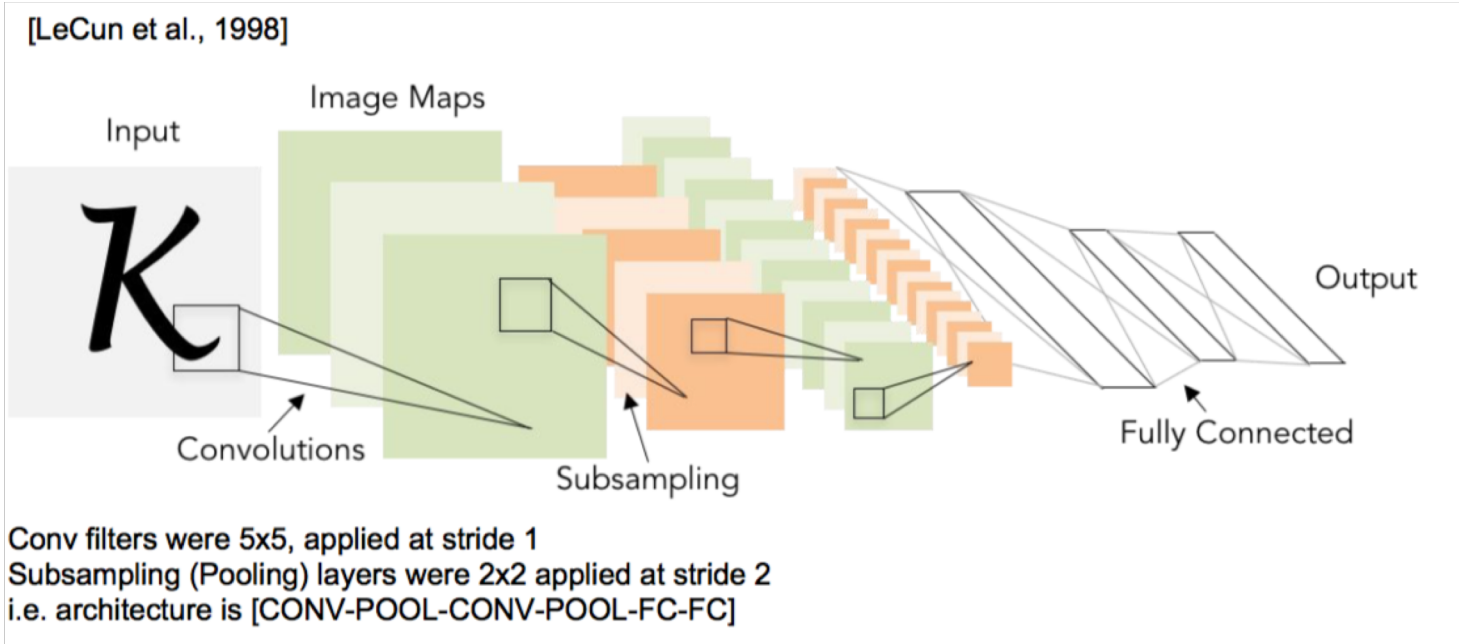
Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

History

What is convolutional neural network?

Why is it so important?

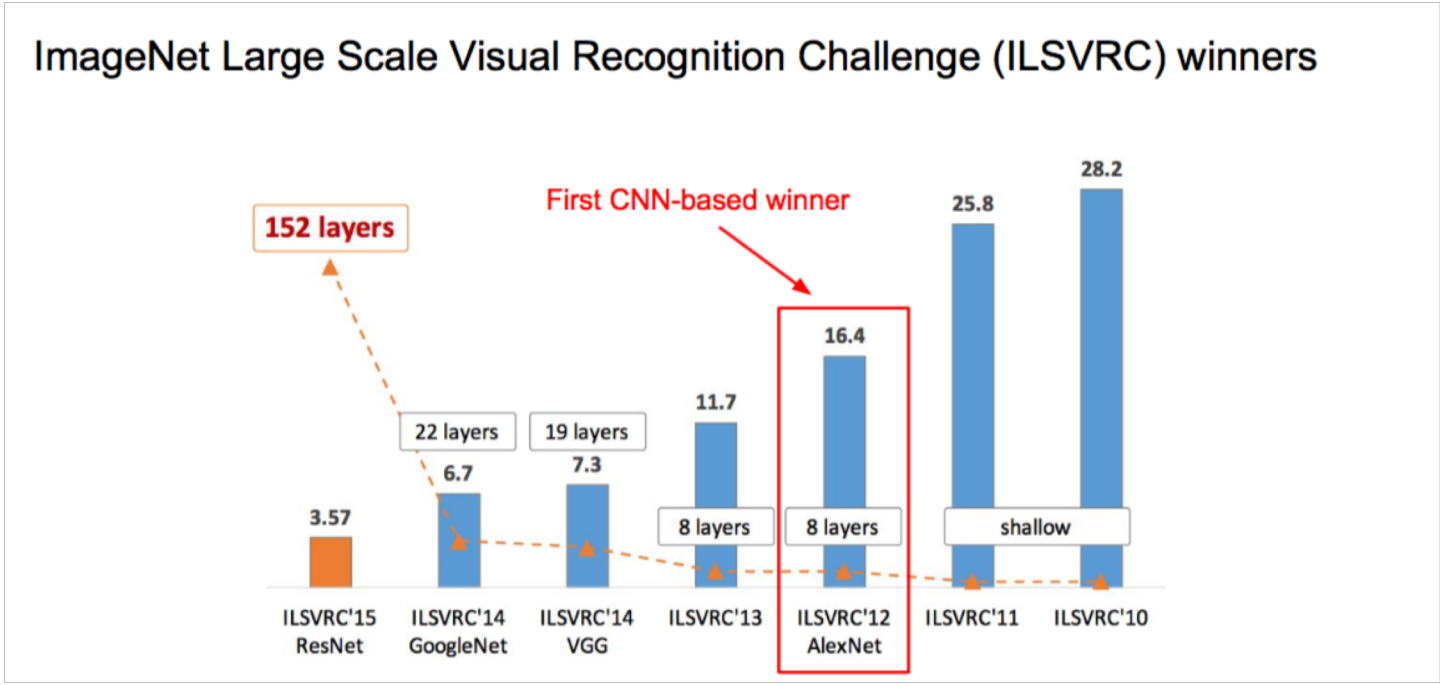
History



Link: 2D Visualization (http://scs.ryerson.ca/~aharley/vis/conv/flat.html)
3D Visualization (http://scs.ryerson.ca/~aharley/vis/conv/)

Taken from slides of CS231n course at Stanford University

History

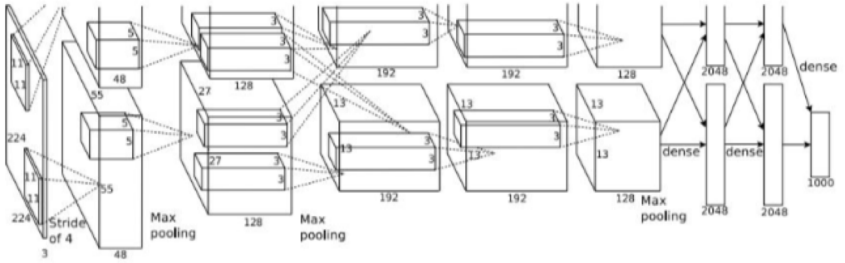


Taken from slides of CS231n course at Stanford University

History

Case Study: AlexNet

[Krizhevsky et al. 2012]

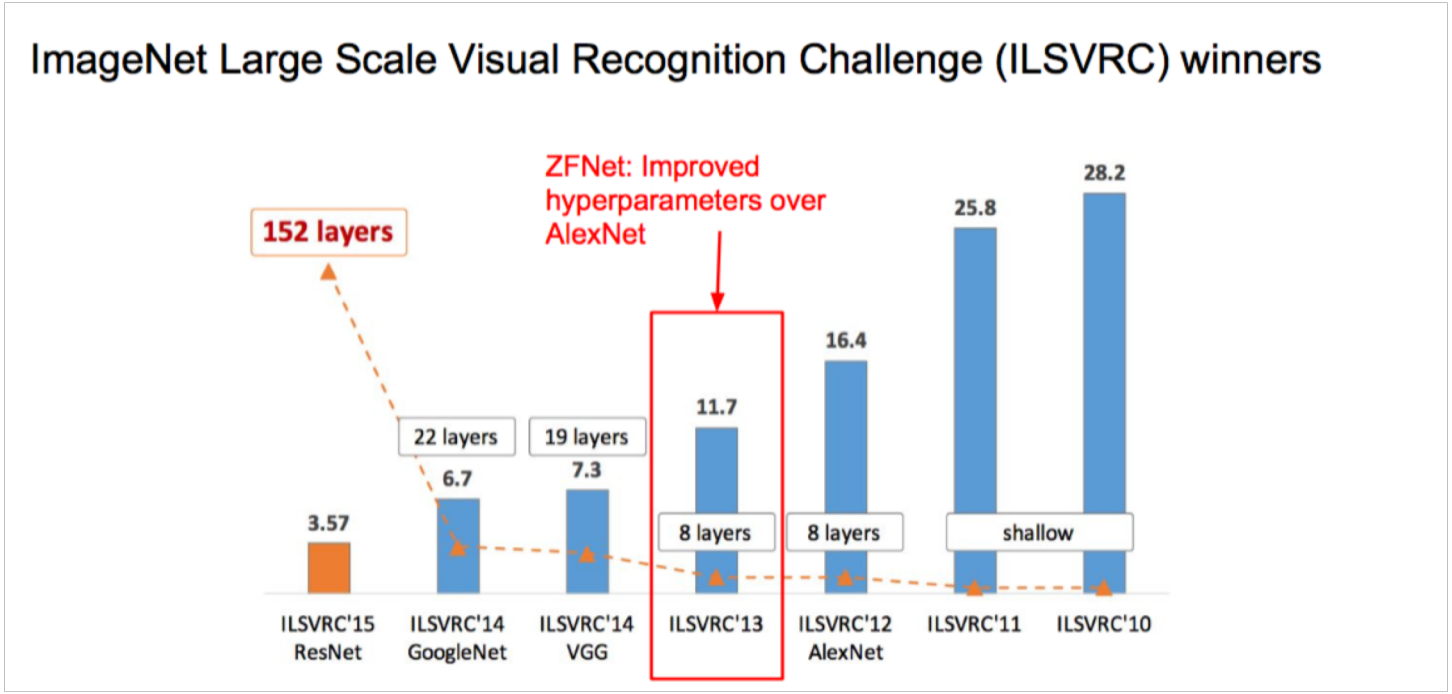


Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Details/Retrospectives:
- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

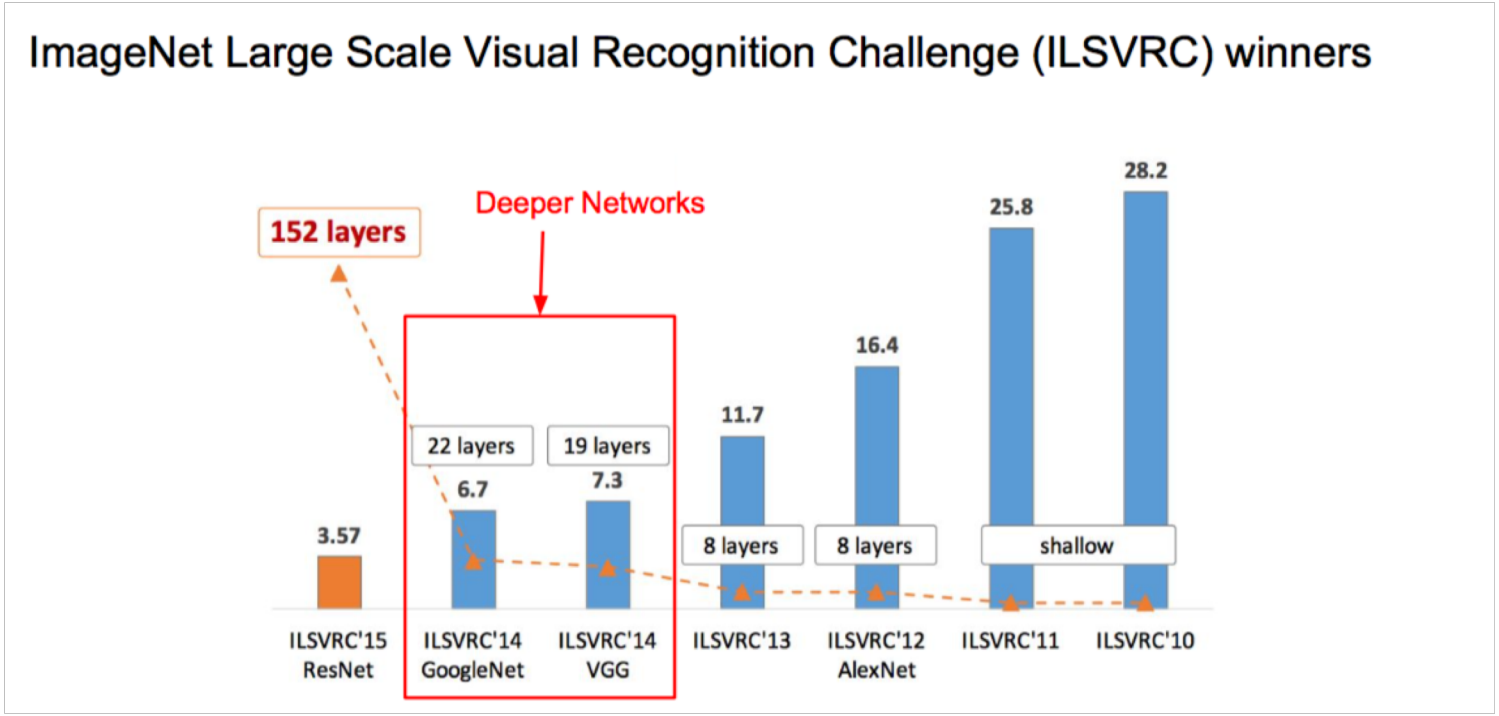
Taken from slides of CS231n course at Stanford University

History



Taken from slides of CS231n course at Stanford University

History



Taken from slides of CS231n course at Stanford University

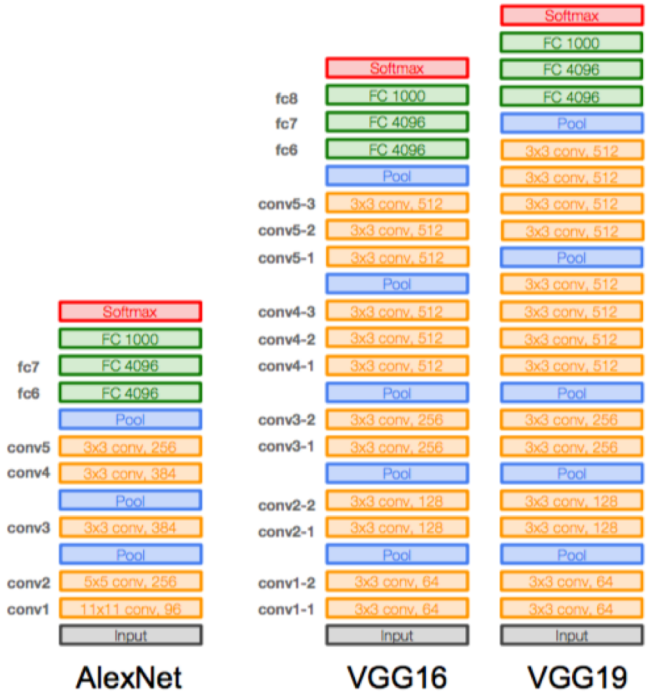
History

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

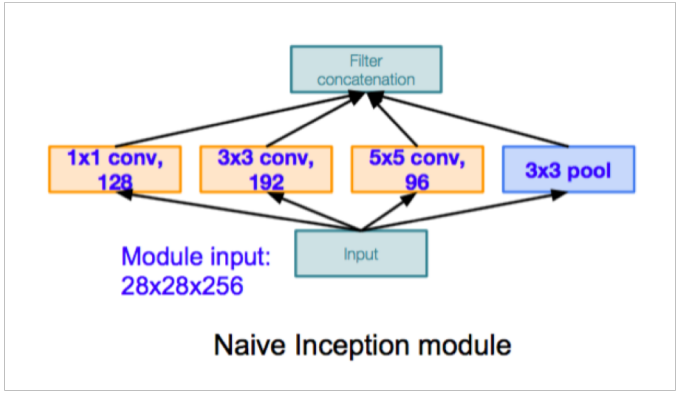
Details:

- ILSVRC'14 2nd in classification, 1st in localization
- Similar training procedure as Krizhevsky 2012
- No Local Response Normalisation (LRN)
- Use VGG16 or VGG19 (VGG19 only slightly better, more memory)
- Use ensembles for best results
- FC7 features generalize well to other tasks



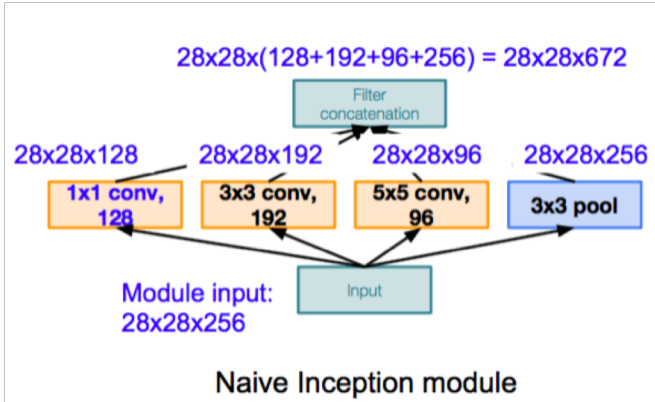
Taken from slides of CS231n course at Stanford University

GoogLeNet (Inception Modules)



Taken from slides of CS231n course at Stanford University

GoogLeNet (Inception Modules)



Conv Ops:
[1x1 conv, 128] 28x28x128x1x1x256
[3x3 conv, 192] 28x28x192x3x3x256
[5x5 conv, 96] 28x28x96x5x5x256
Total: 854M ops

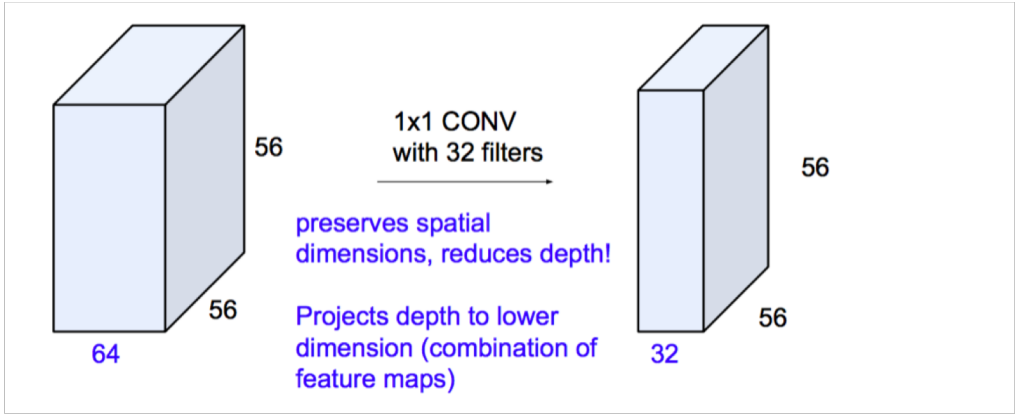
Very expensive compute

Pooling layer also preserves feature depth, which means total depth after concatenation can only grow at every layer!

Taken from slides of CS231n course at Stanford University

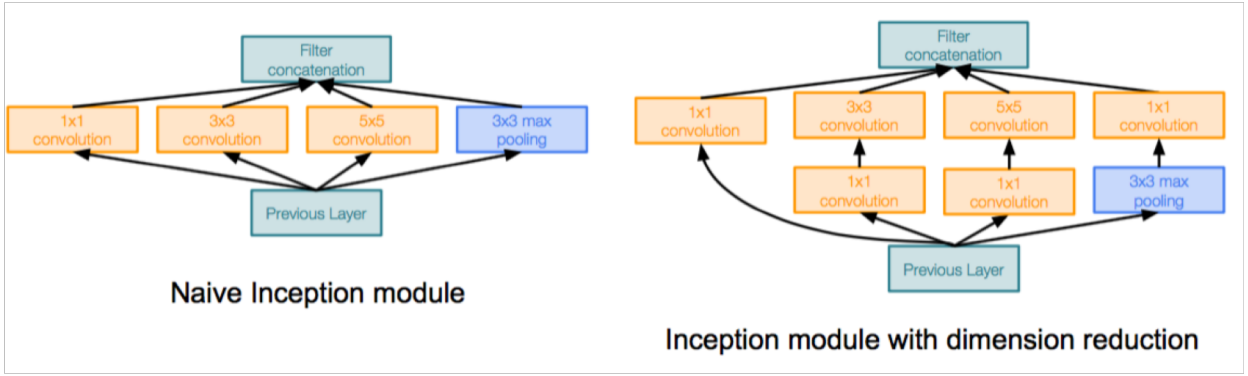
GoogLeNet (Inception Modules)

1 x 1 convolutions



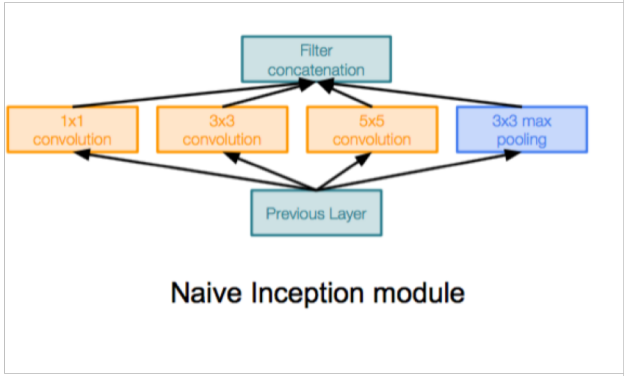
Taken from slides of CS231n course at Stanford University

GoogLeNet (Inception Modules)

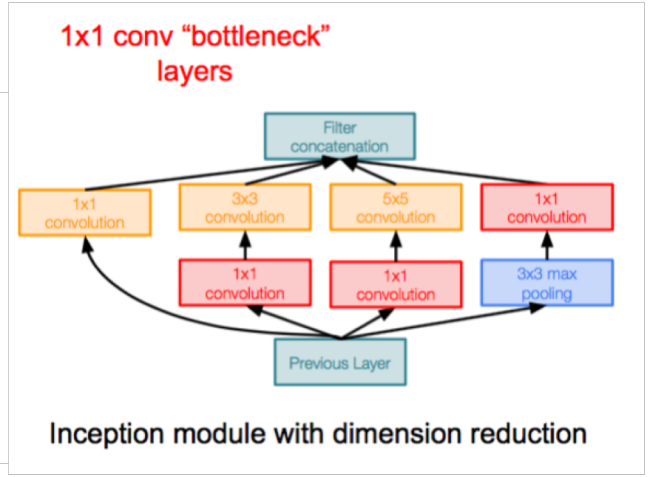


Taken from slides of CS231n course at Stanford University

GoogLeNet (Inception Modules)



Naive Inception module

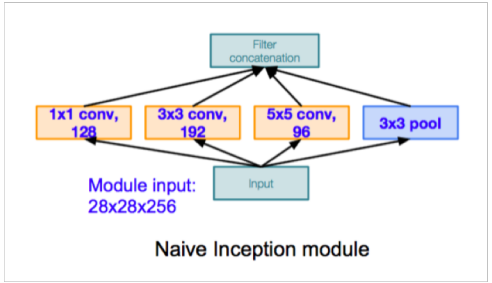


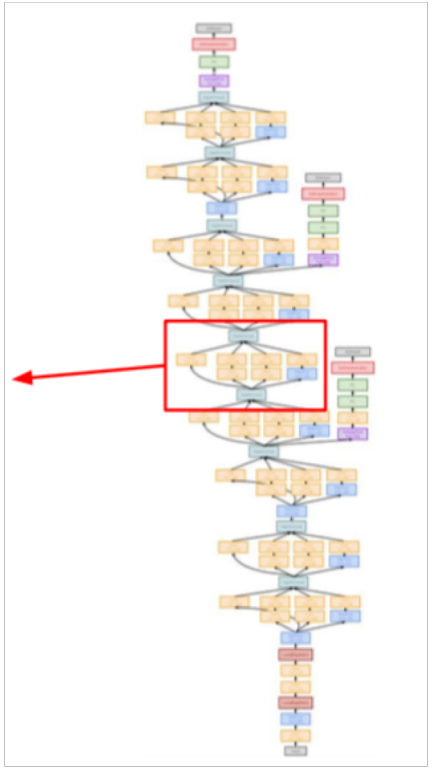
Inception module with dimension reduction

Taken from slides of CS231n course at Stanford University

GoogLeNet (Inception Modules)

Stack inception modules with dimension reduction on top of each other

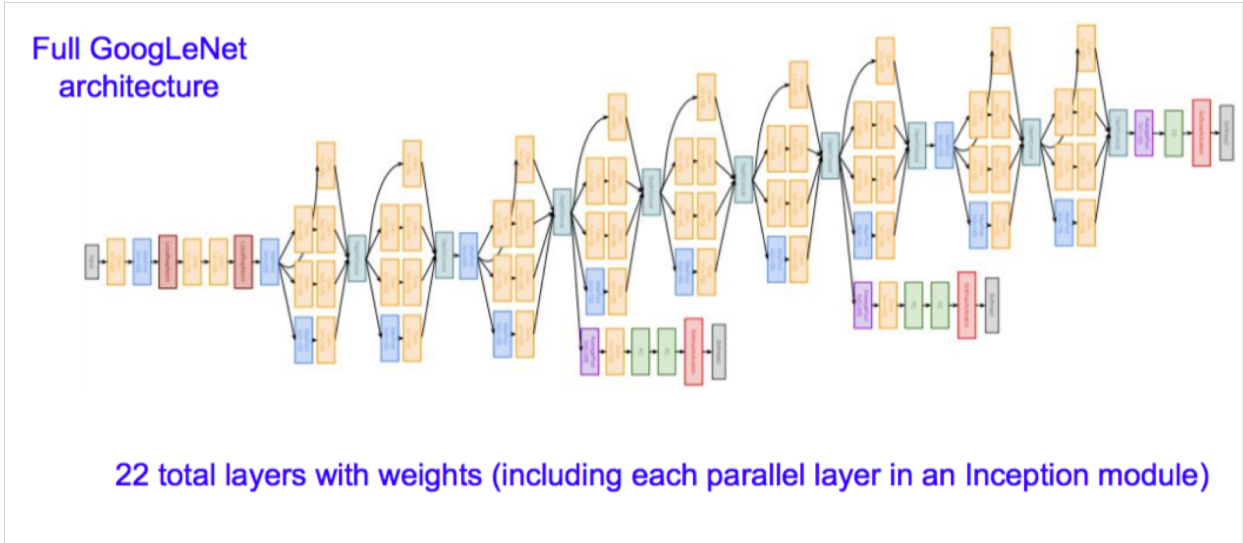




Taken from slides of CS231n course at Stanford University

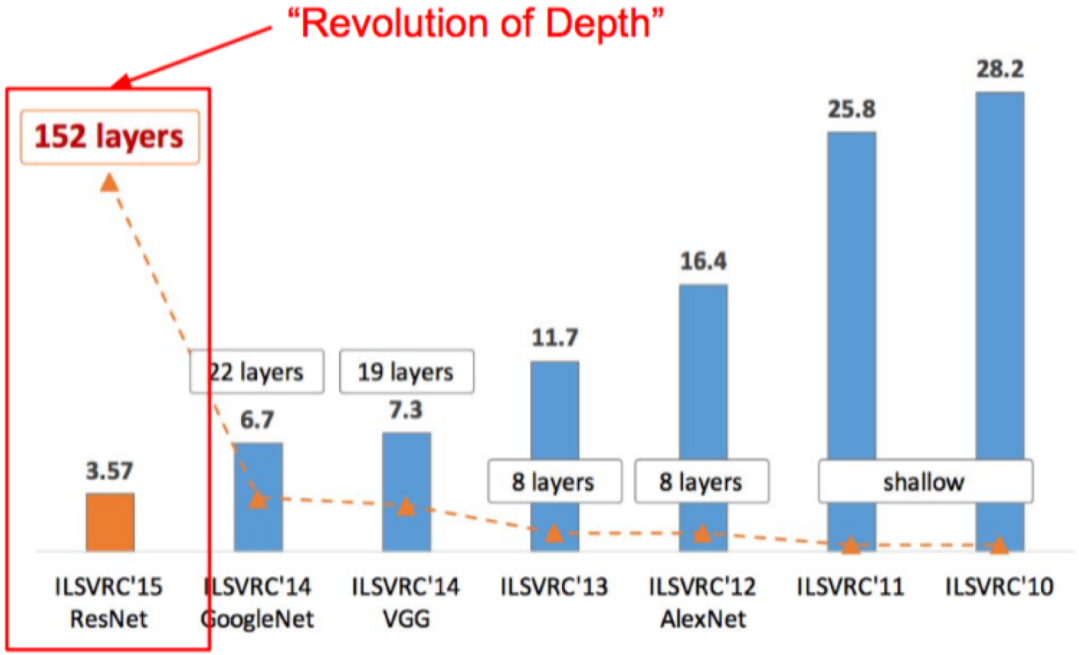
GoogLeNet (Inception Modules)

- 22 layers
- Efficient “Inception” module
- No FC layers
- 12x less params than AlexNet
- ILSVRC’14 classification winner (6.7% top 5 error)



Taken from slides of CS231n course at Stanford University

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



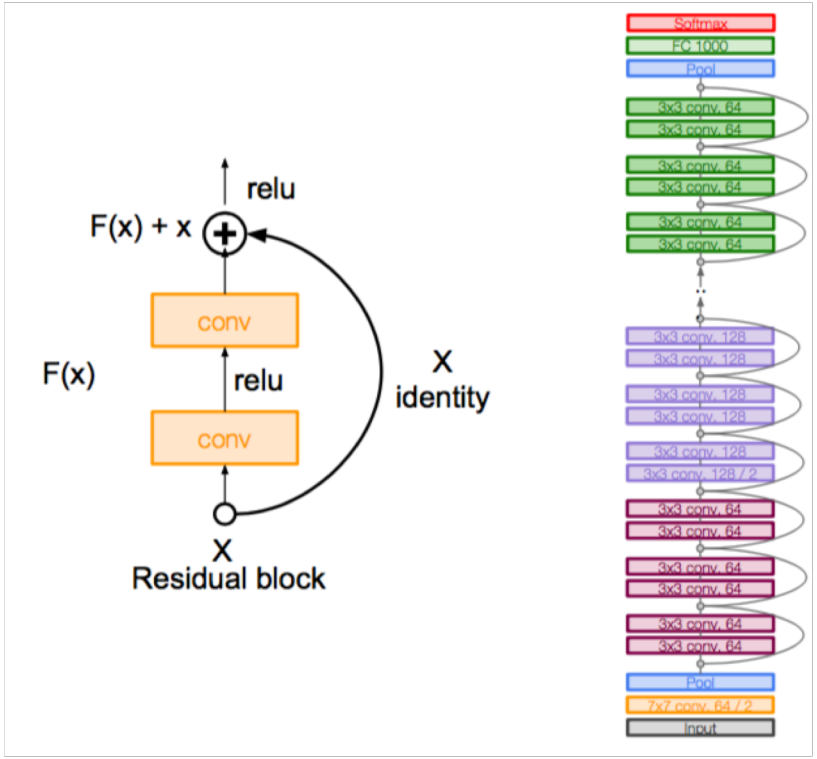
Taken from slides of CS231n course at Stanford University

ResNet (Identity Mapping)

[He et al., 2015]

Very deep networks using residual connections

- 152-layer model for ImageNet
- ILSVRC'15 classification winner (3.57% top 5 error)
- Swept all classification and detection competitions in ILSVRC'15 and COCO'15!



Taken from slides of CS231n course at Stanford University

ResNet (Identity Mapping)

[He et al., 2015]

What happens when we continue stacking deeper layers on a “plain” convolutional neural network?

56-layer model performs worse on both training and test error

*The deeper model performs **worse**, but it’s not caused by overfitting!*

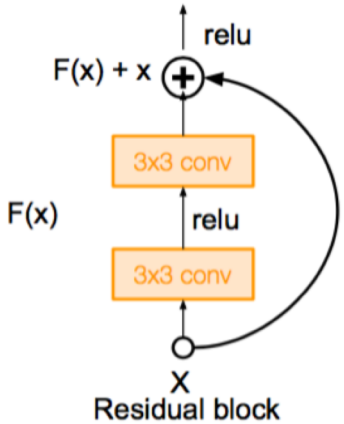
Taken from slides of CS231n course at Stanford University

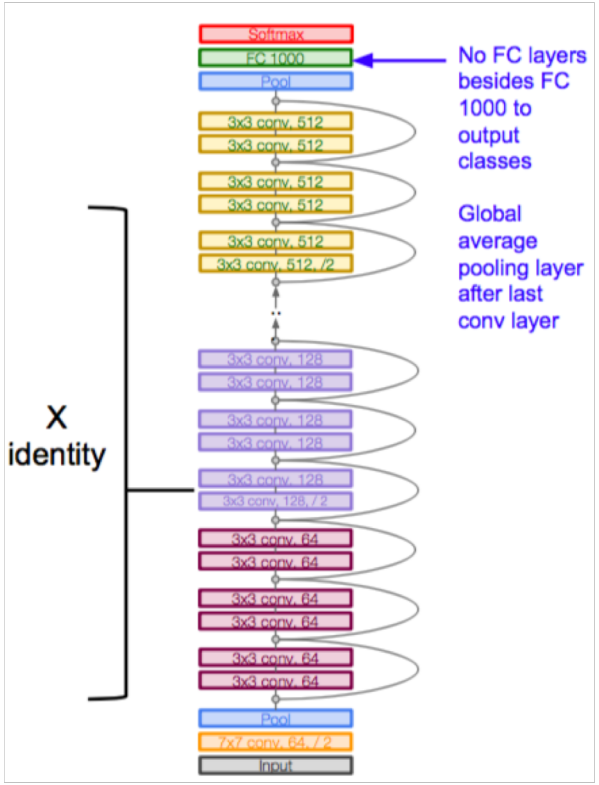
ResNet (Identity Mapping)

[He et al., 2015]

Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes)



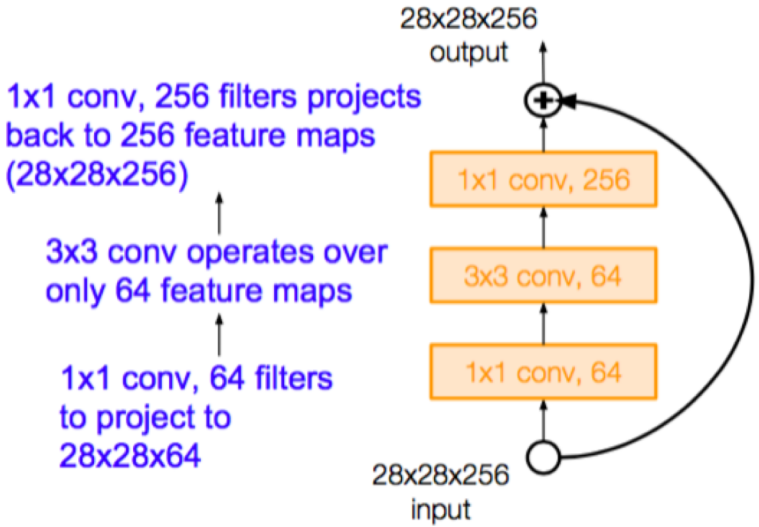


Taken from slides of CS231n course at Stanford University

ResNet (Identity Mapping)

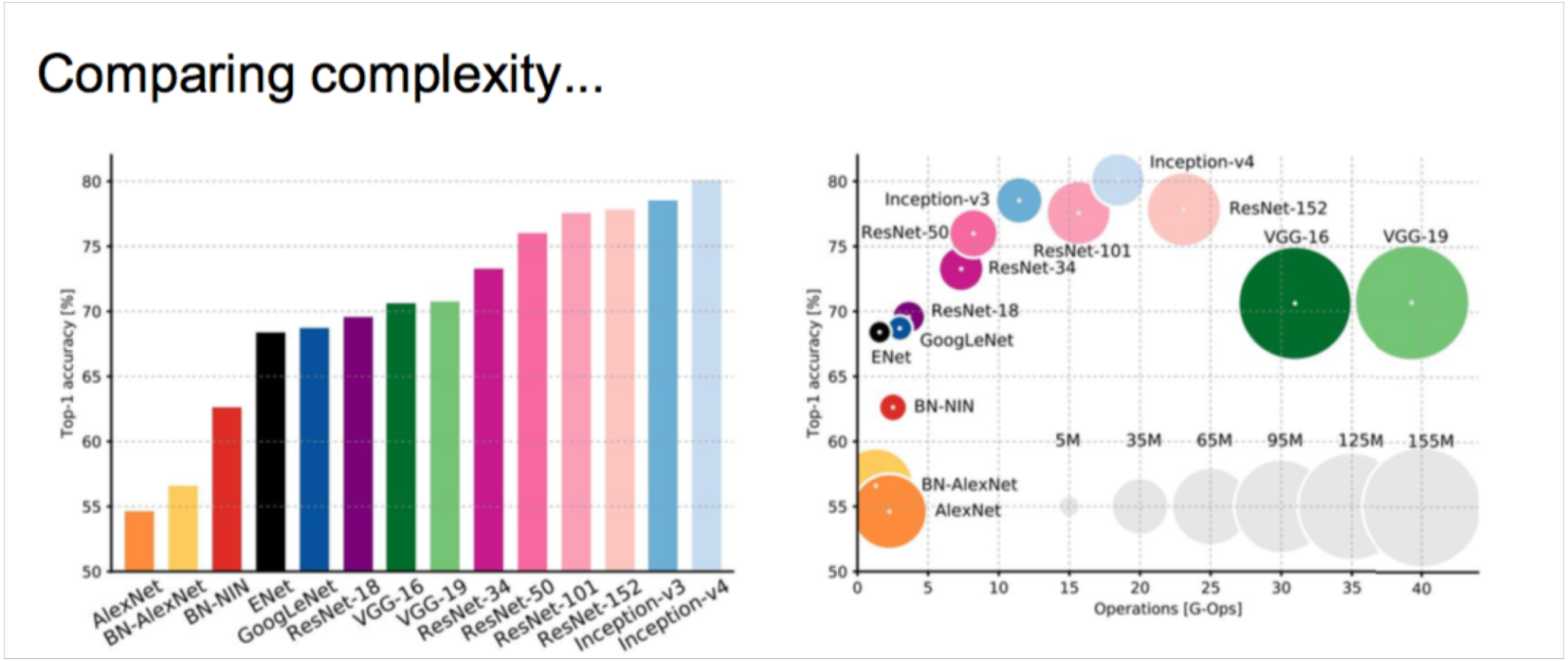
[He et al., 2015]

For deeper networks
(ResNet-50+), use “bottleneck”
layer to improve efficiency
(similar to GoogLeNet)



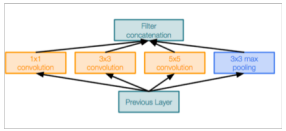
Taken from slides of CS231n course at Stanford University

Until Now ...



Taken from slides of CS231n course at Stanford University

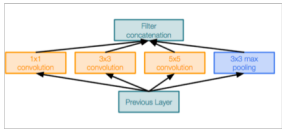
Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning



v/
s

Inception v1, v2 ..

v/
s

(

+

)

ResNet v1, v2 ..

Inception ResNet v1, v2
...

What is the problem being solved?

To gain a better performance over the SOTA in ILSVRC 2015.

What is the problem being solved?

To gain a better performance over the SOTA in ILSVRC 2015.

- *To show that training with **residual connections accelerates** the training of **inception networks** significantly.*
- *Compare residual and non-residual inception networks.*
- *Show that an **ensemble** of three residual and one Inception-v4 you can establish a new SOTA.*

What are the metrics of success?

- Top-1 error
- Top-3 error
- Top-5 error

What are the metrics of success?

- Top-1 error
- Top-3 error
- Top-5 error

Top-1 error
If the correct answer is the same as top 1 predicted answer by the model.

Top-3 error
If the correct answer is within the top 3 predicted answers of the model.

Top-5 error
If the correct answer is within the top 5 predicted answers of the model.

What is the proposed idea/method/ technique/system?

- Training using TensorFlow without partitioning the replicas.

This is enabled in part by recent optimizations of memory used by backpropagation, achieved by carefully considering what tensors are needed for gradient computation and structuring the computation to reduce the number of such tensors.

- Simplifying Inception-v3 and make uniform choices for inception blocks for each grid size

Not simplifying earlier choices in Inception-v3 resulted in networks that looked more complicated that they needed to be. In the newer experiments, for Inception-v4 they shed the unnecessary baggage and made uniform choices for the Inception blocks for each grid size.

What is the proposed idea/method/ technique/system?

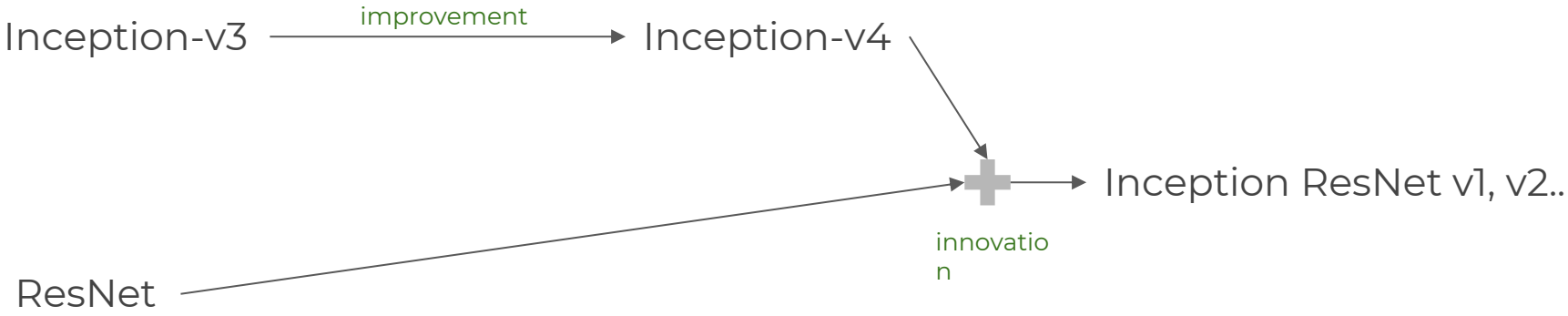
- Training using TensorFlow without partitioning the replicas.

This is enabled in part by recent optimizations of memory used by backpropagation, achieved by carefully considering what tensors are needed for gradient computation and structuring the computation to reduce the number of such tensors.

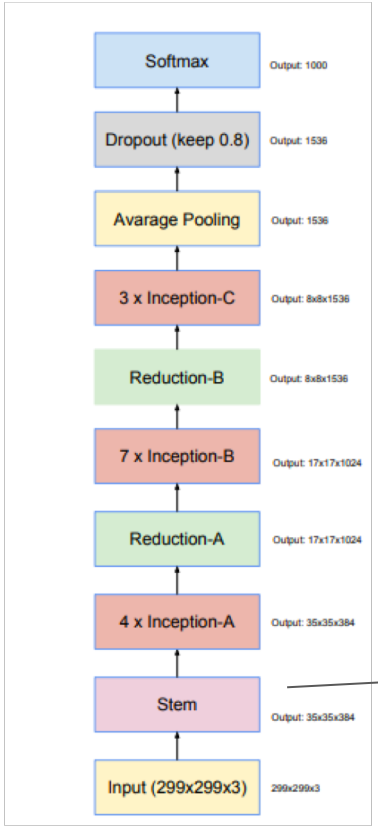
- Simplifying Inception-v3 and make uniform choices for inception blocks for each grid size

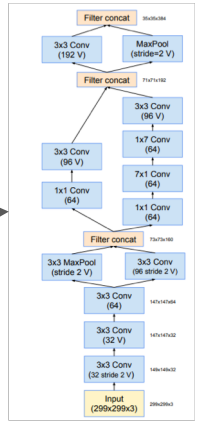
Not simplifying earlier choices in Inception-v3 resulted in networks that looked more complicated that they needed to be. In the newer experiments, for Inception-v4 they shed the unnecessary baggage and made uniform choices for the Inception blocks for each grid size.

What is the key innovation over prior work?

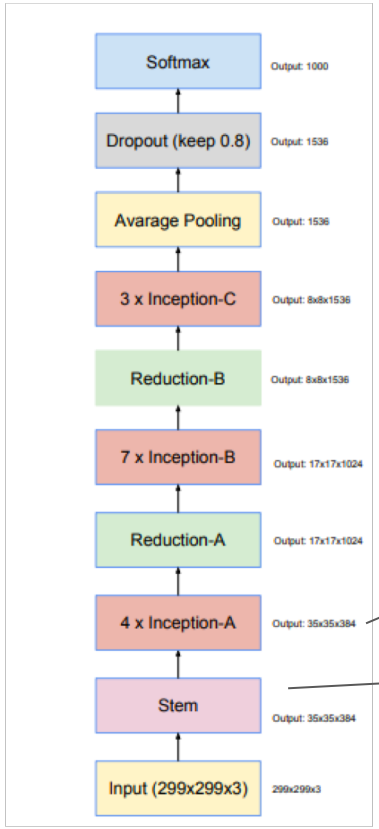


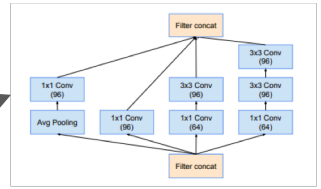
Inception-v4

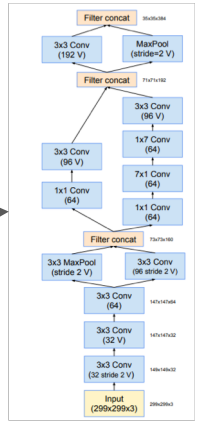




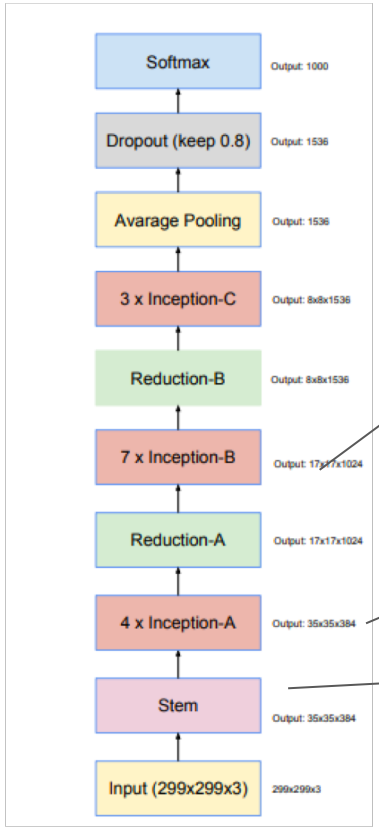
Inception-v4

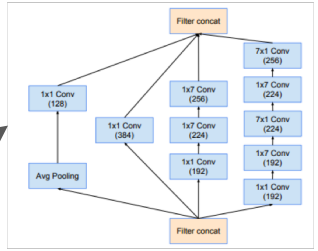


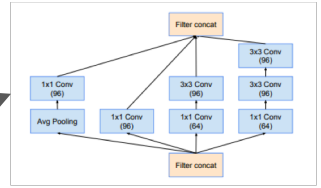


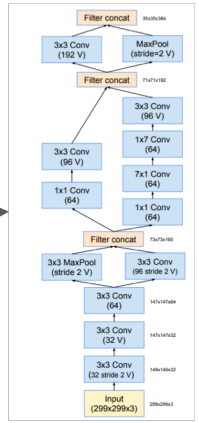


Inception-v4

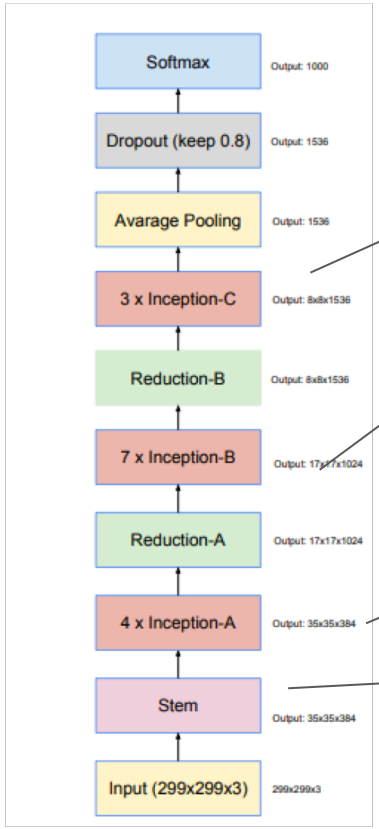


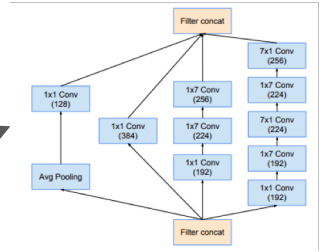


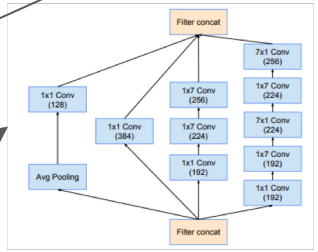


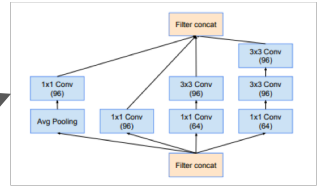


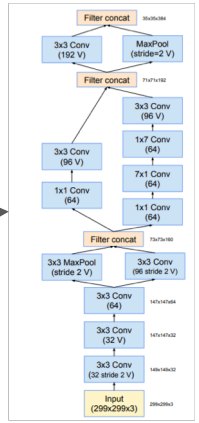
Inception-v4



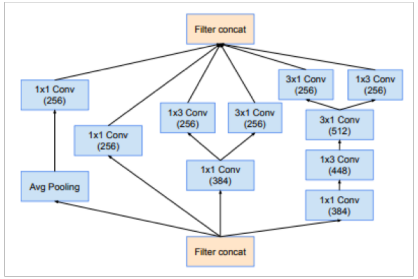




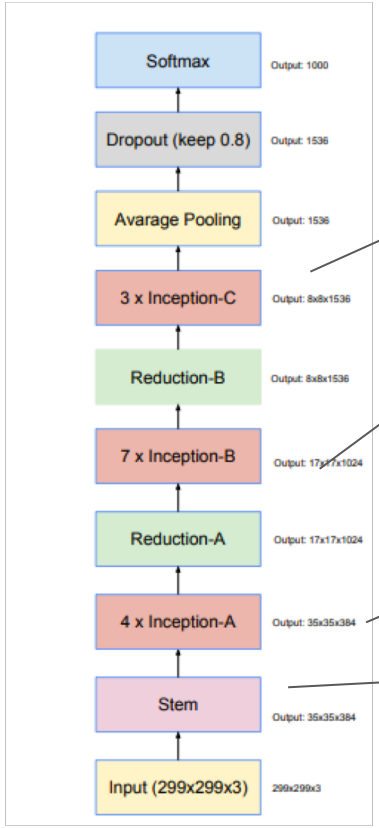


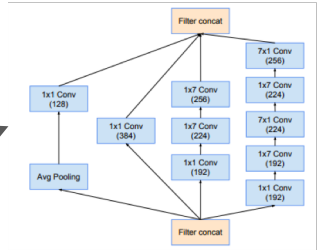


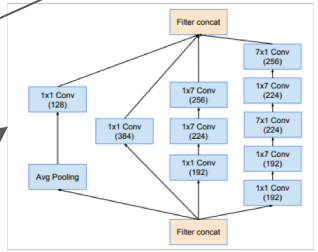
Inception-v4

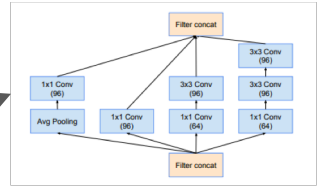


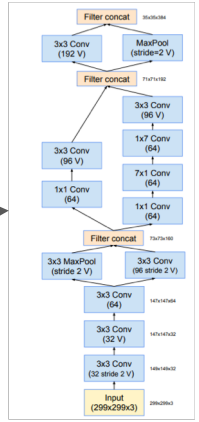
Reduction Module



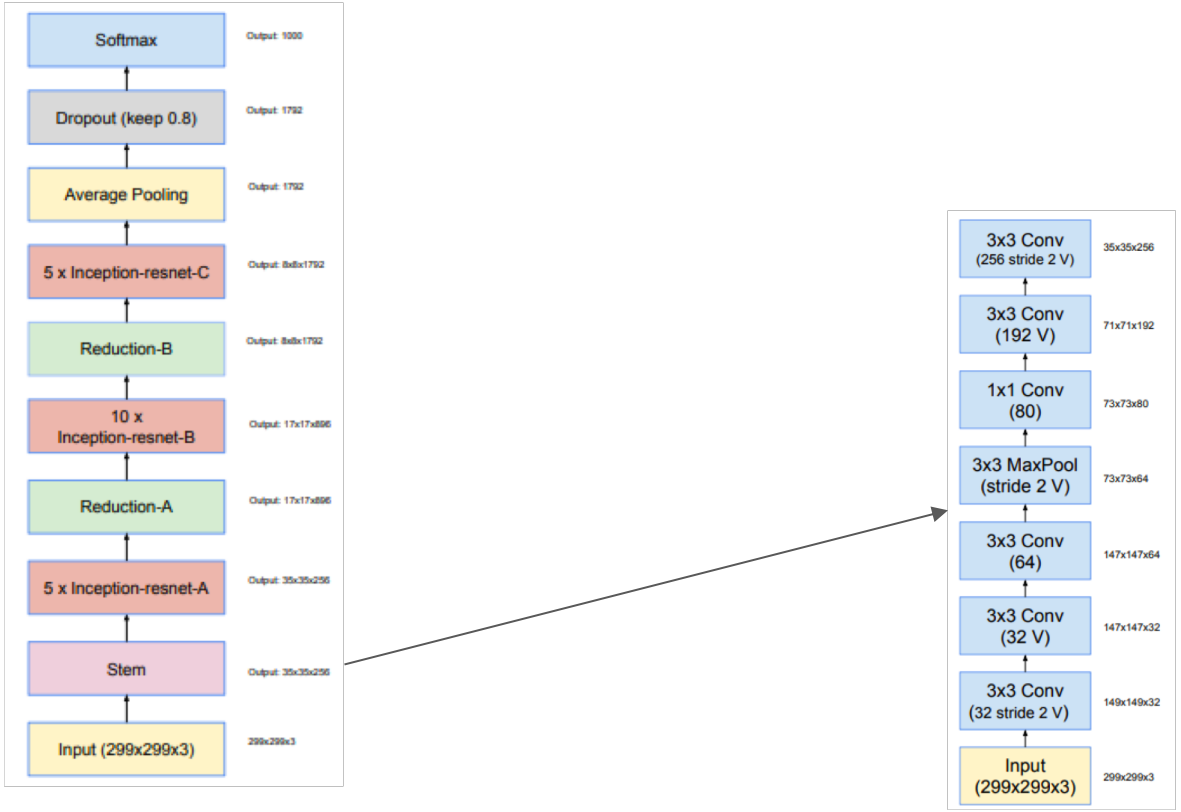




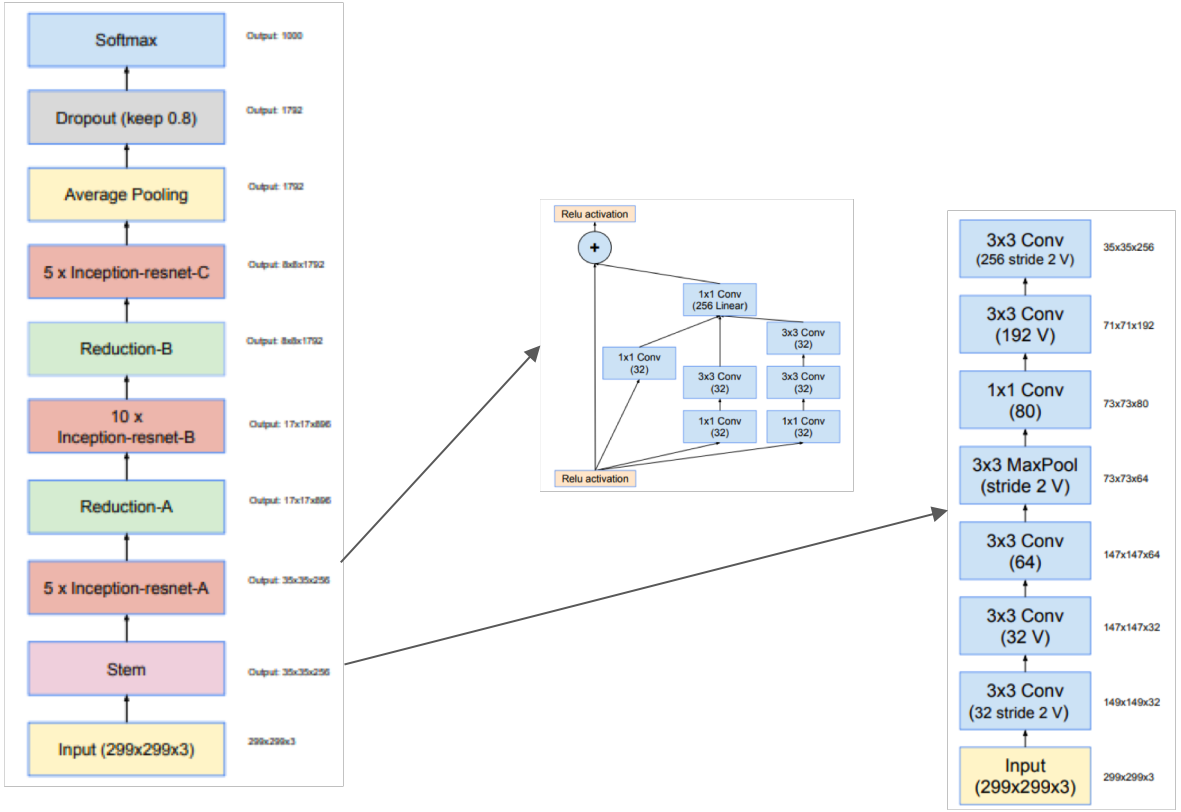




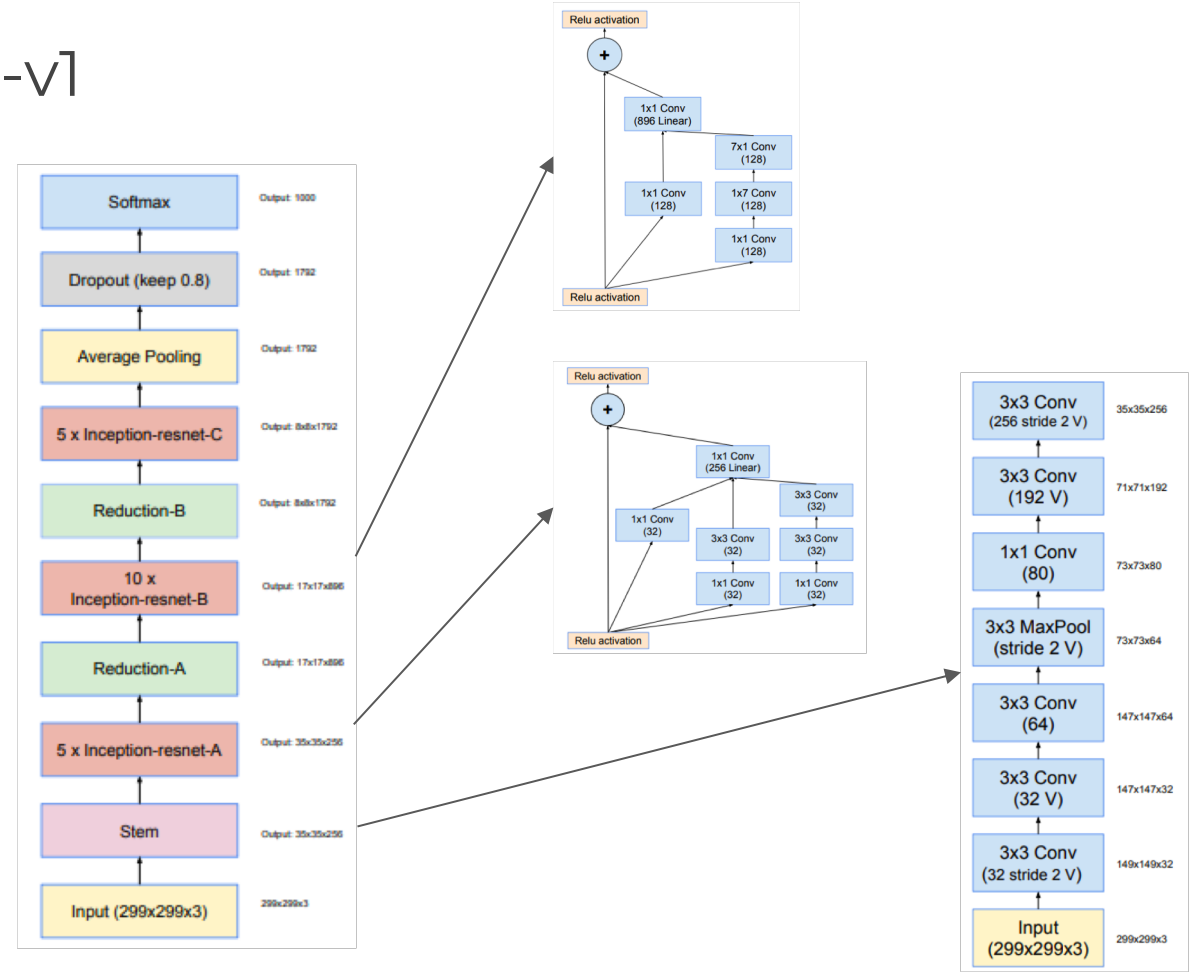
Inception ResNet-v1



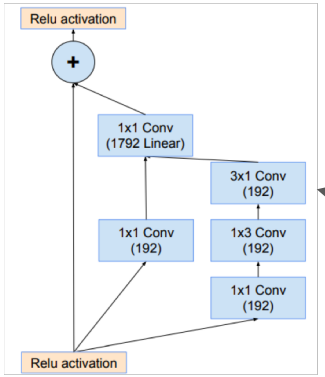
Inception ResNet-v1

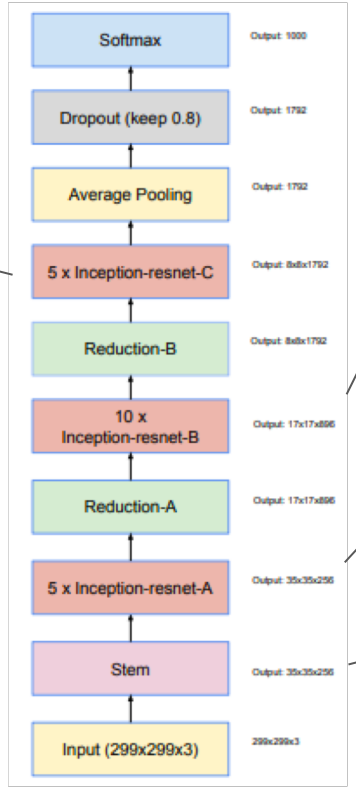


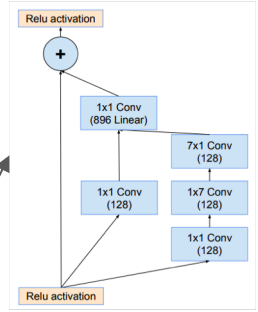
Inception ResNet-v1

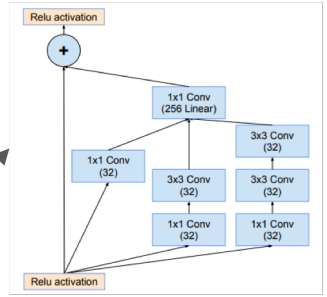


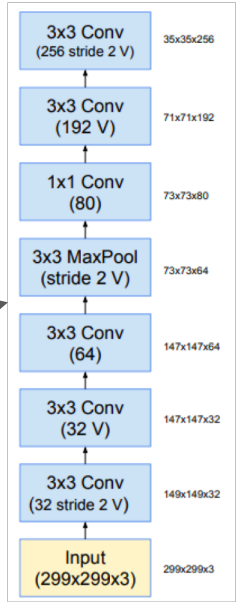
Inception ResNet-v1



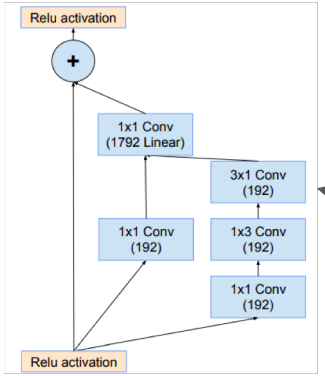


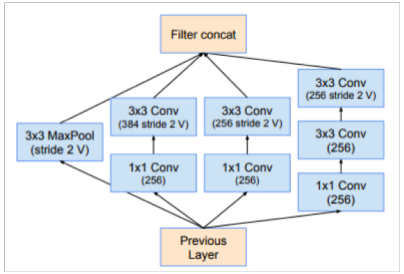




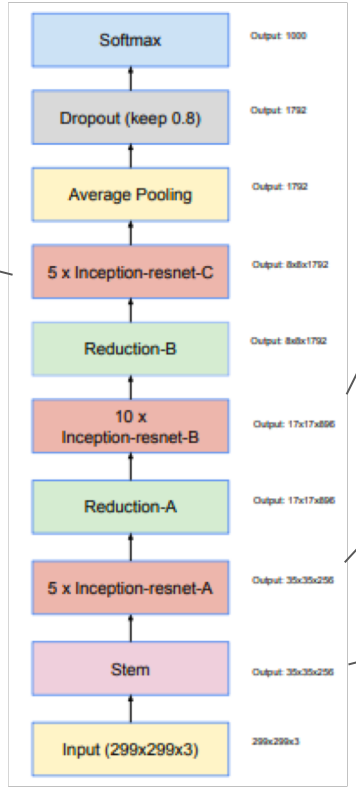


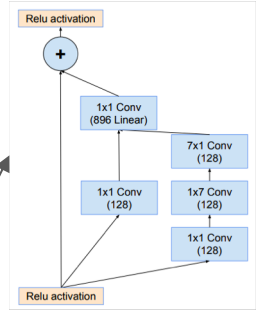
Inception ResNet-v1

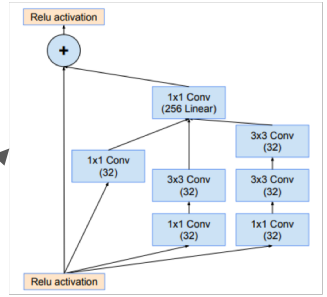


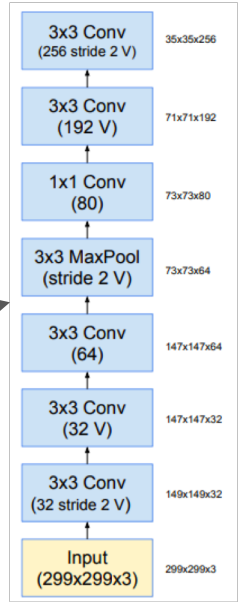


Reduction Module

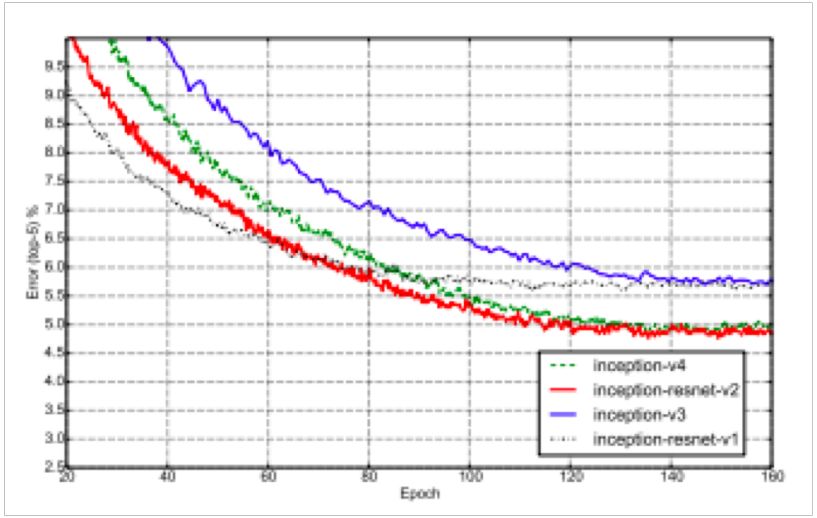




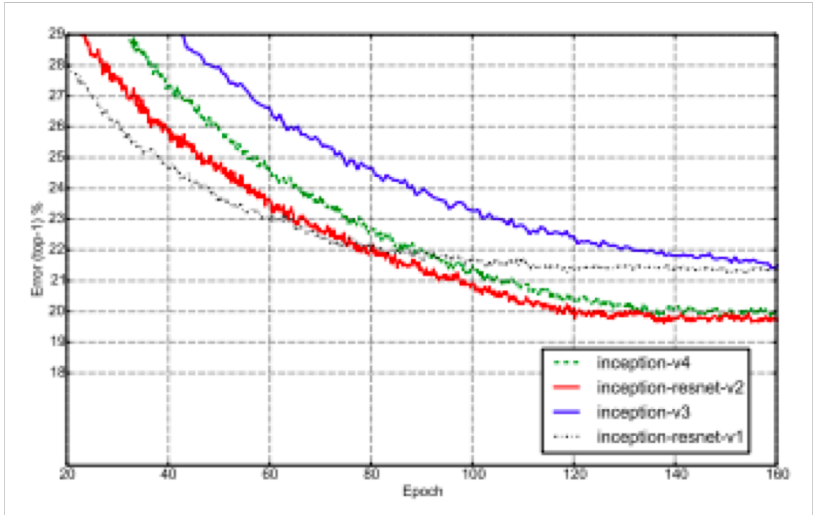




Key Results



Top 5 error evolution of all 4 models during training.



Top 1 error evolution of all 4 models during training.

Key Results

- Inception-ResNet-v1: a hybrid Inception version that has a similar computational cost to Inception-v3
- Inception-ResNet-v2: a costlier hybrid Inception version with significantly improved recognition performance
- Inception-v4: a pure Inception variant without residual connections with roughly the same recognition performance as Inception-ResNet-v2.

Key Results

Ensemble results

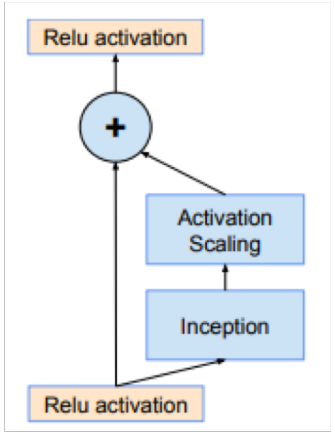
Network	Models	Top-1 Error	Top-5 Error
ResNet-151 [5]	6	–	3.6%
Inception-v3 [15]	4	17.3%	3.6%
Inception-v4 + 3× Inception-ResNet-v2	4	16.5%	3.1%

Limitations

If the number of filters exceeded 1000, the residual variants started to exhibit instabilities and the network has just “died” early in the training,

Meaning that the last layer before the average pooling started to produce only zeros after a few tens of thousands of iterations.

This could **NOT** be prevented, neither by lowering the learning rate, nor by adding an extra batch-normalization to this layer.



Long Term Impact

Introduction of **residual connections** led to dramatically **improved training** speed for the **Inception** architecture.

Also the latest models (with and without residual connections) **outperform all the previous networks**, just by virtue of the increased model size.

Thank You