

Rechnerarchitektur (RA)

Sommersemester 2020

Overview of Deep Neural Networks

Jian-Jia Chen

Informatik 12

Jian-jia.chen@tu-..

<http://ls12-www.cs.tu-dortmund.de/daes/>

Tel.: 0231 755 6078

Neurons, Synapses, Network

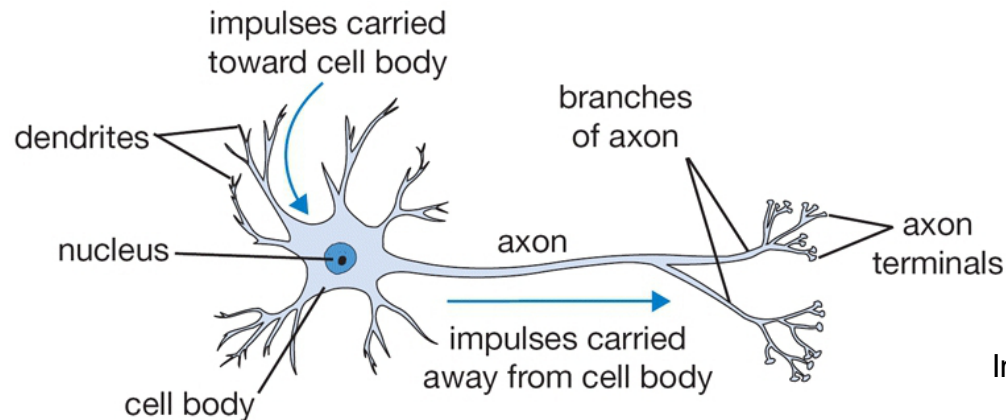


Image Source: Stanford

Functional units and links

- The basic computational unit is a **neuron**
- Neurons are connected with nearly $10^{14} - 10^{15}$ **synapses**

Operations

- Neurons receive input signal from **dendrites** and produce output signal along **axon** which interact with the dendrites of other neurons

Neural Networks

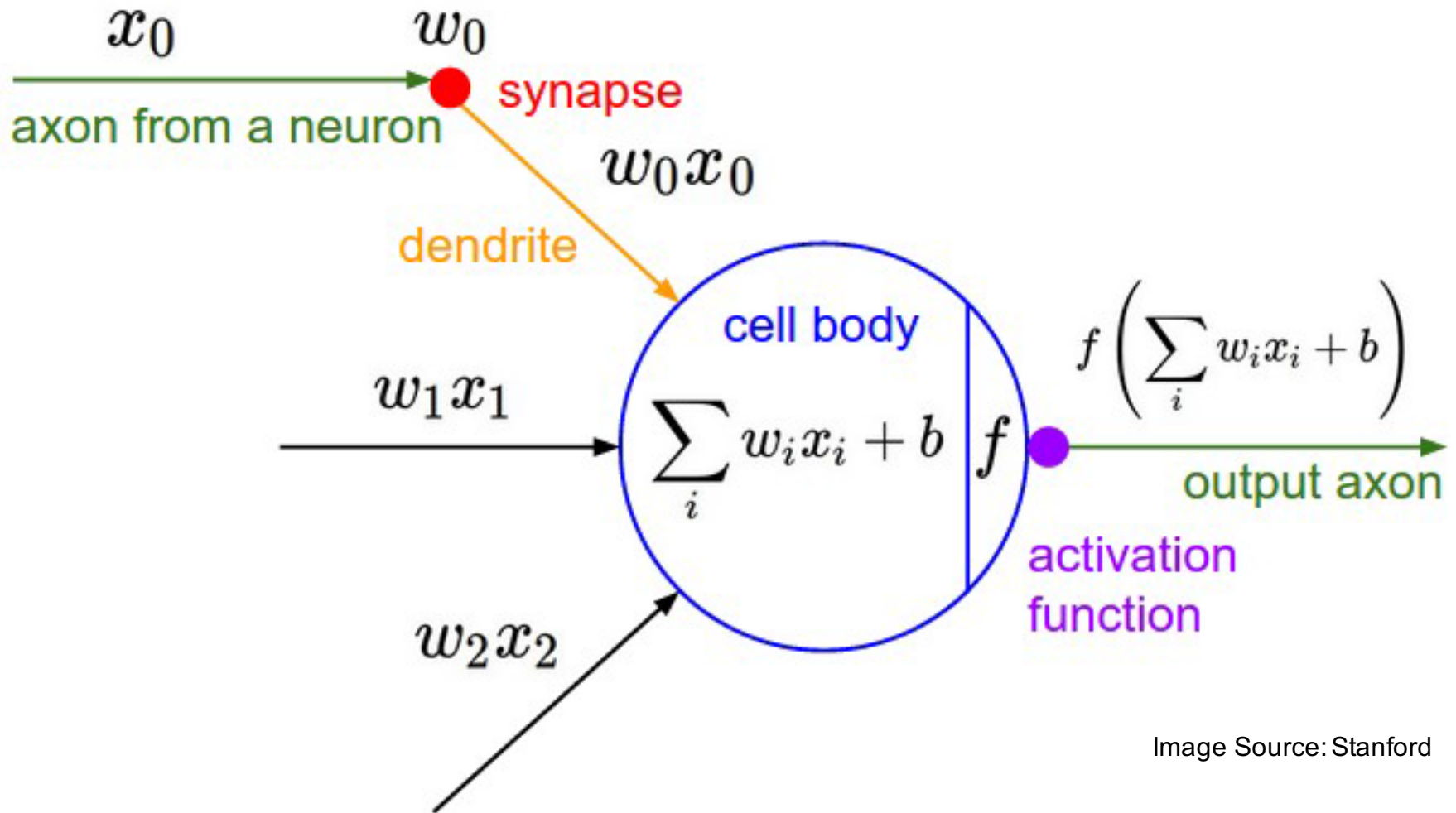
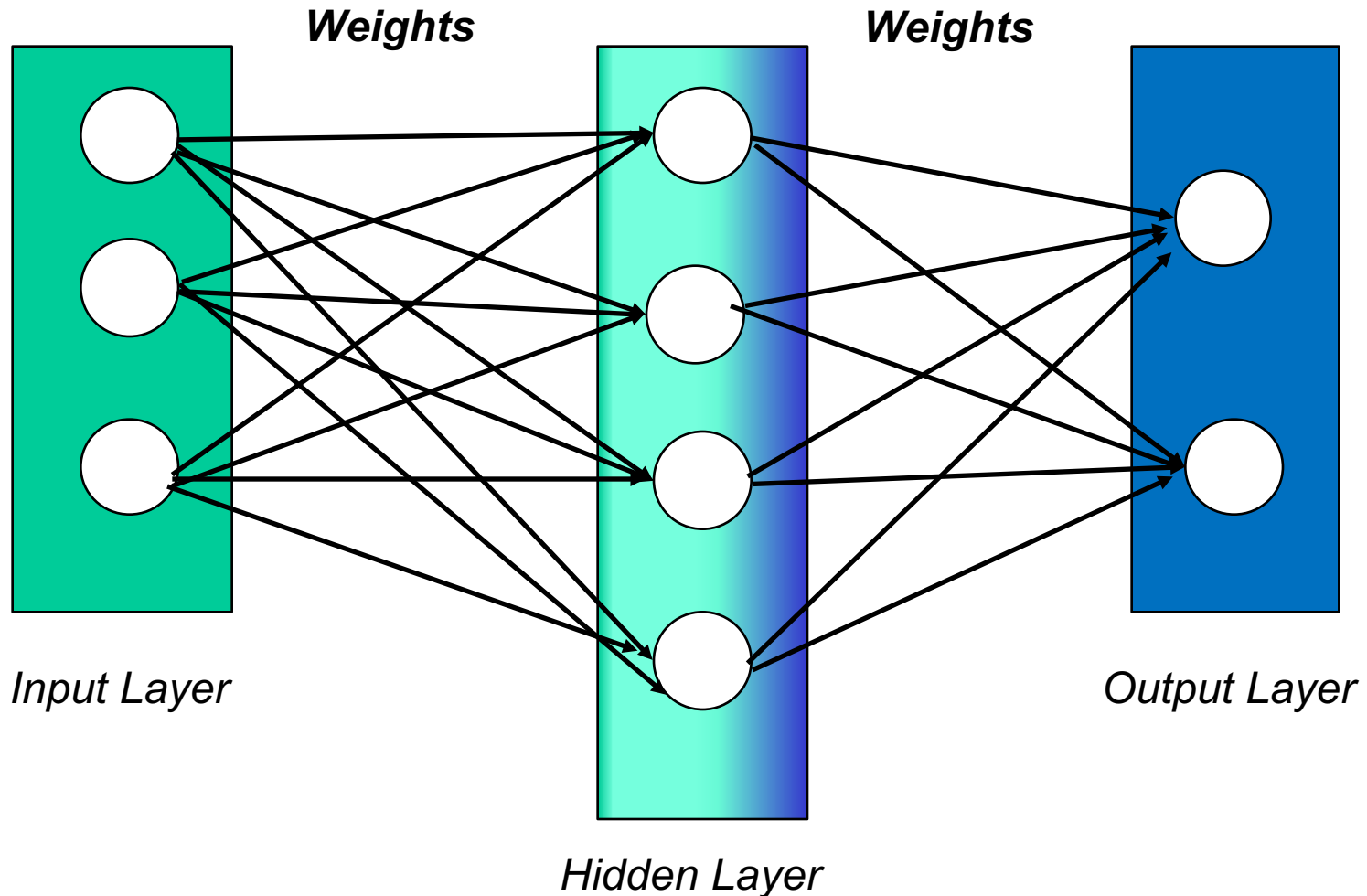
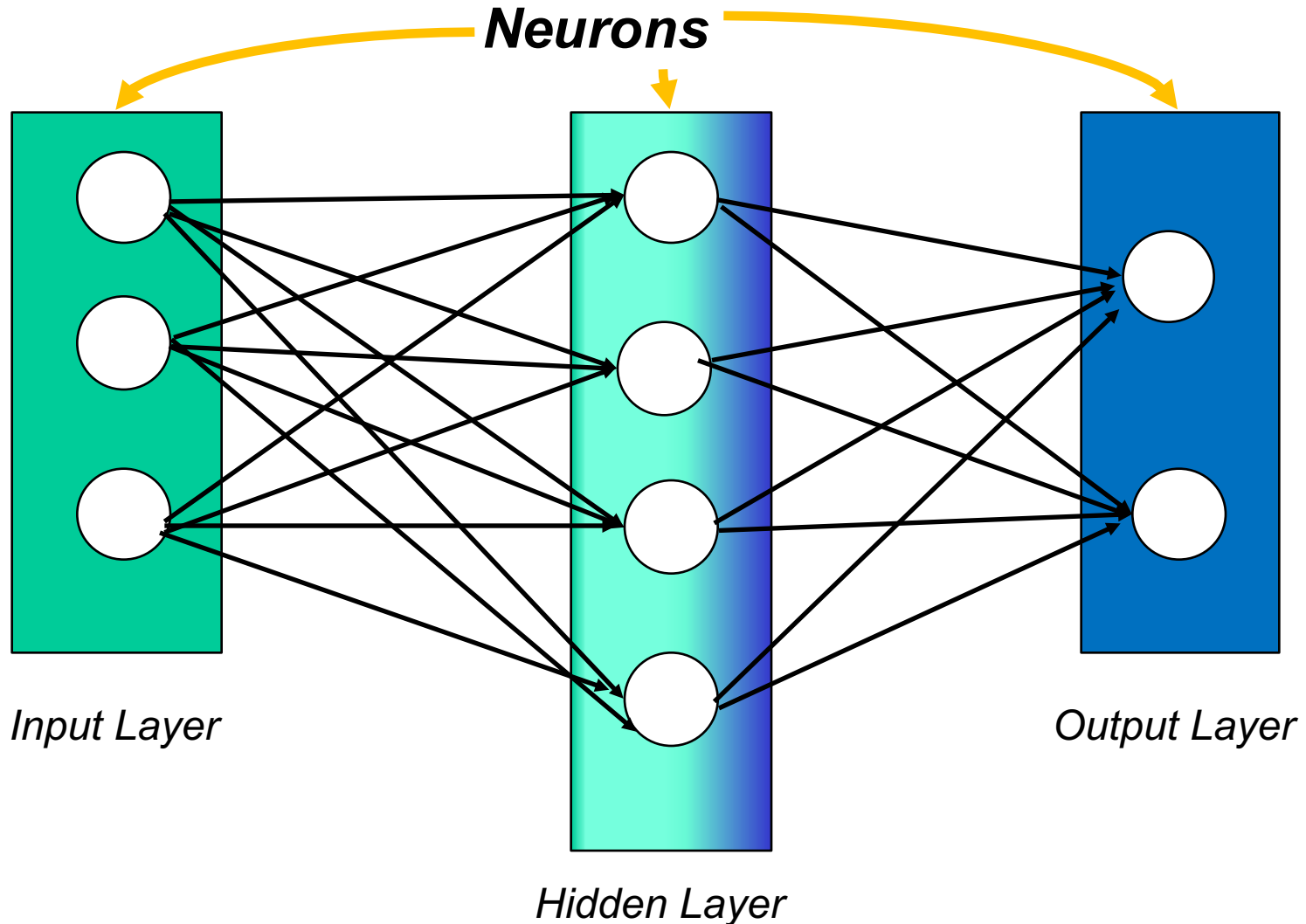


Image Source: Stanford

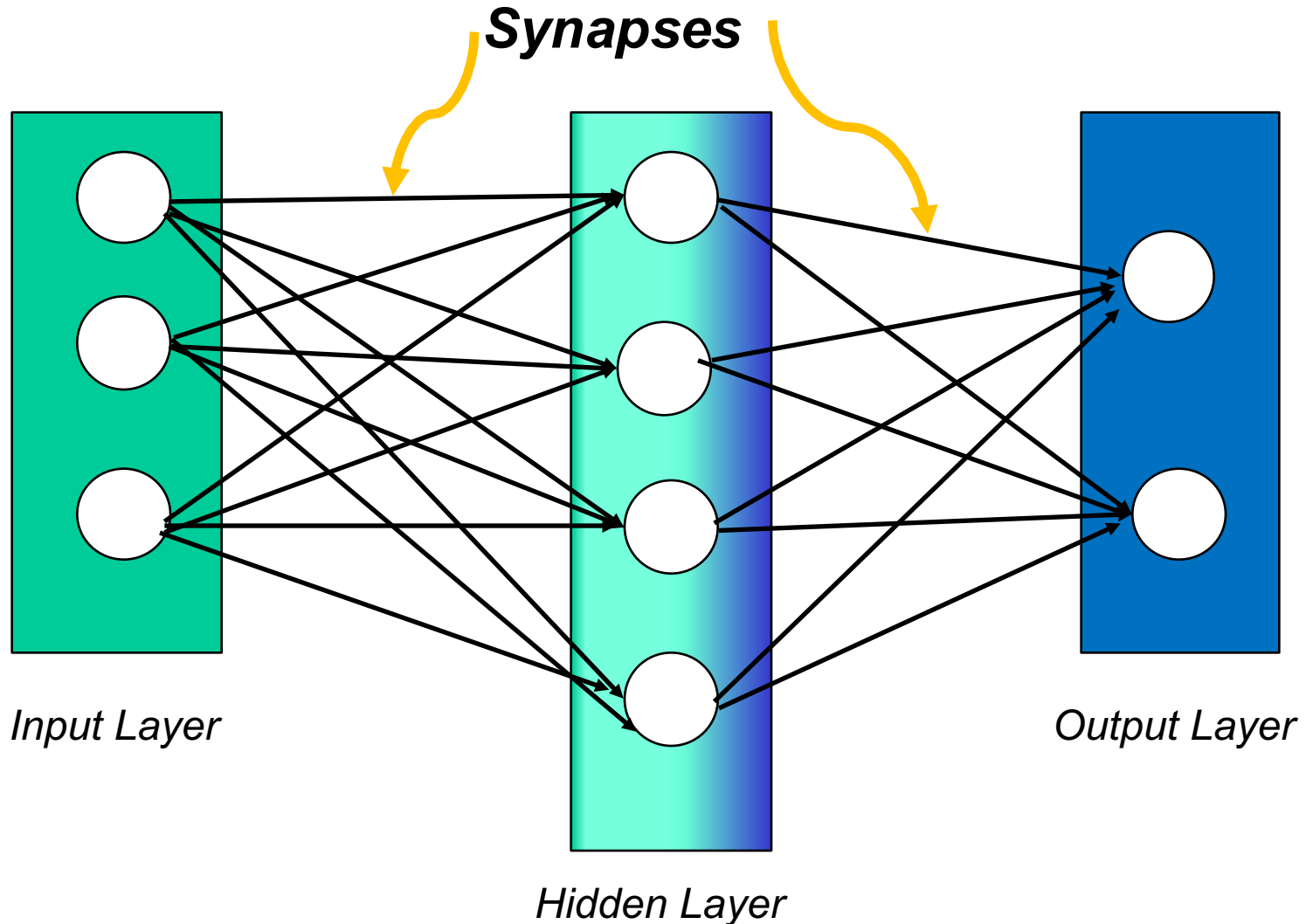
Many Weighted Sums



Many Weighted Sums

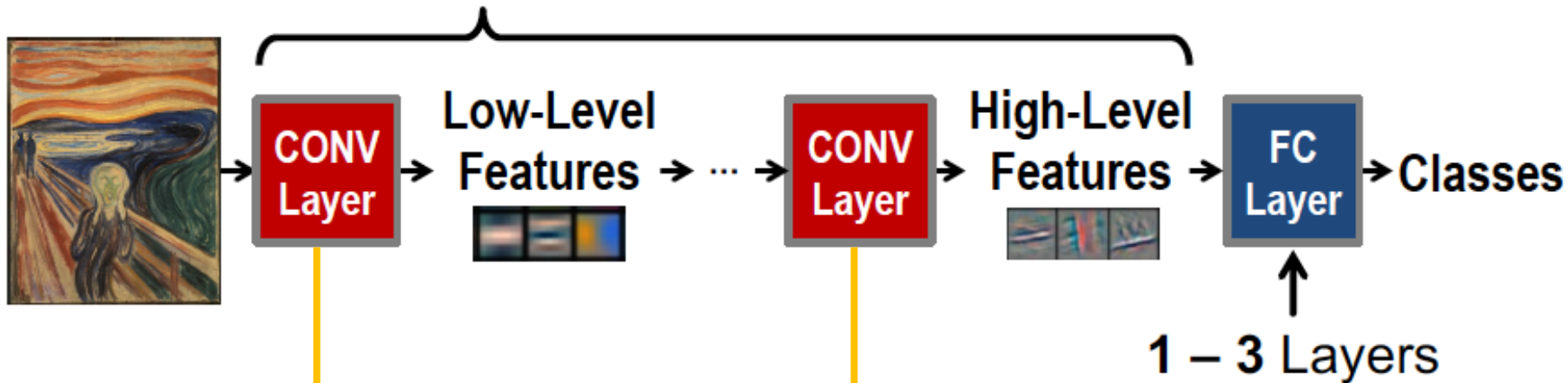


Many Weighted Sums



Deep Convolutional Neural Network

Modern **Deep CNN**: 5 – 1000 Layers



The diagram shows a grayscale version of 'The Scream' with three overlapping colored rectangles (red, green, and blue) representing convolution kernels. Lines connect these kernels to a corresponding grid of gray squares, illustrating how the convolution operation extracts local features from the input image.

Convolutions account for more than 90% of overall computation, dominating **runtime** and **energy consumption**

Image Source: Emer et al. ISCA Tutorial 2019

Convolution Layer

Filter



Input Feature
Map (fmap)

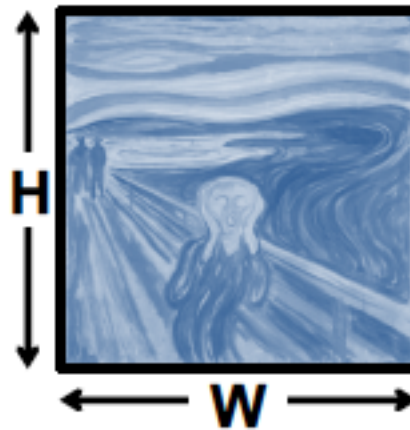


Image Source: Emer et al. ISCA Tutorial 2019

Convolution Layer

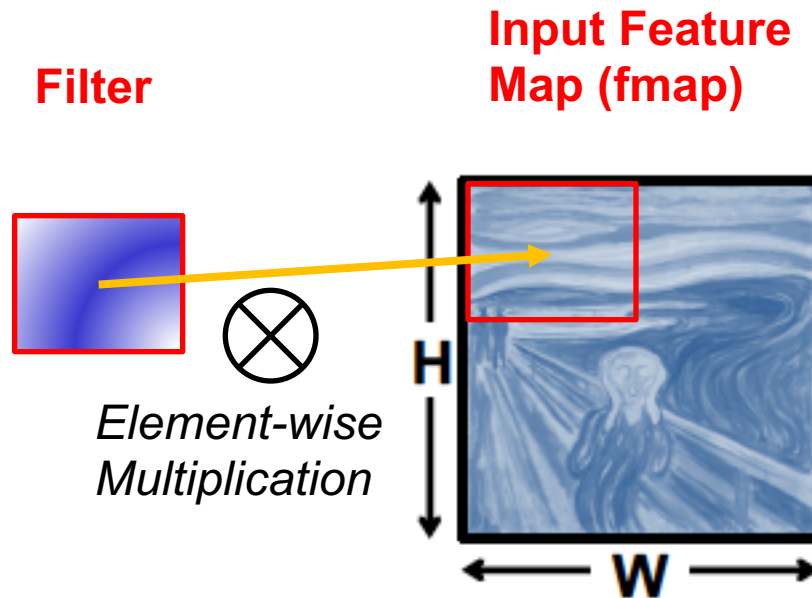


Image Source: Emer et al. ISCA Tutorial 2019

Convolution Layer

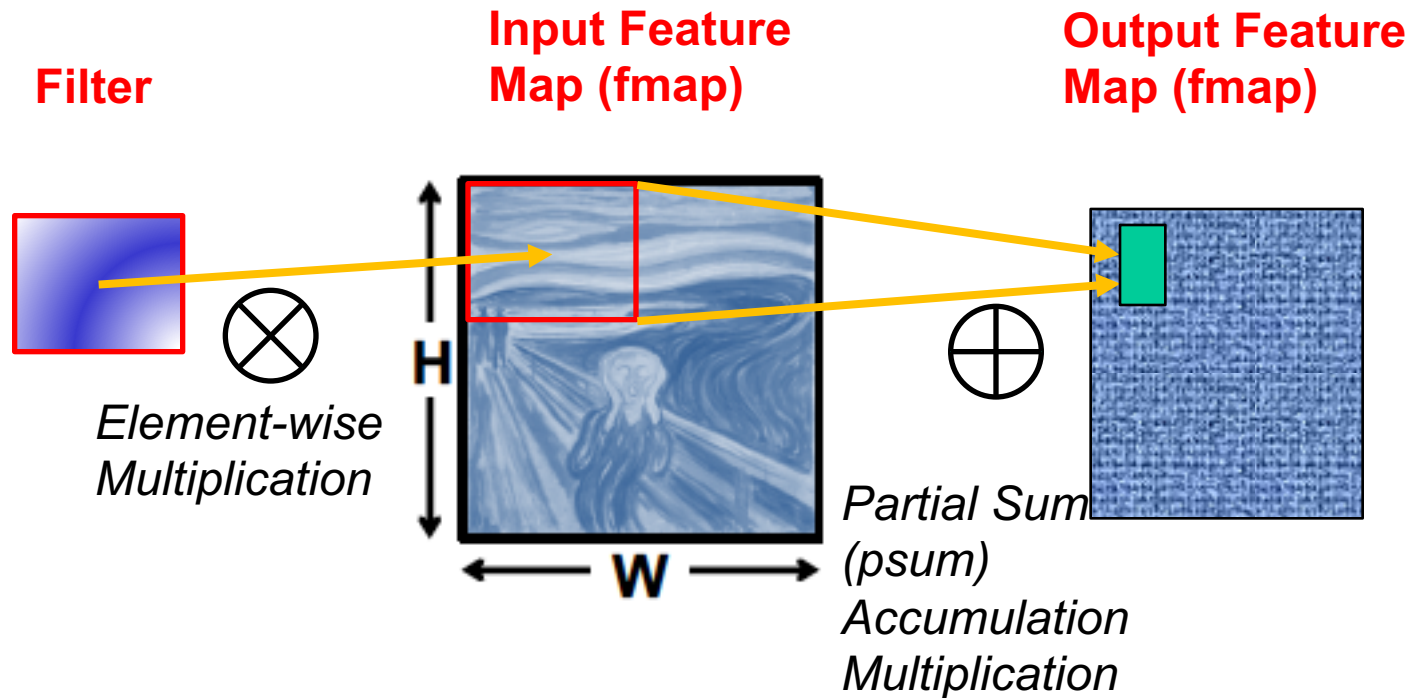


Image Source: Emer et al. ISCA Tutorial 2019

Convolution Layer Operations

Output fmaps (O)

Input fmaps (I)

Biases (B)

Filter weights (W)

$$\underline{O[n][m][x][y]} = \text{Activation}(\underline{B[m]} + \sum_{i=0}^{R-1} \sum_{j=0}^{S-1} \sum_{k=0}^{C-1} \underline{I[n][k][Ux+i][Uy+j]} \times \underline{W[m][k][i][j]}),$$

$$0 \leq n < N, 0 \leq m < M, 0 \leq y < E, 0 \leq x < F,$$

$$E = (H - R + U)/U, F = (W - S + U)/U.$$

Shape Parameter	Description
N	fmap batch size
M	# of filters / # of output fmap channels
C	# of input fmap/filter channels
H/W	input fmap height/width
R/S	filter height/width
E/F	output fmap height/width
U	convolution stride

Source: Emer et al. ISCA Tutorial 2019

A Naïve Implementation

```
for (n=0; n<N; n++) {  
  for (m=0; m<M; m++) {  
    for (x=0; x<F; x++) {  
      for (y=0; y<E; y++) {
```

} for each output fmap value

convolve
a window
and apply
activation

```
    O[n][m][x][y] = B[m];  
    for (i=0; i<R; i++) {  
      for (j=0; j<S; j++) {  
        for (k=0; k<C; k++) {  
          O[n][m][x][y] += I[n][k][Ux+i][Uy+j] × W[m][k][i][j];  
        }  
      }  
    }  
    O[n][m][x][y] = Activation(O[n][m][x][y]);
```

```
  }  
}
```

Source: Emer et al. ISCA Tutorial 2019