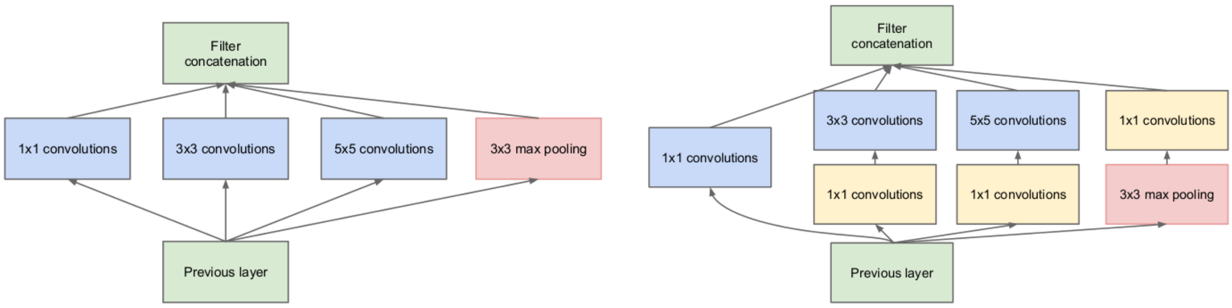
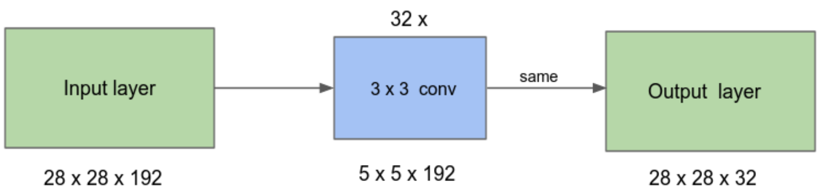
Inception module.

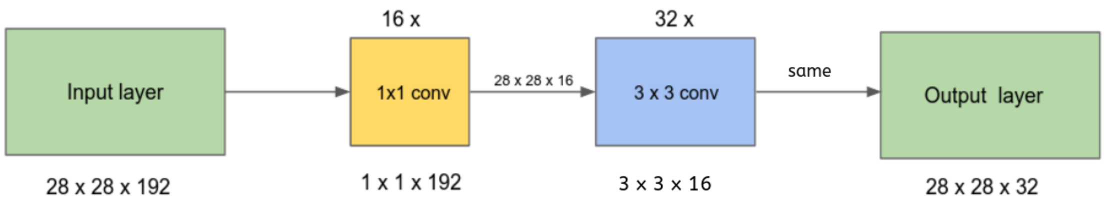
- During ILSVLC-2014, they achieved 1st place at the classification task (top-5 test error = 6.67%)
- It has around 6.7977 million parameters (without auxilaries layers) which is 9x fewer than AlexNet (ILSVRC-2012 winner) and 20x fewer than its competitor VGG-16.
- In most of the standard network architectures, the intuition is not clear why and when to perform the max-pooling operation, when to use the convolutional operation. For example, in AlextNet we have the convolutional operation and max-pooling operation following each other whereas in VGGNet, we have 3 convolutional operations in a row and then 1 max-pooling layer.
- Thus, **the idea behind GoogLeNet is to use all the operations at the same time**. It computes multiple kernels of different size over the same input map in parallel, concatenating their results into a single output. This is called an **Inception module**.



(a) Inception module, naïve version (b) Inception module with dimension reductions

- Consider the following:

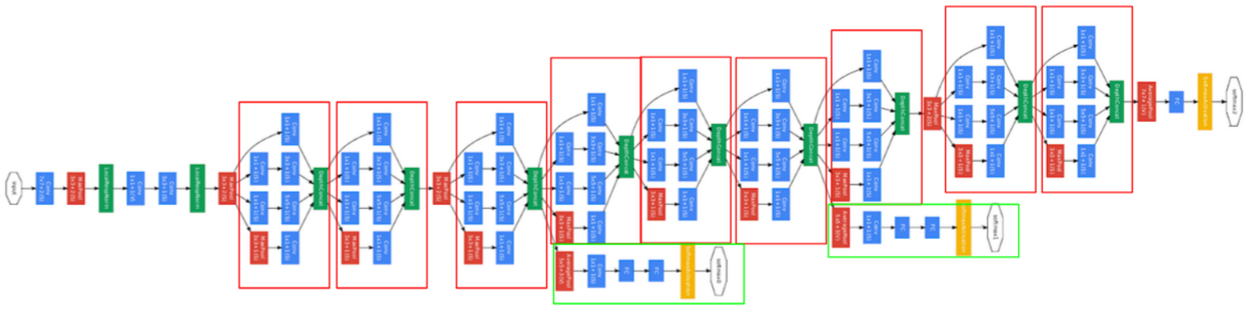




- The Naive approach is computationally expensive:
  - Computation cost = ((28 x 28 x 5 x 5) x 192) x 32 $\simeq$ **120 Mil**
    - We perform (28 x 28 x 5 x 5) operations along 192 channels for each of the 32 filters.
- The dimension reduction approach is **less** computationally expensive:

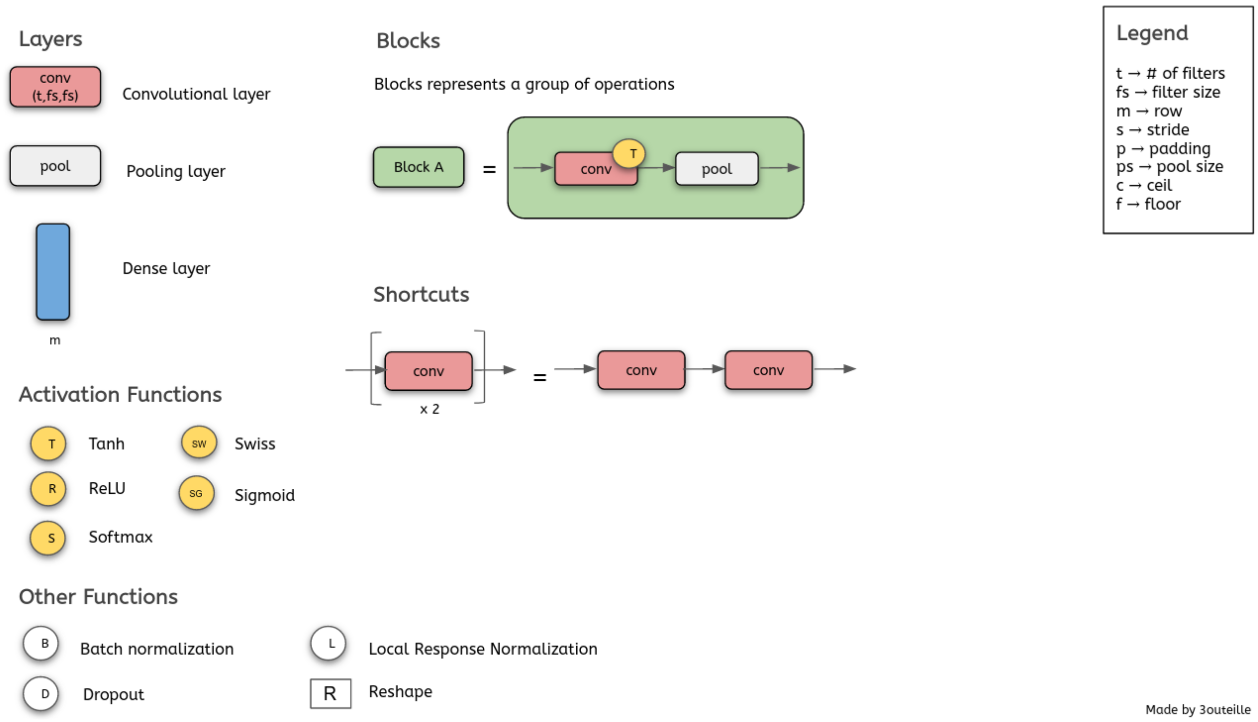
- 1st layer computation cost = ((28 x 28 x 1 x 1) x 192) x 16 $\simeq$ 2.4 Mil
- 2nd layer computation cost = ((28 x 28 x 5 x 5) x 16) x 32 $\simeq$ 10 Mil
- Total computation cost $\simeq$ **12.4 Mil**

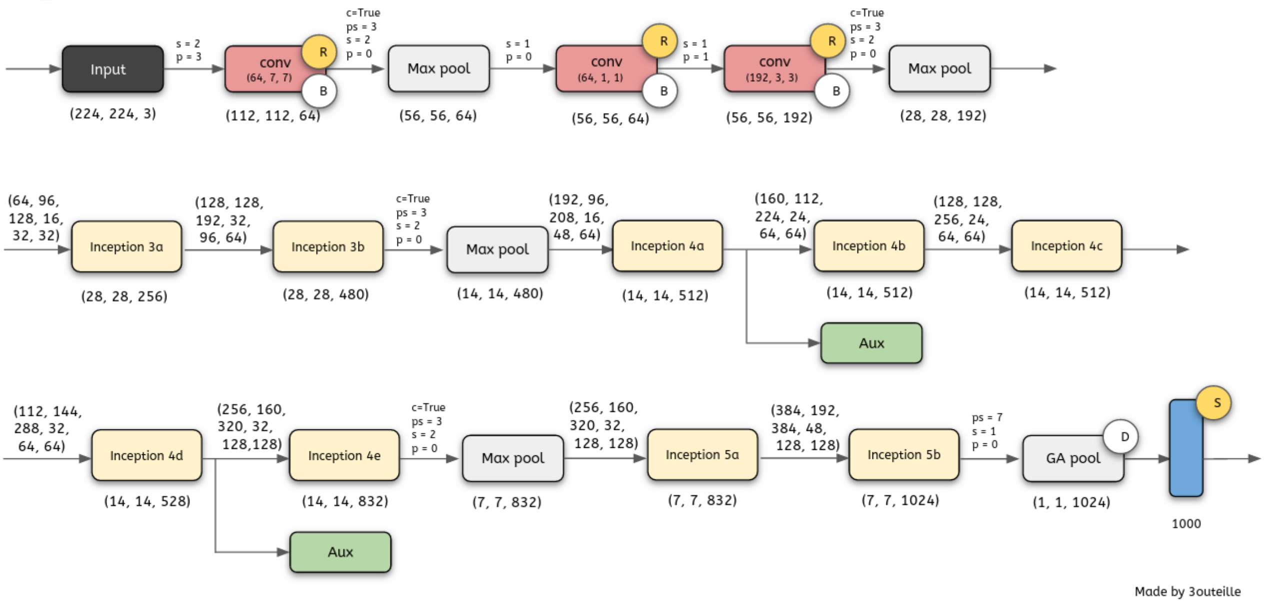
Here its architecture:

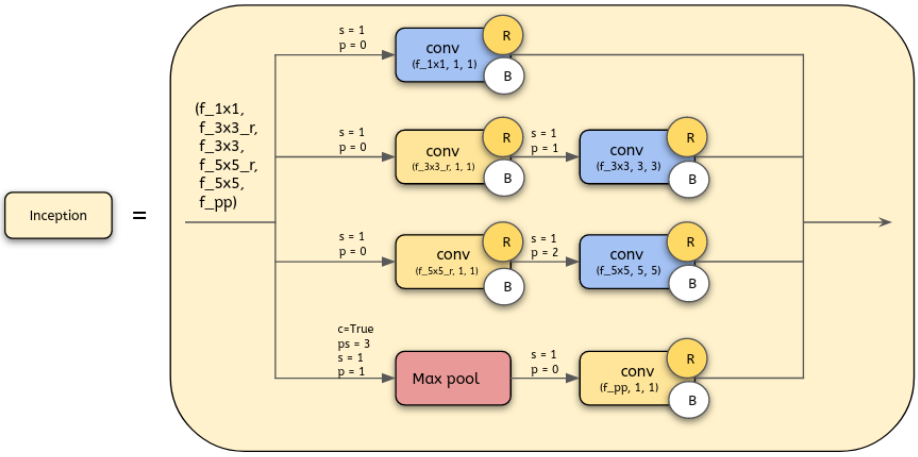


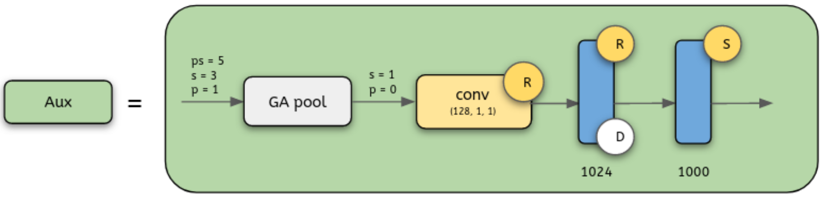
- There are:
  - 9 Inception modules (red box)
  - Global Average pooling were used instead of a Fully-connected layer.
    - It enables adapting and fine-tuning on the network easily.
  - 2 auxilaries softmax layer (green box)
    - Their role is to push the network toward its goal and helps to ensure that the intermediate features are good enough for the network to learn.
    - It turns out that softmax0 and sofmax1 gives regularization effect.
    - During training, their loss gets added to the total loss with a discount weight (the losses of the auxiliary classifiers were weighted by 0.3).
    - During inference, they are discarded.
    - Structure:
      - Average pooling layer with 5×5 filter size and stride 3 resulting in an output size:
        - For 1st green box: 4x4x512.
        - For 2nd green box: 4x4x528.
      - 128 1x1 convolutions + ReLU.
      - Fully-connected layer with 1024 units + ReLU.
      - Dropout = 70%.
      - Linear layer (1000 classes) + Softmax.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								









Legend

f_1x1 → # of filters for 1x1 conv.

f_3x3_r → # of filters for 1x1 conv before 3x3 conv.

f_3x3 → # of filters for 3x3 conv.

f_5x5_r → # of filters for 1x1 conv before 5x5 conv.

f_5x5 → # of filters for 5x5 conv.

f_pp → # of filters for 1x1 conv after pooling.

Made by 3outeille