Title: Inception (deep learning architecture)

URL: https://en.wikipedia.org/wiki/Inception_(deep_learning_architecture)

PageID: 60320598

Categories: Category:Artificial neural networks, Category:Computer vision, Category:Google software

---

Inception [1] is a family of convolutional neural network (CNN) for computer vision , introduced by researchers at Google in 2014 as GoogLeNet (later renamed Inception v1). The series was historically important as an early CNN that separates the stem (data ingest), body (data processing), and head (prediction), an architectural design that persists in all modern CNN. [2]

Version history

Inception v1

In 2014, a team at Google developed the GoogLeNet architecture, an instance of which won the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). [1] [3]

The name came from the LeNet of 1998, since both LeNet and GoogLeNet are CNNs. They also called it "Inception" after a "we need to go deeper" internet meme, a phrase from Inception (2010) the film. [1] Because later, more versions were released, the original Inception architecture was renamed again as "Inception v1".

The models and the code were released under Apache 2.0 license on GitHub. [4]

The Inception v1 architecture is a deep CNN composed of 22 layers. Most of these layers were "Inception modules". The original paper stated that Inception modules are a "logical culmination" of Network in Network [5] and (Arora et al, 2014). [6]

Since Inception v1 is deep, it suffered from the vanishing gradient problem . The team solved it by using two "auxiliary classifiers", which are linear-softmax classifiers inserted at 1/3-deep and 2/3-deep within the network, and the loss function is a weighted sum of all three: $L = 0.3 L_{aux,1} + 0.3 L_{aux,2} + L_{real}$

These were removed after training was complete. This was later solved by the ResNet architecture .

The architecture consists of three parts stacked on top of one another: [2]

The stem (data ingestion): The first few convolutional layers perform data preprocessing to downscale images to a smaller size.

The body (data processing): The next many Inception modules perform the bulk of data processing.

The head (prediction): The final fully-connected layer and softmax produces a probability distribution for image classification.

This structure is used in most modern CNN architectures.

Inception v2

Inception v2 was released in 2015, in a paper that is more famous for proposing batch normalization . [7] [8] It had 13.6 million parameters.

It improves on Inception v1 by adding batch normalization, and removing dropout and local response normalization which they found became unnecessary when batch normalization is used.

Inception v3

Inception v3 was released in 2016. [7] [9] It improves on Inception v2 by using factorized convolutions.

As an example, a single 5×5 convolution can be factored into 3×3 stacked on top of another 3×3. Both has a receptive field of size 5×5. The 5×5 convolution kernel has 25 parameters, compared to just 18 in the factorized version. Thus, the 5×5 convolution is strictly more powerful than the factorized version. However, this power is not necessarily needed. Empirically, the research team found that factorized convolutions help.

It also uses a form of dimension-reduction by concatenating the output from a convolutional layer and a pooling layer . As an example, a tensor of size 35 × 35 × 320 {\displaystyle 35\times 35\times 320} can be downscaled by a convolution with stride 2 to 17 × 17 × 320 {\displaystyle 17\times 17\times 320} , and by maxpooling with pool size 2 × 2 {\displaystyle 2\times 2} to 17 × 17 × 320 {\displaystyle 17\times 17\times 320} . These are then concatenated to 17 × 17 × 640 {\displaystyle 17\times 17\times 640} .

Other than this, it also removed the lowest auxiliary classifier during training. They found that the auxiliary head worked as a form of regularization .

They also proposed label-smoothing regularization in classification. For an image with label c {\displaystyle c} , instead of making the model to predict the probability distribution δ c = (0 , 0 , … , 0 , 1 ■ c -th entry , 0 , … , 0) {\displaystyle \delta _{c}=(0,0,\dots ,0,\underbrace {1} _{c{\text{-th entry}}},0,\dots ,0)} , they made the model predict the smoothed distribution (1 – ■) δ c + ■ / K {\displaystyle (1-\epsilon)\delta _{c}+\epsilon /K} where K {\displaystyle K} is the total number of classes.

Inception v4

In 2017, the team released Inception v4, Inception ResNet v1, and Inception ResNet v2. [10]

Inception v4 is an incremental update with even more factorized convolutions, and other complications that were empirically found to improve benchmarks.

Inception ResNet v1 and v2 are both modifications of Inception v4, where residual connections are added to each Inception module, inspired by the ResNet architecture. [11]

Xception

Xception ("Extreme Inception") was published in 2017. [12] It is a linear stack of depthwise separable convolution layers with residual connections. The design was proposed on the hypothesis that in a CNN, the cross-channels correlations and spatial correlations in the feature maps can be entirely decoupled .

Training each network took 3 days on 60 K80 GPUs, or approximately 0.5 petaFLOP-days. [13]

References

External links

A list of all Inception models released by Google: "models/research/slim/README.md at master · tensorflow/models" . GitHub . Retrieved 2024-10-19 .

v

t

e

Google

Google Brain

Google DeepMind

AlphaGo (2015)

Master (2016)

AlphaGo Zero (2017)

AlphaZero (2017)

MuZero (2019)
Fan Hui (2015)
Lee Sedol (2016)
Ke Jie (2017)
AlphaGo (2017)
The MANIAC (2023)
AlphaFold (2018)
AlphaStar (2019)
AlphaDev (2023)
AlphaGeometry (2024)
AlphaGenome (2025)
Inception (2014)
WaveNet (2016)
MobileNet (2017)
Transformer (2017)
EfficientNet (2019)
Gato (2022)
Quantum Artificial Intelligence Lab
TensorFlow
Tensor Processing Unit
Assistant (2016)
Sparrow (2022)
Gemini (2023)
BERT (2018)
XLNet (2019)
T5 (2019)
LaMDA (2021)
Chinchilla (2022)
PaLM (2022)
Imagen (2023)
Gemini (2023)
VideoPoet (2024)
Gemma (2024)
Veo (2024)
DreamBooth (2022)
NotebookLM (2023)
Vids (2024)
Gemini Robotics (2025)

" Attention Is All You Need "
Future of Go Summit
Generative pre-trained transformer
Google Labs
Google Pixel
Google Workspace
Robot Constitution
Category
Commons
v
t
e
Differentiable programming
Information geometry
Statistical manifold
Automatic differentiation
Neuromorphic computing
Pattern recognition
Ricci calculus
Computational learning theory
Inductive bias
IPU
TPU
VPU
Memristor
SpiNNaker
TensorFlow
PyTorch
Keras
scikit-learn
Theano
JAX
Flux.jl
MindSpore
Portals Computer programming Technology
Computer programming
Technology