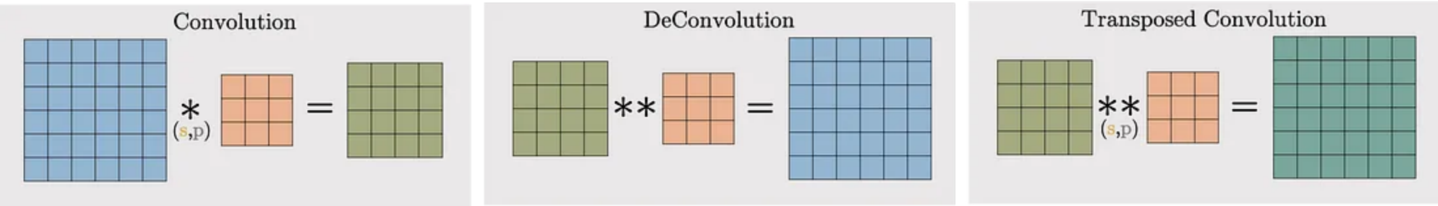
What is Transposed Convolutional Layer?

Explained through animated gifs and python code

Aqeel Anwar · Follow
Published in Towards Data Science
5 min read · Mar 6, 2020

Listen Share More

The transposed Convolutional Layer is also (wrongfully) known as the Deconvolutional layer. A deconvolutional layer reverses the operation of a standard convolutional layer i.e. if the output generated through a standard convolutional layer is deconvolved, you get back the original input. The transposed convolutional layer is similar to the deconvolutional layer in the sense that the spatial dimension generated by both are the same. Transposed convolution doesn't reverse the standard convolution by values, rather by dimensions only.



The transposed convolutional layer does exactly what a standard convolutional layer does but on a modified input feature map. Before explaining the similarity, let's first have a look at how does a standard convolutional layer works.

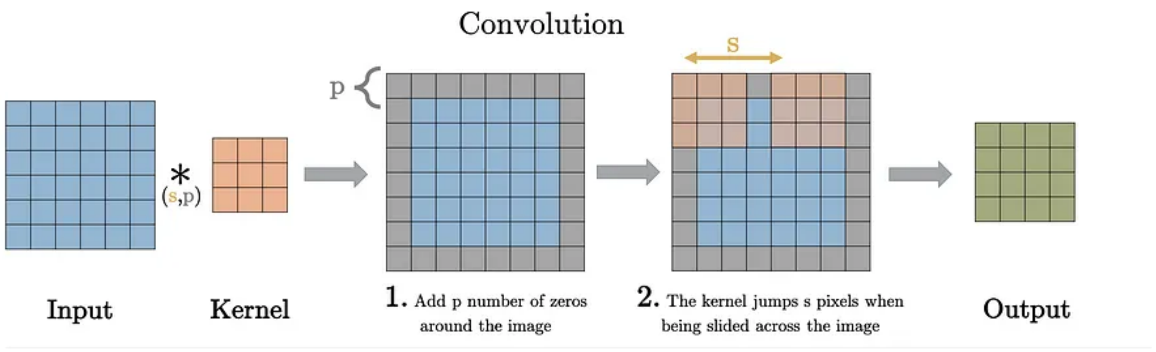
Standard Convolutional Layer:

A standard convolutional layer on an input of size *ixi* is defined by the following two parameters.

- **Padding** *(p):* The number of zeros padded around the original input increasing the size to *(i+2*p)x(i+2*p)*

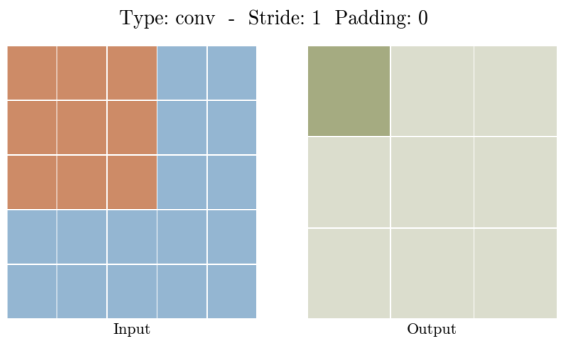
- **Stride *(s)*:** The amount by which the kernel is shifted when sliding across the input image.

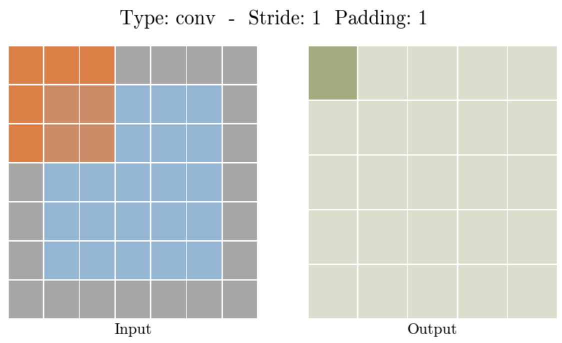
The figure below shows how a convolutional layer works as a two-step process.

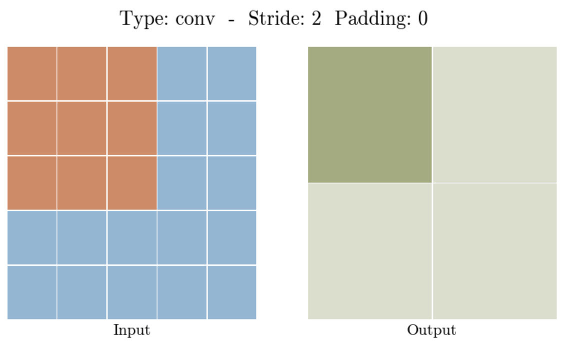

Convolution
*
(s,p)
p
s
Input
Kernel
1. Add p number of zeros around the image
2. The kernel jumps s pixels when being slided across the image
Output

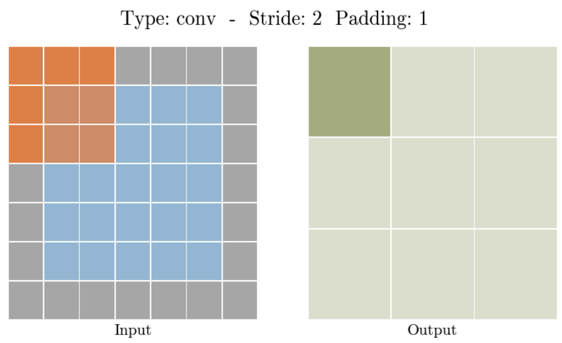
In the first step, the input image is padded with zeros, while in the second step the kernel is placed on the padded input and slid across generating the output pixels as dot products of the kernel and the overlapped input region. The kernel is slid across the padded input by taking jumps of size defined by the stride. The convolutional layer usually does a down-sampling i.e. the spatial dimensions of the output are less than that of the input.

The animations below explain the working of convolutional layers for different values of stride and padding.









For a given size of the input *(i),* kernel *(k),* padding *(p),* and stride *(s),* the size of the output feature map *(o)* generated is given by

$$o = \frac{i + 2p - k}{s} + 1$$

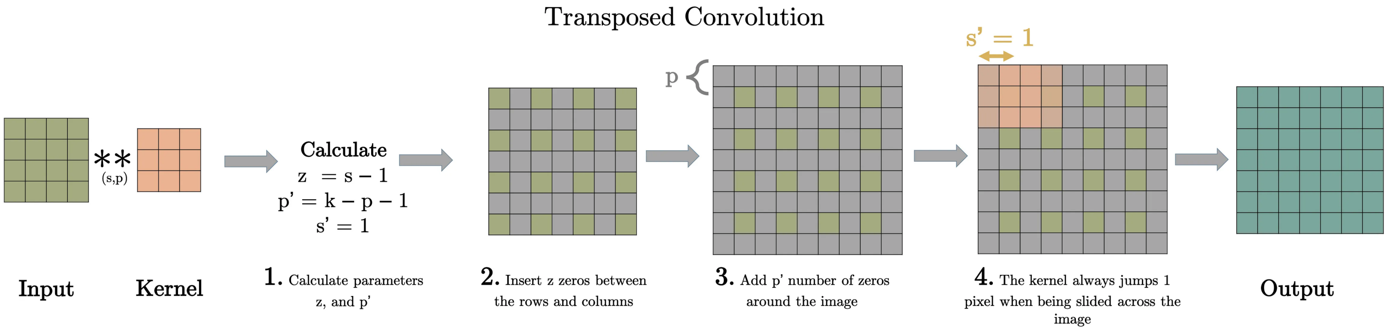
Transposed Convolutional Layer:

A transposed convolutional layer, on the other hand, is usually carried out for upsampling i.e. to generate an output feature map that has a spatial dimension greater than that of the input feature map. Just like the standard convolutional layer, the transposed convolutional layer is also defined by the padding and stride. These values of padding and stride are the one that hypothetically was carried out on the output to generate the input. i.e. if you take the output, and carry out a standard convolution with stride and padding defined, it will generate the spatial dimension same as that of the input.

Implementing a transposed convolutional layer can be better explained as a 4 step process

- **Step 1:** Calculate new parameters z and p'
- **Step 2:** Between each row and columns of the input, insert z number of zeros. This increases the size of the input to *(2*i-1)x(2*i-1)*
- **Step 3:** Pad the modified input image with p' number of zeros
- **Step 4:** Carry out standard convolution on the image generated from step 3 with a stride length of 1

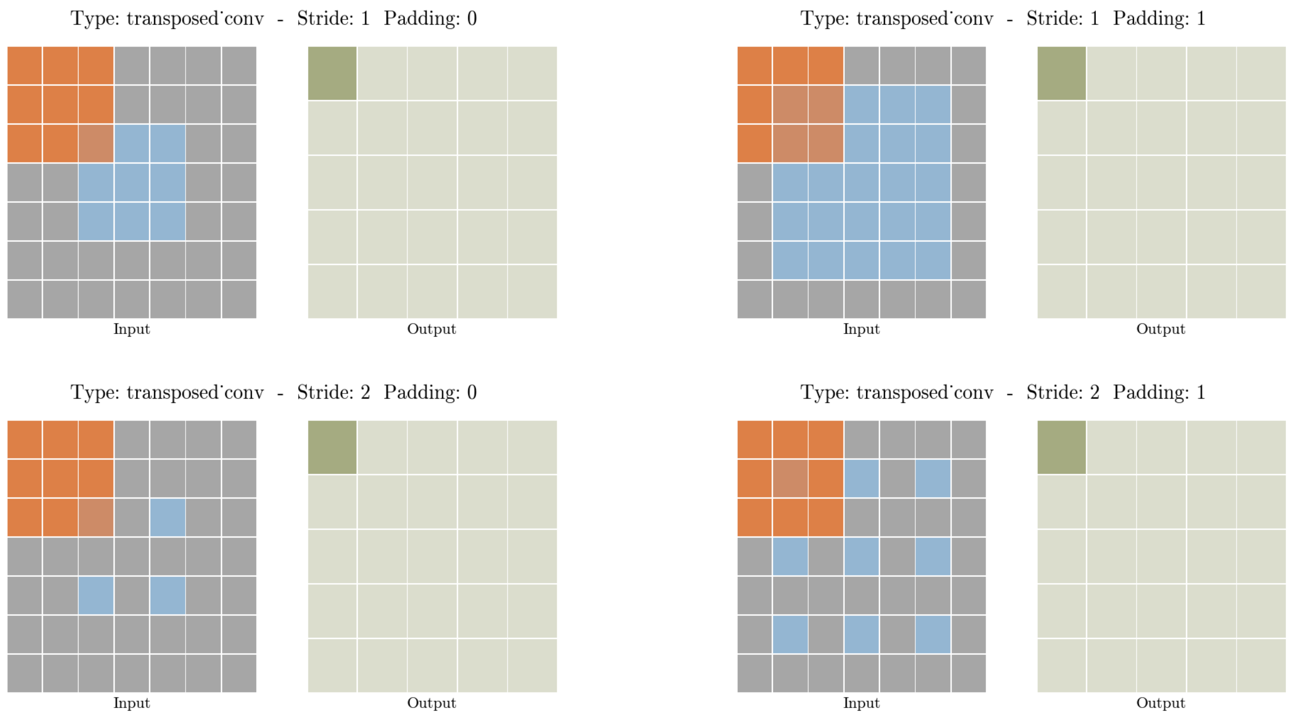
The complete steps can be seen in the figure below.



Transposed Convolution
s' = 1
p
**
(s,p)
Calculate
z = s − 1
p' = k − p − 1
s' = 1
Input
Kernel
1. Calculate parameters z, and p'
2. Insert z zeros between the rows and columns
3. Add p' number of zeros around the image
4. The kernel always jumps 1 pixel when being slided across the image
Output

Image By Author

The animations below explain the working of convolutional layers for different values of stride and padding.

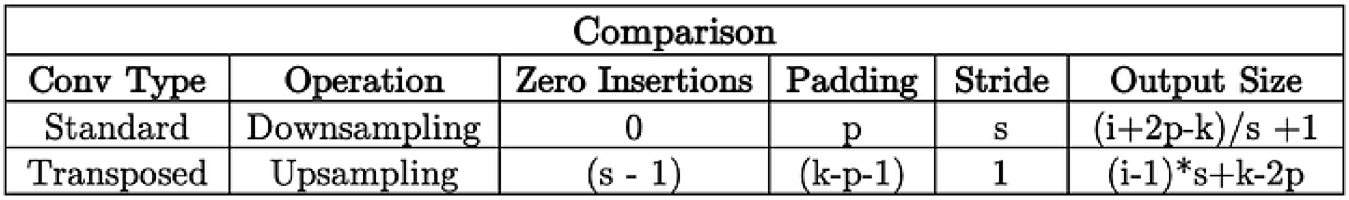

Type: transposed˙conv - Stride: 1 Padding: 0
Input
Output
Type: transposed˙conv - Stride: 1 Padding: 1
Input
Output
Type: transposed˙conv - Stride: 2 Padding: 0
Input
Output
Type: transposed˙conv - Stride: 2 Padding: 1
Input
Output

For a given size of the input ***(i)***, kernel ***(k)***, padding ***(p)***, and stride ***(s)***, the size of the output feature map ***(o)*** generated is given by

$$o = (i - 1) \times s + k - 2p$$

Summary:

The table below summarizes the two convolutions, standard and transposed.



Comparison					
Conv Type	Operation	Zero Insertions	Padding	Stride	Output Size
Standard	Downsampling	0	p	s	(i+2p-k)/s +1
Transposed	Upsampling	(s - 1)	(k-p-1)	1	(i-1)*s+k-2p

- The idea behind transposed convolution is to carry out trainable upsampling
- Transposed convolutions are standard convolutions but with a modified input feature map.

- The stride and padding **do not** correspond to the number of zeros added around the image and the amount of shift in the kernel when sliding it across the input, as they would in a standard convolution operation.

Python Code:

The gifs were generated using python. The complete code can be found at https://github.com/aqeelanwar/conv_layers_animation

Bonus:

Compact cheat sheets for this topic and many other important topics in Machine Learning can be found in the link below

Cheat Sheets for Machine Learning Interview Topics

A visual cheatsheet for ML interviews (www.cheatsheets.aqeel-anwar.com)

medium.com

If this article was helpful to you, feel free to clap, share and respond to it. If you want to learn more about Machine Learning and Data Science, follow me @ Aqeel Anwar or connect with me on *LinkedIn*.

Machine Learning

Convolution Neural Net

Convolutional Layer

Towards Data Science

Artificial Intelligence

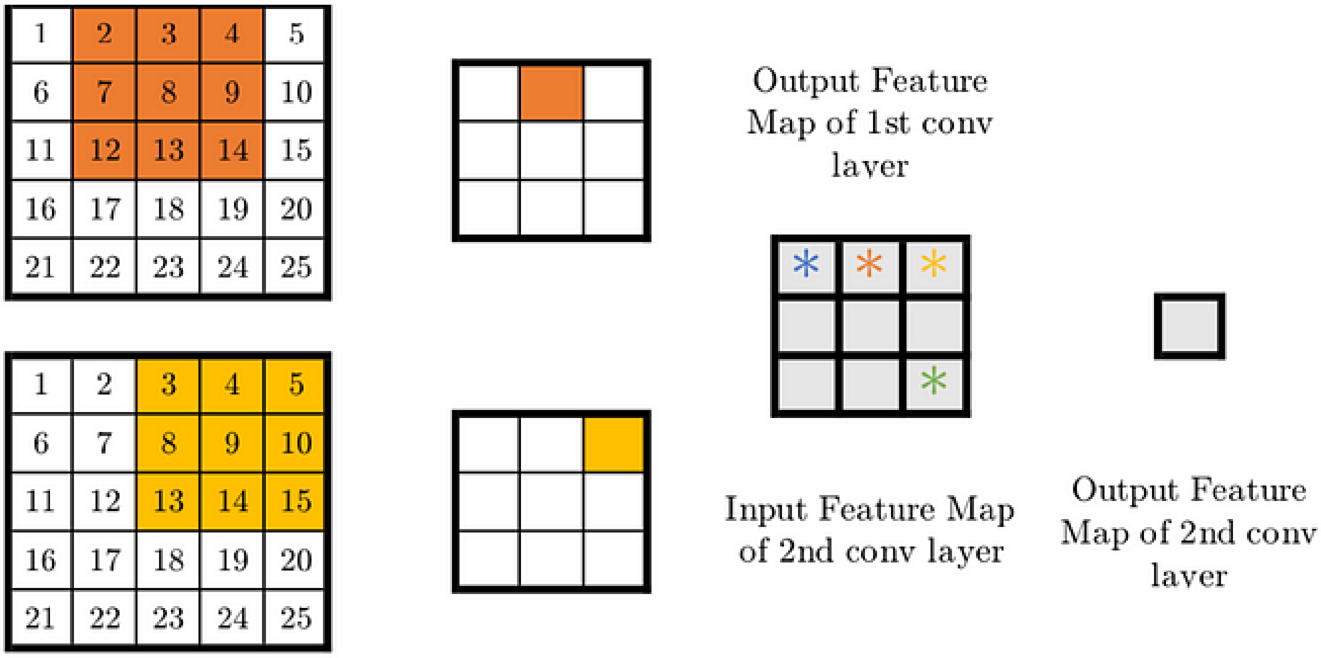
Follow

Written by Aqeel Anwar

3.3K Followers · Writer for Towards Data Science

Senior ML Engineer @NVIDIA | ex-Samsung | GeorgiaTech | Writer | Researcher | Traveler | www.aqeel-anwar.com | https://twitter.com/_aqeelanwar

More from Aqeel Anwar and Towards Data Science



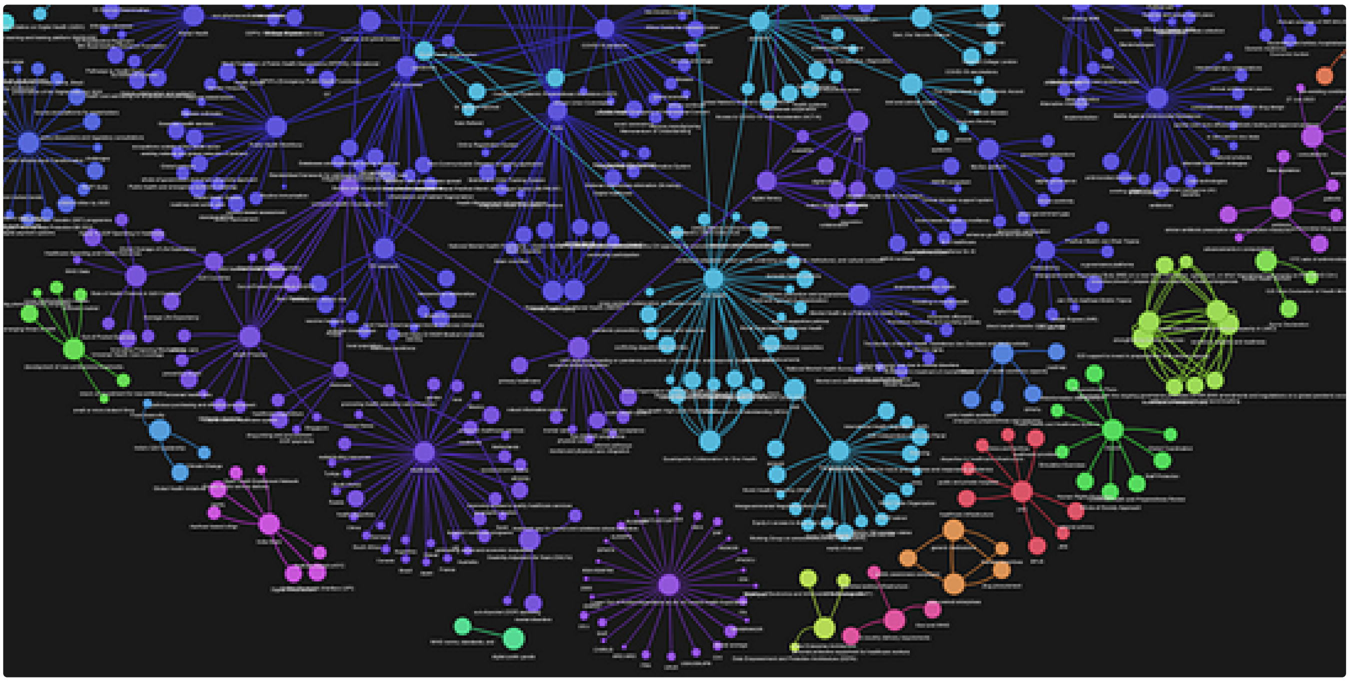
Aqeel Anwar in Towards Data Science

Difference between AlexNet, VGGNet, ResNet and Inception

AlexNet, VGGNet, ResNet and Inception explained

9 min read · Jun 7, 2019

731 10



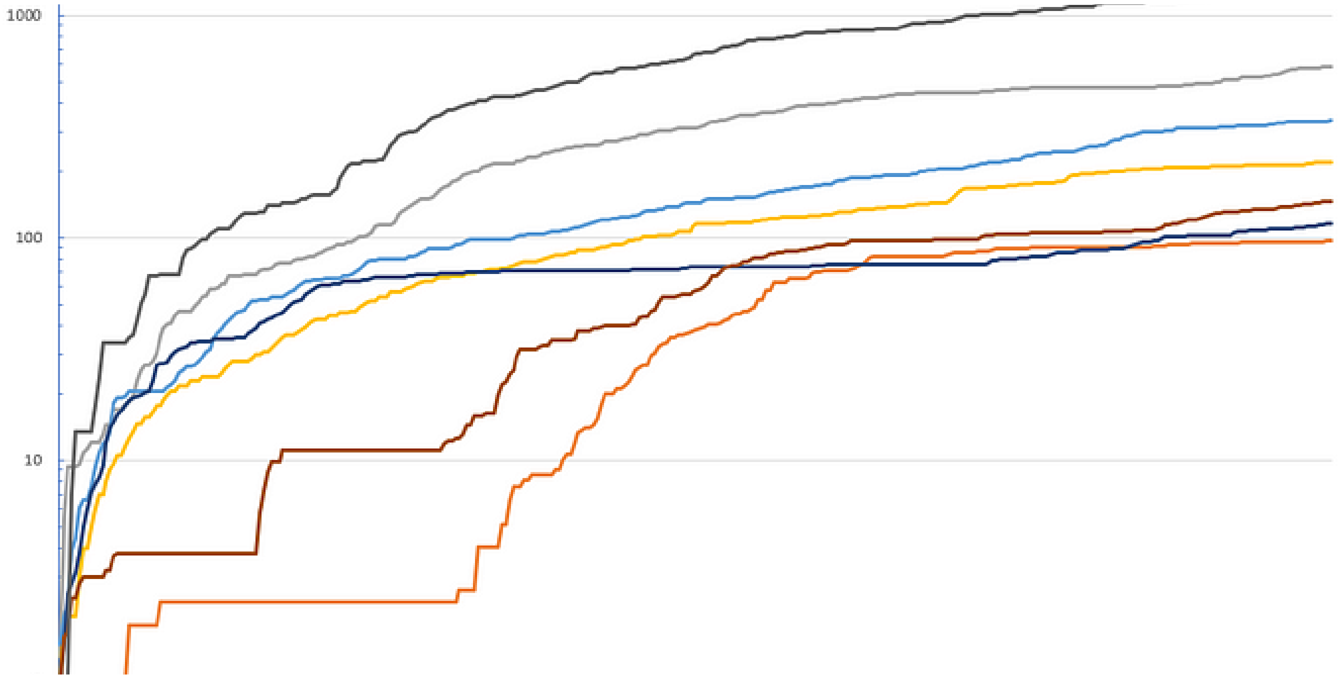
Rahul Nayak in Towards Data Science

How to Convert Any Text Into a Graph of Concepts

A method to convert any text corpus into a Knowledge Graph using Mistral 7B.

12 min read · Nov 10

👏 4.2K 💬 40



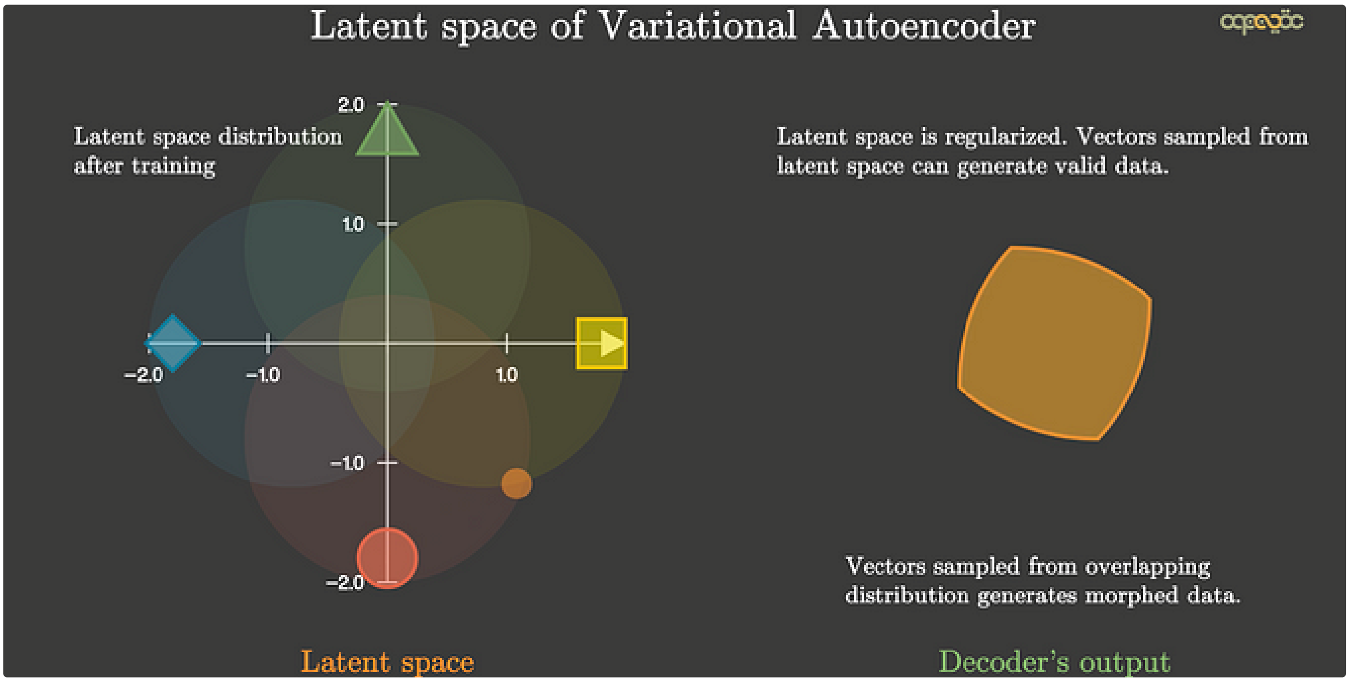
Pau Blasco i Roca in Towards Data Science

My Life Stats: I Tracked My Habits for a Year, and This Is What I Learned

I measured the time I spent on my daily activities (studying, doing sports, socializing, sleeping…) for 332 days in a row.

12 min read · Nov 20

4.4K 81


Latent space of Variational Autoencoder
Latent space distribution after training
2.0
1.0
−2.0
−1.0
1.0
−1.0
−2.0
Latent space is regularized. Vectors sampled from latent space can generate valid data.
Vectors sampled from overlapping distribution generates morphed data.
Latent space
Decoder's output

Aqeel Anwar in Towards Data Science

Difference between AutoEncoder (AE) and Variational AutoEncoder (VAE)

How can you compress data or even generate data from random values? That is what Autoencoder and Variational Autoencoder are.

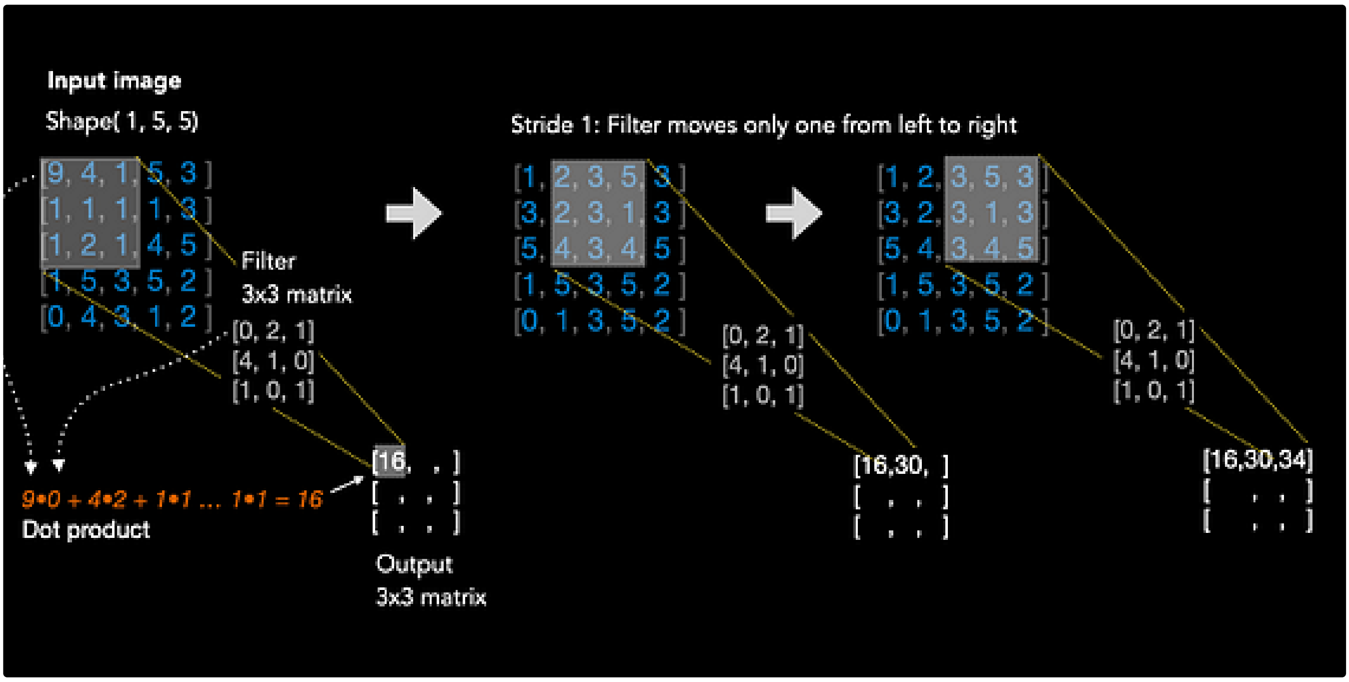
8 min read · Nov 3, 2021

430 7

See all from Aqeel Anwar

See all from Towards Data Science

Recommended from Medium

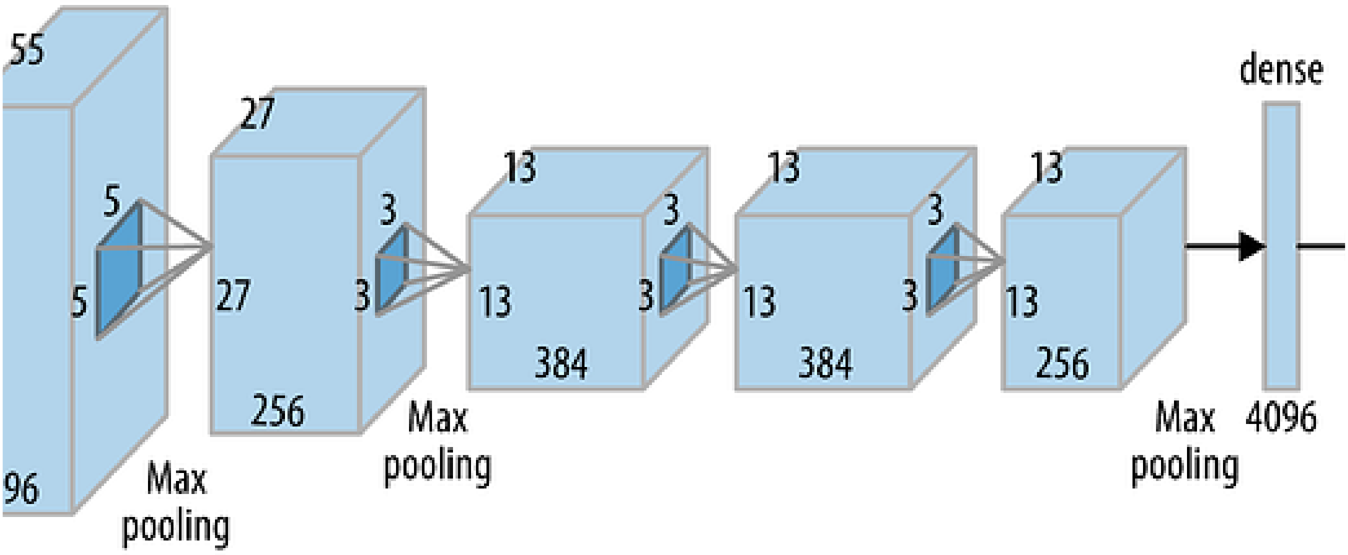


Frederik vl in Advanced Deep Learning

Understanding the Convolutional Filter Operation in CNN’s.

5 min read · Aug 18

125 2



Karan Singh

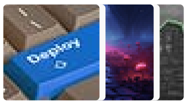
AlexNet, VGGNet, ResNet, and Inception

Exploring AlexNet: A Groundbreaking Deep Convolutional Network

6 min read · Jun 21

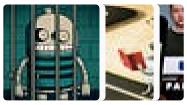
85 1

Lists



Predictive Modeling w/ Python

20 stories · 680 saves



AI Regulation

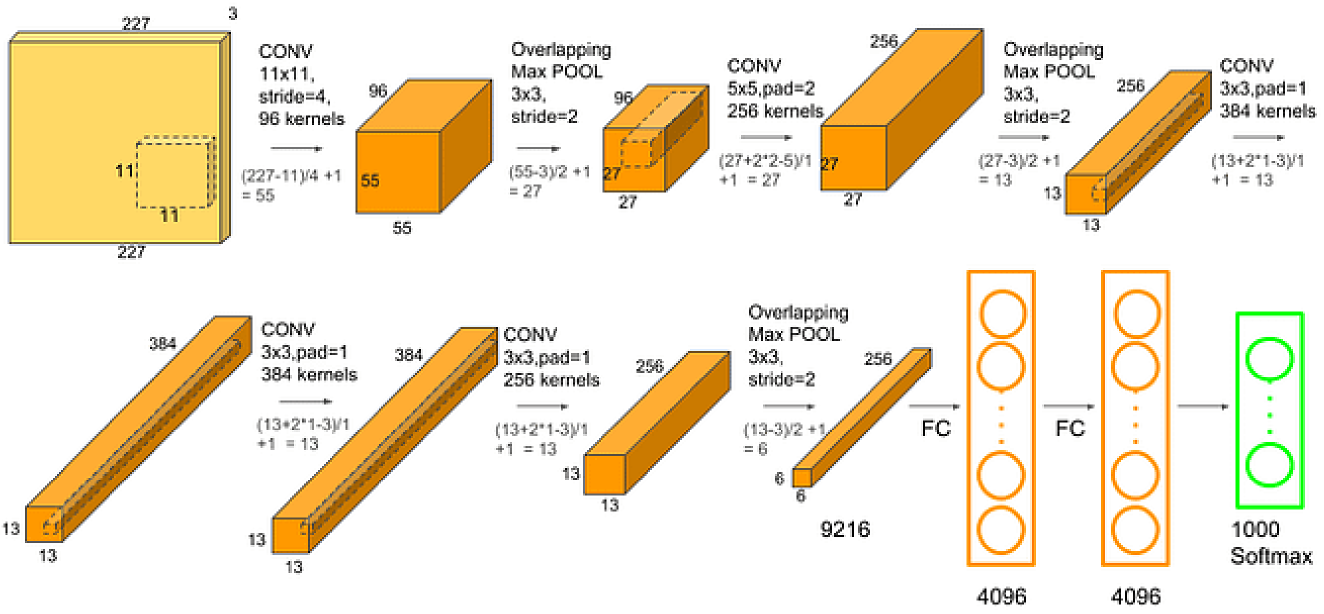
6 stories · 212 saves

Natural Language Processing

965 stories · 462 saves

Practical Guides to Machine Learning

10 stories · 774 saves



227
3
11
11
227
CONV 11x11, stride=4, 96 kernels
(227-11)/4 +1 = 55
96
55
55
Overlapping Max POOL 3x3, stride=2
(55-3)/2 +1 = 27
96
27
27
CONV 5x5,pad=2 256 kernels
(27+2*2-5)/1 +1 = 27
256
27
27
Overlapping Max POOL 3x3, stride=2
(27-3)/2 +1 = 13
256
13
13
CONV 3x3,pad=1 384 kernels
(13+2*1-3)/1 +1 = 13
384
13
13
CONV 3x3,pad=1 384 kernels
(13+2*1-3)/1 +1 = 13
384
13
13
CONV 3x3,pad=1 256 kernels
(13+2*1-3)/1 +1 = 13
256
13
13
Overlapping Max POOL 3x3, stride=2
(13-3)/2 +1 = 6
256
6
6
9216
FC
4096
FC
4096
1000 Softmax

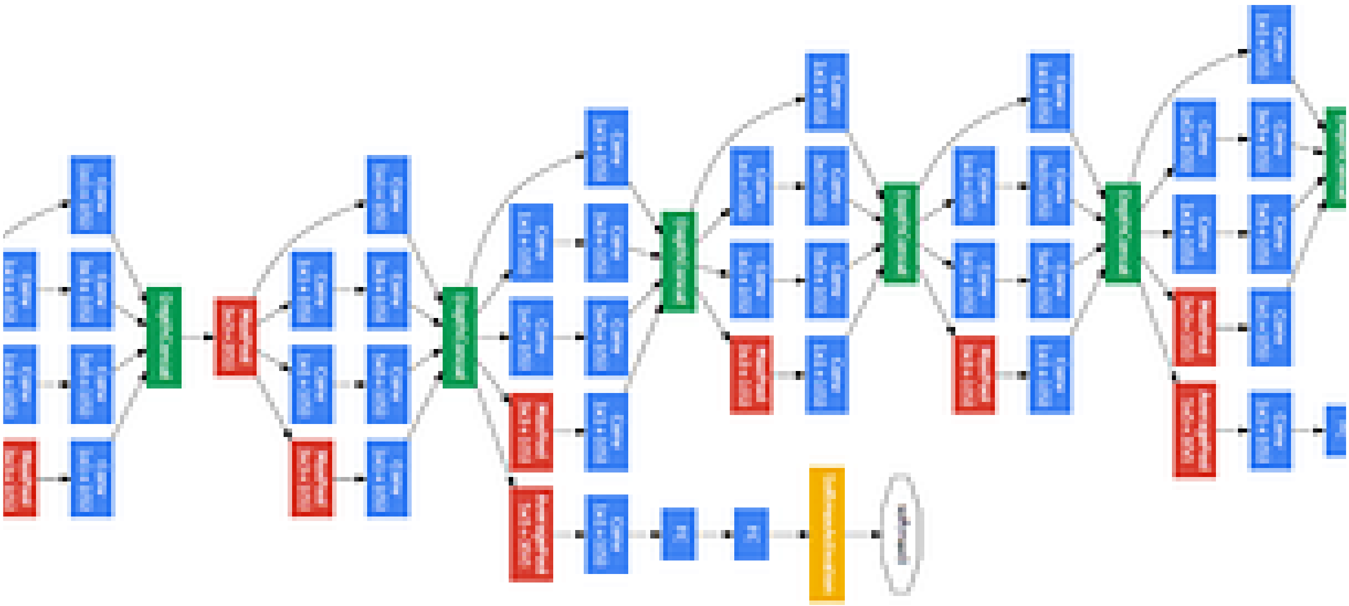
Carla Martins

Building Your First Convolution Neural Network — AlexNet

Using Keras Layers API

· 9 min read · Jul 17

54



Everton Gomede, PhD

Exploring GoogLeNet: A Revolutionary Deep Learning Architecture

Introduction

7 min read · Oct 1

5

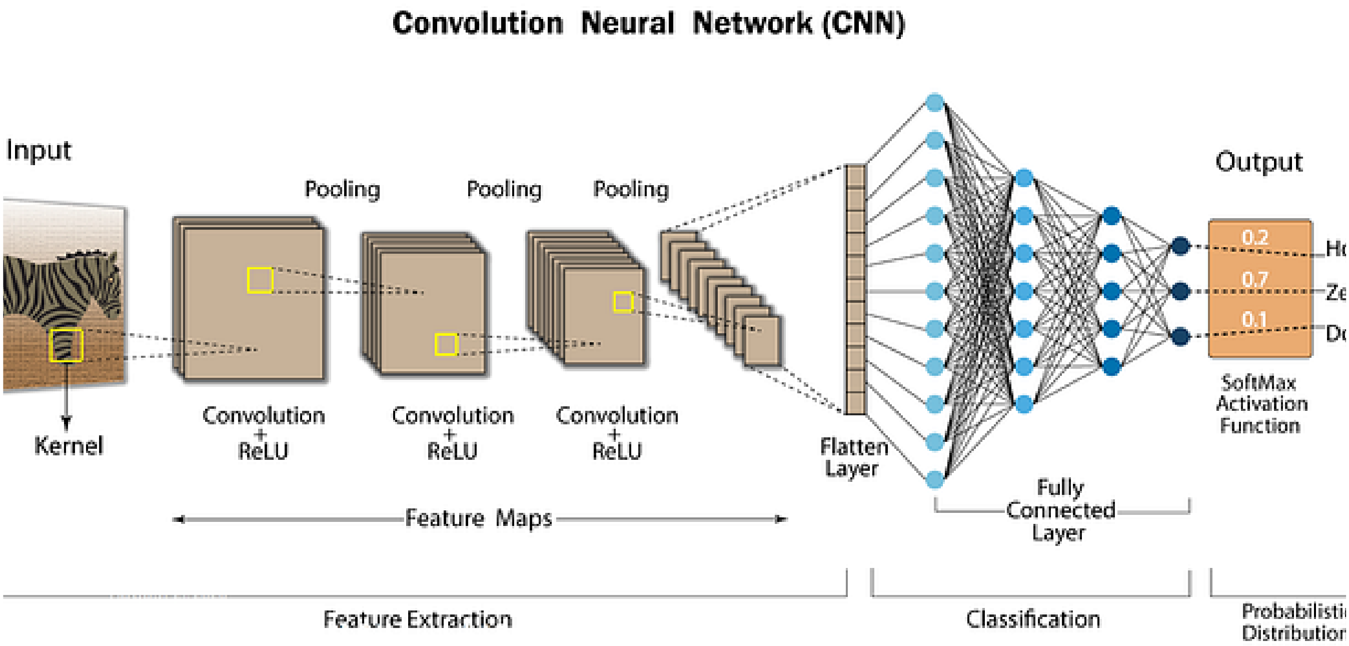
Vijay Vignesh in Level Up Coding

Computer Vision—Roadmap

The path from Beginner to Proficient: Free Learning Resources to Guide You

6 min read · Sep 12

120 1



Prathammodi

Convolutional Neural Networks for Dummies

So you want to learn about Convolutional Neural Networks, CNNs, huh? Well, you've come to the right place.

6 min read · Oct 14

51

See more recommendations