

# Convolutional Neural Network (LeNet)

Chang Liu

## Introduction

This document mainly gives a summary for my understanding of the CNN, especially for the **LeNet-5** structure understanding.

Many of the ideas in this document is not specific to the LeNet, but for all Convolutional Neural Network, as it's a general concept.

## Sparse Connectivity

The input of the hidden units in layer  $m$  are from a subset of units in layer  $m-1$ , as this figure shows:

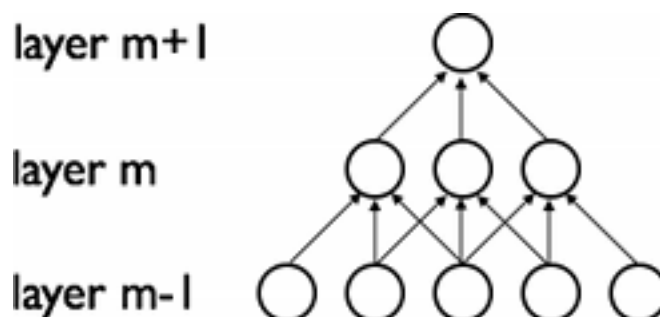


Figure 1: CNN figure

Imagine that layer **m-1** is the input layer, and unit layer **m** has the *receptive field* of width 3 in the input, thus it is connected to 3 adjacent neurons in the retina layer(=layer  $m-1$ )

Units in layer **m+1** have a similar connectivity with the layer below. We say that their receptive field with respect to the layer below is also 3, but their receptive field with respect to the input is larger (5).

**COMMENT:** because their input is just the 5 neurons in the layer **m-1**, taking the input layer into consideration, it's just larger as it's 5 instead of 3.

## Shared Weights

In CNNs, each filter  $h_i$  is replicated across the entire visual field. These replicated units share the same parameterization (weight vector and bias) and form a feature map.

As the figure shows:

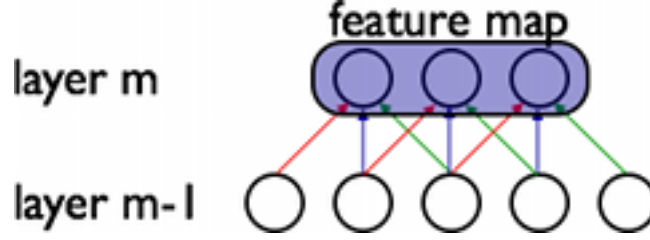


Figure 2: shared\_weights

The 3 hidden units belongs to the same feature map, weights of the same colors are shared, constrained to be identical. Gradient descent can still be used to learn such shared parameters, with only a small change to the original algorithm. The gradient of a shared weight is simply the sum of the gradients of the parameters being shared.

## Details and Notation

A feature map is obtained by repeated application of a function across sub-regions of the entire image, in other words, by *convolution* of the input image with a linear filter, adding a bias term and then applying a non-linear function

as this equation shows:

$$h_{ij}^k = \tanh((W_k * x)_{ij} + b_k)$$

If we denote the k-th feature map at a given layer as  $h^k$ , whose filters are determined by the weights  $W^k$  and bias  $b_k$ , then the feature  $h_k$  can be obtained using the equation listed above.

## More detailed illustration

To form richer representation of the data, each hidden layer is composed of multiple feature maps,  $h^{(k)}$ ,  $k = 0 \dots K$ . The weights  $W$  of a hidden layer can be represented in a 4D tensor containing elements for every combination of destination feature map, source feature map, source vertical position, and source horizontal position. The biases  $b$  can be represented as a vector containing one element for every destination feature map

As the following figure shows:

The figure shows two layers of CNN. layer **m-1** has four feature maps, and layer **m** has 2 feature maps( $h_0$  and  $h_1$ ).

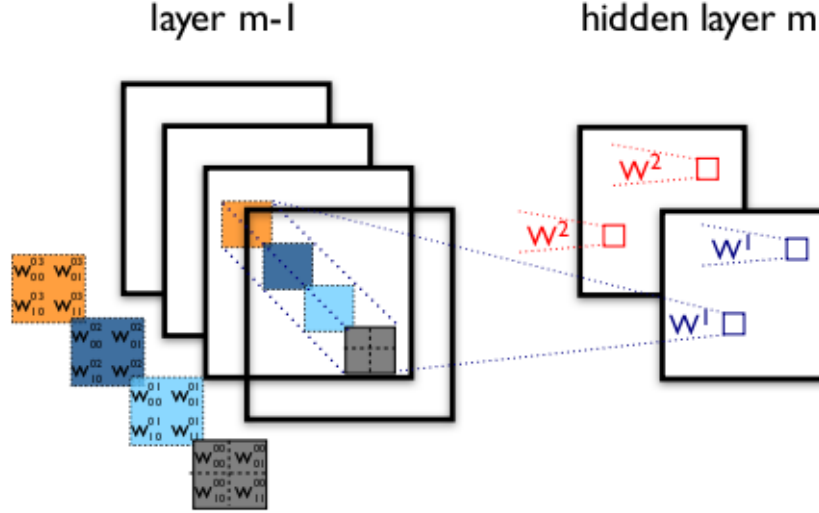


Figure 3: CNN details

Pixels(neuron outputs) in  $h_0$  and  $h_1$  are computed from pixels of layer (m-1) which fall within their 2x2 receptive field in the layer below (shown as colored rectangles)

$W_{ij}^{kl}$  denotes the weight connecting each pixel of the  $k$ -th feature map at layer **m**, with the pixel at coordinates  $(i, j)$  of the  $l$ -th feature map of layer (**m-1**)