Как жить без batch-norm?

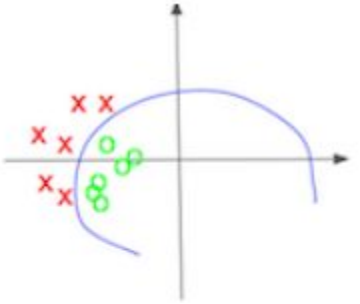
Агапова Ольга, 181 группа

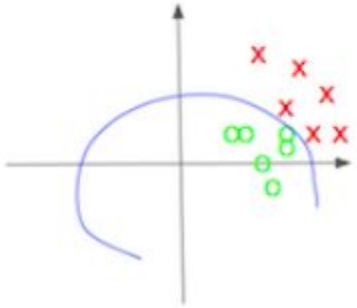
Как жить с batch-norm?

- Есть проблема: ковариантный сдвиг
- Есть решение: batch-norm
- По счастливому стечению обстоятельств, batch-norm решает еще массу проблем

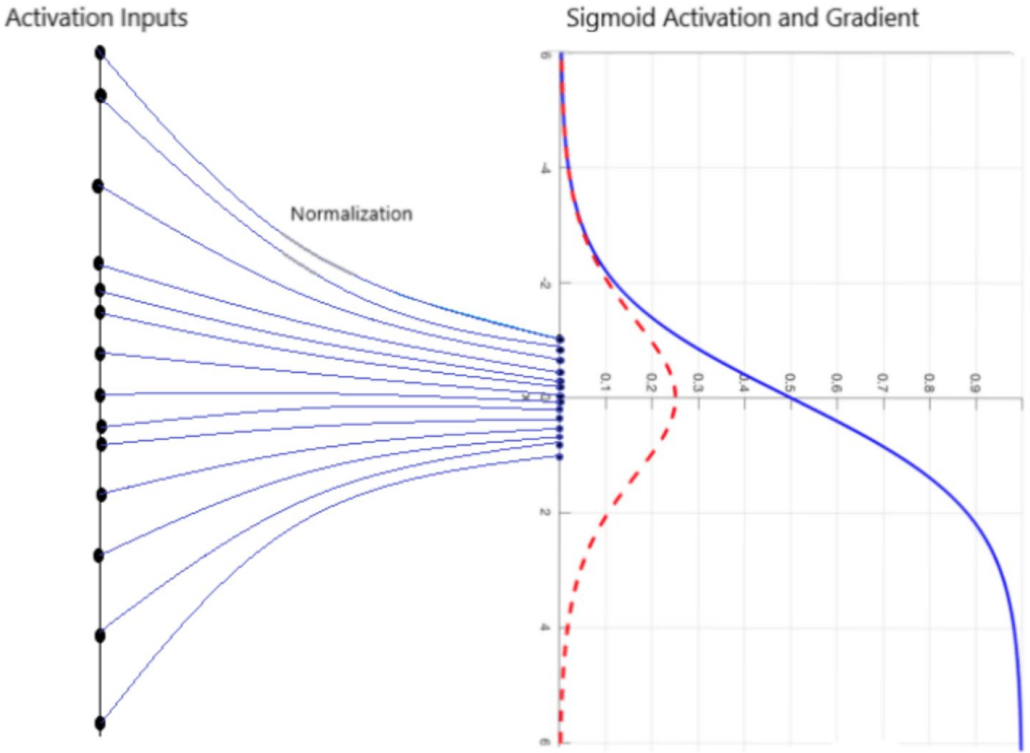
Ковариантный сдвиг

Ситуация, когда в обучающей и тестовой выборке разные параметры распределения значений признаков: разброс распространяется с каждым слоем





Решение: batch-normalization



Некоторым слоям нейронной сети на вход подаются данные, предварительно обработанные и имеющие нулевое мат. ожидание и единичную дисперсию

Вход: значения x из пакета $B = \{x_1, \dots, x_m\}$; настраиваемые параметры γ, β;
Выход: $\{y_i = BN_{\gamma,\beta}(x_i)\}$

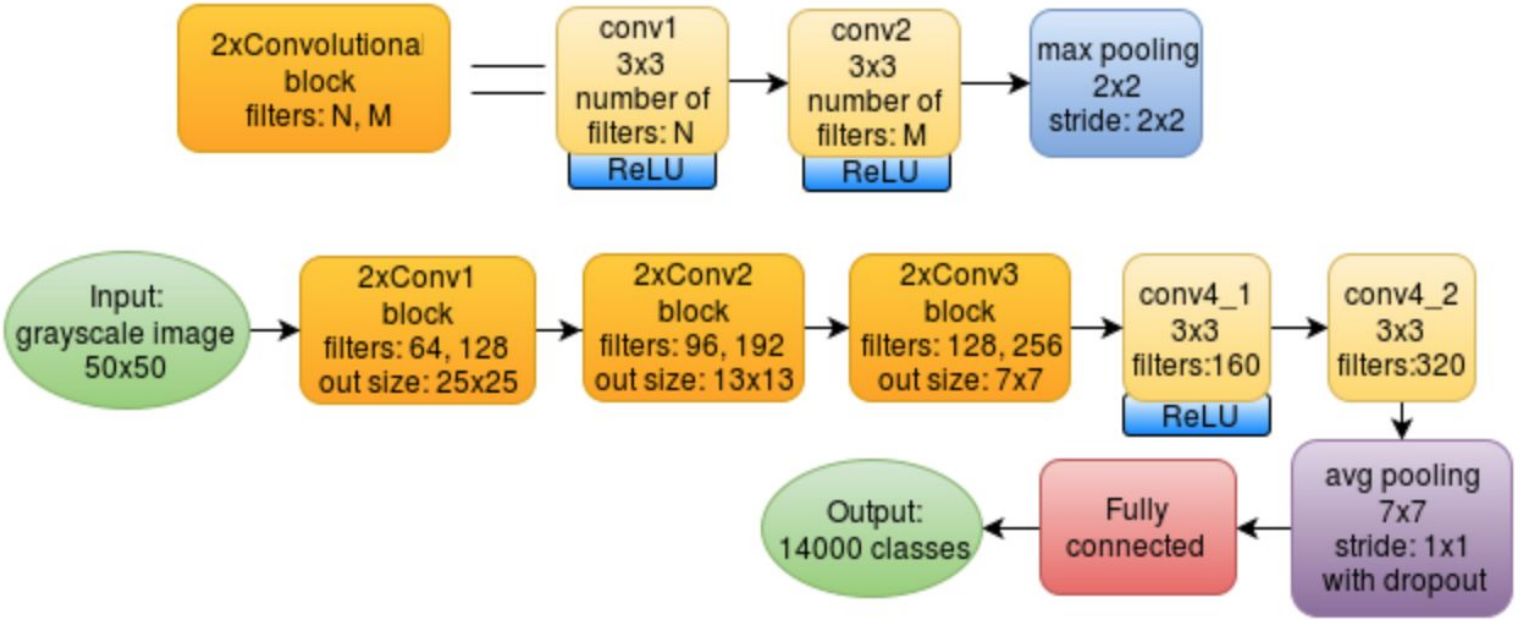
$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \quad \text{// математическое ожидание пакета}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \quad \text{// дисперсия пакета}$$

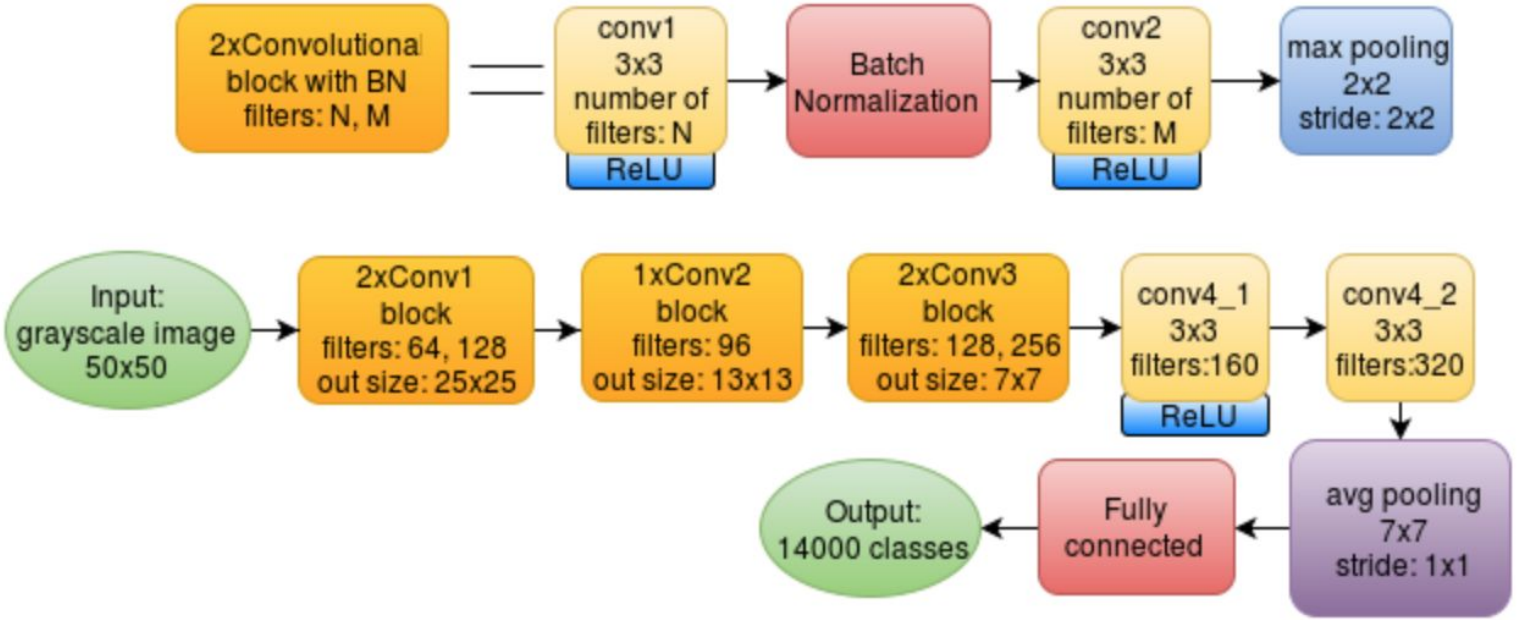
$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad \text{// нормализация}$$

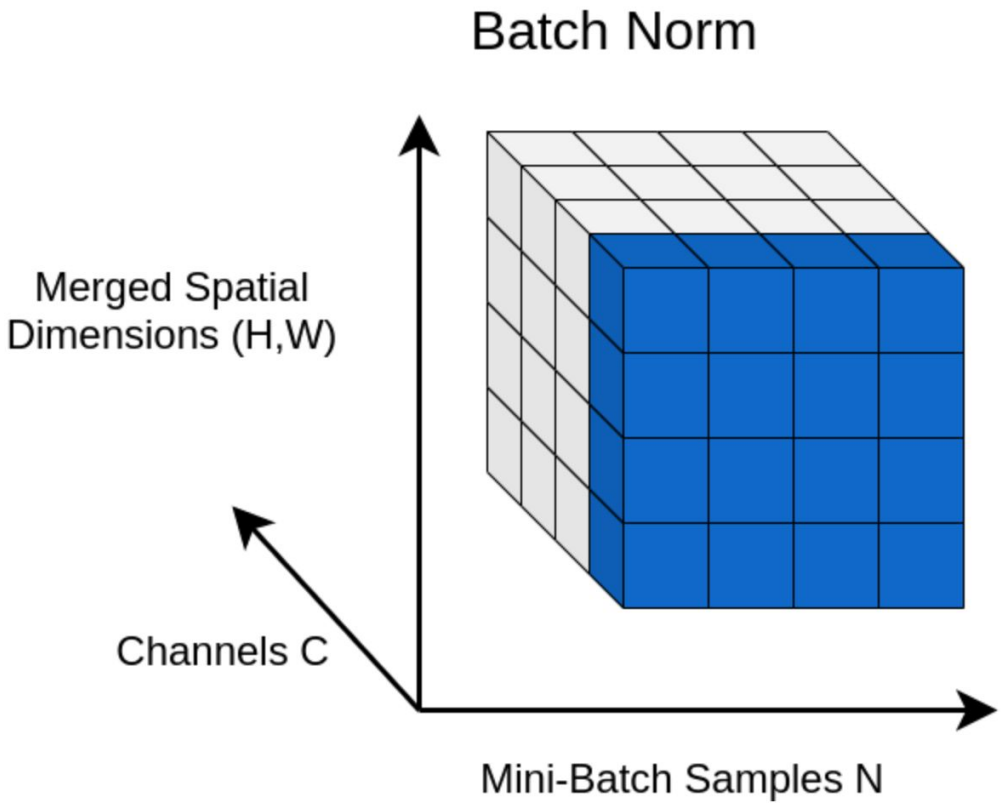
$$y_i = \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i) \quad \text{// сжатие и сдвиг}$$

Сеть без BN:


2xConvolutional block
filters: N, M
=
conv1
3x3
number of filters: N
ReLU
conv2
3x3
number of filters: M
ReLU
max pooling
2x2
stride: 2x2
Input:
grayscale image
50x50
2xConv1
block
filters: 64, 128
out size: 25x25
2xConv2
block
filters: 96, 192
out size: 13x13
2xConv3
block
filters: 128, 256
out size: 7x7
conv4_1
3x3
filters:160
ReLU
conv4_2
3x3
filters:320
avg pooling
7x7
stride: 1x1
with dropout
Fully connected
Output:
14000 classes

Сеть с BN:


2xConvolutional block with BN filters: N, M
conv1 3x3 number of filters: N
ReLU
Batch Normalization
conv2 3x3 number of filters: M
ReLU
max pooling 2x2 stride: 2x2
Input: grayscale image 50x50
2xConv1 block filters: 64, 128 out size: 25x25
1xConv2 block filters: 96 out size: 13x13
2xConv3 block filters: 128, 256 out size: 7x7
conv4_1 3x3 filters:160
ReLU
conv4_2 3x3 filters:320
avg pooling 7x7 stride: 1x1
Fully connected
Output: 14000 classes


Batch Norm
Merged Spatial
Dimensions (H,W)
Channels C
Mini-Batch Samples N

Плюсы batch-norm

- Точность не снижается, потому что BN -- тождественное отображение
- Достигается более быстрая сходимость моделей, несмотря на выполнение дополнительных вычислений;
- Можно использовать более высокий learning rate, так как BN гарантирует, что выходы узлов нейронной сети не будут иметь слишком больших или малых значений;
- BN привносит в выходы узлов скрытых слоев некоторый шум, аналогично методу dropout, поэтому часто его заменяет;
- Модели становятся менее чувствительны к начальной инициализации весов.

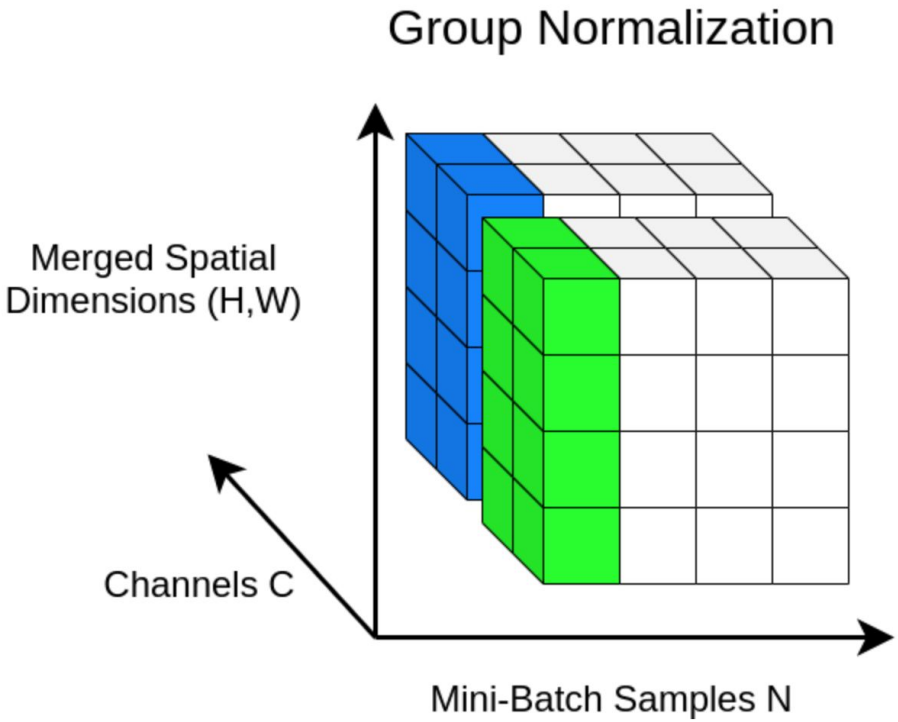
Тогда зачем жить без batch-norm?

- Размер мини-батча очень важен. Но с маленькими мини-батчами результаты плохие, а использование больших ограничивает модели по памяти
- BN рушит независимость между обучающими примерами в мини-батче. Засчет этого результаты часто трудно повторить + возможна утечка информации и переобучение *

*исследование с такой проблемой

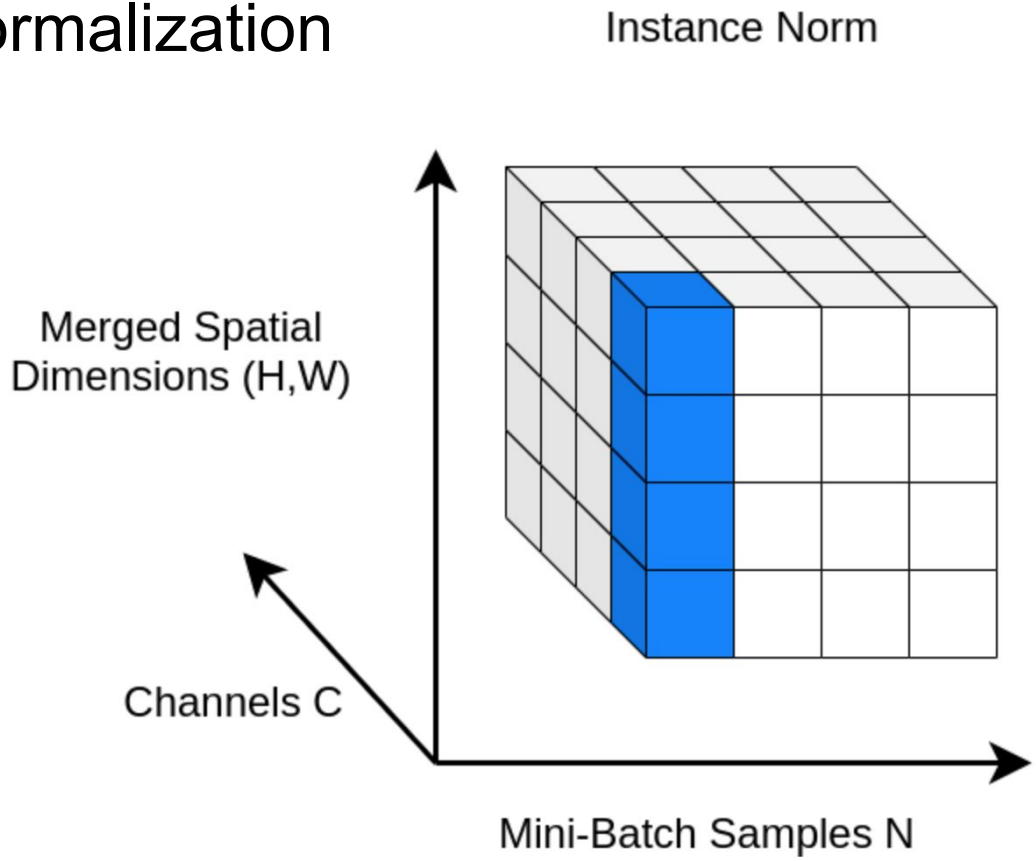
Group Normalization (FAIR)

- Разделяет каналы на группы и нормирует внутри каждой группы
- Не зависит от размера батча и более стабилен при разных размерах
- На ResNet-50, обученной на ImageNet, GN дает ошибку на 10.6% меньше чем аналогичный BN с размером батча 2 *
- В общем случае, GN дает результаты лучше других нормализаций и сравним по результатам с BN.
- GN часто дает лучшие чем у BN результаты для задач object detection.



*исследование [тут](#).

Instance Normalization



Switchable Normalization(SN)

Switchable Normalization (SN) обучается выбирать разные нормализаторы для разных слоев нейронной сети.

У SN нет чувствительного гипер-параметра, в отличии от GN (у него гипер-параметр -- количество групп).

SN сочетает 3 типа нормализаций и переключается между ними, изучая веса их важности:

- Instance Norm (это как BN, только размер батча - 1 объект)
- Layer Norm (это как GN, только размер группы - 1 канал)
- и Batch Norm.

исследование тут.

Еще существующие решения: работы Andrew Brock et al.

- Авторы предложили **Adaptive Gradient Clipping (AGC)**, который обрезает градиенты относительно отношения норм градиентов к нормам параметров, и оказывается, это позволяет обучать сети на больших батчах.
- Авторы разработали Normalizer-Free ResNets, т.н. NFNets, и достигли рекордной точности и скорости обучения (более чем в 8 раз быстрее предыдущей state-of-the-art модели)
- В эксперименте исследования NFNets достигают устойчиво лучшей точности, чем аналогичная модель с BN.

NF-Nets

NF-ResNets: ResNet, у которой веса подвергаются *Scaled Weight Standardization.*

Веса стандартизируются так, чтобы во время back-propagation нормализовались градиенты

NF-Nets: к NF-ResNet добавляется Adaptive Gradient Clipping.

$$\widehat{W}_{ij} = \frac{W_{ij} - \mu_i}{\sqrt{N}\sigma_i}$$

Gradient Clipping

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

Польза: запрещаем делать слишком большие прыжки на градиентном спуске

Проблема: чувствительность к гипер-параметру, после любого изменения параметров модели его нужно тоже перенастраивать

Adaptive Gradient Clipping

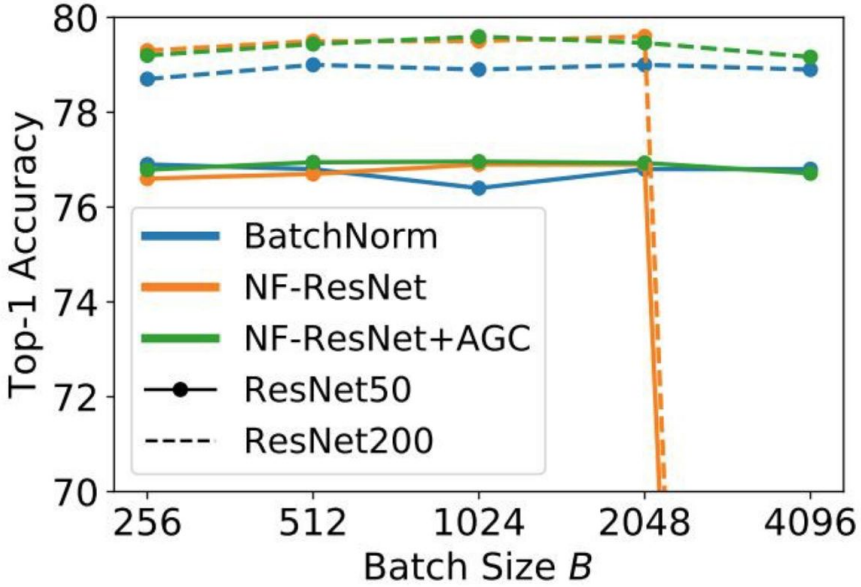
- Так же “урезаем” значение градиента, но не просто так, а относительно отношения нормы градиентов по слою к норме весов по слою. Короче -- “насколько большой градиентный шаг” делить на “насколько большой вес, в отношении которого мы шагаем”.

$$G_i^l = \begin{cases} \lambda \frac{\|W_i^l\|_F}{\|G_i^l\|_F} G_i^l & \frac{\|G_i^l\|_F}{\|W_i^l\|_F} > \lambda \\ G_i^l & otherwise \end{cases}$$

$$\|W_i\|_F = max(\|W_i\|_F, \epsilon)$$

чтобы параметры, инициализированные нулем, не обрезались обратно к нулю

AGC позволяет работать с большими батчами и не падать



Список литературы

1. What Are The Alternatives To Batch Normalization In Deep Learning? Ram Sagar, 2019
2. An Alternative To Batch Normalization, Rahil Vijay, 2019
3. BatchNormalization is not a norm! Questioning basic elements in a Deep Neural Network, Prateek Gulati, 2019
4. A Simple Framework for Contrastive Learning of Visual Representations, Google Research, Brain Team, 2020