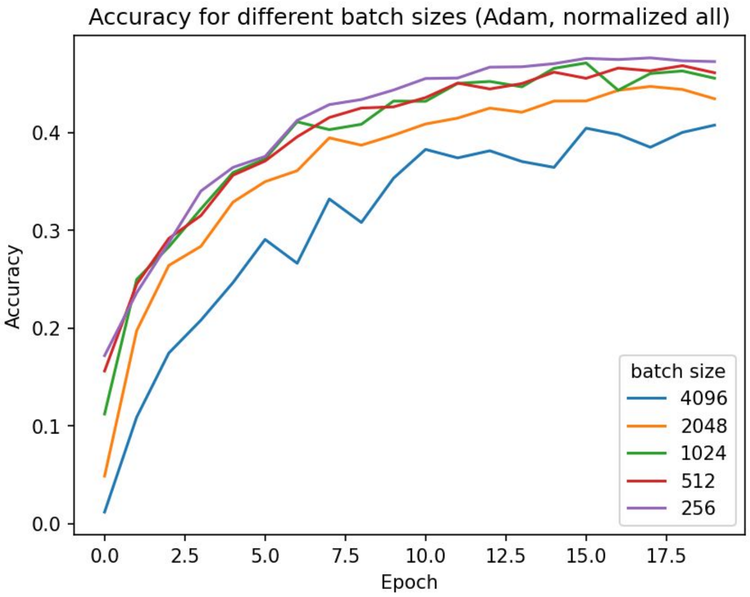


better, but not too much

Adam scaling rule (Malladi et al. 2022): large batches

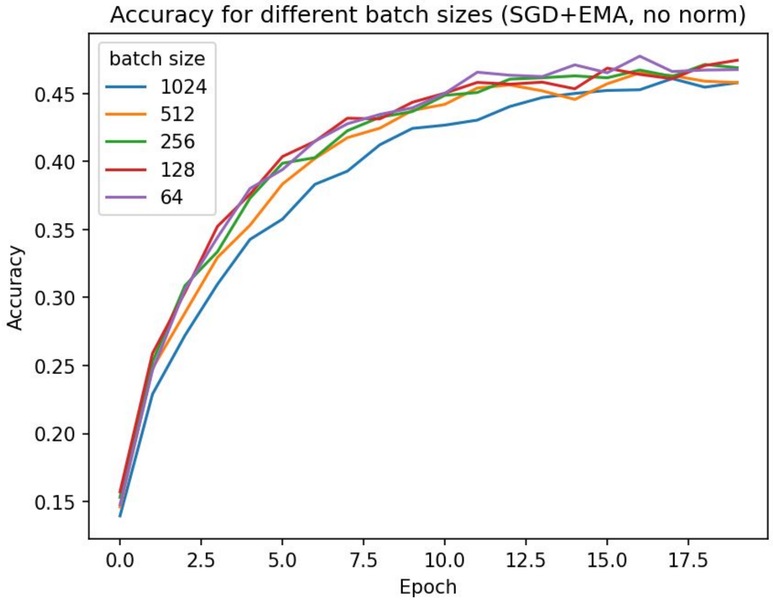
Definition C.3 (Adam Scaling Rule). *When running Adam (Kingma & Ba, 2015) with batch size $\hat{B} = \kappa B$, use a learning rate $\hat{\eta} = \sqrt{\kappa}\eta$, beta coefficients $\hat{\beta}_1 = 1 - \kappa \times (1 - \beta_1)$, $\hat{\beta}_2 = 1 - \kappa \times (1 - \beta_2)$, and adaptivity parameter $\hat{\epsilon} = \frac{\epsilon}{\sqrt{\kappa}}$ (Malladi et al., 2022).*

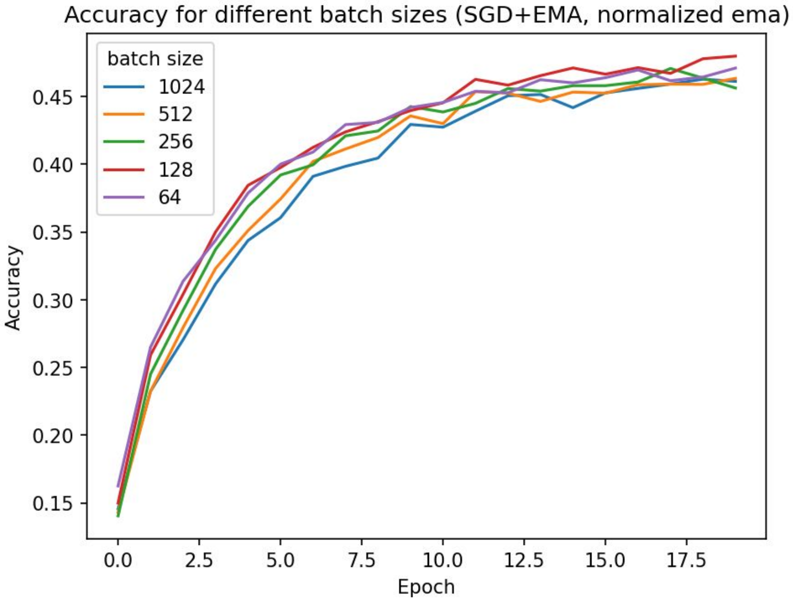




EMA scaling rule (Busbridge et al. 2023)

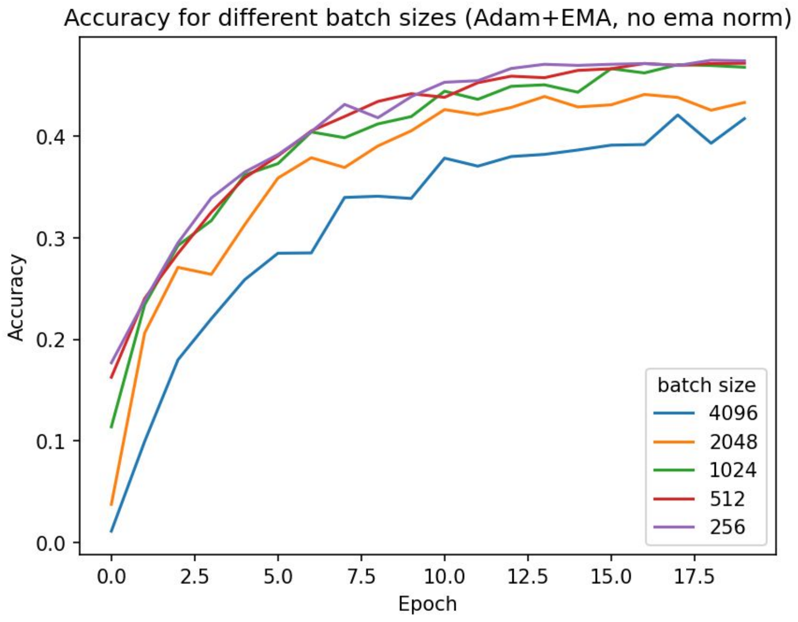
Definition 1.2 (EMA Scaling Rule). *When computing the EMA update (Definition 1.1) of a model undergoing stochastic optimization with batch size $\hat{B} = \kappa B$, use a momentum $\hat{\rho} = \rho^{\kappa}$ and scale other optimizers according to their own scaling rules.*

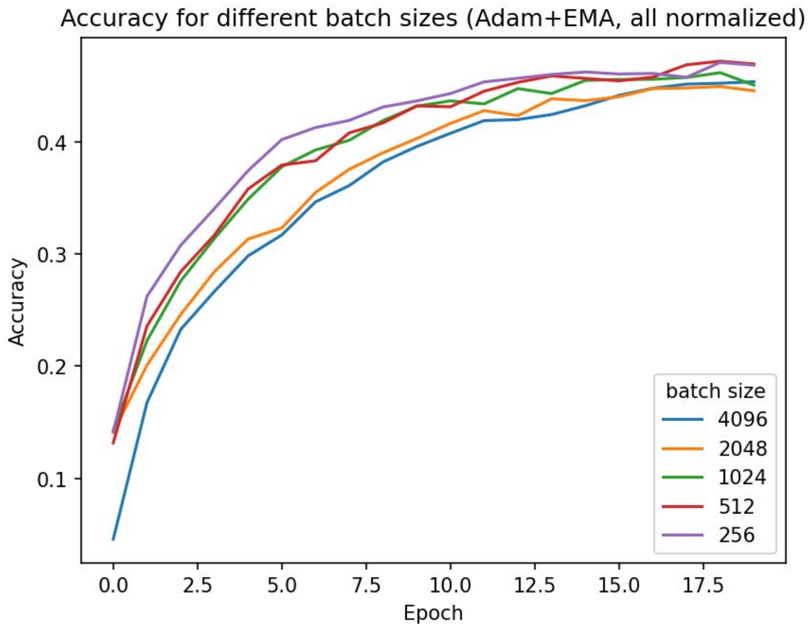




EMA scaling rule (Busbridge et al. 2023): Adam

Definition 1.2 (EMA Scaling Rule). *When computing the EMA update (Definition 1.1) of a model undergoing stochastic optimization with batch size $\hat{B} = \kappa B$, use a momentum $\hat{\rho} = \rho^{\kappa}$ and scale other optimizers according to their own scaling rules.*





Links:

- Busbridge et al. How to Scale Your EMA. [link]
- Goyal et al. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. [link]
- Malladi et al. On the SDEs and Scaling Rules for Adaptive Gradient Algorithms.[link]
- You et al. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. [link]
- Godbole et al. Deep Learning Tuning Playbook. [link]

LAMB scaling rule* (You et al. 2020)

Batch Size	512	1K	2K	4K	8K	16K	32K
Learning Rate	$\frac{4}{2^{3.0}\times 100}$	$\frac{4}{2^{2.5}\times 100}$	$\frac{4}{2^{2.0}\times 100}$	$\frac{4}{2^{1.5}\times 100}$	$\frac{4}{2^{1.0}\times 100}$	$\frac{4}{2^{0.5}\times 100}$	$\frac{4}{2^{0.0}\times 100}$
Warmup Epochs	0.3125	0.625	1.25	2.5	5	10	20
Top-5 Accuracy	0.9335	0.9349	0.9353	0.9332	0.9331	0.9322	0.9308
Top-1 Accuracy	0.7696	0.7706	0.7711	0.7692	0.7689	0.7666	0.7642

SGD

$$v_{t+1} = \mu \cdot v_t - \eta \cdot \nabla_\theta J(\theta)$$
$$\theta_{t+1} = \theta_t + v_{t+1}$$

Where:

- v_t is the momentum term at iteration t.
- μ is the momentum coefficient.

LARS

$$\eta^l = \eta \cdot \frac{\|\theta^l\|}{\|\nabla_\theta J(\theta^l)\| + \lambda \cdot \|\theta^l\|}$$
$$\theta^l_{t+1} = \theta^l_t - \eta^l \cdot \nabla_\theta J(\theta^l)$$

Here:

- θ^l and $\nabla_\theta J(\theta^l)$ are the parameters and their gradients for layer l.
- η^l is the adaptive learning rate for layer l.
- η is the global learning rate.
- λ is a weight decay parameter.

LAMB

1. **Compute the moment estimates:**
   $$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla_\theta J(\theta)$$
   $$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot \nabla_\theta J(\theta)^2$$
2. **Compute bias-corrected moment estimates:**
   $$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
   $$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$
3. **Layer-wise learning rate adaptation:**
   $$r^l_t = \frac{\|\hat{m}^l_t\|}{\sqrt{\hat{v}^l_t} + \epsilon}$$
   $$\eta^l = \eta \cdot \frac{\|\theta^l\|}{r^l_t}$$
4. **Parameter update:**
   $$\theta^l_{t+1} = \theta^l_t - \eta^l \cdot \hat{m}^l_t$$