About CNN, kernels and scale/rotation invariance

Ask Question

13

I have a couple of questions that are confusing me regarding the CNN.

1) The features extracted using CNN are scale and rotation invariant?

2) The Kernels we use to convolution with our data are already defined in the literature? what kind of these kernels are? is it different for every application?

8

neural-networks deep-learning conv-neural-network

edited Oct 8 '16 at 3:09

GeoMatt22

10.2k 2 22 52

asked Oct 8 '16 at 1:09

Aadnan Farooq A

108 2 3 13

4 Answers

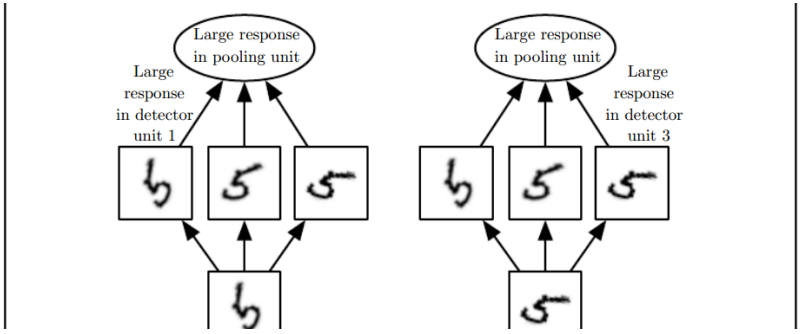
16

1) The features extracted using CNN are scale and rotation invariant?

A feature in itself in a CNN is not scale or rotation invariant. For more details, see: Deep Learning. *Ian Goodfellow and Yoshua Bengio and Aaron Courville. 2016:* http://egrcc.github.io/docs/dl/deeplearningbook-convnets.pdf ; http://www.deeplearningbook.org/contents/convnets.html:

> Convolution is not naturally equivariant to some other transformations, such as changes in the scale or rotation of an image. Other mechanisms are necessary for handling these kinds of transformations.

It is the max pooling layer that introduces such invariants:



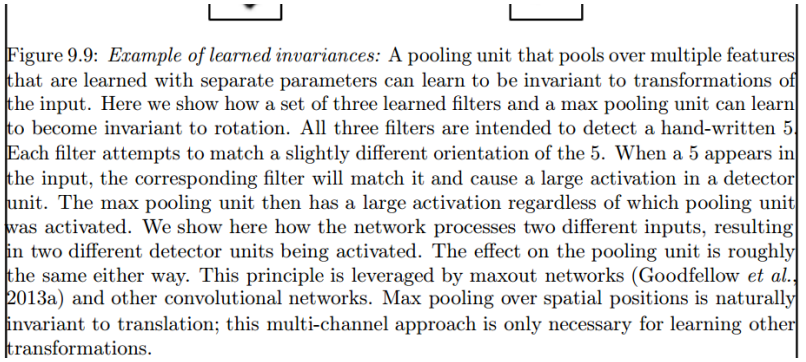


Figure 9.9: *Example of learned invariances:* A pooling unit that pools over multiple features that are learned with separate parameters can learn to be invariant to transformations of the input. Here we show how a set of three learned filters and a max pooling unit can learn to become invariant to rotation. All three filters are intended to detect a hand-written 5. Each filter attempts to match a slightly different orientation of the 5. When a 5 appears in the input, the corresponding filter will match it and cause a large activation in a detector unit. The max pooling unit then has a large activation regardless of which pooling unit was activated. We show here how the network processes two different inputs, resulting in two different detector units being activated. The effect on the pooling unit is roughly the same either way. This principle is leveraged by maxout networks (Goodfellow *et al.*, 2013a) and other convolutional networks. Max pooling over spatial positions is naturally invariant to translation; this multi-channel approach is only necessary for learning other transformations.

> 2) The Kernels we use to convolution with our data are already defined in the literature? what kind of these kernels are? is it different for every application?

The kernels are learnt during the training phase of the ANN.

edited Oct 8 '16 at 17:51

answered Oct 8 '16 at 1:54

Franck Dernoncourt
25.9k 16 101 217

I cannot speak to the details in terms of the current state of the art, but on the topic of point 1, I found this interesting. – GeoMatt22 Oct 8 '16 at 2:06

@Franck 1) That means, we don't take any special steps to make over system rotation invariant? and how about the scale invariant, is it possible to get the scale invariant from the max pooling? – Aadnan Farooq A Oct 8 '16 at 2:08

2) The kernels are the features. I didn't get that. [Here] (wildml.com/2015/11/…) They have mentioned that " For example, in Image Classification a CNN may learn to detect edges from raw pixels in the first layer, then use the edges to detect simple shapes in the second layer, and then use these shapes to deter higher-level features, such as facial shapes in higher layers. The last layer is then a classifier that uses these high-level features." – Aadnan Farooq A Oct 8 '16 at 2:10

(Cont (2)) Therefore, what i want to is the Kernel/Filter we use in our First, second and third layer are defined by the user or these kernels are fixed 3x3, 5x5 ....etc matrix to extract the edges? – Aadnan Farooq A Oct 8 '16 at 2:10

@AadnanFarooqA sorry, I should've said that kernels are learnt during the training phase. "Features" loosely refer to the input of a layer. I corrected the answer. – Franck Dernoncourt Oct 8 '16 at 4:27

@FranckDernoncourt Thanks for correction. But it confuse me more. we usually do convolution between kernel and the input image at first layer.. is it right? – Aadnan Farooq A Oct 8 '16 at 7:55

@AadnanFarooqA the ANN may have several convolution layers. – Franck Dernoncourt Oct 8 '16 at 14:38

@FranckDernoncourt Yes it may have.. but let's talk about the first only.. we convolve out input image with the kernel.. that's correct i think. I want to know how we define that kernel?? – Aadnan Farooq A Oct 9 '16 at 1:00

1 Note that the pooling you're talking about is refered to as cross-channel pooling and is *not* the type of pooling which is usually refered to when talking about "max-pooling", which only pools over spatial dimensions (not over different input channels). – Soltius Jun 13 '18 at 15:35

1 Does this imply a model which does not have any max-pool layers (most of the current SOTA architectures do not use pooling) is completely scale dependent? – shubhamgoel27 Aug 10 '18 at 7:46

4

I think there are a couple of things confusing you, so first things first.

Given a signal $x[n]$, and a kernel (also called a filter) $h[n]$, then the convolution of $x[n]$ with $h[n]$ is written as $y[n] = (x \star h)[n]$, and is computed via a sliding dot-product, mathematically given by:

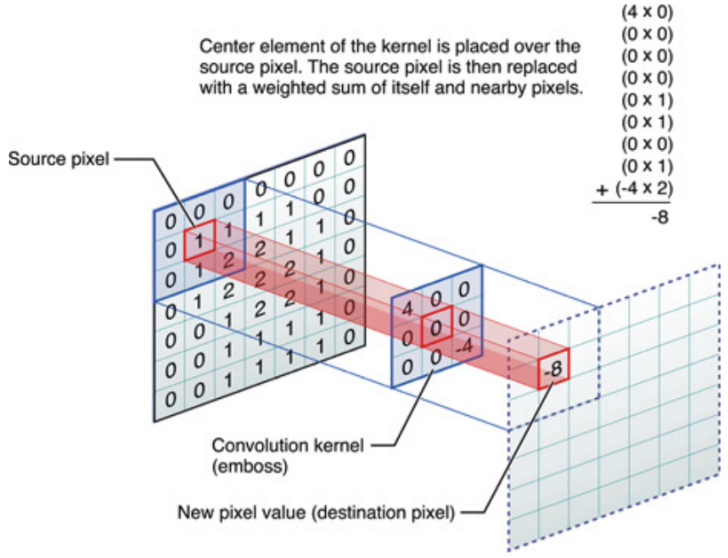
$$\sum_{-\infty}$$

$$y[n] = \sum_{m=-\infty}^{\infty} x[m]\ h[n-m]$$

The above if for one-dimensional signals, but the same can be said for images, which are just two-dimensional signals. In that case, the equation becomes:

$$I_{new}[r,c] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} I_{old}[u,v]\ k[r-u, c-v]$$

Pictorially, this is what is happening:



At any rate, the thing to keep in mind, is that the *kernel*, in actually *learned* during training a Deep Neural Network (DNN). A kernel is just going to be what you convolve your

(DNN). A kernel is just going to be what you convolve your input with. The DNN will learn the kernel, such that it brings out certain facets of the image (or previous image), that are going to be good for lowering the loss of your target objective.

This is the first crucial point to understand: Traditionally people have *designed* kernels, but in Deep Learning, we let the network decide what the best kernel should be. The one thing we do specify however, is the kernel dimensions. (This is called a hyperparameter, for example, 5x5, or 3x3, etc).

answered Oct 10 '16 at 16:06

Tarin Ziyaee
529 2 7

Nice explanation. Can you please answer the First part of the question. About the CNN is scale/rotation Invariant? – Aadnan Farooq A Oct 11 '16 at 4:58

1 @AadnanFarooqA I will do so tonight. – Tarin Ziyaee Oct 11 '16 at 19:49

1

Many authors including Geoffrey Hinton (who proposes Capsule net) try to solve the issue, but qualitatively. We try to address this issue quantitatively. By having all convolution kernels be symmetric (dihedral symmetry of order 8 [Dih4] or 90-degree increment rotation symmetric, et al) in the CNN, we would provide a platform for the input vector and resultant vector on each convolution hidden layer be rotated synchronously with the same symmetric property (i.e., Dih4 or 90-increment rotation symmetric, et al). Additionally, by having the same symmetric property for each filter (i.e., fully connected but weighs sharing with the same symmetric pattern) on the first flatten layer, the resultant value on each node would be quantitatively identical and lead to the CNN output vector the same as well. I called it transformation-identical CNN (or TI-CNN-1). There are other methods that

can also construct transformation-identical CNN using symmetric input or operations inside the CNN (TI-CNN-2). Based on the TI-CNN, a geared rotation-identical CNNs (GRI-CNN) can be constructed by multiple TI-CNNs with the input vector rotated by a small step angle. Furthermore, a composed quantitatively identical CNN can also be constructed by combining multiple GRI-CNNs with various transformed input vectors.

1. "Transformationally Identical and Invariant Convolutional Neural Networks through Symmetric Element Operators" https://arxiv.org/abs/1806.03636 (June 2018)
2. "Transformationally Identical and Invariant Convolutional Neural Networks by Combining Symmetric Operations or Input Vectors" https://arxiv.org/abs/1807.11156 (July 2018)
3. "Geared Rotationally Identical and Invariant Convolutional Neural Network Systems" https://arxiv.org/abs/1808.01280 (August 2018)

answered Oct 28 '18 at 20:32

Ben Lo
11 1

-1

I think max pooling can reserve translational and rotational invariances only for translations and rotations smaller than the stride size. If greater, no invariance

answered Mar 12 at 10:31

feynman
90 7

1 could you expand a little bit? We encourage answers on this site to be a little bit more detailed than this (right now, this looks more to a comment). Thank you! – Antoine Mar 17 at 8:12