

[Course](#)[Schedule](#)[Discussion](#)[Progress](#)

Support

[All Topics](#) > [Mod4-3 Tutorials](#)[Add a Post](#)

Search all posts

[Search](#)

Show all posts ▾

by recent activity ▾

- | | | |
|-------------------------------------|--|---|
| <input checked="" type="checkbox"/> | Help with determining the number of parameters
Hi all -- kind of struggling determining the number of parameters. Has anyone been able to ... | 8 |
| | Question 8
Which CNN we are supposed to use to answer 8. The default one, the one with the 2 pooling... | 2 |
| <input checked="" type="checkbox"/> | module 'cntk' has no attribute 'LeakyRelu'
for question 3, I am getting this error when using LeakyRelu. anyone knows why? Other one... | 5 |
| | Best Average test error
My best Average test error is 0.87. What is yours? | 7 |
| | Question 3. Best result after trying all possibilities marked as incorrect.
In Question 3 (*Which of the following activation function results in the lowest average test ... | 7 |

Help with determining the number of parameters



question posted 19 days ago by [amthomas46](#)

Hi all -- kind of struggling determining the number of parameters. Has anyone been able to find some sort of hard and fast formula or cheat sheet? I'd really appreciate it!

Thanks!

Related to: [Mod4-3 Tutorials / Homework Assignment](#)

This post is visible to everyone.

[jonsan21](#) (Staff)



18 days ago - marked as answer 18 days ago by [jonsan21](#) (Staff)



I suggest you review the notebook.

Understanding Parameters:

Our model has 2 convolution layers each having a weight and bias. This adds up to 4 parameter tensors. Additionally the dense layer has weight and bias tensors. Thus, the 6 parameter tensors. Remember that in a convolutional layer, the number of parameters is not dependent on the number of nodes, only on the shared weights and bias of each filter.

Let us now count the number of parameters:

- First convolution layer: There are 8 filters each of size $(1 \times 5 \times 5)$ where 1 is the number of channels in the input image. This adds up to 200 values in the weight matrix and 8 bias values.
- Second convolution layer: There are 16 filters each of size $(8 \times 5 \times 5)$ where 8 is the number of channels in the input to the second layer (= output of the first layer). This adds up to 3200 values in the weight matrix and 16 bias values.
- Last dense layer: There are $16 \times 7 \times 7$ input values and it produces 10 output values corresponding to the 10 digits in the MNIST dataset. This corresponds to $(16 \times 7 \times 7) \times 10$ weight values and 10 bias values.

Could you please explain how did you arrive at the 16x7x7 input values to the dense layer? Is it 16 filters in second layer $28/(2*2)$, where 28 is the image size and 2 is the stride in first and second layer resulting in 7?



But now I am wondering about pooling. Adding two pooling layers with a stride of 2:

```
C.layers.AveragePooling((5,5), strides=(2,2), pad=True)
```

and setting the convolution to a stride of 1 in the convolution layer, what does it result in and why? I would think $28/(2*2)$

I did verified it by:

```
print(z.classify.parameters)
```

But than the answer : 11274 (since avg pooling does not add any parameters) should be correct, which you (staff?) claim is not.

Could you please check?

posted 3 days ago by [witoldw](#)

Output of last convolution has 16 filters, and the shape of each filter is 7 x 7, when flattened this becomes 16 x 7 x 7. Pooling layers in themselves do not add any parameters. However, use of strides of 2, means you are going to skip every other input, so the shape of the output of the pooling layer is smaller. If you set the stride to be 1, then the output shape will not skip every other pixel during the pooling stage.



posted 2 days ago by [SayanPathak](#) (Staff)

But Output of last convolution layer when flattened will be 16 x 7 x 7 if one of the following is true:



(1) both convolution layers have 2x2 stride and 'same' padding and there is no pooling layer

(2) both convolution layers have 1x1 stride and 'same' padding and there are 2 pooling layers in between with size and stride 2x2

Output of last convolution layer when flattened will be 16 x 28 x 28 when both convolution layers have 1x1 stride and 'same' padding and there is no pooling layer, is not that correct?

posted a day ago by [sandipan_dey](#)

Add a comment



Add a Response

2 other responses

[witoldw](#)

2 days ago



Could you please explain how did you arrive at the 16x7x7 input values to the dense layer? Is it 16 filters in second layer $28/(2*2)$, where 28 is the image size and 2 is the stride in first and second layer resulting in 7?

But now I am wondering about pooling. Adding two pooling layers with a stride of 2:

`C.layers.AveragePooling((5,5), strides=(2,2), pad=True)`
and setting the convolution to a stride of 1 in the convolution layer, what does it result in and why? I would think $28/(2*2)$

I did verified it by:

```
print(z.classify.parameters)
```

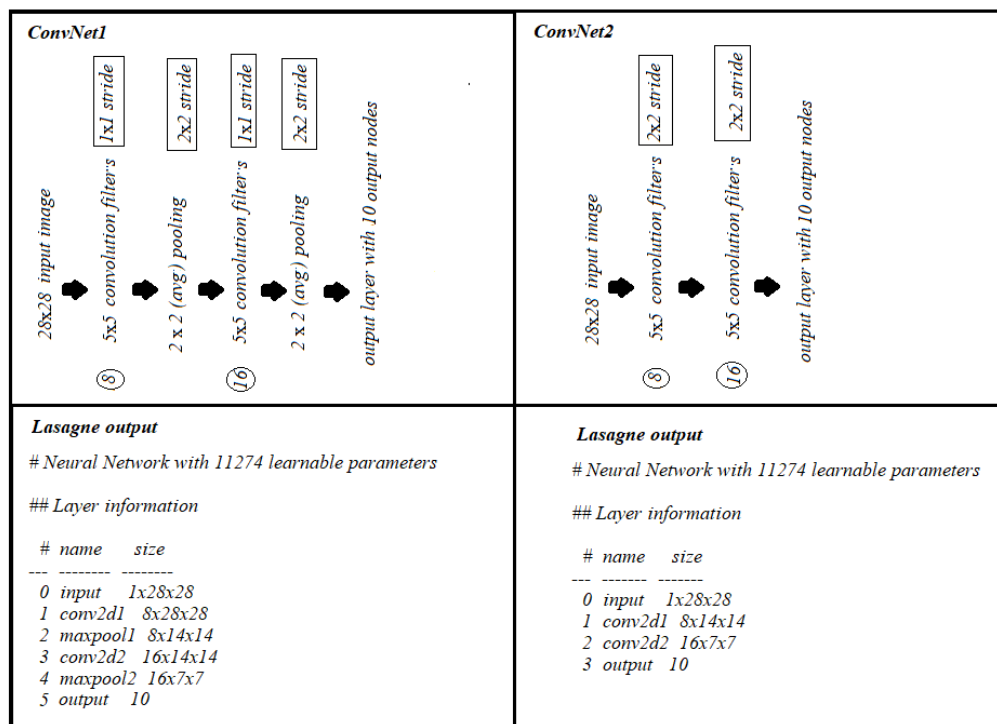
But than the answer : 11274 (since avg pooling does not add any parameters) should be correct, which you (staff?) claim is not.

Could you please check?

+1 @witoldw, i am also wondering why 11274 is not the correct answer.

Please refer to <https://stackoverflow.com/questions/42786717/how-to-calculate-learnable-parameters-for-neural-network/45621048#45621048> and the following figure (obtained by running the conv-net models with Lasagne/theano, conv layers with 'same' padding), which clearly computes the number of parameters as 11274.

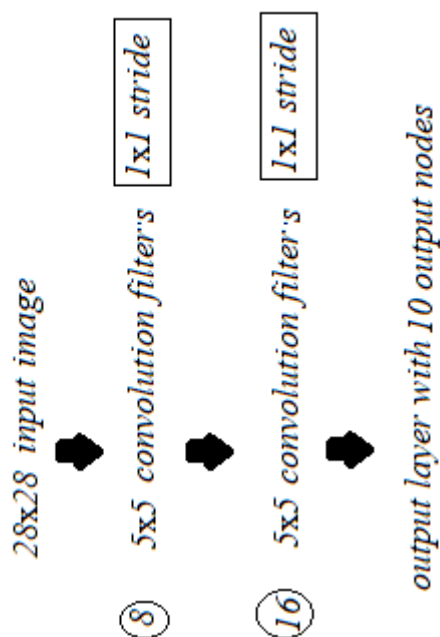
As can be seen, (1) convolution layer with 2x2 stride and (2) convolution layer 1x1 stride + (max/avg) pooling with 2x2 stride, each contributes same numbers of parameters with 'same' padding.



Moreover, the number of parameters for the conv-net shown below with 1x1 stride without pooling is 128874 and **not** 11274 (as stated in the question), as computed / shown in the next figure, along with lasagne/theano output (since with 1x1 stride and with 'same')

padding, the size of the input image does not get reduced to 7x7x16, it remains 28x28x16 at the end of the convolution layers).

ConvNet



Lasagne output

Neural Network with 128874 learnable parameters

Layer information

#	name	size
0	input	1x28x28
1	conv2d1	8x28x28
2	conv2d2	16x28x28
3	output	10

<p>Number of params</p> $= (5 \times 5 + 1) \times 8 +$ $(5 \times 5 \times 8 + 1) \times 16 +$ $(28 \times 28 \times 16 + 1) \times 10$ $= 128874$

@STAFF could you please look into this?

posted 2 days ago by [sandipan_dey](#)

Add a comment

[AbrianMihalko](#)

about 8 hours ago

Q5: Why is 11,274 the wrong answer?

Is this not the math?:

$(200 + 8) + (3200 + 16) + (16 \times 7 \times 7 \times 10 + 10) = 11,274$

Also, `C.logging.log_number_of_parameters(z)` outputs: Training 11274 parameters in 6 parameter tensors.

Add a comment

Support

Showing all responses

Add a response:

Preview

Submit



English ▼

© 2012–2017 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open edX logos are registered trademarks or trademarks of edX Inc. | 粤ICP备17044299号-2

POWERED BY
OPENedX®

