Computer Vision 1, Master AI
Tutorial Lecture 5: ConvNets
With Answers

Amir Ghodrati and Thomas Mensink and Theo Gevers

1 Conv Layer Arithmetics

1.a How many parameters are needed for a fully connected layer, with the following configurations: Input Width (W) = 100 nodes, the Input Height (O) = 100 nodes, and Hidden Layer (H) = 1000 nodes? Note: Here an image of the mentioned dimensions are flattened out.

Answer: $W \times O \times H = 10M$

1.b How many parameters are needed for a locally constrained layer, where each neuron looks at a 10x10 window, when using W=H=100, and stride of 5?

Answer: Hidden = $20 \times 20 = 400$, so $10 \times 10 \times 400 = 40K$ (Assuming a 0 pad configuration)

1.c How many parameters are needed when a convolutional layer is used, using a filter of size 10x10, and using 100 different filters?

Answer: $10 \times 10 \times 100 = 10K$

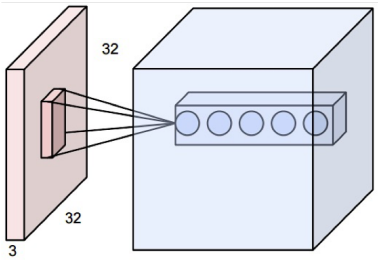


Figure 1: Input volume: $32 \times 32 \times 3$, receptive field: 5×5, number of neurons: 5, along the depth dimension

1.d What is the size of the output volume with stride 1?

Answer: Assuming we don't do padding, the output is $(32 - 5)$/Stride + 1 = 28, so $28 \times 28 \times 5$

1.e How many weights are learned in one filter? How many in this layer?

Answer: There are $5 \times 5 \times 3 = 75$ weights for one kernel. For this layer there are 75×5 = 375 parameters.

1.f What is the depth of a filter in the next layer?

Answer: 5

1.g What is the size of the output volume with stride 3?

Answer: We now have: $(32 - 5)/3 + 1 = 10$, so $10 \times 10 \times 5$

1.h How many weights are learned in that case?

Answer: Doesn't change, see 1.e

2 Sharing of Weights

2.a Describe a scenario where weight sharing -as is done in the Convolutional Layer - is not beneficial for recognition or training.

Answer: For example for face identification, you want to have task specific neurons, like an ”eye”-neuron, and a ”mouth”-neuron. Furthermore, for this task, only the bounding box is considered, so the features are limited.

3 Gradients

3.a What is the difference between the analytical gradient and the numerical gradient?

Answer: Analytical gradients are in the continuous domain while numerical gradient are in the discrete domain. Numerical gradients are approximations using small perturbations, while the analytical gradients are based on analytical derivative functions of a given function.

3.b Consider the following conv notation: $a_{rc} = \boldsymbol{x}_r^\top \boldsymbol{\theta}$, what is the gradient wrt the parameters?

Answer: That is $\boldsymbol{x}_r$, however, for the full gradient, it needs to be summed over all regions, including the loss of each region from the layer above.

3.c Consider the ReLu non-linearity $z = \max(0, a)$, what is the gradient?

Answer: The gradient is

$$\nabla z = \begin{cases} 1, & \text{if } a > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

3.d A ReLU is considered to be ‘dead’ if it never updates. Describe (a) when this happens, and (b) one method to circumvent dead ReLUs

Answer:

a When the input data of the neuron is always negative, it will never receive gradient.

b Use Leaky-ReLu’s or initialise ReLu’s with some positive bias

3.e Describe the max-pooling layer, and the gradient of the layer

Answer: It is a form of quantisation, it selects the maximum value of the region and passes that through to the next layer. The gradient is

$$\nabla_i z = \begin{cases} 1, & \text{if argmax} = i, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In other words, it only passes through the ”pixel” with the maximum of the receptive field of the max-filter.

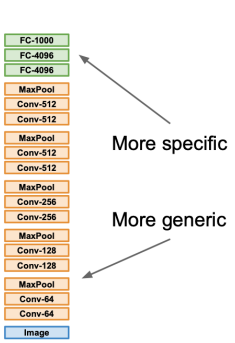
4 Transfer Learning and Invariances

4.a

	very similar new data	very different new data
very little new train data	**A**	**C**
quite a lot new train data	**B**	**D**

Given a pre-trained network, with a new set of data that you want to train a classifier on, please give an advise about learning scheme for a ConvNet scheme for the 4 scenarios (**A-D**) above.

Answer:



	very similar dataset	very different dataset
very little data	Use Linear Classifier on top layer	You're in trouble… Try linear classifier from different stages
quite a lot of data	Finetune a few layers	Finetune a larger number of layers

4.b How could you test the invariance of a convnet towards rotation?

Answer: Do an experiment! Train your network on ImageNet, use a small image/patch and rotate it. Check for the performance as function of degree of rotation.