Deep Learning for NLP

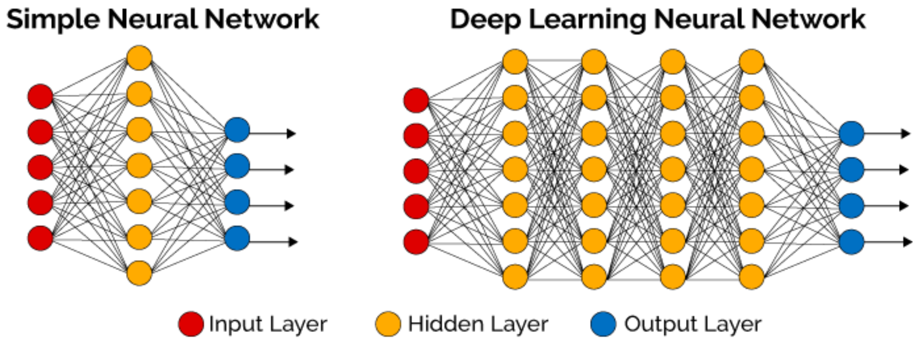
Analytics Vidhya
Learn everything about analytics

About the Module

- ❏ What is Deep Learning
- ❏ Why Exploring Deep Learning
- ❏ Deep Learning Applications in NLP
- ❏ Text Classification using Deep Learning

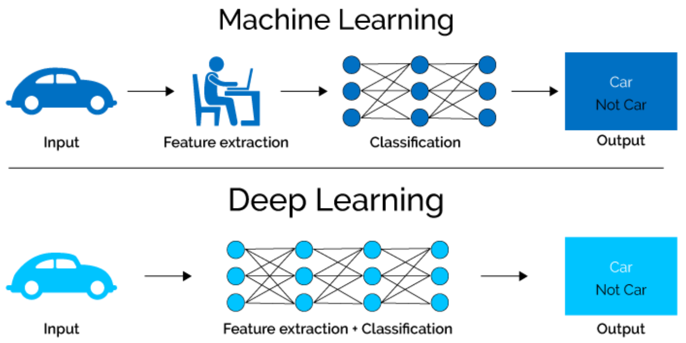
What is Deep Learning

Deep Learning Models
- Subfield of machine learning
- Neural network family algorithms
- Many complex layers stacked together



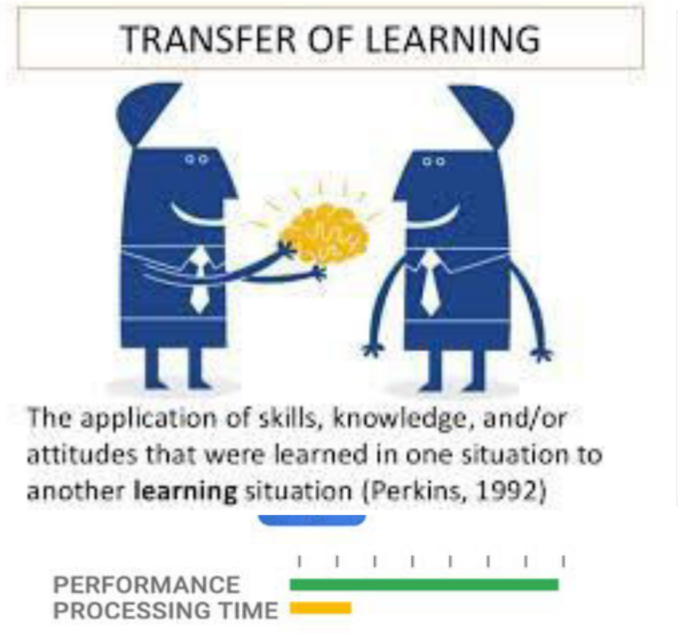
Machine Learning
- Describe Data as Features
- Learning Algorithm : Optimize parameters

Deep Learning
- Automatic learn good features and representations
- Attempt to learn multiple levels of feature representations
- Optimize weights, parameters and feature representations



Why exploring Deep Learning

- Availability of large amounts of datasets
- Availability of more computational power
- Low level features can be recognized
- Learned features are highly informative
- Transfer learning and reusability
- Many deep learning success stories

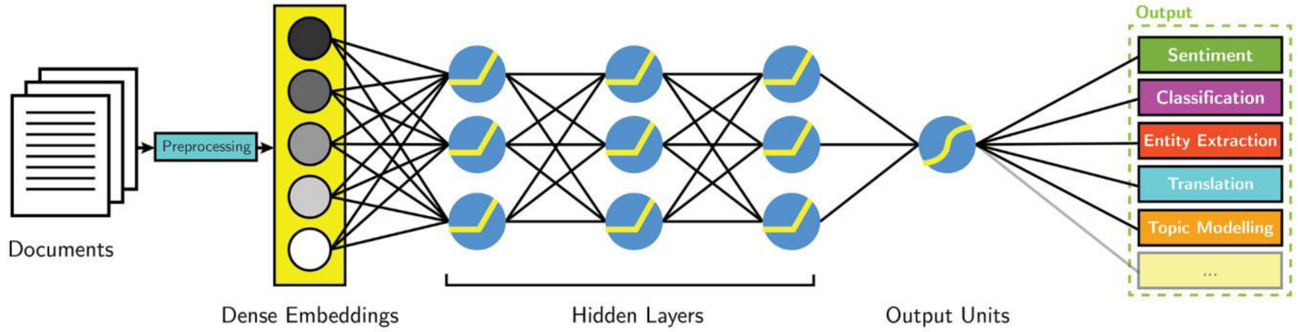


The application of skills, knowledge, and/or attitudes that were learned in one situation to another **learning** situation (Perkins, 1992)

Deep Learning for NLP applications

- Machine Translation
- Text Summarization
- Question Answering Systems
- Improved Text Classification Systems
- Language Modelling, ex – Image Captioning, Auto Replies

Deep Learning for NLP Architectures



Machine Learning with NLP

- Context is not considered : "he was good", "he was not good"
- Training history is not captured : "what is learned so far"

Deep Learning Architectures

- Convolutional Neural Networks
- Recurrent Neural Networks

Analytics Vidhya
Learn everything about analytics

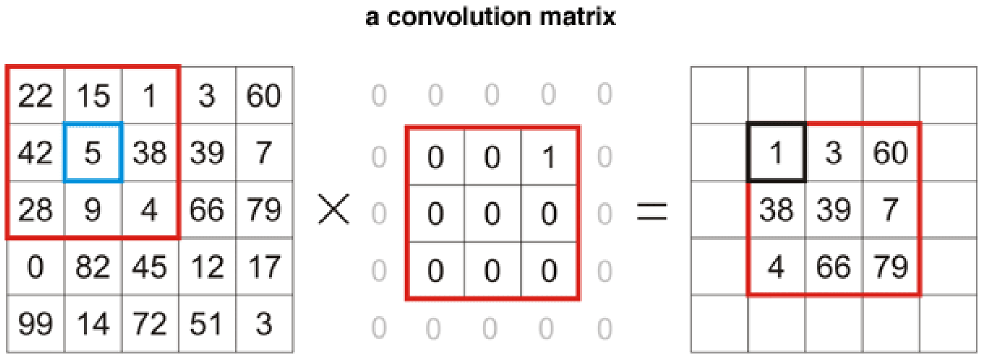
Convolutional Neural Networks

Convolutions:

- Filters (matrix / vectors) with learnable parameters
- Used to extract low-dimensional features from text

Convolution Features:

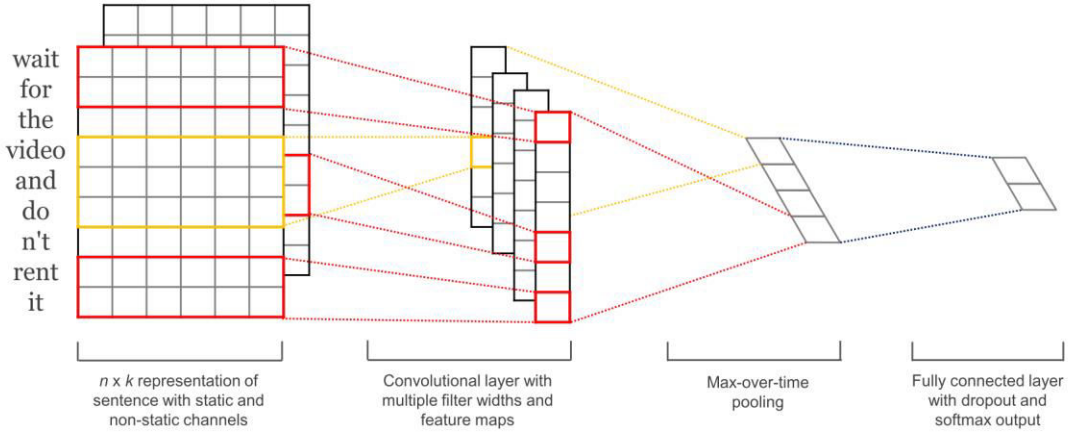
- Filter movement over input in sliding window fashion
- Outputs a weighted sum (of weights and input)



Convolutional Neural Networks

Feature Variety : Different filters for different types of features

Example: I like movies very much
- a certain filter might detect that "like" is a positive word and not a similarity comparison
- a certain filter of size 2 might detect "very much" and detect that it is an expression of degree


wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Analytics Vidhya
Learn everything about analytics

Convolutional Neural Networks : Text Classification

Input >> Embeddings >> Convolutions >> Pooling >> SoftMax (Classification)

- ***Input:*** A corpus - set of documents / sentences
- **Embedding Representation:**
  Vector notation of every document
  Document Vector Notation
  - Average over word vectors or weighted average using tfidf score as the weight
- ***Convolutions:***
  Generate features using multiple convolutions. Example - sliding over 3, 4 or 5 words at a time.
- ***Max Pooling:***
  Most informative features are extracted from the results of convolution layer
- **SoftMax:**
  Classification of learned vectors into one of the class

Text Classification : Convolutional Neural Networks

```
from keras import layers, models

input_layer = layers.Input((input_size, ))

embedding_layer = layers.Embedding(vocabulary_size, embedding_size)(input_layer)

conv_layer = layers.Convolution1D(100, 3, activation="relu")(embedding_layer)

pooling_layer = layers.GlobalMaxPool1D()(conv_layer)

output_layer1 = layers.Dense(50, activation="relu")(pooling_layer)

output_layer2 = layers.Dense(1, activation="sigmoid")(output_layer1)

model = models.Model(inputs=input_layer, outputs=output_layer2)
```

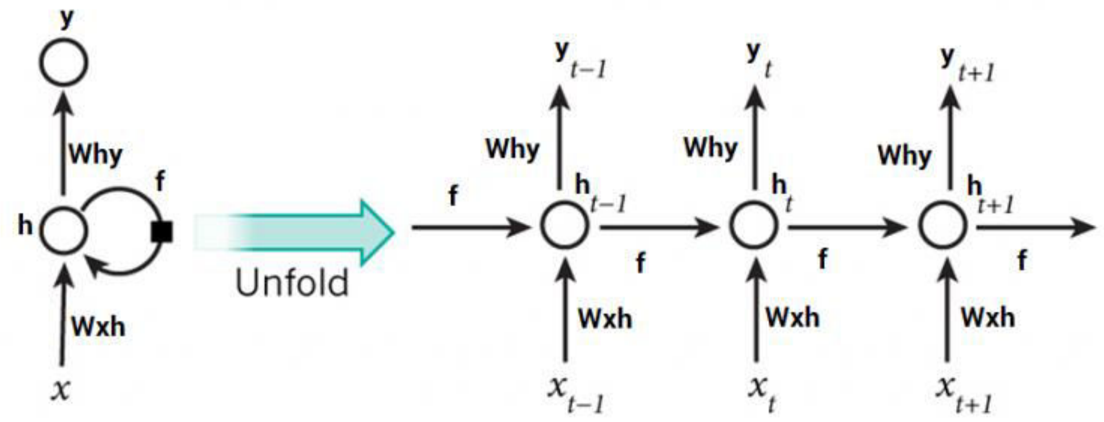
Analytics Vidhya
Learn everything about analytics

Text Classification : Recurrent Neural Networks

Neural Networks with memory state and activation loops

Activations propagate in both directions – outputs and inputs

Allows the neurons an ability to remember what have been learned so far



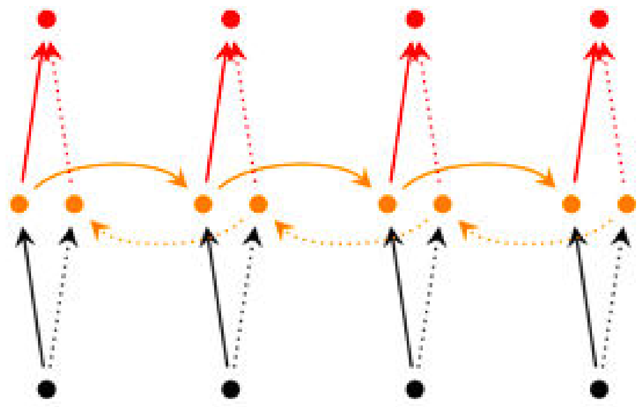
Text Classification : Recurrent Neural Networks

Condition on **all previous words,** Use same set of weights at all time steps

The memory state in RNNs gives an advantage over traditional neural networks but a problem called Vanishing Gradient is associated with them. In this problem, while learning with many layers, it becomes really hard for the network to learn and tune the parameters of the earlier layers. To address this problem, A new type of RNNs called LSTMs (Long Short-Term Memory) Models have been developed.

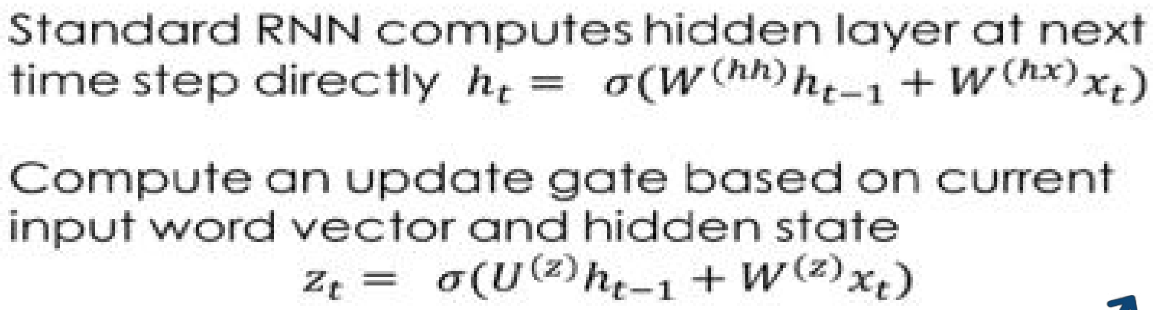
Text Classification : RNN Variants

BiDirectional RNNs
Main idea: incorporate both left and right context
output may not only depend on the **previous** elements
in the sequence, but also **future** elements.



GRUs : Main idea:
keep around memory to capture **long dependencies**
Allow error messages to flow at **different strengths** depen

LSTMs



Standard RNN computes hidden layer at next time step directly $h_t = \sigma(W^{(hh)} h_{t-1} + W^{(hx)} x_t)$

Compute an update gate based on current input word vector and hidden state

$$z_t = \sigma(U^{(z)} h_{t-1} + W^{(z)} x_t)$$

Deep Learning for NLP - 2

Analytics Vidhya
Learn everything about analytics

About the Module

- ❑ Sequence to Sequence Models
- ❑ Recurrent Neural Networks
- ❑ Long Short Term Memory Units
- ❑ LSTMs for Text Generation

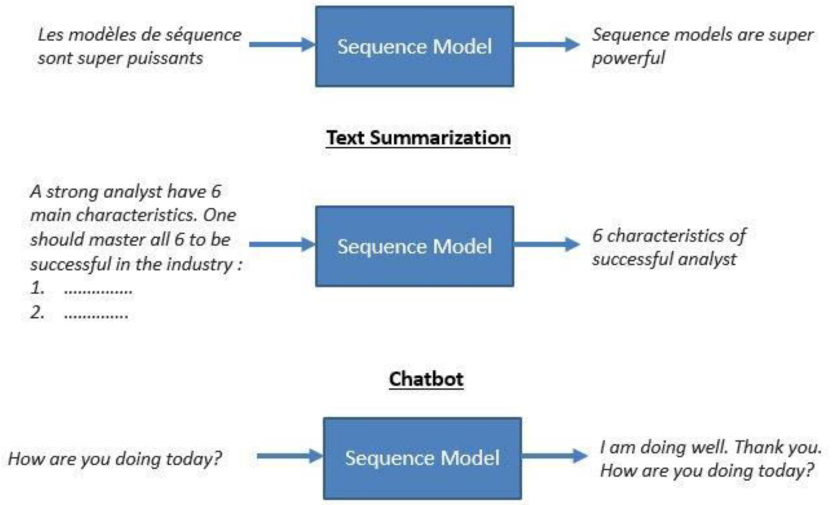
Sequence to Sequence Models

- Sequence model learns the probability of occurrence of an element based on previous elements
- Sequence Data Examples :

- Audio, Signals, Time Series
- Text: Sequence of words

Applications with NLP

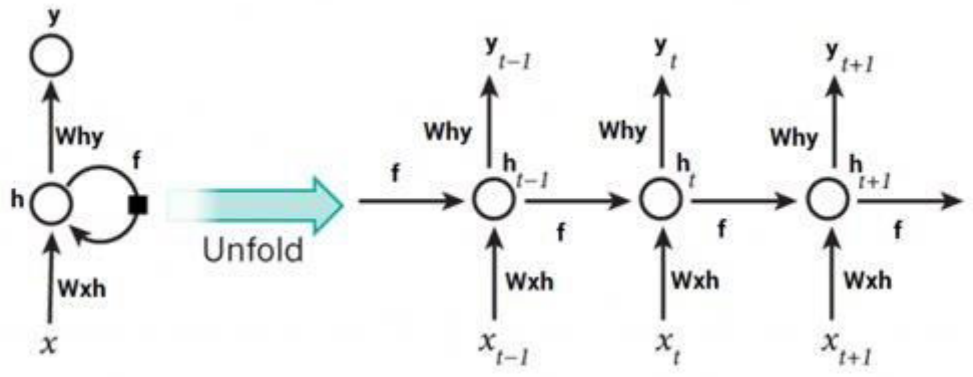
- Machine Translation
- Text Summarization
- Chatbots
- Text Generation


Les modèles de séquence sont super puissants
Sequence Model
Sequence models are super powerful
Text Summarization
A strong analyst have 6 main characteristics. One should master all 6 to be successful in the industry :
1. ....
2. ....
Sequence Model
6 characteristics of successful analyst
Chatbot
How are you doing today?
Sequence Model
I am doing well. Thank you. How are you doing today?

Analytics Vidhya
Learn everything about analytics

Recurrent Neural Networks

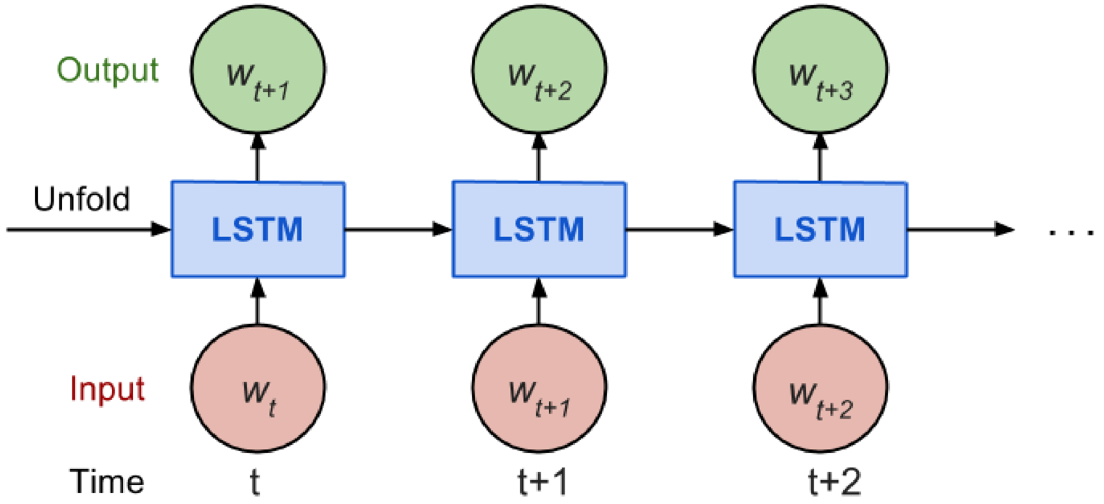
- Feed-Forward Networks - Activation outputs propagates only in one direction
- Recurrent Networks - Activation outputs propagates in both directions
  - From Inputs to Outputs, From Outputs to Inputs



- Loops act as 'memory state' of the neurons
- Allows the neurons an ability to remember what have been learned so far
- Vanishing Gradient problem : network tends to forget was learned in the earlier layers

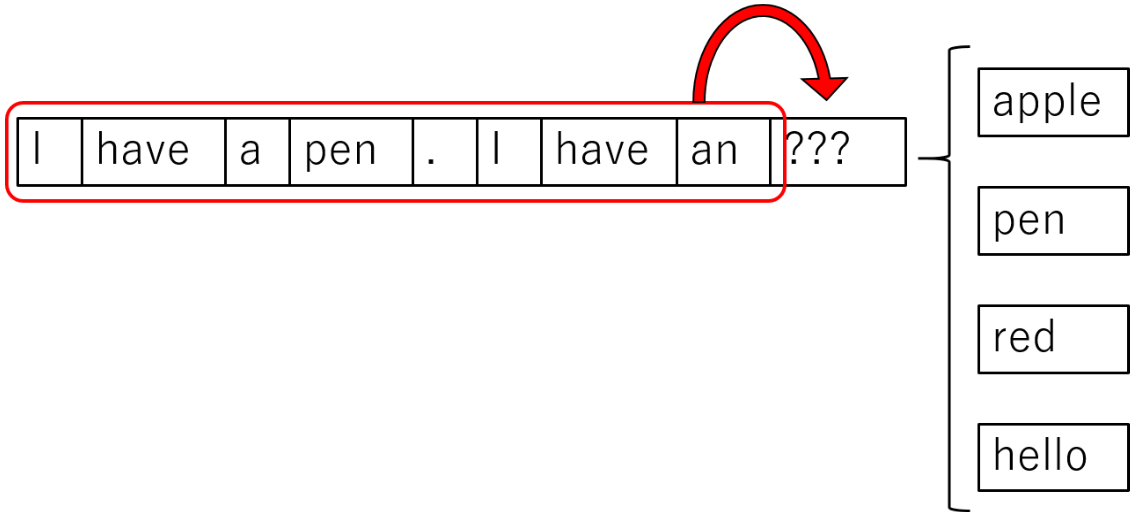
Long Short Term Memory (LSTMs)

- LSTMs are a special types of RNNs with an additional state 'cell state'
- Used to make adjustment in the information flow.
- With cell state, model can remember or forget the leanings more selectively.



LSTMs for Text Generation

- Predict the likelihood of occurrence of a word based on the previous sequence of words



Dataset Preparation

Create Word Index
Text: I will play football today

Tokens : [I, will, play, football, today]
Token Index : [0, 1, 2, 3, 4]

Convert text into sequence of tokens
N grams: Sequence of Tokens

Ngram	Sequence of Tokens	Padded Sequence
i will	[0, 1]	[0, 1, -1, -1, -1]
i will play	[0, 1, 2]	[0, 1, 2, -1, -1]
i will play football	[0, 1, 2, 3]	[0, 1, 2, 3, -1]
i will play football today	[0, 1, 2, 3, 4]	[0, 1, 2, 3, 4]

Analytics Vidhya
Learn everything about analytics

LSTM Model Architecture

Input -> Embedding -> LSTM -> MultiClass Classification

```
model = Sequential()
input_len = max_sequence_len – 1

# Add Input Embedding Layer
model.add(Embedding(total_words, 10, input_length=input_len))

# Add Hidden Layer 1 - LSTM Layer
model.add(LSTM(100))

# Add Output Layer
model.add(Dense(total_words, activation='softmax’))

model.compile(loss='categorical_crossentropy', optimizer='adam')
```

Analytics Vidhya
Learn everything about analytics