10-403 Recitation (1/17/20): Introduction to Deep Learning, CNN & RNN

Checklist for logistics

- Can you log into Piazza and Gradescope?
  - If you are not added, email to amritsin@andrew.cmu.edu with **your name and Andrew ID**
- How do you know when the first assignment will be released?

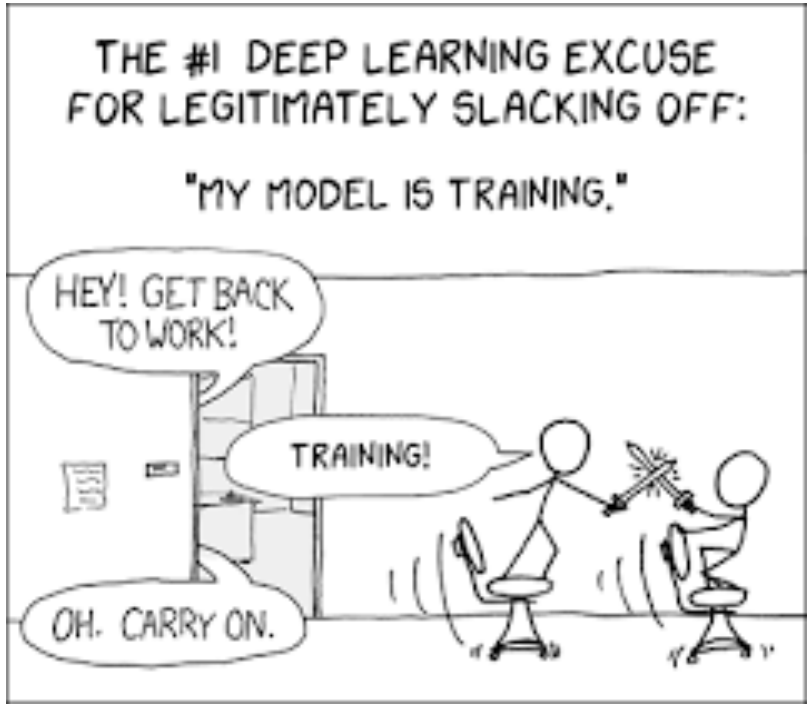
Questions to be answered in this recitation

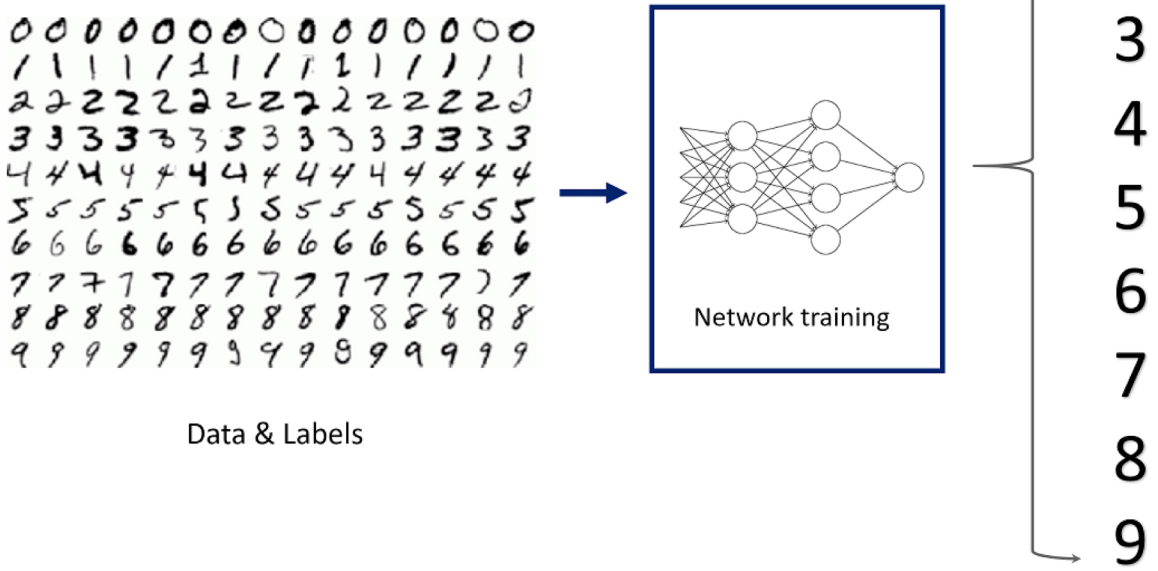
- Why do we need deep learning?
- How do we tailor neural network to each data domain?
- Why are CNN and RNN important in real-world application?
- What is intuition behind CNN and RNN?

Disclaimer: Some of the material were borrowed from 10-707 (Deep Learning)

Deep Learning

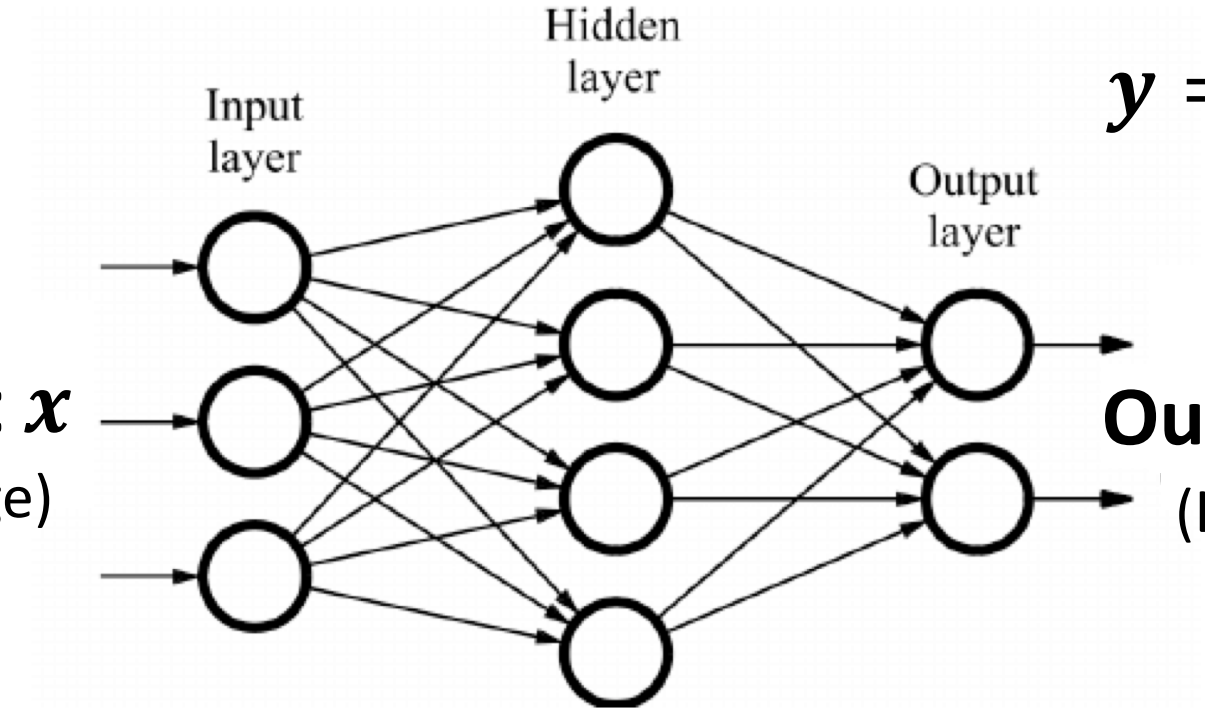
- What is it?
- Why the heck is it so popular?





Feed Forward Neural Network

- Feed-forward Neural Network = Multi-level perceptron = Fully-connected layer
- Hidden layer = Hidden units = Hidden nodes



Input x
(Image)

$$y = f(x)$$

Output y
(Label)

$$a = b_h + W_{xh}x$$

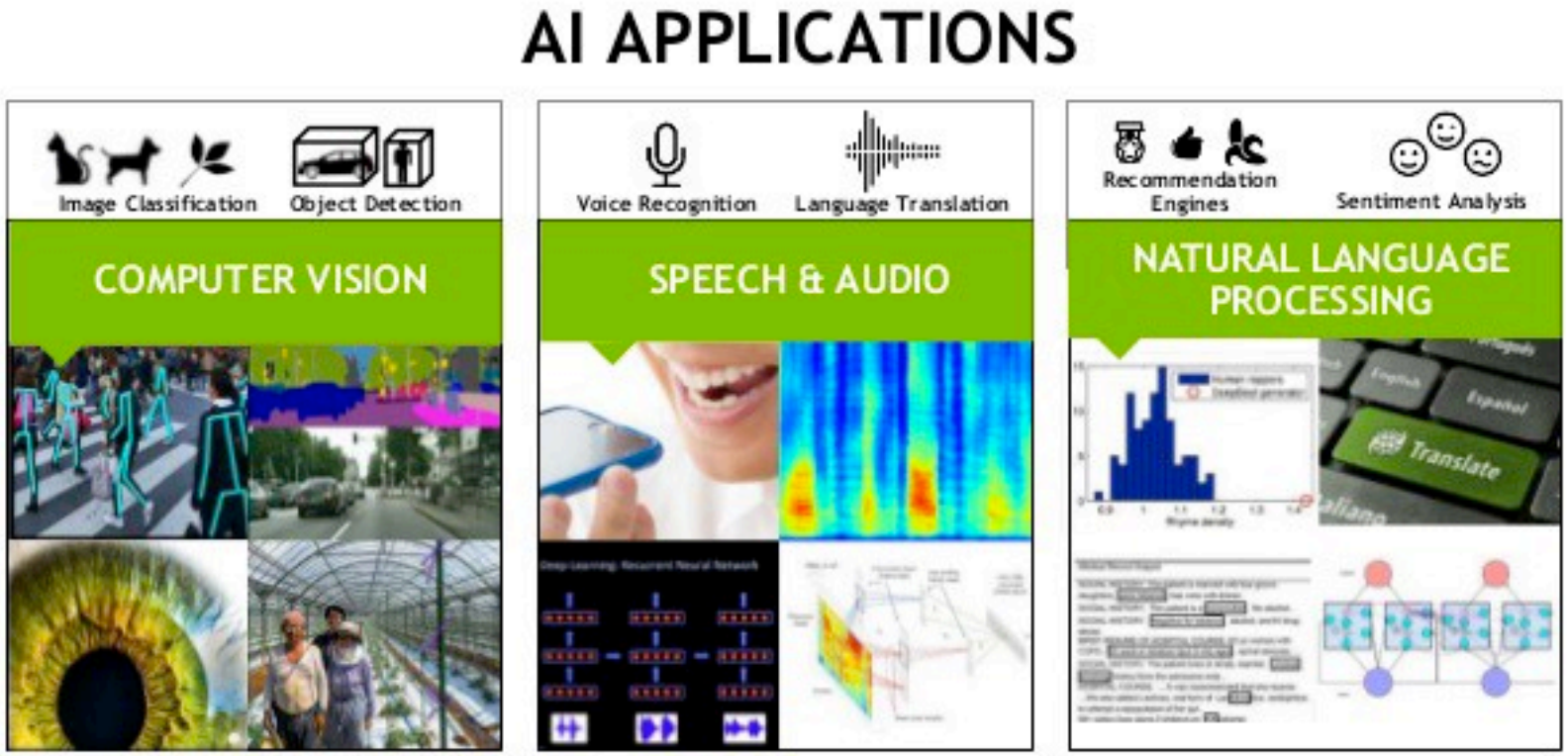
$$h = \tanh(a)$$

$$o = b_o + W_{ho}h$$

$$y = softmax(o)$$

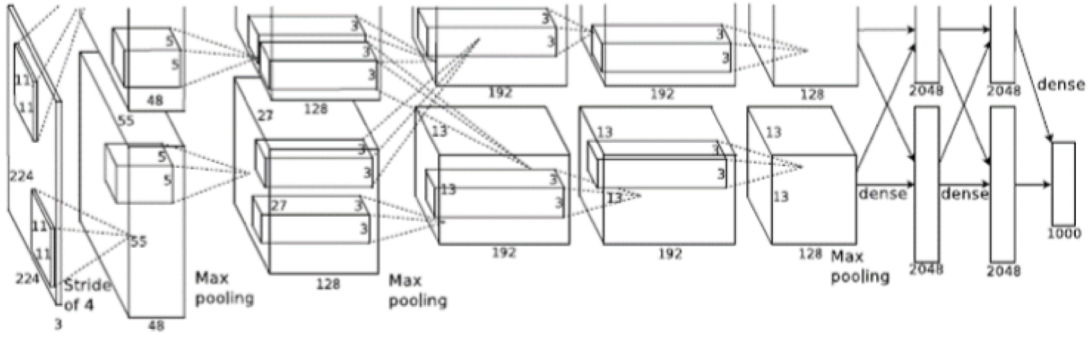
Deep Learning (= Deep Neural Network)

- Does feed forward neural network works for all data domain?
- What is key question to design neural network for each domain?



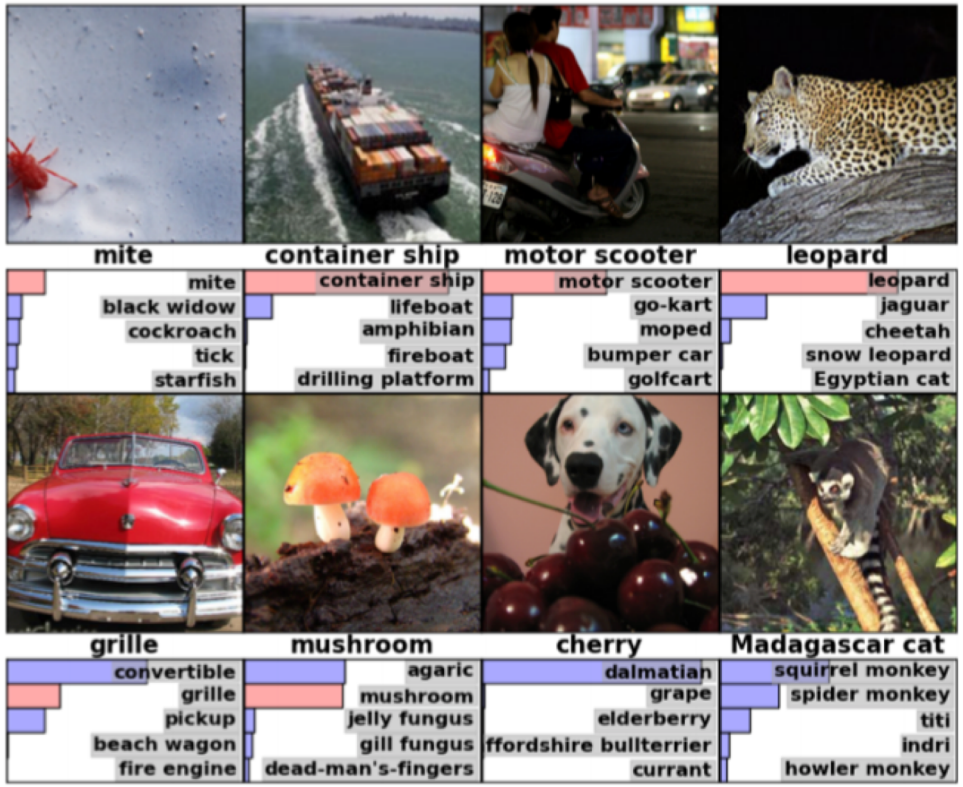
Convolutional Neural Network

Application: Computer Vision

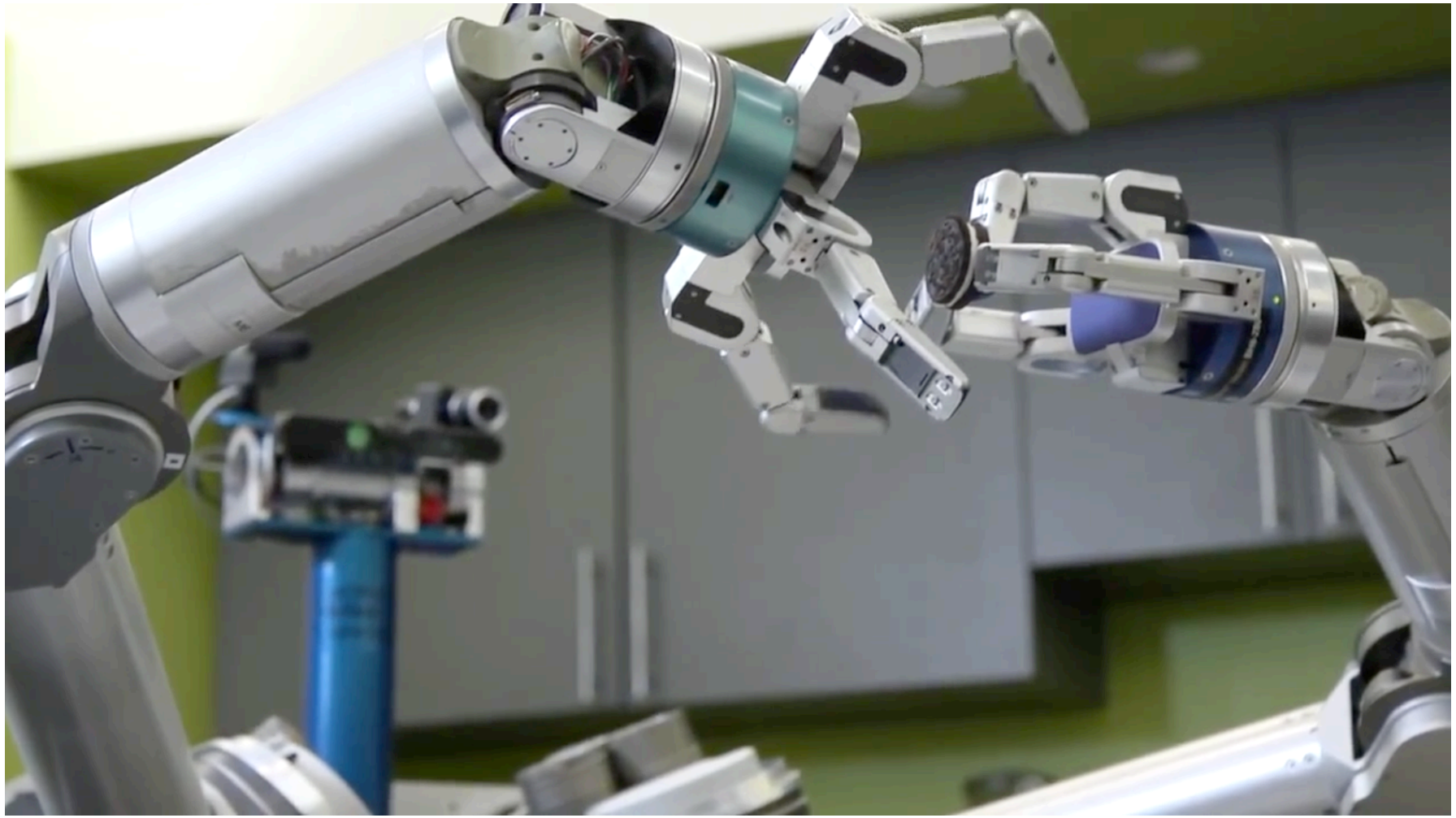


IMAGENET

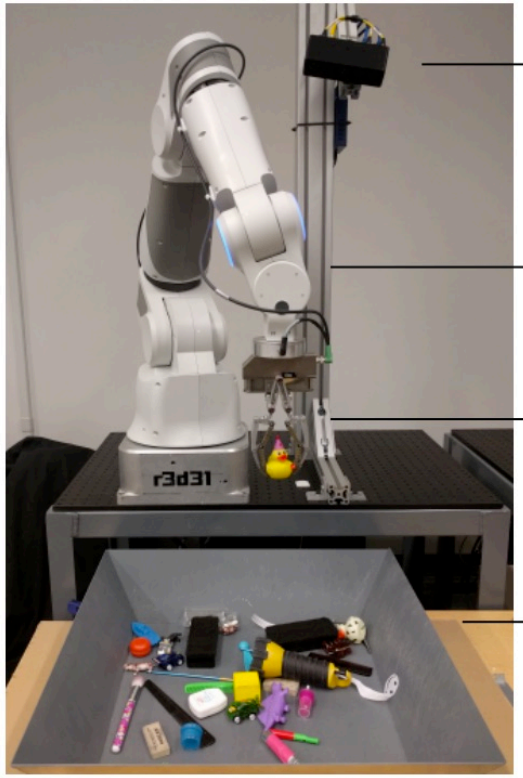
1.2 million training images
1000 classes



Application: Computer Vision



Application: Computer Vision



Shared Objective

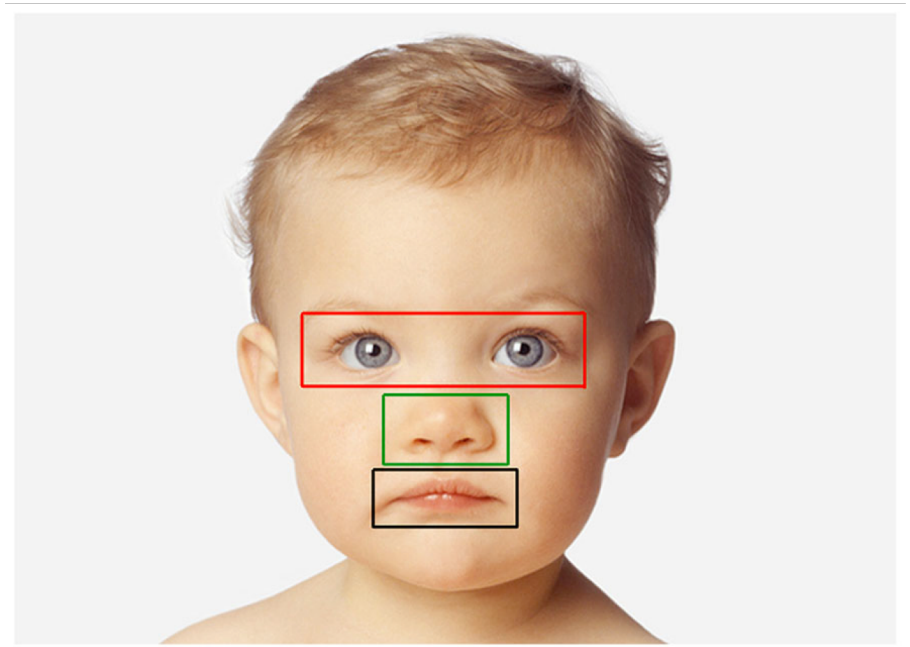
- Design algorithms that can process **visual** data to accomplish a given task (e.g. object recognition)

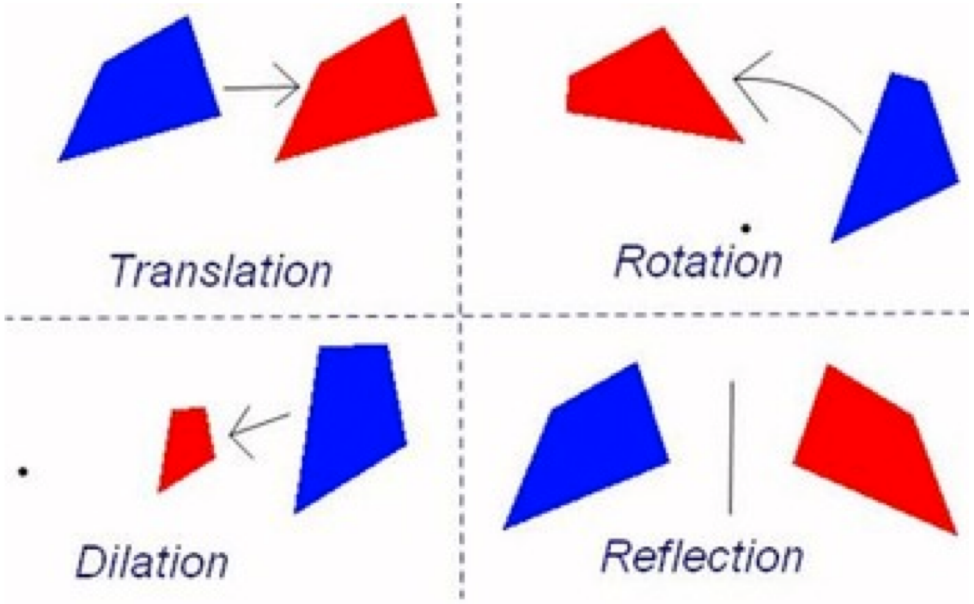
Shared Objective

- Design a **neural network** that can process **visual** data to accomplish a given task (e.g. object recognition)

Characteristics of Visual Data

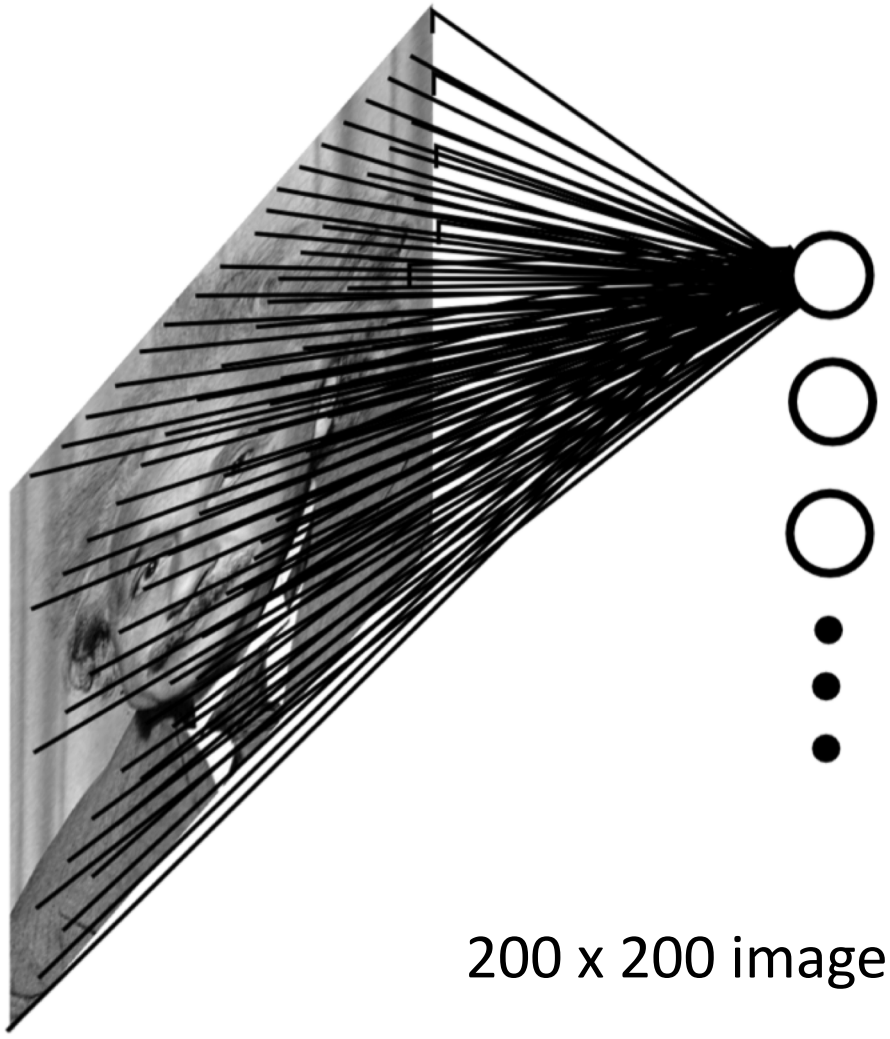
- **High-dimensional** inputs: 150 x 150 pixels = 22500 inputs (x 3 if RGB)
- 2D or 3D **Topology** of pixels (Spatial correlation)
- **Invariant properties** to certain variations: translation, rotation, etc.





Naïve approach

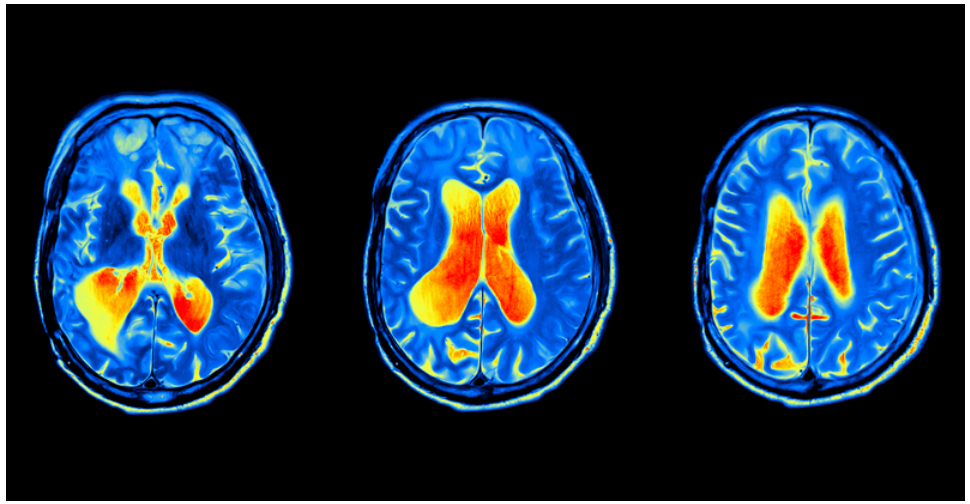
- Assume grayscale image and 40K hidden units
- How many parameters?
- What's the problem?

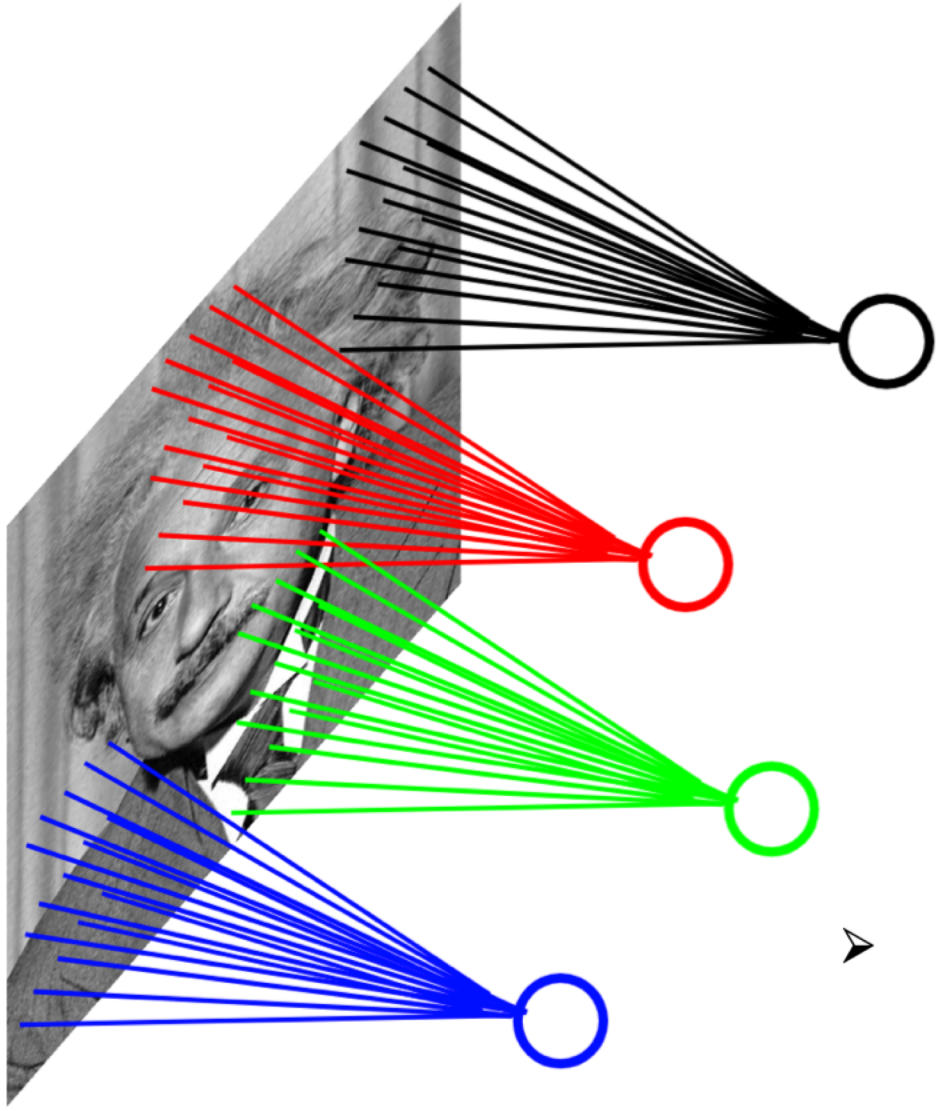


200 x 200 image

Local Connectivity

- Assume cell (= filter) size of 10x10 in a grid (20 x 20) and 40K hidden units
- How many parameters?
- What's the problem?

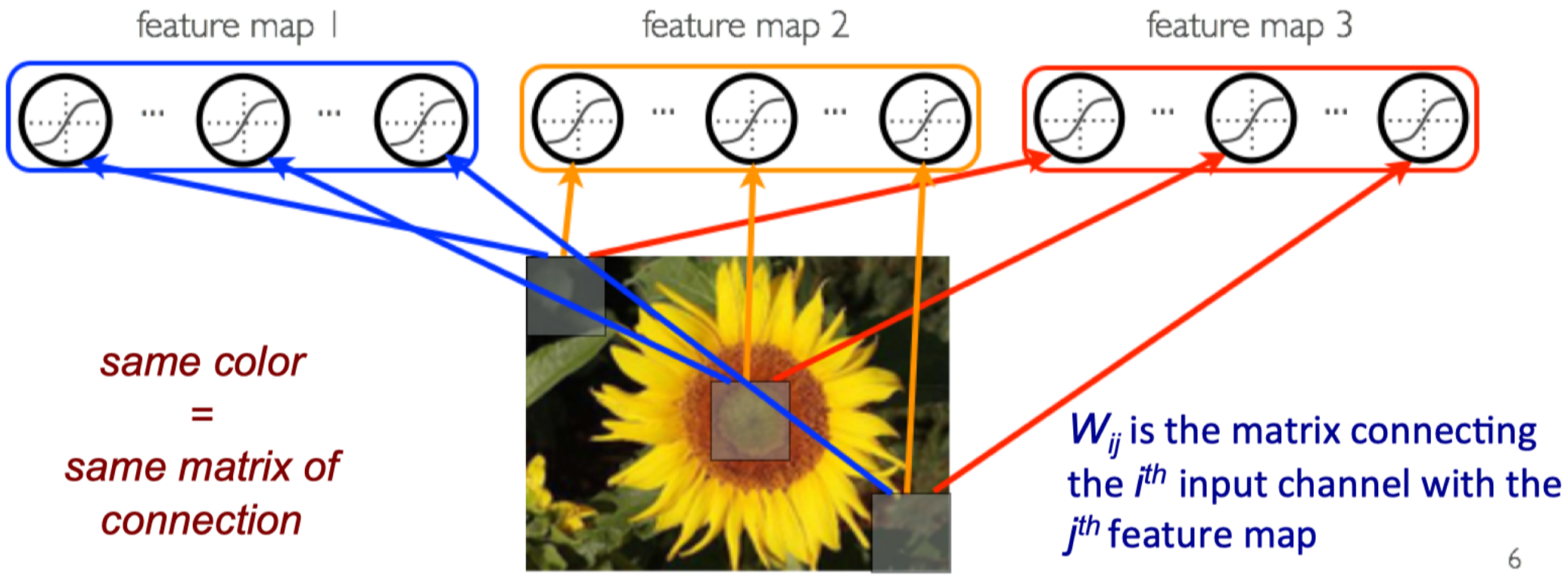




200 x 200 image

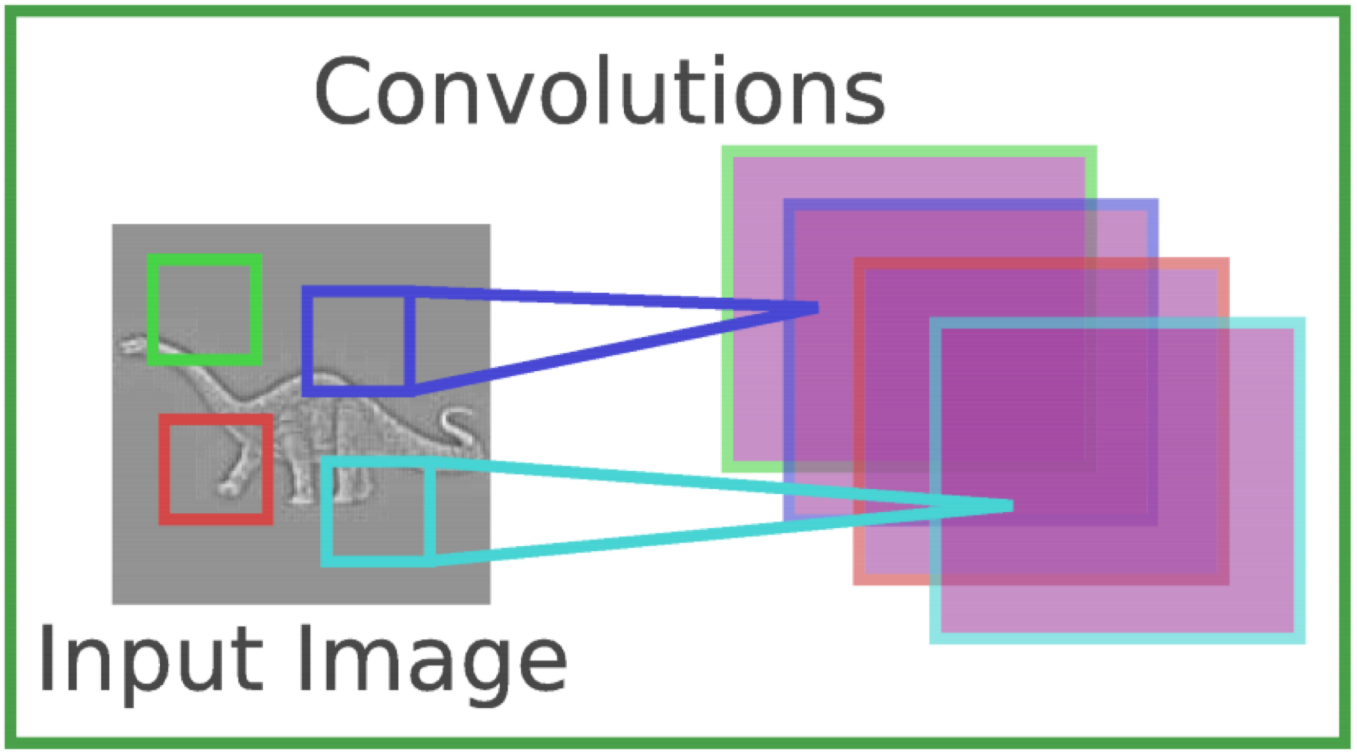
Parameter sharing

- Feedforward NN where hidden units are organized into multiple groups called **feature map** and hidden units in each feature map **share parameters**



Convolution

- Each feature map forms a 2D grid of features computed through **convolution** with a **sliding** filter over an image

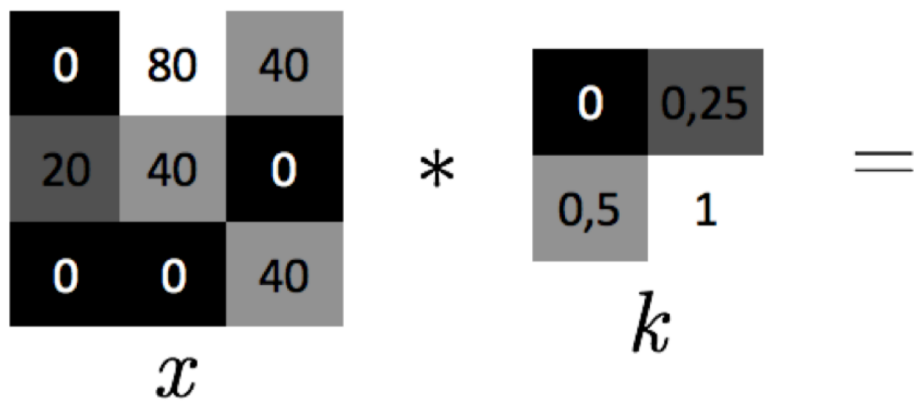


Discrete Convolution

• The convolution of an image x with a kernel k is computed as follows:

$$(x * k)_{ij} = \sum_{pq} x_{i+p,j+q} k_{r-p,r-q}$$

• Example:

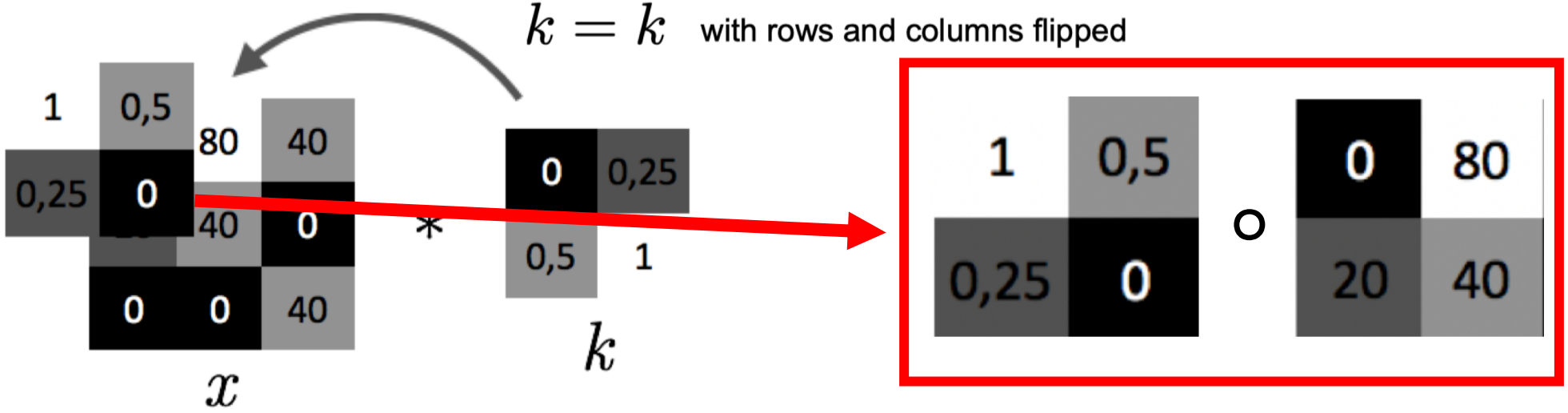


Discrete Convolution

- The convolution of an image x with a kernel k is computed as follows:

$$(x * k)_{ij} = \sum_{pq} x_{i+p,j+q} k_{r-p,r-q}$$

- Example:

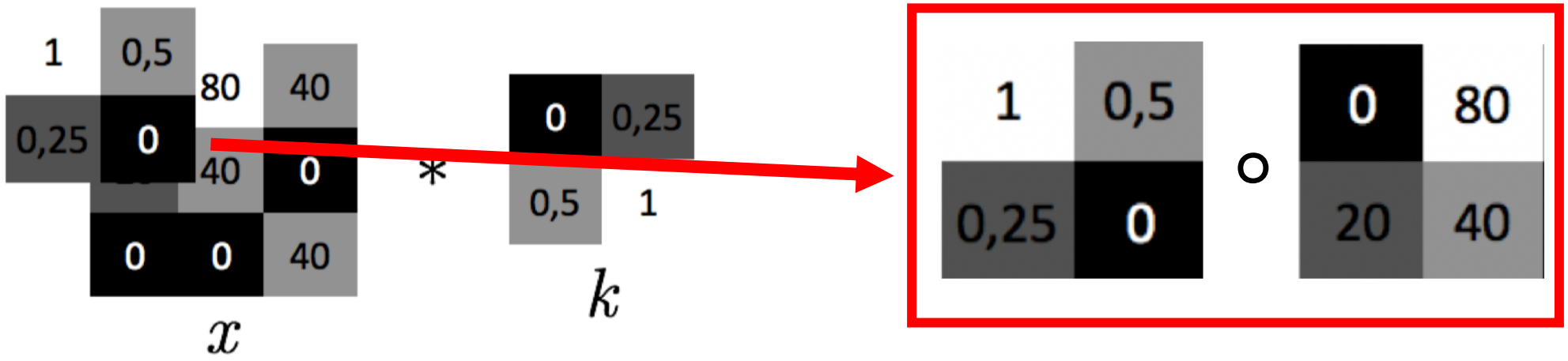


Discrete Convolution

• The convolution of an image x with a kernel k is computed as follows:

$$(x * k)_{ij} = \sum_{pq} x_{i+p,j+q} k_{r-p,r-q}$$

• Example: 1 x 0 + 0.5 x 80 + 0.25 x 20 + 0 x 40 =

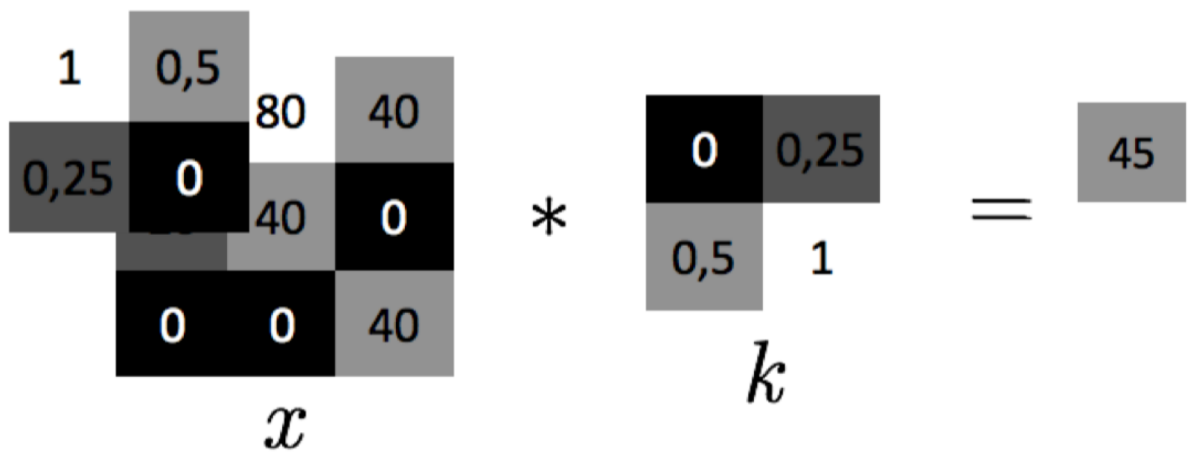


Discrete Convolution

• The convolution of an image x with a kernel k is computed as follows:

$$(x * k)_{ij} = \sum_{pq} x_{i+p,j+q} k_{r-p,r-q}$$

• Example: 1 x 0 + 0.5 x 80 + 0.25 x 20 + 0 x 40 = 45

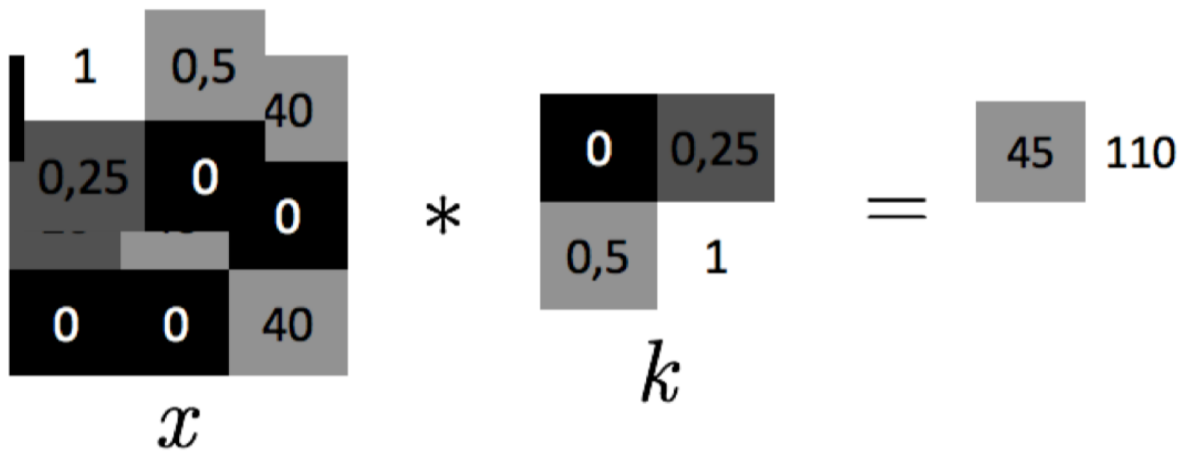


Discrete Convolution

• The convolution of an image x with a kernel k is computed as follows:

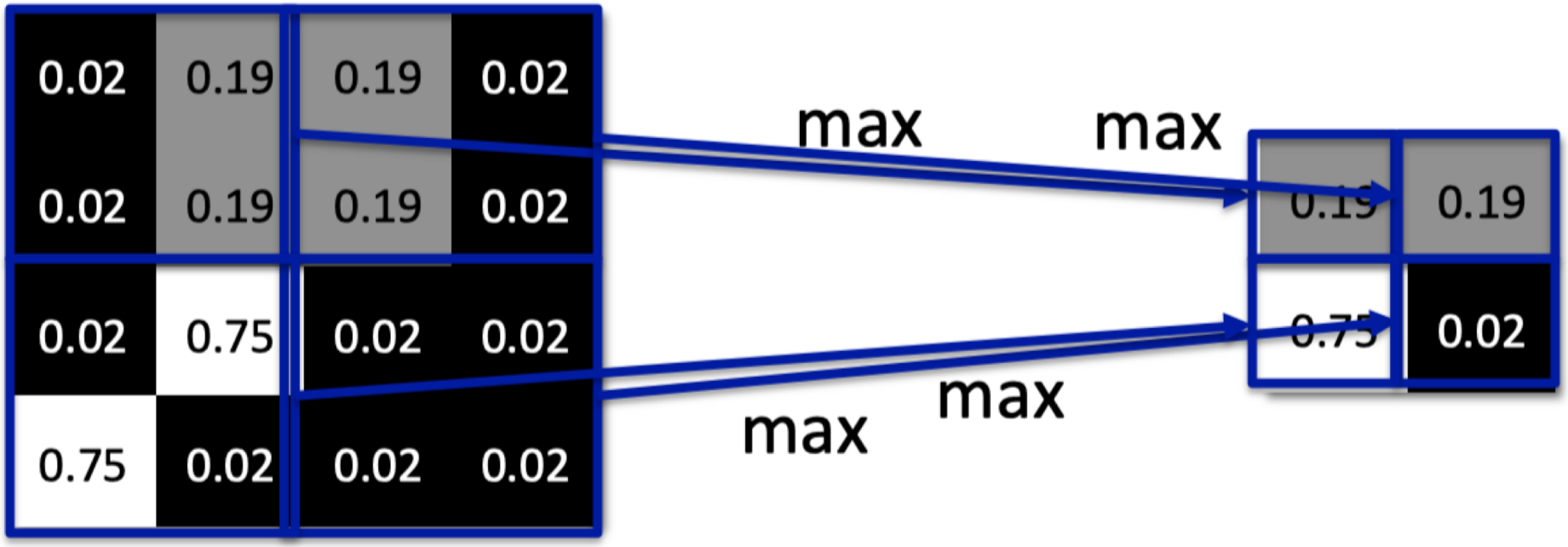
$$(x * k)_{ij} = \sum_{pq} x_{i+p,j+q} k_{r-p,r-q}$$

• Example: 1 x 80 + 0.5 x 40 + 0.25 x 40 + 0 x 0 = 110



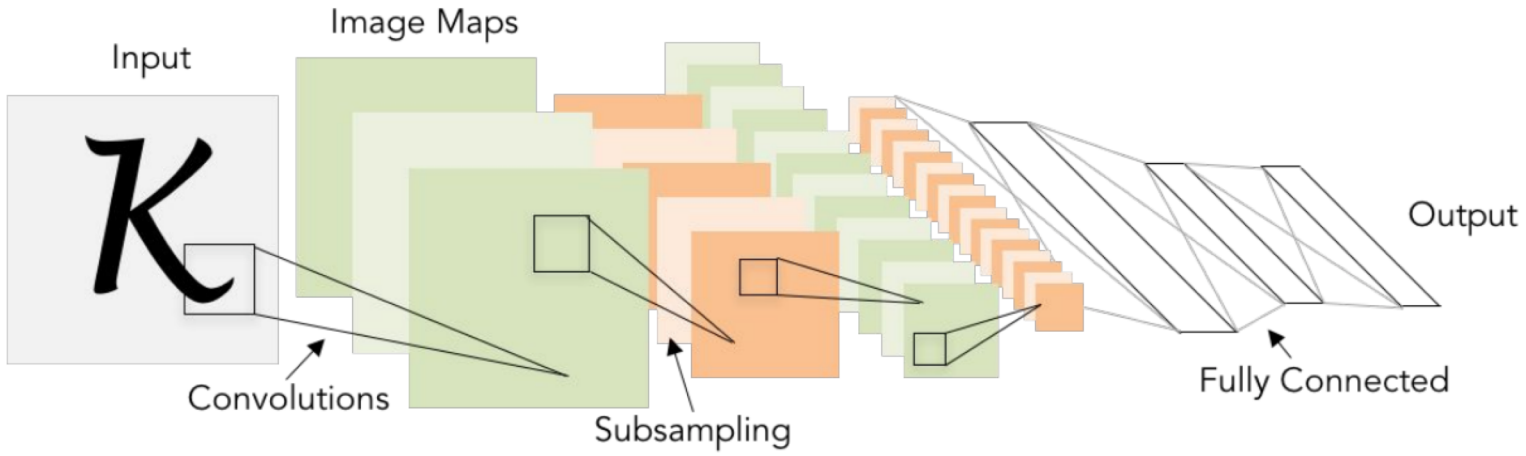
Pooling/Subsampling

- Pooling is performed in non-overlapping neighborhoods
- What are the advantages?



Convolutional Neural Network in a Nutshell

- Local connectivity
- Parameter sharing
- Convolution
- Pooling / subsampling hidden units



LeNet-5

How does it work in mathematical detail?

- Deep learning textbook, Chapter 9
- http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture05.pdf

Recurrent Neural Network

Application: Sequential Data (Stock Market)

Application: Sequential Data (Language)

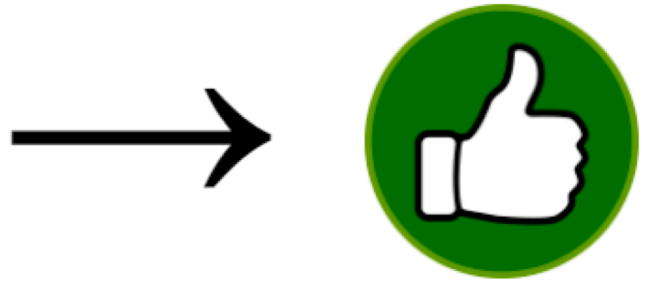
"I love this movie.
I've seen it many times
and it's still awesome."

"This movie is bad.
I don't like it it all.
It's terrible."

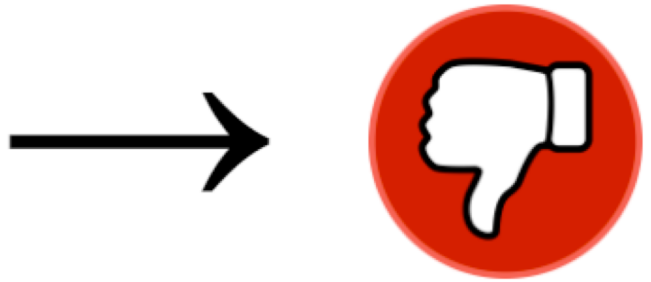
Shared Objective

- Design algorithms that can process **sequential** data to accomplish a given task (e.g. sentiment classification)

"I love this movie.
I've seen it many times
and it's still awesome."



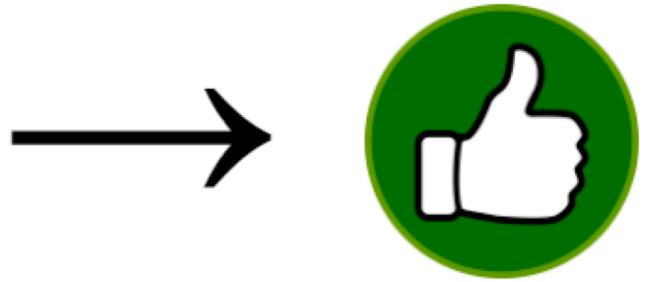
"This movie is bad.
I don't like it it all.
It's terrible."



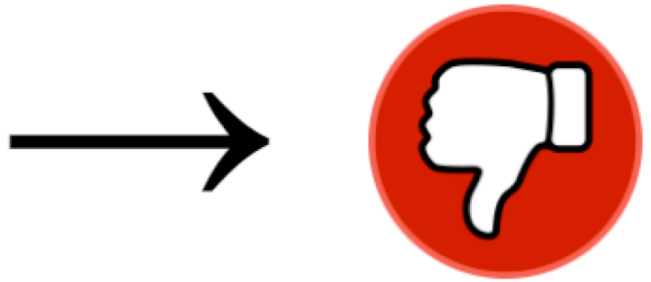
Shared Objective

- Design a **neural network** that can process **sequential** data to accomplish a given task (e.g. sentiment classification)

"I love this movie.
I've seen it many times
and it's still awesome."

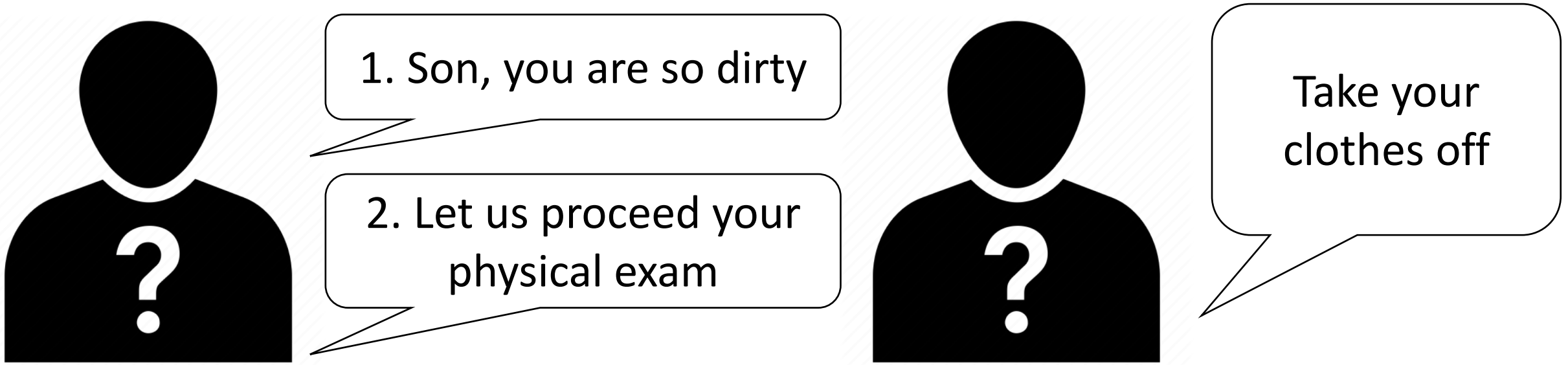


"This movie is bad.
I don't like it it all.
It's terrible."



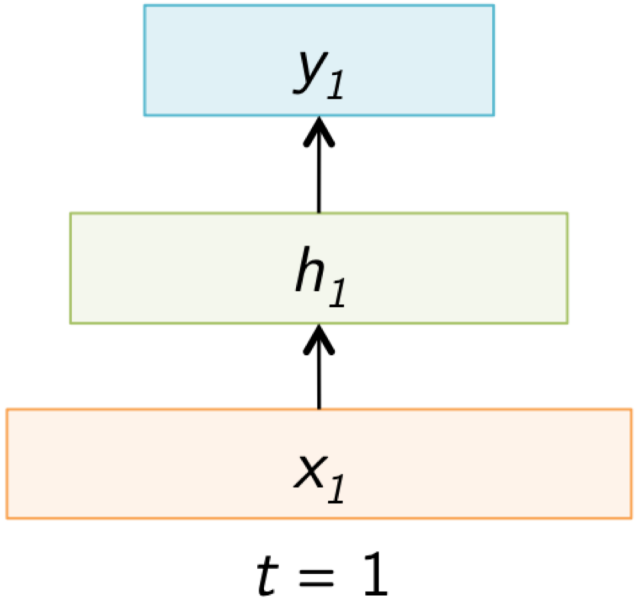
Characteristics of Sequential Data

- **Variable-length** input: “I like you” vs. “I think I like you”
- **Context** information: “Take your clothes off”
- **Variable-length** context window

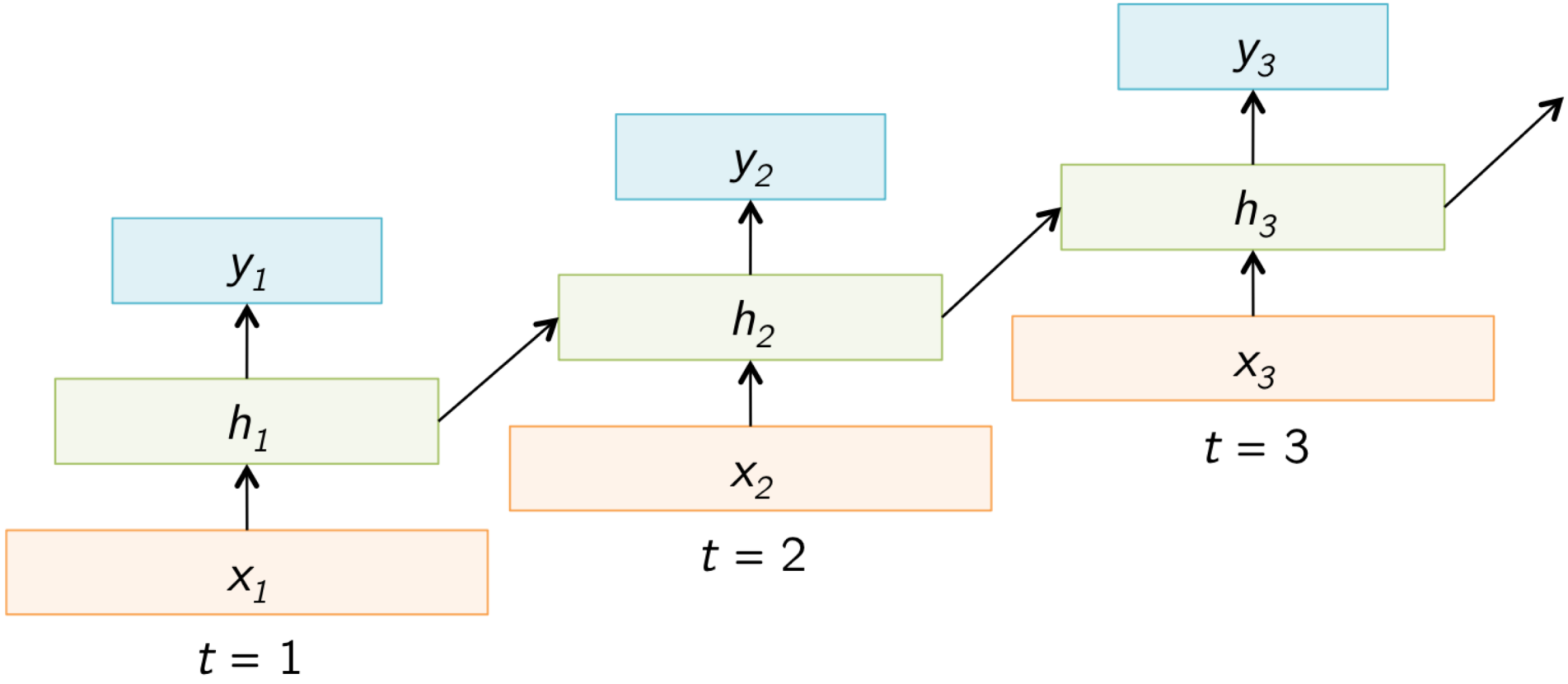


Naïve approach

- What's the problem?



Recurrent Neural Network (RNN)



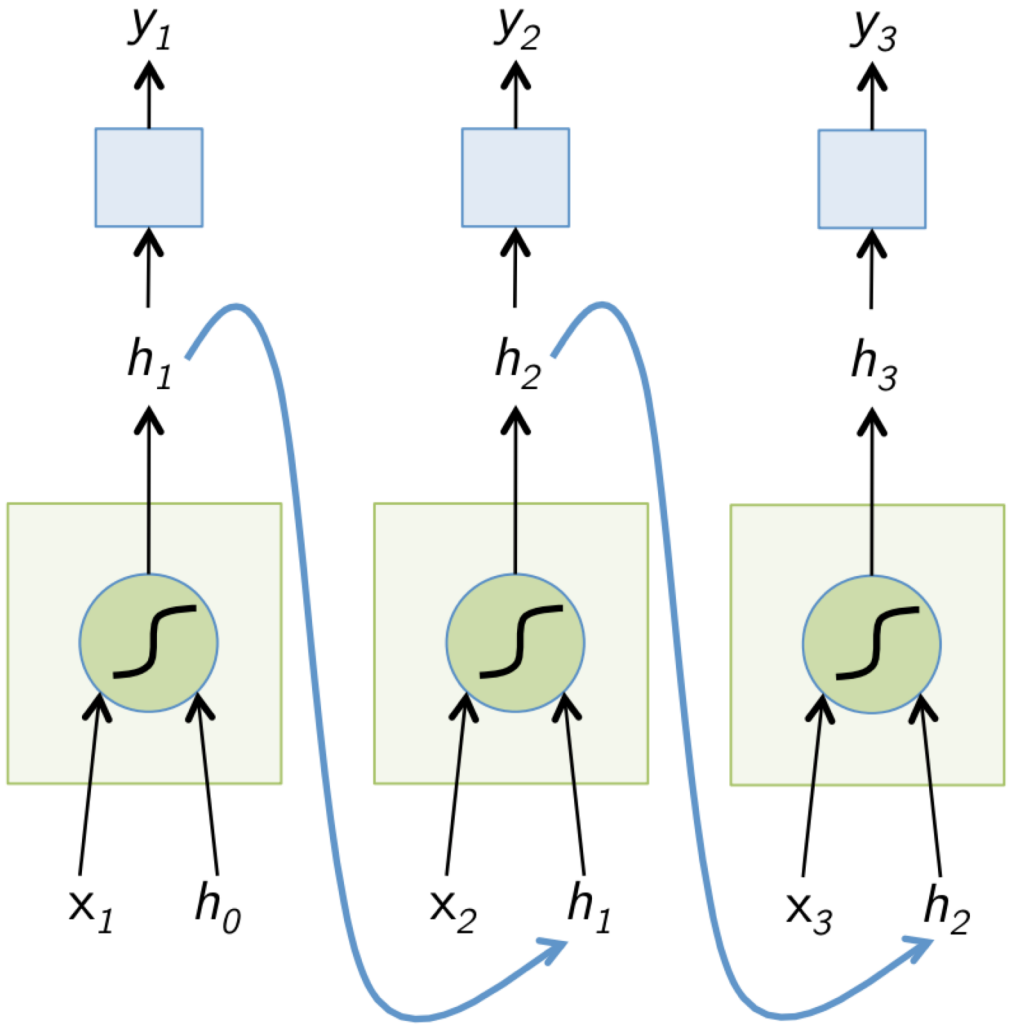
Recurrent Neural Network (RNN)

$$a_t = b_h + W_{hh} h_{t-1} + W_{xh} x_t$$

$$\mathrm{h_t} = \tanh(a_t)$$

$$\mathrm{o_t} = b_o + W_{hy} h_t$$

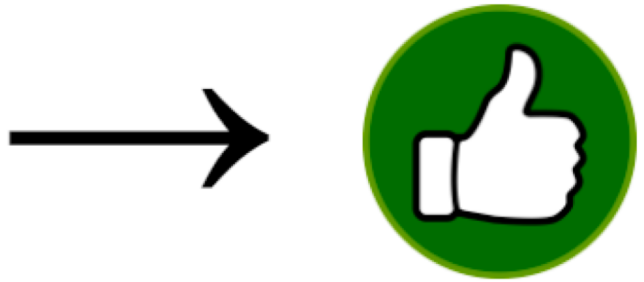
$$\mathrm{y_t} = softmax(o_t)$$



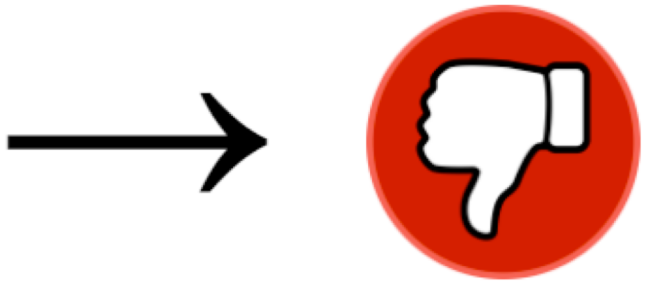
Example: Sentiment Classification

- Input?
- Output?

"I love this movie.
I've seen it many times
and it's still awesome."



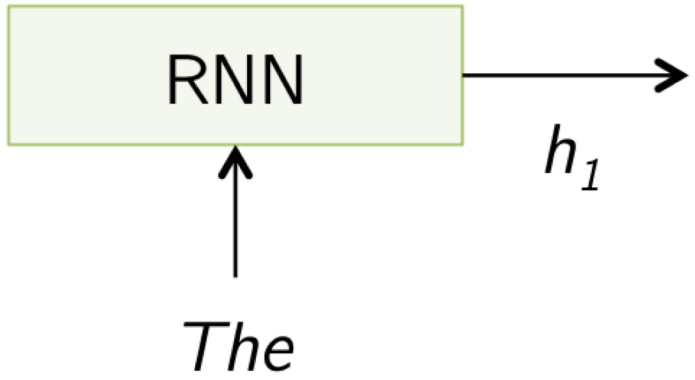
"This movie is bad.
I don't like it it all.
It's terrible."



Example: Sentiment Classification

Input : “The food is really good”

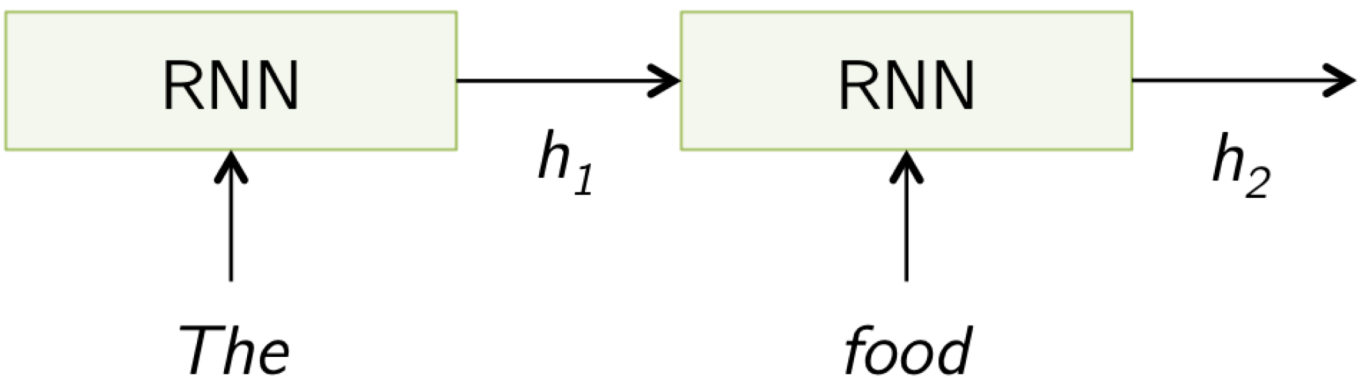
Output: 0 or 1 (Negative / Positive)



Example: Sentiment Classification

Input : “The food is really good”

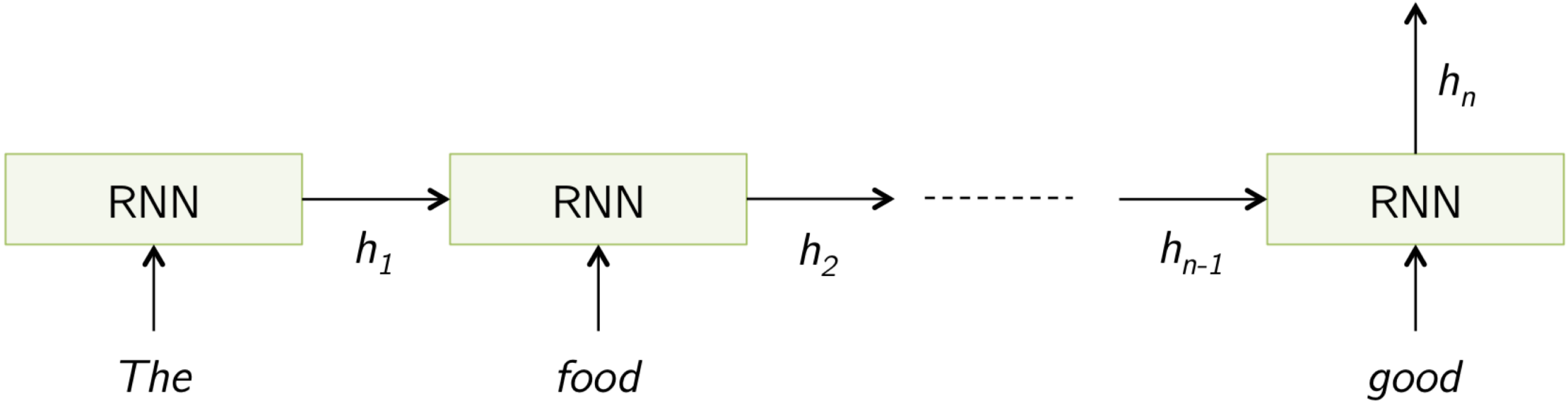
Output: 0 or 1 (Negative / Positive)



Example: Sentiment Classification

Input : “The food is really good”

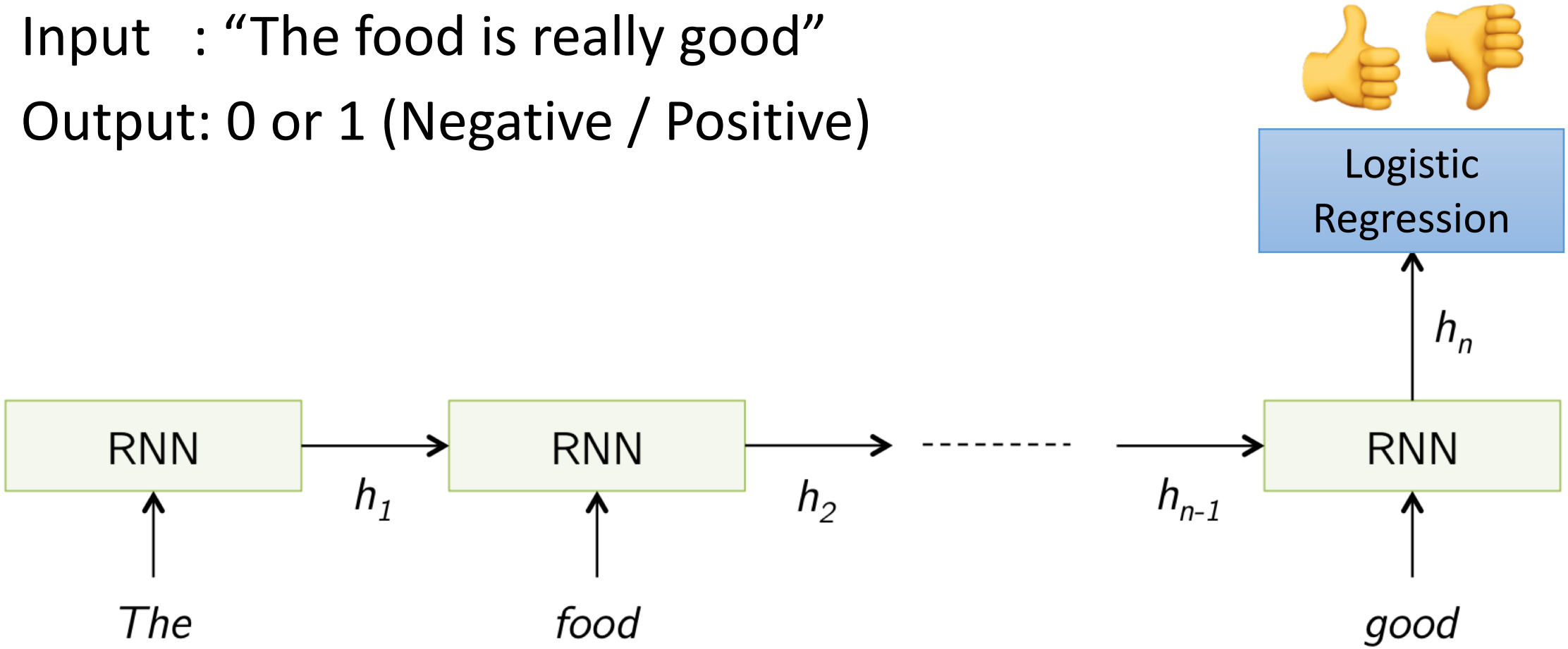
Output: 0 or 1 (Negative / Positive)



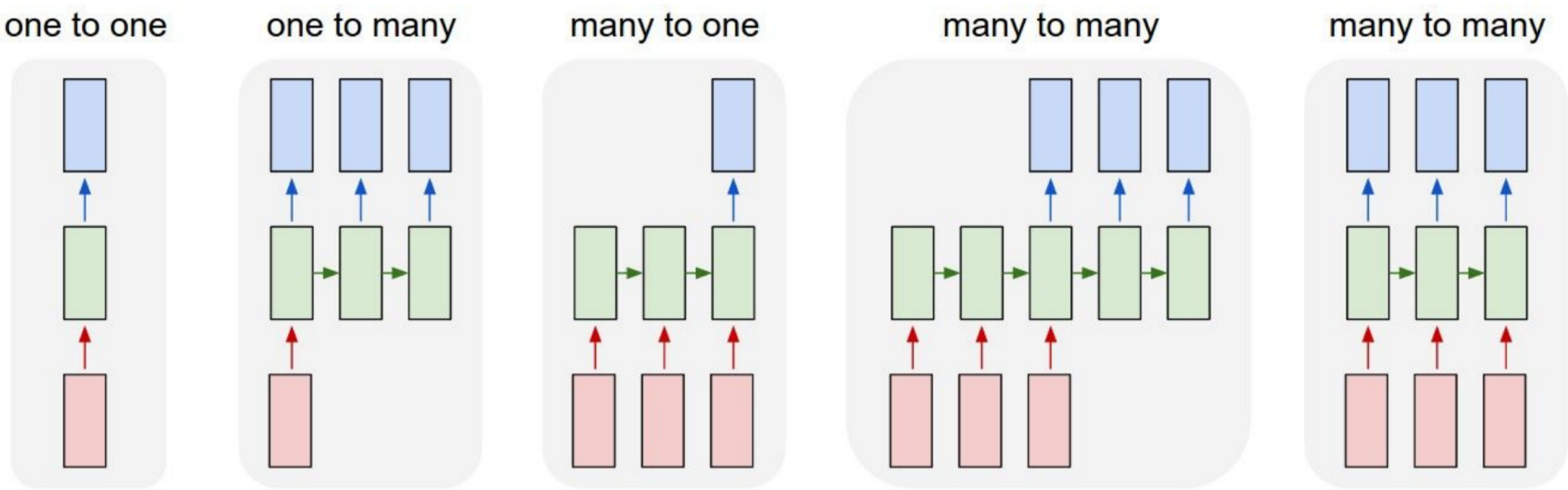
Example: Sentiment Classification

Input : “The food is really good”

Output: 0 or 1 (Negative / Positive)



Variants of RNN



How does it work in mathematical detail?

- Deep learning textbook, Chapter 10
- http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf
- https://www.cs.ubc.ca/labs/lci/mlrg/slides/rnn.pdf

Take-home Messages

- Key question to design neural network solution for domain: How do we parametrize neural networks based on domain characteristic?
  - In other words, how do we share parameters effectively?
- CNN & RNN are amazing representation learning tool changing the world
  - Visual data → CNN, Sequential data → RNN
- CNN processes grid-like topology by employing specialized operations such as convolution and pooling.
- RNN processes variable-length inputs and outputs by maintaining state information across time steps