African Institute for Mathematical Sciences
RWANDA

AMMI Review sessions

Deep Learning (5)

Convnets II

18 Feb 2020

Parameter sharing

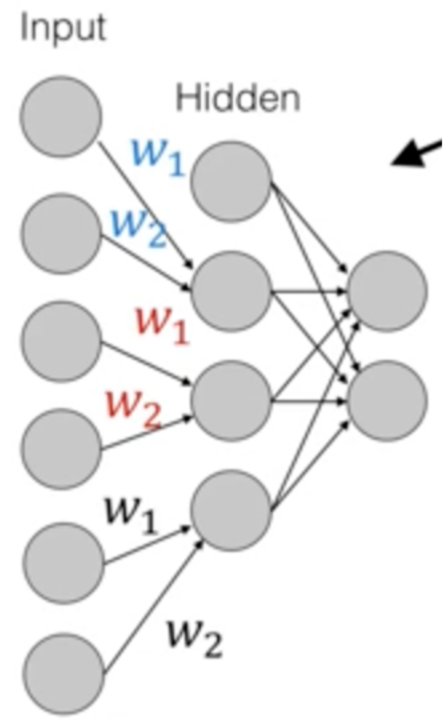
- Parameter sharing refers to using the same parameter for more than one function in a model, i.e the value of the weight applied to one input is tied to the value of a weight applied elsewhere

- kernels can extract similar features at multiple places in an image

- Convolutions are **equivariant to translation** - i.e. translating an input in space will result in a translated convolution output

- Convolution is not naturally equivariant to some other transformations, such as changes in the scale or rotation of an image.

- Useful features which have been learned can be **reused** at multiple regions in an image



Convolution arithmetics

To Calculate the output dimensions of an input (W x H) and a filter with the size ($F_{w,}$ F_h), horizontal stride S_w and a vertical one S_h with Padding P:

- Output width = $\frac{(W - F_w + 2P)}{S_w} + 1$
- Output height = $\frac{(H - F_h + 2P)}{S_h} + 1$

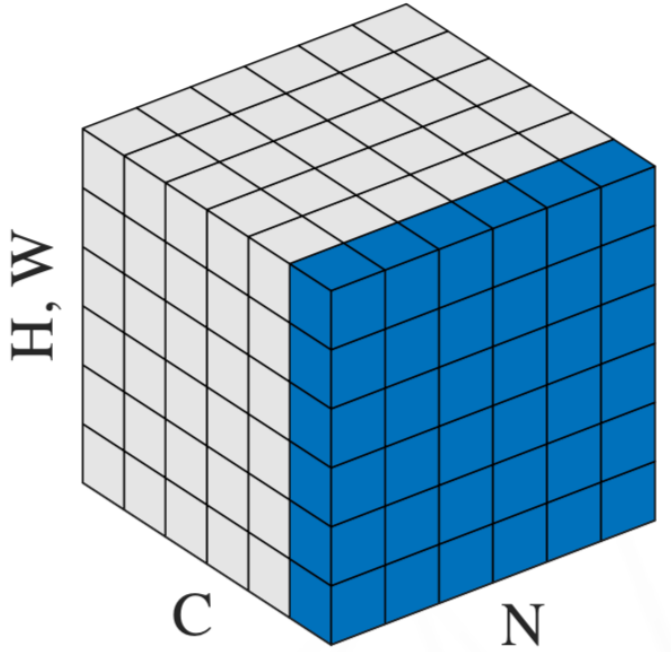
To calculate the output dimensions of pooling operation

- Output width = $\frac{(W - F)}{S} + 1$
- Output height = $\frac{(H - F)}{S} + 1$

Batch Normalization

Z-score input: $\hat{\mathbf{x}} = \dfrac{\mathbf{x} - \mu}{\sigma}$

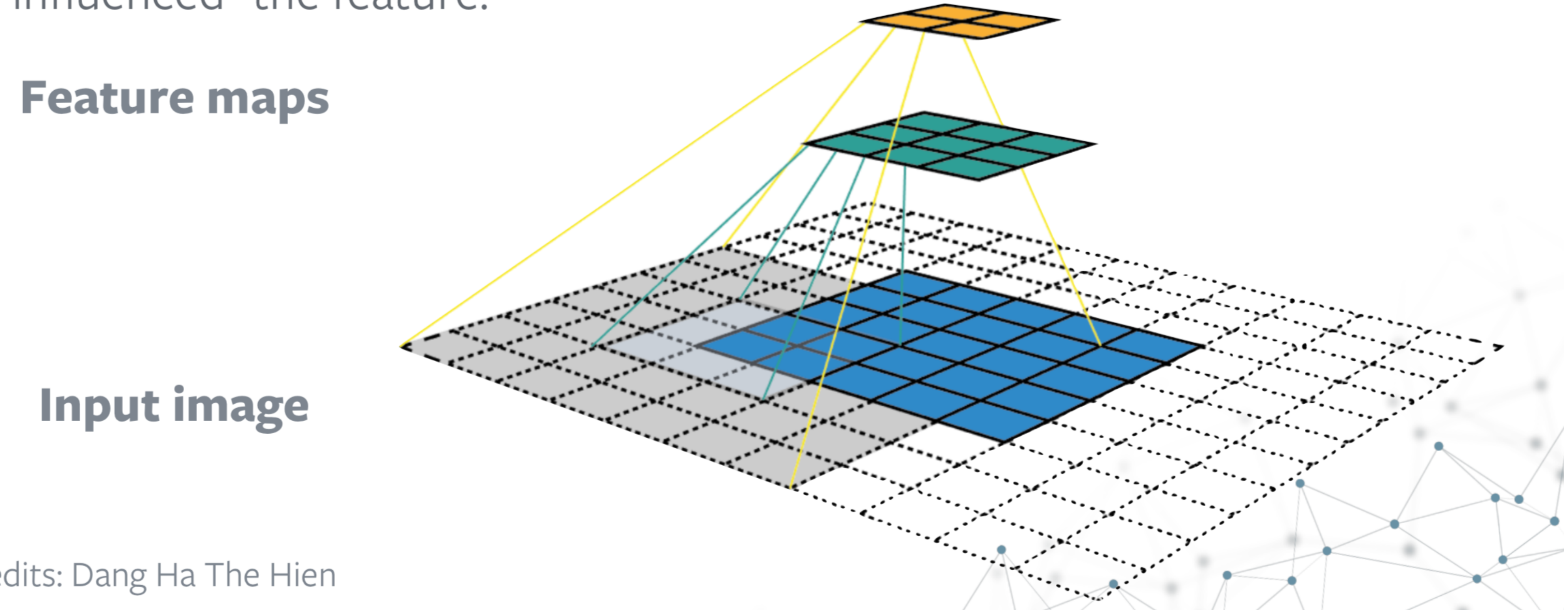
Affine transform: $\mathbf{y} = \gamma\hat{\mathbf{x}} + \beta$



* Note: Batch normalization works differently in training and test time.

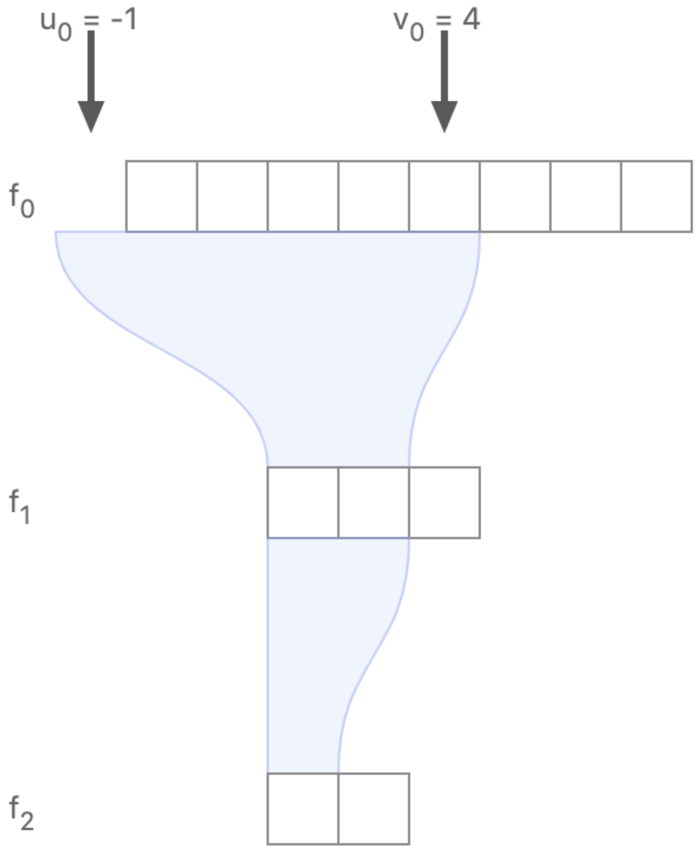
Receptive Field

- The **receptive field** of a feature is the part of the input that ″influenced″ the feature:



* Credits: Dang Ha The Hien

Receptive Field



Kernel Size (k_1): 3
Padding (p_1, q_1): 1
Stride (s_1): 3

Kernel Size (k_2): 2
Padding (p_2, q_2): 0
Stride (s_2): 1

Receptive Field

Filter sizes

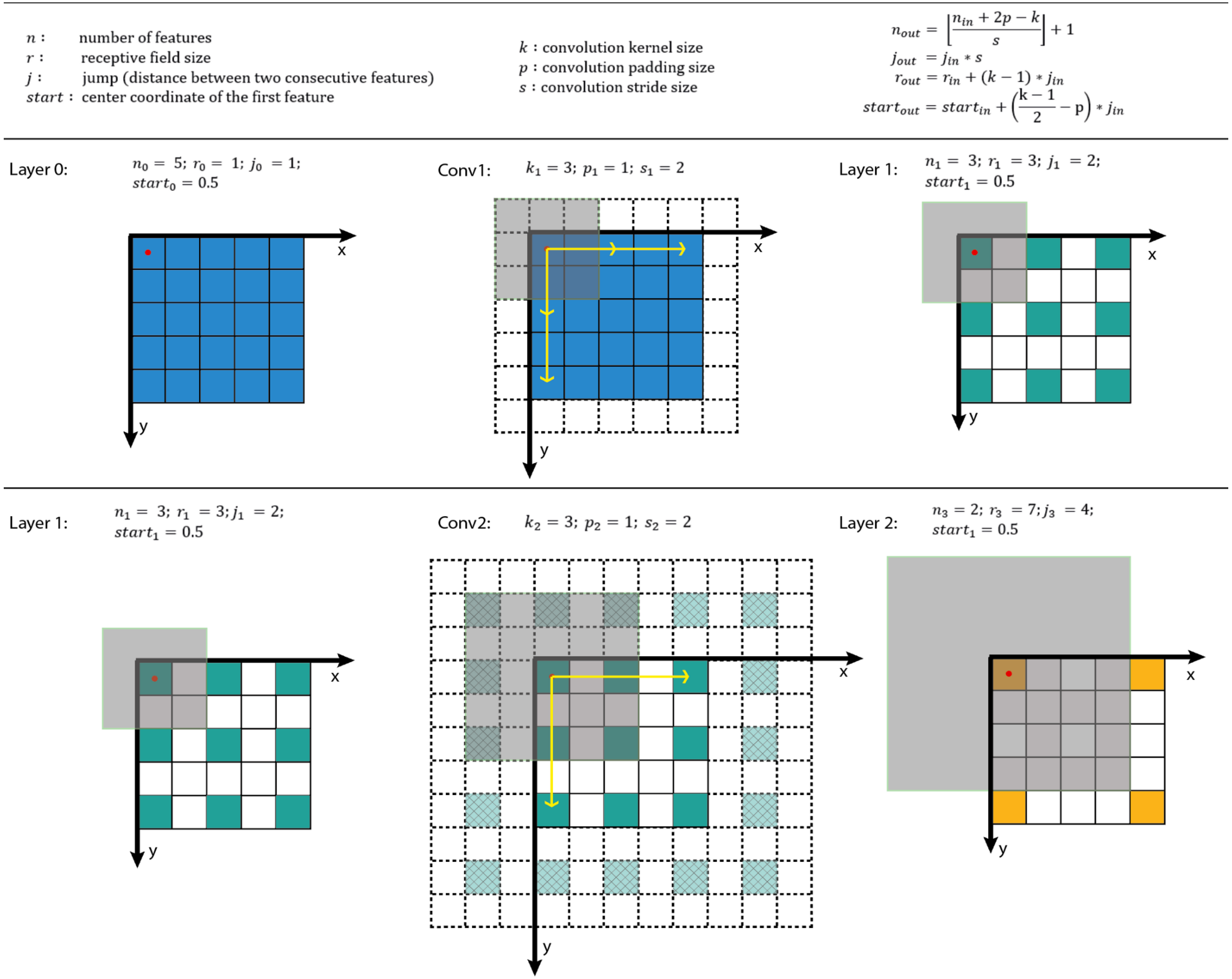
- What is the receptive field of one **5x5 filter**?
  - And how much compute does it take?

- What is the receptive field of **two 3x3 filters**?
  - How much compute does that take?

- Many current architecture use primarily **3x3 filters** for this reason

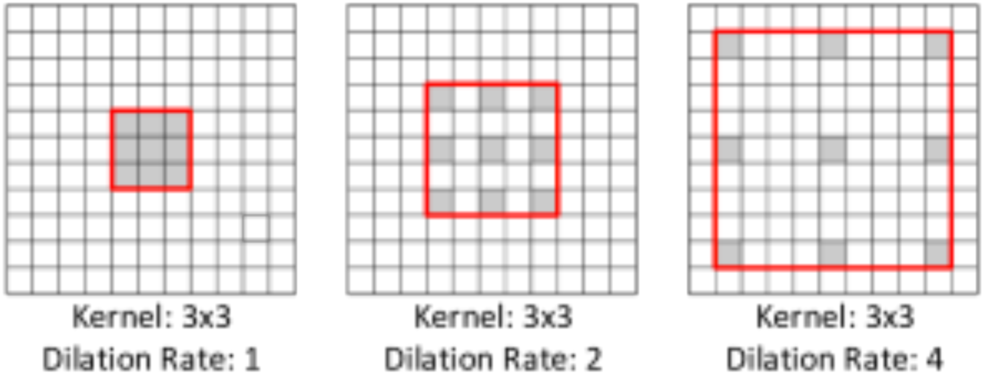
Receptive Field

- The deeper you go, the wider the receptive field



Dilated convolution

- Another way to downsample our input is by using **dilated convolutions** (or *atrous* convolution). The **dilation coefficient** *d* determines pixel spacing between convolutions.



Transposed convolution

- **Transposed convolutions** are used to **increase** the resolution of a **feature map**. It can be thought of as an interpolation, or a convolution on the output. It is often used in the context of **upsampling** the output of a network.

- Convolution as a matrix operation

  For example This linear operation takes the input matrix flattened as a 16-dimensional vector and

  produces a 4-dimensional vector that is later reshaped as the 2 × 2 output matrix.

$$\begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix}$$

- To get the other way around from 4 dimensions to 16 we can use the transposed matrix
- The gradients w.r.t. the inputs can be computed by “Transposed convolution

References

- Dang Ha The Hien, A guide to receptive field arithmetic for Convolutional Neural Networks.
- Vincent Dumoulin, Francesco Visin, A guide to convolution arithmetic for deep learning
- Ivado-mila Deep learning summer school 2019