CS230: Lecture 3
Various Deep Learning Topics
Kian Katanforoosh, Andrew Ng
Today’s outline

We will learn how to:
- Analyse a problem from a deep learning approach
- Choose an architecture
- Choose a loss and a training strategy

I. Day’n’Night classification
II. Face Recognition
III. Art generation
IV. Object detection
V. Image Segmentation
Day’n’Night classification (warm-up)

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

1. **Data?** 10,000 images  
2. **Input?** Resolution? (64, 64, 3)  
3. **Output?** y = 0 or y = 1  
4. **Architecture?** Shallow network should do the job pretty well  
5. **Loss?** \[ L = y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}) \]
Server-based or on-device?

**Server-based**
- App is light-weight

**On-device**
- Faster predictions

Andrew Ng, Kian Katanforoosh
**Face Recognition**

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

4. **What architecture?**

Simple solution:

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

**Issues:**

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- ID photo can be outdated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

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

4. **What architecture?**

Our solution: encode information about a picture in a vector

![Diagram](image)

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

\[
L = \|\text{Enc}(A) - \text{Enc}(P)\|_2^2 - \|\text{Enc}(A) - \text{Enc}(N)\|_2^2 + \alpha
\]
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?
**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

*Leon A. Gatys, Alexander S. Ecker, Matthias Bethge: A Neural Algorithm of Artistic Style, 2015*
Art generation (Neural Style Transfer)

4. Architecture?
We want a model that **understands images** very well
We load an **existing model trained on ImageNet** for example

![Deep Network](image)

When this image forward propagates, we can get information about its content & its style by inspecting the layers.

5. Loss?

\[
L = \|\text{Content}_C - \text{Content}_G\|_2^2 + \|\text{Style}_S - \text{Style}_G\|_2^2
\]

We are not learning parameters by minimizing L. We are learning an image!
Art generation (Neural Style Transfer)

Correct Approach

\[ L = \|\text{Content}_C - \text{Content}_G\|^2 + \|\text{Style}_S - \text{Style}_G\|^2 \]

After 2000 iterations

Deep Network (pretrained)

compute loss

update pixels
Image Segmentation

**Goal**: Separate the foreground from the background on a picture

1. **Data?**
   - Image

2. **Input?**
   - Labels

3. **Output?**
   - Image labels

4. Architecture?

(600, 400, 3)

Convolutions
(reduces volume height and width)

Information Encoded

Encoding

De-convolutions
(increases volume height and width)

Per-Pixel Classification
(600, 400, 1)
4. Loss?

pixel-wise cross-entropy

\[ L = \sum_{\text{pixels}} \sum_{\text{classes}} y \log(\hat{y}) \]
Object Detection

**Goal:** Find objects in images

1. **Data?**
   
   Very large set of labelled images

2. **Input?**

   ![Input Image]

3. **Output?**

   
   \[ y_1 = (b_x, b_y, b_h, b_w, p_c, c) \]
   
   \[ y_2 = (b_x, b_y, b_h, b_w, p_c, c) \]
   
   \[ y_k = (b_x, b_y, b_h, b_w, p_c, c) \]

   Problem: size of output varies
   1. Use a mask?
   2. Change the output of the model

   \[ y = (b_x, b_y, b_h, b_w, p_c, c) \]
4. Architecture?

We have a lot of boxes
We select the most likely ones using thresholding and other methods
5. Loss?

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

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

\[ + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij} (C_i - \hat{C}_i)^2 \]

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

\[ + \sum_{i=0}^{S^2} \mathbb{1}_{ij} \sum_{\text{classes} \in \{c\}} (p_i(c) - \hat{p}_i(c))^2 \]
**Goal**: Find objects in images

1. **Data?**
   Very large set of labelled images

2. **Input?**
   ![Image with bounding boxes]

3. **Output?**
   
   \[ y_1 = (b_x, b_y, b_h, b_w, p_c, c) \]
   
   \[ y_2 = (b_x, b_y, b_h, b_w, p_c, c) \]
   
   \[ y_k = (b_x, b_y, b_h, b_w, p_c, c) \]

Problem: size of output varies
1. Use a mask?
2. Change the output of the model

\[ y = (b_x, b_y, b_h, b_w, p_c, c) \]

Andrew Ng, Kian Katanforoosh