CS230: Lecture 2
Deep Learning Intuition
Kian Katanforoosh
Recap
Learning Process

Model = Architecture + Parameters

Input → Model → Output

Loss → Gradients

Things that can change
- Activation function
- Optimizer
- Hyperparameters
- ...

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
Logistic Regression as a Neural Network

\[
\begin{pmatrix}
255 \\
231 \\
94 \\
142 \\
\end{pmatrix}
\rightarrow \begin{align*}
x_1^{(i)} \\
x_2^{(i)} \\
x_{n-1}^{(i)} \\
x_n^{(i)} \\
\end{align*}

w^T x^{(i)} + b \rightarrow \begin{array}{c}
\sigma \\
0.73 \\
\end{array}

0.73 > 0.5 \rightarrow \text{“it’s a cat”}

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
Multi-class

\[
\begin{pmatrix}
255 \\
231 \\
\vdots \\
94 \\
142
\end{pmatrix}
\]

\[
\begin{array}{c}
\frac{x_1^{(i)}}{255} \\
\frac{x_2^{(i)}}{255} \\
\vdots \\
\frac{x_{n-1}^{(i)}}{255} \\
\frac{x_n^{(i)}}{255}
\end{array}
\]

\[
\mathbf{w}^T \mathbf{x}^{(i)} + b
\]

\[
\sigma
\]

\[
\frac{0.73 > 0.5}{\text{Cat?}}
\]

\[
\frac{0.12 < 0.5}{\text{Dog?}}
\]

\[
\frac{0.04 < 0.5}{\text{Giraffe?}}
\]

\[
\frac{0.73}{\text{Cat?}}
\]

\[
\frac{0.12}{\text{Dog?}}
\]

\[
\frac{0.04}{\text{Giraffe?}}
\]
Neural Network (Multi-class)

\[
\begin{pmatrix}
255 \\
231 \\
\vdots \\
94 \\
142
\end{pmatrix} \rightarrow \begin{cases}
x^{(i)}_1 \\
x^{(i)}_2 \\
\vdots \\
x^{(i)}_{n-1} \\
x^{(i)}_n
\end{cases}
\]

\[
x^{(i)}_1 = w^T x^{(i)} + b
\]

\[
x^{(i)}_2 = w^T x^{(i)} + b
\]

\[
x^{(i)}_{n-1} = w^T x^{(i)} + b
\]

\[
x^{(i)}_n = w^T x^{(i)} + b
\]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
Neural Network (1 hidden layer)

Hidden layer

output layer

0.73

Cat

0.73 > 0.5

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
Deeper network: Encoding

Technique called “encoding”
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
V. Shipping model
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)
   ![Input Image](image)

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**
**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)
   - 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:

- Database image
- Input image
- 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
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

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

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

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

Model = Architecture + Parameters

Input

Output

Loss

Gradients

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

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?
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
4. **Architecture?**

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

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

5. **Loss?**
LiveSlides web content

To view

Download the add-in.
liveslides.com/download

Start the presentation.
Art generation (Neural Style Transfer)

Correct Approach

\[ L = \|Content_C - Content_G\|_2^2 + \|Style_S - Style_G\|_2^2 \]

We are not learning parameters by minimizing \( L \). We are learning an image!

After 2000 iterations

Deep Network (pretrained)

compute loss

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

1. Data? A bunch of 10s audio clips
2. Input? x = A 10sec audio clip
3. Output? y = 0 or y = 1
Let's have an experiment!
**Trigger word detection**

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

1. **Data?**
   - A bunch of 10s audio clips

2. **Input?**
   - $x = \text{A 10sec audio clip}$
     - $y = 0$ or $y = 1$

3. **Output?**
   - $y = 00..000100000..000$
   - $y = 00..00001..100000..000$

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

5. **Loss?**
   - $L = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$
     - (sequential)
What is critical to the success of this project?

1. **Strategic data collection/labelling process**
   - Positive word
   - Negative words
   - Background noise

   000000..000001..10000..000

   Automated labelling

   + Error analysis

2. **Architecture search & Hyperparameter tuning**

   Fourier transform

   000000..000001..10000..000

   Never give up

   Trigger word detection
Another way of solving the TWD problem?
**Trigger word detection (other method)**

**Goal**: Given an audio speech, detect the word “lion”.

4. **What architecture?**

\[ y = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) \]

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

\[ -\| Enc(A) - Enc(N) \|_2^2 \]

\[ + \alpha \]

Threshold: 0.6

[Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri]

[For more on query-by-example trigger word detection, check: Guoguo Chen et al.: Query-by-example keyword spotting using long short-term memory networks (2015)]
\[
\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j} \left[ (\sqrt{w}_i - \sqrt{\hat{w}_i})^2 + (\sqrt{h}_i - \sqrt{\hat{h}_i})^2 \right] \\
+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j} \left( C_i - \hat{C}_i \right)^2 \\
+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j} \left( C_i - \hat{C}_i \right)^2 \\
+ \sum_{i=0}^{S^2} \sum_{c \in \text{classes}} \left( p_i(c) - \hat{p}_i(c) \right)^2
\]
App implementation
Server-based or on-device?

Server-based

On-device

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
Server-based or on-device?

**Server-based**
- App is light-weight
- App is easy to update

**On-device**
- Faster predictions
- Works offline
Duties for next week

For Wednesday 10/09, 10am:

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 (10/05): focus on git and neural style transfer.
• Fill-in AWS Form to get GPU credits for your projects