**Understanding Parameter Sharing (or weights replication) Within Convolutional Neural Networks**

# <https://towardsdatascience.com/understanding-parameter-sharing-or-weights-replication-within-convolutional-neural-networks-cc26db7b645a>

## **Parameter sharing or weights replication is a topic area that can be overlooked within Deep learning studies. Understanding this simple concept aids a broader grasp of the internals of the convolutional neural network.**

Affine invariance karakteristika CNN-a

Convolutional Neural Networks (CNN) have characteristics that enable invariance to the affine transformations of images that are fed through the network. This provides the ability to recognize patterns that are shifted, tilted or slightly warped within images.

These characteristics of affine invariance are introduced due to three main properties of the CNN architecture.

1. [Local Receptive Fields](https://towardsdatascience.com/understand-local-receptive-fields-in-convolutional-neural-networks-f26d700be16c)
2. **Shared Weights (parameter sharing)**
3. Spatial Sub-sampling

In this article, we’ll be exploring **Shared Weights** and understanding their purpose and the advantages they serve within CNN architectures.

## INTRODUCTION

Let’s begin by creating a mental illustration of a single convolutional layer within a CNN.

A convolutional layer (*conv layer*) within a CNN contains a set of units, which can also be referred to as neurons.

The conv layer also includes several filters within the layer, and this is a predefined hyperparameter.

The number of filters within a layer indicates the depth dimension of the output volume of the activation/feature maps that are created by the conv layer as input to the next layer.

Each of these filters has a set width and height, which corresponds to the local receptive field of a single unit within the layer. The filters acting upon the input data creates the output of a convolutional layer, the feature map.

Feature map -> rezultat primjene filtera (output) na ulaznim podacima (događa se u konvolucijskom sloju)

Govorimo o težinama unutar pojedinog filtera

The weight values within filters are learnable during the training phase of a CNN.

Dimenzija feature map, (izlaz iz konvolucijskog sloja) ima komponentu dubine

The output dimension of the convolutional layer has a depth component, if we partition each segment of the output we will obtain a 2D plane of a feature map. The filter used on a single 2D plane contains a weight that is shared across all filters used across the same plane.

Zašto koristit iste težine

The advantage of this is that we maintain the same feature detector used in one part of the input data across other sections of the input data.

Opis feature mape

The output of a convolutional layer is a set of feature maps, where each feature map is the result of a convolution operation between the fixed weight parameters within the unit and the input data.

One of the essential characteristics of the convolutional neural network layer is its ability for the feature map to reflect any affine transformations that are made to the input image that is fed through the input layer.

So any shift, tilt or orientation made to the input data, the feature map will provide an output that is shifted, tilted or orientated by the amount the input data was subjected to.

## PUTTING THEORY INTO PRACTICE

The goal of this section is to bring to light the benefits of weight sharing that occurs within convolutional neural networks.

We are going to derive the number of trainable weights without weight sharing and also with weight sharing, within the first convolutional layer of two popular CNN architectures: [LeNet](http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf)and [AlexNet](https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf).

### STEPS

1. **Obtain the output width of the conv layer**

((Input size width — filter size + (2\*Padding)) / stride )+ 1 = output width of convolutional layer)

2. **Calculate the number of neurons/units within the conv layer**

3.**Calculate the number of training parameters (include bias) without weight sharing**

4. **Calculate the number of training parameters (include bias) with weight sharing**

*The table below depicts the information from the AlexNet and LeNet CNN architectures will be used to derive the number of training parameters/weights within a convolutional layer.*

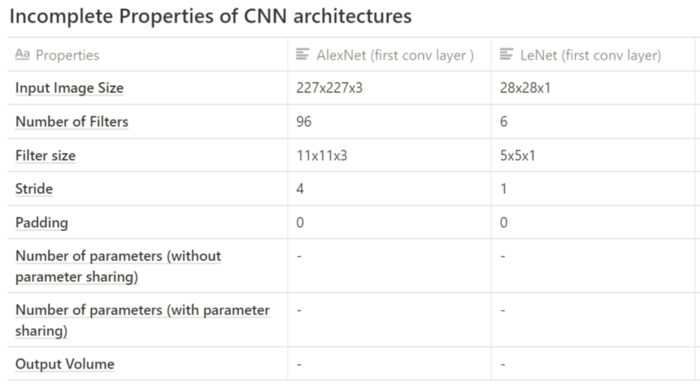


Table showing incomplete properties of CNN architectures

**AlexNet**

1. **Output width of conv layer:**

= ((227–11) / 4) +1

=**55**(conv layer output width)

2. **Number of neurons/units within the conv layer**

= output height \* output width \* number of feature maps

= 55x55x96 (conv output volume)

=**290,400 units**

3. **Number of training parameters or weights within the conv layer (without weight sharing)**

= 290400 \* ((11 \* 11 \* 3) + 1 bias)

=

4. **Number of training parameters or weights with weight sharing (with weight sharing)**

= 96\*((11\*11\*3) + 1 bias)

## **= 34,944 weights**

**LeNet**

1. **Output width of conv layer:**

= ((28–5) / 1) +1

=**24**(conv layer output width)

2. **Number of neurons/units within the conv layer**

= output height \* output width \* number of feature maps

= 24x24x6 (conv output volume)

=**3,456 units**

3. **Number of training parameters or weights within the conv layer (without weight sharing)**

= 3456\* ((5 \* 5 \* 1) + 1 bias)

=

4. **Number of training parameters or weights with weight sharing (with weight sharing)**

= 6\*((5\*5\*1) + 1 bias)

## **= 156 weights**

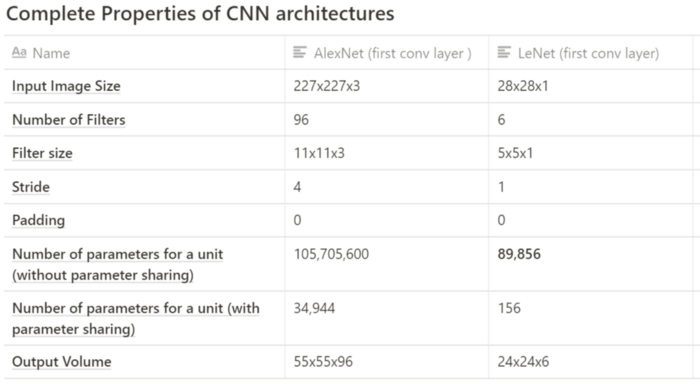


Table showing complete properties of CNN architectures

## CONCLUSION

It’s evident that through parameter sharing, we can reduce the number of weights within a conv layer.

Parameter sharing is used in all conv layer within the network.

Parameter sharing reduces the training time; this is a direct advantage of the reduction of the number of weight updates that have to take place during backpropagation.

To reiterate parameter sharing occurs when a feature map is generated from the result of the convolution between a filter and input data from a unit within a plane in the conv layer. All units within this layer plane share the same weights; hence it is called weight/parameter sharing.