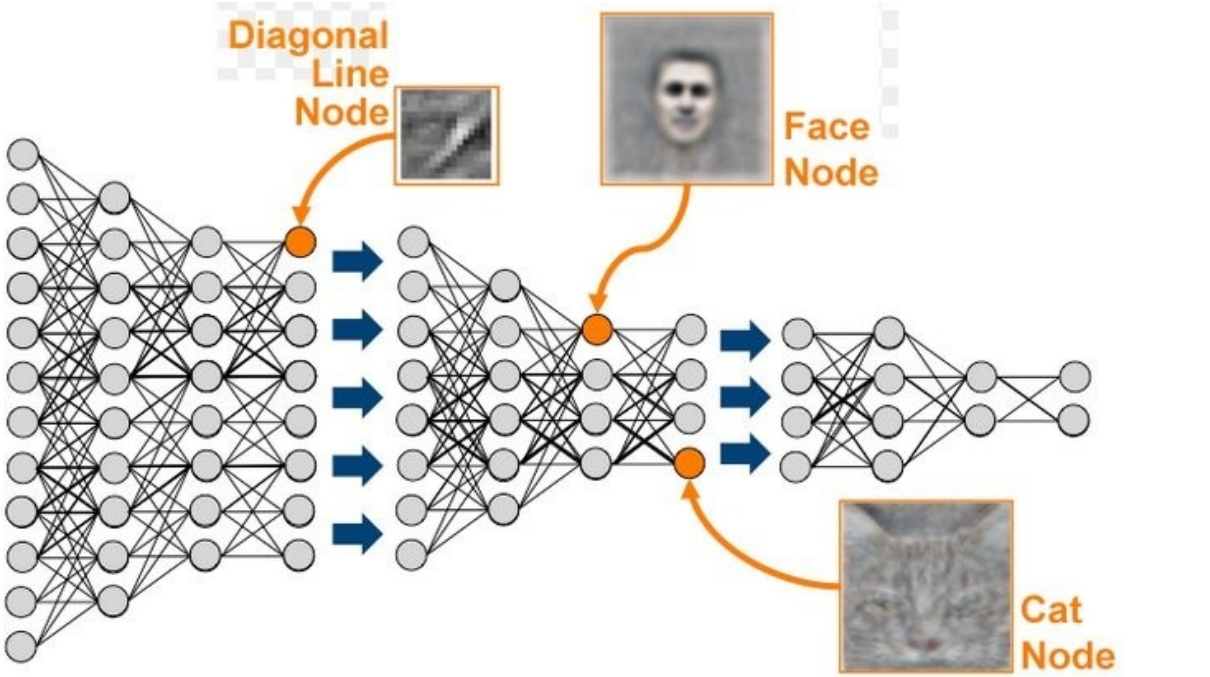
Universal approximation theorem

The “universal approximation theorem” (Cybenko, 1989, Hornik, 1991) proves, that already with a single hidden layer, we can essentially learn every continuous function (with a few mild assumptions).

So why introduce more than one hidden layer?

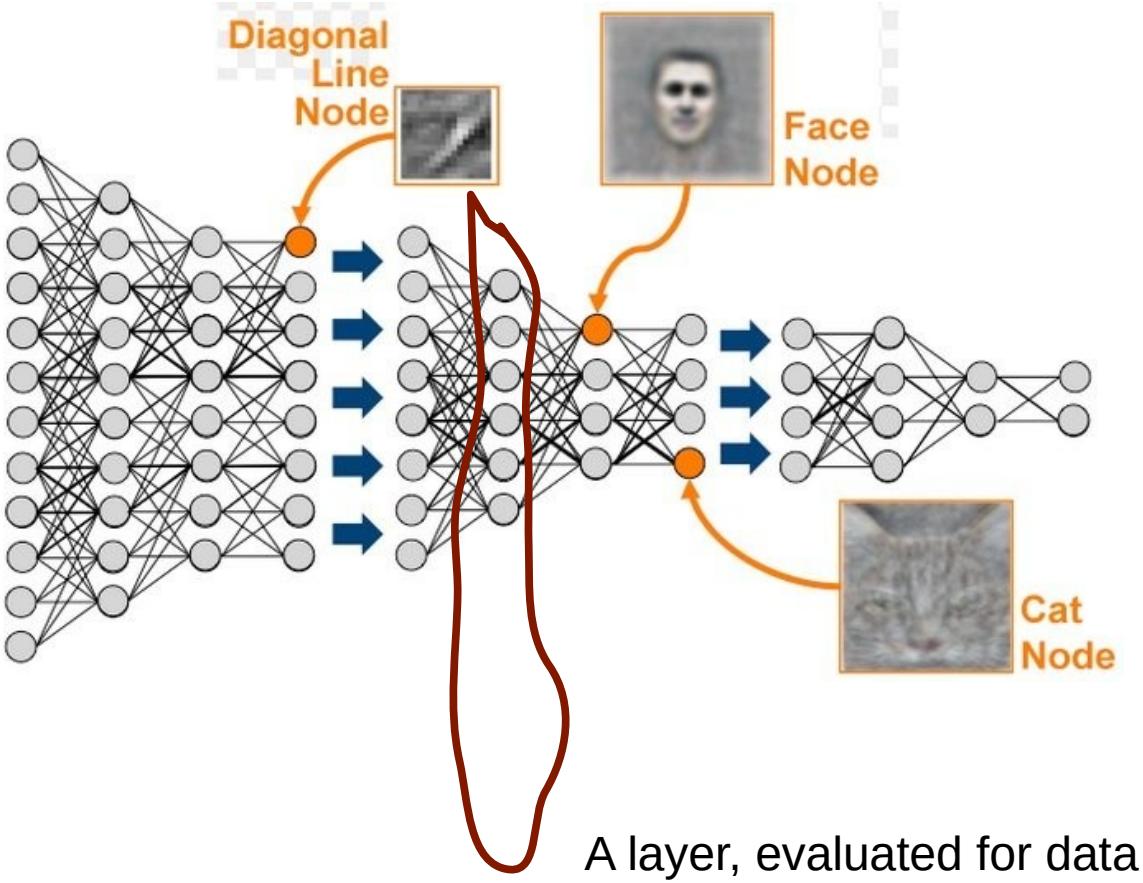
Deep learning = learning abstractions

Because when done right, a deep network can learn “abstract features” of the input data:



Deep learning = hierarchical representation learning

Because when done right, a deep network can learn “abstract features” of the input data:

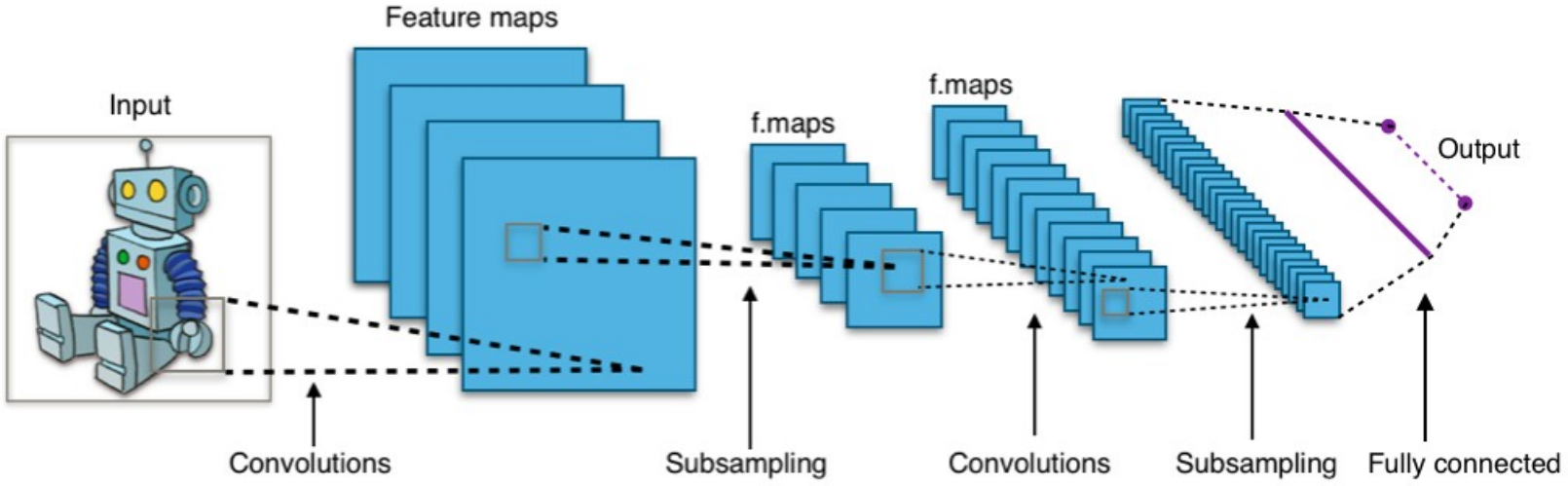


A layer, evaluated for data = a “representation” of the data

Convolutional layers

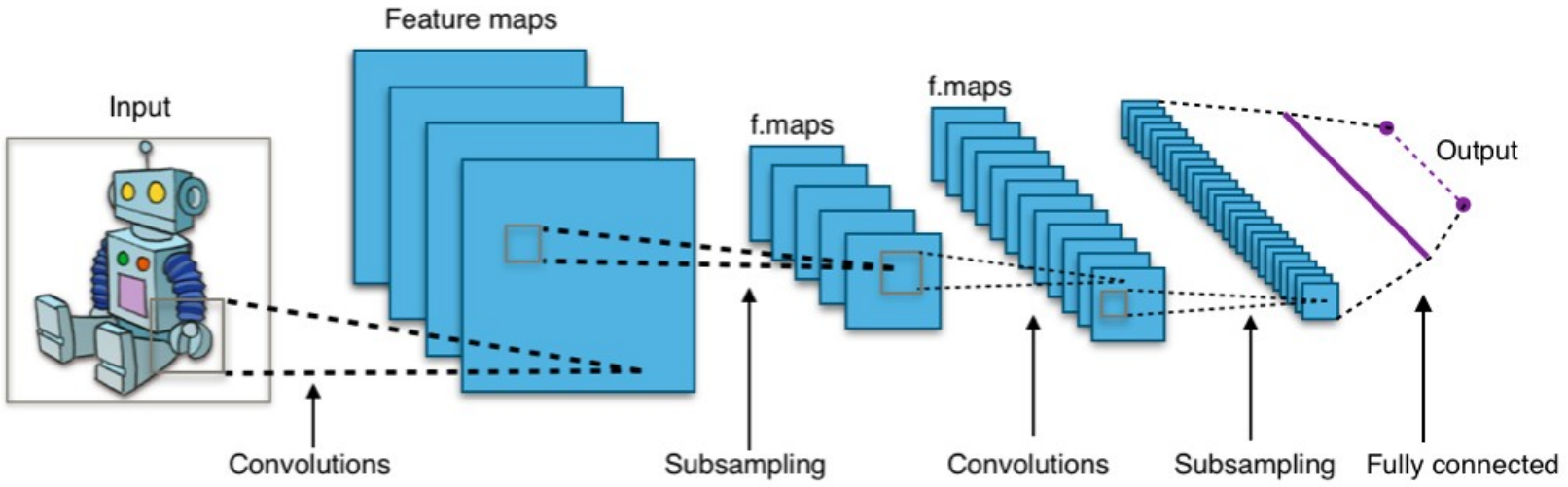
Think computer vision: we want to detect a cat,

independent of where she is in the photo: translation invariance!
Answer: convolutional layers: many several small convolution matrices (“filters”, “maps”, “kernels”)



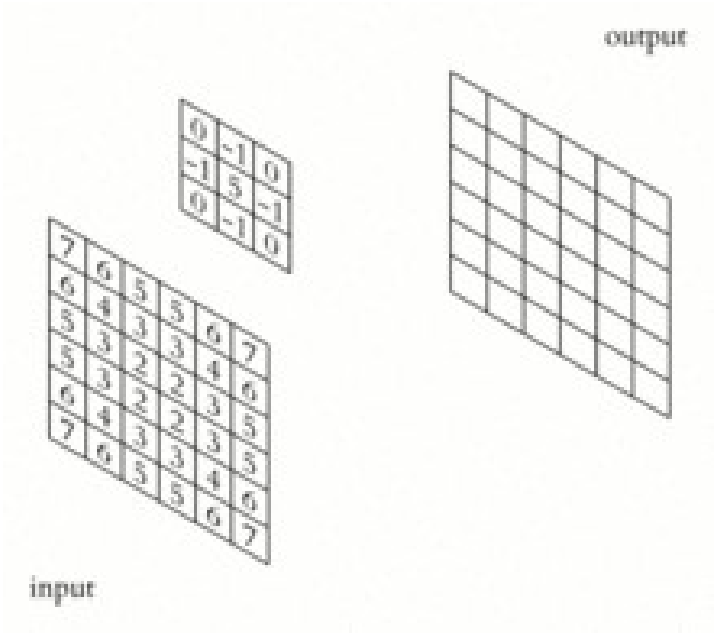
Convolutional layers

Answer: convolutional layers: many several small convolution matrices (“kernels”) that “scan” over the input image, “sharing” the learned weights and thus detect features independent of their position

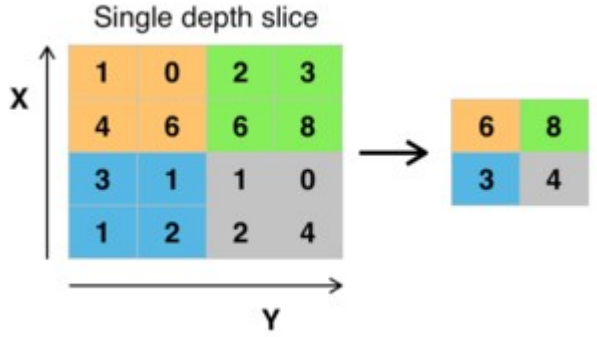


Convolutional layers

Answer: convolutional layers: many several small convolution matrices (“kernels”) that “scan” over the input image, “sharing” the learned weights and thus detect features independent of their position



Convolutions



Max Pooling (subsampling)

Inspired by the “receptive field” in biology