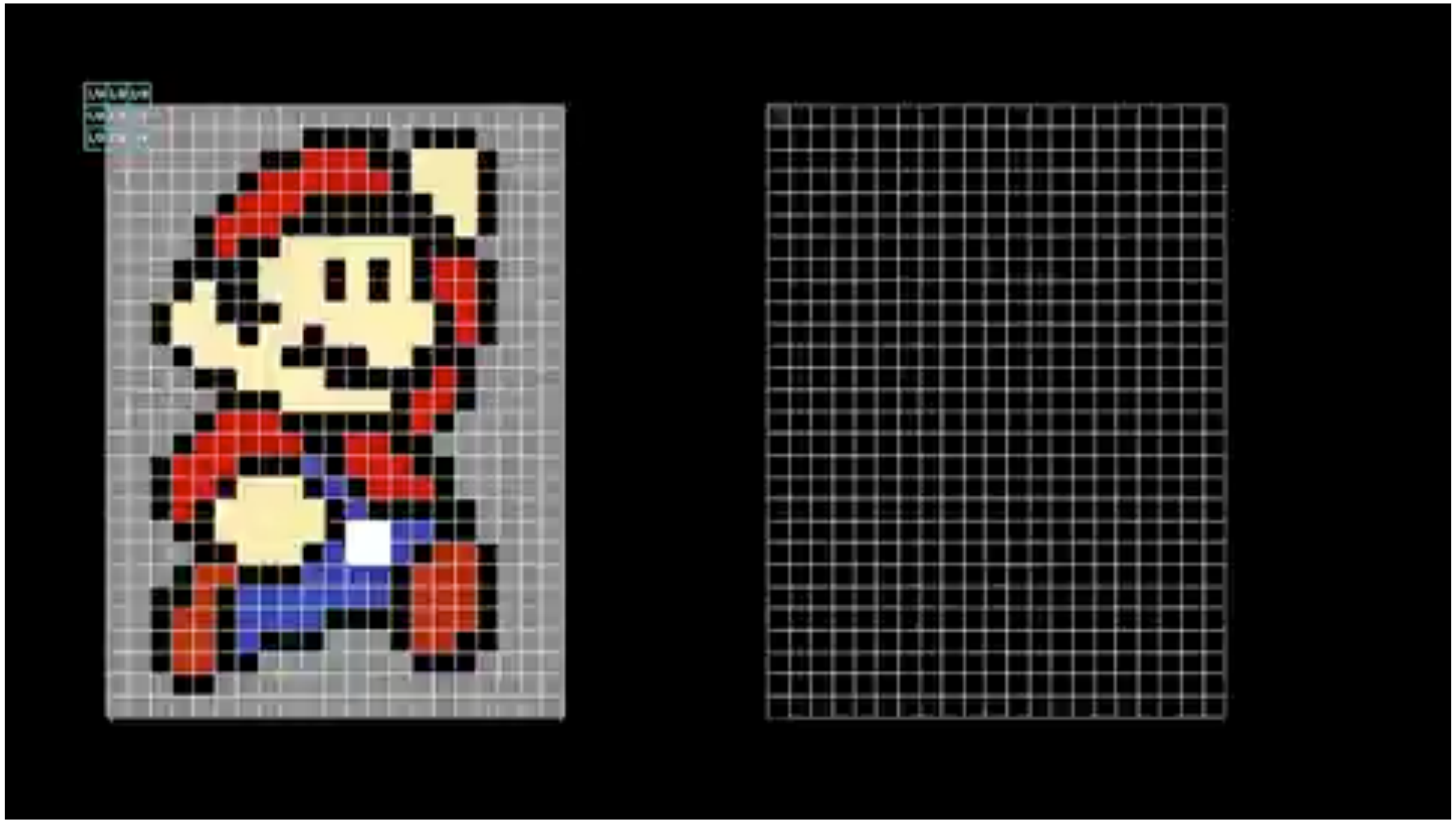
Convolutional neural networks

Romain Tavenard (Université de Rennes)

NB: Most figures in these slides are from
Dumoulin & Visin. A guide to convolution arithmetic for deep learning. 2016

Convolution in practice

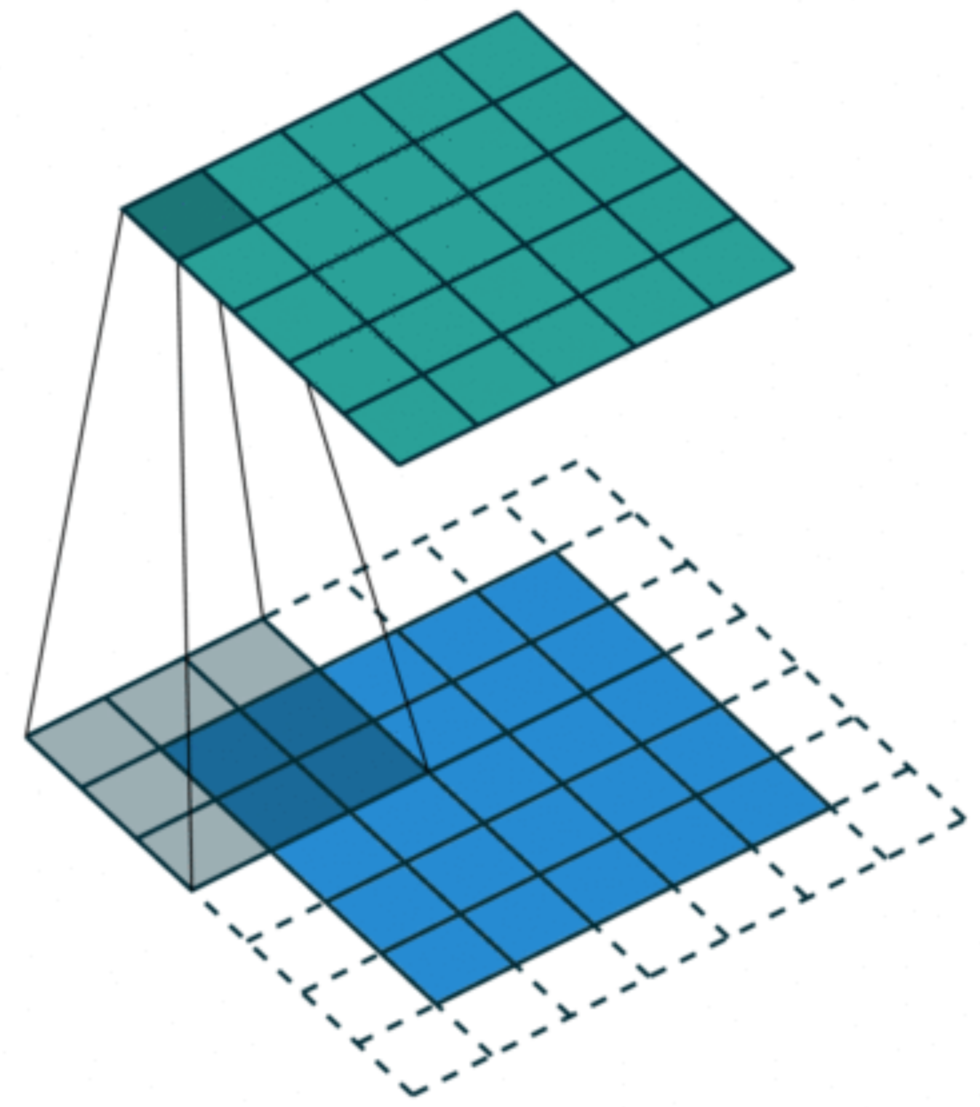


Source : Grant Sanderson, Twitter
https://twitter.com/3blue1brown/status/1303489896519139328?s=20

The convolution operator

- 2D convolution
  - Blue: input image
  - Gray: convolution kernel
  - Cyan: activation map

- Convolution operation = Dot product between
  - convolution kernel (aka filter)
  - subpart of the input

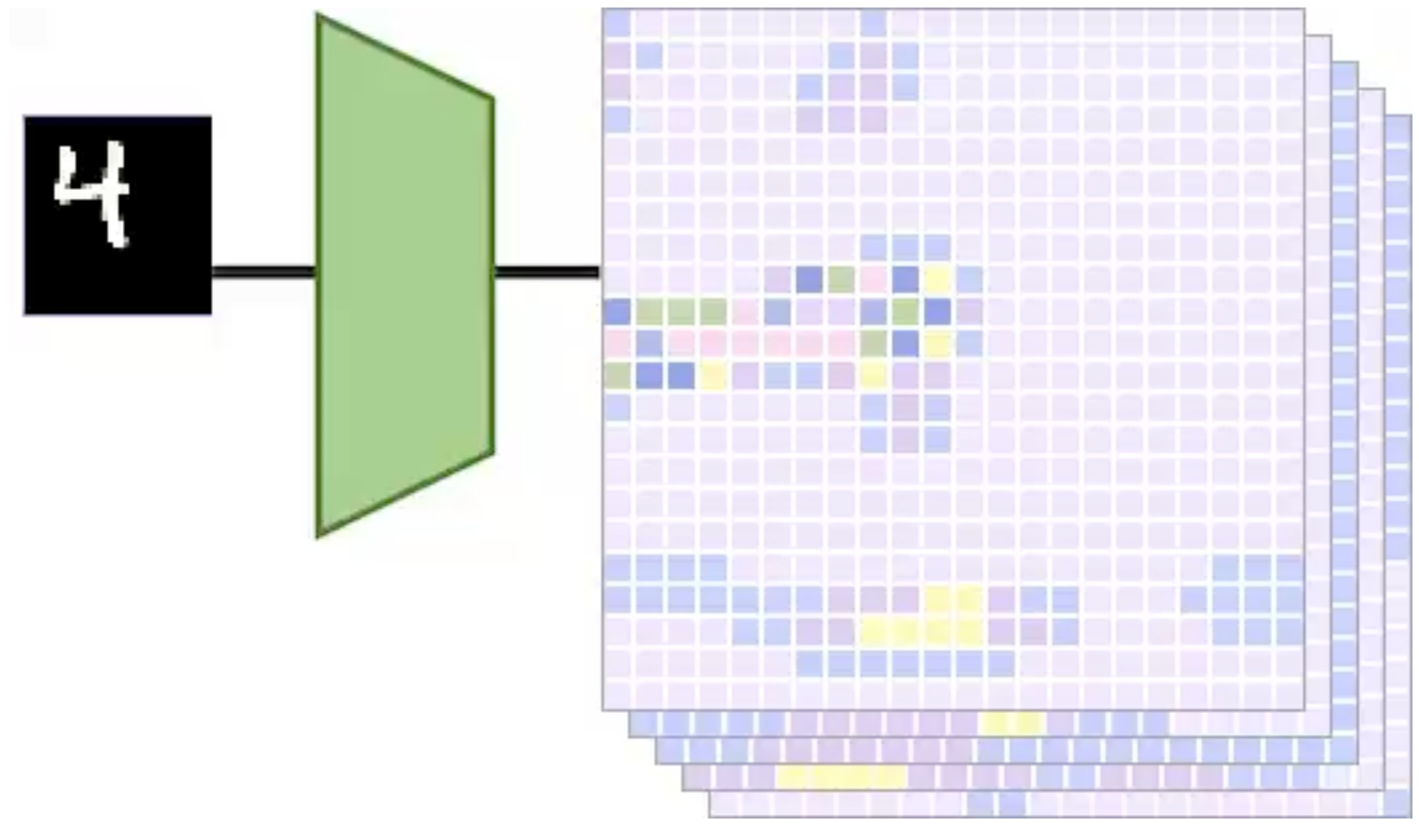


Convolutional layers in NN (1/2)

- A convolution layer is made of
  - convolution kernels
  - biases (1 per kernel)
  - an activation function

- Useful because
  - reduces #parameters
  - encodes translation invariance

Convolution and translation

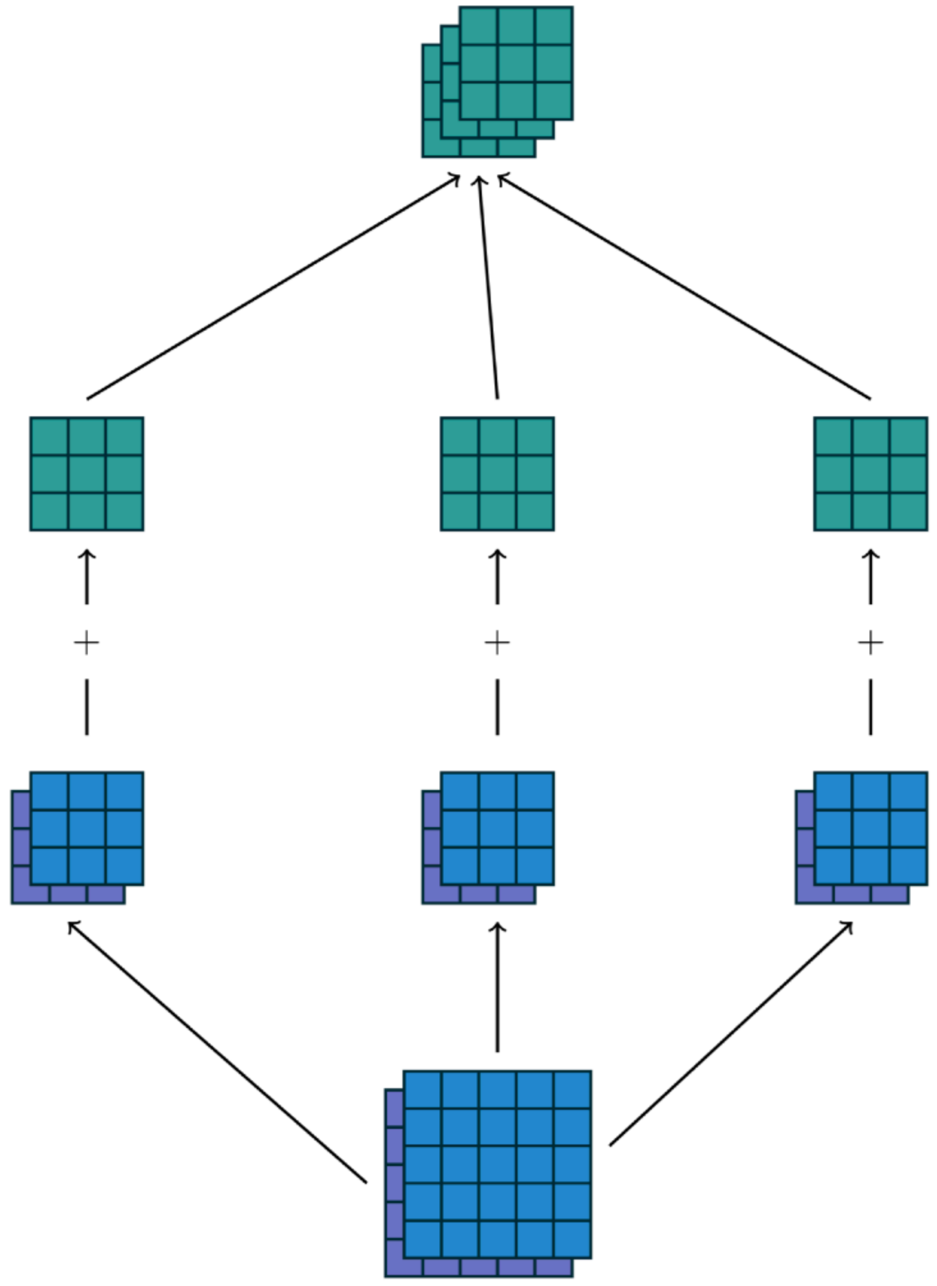


Source : Christian Wolf, Twitter

https://twitter.com/chriswolfvision/status/1313059518574718977?s=20

Convolutional layers in NN (2/2)

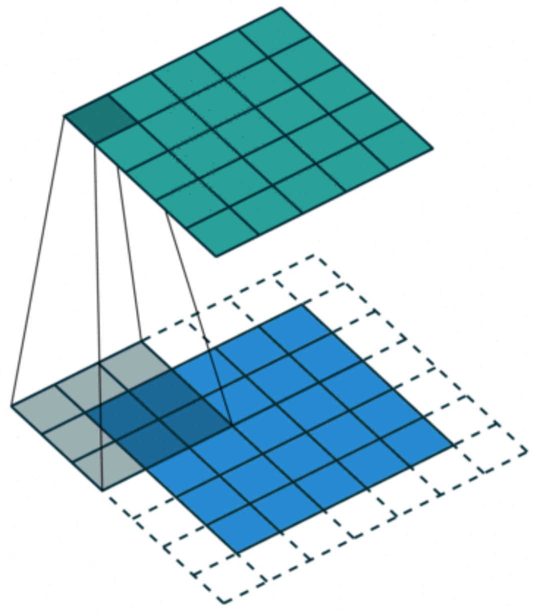
- Multiple input channel case
  - sum the response over all channels
- Multiple kernel case
  - each kernel leads to one output channel

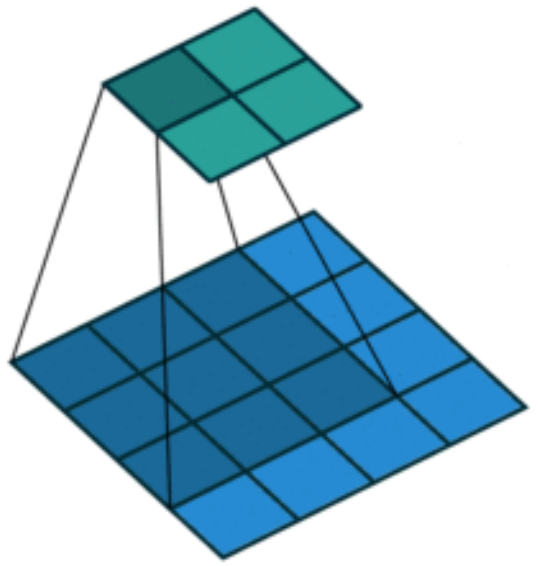


2 input channels, 3 kernels

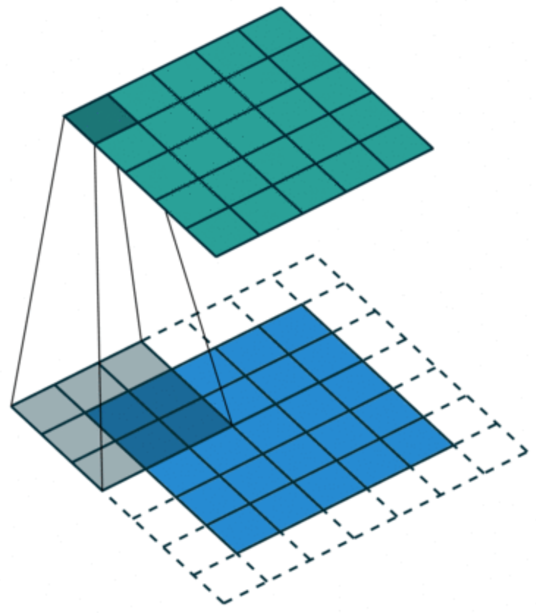
Convolutional layers in NN: hyper parameters

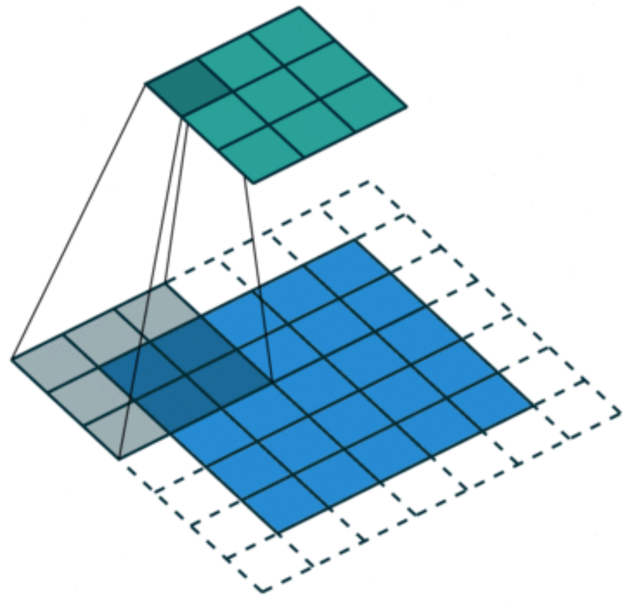
- Padding


padding="same"


padding="valid"

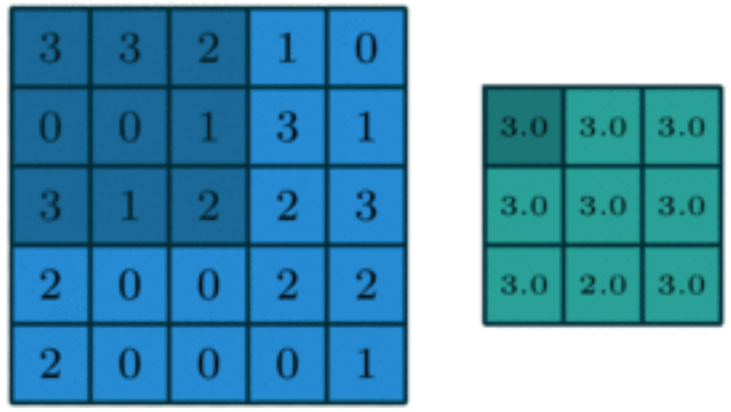
- Strides


strides=1


strides=2

Pooling (aka subsampling) layers in NN

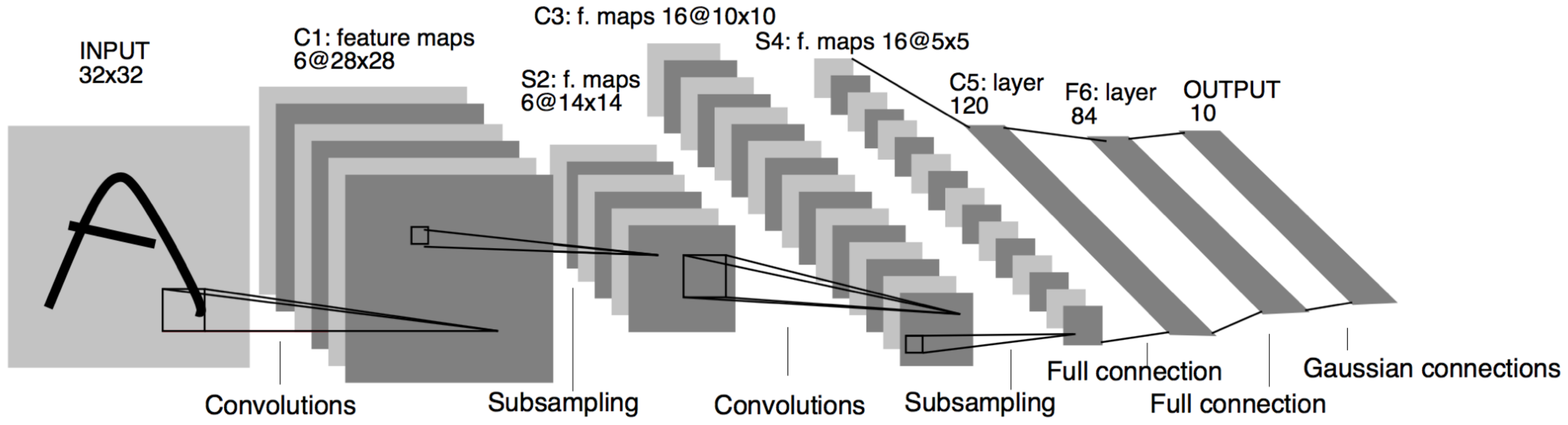
- Max pooling / Average pooling
- Hyper-parameters
  - pool size
  - strides (use None in keras)
  - padding (use "valid" in keras)



pool_size=3, strides=1
(not recommended)

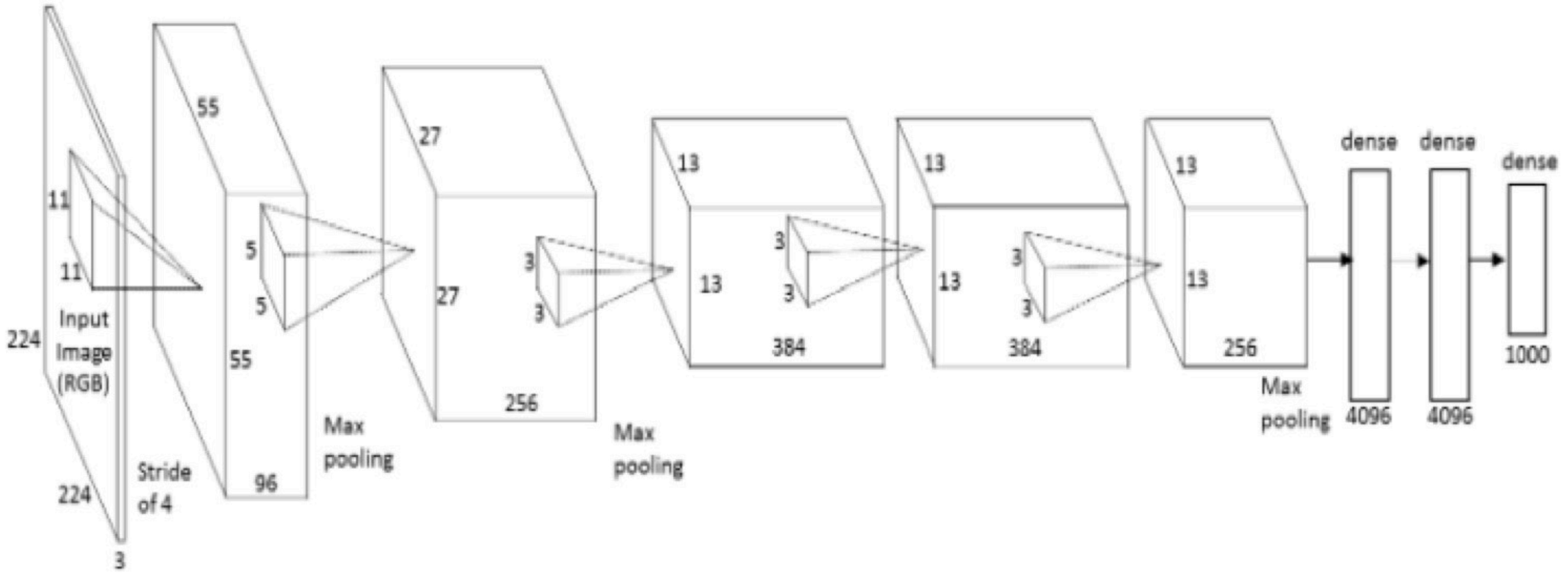
Convolutional model zoo
1. LeNet [LeCun *et al.*, 1989]

- 60k parameters



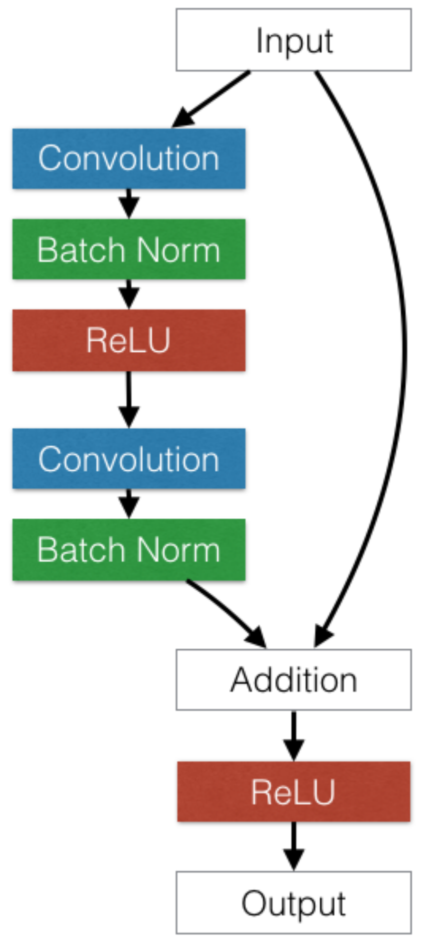
Convolutional model zoo
2. AlexNet [Krizhevsky *et al.*, 2012]

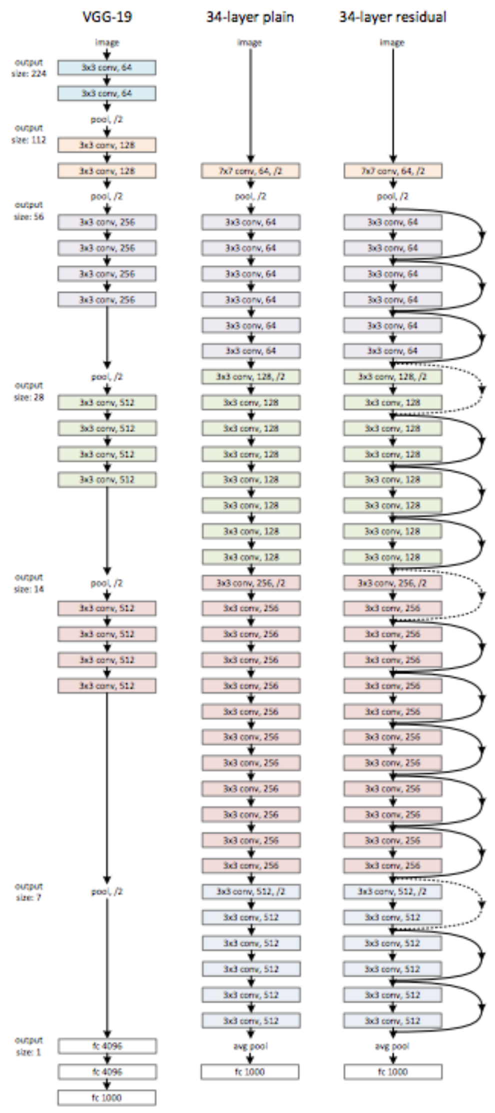
- 60M parameters

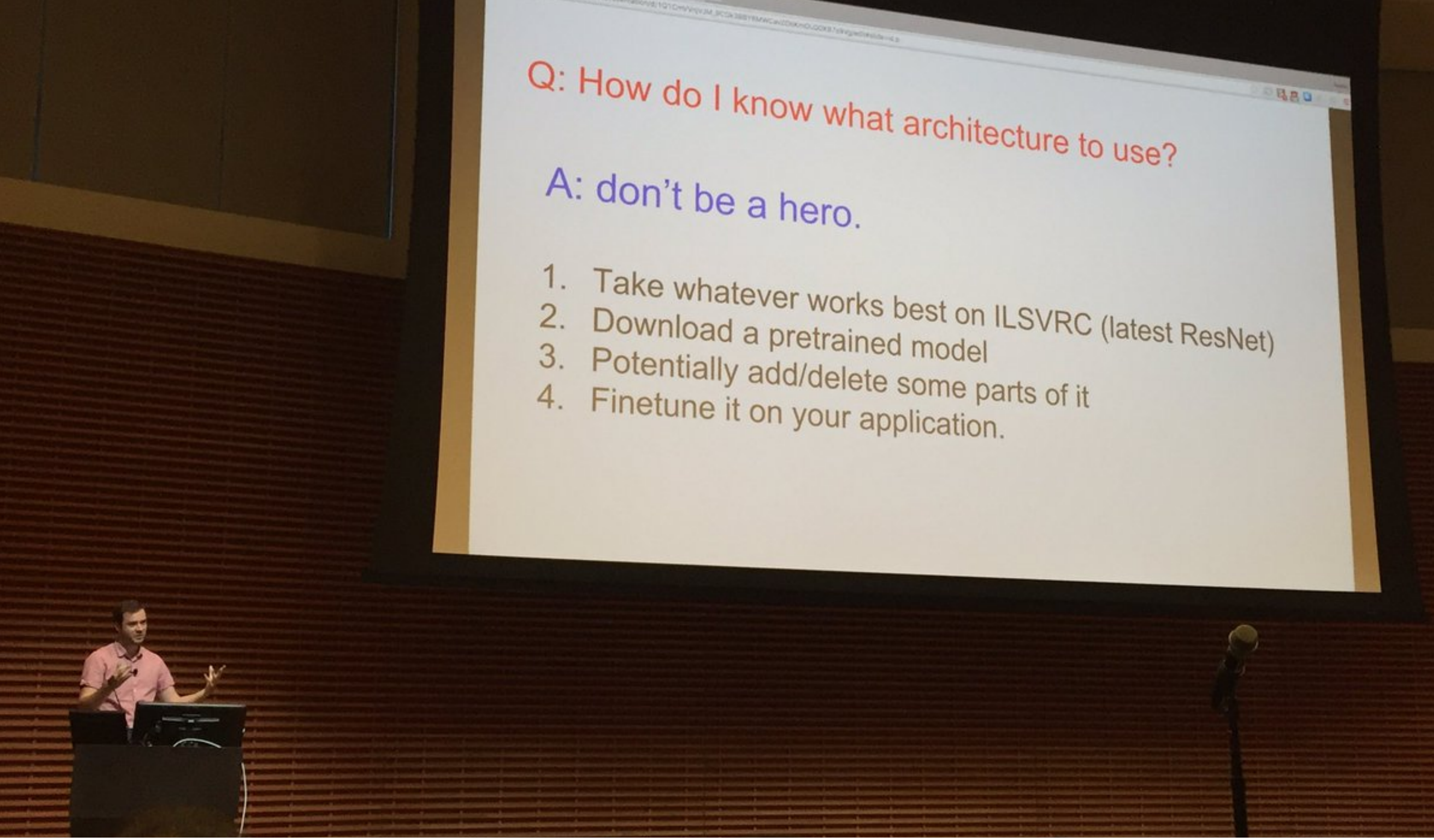


Convolutional model zoo
3. Residual Networks [He *et al.*, 2016]

- Aims at facing the *vanishing gradient* effect
- ResNet-110: ~2M parameters







Q: How do I know what architecture to use?

A: don't be a hero.

1. Take whatever works best on ILSVRC (latest ResNet)
2. Download a pretrained model
3. Potentially add/delete some parts of it
4. Finetune it on your application.

Andrej Karpathy,
Deep Learning Summer School,
2016