Title: Residual neural network

URL: https://en.wikipedia.org/wiki/Residual_neural_network

PageID: 55867424

Categories: Category:Deep learning, Category:Neural network architectures

---

A residual neural network (also referred to as a residual network or ResNet) [1] is a deep learning architecture in which the layers learn residual functions with reference to the layer inputs. It was developed in 2015 for image recognition , and won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) of that year. [2] [3]

As a point of terminology, "residual connection" refers to the specific architectural motif of x ■ f (x) + x {\displaystyle x\mapsto f(x)+x} , where f {\displaystyle f} is an arbitrary neural network module. The motif had been used previously (see §History for details). However, the publication of ResNet made it widely popular for feedforward networks , appearing in neural networks that are seemingly unrelated to ResNet.

The residual connection stabilizes the training and convergence of deep neural networks with hundreds of layers, and is a common motif in deep neural networks, such as transformer models (e.g., BERT , and GPT models such as ChatGPT), the AlphaGo Zero system, the AlphaStar system, and the AlphaFold system.

Mathematics

Residual connection

In a multilayer neural network model, consider a subnetwork with a certain number of stacked layers (e.g., 2 or 3). Denote the underlying function performed by this subnetwork as H (x) {\displaystyle H(x)} , where x {\displaystyle x} is the input to the subnetwork. Residual learning re-parameterizes this subnetwork and lets the parameter layers represent a "residual function" F (x) = H (x) – x {\displaystyle F(x)=H(x)-x} . The output y {\displaystyle y} of this subnetwork is then represented as:

y = F (x) + x {\displaystyle y=F(x)+x}

The operation of " + x {\displaystyle +\ x} " is implemented via a "skip connection" that performs an identity mapping to connect the input of the subnetwork with its output. This connection is referred to as a "residual connection" in later work. The function F (x) {\displaystyle F(x)} is often represented by matrix multiplication interlaced with activation functions and normalization operations (e.g., batch normalization or layer normalization). As a whole, one of these subnetworks is referred to as a "residual block". [1] A deep residual network is constructed by simply stacking these blocks.

Long short-term memory (LSTM) has a memory mechanism that serves as a residual connection. [4] In an LSTM without a forget gate , an input x t {\displaystyle x_{t}} is processed by a function F {\displaystyle F} and added to a memory cell c t {\displaystyle c_{t}} , resulting in c t + 1 = c t + F (x t) {\displaystyle c_{t+1}=c_{t}+F(x_{t})} . An LSTM with a forget gate essentially functions as a highway network .

To stabilize the variance of the layers' inputs, it is recommended to replace the residual connections x + f (x) {\displaystyle x+f(x)} with x / L + f (x) {\displaystyle x/L+f(x)} , where L {\displaystyle L} is the total number of residual layers. [5]

Projection connection

If the function F {\displaystyle F} is of type F : R n → R m {\displaystyle F:\mathbb {R} ^{n}\to \mathbb {R} ^{m}} where n ≠ m {\displaystyle n\neq m} , then F (x) + x {\displaystyle F(x)+x} is undefined. To handle this special case, a projection connection is used:

y = F (x) + P (x) {\displaystyle y=F(x)+P(x)}

where P {\displaystyle P} is typically a linear projection, defined by P (x) = M x {\displaystyle P(x)=Mx} where M {\displaystyle M} is a m × n {\displaystyle m\times n} matrix. The matrix is trained via backpropagation , as is any other parameter of the model.

Signal propagation

The introduction of identity mappings facilitates signal propagation in both forward and backward paths. [6]

Forward propagation

If the output of the ■ {\displaystyle \ell } -th residual block is the input to the (■ + 1) {\displaystyle (\ell +1)} -th residual block (assuming no activation function between blocks), then the (■ + 1) {\displaystyle (\ell +1)} -th input is:

x ■ + 1 = F (x ■) + x ■ {\displaystyle x_{\ell +1}=F(x_{\ell })+x_{\ell }}

Applying this formulation recursively, e.g.:

x ■ + 2 = F (x ■ + 1) + x ■ + 1 = F (x ■ + 1) + F (x ■) + x ■ {\displaystyle {\begin{aligned}x_{\ell +2}&=F(x_{\ell +1})+x_{\ell +1}\\&=F(x_{\ell +1})+F(x_{\ell })+x_{\ell }\end{aligned}}}

yields the general relationship:

x L = x ■ + ∑ i = ■ L − 1 F (x i) {\displaystyle x_{L}=x_{\ell }+\sum _{i=\ell }^{L-1}F(x_{i})}

where L {\textstyle L} is the index of a residual block and ■ {\textstyle \ell } is the index of some earlier block. This formulation suggests that there is always a signal that is directly sent from a shallower block ■ {\textstyle \ell } to a deeper block L {\textstyle L} .

Backward propagation

The residual learning formulation provides the added benefit of mitigating the vanishing gradient problem to some extent. However, it is crucial to acknowledge that the vanishing gradient issue is not the root cause of the degradation problem, which is tackled through the use of normalization. To observe the effect of residual blocks on backpropagation, consider the partial derivative of a loss function E {\displaystyle {\mathcal {E}}} with respect to some residual block input x ■ {\displaystyle x_{\ell }} . Using the equation above from forward propagation for a later residual block L > ■ {\displaystyle L>\ell } : [6]

∂ E ∂ x ■ = ∂ E ∂ x L ∂ x L ∂ x ■ = ∂ E ∂ x L (1 + ∂ ∂ x ■ ∑ i = ■ L − 1 F (x i)) = ∂ E ∂ x L + ∂ E ∂ x L ∂ ∂ x ■ ∑ i = ■ L − 1 F (x i) {\displaystyle {\begin{aligned}{\frac {\partial {\mathcal {E}}}{\partial x_{\ell }}}&={\frac {\partial {\mathcal {E}}}{\partial x_{L}}}{\frac {\partial x_{L}}{\partial x_{\ell }}}\\&={\frac {\partial {\mathcal {E}}}{\partial x_{L}}}\left(1+{\frac {\partial }{\partial x_{\ell }}}\sum _{i=\ell }^{L-1}F(x_{i})\right)\\&={\frac {\partial {\mathcal {E}}}{\partial x_{L}}}+{\frac {\partial {\mathcal {E}}}{\partial x_{L}}}{\frac {\partial }{\partial x_{\ell }}}\sum _{i=\ell }^{L-1}F(x_{i})\end{aligned}}}

This formulation suggests that the gradient computation of a shallower layer, ∂ E ∂ x ■ {\textstyle {\frac {\partial {\mathcal {E}}}{\partial x_{\ell }}}} , always has a later term ∂ E ∂ x L {\textstyle {\frac {\partial {\mathcal {E}}}{\partial x_{L}}}} that is directly added. Even if the gradients of the F (x i) {\displaystyle F(x_{i})} terms are small, the total gradient ∂ E ∂ x ■ {\textstyle {\frac {\partial {\mathcal {E}}}{\partial x_{\ell }}}} resists vanishing due to the added term ∂ E ∂ x L {\textstyle {\frac {\partial {\mathcal {E}}}{\partial x_{L}}}} .

Variants of residual blocks

Basic block

A basic block is the simplest building block studied in the original ResNet. [1] This block consists of two sequential 3x3 convolutional layers and a residual connection. The input and output dimensions of both layers are equal.

Bottleneck block

A bottleneck block [1] consists of three sequential convolutional layers and a residual connection. The first layer in this block is a 1×1 convolution for dimension reduction (e.g., to 1/2 of the input dimension); the second layer performs a 3×3 convolution; the last layer is another 1×1 convolution for dimension restoration. The models of ResNet-50, ResNet-101, and ResNet-152 are all based on bottleneck blocks. [1]

Pre-activation block

The pre-activation residual block [6] applies activation functions before applying the residual function F {\displaystyle F} . Formally, the computation of a pre-activation residual block can be written as:

x ■ + 1 = F (φ (x ■)) + x ■ {\displaystyle x_{\ell +1}=F(\phi (x_{\ell }))+x_{\ell }}

where φ {\displaystyle \phi } can be any activation (e.g. ReLU) or normalization (e.g. LayerNorm) operation. This design reduces the number of non-identity mappings between residual blocks, and allows an identity mapping directly from the input to the output. This design was used to train models with 200 to over 1000 layers, and was found to consistently outperform variants where the residual path is not an identity function. The pre-activation ResNet with 200 layers took 3 weeks to train for ImageNet on 8 GPUs in 2016. [6]

Since GPT-2 , transformer blocks have been mostly implemented as pre-activation blocks. This is often referred to as "pre-normalization" in the literature of transformer models. [7]

Applications

Originally, ResNet was designed for computer vision . [1] [8] [9]

All transformer architectures include residual connections. Indeed, very deep transformers cannot be trained without them. [10]

The original ResNet paper made no claim on being inspired by biological systems. However, later research has related ResNet to biologically-plausible algorithms. [11] [12]

A study published in Science in 2023 [13] disclosed the complete connectome of an insect brain (specifically that of a fruit fly larva). This study discovered "multilayer shortcuts" that resemble the skip connections in artificial neural networks, including ResNets.

History

Previous work

Residual connections were noticed in neuroanatomy , such as Lorente de No (1938). [14] : Fig 3 McCulloch and Pitts (1943) proposed artificial neural networks and considered those with residual connections. [15] : Fig 1.h

In 1961, Frank Rosenblatt described a three-layer multilayer perceptron (MLP) model with skip connections. [16] : 313, Chapter 15 The model was referred to as a "cross-coupled system", and the skip connections were forms of cross-coupled connections.

During the late 1980s, "skip-layer" connections were sometimes used in neural networks. Examples include: [17] [18] Lang and Witbrock (1988) [19] trained a fully connected feedforward network where each layer skip-connects to all subsequent layers, like the later DenseNet (2016). In this work, the residual connection was the form x ■ F (x) + P (x) {\displaystyle x\mapsto F(x)+P(x)} , where P {\displaystyle P} is a randomly-initialized projection connection. They termed it a "short-cut connection". An early neural language model used residual connections and named them "direct connections". [20]

Degradation problem

Sepp Hochreiter discovered the vanishing gradient problem in 1991 [21] and argued that it explained why the then-prevalent forms of recurrent neural networks did not work for long sequences. He and Schmidhuber later designed the LSTM architecture to solve this problem, [4] [22] which has a "cell state" c t {\displaystyle c_{t}} that can function as a generalized residual connection. The highway network (2015) [23] [24] applied the idea of an LSTM unfolded in time

to feedforward neural networks , resulting in the highway network. ResNet is equivalent to an open-gated highway network.

During the early days of deep learning, there were attempts to train increasingly deep models. Notable examples included the AlexNet (2012), which had 8 layers, and the VGG-19 (2014), which had 19 layers. [25] However, stacking too many layers led to a steep reduction in training accuracy, [26] known as the "degradation" problem. [1] In theory, adding additional layers to deepen a network should not result in a higher training loss , but this is what happened with VGGNet . [1] If the extra layers can be set as identity mappings , however, then the deeper network would represent the same function as its shallower counterpart. There is some evidence that the optimizer is not able to approach identity mappings for the parameterized layers, and the benefit of residual connections was to allow identity mappings by default. [6]

In 2014, the state of the art was training deep neural networks with 20 to 30 layers. [25] The research team for ResNet attempted to train deeper ones by empirically testing various methods for training deeper networks, until they came upon the ResNet architecture. [27]

Subsequent work

Wide Residual Network (2016) found that using more channels and fewer layers than the original ResNet improves performance and GPU-computational efficiency, and that a block with two 3×3 convolutions is superior to other configurations of convolution blocks. [28]

DenseNet (2016) [29] connects the output of each layer to the input to each subsequent layer:

x ■ + 1 = F (x 1 , x 2 , … , x ■ – 1 , x ■) {\displaystyle x_{\ell +1}=F(x_{1},x_{2},\dots ,x_{\ell -1},x_{\ell })}

Stochastic depth [30] is a regularization method that randomly drops a subset of layers and lets the signal propagate through the identity skip connections. Also known as DropPath , this regularizes training for deep models, such as vision transformers . [31]

ResNeXt (2017) combines the Inception module with ResNet. [32] [8]

Squeeze-and-Excitation Networks (2018) added squeeze-and-excitation (SE) modules to ResNet. [33] An SE module is applied after a convolution, and takes a tensor of shape R H × W × C {\displaystyle \mathbb {R} ^{H\times W\times C}} (height, width, channels) as input. Each channel is averaged, resulting in a vector of shape R C {\displaystyle \mathbb {R} ^{C}} . This is then passed through a multilayer perceptron (with an architecture such as linear-ReLU-linear-sigmoid) before it is multiplied with the original tensor. It won the ILSVRC in 2017. [34]

References

v

t

e

History timeline

timeline

Companies

Projects

Parameter Hyperparameter

Hyperparameter

Loss functions

Regression Bias–variance tradeoff Double descent Overfitting

Bias–variance tradeoff

Double descent

Overfitting

Clustering

Gradient descent SGD Quasi-Newton method Conjugate gradient method

SGD

Quasi-Newton method

Conjugate gradient method

Backpropagation

Attention

Convolution

Normalization Batchnorm

Batchnorm

Activation Softmax Sigmoid Rectifier

Softmax

Sigmoid

Rectifier

Gating

Weight initialization

Regularization

Datasets Augmentation

Augmentation

Prompt engineering

Reinforcement learning Q-learning SARSA Imitation Policy gradient

Q-learning

SARSA

Imitation

Policy gradient

Diffusion

Latent diffusion model

Autoregression

Adversary

RAG

Uncanny valley

RLHF

Self-supervised learning

Reflection

Recursive self-improvement

Hallucination

Word embedding

Vibe coding

Machine learning In-context learning

In-context learning

Artificial neural network Deep learning

Deep learning

Language model Large language model NMT

Large language model

NMT

Reasoning language model

Model Context Protocol

Intelligent agent

Artificial human companion

Humanity's Last Exam

Artificial general intelligence (AGI)

AlexNet

WaveNet

Human image synthesis

HWR

OCR

Computer vision

Speech synthesis 15.ai ElevenLabs

15.ai

ElevenLabs

Speech recognition Whisper

Whisper

Facial recognition

AlphaFold

Text-to-image models Aurora DALL-E Firefly Flux Ideogram Imagen Midjourney Recraft Stable Diffusion

Aurora

DALL-E

Firefly

Flux

Ideogram

Imagen

Midjourney

Recraft

Stable Diffusion

Text-to-video models Dream Machine Runway Gen Hailuo AI Kling Sora Veo

Dream Machine

Runway Gen

Hailuo AI

Kling

Sora

Veo

Music generation Riffusion Suno AI Udio

Riffusion

Suno AI

Udio

Word2vec

Seq2seq

GloVe

BERT

T5

Llama

Chinchilla AI

PaLM

GPT 1 2 3 J ChatGPT 4 4o o1 o3 4.5 4.1 o4-mini 5

1

2

3

J

ChatGPT

4

4o

o1

o3

4.5

4.1

o4-mini

5

Claude

Gemini Gemini (language model) Gemma

Gemini (language model)

Gemma

Grok

LaMDA
BLOOM
DBRX
Project Debater
IBM Watson
IBM Watsonx
Granite
PanGu-Σ
DeepSeek
Qwen
AlphaGo
AlphaZero
OpenAI Five
Self-driving car
MuZero
Action selection AutoGPT
AutoGPT
Robot control
Alan Turing
Warren Sturgis McCulloch
Walter Pitts
John von Neumann
Claude Shannon
Shun'ichi Amari
Kunihiko Fukushima
Takeo Kanade
Marvin Minsky
John McCarthy
Nathaniel Rochester
Allen Newell
Cliff Shaw
Herbert A. Simon
Oliver Selfridge
Frank Rosenblatt
Bernard Widrow
Joseph Weizenbaum
Seymour Papert
Seppo Linnainmaa

Paul Werbos
Geoffrey Hinton
John Hopfield
Jürgen Schmidhuber
Yann LeCun
Yoshua Bengio
Lotfi A. Zadeh
Stephen Grossberg
Alex Graves
James Goodnight
Andrew Ng
Fei-Fei Li
Alex Krizhevsky
Ilya Sutskever
Oriol Vinyals
Quoc V. Le
Ian Goodfellow
Demis Hassabis
David Silver
Andrej Karpathy
Ashish Vaswani
Noam Shazeer
Aidan Gomez
John Schulman
Mustafa Suleyman
Jan Leike
Daniel Kokotajlo
François Chollet
Neural Turing machine
Differentiable neural computer
Transformer Vision transformer (ViT)
Vision transformer (ViT)
Recurrent neural network (RNN)
Long short-term memory (LSTM)
Gated recurrent unit (GRU)
Echo state network
Multilayer perceptron (MLP)
Convolutional neural network (CNN)

Residual neural network (RNN)

Highway network

Mamba

Autoencoder

Variational autoencoder (VAE)

Generative adversarial network (GAN)

Graph neural network (GNN)

Category