Why and how are convolutional neural networks translation-invariant?

Dimitris Trigkakis · Follow
Ph.D. student in Computer Vision / Machine Learning · Nov 22

Convolutional neural networks are not translation-invariant. They have different responses to different translations.

They are translation-equivariant because convolutions will produce a shifted output if you shift the input.

The problem begins when you realize that this only applies to convolutions, and other layers may not learn how to respond correctly to the translated input.

Pooling operations come to the rescue, since they can reduce the effect of the shift. They can activate the same way whatever region caused an activation to happen below them. But this is not true for large translations.

So be very careful with translation-invariance assumptions. Recent research :

https://arxiv.org/pdf/2110.05861.pdf
shows that pretraining on Imagenet, with random cropping augmentations, essentially “teaches” translation invariance, but after fine-tuning on some custom single-location dataset, this effect can be forgotten. Still, it is much better than training from scratch, as the translation-invariance properties of the network are completely gone.

Be very careful!

Aditya Kumar Praharaj · Follow
I dream of smart machines · 7y

Originally Answered: Why and how convolutional neural networks are translation-invariant?

Recall what's happening in the convolution step. Given an image I of lets say dimensions $N \times N$ and a kernel K of lets say $M \times M$, then the convolution of the image with the kernel at position is given by $C_{ij} = \sum_{p,q} I_{i-p,j-q} K_{p,q}$. Now lets translate the image lets say k positions down and l positions right. What happens? All the pixels that were at position (i, j) now move to $(i + k, h + j)$. Now lets recompute convolution at this new position. Its $C_{i^{'},j^{'}} = \sum_{p,q} I_{i+k-p,j+l-q} K_{p,q}$. Now compare this with the notation above. We get $i^{'} = i + k$ and $j^{'} = j + l$. Hence the convolution for the image at the position (i, j) also shifts by (k, l) units.

In a more continuous sense, consider the calculus definition of convolution. Its $(f * g)(t) = \int_{-\inf}^{\inf} f(x)g(t - x)dx.$. Now translate g by l units. What happens? $(f * g)(t)_{new} = \int_{-\inf}^{\inf} f(x)g(t + l - x)dx = (f * g)(t + l)$.

One's not done yet. The pooling follows which basically pools all the possible translations within a certain receptive field into a single pixel. So basically every translation within this field is mapped to this pixel. This is what introduces translation invariance. This enables the network to learn the object features irrespective of wherever they are.

There are other kinds of invariances as well. All this happens because of weight sharing (visualize the kernels as weight matrices; certain submatrices of the weight matrix share the weights) in Convolutional Nets, which inherently allow this invariance. You can change this degree of weight sharing as well. See Tiled CNNs for more details.

Brando Miranda · Follow
member of MIT CBMM (Center for Brains Minds and Machines) · 5y ·

Convolution + Max pooling ≈ translation invariance (as far as I know from the deep learning book...also if you don’t remember what translation invariance is check out: What is translation invariance in computer vision and convolutional netral network? [https://stats.stackexchange.com/questions/208936/what-is-translation-invariance-in-computer-vision-and-convolutional-netral-netwo]).

Related questions More answers below

How exactly does max pooling create translation invariance?

Are CNNs rotation invariant?

How is a convolutional neural network able to learn invariant features?

Tapa Ghosh · Follow
Founder and CEO at Vathys · Updated 5y

The convolution operation on images is translation equivariant. Sliding the image over will have the same output but simply translated over. Therefore, convolutional neural networks which are composed of convolutions, are also translation-equivariant.

Upvote · 3 5

Micheal Bee · Follow
Former Principal Technical Architect at IBM Informix · 5y

If convolution changed an image to the extent it was unrecognizable, what could possibly be the benefit?

Upvote · 1

Youssef Kashef · Follow
PhD student in Computer Vision and Machine Learning · Upvoted by Sadid Hasan, Ph.D Computational Linguistics & Machine Learning, University of Lethbridge (2013) · 6y

Related **How is Fully Convolutional Network (FCN) different from the original Convolutional Neural Network (CNN)?**

Fully **convolutional** indicates that the neural network is composed of convolutional layers without any fully-**connected** layers or MLP usually found at the end of the network. A CNN with fully **connected** layers is just as end-to-end learnable as a fully **convolutional** one. The main difference is that the fully **convolutional** net is learning filters every where. Even the decision-making layers at the end of the network are filters.

A fully convolutional net tries to learn representations and make decisions based on **local** spatial input. Appending a fully connected layer enables the network to learn something using **global** information where the spatial arrangement of the input falls away and need not apply.

87.5K views · View 195 upvotes · View 9 shares

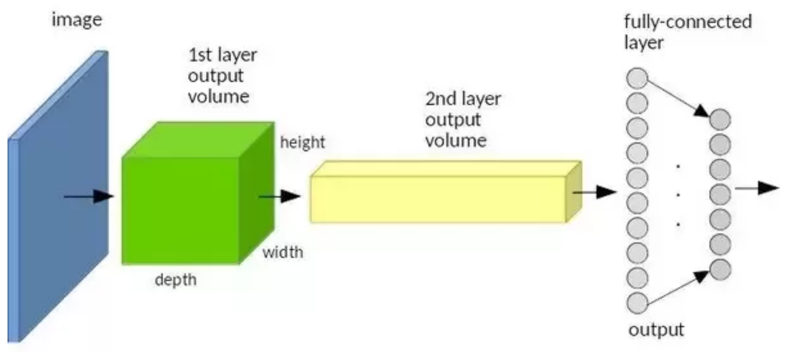
Upvote · 195 4 9

Jean Da Rolt · Follow
PhD, Computer Engineer, Professor · Upvoted by Joseph Antony, PhD Deep Learning & Computer Vision, DCU (2017) and Elijah Philpotts, M.S. Computer Science & Machine Learning, Georgia Institute of Technology (2016) · Updated 5y

Related **How is a convolutional neural network able to learn invariant features?**

After some thought, I do not believe that the pooling operation is the main reason for the translation invariant property in CNNs. I believe that invariance (at least to translation) is due to the convolution filters (not specifically the pooling) while the fully-connected layers at the end are “position-dependent”,

For instance, let's use the Fig. 1 as reference:



The blue volume represents the input image, while the green and yellow volumes represent layer 1 and layer 2 output activation volumes (see CS231n Convolutional Neural Networks for

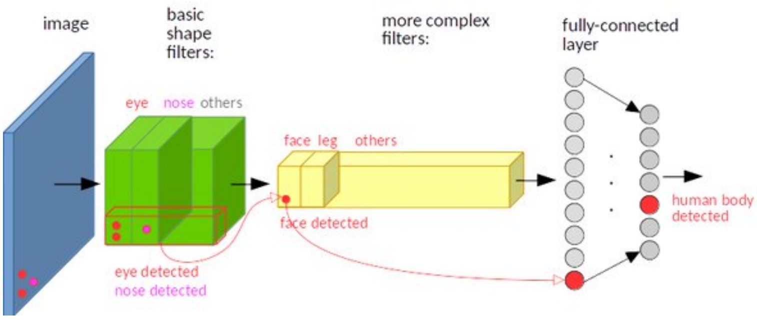
Messages
No messages
Connect with others on Quora by beginning a new conversation.
New message

Related questions

These volumes are built using a convolution plus a pooling operation. The pooling operation reduces the height and width of these volumes, while the increasing number of filters in each layer increases the volume depth.

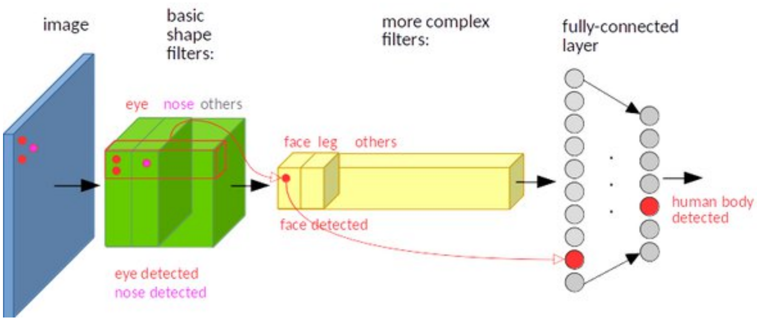
For the sake of the argument, let's suppose that we have very "ludic" filters, as show in Fig. 2:

- the first layer filters (which will generate the green volume) detect eyes, noses and other basic shapes (in real CNNs, first layer filters will match lines and very basic textures);
- The second layer filters (which will generate the yellow volume) detect faces, legs and other objects that are aggregations of the first layer filters. Again, this is only an example: real life convolution filters may detect objects that have no meaning to humans.



Now suppose that there is a face at one of the corners of the image (represented by two red and a magenta point). The two eyes are detected by the first filter, and therefore will represent two activations at the first slice of the green volume. The same happens for the nose, except that it is detected for the second filter and it appears at the second slice. Next, the face filter will find that there are two eyes and a nose next to each other, and it generates an activation at the yellow volume (within the same region of the face at the input image). Finally, the fully-connected layer detects that there is a face (and maybe a leg and an arm detected by other filters) and it outputs that it has detected an human body.

Now suppose that the face has moved to another corner of the image, as shown in Fig. 3:



The same number of activations occurs in this example, however they occur in a different region of the green and yellow volumes. Therefore, any activation point at the first slice of the yellow volume means that a face was detected, INDEPENDENTLY of the face location. Then the fully-connected layer is responsible to "translate" a face and two arms to an human body. In both examples, an activation was received at one of the fully-connected neurons. However, in each example, the activation path inside the FC layer was different, meaning that the FC layer needs to learn that a face may occur at both locations (i.e. there is no spatial invariance at these layers).

It must be noticed that the pooling operation only "compresses" the activation volumes, if there was no pooling in this example, an activation at the first slice of the yellow volume would still mean a face.

In conclusion, what makes a CNN invariant to object translation is the presence of convolution filters. Additionally, I believe that if a CNN is trained showing faces only at one corner during the learning process, the fully-connected layer may become insensitive to faces in other corners.

Are CNNs rotation inv
How is a convolutiona able to learn invariant
I was told that "poolir convolutional neural r
Why are normal neura translation-invariant, l
What is the difference between equivariance and invariance in...
Add question

Neural networks is a generic name for a large class of machine learning algorithms, including but not limited to: perceptrons, Hopfield networks, Boltzmann machines, fully connected neural networks, convolutional neural networks, recurrent neural networks, long short term memory neural networks, autoencoders, deep belief networks, generative adversarial networks and many more. Most of them are trained with an algorithm called backpropagation.

In the late eighties, early to mid nineties, the dominating algorithm in neural nets (and machine learning in general) was fully connected neural networks… (more)

Upvote · 96 4 11

Related What are the advantages of Fully Convolutional Networks over CNNs?

First the definition. A fully convolutional CNN (FCN) is one where all the learnable layers are convolutional, so it doesn't have any fully connected layer.

The key differences between a CNN which has a some convolutional layers followed by a few FC (fully connected) layers and an FCN (Fully Convolutional Network) would be:

- **Input image size:** If you don't have any fully connected layer in your network, you can apply the network to images of virtually any size. Because only the fully connected layer expects inputs of a certain size, which is why in architectures like AlexNet, you must provide inpu

… (more)

Upvote · 404 5 26

London and Mariusz Oszust, Ph.D. Computer Science, AGH University of Science and Technology (2013) · Updated Apr 15

Related How can Convolutional Neural Networks be fooled so easily (like mistaking a cat for a dog)?

I'm surprised the other answers don't touch on this: adversarial attacks.

First let's make it clear- all deep learning models trained to classify, whether they be CNNs, LSTMs or something else (yes LSTMs can be used to classify images by flattening them and considering a sequence of pixels one at a time) are vulnerable to adversarial perturbations to input.

What this means, is that there are clever ways to craft noise such that if you add noise to an input image the model will misclassify it, even though a human can easily tell the difference and can't even tell noise was added to it. Here is a … (more)

Upvote · 19

Related How can I improve validation accuracy in neural networks?

Well, there are a lot of reasons why your validation accuracy is low, let's start with the obvious ones :
1. Make sure that you are able to over-fit your train set
2. Make sure that you train/test sets come from the same distribution
3. Add drop out or regularization layers
4. shuffle your train sets while learning
5. sample your train set if you have unbalanced classification
6. pray to god if all of this doesn't work

Upvote · 6

Related How do you make convolutional neural networks invariant to scale?

The most common way would be augment your dataset by taking your pre-existing images, and zooming them in or out to different random scales so that by the end of this process you have a bunch of images of different scales. Use this for training and your CNN will likely be able to accomodate the 'scale-range' that you used to augment your dataset. The downside is,you'd need a bigger CNN with more layers to account for the increased complexity.

Try using this heuristic:
If you have N images in your dataset and M parameters in your CNN to start with, and your CNN works pretty well apart from scal… (more)

Upvote · 99 1 25

Related **What is a receptive field in a convolutional neural network?**

When dealing with high-dimensional inputs such as images, it is impractical to connect neurons to all neurons in the previous volume. Instead, we connect each neuron to only a local region of the input volume. The spatial extent of this connectivity is a hyperparameter called the **receptive field** of the neuron (equivalently this is the filter size). The extent of the connectivity along the depth axis is always equal to the depth of the input volume. It is important to emphasize again this asymmetry in how we treat the spatial dimensions (width and height) and the depth dimension: The connection… (more)

Upvote · 62 3 2

National Taiwan University and Abhinav Sharma, Product Designer at Quora, former ML Engineer at Facebook · 6y

Related **What is the purpose of using more than 1 fully-connected layer in a convolutional neural network?**

The purpose of convolutional layers in image processing nets is to build features from raw data. In layman's terms, they look for any objects they have seen before, but they don't make decisions about what they see.

After the forward pass reaches the “classifier” part of the neural network, you have a vector representation of *some* object your net has found. The relationship between different classes in training data may be very complex (for example, if you have lots of classes or they are similar to each other) - that's where you need something more sophisticated than a softmax layer.

Upvote · 36 2

Related **What is the motivation for pooling in convolutional neural networks (CNN)?**

Pooling in CNN is used mainly for -

1. **Dimension Reduction:** In deep learning when we train a model, because of excessive data size the model can take huge amount of time for training. Now consider the use of max pooling of size 5x5 with 1 stride. It reduces the successive region of size 5x5 of the given image to a 1x1 region with max value of the 5x5 region. Here pooling reduces the 25 (5x5) pixel to a single pixel (1x1) to avoid curse of dimensionality.
2. **Rotational/Position Invariance Feature Extraction :** Pooling can also be used for extracting rotational and position invariant feature. Consider th

… (more)

Upvote · 25 2 7

Related **What are temporal convolutional neural networks?**

Temporal Convolutional Networks, or simply TCN is a variation over Convolutional Neural Networks for sequence modelling tasks. Rather, it’s quite a descriptive term for a family of architectures.

Motivation:

- TCNs exhibit longer memory than recurrent architectures with the same capacity.
- Constantly performs better than LSTM/GRU architectures on a vast range of tasks (Seq. MNIST, Adding Problem, Copy Memory, Word-level PTB…).
- Parallelism, flexible receptive field size, stable gradients, low memory requirements for training, variable length inputs.

Some **distinguishable characteristics for TCNs** are:

1. T

… (more)

Upvote · 7 · 1

Related **What are the pros and cons of convolutional neural networks?**

I work with unstructured text so my opinion will be based on working with text data only:

the pros of convolutional neural network:

1. speed!!
2. perfect when orthology matters or if you are working with characters such as emojis or byte
3. great for short texts (e.g., headlines)
4. most likely can improve your text classifier model cnn by adding a linear layer (like cbow) after for pooling layer

cons of convolutional neural network:

1. can be more difficult with long texts when you really need a large vocabular
2. does not have the benefits that some of these larger transformer models
3. e.g.,trained a model for 18 hours

... (more)

Upvote · 9 · 1

Related **Why are only convolutional neural networks used for images, rather than other deep learning techniques?**

Correction: Convolutions are generally used on images with RNNs sometime used. Also most of convolutional neural networks have FC layers, pure convolutional networks are rare.

So Why ?

Because you could theoretically use fully DenseNets of images as well, but that would require even larger datasets than current convnets do!! Convnets have a unique architecture where convolutions lower the dimensions of image by extracting local features, so that the data on which dense layer are to be trained is lesser and they can be made to fit.

If you ask Convnets vs RNNs in images, it just happens that convne... (more)

Upvote · 75 · 1 · 7

Related **What is the difference between a convolutional neural network and a multilayer perceptron?**

Convolutional Neural Networks are MLPs with a special structure.

CNNs have repetitive blocks of neurons that are applied across space (for images) or time (for audio signals etc). For images, these blocks of neurons can be interpreted as 2D convolutional kernels, repeatedly applied over each patch of the image. For speech, they can be seen as the 1D convolutional kernels applied across time-windows. At training time, the weights for these repeated blocks are 'shared', i.e. the weight gradients learned over various image patches are averaged.

The reason for choosing this special structure, is t... (more)

Upvote · 10

Related **Can I use CNN for deep learning with textual data, and how?**

There are a lot of papers and works have shown that one-dimensional convolution is suitable for textual data, and sometimes even out-performs RNN.

Though RNN better exploits the fact that textual data are much connected within several words, CNNs are powerful at summarization and are much faster to train, because of the

1. Keras example on use CNN for sentiment classification (achieve 7% higher accuracy than LSTM approach and is 15 times faster to train).
2. A recent paper systemically shows CNN out-preform RNN in sentiment classif

... (more)

Upvote · 7

Related What is the "spatial information" exactly in convolutional neural network?

The pixels in the image are arranged in a specific way, like the pixel at some (x,y) coordinate of image. If i change a pixel from one place to another, the visual appearance of image will be changed.

In traditional image processing algorithms, we do not used to bother much about the specific arrangement of pixels in image and we just extract statistical properties like mean, variance or some geometric attributes like edges (we do not preserve the spatial information). But when we use ConvNets, the spatial information is preserved (because neurons in convolutional layers process each part of im... (more)

Upvote · 51 1

Machine Learning · 7y

Related How does the conversion of last layers of CNN from fully connected to fully convolutional allow it to process images of different size?

Originally Answered: How does the conversion of last layers of CNN from fully connected to convolutional fully connected allows it to process images of different size?

"There's no such thing as fully connected layer" (Yann LeCun - In Convolutional Nets, there is no such thing...)

In short, the decision making layers at the end of an conv. net may just as well be kernels of a conv. layer. If you used the weights of these layers as weights of a kernel and convolved it with the feature maps produced by the preceeding convolutions you are effectively performing the classification on a local patch. This yields a coarse localization of which classes were recognized were.

Some basics first

Conv. Nets for image classification commonly take an image (3-dim matrix of si... (more)

Upvote · 31 3

Related What are the advantages of a convolutional neural network (CNN) compared to a simple neural network from the theoretical and practical perspective?

Here's what I know ,

1. The usage of CNNs are motivated by the fact that they can capture / are able to learn relevant features from an image /video (sorry I dont know about speech / audio) at different levels similar to a human brain. This is **feature learning !** Conventional neural networks cannot do this.
2. Another main feature of CNNs is **weight sharing**. Lets take an example to explain this. Say you have a one layered CNN with 10 filters of size 5x5. Now you can simply calculate parameters of such a CNN, it would be 5*5*10 weights and 10 biases i.e **5* 5*10 + 10 = 260 parameters**. Now lets take a simp

... (more)

Upvote · 10

Related What is the difference between FCN (Fully Convolutional Network) and autoencoders?

Messages

No messages

Connect with others on Quora by beginning a new conversation.

New message

An autoencoder is a Neural Network that tries to regenerate the input as output creating an information bottleneck in the middle, which can be used as a dense representation of the input.

FCNs can be used as autoencoders and also for supervised tasks like object detection, classification and segmentation. Famous FCN architectures are FCN(X) architectures for segmentation, UNets, ResNets (they have no FC layers) for classification, Hour... (more)

Related questions

How exactly does max pooling create translation invariance?

Are CNNs rotation invariant?

How is a convolutional neural network able to learn invariant features?

I was told that "pooling" in a convolutional neural network leads to invariance. Is there a rigorous mathematical proof of this? Is this true for any type of pooling or only some types ...

Why are normal neural networks not translation-invariant, but convolutional nets are translation-invariant?

What is the difference between equivariance and invariance in Convolution neural networks?

How do you make convolutional neural networks invariant to scale?

What is shift invariance in a convolutional neural network (CNN)?

What is the benefit of using average pooling rather than max pooling?

Is CNN translation invariant?

What does it mean that a neural network is invariant to permutation? When does this happen?

What does a 1x1 convolutional layer do?

Do convolutional neural networks learn to be spatially invariant at the last layer of the network (fully connected layer)? Convolution layers produce spatially equivariant output but what...

How can Convolutional Neural Networks be fooled so easily (like mistaking a cat for a dog)?

What is convolutional neural network in layman's terms?