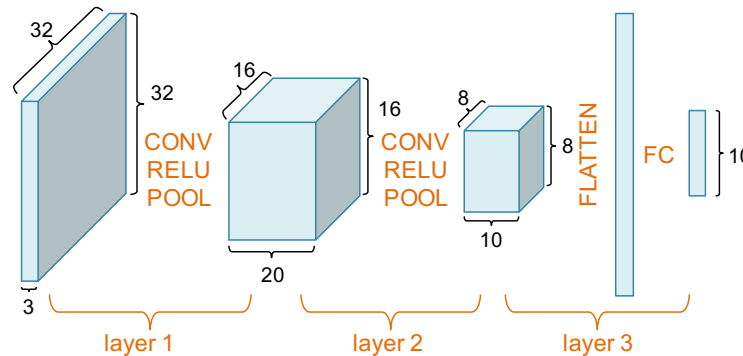


Convolutional Neural Networks (CNNs)*(find and work with a partner)*

1. Say we use a 3-layer CNN with architecture shown below. Our inputs have shape $(32 \times 32 \times 3)$.



- In the CONV step of the first layer, we use 20 filters, each (5×5) in width and height, but all the way through the depth. We use a stride of 1 in each dimension, and use padding = “SAME” to make sure the input and output width/height are the same. After CONV, we apply a RELU non-linearity, then apply POOL using a max-pooling strategy with (2×2) filters and stride 2 in width/height. This reduces the width and height by a factor of 2.
 - In the second layer we use 10 filters, each (3×3) in width and height, but all the way through the depth. The stride and padding follow the same procedure as the first layer. ReLU and pooling also follow the same strategy.
 - Finally, we flatten the volume in preparation for the full connected layer. The FC layer transforms the flattened volume into scores for 10 classes.
- (a) Which steps (i.e. CONV, RELU, POOL, FLATTEN, FC) require parameter learning through gradient descent? Which steps don’t?
- (b) How many parameters do we need to learn for the first layer? What if we also included a bias for each filter?
- (c) How many parameters do we need to learn for the second layer? What if we also included a bias for each filter?
- (d) How many parameters do we need to learn for the third (FC) layer? What if we also included a bias for each class?
- (e) Assuming we keep the biases for each layer, how many parameters total do we need to learn?