Convolutional Neural Networks Variants

Dinesh Kumar Vishwakarma, Ph.D.

PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY, DELHI.

Webpage: http://www.dtu.ac.in/Web/Departments/InformationTechnology/faculty/dkvishwakarma.php

Email: dinesh@dtu.ac.in

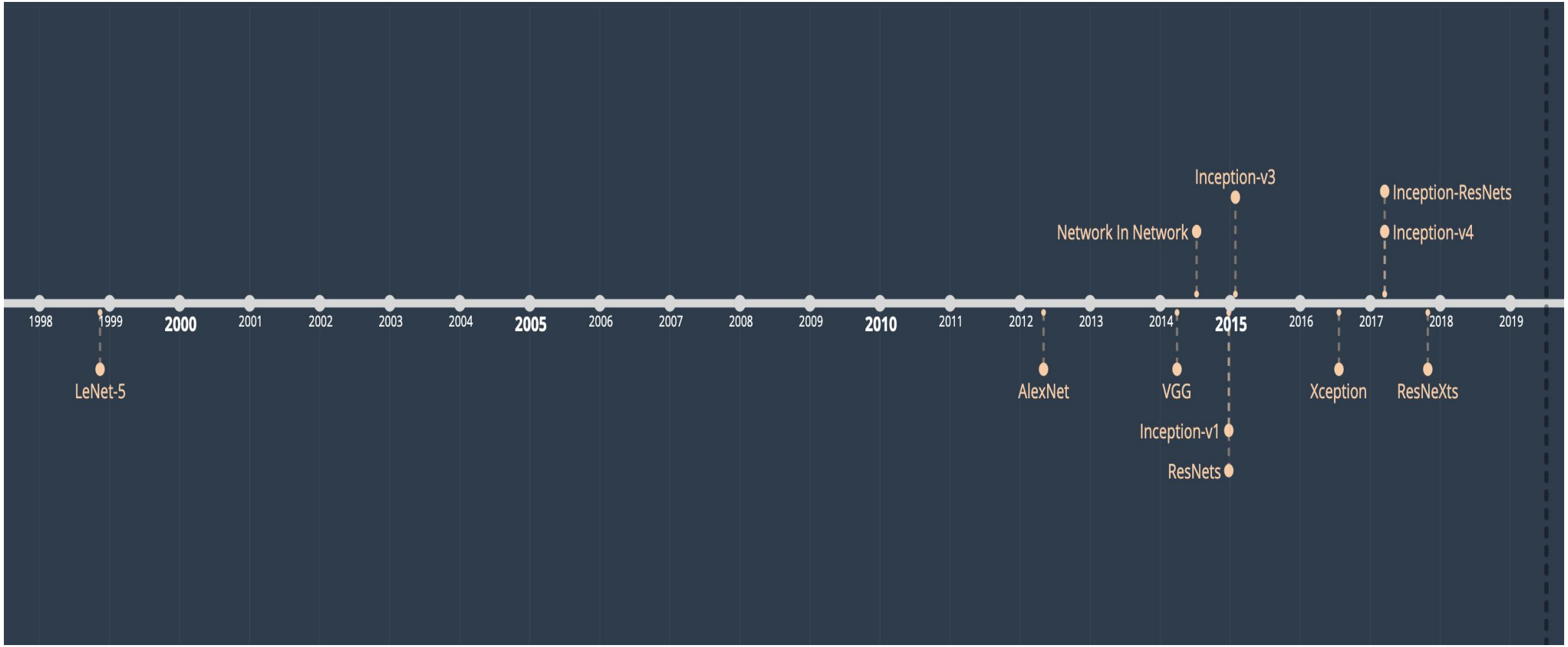
Introduction

Since 1998, number of CNN models are developed.

- Recent models are so deep and extremely difficult to visualize. Hence, it becomes like 'black box'.
- Starting from LeNet 5 to Latest ResNeXt-50.

1. LeNet-5	2. AlexNet	3. VGG-16	4. Inception-v1	5. Inception-v3
6. ResNet-50	7. Xception	8. Inception-v4	9. Inception-Res Nets	10. ResNeXt-50

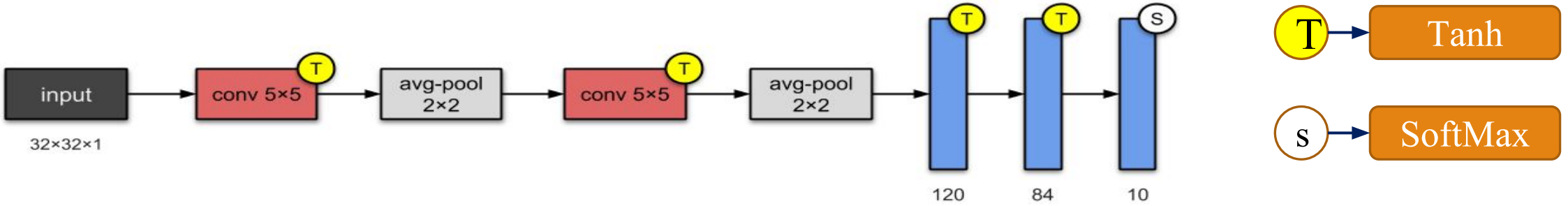
Historical Progress Year wise as on 05.04.20



LeNet 5

Developed in 1998 by Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner.

- One of the simplest architecture.
- It has **02-Convo** layer and **03-FC layer**. Hence, total **05 layers** and very commonly known as **LeNet 5**.
- It has 60000 parameters.
- Implemented on CPU.
- **Novel:** Stacking of Conv layer and FC Layer.

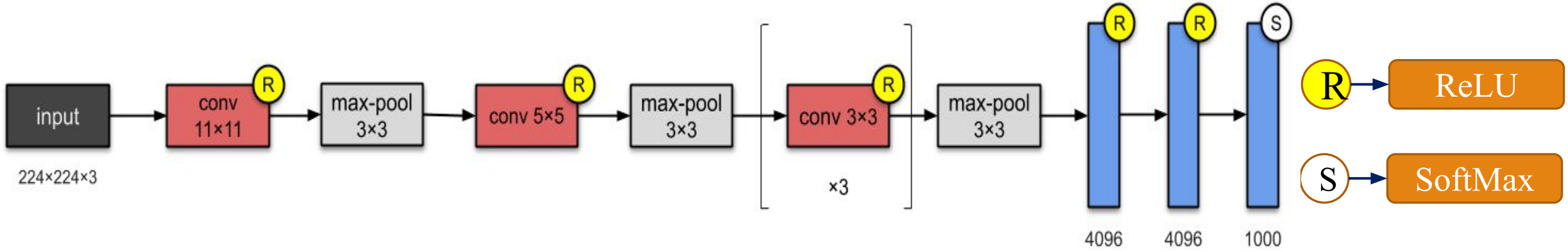


Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

AlexNet

Developed in 2012 by Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton. University of Toronto, Canada.

- It has **05-Convo** layer and **03-FC layer**. Hence, total **08 layers** and very commonly known as **AlexNet**. **Novel:** First to use ReLU
- It has **60M** parameters and stacked **few more layers** onto **LeNet-5**.
- In 2012, it was “one of the largest CNNs to date on the subsets of ImageNet.” it also won ImageNet challenge in 2012.

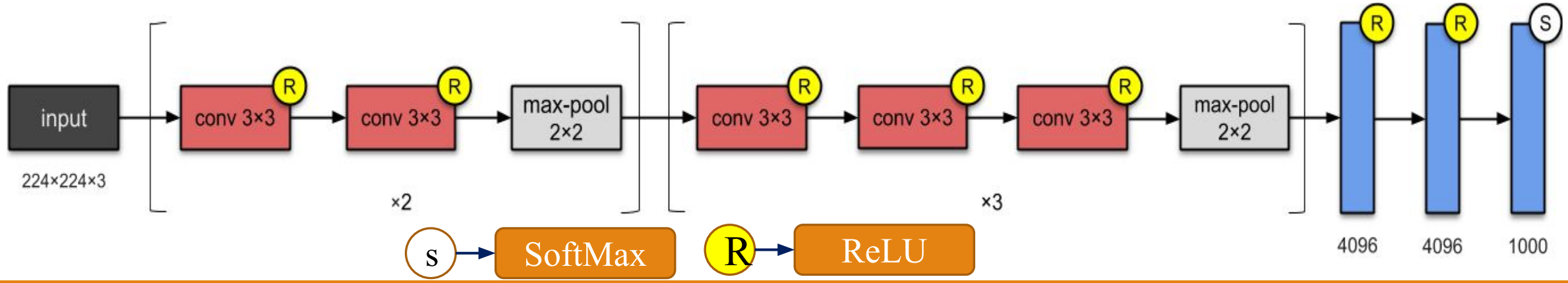


Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In *Advances in neural information processing systems*, pp. 1097-1105. 2012.

VGG-16

VGG-16 invented by Simonyan et al. (2014) of Visual Geometry Group, CNN gets deeper and deeper with time. They also designed VGG-19.

- It has **13-Convo** layer, **03-FC layer** and ReLU Activation. Total **16 layers** and that why known as **VGG-16**. Novel: designing of deep N/W, twice as compared to AlexNet.
- It has **130M** parameters, take **500MB storage space**, and **stacked more layers** onto **AlexNet**. Uses small filter size **2×2 and 3×3**.



Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014 Sep 4.

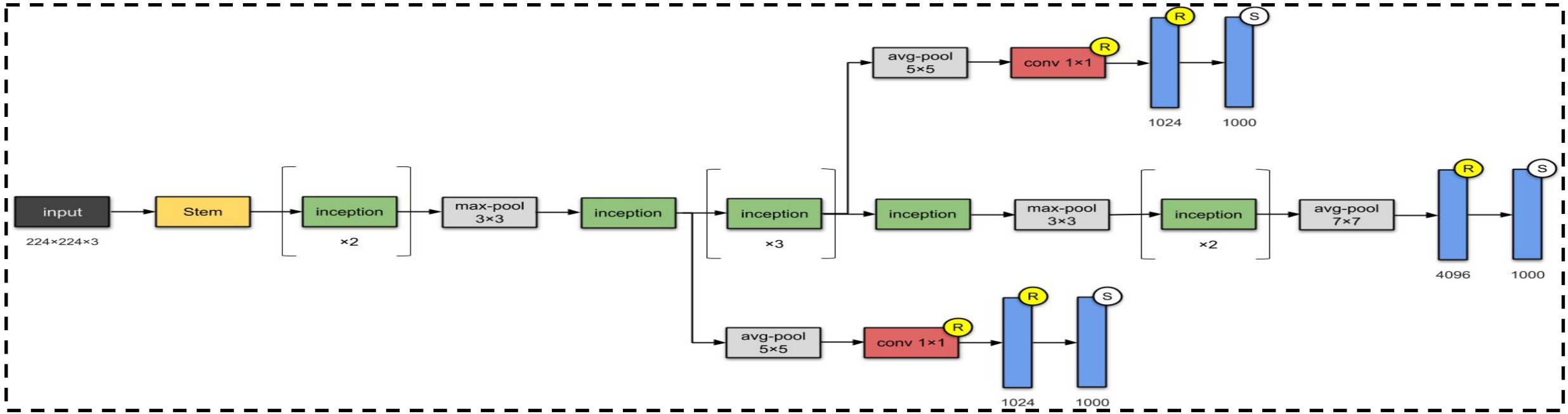
Inception V1 or GoogLeNet

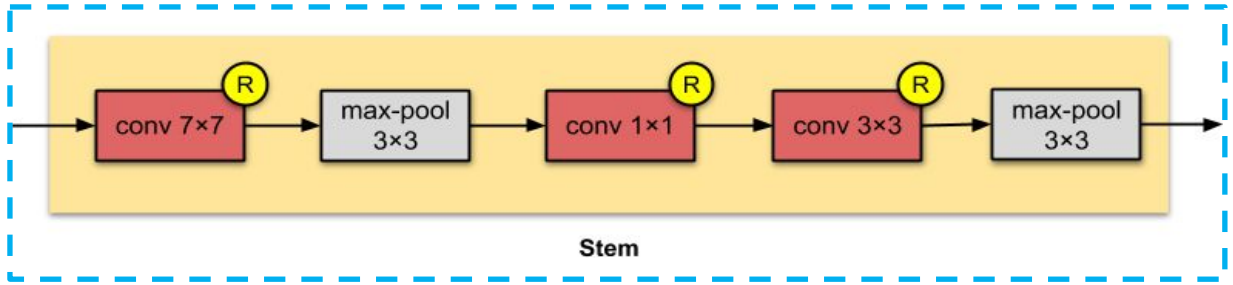
Developved by Szegedy et al.2014. It has **22-**layer architecture with **5M** parameters.

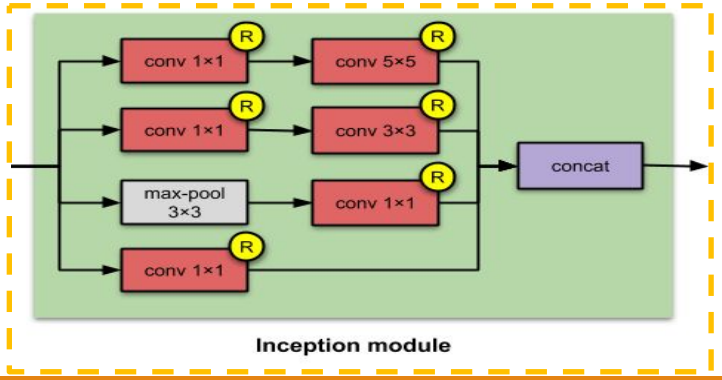
- It uses the concept of N/W in N/W approach. It is done by Inception module.
- The Inception module is designed by *a product of research on approximating sparse structures* and each module presents three ideas.
  - a) Having parallel towers of convolutions with different filters, b) followed by concatenation, captures different features at 1×1, 3×3 and 5×5, thereby c) 'clustering' them.
  - The above idea is motivated by Arora et al. 2014, where it was suggested that *a layer-by layer construction in which one should analyse the correlation statistics of the last layer and cluster them into groups of units with high correlation*.
  - 1×1 convolutions are used for dimensionality reduction to remove computational bottlenecks.
  - Due to the activation function from 1×1 convolution, its addition also adds nonlinearity. This idea is based on the Network In Network concept.
  - The authors also introduced two auxiliary classifiers to encourage discrimination in the lower stages of the classifier, to increase the gradient signal that gets propagated back, and to provide additional regularization.

C. Szegedy *et al*., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9.

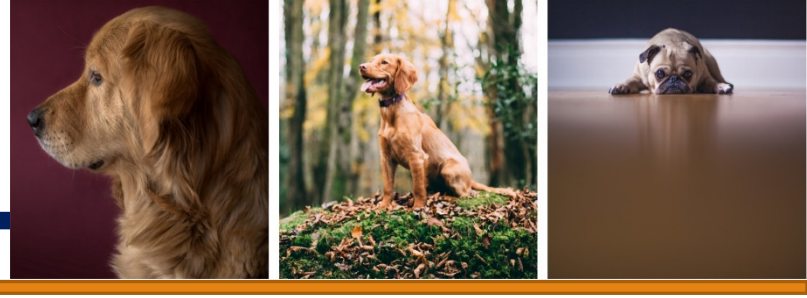
Inception V1 or GoogLeNet…







Inception V1 or GoogLeNet.

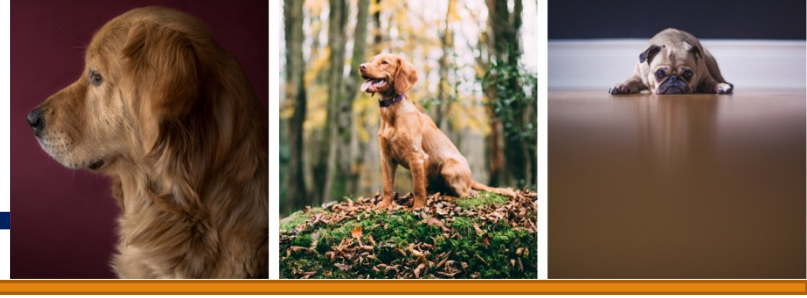


Challenges involved in image recognition leads to the development of more deeper network with variation of filter size.

- **Salient parts** in the image can have extremely **large variation** in size. Ex. The area occupied by the dog is different in each image.
- Because of the huge variation in the location of the information, choosing the **right kernel size** for the convolution operation becomes tough. A **larger kernel** is preferred for information that is distributed more **globally**, and a **smaller kernel** is preferred for information that is distributed more **locally.**
- **Very deep networks** are prone to **overfitting** and also difficult to pass gradient updates through the entire network.
- Naively stacking large convolution operations is **computationally expensive**.

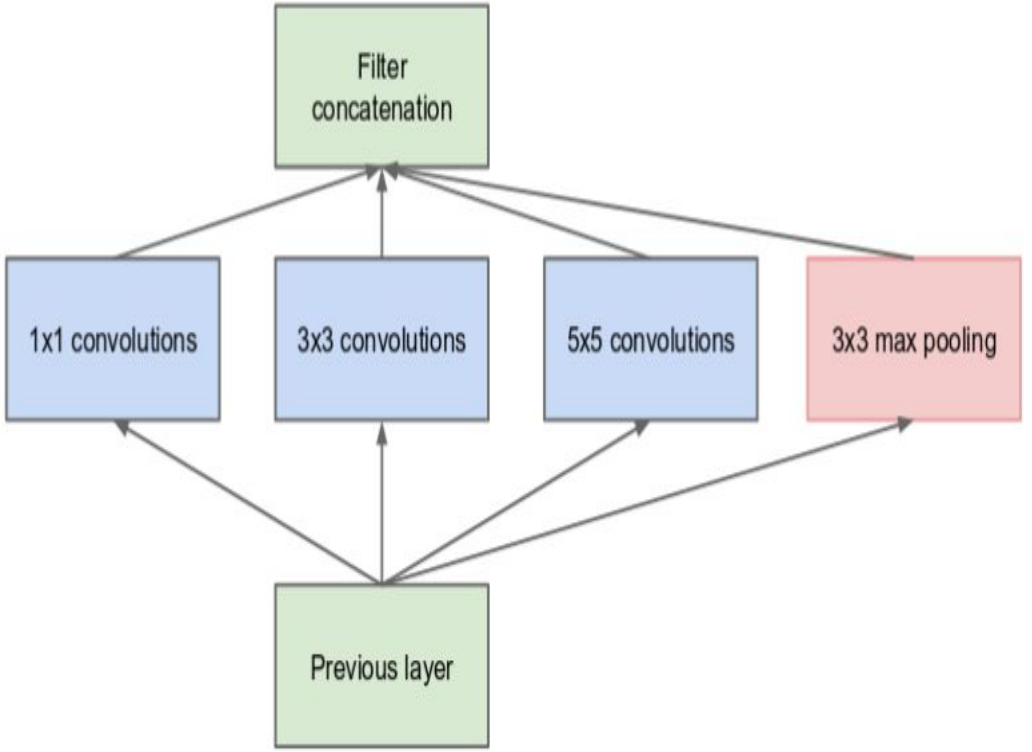
C. Szegedy *et al*., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9.

Inception V1 or GoogLeNet.



Solution of these challenges are to the development of more deeper network with variation of filter size.

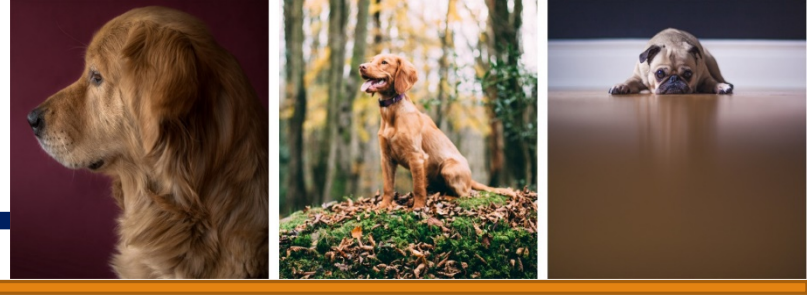
- To have filters with **multiple sizes,** which operates on the **same level.**
- The network layer would get a bit "**wider**" rather than "deeper".
- Hence, **inception module** is designed to perform **convolution** on an I/P with **3 different sizes of filters** (1x1, 3x3, 5x5).
- Additionally, **max pooling** is also performed.
- The outputs are **concatenated** and sent to the next inception module.



(a) Inception module, naïve version

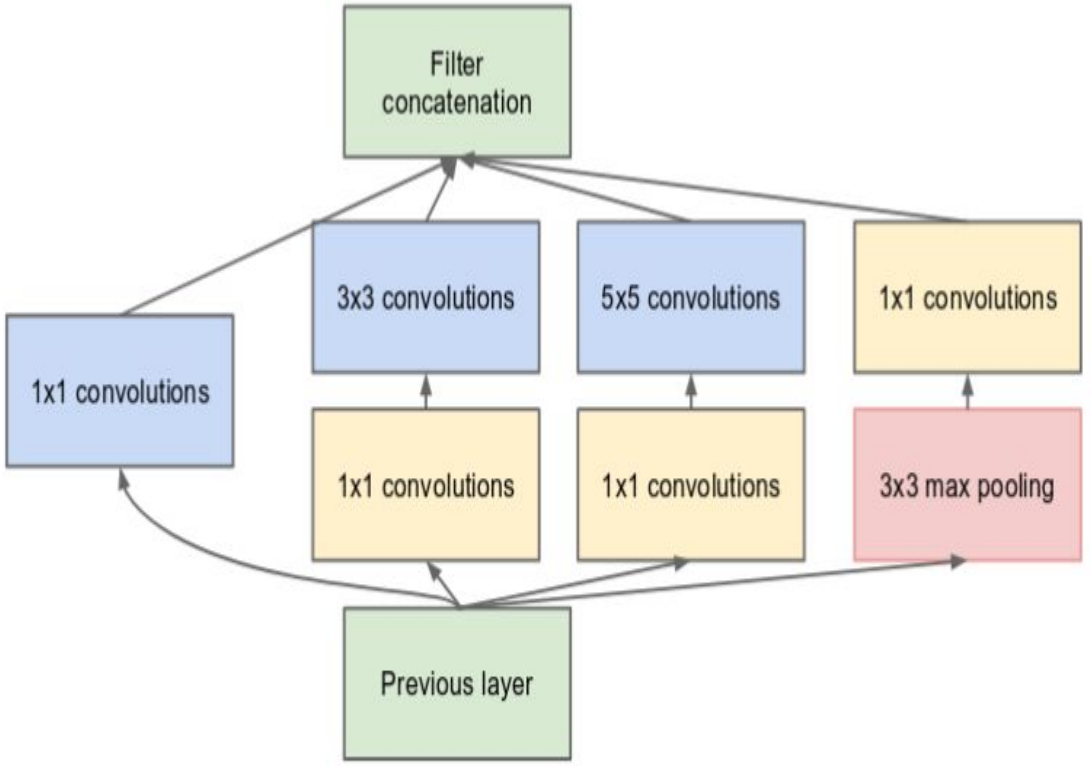
C. Szegedy *et al*., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9.

Inception V1 or GoogLeNet.



Solution to ***computational expensive*** of deep network is addressed.

- To make it cheaper, the authors **limit** the number of **input channels** by adding an **extra 1x1 convolution** before the 3x3 and 5x5 convolutions.
- Though adding an extra operation may seem counterintuitive, 1x1 convolutions are far more cheaper than 5x5 convolutions, and the reduced number of input channels also help.
- Do note that however, the 1x1 convolution is introduced after the max pooling layer, rather than before.

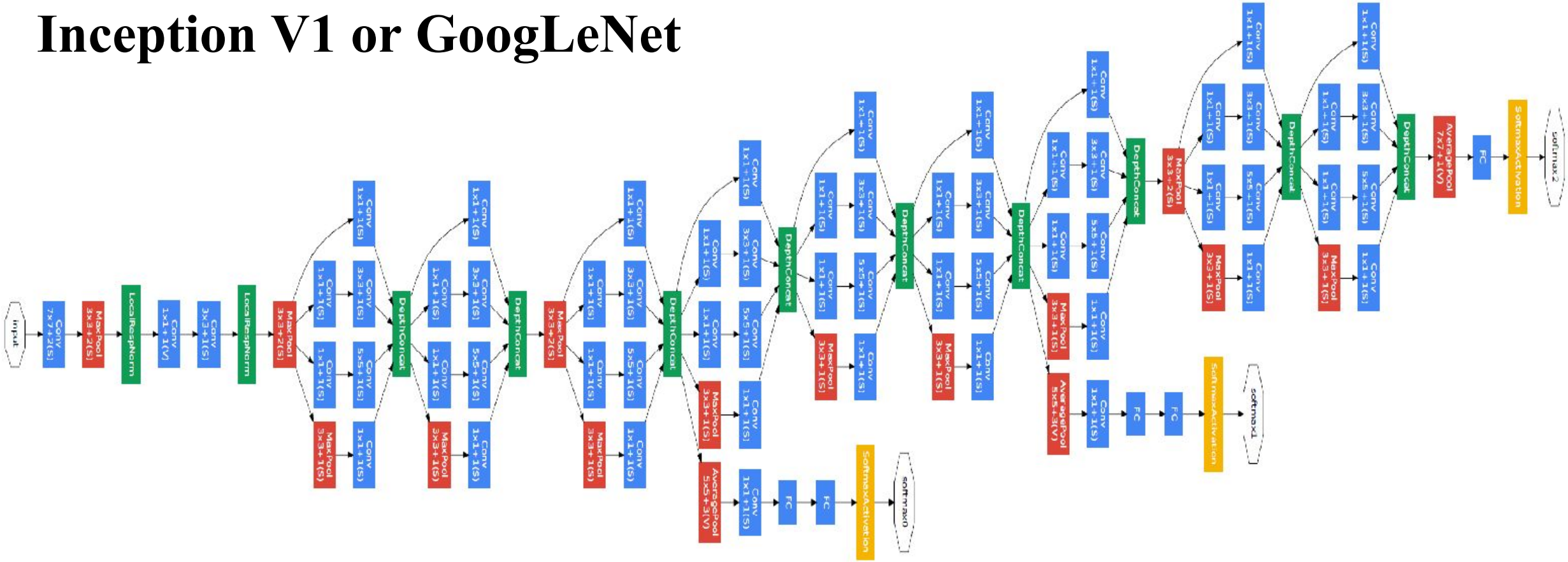


(b) Inception module with dimension reductions

C. Szegedy *et al*., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9.

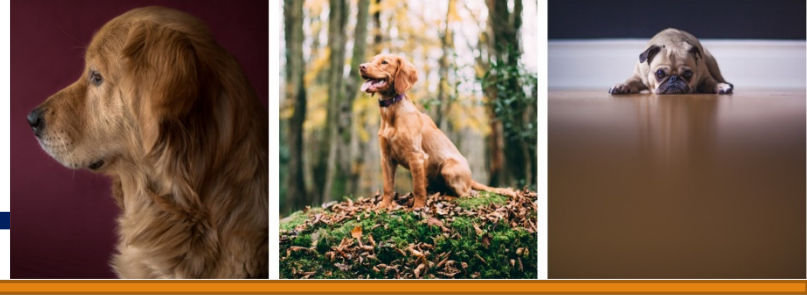
Actual Diagram of Network

Inception V1 or GoogLeNet



C. Szegedy *et al.*, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9.

Inception V1 or GoogLeNet.



Architecture — Novel: stacking of modules instead of Conv layer

- GoogLeNet has 9 inception modules stacked linearly. It is 22 layers deep (27, including the pooling layers). It uses global average pooling at the end of the last inception module.
- Needless to say, it is a pretty **deep classifier**. As with any very deep network, it is subject to the **vanishing gradient problem**.
- To prevent the **middle part** of the network from "**dying out**", the authors introduced **two auxiliary classifiers**.
- They essentially applied softmax to the outputs of two of the inception modules, and computed an **auxiliary loss** over the same labels.
- The **total loss function** is a **weighted sum** of the **auxiliary loss** and the **real loss**. Weight value used in the paper was 0.3 for each auxiliary loss and used for training purpose.

C. Szegedy *et al*., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9.

Inception V2

Inception V2 and V3 both were presented in same work by Szegedy et al. (Dec. 2015).

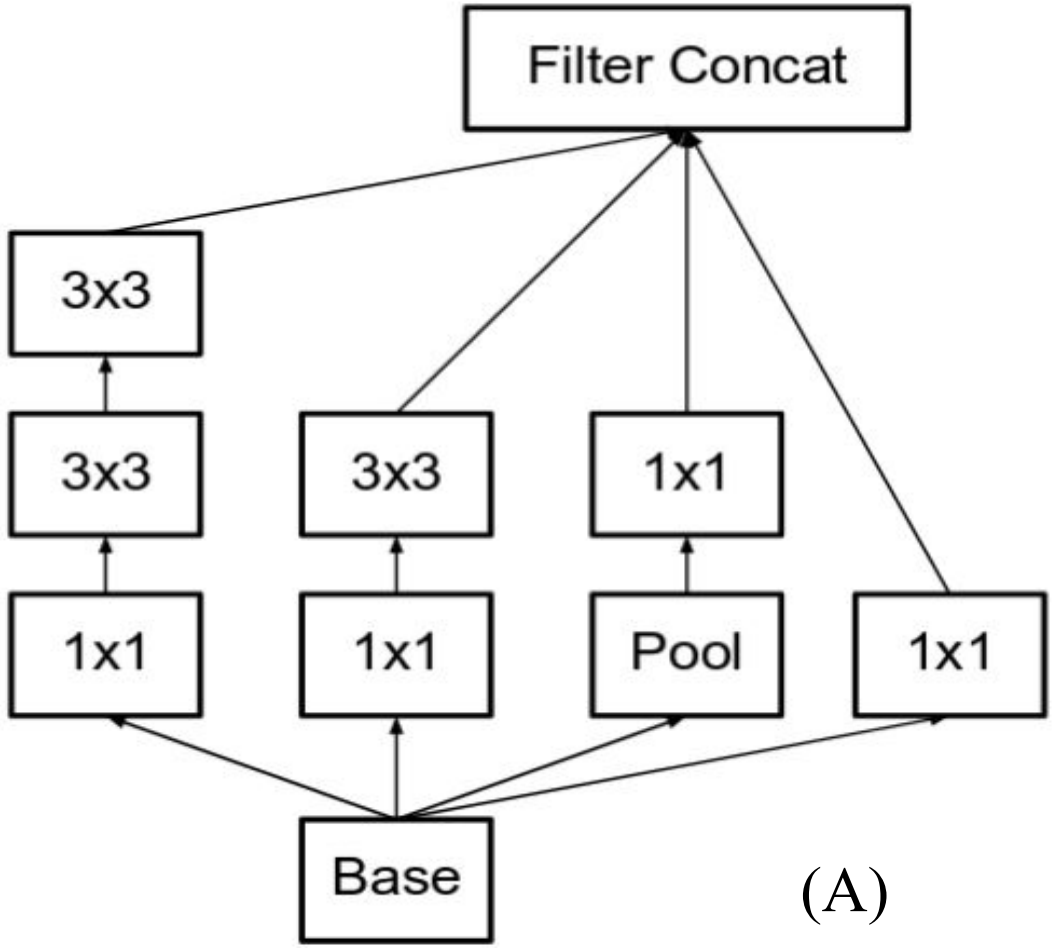
- Number of variants are proposed to increase accuracy and reduce the computational cost.
- **Inception V2 explores:** Reduce **representational bottleneck**.
- The intuition was that, neural networks perform better when convolutions didn't alter the dimensions of the input drastically.
- Reducing the dimensions too much may cause loss of information, known as a "representational bottleneck".
- Using smart factorization methods, convolutions can be made more efficient in terms of computational complexity.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

Inception V2…

Solutions of these problems are:

- **Factorize 5x5** convolution **to two 3x3** convolution operations to improve computational speed.
- Although this may seem counterintuitive, a 5x5 convolution is **2.78 times more expensive** than a 3x3 convolution.
- So stacking two 3x3 convolutions infact leads to a boost in performance. This is illustrated in the image.

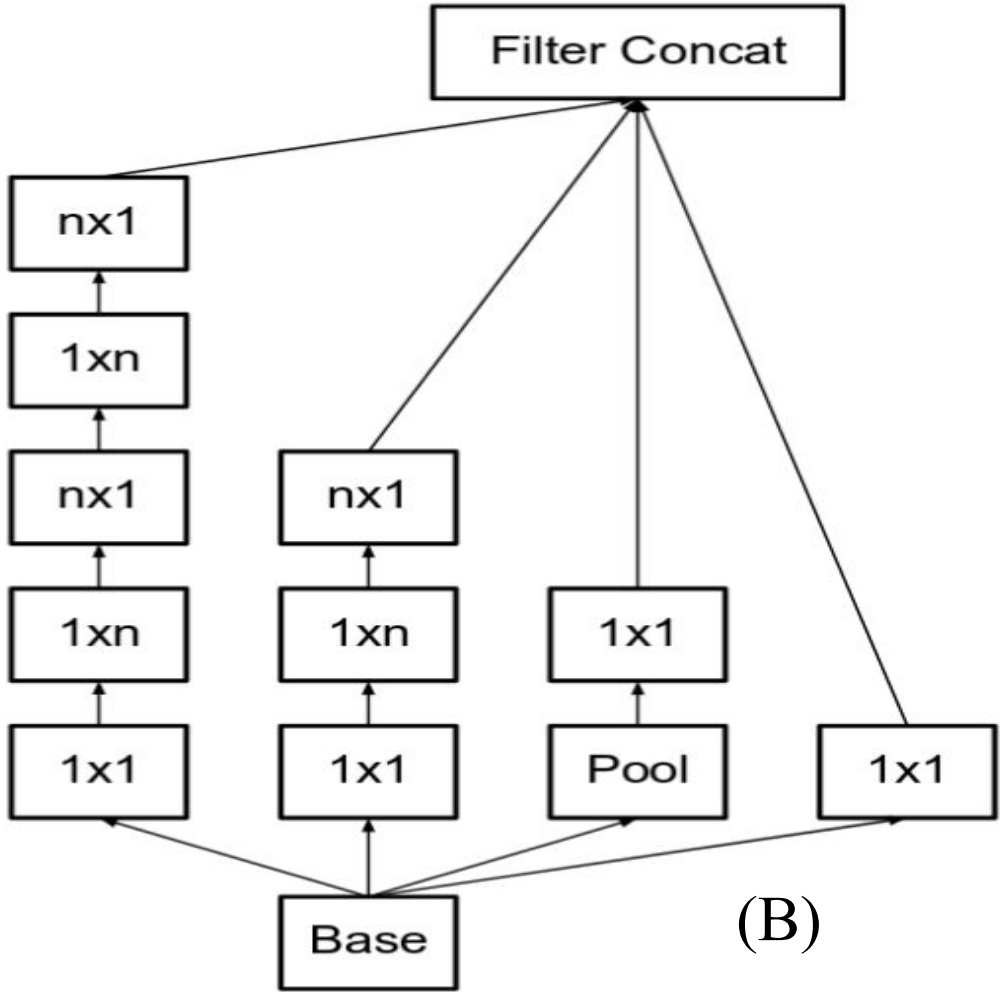


Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

Inception V2…

Solutions of these problems are:

- Moreover, they **factorize** convolutions of filter size **nxn** to a **combination** of **1xn and nx1** convolutions.
- For example, a 3x3 convolution is equivalent to first performing a 1x3 convolution, and then performing a 3x1 convolution on its output.
- They found this method to be **33% more cheaper** than the single 3x3 convolution. This is illustrated in the below image.

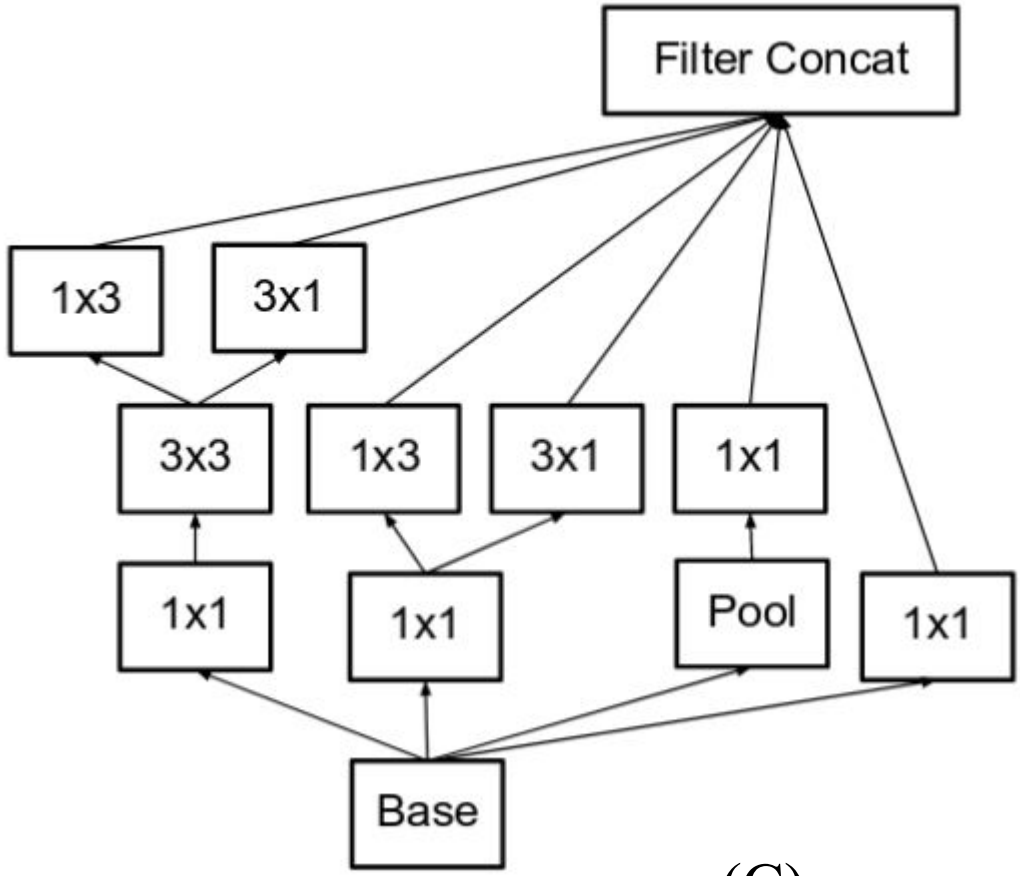


Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

Inception V2…

Solutions of representational bottleneck are:

- The **filter banks** in the module were **expanded** (made wider instead of deeper) to remove the representational bottleneck. If the module was made deeper instead, there would be excessive reduction in dimensions, and hence loss of information.
- The above three principles were used to build three different types of inception modules and these may be denoted as **Fig. 5: A, Fig. 6: B** and **Fig. 7: C. (Original Paper)**



(C)

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

Inception V2 Architecture Parameters

type	patch size/stride or remarks	input size
conv	$3\times3/2$	$299\times299\times3$
conv	$3\times3/1$	$149\times149\times32$
conv padded	$3\times3/1$	$147\times147\times32$
pool	$3\times3/2$	$147\times147\times64$
conv	$3\times3/1$	$73\times73\times64$
conv	$3\times3/2$	$71\times71\times80$
conv	$3\times3/1$	$35\times35\times192$
$3\times$Inception	As in figure 5	$35\times35\times288$
$5\times$Inception	As in figure 6	$17\times17\times768$
$2\times$Inception	As in figure 7	$8\times8\times1280$
pool	8×8	$8\times8\times2048$
linear	logits	$1\times1\times2048$
softmax	classifier	$1\times1\times1000$

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

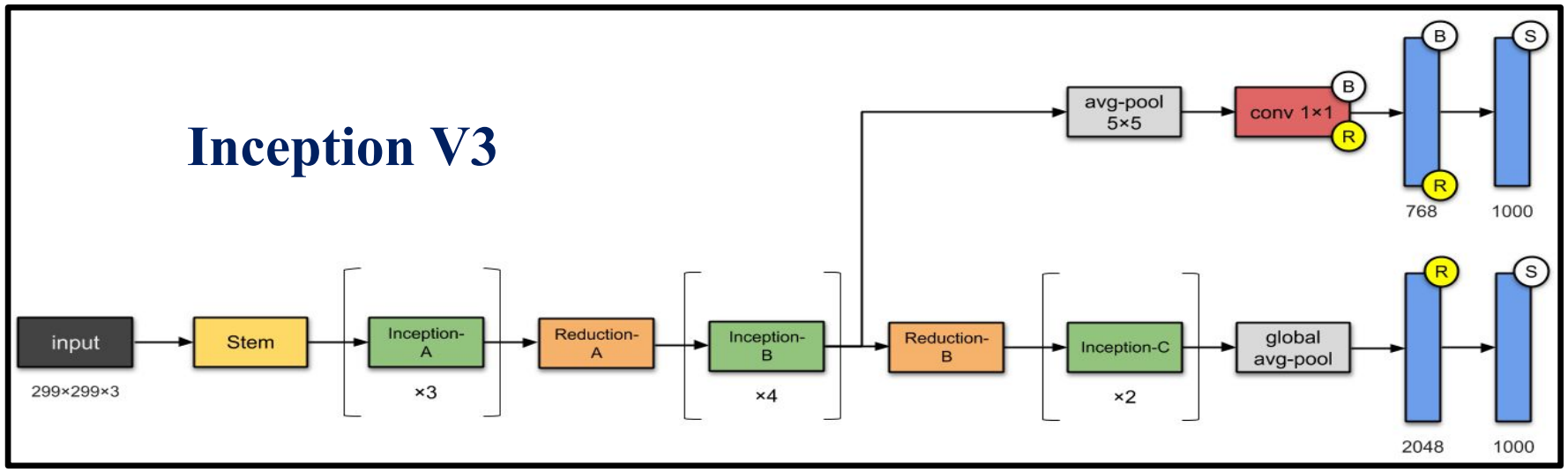
Inception V3

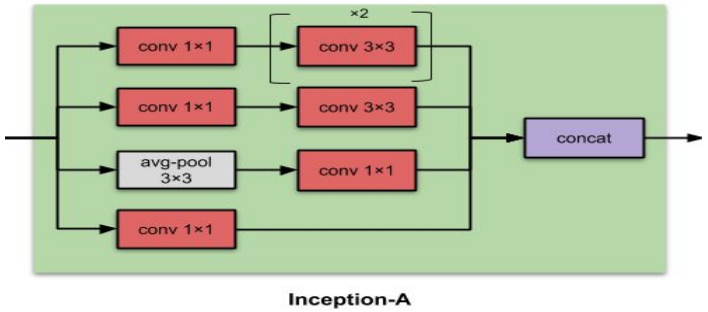
- In Inception V2, it was noticed that the **auxiliary classifiers** didn't contribute much until near the end of the training process, when accuracies were nearing saturation.
- They argued that the function as **regularizes**, especially if they have **BatchNorm** or **Dropout** operations.
- Possibilities to improve on the Inception v2 without drastically changing the modules were to be investigated.

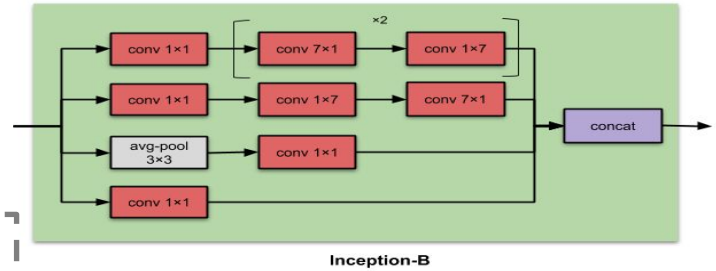
- In **Inception V3** uses the following in addition to inception V2.
  - RMSProp Optimizer.
  - Factorized 7x7 convolutions.
  - BatchNorm in the Auxillary Classifiers.
  - Label Smoothing (A type of regularizing component added to the loss formula that prevents the network from becoming too confident about a class. Prevents over fitting).

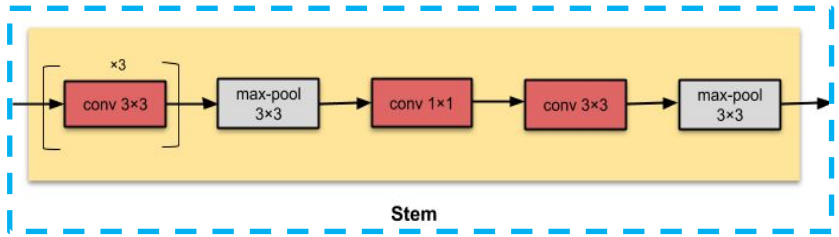
Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

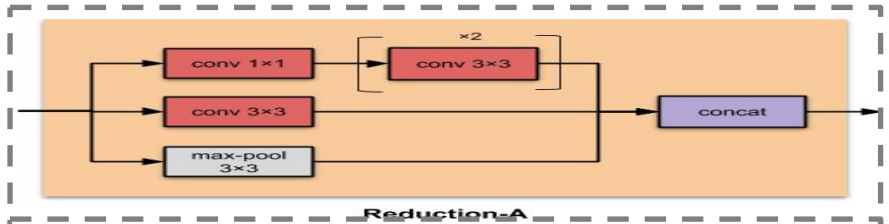
Layout Diagram of Inception V3

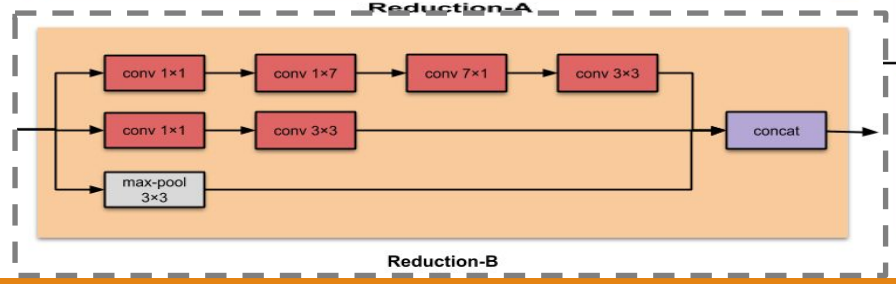


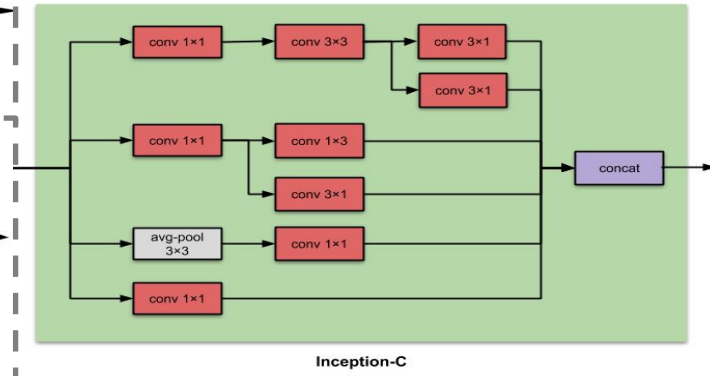












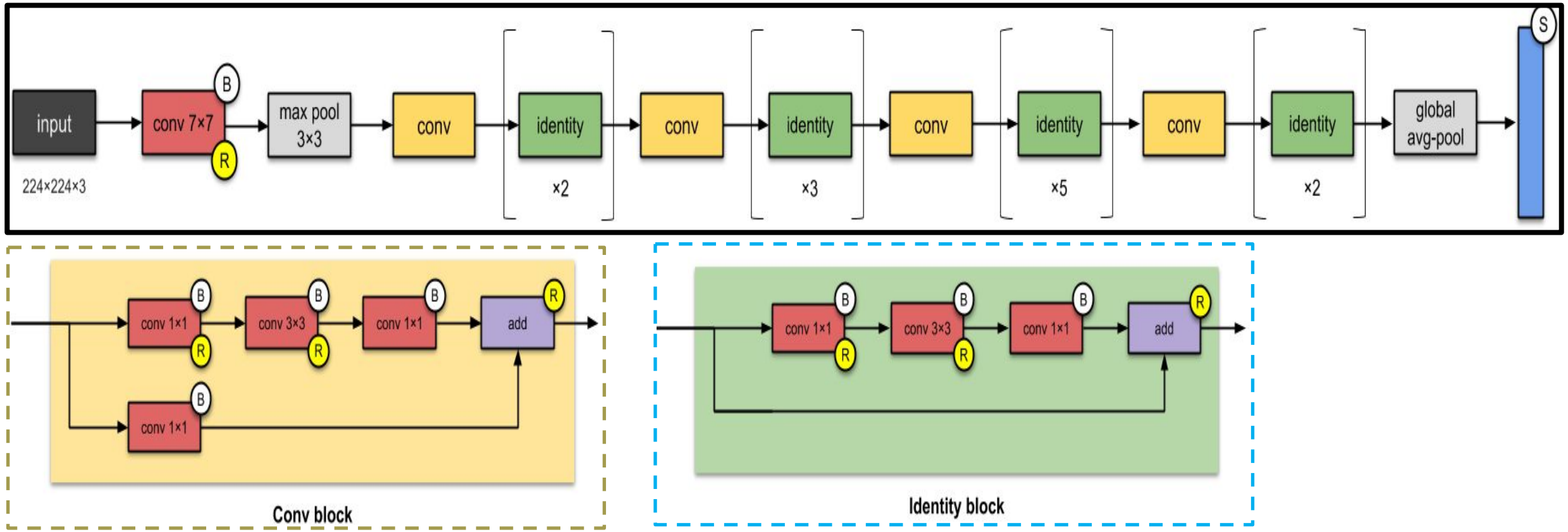
Novel: First Network to have batch normalization.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

ResNet-50 (2015)

- Last few CNNs models, It was seen that increasing number of layers in the design, and achieving better performance.
- With the network depth increasing, accuracy gets saturated and then degrades rapidly.
- Hence, Kaiming et al. from Microsoft Research addressed this problem with ResNet using skip connections (a.k.a. shortcut connections, residuals), while building deeper models.
- ResNet is one of the early adopters of batch normalization.
- ResNet-50 has **26M** parameters.
- The basic building block for ResNets are the **conv** and **identity blocks.**

ResNet-50 (2015)…



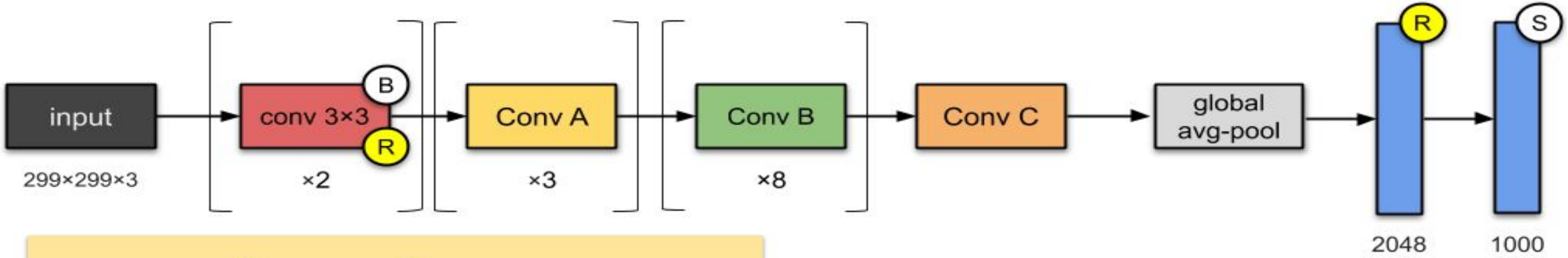
Novel: a) *Popularised* skip connections **b)** Designing even deeper CNNs (up to 152 layers) without compromising model's generalization power **c)** Among the first to use batch normalization

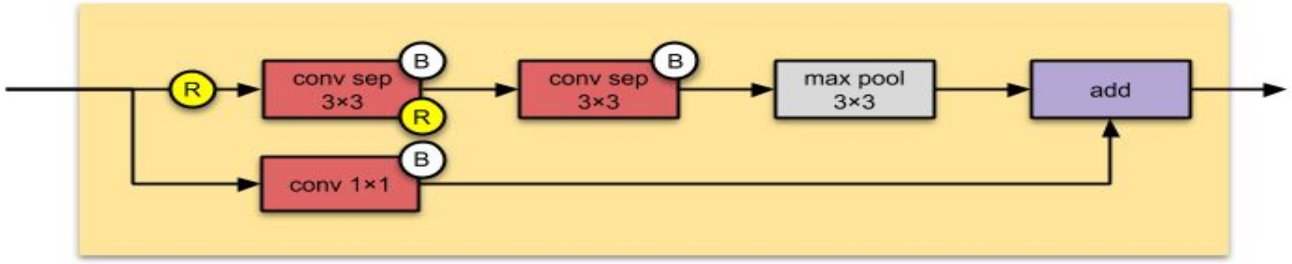
He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

Xception (2016)

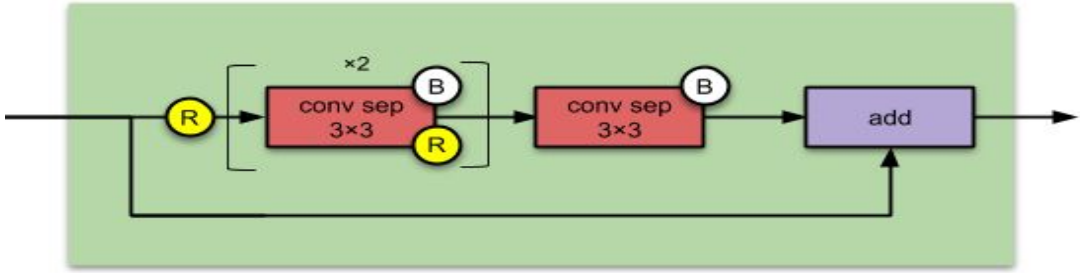
- Xception is an adaptation from Inception, where the Inception modules have been replaced with depthwise separable convolutions.
- It has also roughly the same number of parameters as Inception-v1 (**23M**).
- Xception takes the Inception hypothesis to an *eXtreme.*
  - Firstly, cross-channel (or cross-feature map) correlations are captured by 1×1 convolutions.
  - Consequently, spatial correlations within each channel are captured via the regular 3×3 or 5×5 convolutions
- The idea of an extreme means performing 1×1 to *every* channel, then performing a 3×3 to *each* output.
- This is identical to replacing the Inception module with depthwise separable convolutions.
- **Novel:** Introduced CNN based entirely on depthwise separable convolution layers.

Xception (2016)

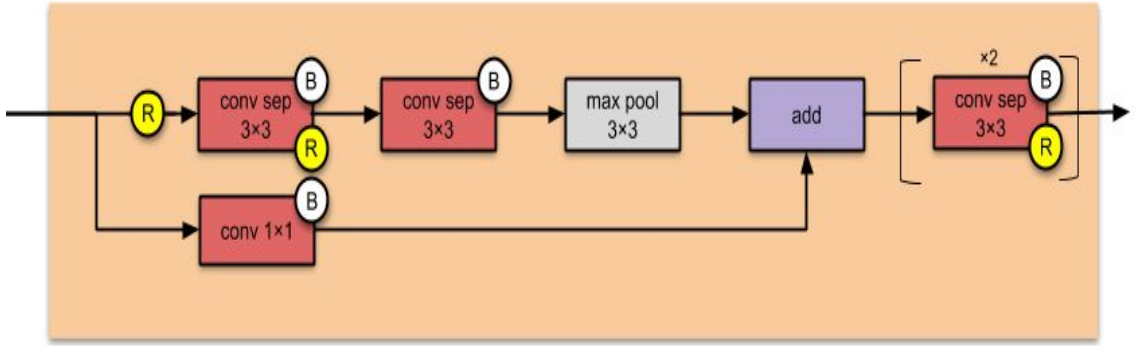




Conv A



Conv B



Conv C

Inception V4 (2016)

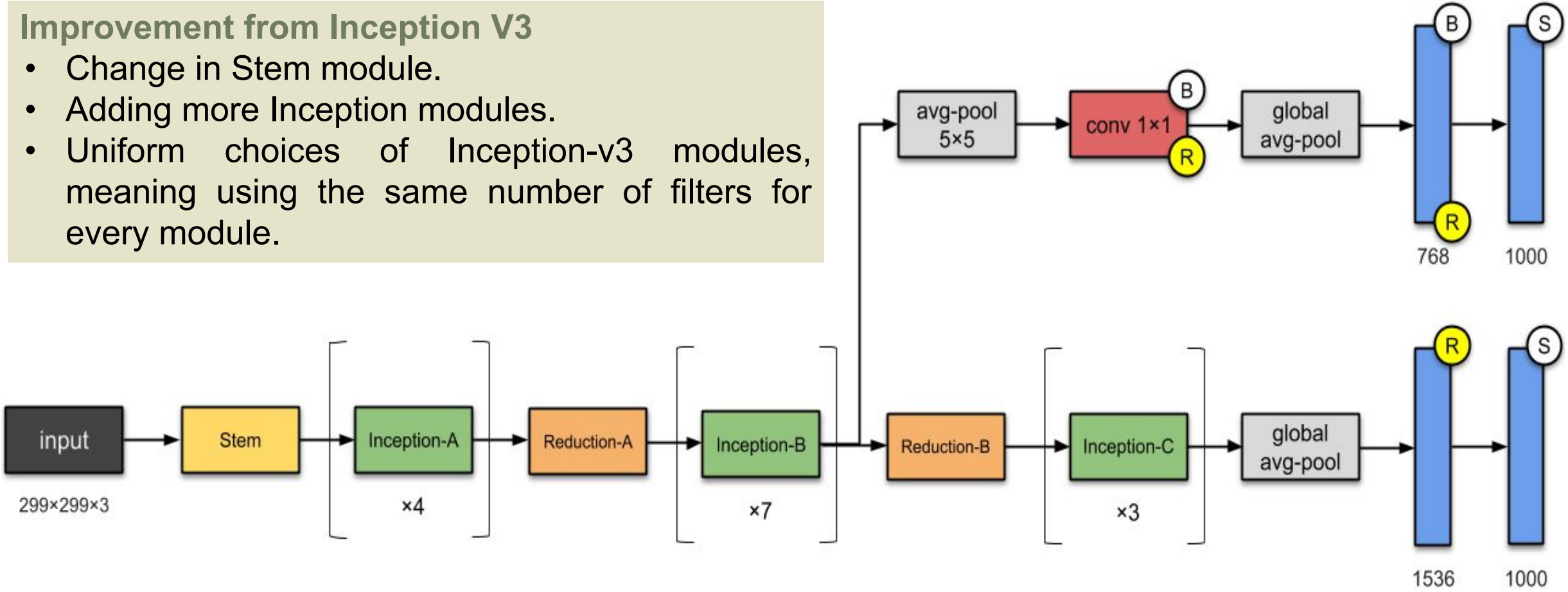
- The researchers from Google strike again with Inception-v4, **43M** parameters.
- Again, this is an improvement from Inception-v3.
- The main difference is the Stem group and some minor changes in the Inception-C module.
- The authors also “made uniform choices for the Inception blocks for each grid size.”
- They also mentioned that having “residual connections leads to dramatically improved training speed.”
- Inception-v4 works better because of increased model size.

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning." In *Thirty-first AAAI conference on artificial intelligence*. 2017.

Inception V4 (2016)…

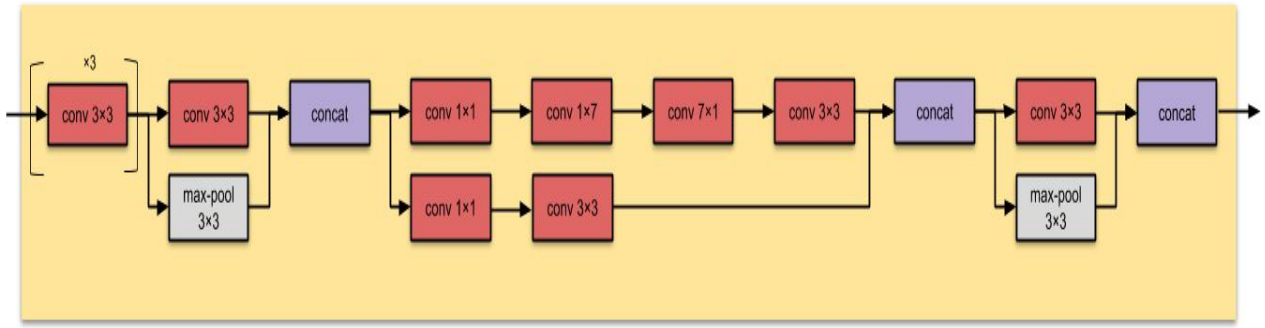
Improvement from Inception V3

- Change in Stem module.
- Adding more Inception modules.
- Uniform choices of Inception-v3 modules, meaning using the same number of filters for every module.

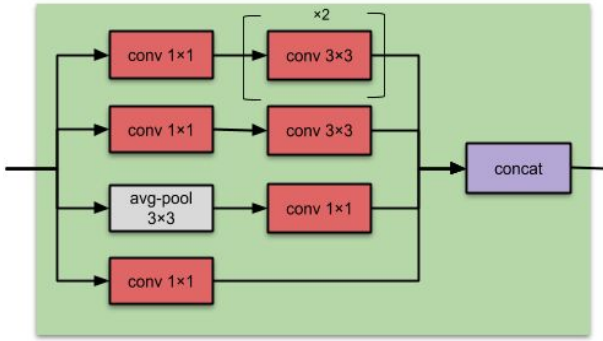


Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning." In *Thirty-first AAAI conference on artificial intelligence*. 2017.

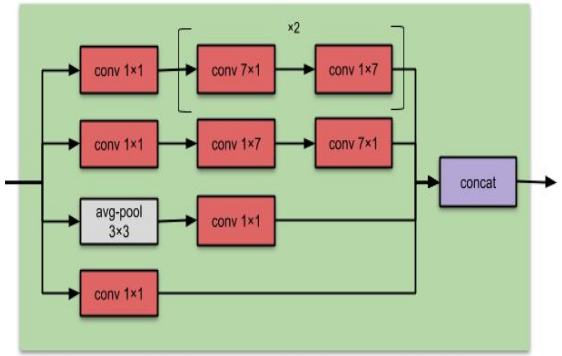
Inception V4 (2016)…



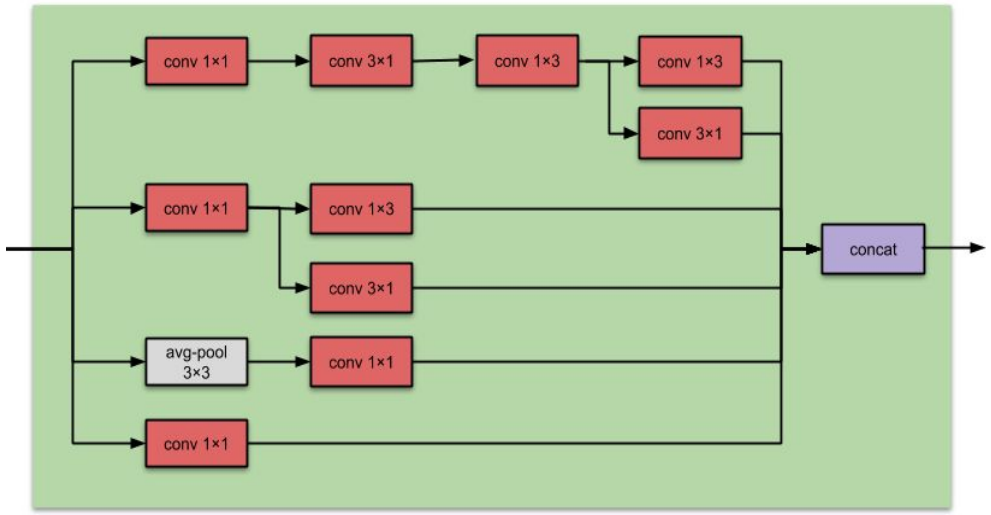
Stem



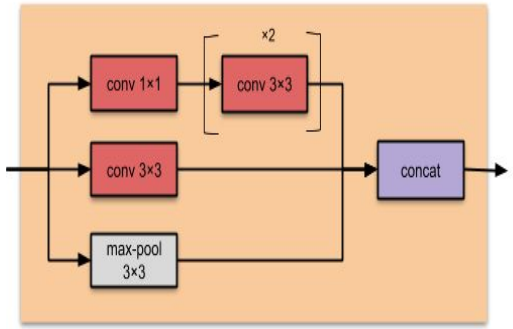
Inception-A



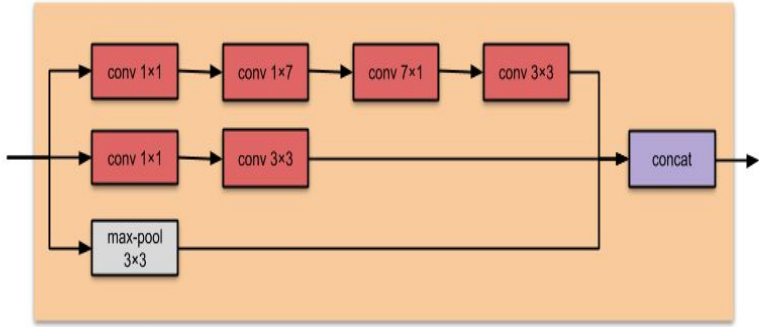
Inception-B



Inception-C



Reduction-A



Reduction-B

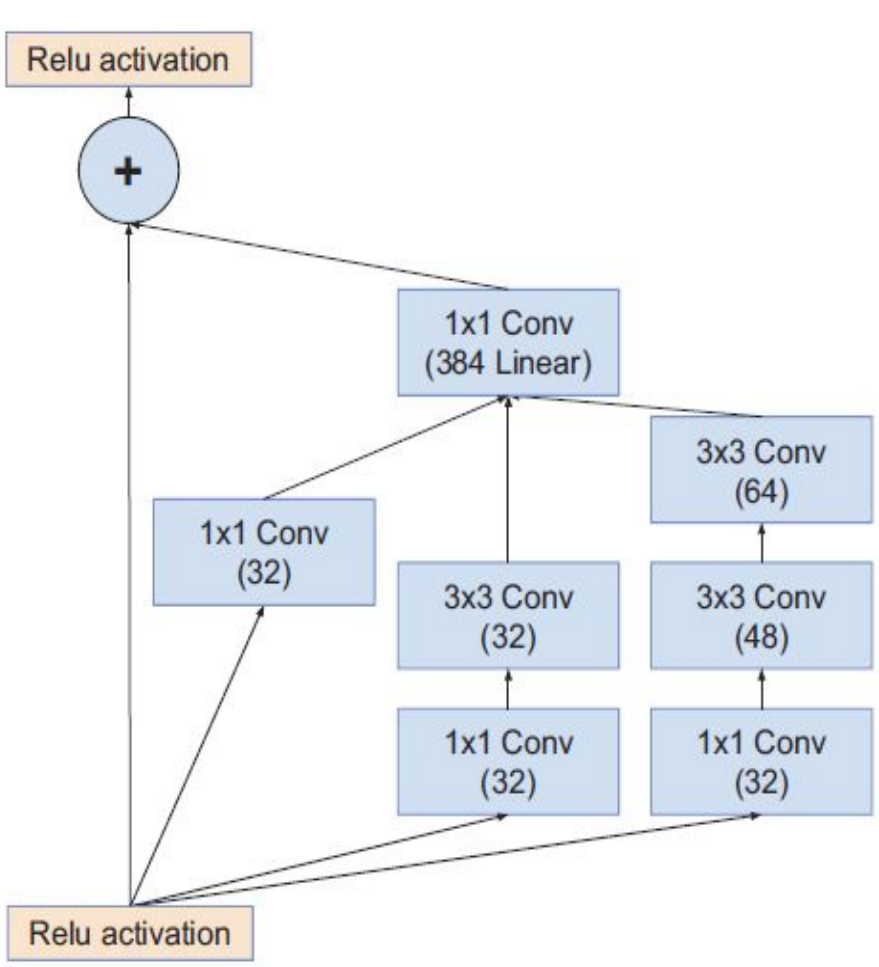
Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning." In *Thirty-first AAAI conference on artificial intelligence*. 2017.

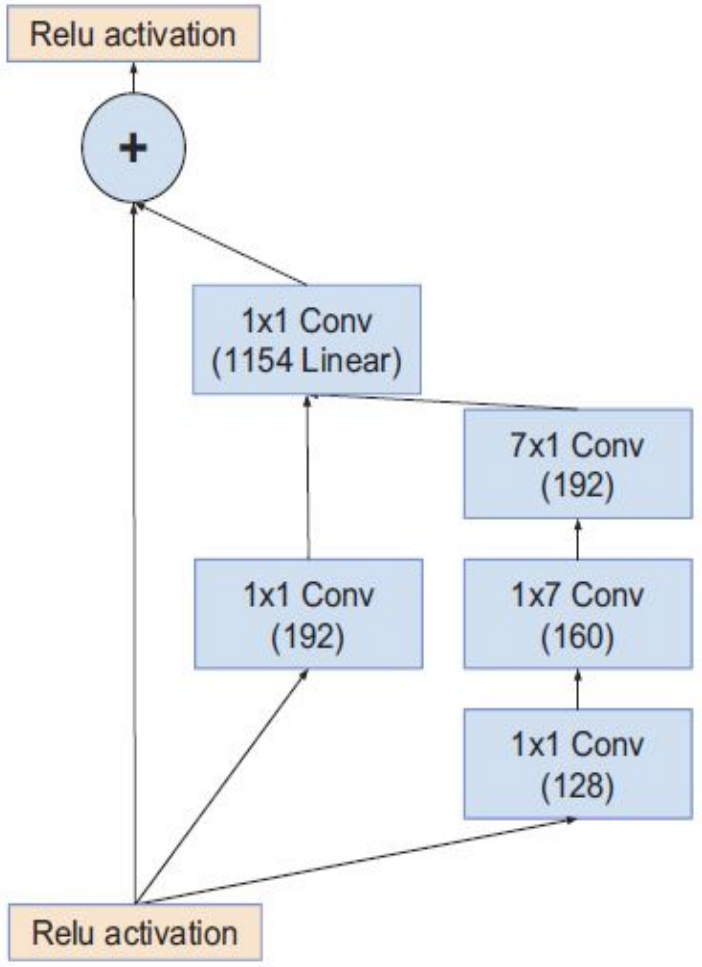
Inception-ResNet-V2 (2016)

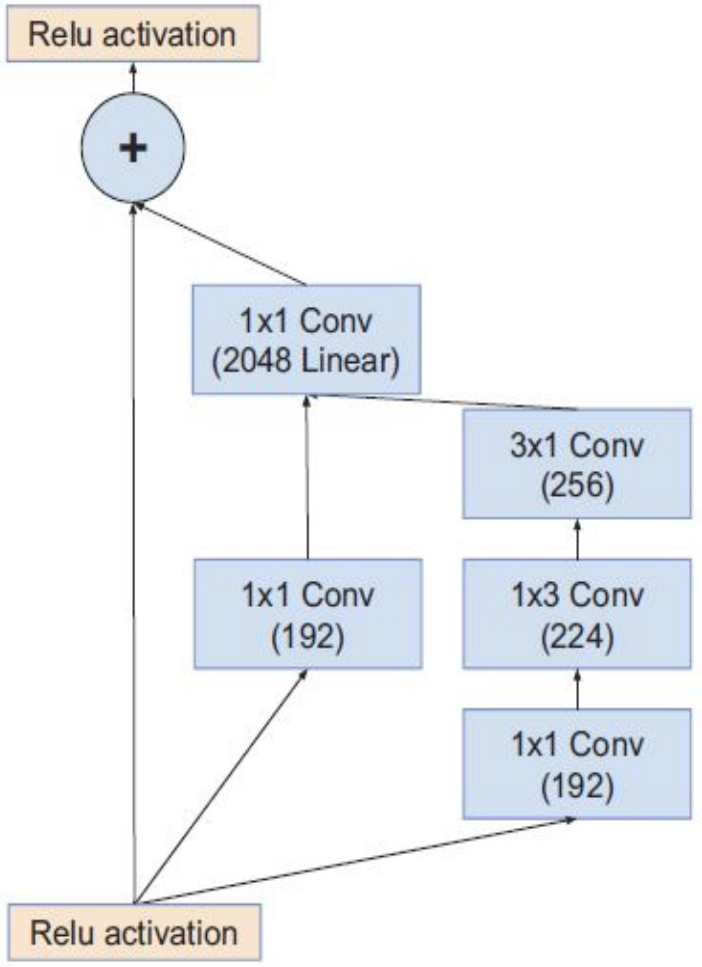
- In the same paper as Inception-v4, the same authors also introduced Inception-ResNets — a family of Inception-ResNet-v1 and Inception-ResNet-v2.
- The latter member of the family has **56M** parameters.
- **What's improved from the previous version, Inception-v3?**
  - Converting Inception modules to *Residual Inception blocks.*
  - Adding more Inception modules.
  - Adding a new type of Inception module (Inception-A) after the Stem module.

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning." In *Thirty-first AAAI conference on artificial intelligence*. 2017.

Inception-ResNet-V2 (2016)…



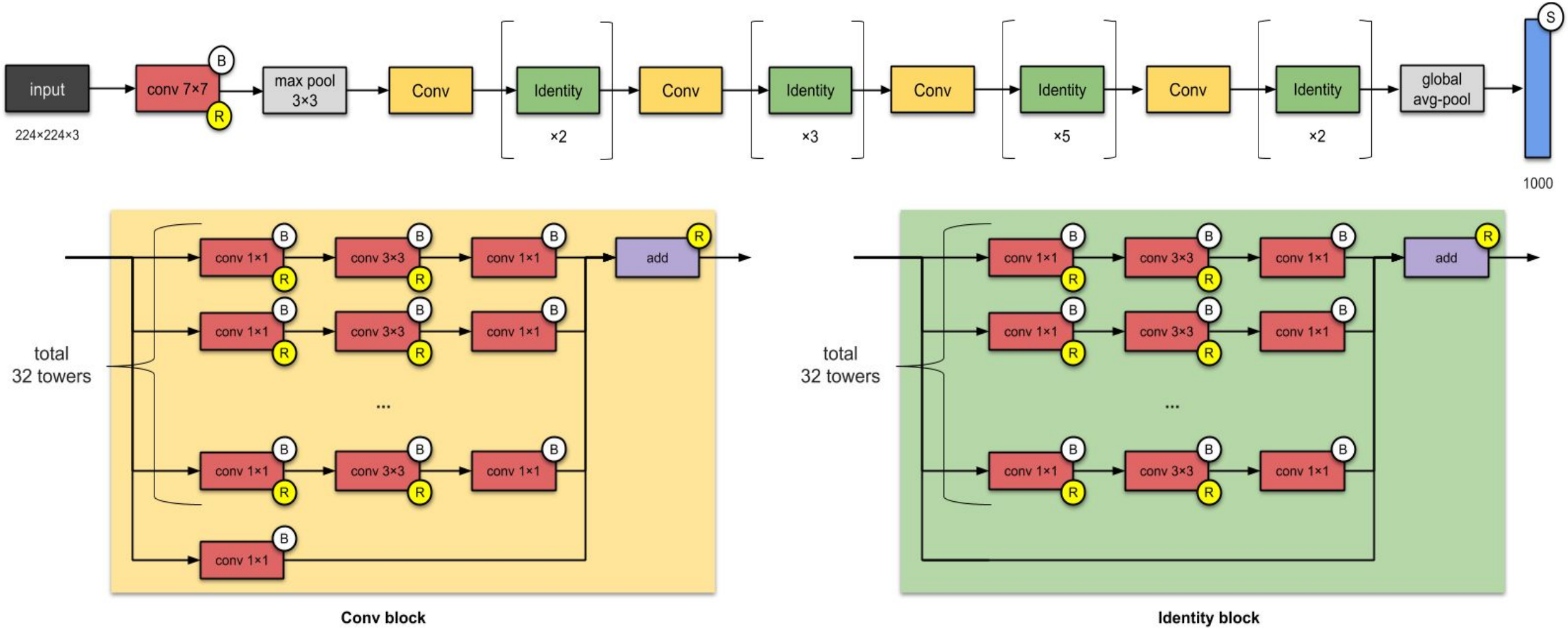




ResNeXt-50 (2017)

- ResNeXt-50 has **25M** parameters (ResNet-50 has 25.5M).
- What's different about ResNeXts is the adding of parallel towers/branches/paths within each module, as seen above indicated by 'total 32 towers.'
- The model name, ResNeXt, contains Next.
- It means the *next* dimension, on top of the ResNet.
- This next dimension is called the "***cardinality***" **dimension**. And ResNeXt becomes the **1st Runner Up of ILSVRC classification task**.
- **What's novel?**
  - Scaling up the number of parallel towers ("cardinality") within a module (well I mean this has already been explored by the Inception network…)

ResNeXt-50 (2017)…



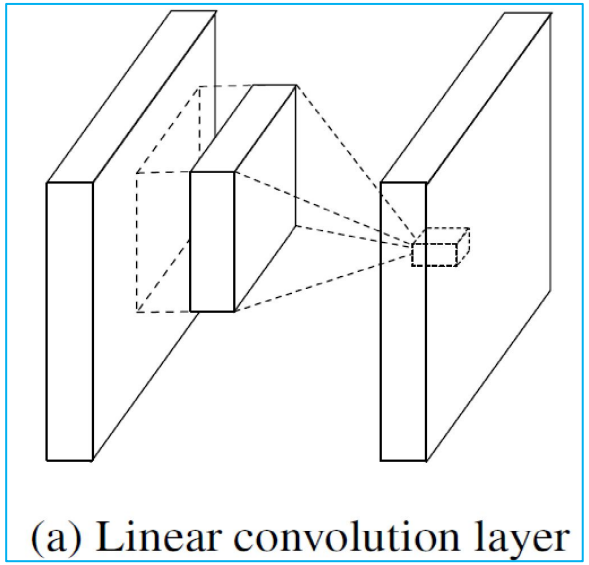
Xie, Saining, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated residual transformations for deep neural networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492-1500. 2017.

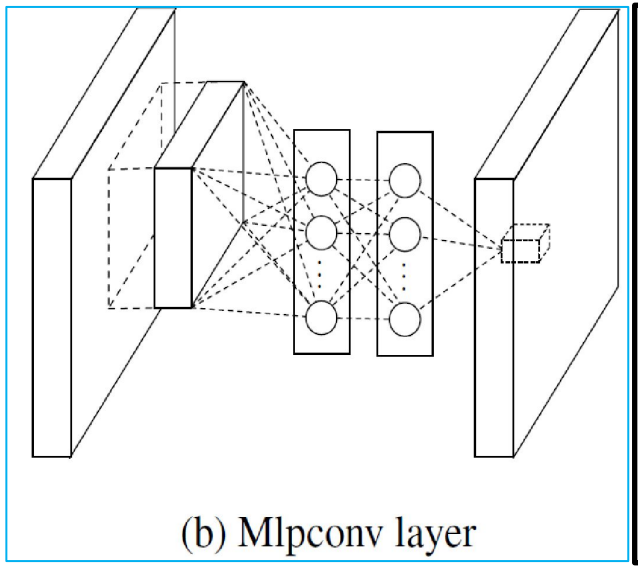
Network In Network (2014)

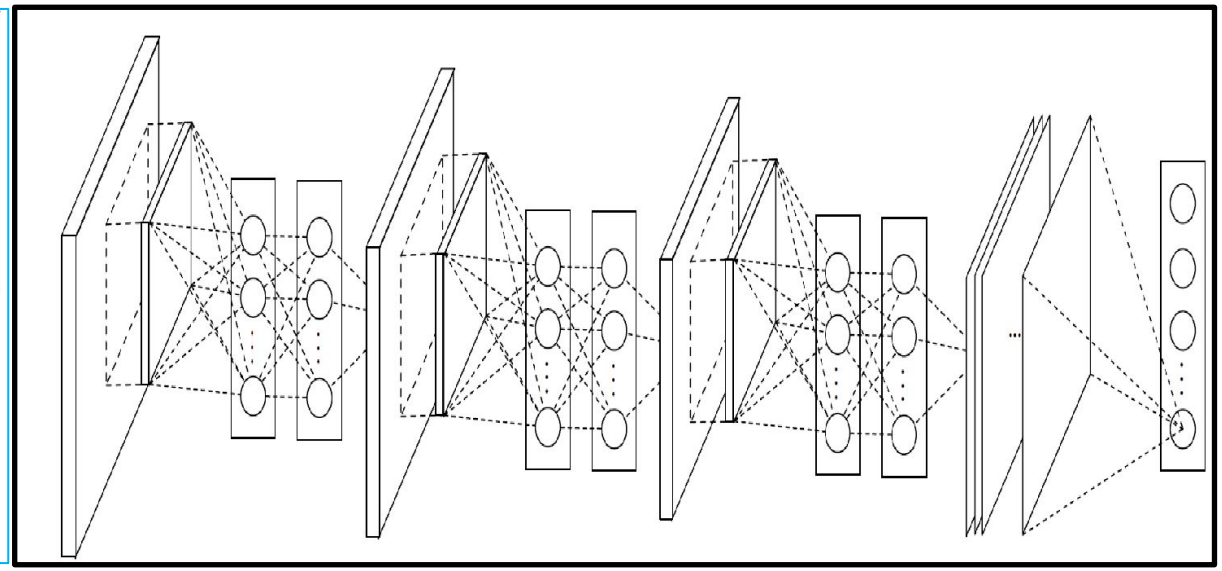
- Recall that in a convolution, the value of a pixel is a linear combination of the weights in a filter and the current sliding window.
- The authors proposed that instead of this linear combination, let's have a mini neural network with 1 hidden layer.
- This is what they coined as **Mlpconv**.
- So what we're dealing with here is a (simple 1 hidden layer) network in a (convolutional neural) network.
- This idea of Mlpconv is likened to 1×1 convolutions, and became the main feature for Inception architectures.
- **What's novel?**
  - MLP convolutional layers, 1×1 convolutions
  - Global average pooling (taking average of each feature map, and feeding the resulting vector into the softmax layer).

Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).

Network In Network (2014)…







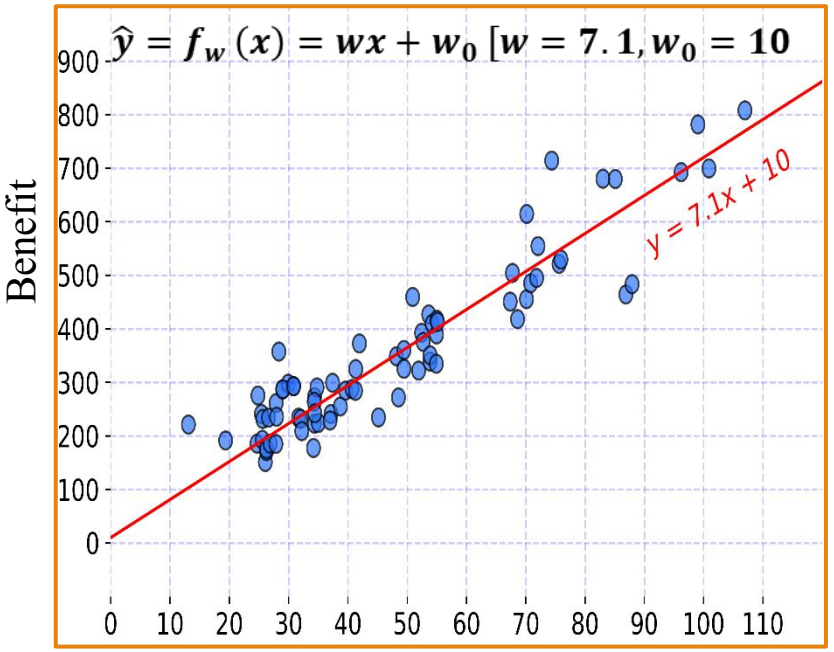
NINs

- The overall structure of Network In Network. NINs include the stacking of three mlpconv layers and one global average pooling layer

Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).

Cost Function

- To understand the cost function, let us consider a simple and most popular ML algorithm Linear Regression.
- Linear regression is used to estimate linear relationships between continuous or/and categorical data and a continuous output variable.
- Ex. $y = w^T x$. Here x is independent variable while y is dependent variable and w is weight value, which may be a slope.
- The quality of regression model is measured in terms of Mean Square Error (MSE), which is defined as: $\frac{1}{2n}\sum_{i=1}^{n}(\hat{y}^{(i)} - y^{(i)})^2$.

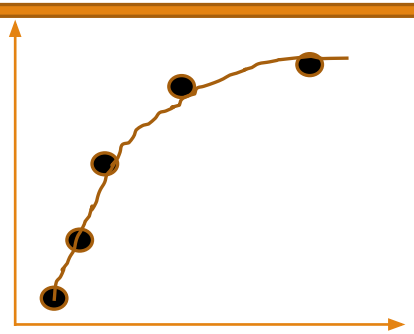


Company

Cost Function…

- To have best LR model the MSE must be minimum and it denoted as

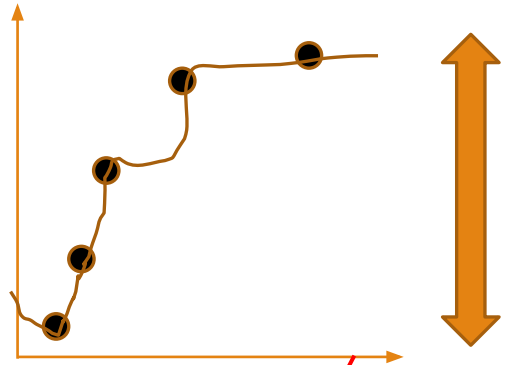
$$\underbrace{min}_{w} \frac{1}{2n} \sum_{i=1}^{n} (\hat{y}^{(i)} - y^{(i)})^2$$



$$f_w(x) = w_0 + w_1 x + w_2 x^2$$

- Here, it is a optimization problem, hence a objective function can be defined as

$$J(w, w_0) = \frac{1}{2n} \sum_{i=1}^{n} (f_w(x)^{(i)} - y^{(i)})^2$$



$$f_w(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

- It is also called as cost function. Let we optimize it by penalize and make w_3, and w_4 very small.

$$\underbrace{min}_{w} \frac{1}{2n} \sum_{i=1}^{n} (f_w(x)^{(i)} - y^{(i)})^2 + 100 w_3^2 + 1000 w_4^2$$

A minimum value can be obtained by keeping w_3 and w_4 equal to zero.

Regularization in ML

- One of the major aspects of training your machine learning model is avoiding overfitting.
- *The model will have a low accuracy if it is overfitting.*
- This happens because model is trying too hard to capture the noise in your training dataset. ***By noise we mean the data points that don't really represent the true properties of your data, but random chance***.
- Learning such data points, makes your model more flexible, at the risk of overfitting.

Regularization in ML…

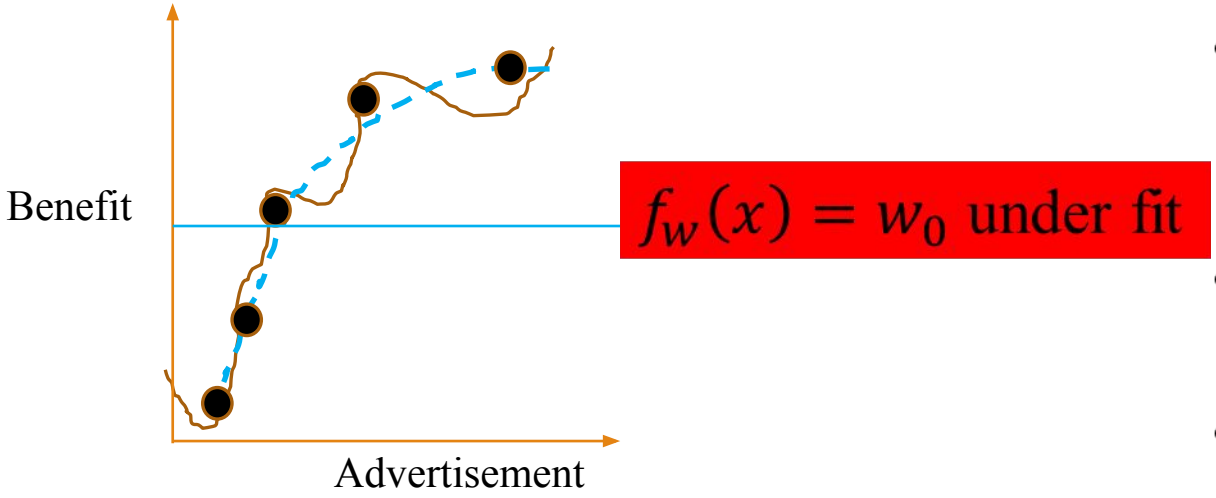
- Small value of parameters $w_1, w_2, w_3, \ldots w_m$
  - ✓ Model is simpler
  - ✓ Less prone to overfitting
- **Company**: Benefit and Advertisement
  - ✓ Features $x_1, x_2, x_3, x_4 \ldots . x_n$
  - ✓ Parameters $w_1, w_2, w_3, \ldots w_m$
- We don't know which parameters has to penalize, hence we modified the cost function as

Regularization Term

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(f_w(x)^{(i)} - y^{(i)})^2 + \lambda\sum_{i=1}^{m} {w_i}^2$$

Regularization Parameter

Regularization in ML…



$f_w(x) = w_0$ under fit

$$f_w(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(f_w(x)^{(i)} - y^{(i)})^2 + \lambda \sum_{i=1}^{m} {w_i}^2$$

- If λ is more then it may be **over fit** and parameters values are more then it will be higher order polynomial.
- If parameter values are very low or equal to zero then it will be **under fit**.
- $f_w(x) = w_0$ and rest parameters are zero.
- Hence, the regularization parameter plays an import role of any machine learning model.

https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#e276

Thank You

Contact: dinesh@dtu.ac.in
Mobile: +91-9971339840