ResNet50

ResNet50 is the Convolutional Residual Network comprised of 50 layers; ResNet50 is adopted to eliminate the Vanishing Gradient problem encountered with Deep Learning utilizing the Skip Connection concept. One of the prominent open source projects utilizing this network is the image-net.org.

DEEP LEARNING, VANISHING GRADIENT and SKIP CONCEPT

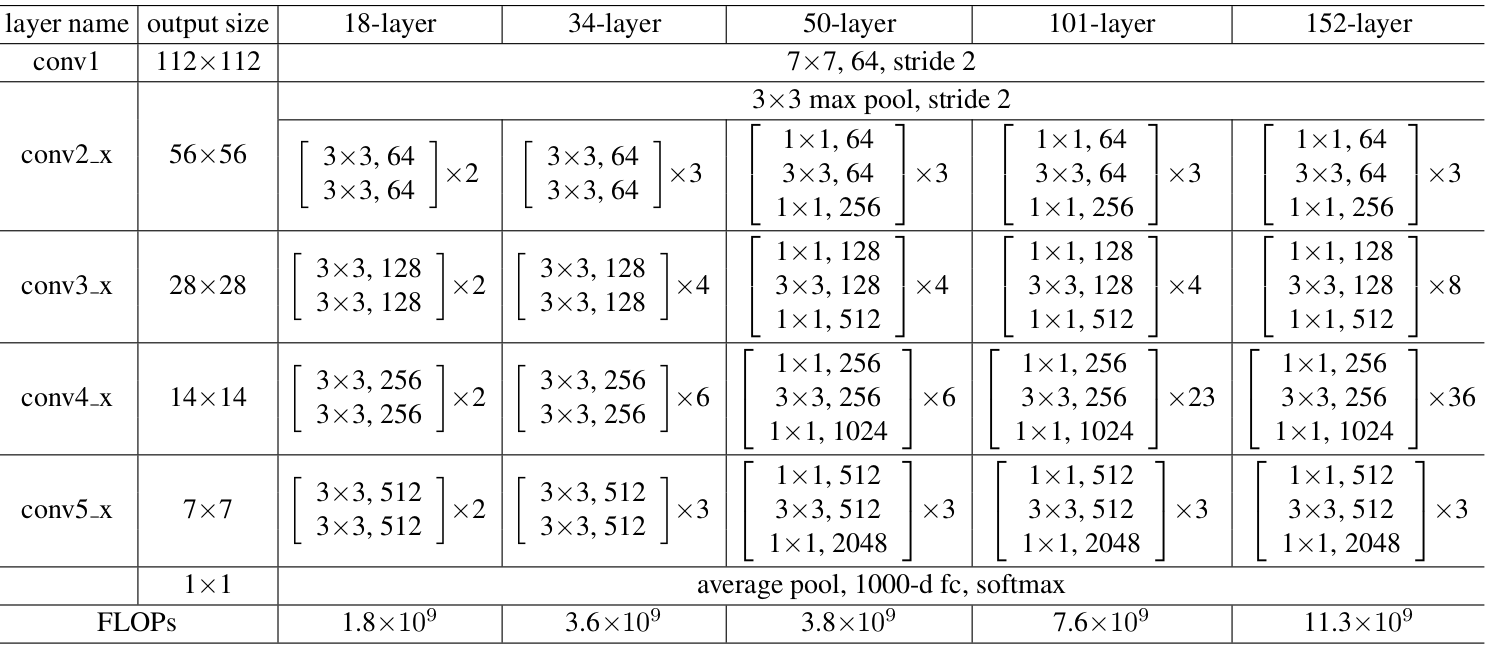
In Deep Learning as more layers using certain activation functions are added to neural networks, the gradients of the loss function approaches zero, making the network hard to train.

Gradients of neural networks are found using backpropagation. Simply put, backpropagation finds the derivatives of the network by moving layer by layer from the final layer to the initial one. By the chain rule, the derivatives of each layer are multiplied down the network (from the final layer to the initial) to compute the derivatives of the initial layers

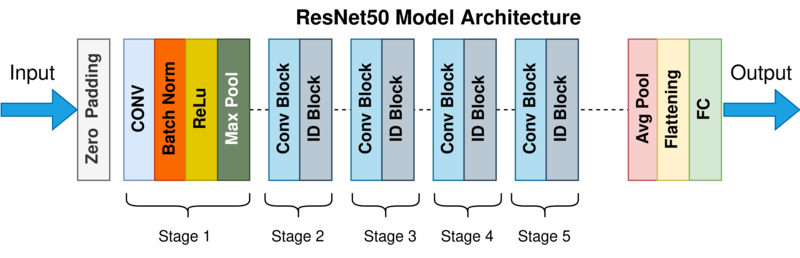
However, when n hidden layers use activation like the sigmoid function, n small derivatives are multiplied together. Thus, the gradient decreases exponentially as we propagate down to the initial layers leading to the Vanishing Gradient problem.

The concept of skip connection was first presented by ResNet. The skip connection is depicted in the diagram below. The figure on the left shows convolution layers stacked one on top of the other. We still stack convolution layers on the right, but we now also add the original input to the convolution block's output. This is known as a skip connection.

ResNet50 - ARCHITECTURE



ResNet50 – LAYERS OF PROCESS

The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. 

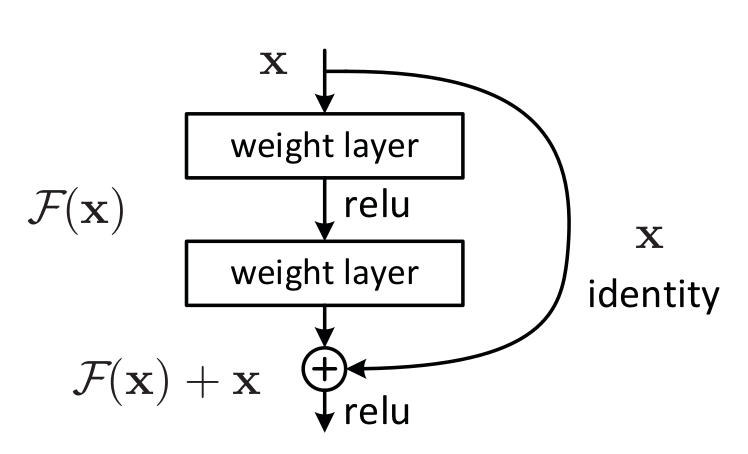
Convolutional Neural Networks and ResNet

Neural network-based machine learning algorithms don't need to be designed with particular rules that describe what to expect from the input. Instead, the neural net learning algorithm learns by analysing a large number of labelled instances (data with "answers") provided during training and using this answer key to determine which input properties are required to construct the correct output. After a sufficient number of examples have been processed, the neural network can begin to receive new, unknown inputs and deliver accurate results. Because the computer learns from experience, the more examples and types of inputs it sees, the more accurate the outputs become.

Given that Deep Convolutional Neural Networks excel at identifying low, mid, and high level characteristics from images, and that stacking additional layers often improves accuracy, the issue arises: is improving model performance as simple as stacking more layers?

With this question comes the issue of vanishing/exploding gradients, which have been solved in a variety of ways, allowing networks with tens of layers to converge. However, as deep neural networks converge, we see another problem: accuracy saturation and rapid degradation. This was not caused by overfitting, as one might expect, and adding more layers to a suitable deep model simply increased the training error.

Residual Networks, or ResNets, instead of learning unreferenced functions, learn residual functions with reference to the layer inputs. Residual nets allow these layers to suit a residual mapping rather than expecting that each few stacked layers directly match a desired underlying mapping. They build networks by stacking residual blocks on top of each other: a ResNet-50, for example, has fifty layers.



THE LAYERS

Using a sequential model for the network, from the first layer to the last layer of ResNet50 consists of the following elements

* Convolution
* Batch normalization
* Drop Out method
* Max Pooling Operation

Convolution

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system that can take an input image, assign relevance (learnable weights and biases) to various aspects/objects in the image, and distinguish between them. In comparison to other classification algorithms, ConvNet requires substantially less pre-processing. While basic approaches require hand-engineering of filters, ConvNets can learn these filters/characteristics with enough training.

The Convolution Operation's goal is to extract high-level characteristics from the input image, such as edges. ConvNets don't have to have just one Convolutional Layer. The first ConvLayer is traditionally responsible for capturing Low-Level information such as edges, colour, gradient direction, and so on. With further layers, the architecture adjusts to High-Level characteristics as well, giving us a network that understands all of the photos in the dataset in the same way that humans do.

Pooling Layer and max pooling operation

The Pooling layer, like the Convolutional Layer, is responsible for shrinking the Convolved Feature's spatial size. Through dimensionality reduction, the computer power required to process the data is reduced. It's also beneficial for extracting rotational and positional invariant dominant features, which helps keep the model's training process running smoothly.

Pooling can be divided into two types: maximum pooling and average pooling. The maximum value from the portion of the image covered by the Kernel is returned by Max Pooling.

Max Pooling can also be used to reduce noise. It eliminates all noisy activations and conducts de-noising as well as dimensionality reduction. Average Pooling, on the other hand, merely conducts dimensionality reduction as a noise suppression strategy.

Batch Normalization

bscaling the layer's output, for example, the enactments of a node from the previous layer, by explicitly normalising the activations of each input variable per mini-batch. Review that normalisation refers to rescaling data to a zero mean and one standard deviation.

By brightening the inputs to each layer, it would take a step closer to achieving stable input distributions, which would eliminate the negative effects of the internal covariate shift.

Normalizing the earlier layer's activations means that the subsequent layer's assumptions about the spread and distribution of inputs during the weight update won't change, at least not dramatically. This has the effect of stabilising and speeding up the deep neural network preparation training phase.

Drop Out method

Machine learning systems with a large number of parameters, such as deep neural networks, are extremely powerful. Overfitting, on the other hand, is a severe issue in such networks. Large networks are very sluggish to utilise, making it difficult to avoid overfitting by merging predictions from multiple large neural networks at test time. Dropout is a method of dealing with this issue. During training, units (along with their connections) are dropped at random from the neural network. This inhibits units from over-co-adapting. Dropout samples from an exponential number of different thinning networks are taken during training.

At test time, a single unthinned network with fewer weights can be used to approximate the effect of averaging the predictions of all these thinned networks. Overfitting is greatly reduced, and other regularisation approaches are significantly improved. Dropout increases neural network performance on supervised learning tasks in vision, speech recognition, document classification, and computational biology, yielding state-of-the-art results on a variety of benchmark data sets.

FILTER, CONDENSE and DETECT

The last three layers of the network consists of unique elements for filtering, condensing and detecting

Average Pooling

Average pooling method smoother out the image and hence the sharp features may not be identified when this pooling method is used. Max pooling selects the brighter pixels from the image. It is useful when the background of the image is dark and w due to the averaging operation over the feature maps, this makes the model more robust to spatial translations in the data. In other words, as long as the requisite feature is included / or activated in the feature map somewhere, it will still be “picked up” by the averaging operation.e are interested in only the lighter pixels of the image

Flattening

Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer

FC Layer

Completely connected (FC) The fully connected layer (FC) works with a flattened input, which means that each input is connected to all neurons. FC layers, if present, are commonly found near the end of CNN designs and can be utilised to optimise goals like class scores.