CS230: Lecture 2

Practical Approaches to Deep Learning Projects

Kian Katanforoosh

Recap

Learning Process

Input

Model
=
Architecture
+
Parameters

Output

0

Loss

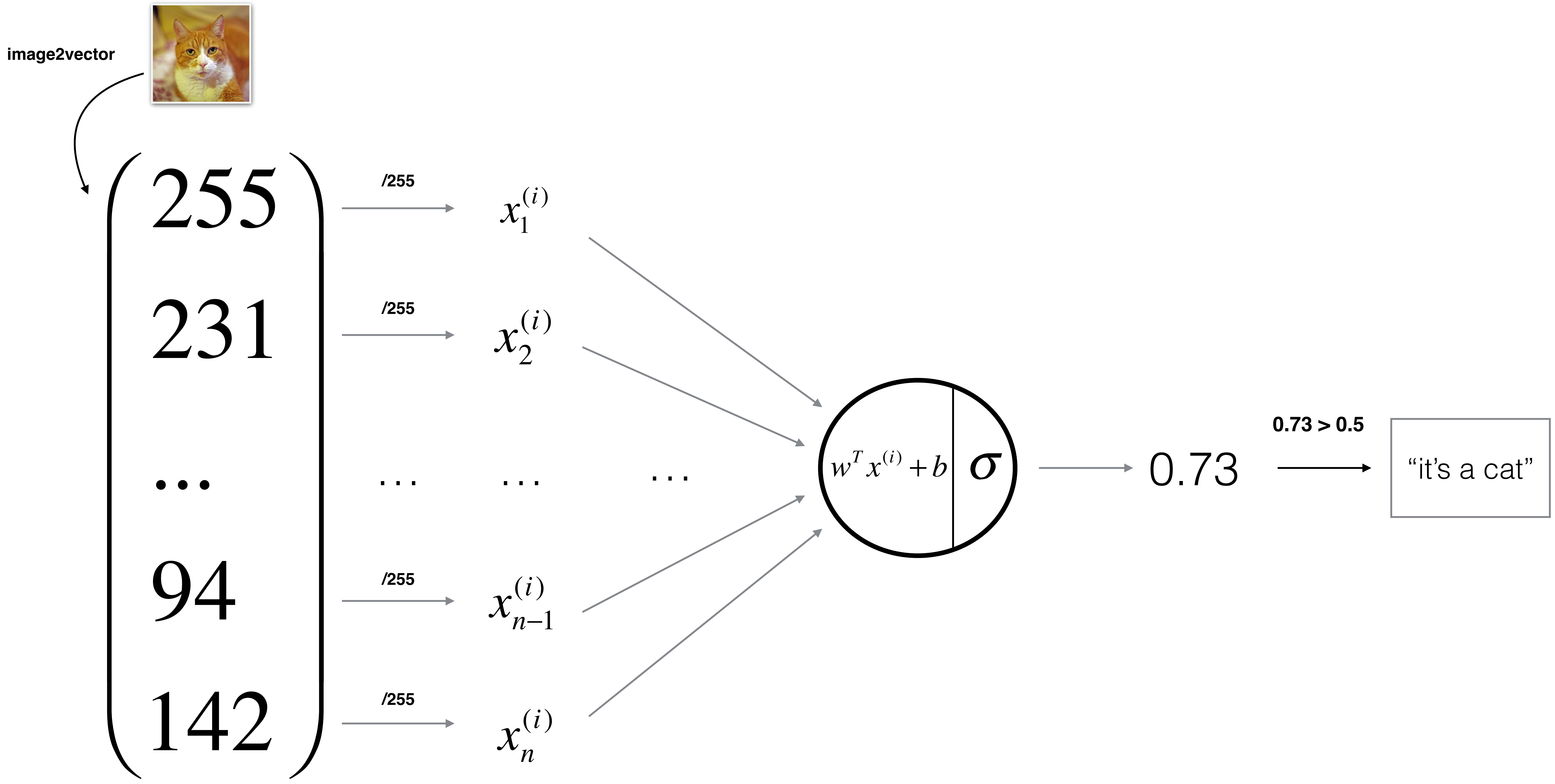
Gradients

Things that can change

- Activation function
- Optimizer
- Hyperparameters
- ...

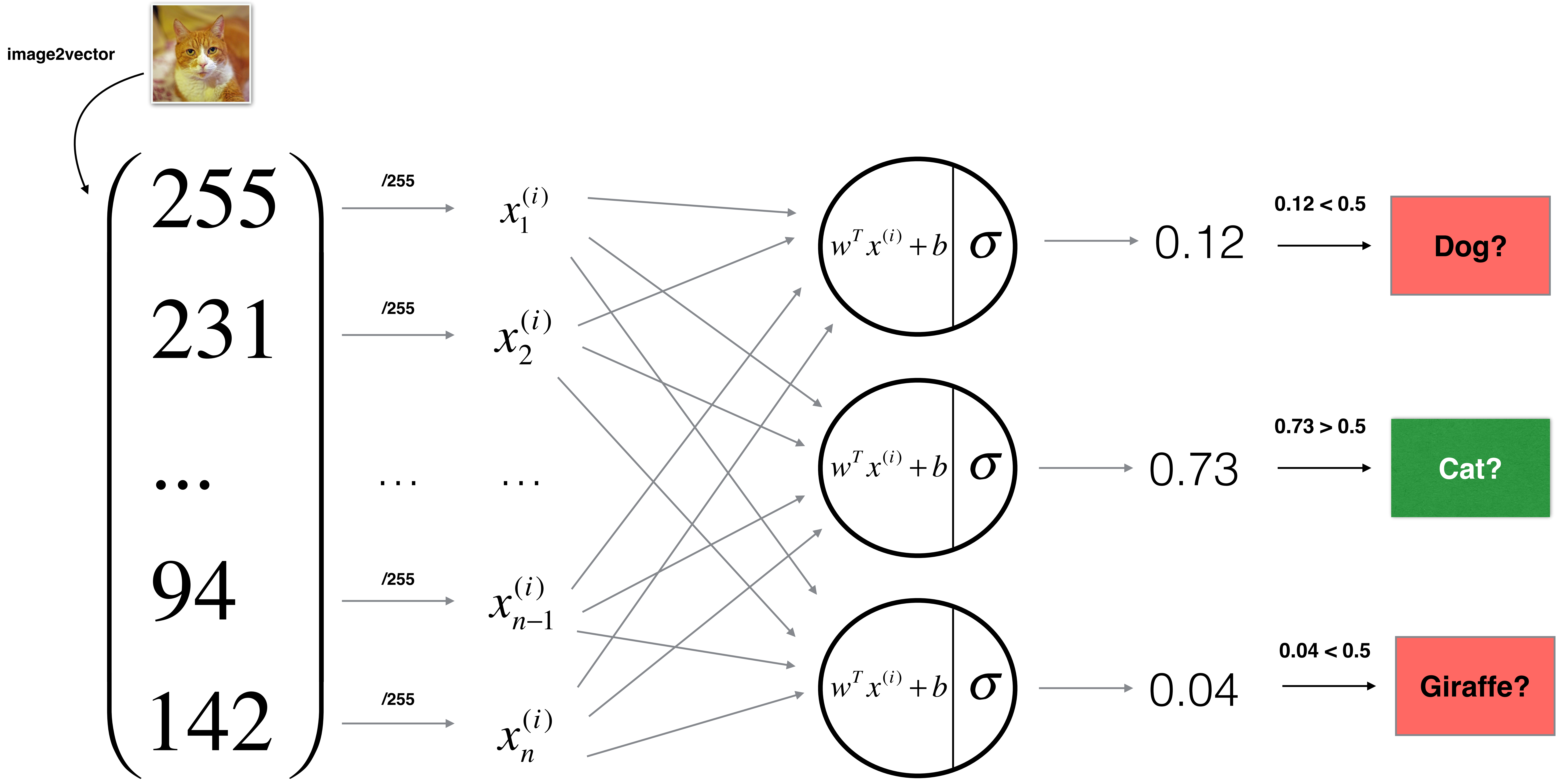
Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Logistic Regression as a Neural Network



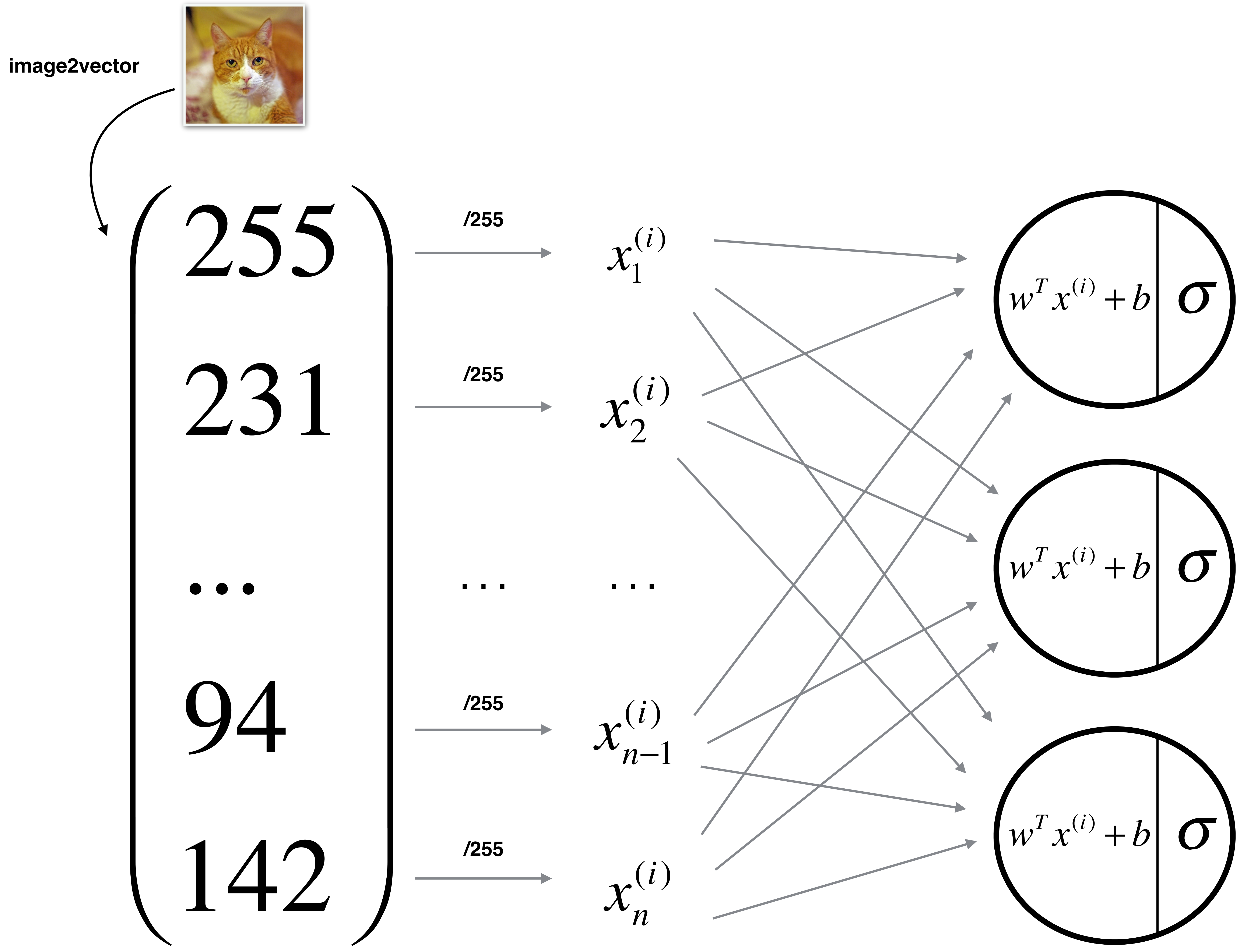
Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Multi-class



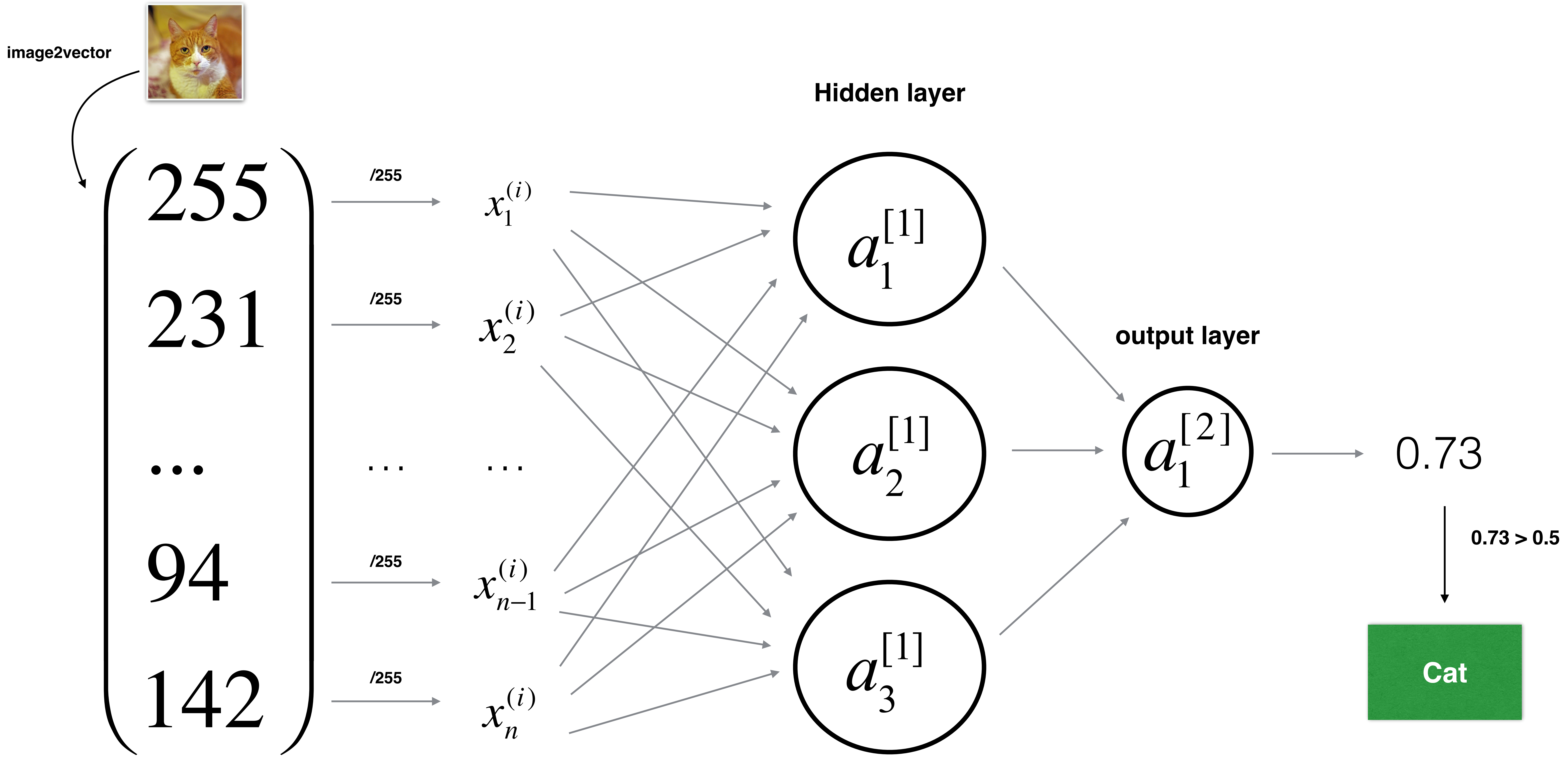
Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Neural Network (Multi-class)



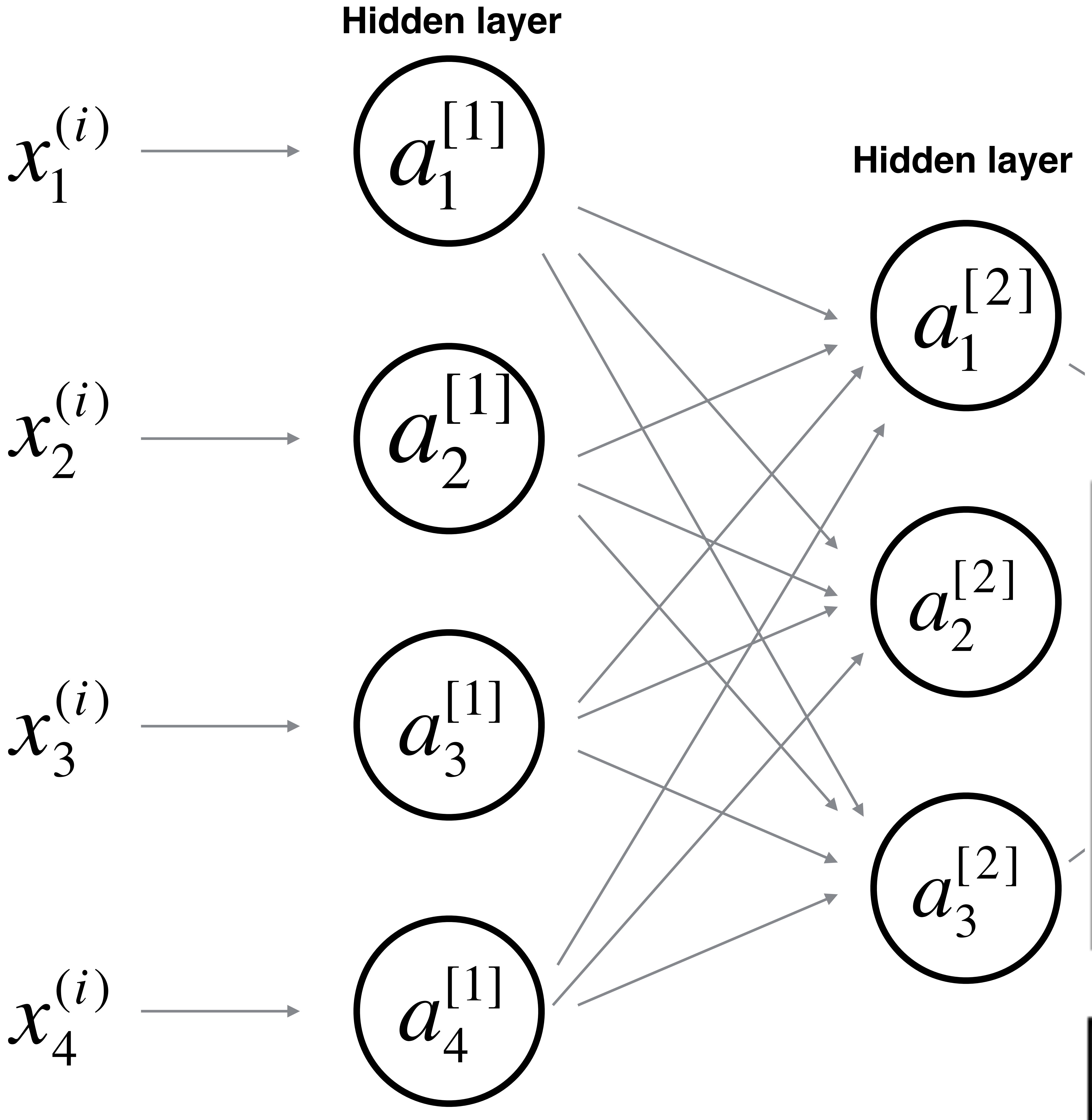
Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

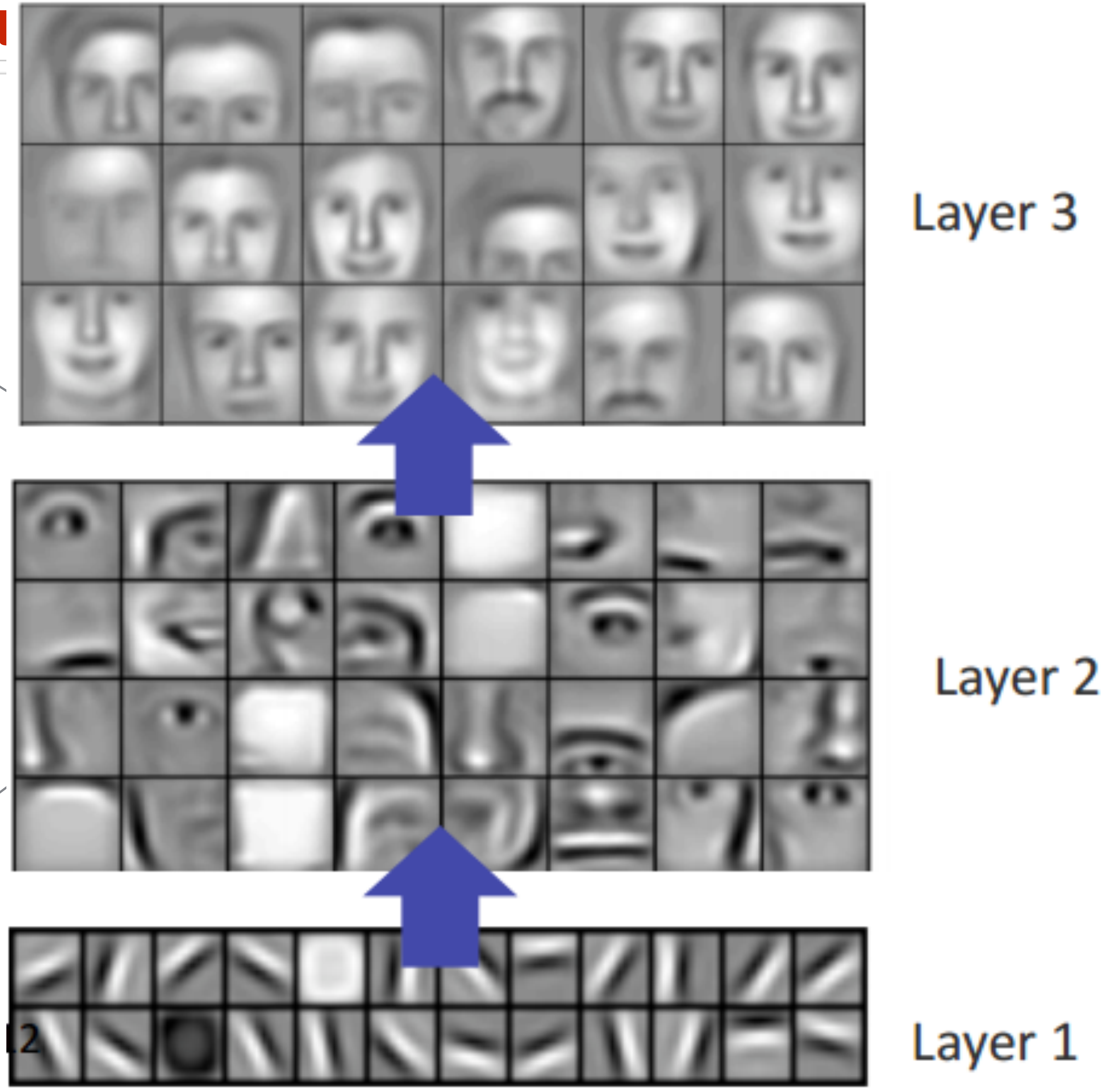
Neural Network (1 hidden layer)



Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

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**.

Let’s build intuition on concrete applications

Today's outline

We will learn tips and tricks to:

- Analyze a problem from a deep learning approach
- Choose an **architecture**
- Choose a **loss** and a **training strategy**

I. Day'n'Night classification

II. Face verification and recognition

III. Neural style transfer (Art generation)

IV. Trigger-word detection

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 ?** Shallow network should do the job pretty well

5. **Loss?** $$L = -[y\log(\hat{y}) + (1-y)\log(1-\hat{y})]$$

Easy warm up

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

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.

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)

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

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

Deep Network

128-d

$$\begin{pmatrix} 0.931 \\ 0.433 \\ 0.331 \\ \vdots \\ 0.942 \\ 0.158 \\ 0.039 \end{pmatrix}$$

Deep Network

$$\begin{pmatrix} 0.922 \\ 0.343 \\ 0.312 \\ \vdots \\ 0.892 \\ 0.142 \\ 0.024 \end{pmatrix}$$

distance $\longrightarrow$ 0.4 $\xrightarrow{0.4\ <\ \text{threshold}}$ y=1

We gather all student faces encoding in a database. Given a new picture, we compute its distance with the encoding of card holder

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Face Recognition

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:

anchor positive negative

minimize encoding distance

maximize encoding distance

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Face Recognition

What we really want:

similar encoding different encoding

So let's generate triplets:

anchor positive negative

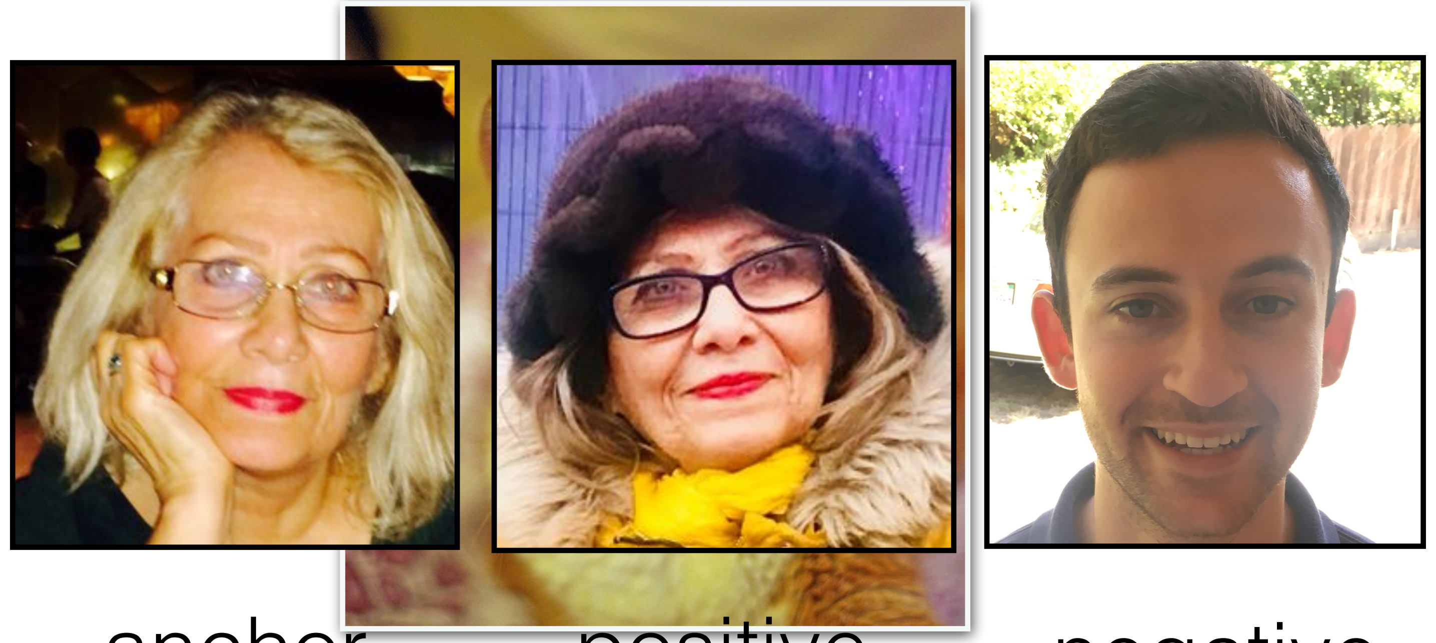
minimize encoding distance

maximize encoding distance

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Recap: Learning Process

Input



anchor positive negative

Model
=
Architecture
+
Parameters

Output

$$\begin{pmatrix}0.13\\0.42\\..\\0.10\\0.31\\0.73\\..\\0.43\\0.33\end{pmatrix} \quad \begin{pmatrix}0.01\\0.54\\..\\0.45\\0.?1\\0.?9\\..\\0.12\\0.01\end{pmatrix} \quad \begin{pmatrix}0.95\\0.45\\..\\0.20\\0.41\\0.89\\..\\0.31\\0.34\end{pmatrix}$$

0

Enc(A) Enc(P) Enc(N)

Loss

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

Gradients

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Face Recognition

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 Recognition

- 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 learnt the difference between **face verification, face identification and face clustering**.

Art generation (Neural Style Transfer)

Goal: Given a picture, make it look beautiful

1. Data?

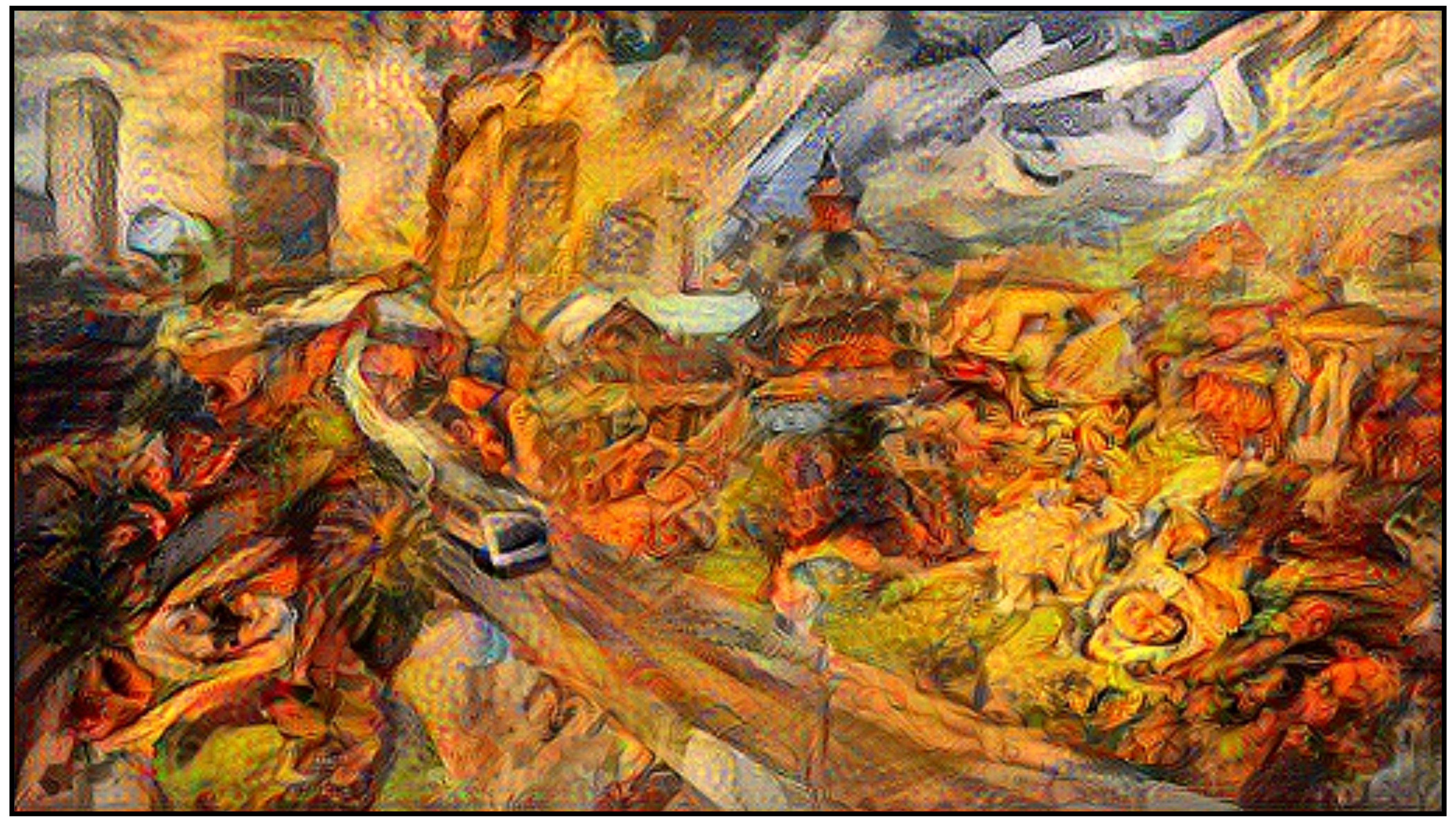
Let's say we have any data

2. Input?

content image

style image

3. Output?



generated image

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge: A Neural Algorithm of Artistic Style, 2015

Kian Katanforoosh

Art generation (Neural Style Transfer)

4. Architecture?

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

Deep Network

(pretrained on ImageNet)

classification

$$Content_C = \begin{pmatrix} 0.43 \\ 0.39 \\ \vdots \\ 0.53 \end{pmatrix}$$

$$\begin{pmatrix} 0.92 \\ 0.01 \\ \vdots \\ 0.53 \end{pmatrix} \xrightarrow{\text{Gram Matrix}} Style_S = \begin{pmatrix} 0.13 \\ 0.32 \\ \vdots \\ 0.92 \end{pmatrix}$$

Deep Network

(pretrained on ImageNet)

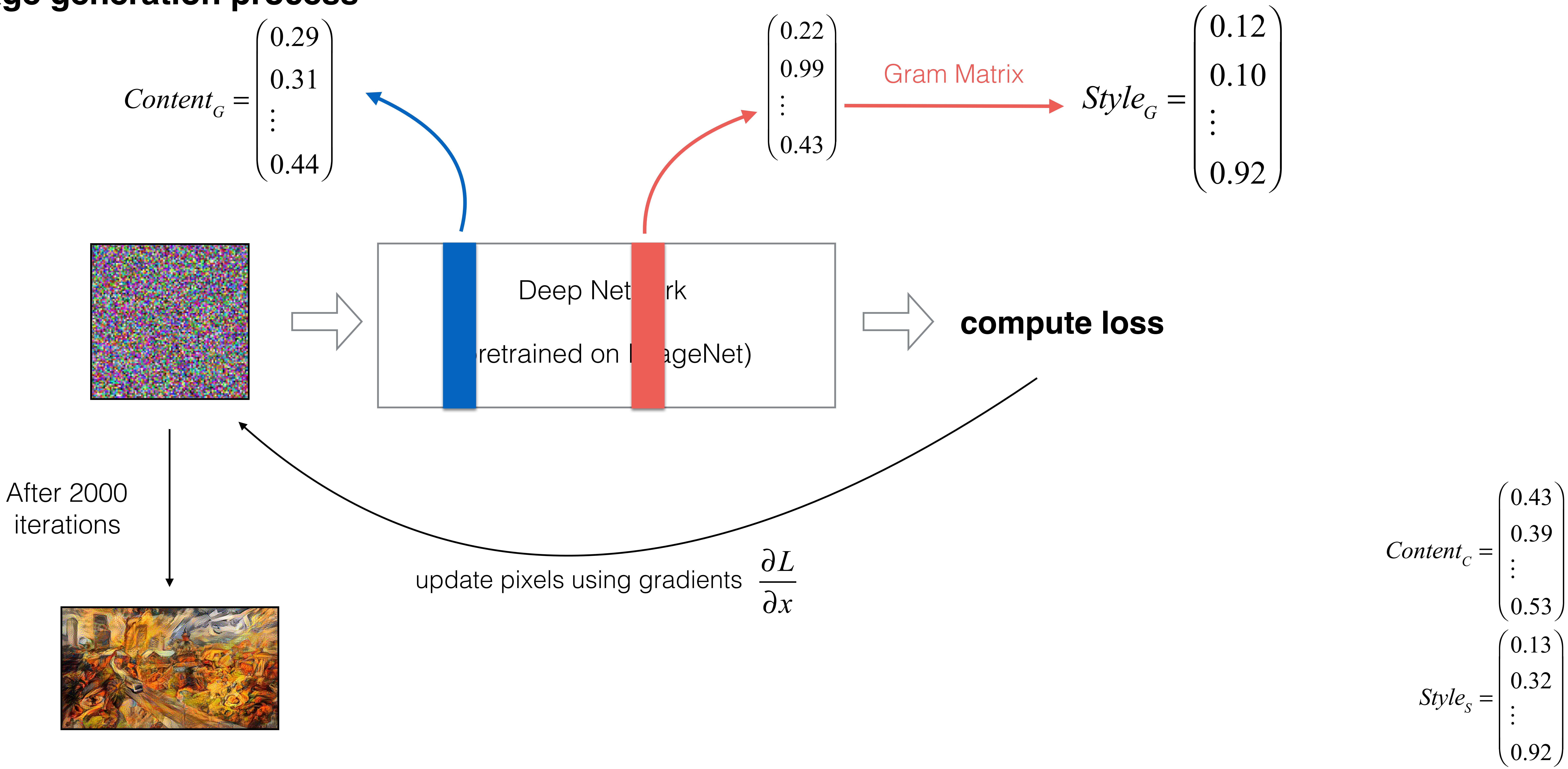
classification

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge: A Neural Algorithm of Artistic Style, 2015

Kian Katanforoosh

Art generation (Neural Style Transfer)

Image generation process



Art generation (Neural Style Transfer)

Which loss should we minimize?

A

$$L = \left\|Content_C - Content_G\right\|_2^2 - \left\|Style_S - Style_G\right\|_2^2$$

B

$$L = \left\|Style_S - Style_G\right\|_2^2 + \left\|Content_C - Content_G\right\|_2^2$$

C

$$L = \left\|Style_S - Style_G\right\|_2^2 - \left\|Content_C - Content_G\right\|_2^2$$

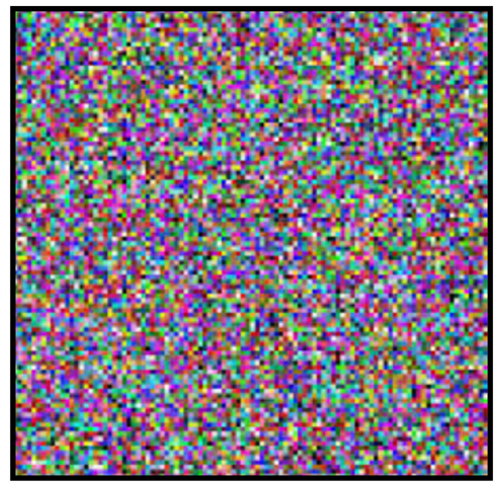
Kian Katanforoosh

Art generation (Neural Style Transfer)

Image generation process

$$Content_G = \begin{pmatrix} 0.29 \\ 0.31 \\ \vdots \\ 0.44 \end{pmatrix}$$

$$\begin{pmatrix} 0.22 \\ 0.99 \\ \vdots \\ 0.43 \end{pmatrix} \xrightarrow{\text{Gram Matrix}} Style_G = \begin{pmatrix} 0.12 \\ 0.10 \\ \vdots \\ 0.92 \end{pmatrix}$$



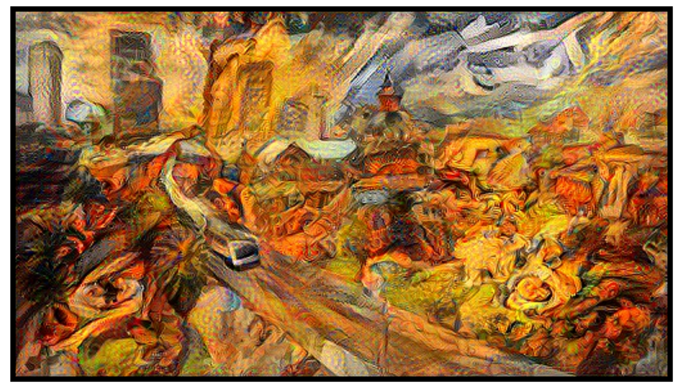
Deep Network

(pretrained on ImageNet)

compute loss

After 2000 iterations

update pixels using gradients $\frac{\partial L}{\partial x}$



where $L = ||Content_C - Content_G||_2^2 + ||Style_S - Style_G||_2^2$

$$Content_C = \begin{pmatrix} 0.43 \\ 0.39 \\ \vdots \\ 0.53 \end{pmatrix}$$

$$Style_S = \begin{pmatrix} 0.13 \\ 0.32 \\ \vdots \\ 0.92 \end{pmatrix}$$

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge: A Neural Algorithm of Artistic Style, 2015

Kian Katanforoosh

Content image

In the style of Hilma af Klint

In the style of Jamini Roy

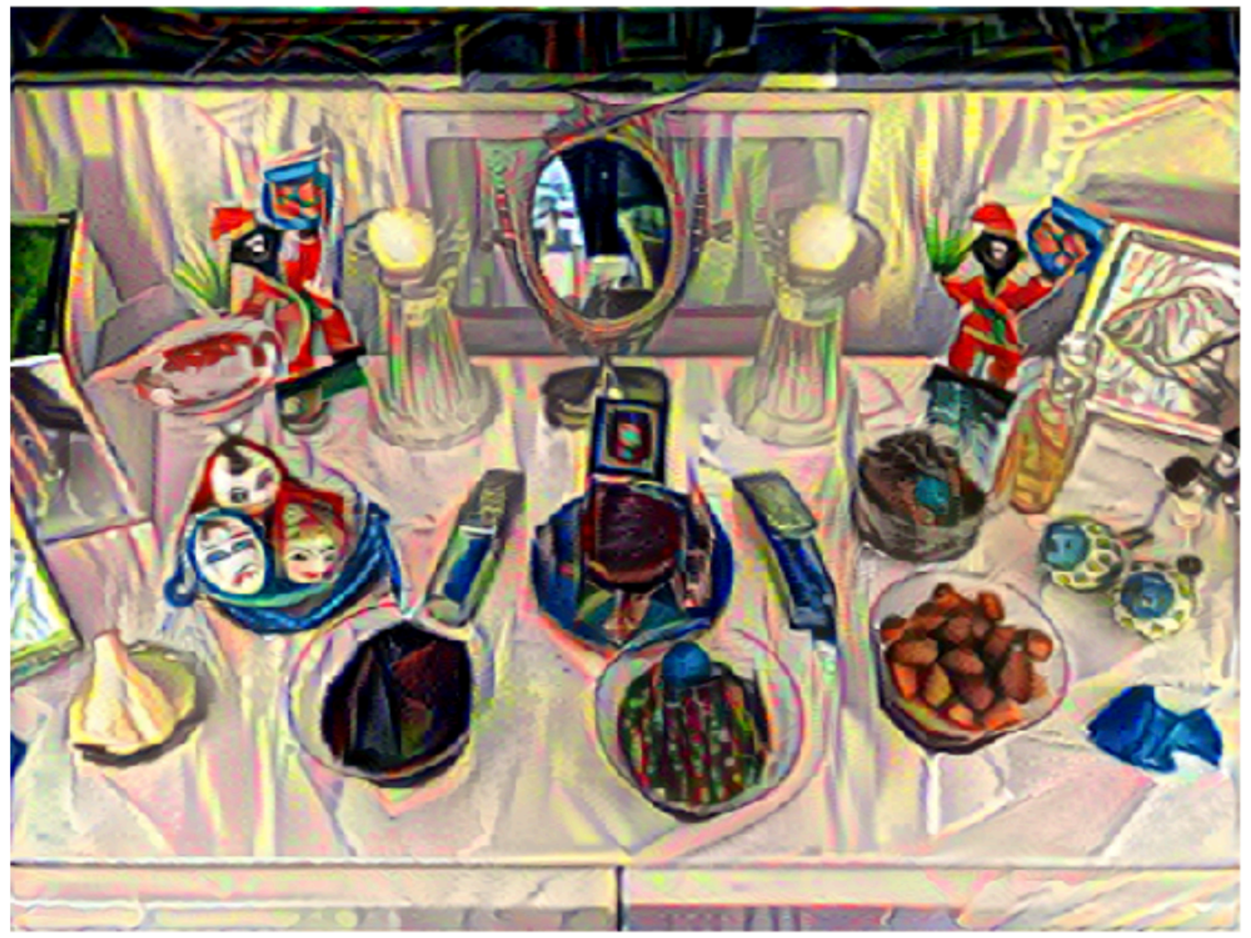
In the style of Claude Monet

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: Neural Style Transfer

- 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.

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 = A 10sec audio clip — Resolution? (sample rate)

3. **Output?** y = 0 or y = 1

Let’s have an experiment!

y = 1

y = 0

y = 1

y = 000000000000000000000000000000000000000**1**0000000000

y = 00000000000000000000000000000000000000000000000000

y = 00000000000**1**00000000000000000000000000000000000000

Kian Katanforoosh

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 = A 10sec audio clip — Resolution? (sample rate)

3. **Output?**
   y = 0 or y = 1
   y = 00..0000**1**00000..000
   y = 00..0000**1..1**000..000
   Last Activation? sigmoid (sequential)

4. **Architecture ?** Sounds like it should be a RNN

5. **Loss?**

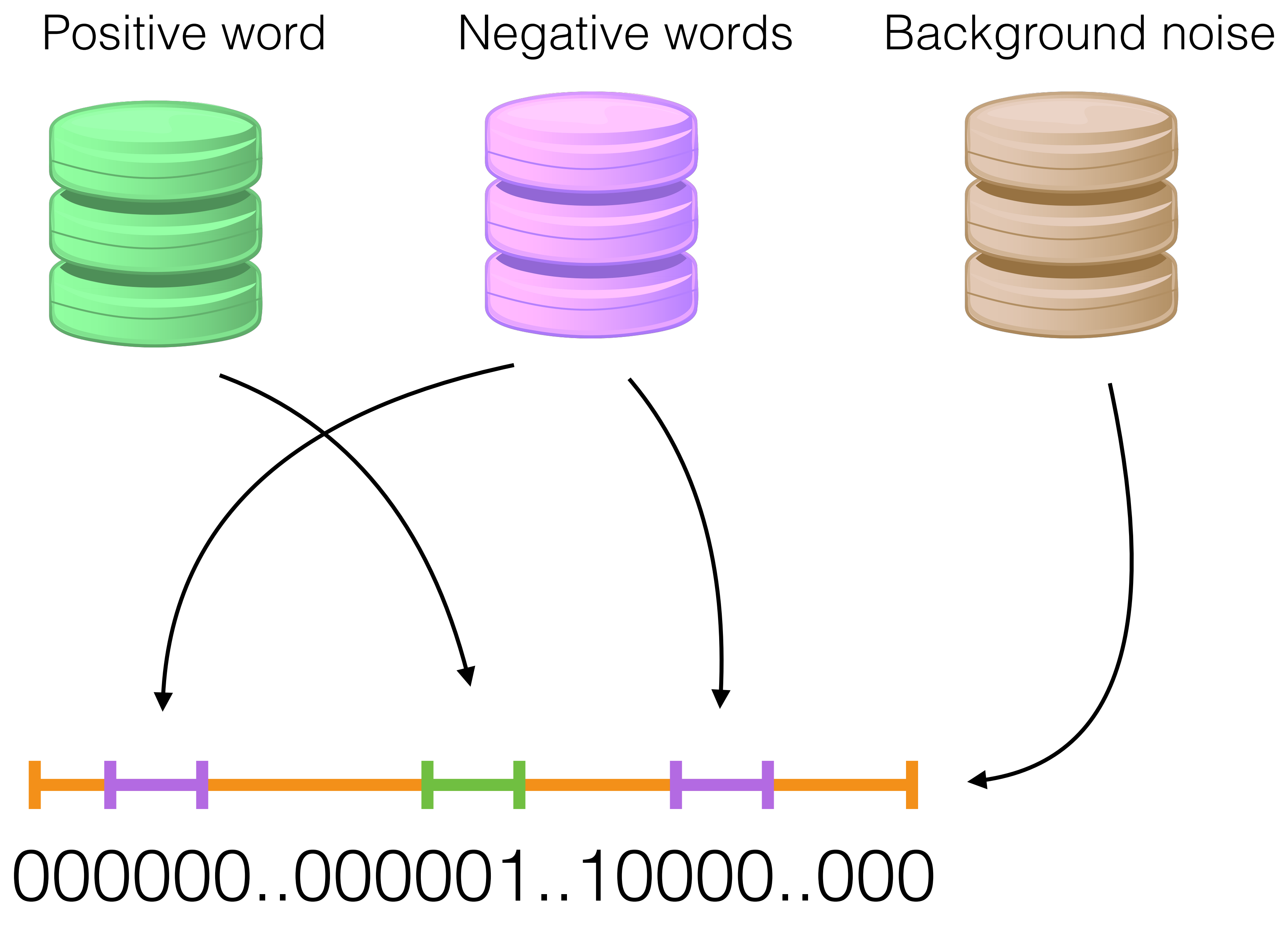
$$L = -(y \log(\hat{y}) + (1-y)\log(1-\hat{y}))$$

(sequential)

Trigger word detection

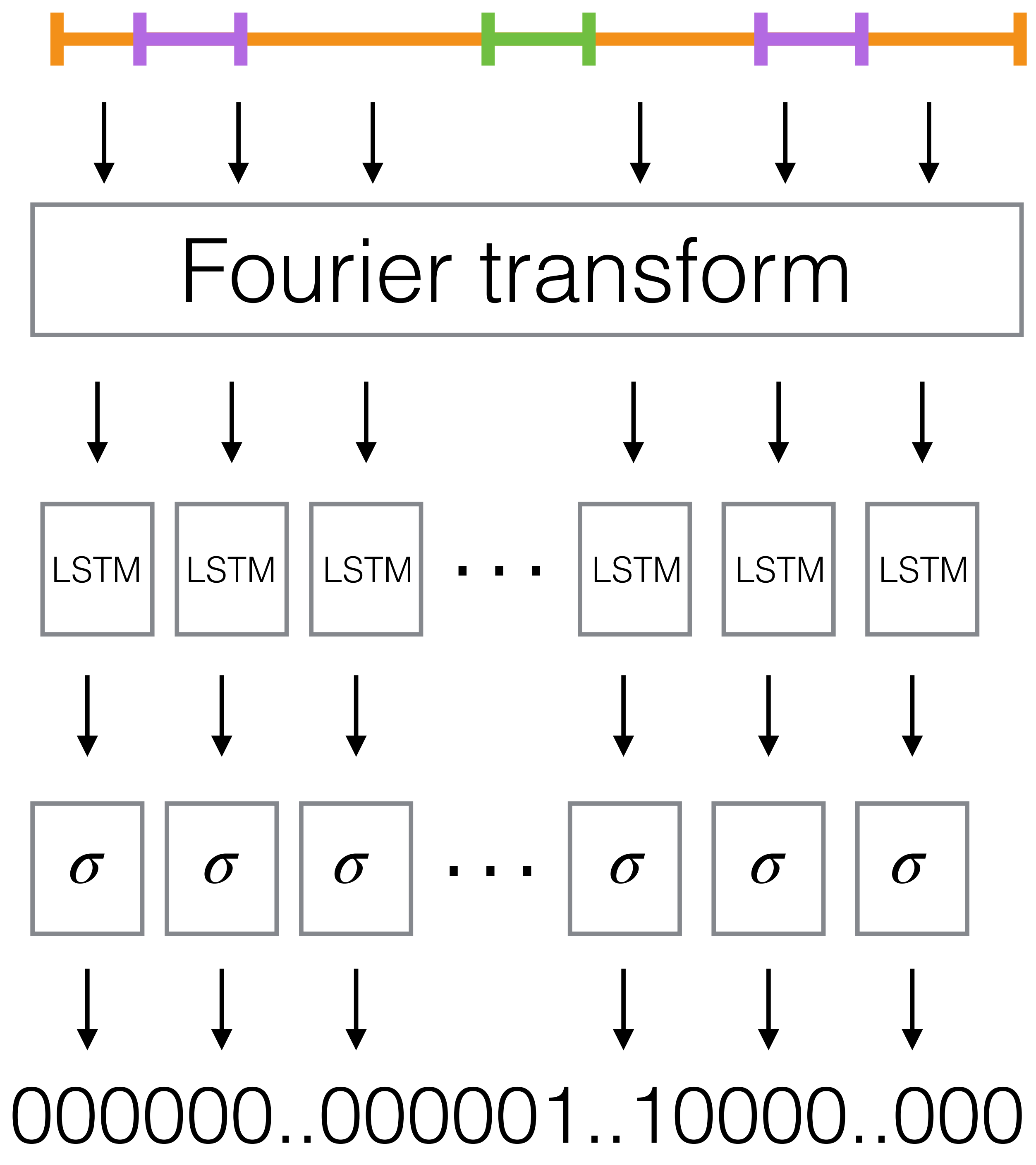
What is critical to the success of this project?

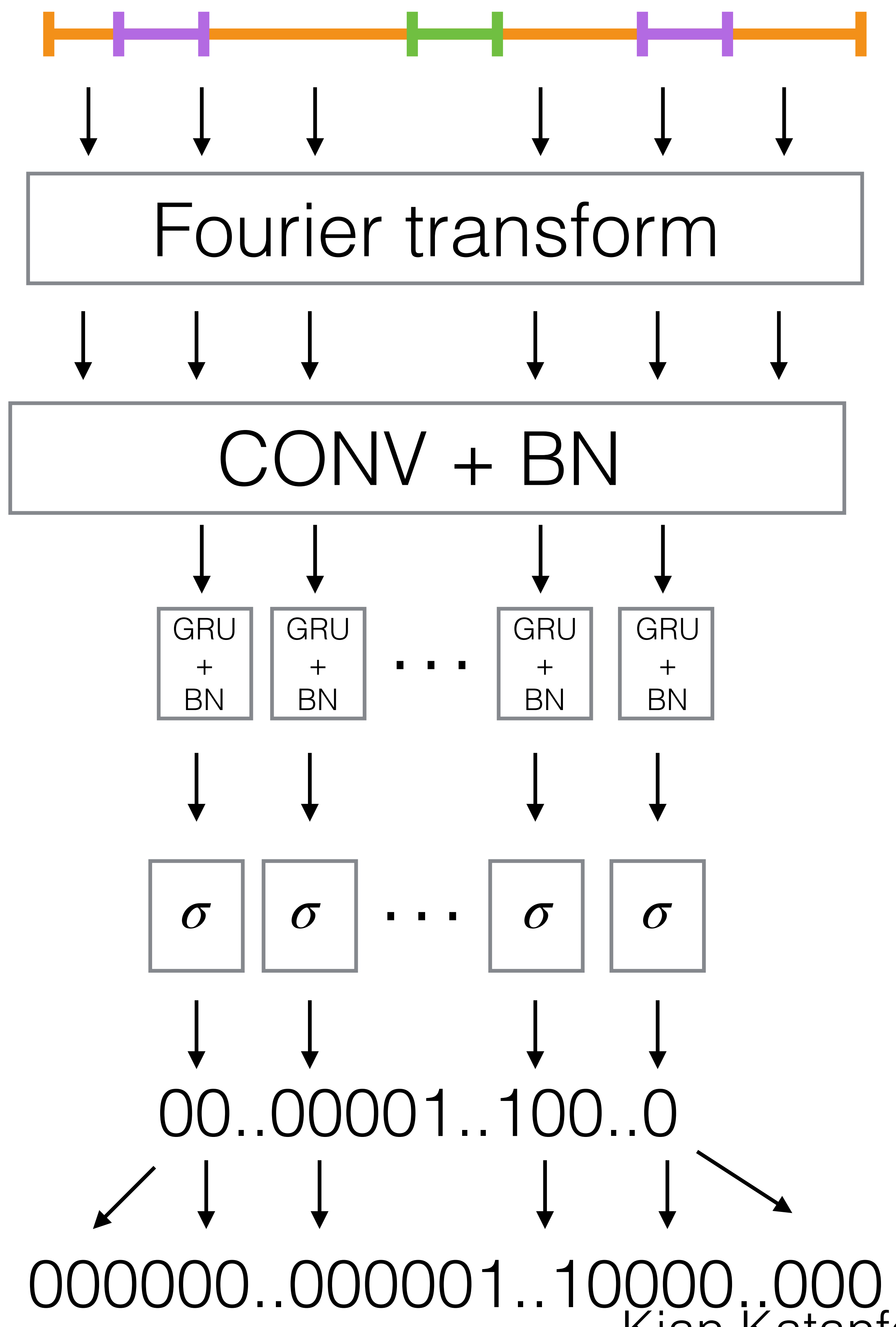
1. Strategic data collection/ labelling process



Automated labelling

2. Architecture search & Hyperparameter tuning





Never give up

+ Error analysis

Kian Katanforoosh

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 in the Magazine “the Most Beautiful Loss functions of 2015”

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection

Duties for next week

For Tuesday 01/21, 8am:

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 this week)**
- **Friday TA section (01/17)**
- **Fill-in AWS Form to get GPU credits for your projects**