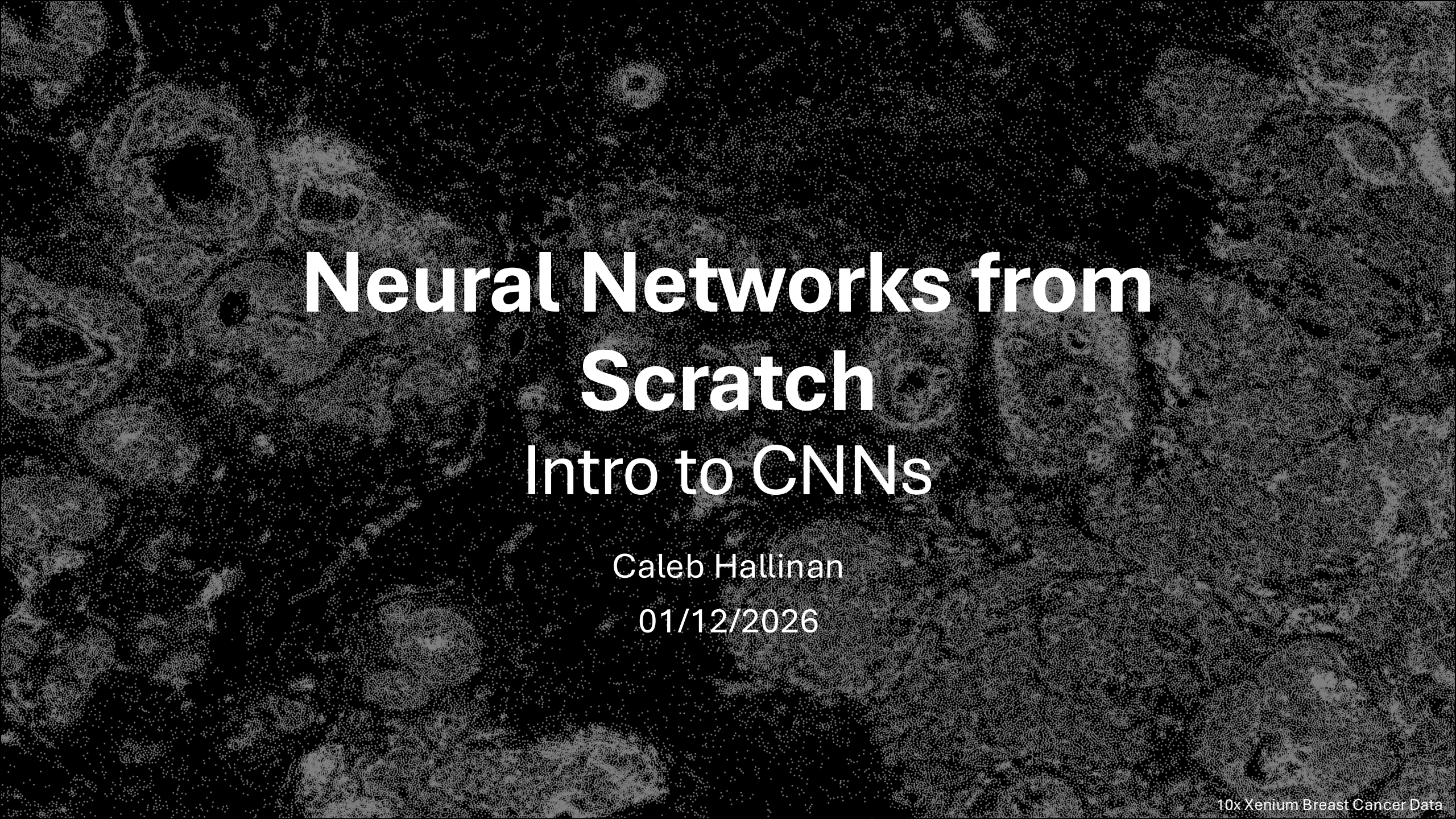
Neural Networks from Scratch

Intro to CNNs

Caleb Hallinan

01/12/2026



10x Xenium Breast Cancer Data

Announcements

- Lecture from Friday is on GitHub
- Today’s code will be from scratch, so nothing is uploaded

- Does anyone know a location we could do in person at?

A few questions from reflection cards

How to choose number of layers and nodes

- No universal rule, but some general guidelines:
  - More data you can use more parameters
  - Use loss plots to help decide:
    - If training loss is too high (low accuracy) then you are probably underfitting
      - So increase layers and nodes
    - If training loss is low but validation loss is high, then you are probably overfitting
      - So decrease layers and nodes
  - Experimentation
    - Try and see what works best
  - “the optimal size of the hidden layer is usually between the size of the input and size of the output layers” –Jeff Heaton https://www.heatonresearch.com/

Saving and Loading models in PyTorch

```
7]
```

```python
# read in packages
import torch
import torchvision.models as models

# To load model weights, you need to create an instance of the same model first, and then load the parameters using load_state_dict() method.
model = models.vgg16(weights='IMAGENET1K_V1')
torch.save(model.state_dict(), 'model_weights.pth')
```

```python
# now we read in the weights
model = models.vgg16() # we do not specify ``weights``, i.e. create untrained model
model.load_state_dict(torch.load('model_weights.pth', weights_only=True))
model.eval()
```

```
<All keys matched successfully>
```

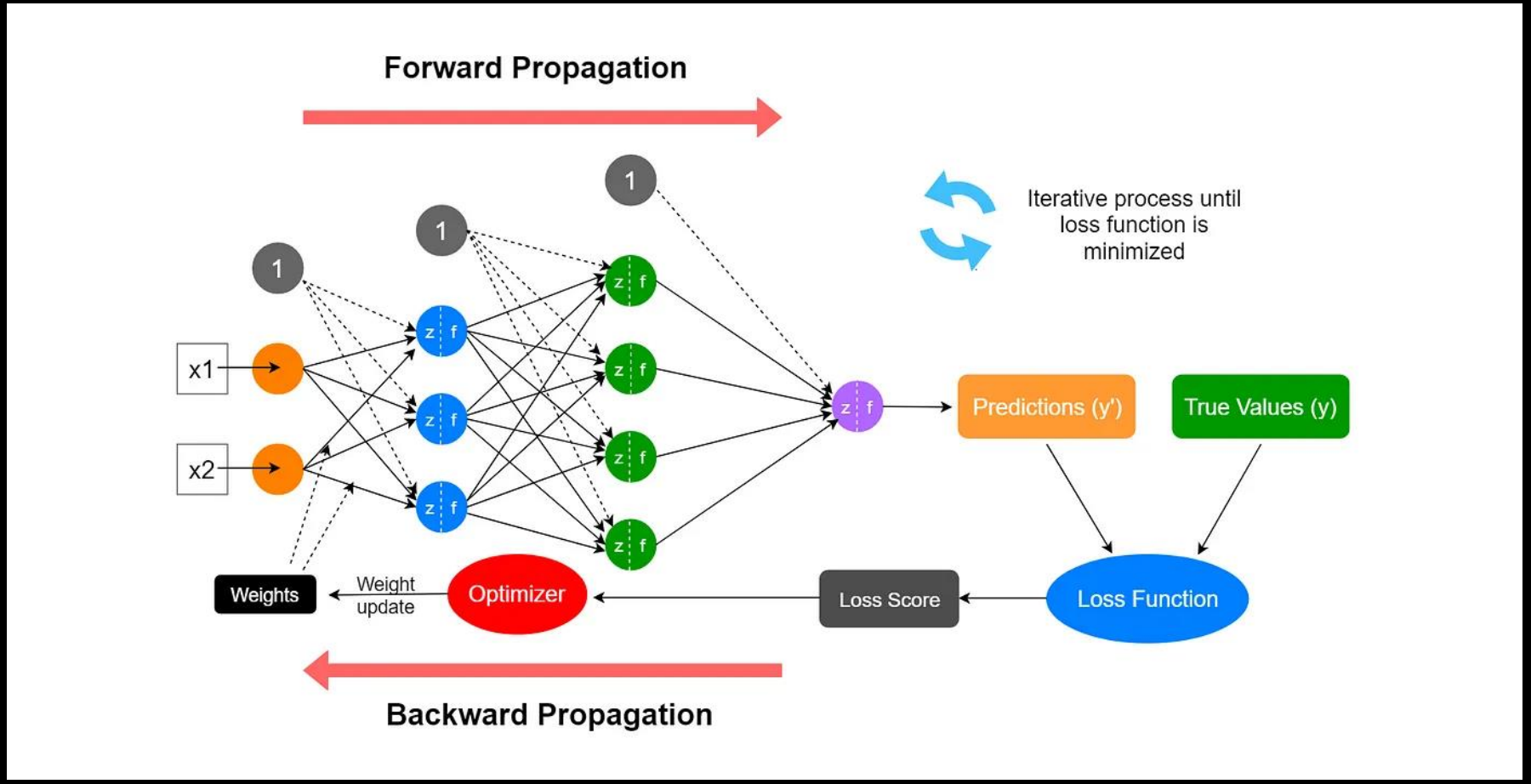
Today’s class

- Common additions you will see in neural networks

- Intro to Convolutional Neural Networks
  - https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175

- Coding CNNs

Full Neural Network process

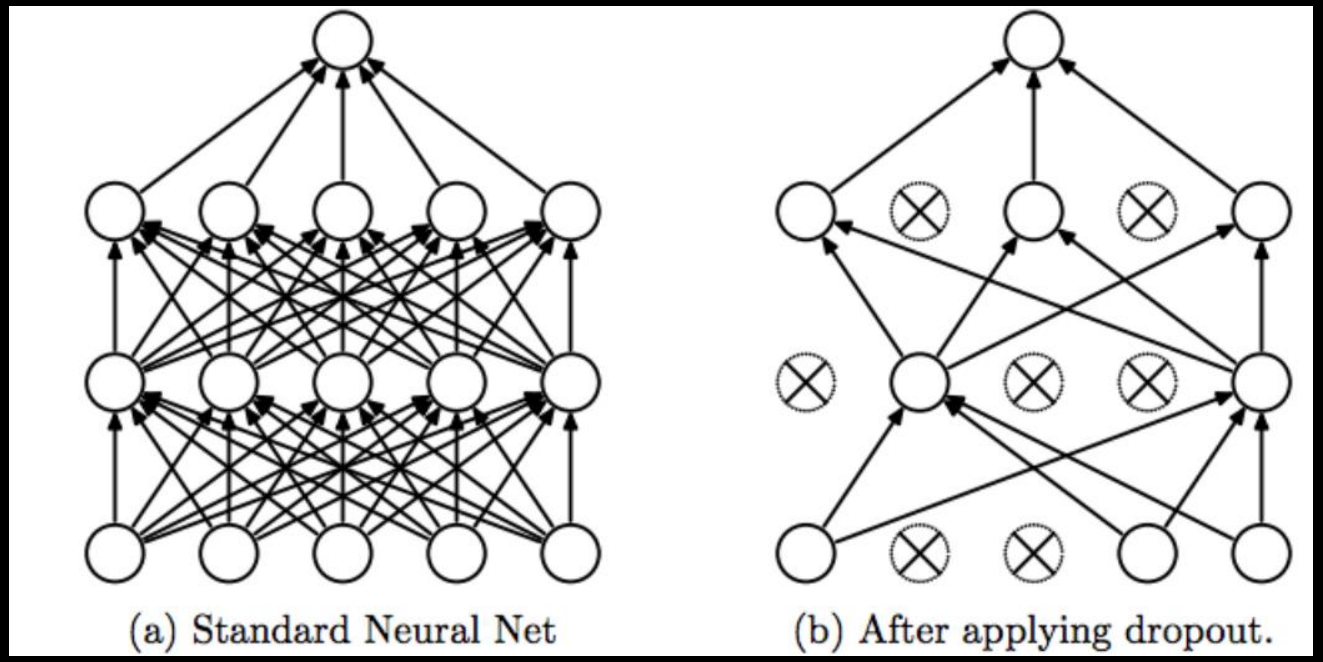


Any questions?

Popular additions in Neural Networks

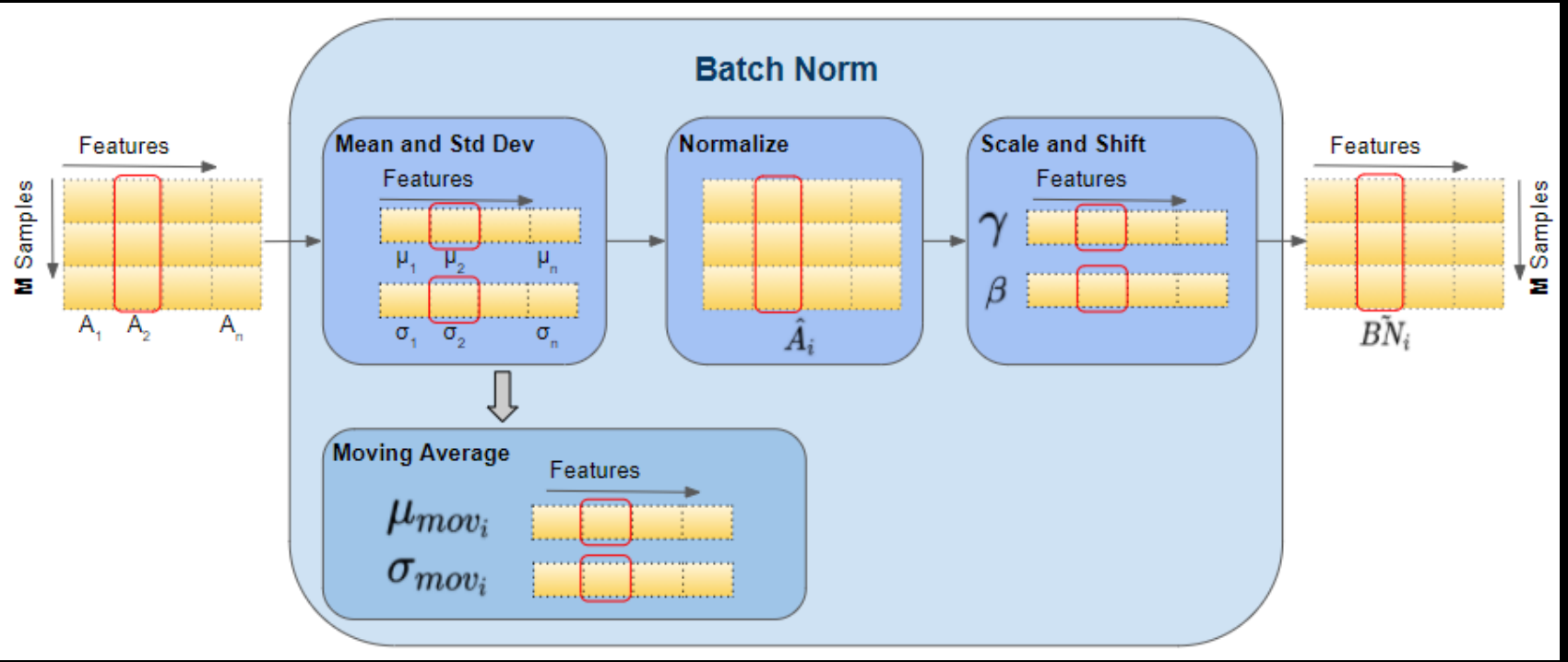
Dropout

- Randomly setting some nodes to output zero during the training process
- What is the benefit of this?



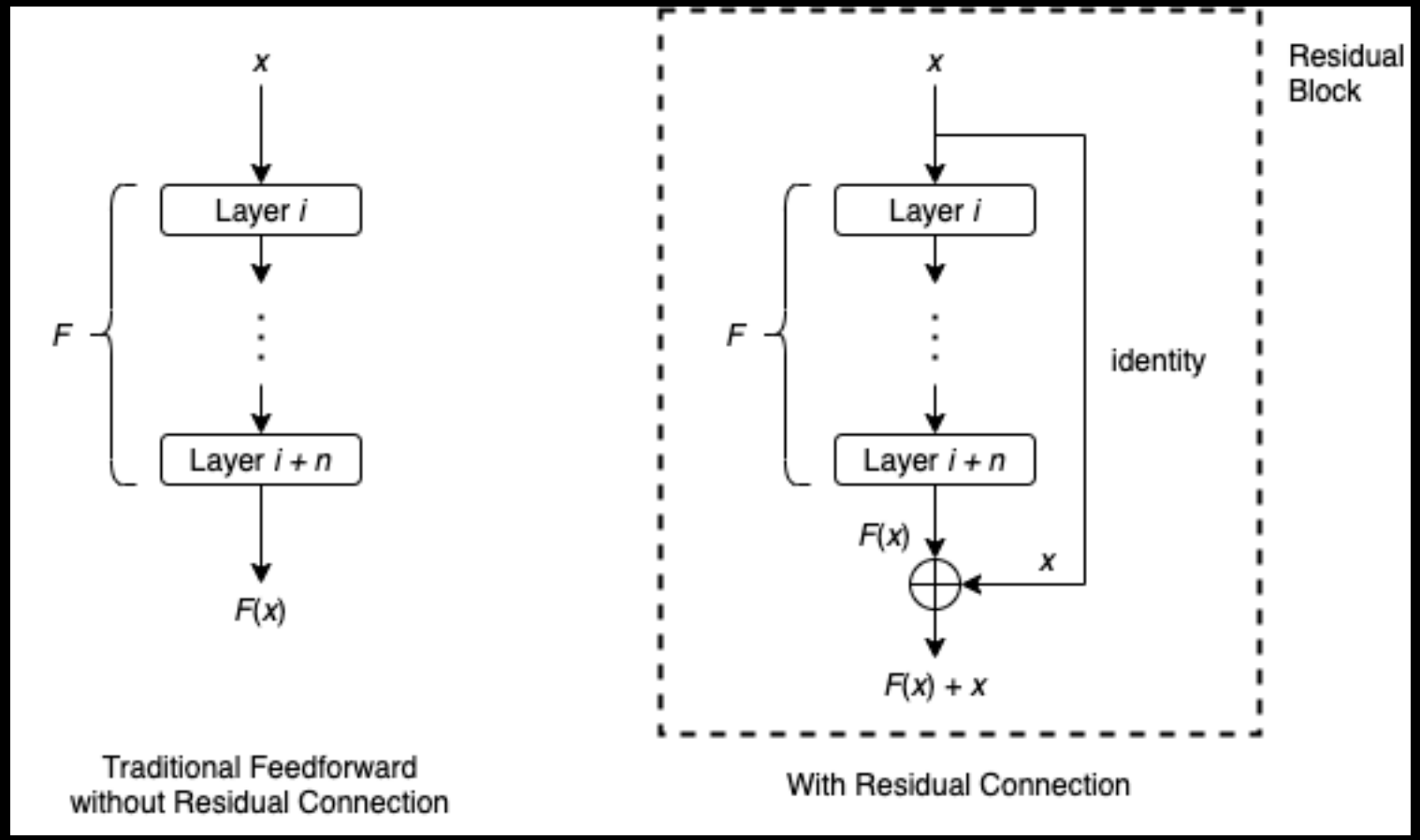
Batch Normalization

- Used to reduce the problem of internal covariate shift in neural networks. It works by normalizing the data within each mini-batch. This means it calculates the mean and variance of data in a batch and then adjusts the values so that they have similar range.



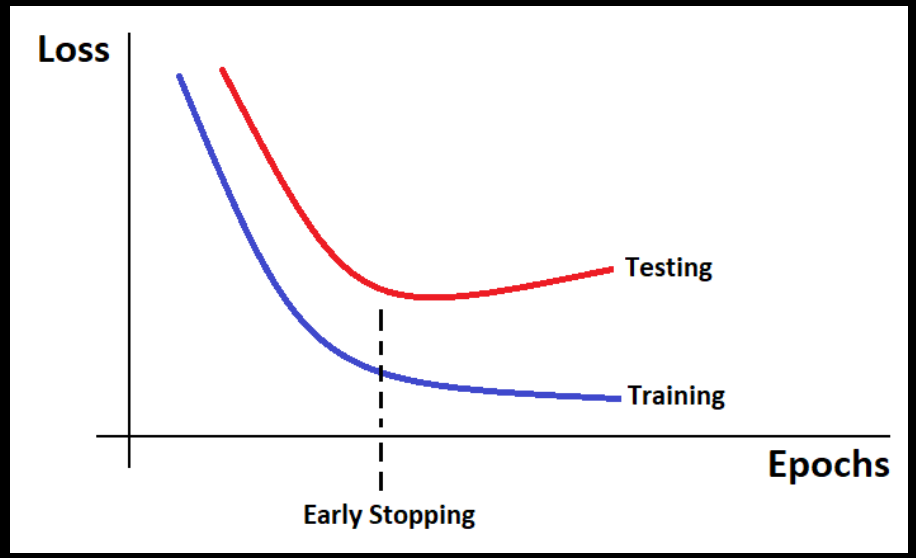
Residual Connections

- Also known as skip connection, provides an alternative path for data to reach later layers by skipping some layers

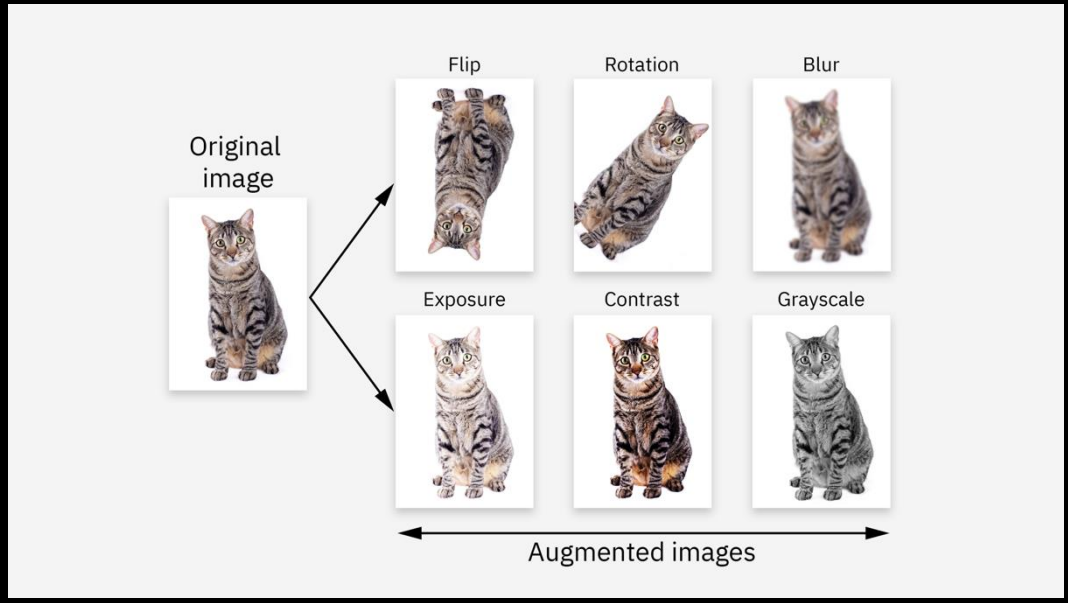


Regularization Techniques

- Early Stopping
  - Stops training when validation loss stops improving

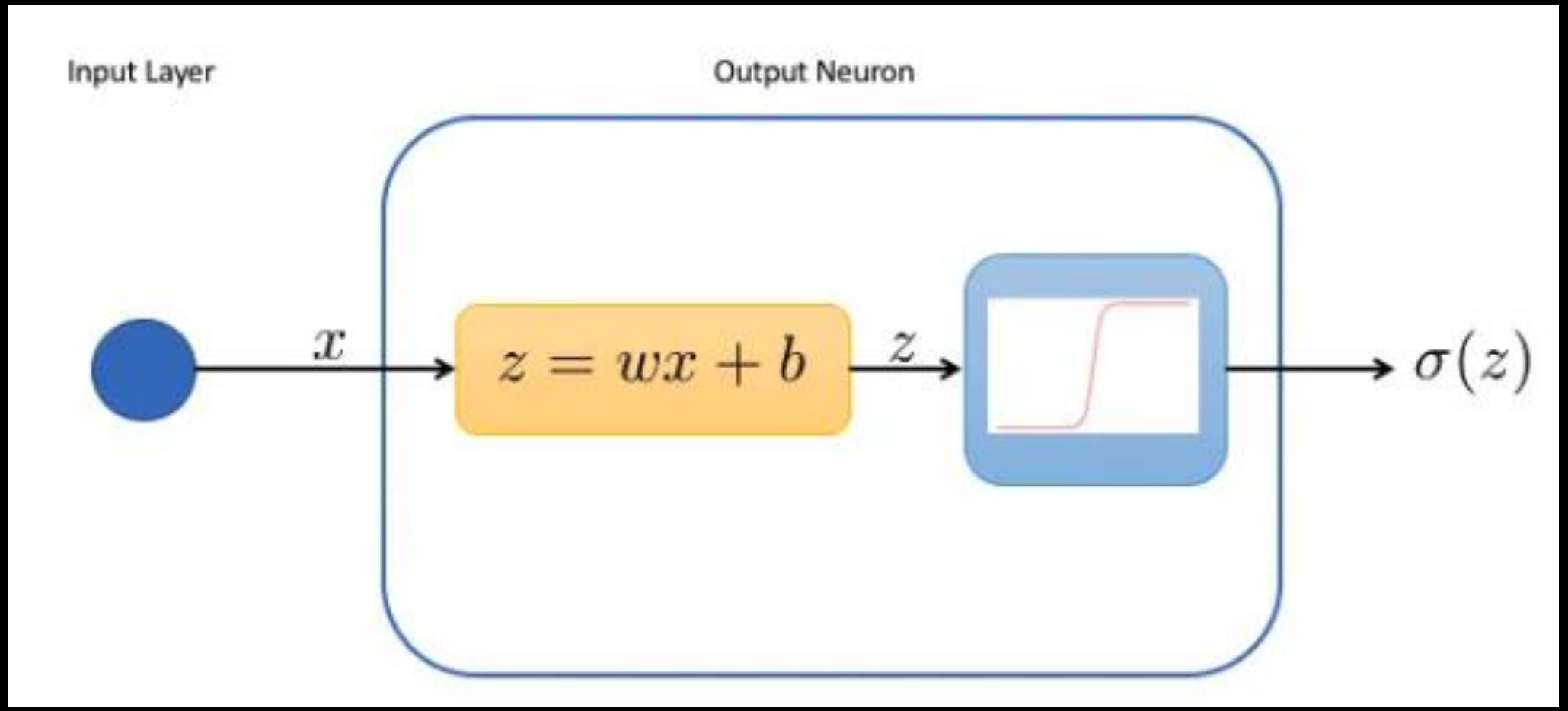


- Data augmentation
  - Adds randomness to input data to make the model generalize better

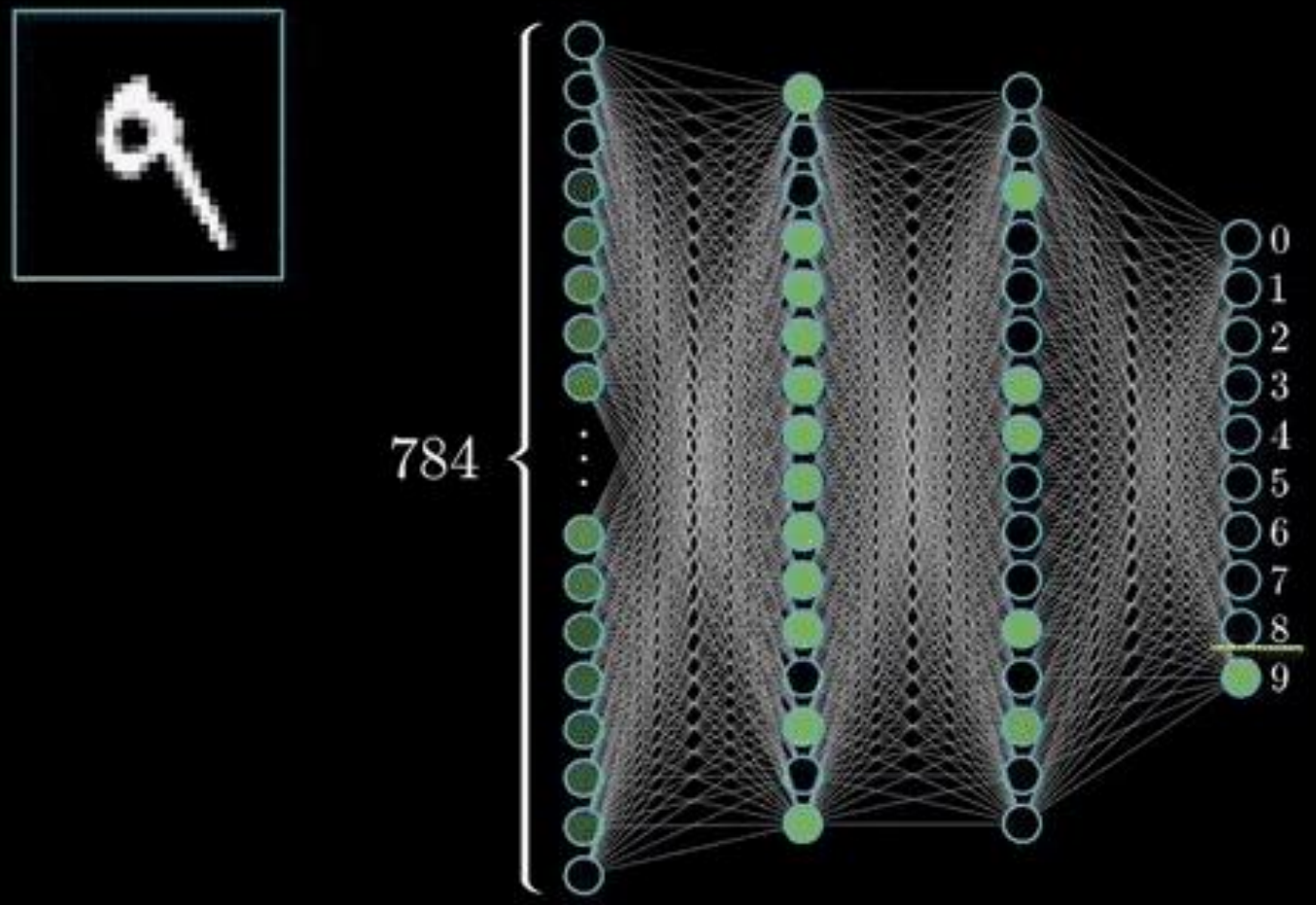


Convolutional Neural Networks

We know about the basic NN

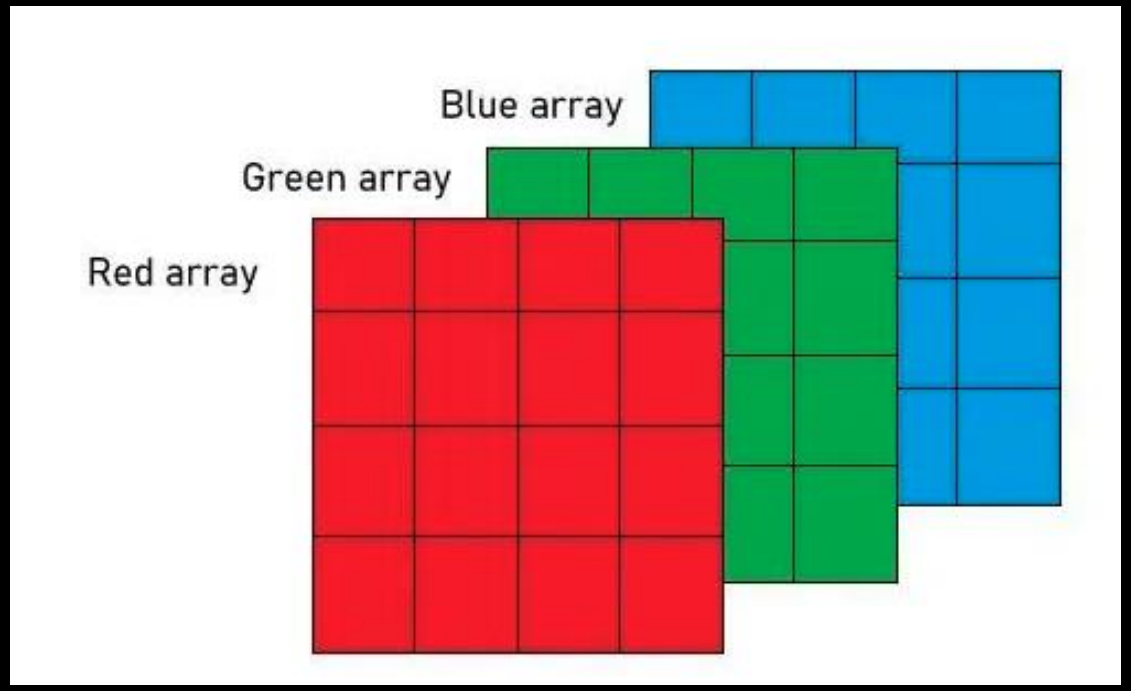


NN example on an image



Doesn’t always capture the features we want… why?

What is an image?

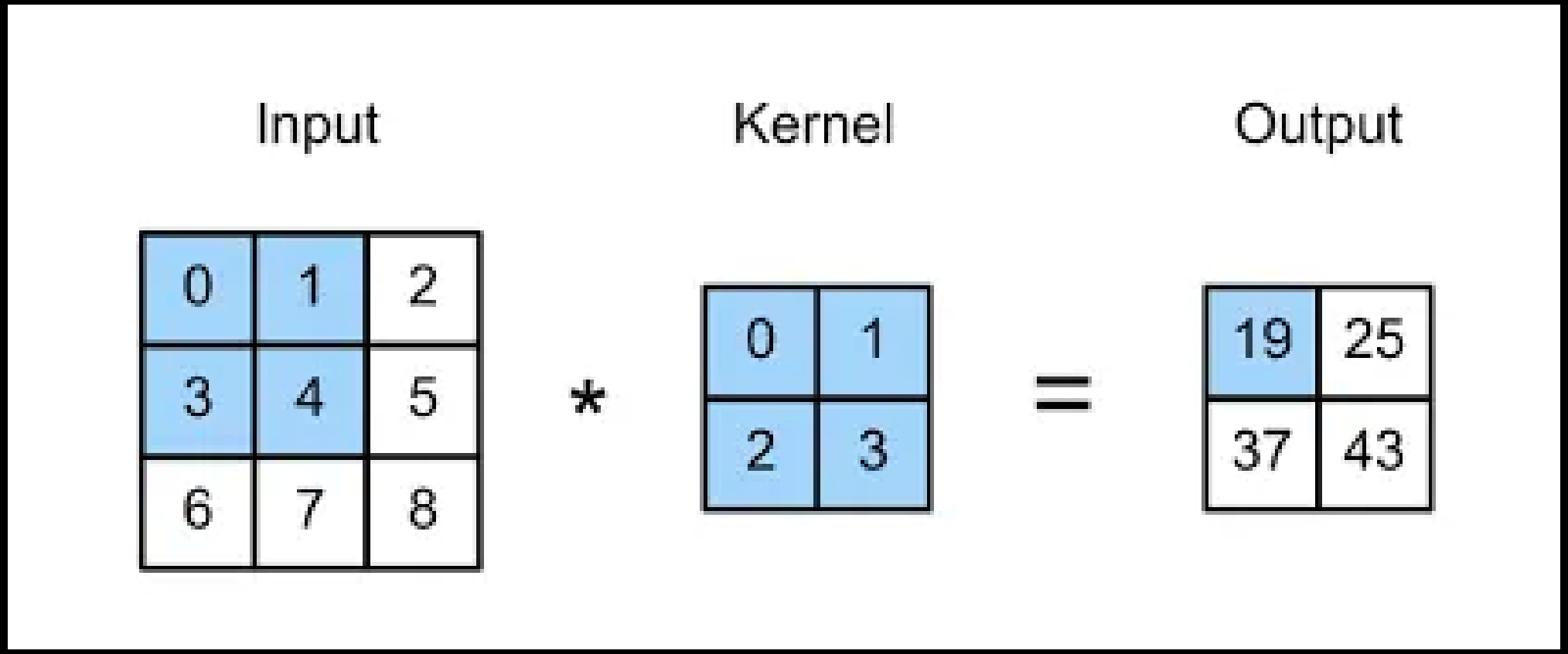


=

Three channels (matrices) stacked on top of each other

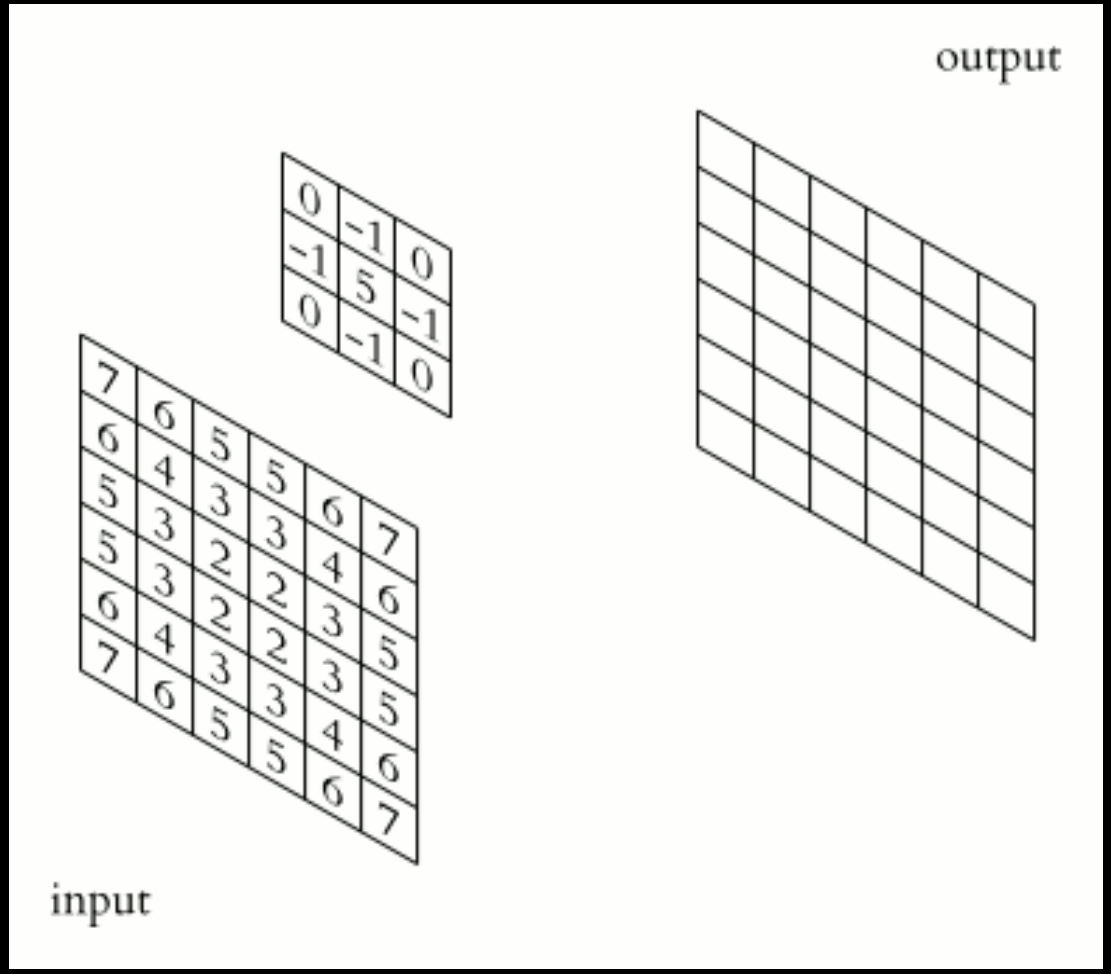
So how can we preserve spatial information in the image for a NN?

Convolutional Operation

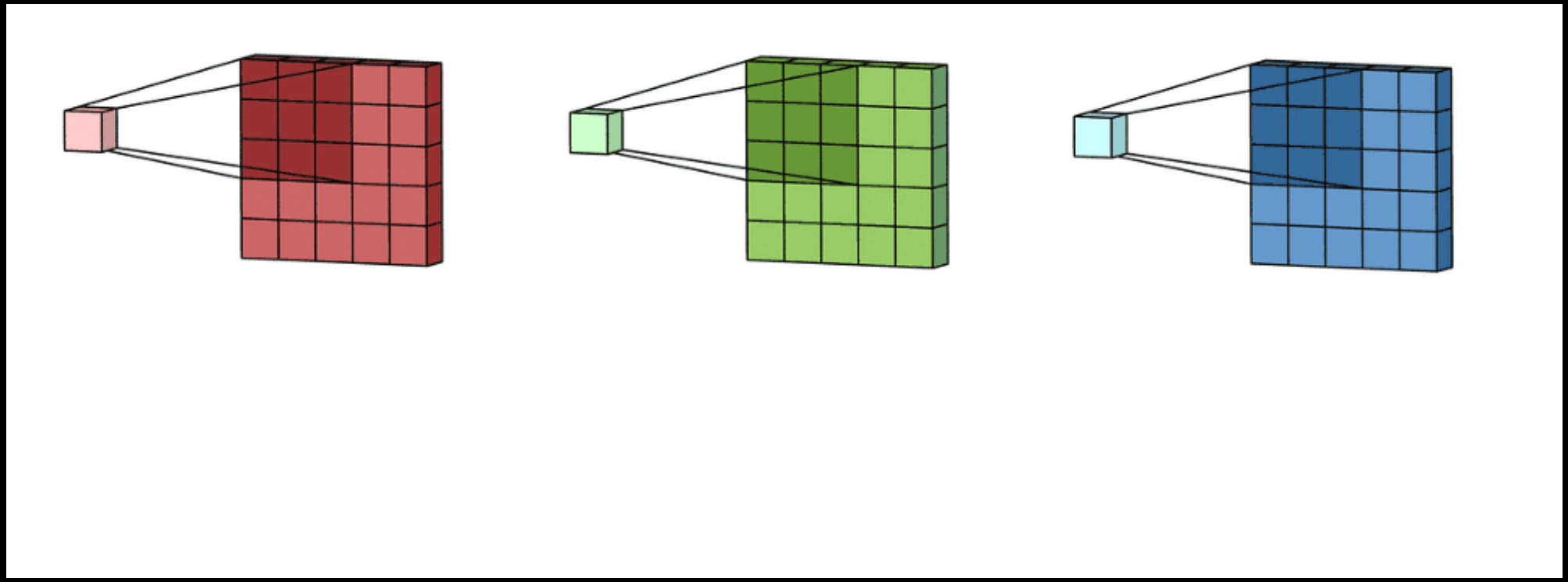


Output matrix is known as a feature map!

Kernels

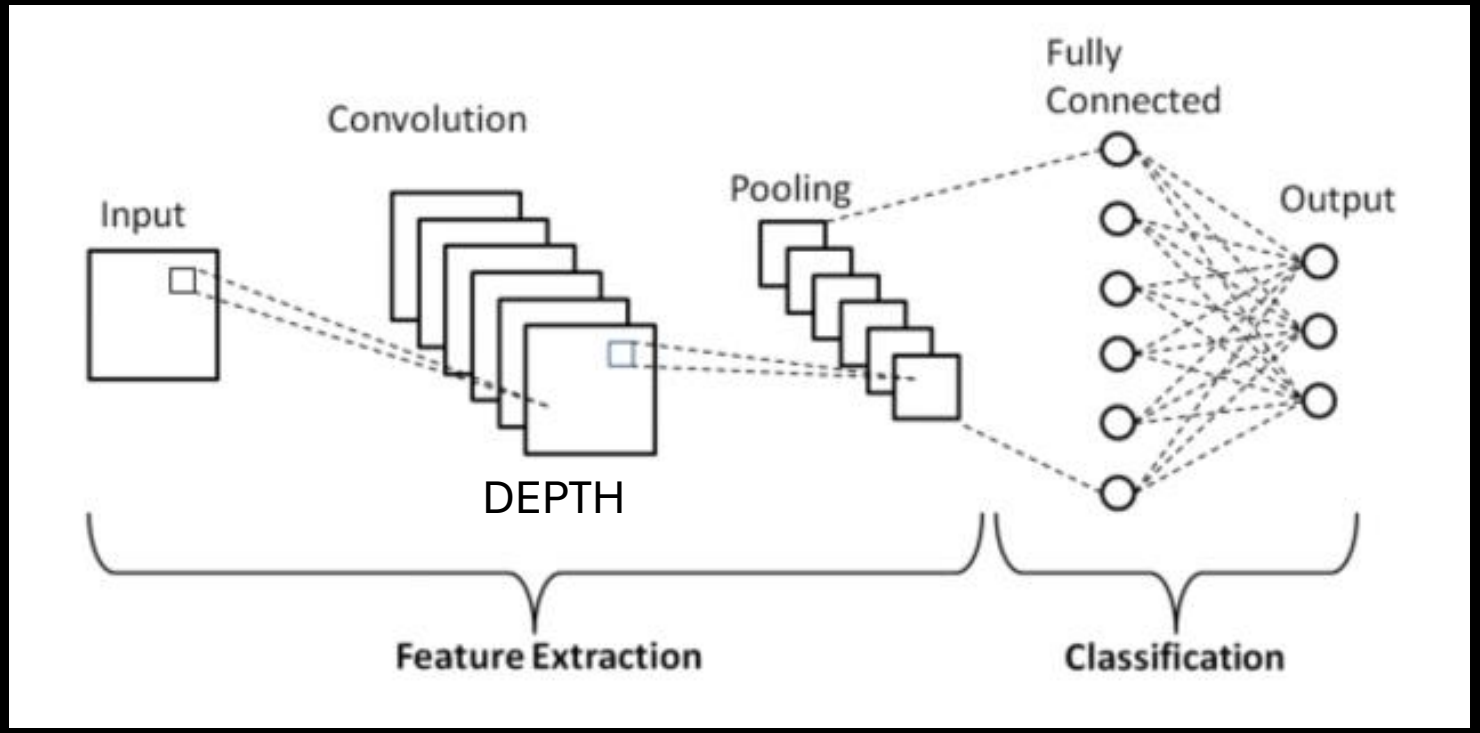

output
0 -1 0
-1 5 -1
0 -1 0
7 6 5 5 6 7
6 4 3 3 4 6
5 3 2 2 3 5
5 3 2 2 3 5
6 4 3 3 4 6
7 6 5 5 6 7
input

There are typically separate kernels for each channel



Depth

The **‘depth’** of a layer refers to the number of kernels it contains. Each filter produces a separate **feature map**, and the collection of these feature maps forms the ***complete output of the layer***

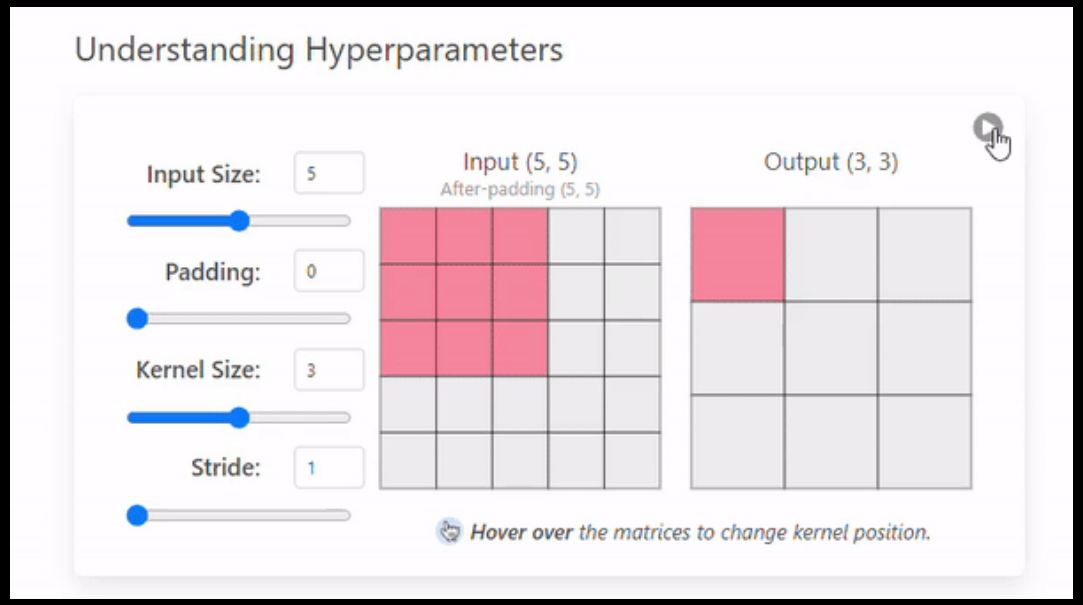


Why would we want more kernels/feature maps?

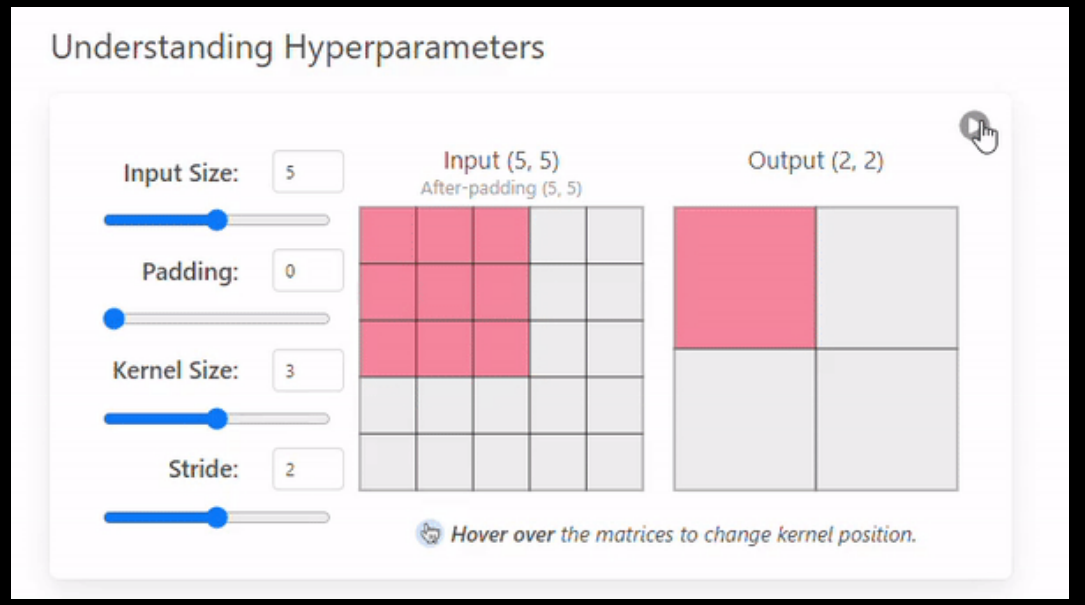
Stride

Stride refers to the number of pixels by which a kernel moves across the input image.

Stride 1



Stride 2

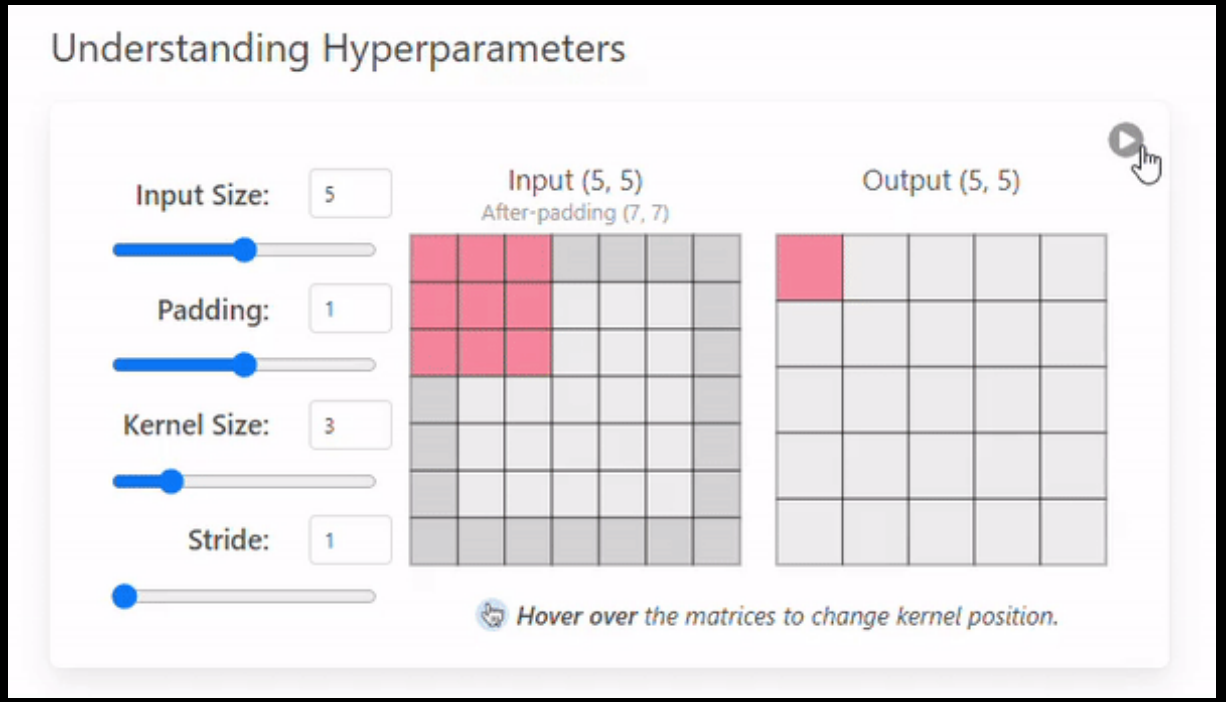


Why would we want a smaller or bigger stride?

Padding

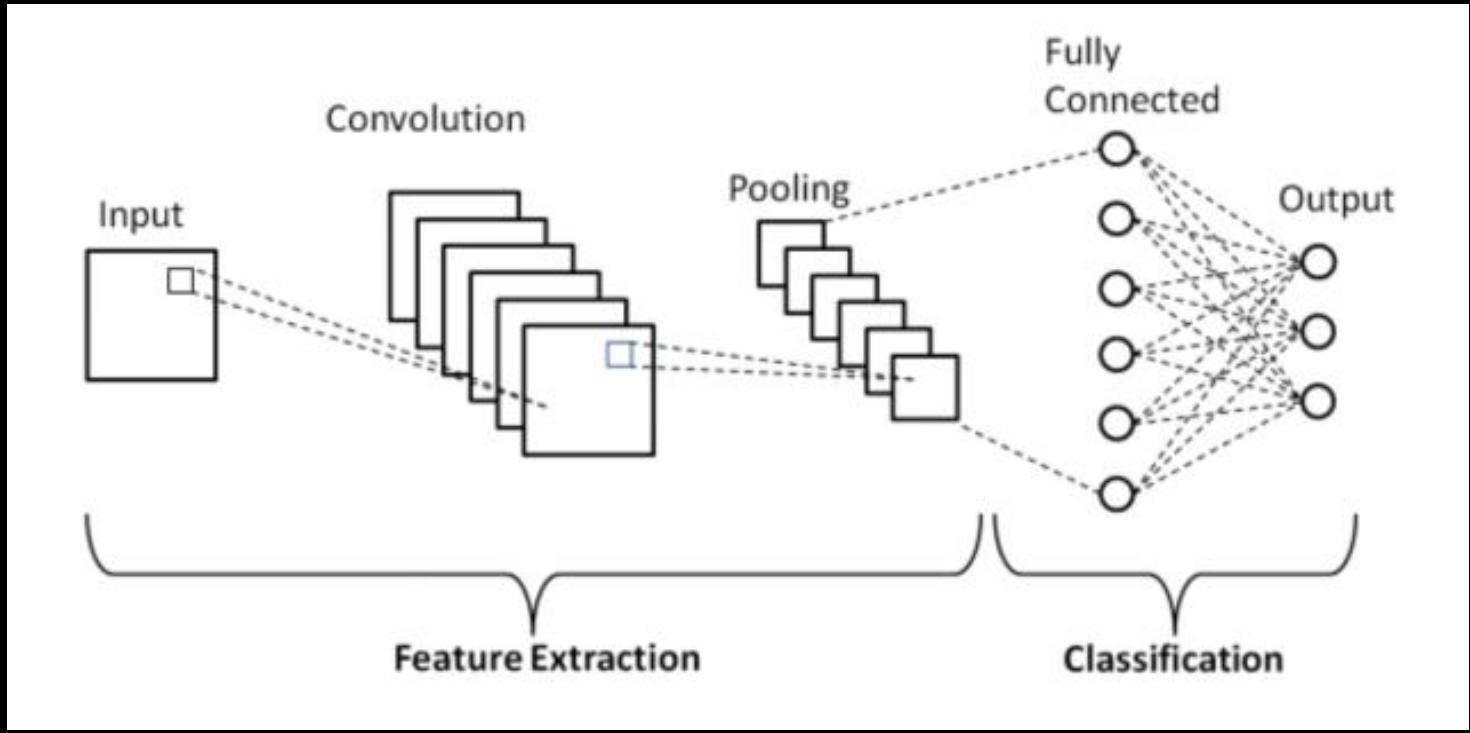
Padding refers to the addition of extra pixels around the edge of the input image.

Padding = 1



Why would we want to add padding?

What is Pooling?

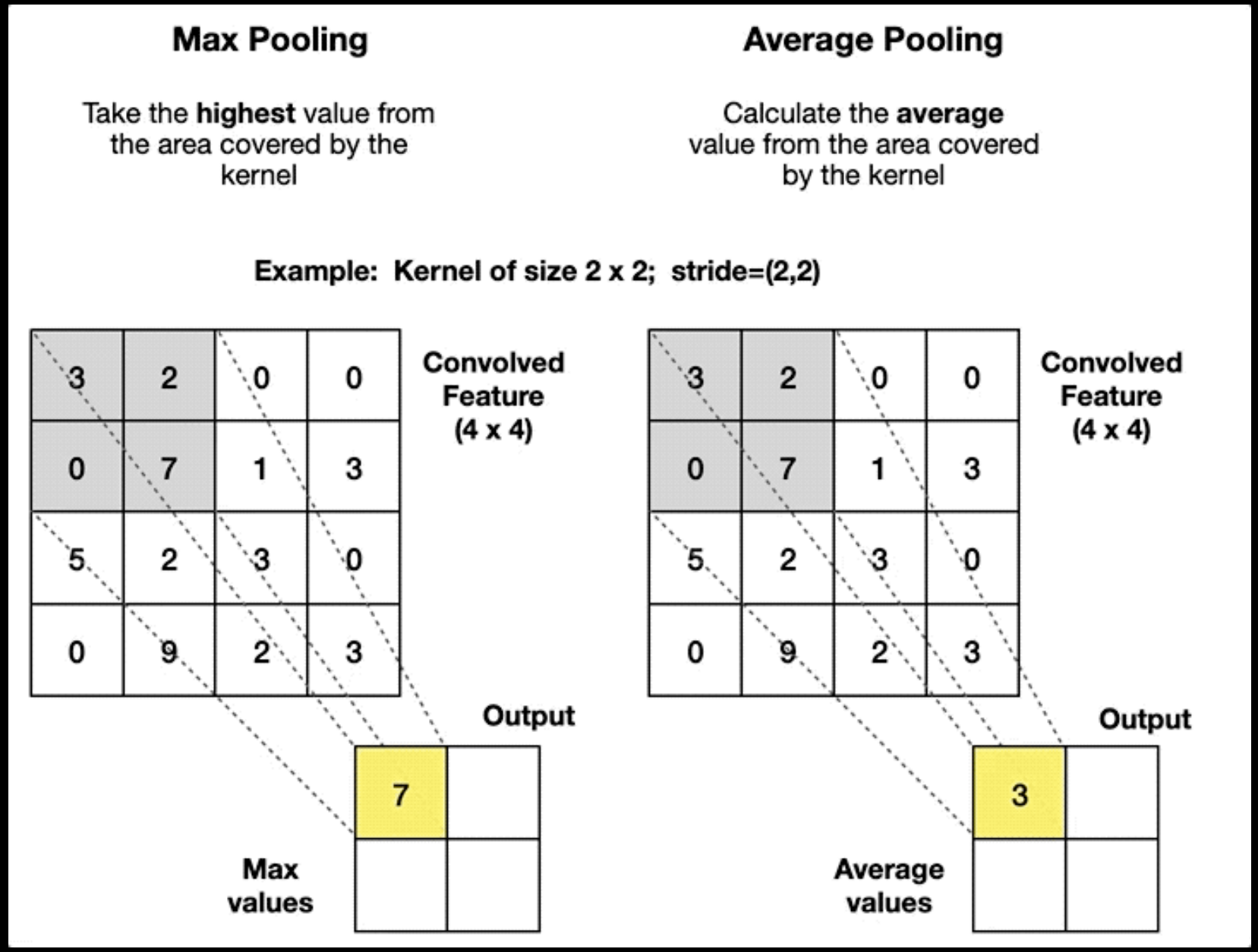


Pooling

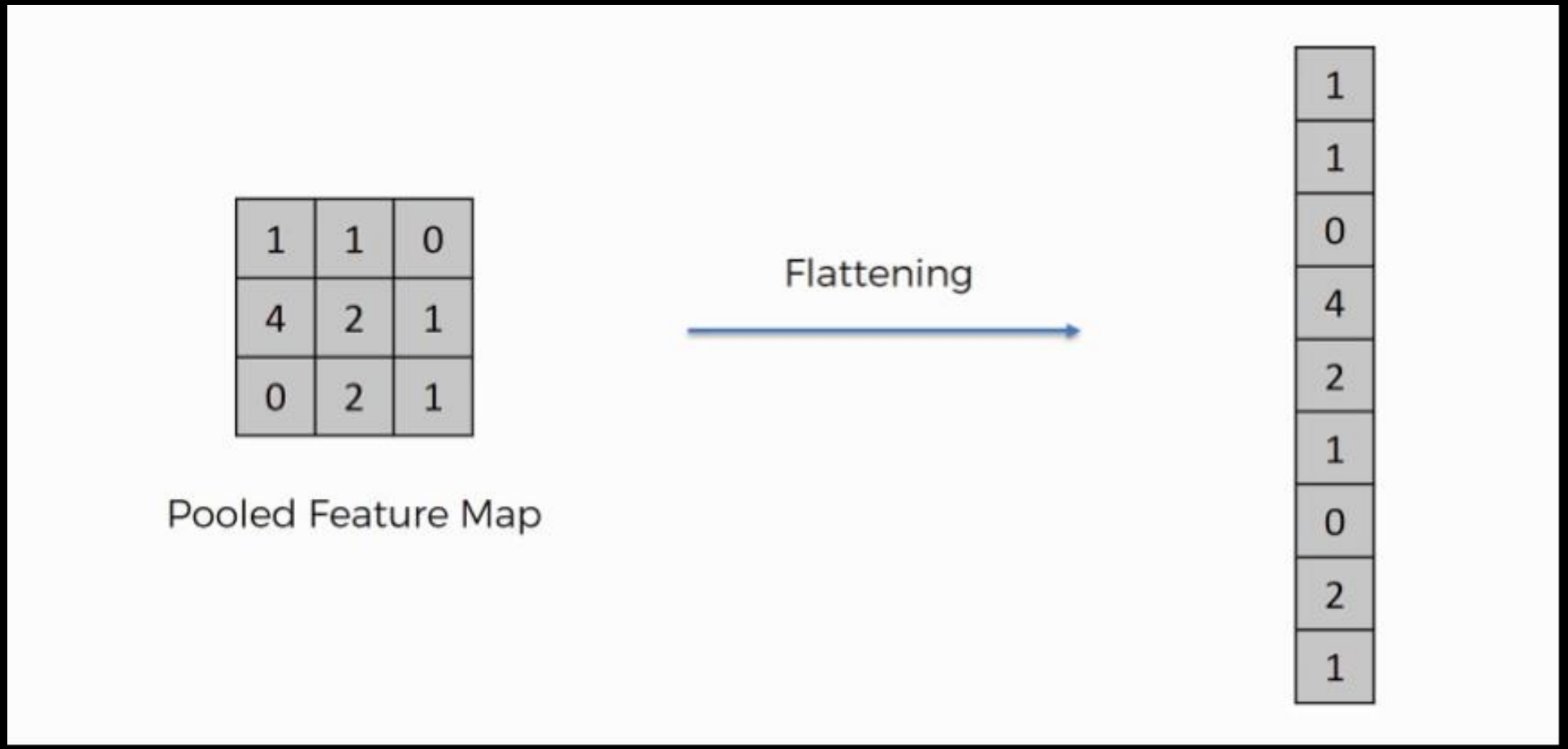
Main objective is dimensionality reduction

Pooling layers do not reduce the number of channels

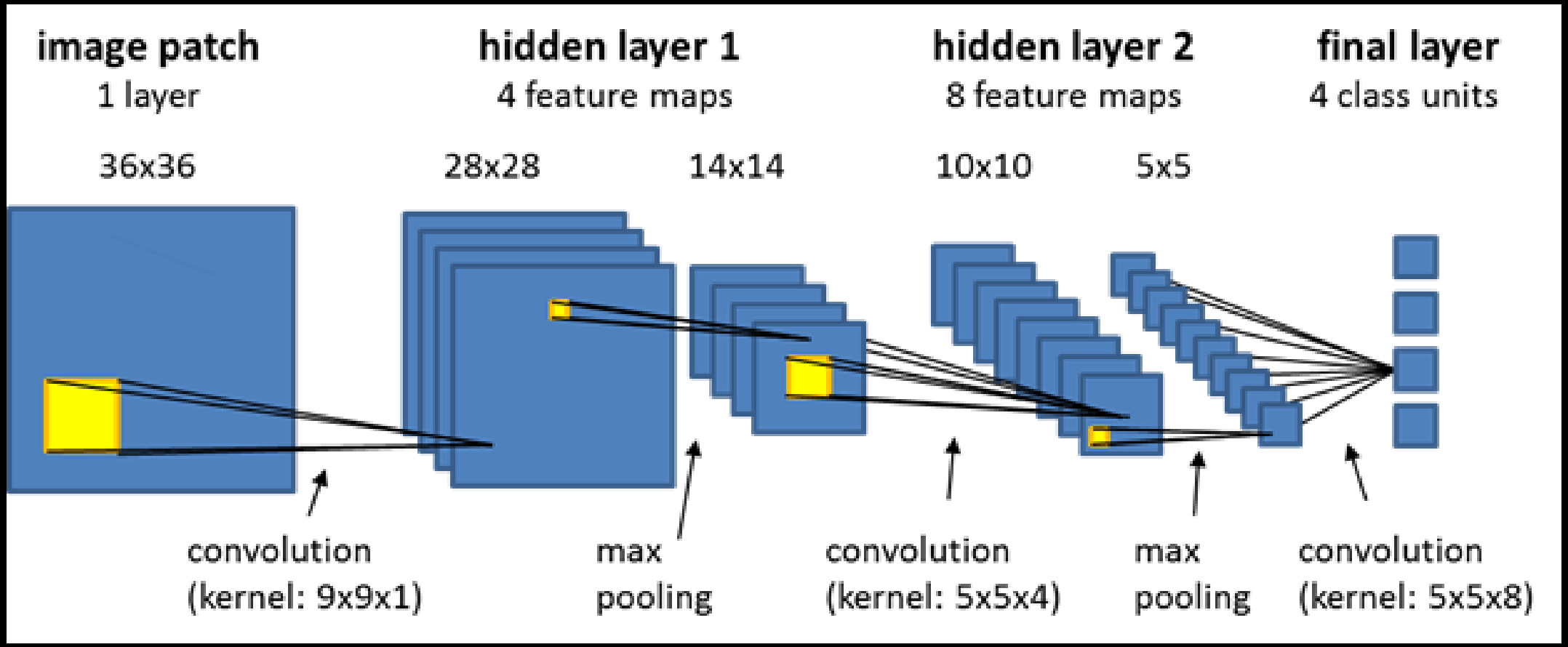
Each pooling operation is applied independently to each channel of the input data



Flattening layer (last layer)

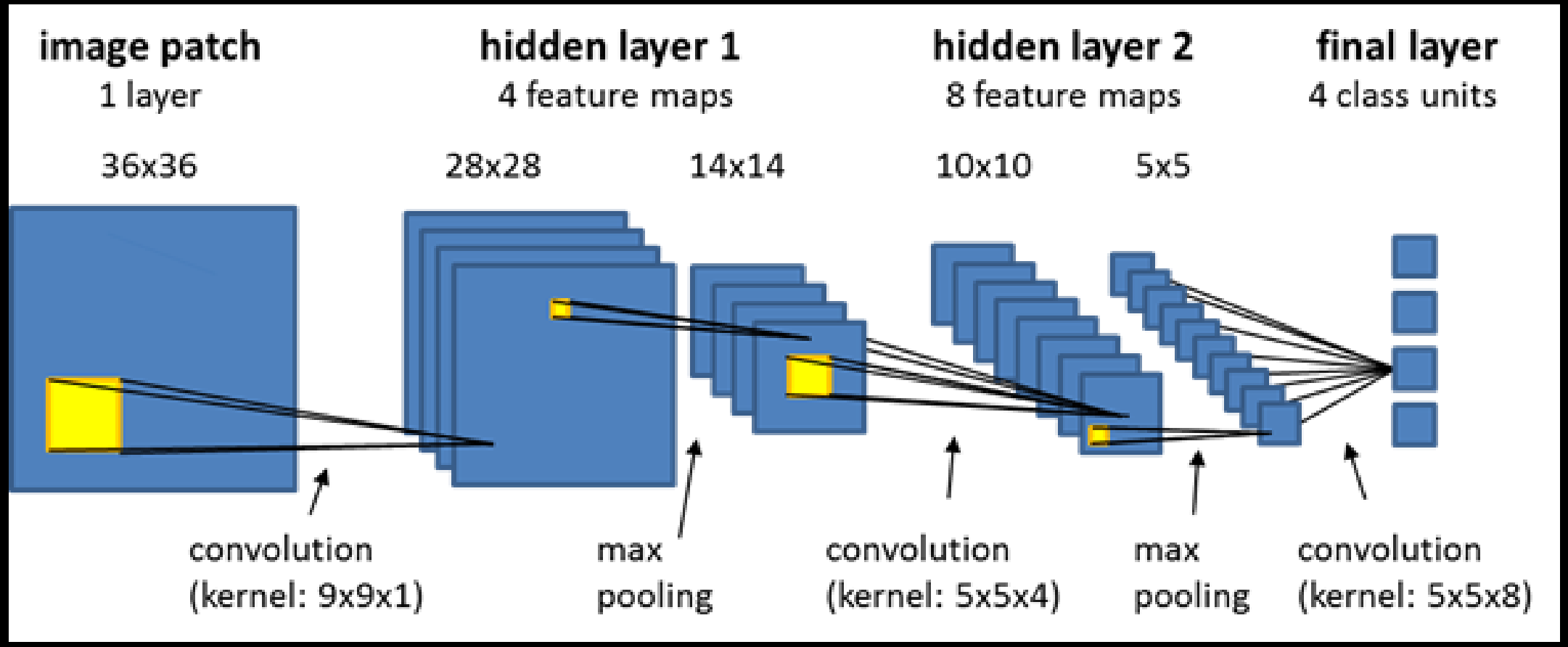


All together so far

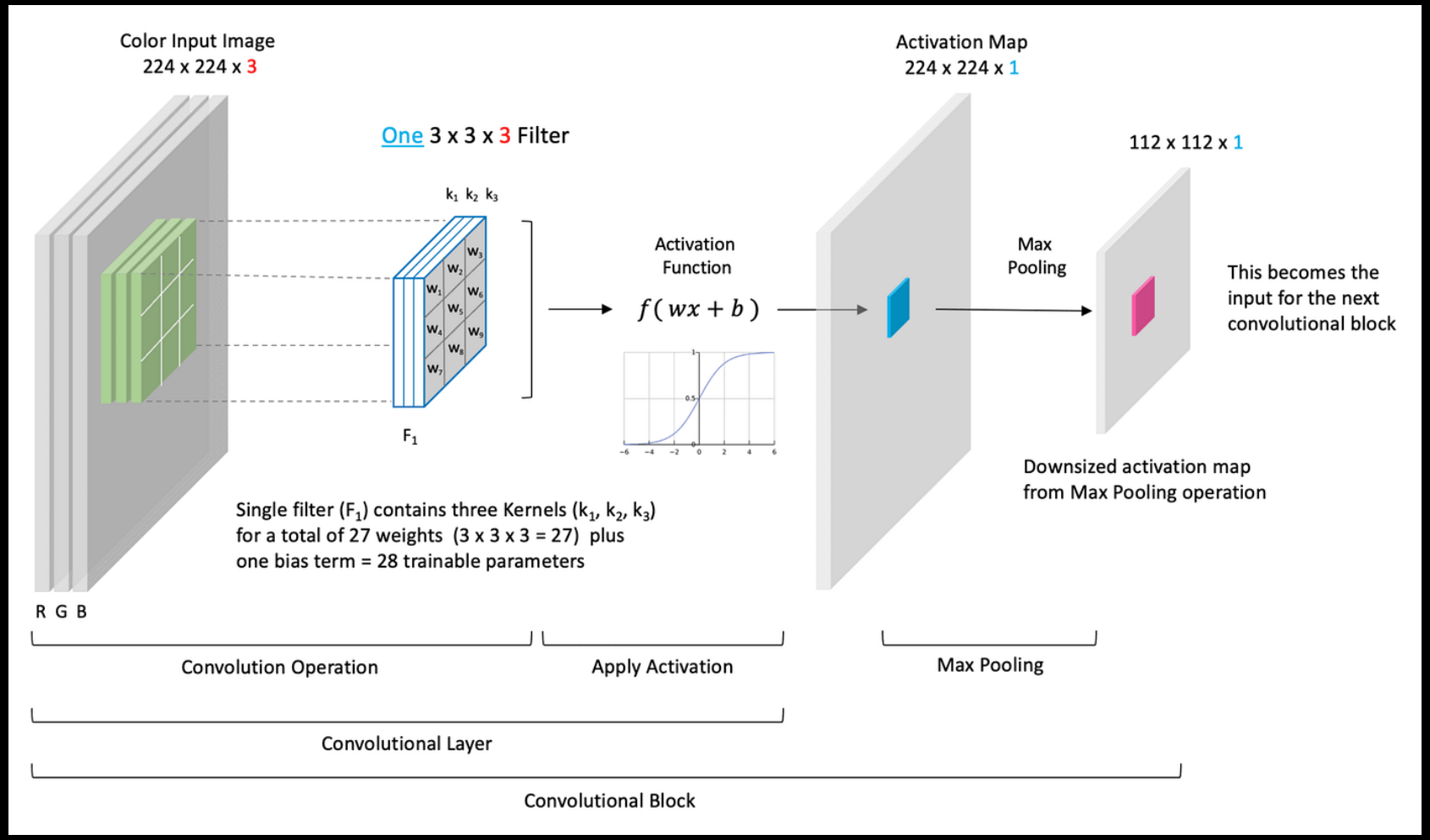


Why do we need Dense Layers in a CNN?

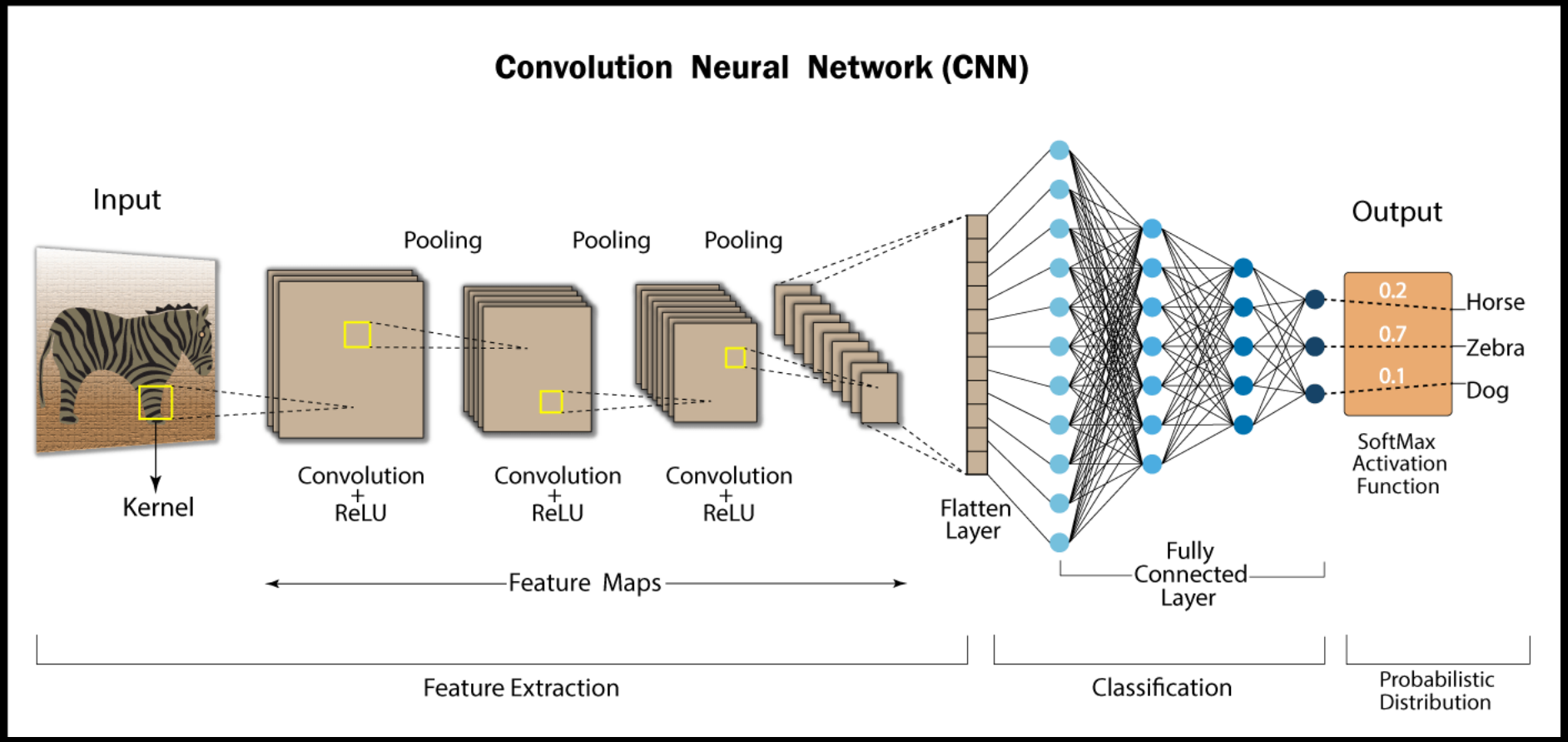
But what are we missing here? 🤔



Activation Functions!



Full CNN

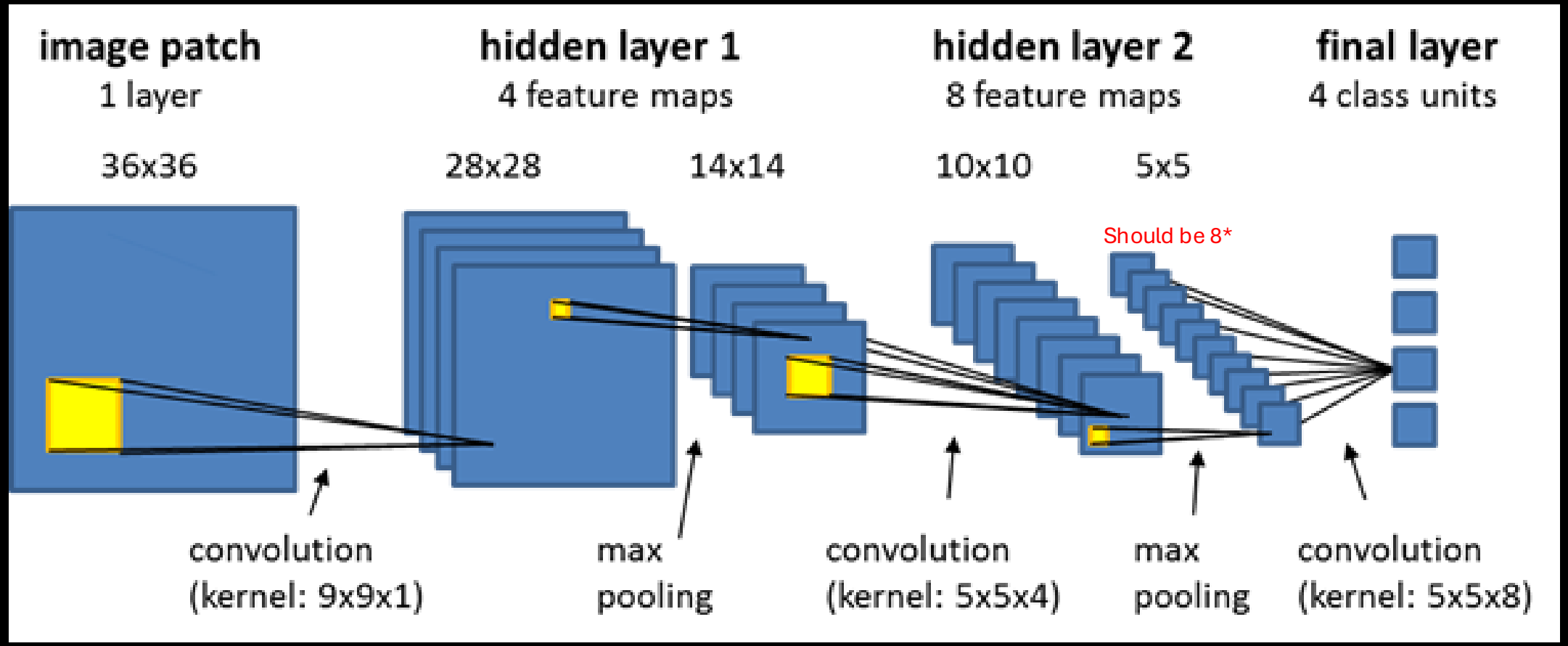

Convolution Neural Network (CNN)
Input
Kernel
Pooling
Pooling
Pooling
Convolution + ReLU
Convolution + ReLU
Convolution + ReLU
Feature Maps
Flatten Layer
Fully Connected Layer
Output
0.2
0.7
0.1
Horse
Zebra
Dog
SoftMax Activation Function
Feature Extraction
Classification
Probabilistic Distribution

How to calculate output size of CNN

$$\text{Output size} = \frac{\text{Input size} + 2 \times \text{Padding} - \text{Kernel size}}{\text{Stride}} + 1$$

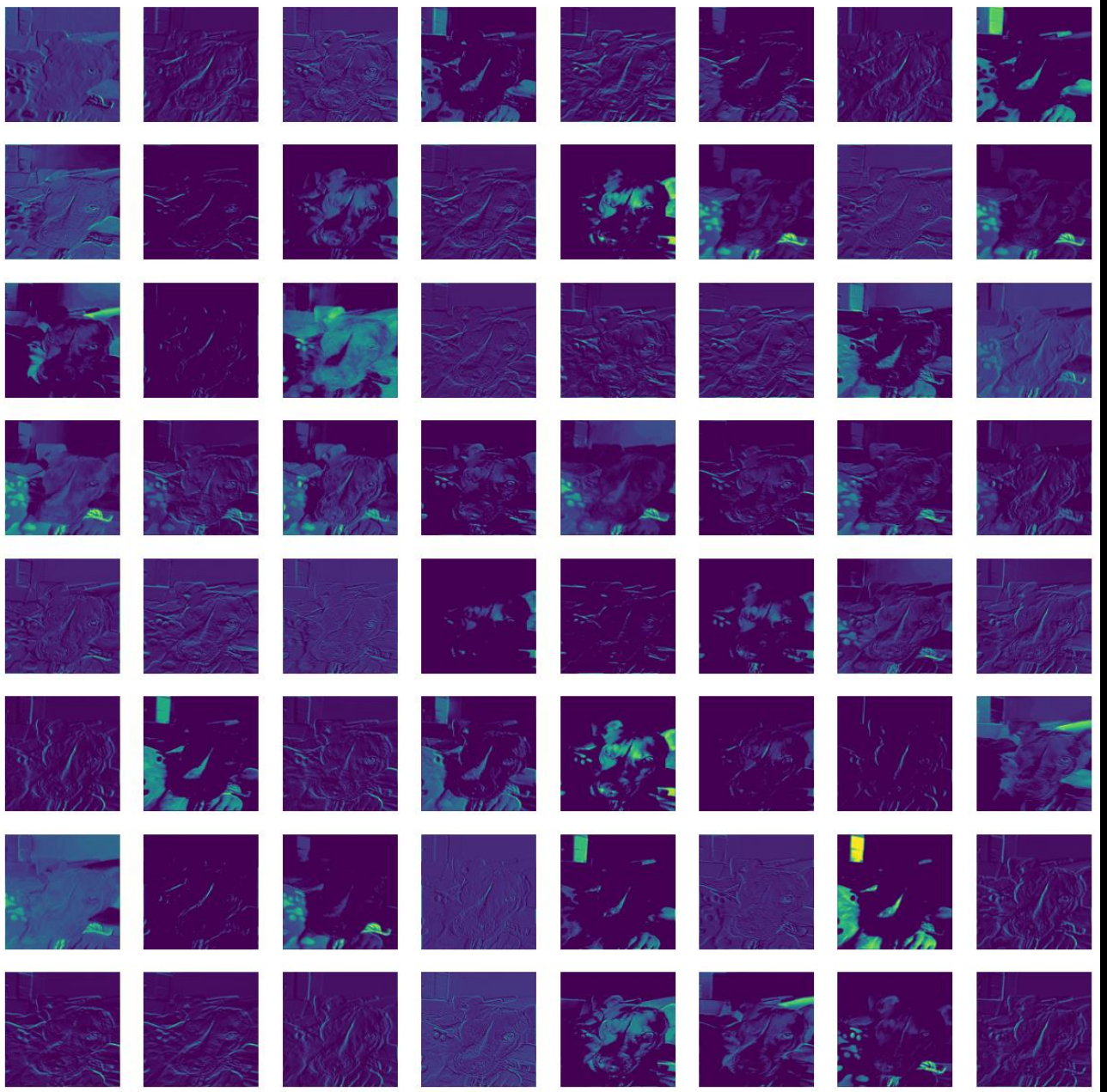
Let's go through this example

$$\text{Output size} = \frac{\text{Input size} + 2 \times \text{Padding} - \text{Kernel size}}{\text{Stride}} + 1$$

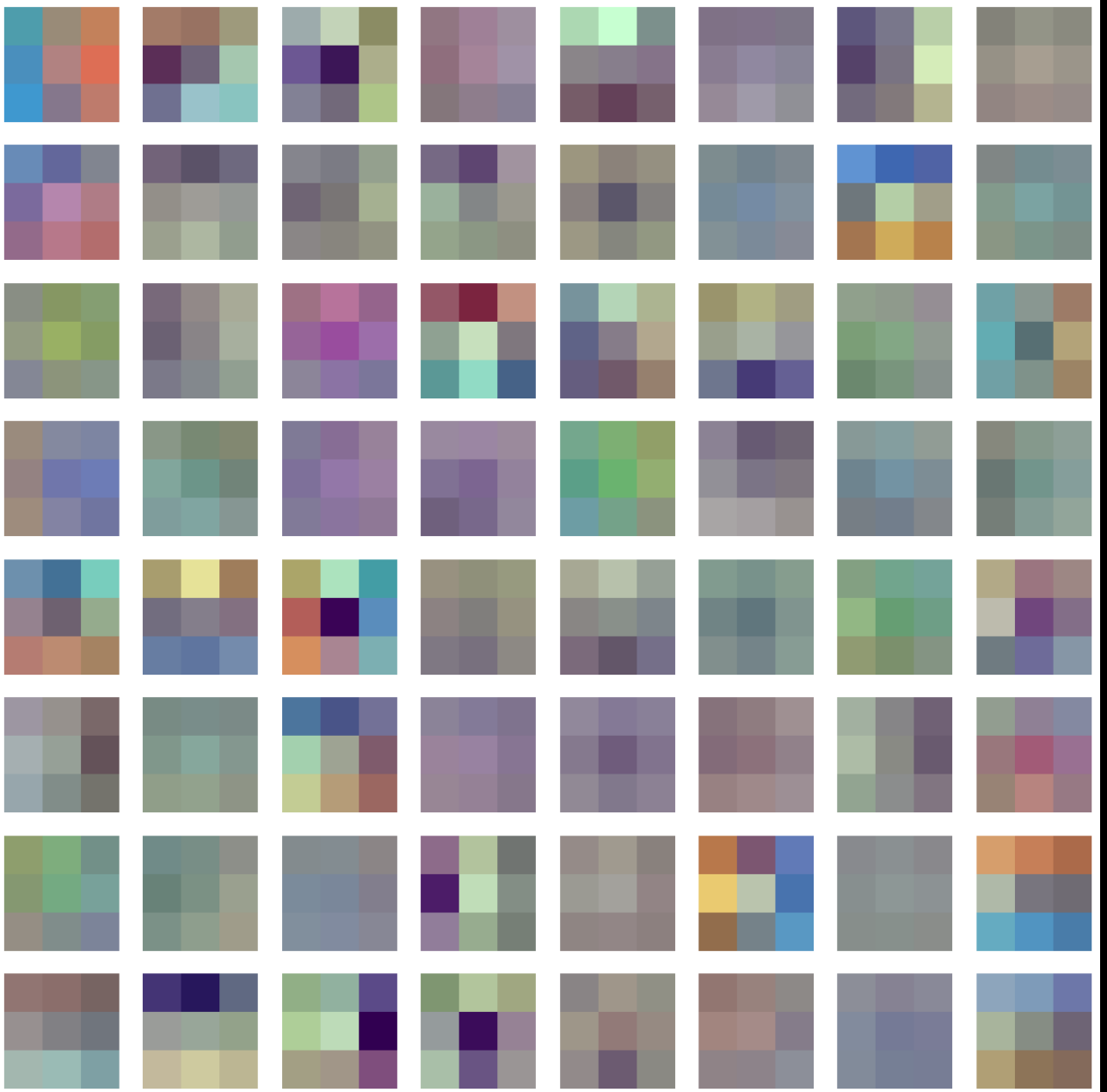


Assume no padding and stride = 1

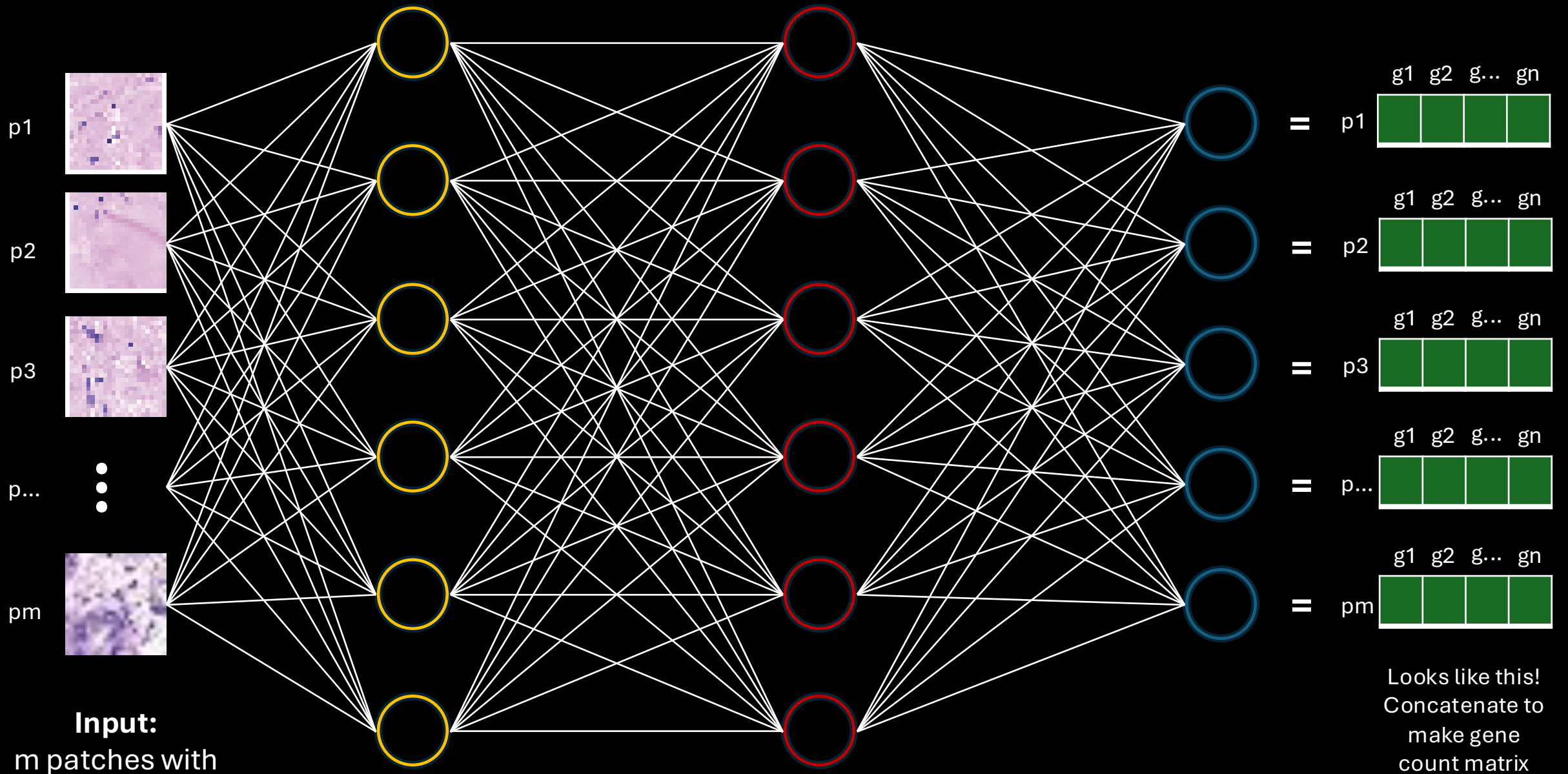
Feature Maps from the First Convolutional Layer



Weights of the First Convolutional Layer



My Research Model Architecture



Input:
m patches with associated counts matrix

Feature Extractor:
Pretrained ResNet50

Hidden Layers:
Dimensionality Reduction

Output:
n gene expression per patch

Looks like this! Concatenate to make gene count matrix

My Research Model Architecture in python

```
GeneExpressionPredictor(
  (feature_extractor): ResNet(
    (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
    (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU()
    (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    (layer1): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu_1): ReLU()
        (relu_2): ReLU()
        (relu_3): ReLU()
        (downsample): Sequential(
          (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
```

⋮

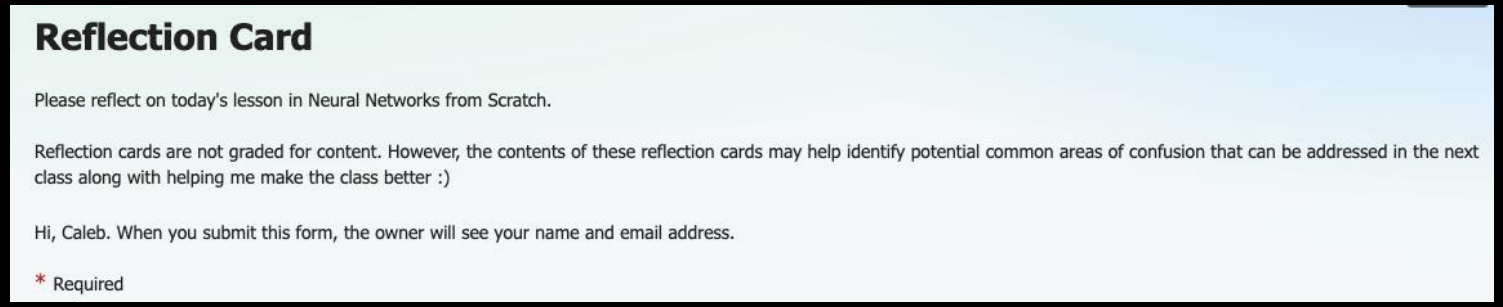
CNN part

```
(feature_layers): Sequential(
  (0): Linear(in_features=2048, out_features=2048, bias=True)
  (1): BatchNorm1d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU()
  (3): Dropout(p=0.2, inplace=False)
  (4): Linear(in_features=2048, out_features=1024, bias=True)
  (5): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (6): ReLU()
  (7): Dropout(p=0.2, inplace=False)
  (8): Linear(in_features=1024, out_features=512, bias=True)
  (9): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (10): ReLU()
  (11): Dropout(p=0.2, inplace=False)
  (12): Linear(in_features=512, out_features=256, bias=True)
  (13): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (14): ReLU()
  (15): Dropout(p=0.2, inplace=False)
)
(output): Linear(in_features=256, out_features=306, bias=True)
```

Parameters we have discussed!

Let’s make a simple CNN in PyTorch

Reflection Cards



Reflection Card

Please reflect on today's lesson in Neural Networks from Scratch.

Reflection cards are not graded for content. However, the contents of these reflection cards may help identify potential common areas of confusion that can be addressed in the next class along with helping me make the class better :)

Hi, Caleb. When you submit this form, the owner will see your name and email address.

* Required

- Essentially a means to help me make this class better!

1. What is something that you learned in today's lecture?
2. What is something that you are still confused about from today's lecture?
3. Do you have any other comments/feedback/thoughts/suggestions/concerns?

https://forms.office.com/r/Kv8LtW4iJH