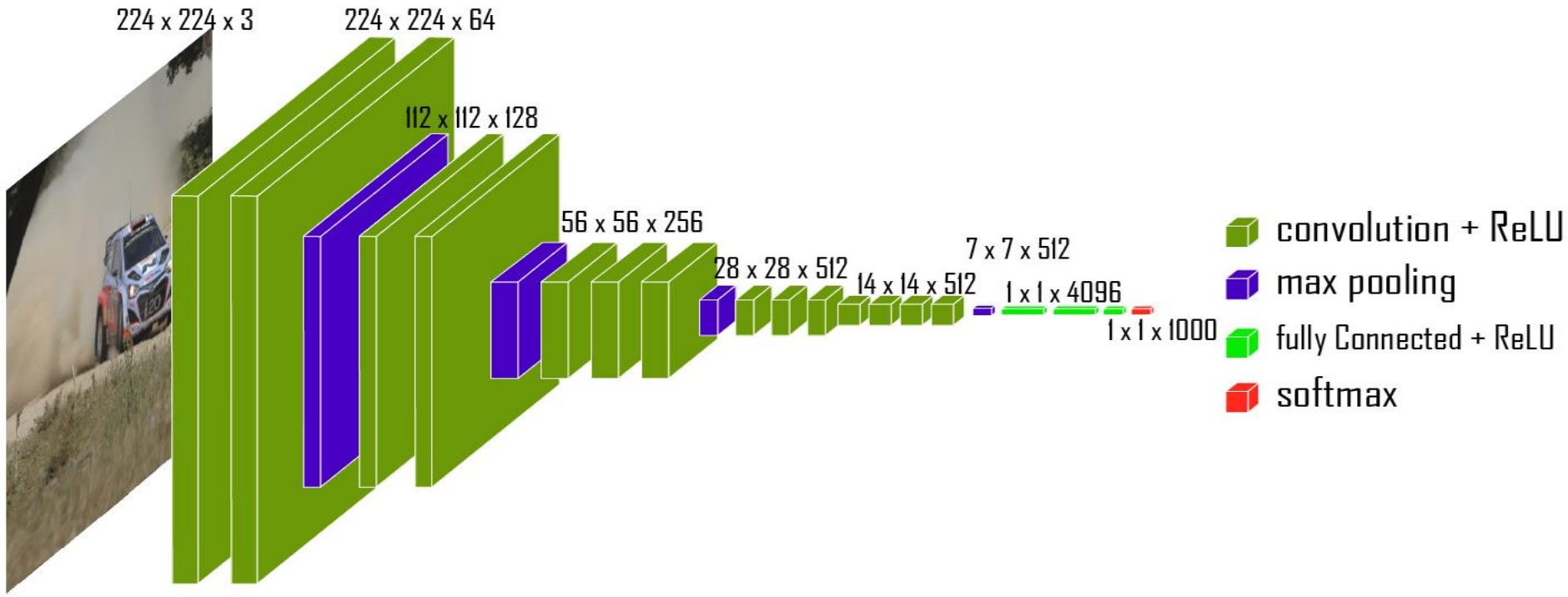
Some Popular CNN Embedding models

VGG-16

Original paper: https://arxiv.org/abs/1409.1556

VGG-16

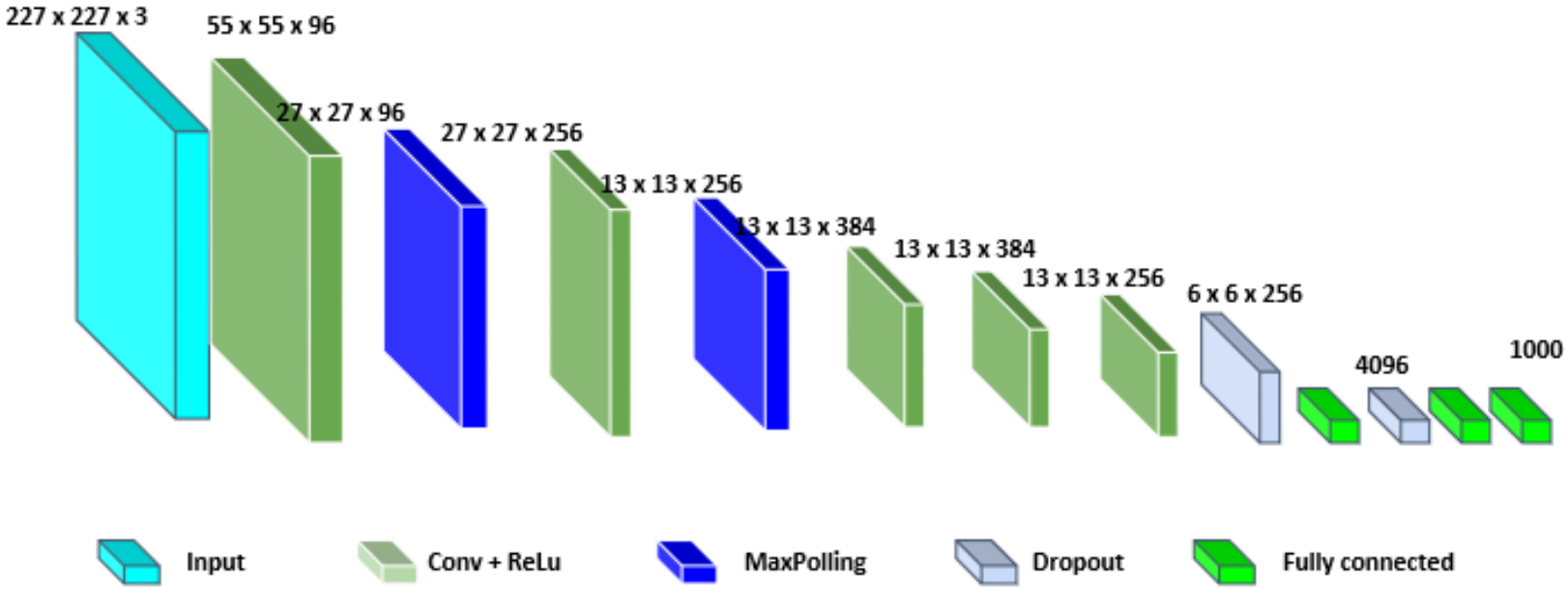


1. VGG-16 architecture consists of 12 convolutional layers, and 4 fully connected layers.
2. 138 millions parameters.

Original paper: https://arxiv.org/abs/1409.1556

Layer	#filters	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	224 x 224 x 3	-
2 x Conv	64	3 x 3	1	224 x 224 x 64	ReLu
MaxPool	-	3 x 3	1	224 x 224 x 64	-
2 x Conv	128	5 x 5	1	112 x 112 x 128	ReLu
MaxPool	-	3 x 3	2	56 x 56 x 128	-
2 x Conv	256	3 x 3	1	56 x 56 x 256	ReLu
MaxPool	-	3 x 3	2	28 x 28 x 256	-
3 x Conv	512	3 x 3	1	28 x 28 x 512	ReLu
MaxPool	-	3 x 3	2	14 x 14 x 512	-
3 x Conv	512	3 x 3	1	14 x 14 x 512	ReLu
MaxPool	-	3 x 3	2	7 x 7 x 512	-
FC1	-	-	-	25088	ReLu
FC2	-	-	-	4096	ReLu
FC3	-	-	-	4096	ReLu
Output	-	-	-	10000	Softmax

AlexNet



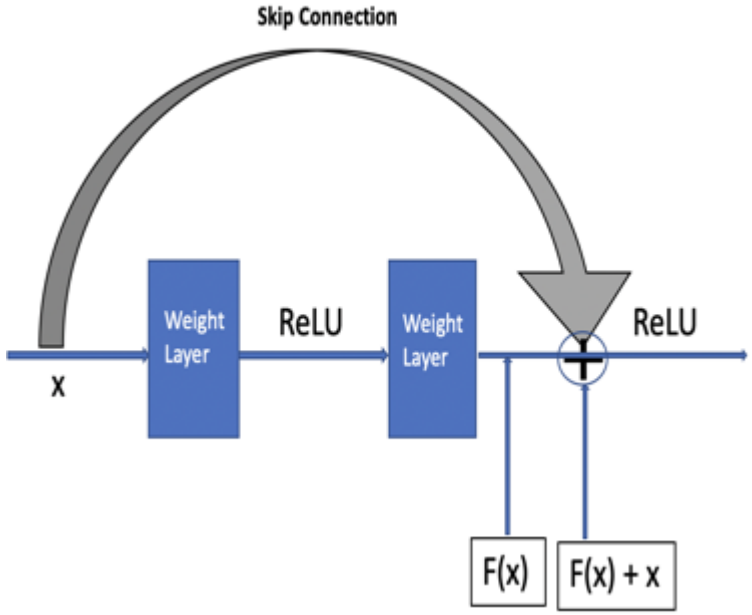
1. AlexNet architecture consists of 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer.
2. Each convolutional layer consists of convolutional filters and a nonlinear activation function ReLU.
3. 61 millions parameters

Layer	#filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-	-	-	227 x 227 x 3	-
Conv 1	96	11 x 11	4	-	55 x 55 x 96	ReLu
MaxPool 1	-	3 x 3	2	-	27 x 27 x 96	-
Conv 2	256	5 x 5	1	2	27 x 27 x 256	ReLu
MaxPool 2	-	3 x 3	2	-	13 x 13 x 256	-
Conv 3	384	3 x 3	1	1	13 x 13 x 384	ReLu
Conv 4	384	3 x 3	1	1	13 x 13 x 384	ReLu
Conv 5	256	3 x 3	1	1	13 x 13 x 256	ReLu
MaxPool 2	-	3 x 3	2	-	6 x 6 x 256	-
Dropout 1	rate=0.5	-	-	-	6 x 6 x 256	-
FC1	-	-	-	-	4096	ReLu
Dropout 1	rate=0.5	-	-	-	4096	-
FC2	-	-	-	-	4096	ReLu
FC3	-	-	-	-	1000	Softmax

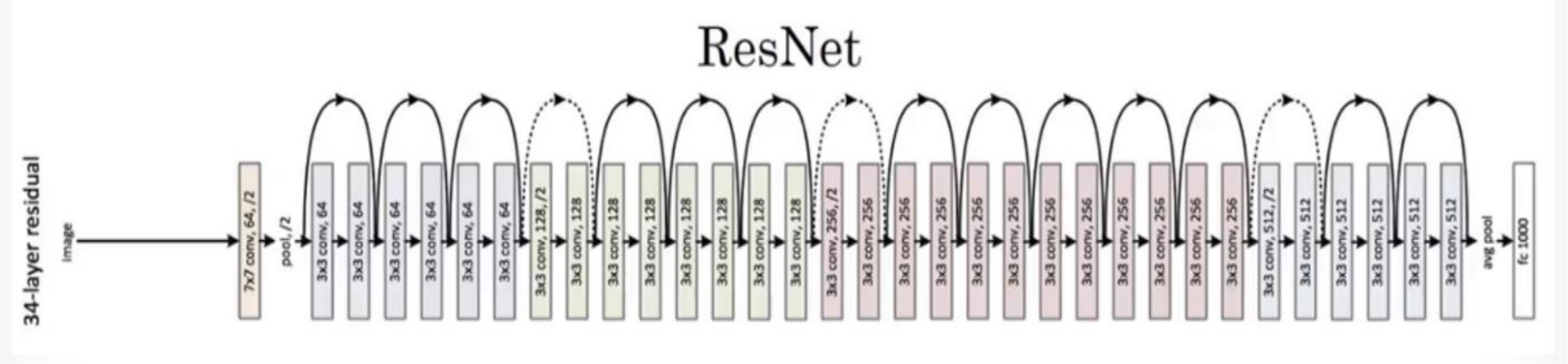
ResNet

Residual Block

- **Residual blocks** is that each layer is fed to the next layer of the network and also directly to the next layers skipping between a few layers in between.
- Residual blocks allow you to train much deeper neural networks.

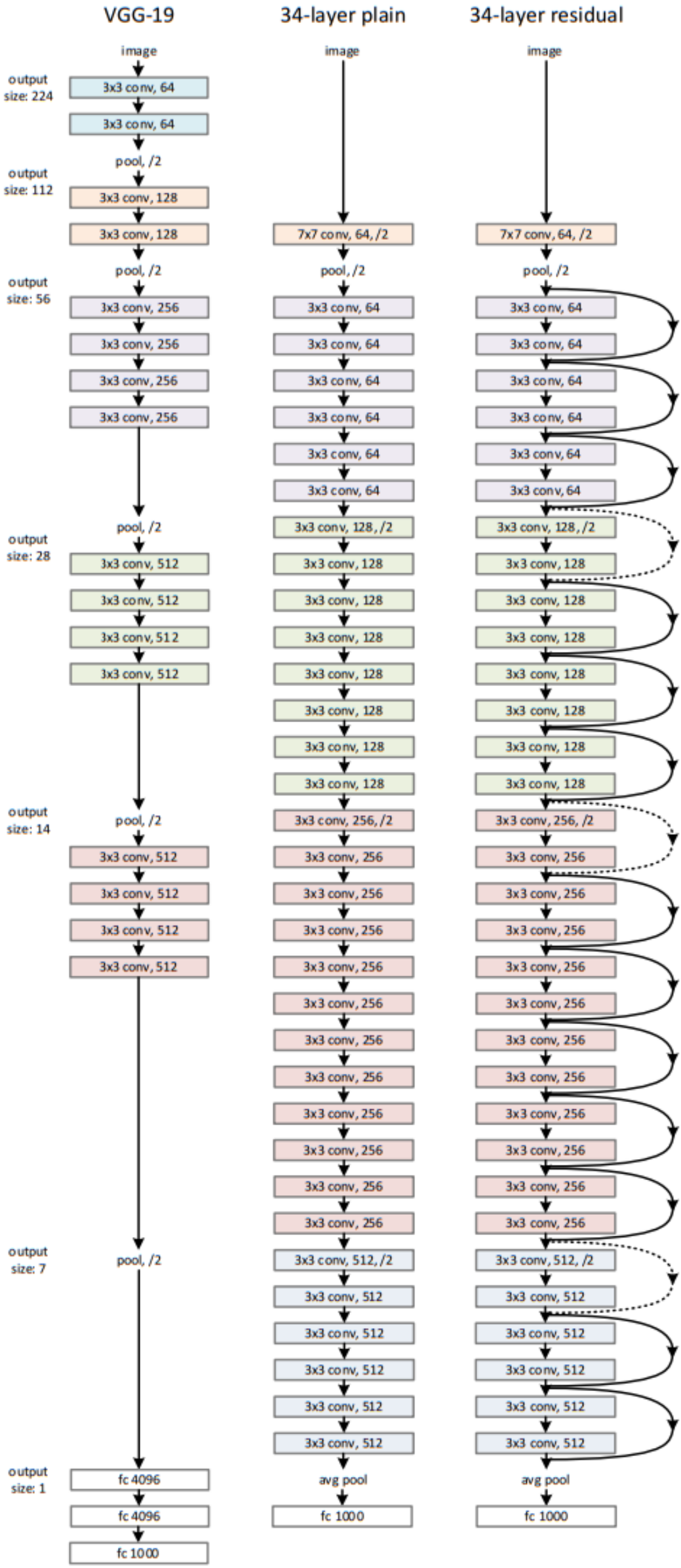


ResNet-34



ResNet-34 is inspired by VGG-19. Its difference is it has residual connections.

63.5 Million parameters



Original paper: https://arxiv.org/abs/1512.03385v1

Inception

All the networks are developed to go deeper, but Inception introduce the concept to go wider. Its main component is the ***inception module***

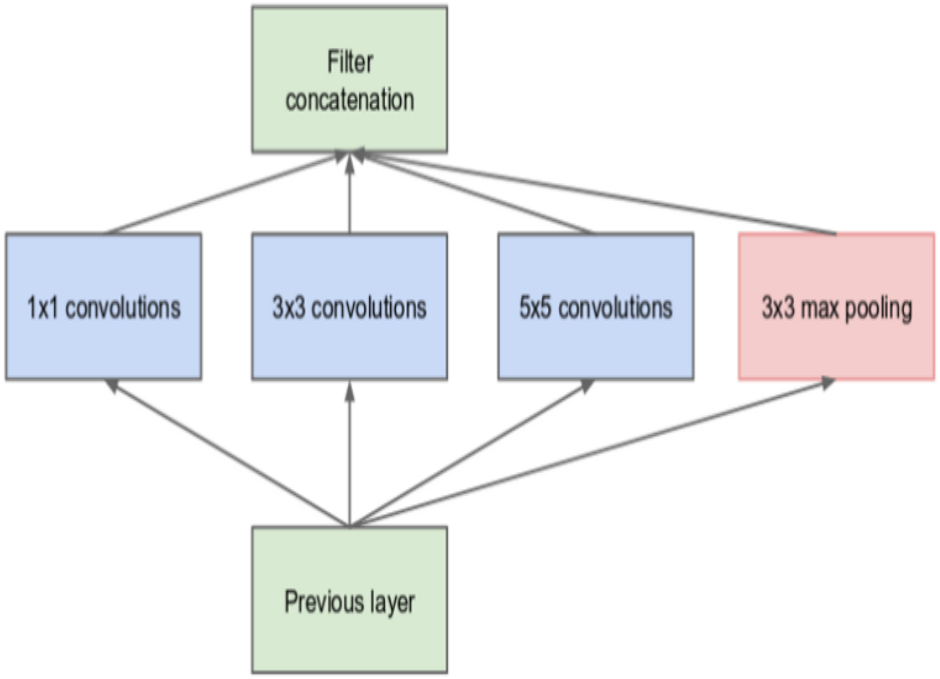
Inception module

- An Inception Module consists of the following components
  - *Input layer*
  - *filters with multiple sizes*
  - *Max pooling layer*
  - *Concatenation layer*

- Among the filters, it uses 1x1 because it learns patterns across the depth of the input

- Filters of other size (it used 3x3 and 5x5) enable the network to learn various spatial patterns at different scales as a result of the varying conv filter sizes.

Inception v1(GoogLeNet)

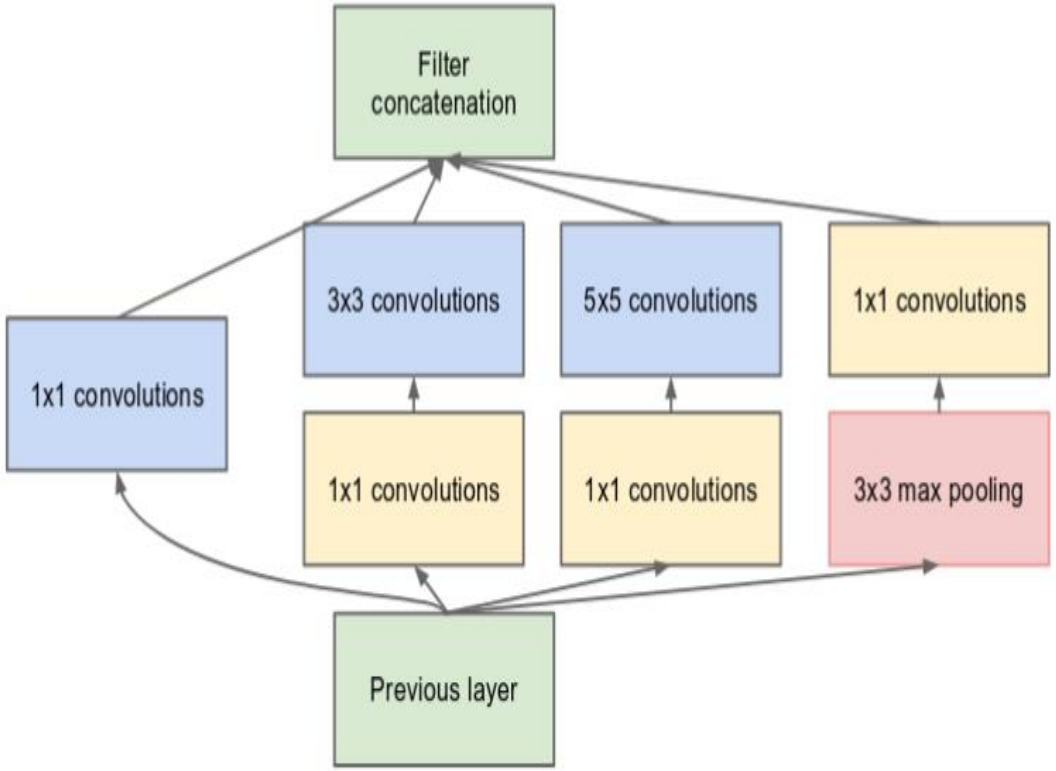


(a) Inception module, naïve version

- The naive inception module allows for the utilisation of varying convolutional filter sizes to learn spatial patterns at different scales. But the problems with the naive inception module is that it can quickly accumulate substantial computational cost.
- The benefits of using the Inception architecture is lost to the massive computational disadvantage.

Original paper: https://www.arxiv.org/pdf/1512.00567v1.pdf

Inception v1(GoogLeNet)

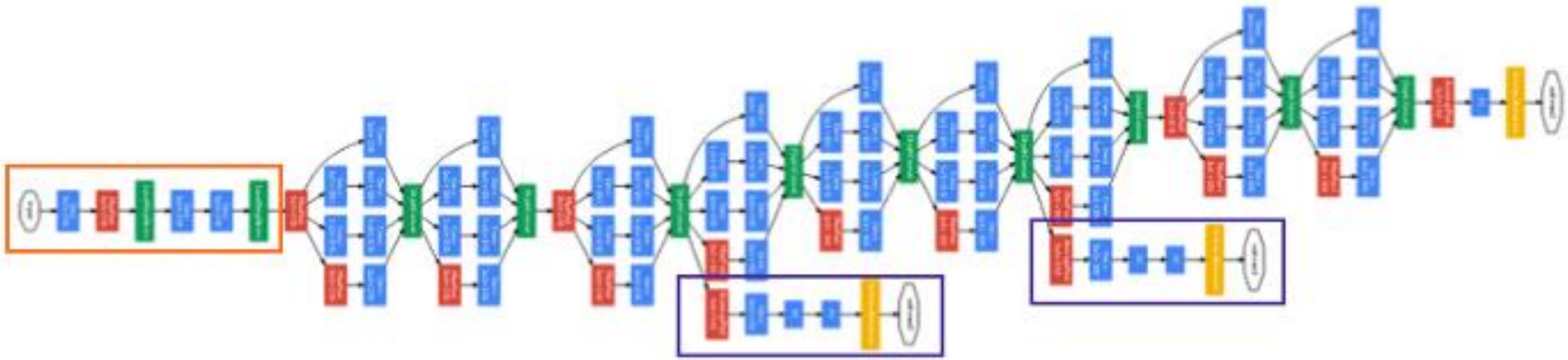


(b) Inception module with dimension reductions

The challenge in naive inception module is addressed by introducing 1x1 conv as shown in the figure

Original paper: https://www.arxiv.org/pdf/1512.00567v1.pdf

Inception v1(GoogLeNet)



Original paper: https://www.arxiv.org/pdf/1512.00567v1.pdf

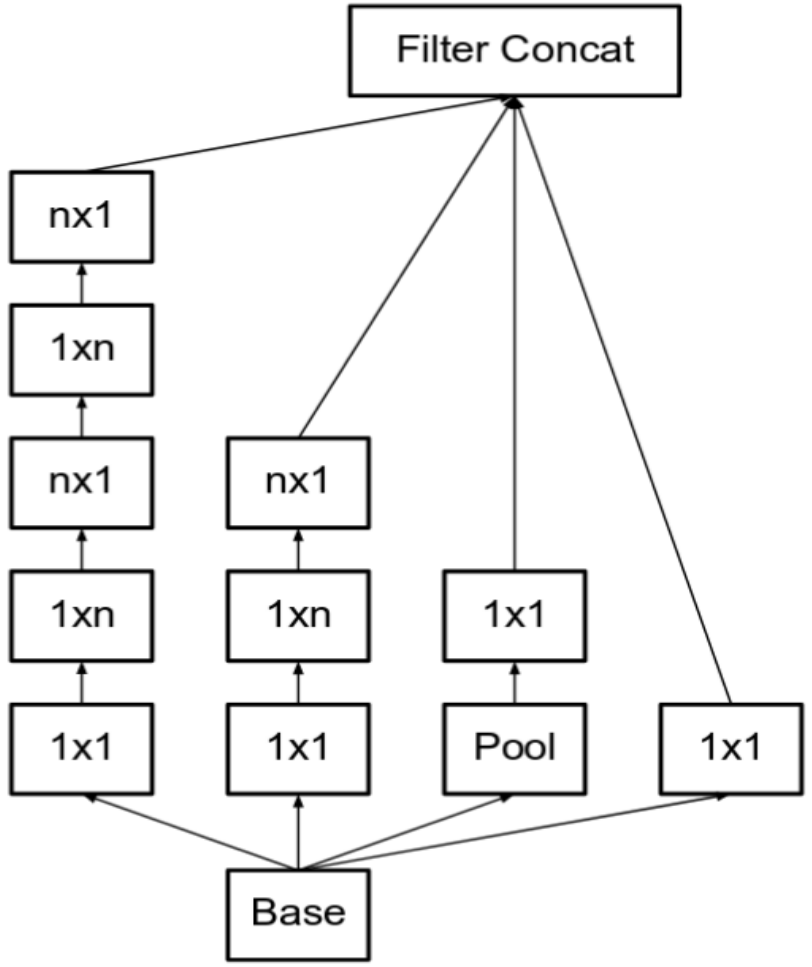
Inception v2

- Inception v2 and Inception v3 were presented in the same paper.
- The authors proposed a number of upgrades to the version 1 which increased the accuracy and reduced the computational complexity.

Upgrades in v1 to get v2

- Reduce time: Factorize convolutions of filter size nxn to a combination of 1xn and nx1 convolutions as shown in the figure. For example, a 3x3 convolution is equivalent to first performing a 1x3 convolution, and then performing a 3x1 convolution on its output.

- Increase accuracy: The filter 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.



Original paper: https://arxiv.org/pdf/1512.00567v3.pdf

Inception v3

Inception v3 is the same architecture as v2 (minor changes) with different training algorithm (RMSprop, label smoothing regularizer, adding an auxiliary head with batch norm to improve training etc).

Inception v4

Ity tries to make the modules more uniform. The authors noticed that some of the modules were more complicated than necessary. Removing and modification tho those unnecessary complicated modules enable the model to boost performance

Some other CNN

- Xception
- MobileNet
- DenseNet
- EfficientNet

Pre-trained CNN models

- Pre-trained model are used in transfer learning (Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.)
- Most of the pretrained version of all the models (except for AlexNet and UNet) are provided by Keras. https://keras.io/api/applications/ (Pretrained AlexNet is available in pytorch https://pytorch.org/hub/pytorch_vision_alexnet/)
- All of them are trained on ImageNet data.
- Imagenet is a large collection of image data containing 1000 categories of images.
- These pretrained models are capable of classifying any image that falls into these 1000 categories of images.
- Some well-known articles discussing transfer learning using pretrained model:

  - https://machinelearningmastery.com/use-pre-trained-vgg-model-classify-objects-photographs/
  - https://towardsdatascience.com/how-to-use-a-pre-trained-model-vgg-for-image-classification-8dd7c4a4a517
  - https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3