NO MORE BATCH NORMALIZATION?

High-Performance Large-Scale Image Recognition Without Normalization

Andrew Brock [1] **Soham De** [1] **Samuel L. Smith** [1] **Karen Simonyan** [1]

Abstract

Batch normalization is a key component of most image classification models, but it has many undesirable properties stemming from its dependence on the batch size and interactions between examples. Although recent work has succeeded in training deep ResNets without normalization layers, these models do not match the test accuracies of the best batch-normalized networks, and are often unstable for large learning rates or strong data augmentations. In this work, we develop an adaptive gradient clipping technique which overcomes these instabilities, and design a significantly improved class of Normalizer-Free ResNets. Our smaller models match the test accuracy of an EfficientNet-B7 on ImageNet while being up to 8.7× faster to train, and our largest models attain a new state-of-the-art top-1 accuracy of 86.5%. In addition, Normalizer-Free models attain significantly better performance than their batch-normalized counterparts when fine-tuning on ImageNet after large-scale pre-training on a dataset of 300 million labeled images, with our best models obtaining an accuracy of 89.2%.[2]

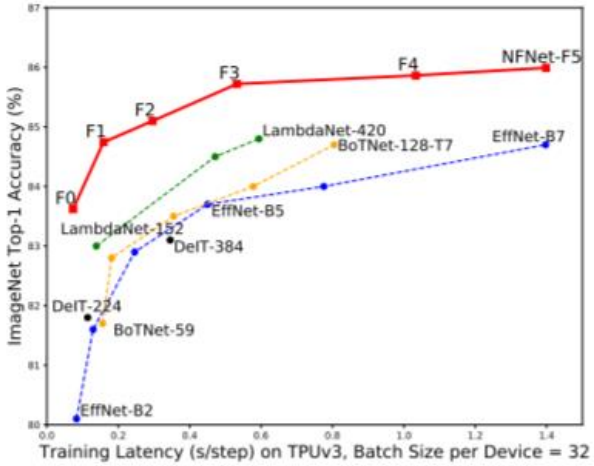


Figure 1. **ImageNet Validation Accuracy vs Training Latency.** All numbers are single-model, single crop. Our NFNet-F1 model achieves comparable accuracy to an EffNet-B7 while being 8.7× faster to train. Our NFNet-F5 model has similar training latency to EffNet-B7, but achieves a state-of-the-art 86.0% top-1 accuracy on ImageNet. We further improve on this using Sharpness Aware Minimization (Foret et al., 2021) to achieve 86.5% top-1 accuracy.

However, batch normalization has three significant practical

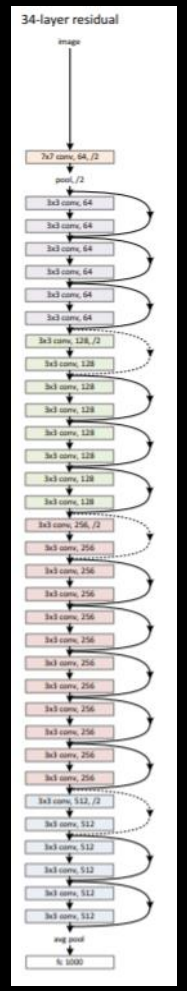
https://arxiv.org/pdf/2101.08692.pdf

https://arxiv.org/pdf/2102.06171.pdf

IMAGE RECOGNITION

Image taken from Google images, not clear who is Copyright owner.

SOTA MODELS EXPLOIT BATCH NORMALIZATION



Deep Residual Network

Image taken from: https://arxiv.org/pdf/1512.03385.pdf

EfficientNet

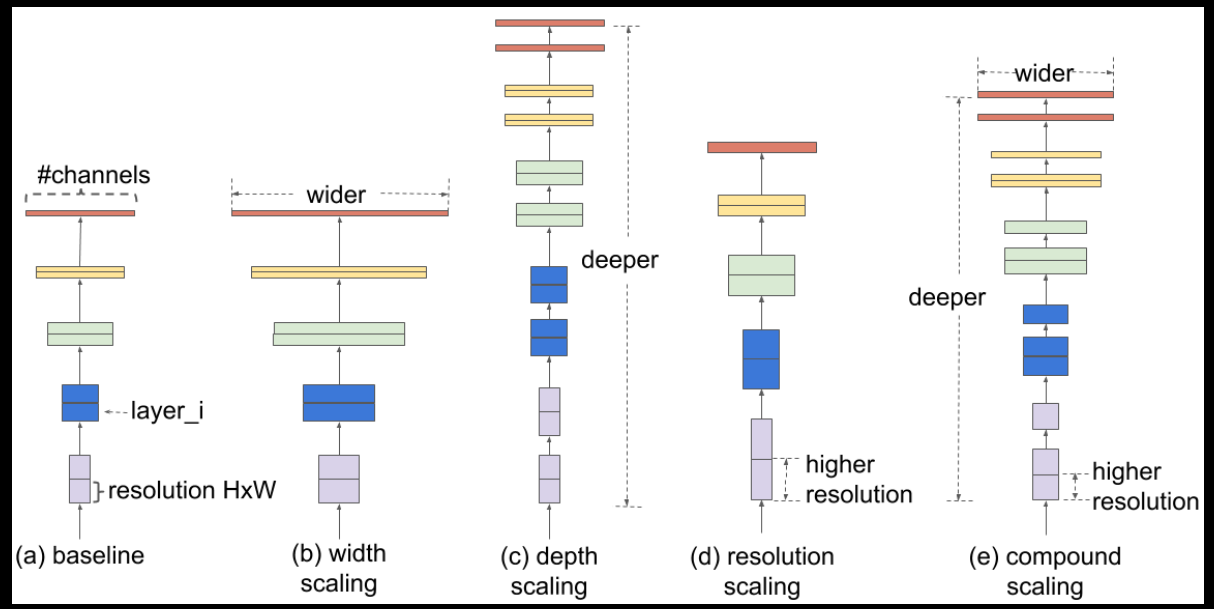


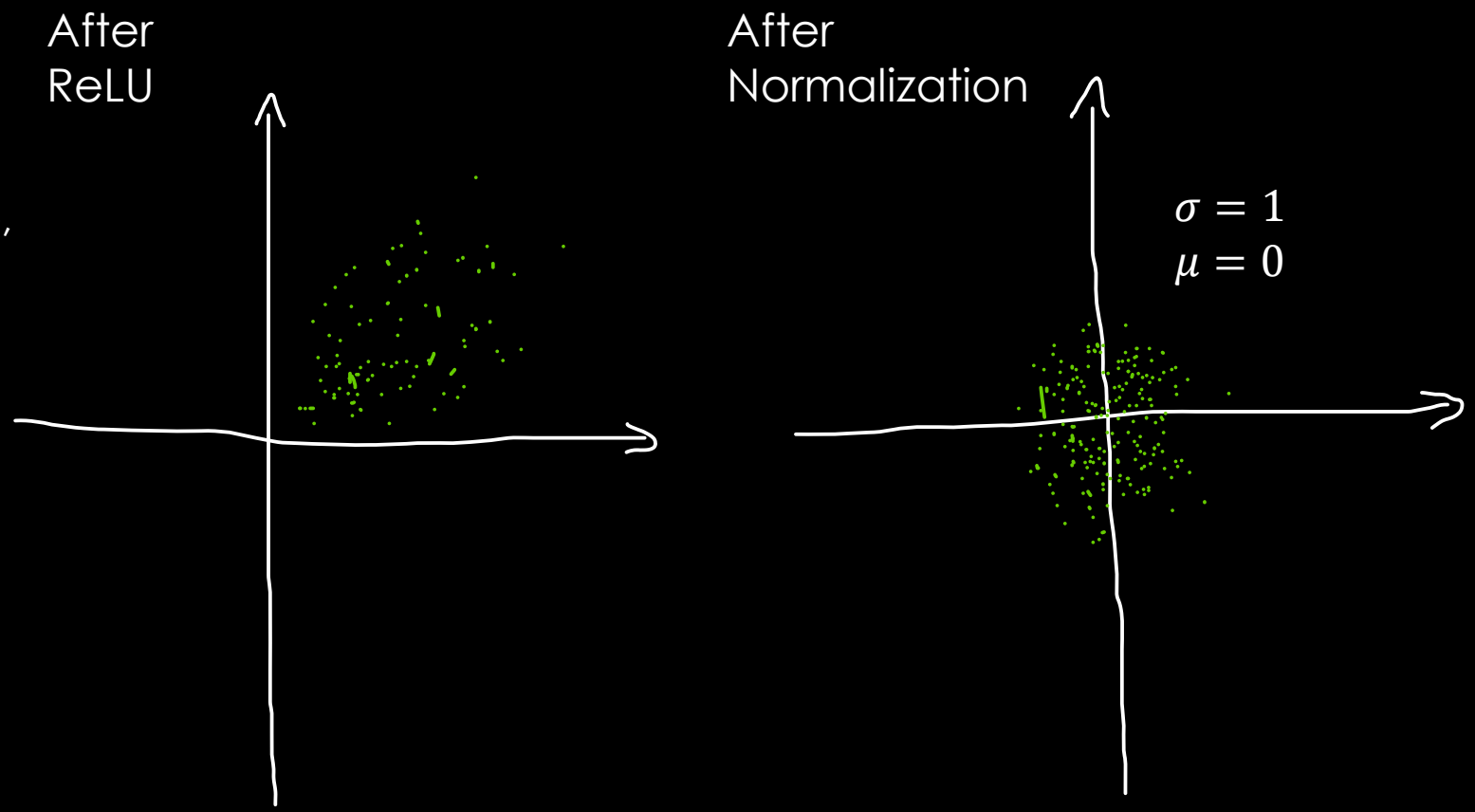
Image taken from: https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

PERCHE BATCH NORMALIZATION

- Durante allenamento di reti neurali profonde, ad ogni step la rappresentazione della distribuzione dell'input tende a non rappresentare piu la reale distribuzione dei nostri dati in ingresso.

- Batch Normalization risolve il problema centrando e ri-scalando la mini-batch in esame.

- Risoluzione Vanishing ed Exploding Gradient, impedendo di avere valori troppo grandi o troppo bassi durante discesa al gradiente.



BATCH NORMALIZATION

Data una mini-batch B, applichiamo Batch Normalization al layer come segue:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - \mu_\beta\right)^2$$

Normalizzazione:

$$\hat{x}_i = \frac{x_i - u_B}{\sqrt{\sigma^2 + \epsilon}}$$

La capacita rappresentativa della rete pero rischia di essere compromessa. Ci serve che alcune attivazioni abbiano comunque valori alti per manetenere predizioni accurate. Per questo introduciamo I learnable parameters γ e β.

$$y_i = \gamma\hat{x}_i + \beta$$

y e scalato per fittare con le caratteristiche del nostro dataset e per mantenere capacita rappresentativa della rete.

SVANTAGGI BATCH NORMALIZATION

- **Computazionalmente e un'operazione costosa**, puo causare overhead di memoria. Calcolare medie e scaling e tenerli In memoria per backpropagation.

- **Discrepanze tra il comportamento del modello nella fase di allenamento ed inferenza**. Un "buffer" contenente le statistiche calcolate durante il training e usato in inferenza. (introdotti anche hyper-params addizionali).

- **Per calcolare il gradiente in alcuni modelli il tempo richiesto aumenta in maniera considerevole**.

- **Batch Normalization tende a "imbrogliare" alcune funzioni di loss**. Intra-batch information leakage puo avvenire sia per alcune metodologie di Contrastive Learning che per modellazione di sequenze temporali, dove c'e il rischio di fare overlap target e training samples.

SVANTAGGI BATCH NORMALIZATION

- **Rompe indipendenza tra I training examples nel mini-batch normalizzato**:

1) Batch size matters! Fondamentale avere batch molto grandi perche ci permette di approssimare la vera media del dataset utilizzato nella fase di allenamento. Un batch piccolo, porta ad una cattiva approssimazione.

2) Distributed Training: diventa problematico perche non solo lo stato del gradiente deve essere comunicato ad ogni unita computazionale, ma anche le statistiche u, σ^2 calcolate da ogni operatore Batch Normalization.

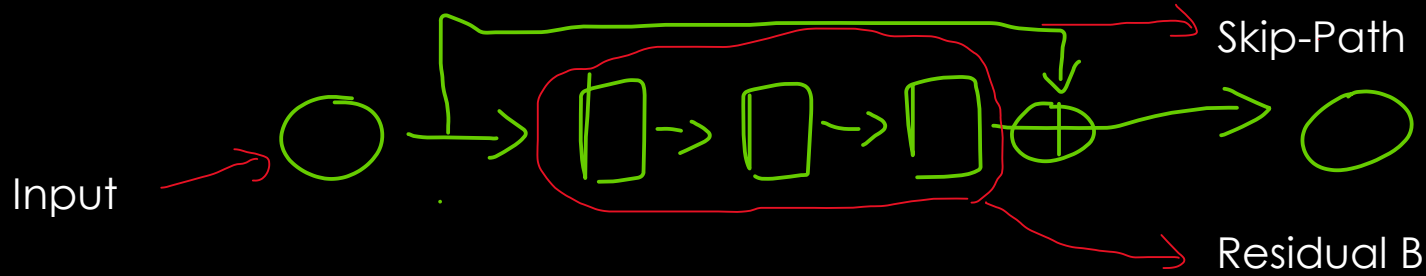
VANTAGGI BATCH NORMALIZATION

- **Batch Normalization elimina "mean shift".** Specialmente per non-linearita quali ReLU, assicura che mean activation in ogni canale sia zero sul batch in esame, ricentrando il tutto.

- **Batch normalization garantisce regolarizzazione**, aumentando accuratezza sul test set. Le statistiche calcolate sono basate su un sample piccolo di dati (la mini-batch in esame), per cui questo rende i valori "rumorosi". Questa aggiunta di rumore causa regolarizzazione.

- **Batch Normalization permette efficienti allenamenti con large-batch**. Rende possibile utilizzare valori alti per learning rate durante allenamento e garantisce una loss landscape molto regolare.

- **Batch Normalization ridimensiona i residual branches:** Batch Normalization implicitamente favorisce lo skip-path facendo downscaling delle attivazioni dei residual branches. Allenamento e cosi piu stabile (stesso obiettivo di ResNet il favorire Skip-Path).



COME CREARE RETI NEURALI LIBERI DA ELEMENTI DI NORMALIZZAZIONE?

Andrew Brock et al, from Google DeepMind propongono:

- Adaptive Gradient Clipping (AGC)

- Nuova famiglia di Normalizer-Free ResNets chiamate NFNets. Stessa accuratezza di EfficientNet ma 8.7x piu veloce nella fase di allenamento.

- Accuratezza nella fase di validazione superiore a reti con batch normalization quando si fa fine-tuning su ImageNet dopo aver pre-allenato su un dataset privato di 300 milioni di immagini annotate.

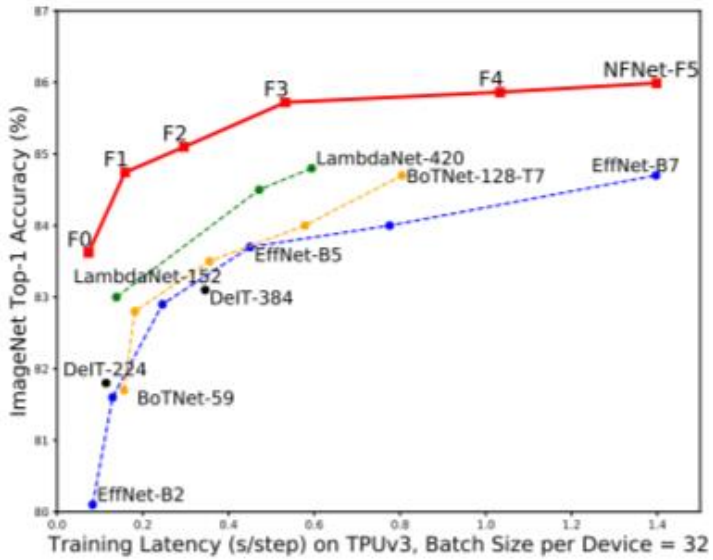


Figure 1. **ImageNet Validation Accuracy vs Training Latency.** All numbers are single-model, single crop. Our NFNet-F1 model achieves comparable accuracy to an EffNet-B7 while being 8.7× faster to train. Our NFNet-F5 model has similar training latency to EffNet-B7, but achieves a state-of-the-art 86.0% top-1 accuracy on ImageNet. We further improve on this using Sharpness Aware Minimization (Foret et al., 2021) to achieve 86.5% top-1 accuracy.

Image Taken from: https://arxiv.org/pdf/2102.06171.pdf

NORMALIZER-FREE RESNETS

Published as a conference paper at ICLR 2021

CHARACTERIZING SIGNAL PROPAGATION TO CLOSE THE PERFORMANCE GAP IN UNNORMALIZED RESNETS

Andrew Brock, Soham De & Samuel L. Smith
Deepmind
{ajbrock, sohamde, slsmith}@google.com

ABSTRACT

Batch Normalization is a key component in almost all state-of-the-art image classifiers, but it also introduces practical challenges: it breaks the independence between training examples within a batch, can incur compute and memory overhead, and often results in unexpected bugs. Building on recent theoretical analyses of deep ResNets at initialization, we propose a simple set of analysis tools to characterize signal propagation on the forward pass, and leverage these tools to design highly performant ResNets without activation normalization layers. Crucial to our success is an adapted version of the recently proposed Weight Standardization. Our analysis tools show how this technique preserves the signal in networks with ReLU or Swish activation functions by ensuring that the per-channel activation means do not grow with depth. Across a range of FLOP budgets, our networks attain performance competitive with the state-of-the-art EfficientNets on ImageNet.

https://arxiv.org/pdf/2101.08692.pdf

EfficientNet SOTA non e stata battuto.

Risultati di accuratezza simili a ResNet su ImageNet con batch size 1024 su test dataset

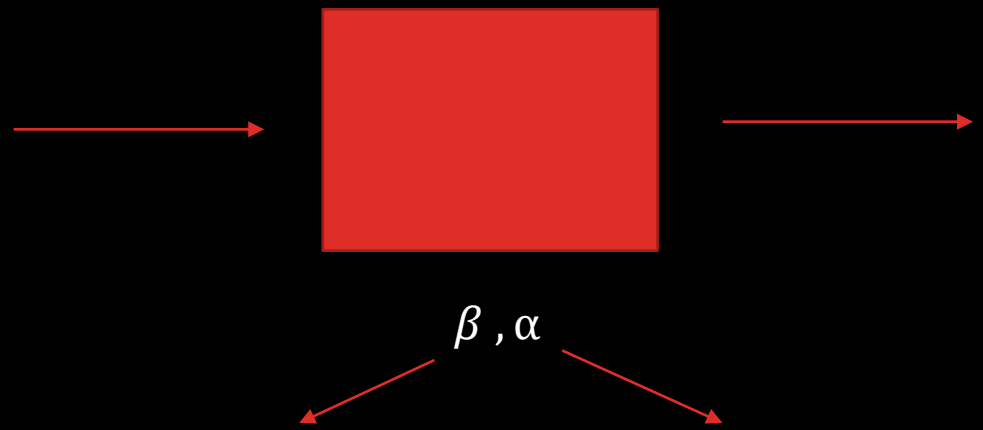
Supera reti batch normalized quando batch size e piccolo, ma e inferiore quando il batch size e grande (4096)

IT IS ALL ABOUT SCALING

Controlliamo cambiamento varianza dei dati per ogni blocco in modo che sia simile a quella della Batch Normalization. (Signal Propagation Plots (SPPs)).

Per ogni blocco consideriamo gli scalari β , α per controllare la varianza del segnale in modo tale che questa sia costante mentre si scende nella rete.

Altre tecniche sono utlizzate per mantenere la varianza, ma da sottolineare anche Scaled Weight Standardization per eliminare mean shift.



GRADIENT CLIPPING

Allenamento non sempre stabile, possiamo avere grossi sbalzi nella discesa al gradiente dovuti al nostro data batch (Puo creare problemi in Adam ecc..).

Gradient Clipping: se il gradiente e piu largo di una certa dimensione fissata lo clippiamo in modo da limitare impatto di questi sbalzi:

$$G \to \begin{cases} \lambda \frac{G}{\|G\|} & \text{if } \|G\| > \lambda, \\ G & \text{otherwise.} \end{cases}$$

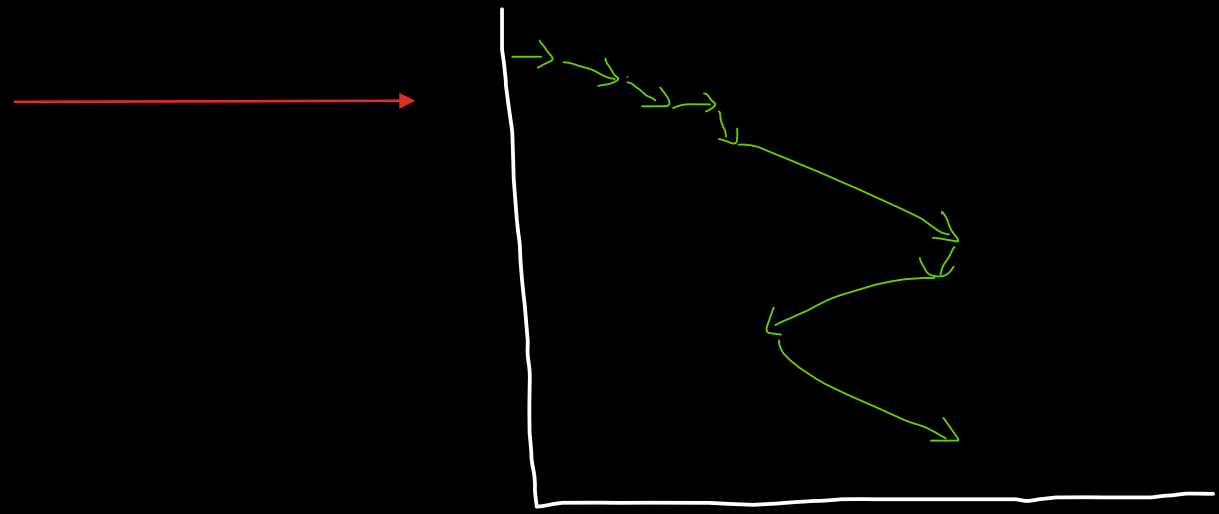


Image Taken from: https://arxiv.org/pdf/2102.06171.pdf

ADAPTIVE GRADIENT CLIPPING

Gradient Clipping e estremamente sensibile al parametro clipping threshold λ (non adaptive).

Adaptive Gradient Clipping: Il gradiente viene limitato dal rapporto tra quanto grande e la norma del gradiente vs quanto grande e la matrice dei pesi sulla quale il gradiente agisce. Matrice piccola, il gradient step dovra essere piccolo, ma se lo step e grande questo non va bene. Accettabile solo se la matrice dei pesi e gia grande.

$$G_i^\ell \rightarrow \begin{cases} \lambda \frac{\|W_i^\ell\|_F^\star}{\|G_i^\ell\|_F} G_i^\ell & \text{if } \frac{\|G_i^\ell\|_F}{\|W_i^\ell\|_F^\star} > \lambda, \\ G_i^\ell & \text{otherwise.} \end{cases}$$

Image Taken from: https://arxiv.org/pdf/2102.06171.pdf

ADAPTIVE GRADIENT CLIPPING RESULTS

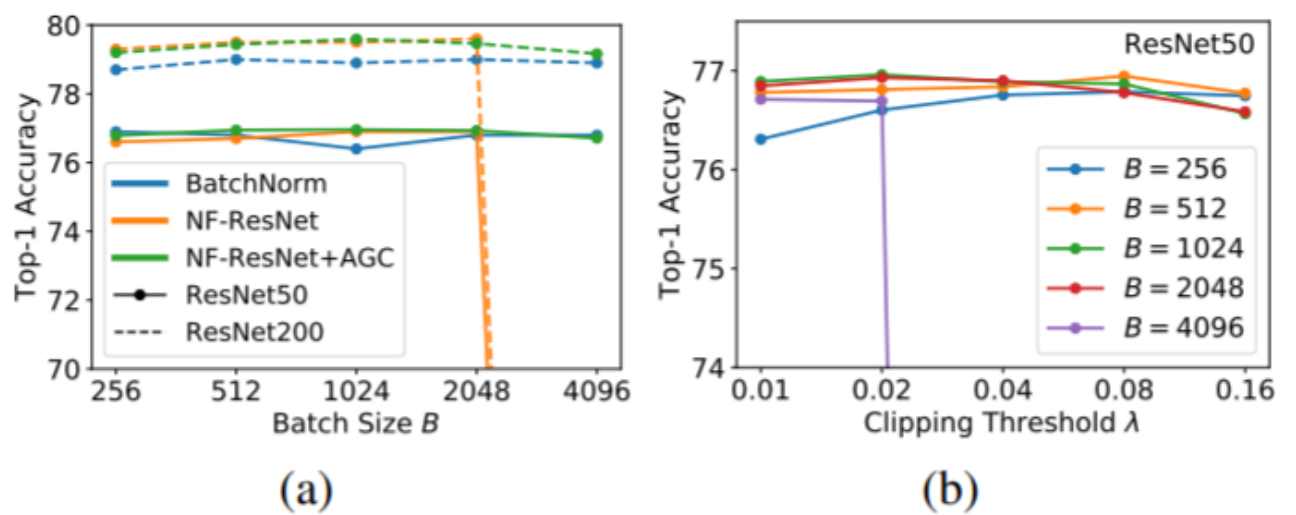


Figure 2. (a) AGC efficiently scales NF-ResNets to larger batch sizes. (b) The performance across different clipping thresholds λ.

Abbiamo ancora implicitamente dipendenza tra I data points nella batch , anche se in maniera molto fine.

Image Taken from: https://arxiv.org/pdf/2102.06171.pdf

YANNIC KILCHER: CONSIDERAZIONI

Il gradiente e composto dal batch di dati che ci facciamo passare attraverso.

Il gradiente e un media degli svariati steps che prenderebbero I singoli data points.

Considerando questi batch avremo comunque un po' di rumore dato che abbiamo una stima di tutto il dataset.

Naturalmente il rumore mediato su questi I.I.D samples diventera piu piccolo in relazione al segnale stesso. Piu grande la batch, minore il rumore.

Limitando solo il gradiente finale, e non sul singolo data point, rimani comunque dipendente dalla grandezza del batch e in qualche maniera interconneti I vari training samples.

Yannic Kilcher: https://www.youtube.com/watch?v=rNkHjZtH0RQ

NFNETS

Due nuovi blocchi per ResNets presentati.

Code:

PyTorch: https://github.com/rwightman/pytorch-image-models

JAX: https://github.com/deepmind/deepmind-research/tree/master/nfnets

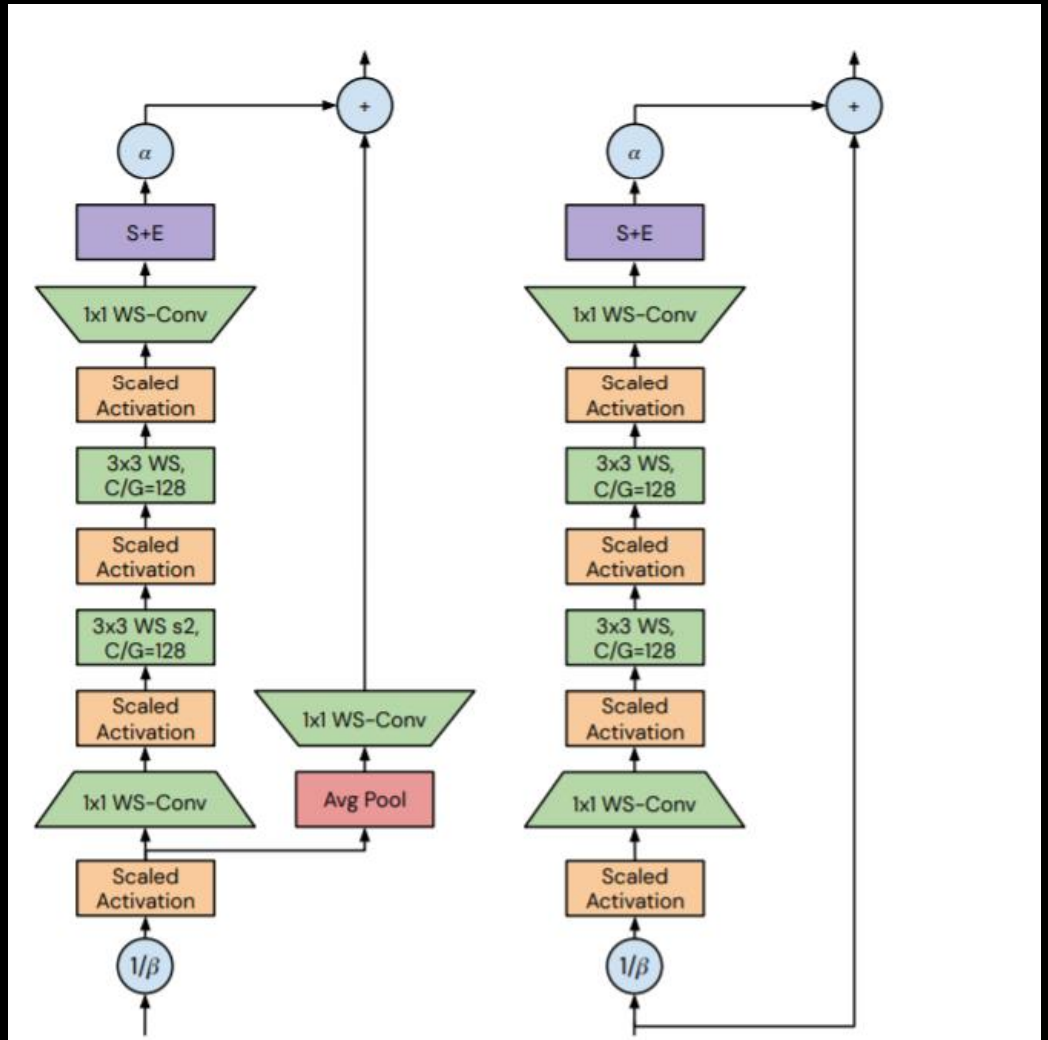


Figure 5. Detailed view of an NFNet transition block. The bottleneck ratio is 0.5, while the group width (the number of channels per group, C/G) in the 3×3 convolutions is fixed at 128 regardless of the number of channels. Note that in this block, the skip path takes in the signal after the variance downscaling with β and the scaled nonlinearity.

Figure 6. Detailed view of an NFNet non-transition block. The bottleneck ratio is 0.5, while the group width (the number of channels per group, C/G) in the 3×3 convolutions is fixed at 128 regardless of the number of channels. Note that in this block, the skip path takes in the signal before the variance downscaling with β.

Image Taken from: https://arxiv.org/pdf/2102.06171.pdf

ARE WE DONE WITH BATCH NORMALIZATION?

- A livello di prestazioni dipendiamo ancora dalla mini-batch size. Potenzialmente ancora presente dipendenza tra i training examples nel mini-batch normalizzato.

- A livello computazionale il problema sembra risolto, dato l'enorme speed-up ottenuto durante fase di allenamento.

- Anche tra inferenza e training, abbiamo tolto necessita di running mean ecc., ma ancora presenza di dropout nella rete, che ha un comportamento differente in training e test.

- Non chiarissimo se i miglioramenti sono proprio dovuti completamente alle tecniche che impediscono batch normalization o alla struttura di queste nuove architetture.

QUESTIONS?