

SAARBRÜCKEN

6TH SUMMER SCHOOL ON COMPUTATIONAL INTERACTION

INFERENCE, OPTIMIZATION AND MODELING FOR THE
ENGINEERING OF INTERACTIVE SYSTEMS | 13 - 18 JUNE 2022

Deep Learning for Human–Computer Interaction **Session 4: Hands-on Deep Learning Frameworks**



PyTorch



Keras



TensorFlow

Luis Leiva & Bereket Yilma
University of Luxembourg

Timeline

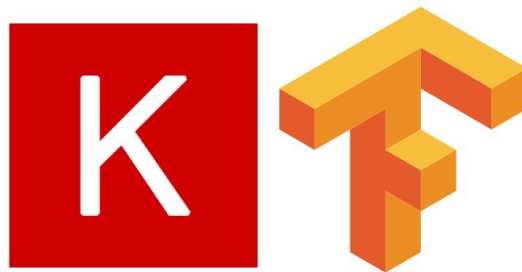
45min [13:30 - 14:15] **Pytorch & Keras overview**

15min [14:15 - 14:30] **Visual clutter classifier**

15min [14:30 - 14:45] **Fitts law regressor**

15min [14:45 - 15:00] Break

30min [15:00 - 15:30] **Clustering of UI wireframes**



Learning outcomes

After this session you will be able to:

- Understand the basic building blocks for creating a DL model
- Know the popular DL frameworks (PyTorch and Keras)
- Learn how to use these frameworks to solve HCI problems

Dependencies

Create a virtual environment and install [requirements](#)

using pip

```
pip install -r requirements.txt
```

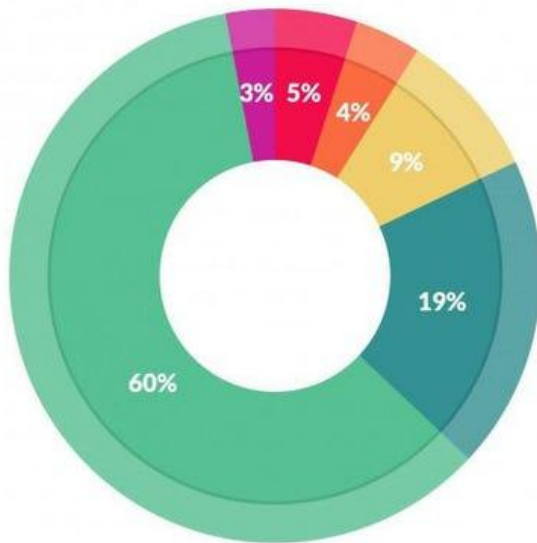
using Conda

```
conda create --name <env_name> --file requirements.txt
```

Steps for training a DL model

1. Loading data ← Data Preprocessing
2. Defining a Neural Network
3. Defining a Loss function and optimizer
4. Training
5. Testing

1. Loading data ← Data Preprocessing



What data scientists spend the most time doing

- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

<https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/>

Steps for training a DL model

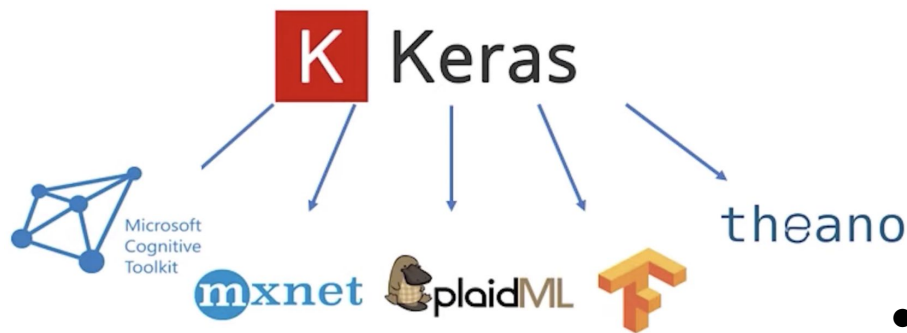
1. Loading data ← Data Preprocessing
2. Defining a Neural Network
3. Defining a Loss function and optimizer
4. Training
5. Testing





- A low-level API developed by Facebook for natural language processing and computer vision.
- Emphasizes flexibility and allows deep learning models to be expressed in basic python

Keras



- A high-level deep learning API written in Python for easy implementation and computation of neural networks.
- It also integrates with various backend engines to help with fast computation while maintaining ease of implementation.

Tensors

- Specialized data structure that are very similar to arrays and matrices.
- Used to encode the inputs and outputs of a DL model, as well as the model's parameters.
- Tensors are **IMMUTABLE**. Just like python tuples and sets you can NOT update the contents of a tensor. You can only create a new tensor.

Multiclass classification

label = 5



label = 0



label = 4



label = 1



label = 9



label = 2



label = 1



label = 3



label = 1



label = 4



label = 3



label = 5



label = 3



label = 6



label = 1



label = 7



label = 2



label = 8



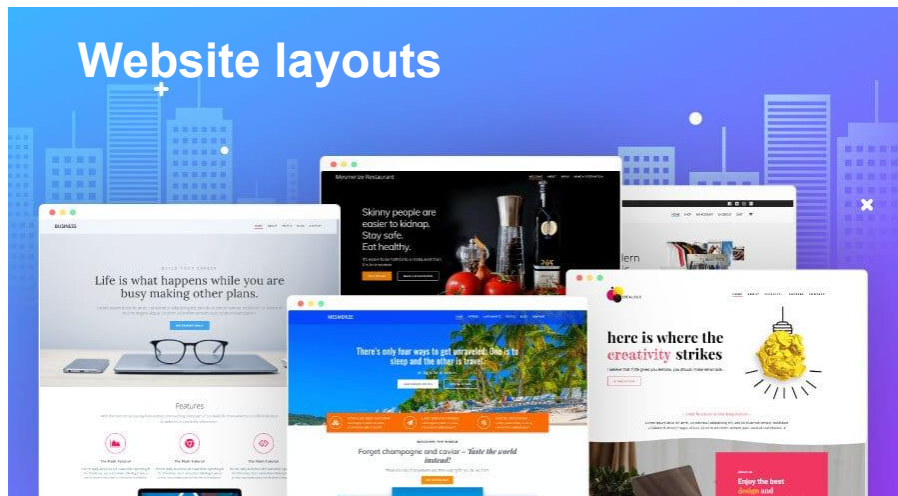
label = 6



label = 9



Exercise 1: Visual clutter classifier



There are many reasons to consider a choice of layout wisely, for example:

- Good layout → important information easily accessible and intuitive to find.
- Bad layout → Frustrates users, hard to find important information, user leaves the site quickly.

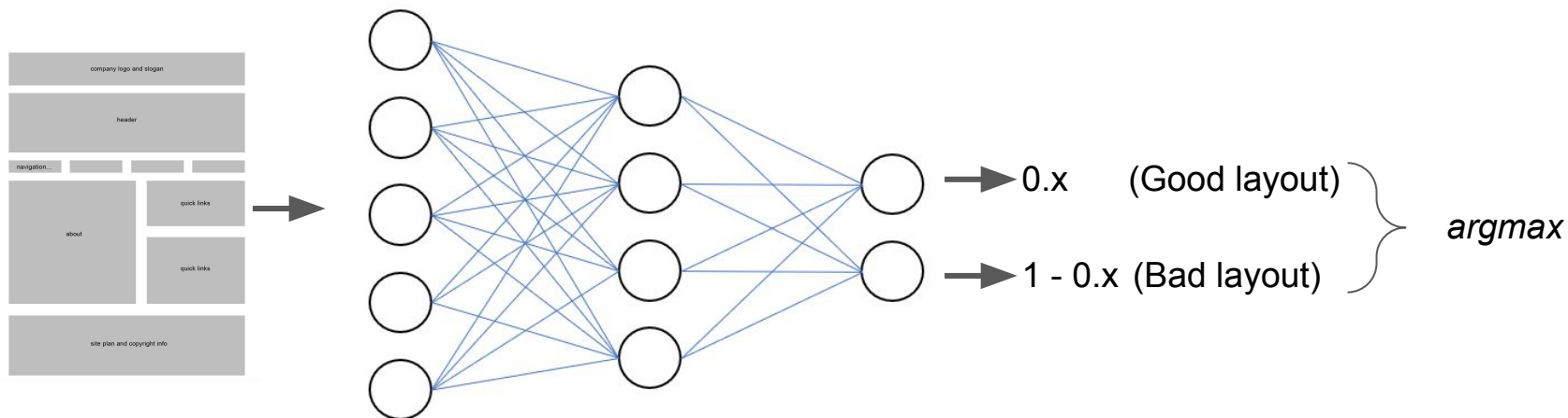
There's a strong relationship between a website's layout and the engagement of users. It determines:

- How long they dwell on the website pages,
- How many pages they browse.
- How often they come back to the website.

According to Adobe research, **38%** of people will stop engaging with a website if the content/layout is unattractive.

Exercise 1: Visual clutter classifier

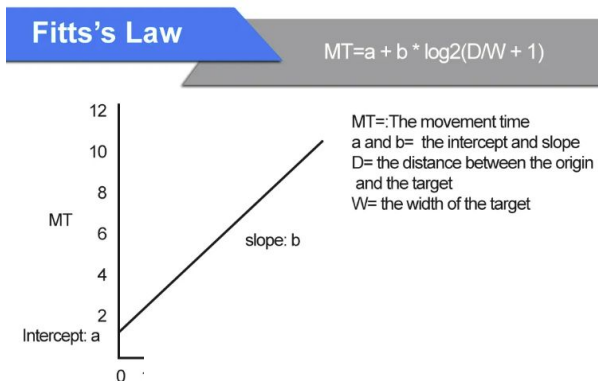
- In this exercise you will develop a DL model that can predict automatically how visually packed and/or disorganized a UI is.
- Cluttered UIs cause decreased recognition performance, therefore this model has potential for providing UI designers with "a priori" estimations of visual clutter, without having to actually look at the UI.



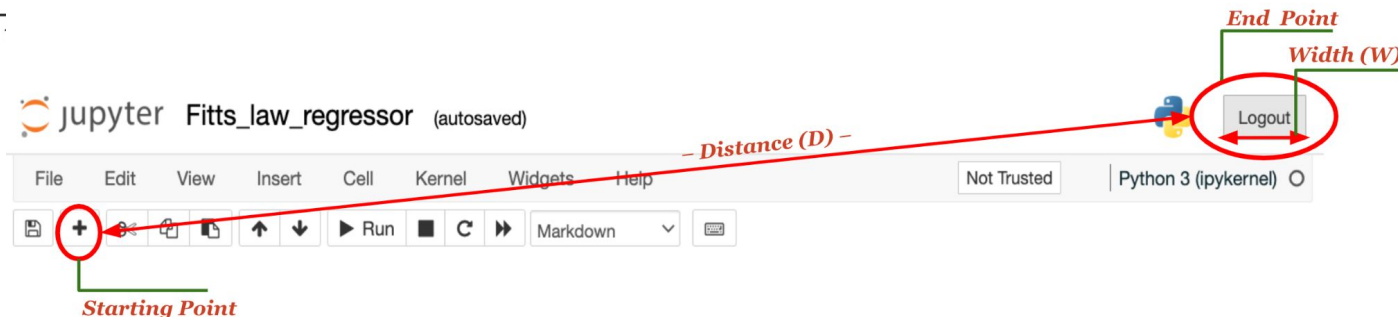
- could be offered as a third-party service or integrated into an existing one such as <https://interfacemetrics.aalto.fi/>.

Exercise 2: Fitts Law Regressor

In this exercise, we are going to use the key concept from human-interaction theory - the Fitts's law.

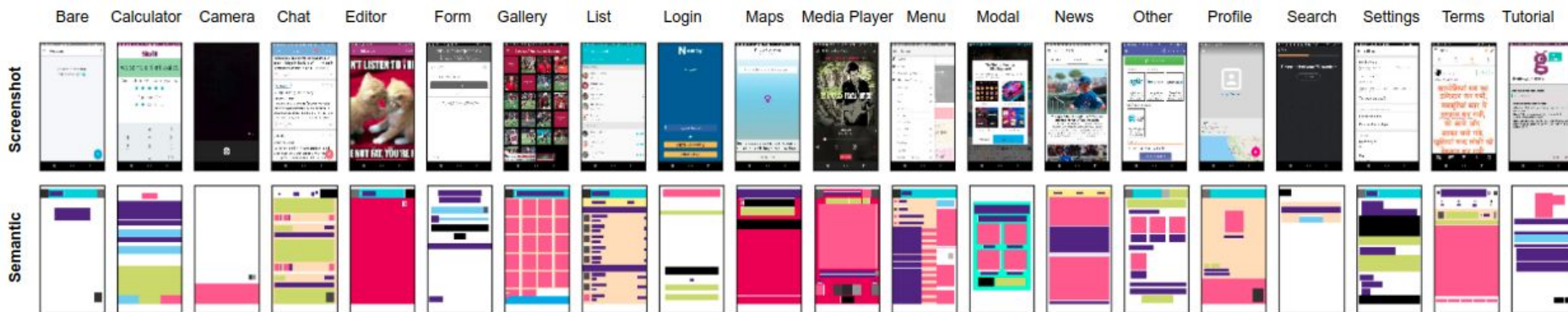


- Time required to move a pointer (e.g., mouse cursor) to a target area is a function of the **distance to the target** divided by the **size of the target**.



Exercise 3: Clustering of UI wireframes

- Layout design categorization contributes to a better understanding of the UI functionality.



- In this exercise, you will develop a deep learning model that can find a good internal representation of UI layouts (**layout embeddings**) that can be used for **downstream tasks**.
 - Clustering (i.e. finding and grouping together similar UIs.)
- You will also learn how to visualize the clustering results through non-linear dimensionality reduction of UI embeddings.