



Neural Network Part 3: Convolutional Neural Networks

CS 760@UW-Madison





Goals for the lecture

you should understand the following concepts

- convolutional neural networks (CNN)
- convolution and its advantage
- pooling and its advantage

Convolutional neural networks



- Strong empirical application performance
- Convolutional networks: neural networks that use convolution in place of general matrix multiplication in at least one of their layers

$$h = \sigma(W^T x + b)$$

for a specific kind of weight matrix W



Convolution

Convolution: discrete version



- Given array u_t and w_t , their convolution is a function s_t

$$s_t = \sum_{a=-\infty}^{+\infty} u_a w_{t-a}$$

- Written as

$$s = (u * w) \quad \text{or} \quad s_t = (u * w)_t$$

- When u_t or w_t is not defined, assumed to be 0

Illustration 1

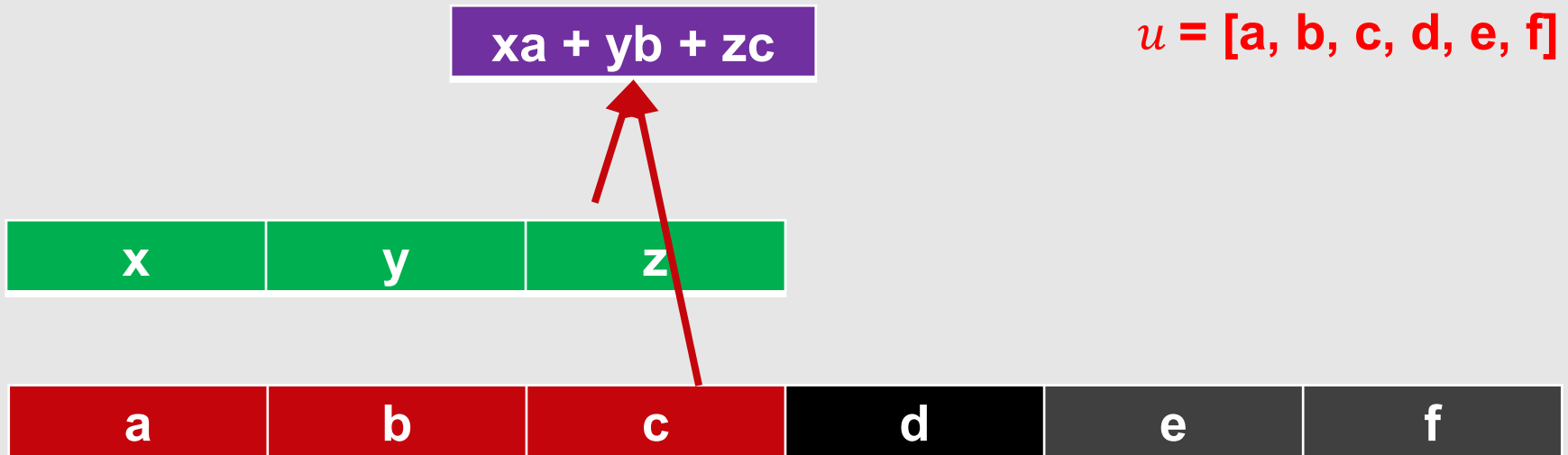


Illustration 1

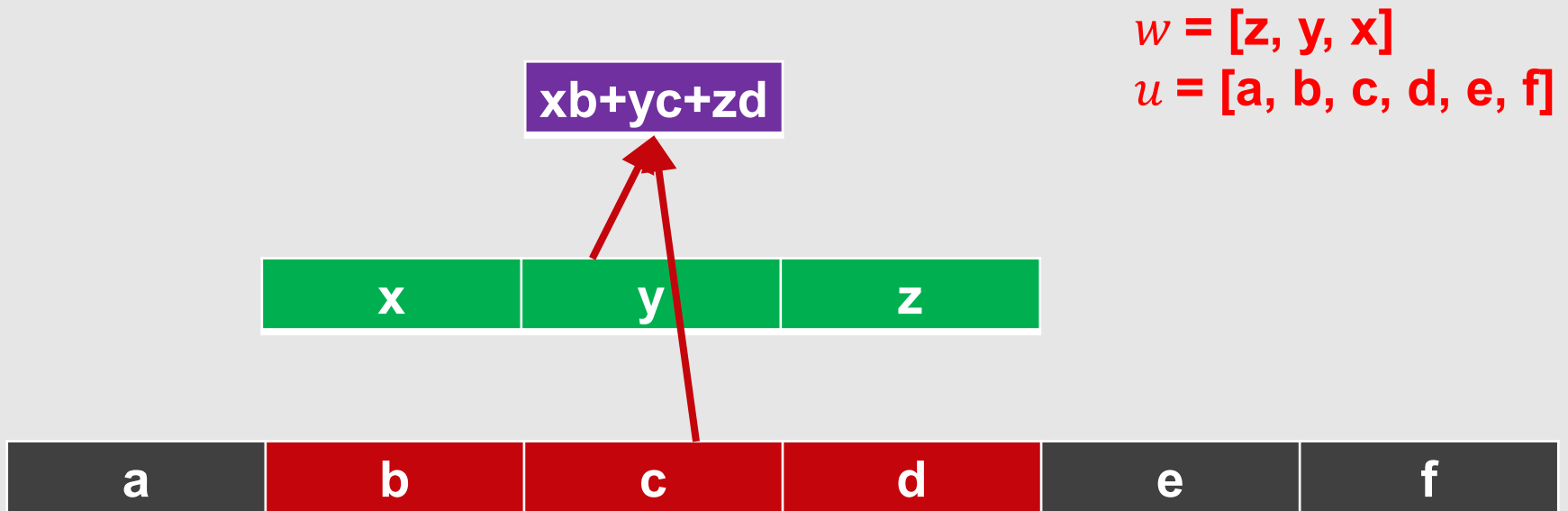


Illustration 1

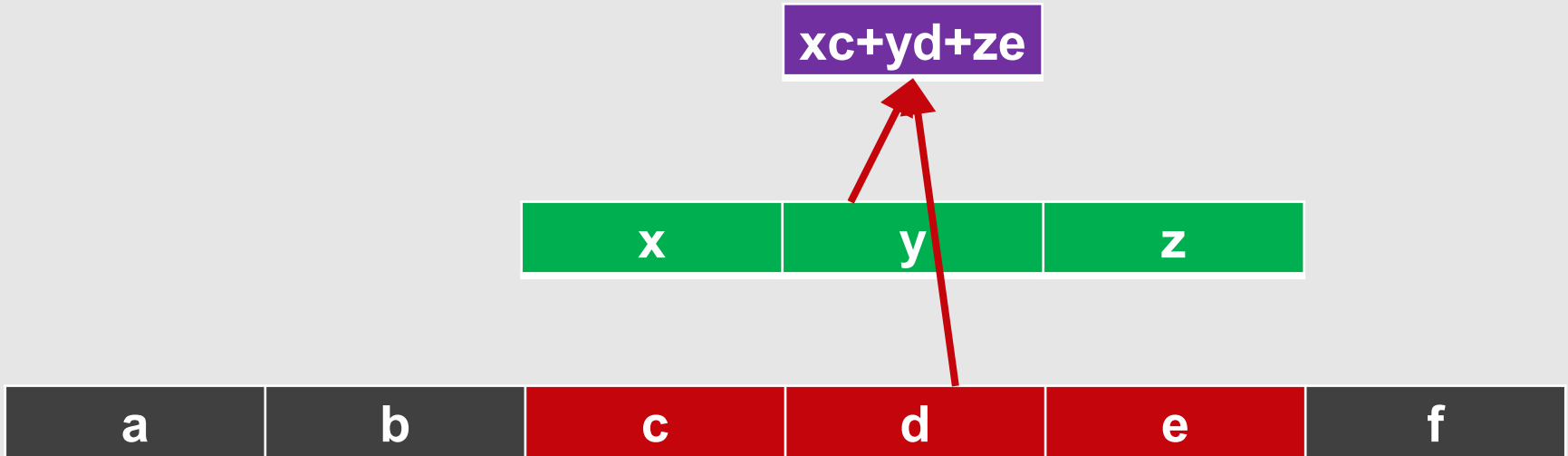


Illustration 1

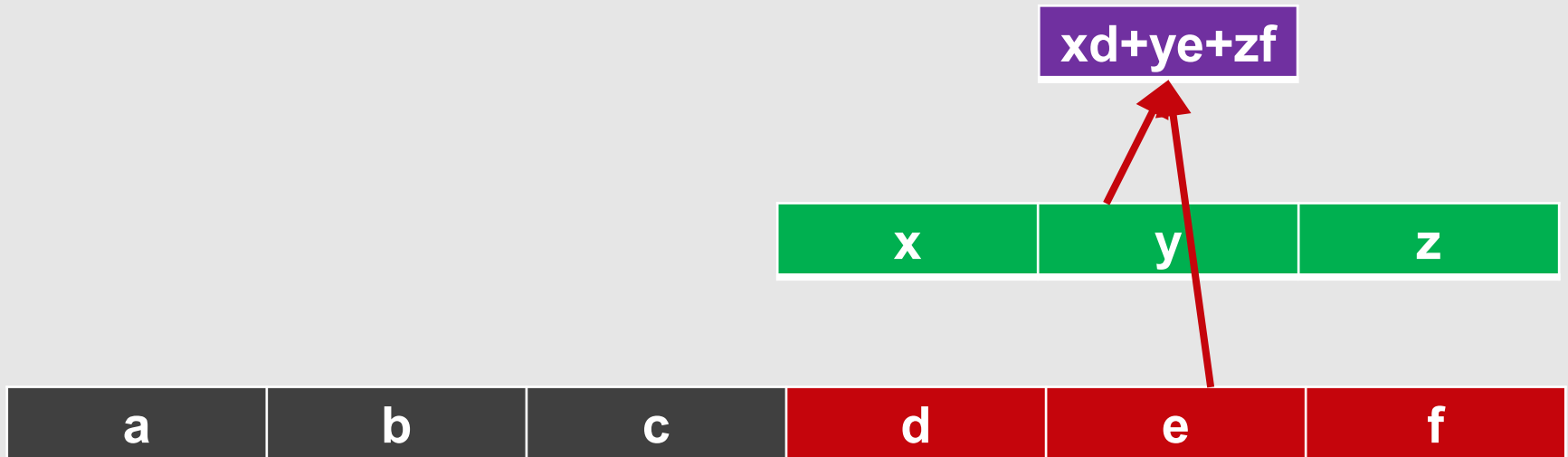


Illustration 1: when to stop



Rightmost output entry

$xd+ye+zf$

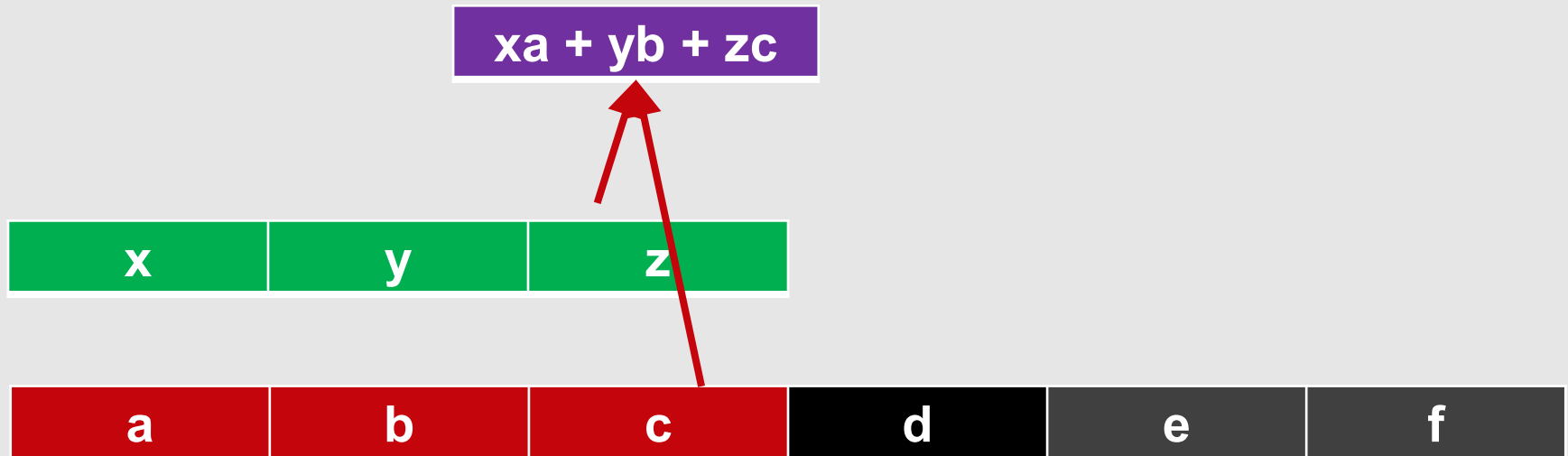


Option 1: VALID, stop when hit the boundary

Illustration 1: when to stop

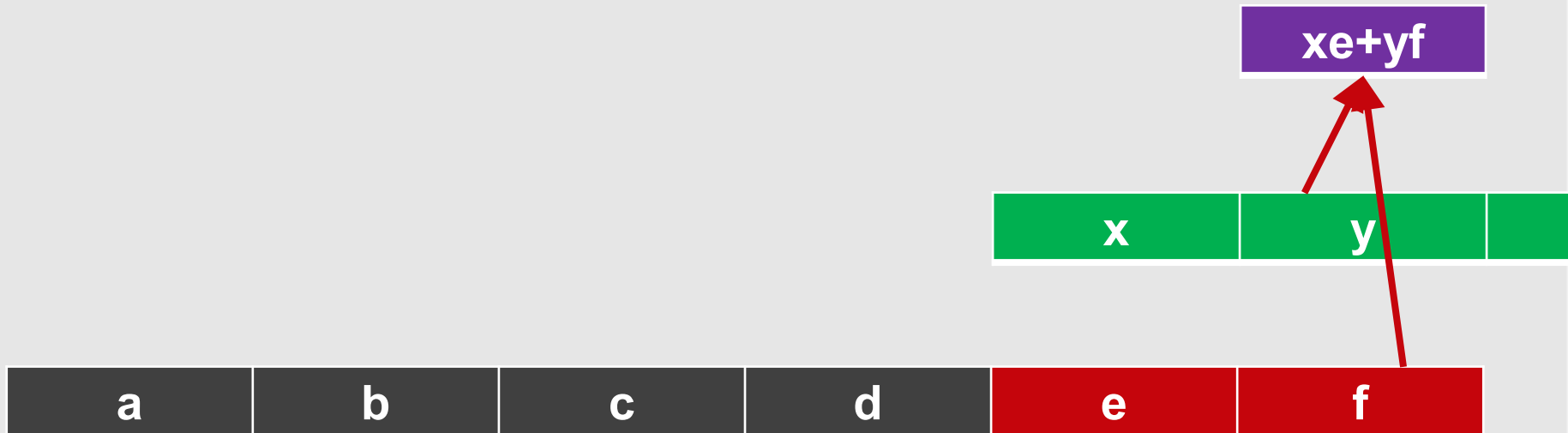


Leftmost output entry



Option 1: VALID, stop when hit the boundary

Illustration 1: when to stop

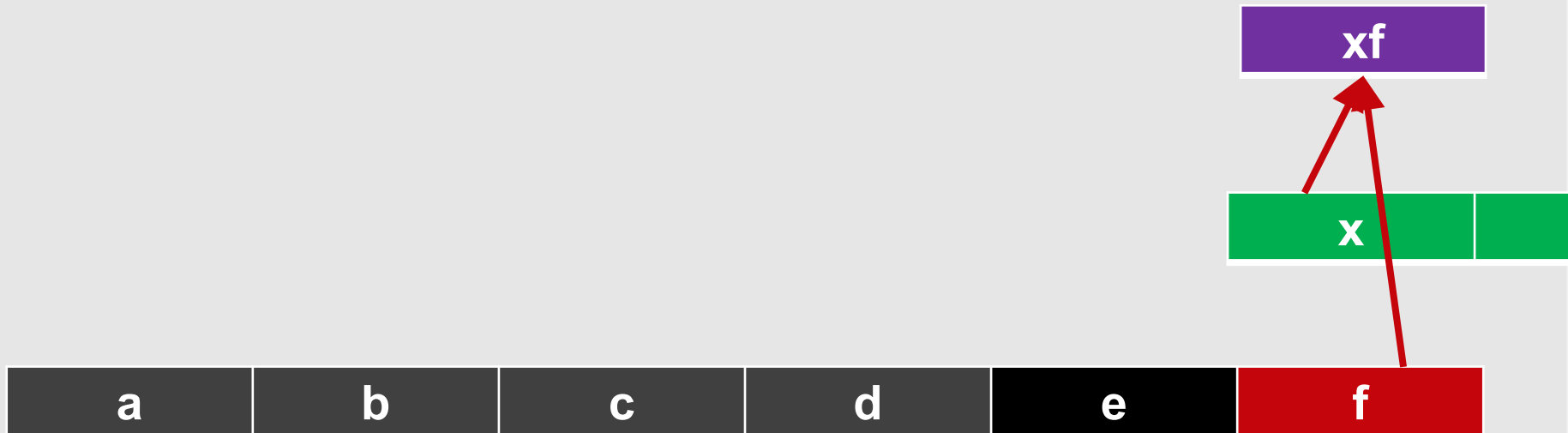


Option 2: SAME, stop when no overlap

Illustration 1: when to stop



Rightmost output entry

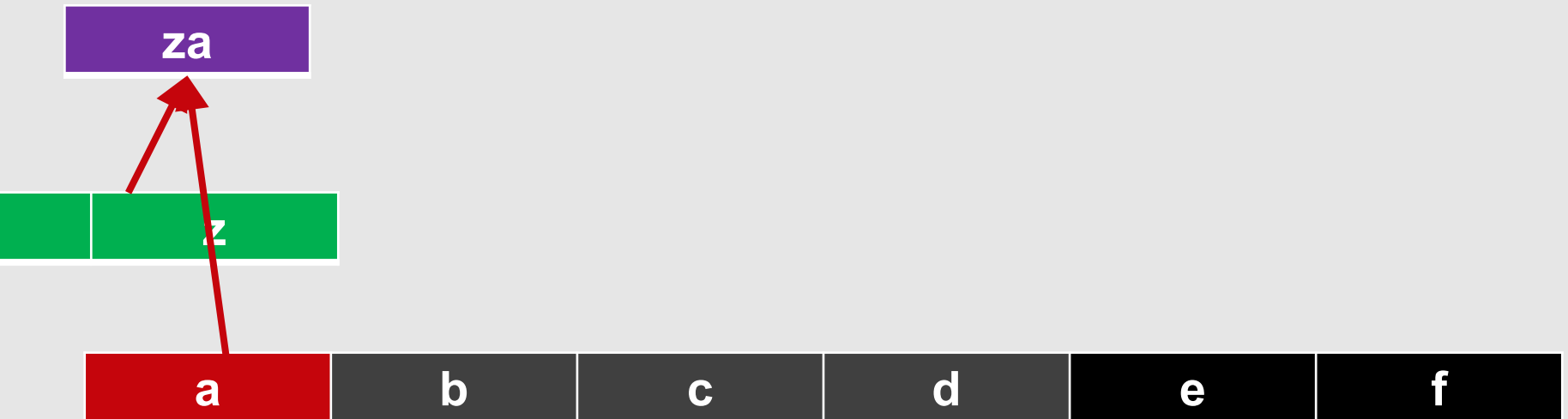


Option 2: SAME, stop when no overlap

Illustration 1: when to stop



Leftmost output entry



Option 2: SAME, stop when no overlap

Illustration 1: stride (#positions sliding per step)

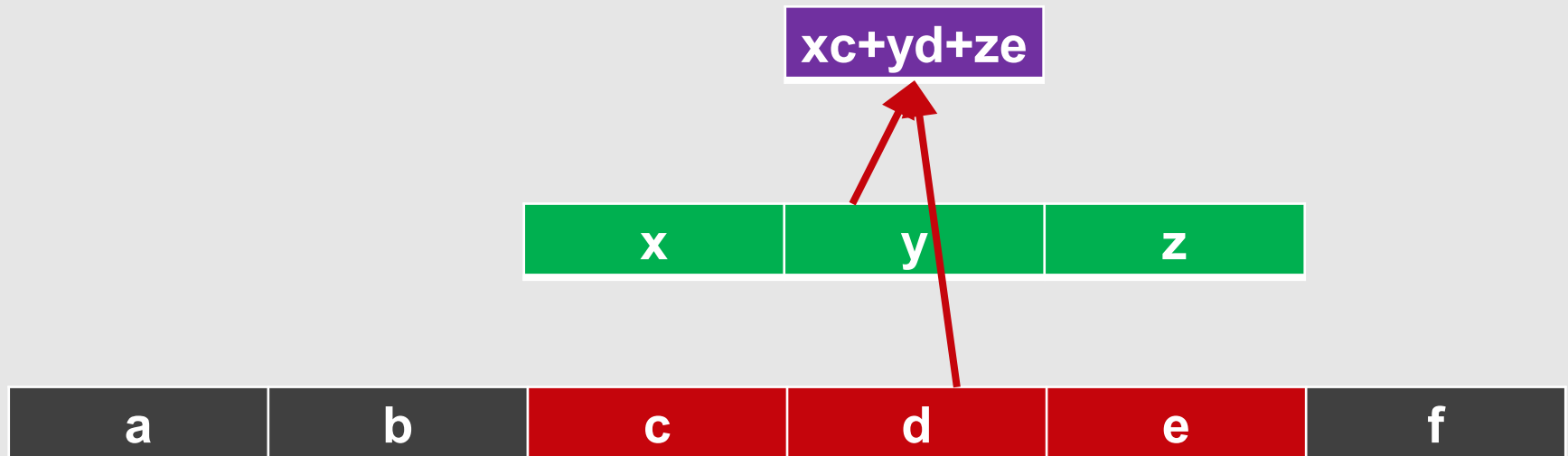
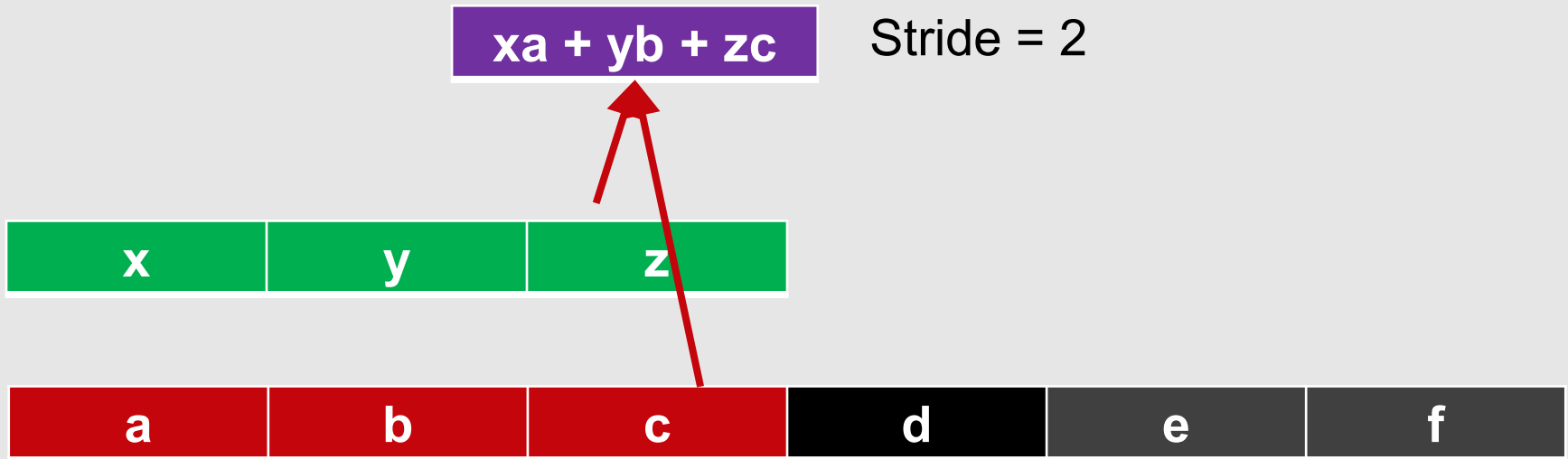
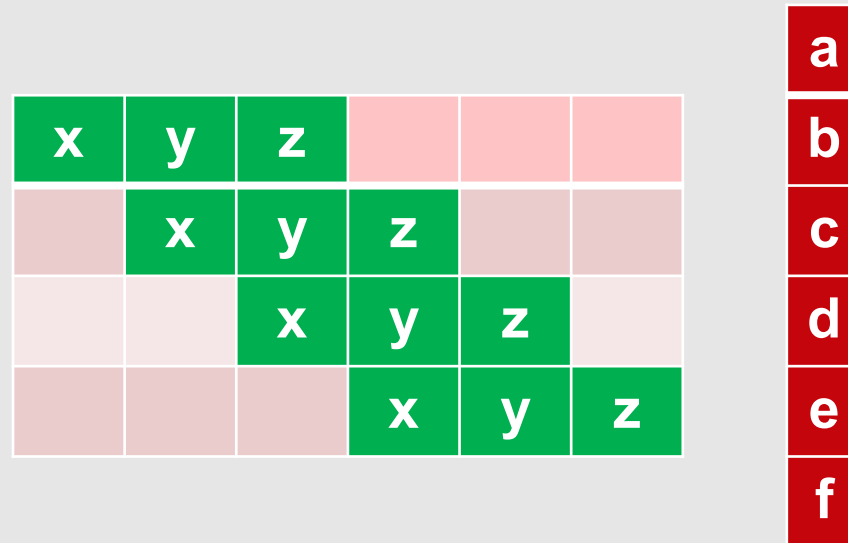


Illustration 1 as matrix multiplication



When using VALID, and Stride=1

Illustration 2: two dimensional case

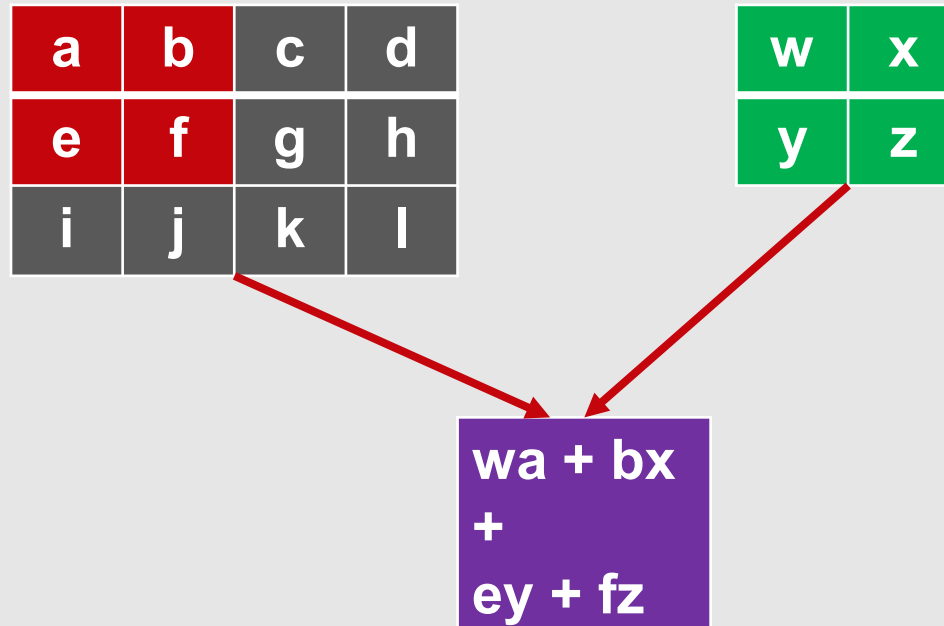


Illustration 2

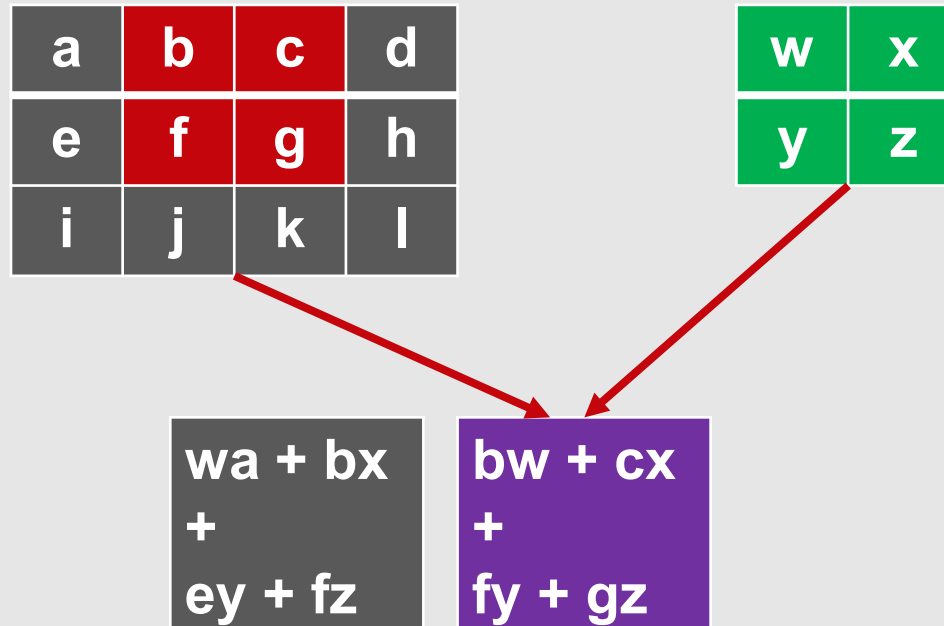


Illustration 2

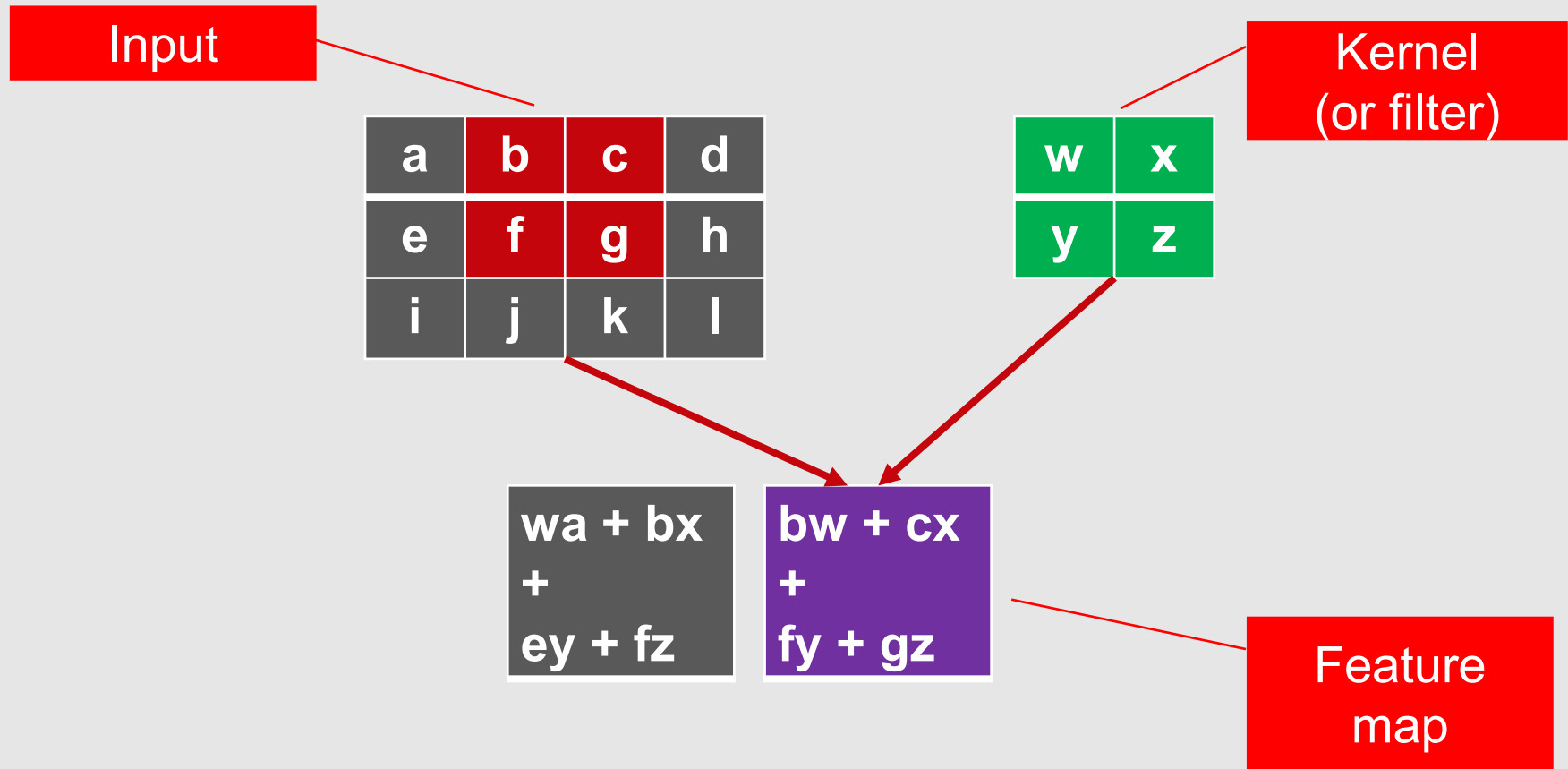
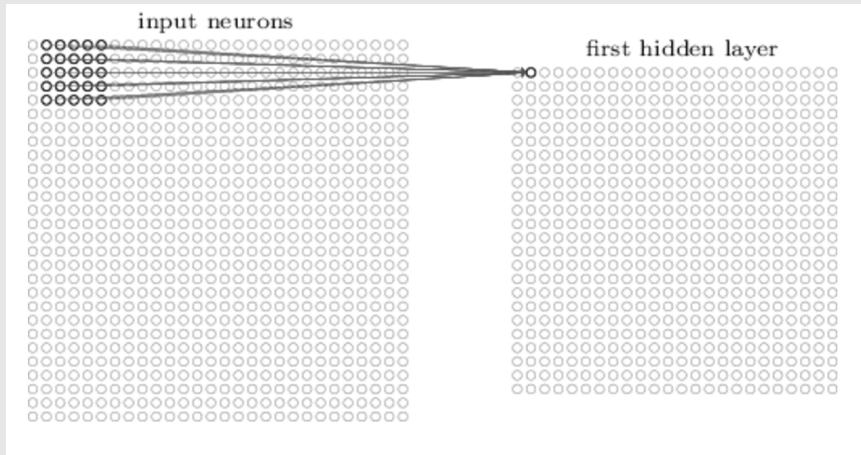
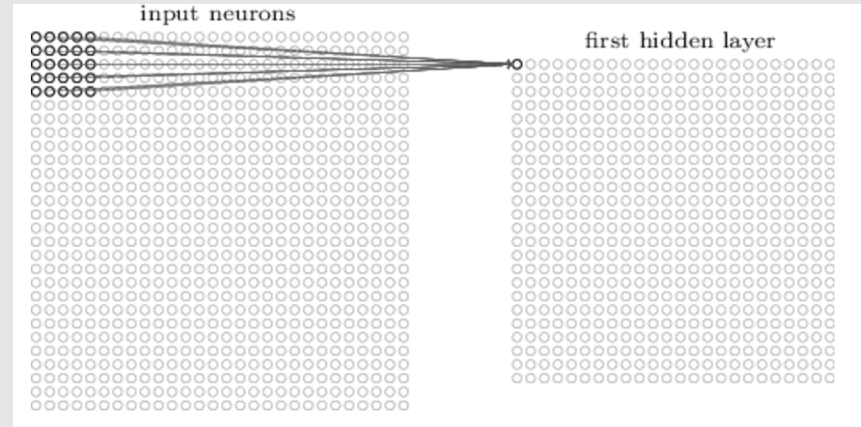


Illustration 2



- All the units used the same set of weights (kernel)
- The units detect the same “feature” but at different locations

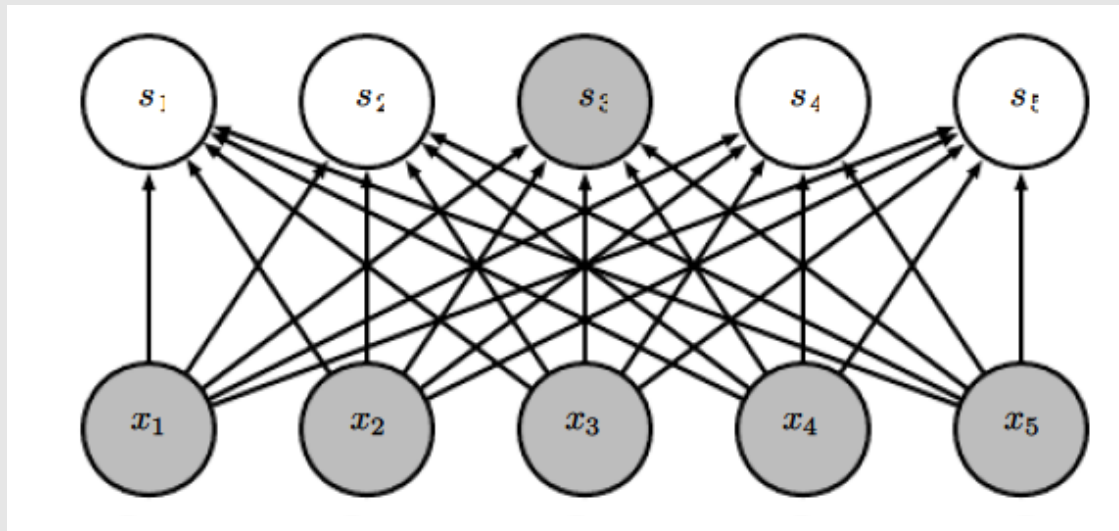


[Figure from neuralnetworksanddeeplearning.com]

Advantage: sparse interaction



Fully connected layer, $m \times n$ edges



m output nodes

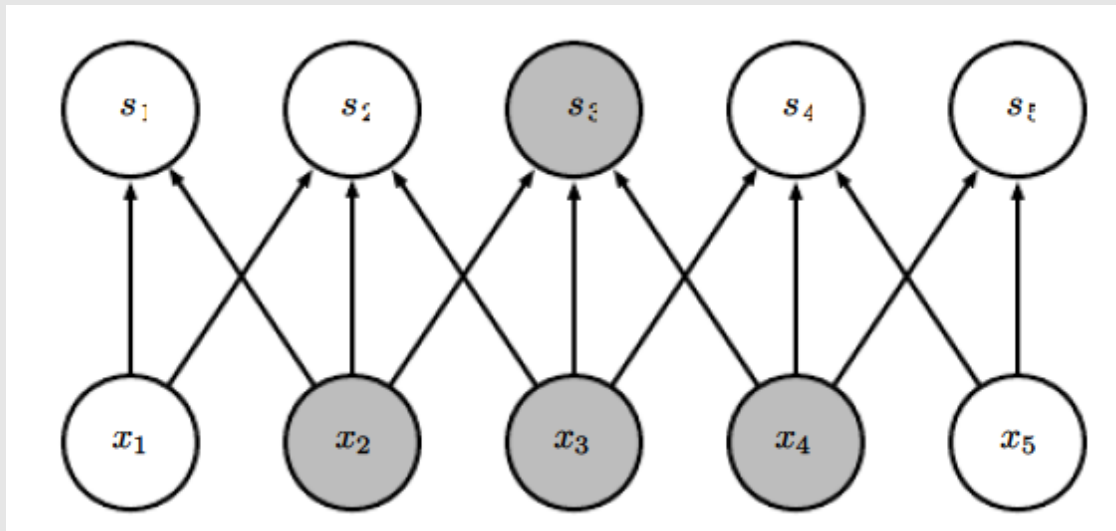
n input nodes

Figure from *Deep Learning*, by Goodfellow, Bengio, and Courville

Advantage: sparse interaction



Convolutional layer, $\leq m \times k$ edges



m output nodes

k kernel size

n input nodes

Figure from *Deep Learning*, by Goodfellow, Bengio, and Courville

Advantage: sparse interaction



Multiple convolutional layers: larger receptive field

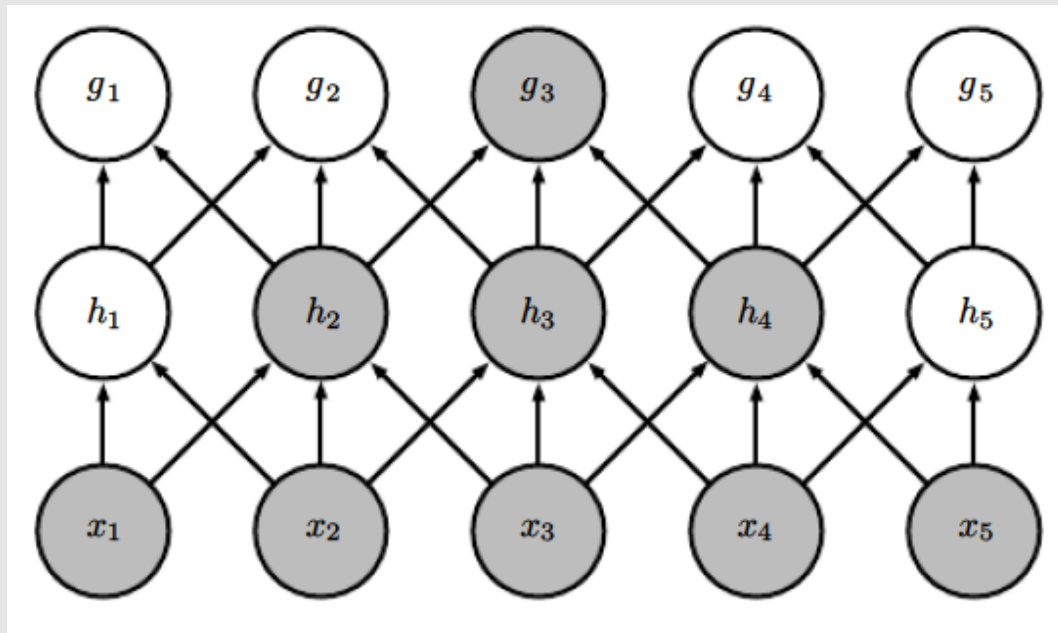
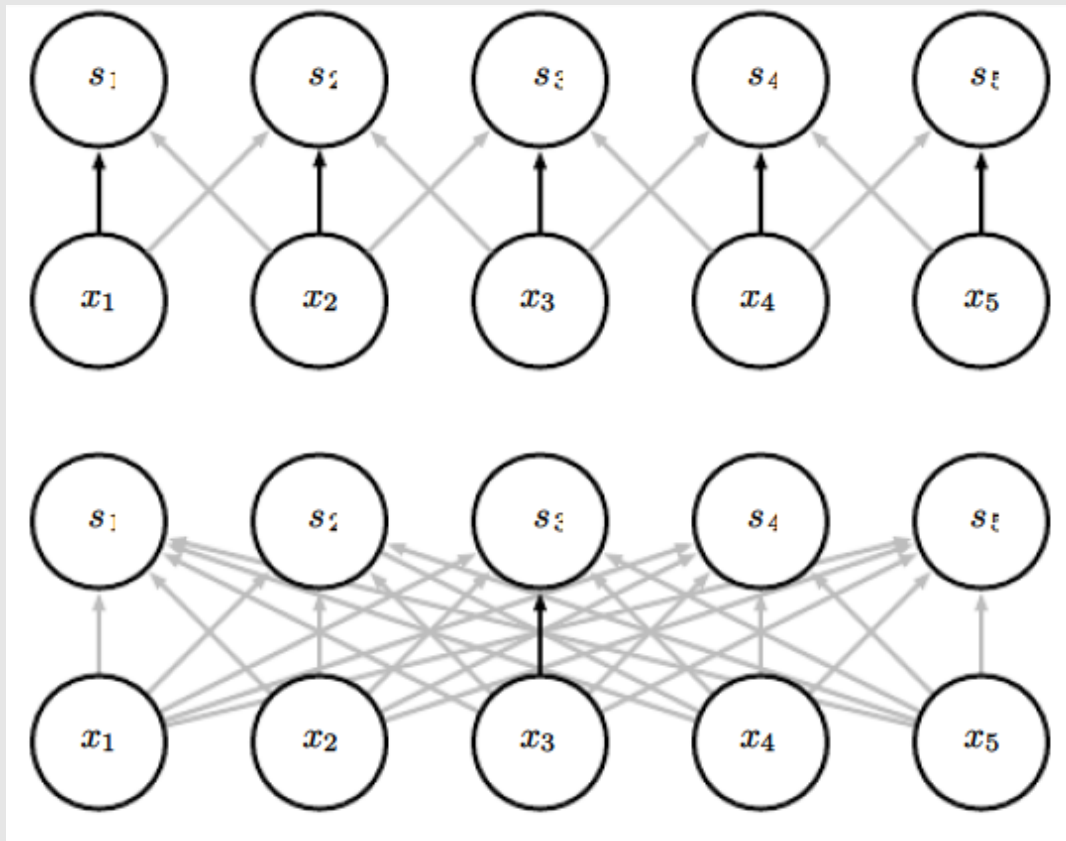


Figure from *Deep Learning*, by Goodfellow, Bengio, and Courville

Advantage: parameter sharing/weight tying



The same kernel are used repeatedly. E.g., the black edge is the same weight in the kernel.

Figure from *Deep Learning*, by Goodfellow, Bengio, and Courville

Advantage: equivariant representations



- Equivariant: transforming the input = transforming the output
- Example: input is an image, transformation is shifting
- $\text{Convolution}(\text{shift}(\text{input})) = \text{shift}(\text{Convolution}(\text{input}))$
- Useful when care only about the **existence** of a pattern, rather than the **location**



Pooling

Terminology

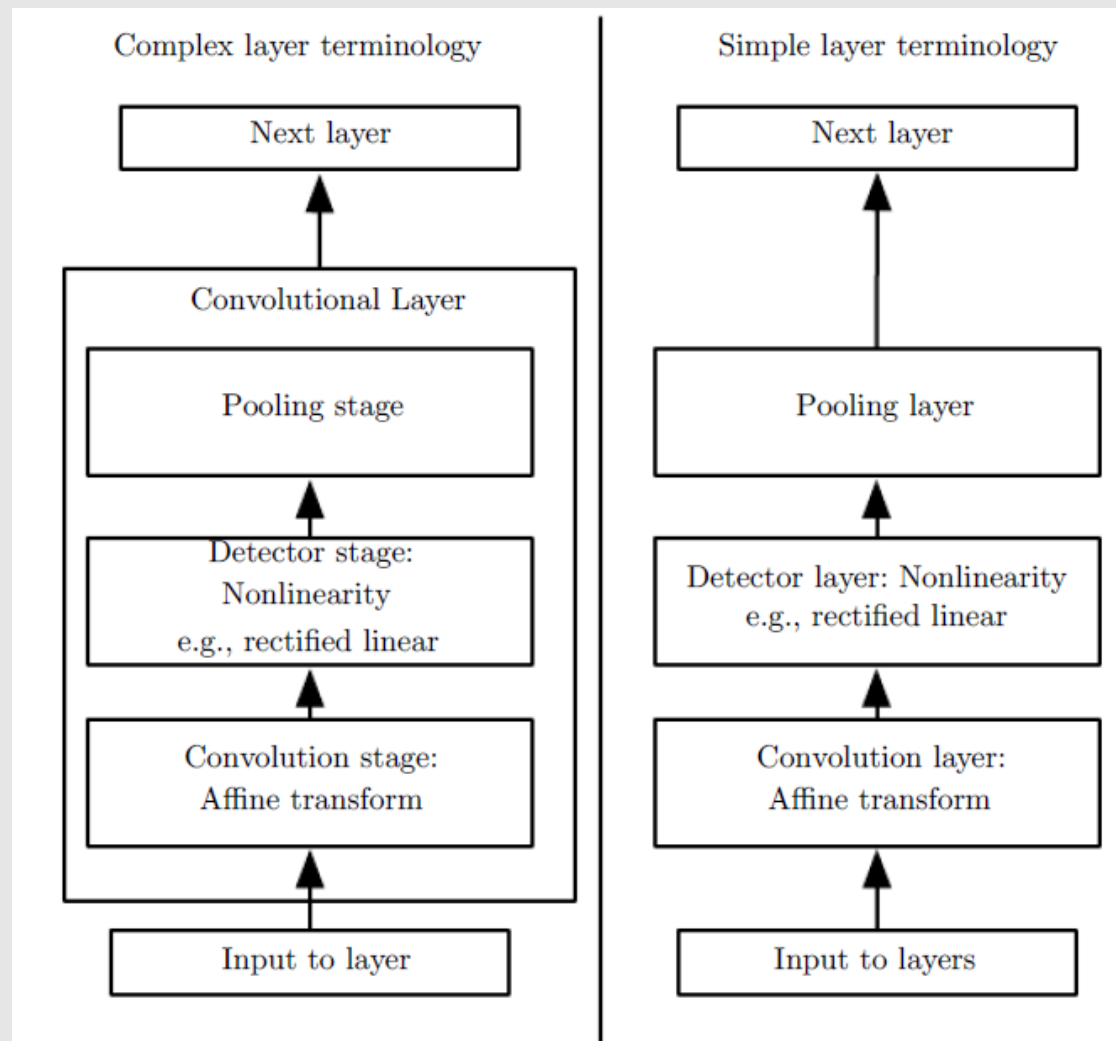


Figure from *Deep Learning*,
by Goodfellow, Bengio,
and Courville

Pooling



- Summarizing the input (i.e., output the max of the input)

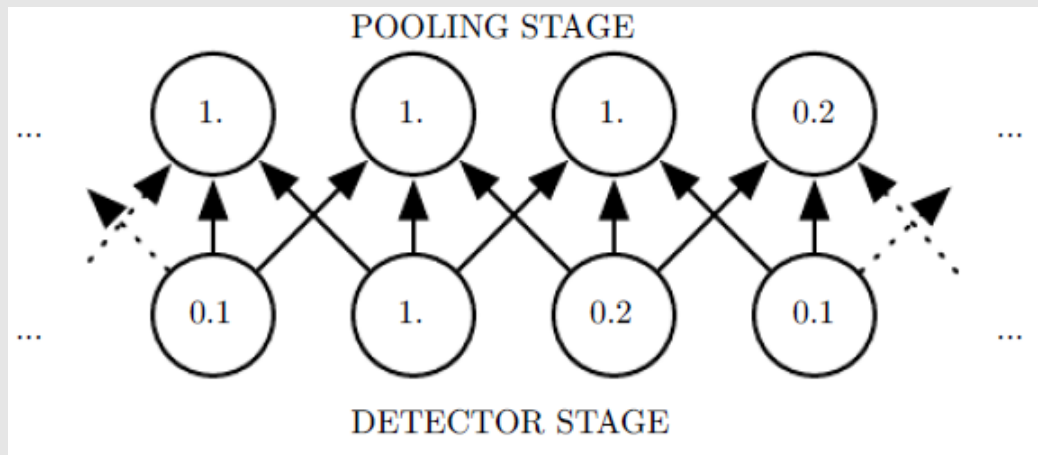
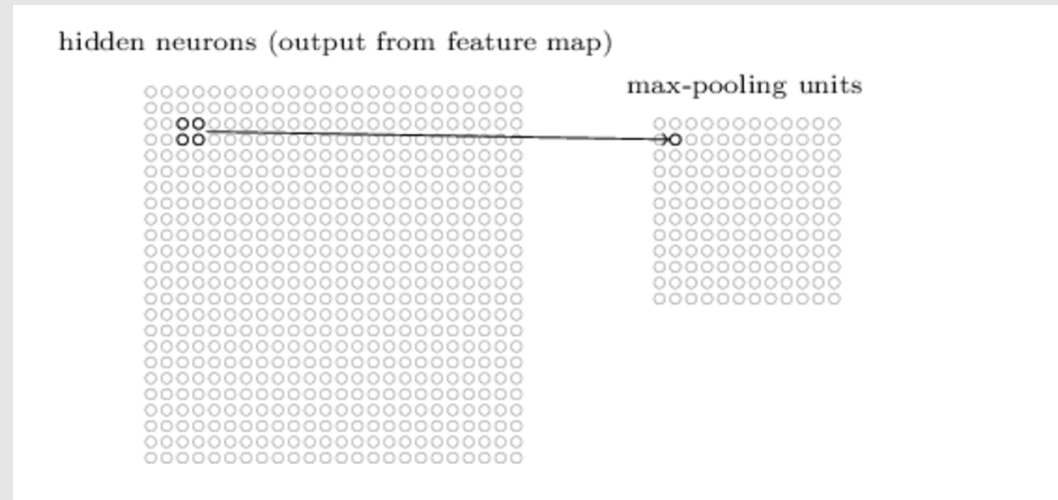


Figure from *Deep Learning*, by Goodfellow, Bengio, and Courville

Illustration



- Each unit in a pooling layer outputs a max, or similar function, of a subset of the units in the previous layer
- Variants: Max-pooling, Average-pooling,



[Figure from neuralnetworksanddeeplearning.com]

Advantage



Induce invariance

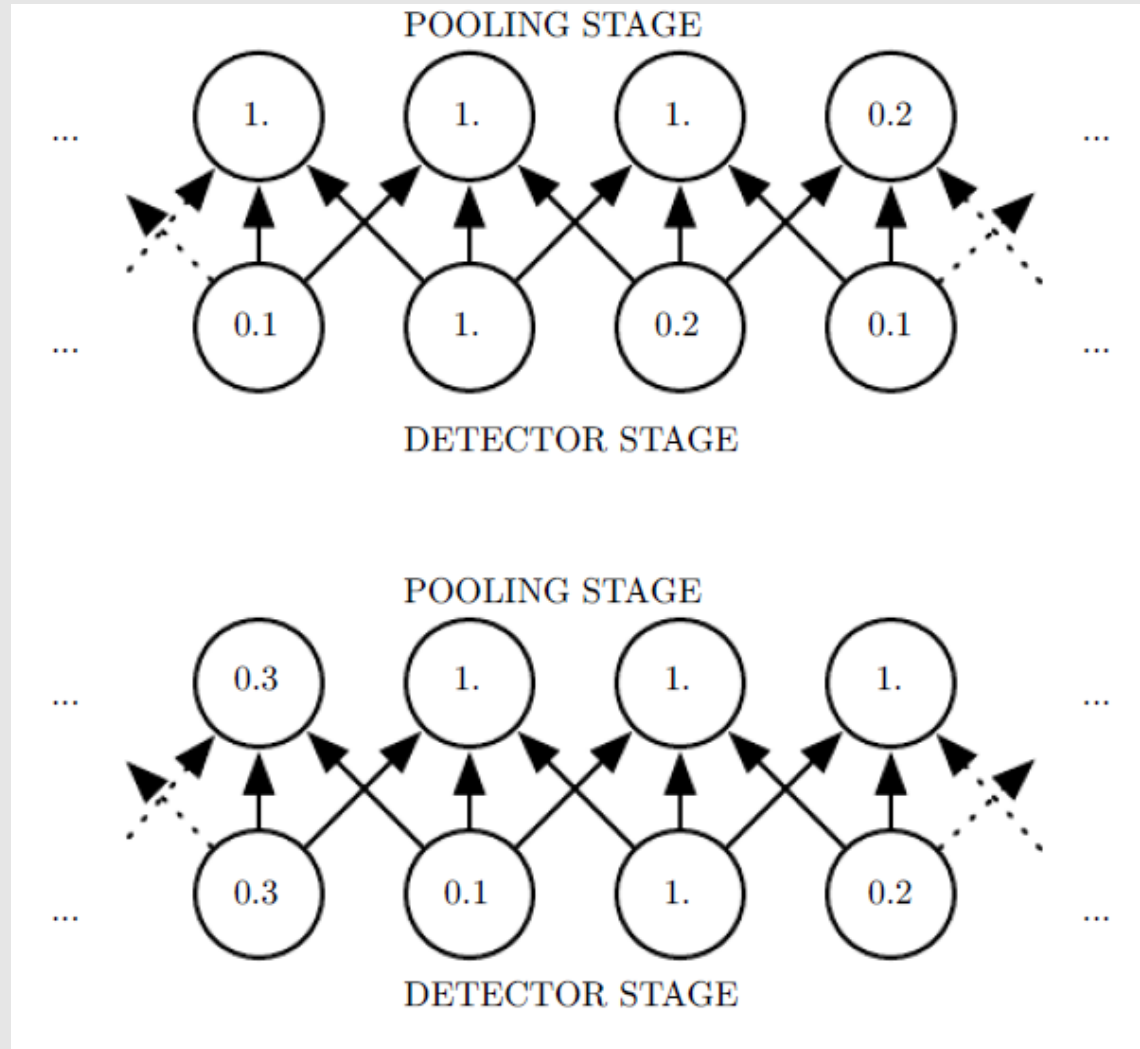


Figure from *Deep Learning*,
by Goodfellow, Bengio,
and Courville

Motivation from neuroscience



- David Hubel and Torsten Wiesel studied early visual system in human brain (V1 or primary visual cortex), and won Nobel prize for this
- V1 properties
 - 2D spatial arrangement
 - Simple cells: inspire convolution layers
 - Complex cells: inspire pooling layers



Example: LeNet

LeNet-5



- Proposed in “*Gradient-based learning applied to document recognition*”, by Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haffner, in *Proceedings of the IEEE*, 1998
- Apply convolution on 2D images (MNIST) and use backpropagation
- Structure: 2 convolutional layers (with pooling) + 3 fully connected layers
 - Input size: 32x32x1
 - Convolution kernel size: 5x5
 - Pooling: 2x2

LeNet-5

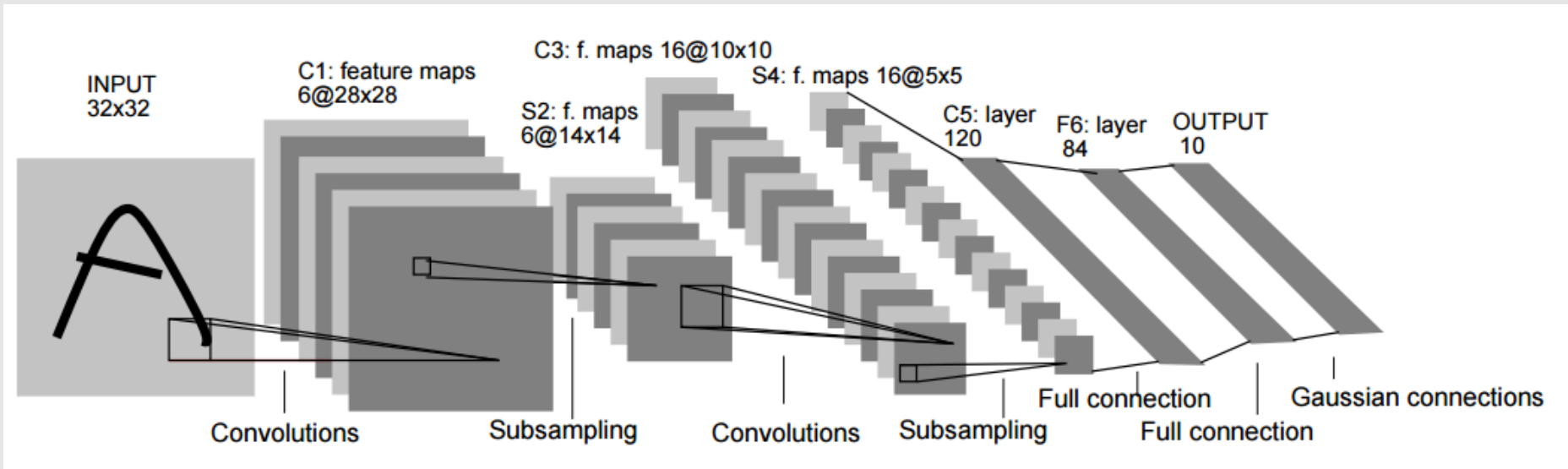


Figure from *Gradient-based learning applied to document recognition*, by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner

LeNet-5

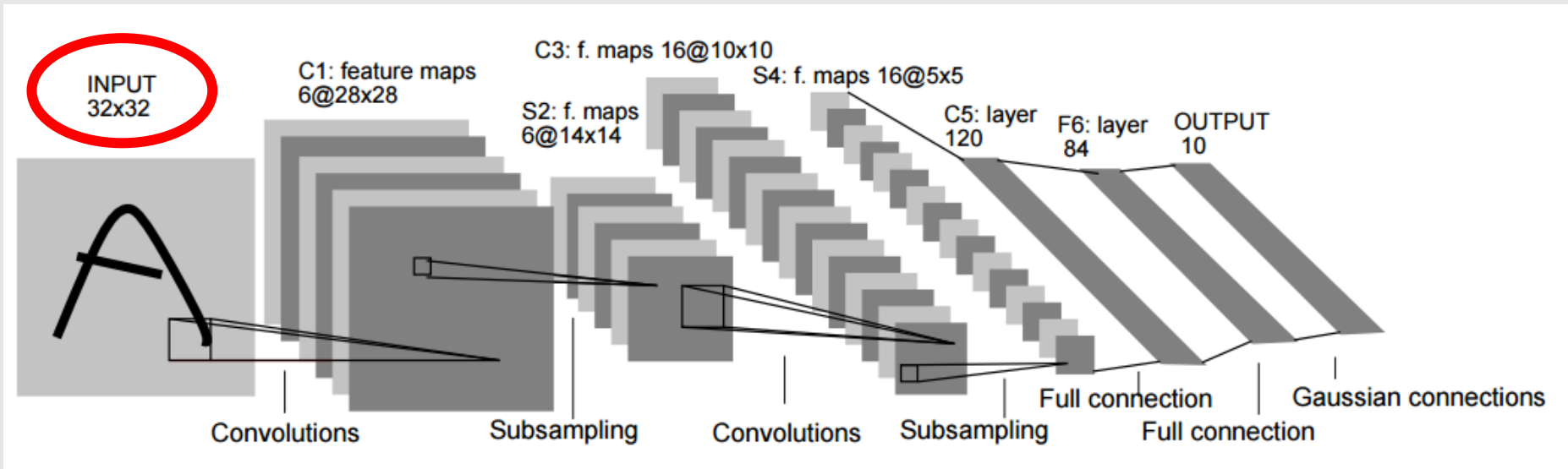


Figure from *Gradient-based learning applied to document recognition*, by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner

LeNet-5

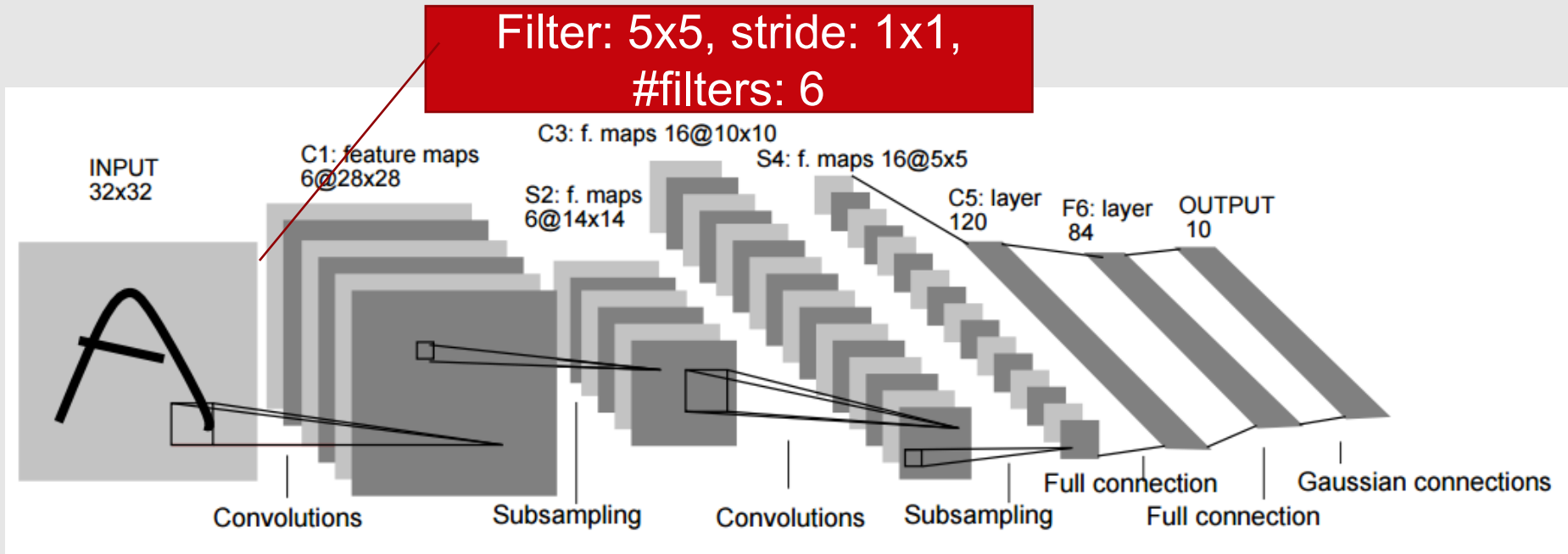


Figure from *Gradient-based learning applied to document recognition*, by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner

LeNet-5

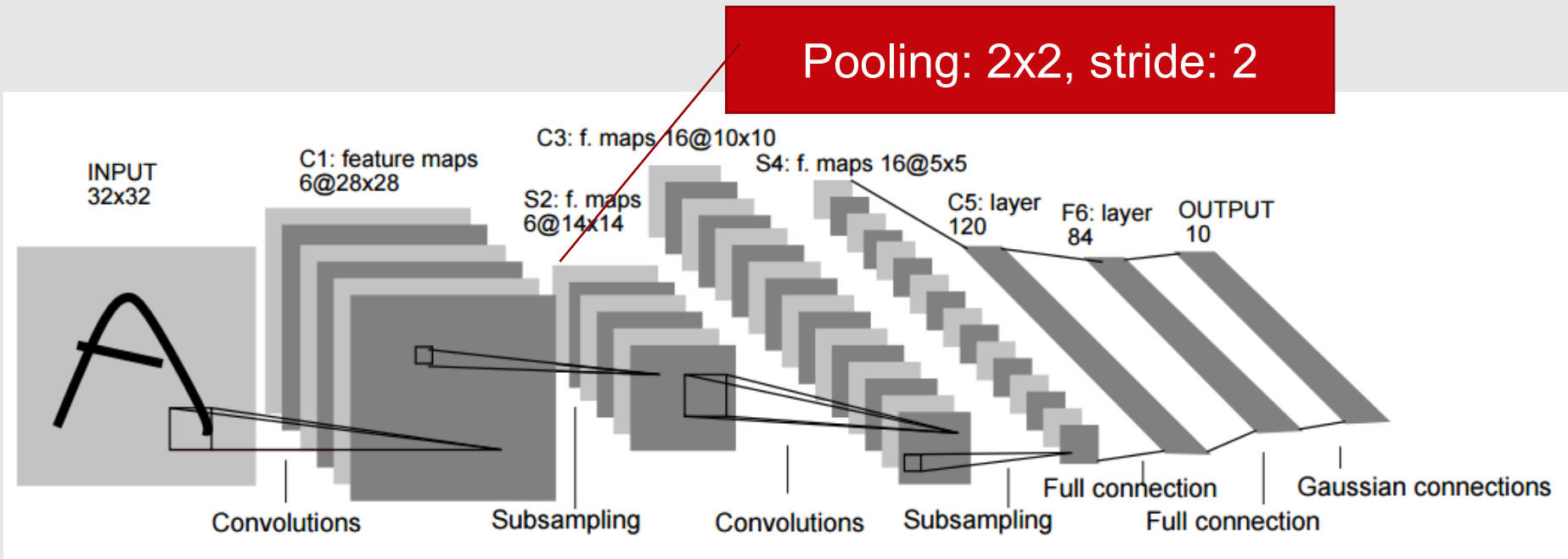


Figure from *Gradient-based learning applied to document recognition*, by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner

LeNet-5

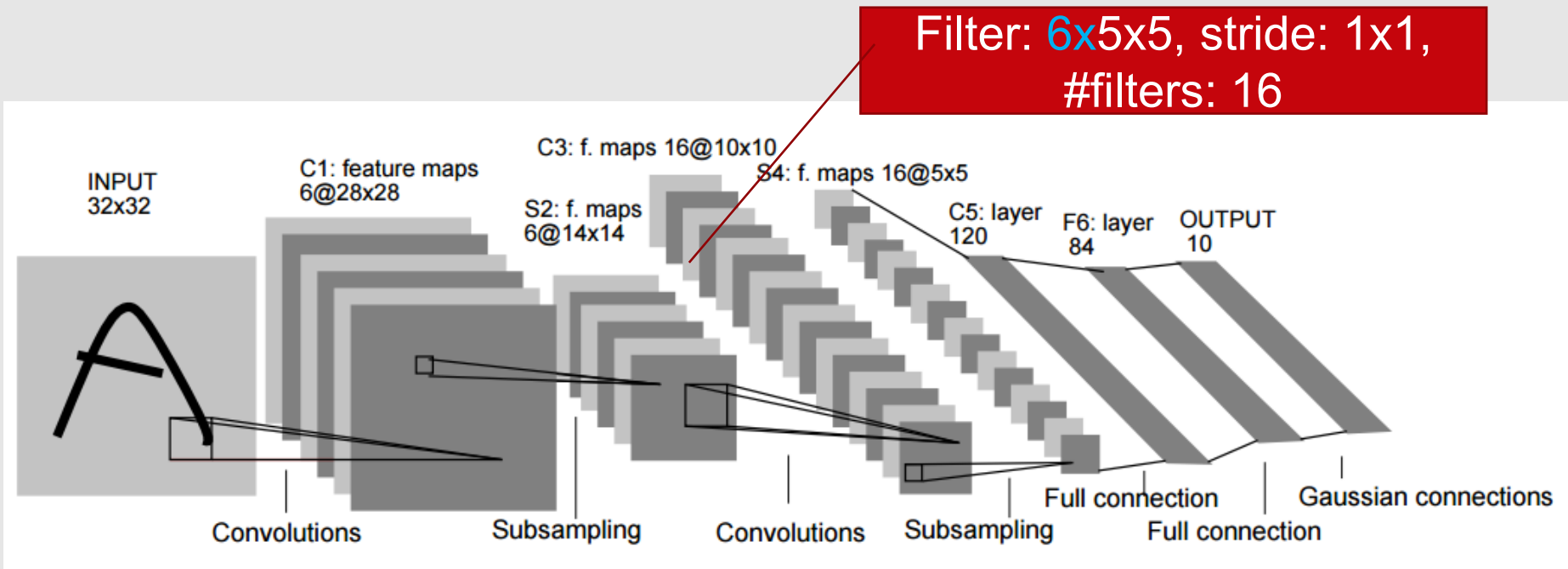


Figure from *Gradient-based learning applied to document recognition*, by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner

LeNet-5

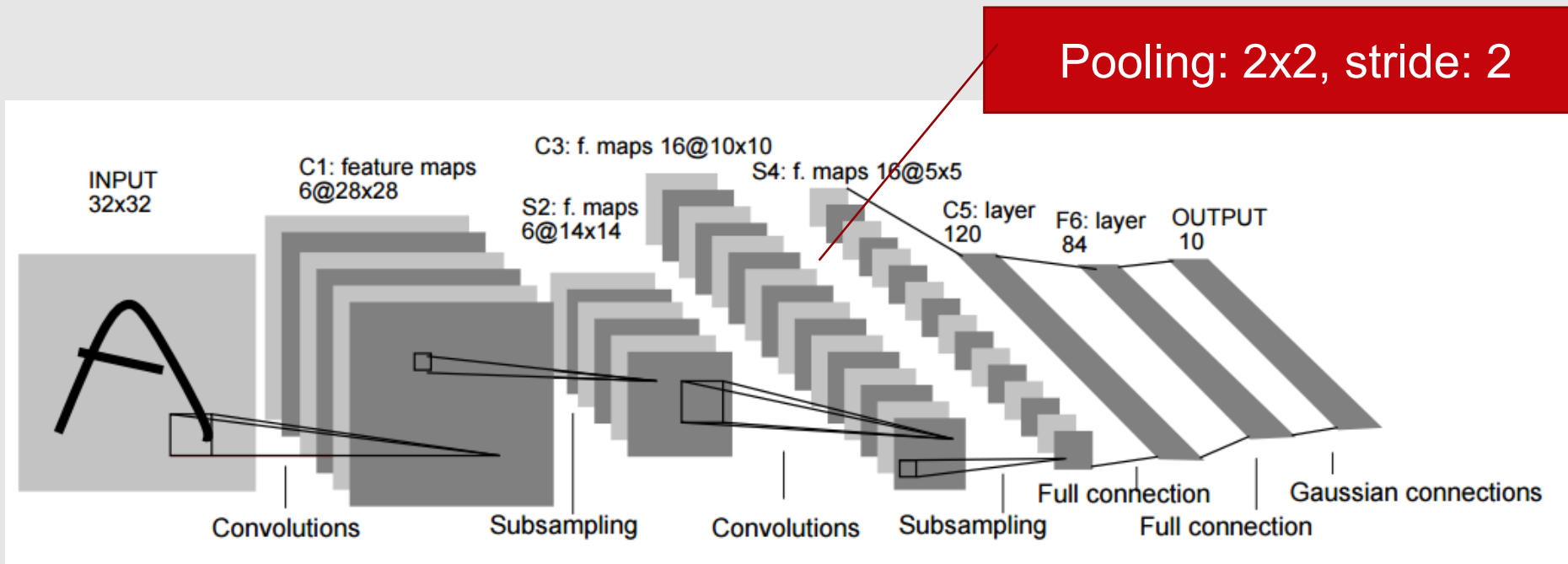


Figure from *Gradient-based learning applied to document recognition*, by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner

LeNet-5



Weight matrix: 400x120

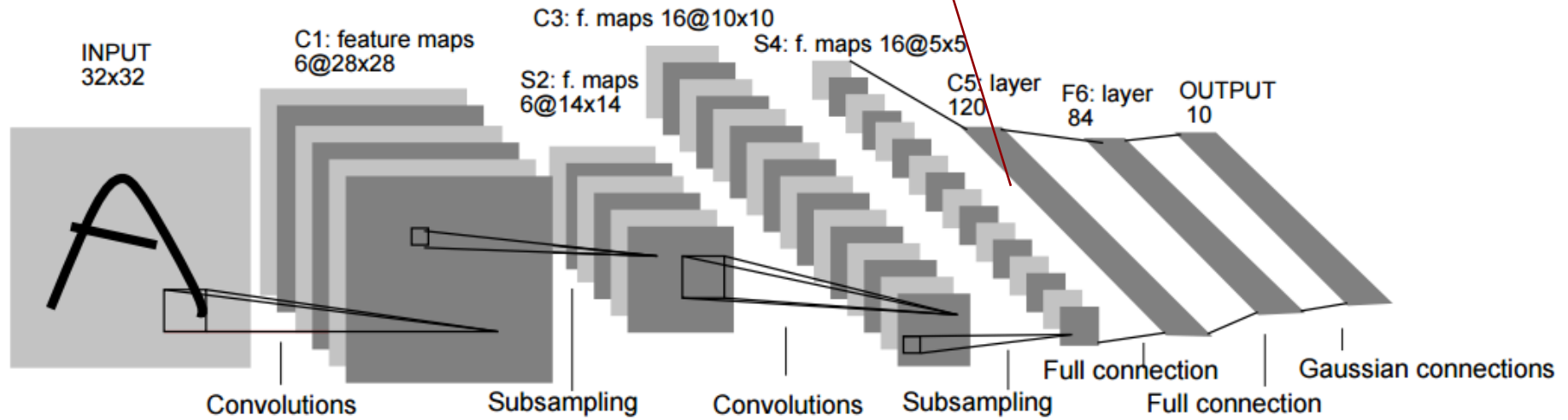


Figure from *Gradient-based learning applied to document recognition*, by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner

LeNet-5

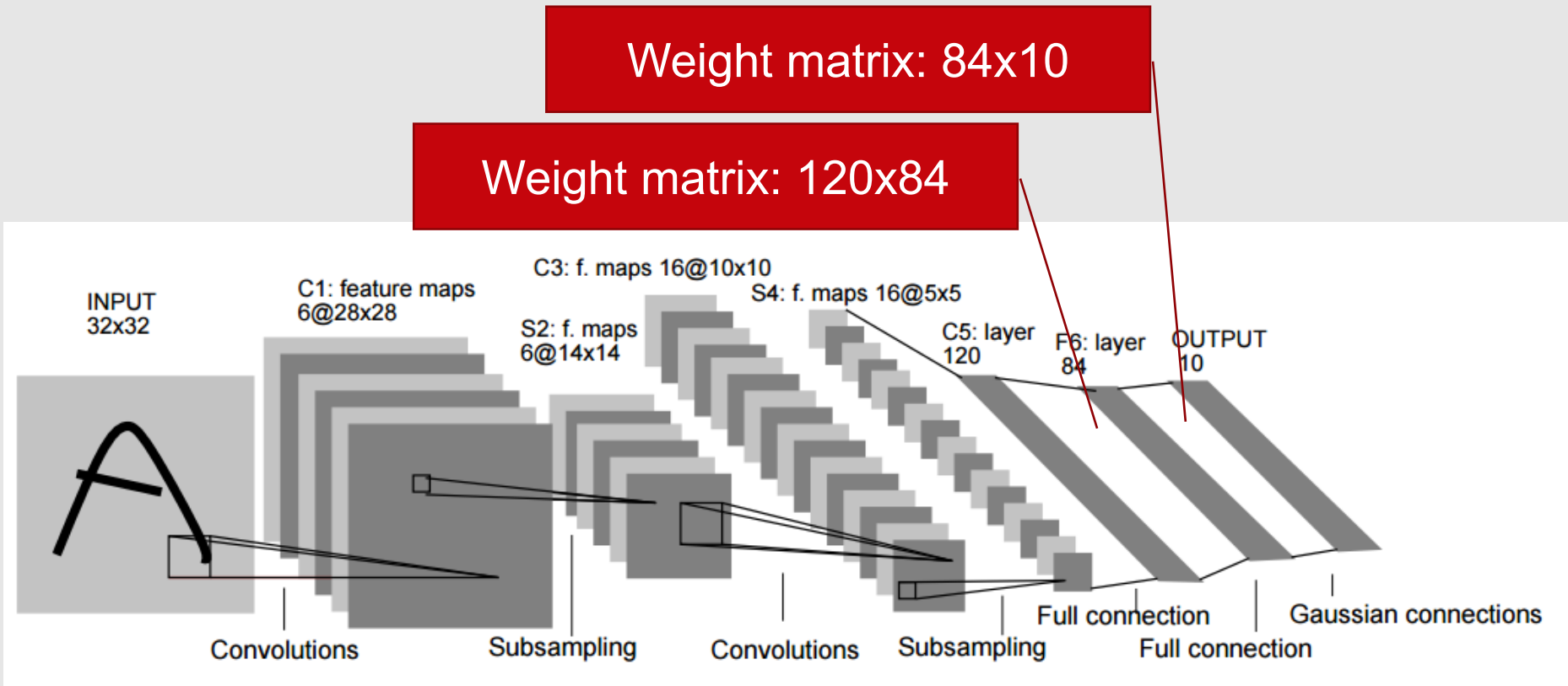


Figure from *Gradient-based learning applied to document recognition*, by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner



Optional: ResNet

ResNet

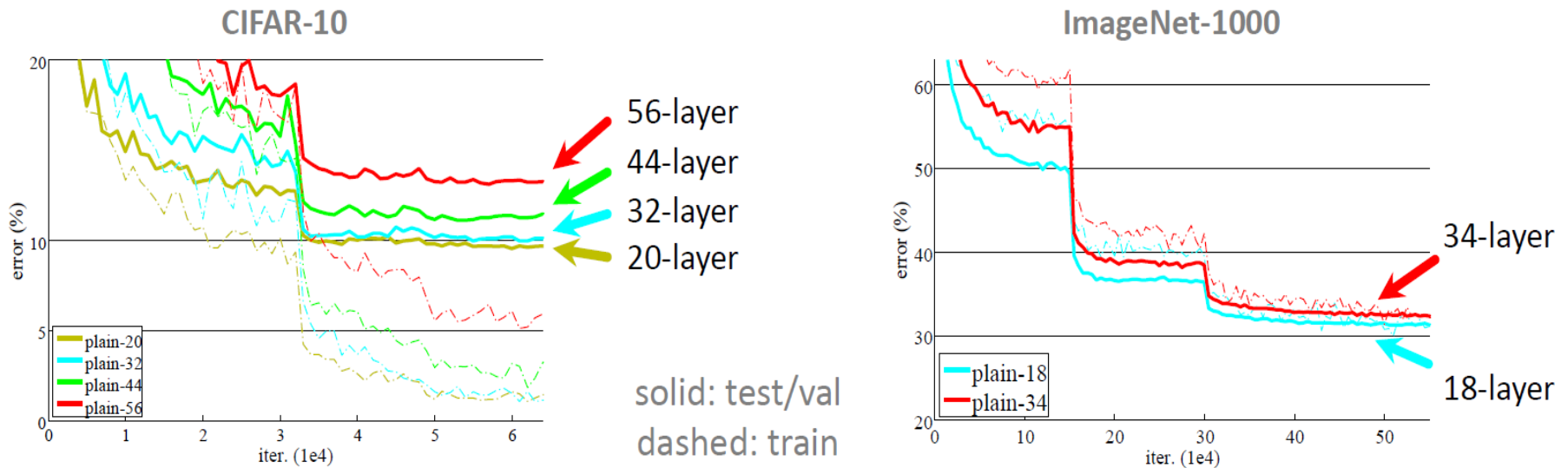


- Proposed in “Deep residual learning for image recognition” by *He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*,. 2016.
- Apply very deep networks with repeated residue blocks
- Structure: simply stacking residue blocks



Plain Network

- “Overly deep” plain nets have higher training error
- A general phenomenon, observed in many datasets

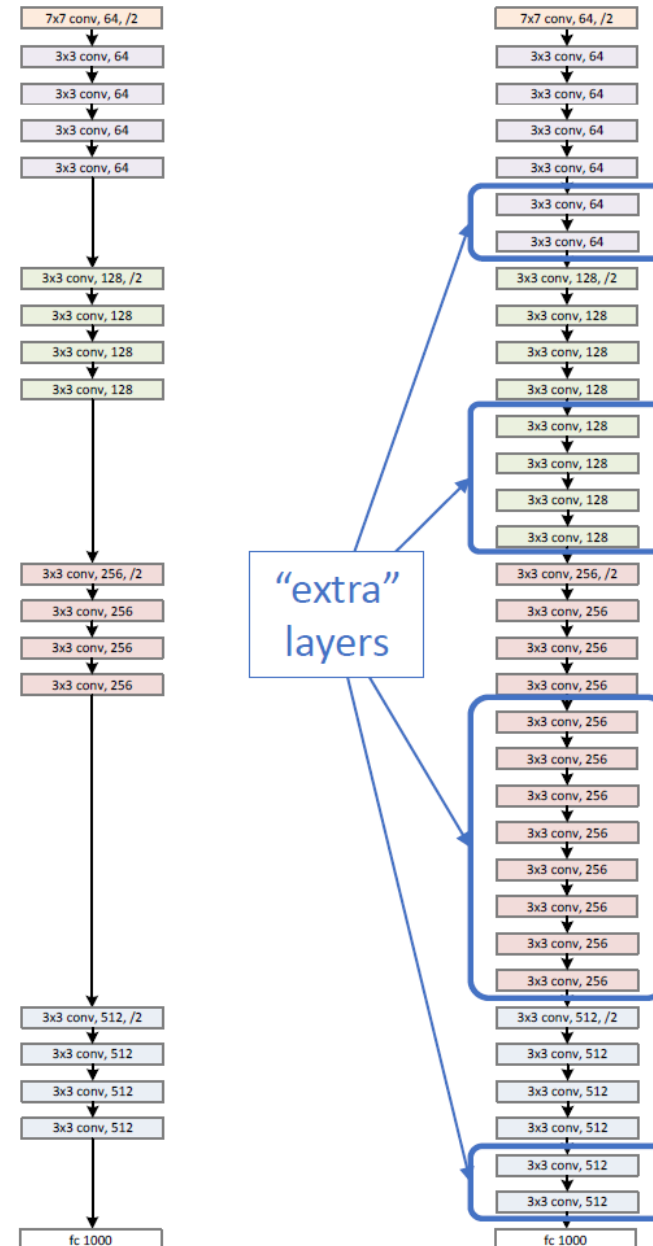


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

Residual Network



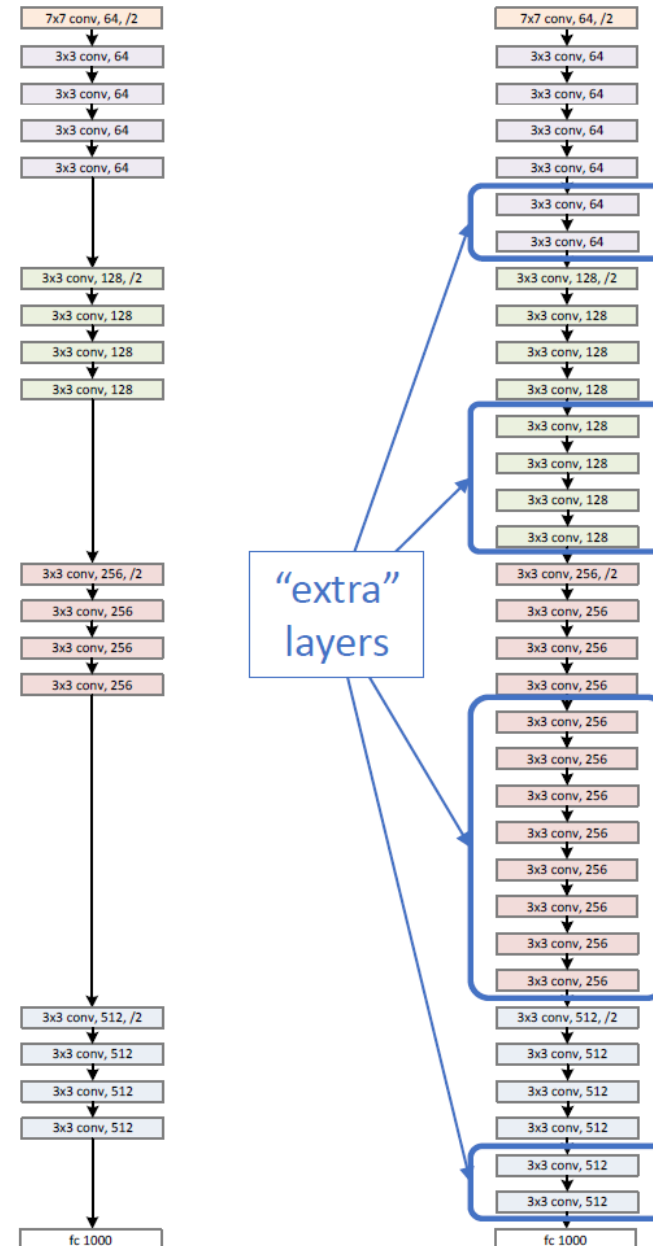
- Naïve solution
 - If extra layers are an **identity** mapping, then a training errors does not increase



Residual Network



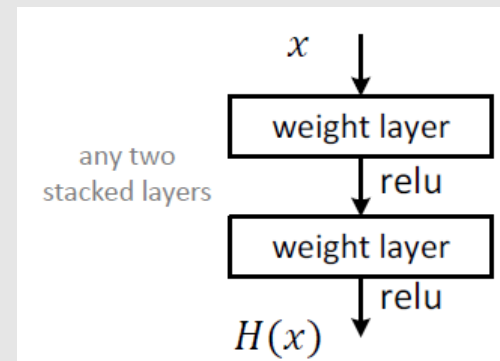
- Deeper networks also maintain the tendency of results
 - Features in same level will be almost same
 - An amount of changes is fixed
 - Adding layers makes smaller differences
 - Optimal mappings are closer to an **identity**



Residual Network



- Plain block
 - Difficult to make identity mapping because of multiple non-linear layers



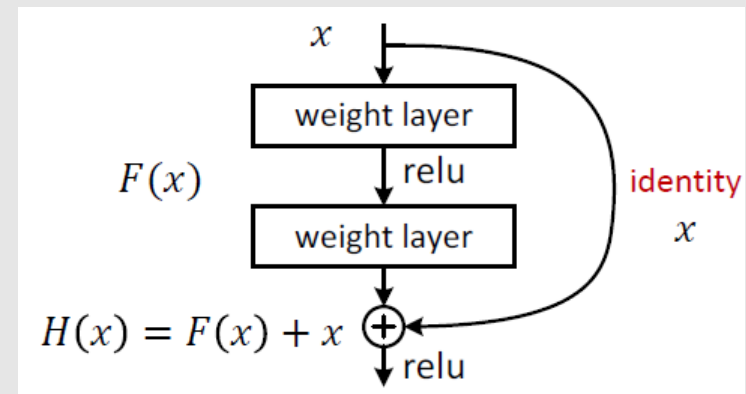
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun.
"Deep Residual Learning for Image Recognition". arXiv 2015.

Residual Network



- Residual block
 - If identity were optimal, easy to set weights as 0
 - If optimal mapping is closer to identity, easier to find small fluctuations

-> Appropriate for treating **perturbation** as keeping a base information



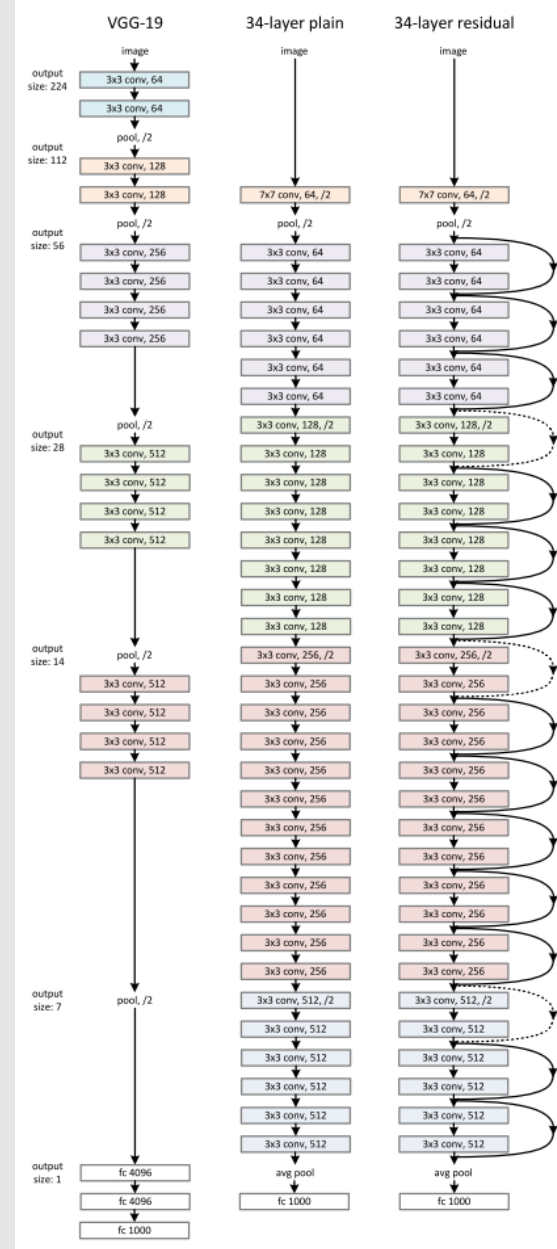
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun.
"Deep Residual Learning for Image Recognition". arXiv 2015.



Network Design

- Basic design (VGG-style)
 - All 3x3 conv (almost)
 - Spatial size/2 => #filters x2
 - Batch normalization
 - Simple design, just deep
- Other remarks
 - No max pooling (almost)
 - No hidden fc
 - No dropout

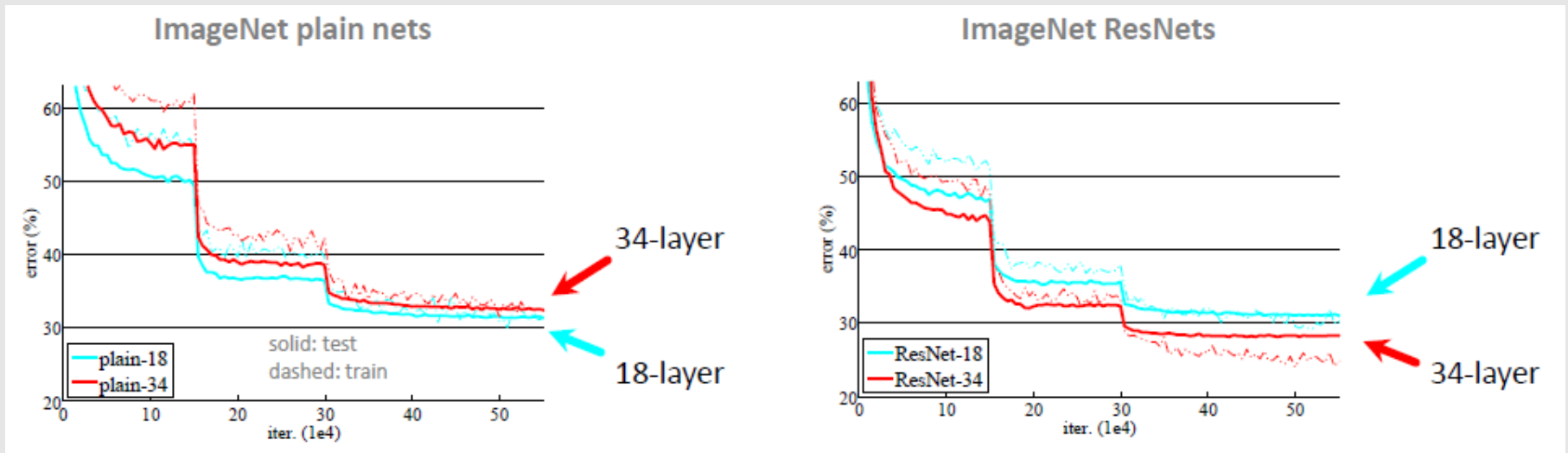
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.





Results

- Deep Resnets can be trained without difficulties
- Deeper ResNets have lower training error, and also lower test error



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.

Results



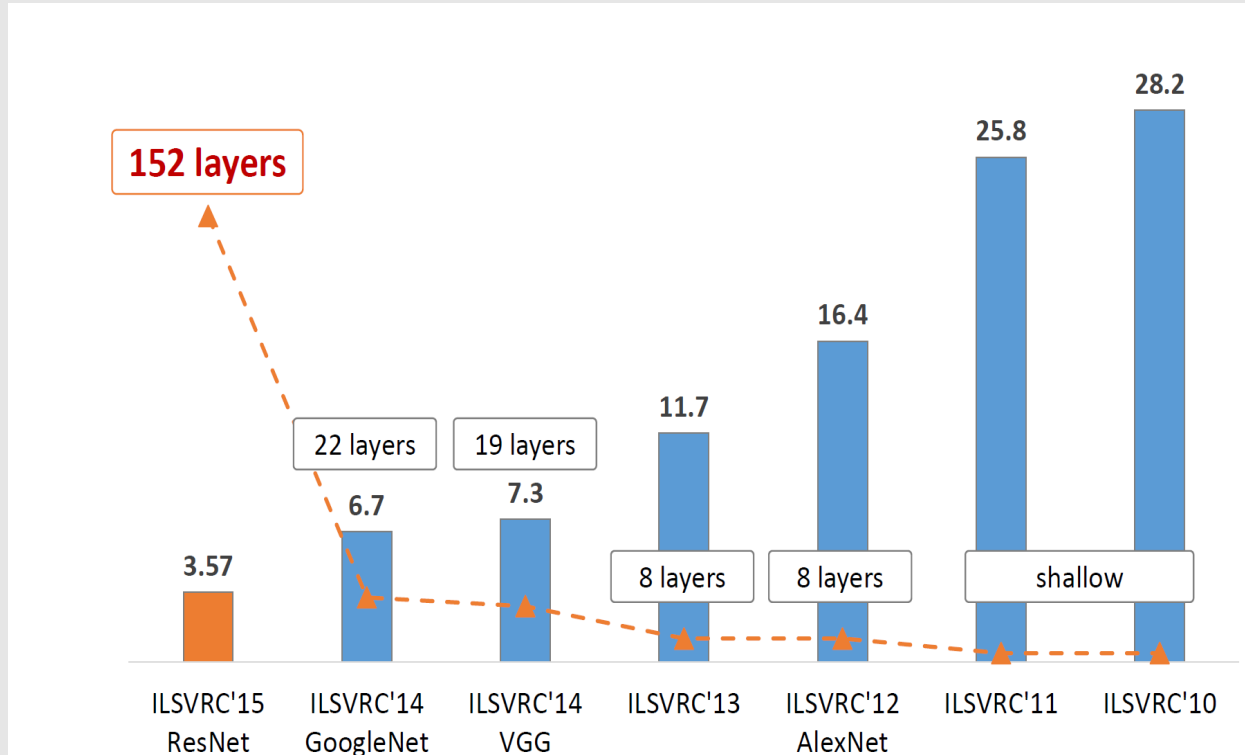
- 1st places in all five main tracks in “ILSVRC & COCO 2015 Competitions”
 - ImageNet Classification
 - ImageNet Detection
 - ImageNet Localization
 - COCO Detection
 - COCO Segmentation

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

Quantitative Results



- ImageNet Classification

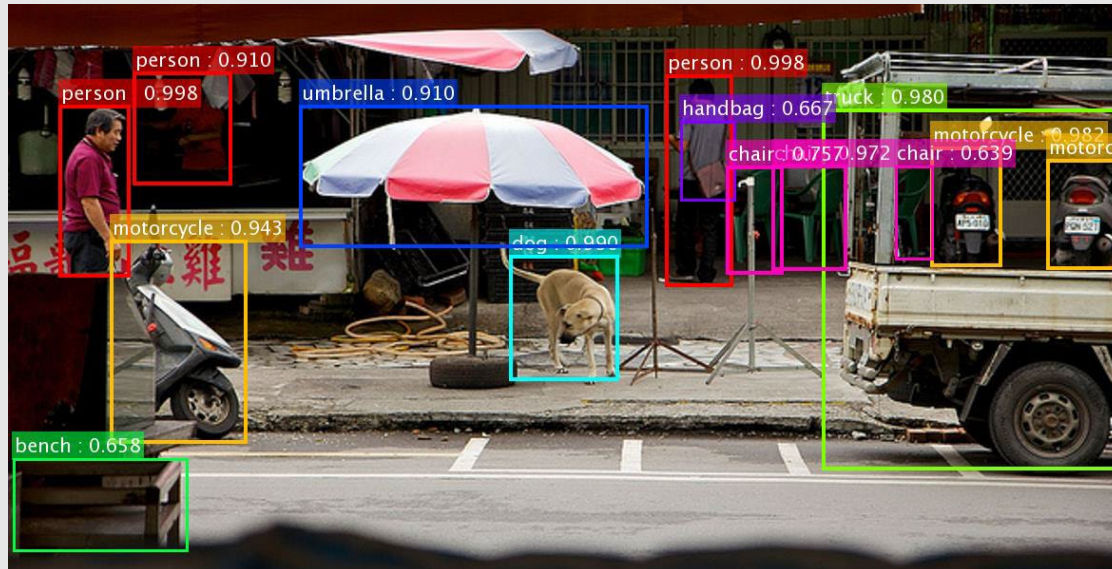


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.

Qualitative Result



- Object detection
 - Faster R-CNN + ResNet

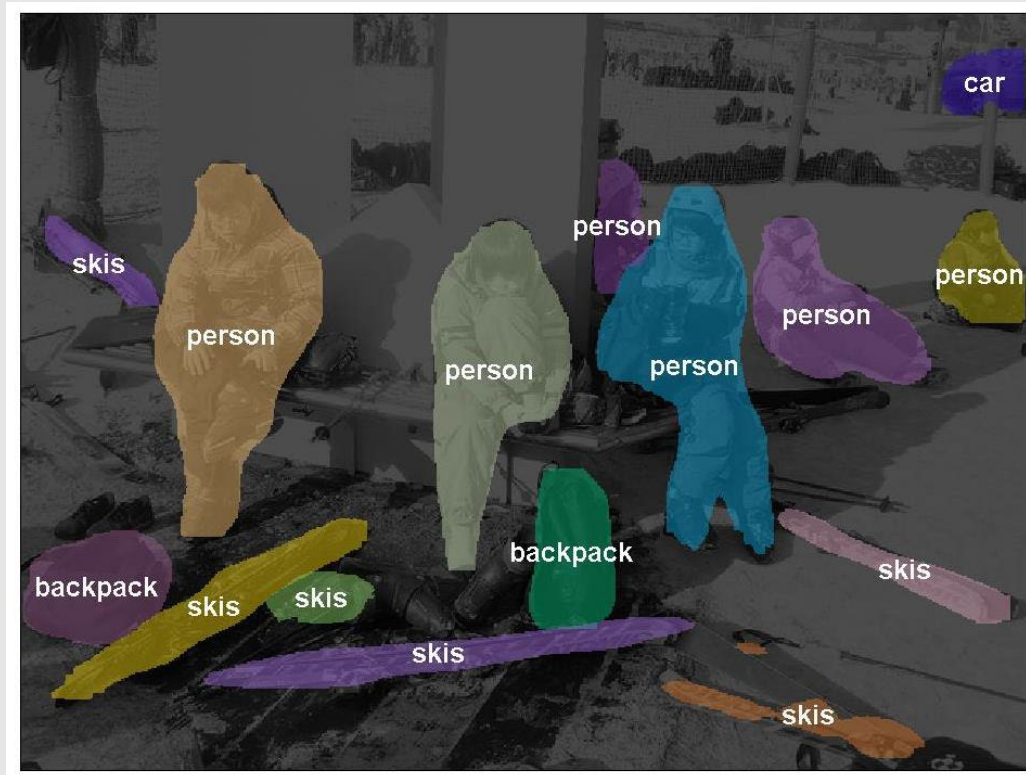


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.
Jifeng Dai, Kaiming He, & Jian Sun. "Instance-aware Semantic Segmentation via Multi-task Network Cascades". arXiv 2015.

Qualitative Results



- Instance Segmentation



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.



THANK YOU

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Matt Gormley, Elad Hazan, Tom Dietterich, and Pedro Domingos.

