How to compute number of weights of CNN?

Asked 1 year, 6 months ago Active 1 year, 6 months ago Viewed 2k times

1

How can we compute number of weights considering a convolutional neural network that is used to classify images into two classes :

- INPUT: 100x100 gray-scale images.
- LAYER 1: Convolutional layer with 60 7x7 convolutional filters (stride=1, valid padding).
- LAYER 2: Convolutional layer with 100 5x5 convolutional filters (stride=1, valid padding).
- LAYER 3: A max pooling layer that down-samples Layer 2 by a factor of 4 (e.g., from 500x500 to 250x250)
- LAYER 4: Dense layer with 250 units
- LAYER 5: Dense layer with 200 units
- LAYER 6: Single output unit

Assume the existence of biases in each layer. Moreover, pooling layer has a weight (similar to AlexNet)

How many weights does this network have?

Some Keras code

```
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Conv2D, MaxPooling2D

model = Sequential()

# Layer 1
model.add(Conv2D(60, (7, 7), input_shape = (100, 100, 1), padding="same",
activation="relu"))

# Layer 2
model.add(Conv2D(100, (5, 5), padding="same", activation="relu"))

# Layer 3
model.add(MaxPooling2D(pool_size=(2, 2)))

# Layer 4
model.add(Dense(250))

# Layer 5
model.add(Dense(200))

model.summary()
```

Share Improve this question Follow

edited Jun 20 '20 at 9:12
Community ♦
1 1

asked Dec 15 '19 at 12:19
estamos
59 10

If you create that model in Keras, it will calculate it for you. – Chris Tosh Dec 15 '19 at 12:48

2 Create the model and call `model.summary()` – Daniel Möller Dec 15 '19 at 13:49

1 Answer

Active Oldest Votes

4

TL;DR - For TensorFlow + Keras

Use `Sequential.summary` - Link to documentation.

Example usage:

```
from tensorflow.keras.models import *

model = Sequential([
    # Your architecture here
]);

model.summary()
```

The output for your architecture is:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 94, 94, 60)        3000
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 90, 90, 100)       150100
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 45, 45, 100)       0
_________________________________________________________________
flatten (Flatten)            (None, 202500)            0
_________________________________________________________________
dense (Dense)                (None, 250)               50625250
_________________________________________________________________
dense_1 (Dense)              (None, 200)               50200
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 201
=================================================================
Total params: 50,828,751
Trainable params: 50,828,751
Non-trainable params: 0
_________________________________________________________________
```

That's 50,828,751 parameters.

Number of weights in a 2D Convolutional layer

For a 2D Convolutional layer having

- `num_filters` filters,
- a filter size of `filter_size * filter_size * num_channels` ,
- and a bias parameter per filter

The number of weights is: `(num_filters * filter_size * filter_size * num_channels) + num_filters`

E.g.: LAYER 1 in your neural network has

- 60 filters
- and a filter size of 7 * 7 * 1. (Notice that the number of channels (1) comes from the input image.)

The number of weights in it is: `(60 * 7 * 7 * 1) + 60` , which is `3000` .

Number of weights in a Dense layer

For a Dense layer having

- `num_units` neurons,
- `num_inputs` neurons in the layer prior to it,
- and a bias parameter per neuron

The number of weights is: `(num_units * num_inputs) + num_units`

E.g. LAYER 5 in your neural network has

- 200 neurons
- and the layer prior to it - LAYER 4 - has 250 neurons.

The number of weights in it is `200 * 250` , which is `50200` .

Share Improve this answer Follow

answered Dec 15 '19 at 13:54
tVishal96
58 6

Flatten and Dense layers I don't think that are needed at my model . Should I add a weight at MaxPooling layer similar to AlexNet ? – estamos Dec 15 '19 at 15:13

As far as I know, MaxPooling layers don't have weights associated with them. I may be wrong, though. – tVishal96 Dec 17 '19 at 16:55

Yes , Maxpooling layer does not have a weight . I wrote some fixed code at top post . – estamos Dec runtime (while training). A Dense layer expects a 2D tensor as input, but the output of a Conv2D layer is a 4D tensor. Flatten will convert the 4D tensor to a 2D tensor as needed. Please see this question for more details. Also, you'll need a Dense layer with 1 neuron at the end of your network because that's the output layer. – tVishal96 Dec 18 '19 at 3:29