Convolutional Neural Networks (ConvNet)

1. Recap from last week (chapters 5.1 & 5.2)
2. Using a pretrained ConvNets (chapter 5.3)
   - Feature extraction
   - Fine-tuning a pretrained convnet
3. Visualizing what ConvNets learn (chapter 5.4)
   - Visualizing activations
   - Visualizing filters
   - Visualizing heatmaps of class activation

ConvNet

- class of deep Artificial Neural Networks (ANN)
- mainly applied for analyzing visual imagery.
- common applications

classification

Cat

object detection

Dog, Dog, Cat

segmentation

Dog, Dog, Cat

video processing, natural language processing, speech recognition

ConvNet architecture

```
model <- keras_model_sequential() %>%
  layer_conv_2d(filters = 32, kernel_size = c(3, 3), activation = "relu",
                input_shape = c(150, 150, 3)) %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 64, kernel_size = c(3, 3), activation = "relu") %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 128, kernel_size = c(3, 3), activation = "relu") %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 128, kernel_size = c(3, 3), activation = "relu") %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_flatten() %>%
  layer_dropout(rate = 0.5) %>%
  layer_dense(units = 512, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")

model %>% compile(
  loss = "binary_crossentropy",
  optimizer = optimizer_rmsprop(lr = 1e-4),
  metrics = c("acc")
)
```

Convolution process

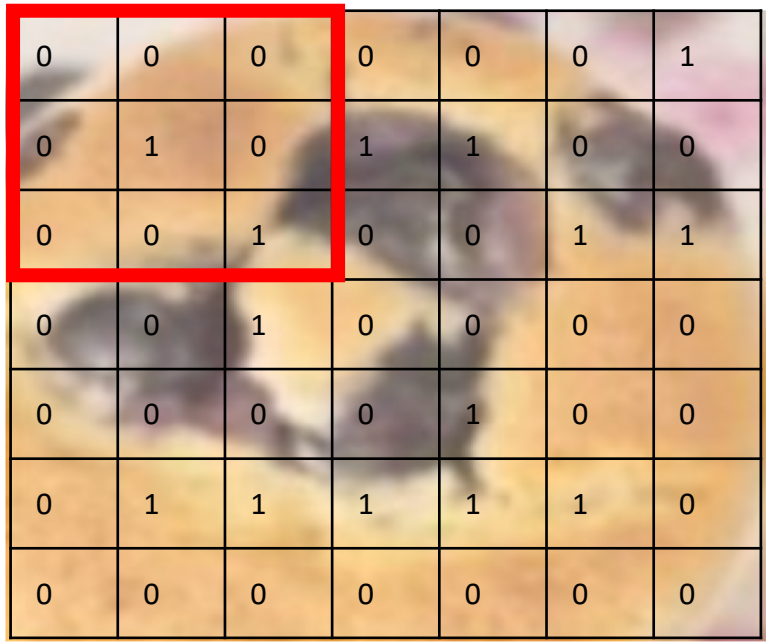
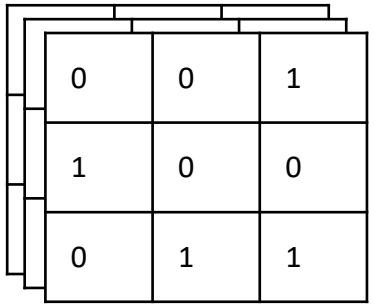


Image input

X

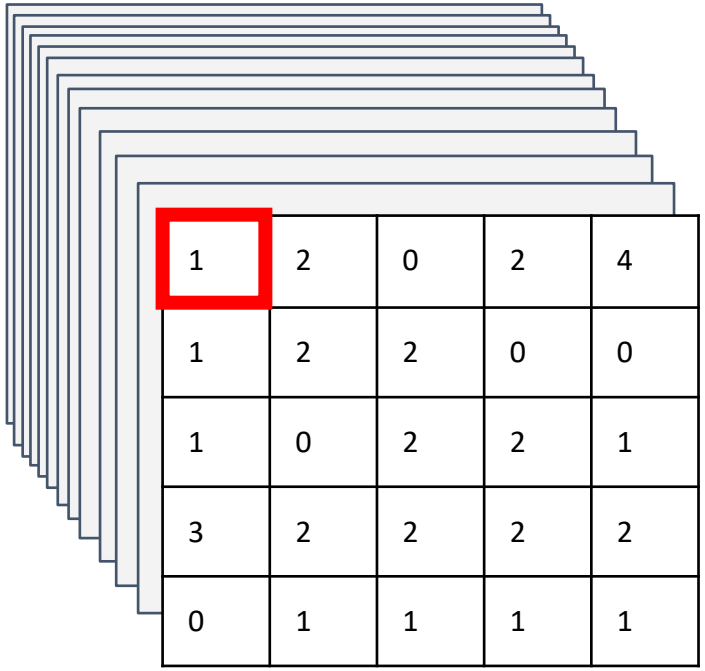
Random learnable **weights**



Kernel (3X3)
Stride (1,1)

=

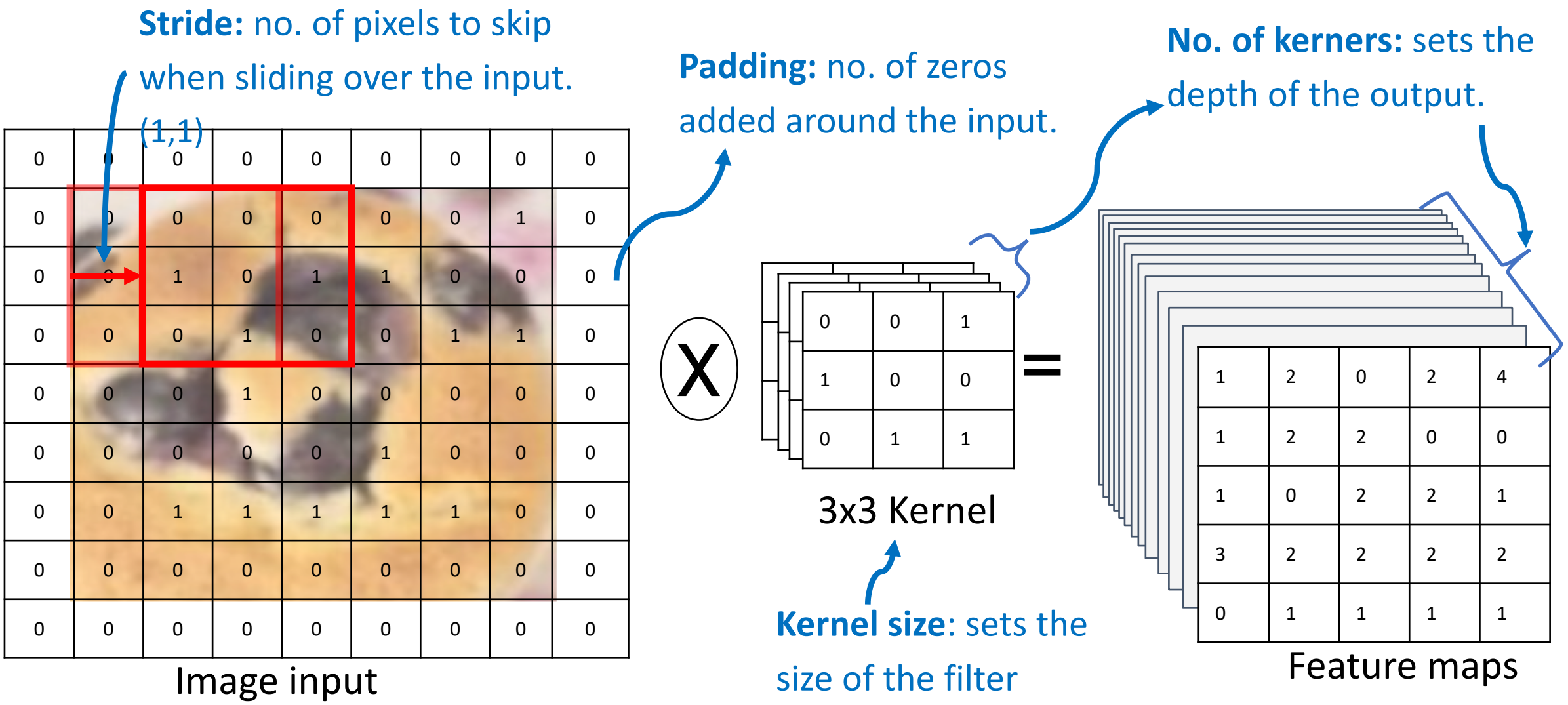
dot-product of image pixels with a kernel weights



Feature maps

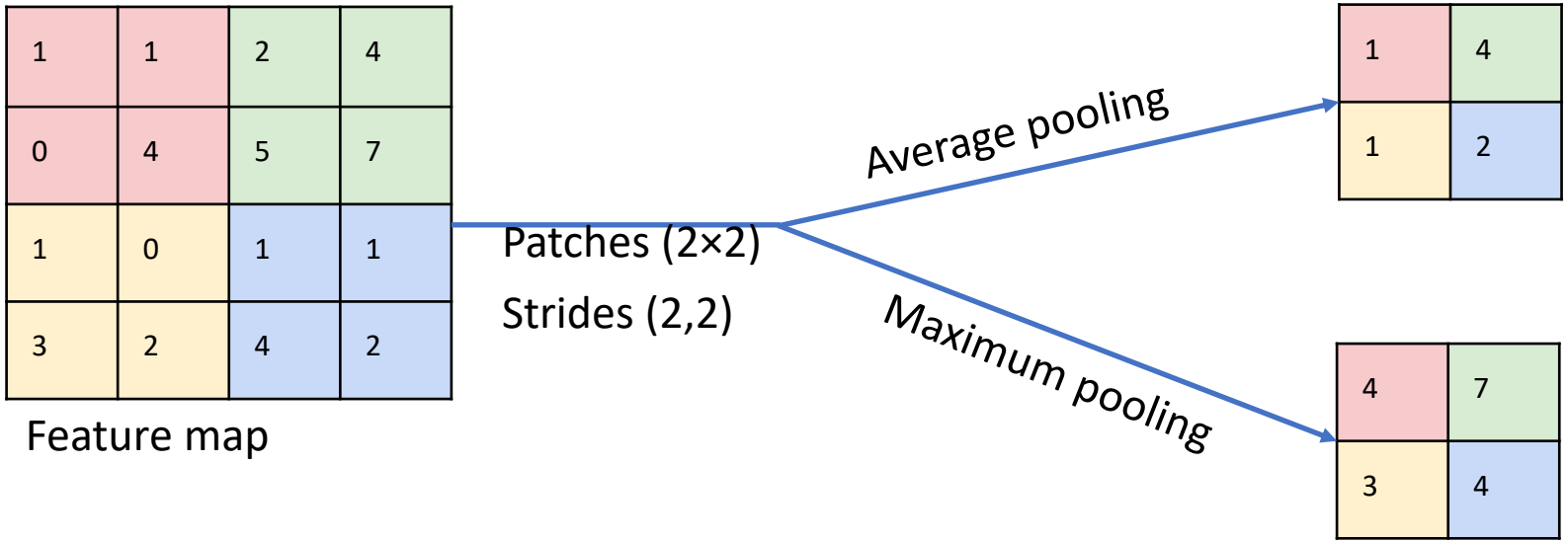
Hyperparameters

- The size of the feature map can be controlled by the following hyper-parameters:



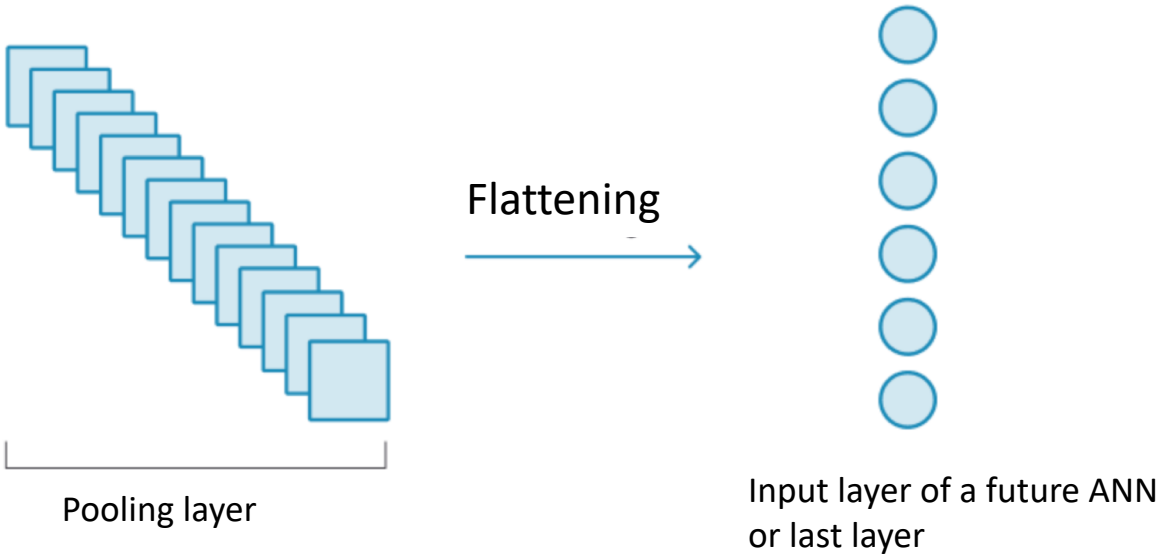
Pooling process

- Reduces the dimensionality of feature maps but retains important information.
  - Focuses on the main features
  - Reduce computational costs - gets rid of extra data
  - Reduce overfitting - combines neighboring values into a single one
- Types of pooling
  - **Average Pooling:** average value for each patch on the feature map.
  - **Maximum Pooling:** maximum value for each patch of the feature map.



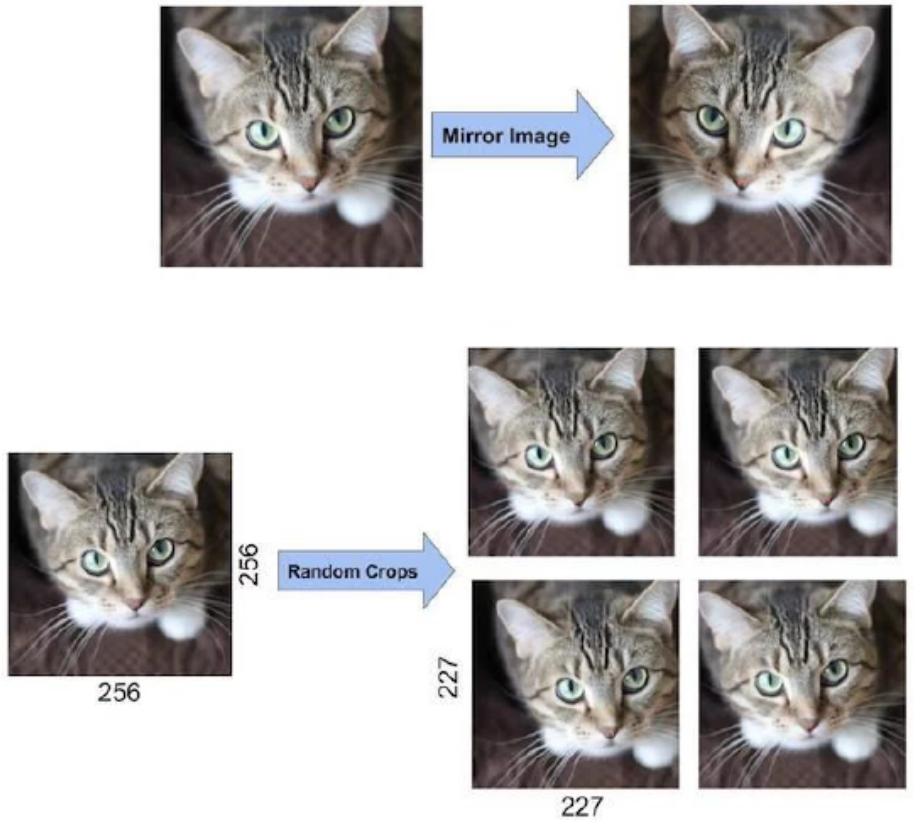
Dense Layer

- Facilitated by a process called Flattening.
- Flattening transforms the feature maps into **one** vector
- The vector fed into a neural network of ***n*** layers.
- Last layer has ***c*** neurons, representing the **number of classes** to classify.

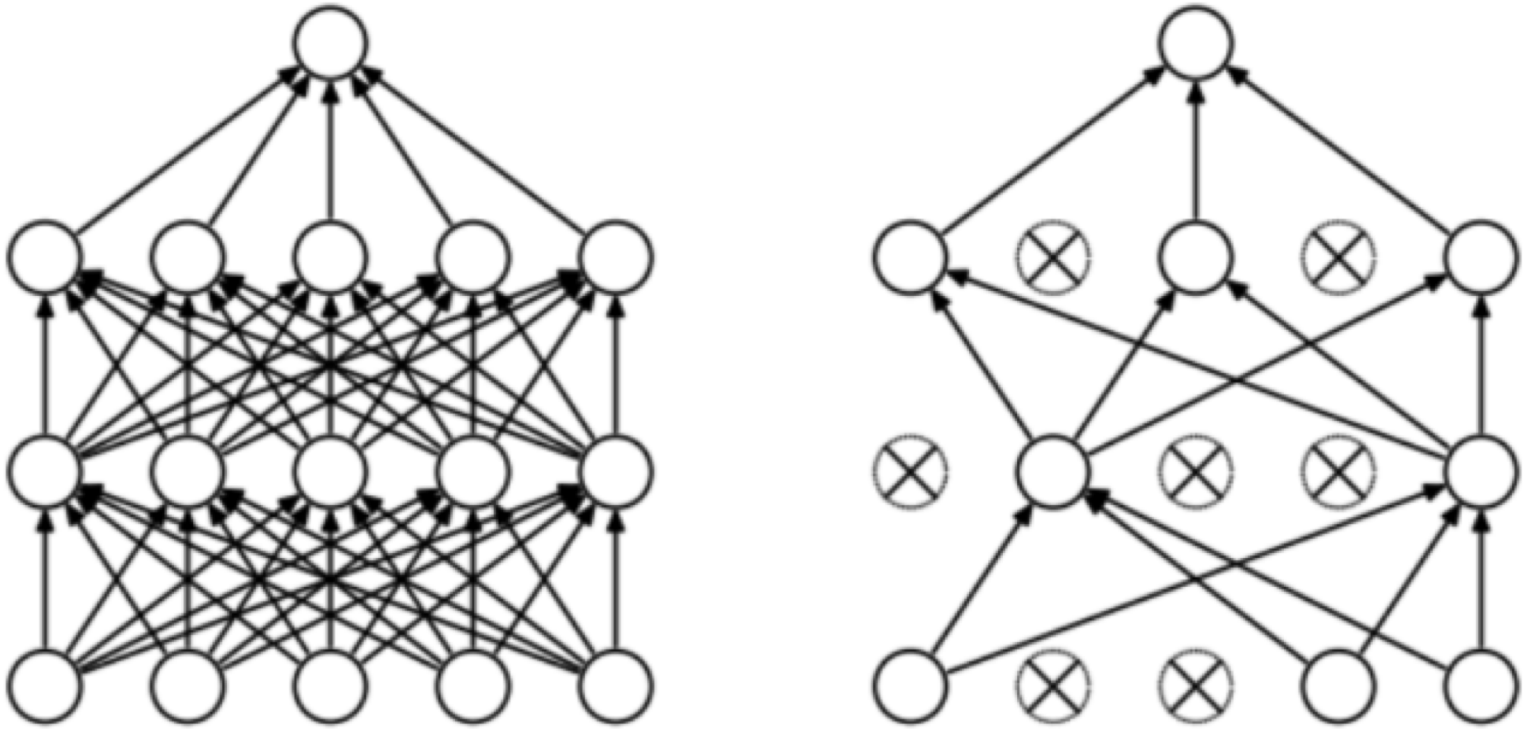


Data augmentation

- Data augmentation is used as a way to make models more robust.
- It reduces overfitting by letting models learn harder and more general data representations.
- Most popular:
  - Noise addition
  - Mirroring
  - Rotating
  - Resizing
  - Cropping
  - Occlusion
- Newer methods:
  - Synthetic data



Drop out



(a) Standard Neural Net

(b) After applying dropout.