Recurrent and Convolutional Neuro Network

WEEK 4

Convolutional Neural Network

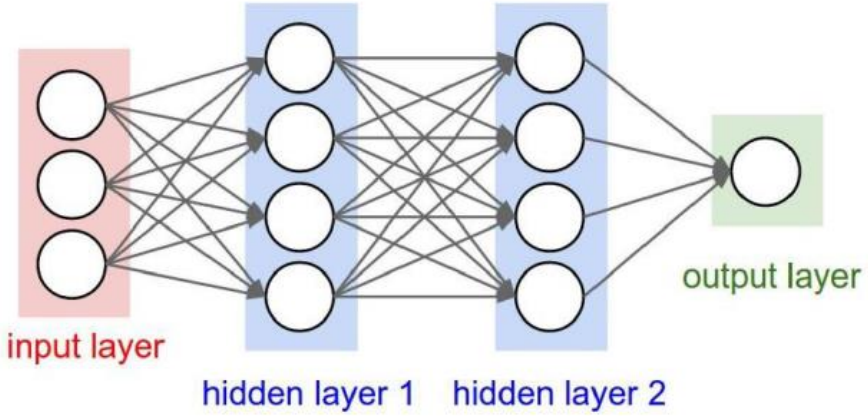
Acknowledgements

(All sources are great reading materials for learning convolutional NNs).

Some of the images, screenshots, quotations, and other properties are extracted from one of the following sources:

- Lecture Notes, CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University. http://cs231n.github.io/convolutional-networks/
- https://en.wikipedia.org/wiki/Convolutional_neural_network
- Illustrational Video, DeepLearning.TV. https://www.youtube.com/watch?v=JiN9p5vWHDY

Convolutional Neural Network



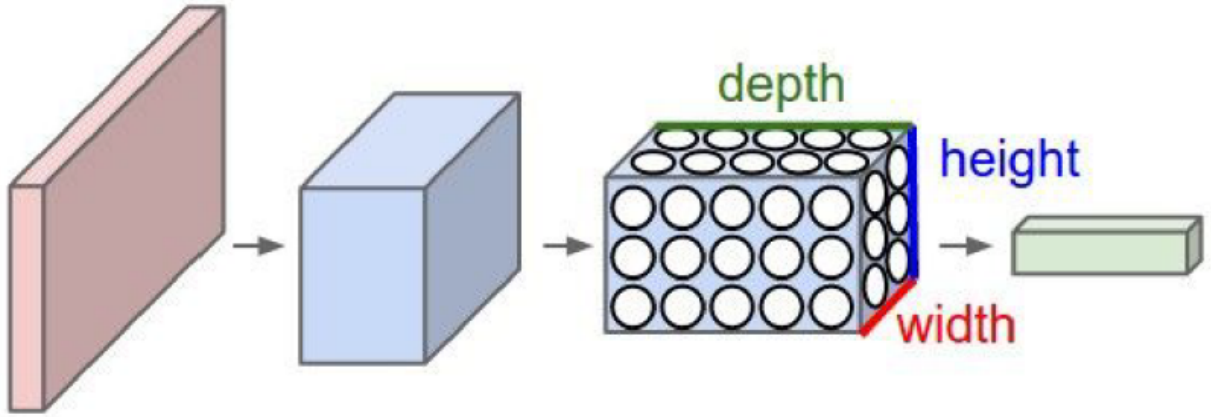
Conventional Feedforward Neural Network

Problems:

- Ignores the local structure of the images.
- Does not scale well with the image size.

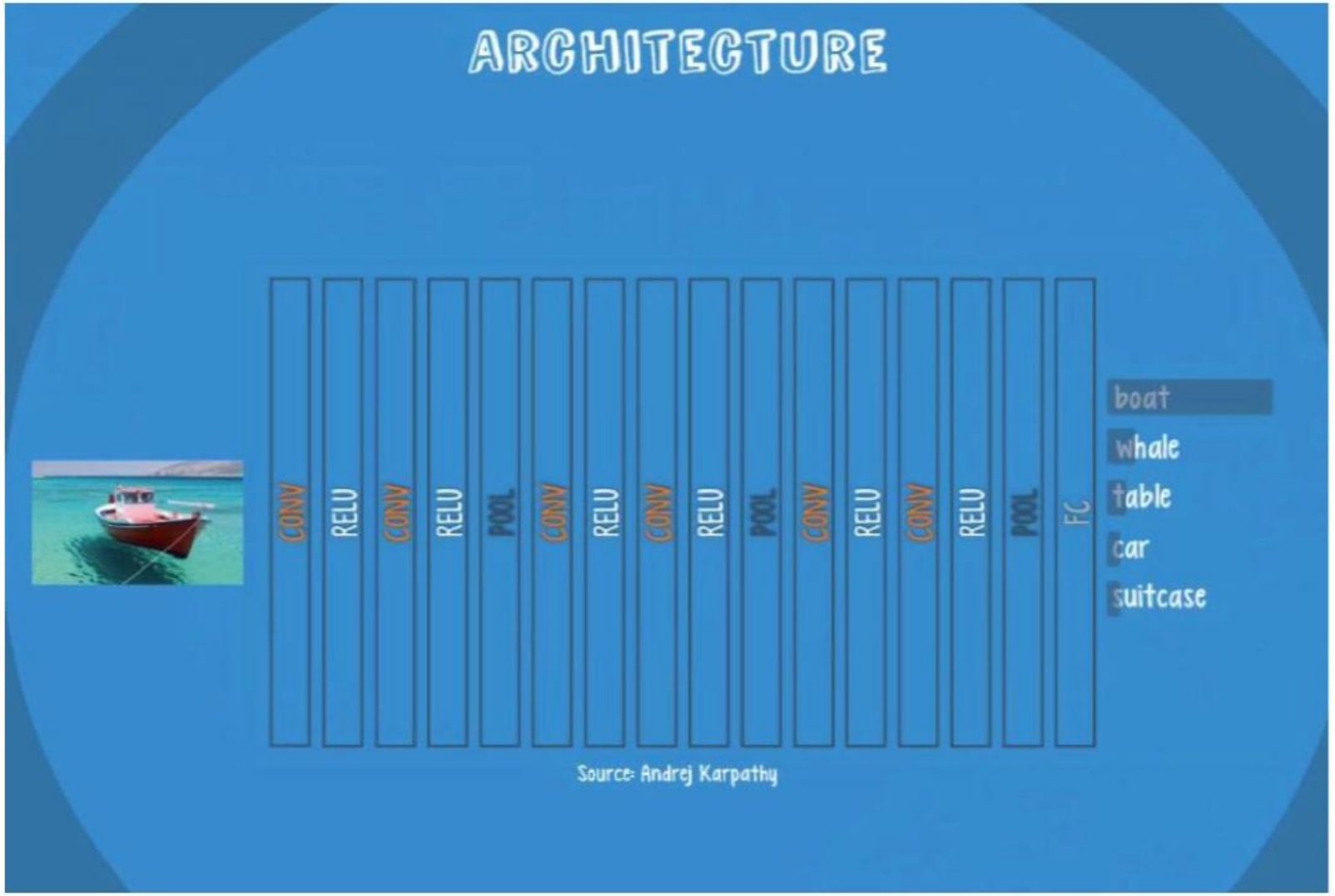
Convolutional Neural Network

- 3d volume of neurons:
  - Width, Height, Depth.
- Layers:
  - Convolutional layer;
  - RELU layer;
  - Pooling layer;
  - Fully connected layer;

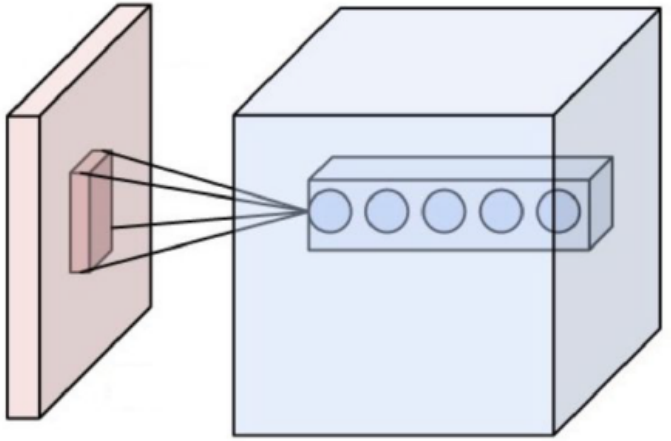


- Typical Layered Structure:
  - INPUT - [(CONV - RELU) * m - POOL] * n - FC

Convolutional Neural Network: Architecture Illustration



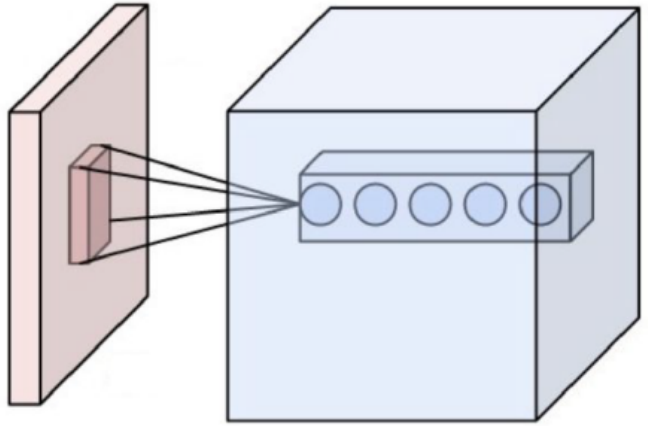
Convolutional Neural Network Structures



Convolutional Layer (Core structure)

- Set of learnable filters.
- Local connectivity
  - Assumes that local features around any unit is important.
  - Hyperparameter: Receptive field (filter size).
- Dimension Control
  - Depth: Neuron along the depth learn different features local features.
  - Stride: Spacing between columns.
  - Zero padding: Control edge (border size) of the image.
- Parameter Sharing.

Convolutional Neural Network Structures



Convolutional Layer (Core structure)

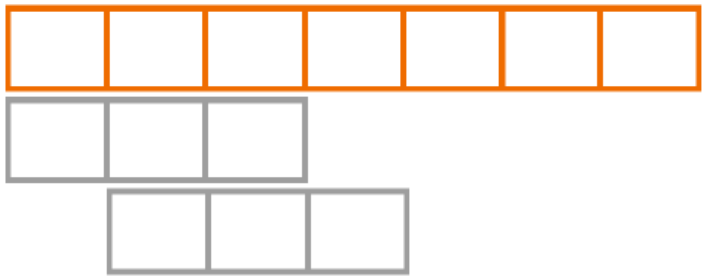
Input volume size W,

- Receptive field size F,
- Stride S,
- Amount of zero padding used P.

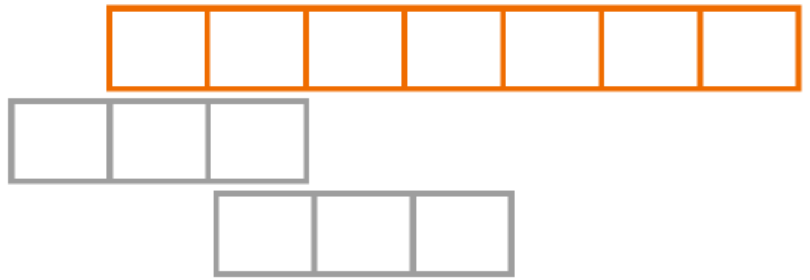
Output volume: $(W-F+2P)/S+1$

Eg. (1) for a 7x7 input and a 3x3 filter with stride 1 and pad 0 we would get a 5x5 output.
(2) With stride 2, 1 padding we would get a 4x4 output.

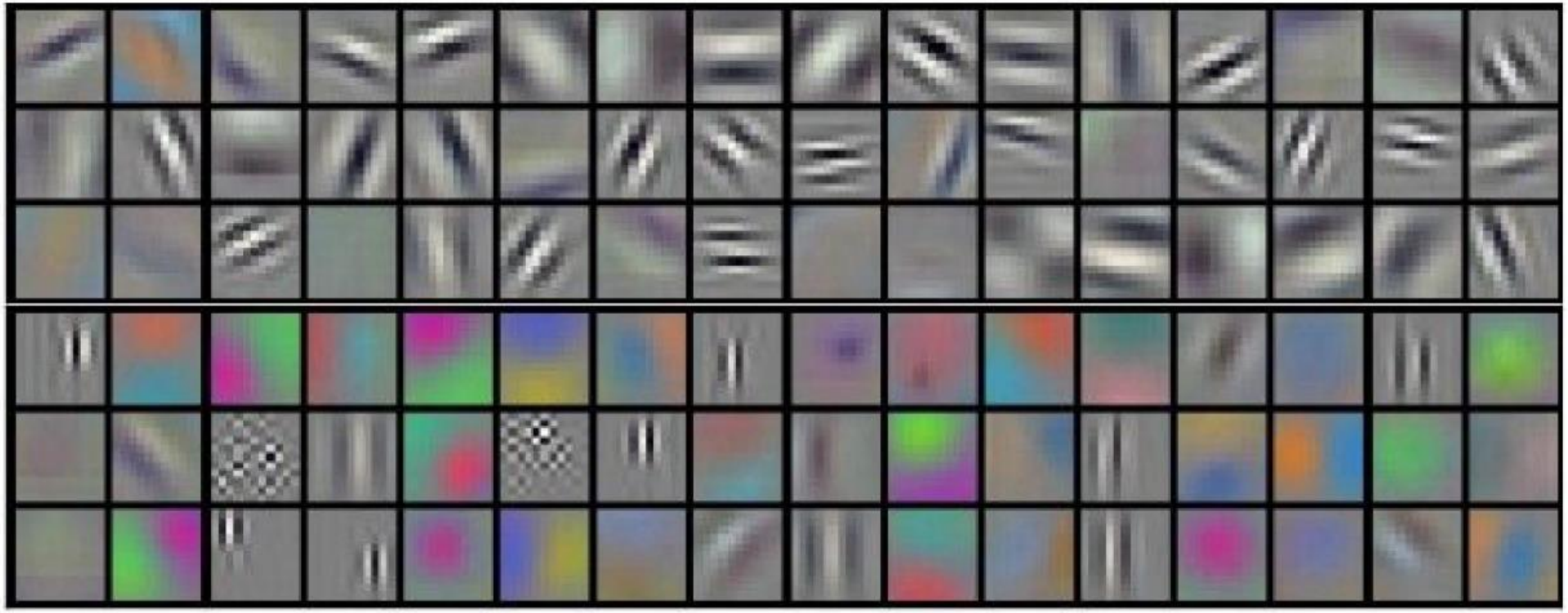
Eg (1)



Eg (2)



Convolutional Neural Network Structures



Example of the learned Convolutional Layer Filters

Convolutional Neural Network Structures

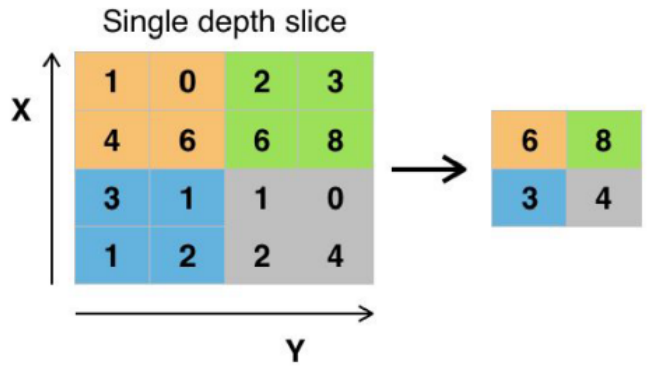
RELU (Rectified Linear Units)

- Add a set of activation function to the filter output
- Many types of activation function
  - x = max(0, x) is the most common function
  - Alternative: sigmoid(x), tanh(x)

Convolutional Neural Network Structures

Pooling Layer

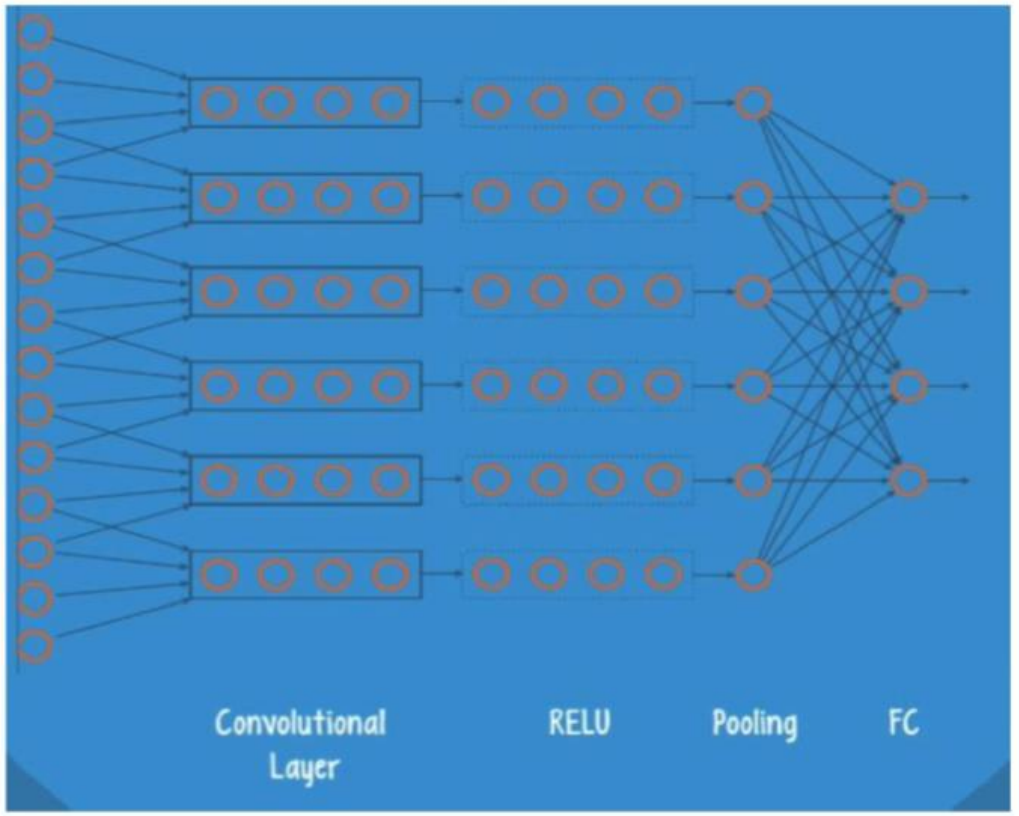
- Performs a non-linear down sampling procedure.
- Used to (1) Reduced the width and height of the layers.
  (2) Add more non-linearity to the model.
- Max pooling is the most common pooling scheme.
  - Intuition: If some features are found, the locations don't matter that much.
  - Alternatives: Average Pooling
- Most used 2*2 with Stride = 2.
  - Reduce 75% size.
  - Alternative 3*3
- Is a special type of non-linear convolutional filter.



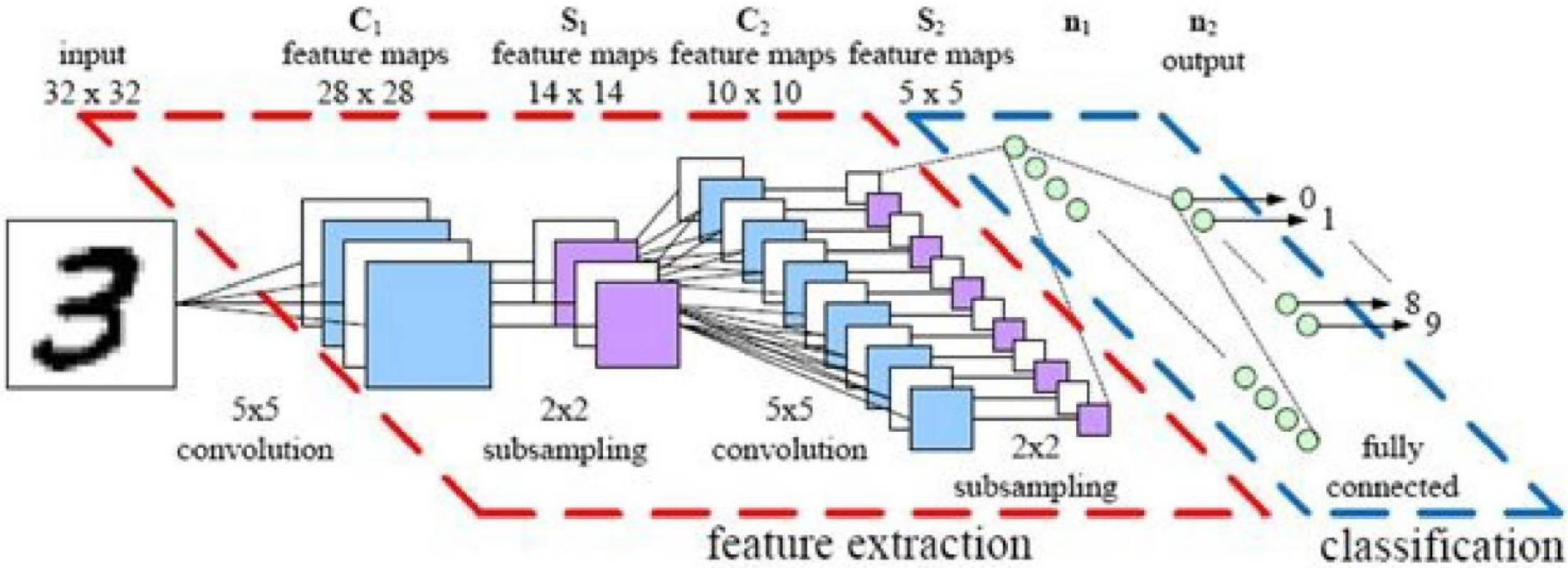
Convolutional Neural Network Structures

Fully Connected Layer

- Is a special type of convolutional layer.
  - Difference: local connected neuron vs all neurons

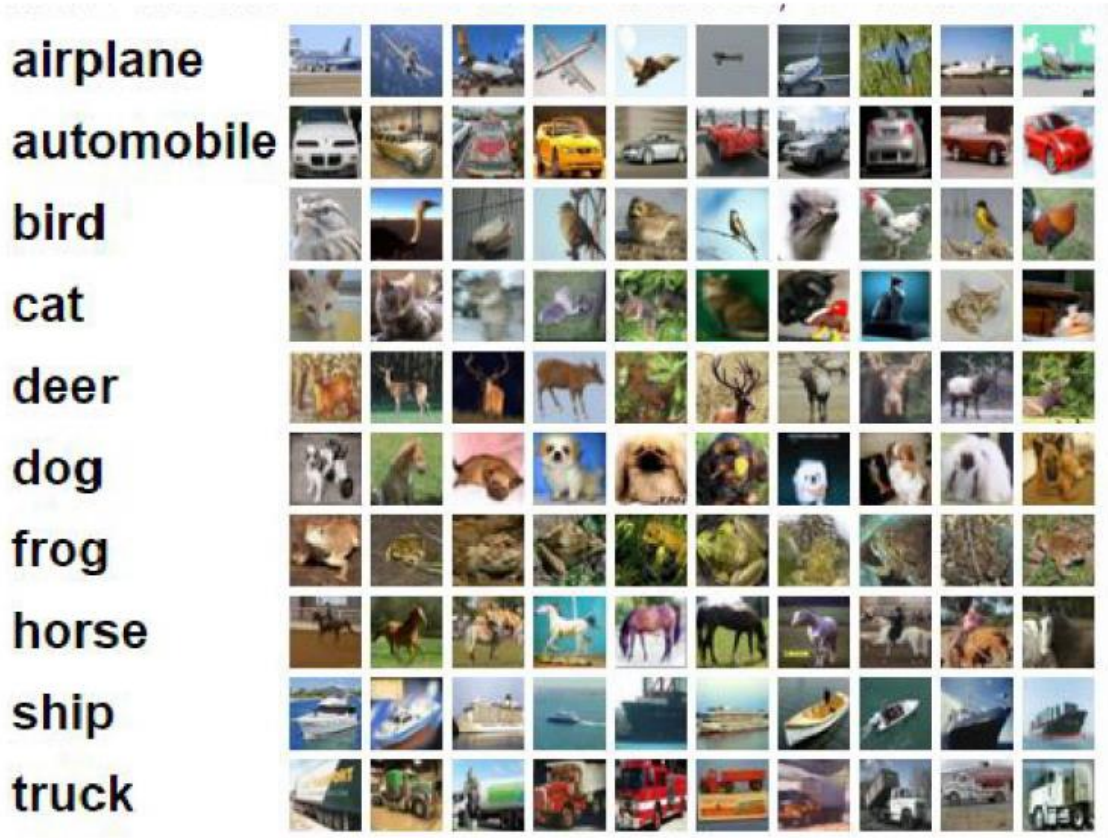


Convolutional Neural Network Structures

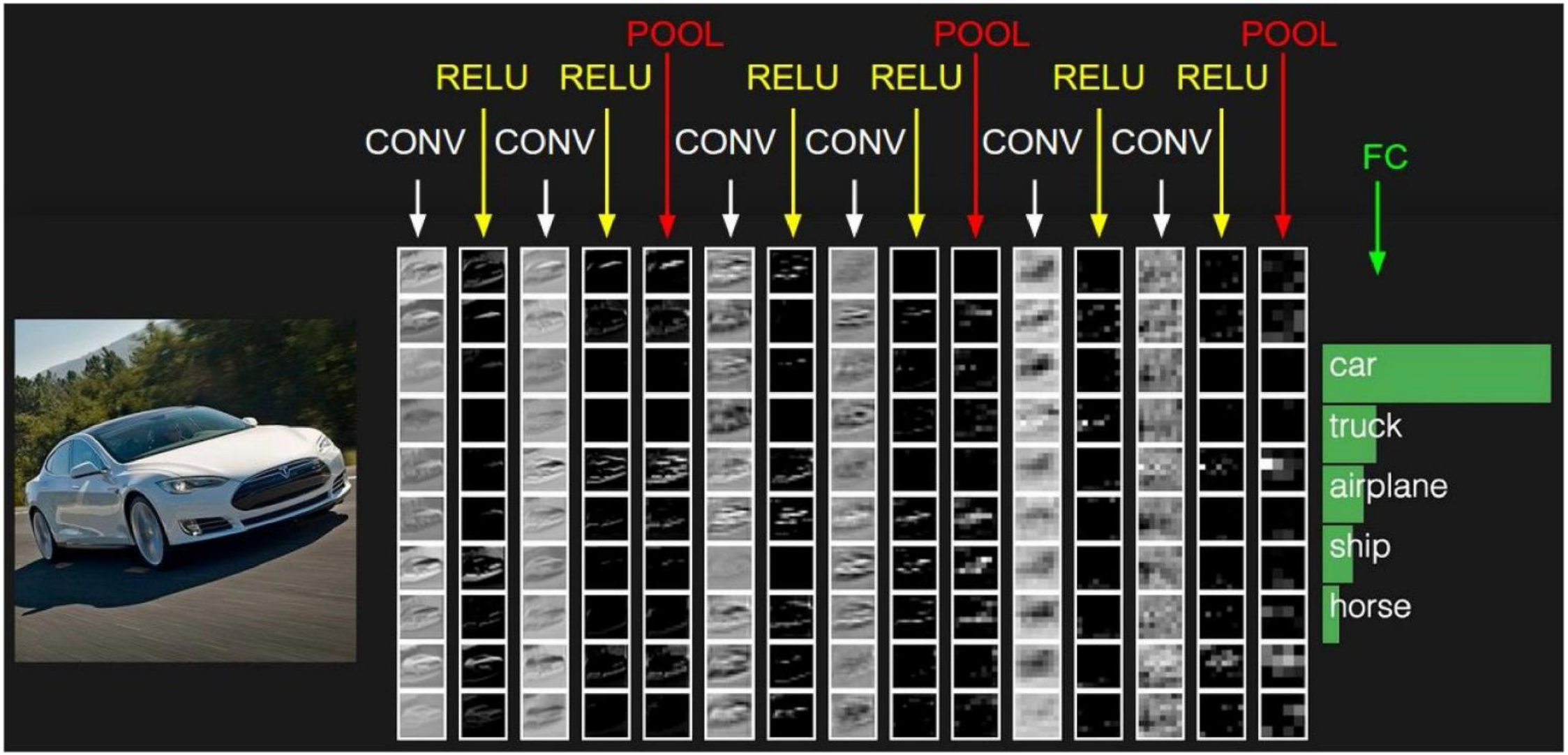


Convolutional Neural Network: Example

- CIFAR-10 Dataset
- 60000 images
- 32x32 pixels
- RGB coloured
- 10 classes
- https://www.cs.toronto.edu/~kriz/cifar.html



Convolutional Neural Network: Example



Recurrent Neuro Network

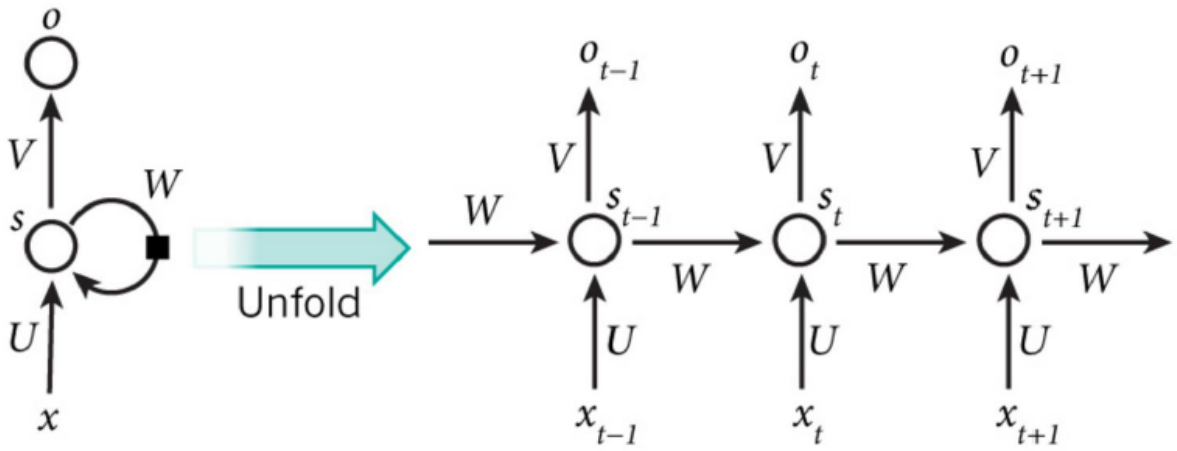
List the alphabet forward....

List the alphabet backward....

Try with the lyrics of a song you know?....

Why is it easier to recall forward than it is to recall backward?

Can you jump into the middle of the second verse?...



Recurrent Neuro Network

(input + empty_hidden) -> hidden -> output
(input + prev_hidden) -> hidden -> output
(input + prev_hidden) -> hidden -> output
(input + prev_hidden) -> hidden -> output

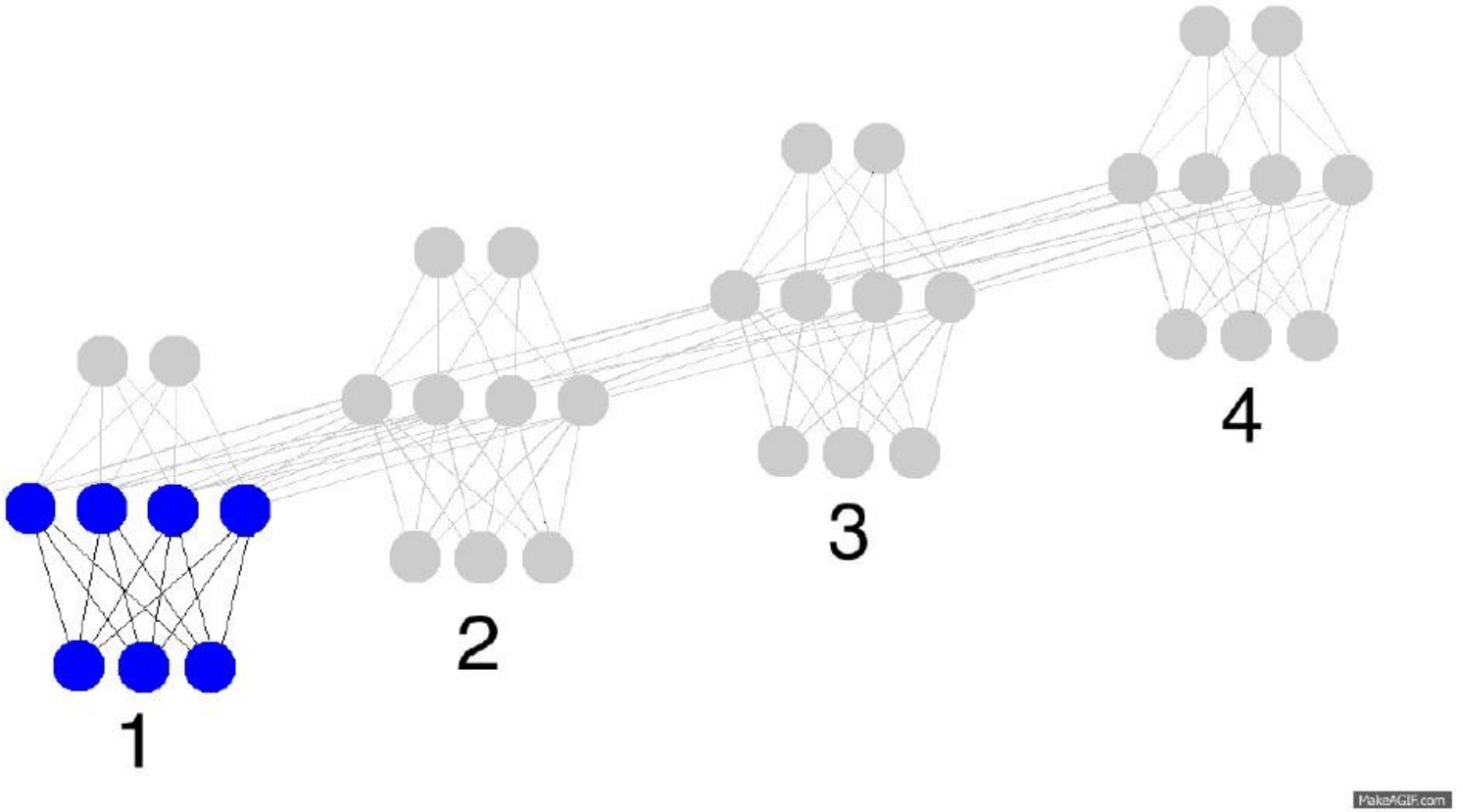
Input Layer Recurrence

V.S

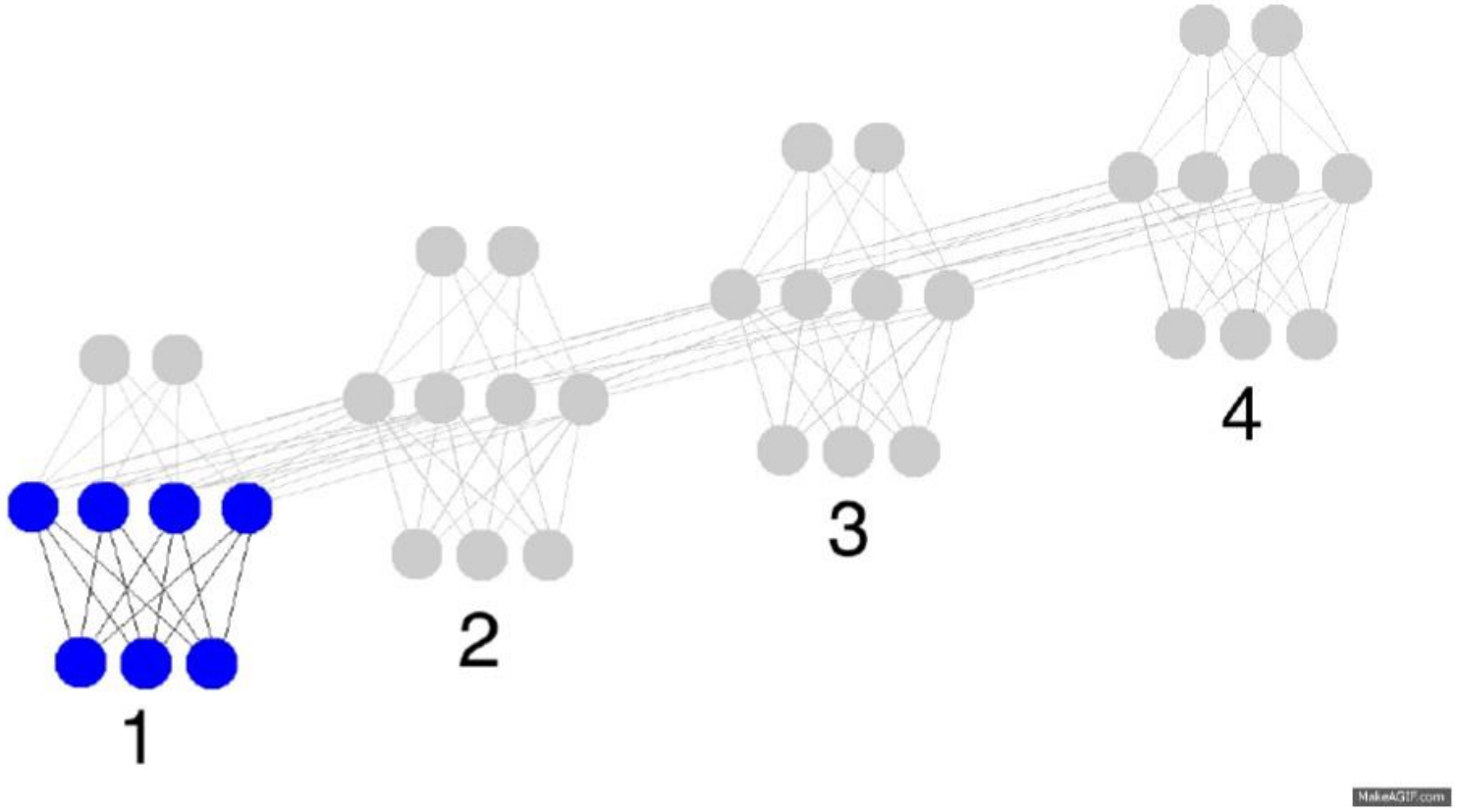
Hidden Layer Recurrence

(input + empty_input) -> hidden -> output
(input + prev_input) -> hidden -> output
(input + prev_input) -> hidden -> output
(input + prev_input) -> hidden -> output

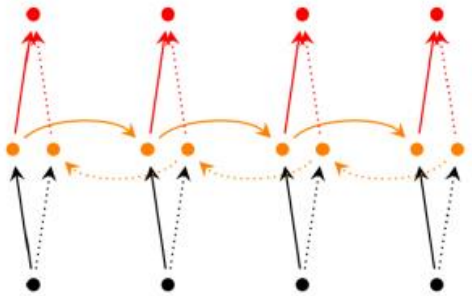
Forward Propagation



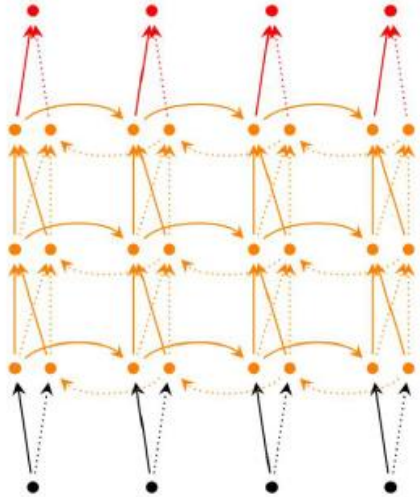
Backward Propagation



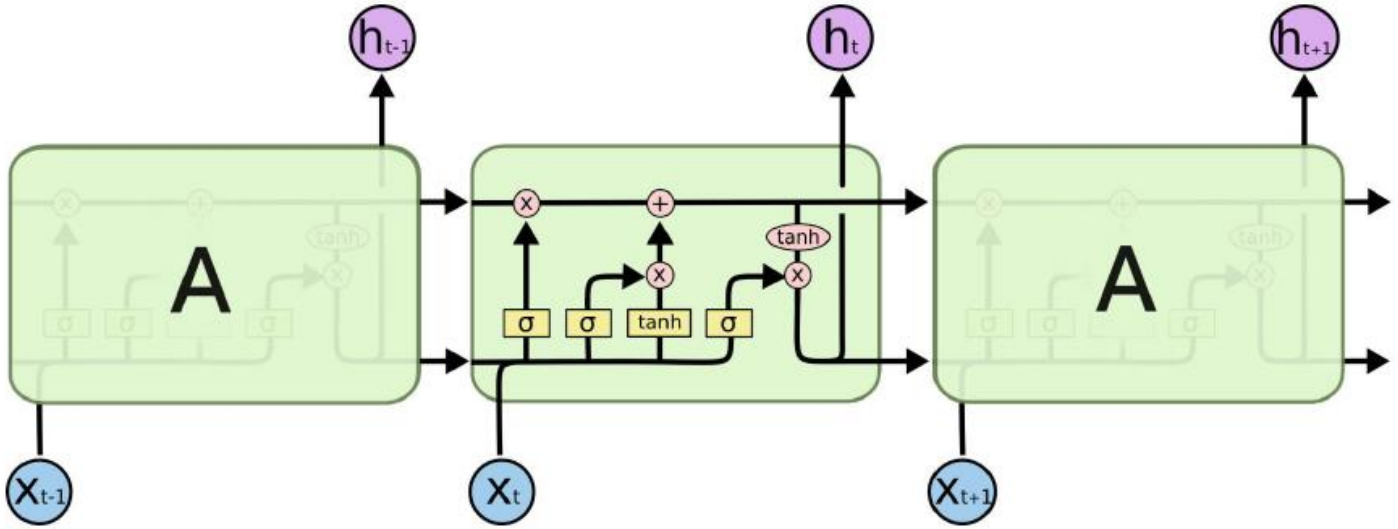
RNN Extensions



Bidirectional RNN

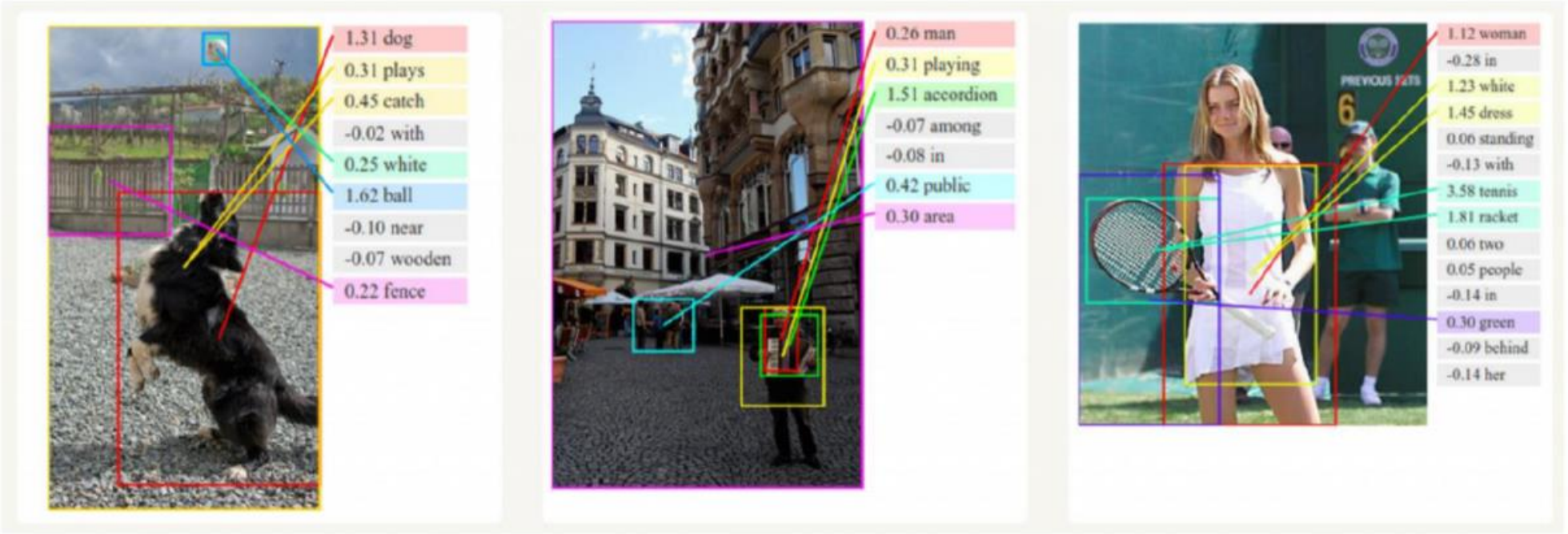


Deep Bidirectional RNN



Long Short Term Memory (LSTM) RNN

Generating Image Descriptions



Deep Visual-Semantic Alignments for Generating Image Descriptions. Source:

http://cs.stanford.edu/people/karpathy/deepimagesent/

RNN Example: Binary Addition

$$
\begin{array}{rll}
 & 11111111 & =-1 \\
+ & & \\
 & 11111110 & =-2 \\
\hline
= & &
\end{array}
$$