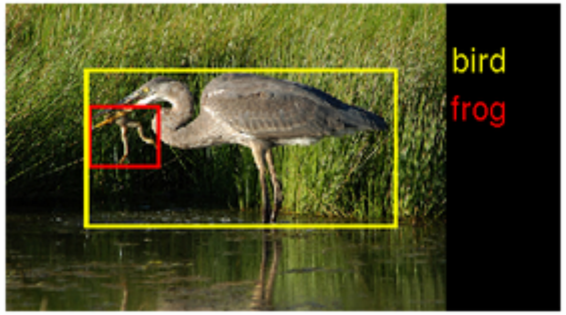
Deep Learning

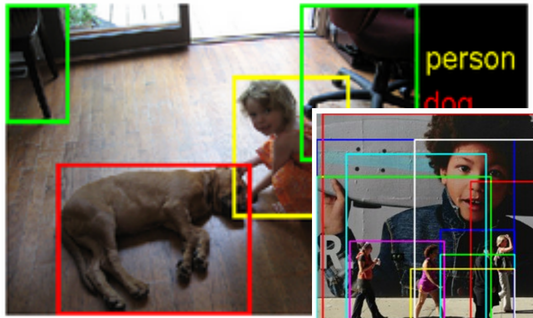
CMPT 733

Spring 2017

Apala Guha

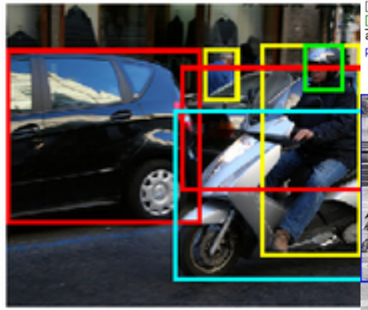


bird
frog



person

person
hammer
flower pot
power drill



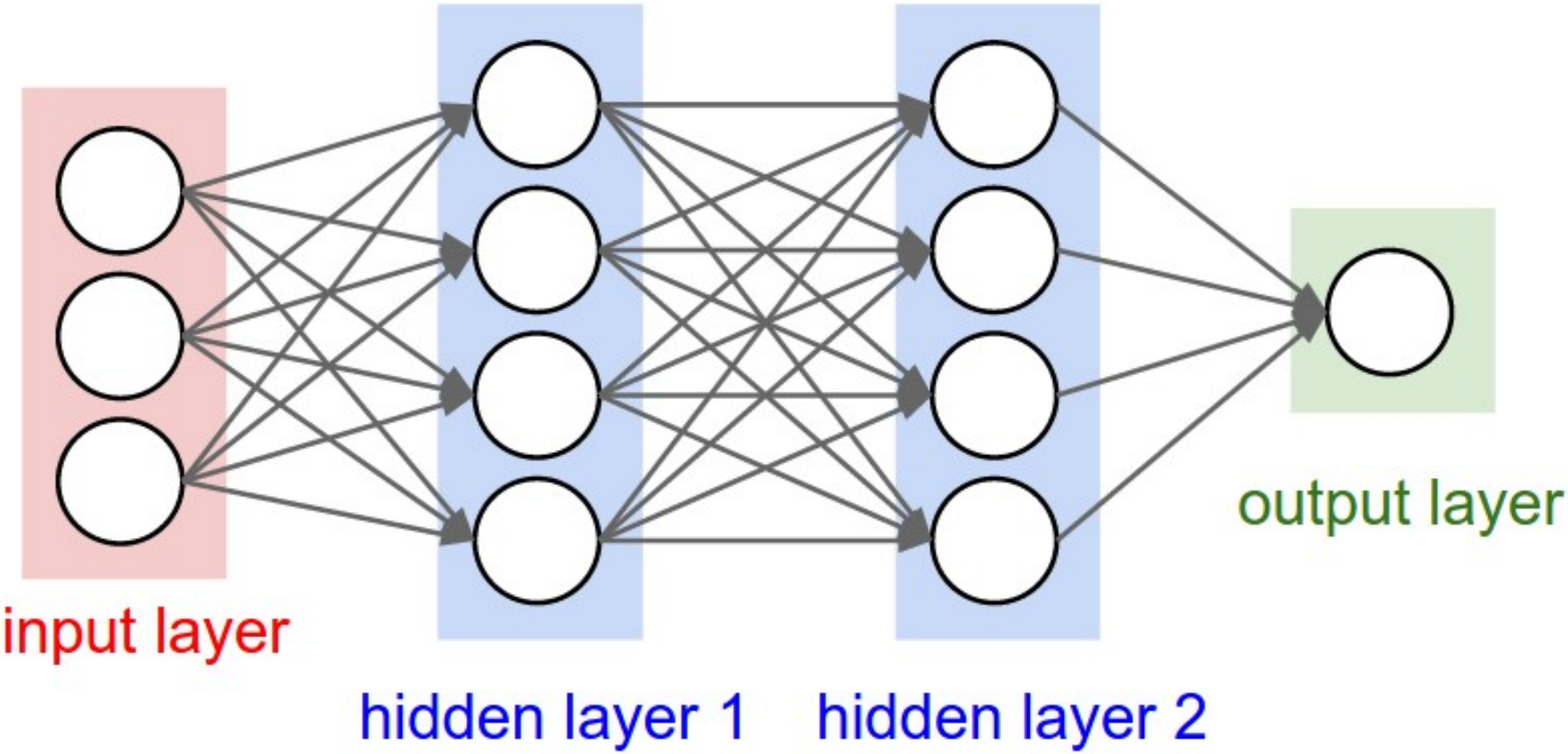
"Due to deep learning, we brought the vehicle's environment perception a significant step closer to human performance and exceeded the performance of classic computer vision."
Ralf G. Herrtwich
Director of Vehicle Automation, Daimler
NVIDIA DRIVE

[men (0.59)] [group (0.66)] [woman (0.64)] [people (0.89)] [holding (0.60)] [playing (0.61)] [tennis (0.69)] [court (0.51)] [standing (0.59)] [skis (0.58)] [street (0.52)] [man (0.77)] [skateboard (0.67)]
a group of people standing next to each other
people stand outside a large ad for gap featuring a young boy
[person (0.55)] [street (0.53)] [holding (0.55)] [group (0.63)] [slope (0.51)] [standing (0.62)] [snow (0.91)] [skis (0.74)] [player (0.54)] [people (0.85)] [men (0.57)] [skiing (0.51)] [skateboard (0.89)] [riding (0.75)] [tennis (0.74)] [trick (0.53)] [skate (0.52)] [woman (0.52)] [man (0.86)] [down (0.61)]
a group of people riding skis down a snow covered slope
a guy on a skate board on the side of a ramp
[airplane (0.57)] [plane (0.58)] [kites (0.93)] [people (0.80)] [flying (0.93)] [man (0.57)] [beach (0.84)] [wave (0.61)] [sky (0.61)] [kite (0.74)] [field (0.75)]
a couple of people flying kites in a field
people in a field flying different styles of kites
[parked (0.72)] [bench (0.63)] [truck (0.70)] [red (0.88)] [train (1.00)] [sitting (0.73)] [cars (0.58)] [traveling (0.52)] [grass (0.65)] [track (0.69)] [car (0.59)] [yellow (0.57)] [field (0.80)] [engine (0.56)] [down (0.54)] [tracks (0.94)]
a train traveling down train tracks near a field
a red train is coming down the tracks
[umbrella (0.59)] [woman (0.52)] [fire (0.96)] [hydrant (0.96)] [street (0.79)] [old (0.50)] [bench (0.81)] [building (0.75)] [standing (0.57)] [baseball (0.55)] [white (0.82)] [sitting (0.65)] [people (0.79)] [photo (0.53)] [black (0.84)] [kitchen (0.54)] [man (0.72)] [water (0.56)]
a black and white photo of a fire hydrant
a courtyard full of poles pigeons and garbage cans also has benches on either side of it one of which shows the back of a large person facin g in the direction of the pigeons
[horse (0.53)] [bear (0.71)] [elephant (0.99)] [elephants (0.95)] [brown (0.68)] [baby (0.62)] [walking (0.57)] [laying (0.61)] [man (0.57)] [standing (0.79)] [field (0.65)] [water (0.83)] [large (0.71)] [dirt (0.65)] [river (0.58)]
a baby elephant standing next to each other on a field
elephants are playing together in a shallow watering hole
[man (0.59)] [beach (0.54)] [sky (0.53)] [bird (0.50)] [field (0.88)] [snow (0.86)] [mountain (0.59)] [standing (0.81)] [white (0.64)] [people (0.51)] [dog (0.60)] [cows (0.55)] [sheep (0.97)] [black (0.84)] [grass (0.64)] [horse (0.60)] [elephants (0.57)] [bear (0.81)]
a black bear standing on top of a grass covered field
a couple of sheep standing up on a small hill
[bus (0.56)] [car (0.79)] [black (0.57)] [truck (0.86)] [street (0.57)] [bed (0.51)] [parked (0.55)] [dog (0.65)] [sitting (0.55)] [man (0.53)] [cat (0.72)]
a dog sitting on top of a car
a cat is lying on the hood of a black car
[street (0.89)] [truck (0.76)] [road (0.58)] [fire (0.95)] [hydrant (0.91)] [sitting (0.53)] [black (0.51)] [red (0.53)] [parking (0.69)] [parked (0.82)] [sign (0.78)]
a fire hydrant on the side of a road
two signs with arrows pointing to each other for detour
a man doing a trick on a skateboard
a skateboarder is is mid air performing a stunt
[monitors (0.56)] [laptop (0.97)] [table (0.74)] [open (0.71)] [sitting (0.61)] [station (0.52)] [desk (0.97)] [computer (0.94)] [keyboard (0.68)] [computers (0.65)] [tv (0.54)] [television (0.50)] [monitor (0.69)]
an open laptop computer sitting on top of a desk
two computers are shown together on a desk
a baseball player swinging a bat at a ball
a boy is playing with a baseball bat

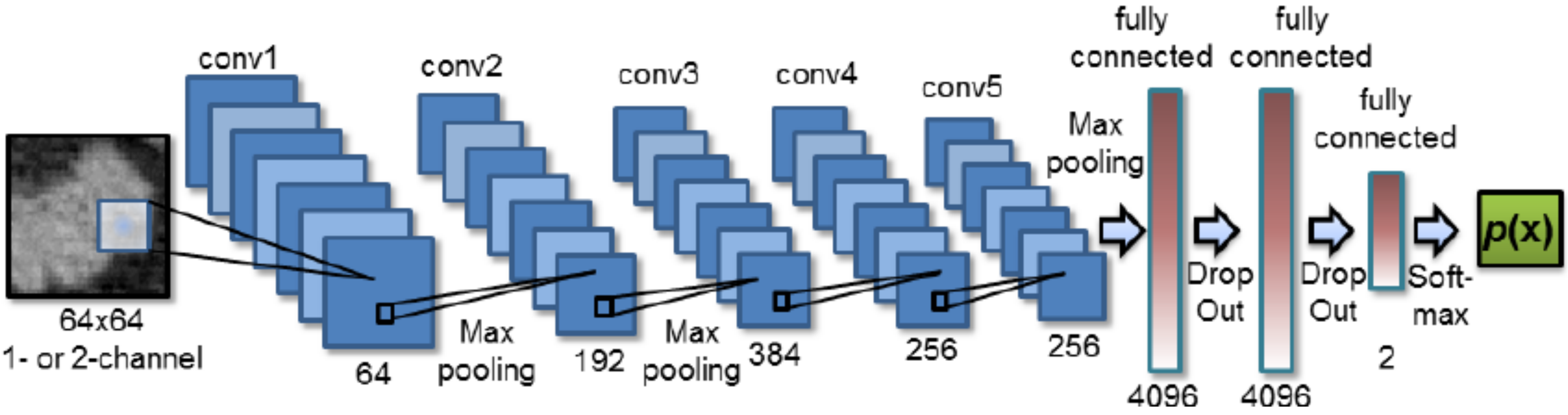
Deep Learning Platforms

- Caffe – BVLC
- AlexNet – U Toronto
- convNet.js – browser-based
- CaffeonSpark - Yahoo
- TensorFlow - Google
- DaDianNao
- TrueNorth – IBM
- CuDNN – Nvidia
- DeepLearning4j, Keras, Theano

Neural Networks



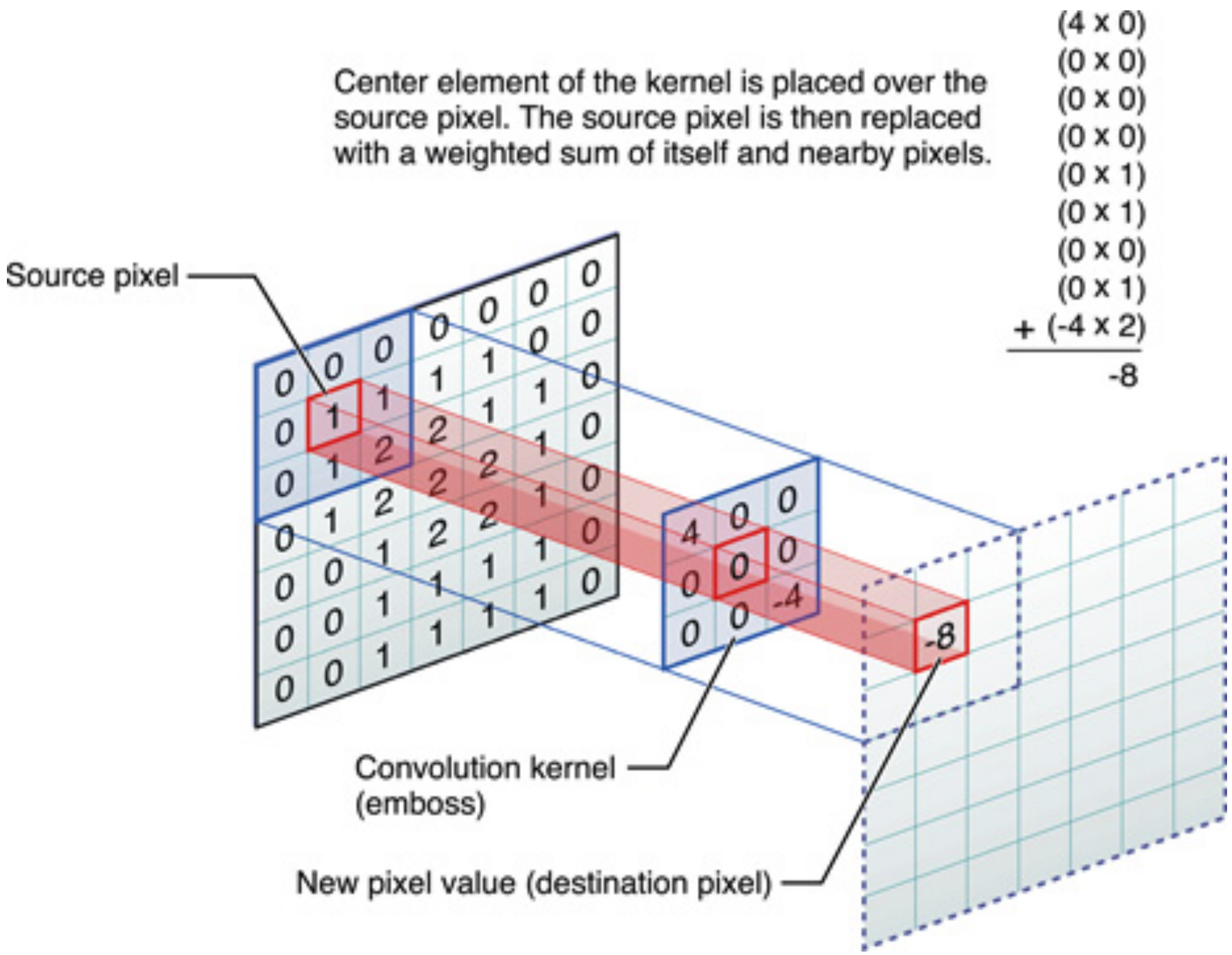
Convolutional Neural Networks



Convnet Layers

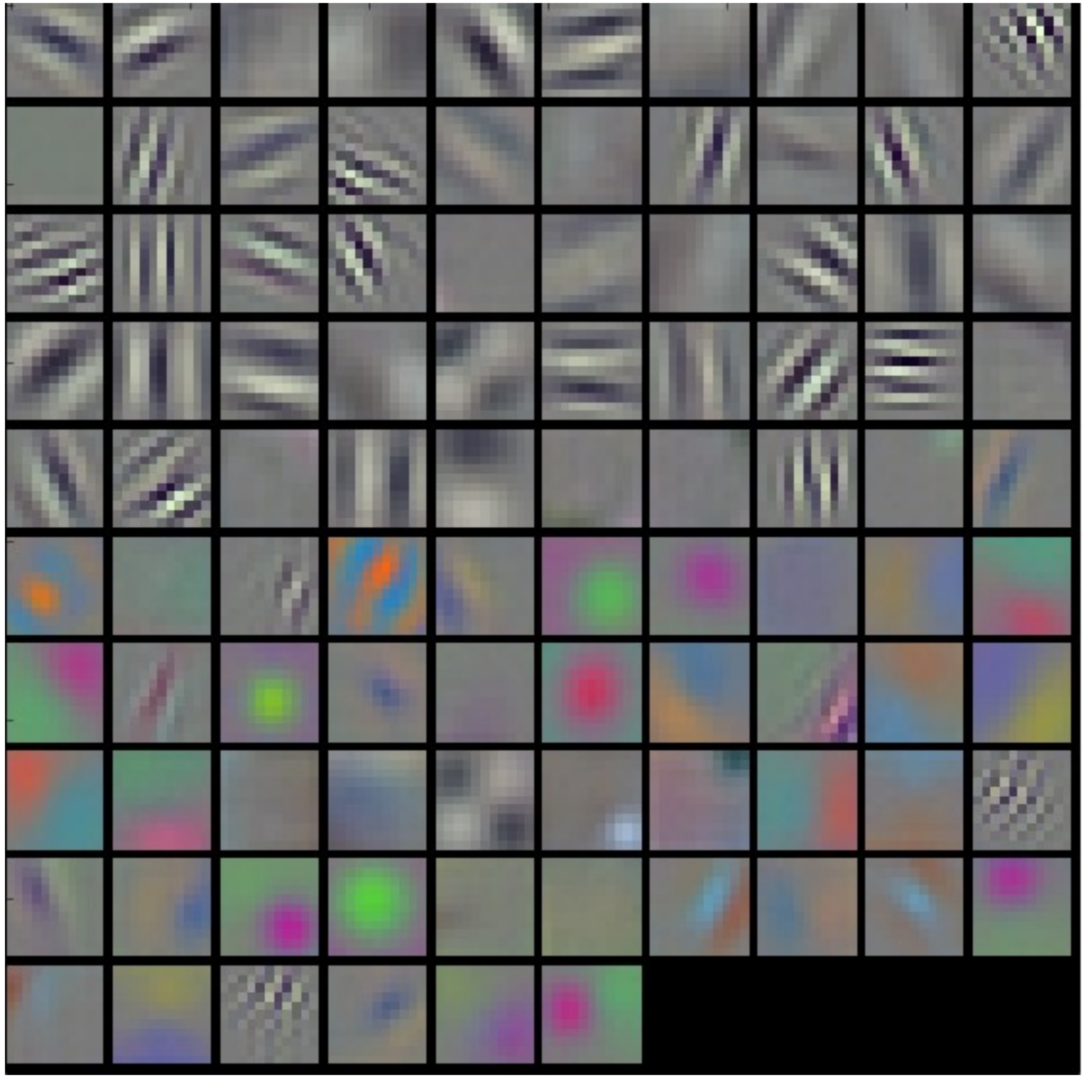
- convolution layer – detects input features
- pooling layer – subsamples input
- fully connected layer – makes classification decisions as a whole

Convolution Layer

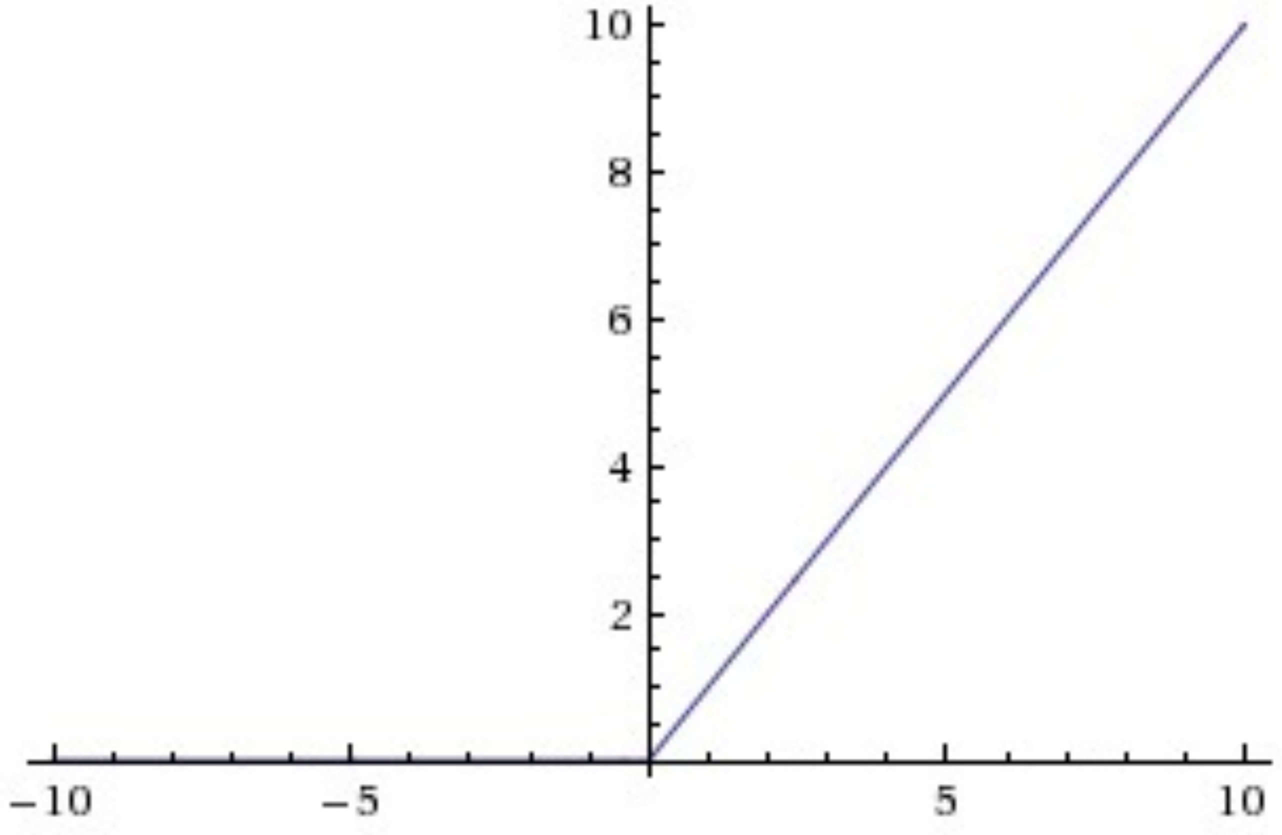


- Kernel forms dot products with different regions of input and produces activation maps
- Kernel may skip over rows and columns *(striding)* when sliding over input
- Kernel = neuron => neurons with same weights consume different areas of input

Visualizing convolution layer - Kernels

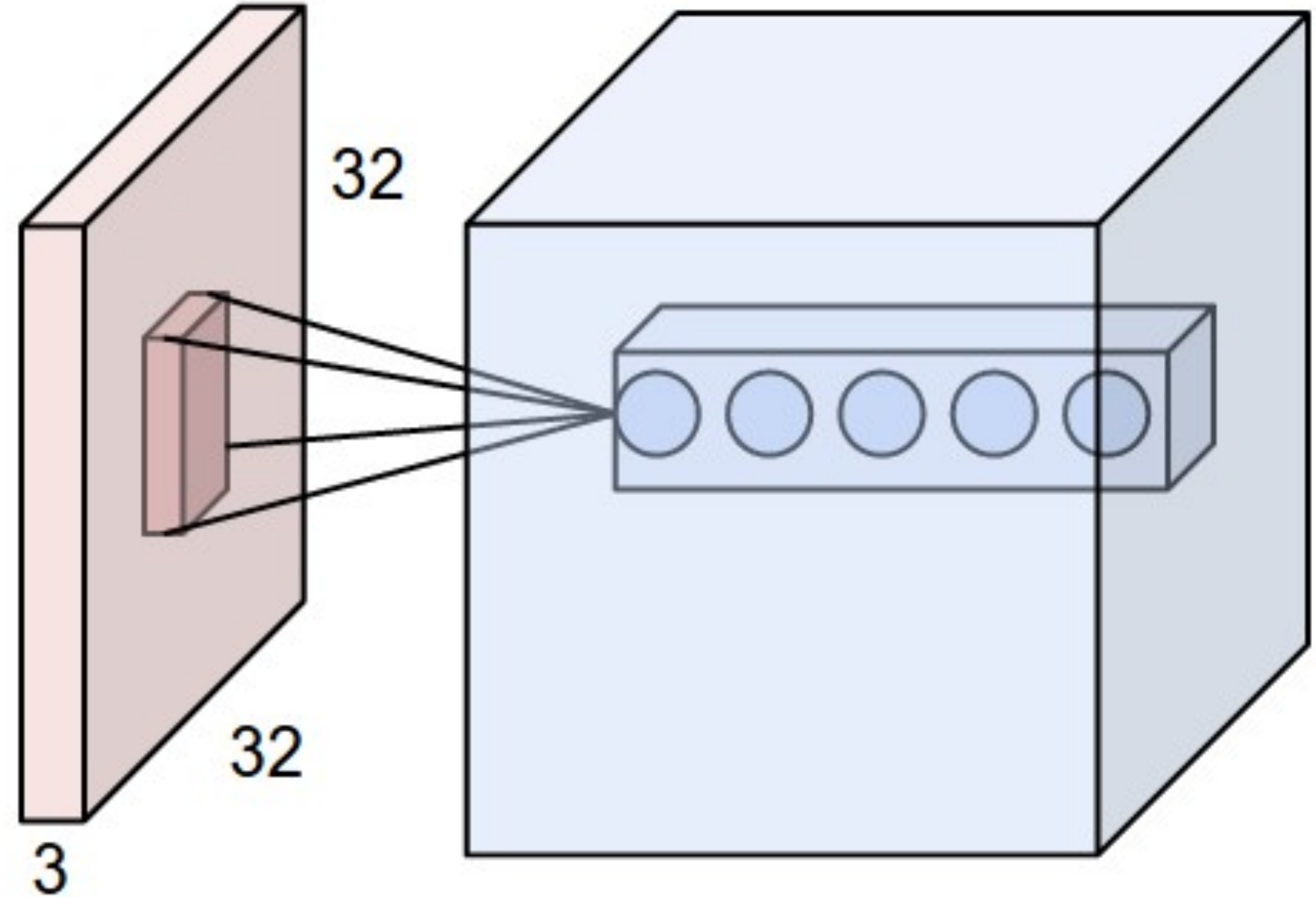


Convolution Layer



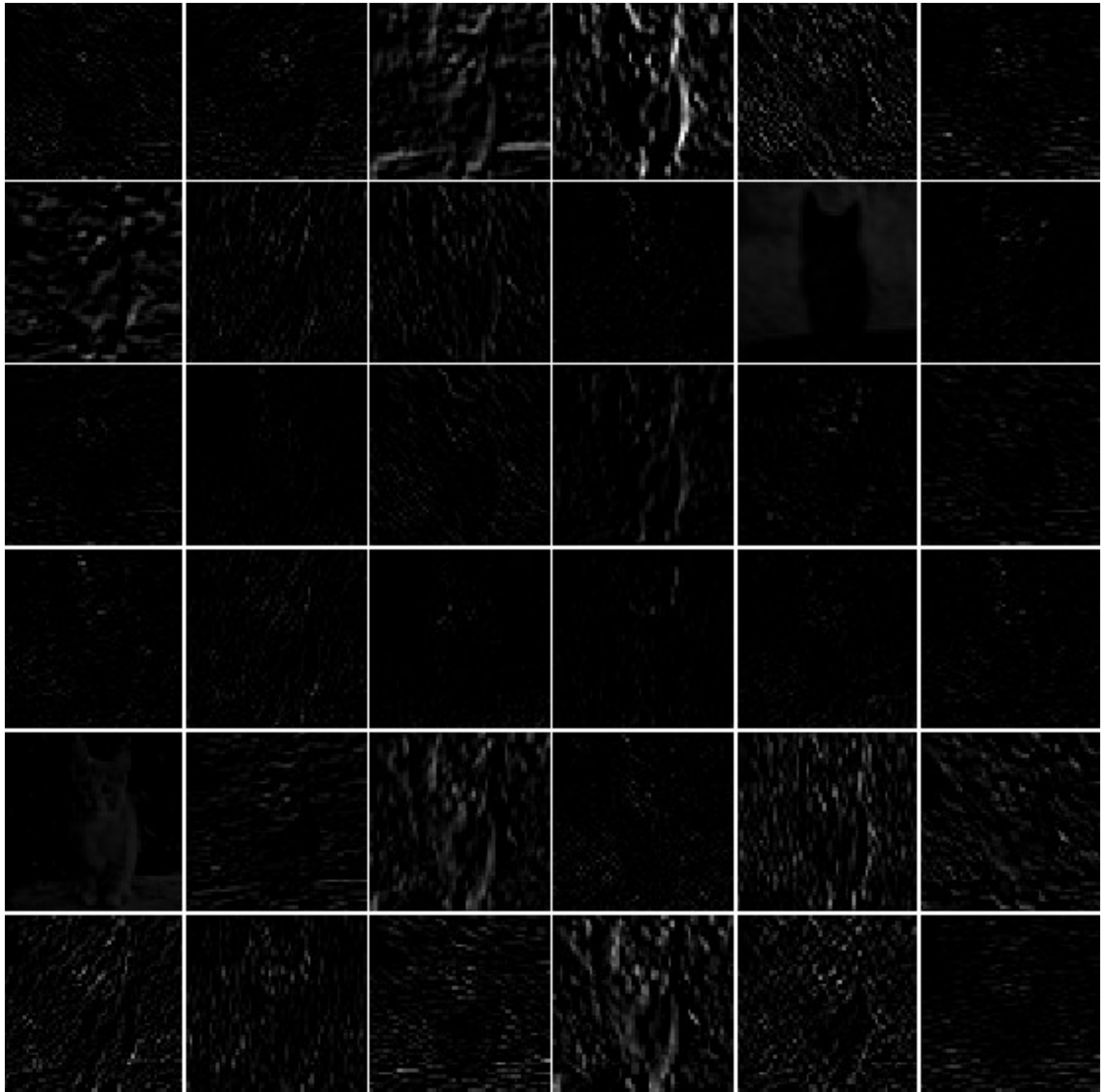
- Activation is a measure of the presence of the kernel feature at a specific location in the input
- Typically we are only interested in positive activation values
- ReLU is applied to activation maps to suppress negative activation

Convolution Layer

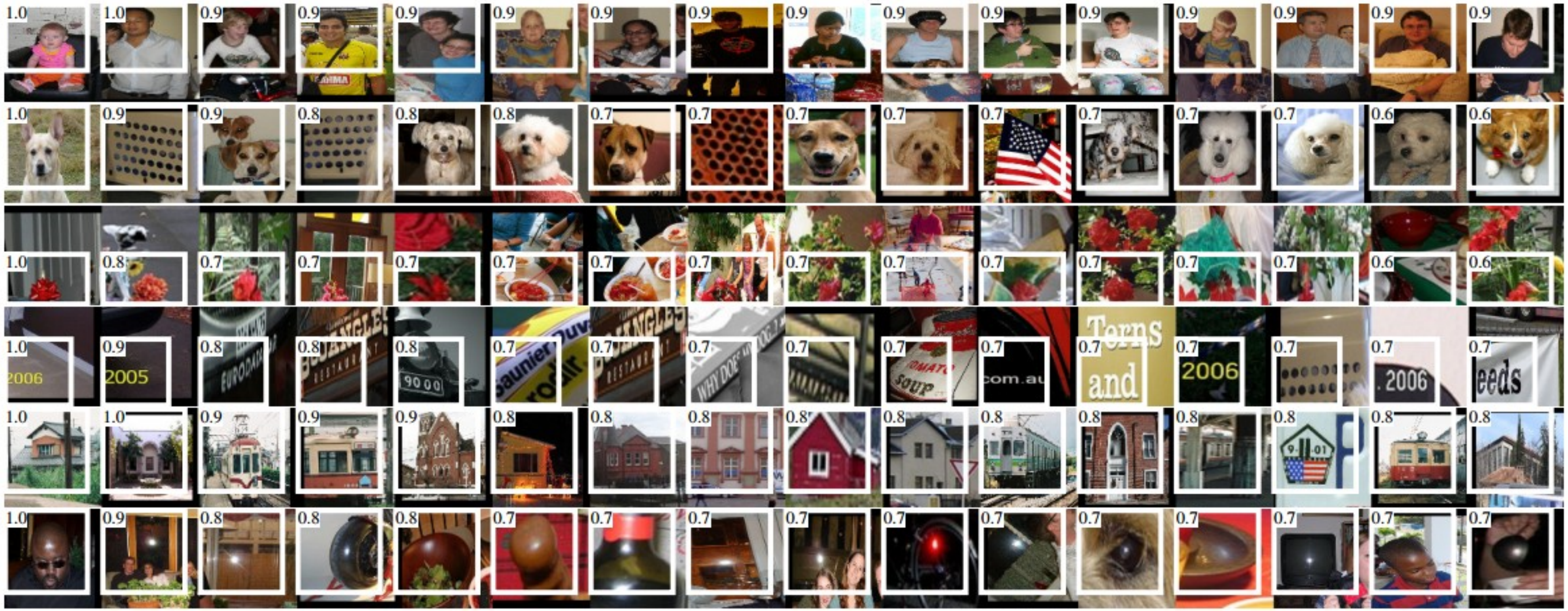


- Inputs have depth (e.g. color channels at lowest layer)
- Kernel convolves through the input depth
- The depth of inputs to higher layers is due to multiple kernels

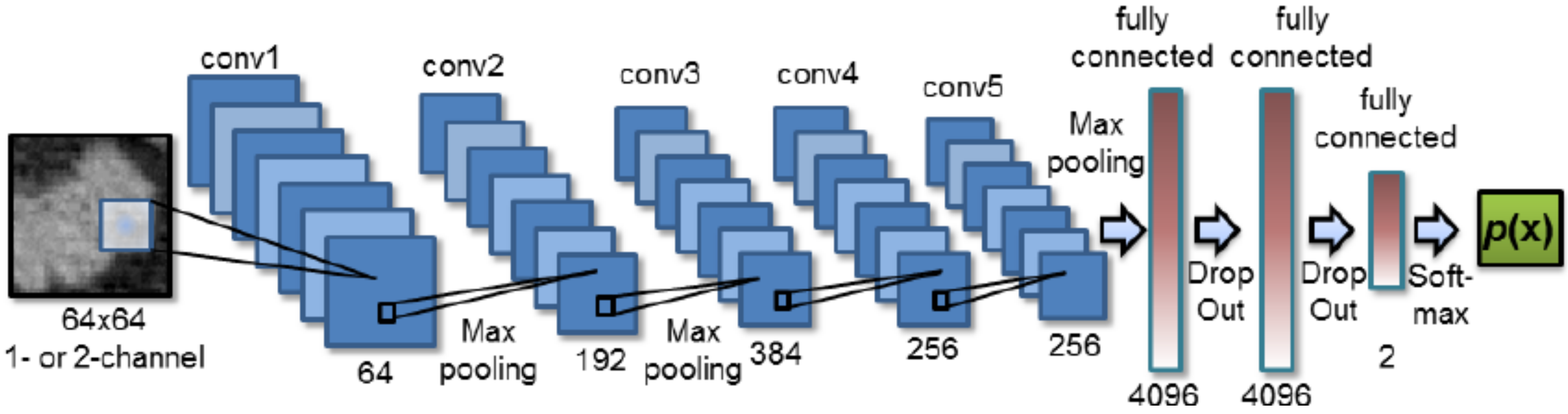
Visualizing convolution layer - activation



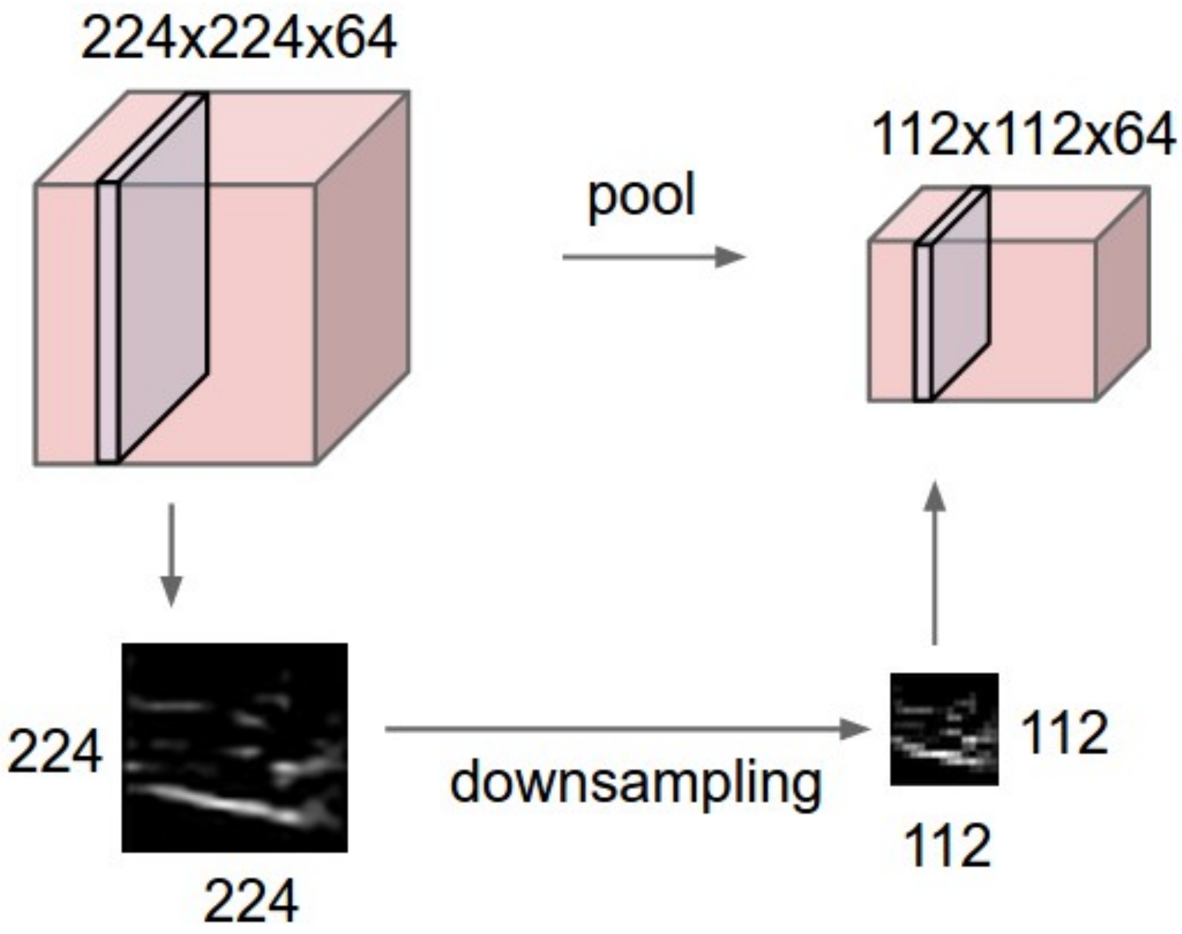
Visualizing convolution layer
– images that most activate a kernel



Convolutional Neural Networks

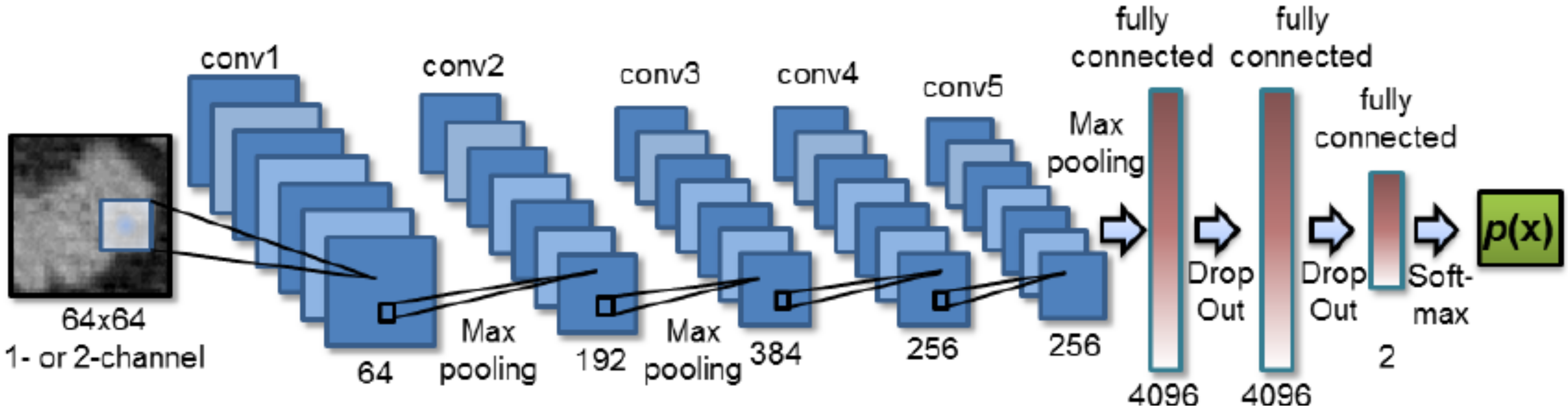


Pooling Layer



- Reduces the size of the output of convolution layers
- Different types possible
  - max pooling
  - average pooling
  - sum pooling

Convolutional Neural Networks



Fully connected layer

- Similar to convolution layer but processes entire input together
- Form the last few layers where they consider all combinations of features to make a final classification
- Sigmoid function ($1 / (1 + e^{-z})$) is applied at the output of FC layer to map prediction to (0, 1) probabilities e.g. (0.3, 0.5, 0.2, 0.4)
- Softmax ($e^z / \Sigma e^z$) function is then applied to ensure that the probabilities add up to 1 e.g. (0.1, 0.1, 0.7, 0.1)

Learning Task

- In the training phase we learn the kernels
- Each weight in a kernel has a gradient on the final cost and is updated in the negative direction of the gradient using a learning rate
- Simple example

Learning task

- Let kernel = g_θ, where θ is a vector of weights
- Assuming a really simple CNN, conv -> ReLU -> pool -> FC
- final prediction = softmax(conv(pool(ReLU(g_θ(x)))))
  = (0.1, 0.5, 0.1, 0.3), for example for 4 classes
  where x = input image
- We want predictions that are more confident e.g. (0.05, 0.8, 0.05, 0.5)
- In the training phase, we have examples of the form (0.0, 1.0, 0.0, 0.0)
- overall cost = sum of the cost of misprediction for each class

Learning Task

- Cost = $-(1/N)\ \Sigma_i\ [\ y_i \log p_i + (1 - y_i)\log(1-p_i)\ ]$
  - y_i = belongs to class i or not => {0, 1}
  - p_i = predicted probability of belonging to class i
  - Taking a log of the prediction rapidly increases cost as it moves away from correct answer

- δ Cost / δ θ = δ cost(softmax(conv(pool(ReLU(g_θ(x)))))) / δ θ
  = (δ cost(r) / δ r) * (δ r / δ θ)
  = (δ cost(r) / δ r) * (δ softmax(s) * δ s) * (δ s / δ θ)
  = …

- Note that ReLU and pool are non-differentiable

- $\theta = \theta$ – (learning rate) * δ Cost / δ θ