Don’t Decay the Learning Rate, Increase the Batch Size

Samuel L. Smith, Pieter-Jan Kindermans, Quoc V. Le

December 9th 2017

slsmith@

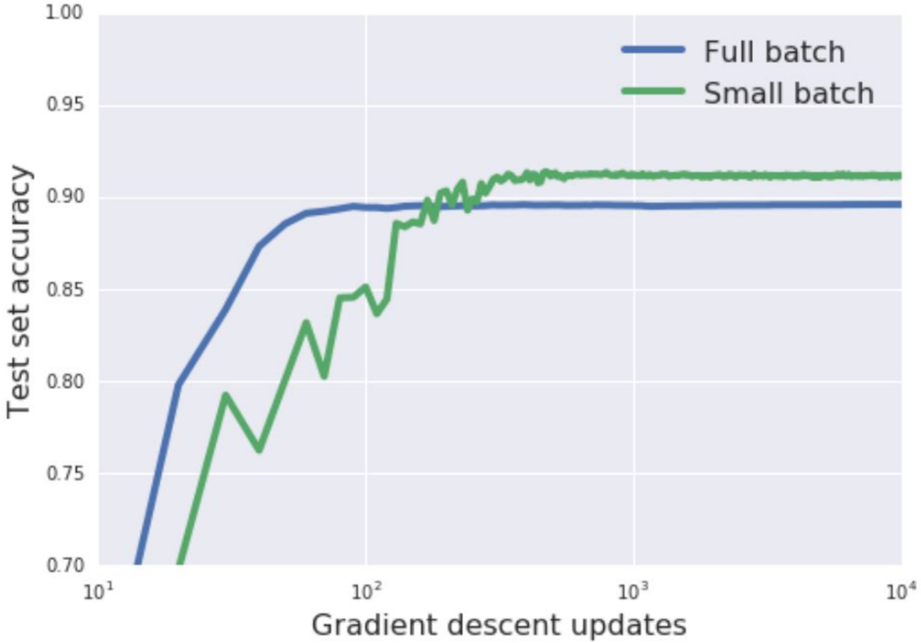
Google Brain

Three related questions:

- *What properties control generalization?*
- *How should we tune SGD hyper-parameters?*
- *Can we train efficiently with large batches? (> 50,000 examples)*

Small batches out-generalize large batches

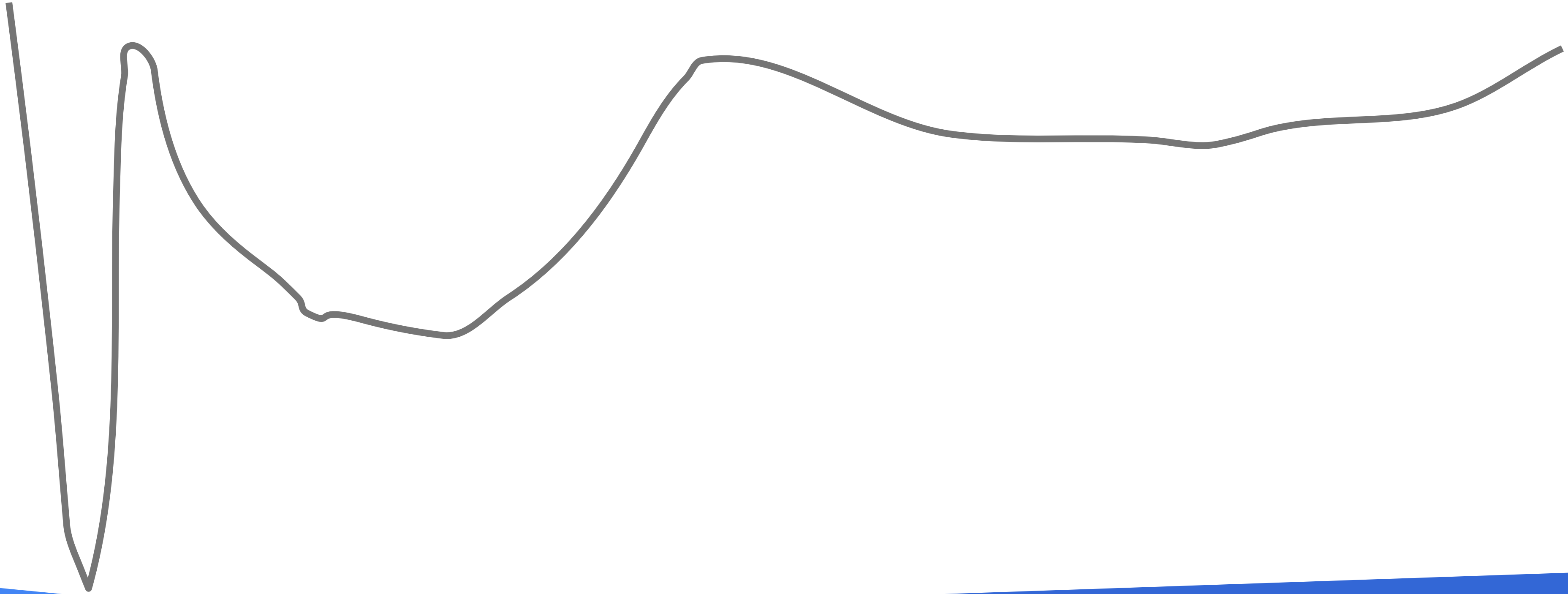
(at constant learning rate)



As observed by:

"On Large Batch Training…", Keskar et al. (2017)

Which minimum is best?



Bayesian model comparison

Bayesian model comparison

$$\frac{P(M_1|\{y\},\{x\})}{P(M_2|\{y\},\{x\})} = \frac{P(\{y\}|\{x\};M_1)}{P(\{y\}|\{x\};M_2)}\frac{P(M_1)}{P(M_2)}$$

Bayesian model comparison

$$\frac{P(M_1|\{y\},\{x\})}{P(M_2|\{y\},\{x\})} = \frac{P(\{y\}|\{x\};M_1)}{P(\{y\}|\{x\};M_2)}\frac{P(M_1)}{P(M_2)}$$

Probability ratio of two competing models

Bayesian model comparison

Prior probability ratio of the models. Usually 1.

$$\frac{P(M_1|\{y\},\{x\})}{P(M_2|\{y\},\{x\})} = \frac{P(\{y\}|\{x\};M_1)}{P(\{y\}|\{x\};M_2)} \frac{P(M_1)}{P(M_2)}$$

Probability ratio of two competing models

Bayesian model comparison

Prior probability ratio of the models. Usually 1.

$$\frac{P(M_1|\{y\},\{x\})}{P(M_2|\{y\},\{x\})} = \frac{P(\{y\}|\{x\};M_1)}{P(\{y\}|\{x\};M_2)}\frac{P(M_1)}{P(M_2)}$$

Probability ratio of two competing models

The evidence ratio!

The Bayesian evidence

(Gaussian approximation)

λ_i is the i^{th} Hessian eigenvalue

λ is the L2 regularization parameter

$$P(\{y\}|\{x\}; M) \approx \exp\left\{-\left(C(\omega_0) + \frac{1}{2}\sum_{i=1}^{P}\ln(\lambda_i/\lambda)\right)\right\}$$

The Bayesian evidence

(Gaussian approximation)

λ_i is the i[th] Hessian eigenvalue

λ is the L2 regularization parameter

$$P(\{y\}|\{x\};M) \approx \exp\left\{-\left(C(\omega_0) + \frac{1}{2}\sum_{i=1}^{P}\ln(\lambda_i/\lambda)\right)\right\}$$

Evidence for a minimum

The Bayesian evidence
(Gaussian approximation)

λ_i is the i^{th} Hessian eigenvalue

λ is the L2 regularization parameter

$$P(\{y\}|\{x\};M) \approx \exp\left\{-\left(C(\omega_0)+\frac{1}{2}\sum_{i=1}^{P}\ln(\lambda_i/\lambda)\right)\right\}$$

Evidence for a minimum

Depth of the minimum

The Bayesian evidence

(Gaussian approximation)

λ_i is the i^{th} Hessian eigenvalue

λ is the L2 regularization parameter

$$P(\{y\}|\{x\};M) \approx \exp\left\{-\left(C(\omega_0) + \frac{1}{2}\sum_{i=1}^{P}\ln(\lambda_i/\lambda)\right)\right\}$$

Evidence for a minimum

Depth of the minimum

Width of the minimum

The Bayesian evidence
(Gaussian approximation)

λ_i is the ith Hessian eigenvalue

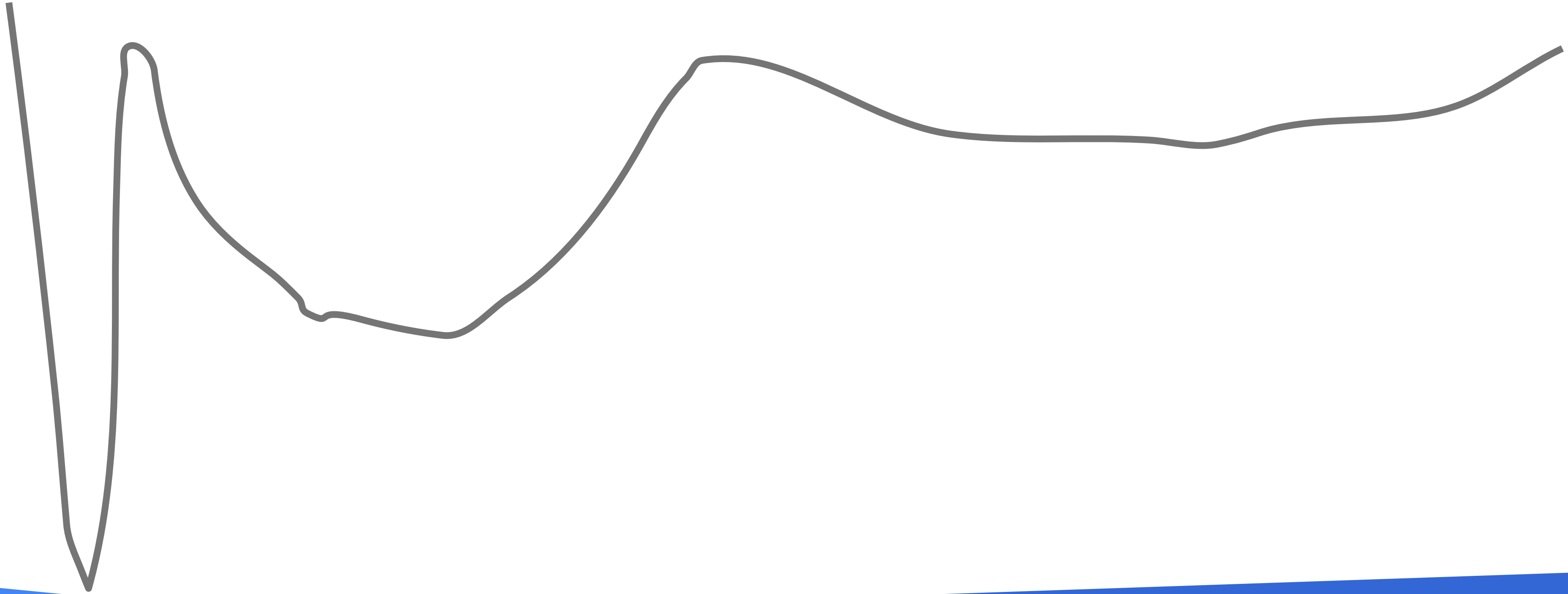
λ is the L2 regularization parameter

$$P(\{y\}|\{x\};M) \approx \exp\left\{-\left(C(\omega_0) + \frac{1}{2}\sum_{i=1}^{P}\ln(\lambda_i/\lambda)\right)\right\}$$

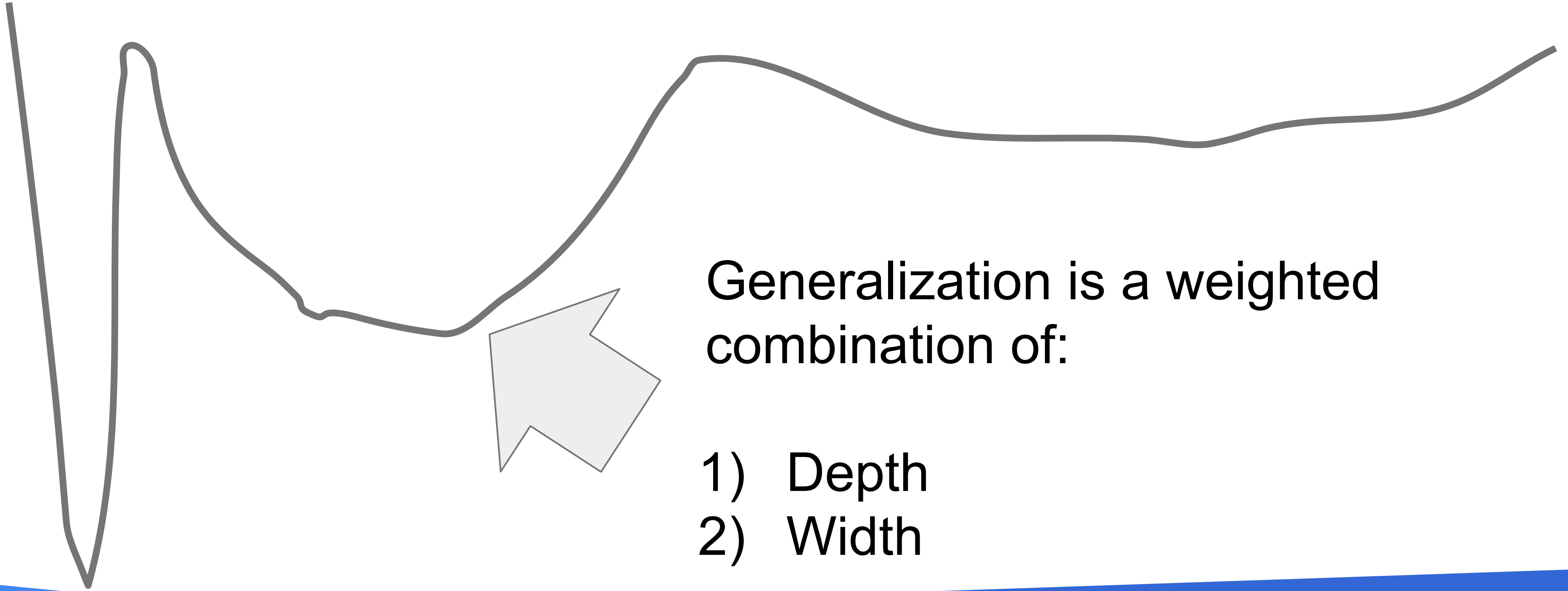
Invariant to changes in model parameterization
(sharp minima can't generalize!)

Width of the minimum

Which minimum is best?



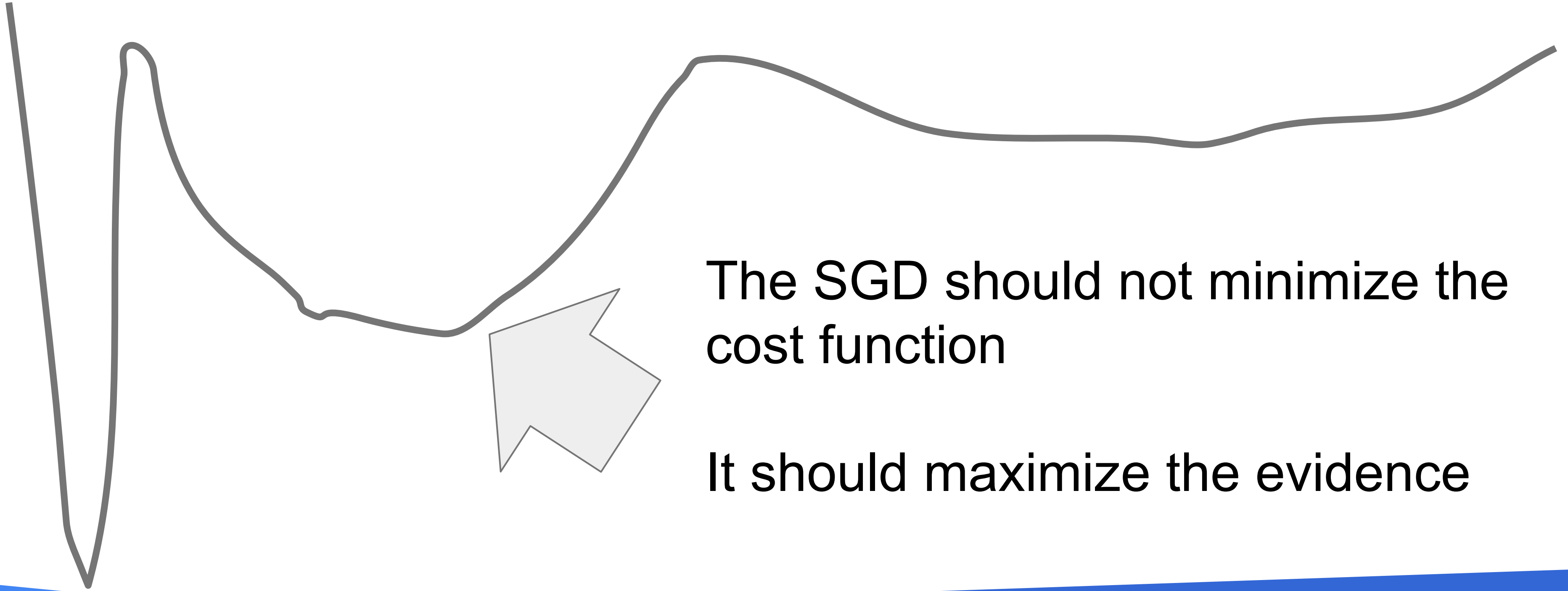
Which minimum is best?



Generalization is a weighted combination of:

1) Depth
2) Width

Which minimum is best?



The SGD should not minimize the cost function

It should maximize the evidence

The SGD gradient update

$$\Delta\omega \quad = \quad \frac{\epsilon}{N}\left(\frac{dC}{d\omega} + \left(\frac{d\hat{C}}{d\omega} - \frac{dC}{d\omega}\right)\right)$$

True gradient

Noise

The SGD gradient update

$$\Delta\omega = \frac{\epsilon}{N}\left(\frac{dC}{d\omega} + \left(\frac{d\hat{C}}{d\omega} - \frac{dC}{d\omega}\right)\right)$$

$$\alpha = \left(\frac{d\hat{C}}{d\omega} - \frac{dC}{d\omega}\right)$$

The SGD gradient update

$$\Delta\omega = \frac{\epsilon}{N}\left(\frac{dC}{d\omega} + \left(\frac{d\hat{C}}{d\omega} - \frac{dC}{d\omega}\right)\right)$$

$$\langle\alpha\rangle = 0$$

$$\alpha = \left(\frac{d\hat{C}}{d\omega} - \frac{dC}{d\omega}\right)$$

$$\langle\alpha^2\rangle \approx N^2 F(\omega)/B$$

The SGD gradient update

$$\Delta\omega = \frac{\epsilon}{N}\left(\frac{dC}{d\omega} + \left(\frac{d\hat{C}}{d\omega} - \frac{dC}{d\omega}\right)\right)$$

Batch size

$$\langle\alpha\rangle = 0$$

$$\alpha = \left(\frac{d\hat{C}}{d\omega} - \frac{dC}{d\omega}\right)$$

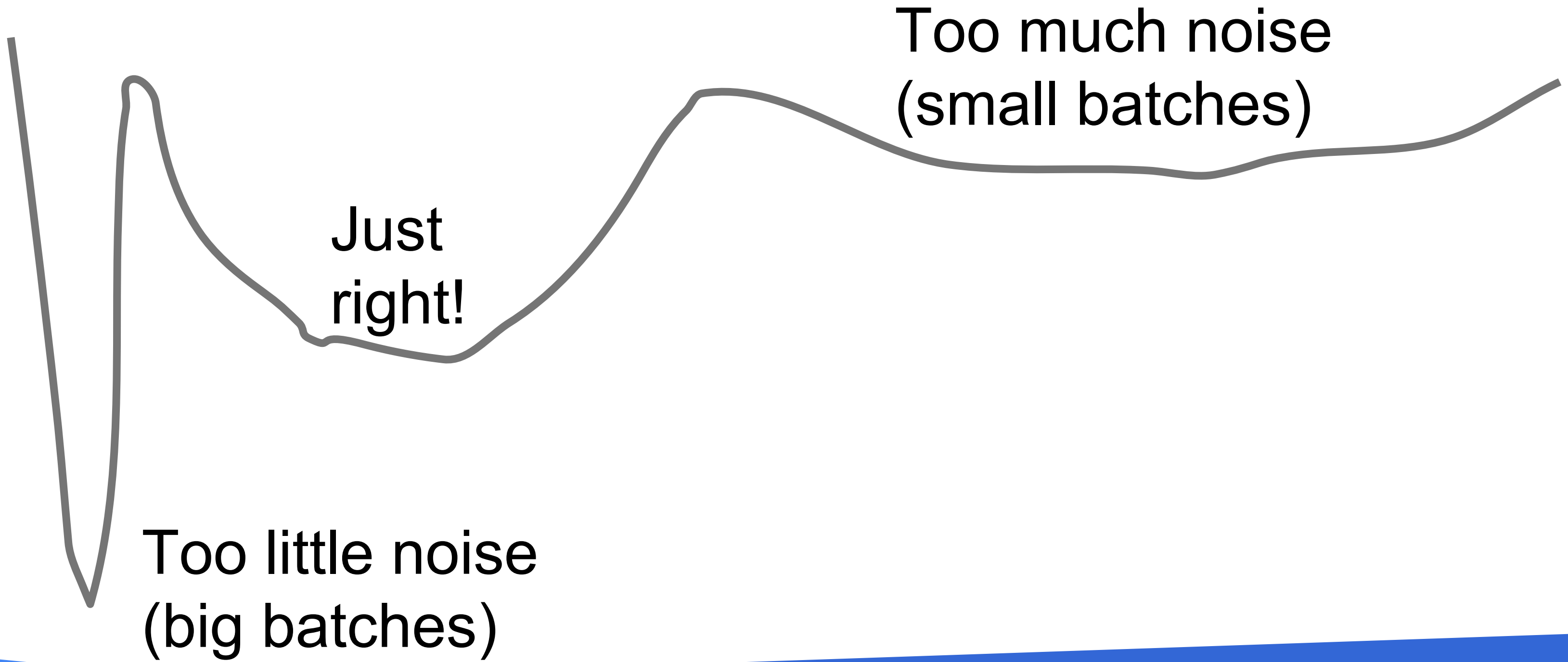
$$\langle\alpha^2\rangle \approx N^2 F(\omega)/B$$

How to choose the batch size?

(at constant learning rate)

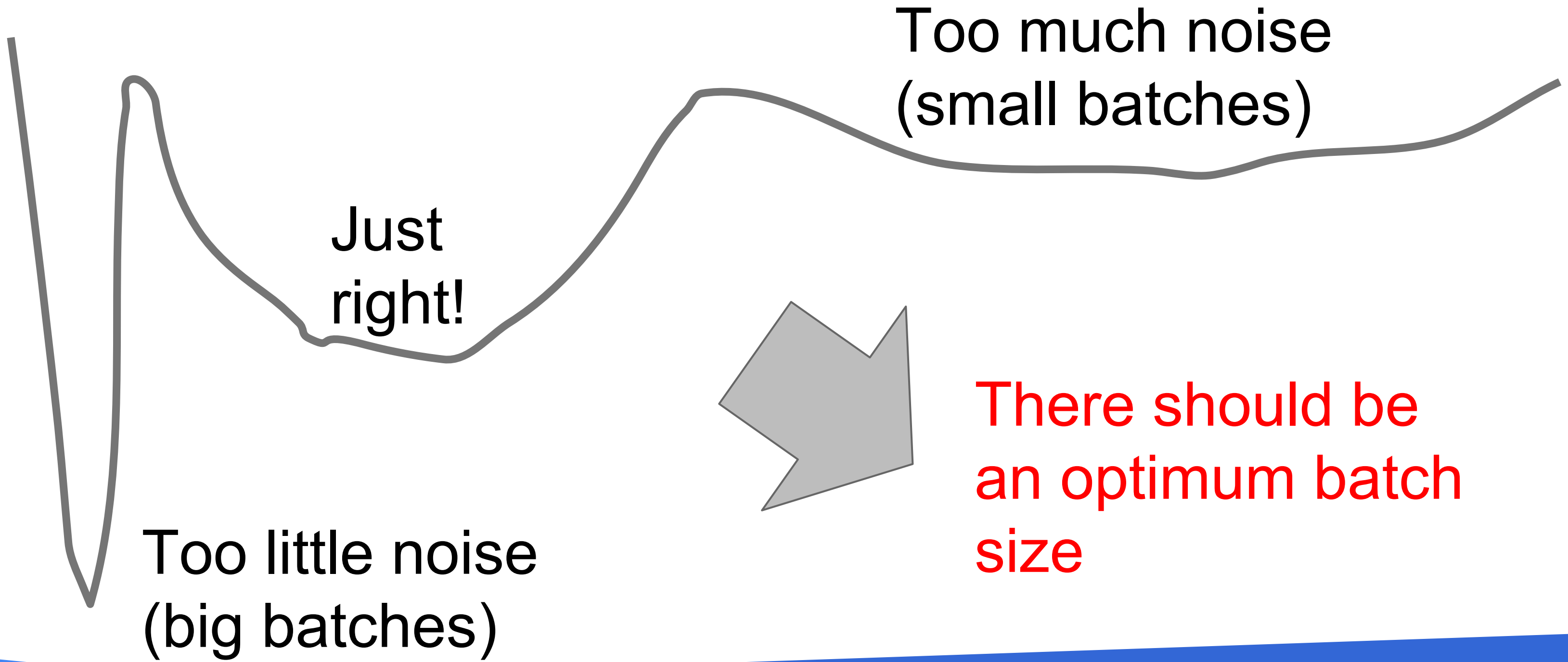
How to choose the batch size?

(at constant learning rate)



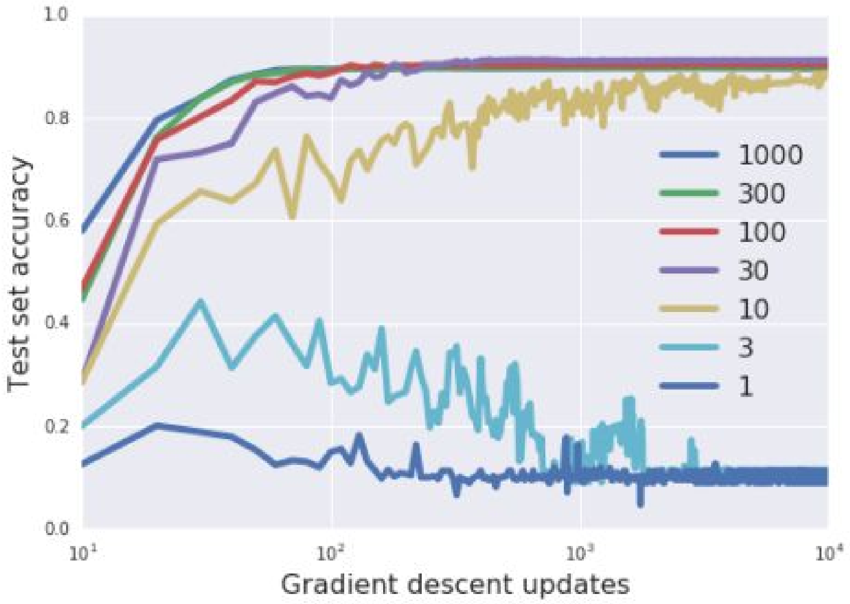
How to choose the batch size?

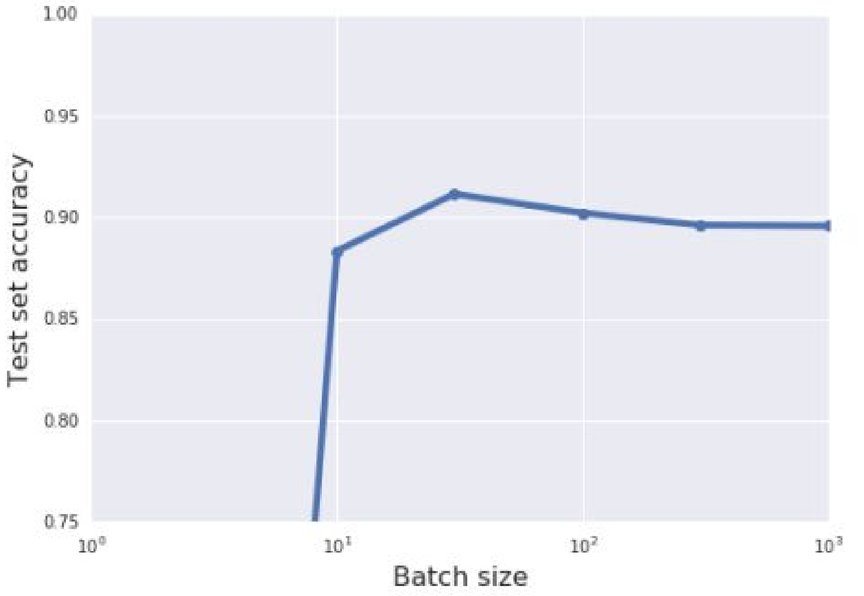
(at constant learning rate)



There should be an optimum batch size

How to choose the batch size?
(at constant learning rate)

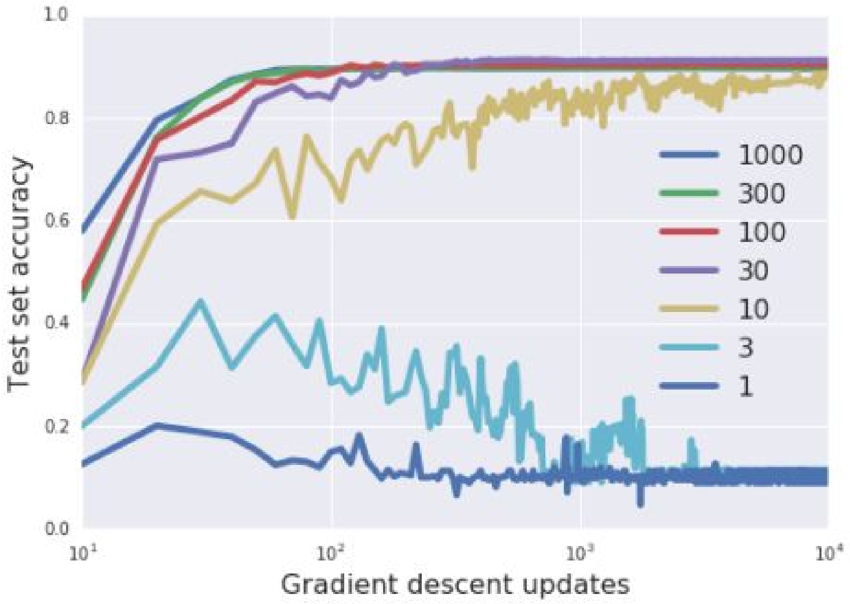


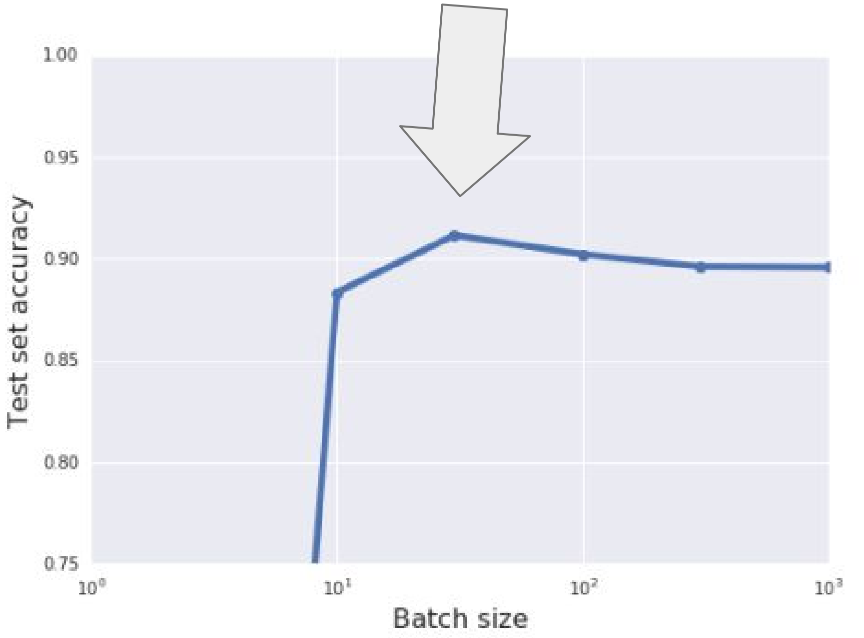


How to choose the batch size?

(at constant learning rate)

As predicted!





Defining the SGD “noise scale”

Defining the SGD “noise scale”

SGD integrates an underlying stochastic differential equation

$$\frac{d\omega}{dt} = \frac{dC}{d\omega} + \eta(t) \qquad \langle \eta(t) \rangle = 0$$

$$\langle \eta(t)\eta(t') \rangle = gF(\omega)\delta(t - t')$$

Defining the SGD "noise scale"

SGD integrates an underlying stochastic differential equation

$$\frac{d\omega}{dt} = \frac{dC}{d\omega} + \eta(t) \qquad \langle \eta(t) \rangle = 0$$

$$\langle \eta(t)\eta(t') \rangle = gF(\omega)\delta(t - t')$$

"Noise scale"

Defining the SGD “noise scale”

SGD integrates an underlying stochastic differential equation

After a little math:

$$g \approx \epsilon N/B$$

Defining the SGD “noise scale”

SGD integrates an underlying stochastic differential equation

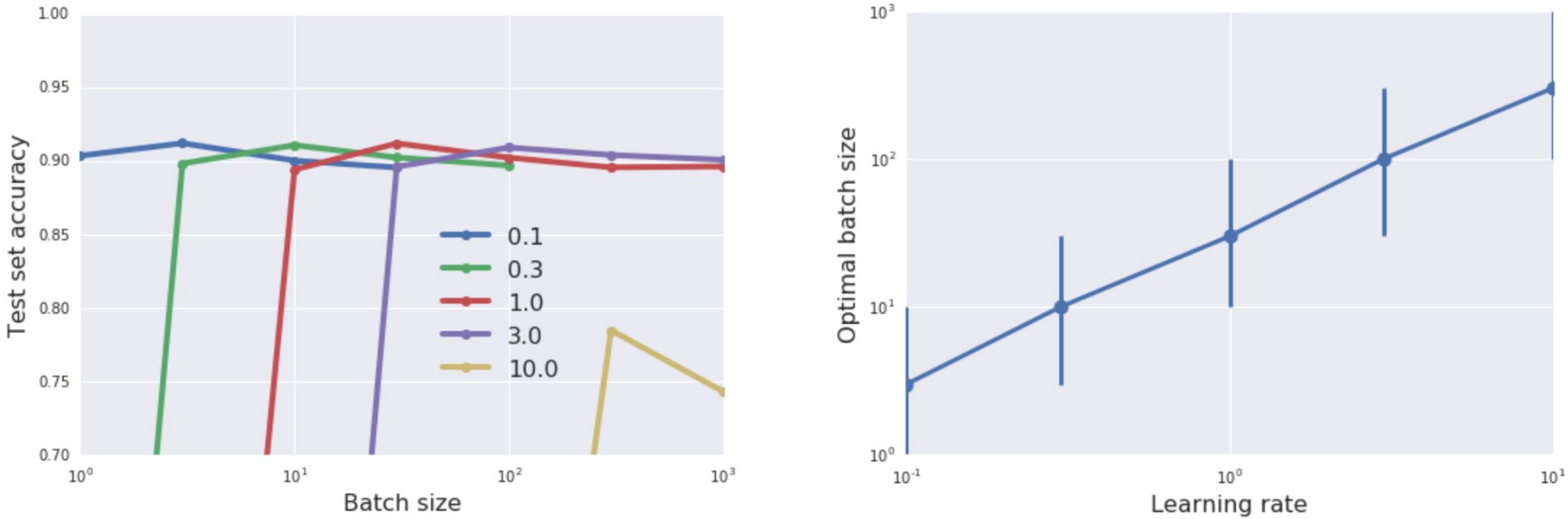
After a little math:

$$g \approx \epsilon N / B$$

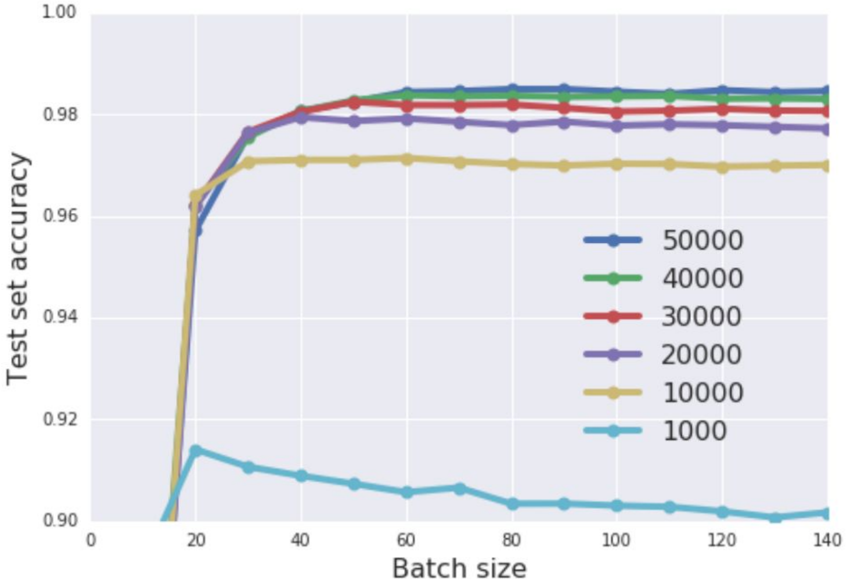
Prediction:

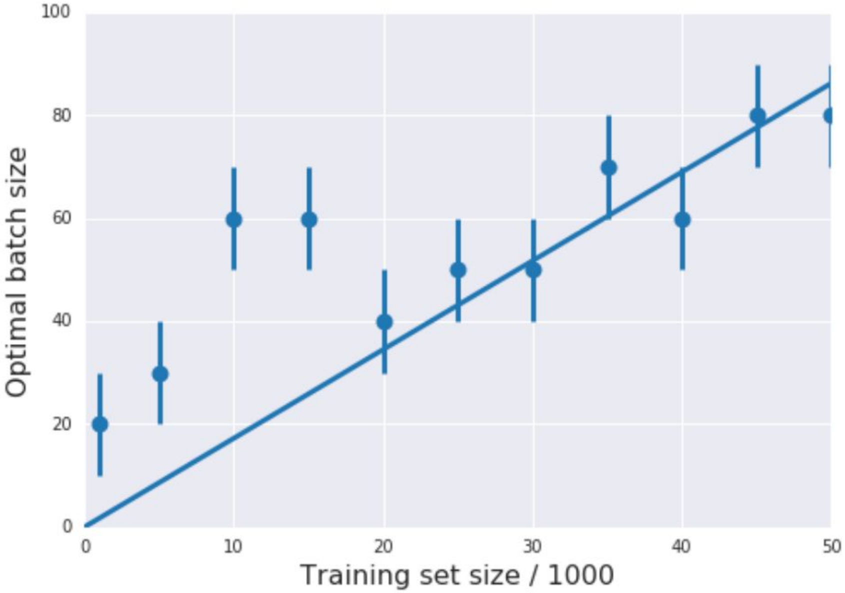
$$B_{opt} \propto \epsilon N$$

$$B_{opt} \propto \epsilon$$



$$B_{opt} \propto N$$





Consequences

1) We can linearly scale batch size and learning rate
   - "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour", Goyal et al. (2017)

2) We expect training sets to grow over time
   - Suggests batch sizes will rise

What about momentum?

What about momentum?

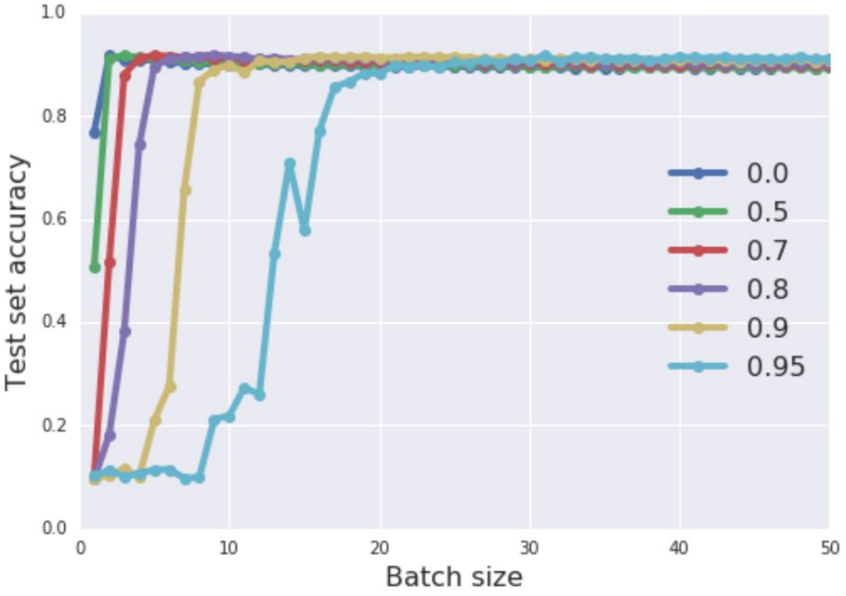
$$g \approx \frac{\epsilon N}{B(1-m)}$$

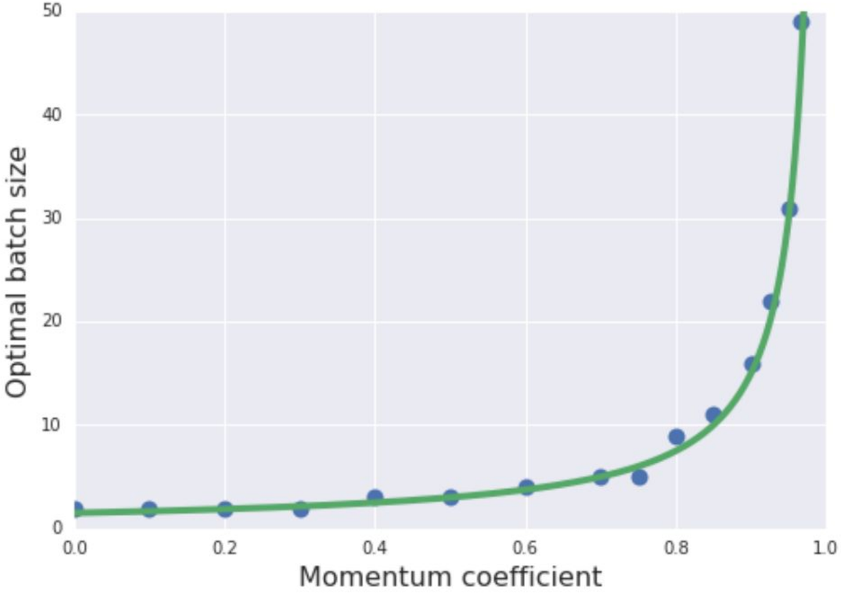
What about momentum?

$$g \approx \frac{\epsilon N}{B(1-m)}$$

$$B_{opt} \propto 1/(1-m)$$

$B_{opt} \propto 1/(1-m)$





Decaying learning rate and increasing batch size are equivalent

Decaying learning rate and increasing batch size are equivalent

$$g \approx \frac{\epsilon N}{B(1-m)}$$

Decaying learning rate and increasing batch size are equivalent

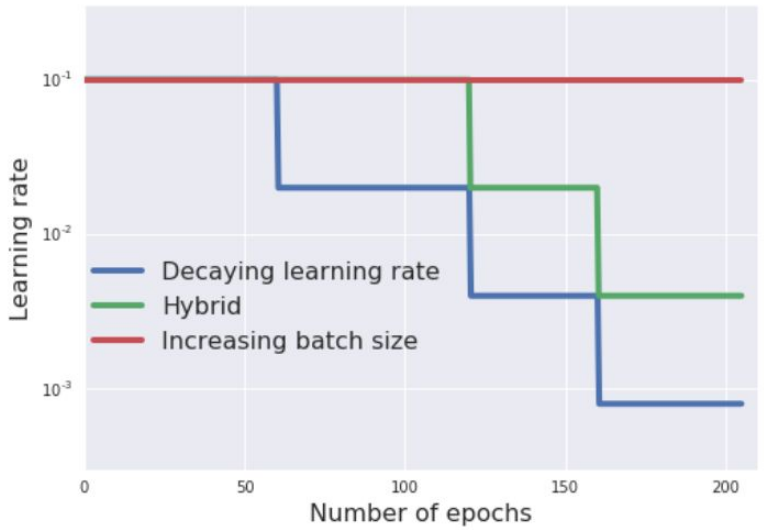
$$g \approx \frac{\epsilon N}{B(1-m)}$$

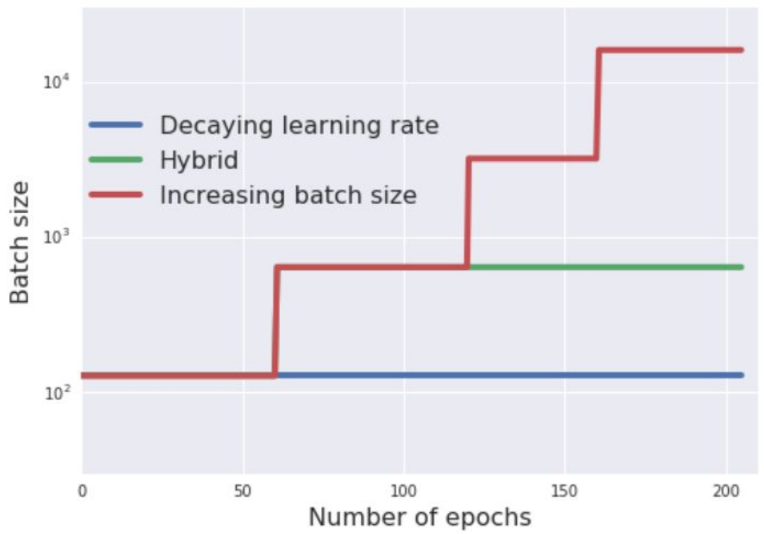
We can choose any combination of ε and B with the same g.

(so long as ε isn't too large)

Three equivalent schedules:

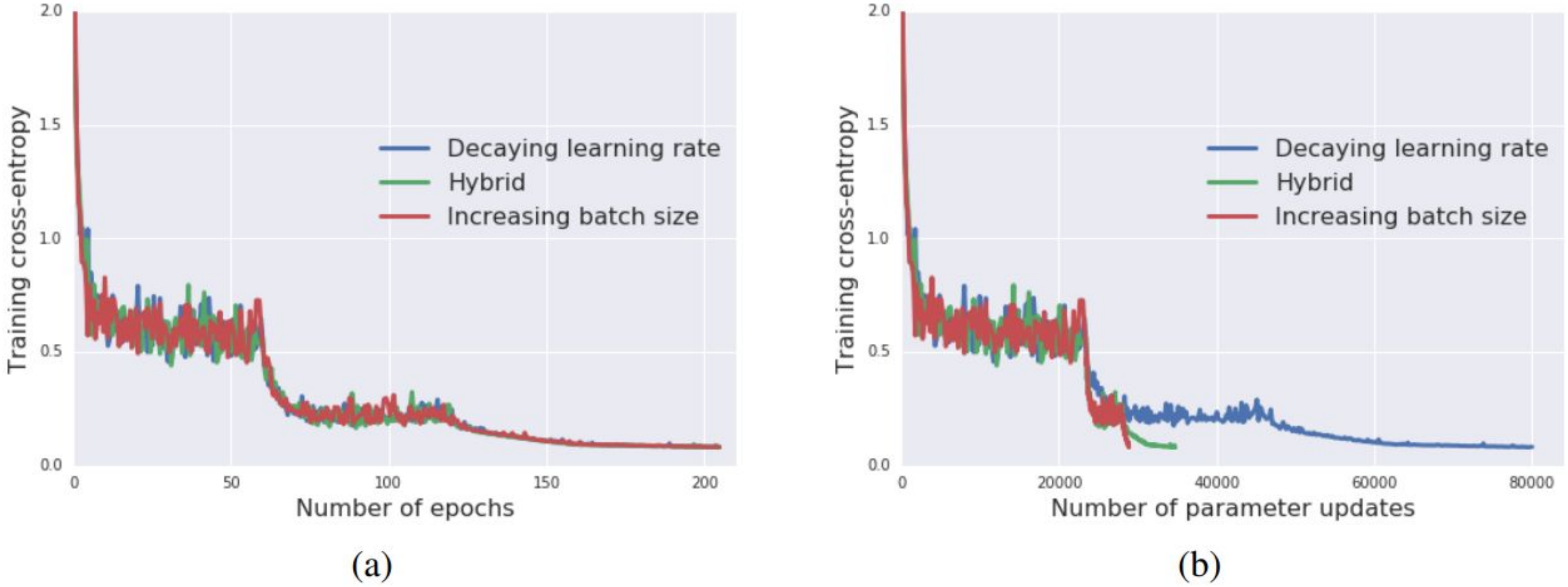
Wide ResNet on CIFAR-10





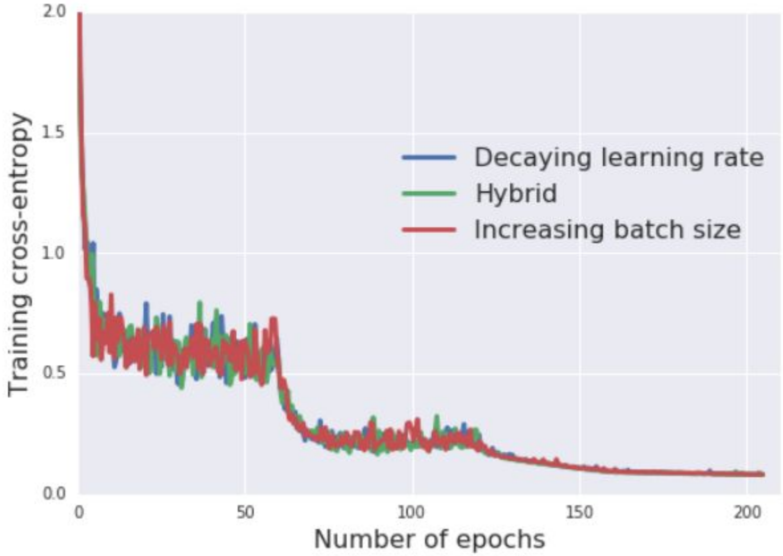
Training curves:

Ghost batch norm,
Hoffer et al., 2017

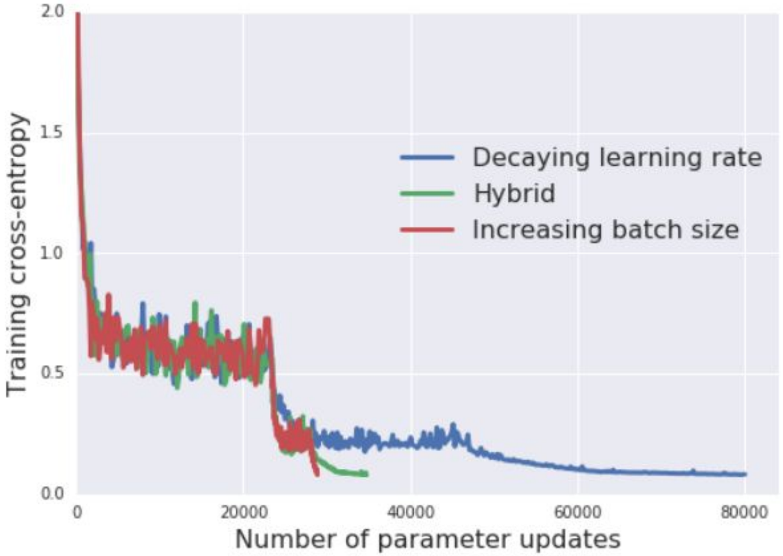


Training curves:

Computational cost constant



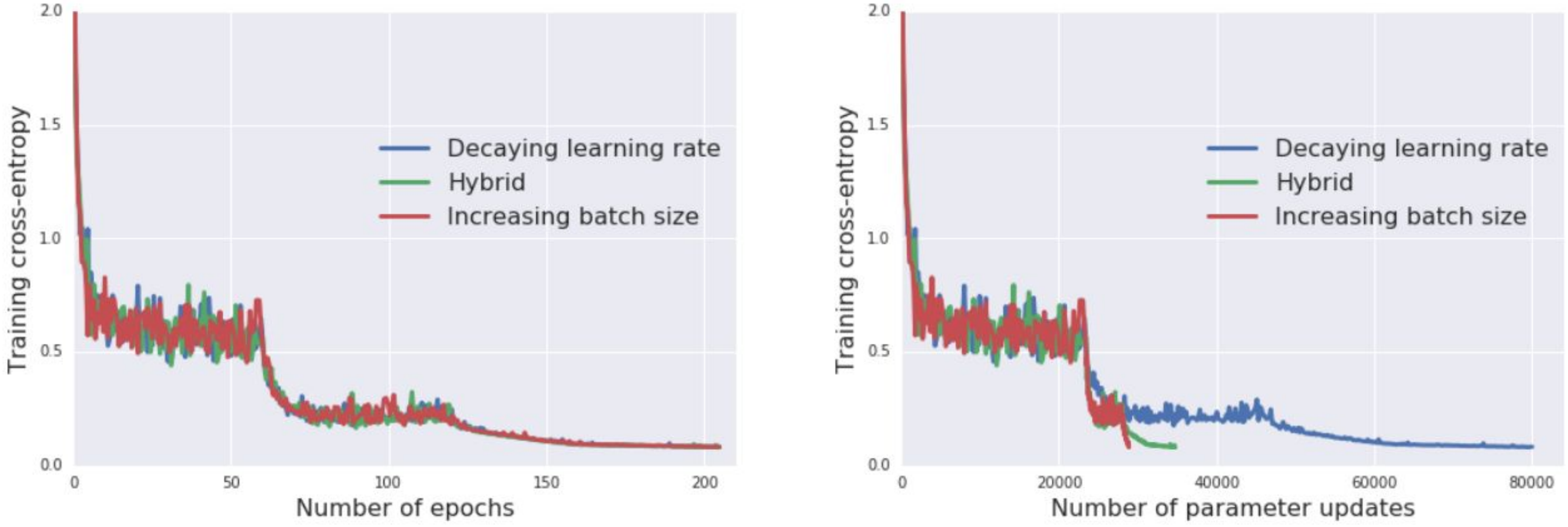
(a)



(b)

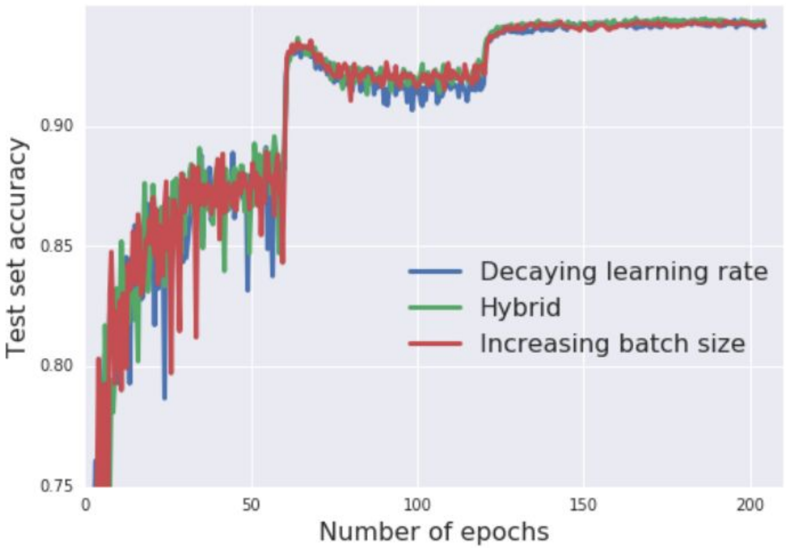
Training curves:

Computational cost constant
But parallelizable

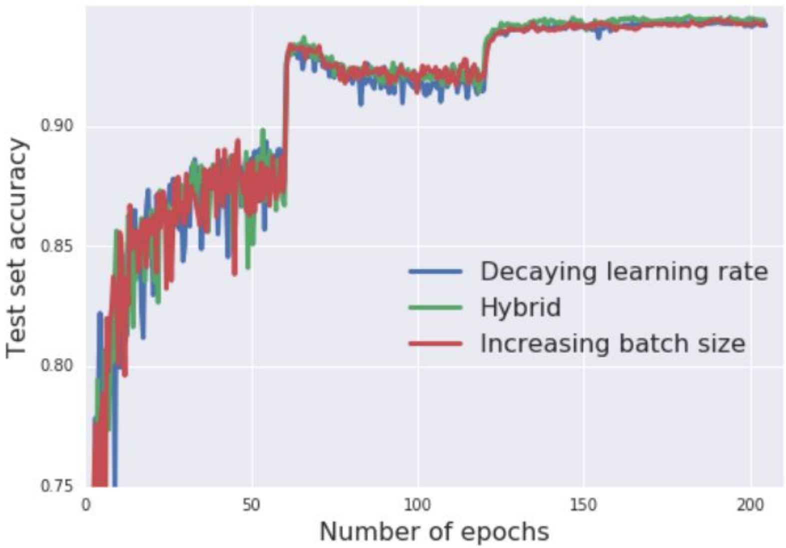


(a) (b)

Test curves:

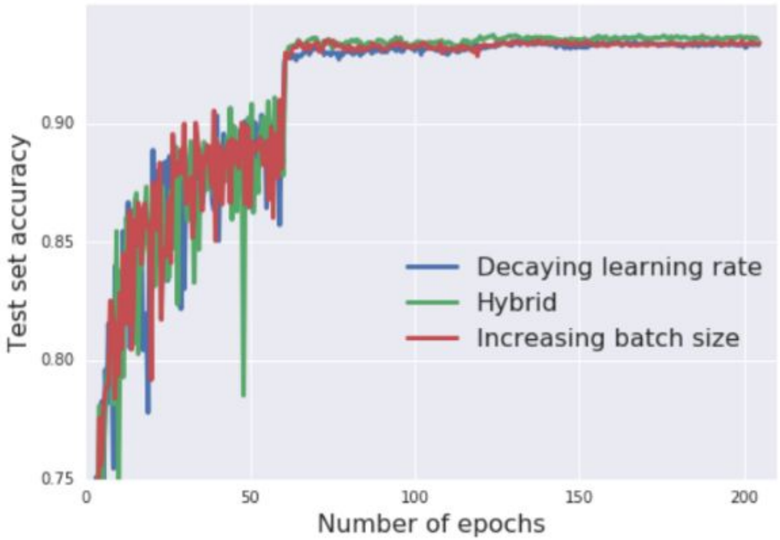


Momentum

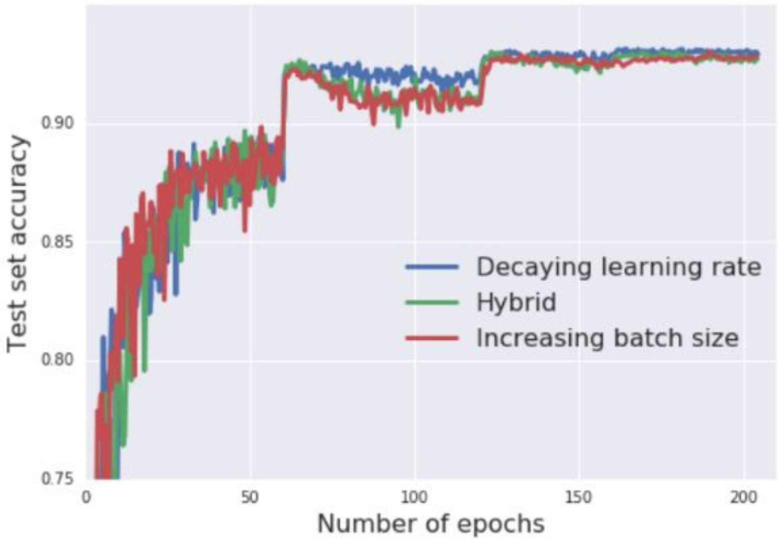


Nesterov momentum

Test curves:

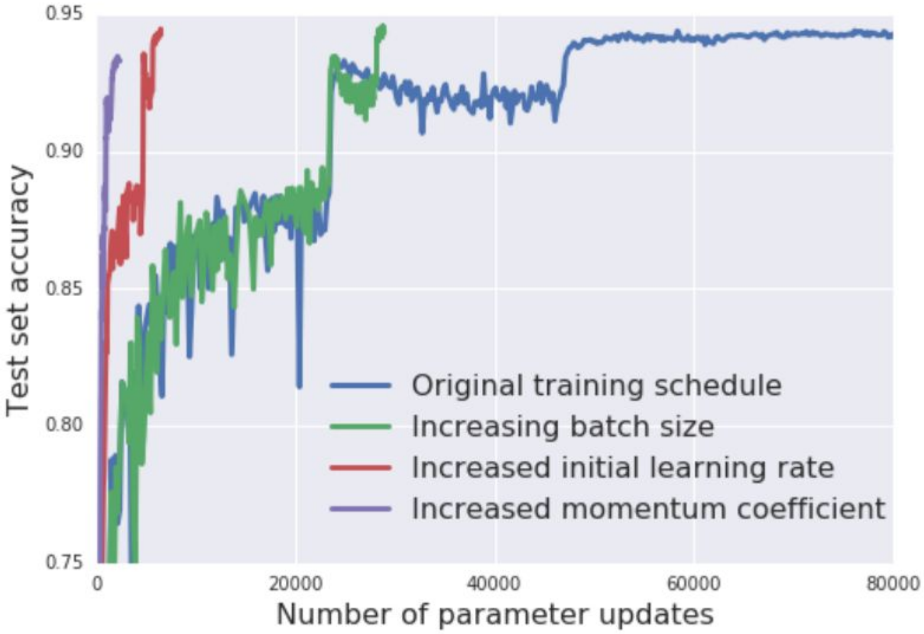


Vanilla SGD

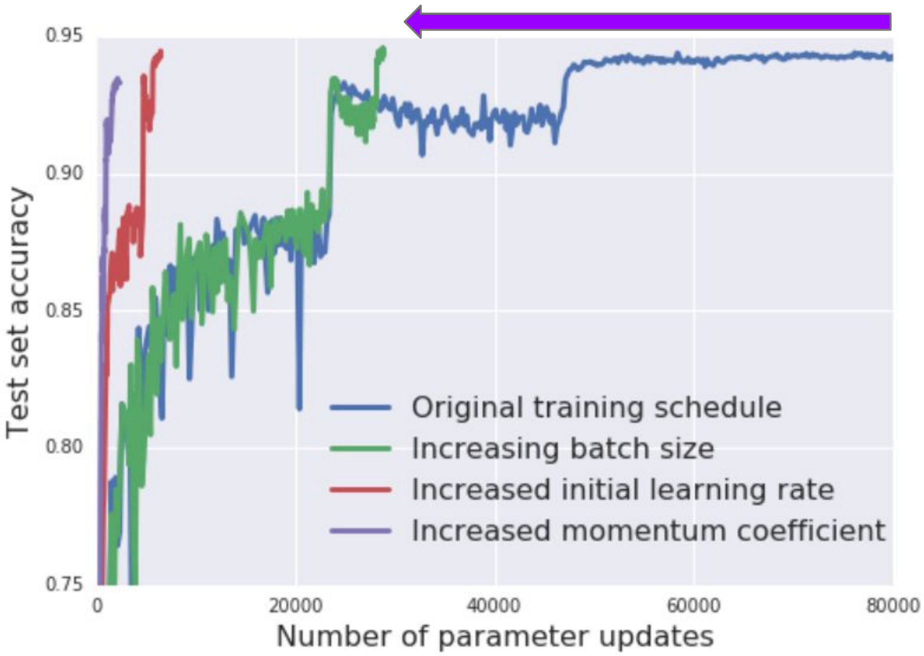


Adam

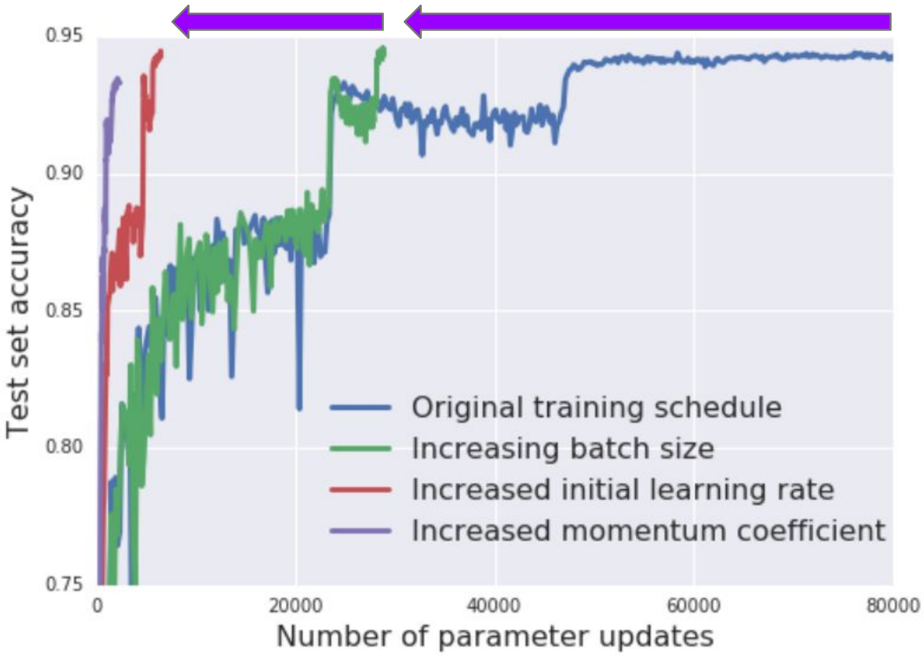
Towards large batch training:



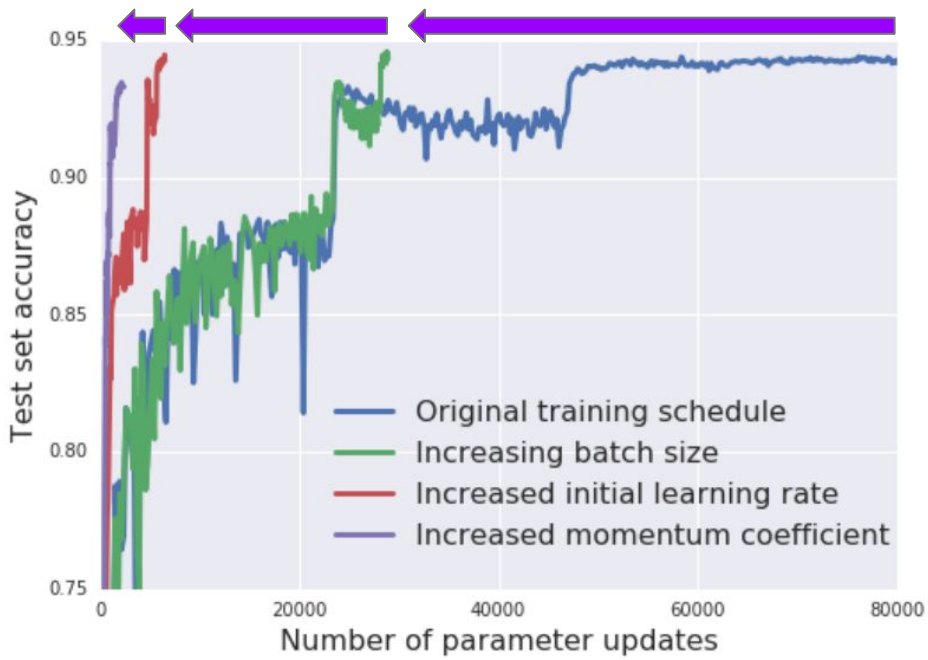
Towards large batch training:



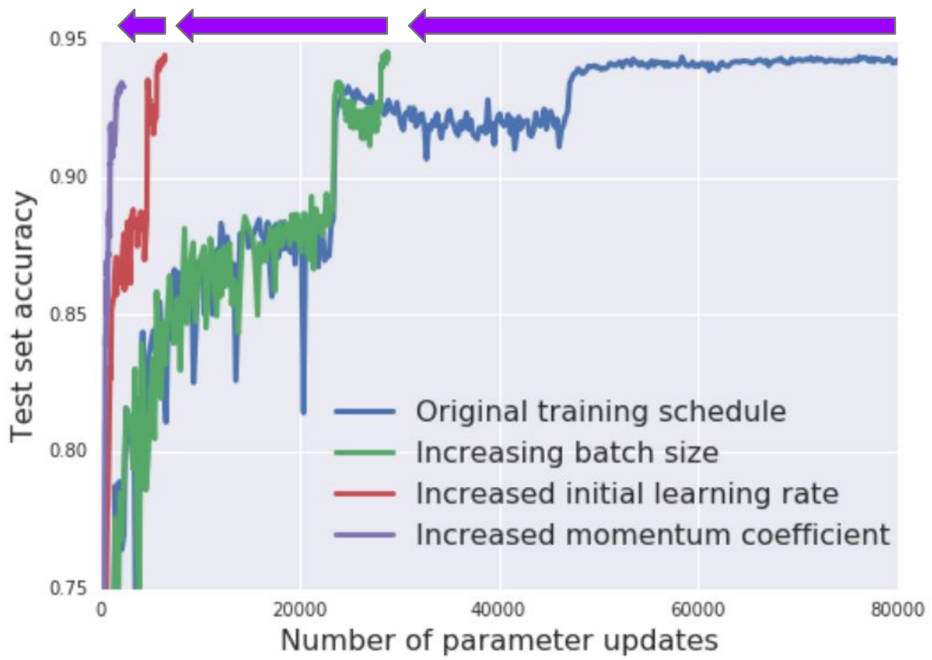
Towards large batch training:



Towards large batch training:



Towards large batch training:



Typical speed-up <u>10-100X</u>

Why does momentum scaling reduce test accuracy?

$$\begin{aligned} \Delta A &= -(1-m)A + \frac{d\hat{C}}{d\omega}, \\ \Delta\omega &= A\epsilon. \end{aligned}$$

Why does momentum scaling reduce test accuracy?

"Accumulation" stores moving average of gradients

$$\Delta A = -(1-m)A + \frac{d\hat{C}}{d\omega},$$
$$\Delta \omega = A\epsilon.$$

Why does momentum scaling reduce test accuracy?

Larger momentum equals longer memory

$$\Delta A = -(1-m)A + \frac{d\hat{C}}{d\omega},$$

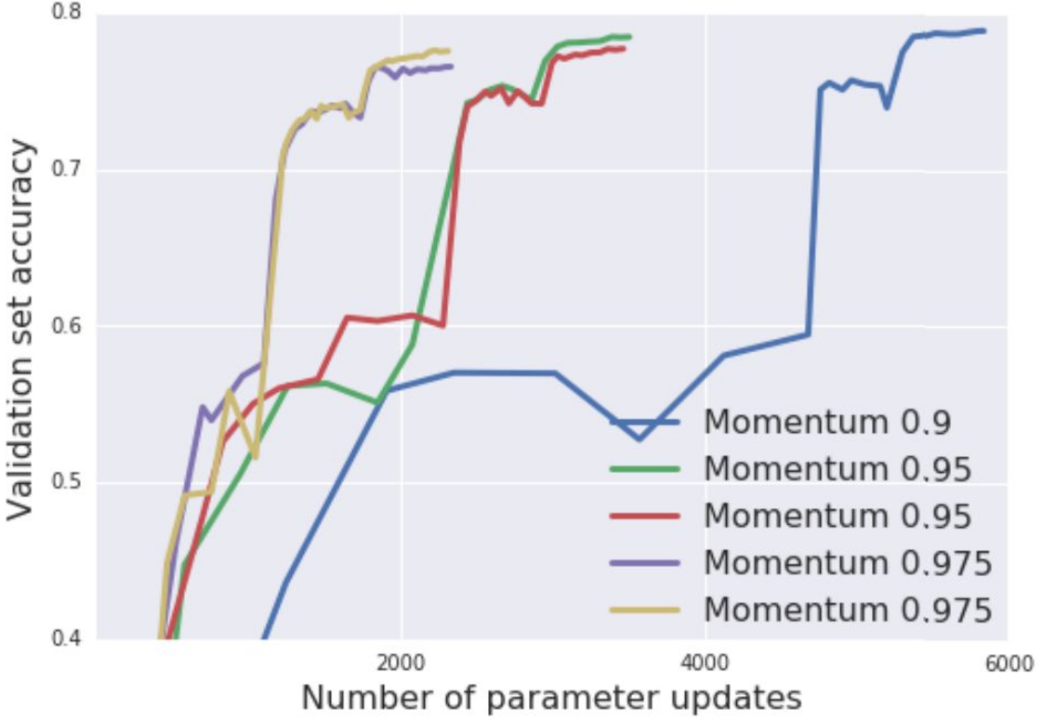
$$\Delta \omega = A\epsilon.$$

Why does momentum scaling reduce test accuracy?

Larger momentum equals longer memory

The gradient changes too slowly as we explore the parameter space

Training ImageNet in under 2500 updates!

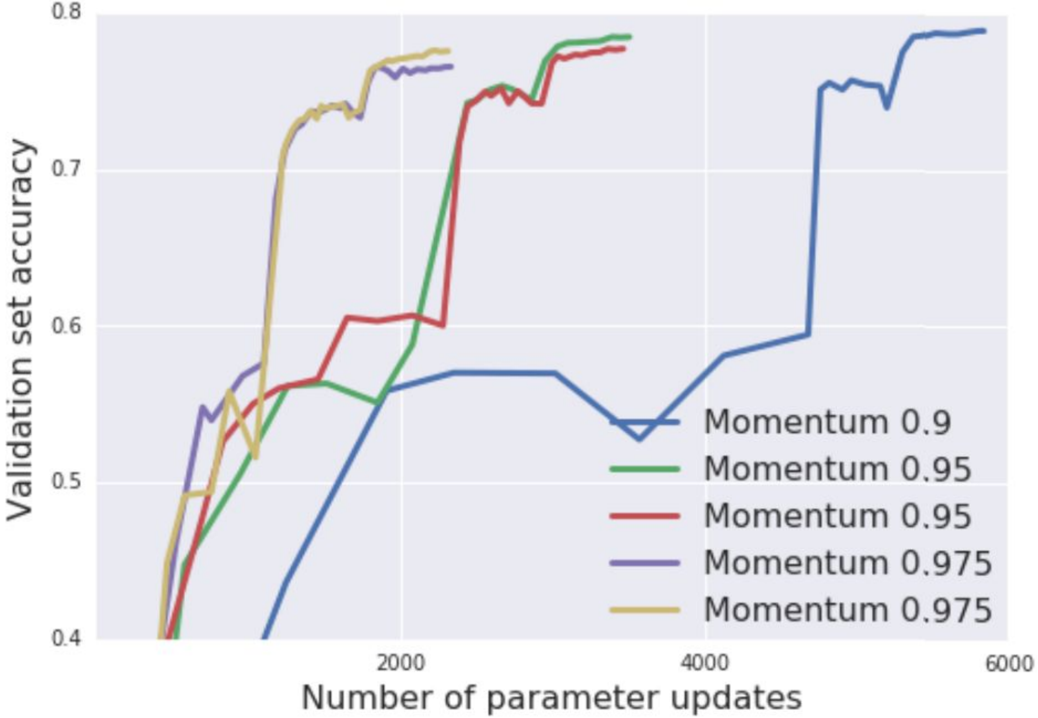


Inception-Resnet-V2

Original implementation:
~ 400,000 updates

"ImageNet in one hour"
Goyal et al., 2017
(learning rate scaling)
~ 14,000 updates

Training ImageNet in under 2500 updates!



79% accuracy in under 6000 updates

77% accuracy in under 2500 updates

Batches of 65,536 images

Thank You!

- “A Bayesian Perspective on Generalization and Stochastic Gradient Descent”, arXiv:1710.06451
  Samuel L Smith and Quoc V. Le

- “Don’t Decay the Learning Rate, Increase the Batch Size”, arXiv:1711.00489
  Samuel L Smith*, Pieter-Jan Kindermans* and Quoc V. Le
  *Equal contribution

- “Stochastic Gradient Descent as Approximate Bayesian Inference”, arXiv:1704.04289
  Stephan Mandt, Matthew D. Hoffman and David M. Blei

slsmith@
pikinder@
qvl@