

COMSM0045: PRACTICAL2

Dima Damen

`Dima.Damen@bristol.ac.uk`

Bristol University, Department of Computer Science
Bristol BS8 1UB, UK

October 10, 2020

CIFAR-10

- ▶ Dataset size: 60,000 images
- ▶ Training split: 50,000 images
- ▶ Test split: 10,000 images (1000 from each class) (**balanced**)

¹<http://groups.csail.mit.edu/vision/TinyImages/>

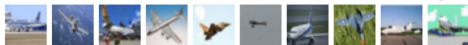
CIFAR-10

- ▶ Dataset size: 60,000 images
- ▶ Training split: 50,000 images
- ▶ Test split: 10,000 images (1000 from each class) (**balanced**)
- ▶ Input size: 32×32 RGB images - $32 \times 32 \times 3 = 3072$ (**tiny images**)
- ▶ These have been collected by Rob Fergus, Antonio Torralba and Bill Freeman from MIT in 2008¹

¹<http://groups.csail.mit.edu/vision/TinyImages/>

CIFAR-10

airplane



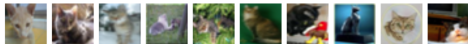
automobile



bird



cat



deer



dog



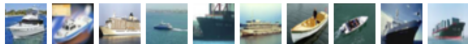
frog



horse



ship



truck



<http://www.cs.toronto.edu/~kriz/cifar.html>

Dima Damen

Dima.Damen@bristol.ac.uk

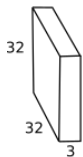
COMSM0045: Lab2 Practical - 2020/2021

CIFAR-10

- ▶ The current state-of-the-art results on CIFAR-10 are available at:
http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#43494641522d3130

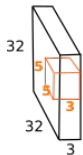
Our First Architecture

- ▶ We start with a $32 \times 32 \times 3$ input x



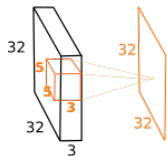
Our First Architecture

- ▶ In the first convolutional layer, one convolution filter is $5 \times 5 \times 3 = 75$ weights



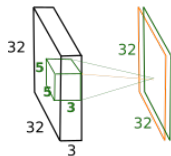
Our First Architecture

- ▶ By convolving it throughout the image, with padding,



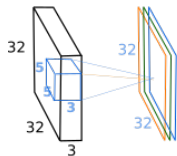
Our First Architecture

- ▶ We can have another filter of the same size, producing a different output layer



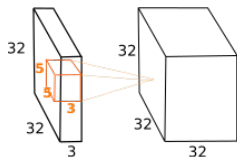
Our First Architecture

- ▶ And another one [until now 75×3 weights to learn]



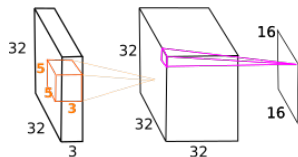
Our First Architecture

- ▶ We propose to have 32 of these = 2400 weights (CONV_1)



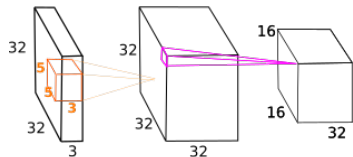
Our First Architecture

- ▶ Following an activation function, we perform max pooling on 2x2 grids



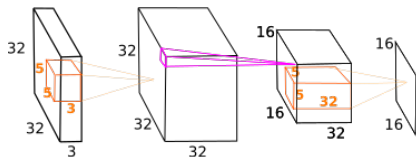
Our First Architecture

- ▶ This is applied for EACH of the 32 output layers



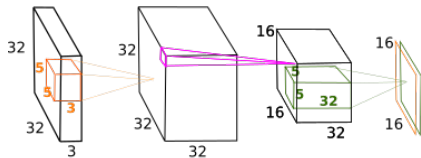
Our First Architecture

- ▶ Second conv layer will have $5 \times 5 \times 32$ convolutional filter = 800 weights



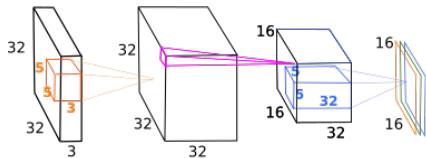
Our First Architecture

- ▶ We can have a second one of these filters



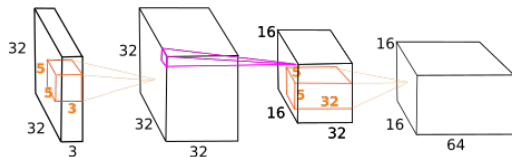
Our First Architecture

► And a third



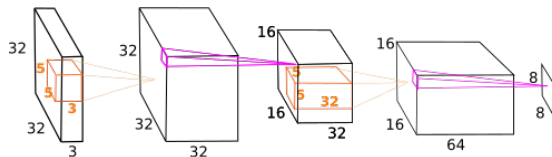
Our First Architecture

- ▶ We will have 64 of these = 51200, along with max-pooling



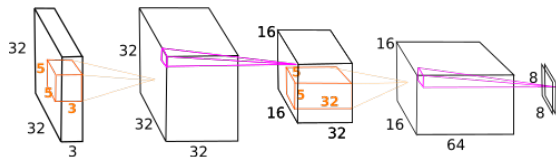
Our First Architecture

- ▶ Followed by max pooling, for each output layer



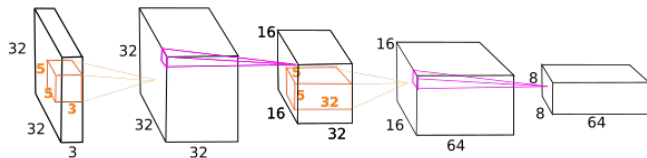
Our First Architecture

- ▶ Doing this for the second filter,



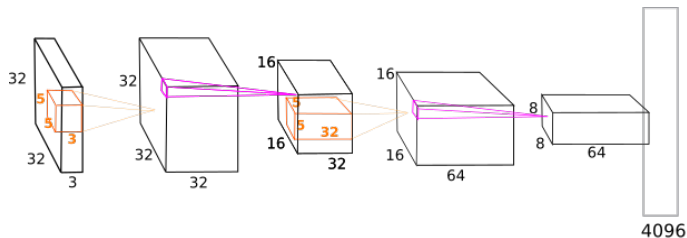
Our First Architecture

- And for all filters,



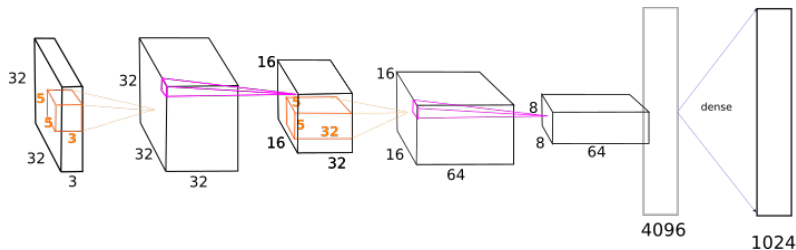
Our First Architecture

- ▶ Our output size is 4096 dimensions, which we reshape into 1D



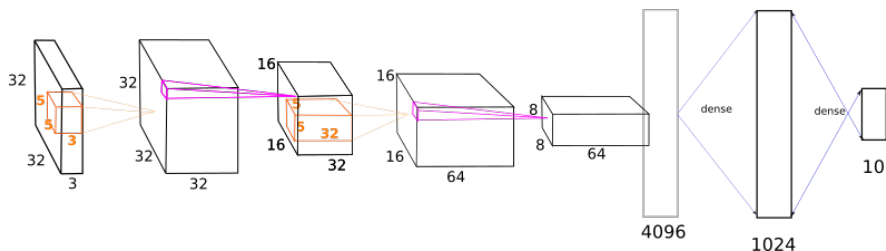
Our First Architecture

- ▶ Followed by 1 fully-connected layer, (4096x1024 weights)



Our First Architecture

- And a final fully connected layer into our 10 classes, (1024x10 weights)



And now....

READY....

STEADY....

GO...