

# CS230: Lecture 2

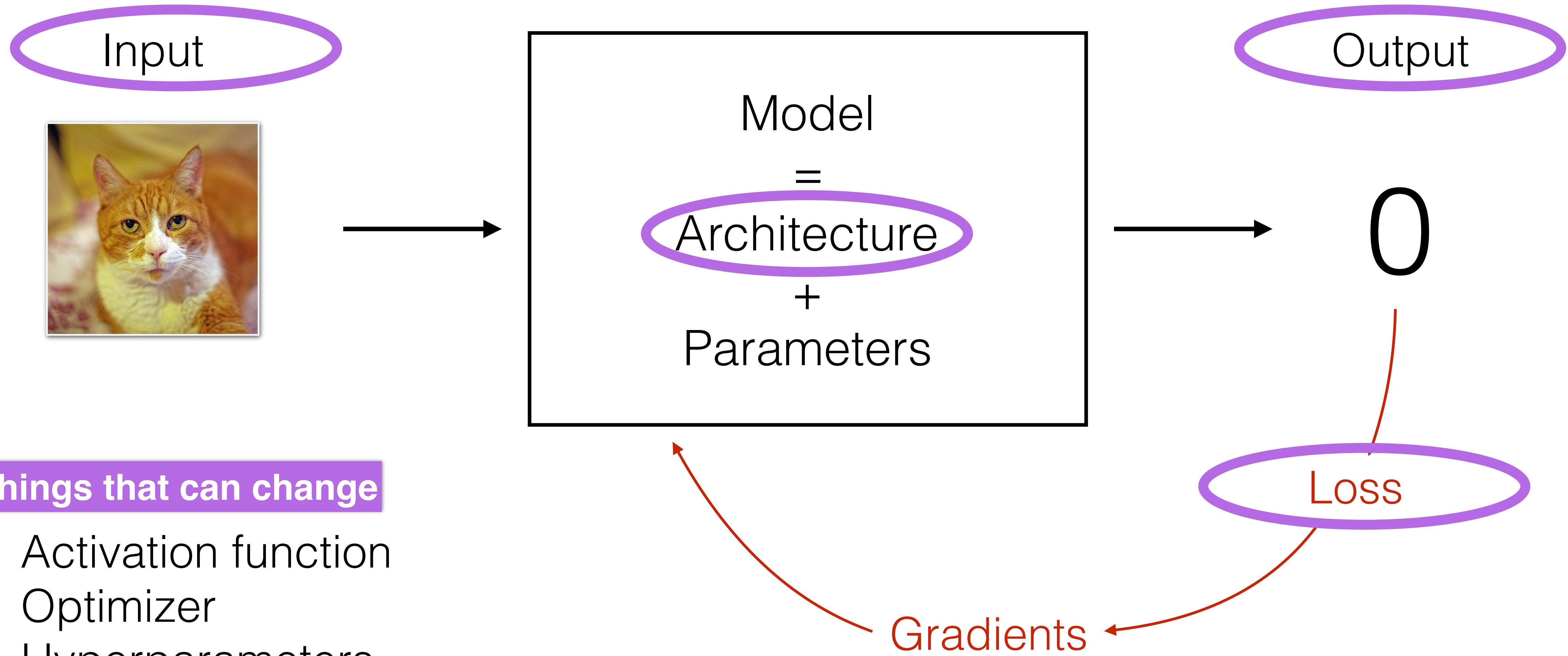
## Decision making in AI projects

Kian Katanforoosh

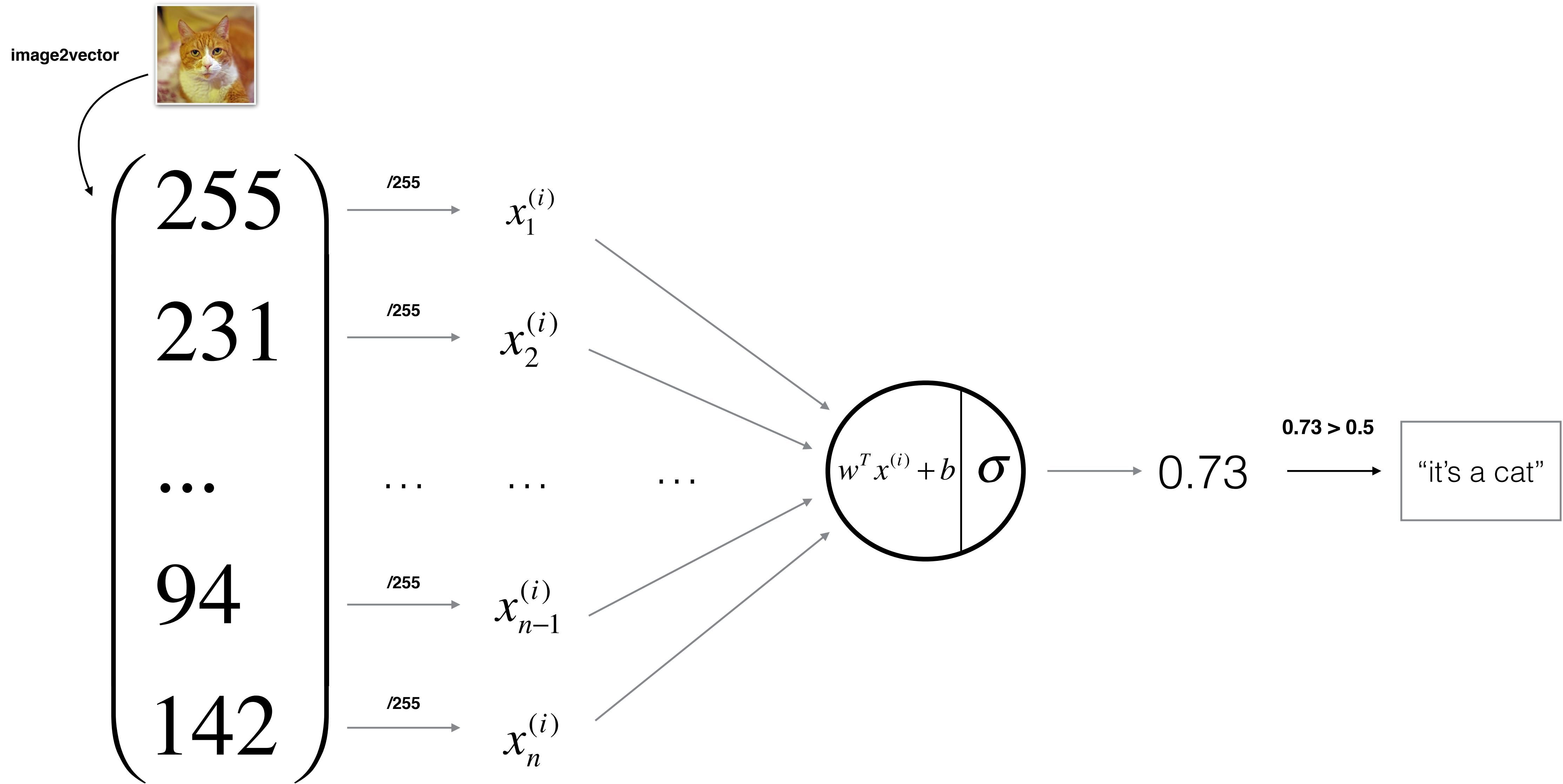


Recap of the week

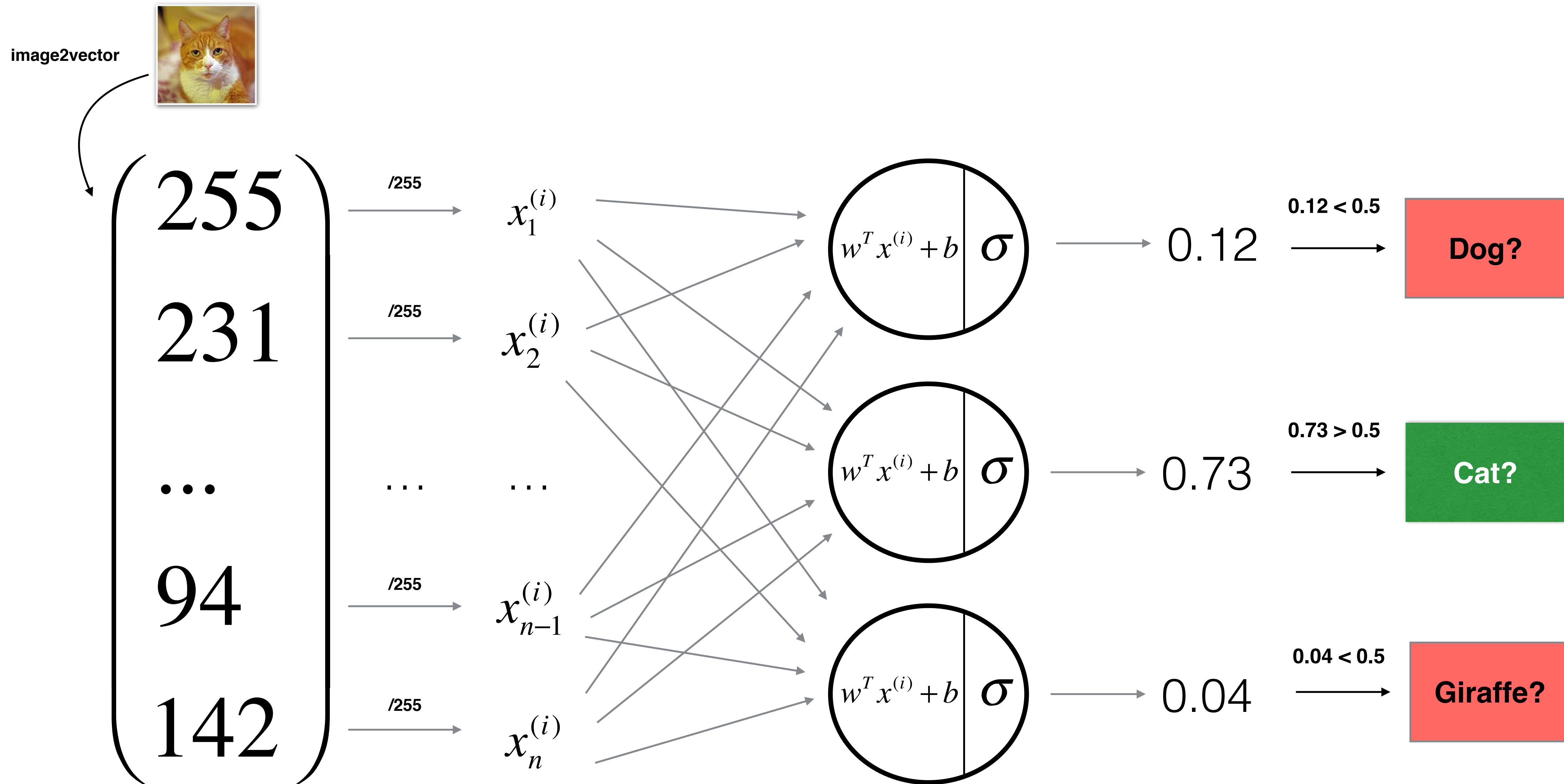
# Learning Process



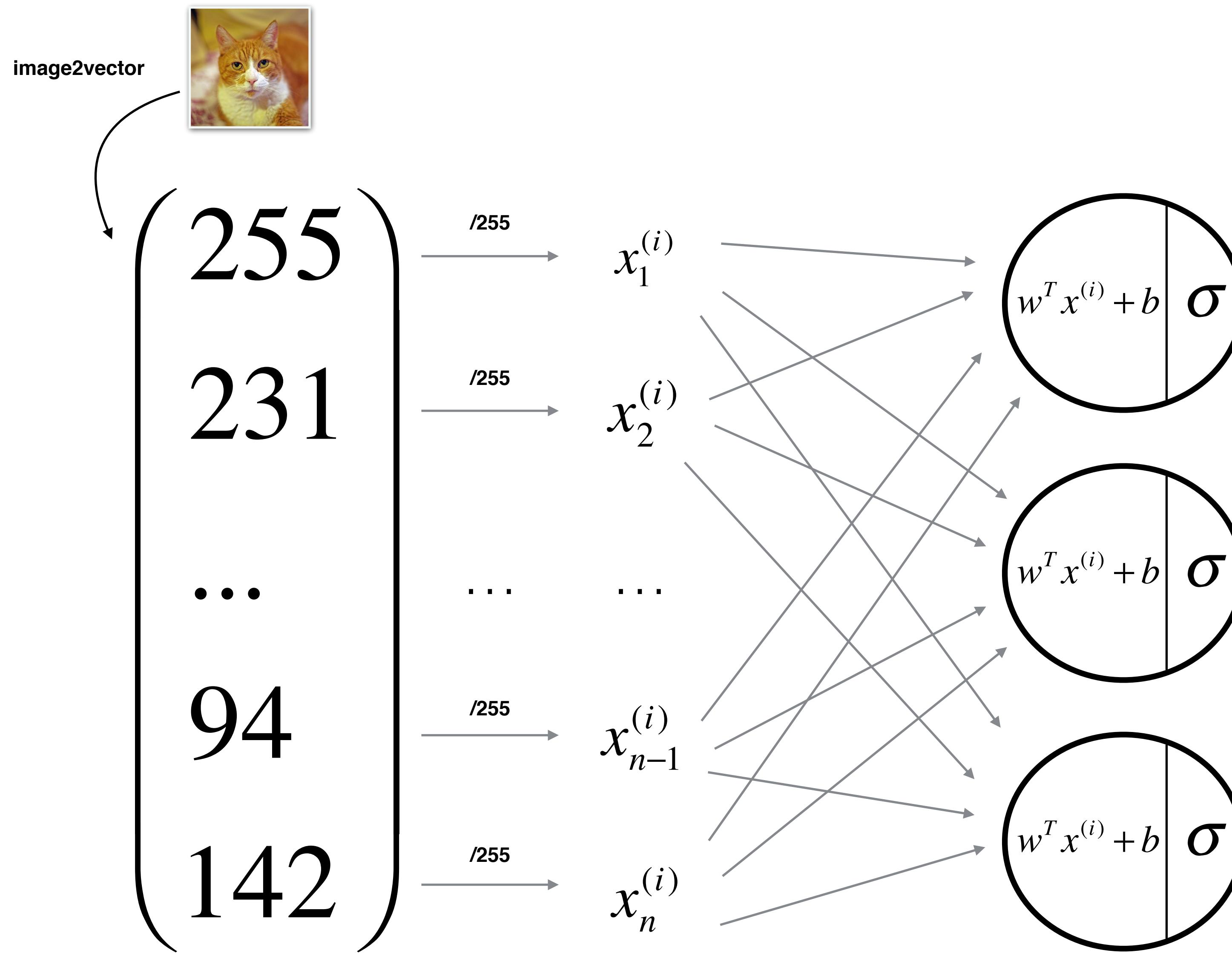
# Logistic Regression as a Neural Network



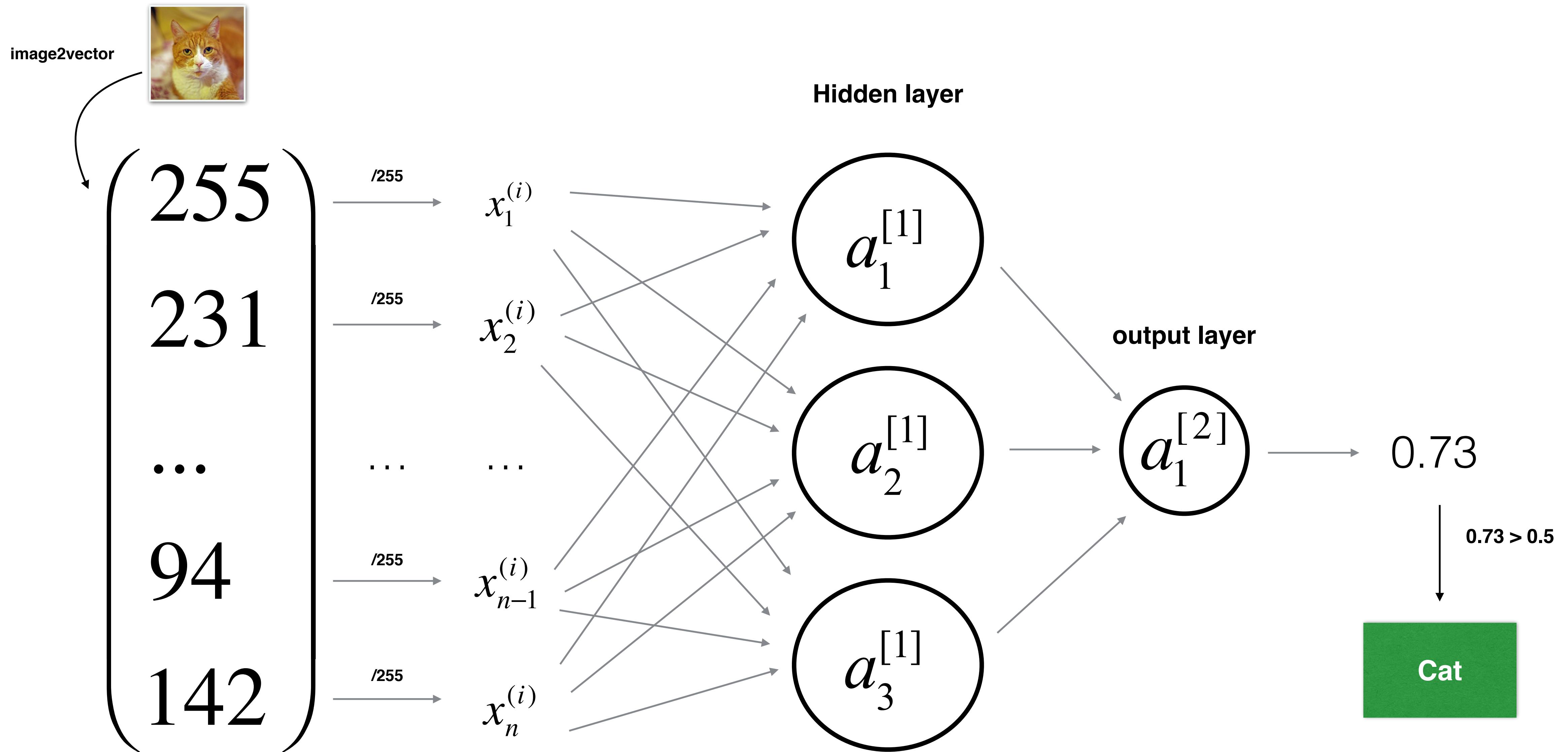
# Multi-class



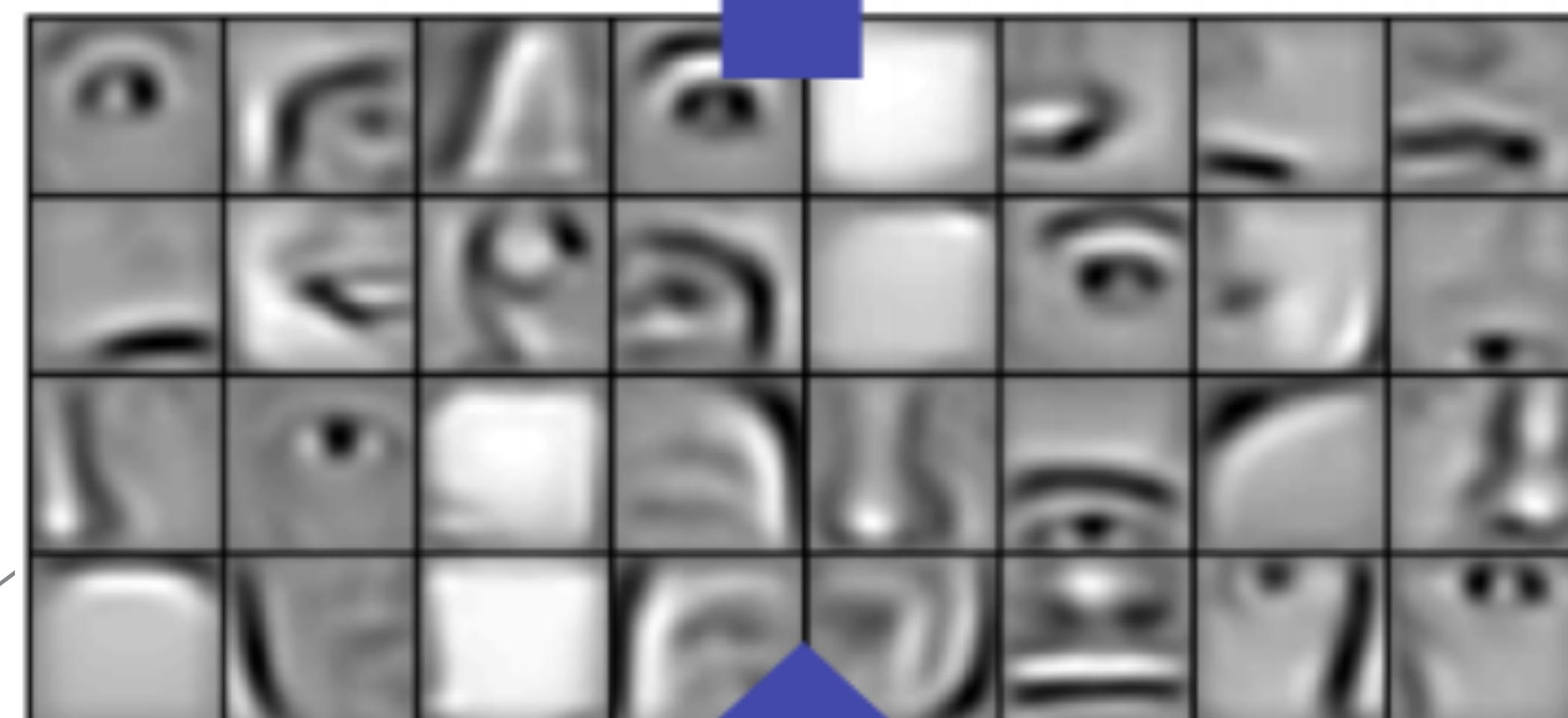
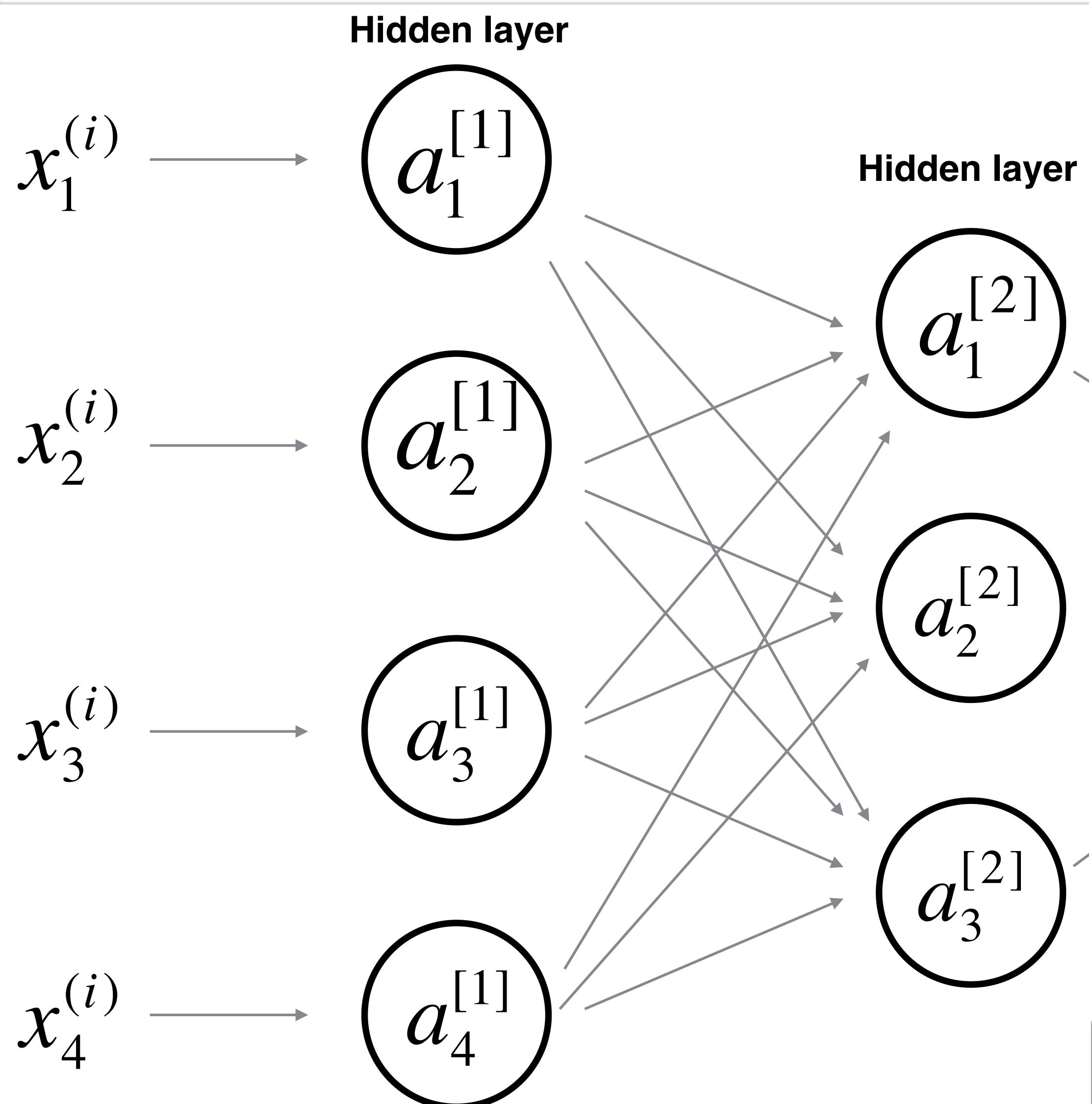
# Neural Network (Multi-class)



# Neural Network (1 hidden layer)



# Deeper net



Technique called “encoding”

# Summary of learnings: Introduction

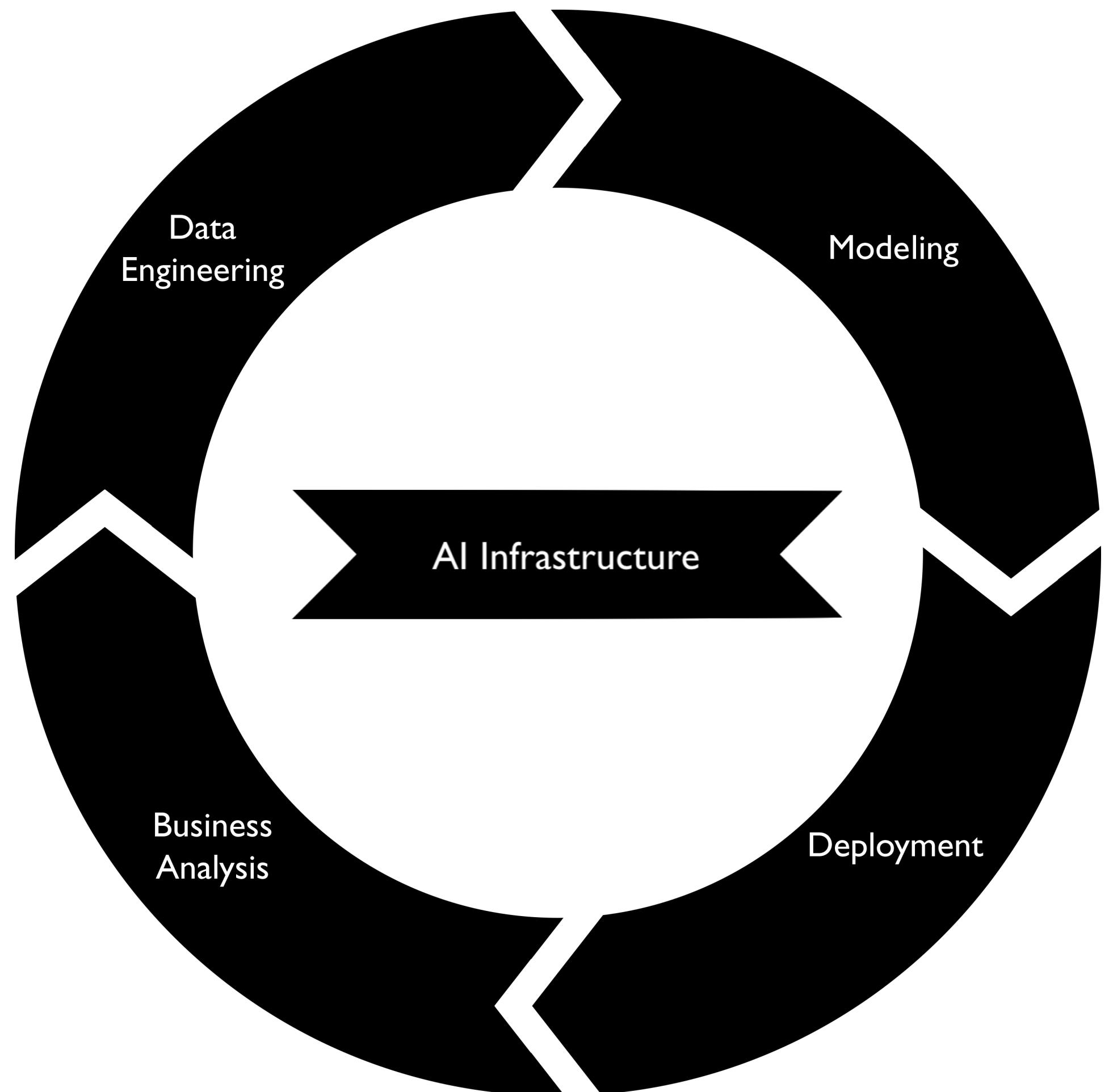
- A **model** is defined by its **architecture** and its **parameters**.
- The labelling strategy matters to successfully train your models. For example, if you're training a 3-class (dog, cat, giraffe) classifier under the constraint of one animal per picture, you might use **one-hot vectors** to label your data.
- We introduced a set of **notations** to differentiate indices for neurons, layers and examples.
- In deep learning, **feature learning** replaces **feature engineering**.



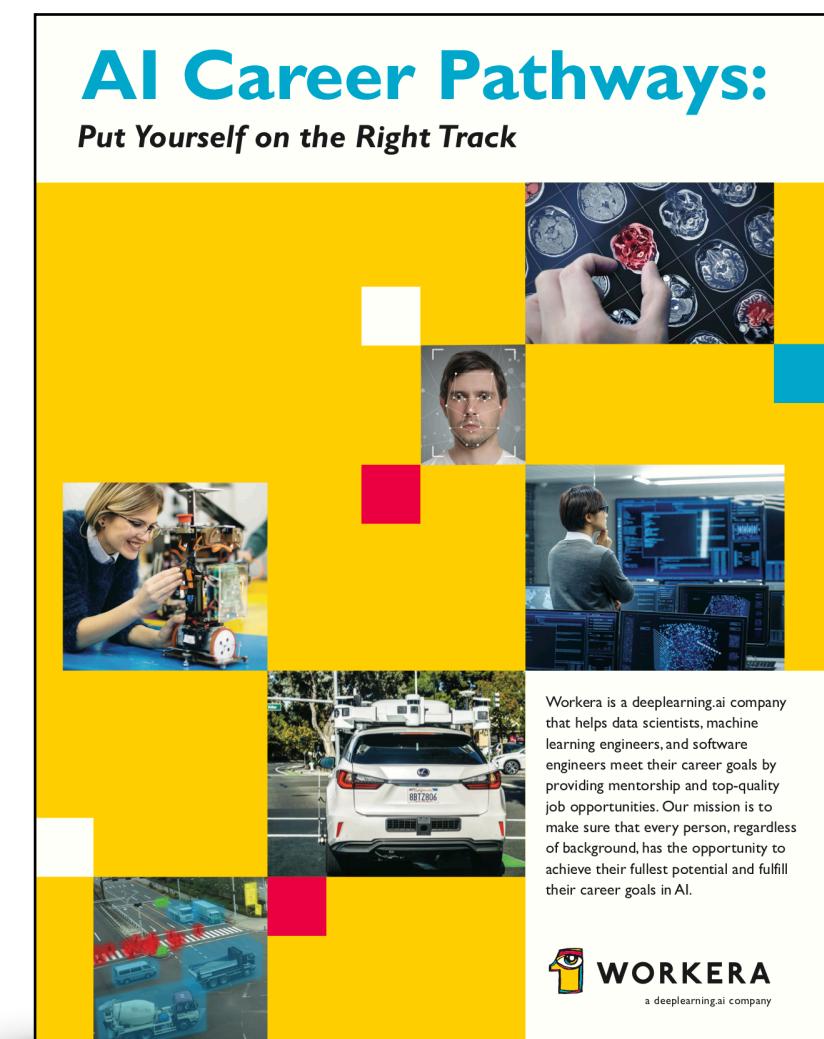
## ~~Recap of the week~~

Let's now talk about decision making and build intuition on concrete applications

# What skills matter to carry out AI projects?

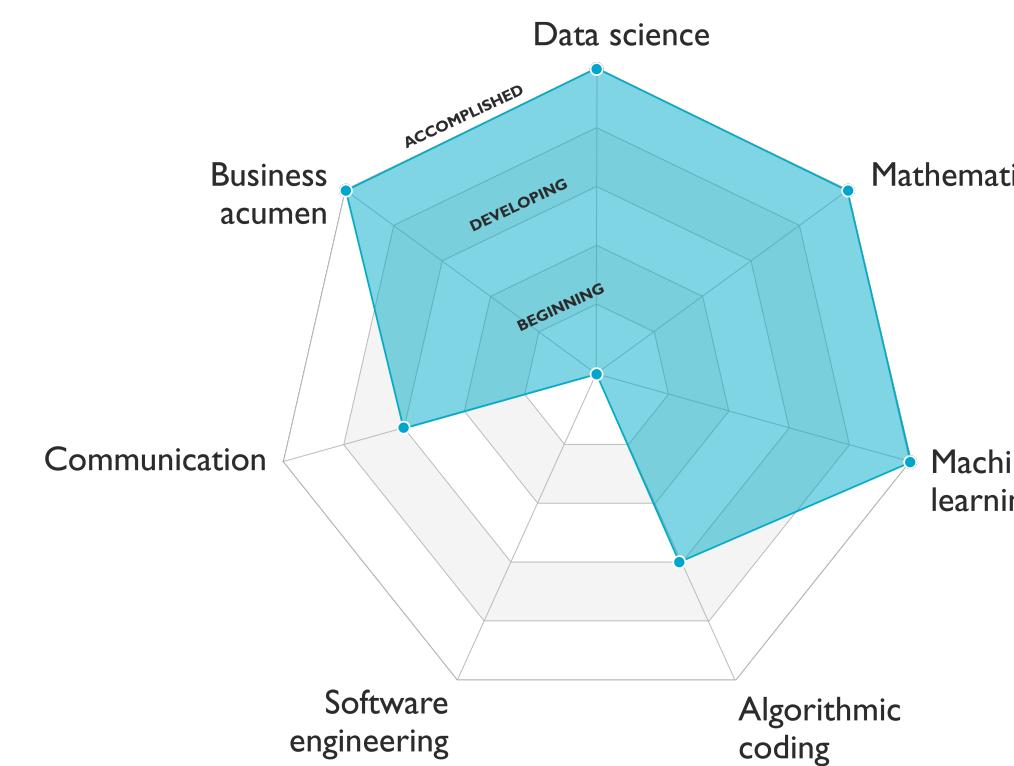


The AI project development lifecycle

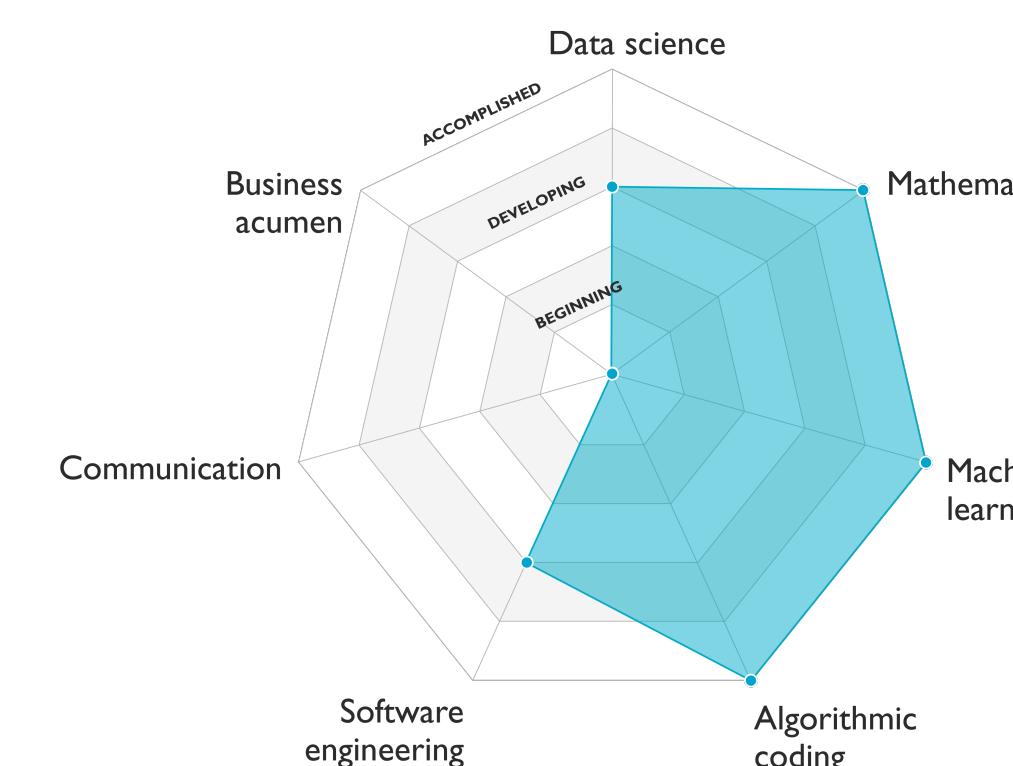


AI career pathways report (2020)  
(Optional reading)

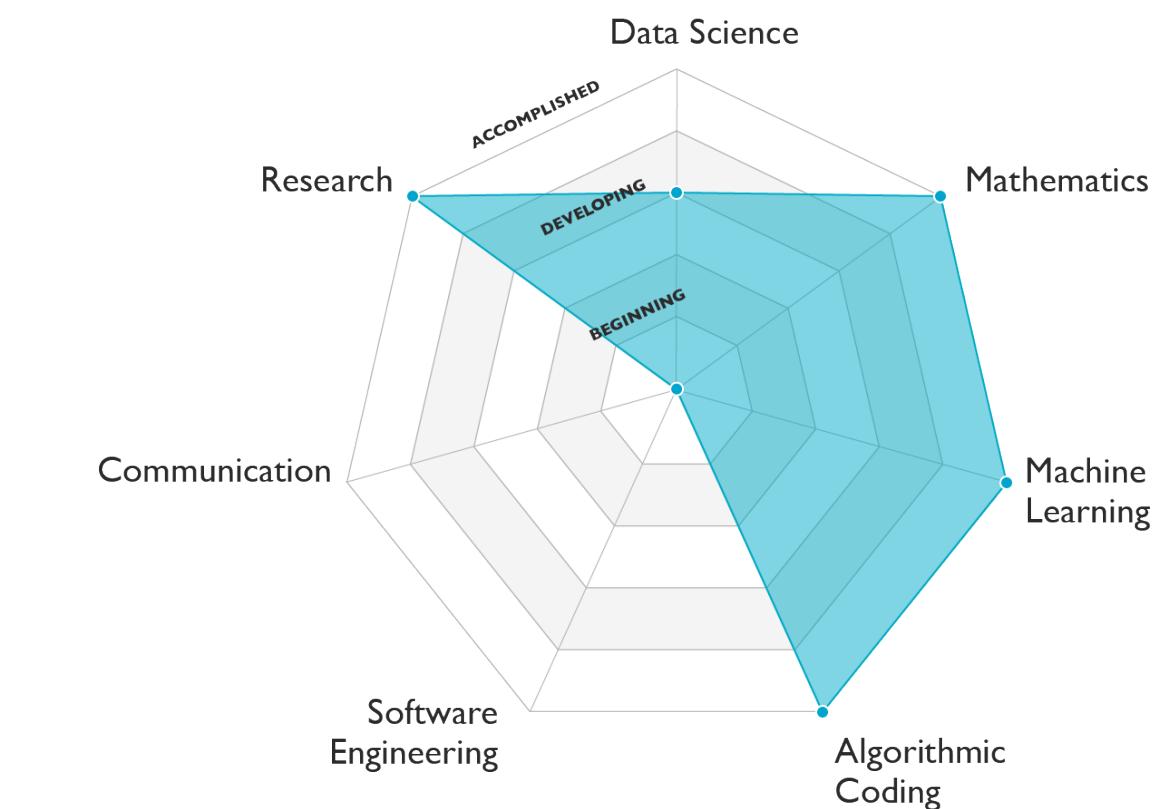
# What skills matter to carry out AI projects?



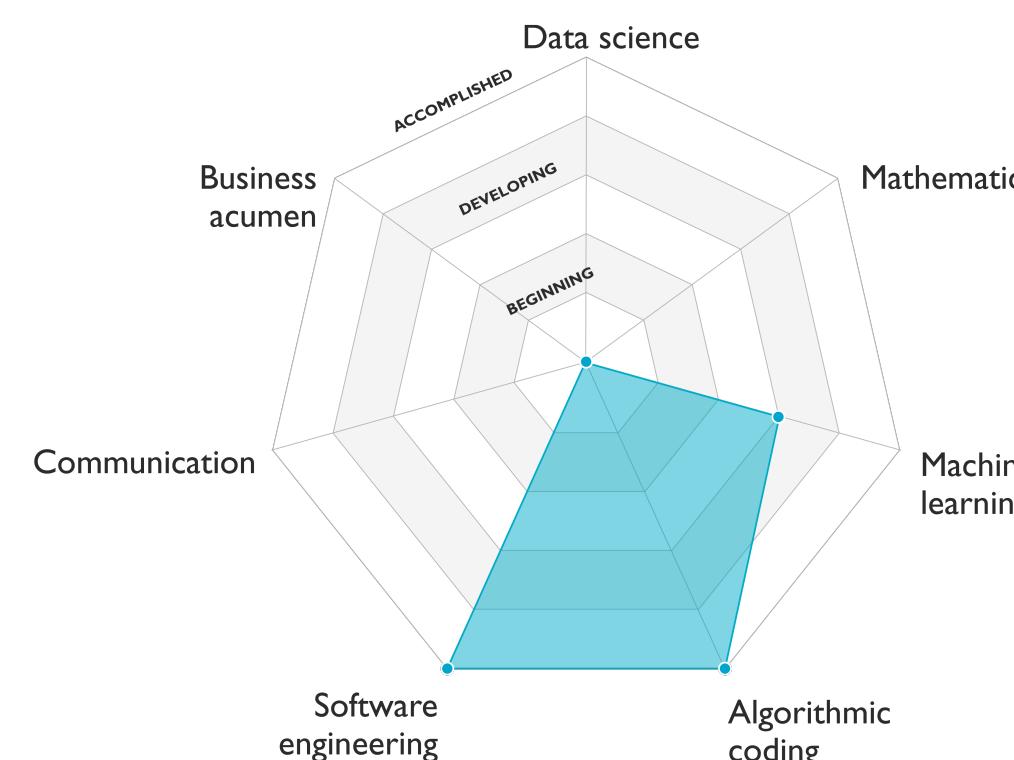
**Data Scientist**



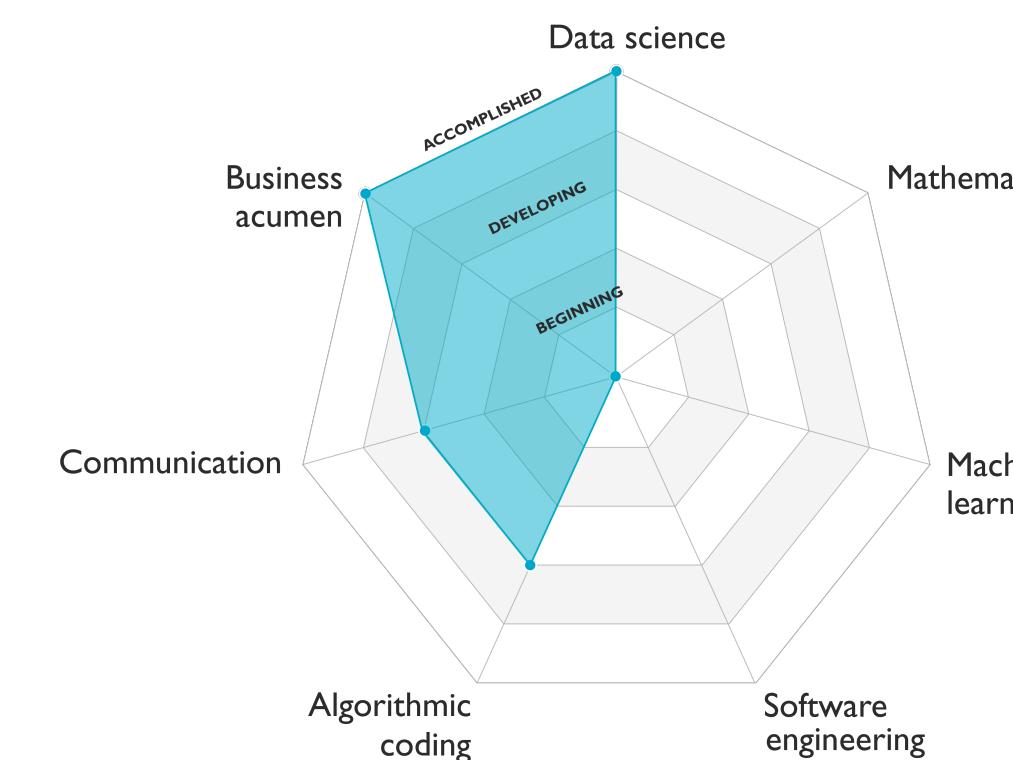
**Machine Learning Engineer**



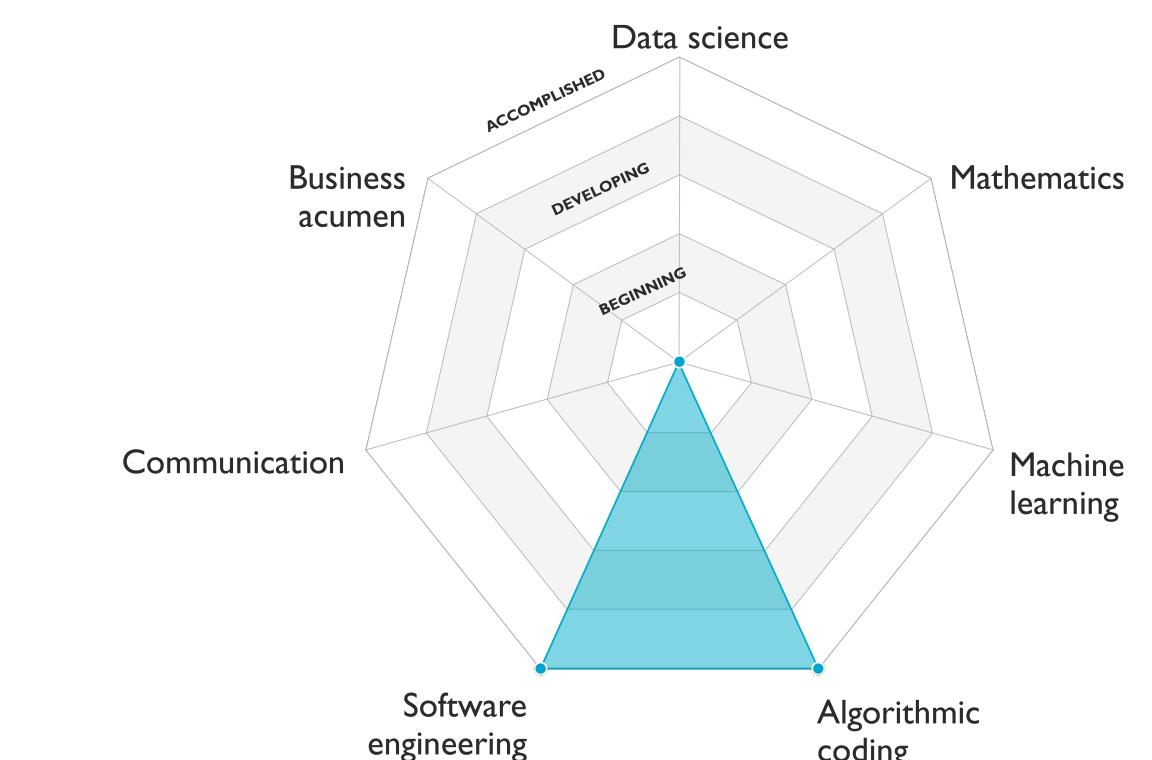
**Machine Learning Research**



**Software Engineer-Machine Learning**



**Data Analyst**



**Software Engineer**

The necessary skills to carry out the tasks of the AI project development lifecycle are a combination of scientific, engineering, behavioral, and decision making skills.

In the rest of this presentation, we will illustrate **AI decision making skills** through real **case studies**. The goal is to learn war stories that you can refer to for your own AI projects.

We will learn to **pose a ML problem, break down a complex ML project into pieces, choose a loss and a training strategy**.

## I. Day 'n' Night classification

## II. Face verification

## III. Neural style transfer (Art generation)

## IV. Trigger-word detection

# Case study 1: Day 'n' Night classification

**Goal:** Given an image, classify as taken “during the day” (0) or “during the night” (1)

**1. Data?**

10,000 images

Split? Bias?



**2. Input?**

Resolution?

(64, 64, 3)

**3. Output?**

y = 0 or y = 1

Last Activation?

sigmoid

**4. Architecture ?**

A shallow CNN should do the job pretty well

**5. Loss?**

$$L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Easy warm up

## Summary of learnings: Day 'n' Night classification

- Use a known **proxy project** to evaluate how much data you need.
- Be scrappy. For example, if you'd like to find a good resolution of images to use for your data, but don't have time for a large scale experiment, **approximate human-level performance by testing your friends** as classifiers.

## Case study 2: Face Verification

**Goal:** A school wants to use Face Verification for validating student IDs in facilities (dinning halls, gym, pool ...)

### 1. Data?

Picture of every student labelled with their name



Bertrand

### 2. Input?



Resolution?  
(412, 412, 3)

### 3. Output?

$y = 1$  (it's you)  
or  
 $y = 0$  (it's not you)

## Case study 2: Face Verification

**Goal:** A school wants to use Face Verification for validating student IDs in facilities (dinning halls, gym, pool ...)

### 4. What architecture?

Simple solution:



compute distance  
pixel per pixel  
if less than threshold  
then  $y=1$



database image

input image

Issues:

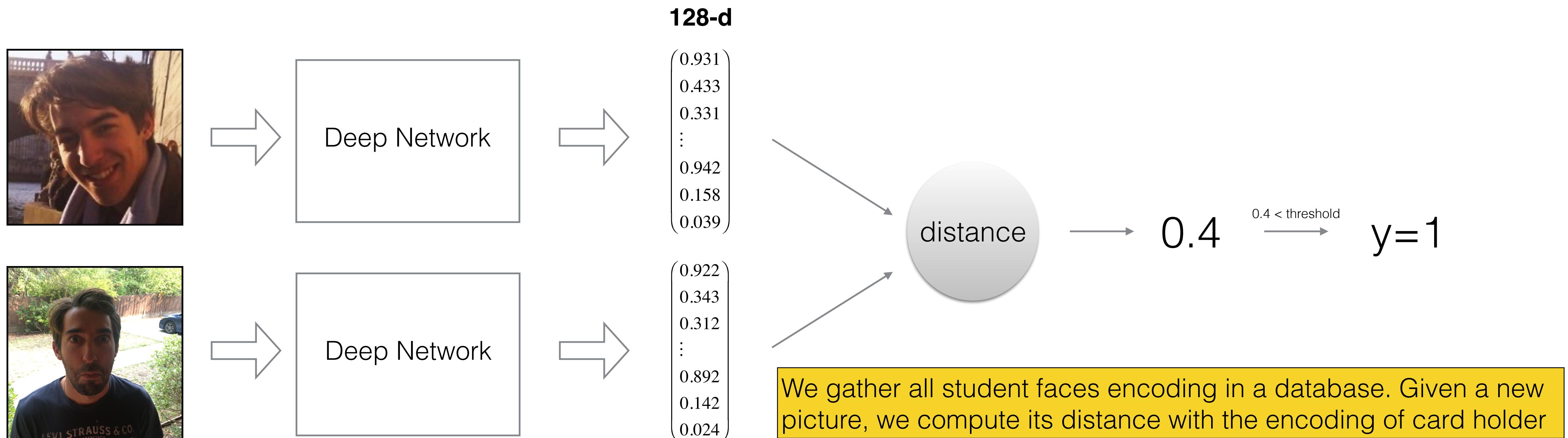
- Background lighting differences
- A person can wear make-up, grow a beard...
- ID photo can be outdated

## Case study 2: Face Verification

**Goal:** A school wants to use Face Verification for validating student IDs in facilities (dinning halls, gym, pool ...)

### 4. What architecture?

Our solution: encode information about a picture in a vector



## Case study 2: Face Verification

**Goal:** A school wants to use Face Verification for validating student IDs in facilities (dinning hall, gym, pool ...)

### 4. Loss? Training?

We need more data so that our model understands how to encode:  
Use public face datasets

What we really want:



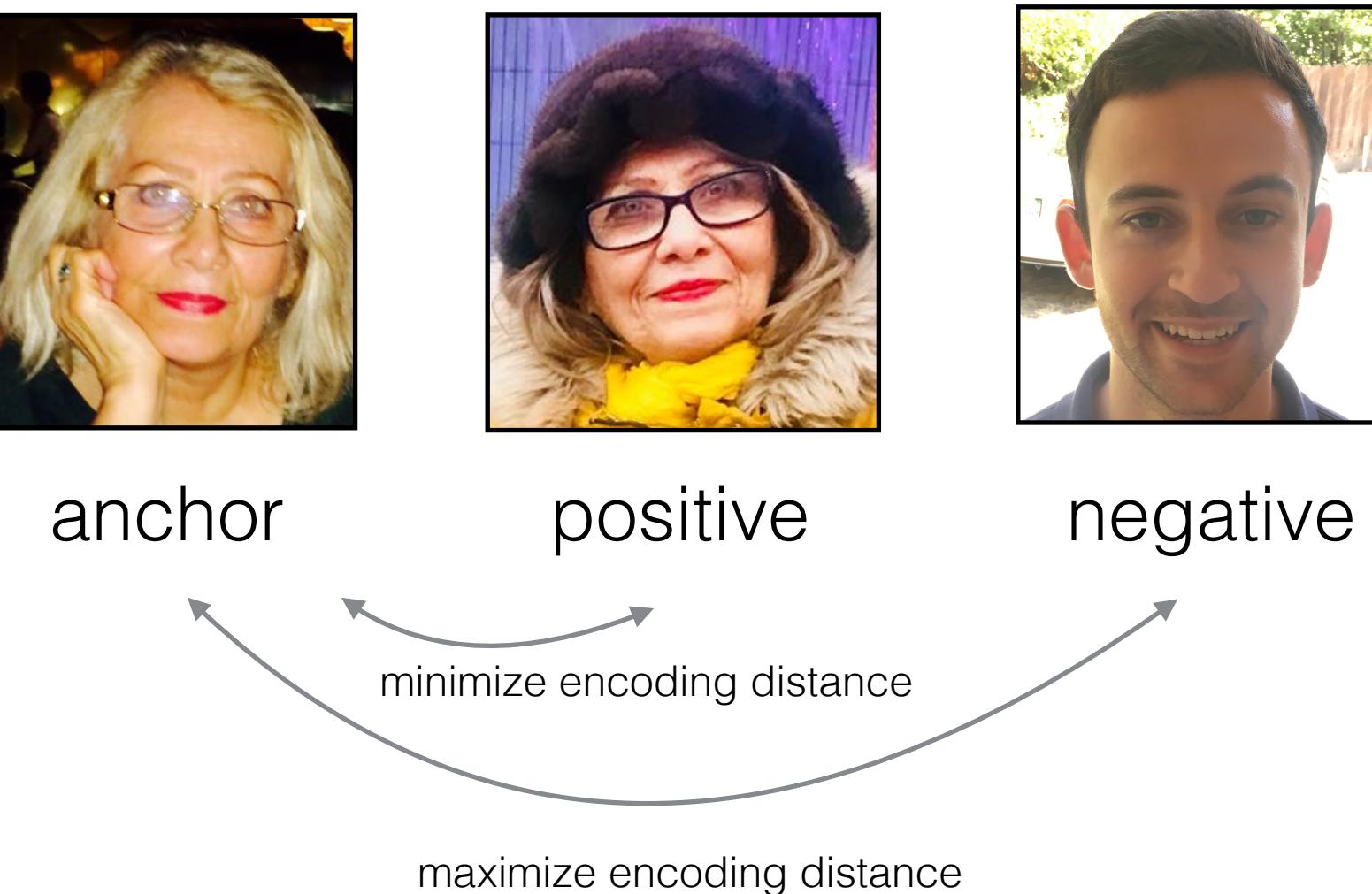
similar encoding



different encoding



So let's generate triplets:



## Case study 2: Face Verification

What we really want:



similar encoding

different encoding

So let's generate triplets:



anchor

positive

negative

minimize encoding distance

maximize encoding distance

## Which loss should you minimize?

$$L = \|Enc(A) - Enc(P)\|_2^2$$

$$- \|Enc(A) - Enc(N)\|_2^2$$

$$L = \|Enc(A) - Enc(N)\|_2^2$$

$$- \|Enc(A) - Enc(P)\|_2^2$$

$$L = \|Enc(P) - Enc(N)\|_2^2$$

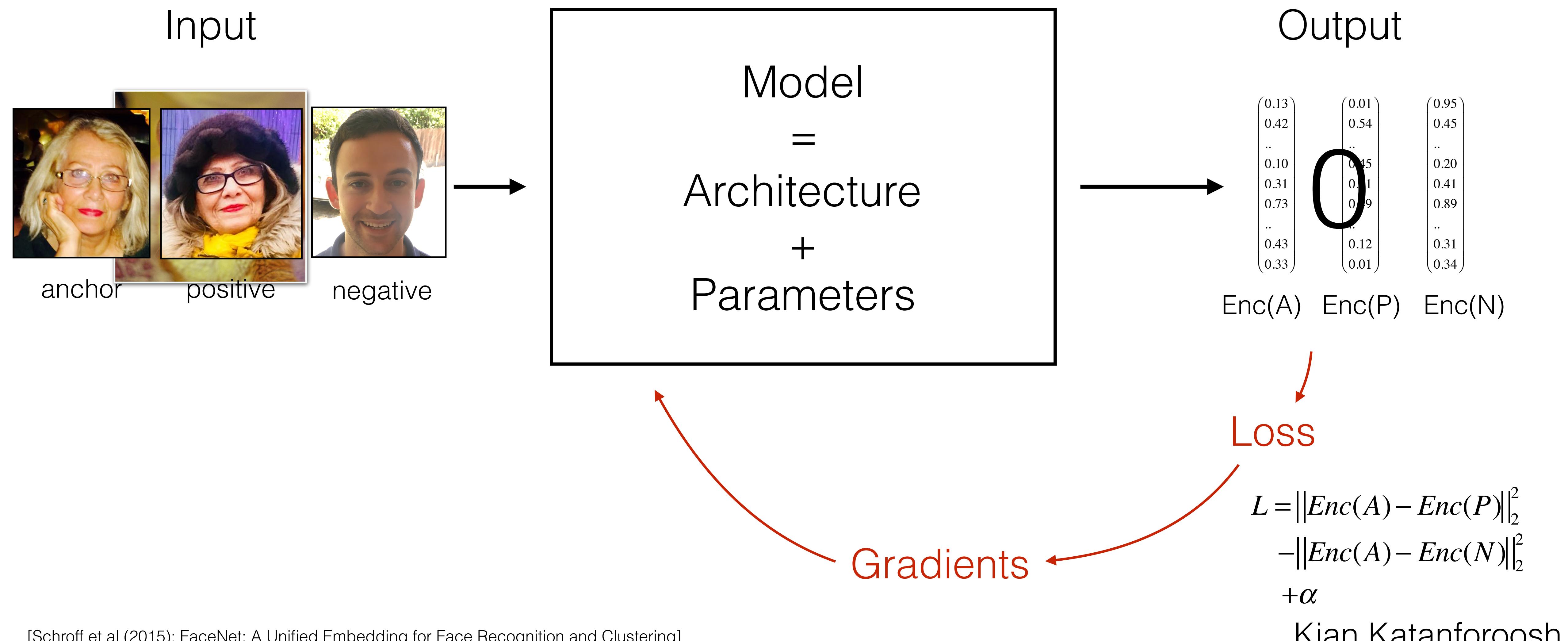
$$- \|Enc(P) - Enc(A)\|_2^2$$

A

B

C

## Case study 2: Face Verification



## Case study 2b: Face Identification and Face Clustering

**Goal:** A school wants to use Face Identification for recognize students in facilities (dinning hall, gym, pool ...)

K-Nearest Neighbors

**Goal:** You want to use Face Clustering to group pictures of the same people on your smartphone

K-Means Algorithm

Maybe we need to detect the faces first?

## Summary of learnings: Face Verification

- In face verification, we have used an **encoder network** to learn a lower dimensional representation (called “**encoding**”) for a set of data by training the network to **focus on non-noisy signals**.
- **Triplet loss** is a loss function where an (**anchor**) input is compared to a **positive** input and a **negative** input. The distance from the anchor input to the positive input is minimized, whereas the distance from the anchor input to the negative input is maximized.
- You learned the difference between **face verification, face identification and face clustering**.

# Case study 3: Art Generation

**Goal:** Given a picture, make it look beautiful

## 1. Data?

Let's say we have  
any data

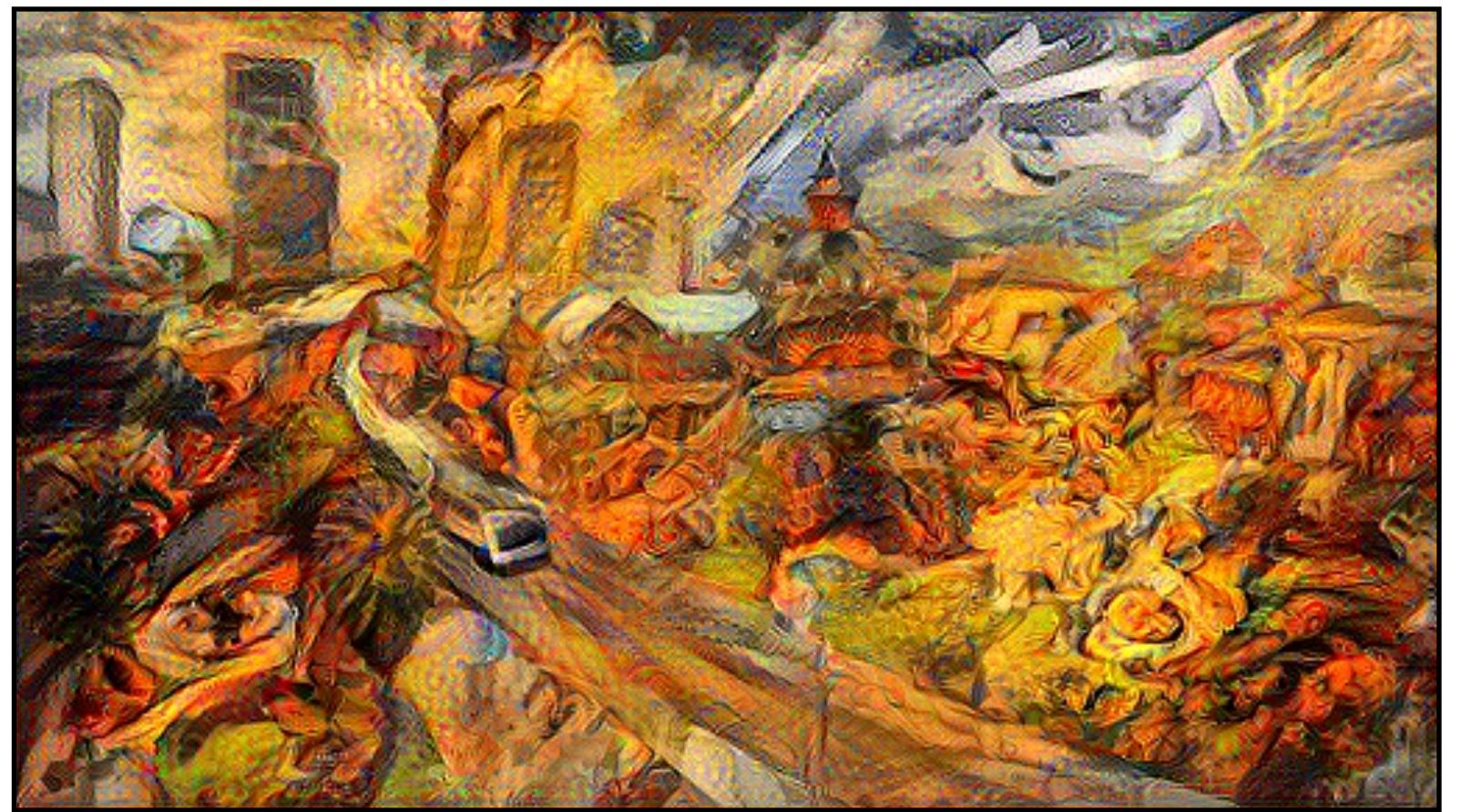


## 2. Input?



content  
image

## 3. Output?



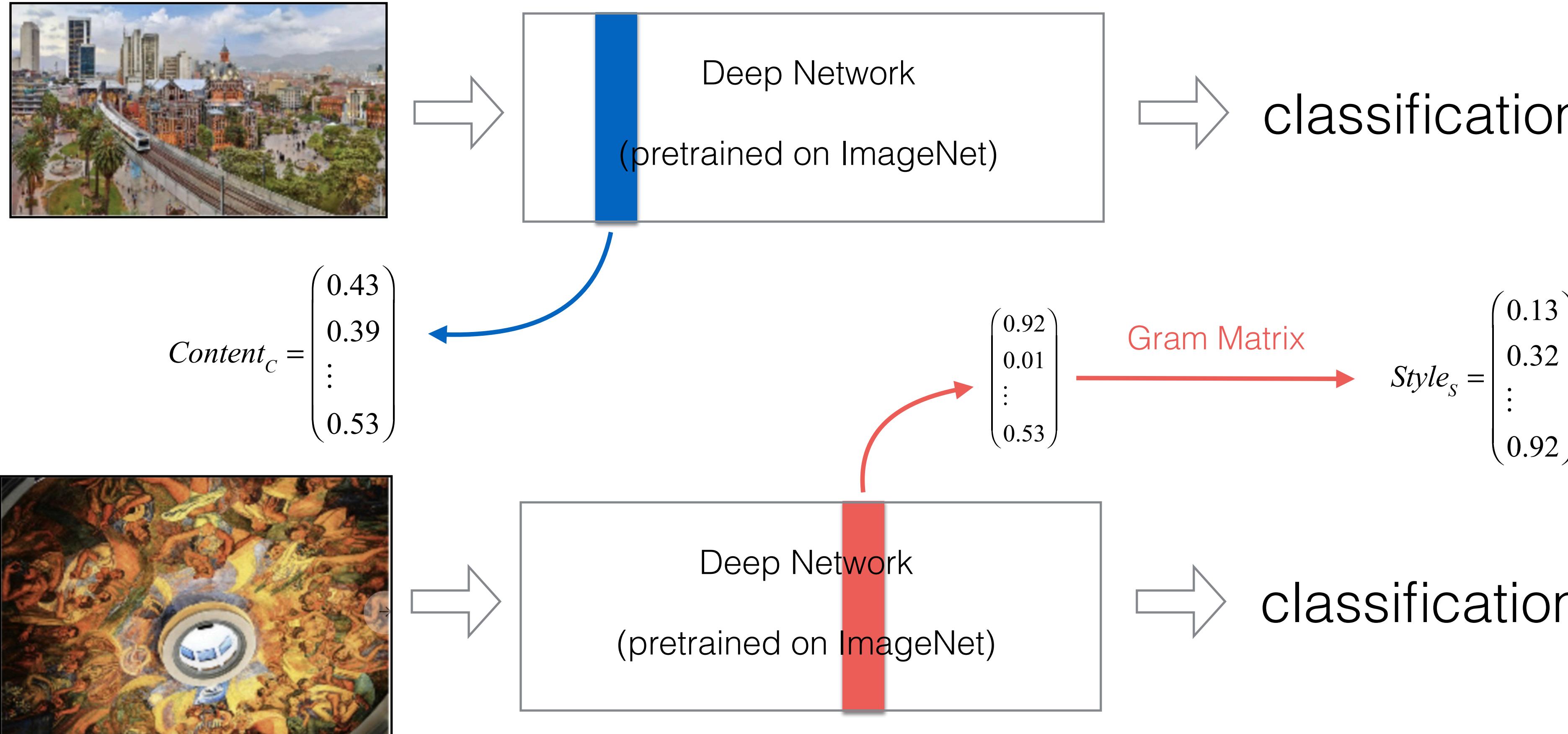
style  
image

generated  
image

# Case study 3: Art Generation

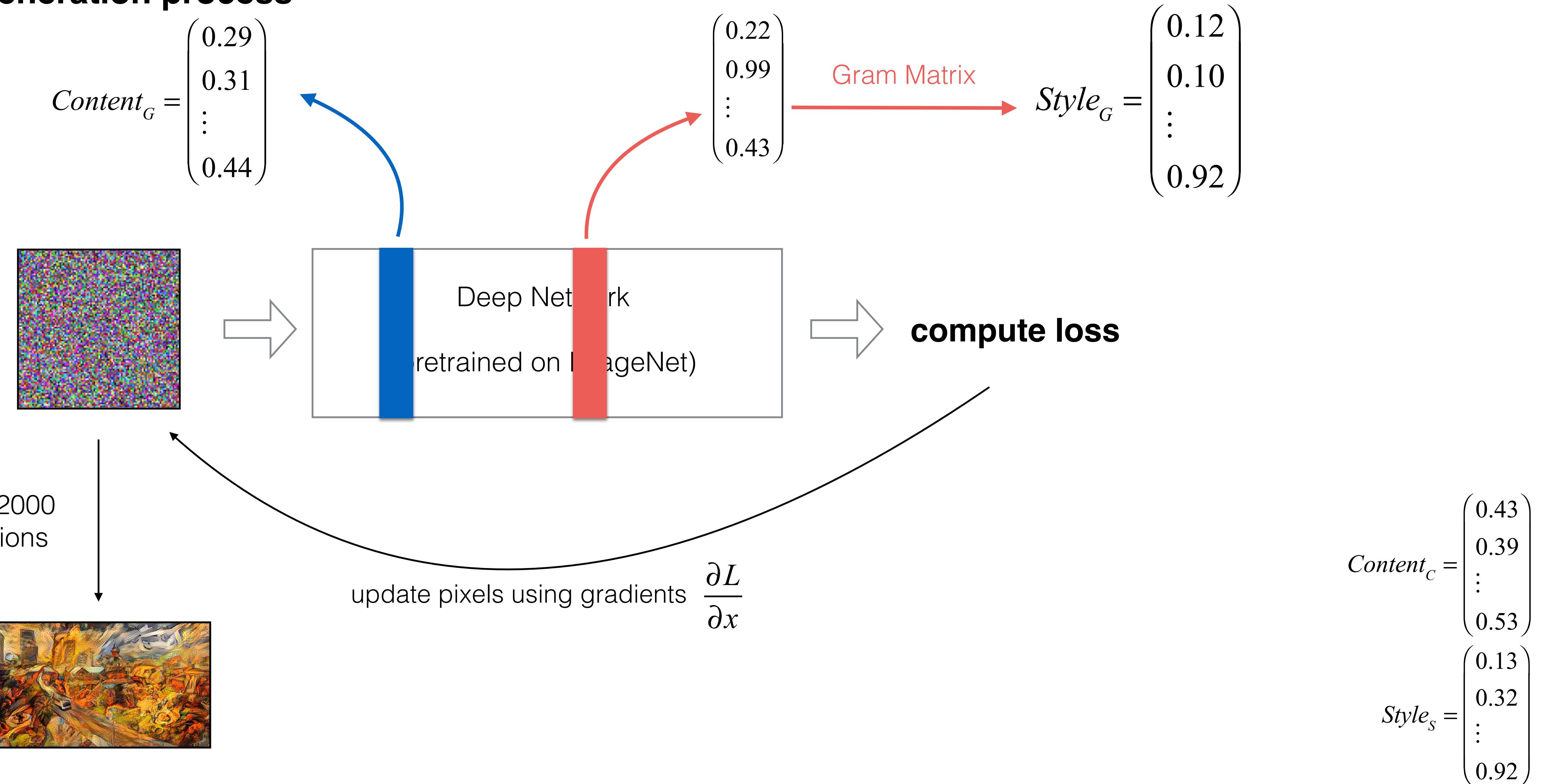
## 4. Architecture?

We use a **pre-trained model** because it **extracts important information** from images.



# Case study 3: Art Generation

## Image generation process



## Case study 3: Art Generation

Which loss should we minimize?

$$L = \|Content_C - Content_G\|_2^2$$

$$- \|Style_S - Style_G\|_2^2$$

$$L = \|Style_S - Style_G\|_2^2$$

$$+ \|Content_C - Content_G\|_2^2$$

$$L = \|Style_S - Style_G\|_2^2$$

$$- \|Content_C - Content_G\|_2^2$$

---

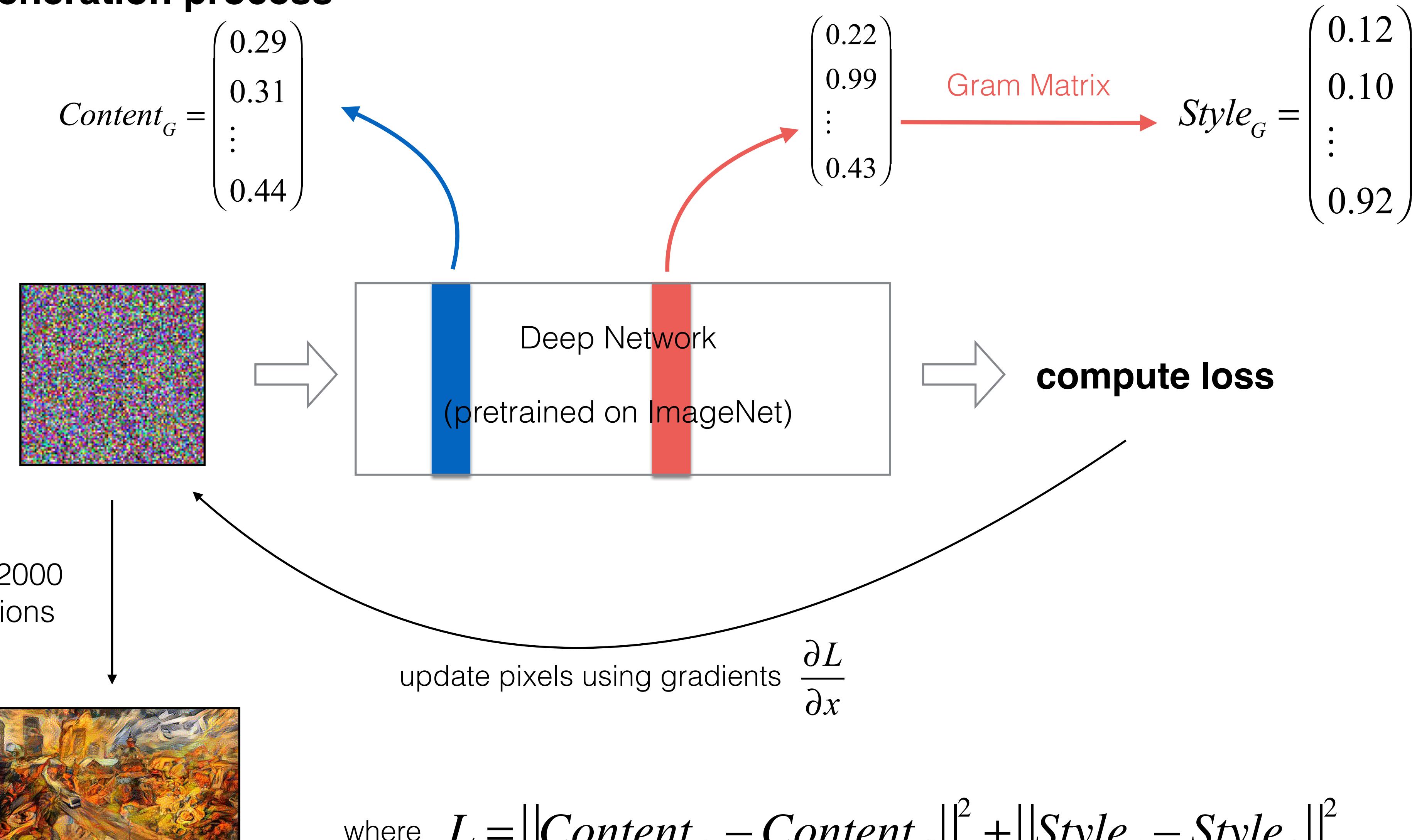
A

B

C

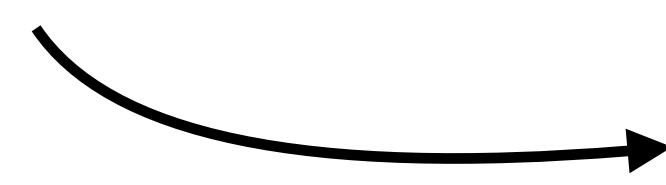
# Case study 3: Art Generation

## Image generation process





Content image



In the style of Hilma af Klint



In the style of Jamini Roy



In the style of Claude Monet



In the style of Yayoi Kusama



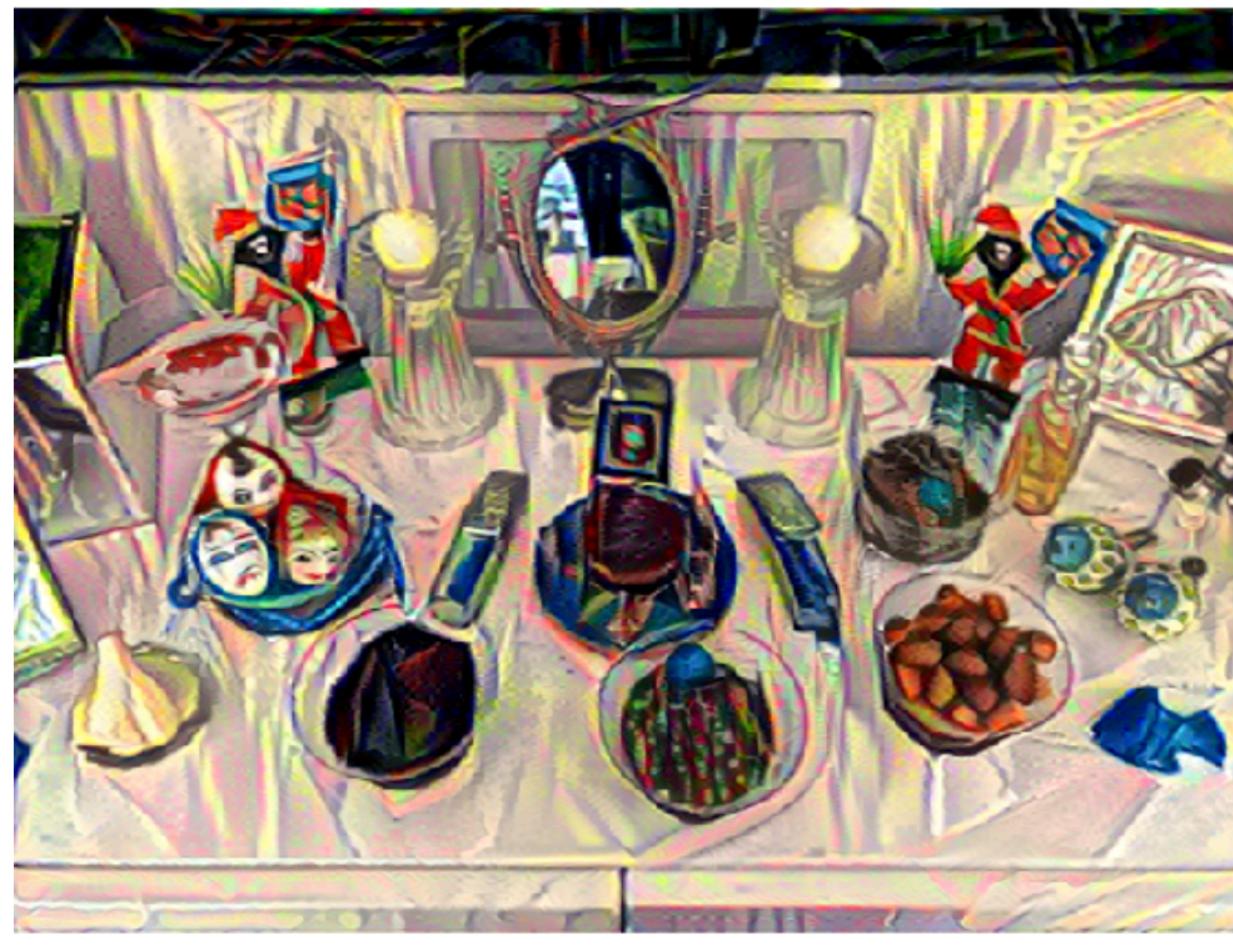
In the style of Eiichiro Oda



In the style of Salvador Dali



In the style of Piet Mondrian



In the style of Pablo Picasso

## Summary of learnings: Art Generation

- In the neural style transfer algorithm proposed by Gatys et al., you **optimize image pixels rather than model parameters**. Model parameters are pretrained and non-trainable.
- You leverage the “knowledge” of a pretrained model to extract the **content** of a content image and the **style** of a style image.
- The loss proposed by Gatys et al. aims to minimize the distances between the **content** of the generated and content images, and the **style** of the generated and style images.

## Case study 4: Trigger word detection

**Goal:** Given a 10sec audio speech, detect the word “activate”.

**1. Data?**

A bunch of 10s audio clips

Distribution?

**2. Input?**

$x = \text{A 10sec audio clip}$

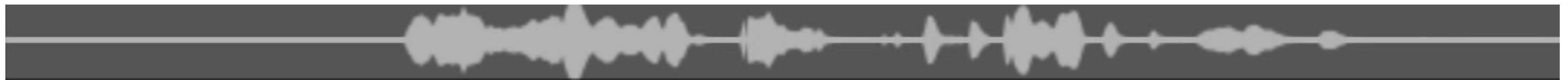


Resolution? (sample rate)

**3. Output?**

$y = 0 \text{ or } y = 1$

# Let's have an experiment!



$$y = 1$$



$$y = 0$$



$$y = 1$$



## Case study 4: Trigger word detection

**Goal:** Given a 10sec audio speech, detect the word “activate”.

**1. Data?**

A bunch of 10s audio clips

Distribution?

**2. Input?**

$x = \text{A 10sec audio clip}$



Resolution? (sample rate)

**3. Output?**

$y = 0 \text{ or } y = 1$

$y = 00..0000\mathbf{1}00000..000$

$y = 00..0000\mathbf{1}..1000..000$

Last Activation?  
sigmoid  
(sequential)

**4. Architecture ?**

Sounds like it should be a RNN

**5. Loss?**

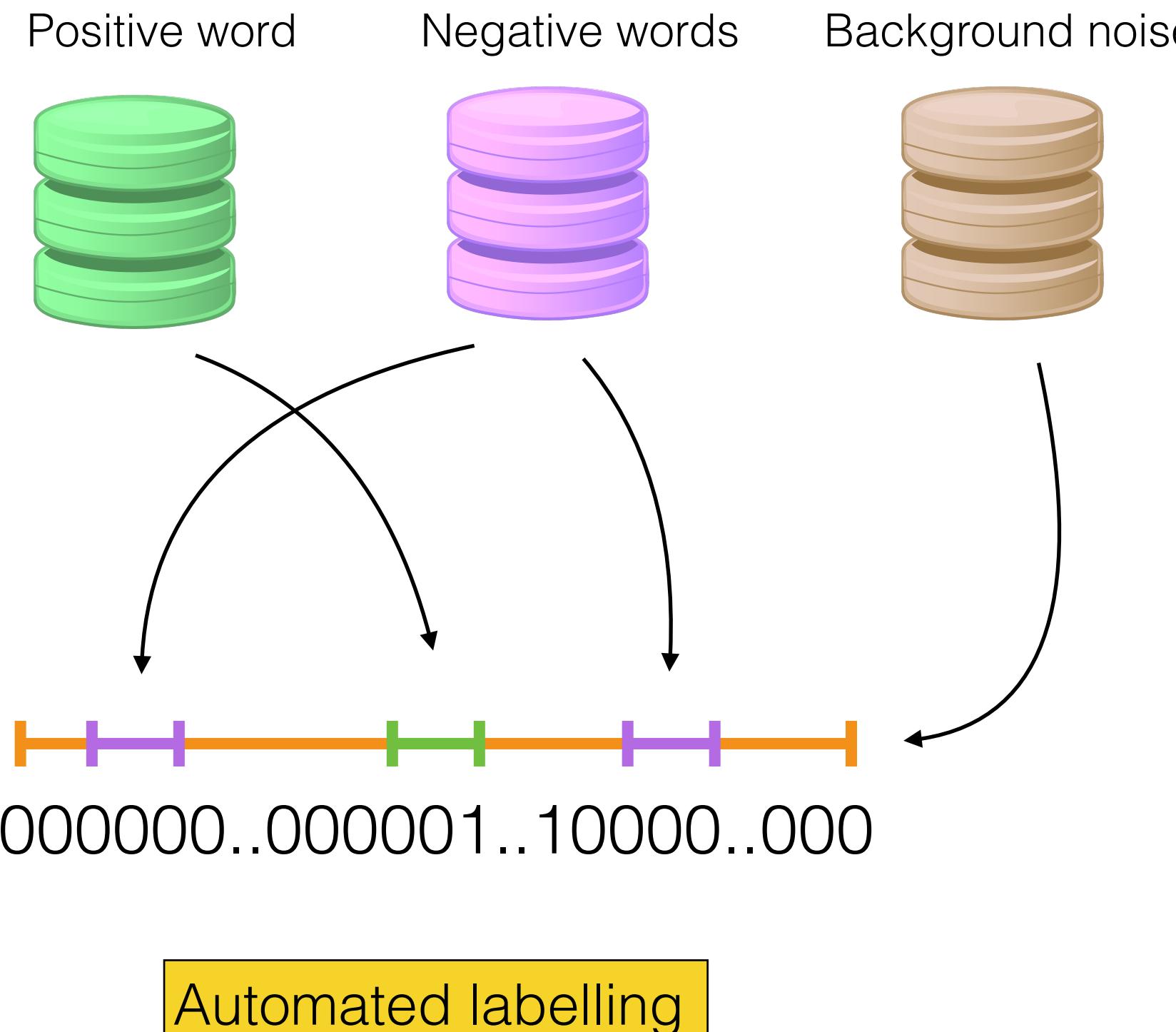
$$L = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

(sequential)

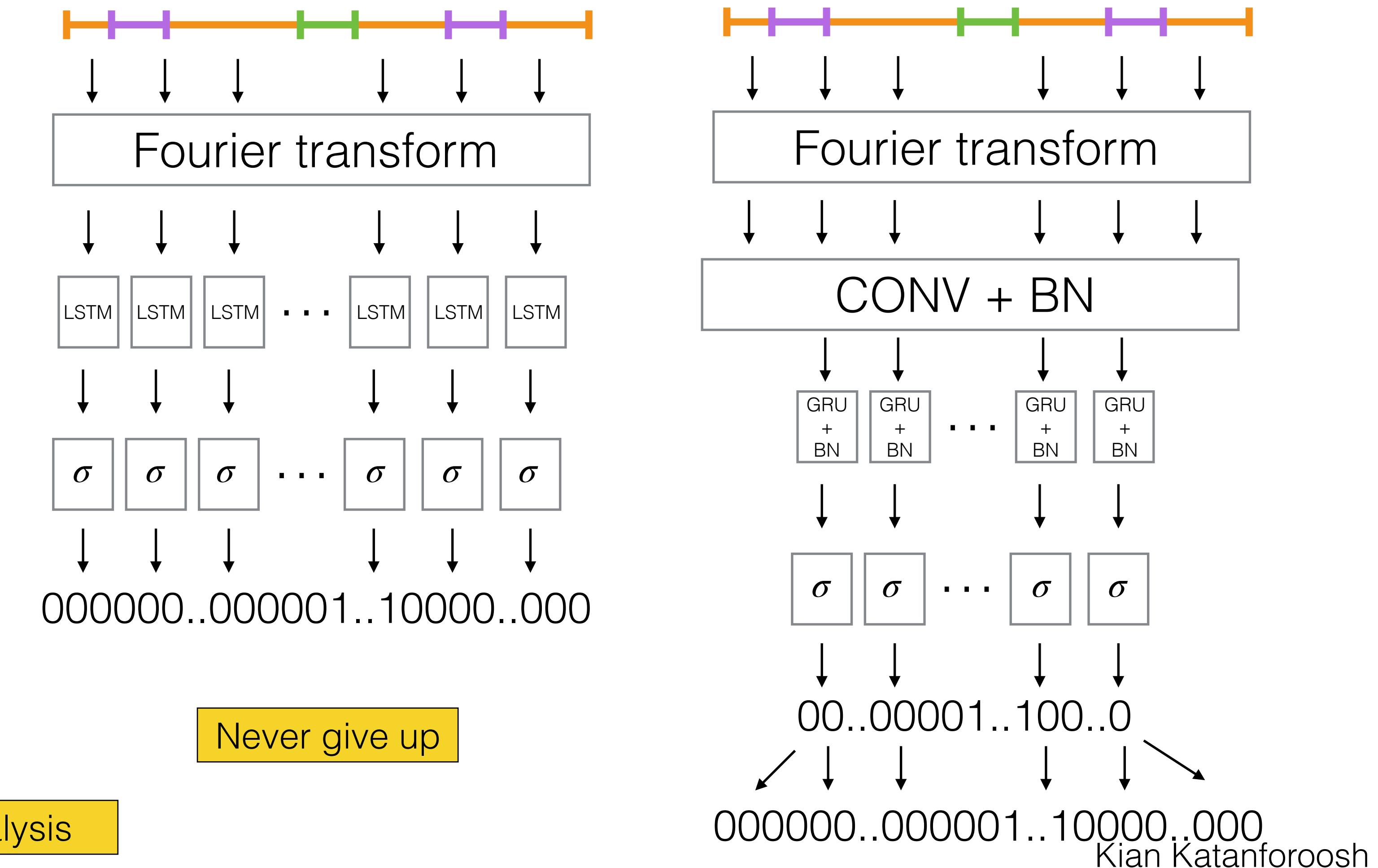
# Case study 4: Trigger word detection

What is critical to the success of this project?

## 1. Strategic data collection/ labelling process



## 2. Architecture search & Hyperparameter tuning



## Summary of learnings: Trigger word detection

- Your **data collection strategy** is critical to the success of your project. (If applicable) Don't hesitate to get out of the building.
- You can gain insights on your labelling strategy by using a **human experiment**.
- **Refer to expert advice** to earn time and be guided towards a good direction.

## Featured among “the most beautiful loss functions of 2015”

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left( C_i - \hat{C}_i \right)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left( C_i - \hat{C}_i \right)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

## Duties for next week

For next week:

### C1M3

- Quiz: Shallow Neural Networks
- Programming Assignment: Planar data classification with one-hidden layer

### C1M4

- Quiz: Deep Neural Networks
- Programming Assignment: Building a deep neural network - Step by Step
- Programming Assignment: Deep Neural Network Application

### Others:

- TA project mentorship (mandatory): Meet with a TA to discuss your proposal.
- Friday TA section
- Fill-in AWS Form to get GPU credits for your projects