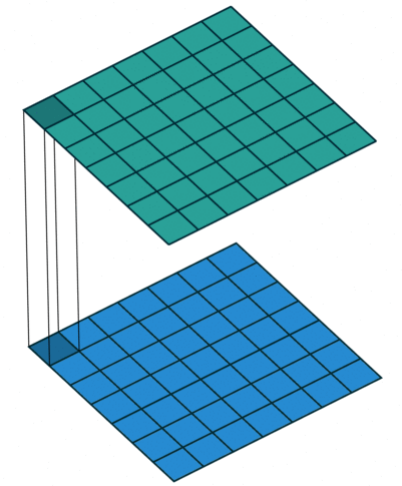
Agenda

CNN components discussion

Regularization techniques discussion

1x1 Convolution

In neural networks 1x1 convolutions are generally used to reduce the dimensions of input in the filter dimension. As the depth of the neural network is increased in general the filter dimension is increased which leads to a lot of computation. To save this effort 1x1 convolutions are used. They were extensively used in Google's inception architecture



Used to mix and match channels

Global Average Pooling

Average Pooling :-

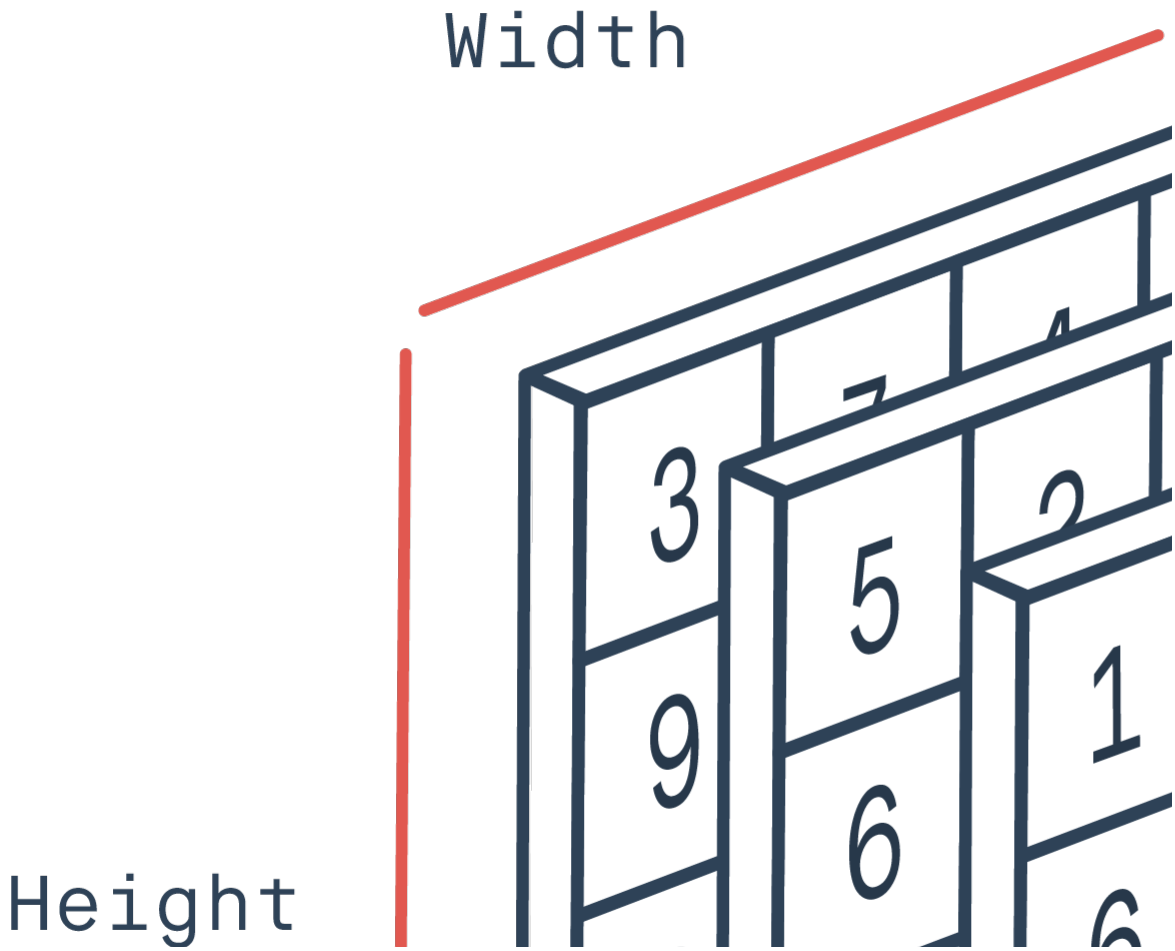
img

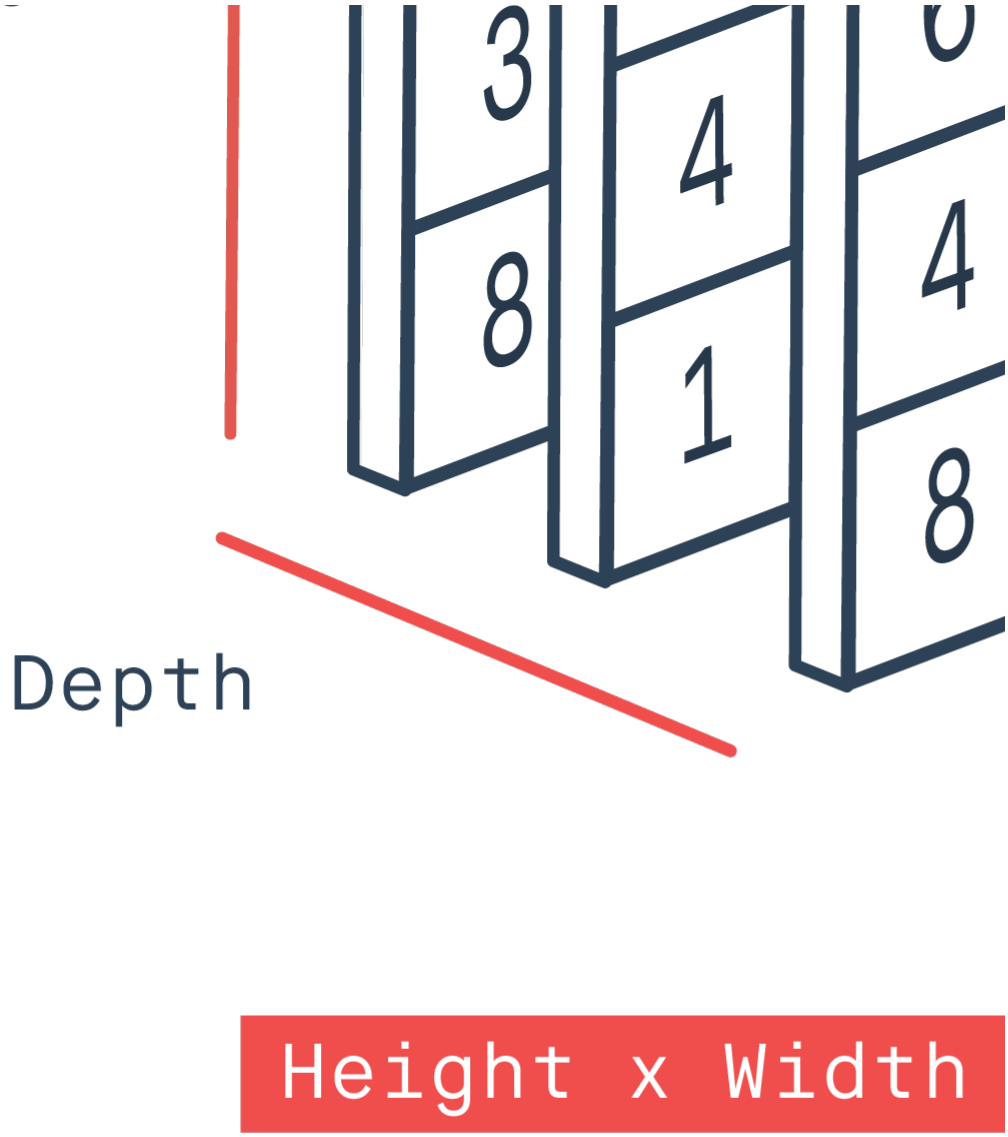
Max pooling preferred over average pooling

Why we need pooling

1. To reduce the input size
2. To achieve local translation and rotation invariance

Average pooling with size equal to that of kernel to constrain it to a single value per channel. Generally used at the output layer in fully convolutional networks





Added to replace fully connected neural networks

Receptive Field

The amount of the image the network looks at. It increases as the depth of the network increases

Regularization techniques

Batch Normalization

Used to normalize the input to every layer

d1, d2, ... dn

m = (d1+d2 + ... + dn)/n

variance = ((d1-m)^2 + (d2-m)^2 + ... + (dn-m)^2))/n

std = variance^1/2

d1 -> (d1-m)/std

d2 -> (d2-m)/std

M = 0

v = 1

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

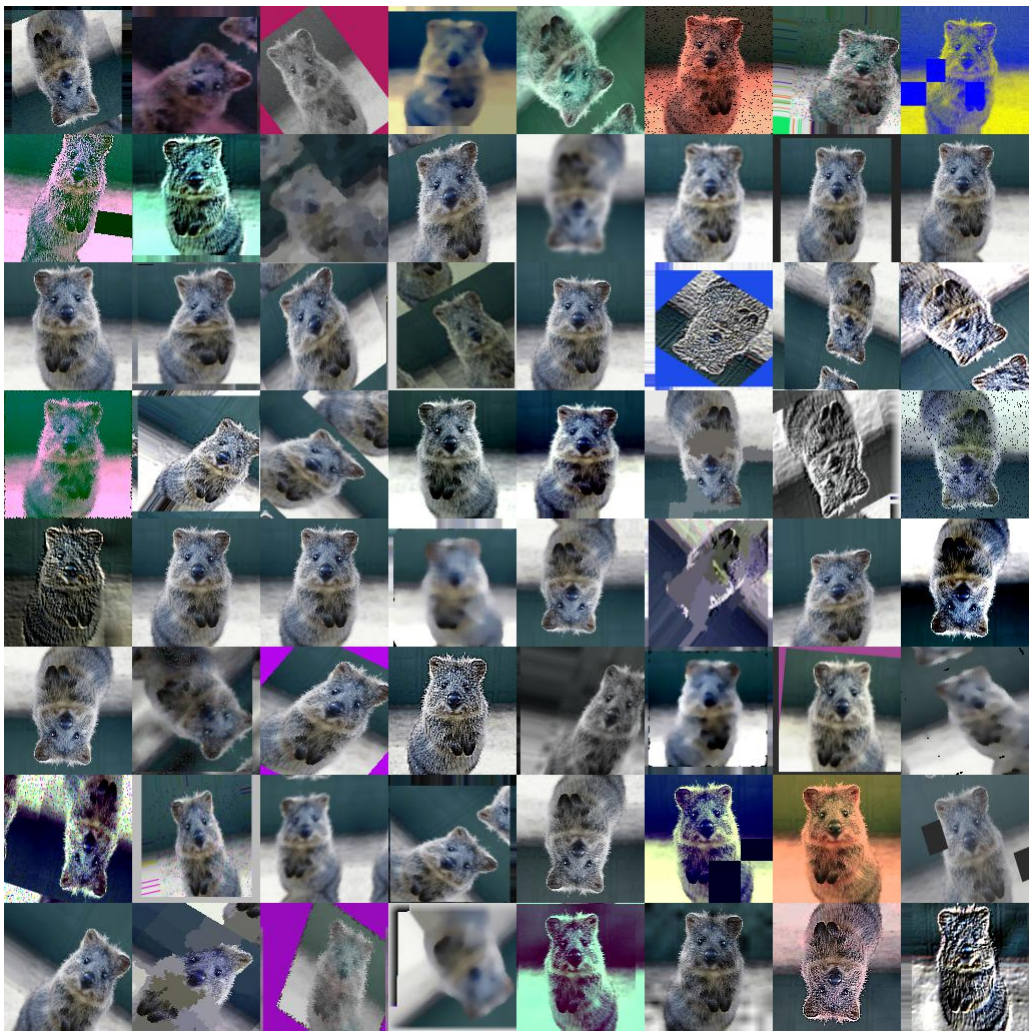
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Dropout

Regularization technique to improve the capability of a network by giving the power of an ensemble of networks

Image Augmentation



L1 and L2 regularization

L1 regularization on least squares:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

L2 regularization on least squares:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

Label Smoothing

Have soft labels instead of hard labels

Dog - 0.9 0.1

Cat - 0.1 0.9

Dog Cat Horse

0.9 0.05 0.05

0.05 0.9 0.05

0.05 0.05 0.9

One sided label smoothing

- Technique proposed in the 1980's
- Instead of having hard labels like 0 and 1, smoothen the labels by making them close to 0 and 1
- For example, 0,1 -> 0.1,0.9