



# Why BN is not used in Transformer

Beomgon Yu([beomgon.yu@gmail.com](mailto:beomgon.yu@gmail.com))



# Contents

1. introduction
2. Batchnorm
3. Transformer
4. Results

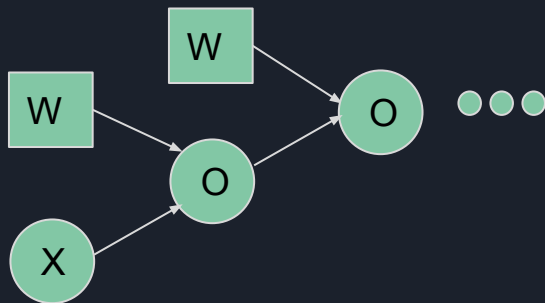


# Introduction

1. Vanishing or exploding gradient Problem
2. How to solve
  - sigmoid -> relu,
  - skip connection
  - careful parameter init
  - batchnorm
  - etc...

# Introduction

1. DNN은 input(x)과 weight(w)가 recursive하게 계속 곱해지는 구조  
W의 값이 1보다 크면 발산하고, 1보다 작으면 최종 output이 0으로 수렴하는 문제 발생
2. scale의 값을 1에 가깝게 하기 위한 다양한 initialization 방법들이 제안됨  
( matrix norm, principle eigen value to 1  
by circular theorem, need to select randomly by mean and variance constraints,  
variance is by inputs size, output size(for forward, backward stable)
3. Batcnorm은 output에 대해 normalization을 해주는 방식의 다른 대안을 제시함



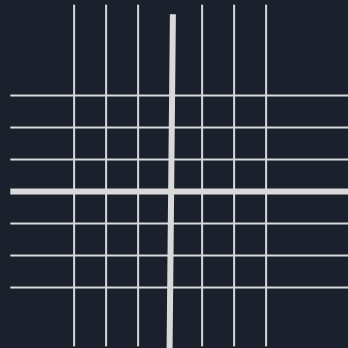
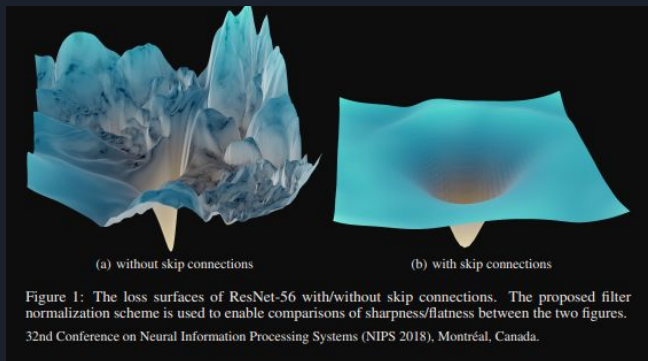


# Batchnorm

1. 1D, 2D, 3D  
channel 단위 normalization, re-scaling in image  
2\*dim (scale, bias)의 parameter,  
test시 moving average(mean, variance)  
Internal covariance shift -> (how does batchnorm help optimization)  
frozen batchnorm  
learning rate vs batch size  
set batch statistic on non linear point
2. Without Batchnorm(NFNet)  
poor in small batch size  
discrepancy between the behaviour of the model during training and at inference  
batch normalization breaks the independence during training

# Batchnorm

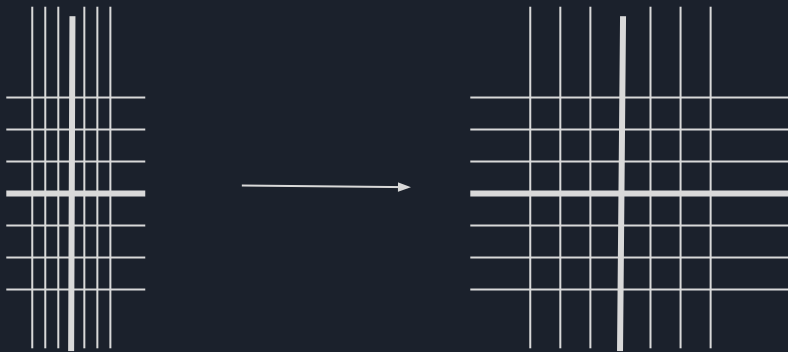
1. loss landscape visualization  
select two axis, make grid,  
get loss by all test set, visualize
2. it seemed easy but difficult, Why??





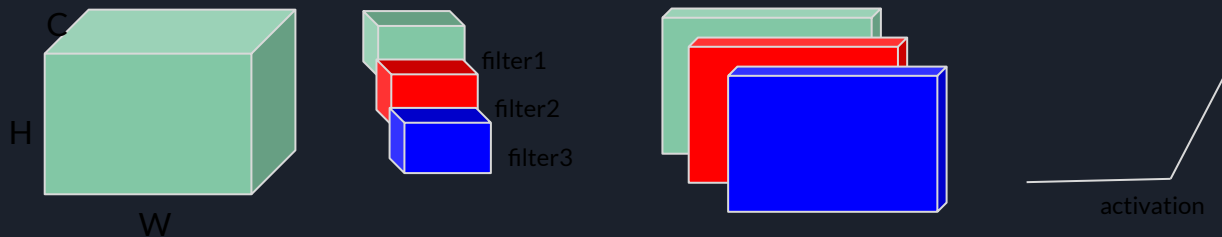
# Batchnorm

1. its not euclidean space, big scale eats small scale  
need to make scale to be equally  
which unit to scale -> filter wise normalization



# Batchnorm

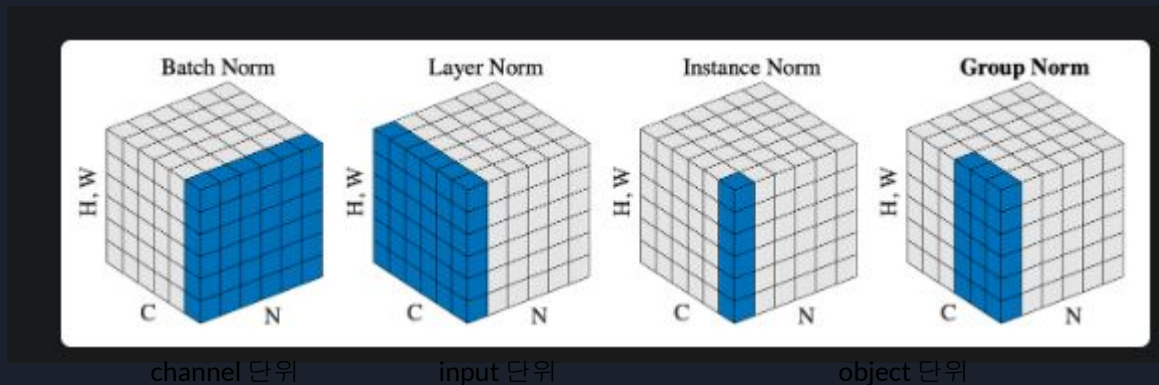
1. How about using filter wise normalization in training??  
it seemed to be difficult direct on weight, how about using on feature map?  
This is Batchnorm,  
normalize each filter, and gave scale  
(각 필터를 normalization 후 중요 필터에 가중치, bias를 적절히 주자)





# Batchnorm

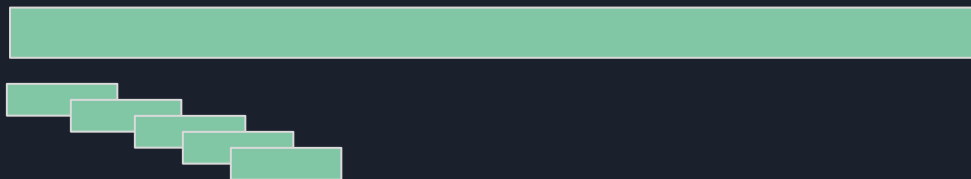
1. LayerNorm은 input image  
instance or group norm의 경우 object 단위의 normalization을 한다고 볼수 있을 듯,  
task에 맞게 적절한 normalization이 필요하다.





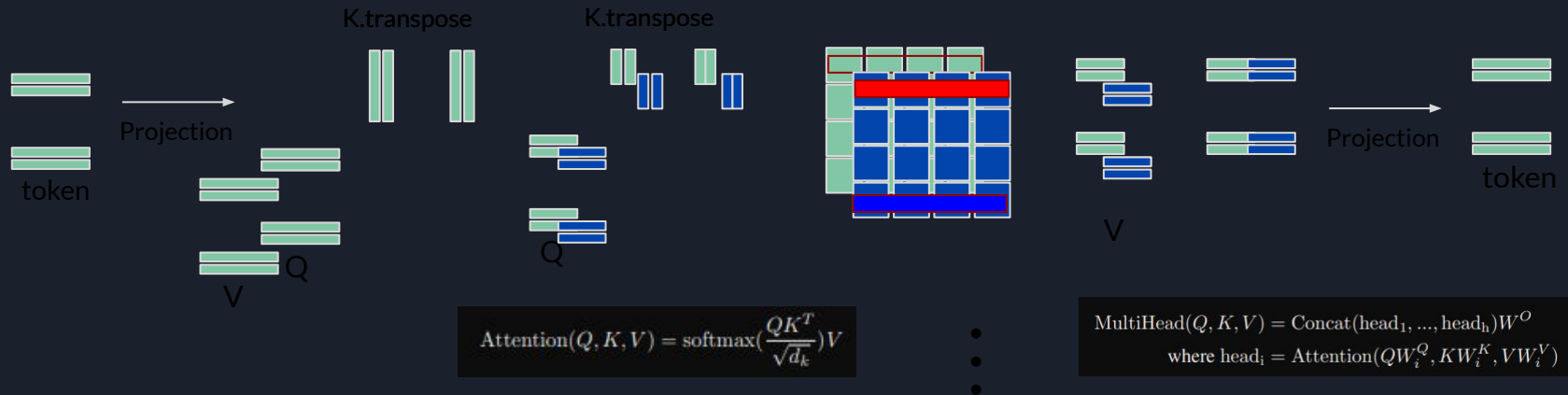
# Transformer

1. Why BatchNorm is not used in Transformer  
sequential한 data에서 leakage등이 발생할 수 있음  
(cheating)
2. But Lots of data, random sampling, is it really problem?
3. view of filter wise normalization



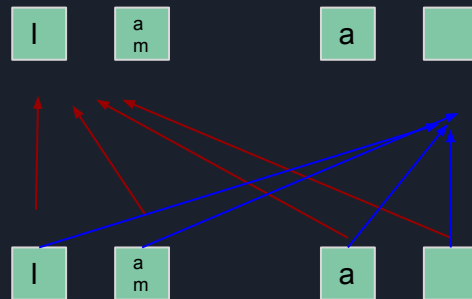
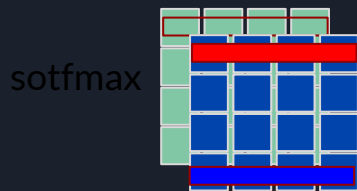
# Transformer

- attention
  - re-encoding, dynamic convolution, long range dependency  
soft filter, trainable projection matrix
  - complexity, weak inductive bias



# Transformer

1. weight is already normalized by softmax,  
it seem that batchnorm is not fit to transformer(layer norm is better)  
But how about value?



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

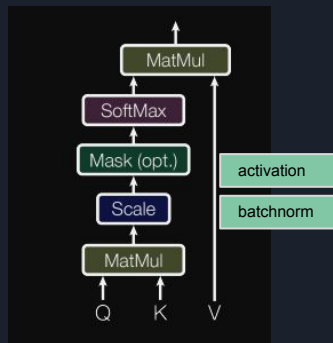
# Transformer

Linear projection is like a convolution of full window size

Batchnorm 1d per channel wise

give activation(기존 addition, GELU in mlp)

attention map을 학습하기 오랜 시간이 걸림, direct로 value쪽에서 의미 있는 feature 뽑아내 보자.





# Results

1. Imagenet 1k,  
SGD with momentum 0.9, weith decay  $1e-4$   
step LR(0.1 decay per 30 epoch)
2. training 내내 val acc가 2~3% 정도 항상 높은 경향을 보임
3. vit의 경우 imagenet 1k보다 큰 dataset으로 테스트 필요

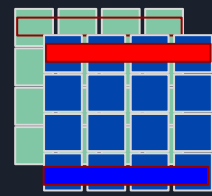
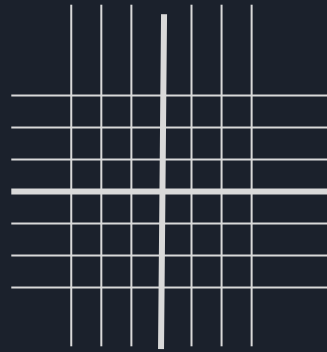
Previous	New
Acc@1 53.784 Acc@5 77.266	Acc@1 56.340 Acc@5 79.732



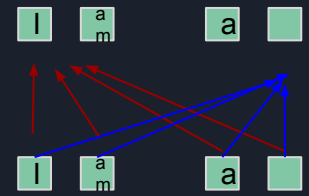
Thank you !!







V





- attention
  - re-encoding, dynamic convolution, long range dependency  
soft filter, trainable projection matrix
  - complexity, weak inductive bias