CS230: Lecture 2
Key AI Concepts Through Case Studies

Kian Katanforoosh

Today's outline

I. Recap' of the week

II. Supervised Learning Projects

   A. Day & Night Classification

   B. Trigger Word Detection

   C. Face Verification

III. Self-Supervised Learning & Weakly Supervised Learning Projects

   A. Image Embeddings

   B. Multi-Modal Embeddings

Today's outline

I. **Recap' of the week**

II. Supervised Learning Projects

   A. Day & Night Classification

   B. Trigger Word Detection

   C. Face Verification

III. Self-Supervised Learning & Weakly Supervised Learning Projects

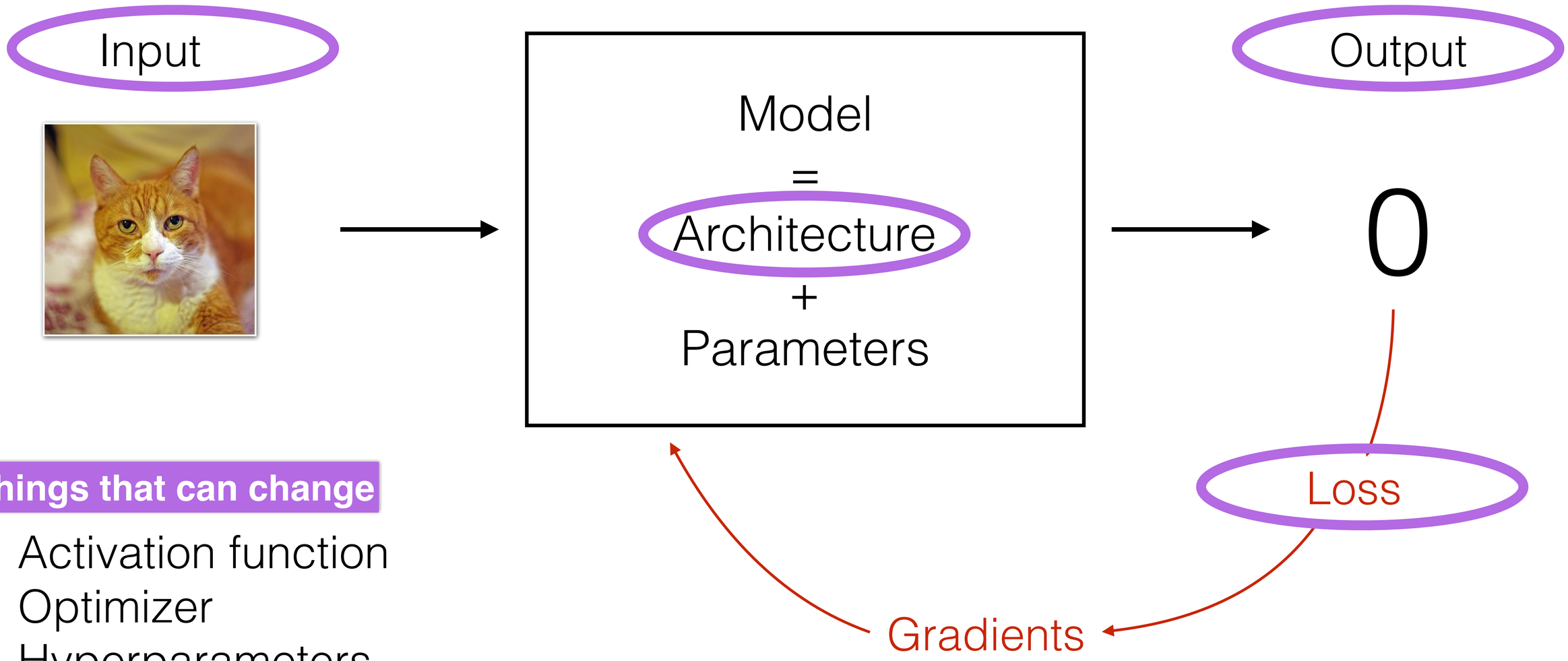
   A. Image Embeddings

   B. Multi-Modal Embeddings

Recap of the week

Kian Katanforoosh

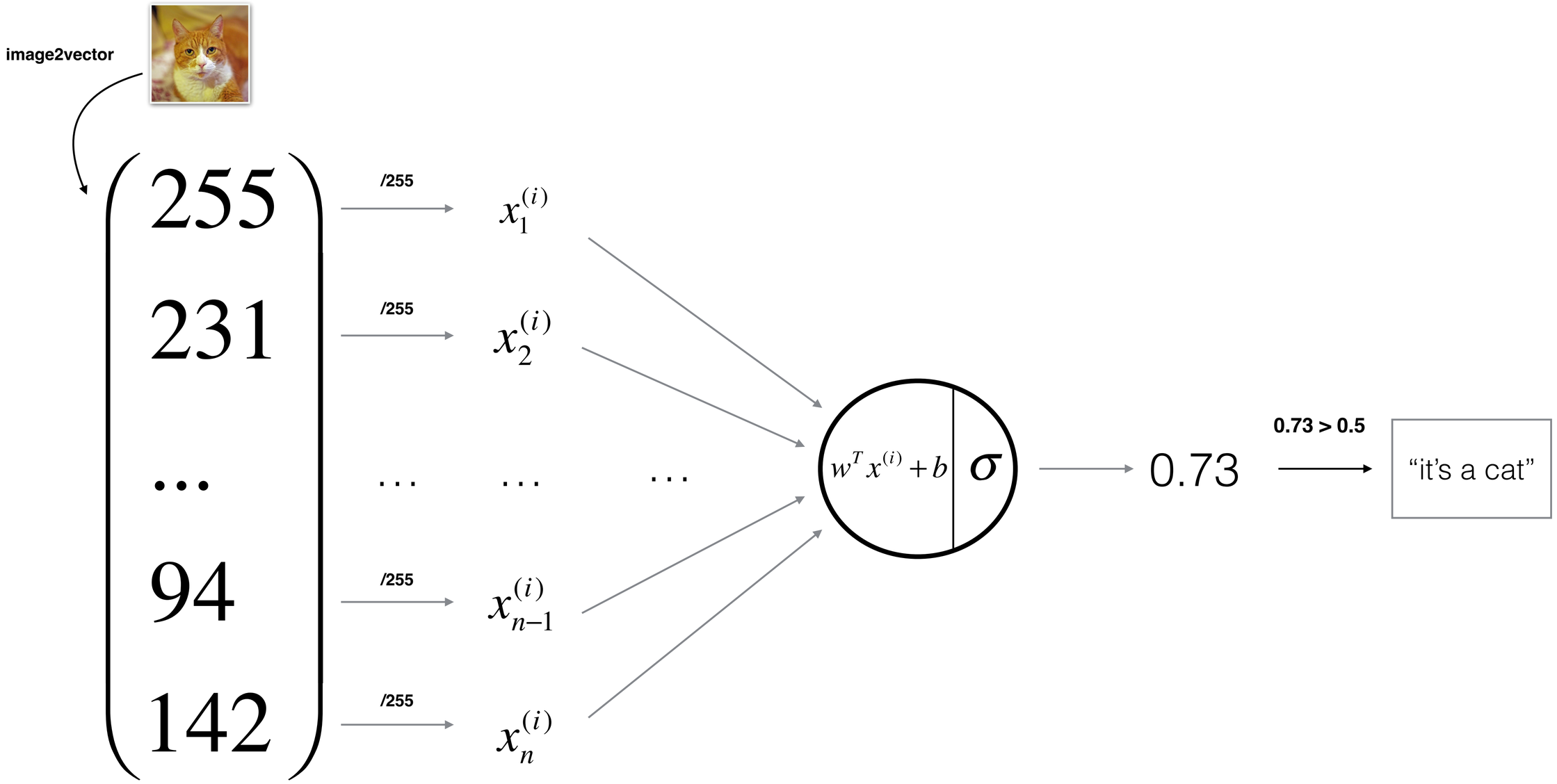
Learning Process



Things that can change

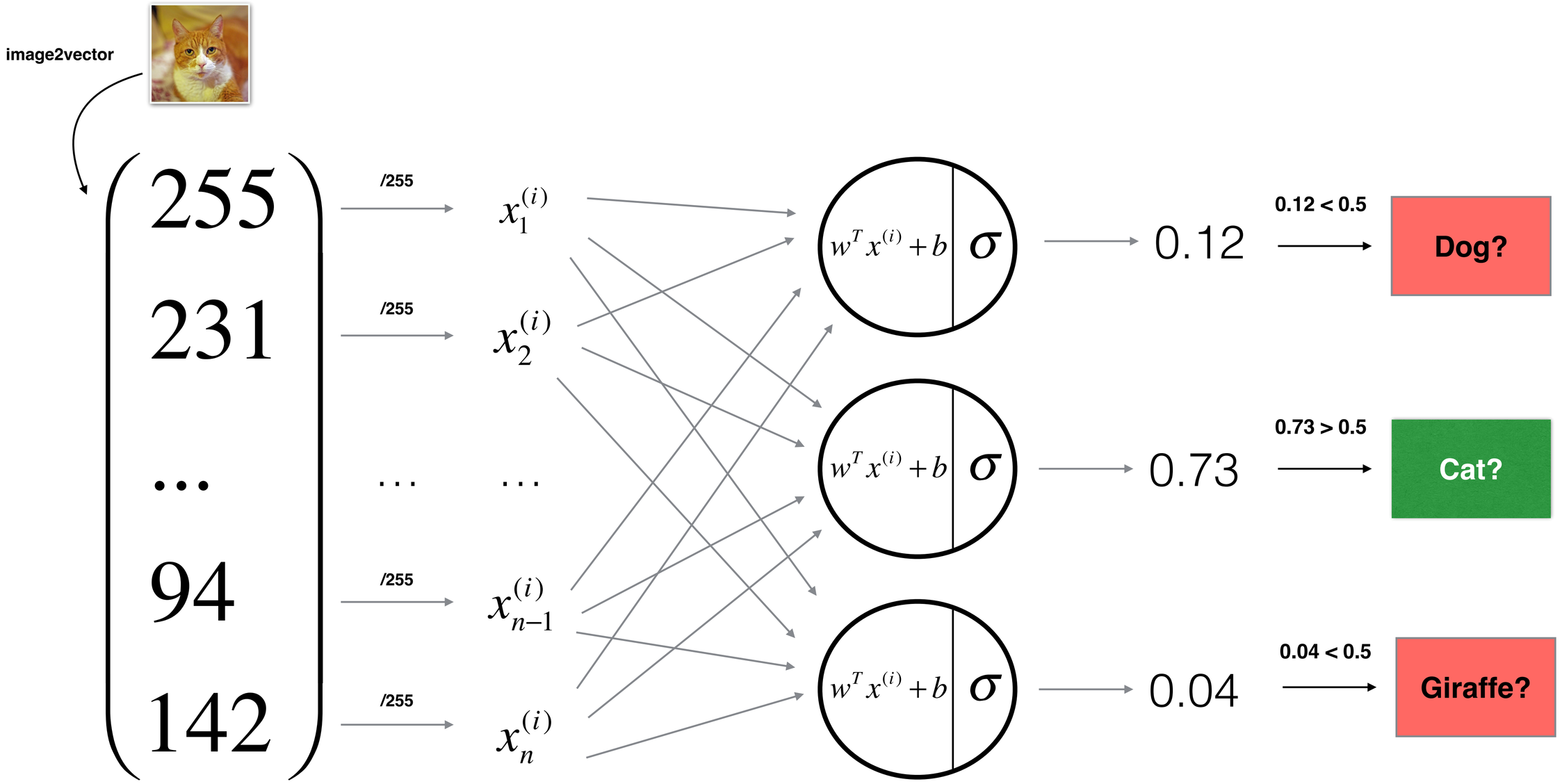
- Activation function
- Optimizer
- Hyperparameters
- …

Logistic Regression as a Neural Network



Kian Katanforoosh

Multi-class



Kian Katanforoosh

Neural Network (Multi-class)

image2vector

$$\begin{pmatrix} 255 \\ 231 \\ \dots \\ 94 \\ 142 \end{pmatrix}$$

/255 → $x_1^{(i)}$

/255 → $x_2^{(i)}$

$\dots$ $\dots$

/255 → $x_{n-1}^{(i)}$

/255 → $x_n^{(i)}$

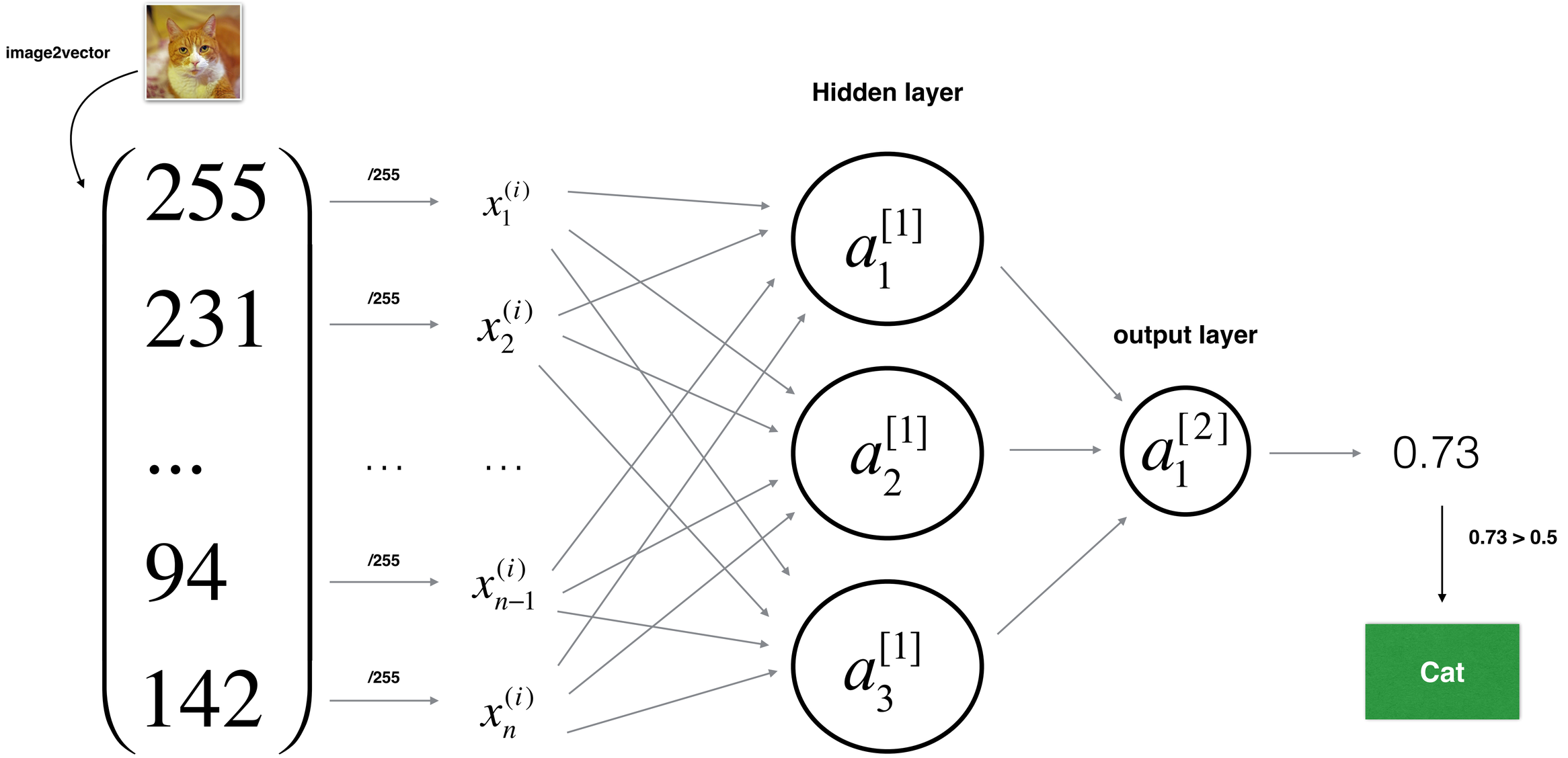
$w^T x^{(i)} + b \mid \sigma$

$w^T x^{(i)} + b \mid \sigma$

$w^T x^{(i)} + b \mid \sigma$

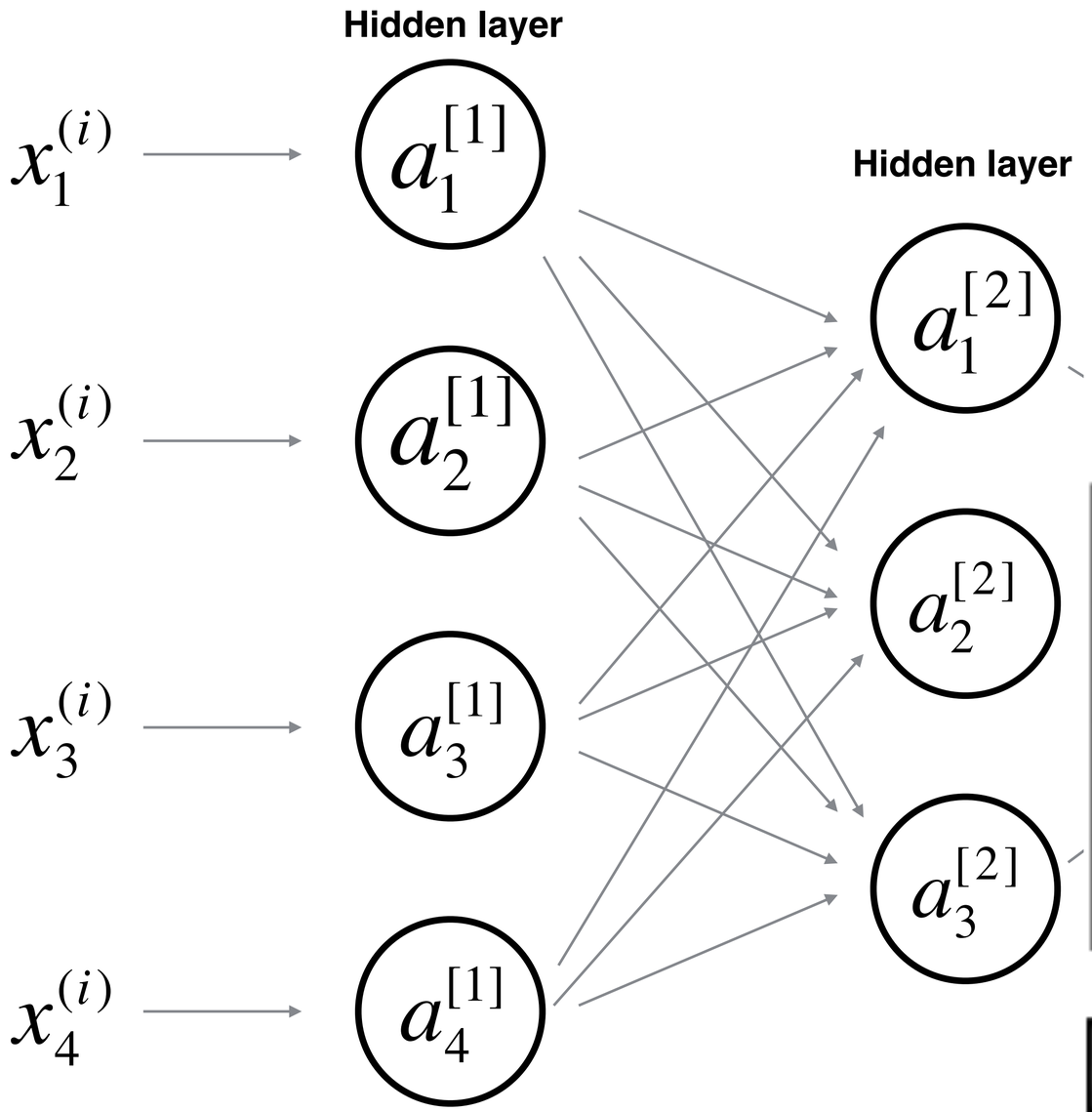
Kian Katanforoosh

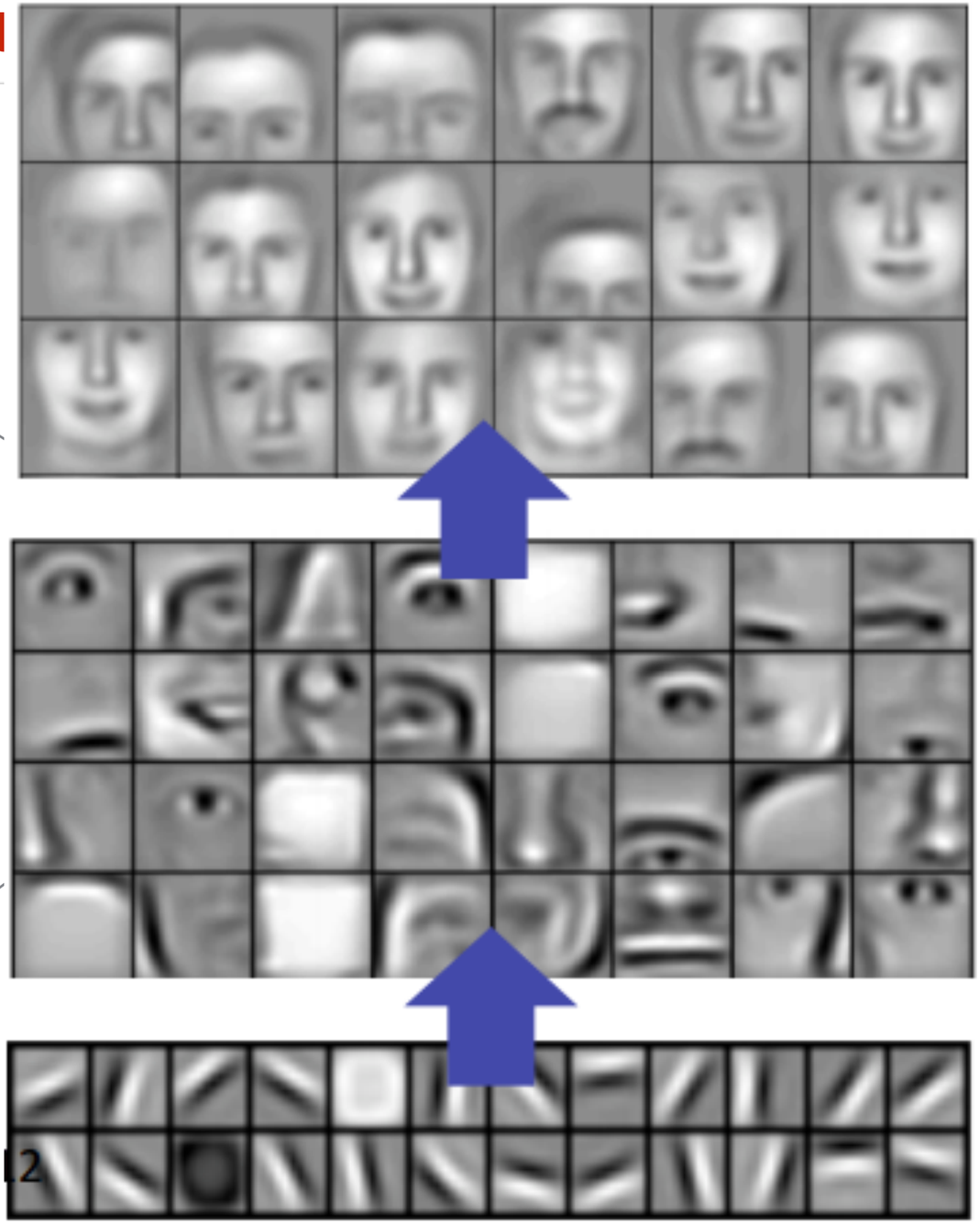
Neural Network (1 hidden layer)



Kian Katanforoosh

Deeper net





Technique called “encoding”

Summary of learnings: Introduction

New Words We Learned

- **Model = Architecture + Parameters**
  - **Architecture**: The blueprint or design of the neural network.
  - **Parameters**: The numerical values (weights, biases) the model learns during training.

- **Feature Engineering**: *Manual* — humans specify features.
- **Feature Learning**: *Automatic* — the network figures out features during training.

- **Encoding**: Any representation in vector form.
- **Embedding**: An encoding where **closeness = similarity**.

- **One-Hot Vector**: Exactly one thing is true.
- **Multi-Hot Vector**: Multiple things can be true at the same time.

~~Recap of the week~~

Kian Katanforoosh

Today's outline

I. Recap' of the week

II. Supervised Learning Projects

- A. Day & Night Classification
- B. Trigger Word Detection
- C. Face Verification

III. Self-Supervised Learning & Weakly Supervised Learning Projects

- A. Image Embeddings
- B. Multi-Modal Embeddings

A) Day & 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

Kian Katanforoosh

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.

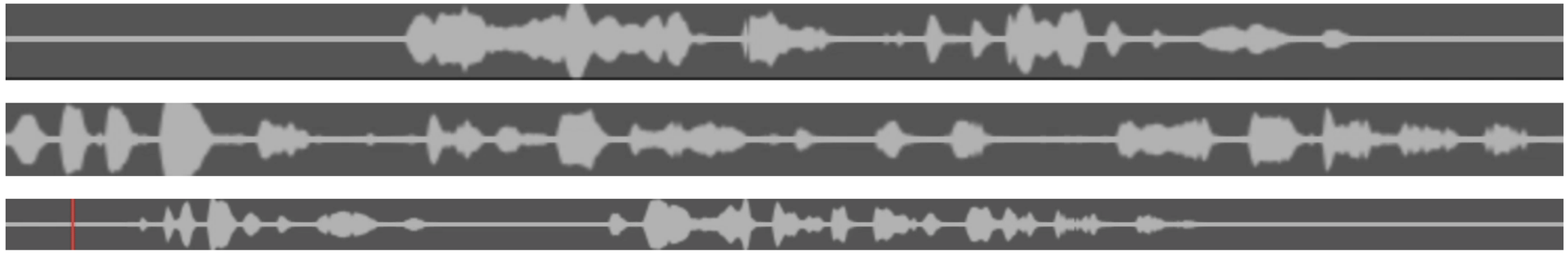
Case study: Trigger word detection

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 = 000000000000000000000000000000000000**1**0000000000

y = 00000000000000000000000000000000000000000000000000

y = 0000000000**1**0000000000000000000000000000000000000

Case study: Trigger word detection

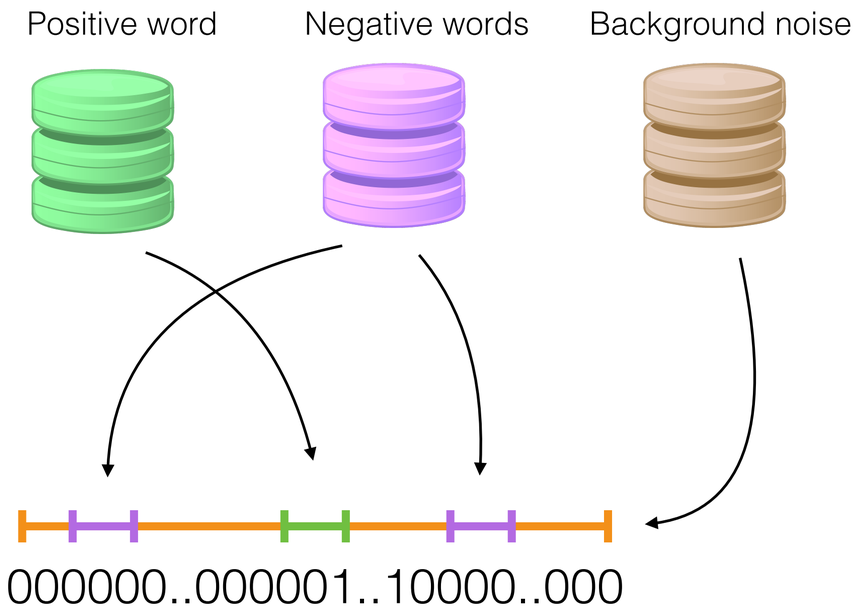
Goal: Given a 10sec audio speech, detect the word “activate”.

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)

Case study: Trigger word detection

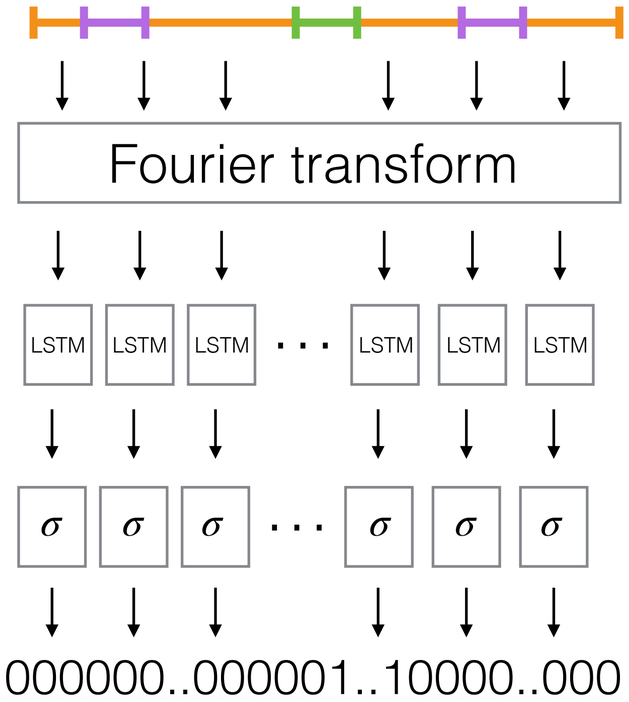
What is critical to the success of this project?

1. Strategic data collection/ labelling process

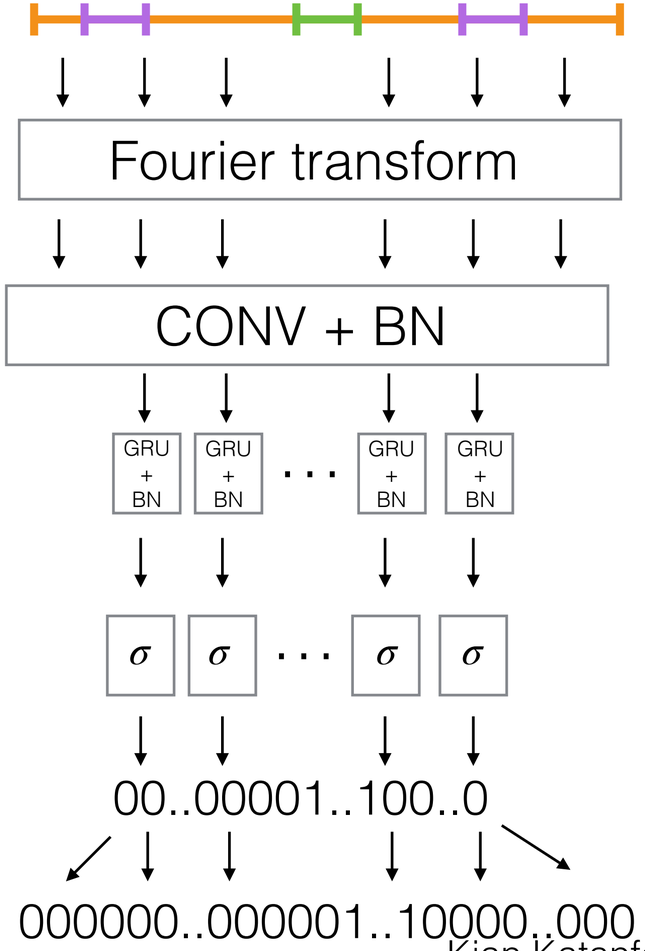


Automated labelling

2. Architecture search & Hyperparameter tuning



Architecture search



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.

Case Study: 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: 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:

database image

compute distance pixel per pixel

if less than threshold then y=1

input image

Issues:

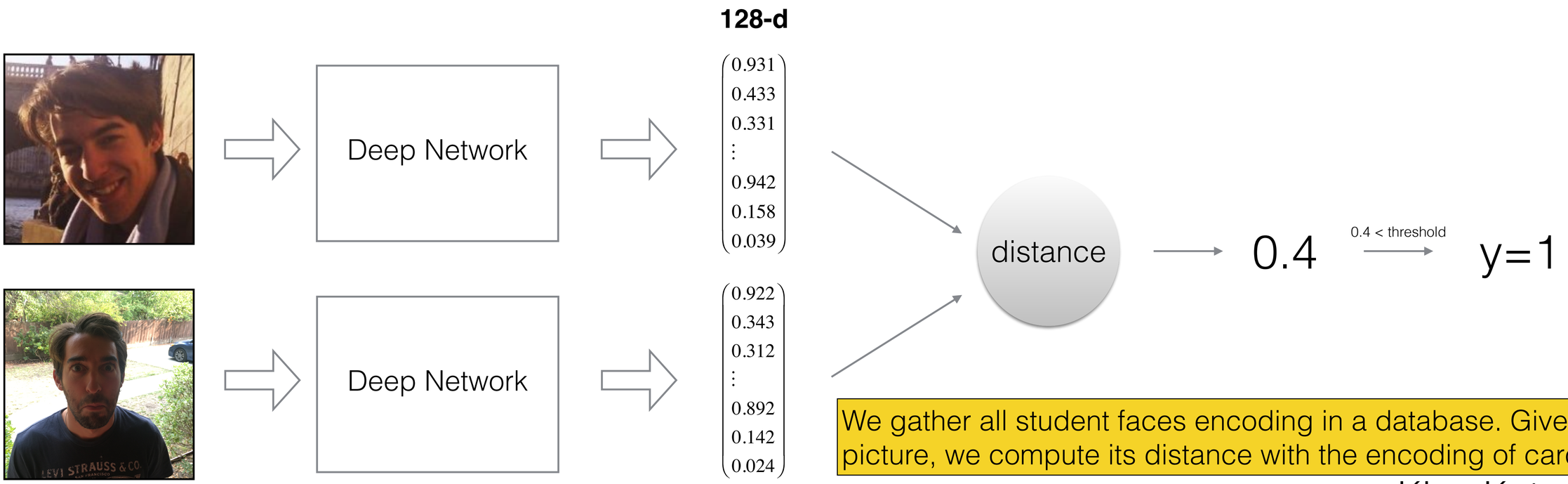
- Background lighting differences
- A person can wear make-up, grow a beard…
- ID photo can be outdated

Case Study: 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



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

Case Study: Face Verification

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

4. Loss? Training?

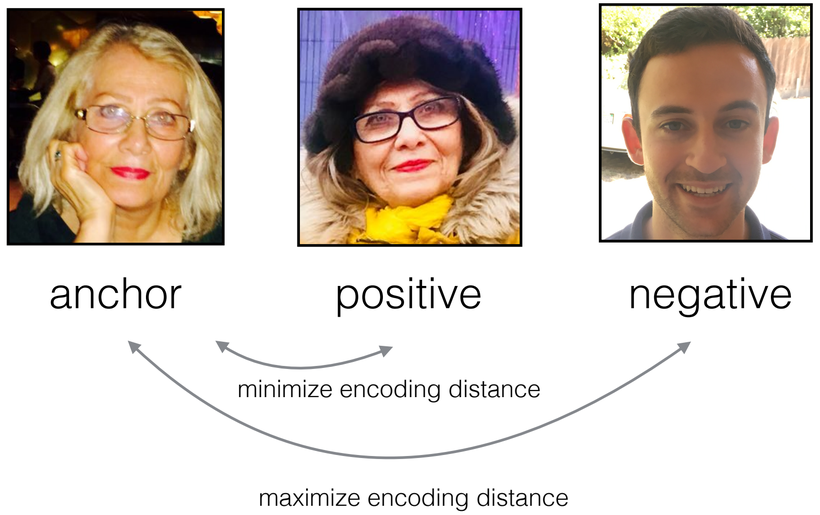
We need more data so that our model understands how to encode:
Use public face datasets

What we really want:

similar encoding

different encoding

So let's generate triplets:



[Schroff et al (2015): FaceNet: A Unified Embedding for Face Recognition and Clustering]

Case Study: 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?

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

A

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

B

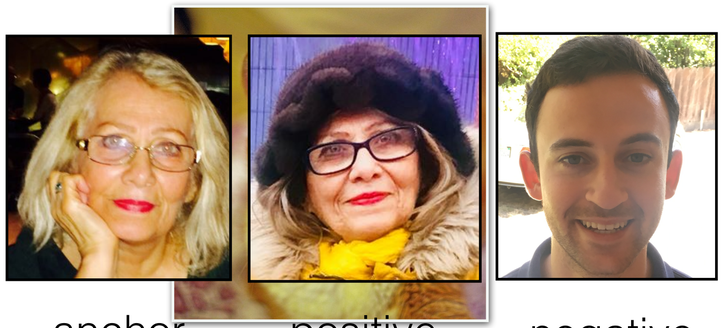
$$L = \left\| Enc(P) - Enc(N) \right\|_2^2 - \left\| Enc(P) - Enc(A) \right\|_2^2$$

C

[Schroff et al (2015): FaceNet: A Unified Embedding for Face Recognition and Clustering]

Case Study: Face Verification

Input



anchor positive negative

Model
=
Architecture
+
Parameters

Output

$$\begin{pmatrix}0.13\\0.42\\..\\0.10\\0.31\\0.73\\..\\0.43\\0.33\end{pmatrix} \quad \mathbf{O} \quad \begin{pmatrix}0.01\\0.54\\..\\0.45\\0.11\\0.89\\..\\0.12\\0.01\end{pmatrix} \quad \begin{pmatrix}0.95\\0.45\\..\\0.20\\0.41\\0.89\\..\\0.31\\0.34\end{pmatrix}$$

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

Loss

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

Gradients

[Schroff et al (2015): FaceNet: A Unified Embedding for Face Recognition and Clustering]

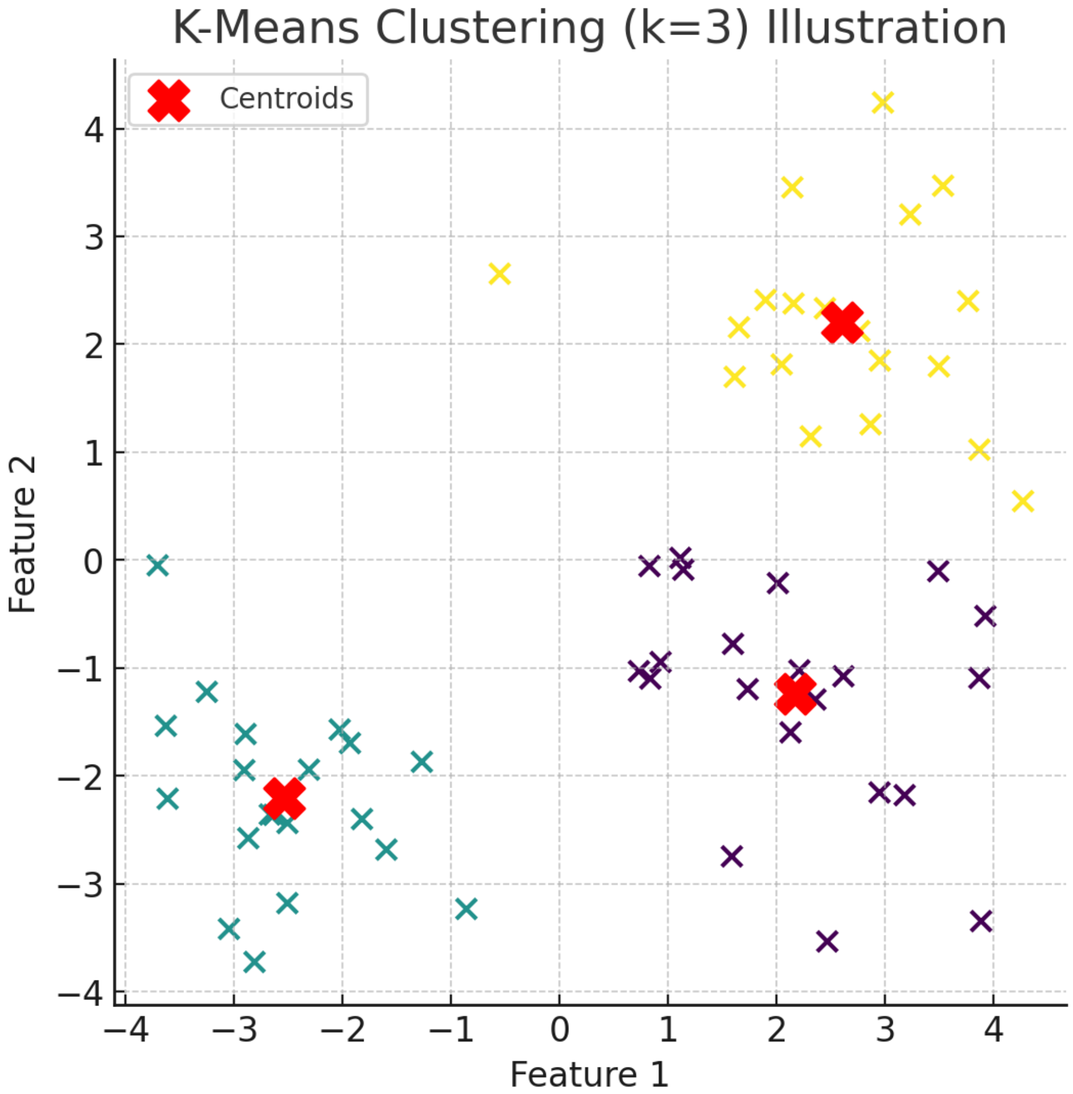
Case Study: Face Identification and Face Clustering

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



Kian Katanforoosh

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**.

Today's outline

I. Recap' of the week

II. Supervised Learning Projects

   A. Day & Night Classification

   B. Trigger Word Detection

   C. Face Verification

III. Self-Supervised Learning & Weakly Supervised Learning Projects

   A. Image Embeddings

   B. Multi-Modal Embeddings

From Supervised to Self-Supervised Approaches

Goal: Labeling can be expensive. How would you create embeddings for facial images without any labels?

In **contrastive self-supervised learning**, you generate *paired views* from the same image (via augmentation) and force representations of those views to agree (positive pairs), while pushing apart other images (negative pairs).

The idea: if a model can reliably tell that two transformed views come from the same image, then the embedding it learned is meaningful (captures content invariant to those augmentations).

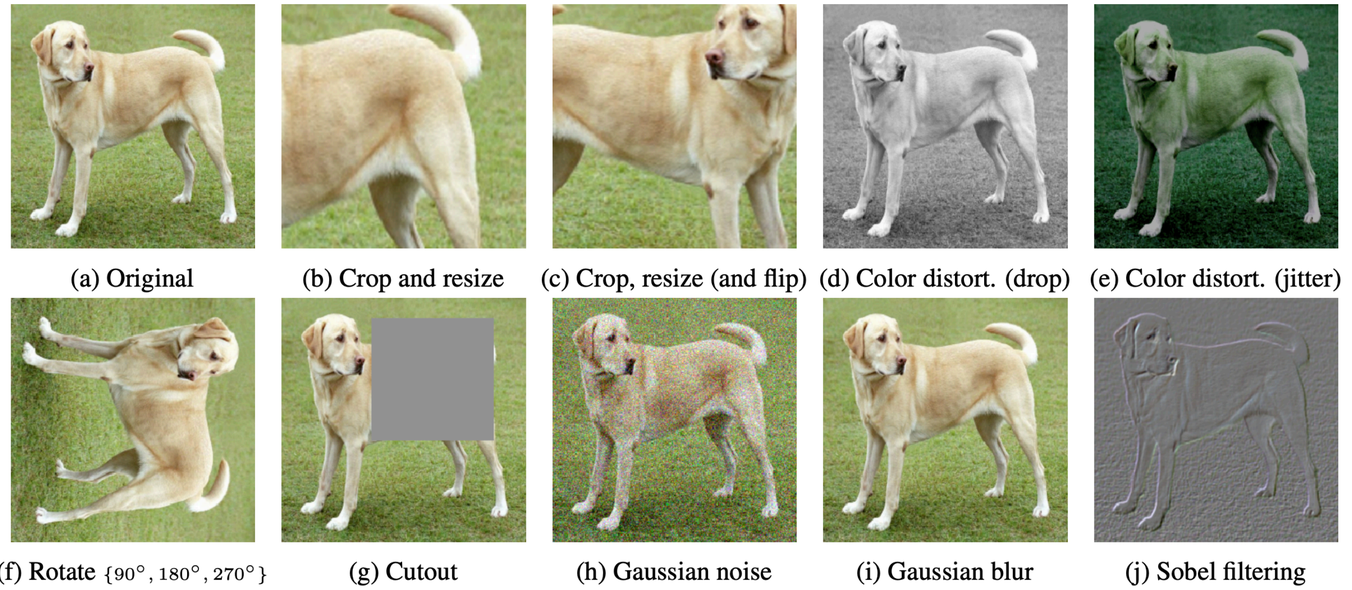


Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes *random crop (with flip and resize)*, *color distortion*, and *Gaussian blur*. (Original image cc-by: Von.grzanka)

This shift from supervised triplets to self-supervised pairs is why modern models can be trained on **billions of unlabeled images**. And the embeddings they learn power today's foundation models.

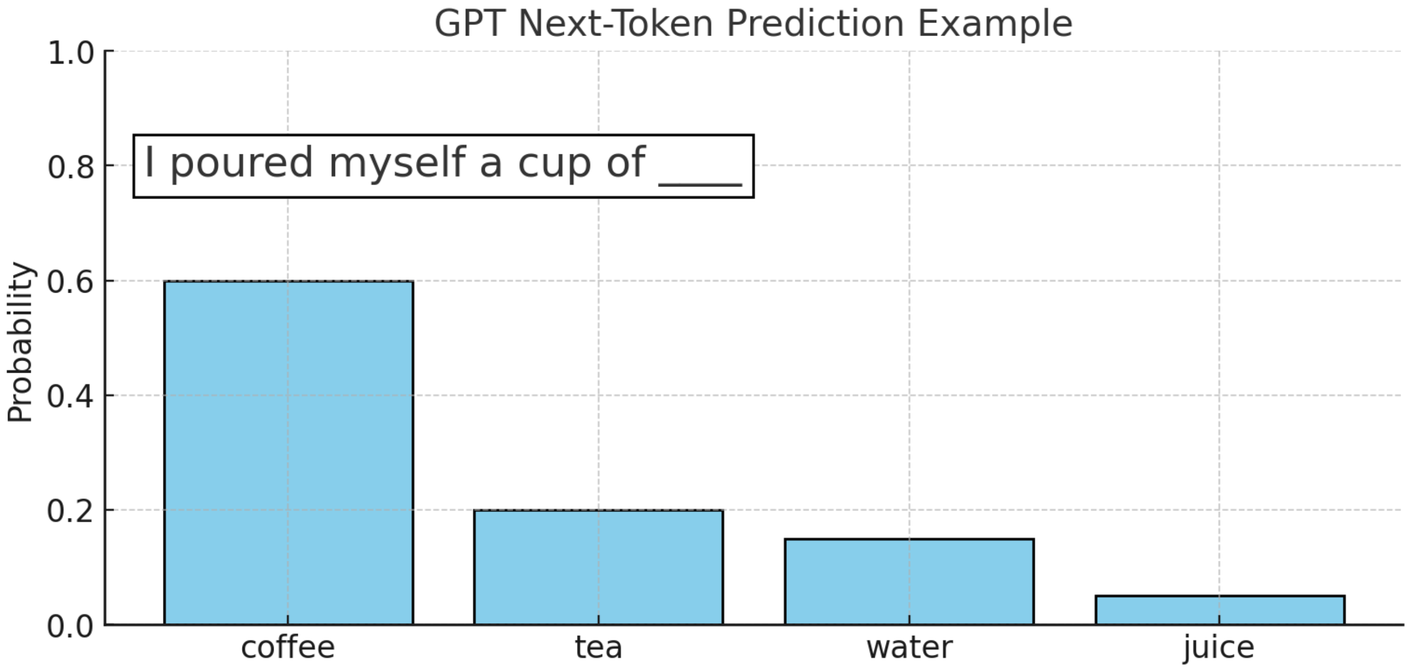
Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton: A Simple Framework for Contrastive Learning of Visual Representations (2020)

GPT also uses self-supervision, but with text.

The principle is the same: predict what belongs together, and push away what doesn't. In images, SimCLR says: two augmented views of the same photo should be close. In text, GPT does says: the next word should be consistent with the context."

What emergent behaviors stem from next-token prediction?

1. *"I poured myself a cup of _____"* → The model learns **ever**
2. *"The capital of France is _____"* → Predicting the next tok
3. *"She unlocked her phone using her _____"* → The model l **learns meanin**
4. *"The cat chased the _____"* → Multiple completions are plau
5. *"If it's raining, I should bring an _____"* → The model can **re likely consequences**



Emergent behaviors are unexpected capabilities that arise from simple training objectives at scale, without being explicitly taught or labeled.

Improving Language Understanding by Generative Pre-Training (Radford et al., 2017)

Kian Katanforoosh

Self-supervision isn't just about text. It's about creating a prediction task from raw data itself.

- **Text** → *Next-token prediction (GPT)*
- **Images** → *Contrastive learning (SimCLR)*

What other examples can you think of?

- **Audio** → *Predict missing audio chunks from surrounding sound.*
- **Video** → *Predict the next frame, or the correct order of shuffled frames.*
- **Biology / Proteins** → *Hide one amino acid in a sequence and predict it*

But the world is multimodal. We experience words, images, sounds, and actions together. How do we connect them?

The multimodal embedding task is technically supervised (because you rely on *paired* data like “image ↔ caption”), but

it’s a very *weak* or *naturally occurring* form of supervision, not like ImageNet-style hand labeling.

This is often called “**weakly supervised**”.

The cat is looking at the camera, a bit confused

Can you think of some examples where different types of data naturally come paired together?

How would you align modalities that are not naturally occurring together in available datasets?

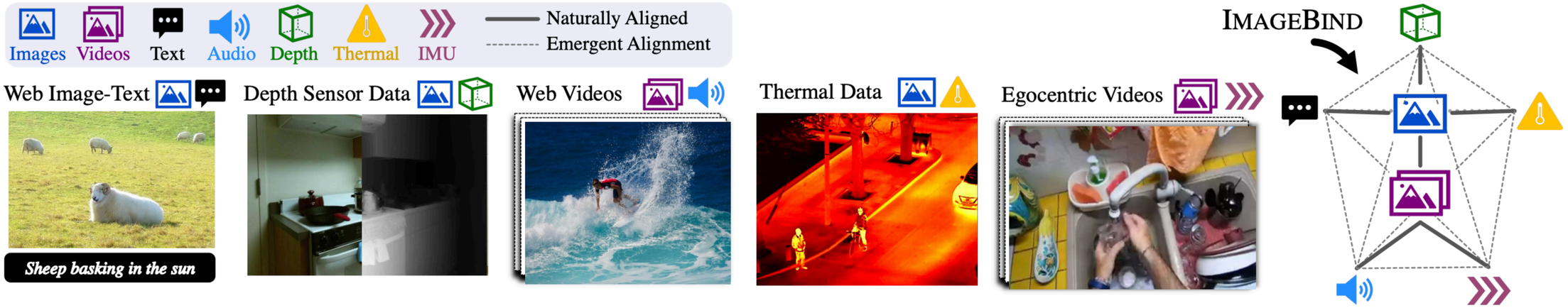


Figure 2. IMAGEBIND overview. Different modalities occur naturally aligned in different data sources, for instance images+text and video+audio in web data, depth or thermal information with images, IMU data in videos captured with egocentric cameras, *etc*. IMAGEBIND links all these modalities in a common embedding space, enabling new emergent alignments and capabilities.

IMAGEBIND: One Embedding Space To Bind Them All (Girdhar et al., 2023) + Demo

Kian Katanforoosh

Summary of learnings: Embeddings (Self-Supervised Learning)

- **Embedding**: An encoding where closeness = similarity.
- **Self-supervised learning** is a method where models create their own training labels from raw data by solving prediction tasks inherent in the data itself.
- **Contrastive learning** uses **data transformations** to create similar pairs, pulling them together while pushing others apart.
- **Next-token prediction** trains a model to guess the most likely next word in a sequence given all the previous words.
- **Weakly supervised** = training models with imperfect or automatically generated labels instead of precise human ones
- We can learn a **shared embedding space** across modalities (images, text, audio, depth, thermal, and IMU sensors) using text as the **central pivot**.