LINKSFOUNDATION.COM

FONDAZIONE links
PASSION FOR INNOVATION

Applied Data Science Project

L11 - Project development tools I

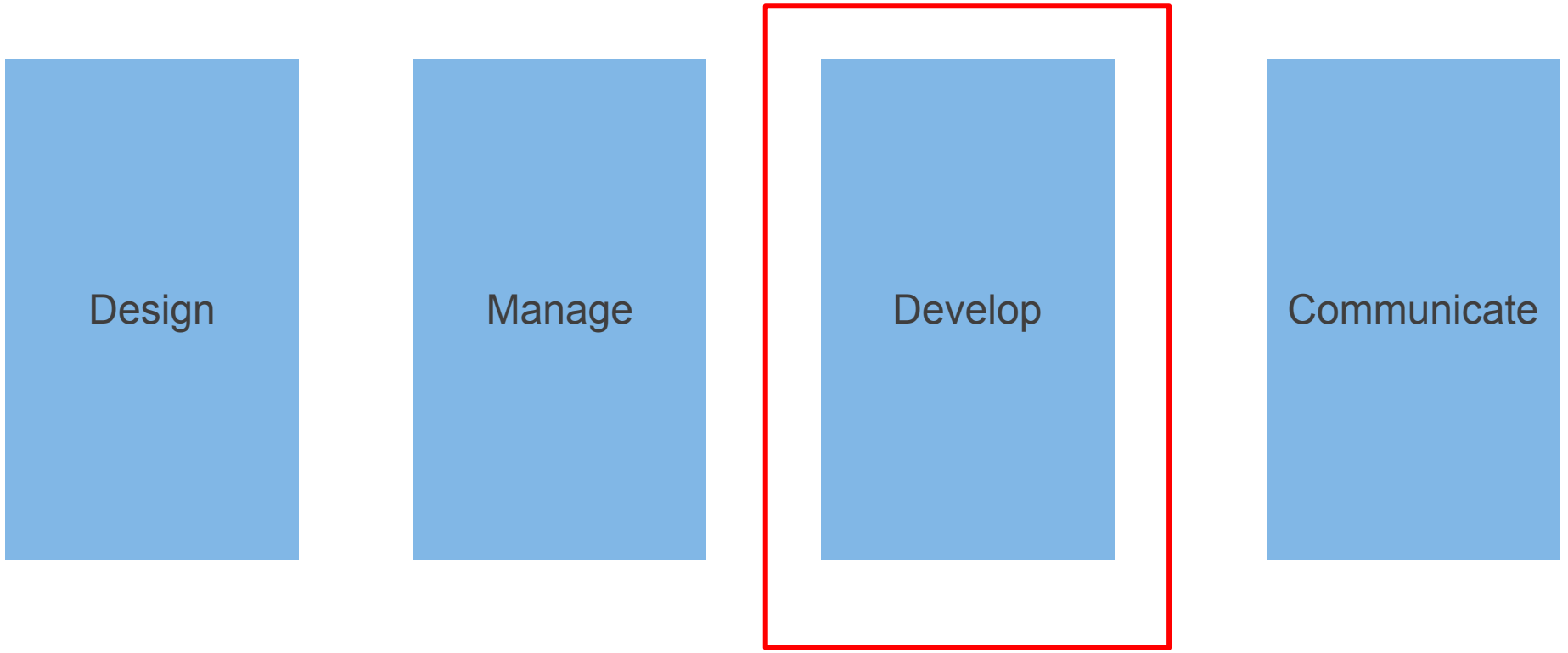
Politecnico di Torino

ellis
European Laboratory for Learning and Intelligent Systems

Giuseppe Rizzo

Turin, October 7, 2021

Pillars



Development

It is where the magic happens

A machine intelligence is created to generate the outputs that meet objectives and requirements with the involvement of a team tasked on activities with due dates

Objectives and requirements have been defined in the Design pillar

Activities and due dates in the Manage pillar

Knowledge tools

- agile and scrum

- collaborative workspaces
  - program development
  - repository
  - communication among project developers

Knowledge tools

- agile and scrum

Lecture 8 & 10

- collaborative workspaces
  - program development
  - repository
  - communication among project developers

Apps

- agile and scrum

- collaborative workspaces
  - program development — Colaboratory
  - version control — Colaboratory & Github
  - communication among project developers — Slack

Program development

- “Divide and rule”
- Tasks are mapped into modules
- One module has one lead developer and, eventually, contributors
- Choose the programming language according to:
  - ecosystem of software modules that you can utilize
  - easiness of model integration
  - familiarity. Do not be afraid to switch to another (similar programming language) since they share most of the features and development patterns

Python is considered the default language for developing machine intelligence nowadays

Colaboratory

https://colab.research.google.com

An application where to develop, share, and also test on dedicated hardware (GPU to speed up the computing)

Overview

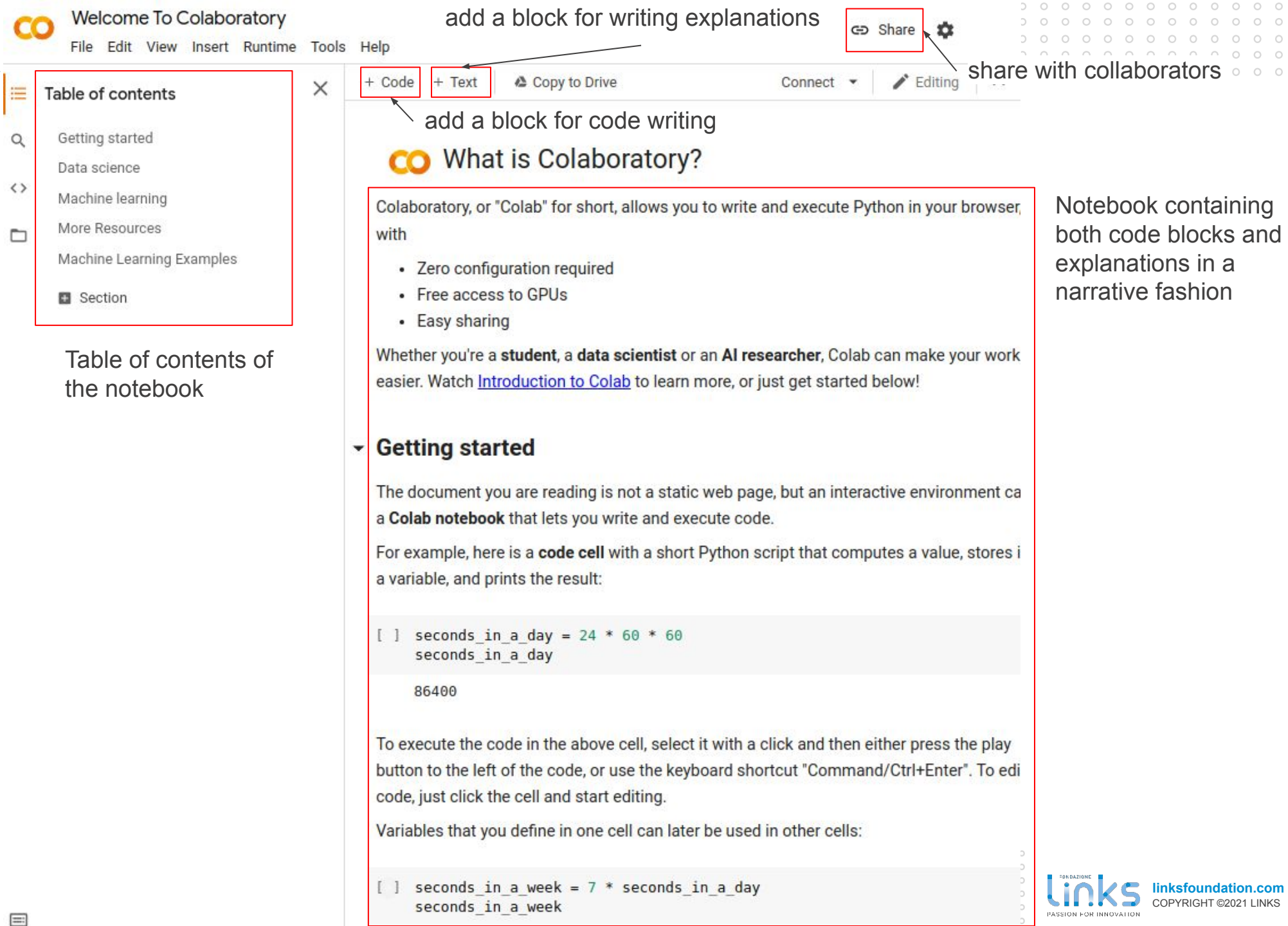
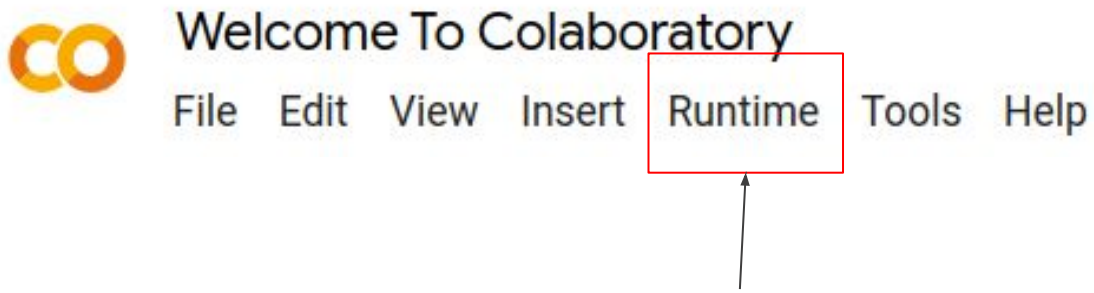


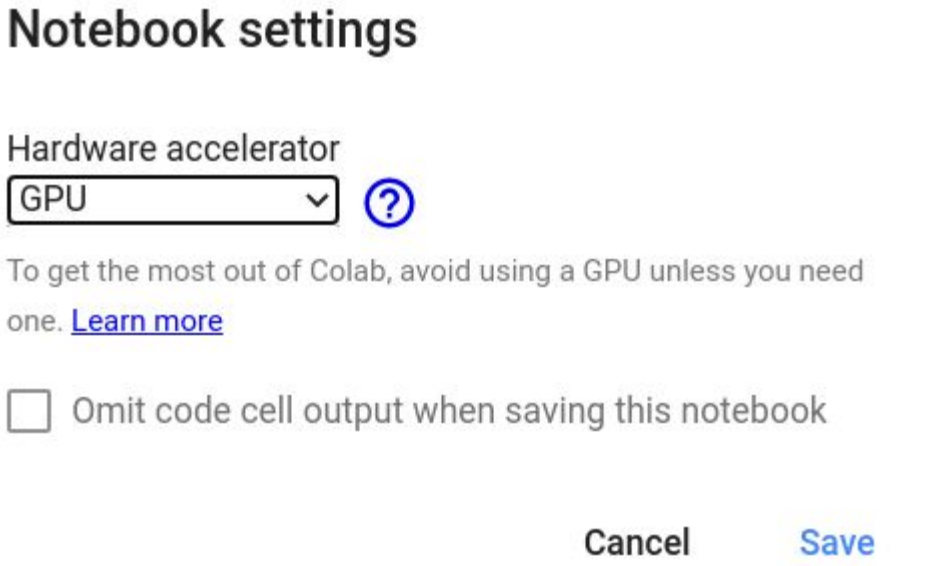
Table of contents of the notebook

Notebook containing both code blocks and explanations in a narrative fashion

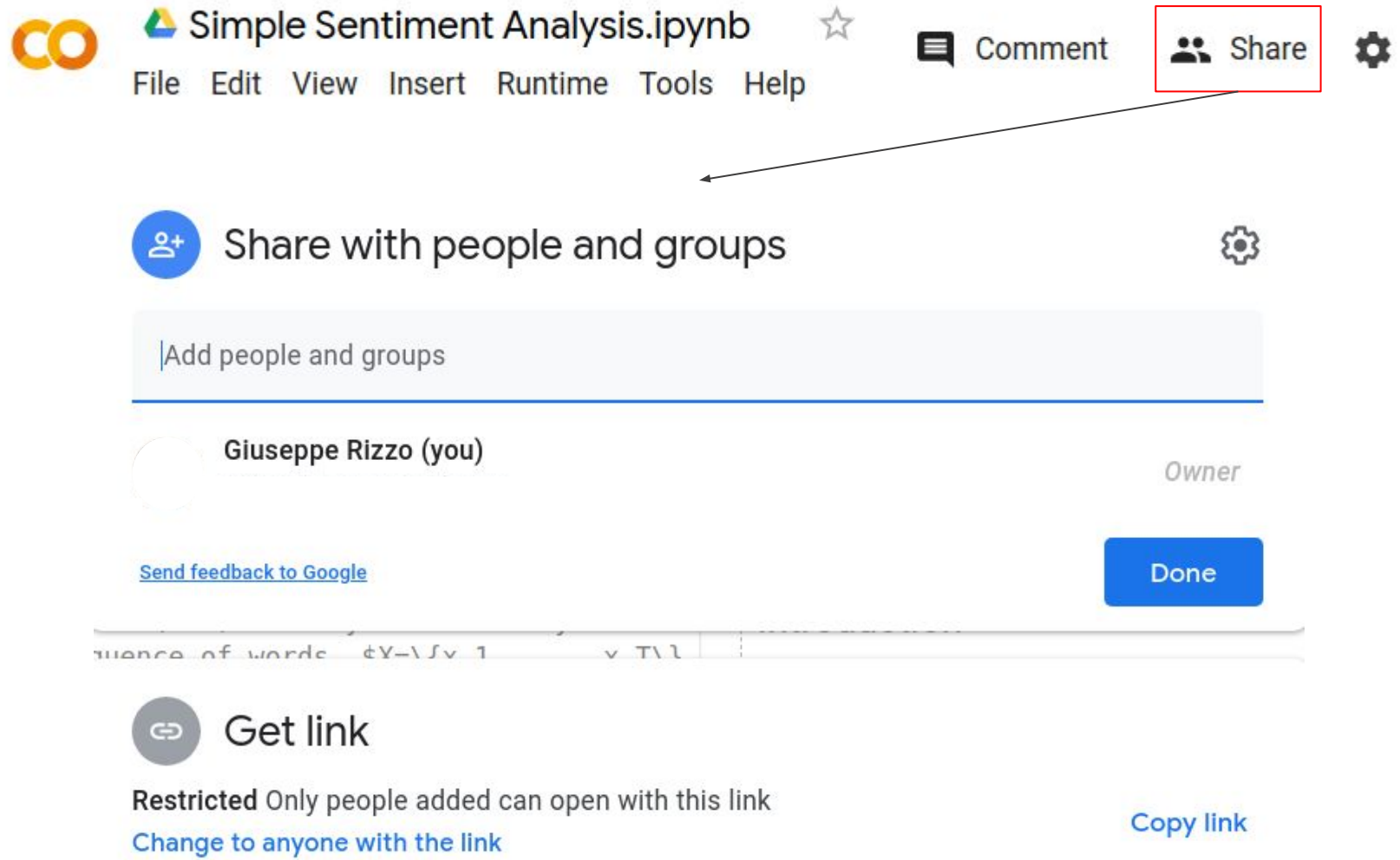
Menu bar



define the runtime configurations, among those the hardware acceleration if needed



Share



Connect your Google Drive

Enable authorization to import data from Google Drive

```
1  from google.colab import drive
2  drive.mount('/content/drive/')
```

List the files in your drive

```
1  !ls "/content/drive/My Drive/"
```

Running with Google Colab

Upload YOUR_PYTHON_FILE.py to Google Drive & Run with Google Colab

```
1  !python3 "/content/drive/My Drive/Colab Notebooks/YOUR_PYTHON_FILE.py"
```

Run with Google Colab to Download YOUR_PYTHON_FILE.py from Google Drive

```
1  from google.colab import files

2  files.download('/content/drive/My Drive/Colab Notebooks/YOUR_PYTHON_FILE.py')
```

Bash commands

Bash commands are executed with the environment “!”

Download an external file

```
1    !wget http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz -P "/content/drive/My Drive/Colab Notebooks"
```

Clone a repository

```
1    !git clone https://github.com/pytorch/examples.git
```

Colab == virtual environment

The environment can be customized with the addition of python packages

Install

```
1    !pip install torchtext
```

Show a version

```
1    !pip show torchtext
```

Example

Simple Sentiment Analysis.ipynb

Simple Sentiment Analysis

In this series we'll be building a machine learning model to detect sentiment (i.e. detect if a sentence is positive or negative) using PyTorch and TorchText. This will be done on movie reviews, using the IMDb dataset.

In this first notebook, we'll start very simple to understand the general concepts whilst not really caring about good results. Further notebooks will build on this knowledge and we'll actually get good results.

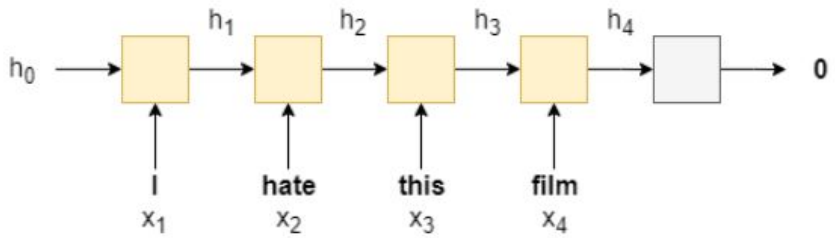
Introduction

We'll be using a **recurrent neural network** (RNN) as they are commonly used in analysing sequences. An RNN takes in sequence of words, $X = \{x_1, \ldots, x_T\}$, one at a time, and produces a *hidden state*, h, for each word. We use the RNN *recurrently* by feeding in the current word x_t as well as the hidden state from the previous word, h_{t-1}, to produce the next hidden state, h_t.

$$h_t = \text{RNN}(x_t, h_{t-1})$$

Once we have our final hidden state, h_T, (from feeding in the last word in the sequence, x_T) we feed it through a linear layer, f, (also known as a fully connected layer), to receive our predicted sentiment, $\hat{y} = f(h_T)$.

Below shows an example sentence, with the RNN predicting zero, which indicates a negative sentiment. The RNN is shown in orange and the linear layer shown in silver. Note that we use the same RNN for every word, i.e. it has the same parameters. The initial hidden state, h_0, is a tensor initialized to all zeros.



Note: some layers and steps have been omitted from the diagram, but these will be explained later.

FONDAZIONE links

PASSION FOR INNOVATION

Thank you for your attention.

Questions?

FONDAZIONE links

PASSION FOR INNOVATION

CONTACTS

Giuseppe Rizzo

Team Leader

p. +39 011 2276244

e. giuseppe.rizzo@linksfoundation.com

FONDAZIONE LINKS

Via Pier Carlo Boggio 61 | 10138 Torino

P. +39 011 22 76 150

LINKSFOUNDATION.COM