



# **Adventures in Data Science**

**Exploring the tools and tricks of Deep Learning  
(DL) and Natural Language Processing (NLP)**

*Salim Hameed*





# **What is this all about ?**

**Organizations have huge volume of business and other data available through internal and external sources.**

**It should be possible to apply modern Deep Learning techniques to this data so that we can have valuable, practical, useful insights into processes, the way operations are done internally/externally and interactions with customers.**



# Motivation

- **Personal interest**
- **Learn more about DL**
- **Learn more about NLP**
- **Refresh Python skills**
- **One of my goals !**
- **A cool thing to do :)**



# Relevant Definitions

- **Deep Learning, broadly speaking, is group of machine learning techniques where artificial neural networks are used for the learning process**
- **Natural Language Processing, loosely speaking, are techniques used for teaching computers to process and learn from language data**



Osprey ?





# What is a Deep Learning Model and how to create one ?

DL models are functions that connect identified features  $X$  of a system with the outputs from the system  $Y$  using a set of parameters  $w$  .

ie.  $f(X,w) = Y$

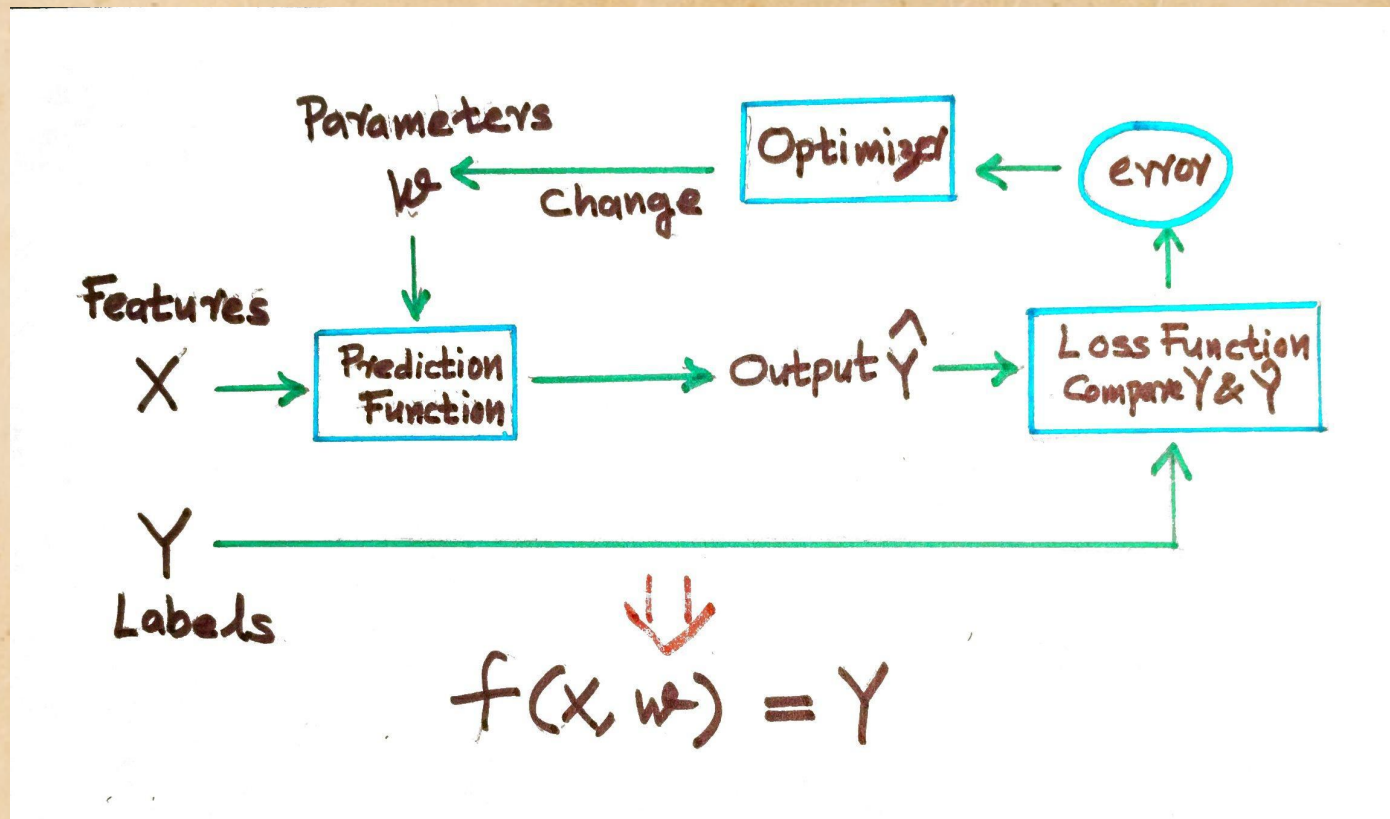
For example  $X$  can be a set of features that can identify sentences that form news headlines. These can be how frequently some words are used, how words are grouped together etc.

$Y$  can be an output like – does the sentence represent a positive sentiment or a negative sentiment



# The process of creating a DL Model – how does it work ?

Repeatedly optimize values of  $w$  so that error is minimized:





# How do simple NLP sentiment classifiers work ?

Start from a list of sentences with known positive or negative sentiments. Compile a list of words eliminating duplicates, punctuation marks and words that do not contribute significantly to the meaning. The model can be trained and tested using subsets from this collection.

Features  $X$  will be the frequencies of words with positive or negative sentiments. Connect this word frequency collection to the corresponding sentiments  $Y$ .






**What I did:**

# **Collecting and analyzing data ...**

**The target was to collect, cleanup, analyze, extract information, model and visualize real life data.**

**From here on we go through how this was done. Tools and techniques used here were selected based on my own prior experience. They may not be the ideal selection under a different set of conditions.**






# **In practice - the process**

- **Collect data – cleanup – modify – process – generate secondary data if needed**
- **Create the model – code – split the available data to train and test sets OR use an already available trained model and start from there**
- **Train the model using the training data OR start from an available trained model and further refine the model using the training data**
- **Test new data using the trained model – get the results – test using test data – find the accuracy of the trained model**
- **Visualize and analyze the results**



# Collect Data



- **Decided to use publicly available data**
  - **XML RSS news feeds from publicly available news sources**
  - **The news feeds were collected converted to HTML, timestamped and saved 24x7 for a month (between the end of June 2020 and the end of July 2020) – approximately 50,000 distinct news headlines and associated data**
  - **The required data was extracted – news headlines**
  - **The headlines were further analyzed to extract place names, names of people and organizations**
  - **Geolocation was done using extracted place names**
- 



# Collect Data - Tools


- **Data collection infrastructure and XML RSS to HTML – bash and Perl scripts run using Linux cron jobs**
- **Extraction of data from HTML – Python scripts using BeautifulSoup library – cleanup using Python regex libraries**
- **Entity detection from news headlines – Python scripts using spaCy library for NLP**
- **Geolocation of detected place names – Python scripts using Nominatim library using OpenStreetMap**

**Data inside the processing scripts were in PadasDataframe format using Pandas library. Data was stored on disks in JSON format**



# Models and Training



- **Commonly used operations required for deep learning models are available in popular python libraries. There are well tested, supported and reliable. Using these libraries sophisticated deep learning models can be created in Python using very little code.**
  - **There are large collections of pre-classified data available as a part of various popular Python deep learning libraries and other public sources on the web.**
  - **Using the above two, models can be created and trained for specific applications.**
- 



# Models and Training - Tools

- The target was to create a model that can categorize (classify) news headlines as positive or negative sentiment.
- Logistic Regression classifier - those who practice deep learning often use this as a baseline technique to verify the accuracy of more sophisticated methods. Simple to implement from scratch.
- NLTK library in Python for NLP, NumPy library in Python for mathematical operations, pandas library in Python for data manipulation internally



# Visualize Data - Tools

- **Data obtained by processing around 50,000 news headlines during the previous two stages was collected and stored in JSON format on disk.**
- **Visualization was done using Streamlit application framework for Python. It is fast and easy to create attractive data visualizations using Streamlit. It has built in support for many existing data visualization frameworks like dec.gl .**



**End result ...**

**Data visualization web application demo ...**




# **Web Application - cases**



## **Example:**

**For large and complex web applications with many concurrent users, models can be trained on application user behavior (using data extracted from logs and other sources).**

**The trained model can be used to predict user behavior to avoid or delay costly process operations or calls to the backend.**







**The end**

