Convolutional and Recurrent Neural Networks

Data Science Decal

Hosted by Machine Learning at Berkeley

-

ML@B

Agenda

Background

Convolutional Neural Networks

Recurrent Neural Networks

Questions

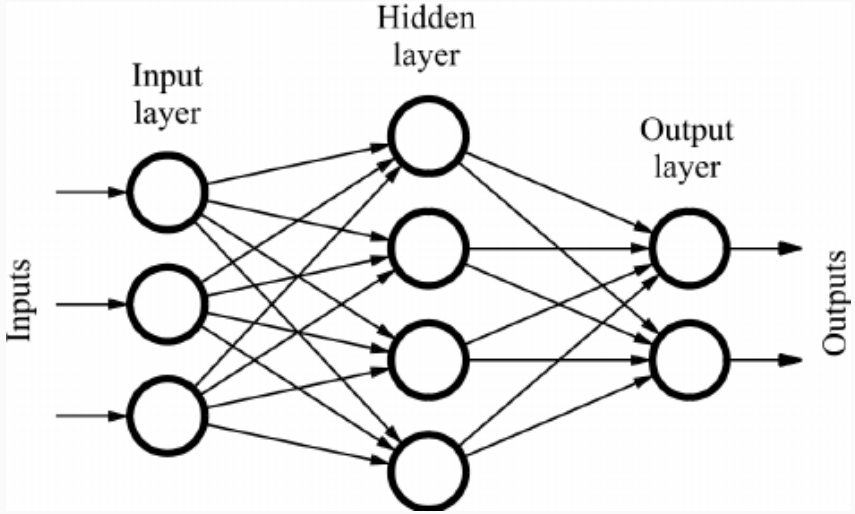
Demos

Background

Going beyond feed-forward neural networks

ML@B

- Where we left off:



- Why would such a network not be good for certain tasks?
- How does a neural network deal with complex or high-dimensional data such as images and text?
- ... if we have two hidden layers with 10,000 neurons each,

How are neural networks being used in industry?

Problem: high-dimensional image data

ML@B

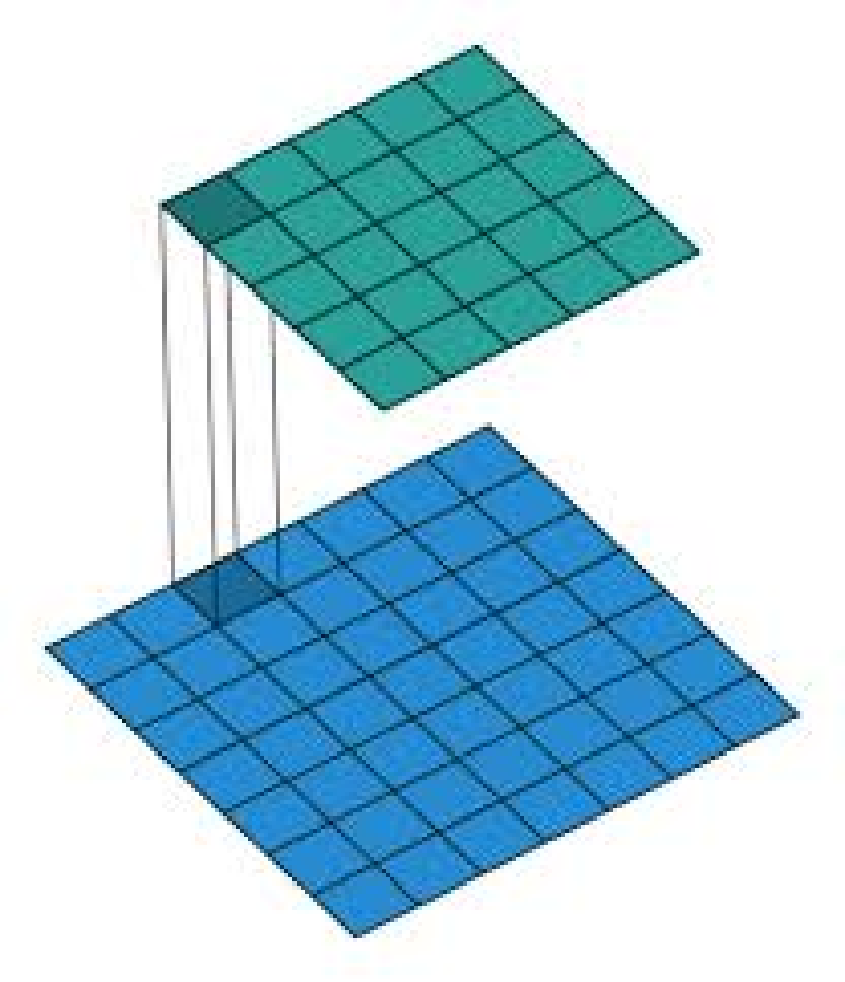
- If we want a neural network to take as input 256×256 images, the input layer will have 65,536 neurons
- ... if we have one hidden layers with 1,000 neurons, we're already at 65 million parameters
- ... if we have two hidden layers with 15,000 neurons each, we're at over a **billion** parameters

Solution: convolutions

ML@B

- Convolutional architectures allow us to **vastly** reduce the amount of parameters in a network
- Convolutions emulate the response of an individual neuron to visual stimuli
- Convolution operators are translation invariant

Convolution gif



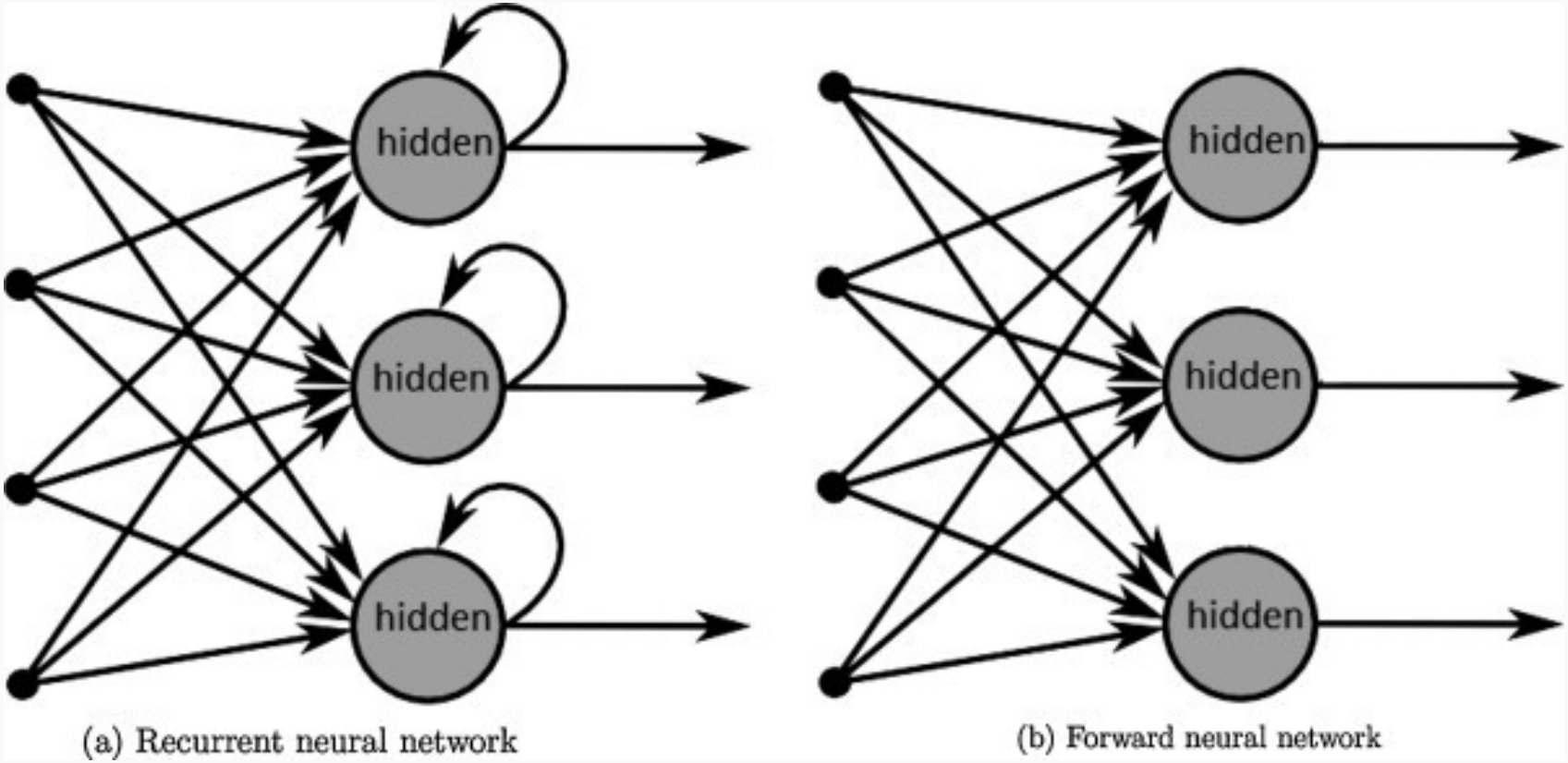
[1] http://iamaaditya.github.io/2016/03/one-by-one-convolution/

Problem: sequential input

ML@B

- What if we want to do machine learning on sequential data (i.e. music, text, time series, etc.)?
- How do we feed in a sequence of inputs into a neural network?
- How does a neural network keep track of all of the information it receives?

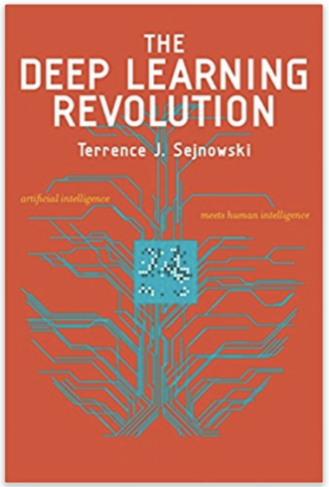
Solution: recurrence!

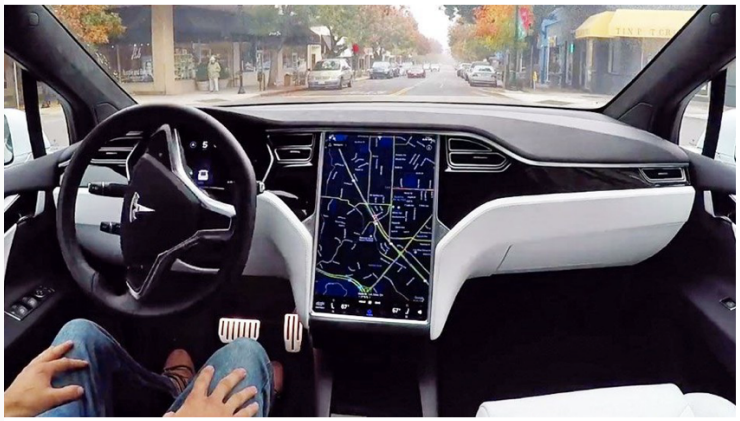


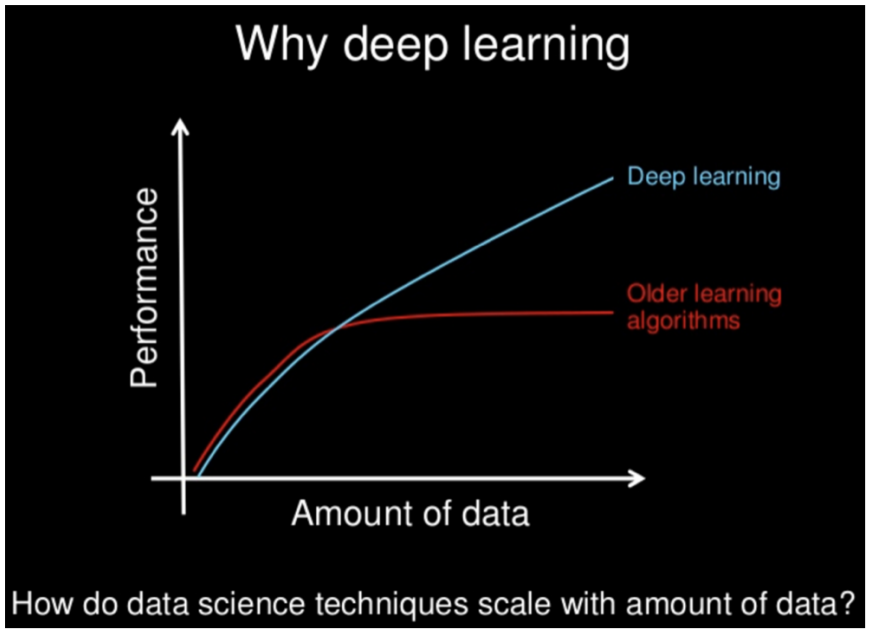
(a) Recurrent neural network

(b) Forward neural network

The deep learning revolution

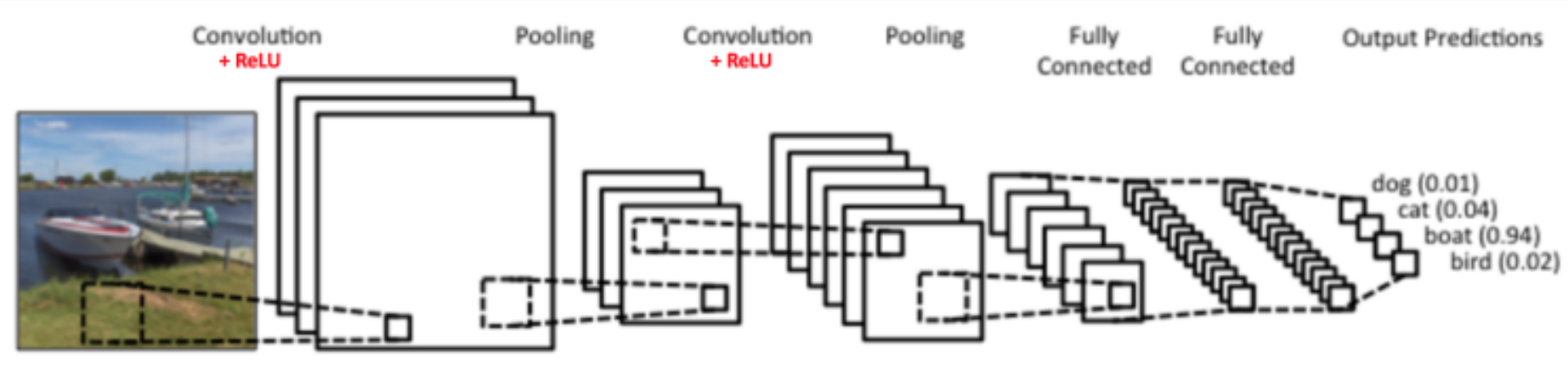




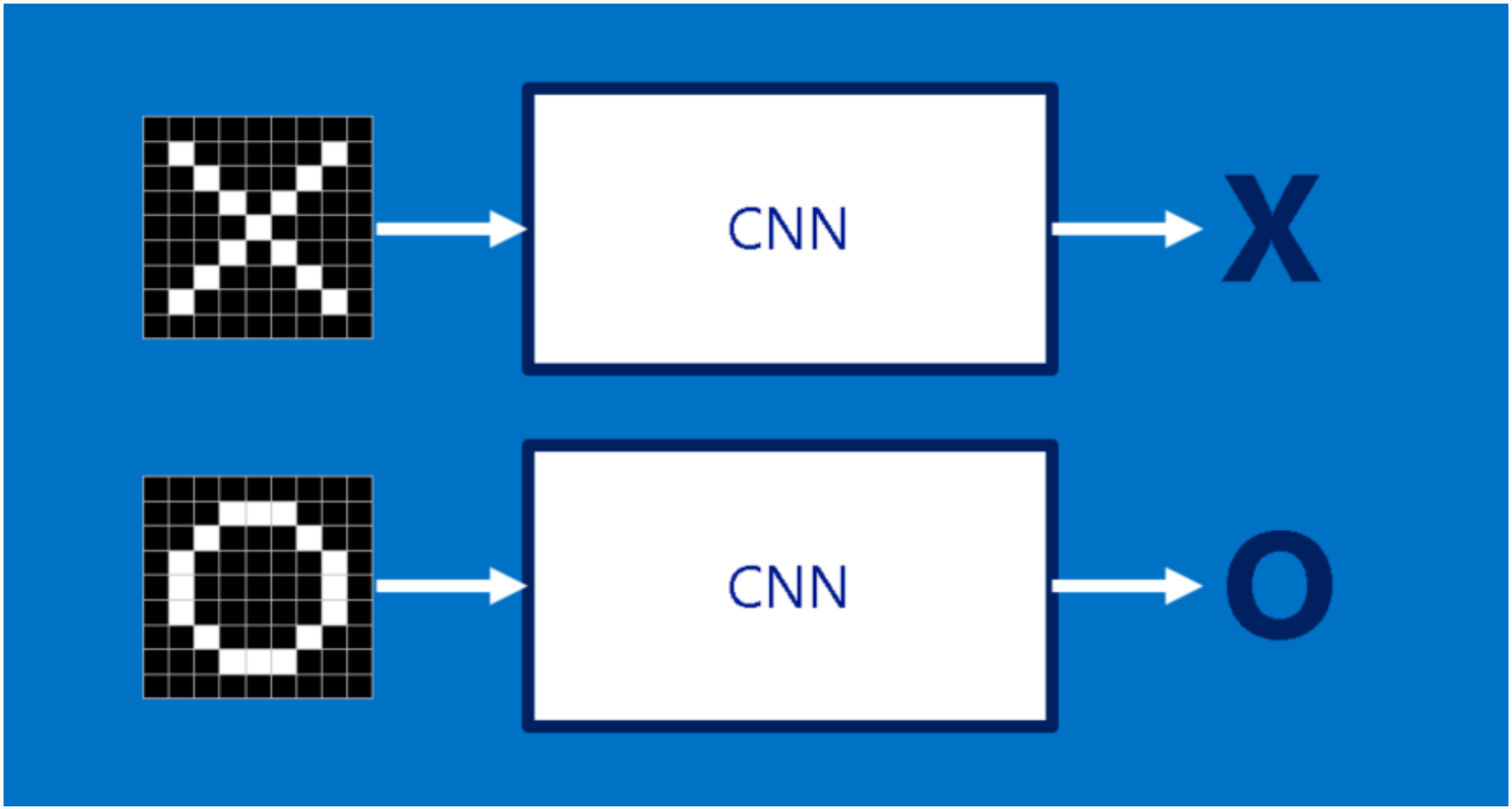


Convolutional Neural Networks

Our first CNN

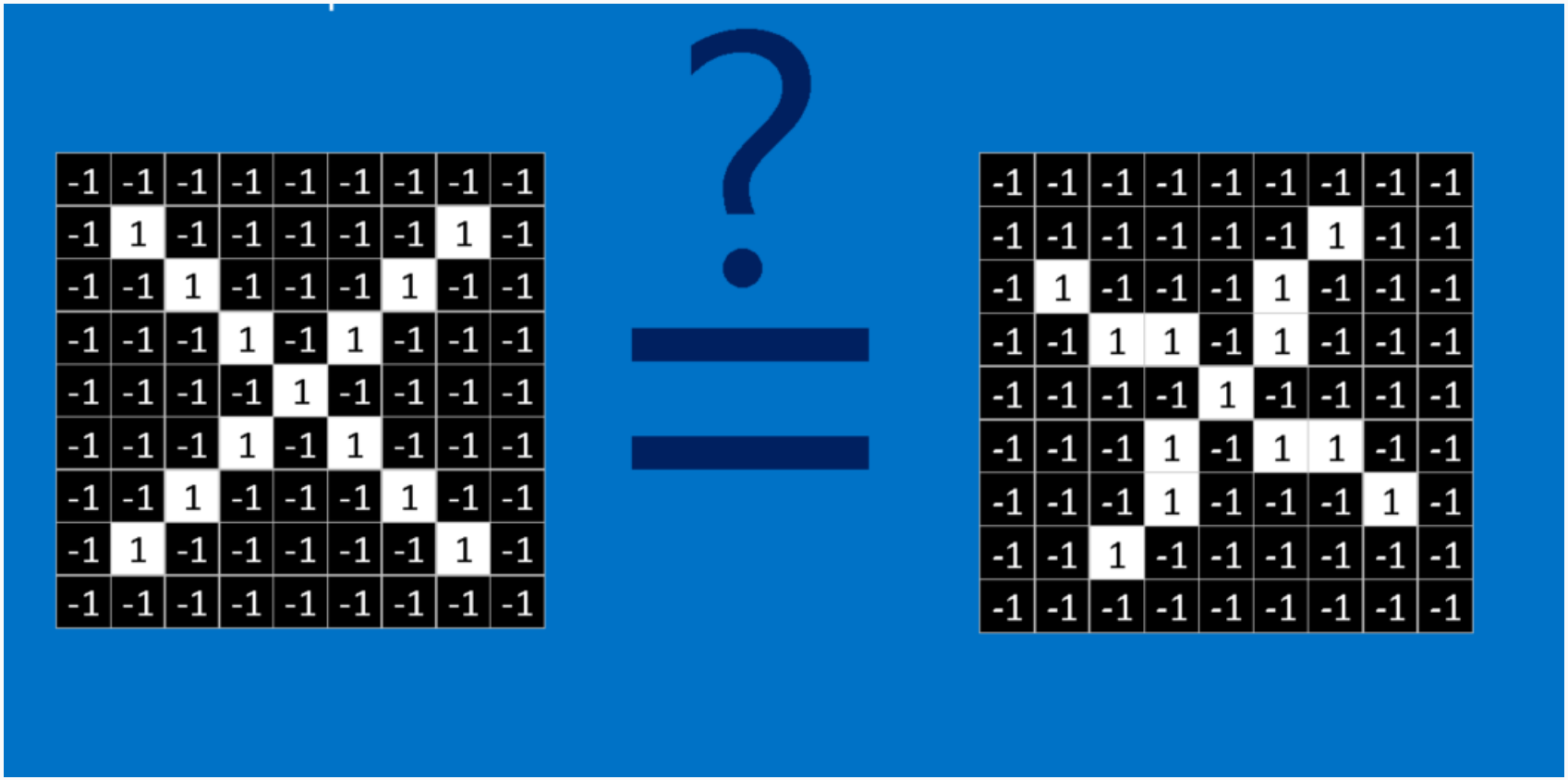


Let's look at a toy example!



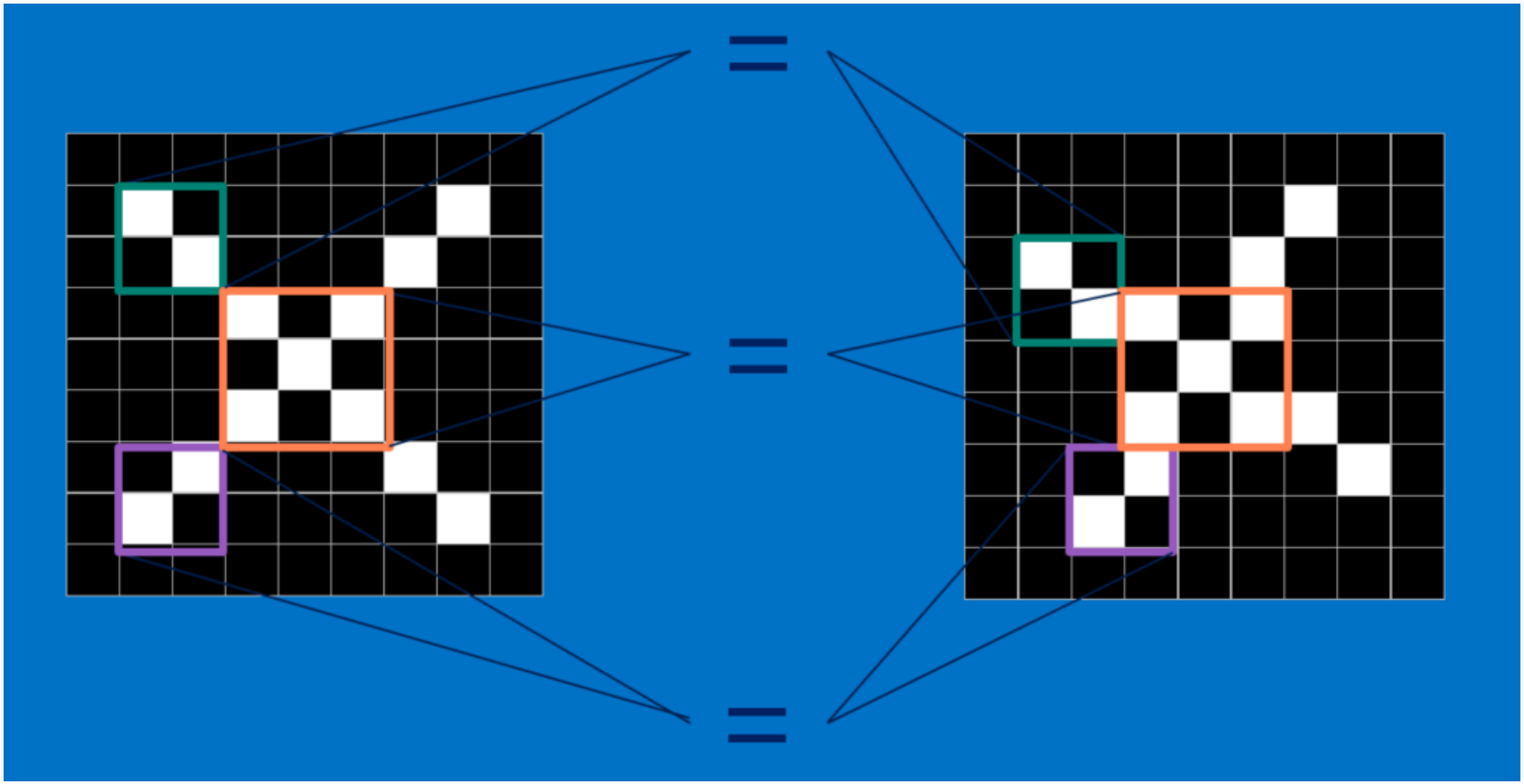
- How would a CNN classify X's vs O's?

Would you classify this as an X?



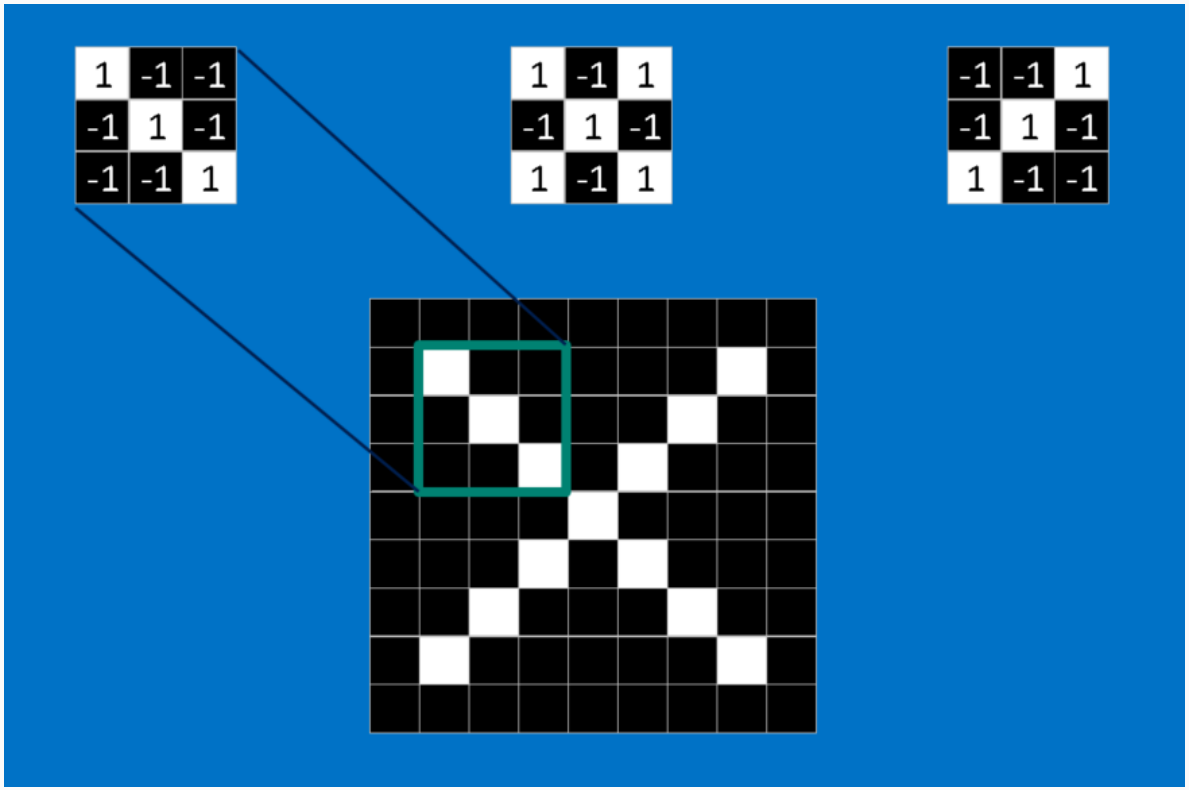
- A computer can't easily distinguish this example as an X.. so what can we do?

What is a feature?



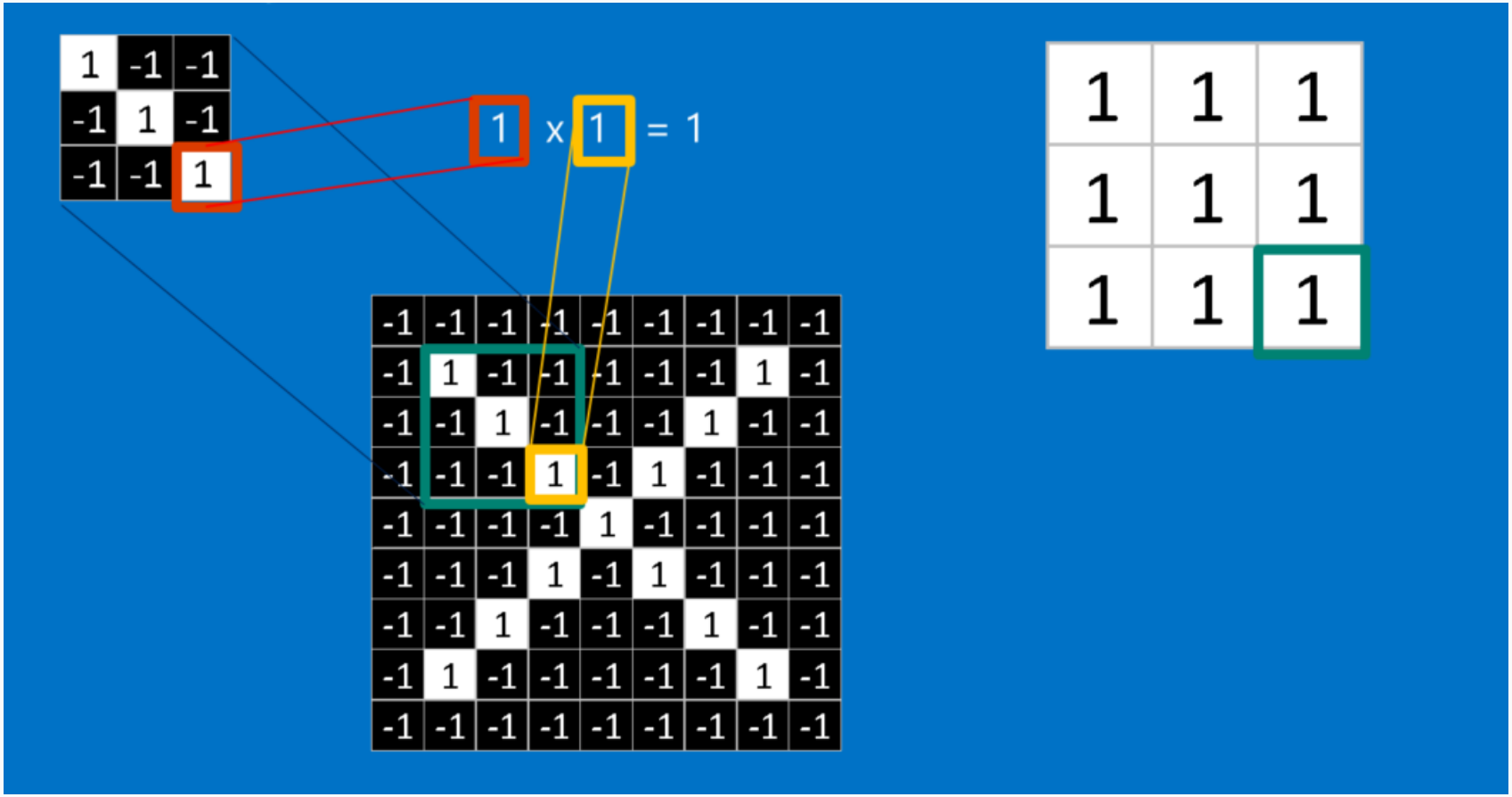
- We can break down the image into smaller images and look for patterns!
- Can we see here the similiar patterns?
- These diagonals and crosses can be thought of as features!

Filters



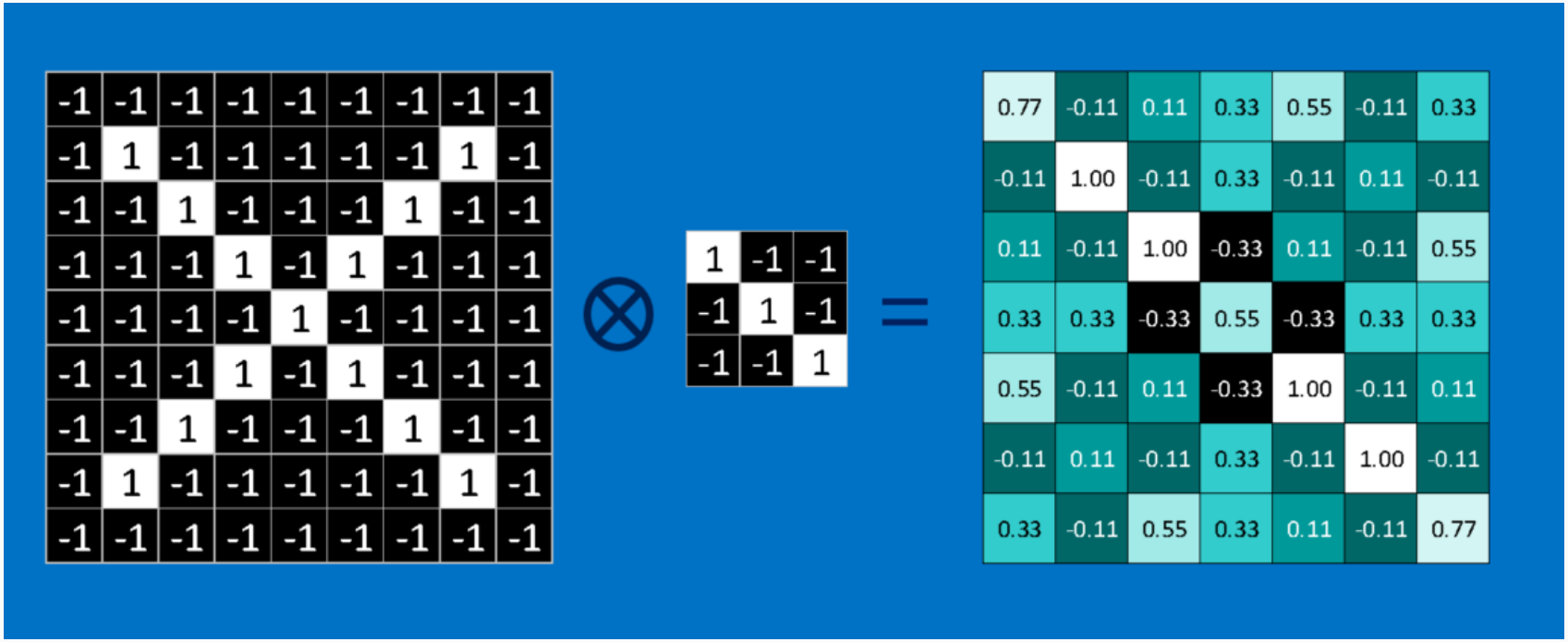
- Let's break this up into smaller problems... what about smaller patterns!

CNN Layers: Convolutions



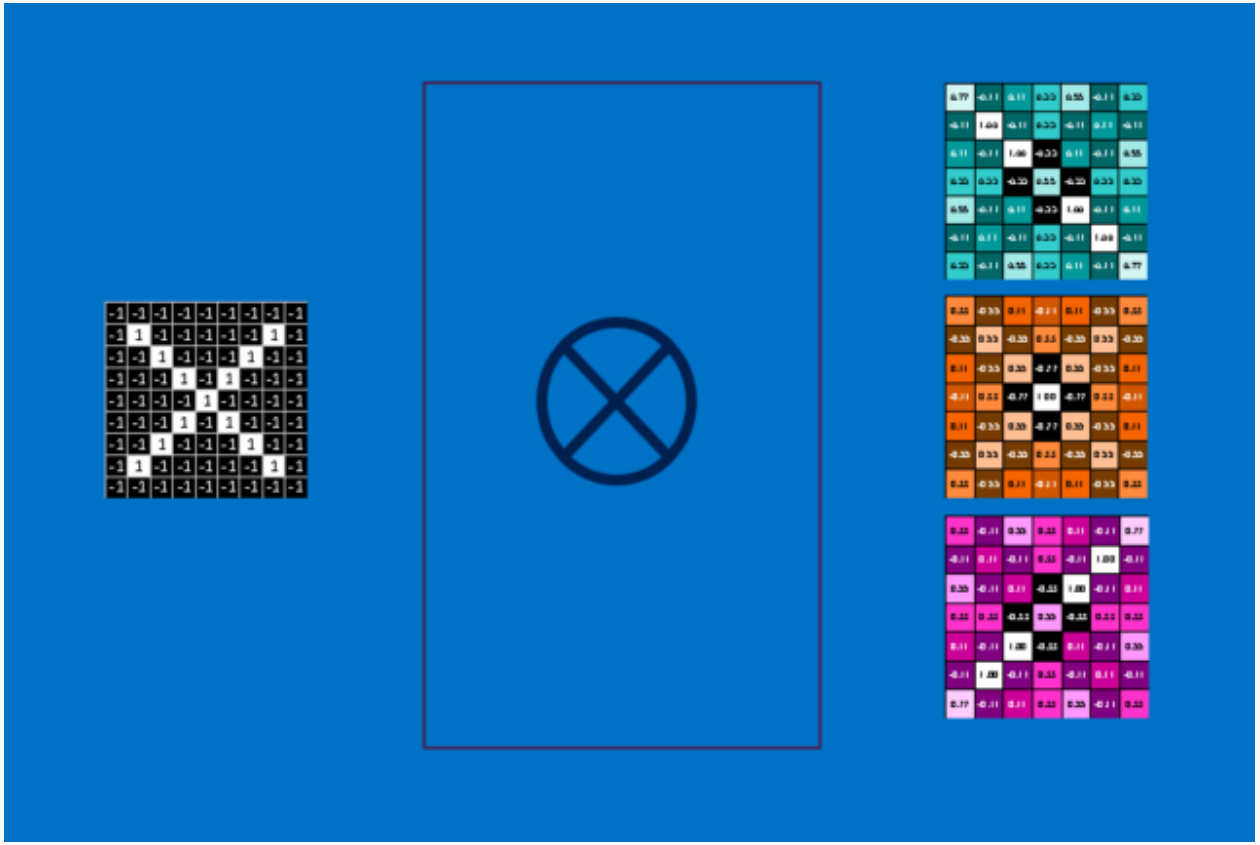
- Now let's compare our smaller pattern with the image.
- This is the basic idea behind convolutions!

CNN Layers: Convolutions



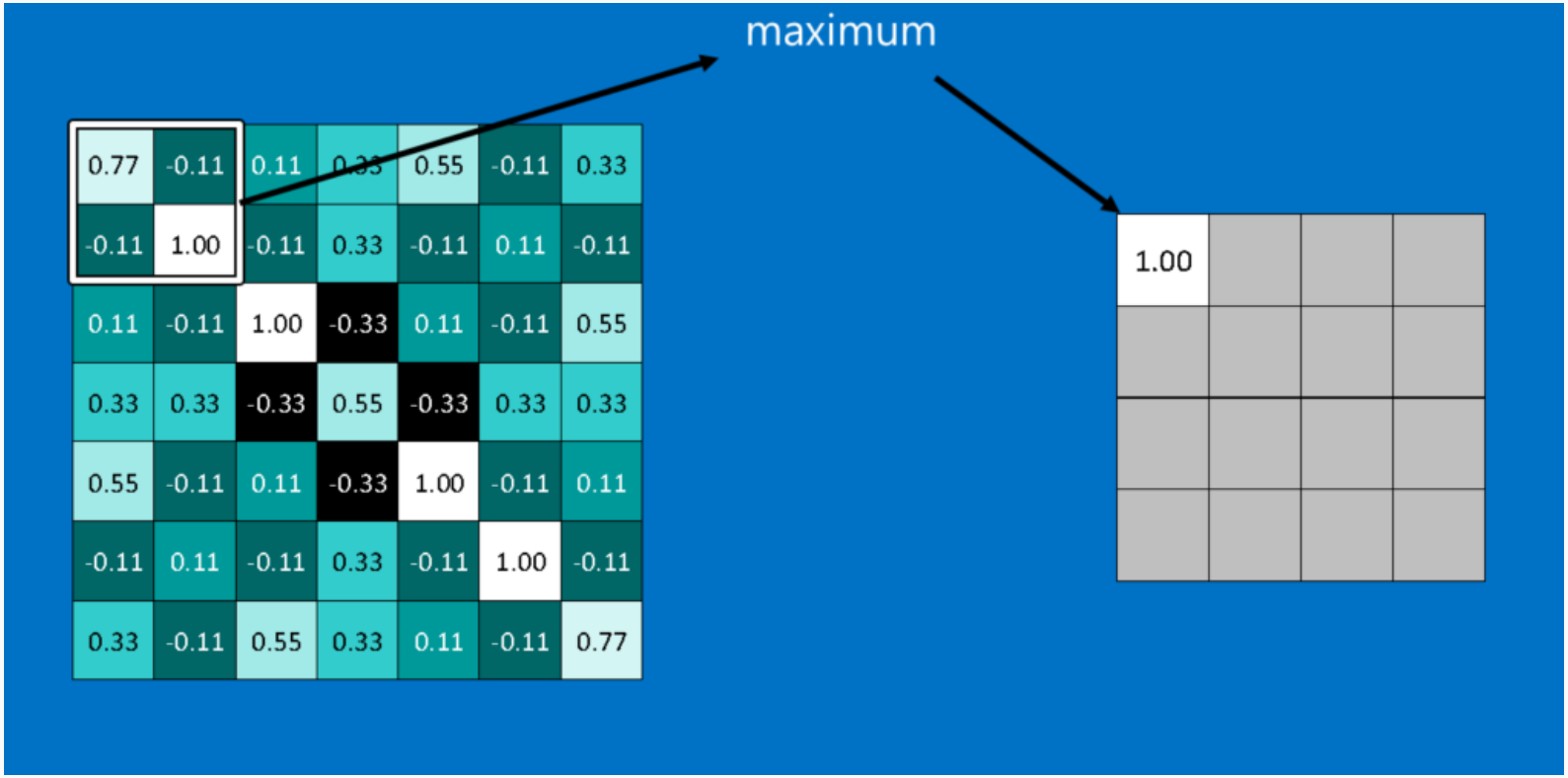
- Now lets apply this process to every part of the board.
- We can look at the new board and call it a filtered version of this specific feature
- Shows us where on the image the feature matches the most

CNN Layers: Convolutions



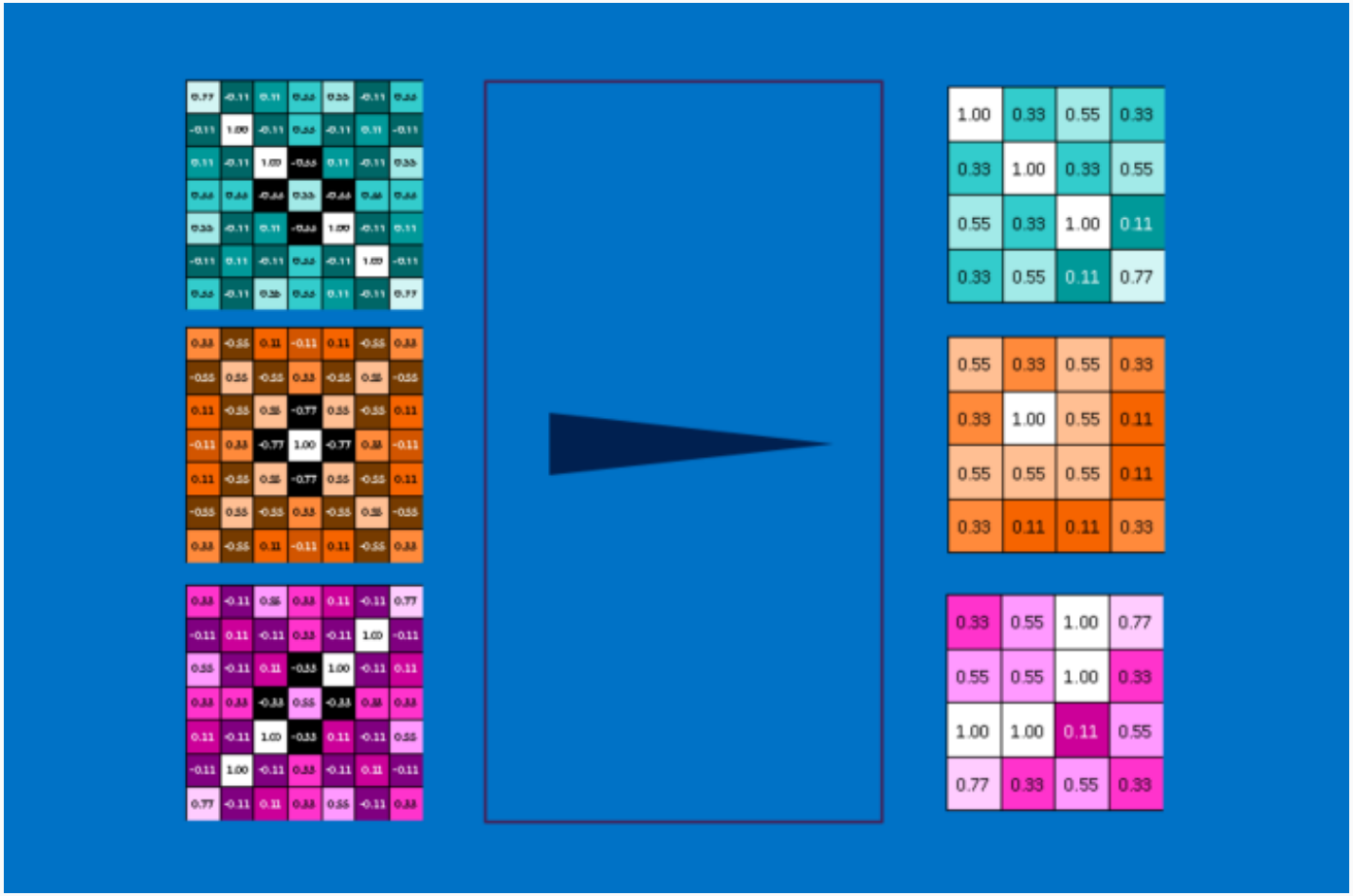
- Now lets apply this process to every part of the image for different features.

CNN Layers: Pooling



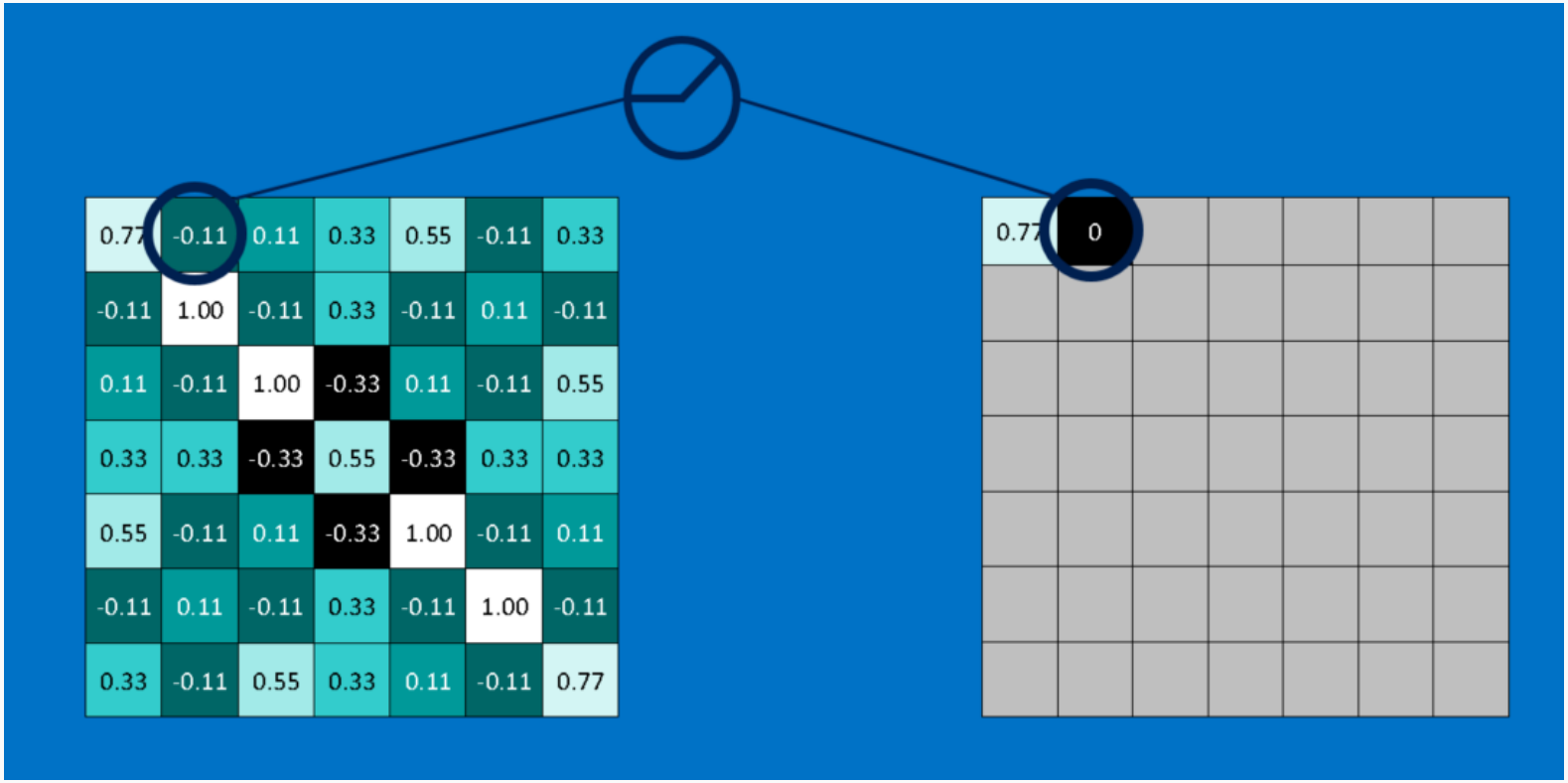
- If CNN's are really deep, doing that math for every possible pixel gets to be quite computationally expensive
- That's why we have pooling layers, that down sample our image!

CNN Architecture



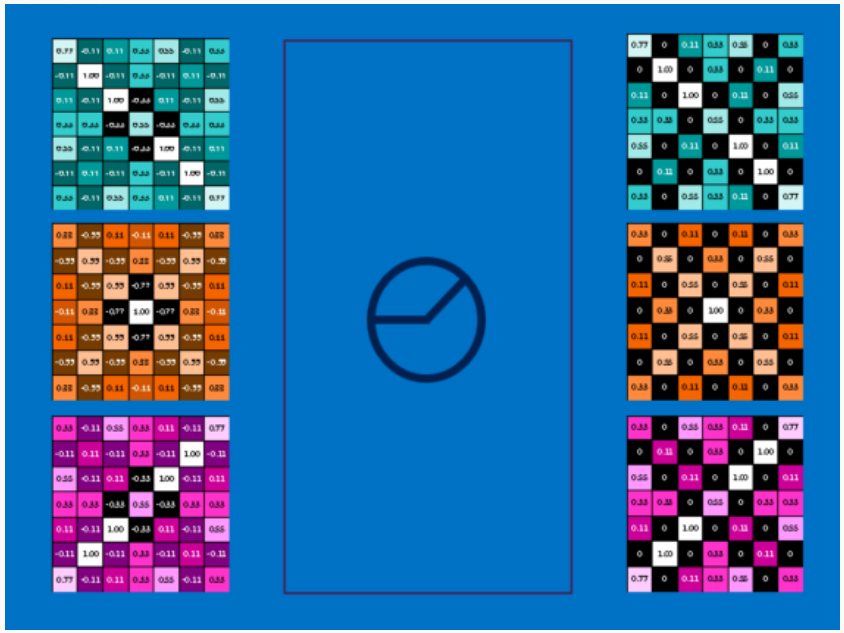
- By applying these pooling layers, we try to capture the patterns and get rid of extra unneeded information.

CNN Layers: Activation



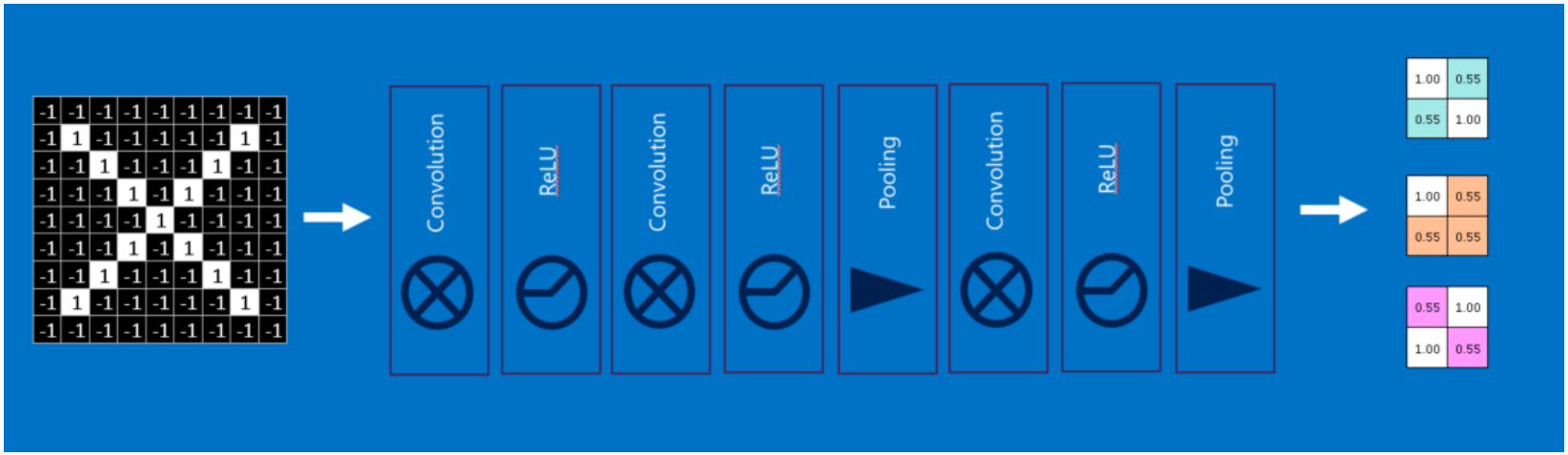
- Remember those activation functions you learned from the last lecture? We use them here too!
- Applying RELU we simply get rid of all the negatives!

CNN Pooling Layers

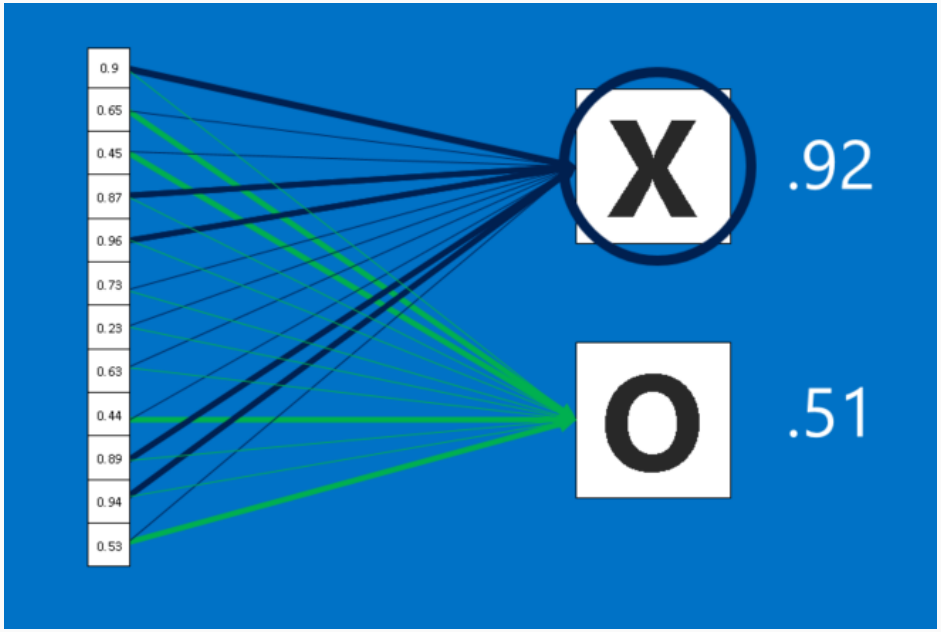


- The idea behind this, is to help keep any one number stuck at 0, or block up towards infinity
- This may not seem like a big issue in our toy example, but in much larger more complicated CNN's having stable values is important!

Let's put it all together!



CNN: Fully Connected Layer



- At the end we use a fully connected layer to hold a vote if the image is either an X or O
- Each value has a weight though, so not all values contribute equally
- This goes back to the ideas you learned last week!

Applications of CNNs

ML@B

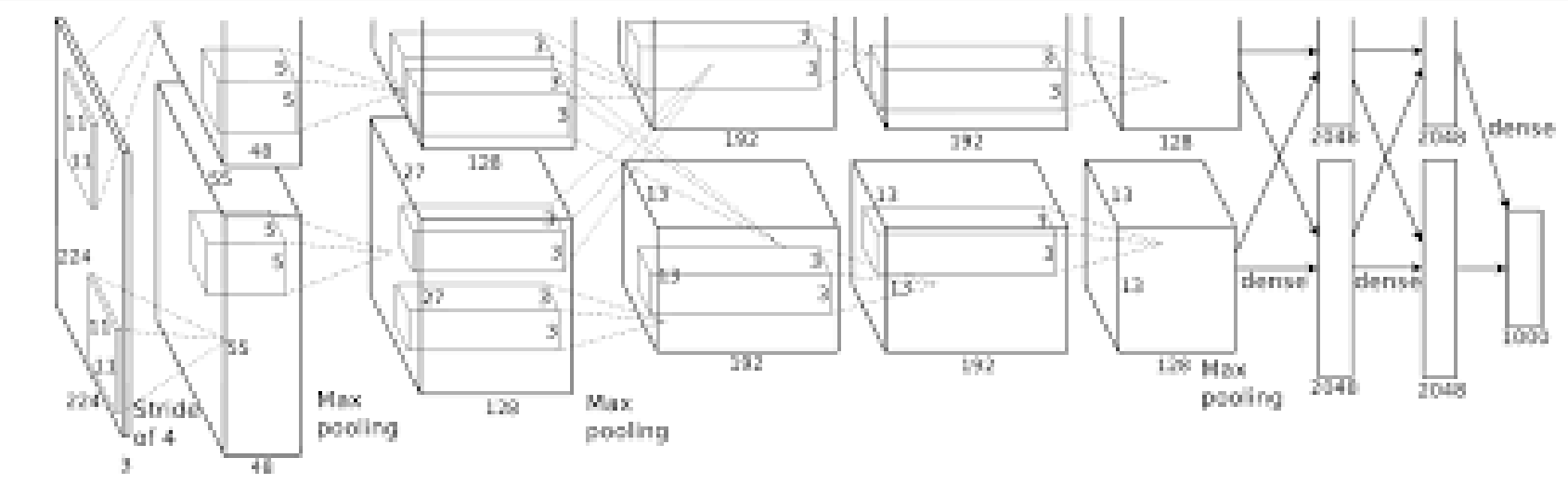
- Computer vision
  - Face recognition
  - Scene labeling
  - Image segmentation
  - Image classification
  - Action recognition
  - Human pose estimation
  - Document analysis
- Natural language processing
  - Speech recognition
  - Text classification
- Games (Checkers, Go)
- Drug discovery
- Medical image analysis
- Aircraft ejection seat testing

Idolized CNN architectures

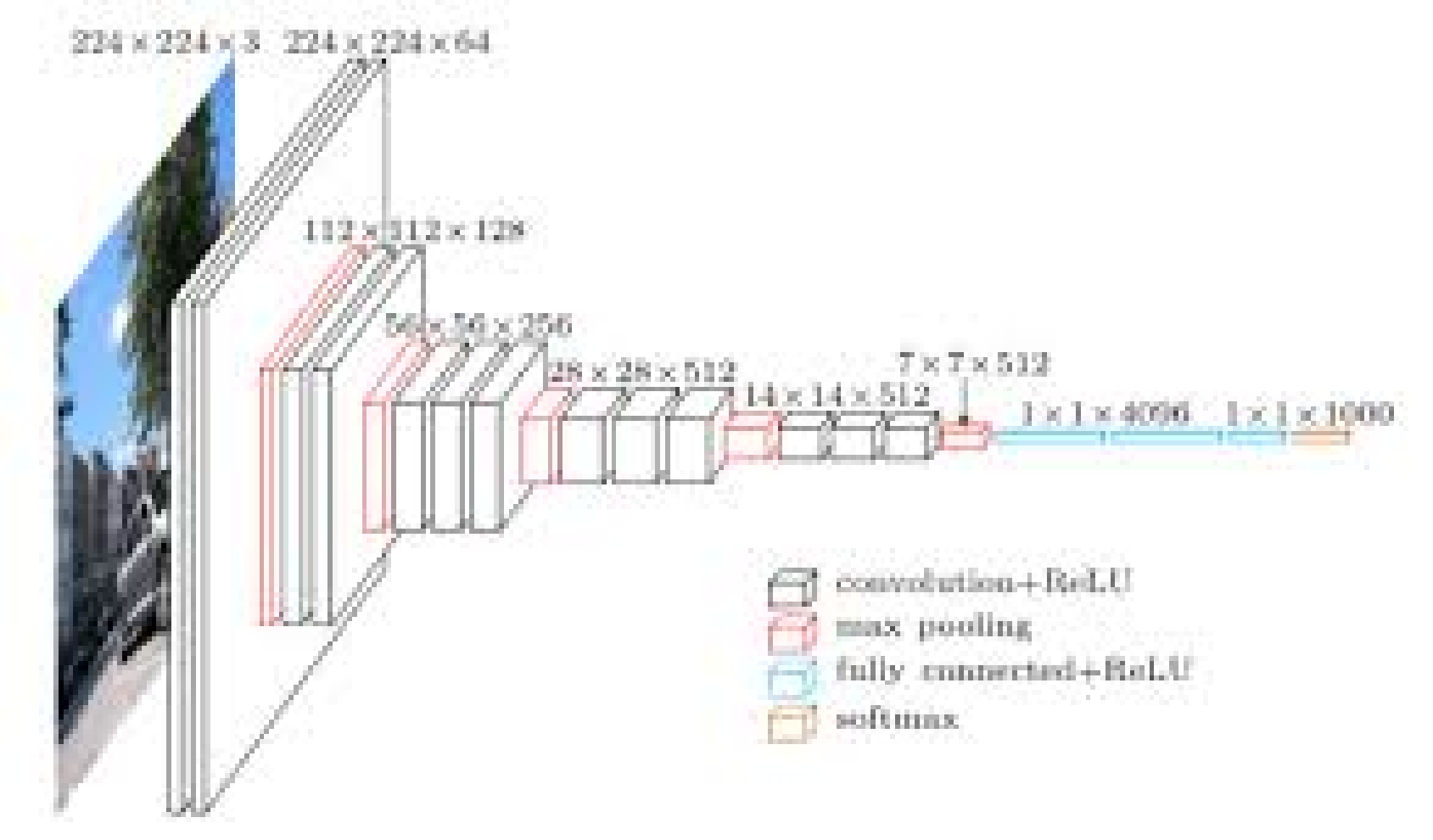
ML@B

- "The 9 Deep Learning Papers You Need to Know About" (https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html)
- Many of these models are used as benchmarks in deep learning
- There exist many pre-trained versions of the above models that can be leveraged for a new task (this is called **transfer learning**)
- Let's look at a few...

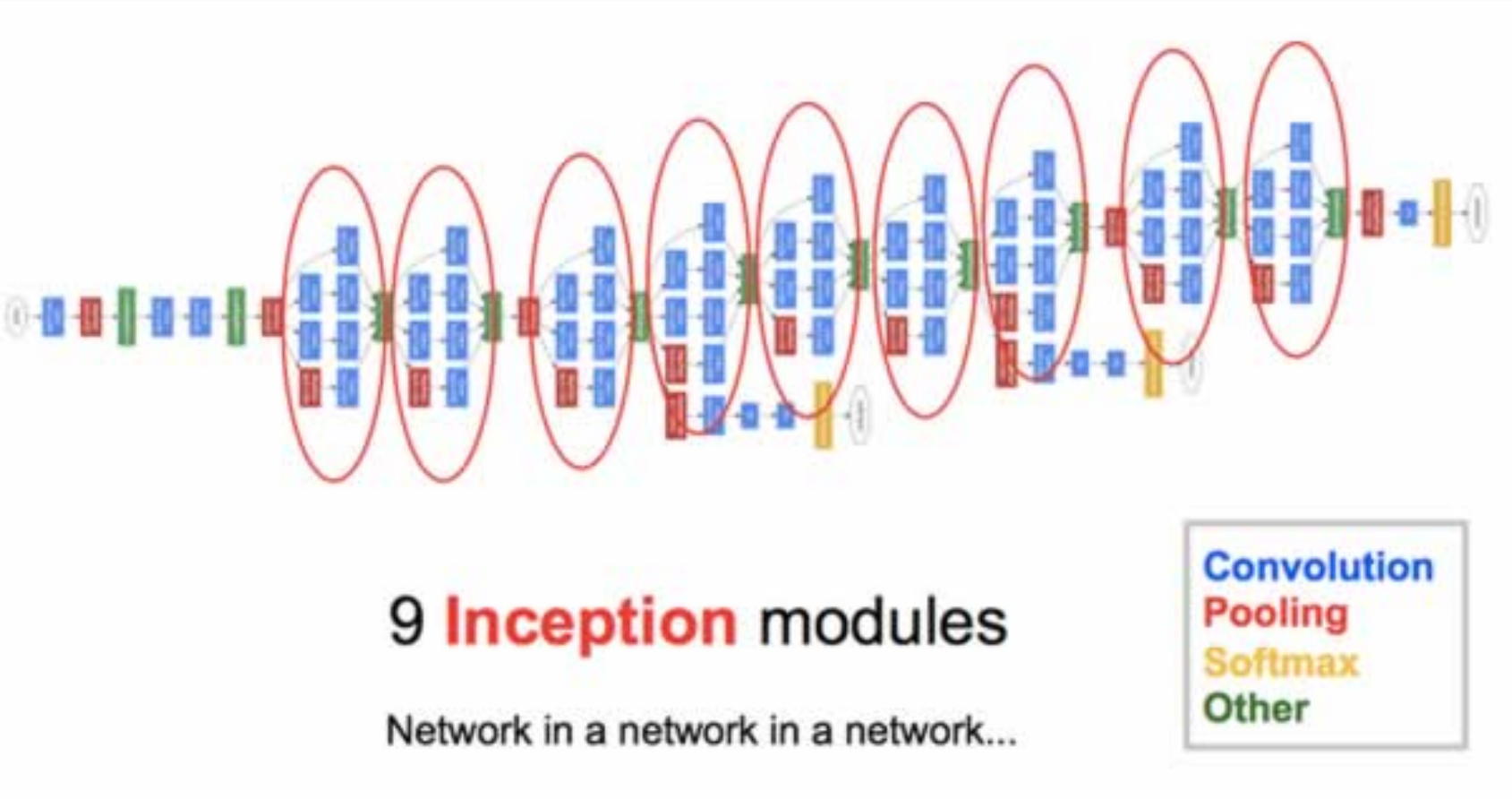
AlexNet (2012)



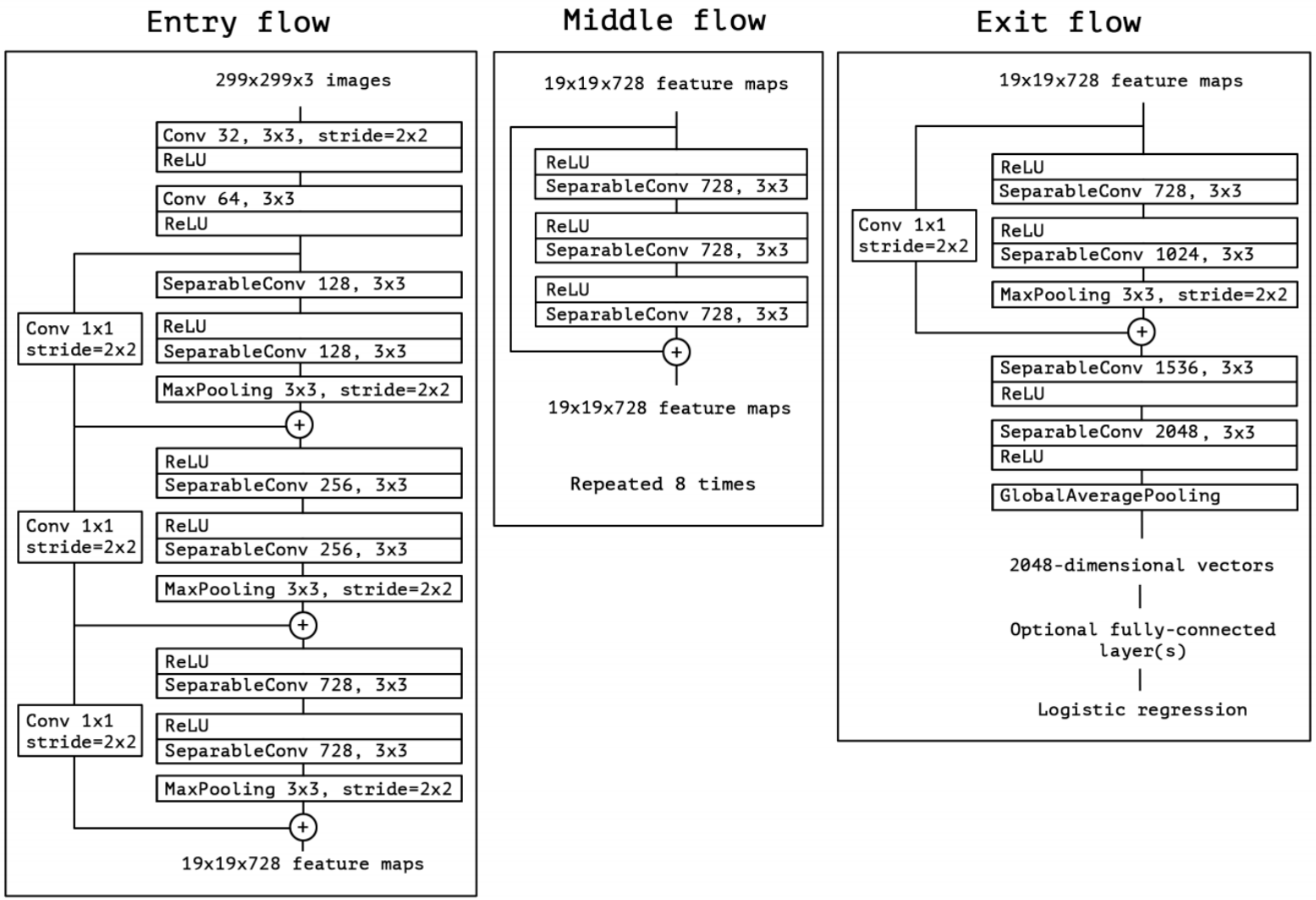
VGG-16 (2014)


224 × 224 × 3
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

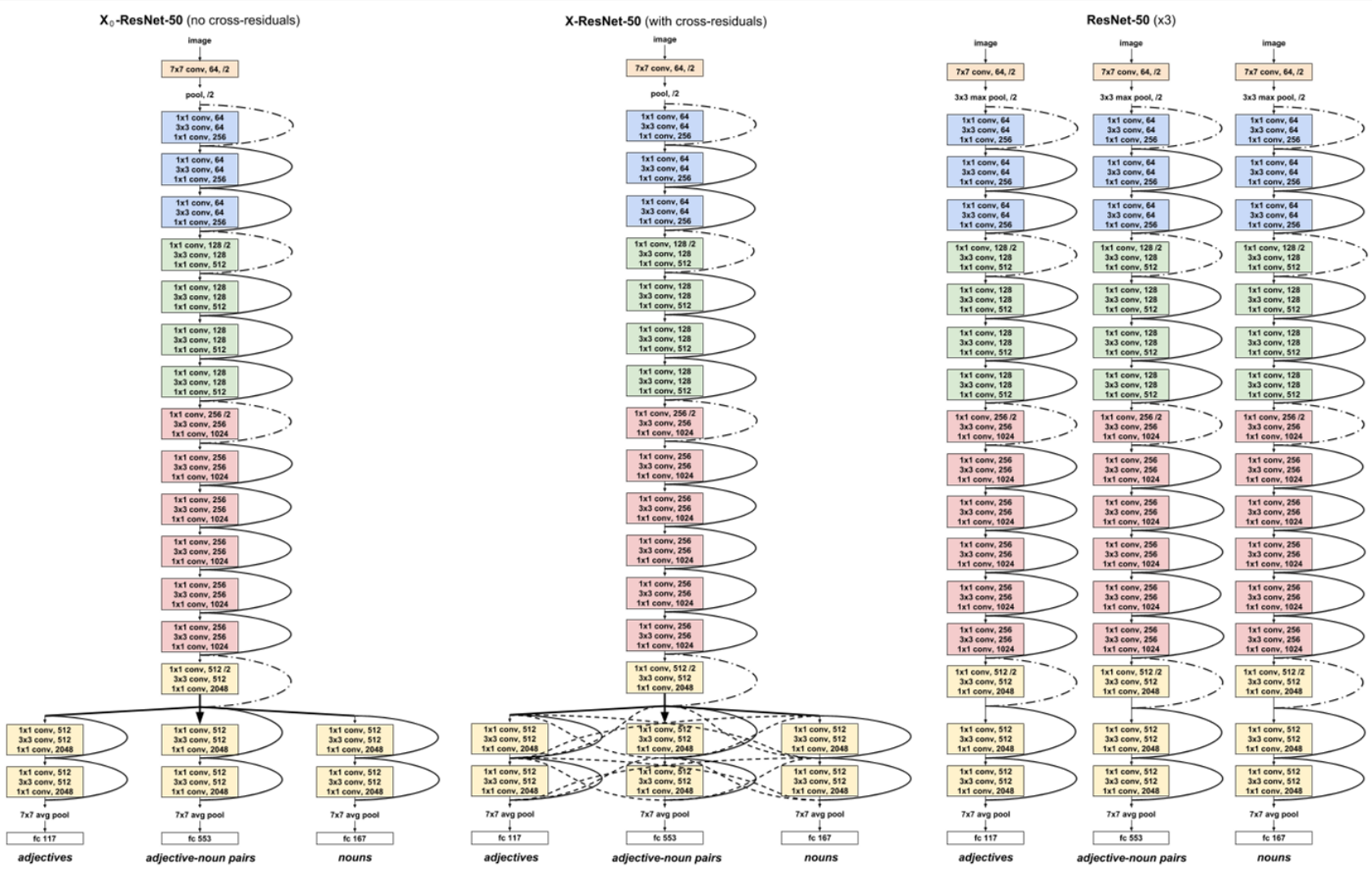
GoogLeNet (2015)



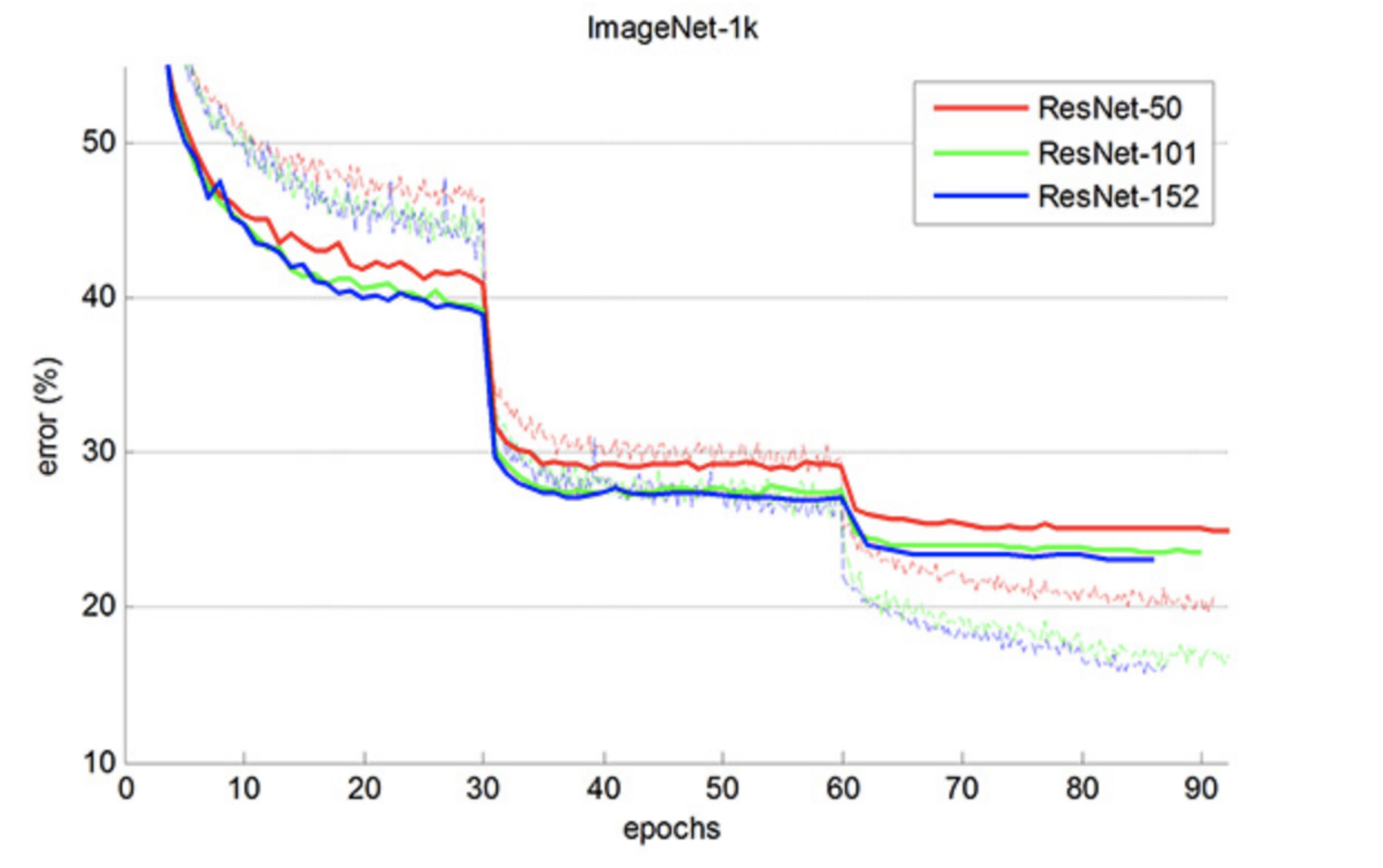
Xception (2016)



ResNet-50 (2015)



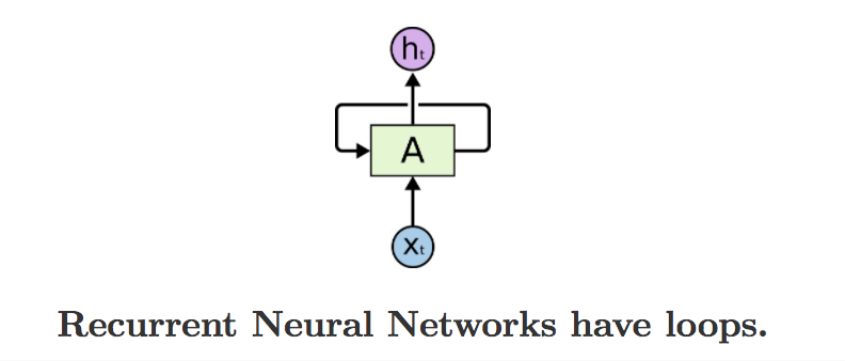
ResNet-152 (2015)



Recurrent Neural Networks

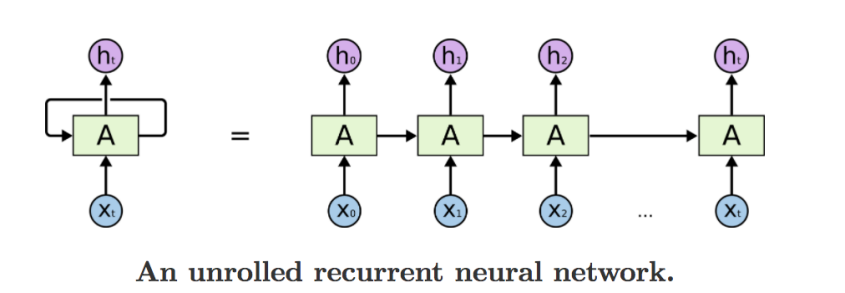
Recurrent neural networks

- RNNs are networks with loops, allowing information over time to persist



Recurrent Neural Networks have loops.

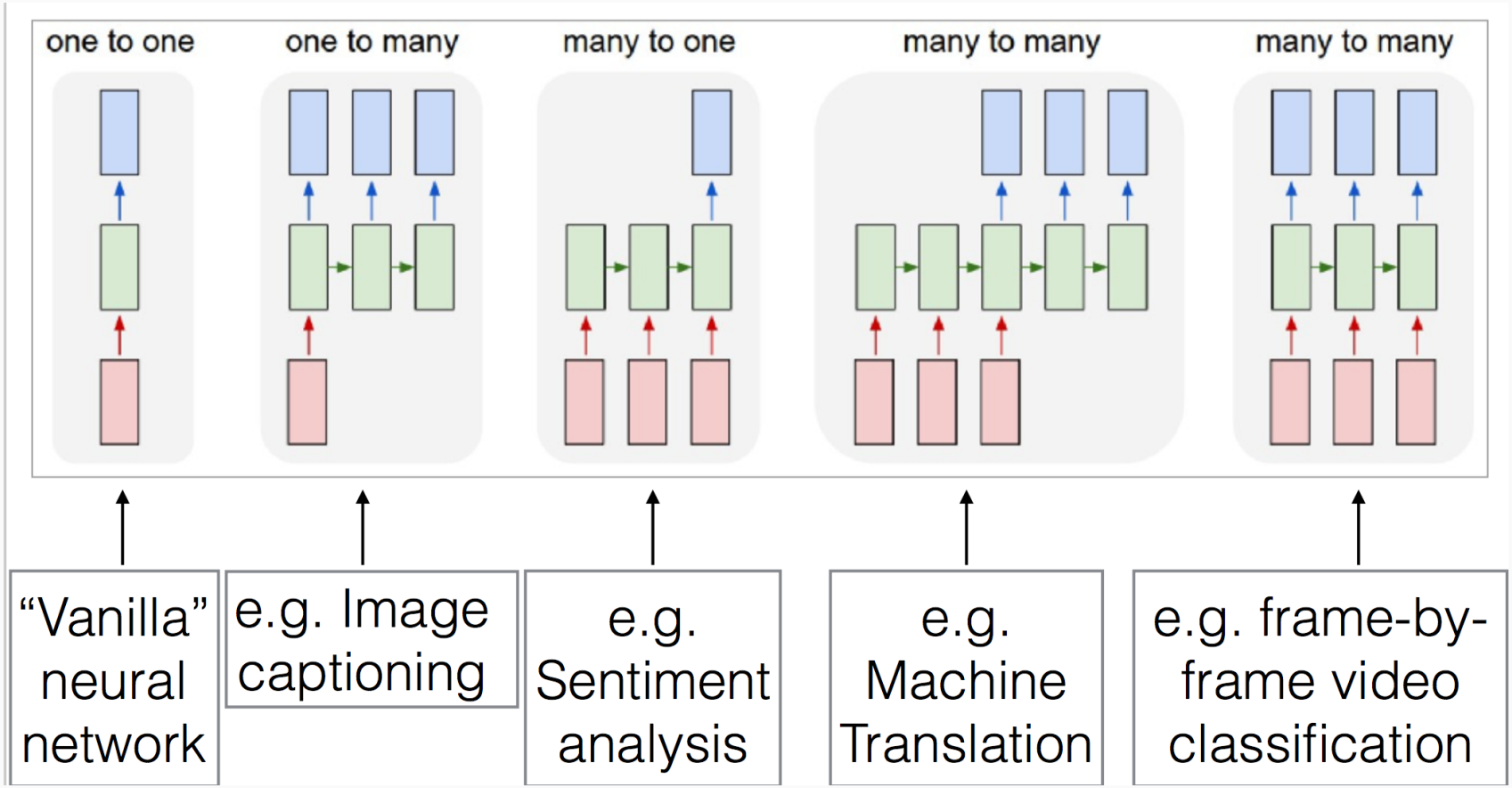
- Consider unrolling the loop:



An unrolled recurrent neural network.

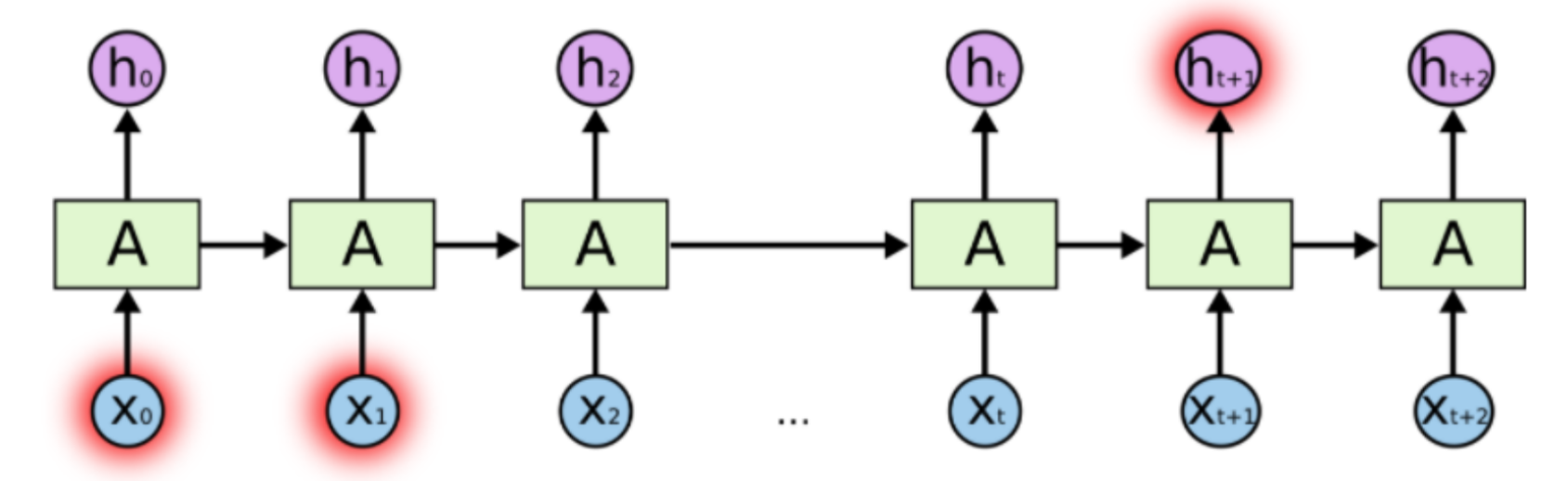
[2]http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Examples of recurrent neural networks



[3] http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Problem: long-term dependencies



"I grew up in *France*... I speak fluent *French*"

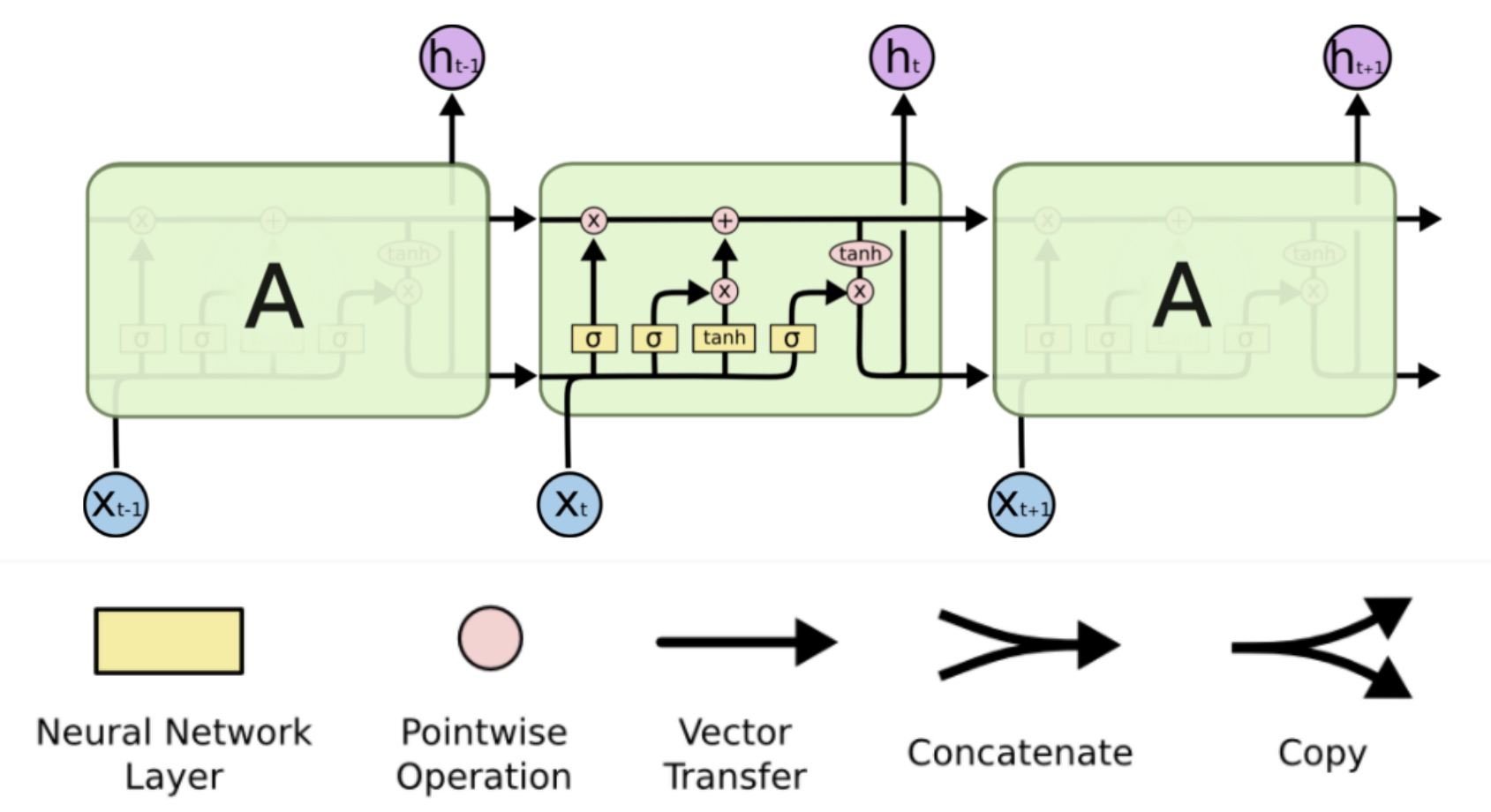
[4]http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Solution: LSTMs

ML@B

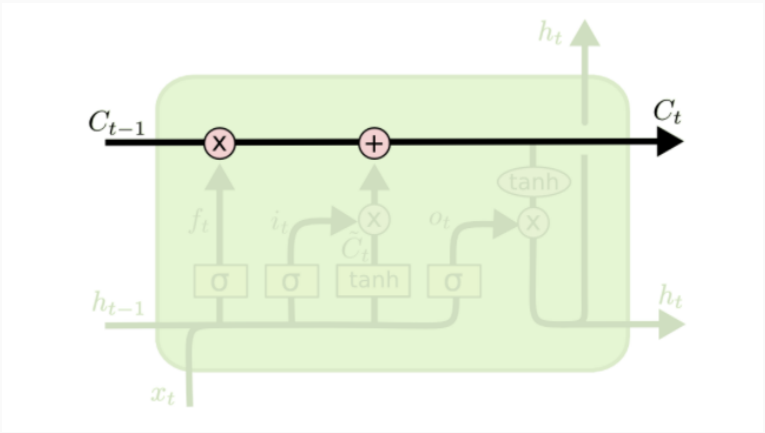
- "Long Short Term Memory Network"
- Introduced by Hochreiter & Schmidhuber in 1997
- Capable of learning long-term dependencies
- Instead of a single neural network layer, there are 4 which interact in a special way

Inside of an LSTM cell



[5] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

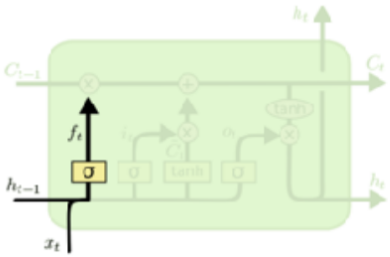
Core idea behind LSTMs



- The cell state is like a conveyor belt on which information flows
- The LSTM can add or remove information from the cell state, regulated by structures called gates
- Gates are made of a sigmoid neural net layer and a pointwise multiplication operation.

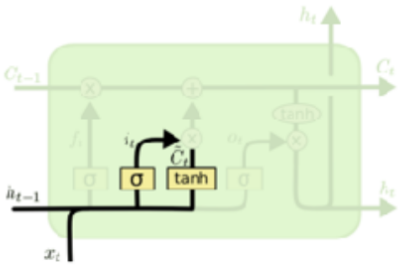
[6] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Steps in an LSTM



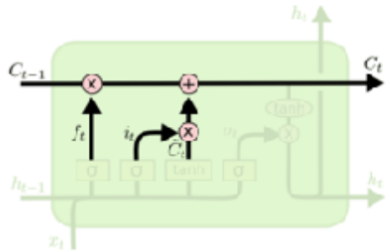
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

1. Input gate



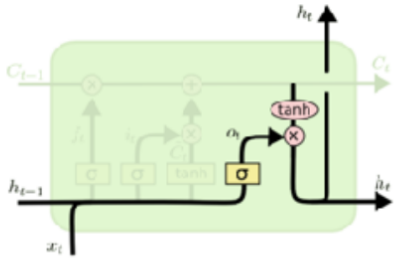
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

2. Forget gate



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

3. Update cell state



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

4. Output new hidden state and new cell state

[7] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Applications of LSTMs

ML@B

- Time series prediction (finance, economics, etc.)
- Speech recognition
- Sentiment analysis
- Machine translation
- Music composition
- Robot control
- Handwriting recognition
- Human action recognition
- Protein homology detection
- Predicting subcellular localization of proteins
- Text generation
- Many, many more...

Cool things we did not have time to talk about today

- Training networks using GPUs
- CNN-RNN combinations
- Network visualization and interpretation
- Transfer learning (see project 3!)
- Other types of neural networks: auto-encoders, Neural Turing machines, Generative adversarial networks, ...
- The future of all of this

Questions

Questions?

Demos

CNN demo: FashionMNIST

RNN demo: sentiment classification