



# Welcome to Text Mining with Deep Learning

Going live at:

Chicago 11:00 am

San Francisco 9:00 am

New York 12:00 pm

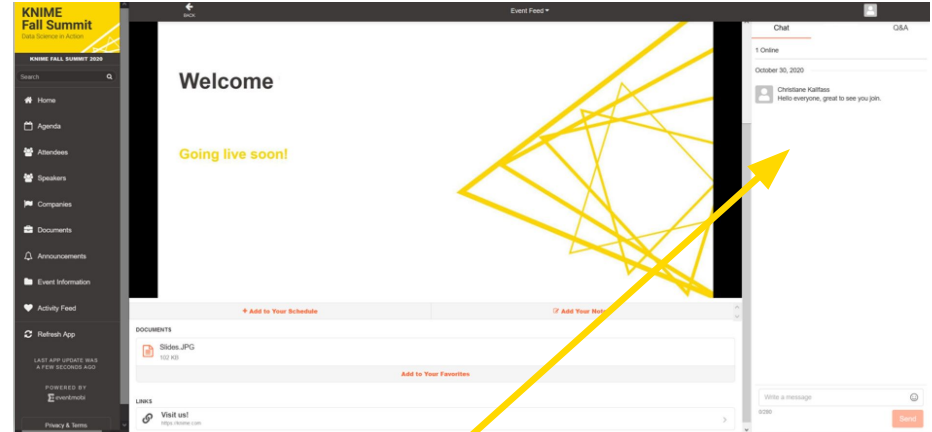
Berlin 6:00 pm



# Housekeeping

- Post in the chat where you are dialing in from and discuss with other attendees
- Questions? Post them in the Q&A

Questions will be answered after the presentation.



# Agenda

---

- Introduction to Sentiment Analysis
  - Review of sentiment analysis techniques
  - Sentiment analysis in-depth: step-by-step
  - Transformation for deep learning
- Introduction to Deep Learning
  - Neural network vs. deep neural network
  - How to build a simple deep neural network
  - Training and classification

# Text Processing Extension

- Other Data Types
  - Text Processing
    - IO
      - Brat Document Writer
      - Dml Document Parser
      - Document Grabber
      - Flat File Document Parser
      - OpenNLP NER Model Reader
      - PDF Parser
      - PubMed Document Parser
      - RSS Feed Reader
      - Sdml Document Parser
      - Tika Parser
      - Word Parser
    - Enrichment
      - Abner Tagger
      - Dictionary Tagger
      - Dictionary Tagger (Multi Column)
      - OpenNLP NE Tagger
      - Oscar Tagger
      - POS Tagger
      - Stanford Tagger
      - StanfordNLP NE Learner
      - StanfordNLP NE Scorer
      - StanfordNLP NE Tagger
      - Wildcard Tagger
- Transformation
  - ABC Bag Of Words Creator
  - ABC Document Data Assigner
  - ABC Document Data Extractor
  - ABC Document Vector
  - ABC Document Vector Applier
  - ABC Document Vector Hashing
  - ABC Document Vector Hashing Applier
  - ABC Meta Info Extractor
  - ABC Meta Info Inserter
  - ABC Sentence Extractor
  - ST String To Term
  - TS Strings To Document
  - TS Tags To String
  - ABC Term Neighborhood Extractor
  - TS Term To String
  - TS Term To Structure
  - ABC Term Vector
  - ABC Unique Term Extractor
- Frequencies
  - 12345 DF
  - 12345 Frequency Filter
  - 12345 ICF
  - 12345 IDF
  - 12345 NGram Creator
  - 12345 TF
  - 12345 Term Co-Occurrence Counter
  - 12345 Term Document Entropy
- Preprocessing
  - Case Converter
  - Diacritic Remover
  - Dictionary Filter
  - Dictionary Replacer
  - Dictionary Replacer (File-based)
  - Hyphenator
  - Kuhlen Stemmer
  - Modifiable Term Filter
  - N Chars Filter
  - Number Filter
  - Porter Stemmer
  - Punctuation Erasure
  - Regex Filter
  - Replacer
  - Snowball Stemmer
  - Stanford Lemmatizer
  - Stop Word Filter
  - Tag Filter
  - Tag Stripper
- Mining
  - Chi-Square Keyword Extractor
  - Keygraph Keyword Extractor
  - ABC StanfordNLP Open Information Extractor
  - ABC StanfordNLP Relation Extractor
  - ABC Topic Extractor (Parallel LDA)
- Misc
  - Category To Class
  - Document Viewer
  - Markup Tag Filter
  - String Matcher
  - Tag Cloud
  - ABC Tika Language Detector
  - ABC Tika Parser URL Input
- Meta Nodes
  - Simple Preprocessing
  - Extended NER Preprocessing
  - 12345 Frequencies
  - 12345 Vector Creation

# Sentiment Analysis – An Example



Samsung

Samsung Galaxy S7 Edge G935A 32GB Unlocked - Gold Platinum



125 customer reviews | 606 answered questions



**Beautiful phone from a wonderful seller!**

By

ay on May 29, 2017

Color: Gold

**Verified Purchase**

This practically new beautiful phone well exceeded my expectations!



**One Star**

By

on August 3, 2016

Color: Black Onyx

**Verified Purchase**

Very bad experience

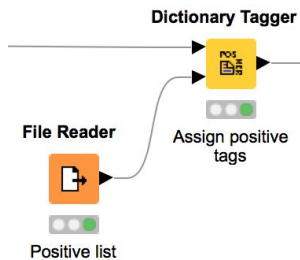


# Sentiment Analysis

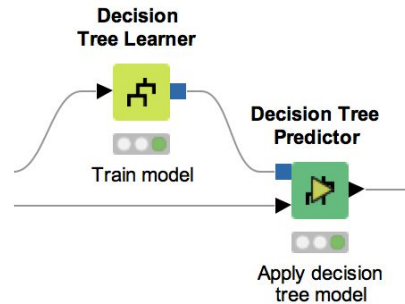
**Task:** Determine the expressed opinion in a document/text, e.g. positive, negative

*Sentiment Analysis = Opinion Mining = Emotion AI*

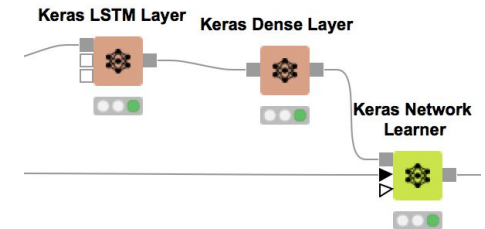
## Lexicon Based



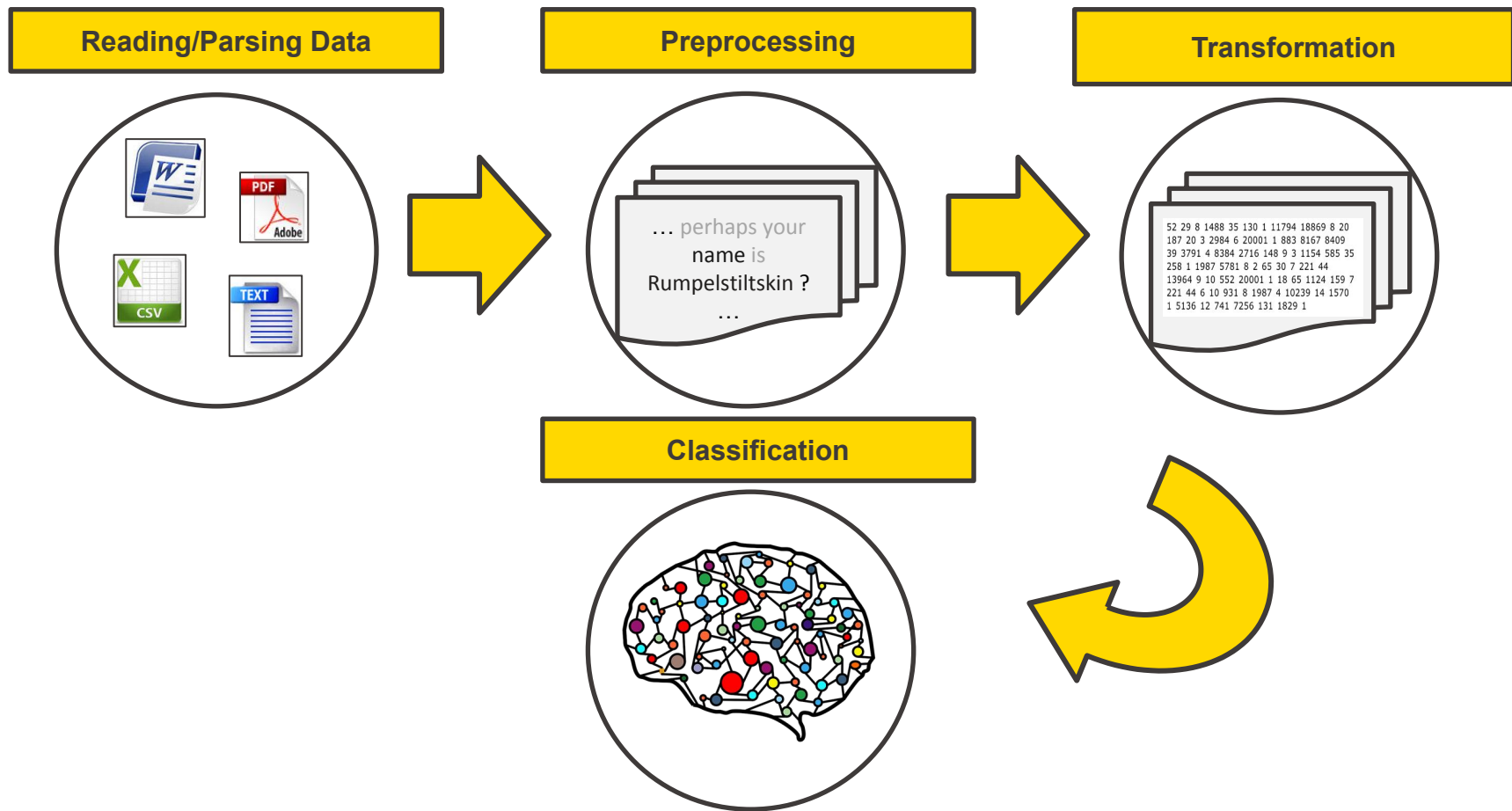
## Machine Learning



## Deep Learning

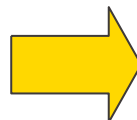
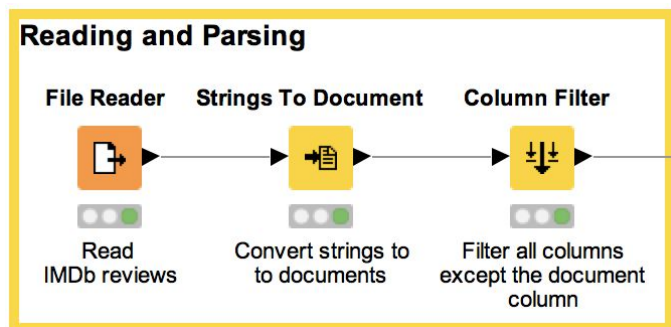


# Philosophy



# Part 1: Reading and Parsing Data

## Read/Parse textual data



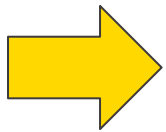
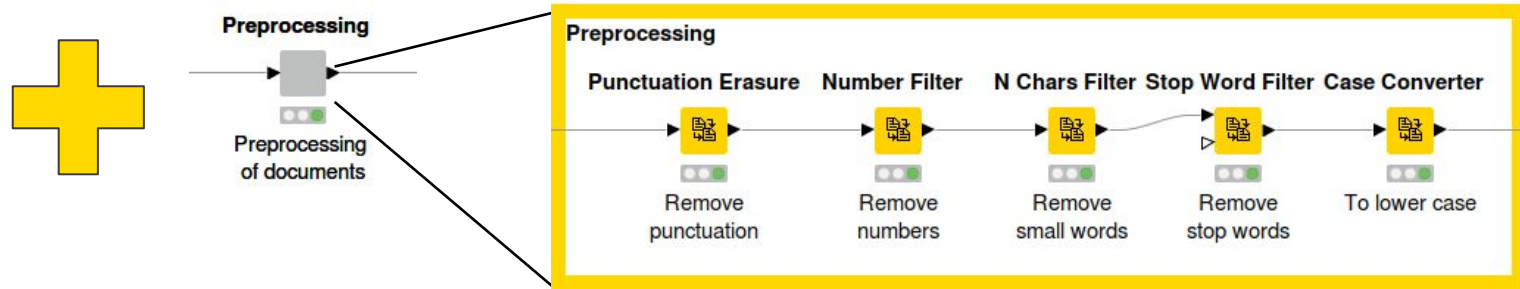
I	sentiment	S	text
0			Although the production and Jerry Jameson's direction
0			Capt. Gallagher (Lemmon) and flight attendant Eve C
0			Towards the end of the movie, I felt it was too techni
0			This is the kind of movie that my enemies content I w
0			I saw 'Descent' last night at the Stockholm Film Festi
0			Some films that you pick up for a pound turn out to k
0			This is one of the dumbest films, I've ever seen. It rip
1			Bromwell High is a cartoon comedy. It ran at the sam
1			Homelessness (or Houselessness as George Carlin s
1			Brilliant over-acting by Lesley Ann Warren. Best dram
1			This is easily the most underrated film inn the Brook
1			This is not the typical Mel Brooks film. It was much le
1			This isn't the comedic Robin Williams, nor is it the qu
1			Yes its an art... to successfully make a slow paced th
1			In this critically acclaimed psychological thriller base
1			THE NIGHT LISTENER (2006) **1/2 Robin Williams, Tor
1			You know, Robin Williams, God bless him, is constant
1			When I first read Armistead Maupins story I was take
1			I liked the film. Some of the action scenes were very



# Part 2: Preprocessing

Example:

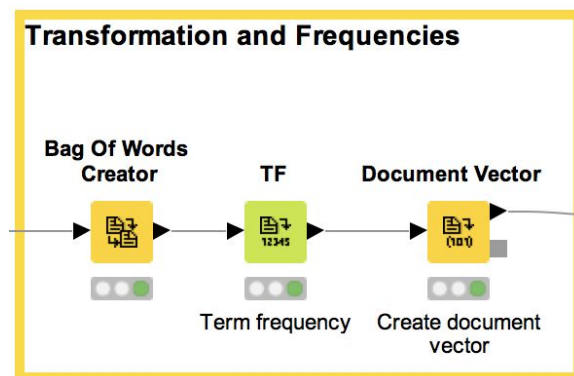
- This movie is horrible. The acting is a waste basket.. Though the scenery is great.
- Even though this movie came out a year before I was born, it's my favorite movie.
- This is definitely one of my favorite comedies.



- movie horrible acting waste basket scenery
- movie born favorite movie
- definitely favorite comedies

# Part 3: Transformation

## Transform documents to numbers



Row ID	D movi	D absolut	D act	D worst	D amaz	D aw	D direct
Row0	1	1	1	1	1	1	1
Row1	1	1	0	0	0	1	0
Row2	0	0	0	0	0	0	0
Row3	1	1	0	1	0	0	0
Row4	1	0	0	0	0	0	0
Row5	0	0	0	0	0	0	0
Row6	1	0	0	0	1	0	0
Row7	1	0	0	0	0	0	0
Row8	1	0	0	0	0	0	0
Row9	1	0	0	0	0	0	0
Row10	1	1	0	1	0	0	0
Row11	1	0	0	0	0	0	0
Row12	1	1	0	1	0	0	0
Row13	1	0	0	0	0	0	0
Row14	0	0	0	0	0	0	0

# Transformation for Deep Learning

Expected input of a network:

- Numerical representation of each document encoding the words and their order

This film is mediocre at best. Angie Harmon is as funny as a bag of hammers. Her bitchy demeanor from Law and Order carries over in a failed attempt at comedy. Charlie Sheen is the only one to come out unscathed in this horrible anti-comedy. The only positive thing to come out of this mess is Charlie and Denise's marriage. Hopefully that effort produces better results.



Row ID	S Term as String	S Truncation	S WordAsInteger
Row1...	he	10147885	53
Row1...	good	10144990	54
Row1...	some	10191253	55
Row1...	more	10163948	56
Row2...	would	10209290	57
Row2...	what	10207660	58
Row1...	time	10199284	59
Row2...	very	10205272	60



Dictionary Replacer

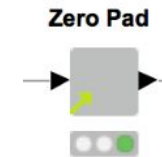
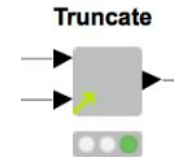


Apply Dictionary



52 29 8 1488 35 130 1 11794 18869 8 20  
187 20 3 2984 6 20001 1 883 8167 8409  
39 3791 4 8384 2716 148 9 3 1154 585 35  
258 1 1987 5781 8 2 65 30 7 221 44  
13964 9 10 552 20001 1 18 65 1124 159 7  
221 44 6 10 931 8 1987 4 10239 14 1570  
1 5136 12 741 7256 131 1829 1

- Equivalent input shape of each document
  - Truncate too long documents
  - Zero pad too short documents



# Transformation Example

- movie horrible acting waste basket scenery
- movie born favorite movie
- definitely favorite comedies

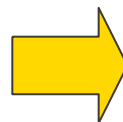
Dictionary Replacer



Apply Dictionary



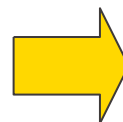
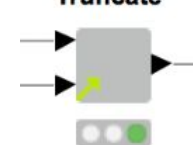
Term	Int
favorite	1
movie	2
acting	3
basket	4
born	5
comedies	6
definitely	7
horrible	8
scenery	9
waste	10



- [2 8 3 10 4 9]  
- [2 5 1 2]  
- [7 1 6]



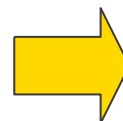
Truncate



- [2 8 3 10 4]  
- [2 5 1 2]  
- [7 1 6]

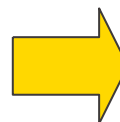
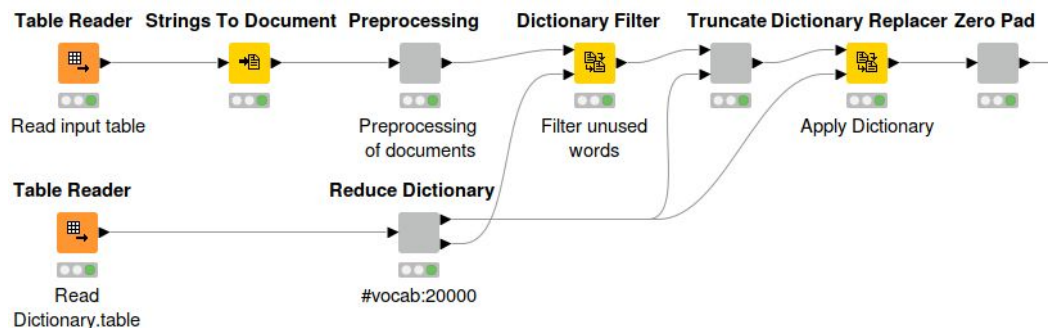


Zero Pad



- [2 8 3 10 4]  
- [2 5 1 2 0]  
- [7 1 6 0 0]

# Transformation for Deep Learning (Demo)



Row ID	sentiment	AggregatedValues
Row0#0	0	[6,4539,21,...]
Row0#1	0	[3521,28,2300,...]
Row0#2	0	[2,162,1727,...]
Row0#3	0	[475,159,273,...]
Row0#4	0	[17,479,390,...]
Row0#5	0	[91,379,82,...]
Row0#6	0	[151,672,1714,...]
Row0#7	0	[991,7268,243,...]
Row0#8	0	[2,971,7475,...]
Row0#9	0	[2,11,6,...]
Row0#10	0	[2,1054,2791,...]
Row0#11	0	[15,3554,20,...]
Row0#12	0	[15,85,1078,...]
Row0#13	0	[2,603,5486,...]
Row0#14	0	[1,49,92,...]
Row0#15	0	[17368,15,55,...]
Row0#16	0	[1,4546,1747,...]

# Dataset

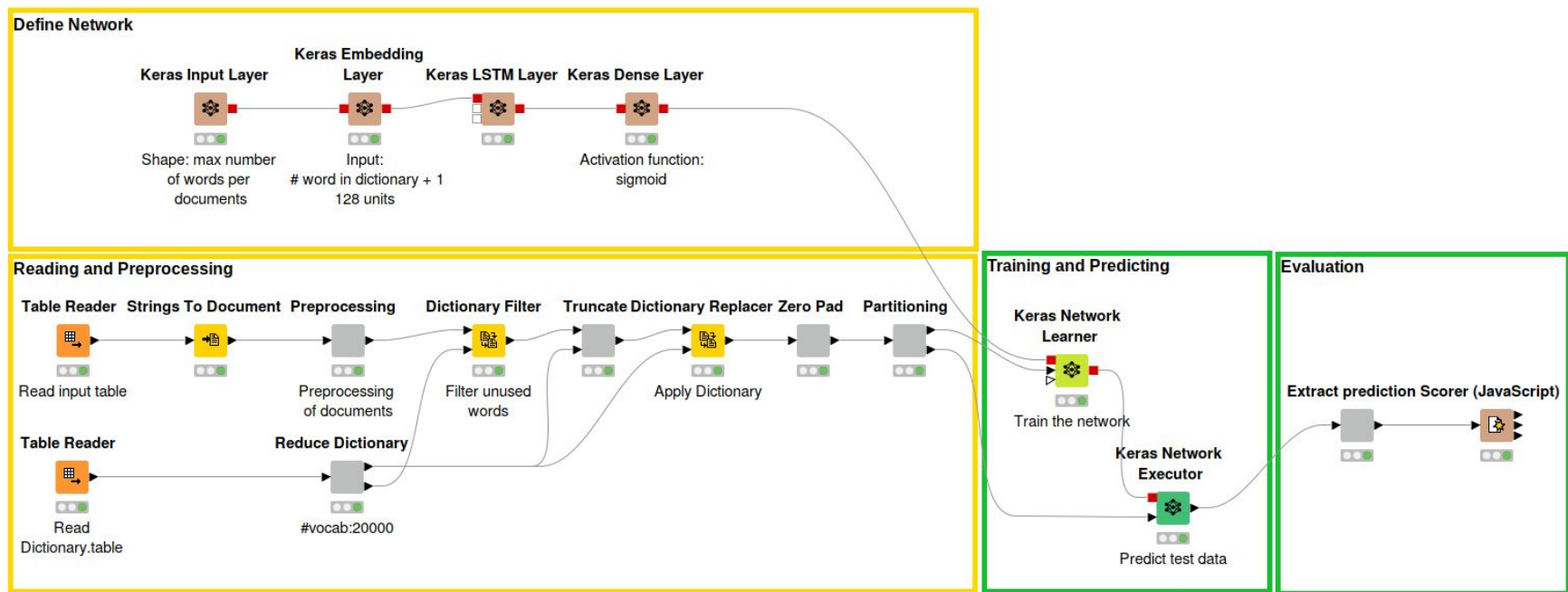
---

- Subset of the IMDb (Internet Movie Database) [Large Movie Review Dataset v1.0](#) with 50000 documents\*
  - 25000 documents from the positive group
  - 25000 documents from the negative group
- Goal: Assign the correct sentiment label to each document

(\*) For details about the data set see <http://ai.stanford.edu/~amaas/data/sentiment/>

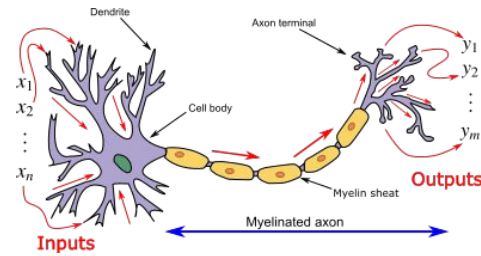
Data citation: Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)

# Part 4: Classification

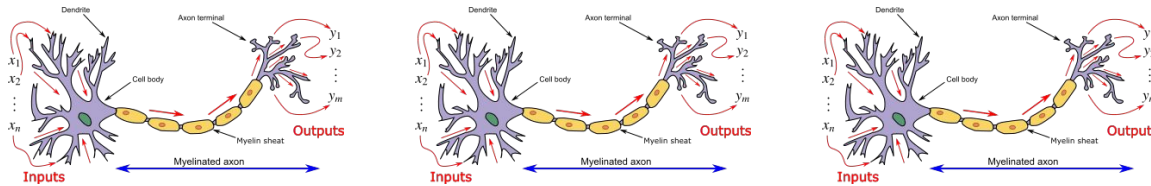


# Neural Network

## Neuron



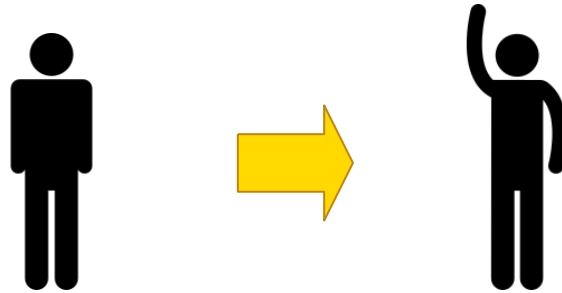
## Neural networks





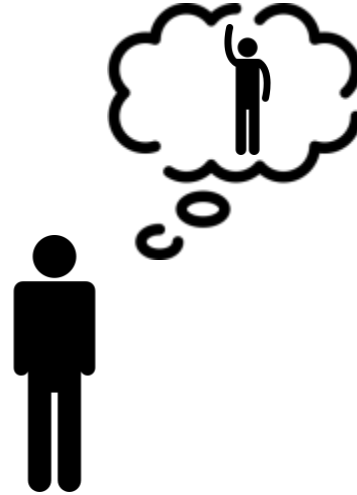
# Neural Network

---



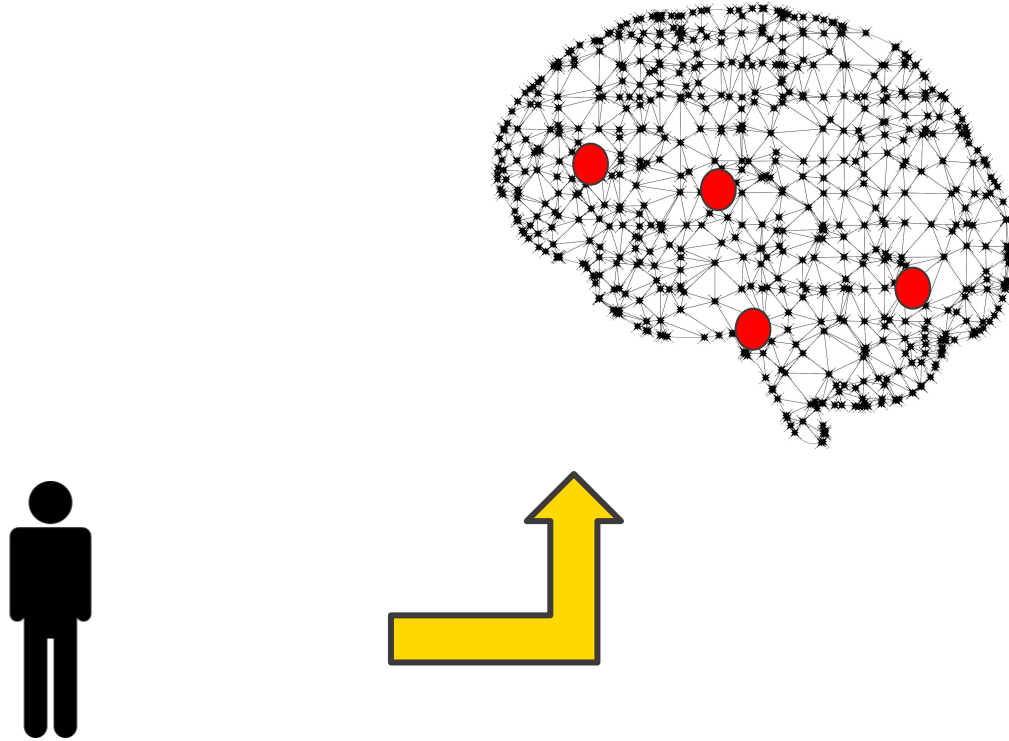
# Neural Network

---



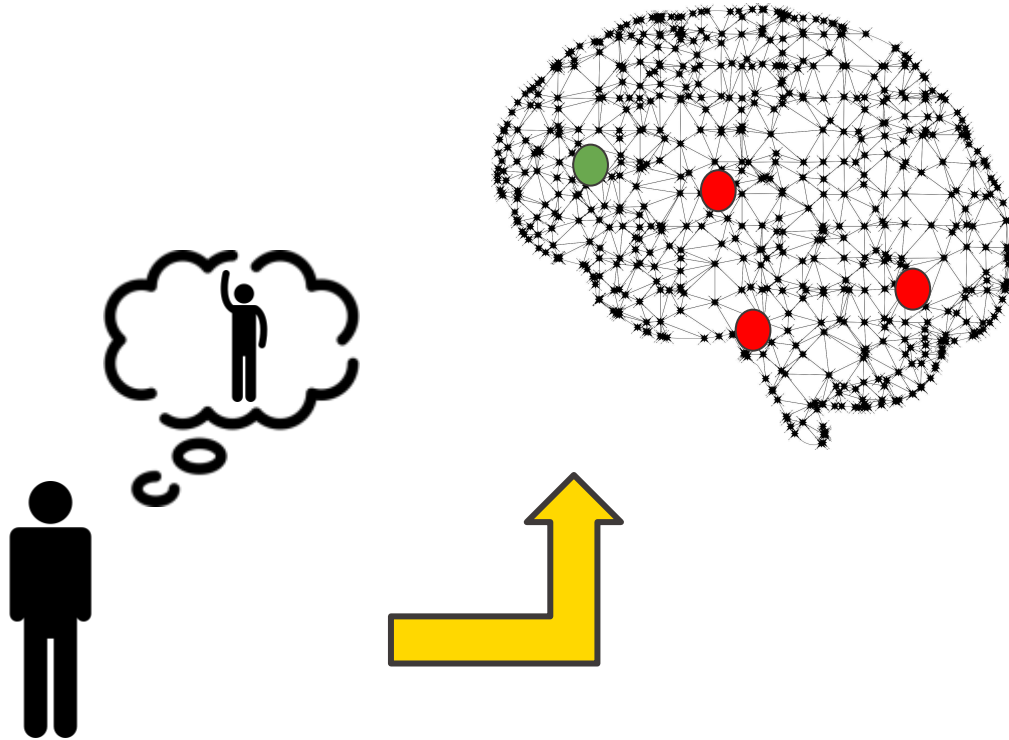
# Neural Network

---



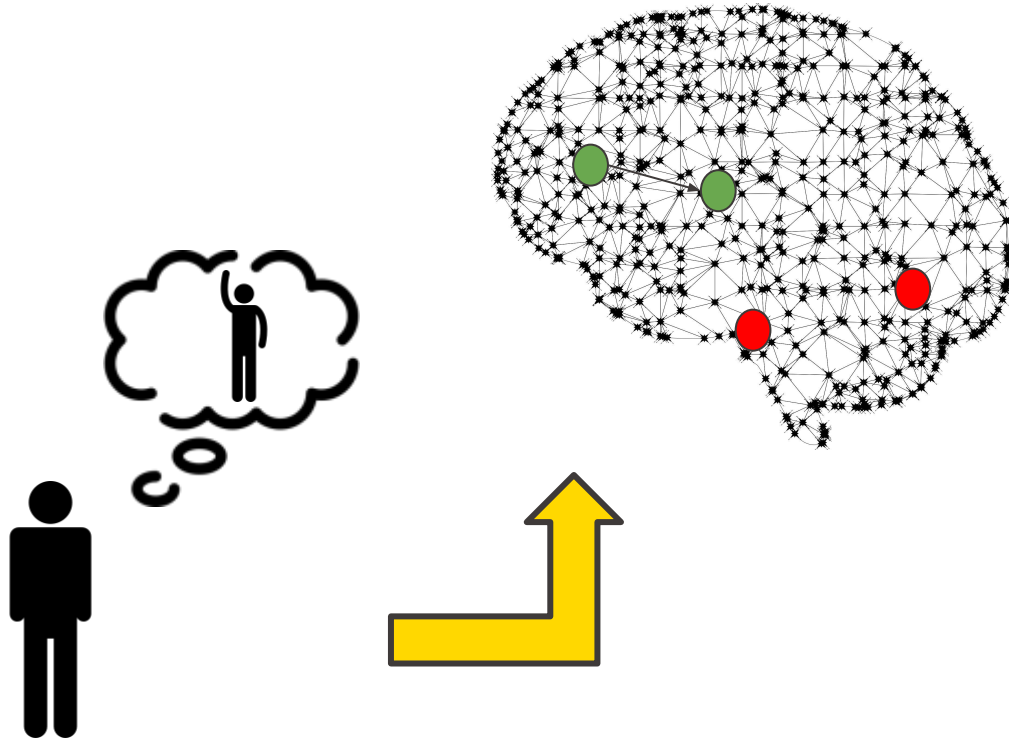
# Neural Network

---



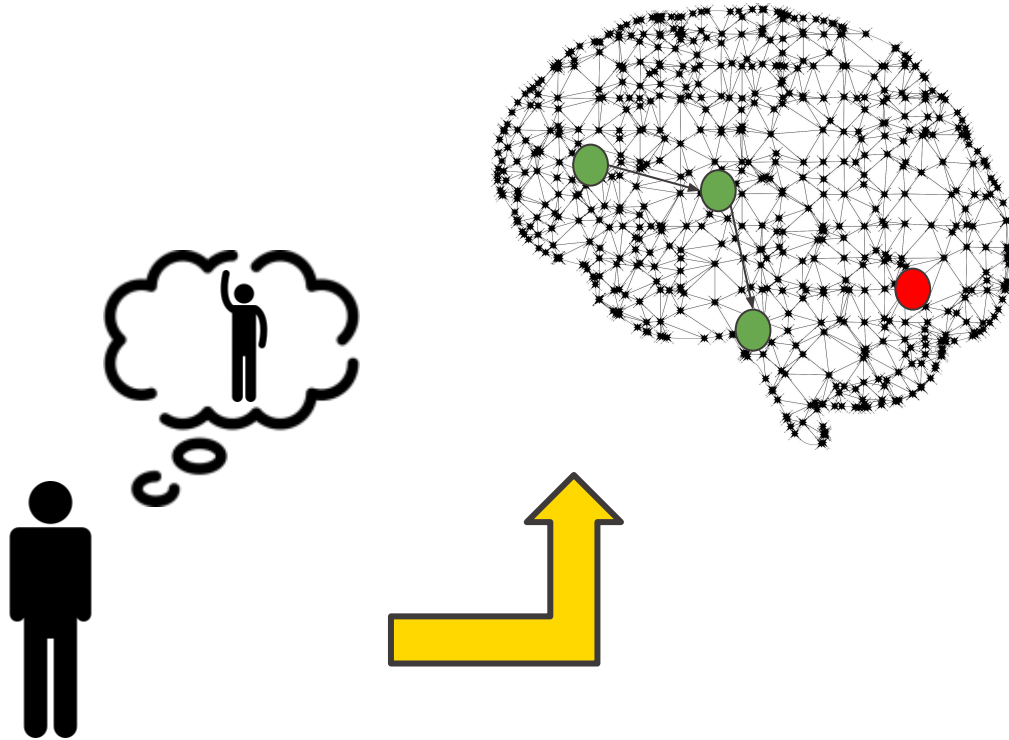
# Neural Network

---



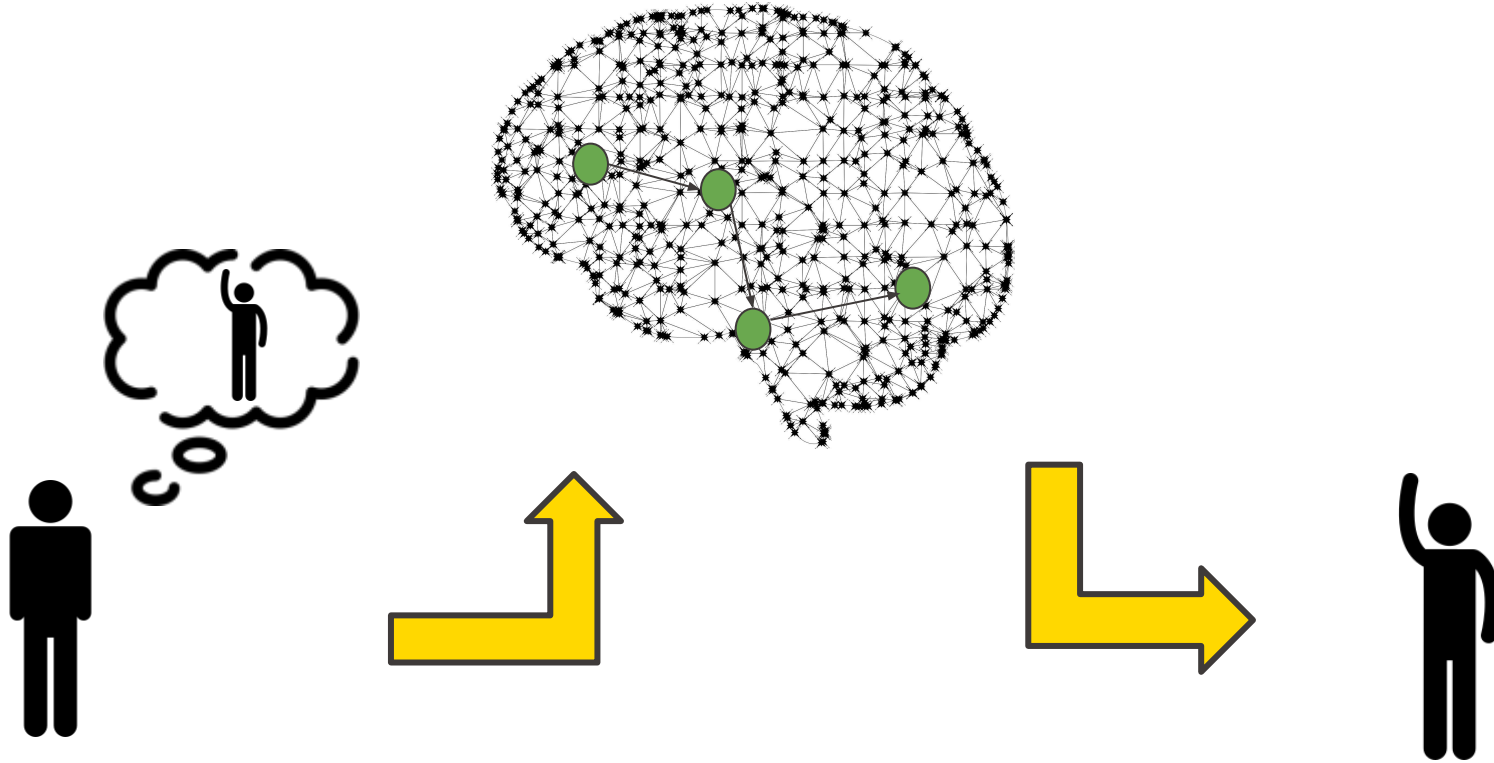
# Neural Network

---



# Neural Network

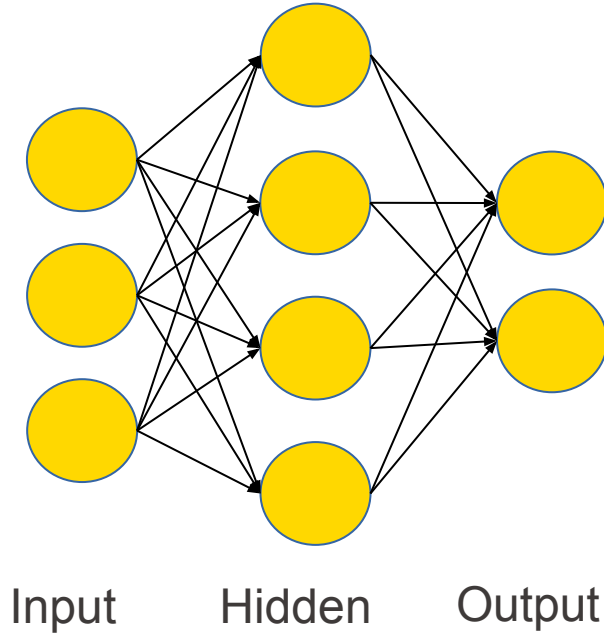
---



# Deep Learning

---

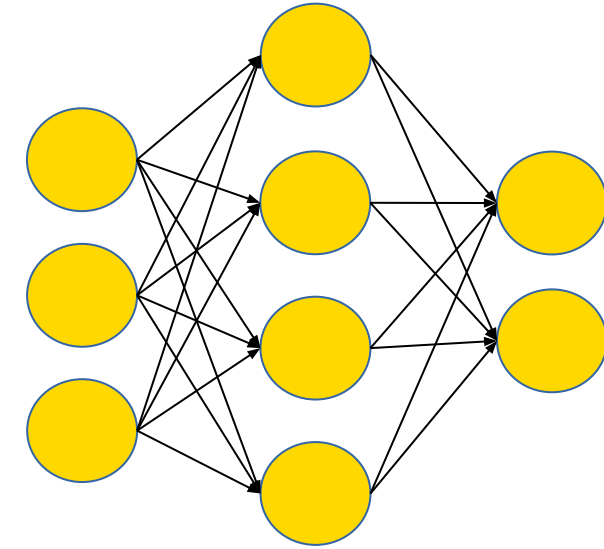
Artificial neural network



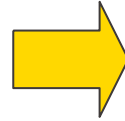


# Deep Learning

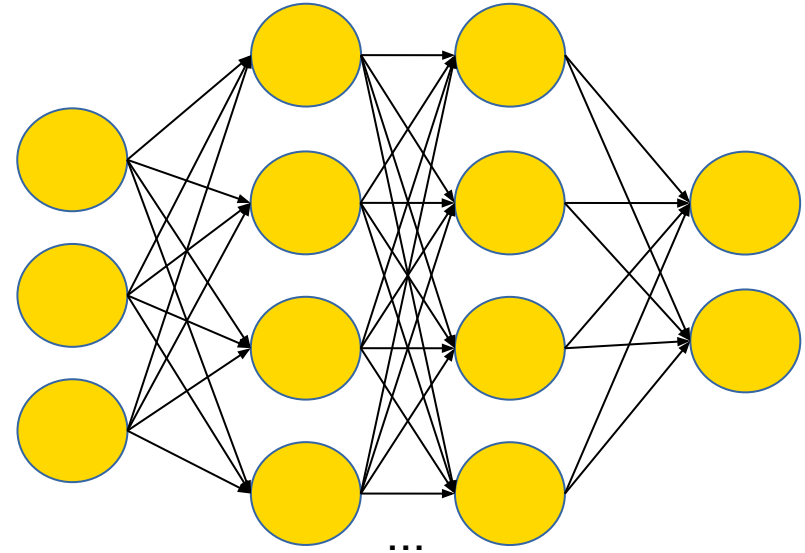
Artificial neural network



Input Hidden Output



Deep neural network

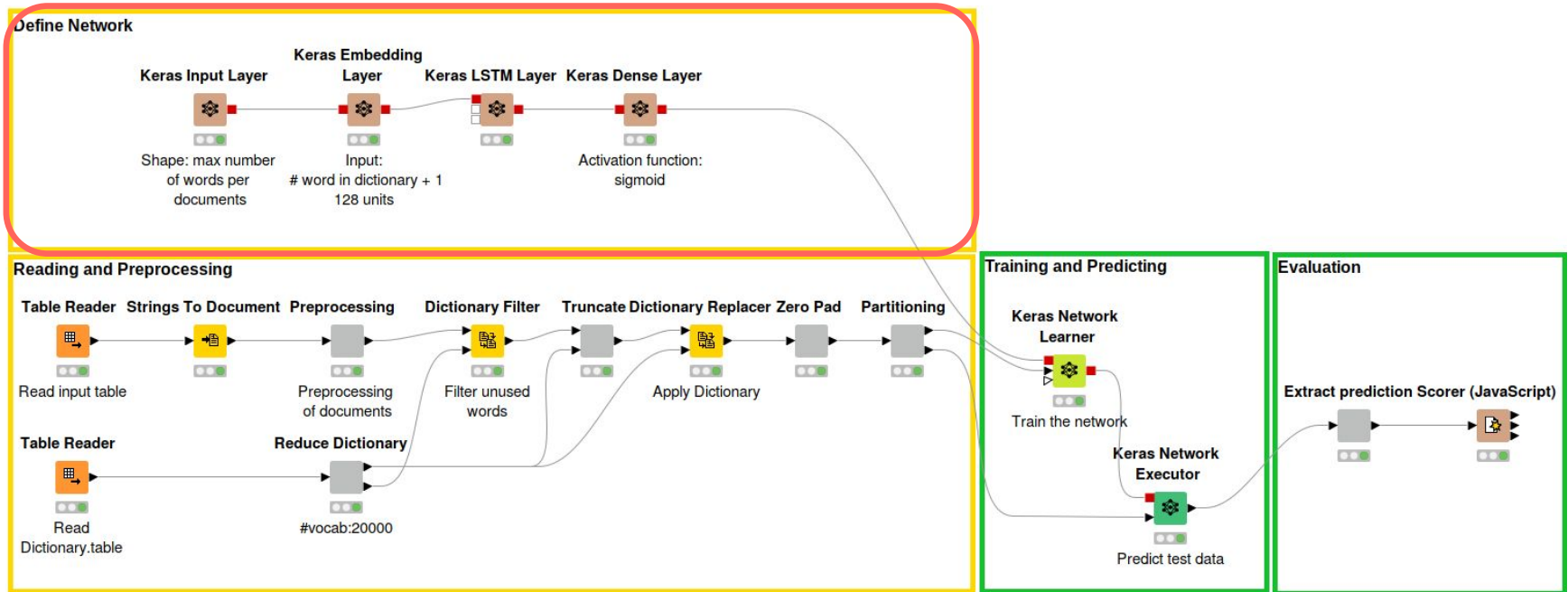


Input Hidden Hidden Output

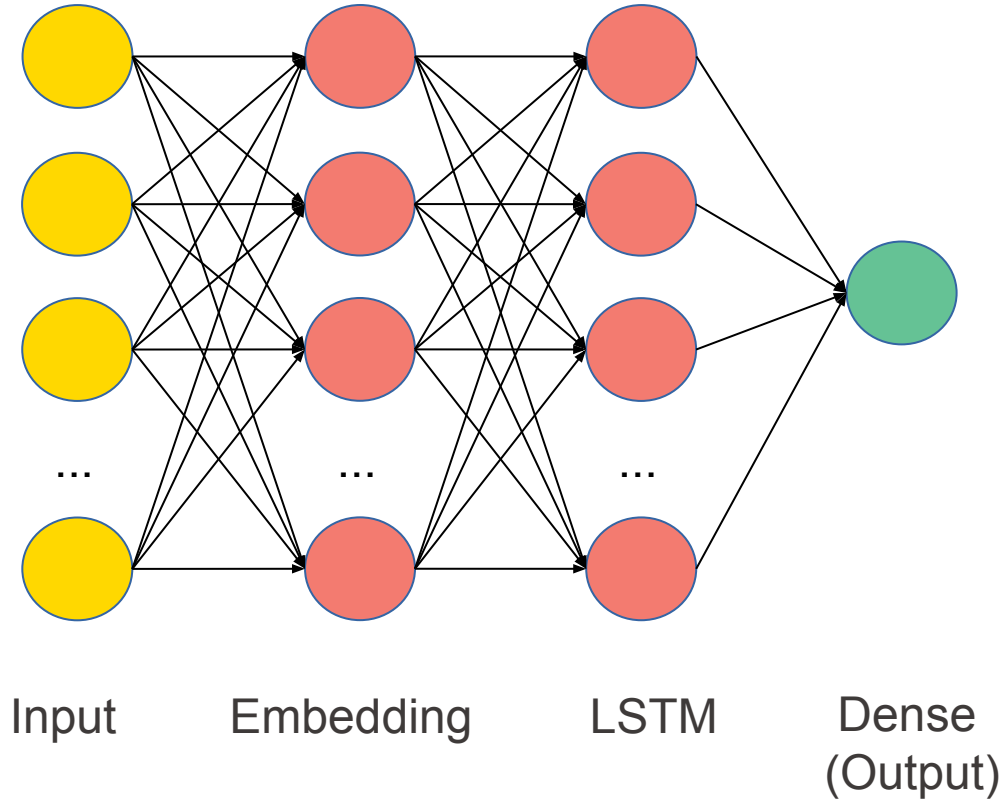
n hidden layers

(convolutional, embedding, dense, recurrent, ...)

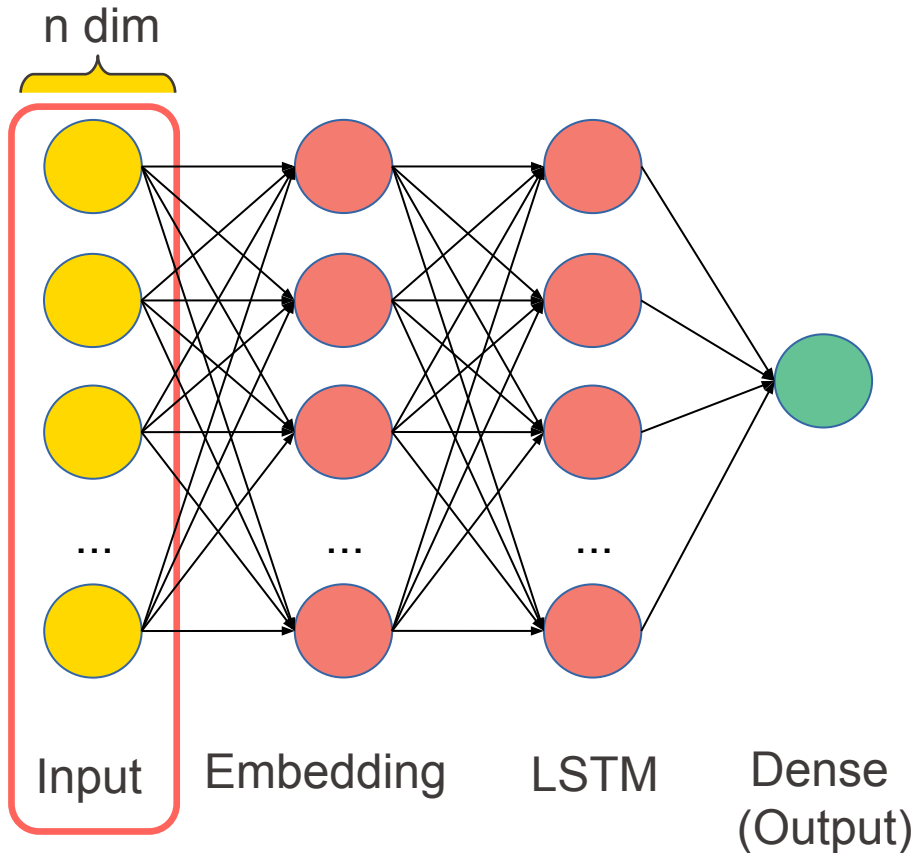
# Part 4: Classification



# Our Network Topology



# Input Layer

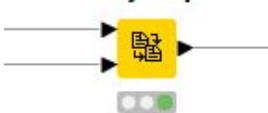


- Input layer passes input data to the first hidden layer
- The dimension is the document size, i.e. the number of words in each document

# Transformation Example

- movie horrible acting waste basket scenery
- movie born favorite movie
- definitely favorite comedies

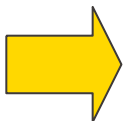
Dictionary Replacer



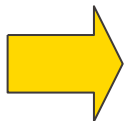
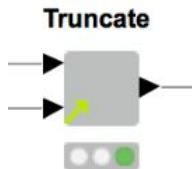
Apply Dictionary



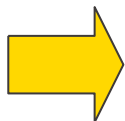
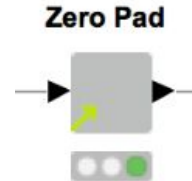
Term	Int
favorite	1
movie	2
acting	3
basket	4
born	5
comedies	6
definitely	7
horrible	8
scenery	9
waste	10



- [2 8 3 10 4 9]  
- [2 5 1 2]  
- [7 1 6]

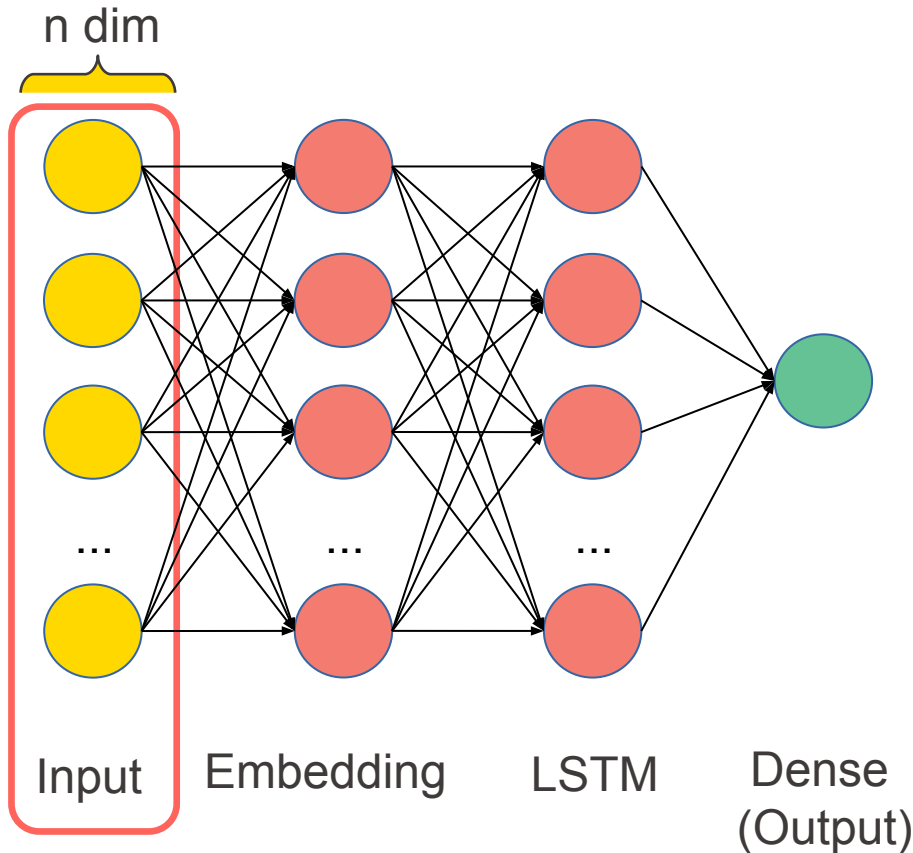


- [2 8 3 10 4]  
- [2 5 1 2]  
- [7 1 6]



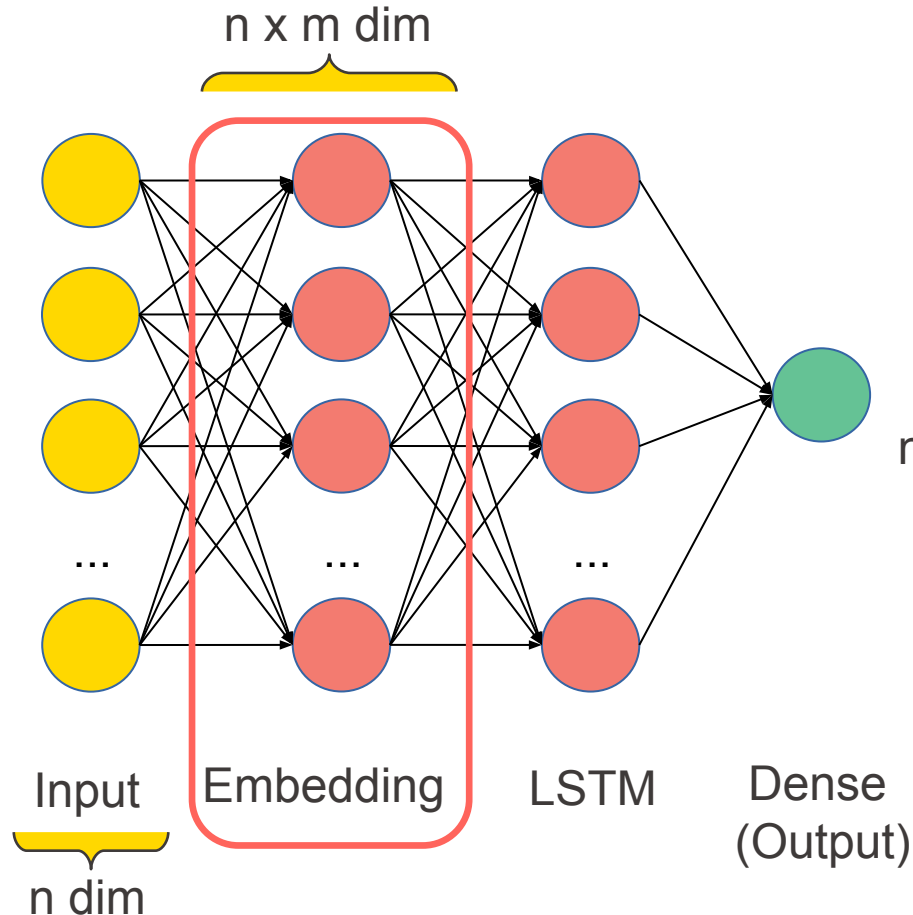
- [2 8 3 10 4]  
- [2 5 1 2 0]  
- [7 1 6 0 0]

# Input Layer



- Input layer passes input data to the first hidden layer
- The dimension is the document size, i.e. the number of words in each document

# Embedding Layer



- Embedding layer embeds each word into a dense, high dimensional vector

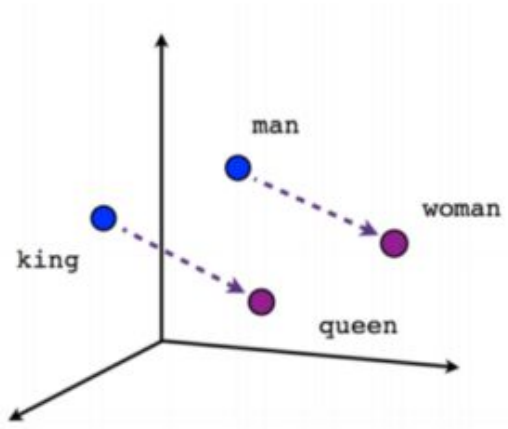
- Example:

$$\left. \begin{matrix} 2 \\ 5 \\ 1 \end{matrix} \right\} n \rightarrow \left\{ \begin{matrix} [0.3, 0.2, \dots, 0.6] \\ [0.4, 0.6, \dots, 0.5] \\ [0.1, 0.8, \dots, 0.3] \end{matrix} \right\} n$$

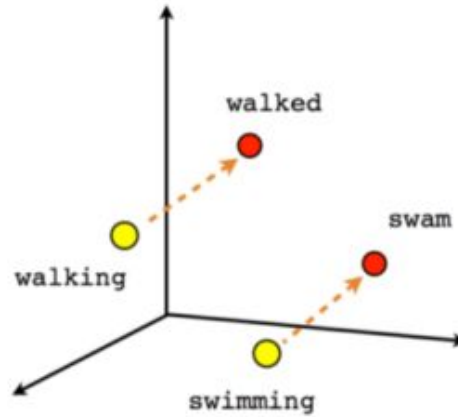
$m$

- Similar words will be embedded near each other in the vector space

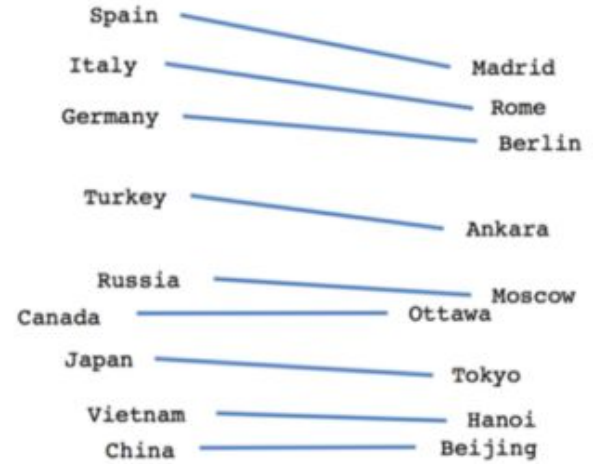
# Embedding Layer



Male-Female



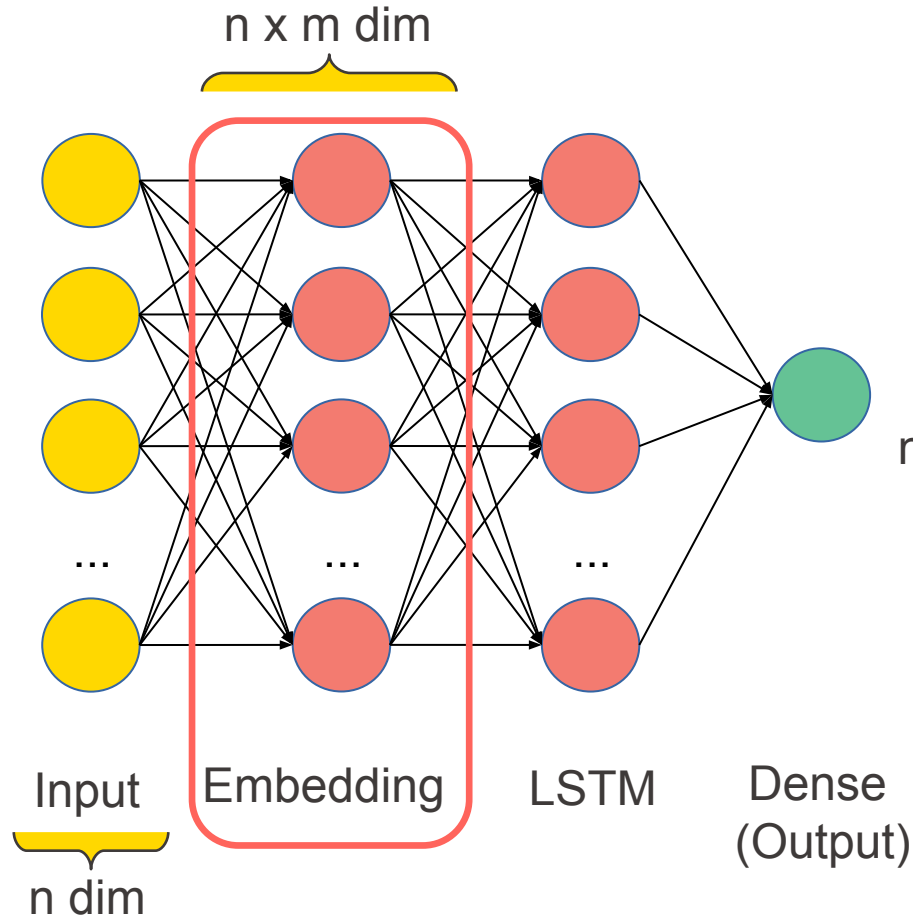
Verb tense



Country-Capital



# Embedding Layer



- Embedding layer embeds each word into a dense, high dimensional vector

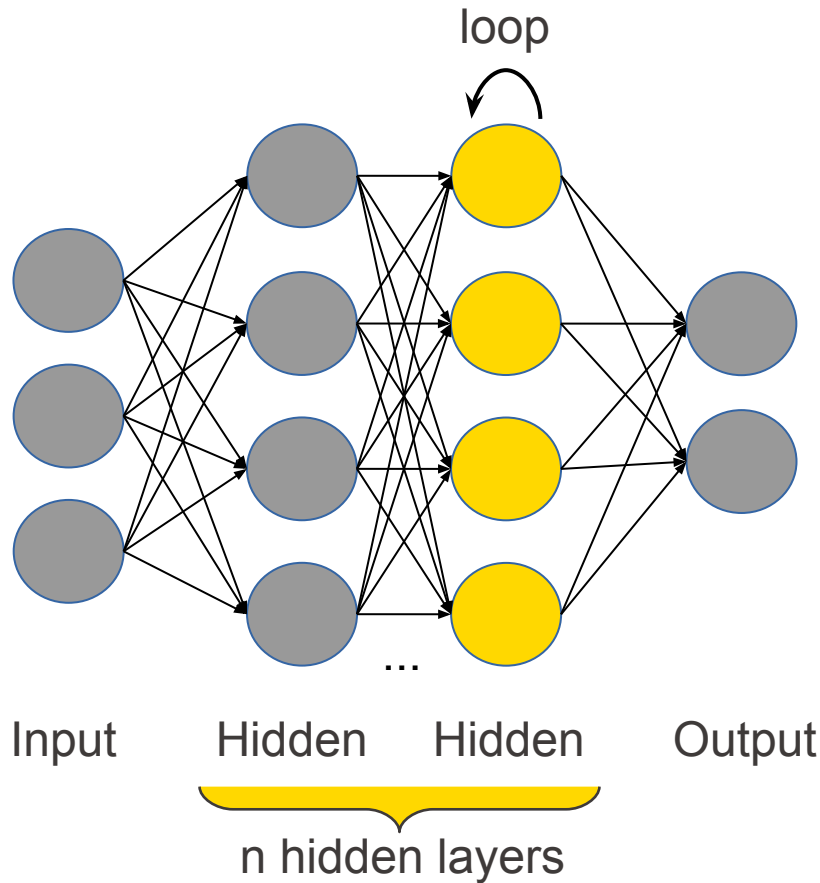
- Example:

$$\left. \begin{matrix} 2 \\ 5 \\ 1 \end{matrix} \right\} n \rightarrow \left\{ \begin{matrix} [0.3, 0.2, \dots, 0.6] \\ [0.4, 0.6, \dots, 0.5] \\ [0.1, 0.8, \dots, 0.3] \end{matrix} \right\} n$$

$m$

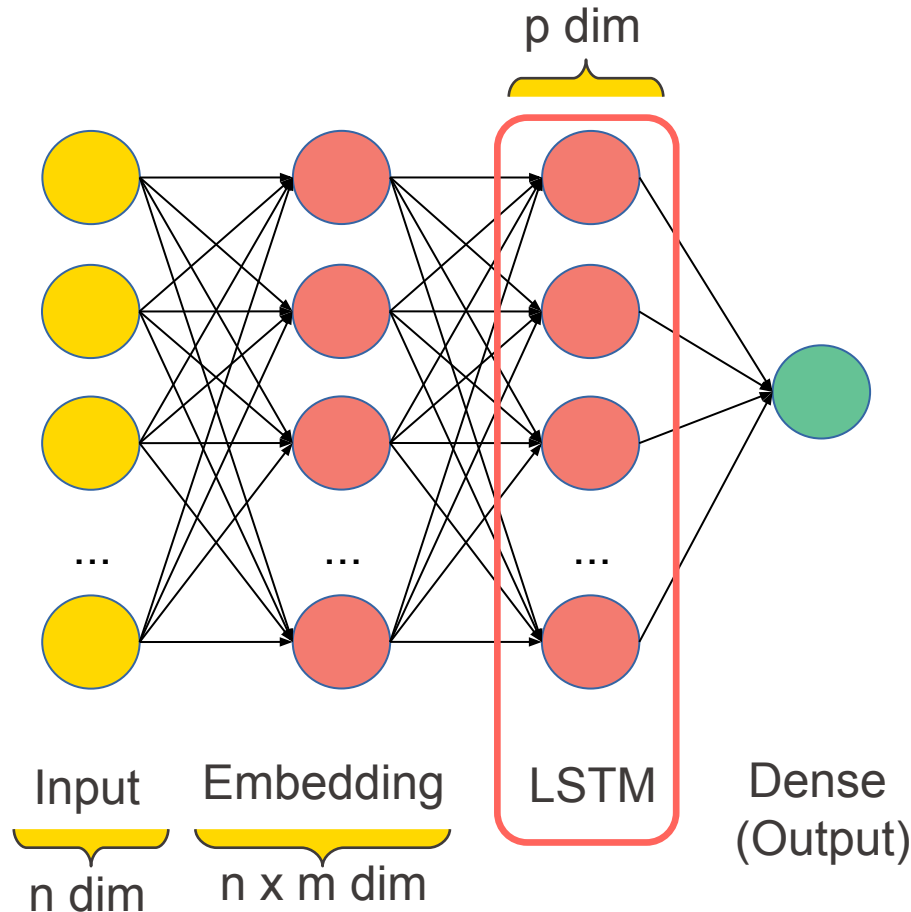
- Similar words will be embedded near each other in the vector space

# RNN (Recurrent Neural Network)



- RNN is a deep learning model
- It has the ability to memorize previous inputs
- Suitable for analysing sequential data, e.g text, time series
- However, unreliable in handling long-term memory

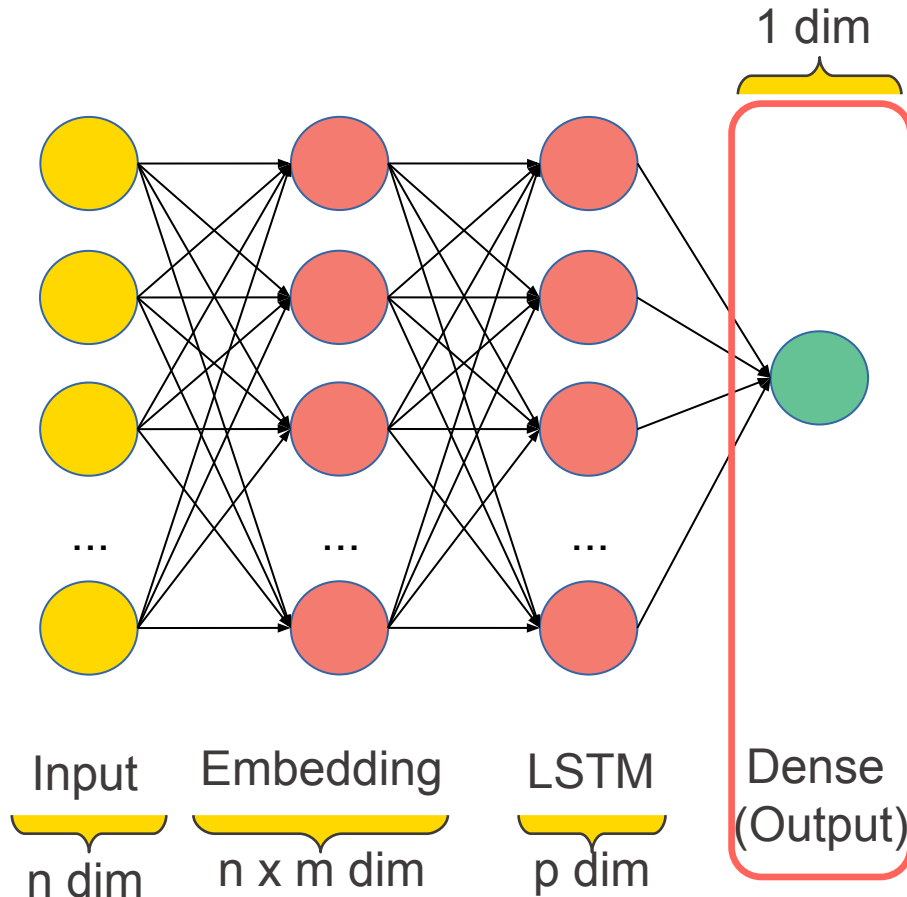
# LSTM (Long Short Term Memory) Layer



- LSTM is a variant of RNN
- Handles long-term memory
- Uses gates to control memorizing process
- Example:

$$\underbrace{n}_{\text{rows}} \left\{ \underbrace{\begin{bmatrix} [0.3, 0.2, \dots, 0.6] \\ [0.4, 0.6, \dots, 0.5] \\ [0.1, 0.8, \dots, 0.3] \end{bmatrix}}_{m} \right\} \Rightarrow \underbrace{\begin{bmatrix} 0.3 \\ 0.4 \\ 0.7 \\ \dots \end{bmatrix}}_p$$

# Dense Layer



- Dense layer connects each unit of the input with each output unit of this layer
- Defines the activation function for the final output
- Example:

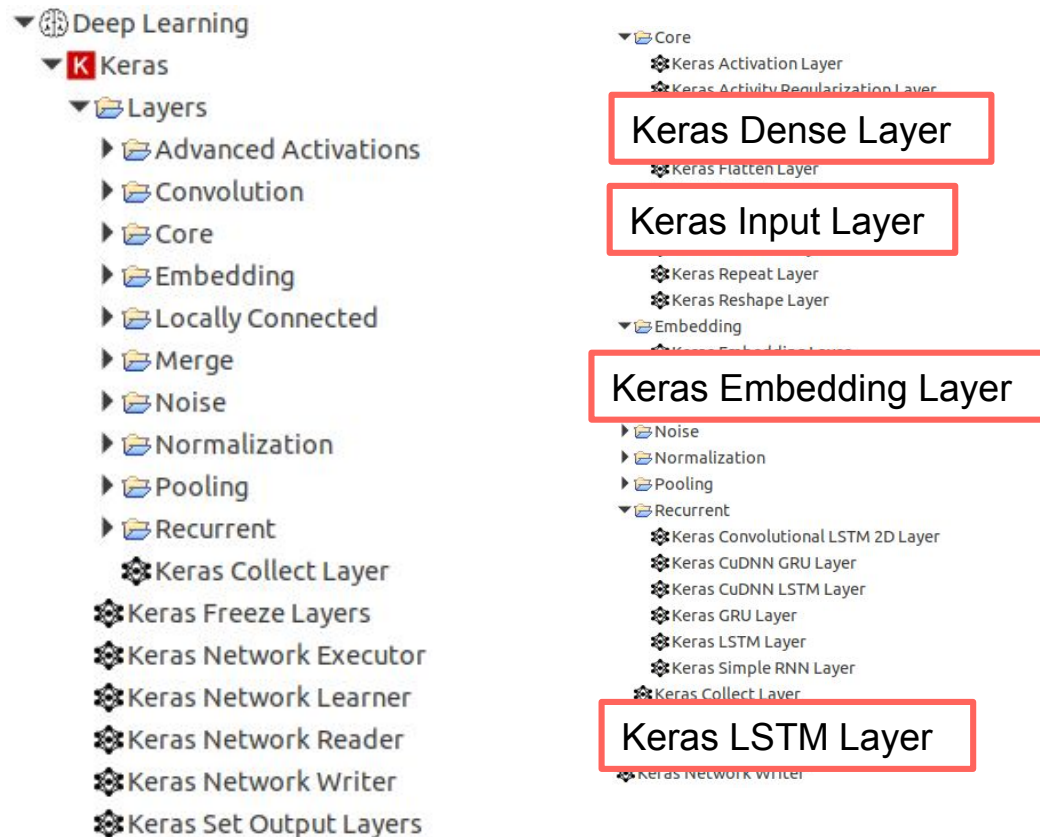
$$p \left\{ \begin{bmatrix} 0.3 \\ 0.4 \\ 0.7 \\ \dots \end{bmatrix} \right\} \Rightarrow \begin{bmatrix} 0.8 \end{bmatrix}$$

# KNIME Deep Learning

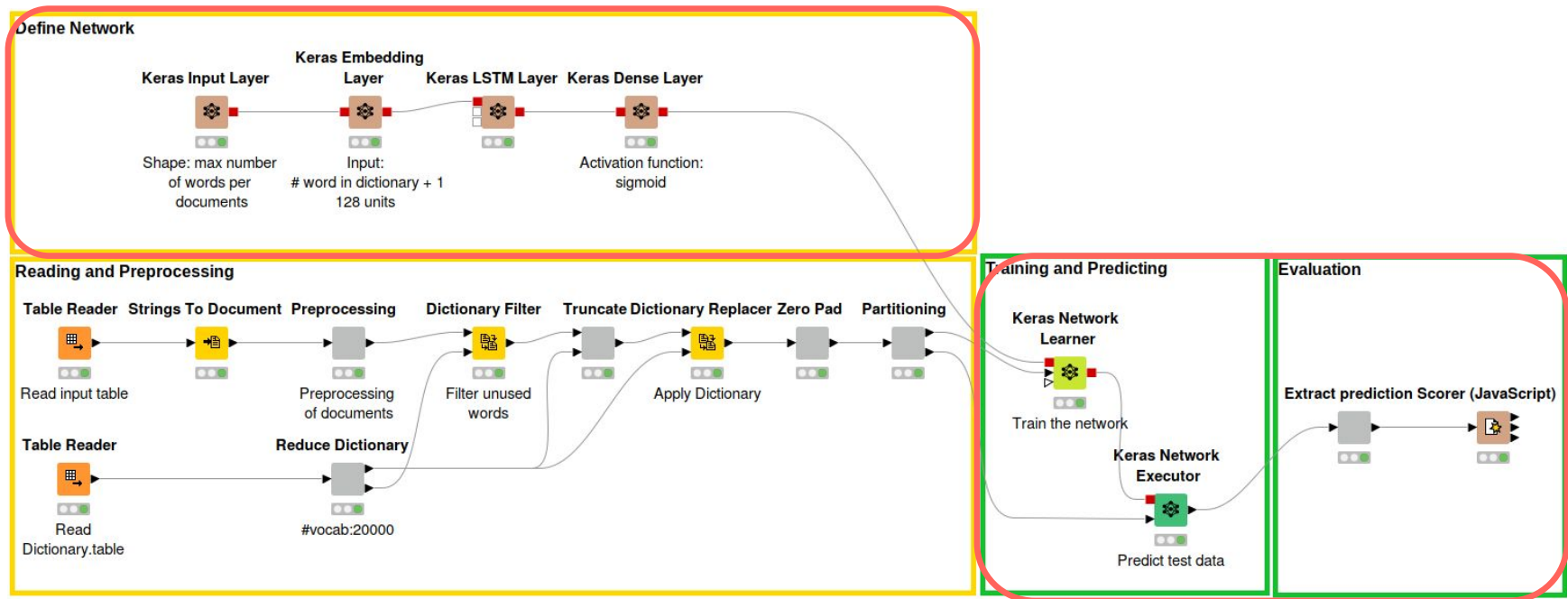
[Keras Integration](#)

[TensorFlow Integration](#)

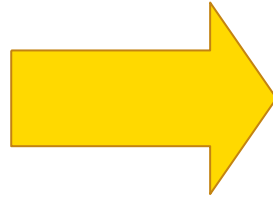
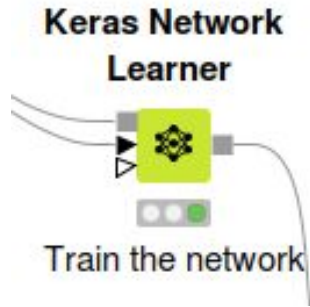
[TensorFlow 2 Integration](#)



# Part 4: Classification



# Training the Network



Dialog - 0:189 - Keras Network Learner (train for 3 epochs)

File

Flow Variables | Job Manager Selection | Memory Policy

Input Data | Target Data | Options | Advanced Options

General Settings

Back end: Keras (TensorFlow)

Epochs: 3

Training batch size: 32

Validation batch size: 32

☐ Shuffle training data before each epoch

☐ Use random seed: 1529490507276 [New seed]

Optimizer Settings

Optimizer: Adam

Learning rate: 0.001

Beta 1: 0.9

Beta 2: 0.999

Epsilon: 1.0E-8

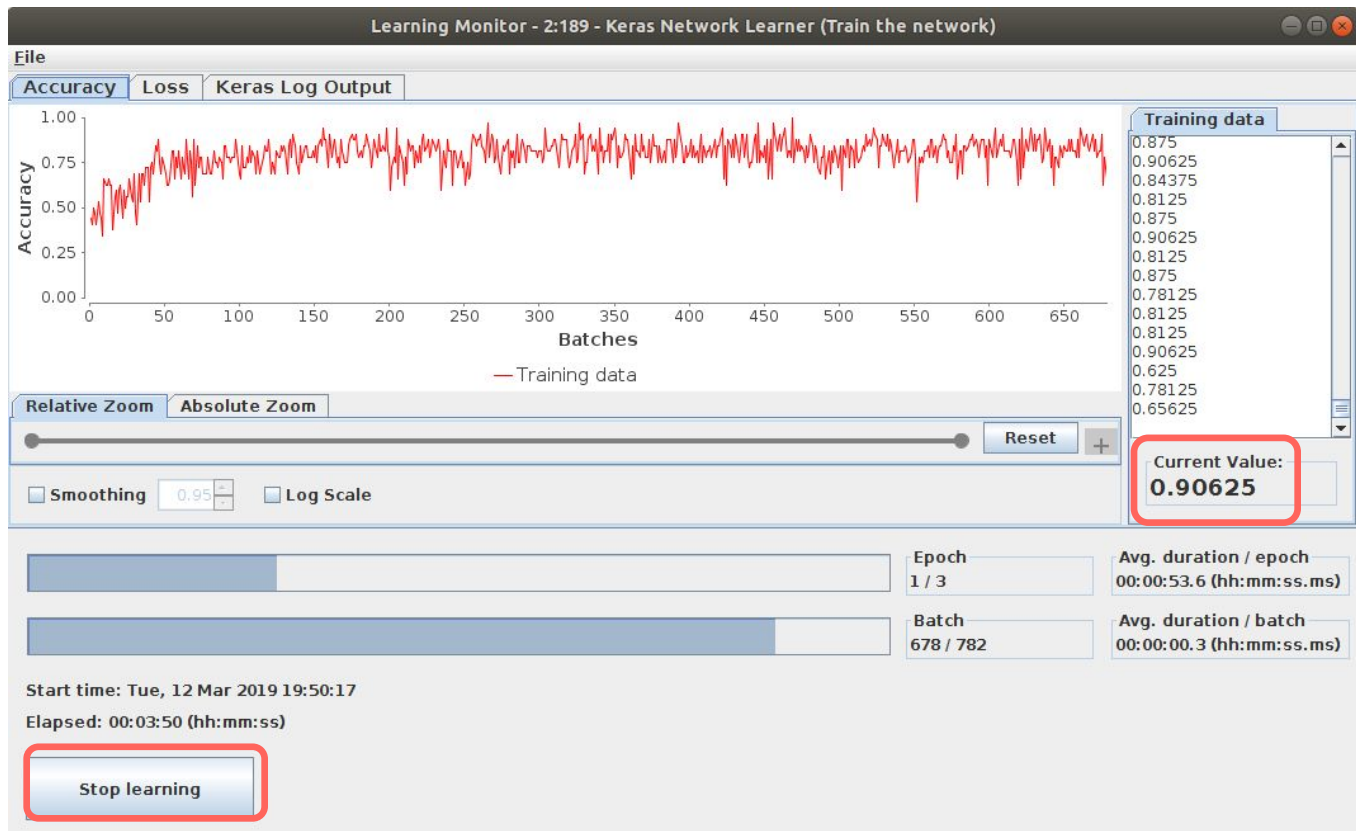
Learning rate decay: 0.0

☐ Clip norm: 1.0

☐ Clip value: 1.0

OK Apply Cancel ?

# Training the Network: Learning Monitor





# Setting up the Keras Integration

---

- Install the KNIME Keras Integration extension
  - Go to File > Install KNIME Extensions...
  - Enter keras into the search box
  - The extension is listed under KNIME Labs Extensions
- Setup Python for KNIME Deep Learning using Anaconda
  - Python environment with Keras and TensorFlow
- Please follow the installation details in the KNIME Keras Integration Installation guide:

[https://docs.knime.com/latest/deep\\_learning\\_installation\\_guide/index.html](https://docs.knime.com/latest/deep_learning_installation_guide/index.html)

# References

---

Word Embedding:

<https://www.knime.com/blog/word-embedding-word2vec-explained>

RNN/LSTM:

<https://www.knime.com/blog/text-generation-with-lstm>

Text Encoding:

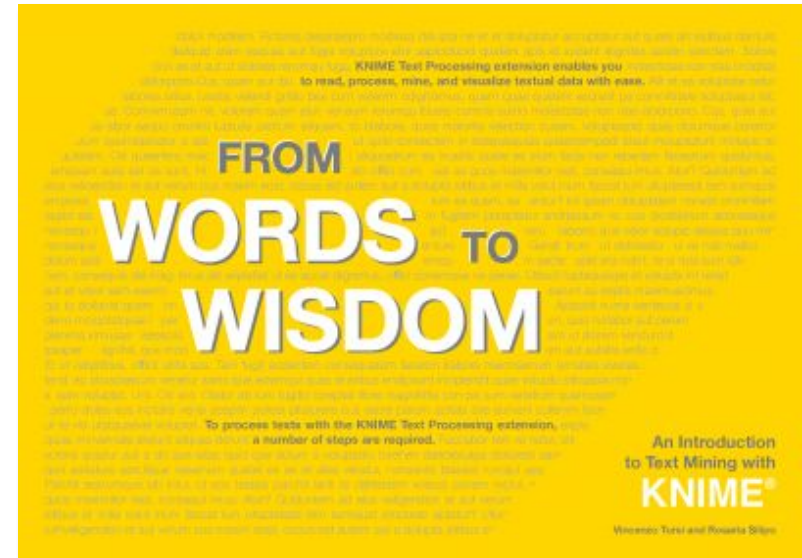
<https://www.knime.com/blog/text-encoding-a-review>

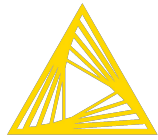
KNIME Deep Learning - Keras Integration

<https://www.knime.com/deeplearning/keras>

# KNIME Books

- Course books downloadable from **KNIME Press**  
<https://www.knime.com/knimepress>
- Get a free version with code:  
**FALL-SUMMIT-WORKSHOP**  
(valid until Jan 31, 2021)





Open for Innovation

**KNIME**

**Thank you for joining!**

