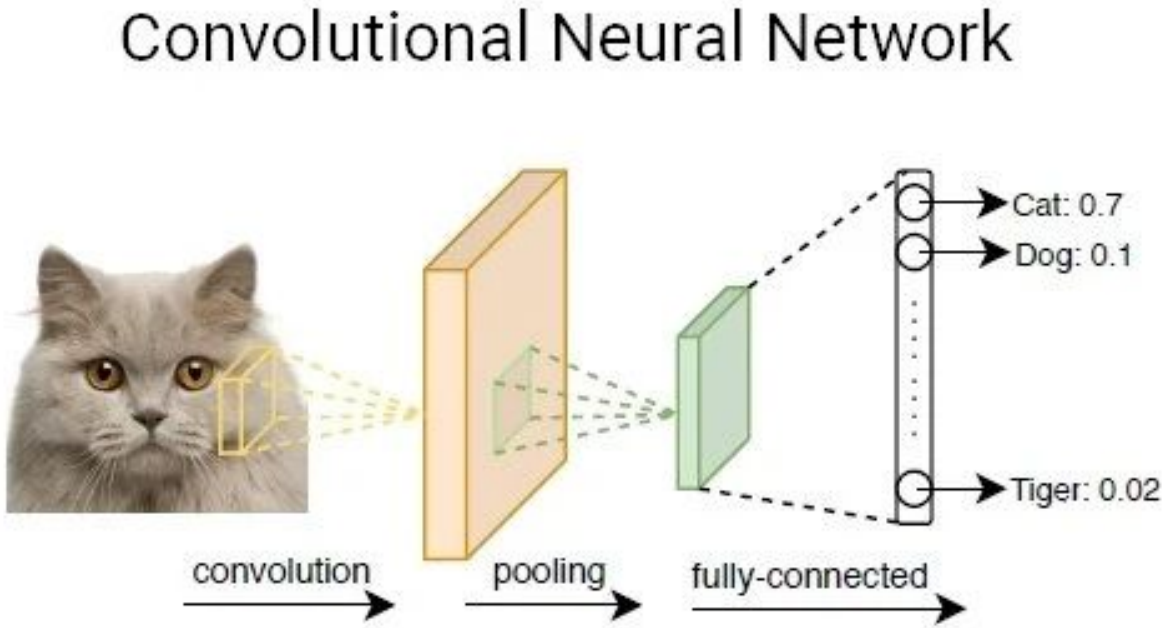
Intro to Convolutional Neural Networks

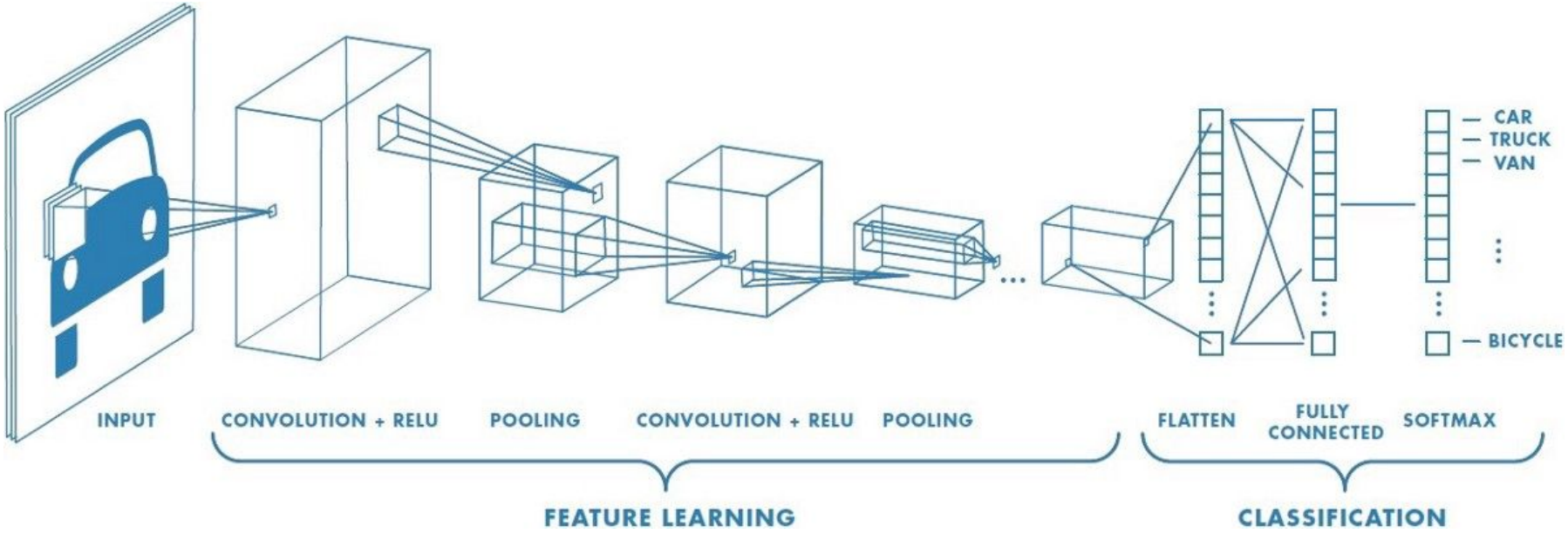
BUDSA – Workshop

By: Christina + Ryan

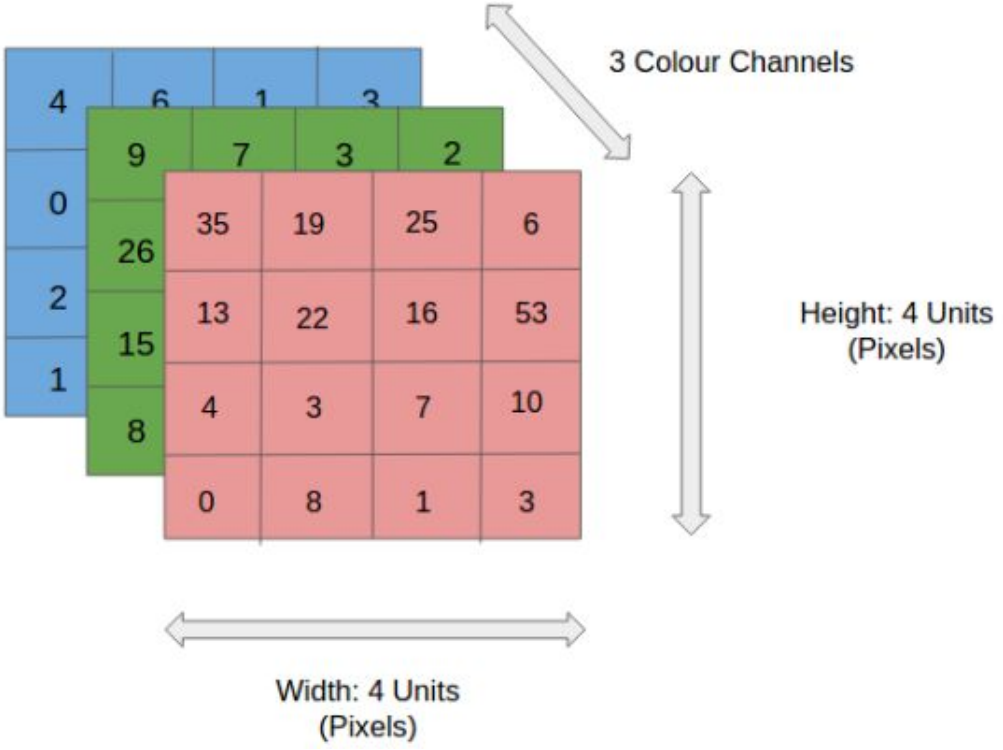
Sample Architecture



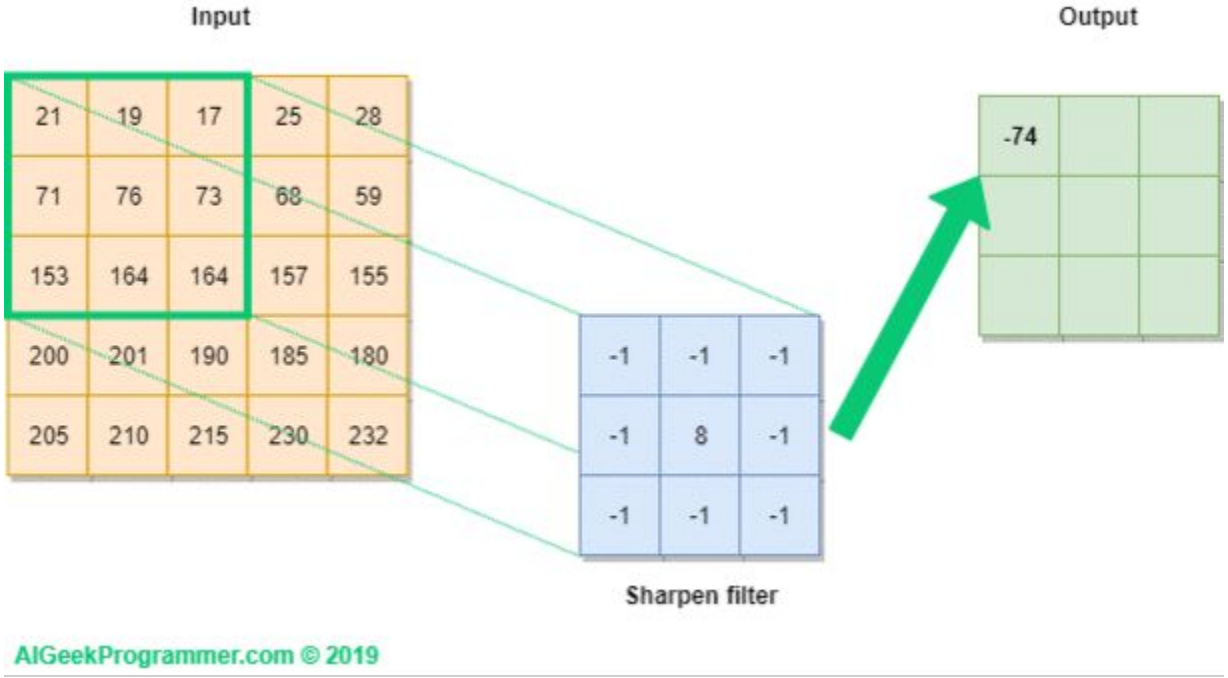
Sample Architecture



How do images work?



Kernels (Filters)

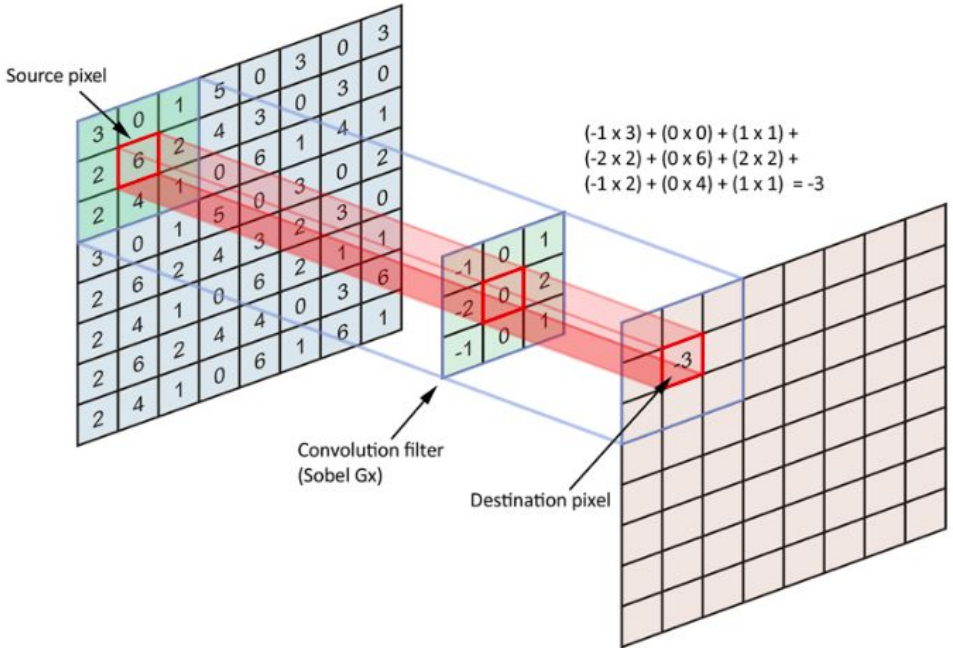


https://setosa.io/ev/image-kernels/

1. Convolutional Layer and Nonlinear Transformation

- Preserves locality
- Finds different kernels (filters)
  - Best weights and biases via gradient descent
- Nonlinear transformation functions: ReLU, Tanh, GeLU

Imagine a special projector onto a screen

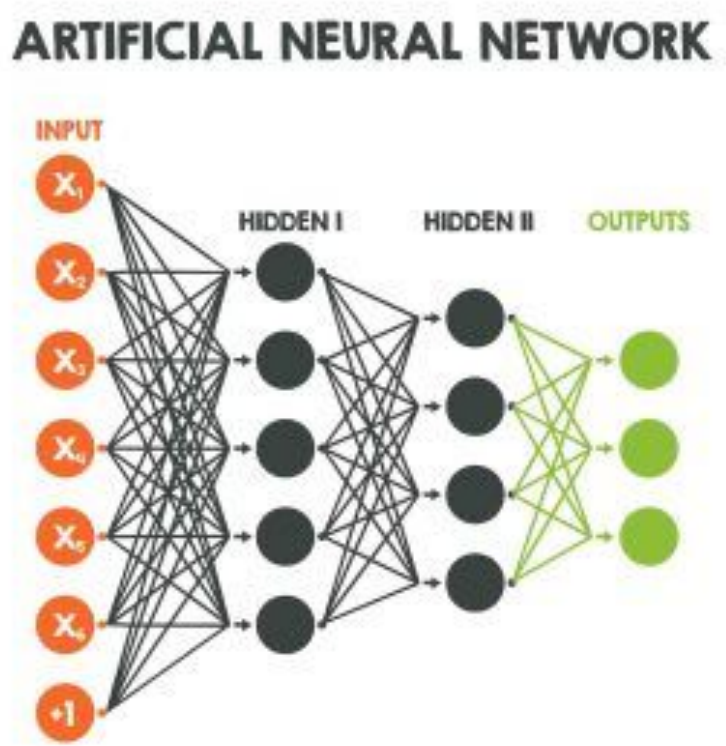


2. Pooling Layer

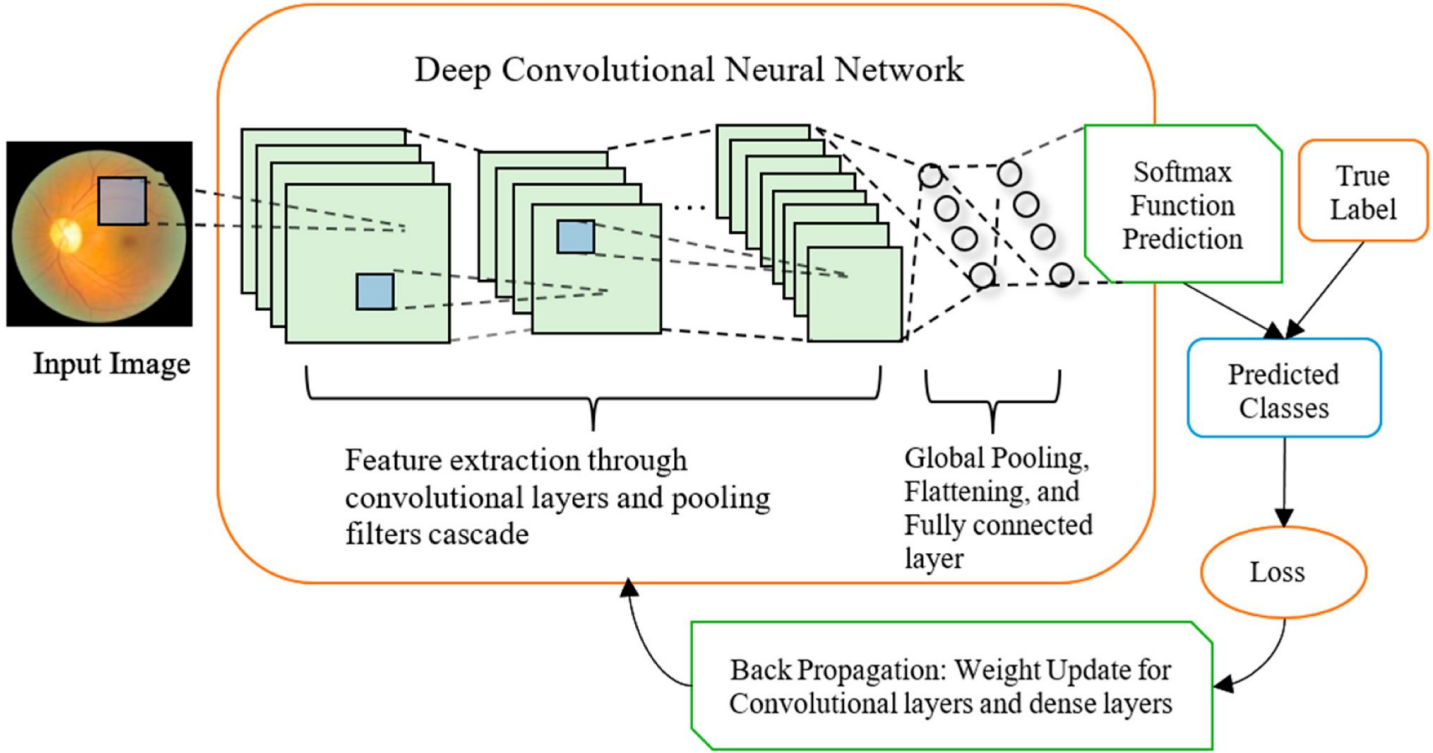
Ex. Max2D Pooling – accentuates prevalent features within the image

3. Flatten and Deep Neural Network

- Flatten the final image into a singular vector
- Fully-connected neural network
- Last layer -- softmax for probabilities of classes, or replace with one neuron for class prediction



Putting it together: Construction of Architecture

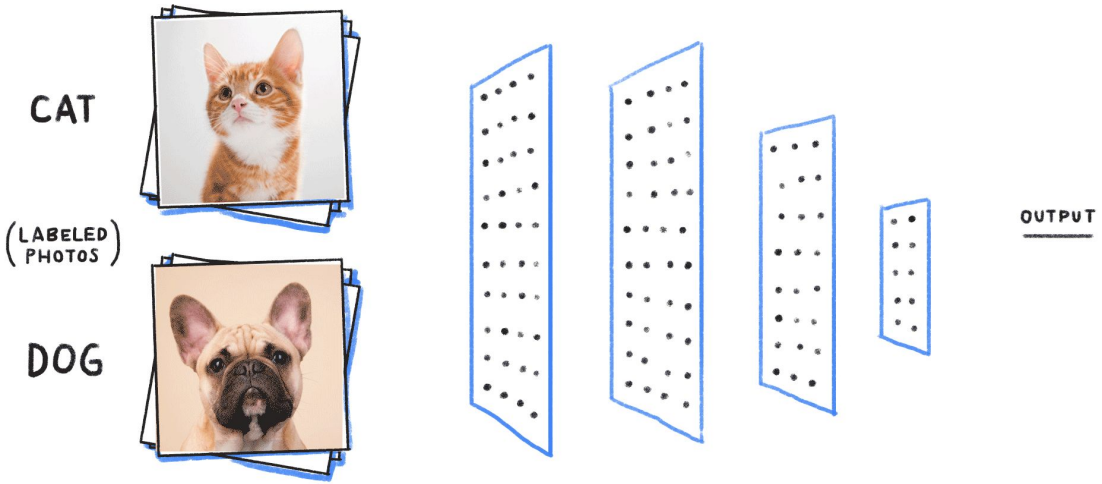


How to train?

When you create the architecture, you initialize the architecture with some weights and biases

Backpropagation is always the answer

- Finds the best kernels (filters)
- Finds the best weights and biases for fully-connected layers



Why do CNN work so well for image processing?

- Convolution preserves locality
- Max2DPooling layers accentuates features
- Nonlinear transformations allow network to look for latent features (unobservable features of ex. cat/dog)

Cons?

- Long to train with limited computing power
- Typically need to augment data to get better results
- Vision Transformers are better for SOTA

Resources

- CNN: https://poloclub.github.io/cnn-explainer/
- CNN Cheatsheet: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks
- Visualizing Kernels: https://setosa.io/ev/image-kernels/
- Statquest on CNN: https://www.youtube.com/watch?v=HGwBXDKFk9I