DEEP LEARNING:
Introduction and Applications

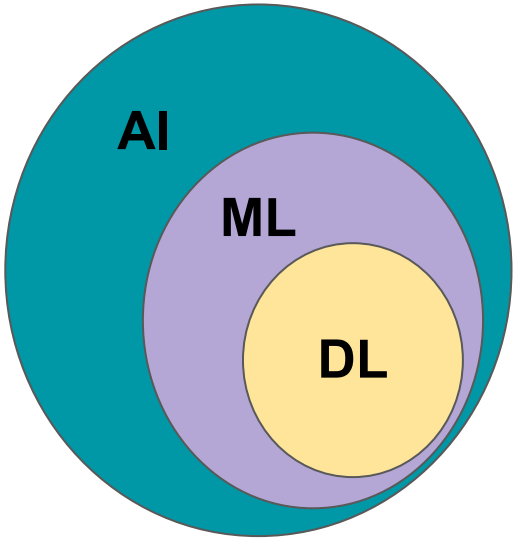
Gedeon M

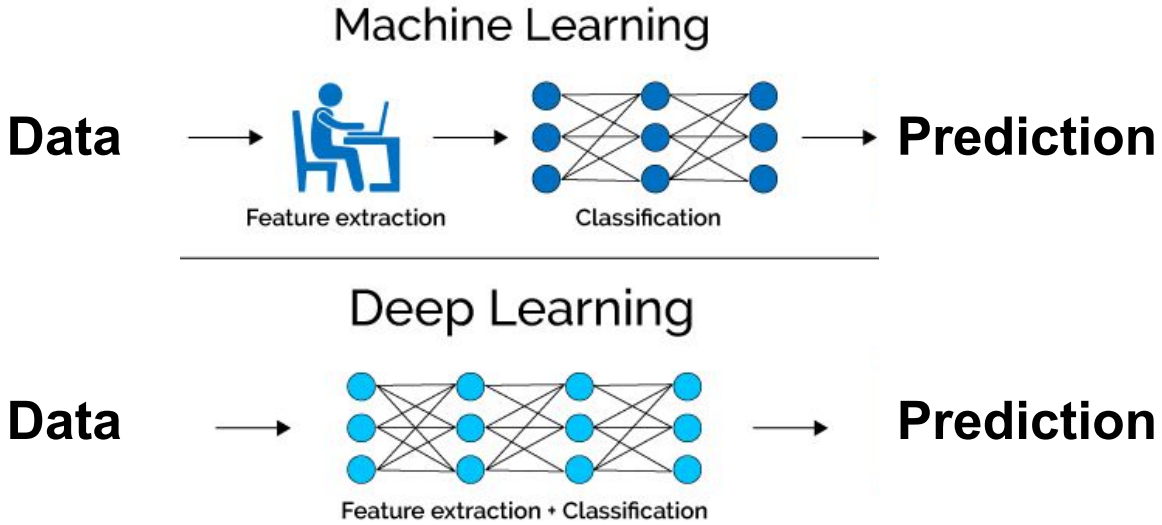
Nyandwi JD

Day 1 Agenda

2:30 - 3:30	**Presentation**
3:30 - 3:50	**Questions**
3:50 - 4:00	**Break**
4:00 - 4:30	**Practice**
4:30 - 5:00	**Questions**

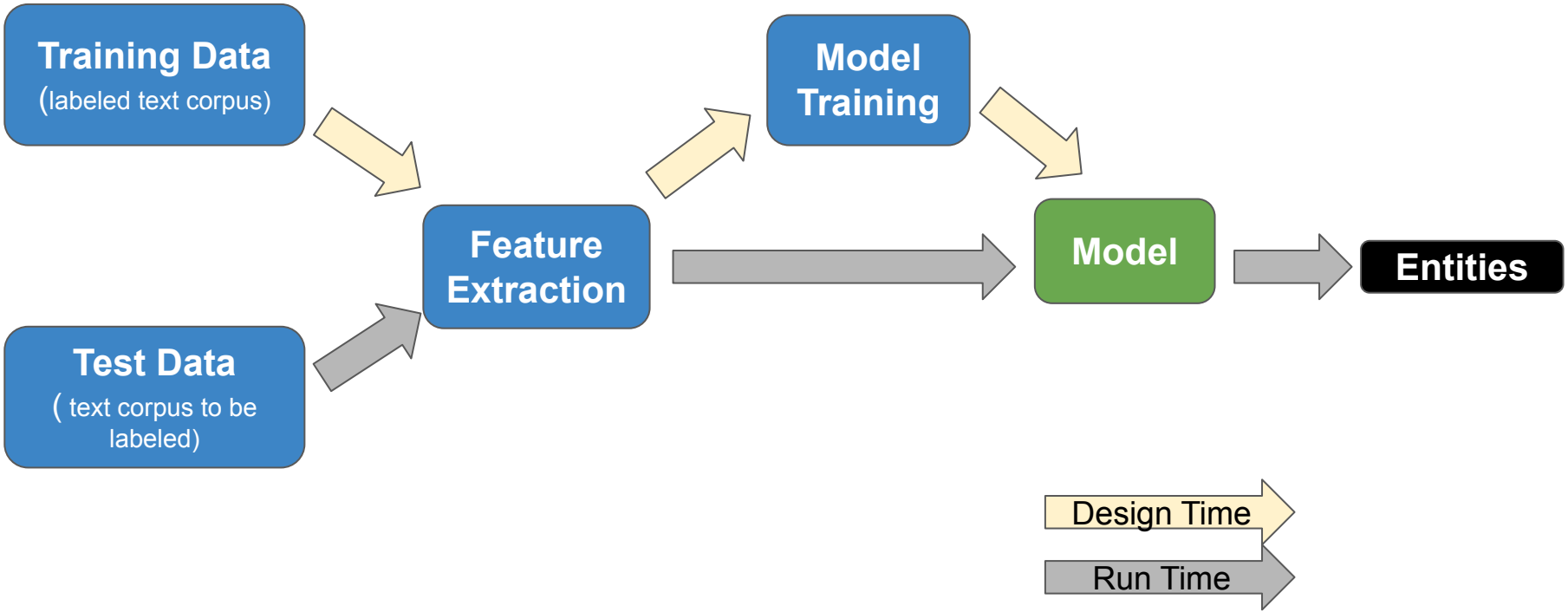
Introduction



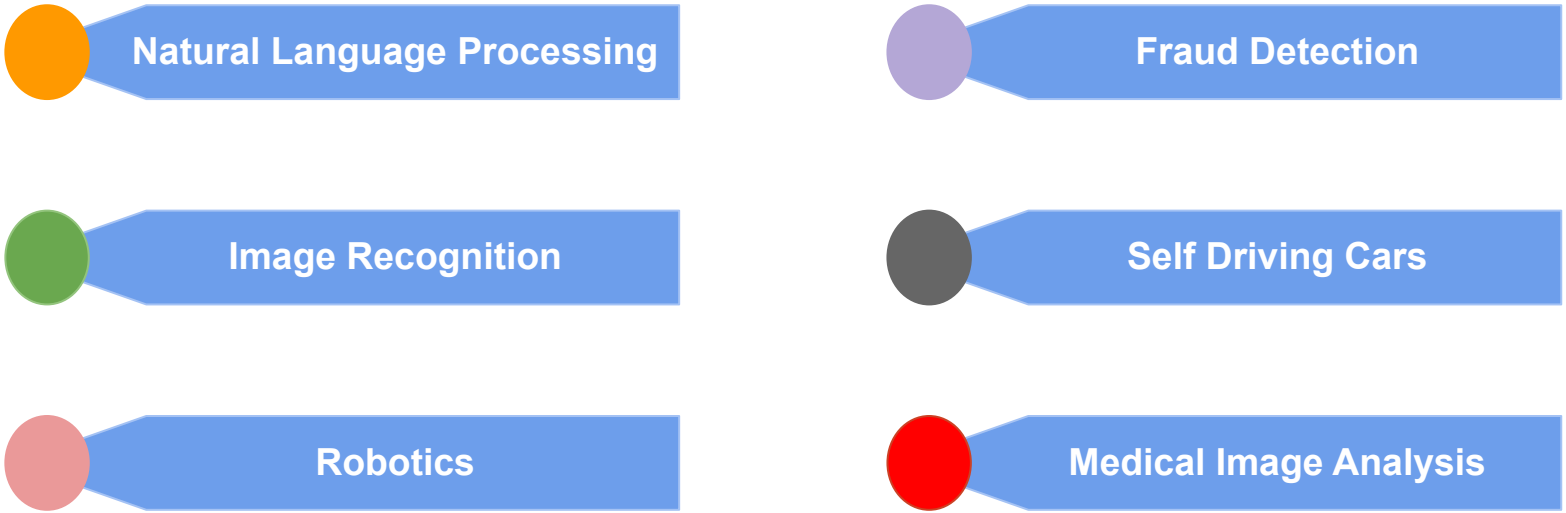


Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning.

Learning from the data



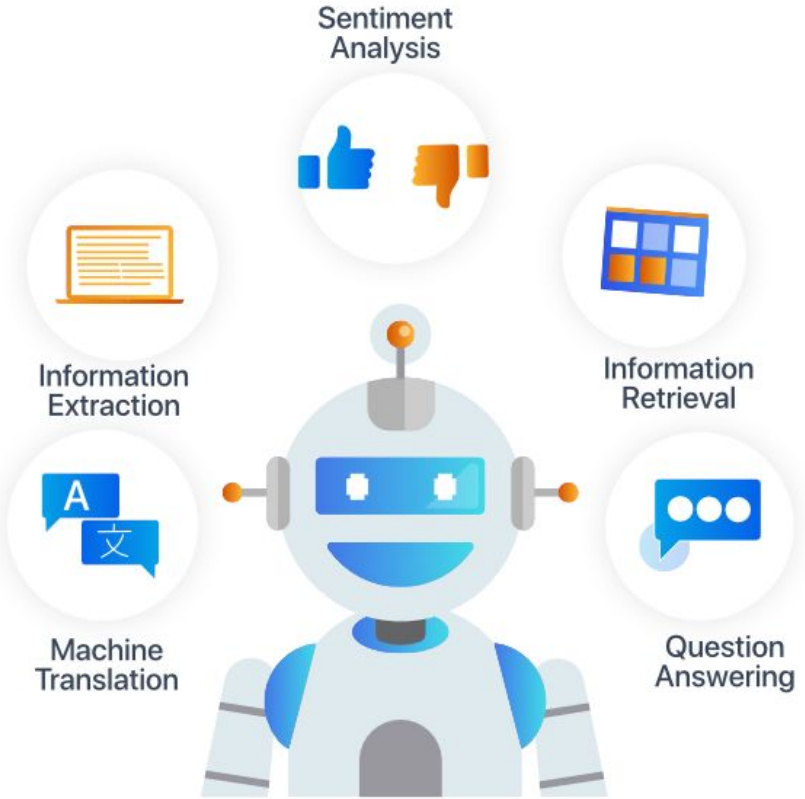
Applications



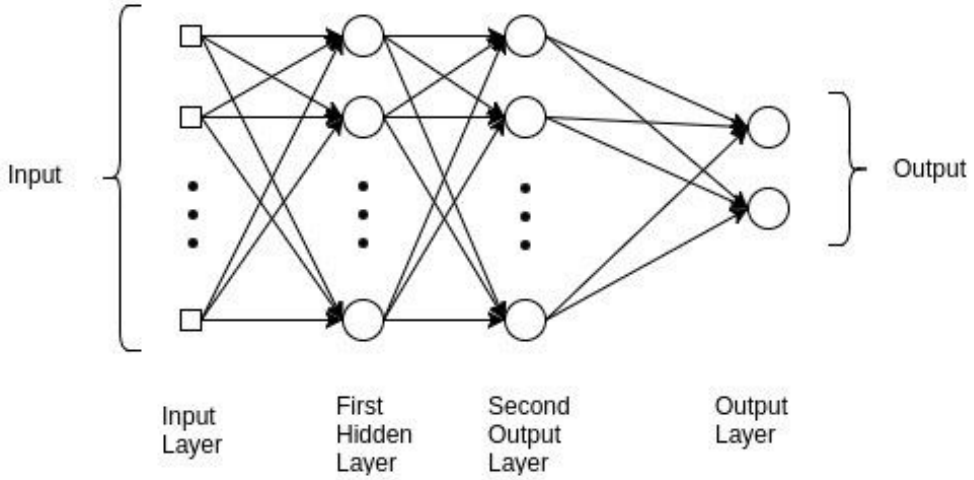
Applications of

Natural Language Processing

in Different Domains



Basics of Training Neural Networks: Multi-layer Perceptron



$$f(\boldsymbol{x}) = f^{(3)}\left(f^{(2)}\left(f^{(1)}(\boldsymbol{x})\right)\right)$$

MLP are typically represented by composing together many different functions.

Basics of Training Neural Networks: The Maths behind

Forward pass and back-prop

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial X_2}\frac{\partial X_2}{\partial W_2}$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial X_2}\frac{\partial X_2}{\partial X_1}\frac{\partial X_1}{\partial W_1}$$

Parameters update $W \leftarrow W - \alpha * \nabla_w L$

Example

Assume Mean squared error loss $L(X_2, Y) = ||X2 - Y||^2$

$\frac{\partial L}{\partial X_2} = 2(X_2 - Y)$

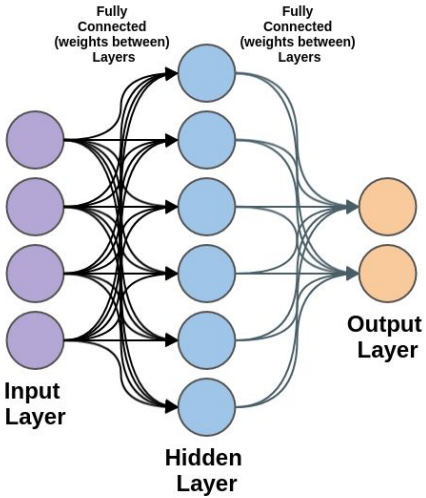
$\frac{\partial X_2}{\partial W_2} = X_1$

$\frac{\partial X_2}{\partial X_1} = W_2$

$\frac{\partial X_1}{\partial W_1} = X$

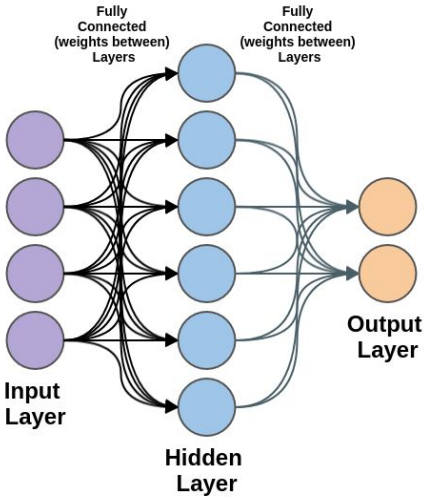
$\nabla_{w_2} L = 2(X_2 - Y)X_1$ and $\nabla_{w_1} L = 2(X_2 - Y)W_2 X$

Basics of Training Neural Networks: Tunable Parameters



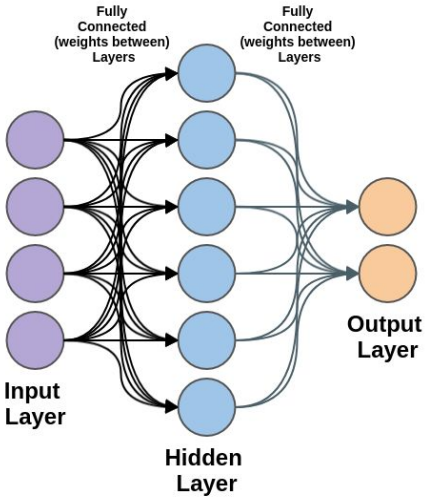
- **Epoch** – one forward pass and one backward pass of all the training examples

Basics of Training Neural Networks: Tunable Parameters



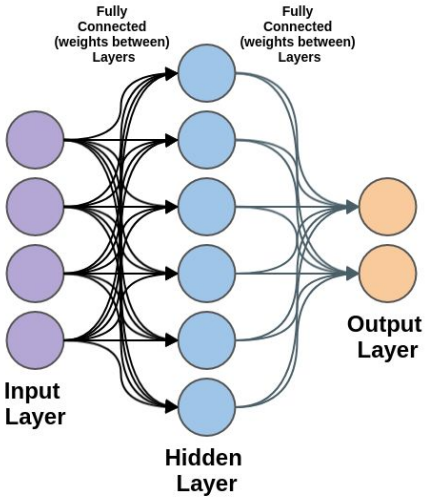
- **Epoch** – one forward pass and one backward pass of all the training examples
- **Batch Size** – The number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.

Basics of Training Neural Networks: Tunable Parameters



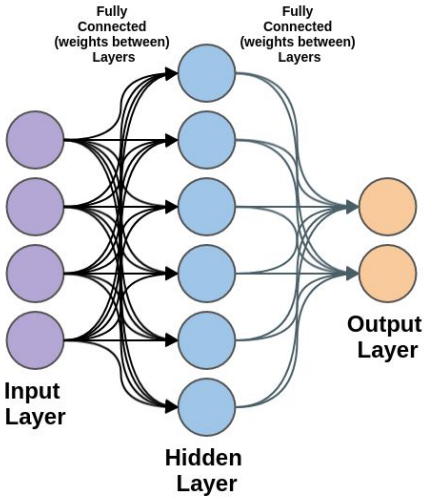
- **Epoch** – one forward pass and one backward pass of all the training examples
- **Batch Size** – The number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- **Iterations** – number of passes, each pass using [batch size] number of examples (one forward pass + one backward pass)

Basics of Training Neural Networks: Tunable Parameters



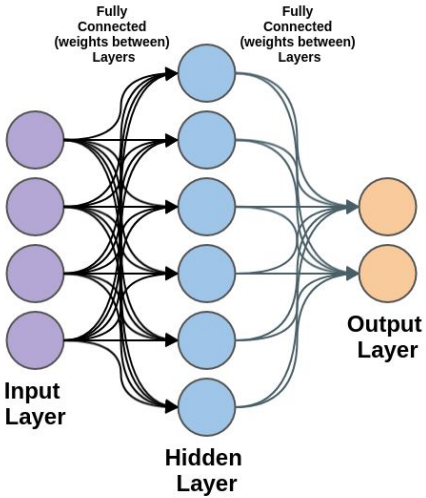
- **Epoch** – one forward pass and one backward pass of all the training examples
- **Batch Size** – The number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- **Iterations** – number of passes, each pass using [batch size] number of examples (one forward pass + one backward pass)
- **Learning rate** – Determines the step size at each iteration while moving toward a minimum of a loss function

Basics of Training Neural Networks: Tunable Parameters



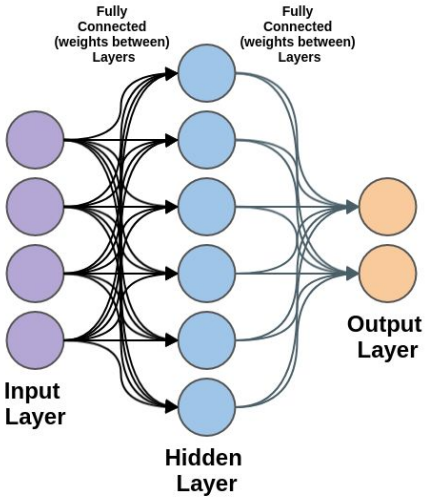
- **Epoch** – one forward pass and one backward pass of all the training examples
- **Batch Size** – The number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- **Iterations** – number of passes, each pass using [batch size] number of examples (one forward pass + one backward pass)
- **Learning rate** – Determines the step size at each iteration while moving toward a minimum of a loss function
- **Optimizers** – A function or an algorithm that modifies the attributes of the neural network, such as weights and learning rate.

Basics of Training Neural Networks: Tunable Parameters



- **Epoch** – one forward pass and one backward pass of all the training examples
- **Batch Size** – The number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- **Iterations** – number of passes, each pass using [batch size] number of examples (one forward pass + one backward pass)
- **Learning rate** – Determines the step size at each iteration while moving toward a minimum of a loss function
- **Optimizers** – A function or an algorithm that modifies the attributes of the neural network, such as weights and learning rate.
- **Model size** – Layers and their size

Basics of Training Neural Networks: Tunable Parameters

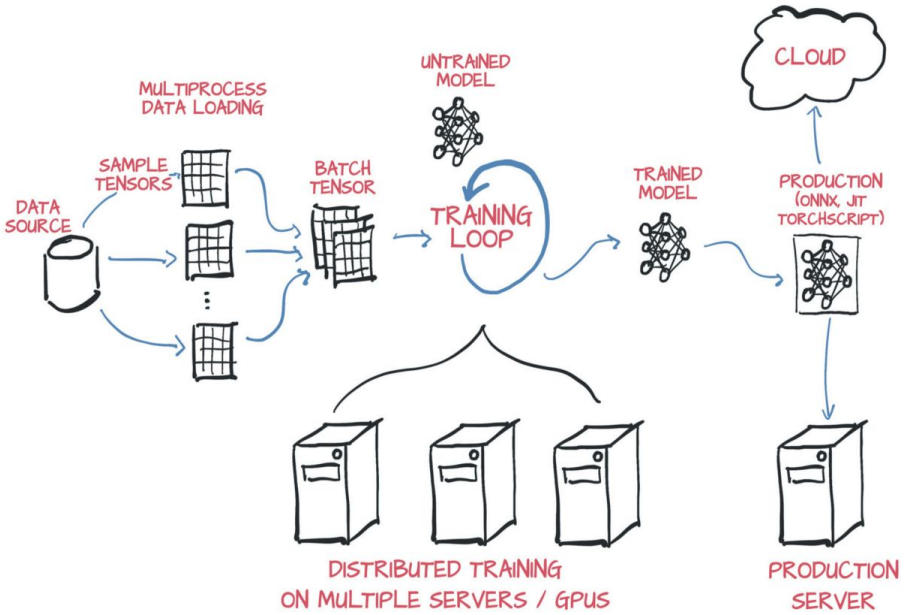


- **Epoch** – one forward pass and one backward pass of all the training examples
- **Batch Size** – The number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- **Iterations** – number of passes, each pass using [batch size] number of examples (one forward pass + one backward pass)
- **Learning rate** – Determines the step size at each iteration while moving toward a minimum of a loss function
- **Optimizers** – A function or an algorithm that modifies the attributes of the neural network, such as weights and learning rate.
- **Model size** – Layers and their size

⚠ **Hyperparameters** are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning.

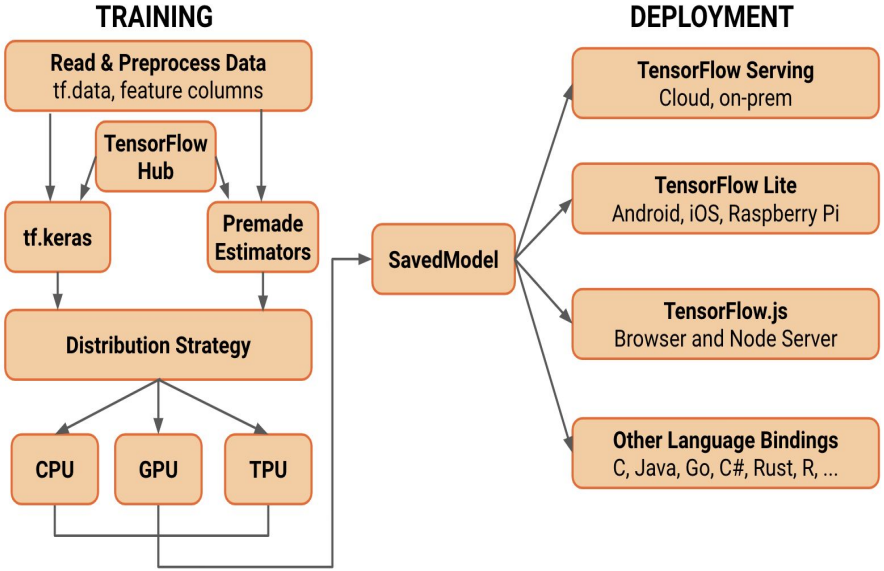
Deep Learning: Common Frameworks

PyTorch



Source: Medium

TensorFlow

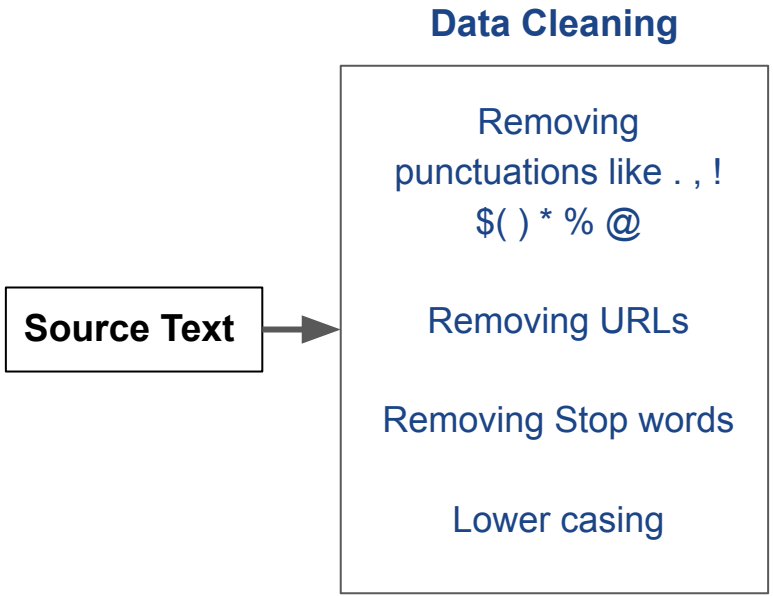


Source: TF blog

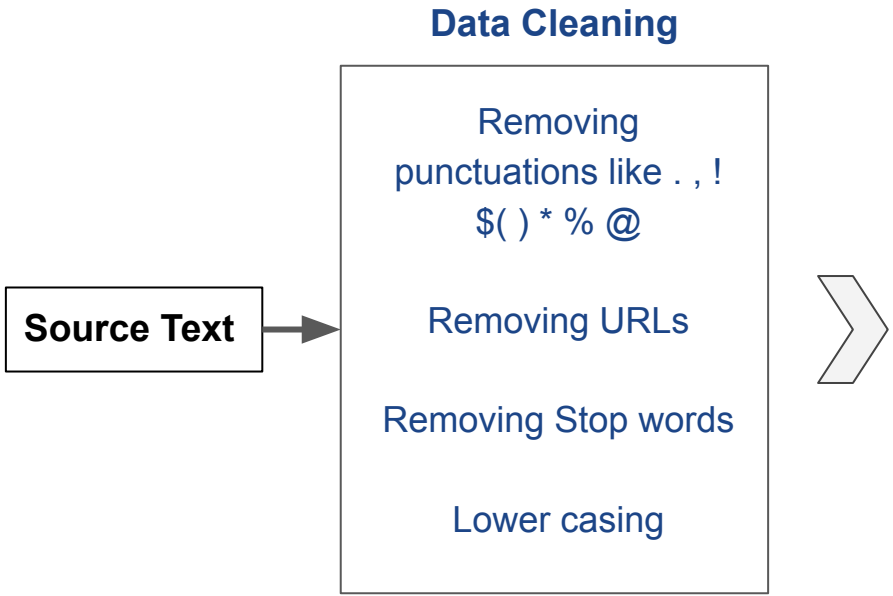
NLP: Data and preprocessing

NLP: Data and preprocessing

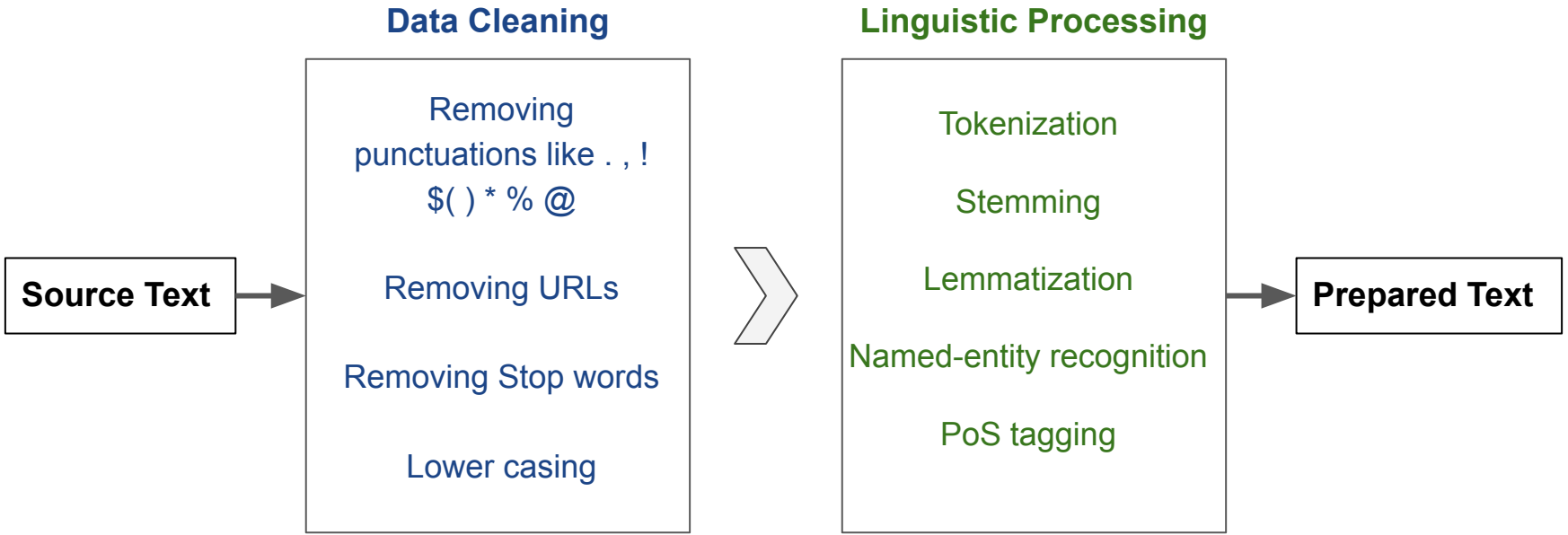
NLP: Data and preprocessing



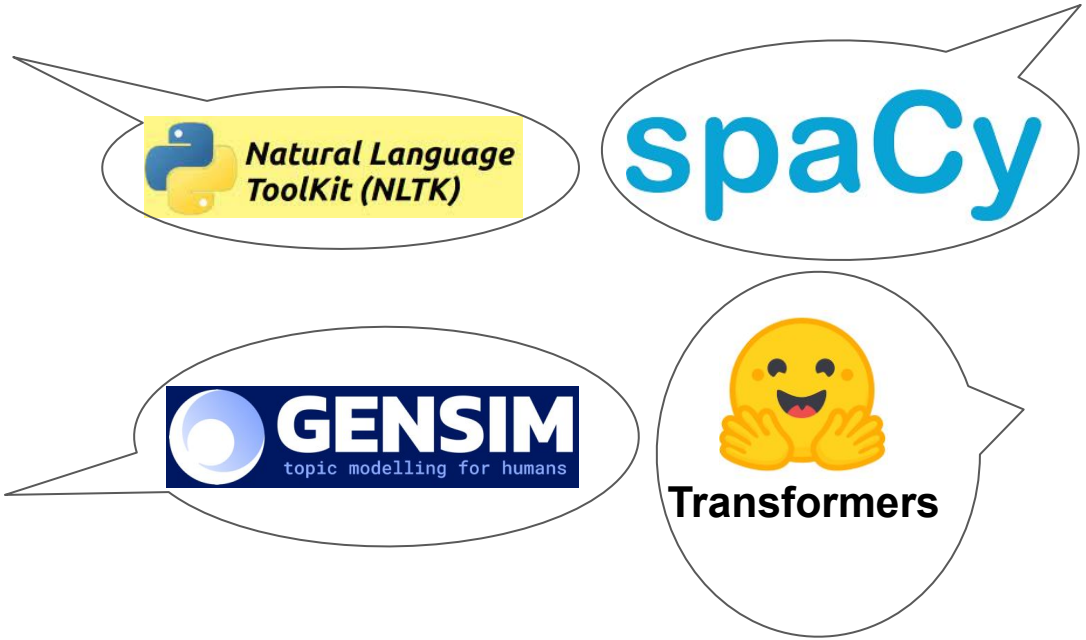
NLP: Data and preprocessing



NLP: Data and preprocessing



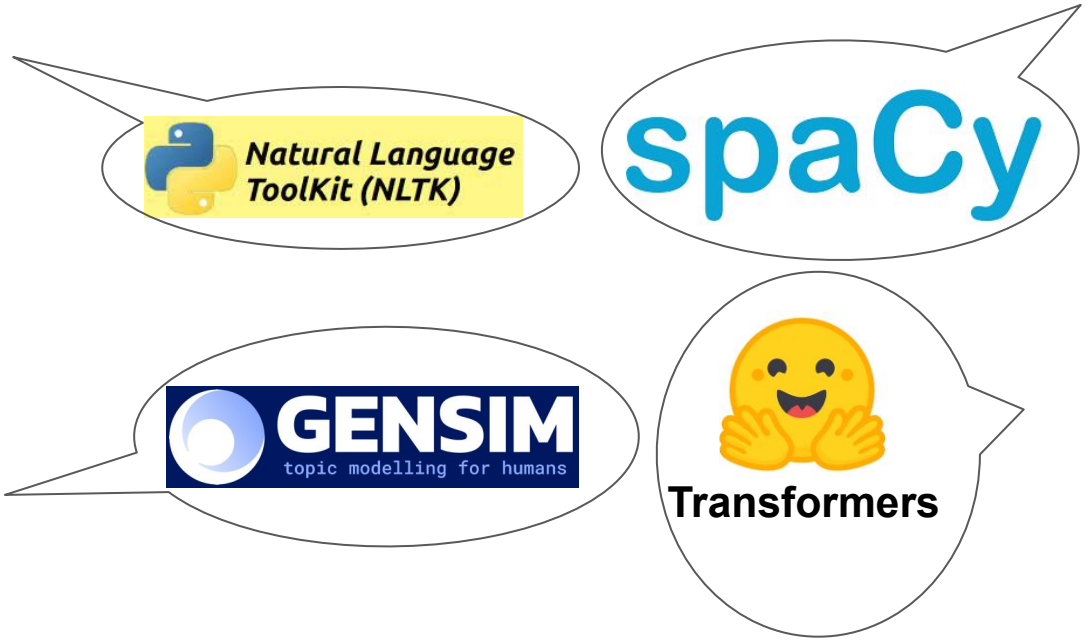
Common Packages: Linguistic Processing, NLP and Deep NLP



Common Packages: Linguistic Processing, NLP and Deep NLP

Corpus to keywords

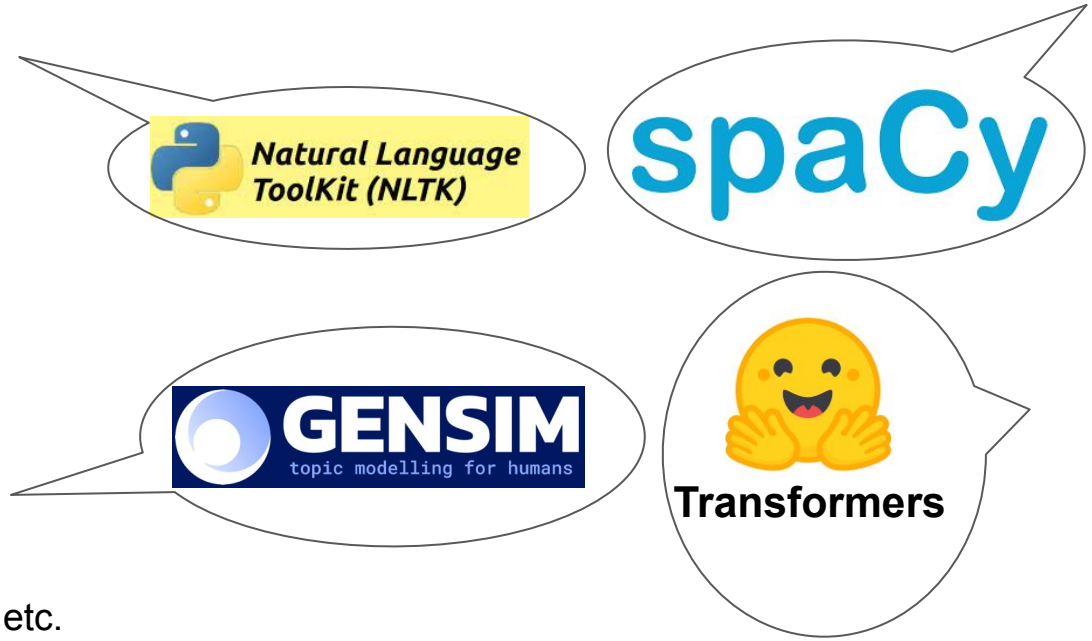
- Tokenize
- Remove stop words
- Lemmatise



Common Packages: Linguistic Processing, NLP and Deep NLP

Corpus to keywords

- Tokenize
- Remove stop words
- Lemmatise



Topic modeling

- LDA
- Similarity matrix, etc.
- Bigrams and trigrams
- Topic modeling

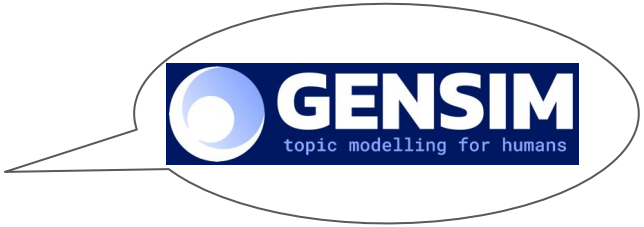
Common Packages: Linguistic Processing, NLP and Deep NLP

Corpus to keywords

- Tokenize
- Remove stop words
- Lemmatise

Linguistics features

- Rephrase sentence structure; Provides all info for each word in the paragraph(pos, stopword, lemma, named entity)
- Semantic search



Topic modeling

- LDA
- Similarity matrix, etc.
- Bigrams and trigrams
- Topic modeling

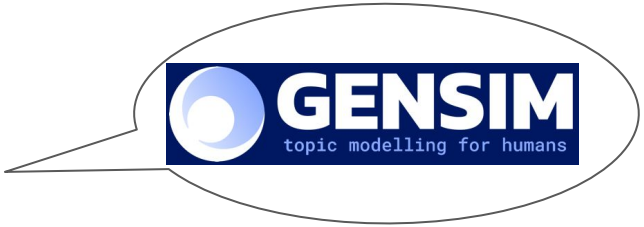
Common Packages: Linguistic Processing, NLP and Deep NLP

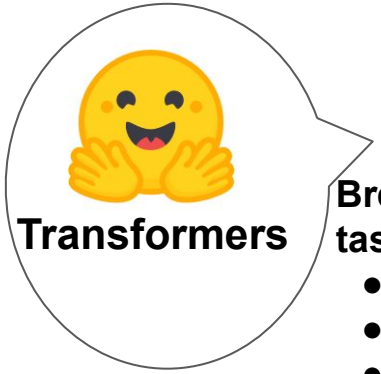
Corpus to keywords

- Tokenize
- Remove stop words
- Lemmatise

Linguistics features

- Rephrase sentence structure; Provides all info for each word in the paragraph(pos, stopword, lemma, named entity)
- Semantic search





Topic modeling

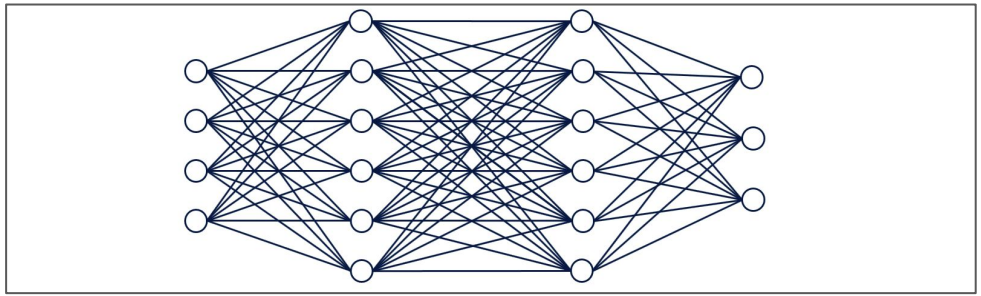
- LDA
- Similarity matrix, etc.
- Bigrams and trigrams
- Topic modeling

Breakthrough for any NLP tasks

- Question Answering
- Summarization
- Sentiment Analysis
- Document Classification
- Machine Translation

Recap

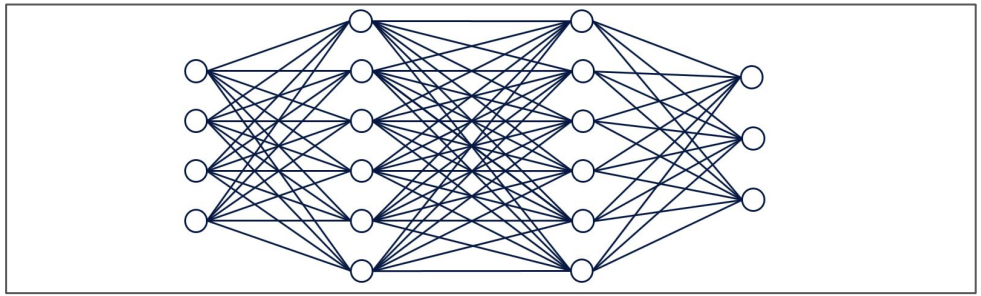
1



Deep learning models are trained by using large sets of data and neural network architectures that learn features directly from the data. Models get trained by back-propagating the gradient of the loss.

Recap

1



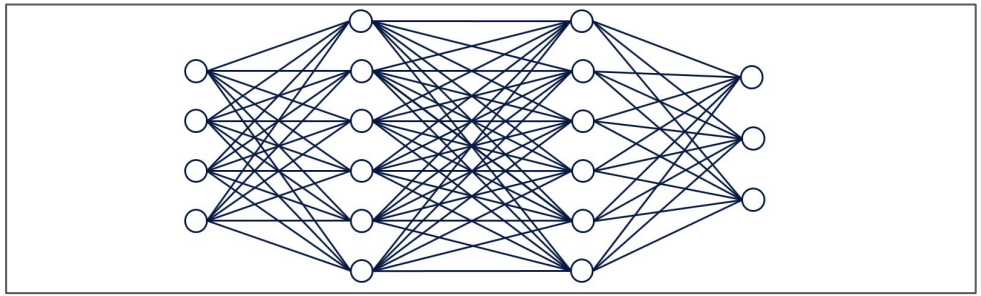
Deep learning models are trained by using large sets of data and neural network architectures that learn features directly from the data. Models get trained by back-propagating the gradient of the loss.

2

PyTorch and TensorFlow are the most popular deep learning frameworks

Recap

1



Deep learning models are trained by using large sets of data and neural network architectures that learn features directly from the data. Models get trained by back-propagating the gradient of the loss.

2

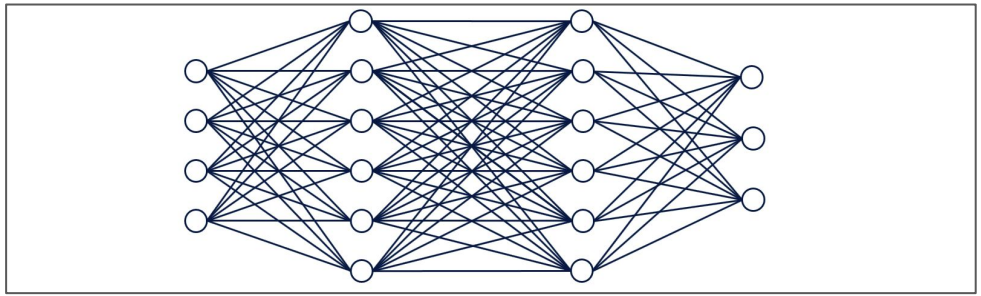
PyTorch and TensorFlow are the most popular deep learning frameworks

3

Hugging faces transformers can help to accomplish common NLP tasks/applications (chatbot, text summarization, machine translation, etc.)

Recap

1



Deep learning models are trained by using large sets of data and neural network architectures that learn features directly from the data. Models get trained by back-propagating the gradient of the loss.

2

PyTorch and TensorFlow are the most popular deep learning frameworks

3

Hugging faces transformers can help to accomplish common NLP tasks/applications (chatbot, text summarization, machine translation, etc.)

4

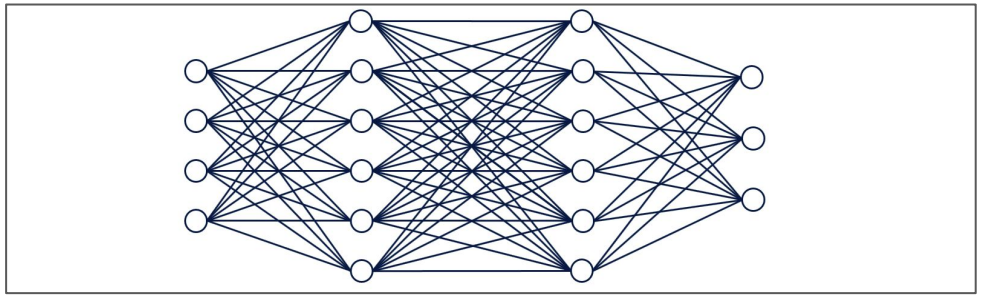
Syntax vs Semantics:

Syntax is the sentence structure

Semantics is the meaning of text, (fundamental take away after reading the text)

Recap

1



Deep learning models are trained by using large sets of data and neural network architectures that learn features directly from the data. Models get trained by back-propagating the gradient of the loss.

2

PyTorch and TensorFlow are the most popular deep learning frameworks

3

Hugging faces transformers can help to accomplish common NLP tasks/applications (chatbot, text summarization, machine translation, etc.)

4

Syntax vs Semantics:

Syntax is the sentence structure

Semantics is the meaning of text, (fundamental take away after reading the text)

https://bit.ly/3SLRsMS

The End!

— THANK YOU —