Mean Subtraction / Standard Deviation Division

We normalize the dataset to **centre** the data.

- We do this so that our backpropagation gradients don't explode or go out of control
- We centre the data so that our weights are getting modified equally
- **Subtracting Per-Channel Mean**
  - More popular
  - You compute the per-channel mean and subtract it from the original image
  - You don't need to resize or crop the original image
  - You can then also divide the per-channel value by the standard deviation
- **Subtracting Mean Image**
  - Less popular
  - You compute the **mean image** where each pixel is the average pixel at that position
  - You need to make sure that your entire dataset has the same resolution
- **Dividing by the standard deviation**
  - After subtracting the per channel mean, you often want to divide the output by the standard deviation of that feature / pixel
  - This makes the resulting dataset have a standard deviation of 1

Batch Normalization

- A method for normalizing data by subtracting the mean and then dividing by the standard deviation
- **You don't need to use bias in your layers if you're using batchnorm**
- Placed in between every layer, such that the **output of every layer** is batch normalized before going to the next
- If you are **using mini batches**, you **normalize over the whole minibatch**
- **Epoch -** one pass over the full training set
- **Batch -** means you use all of the data to compute gradient during one iteration (what we usually call epoch)
- **Mini Batch -** means you use a subset of the data during one iteration (what we usually call batch)
- **SGD Update -** means you use 1 sample from the data during one iteration (what we usually call online)
- **Benefits**
  - **Increase learning rate -** since we have less over fitting
  - **Remove / Reduce Drop-out -** batchnorm adds resistance to overfitting, so there is less of a need for do
  - **Reduce L_2 Weight Regularisation -** batchnorm has regularizing properties so there is less need
  - **Accelerate learning rate decay -** Learning rate can be made to decay around 6 times faster
    - Means the network learns faster with the same learning rate
- Each layer **k** has an output of **N x D**, where **N** is the number of samples in the minibatch and **D** is the number of features (size of layer **k**)

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

Pix Norm

- Used in ProGAN
-