SD TSIA 214 – Deep learning for natural language processing

Chloé Clavel,
chloe.clavel@telecom-paristech.fr,

Telecom ParisTech, France

Outline of the course

Objectives of the course

At the end of this lecture,

- you will be able to explain the "philosophy" of deep learning vs. classical machine learning approaches
- you will master the ML NN architectures for NLP tasks
- you will be able to cite other neural network architectures for NLP tasks and explain their underlying principles

Problem statement

- Training dataset consisting of samples $\{xi, yi\} i = 1, N$
- xi - inputs, e.g. words (indices or vectors !), context windows, sentences, documents, etc.
- yi - labels we try to predict, e.g. other words, class : sentiment, named entities, buy/sell decision,

NLP tasks

Assigning labels to words :

- Part-Of-Speech tagging (POS),
- chunking (CHUNK),
- Named Entity Recognition (NER)
- Semantic Role Labeling (SRL)

Assigning labels to sentence/document :

- Topic classification
- opinon classification (positive vs. negative)

Classical machine learning vs. deep learning

Could speech and language processing be seen as a linear problem ?

NLP requirements

Input-output functions should solve the selectivity-invariance dilemma

- insensitive to irrelevant variations of the inputs
- very sensitive to particular minute variations of the inputs
- (for example : the pitch variation due to the speaker when you want to develop an emotion recognition system)

First option :Classical machine learning

In the simplest cases :

- linear classifiers on top of **hand-engineered features**
- A two-class linear classifier computes a weighted sum of the feature vector component
- if the weighted sum is above a threshold $\rightarrow$ choose the class

First option : Classical machine learning

With this option, the challenge is on the design of hand-engineered features

Using semantics, lexicons, etc. (see Lectures 1 and 2) in order to build feature extractor that solves the selectivity-invariance dilemma : build representations that are

- selective to the aspects of the text that are important for discrimination
- invariant to irrelevant aspects

Requires engineering skill and linguistic expertise

Second option : Deep learning

Statement

- do not use linguistic expertise and build general purpose learning procedures to automatically learn representations

Philosophy

- input : try to pre-process the features as little as possible and
- use a multilayer neural network (NN) architecture trained in an *end-to-end* fashion.
- ex : use characters as input

Second option : Deep learning

Deep learning architecture

Multilayer stack of simple modules

- subject to learning
- that computes non-linear input-output mappings
- that transform their inputs to increase both the selectivity and the invariance of the representation

Second option : Deep learning

Deep learning architecture

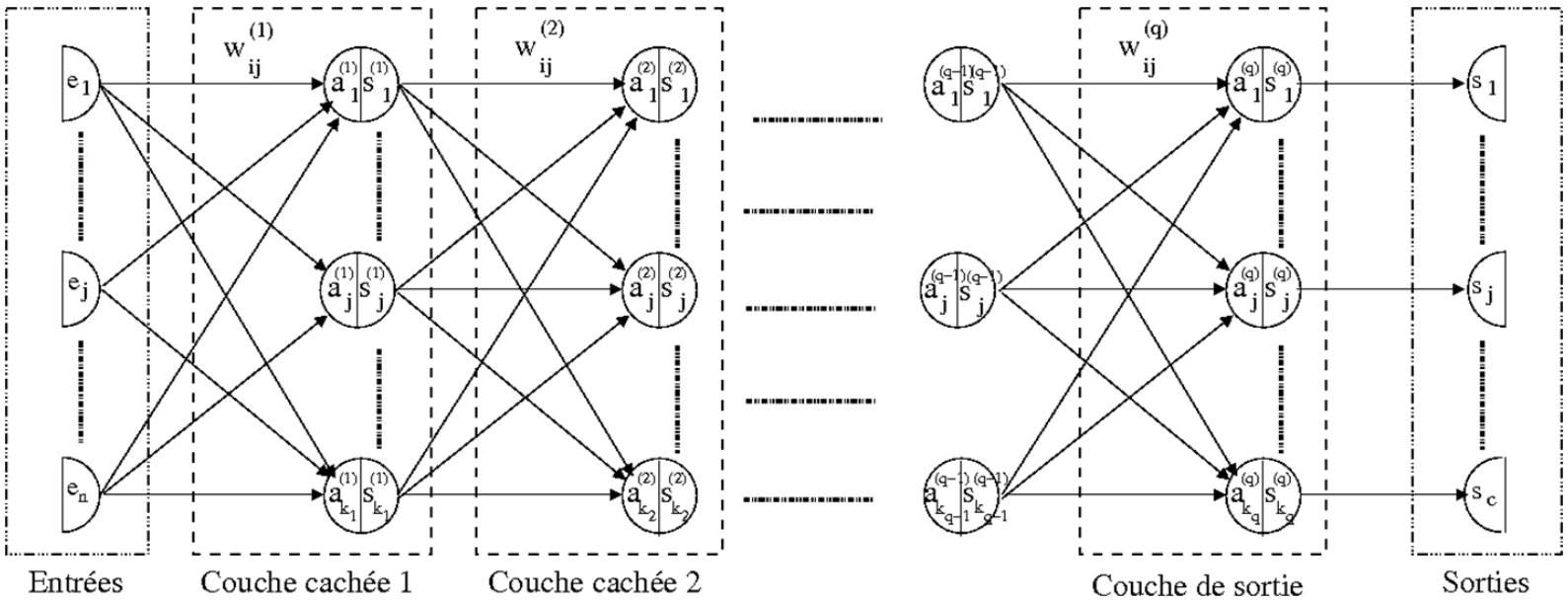
For example, with a depth of 5 to 20 non-linear layers, a system can implement extremely intricate functions of its inputs that are simultaneously sensitive to minutes details and insensitive to large irrelevant variations

Multilayer Neural Networks - ML NN

1. Use for NLP
2. Inputs
3. Outputs
4. Layers and backpropagation

ML Neural networks principles

- A multilayer neural network can distort the input space to make the classes of data linearly separable
- If the weights are set correctly, a neural network with enough neurons and a non-linear activation function can approximate a very wide range of mathematical functions



ML NN use for NLP

- For binary classification problems
- For multiclass classification problems
- More complex structured prediction problems

Advantages : The non-linearity of the network, as well as the ability to easily integrate pre-trained word embeddings, often lead to superior classification accuracy.

ML NN use for NLP

Examples :

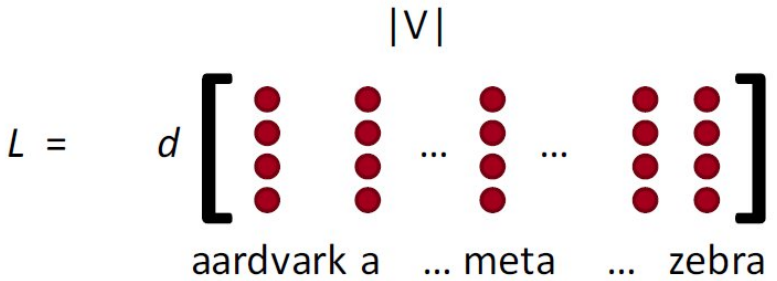
- Syntactic parsing : Chen, D., & Manning, C. (2014). A Fast and Accurate Dependency Parser using Neural Networks. EMNLP 2014
- Dialog state tracking : Henderson, M., Thomson, B., & Young, S. (2013). Deep Neural Network Approach for the Dialog State Tracking Challenge. Sigdial 2013

ML NN Inputs

Reminder from Lecture 2b about word embeddings

INPUT : words are represented as indices taken from a finite dictionary $\mathcal{D}$

OUTPUT : Lookup table feature vector



Conceptually you get a word's vector by left multiplying a one-hot vector e by L

ML NN Inputs

Option 1

- Use pre-trained word vectors (the best to do : if you have a small training dataset). Example :
  - → In Valentin Barriere, Chloé Clavel, Slim Essid : « Attitude Classification in Adjacency Pairs of a Human-Agent Interaction with Hidden Conditional Random Fields », ICASSP 2018
  - → we had about 500 utterances and we use representations learnt from a Google News corpus of 100 billions words `https://code.google.com/archive/p/word2vec/`

ML NN Inputs

Option 2

- Train your word vectors on your database in an unsupervised manner (the best to do : if you have a big dataset with peculiarities). Example :
  - → In Maslowski, I., Lagarde, D., Clavel, C., In-the-wild chatbot corpus from opinion analysis to interaction problem detection, ICNLSSP 2017
  - → we train word2vec on 1,813,934 dialogues.

ML NN Inputs

Option 3

Re-train vectors for your task (the best to do : if you have a big **labelled** dataset)
How to train multilayer neural network (NN) architecture, in an end-to-end fashion ?
STEP 1 : The architecture takes the input sentences and learns several layers of feature extraction that process the inputs.
STEP 2 : The features computed by the deep layers of the network are automatically trained by backpropagation to be relevant to the task.

Window approach

Starting from an example : input : "Museums in Paris are amazing"
output : "O O B_LOC O O"
The output for "Paris" depends on its context of occurrence ("Paris Hilton" will be a person)
$\rightarrow$ build a context window : e.g. we represent each word using a 4-dimensional word vector and we use a 5-word window (the previous 2 and the following 2) as input (as in the above example), then the input $x \in \mathbb{R}^{20}$.

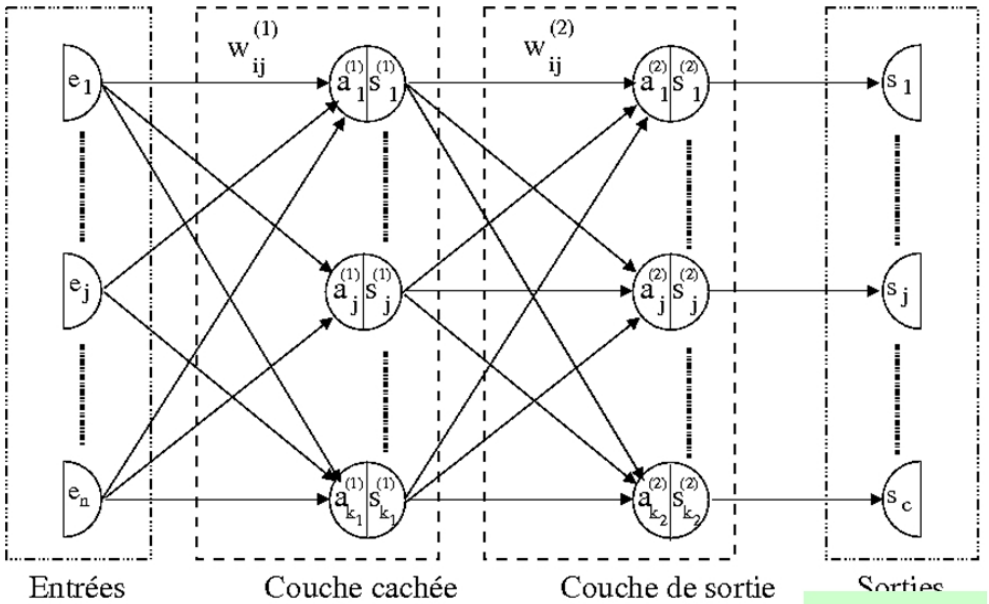
ML NN outputs

Case where the dimension of the outputs $d_{out} = 1$ which means that the network's output is a scalar.
Such networks can be used

- for regression (or scoring) by considering the value of the output
- for binary classification by consulting the sign of the output.

ML NN outputs

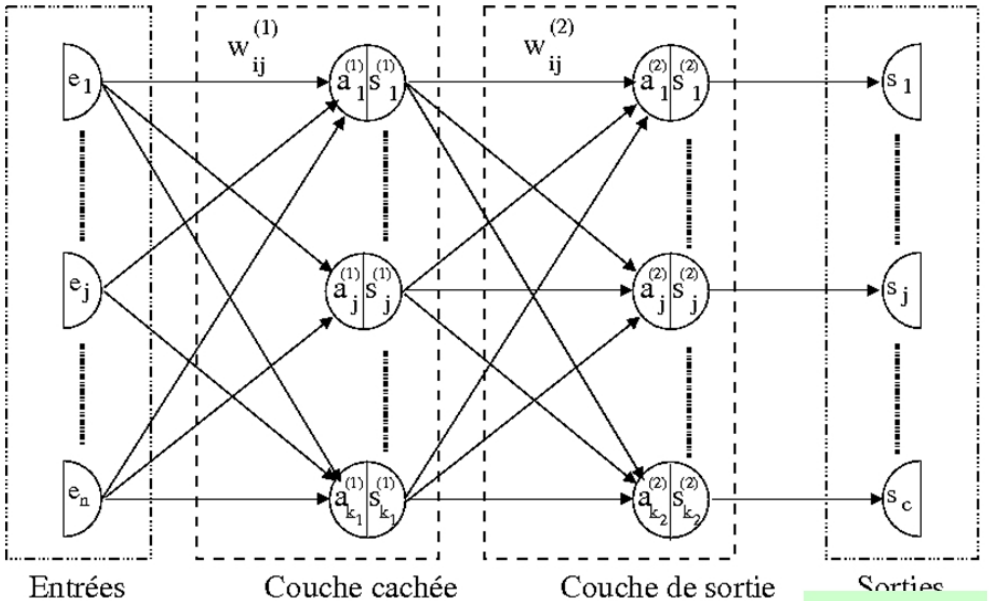
Networks with $d_{out} = c > 1$ can be used for k-class classification, by associating each dimension with a class, and looking for the dimension with maximal value.



ML NN outputs

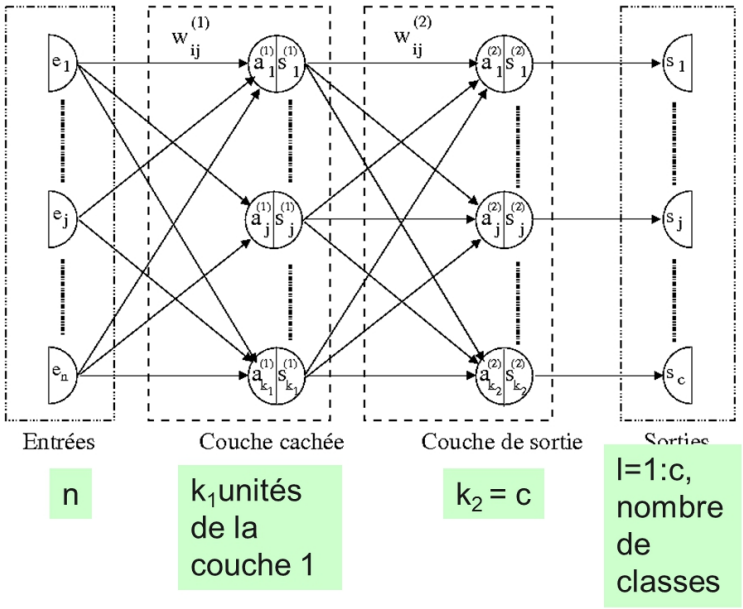
Similarly, if the output vector entries are positive and sum to one, the output can be interpreted as a distribution over class assignments (such output normalization is typically achieved by applying a softmax transformation on the output layer).

ML NN Layers



Hidden layer : between the inputs and the outputs there are layers with hidden outputs

ML NN Layers



FORWARD : we need to compute all the outputs of the $m-1$ layer to compute the outputs of the m layer.
PRACTICE in the case of two layers : try to compute the final outputs

Training and backpropagation algorithm

1. define the loss
2. compute partial derivatives
3. apply gradient descent algorithm from output layers to input layers

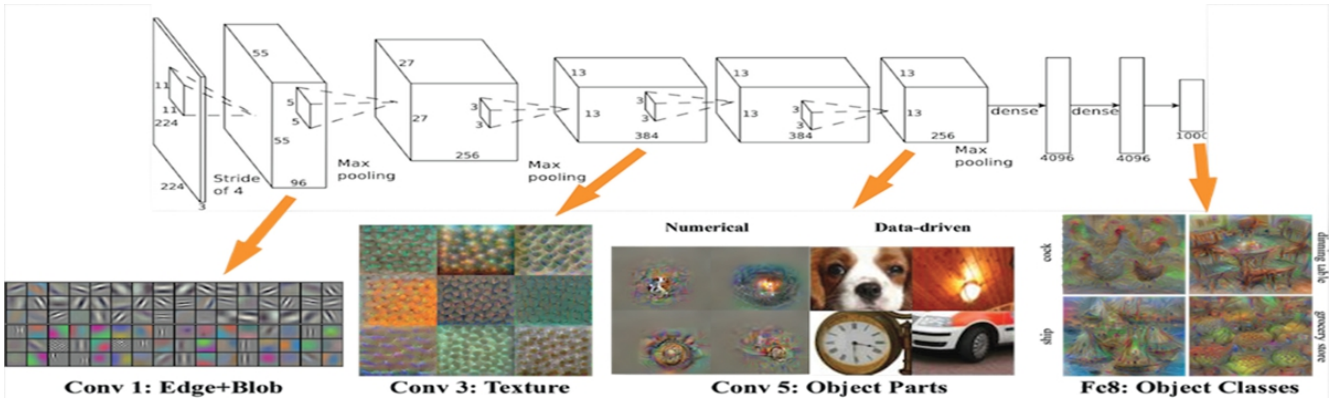
See lecture Neural Networks SDTSIA 210

Other NN architectures

- Convolutional neural networks
- Recursive deep models
- Recurrent neural networks and variants

Convolutional Neural Networks

- Variation of multilayer perceptrons designed to require minimal preprocessing and using *convolutional* layers
- the network learns the filters



Convolutional Neural Networks

Example of use for the text : Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks.

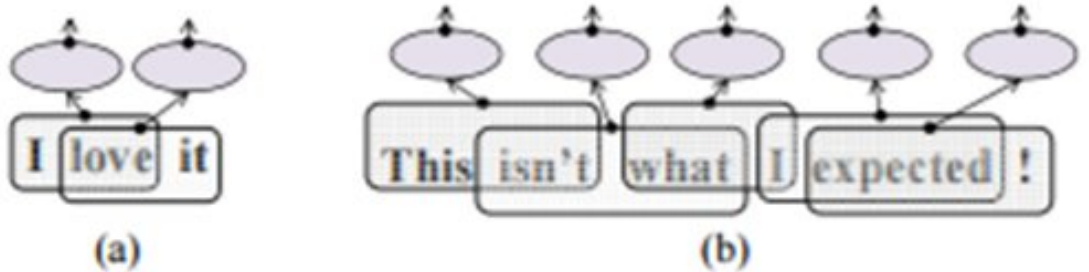
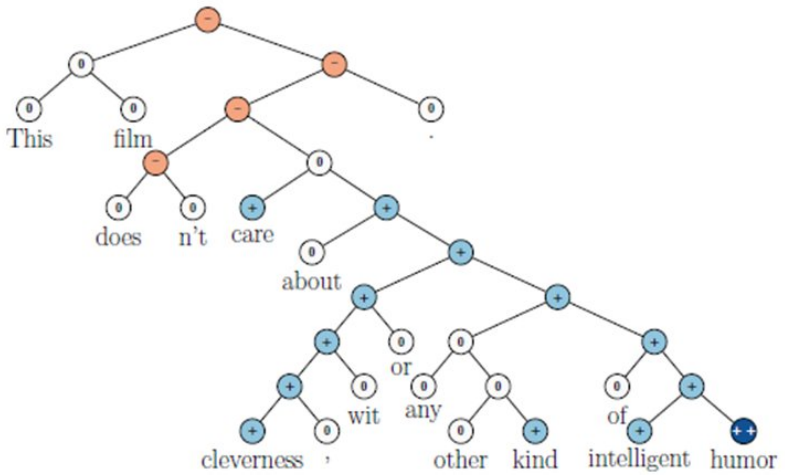


Figure 3: Convolution layer for variable-sized text.

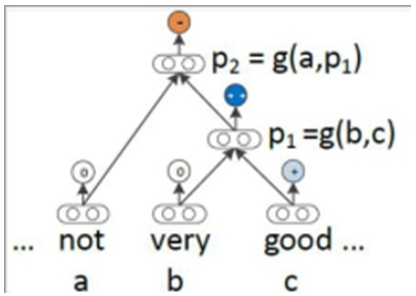
Recursive deep models

Tree representation of movie sentences using Stranford parser
Each node of the tree is labelled in (-, +,0) to provide the structure that is required for the training of a recursive model (Sentiment TreeBank Database)



Recursive deep models

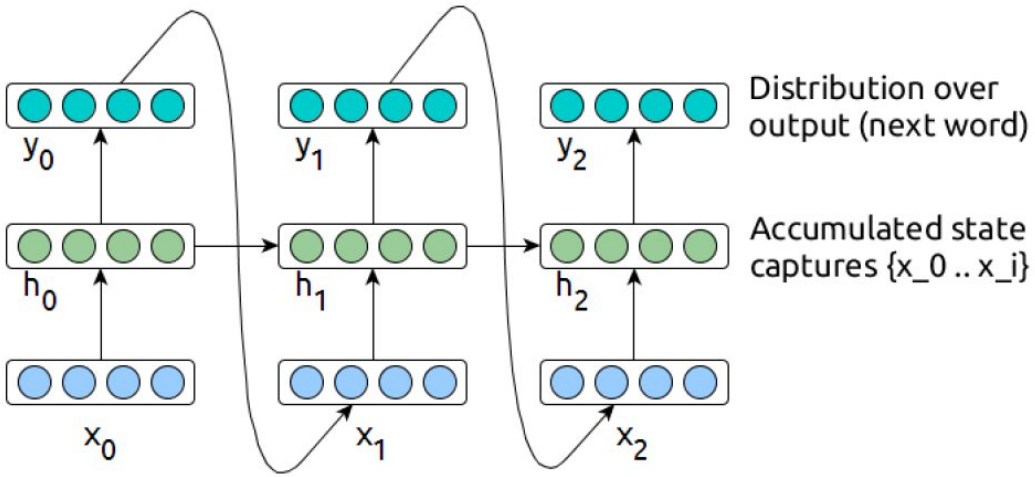
Training step : learning g function that compute the upper outputs in the binary tree



REF : R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, EMNLP 2013.

Recurrent Neural Networks

Use for language models



- Reads inputs xi to accumulate state hi and predict outputs yi
- Variants : LSTM networks (Long Short Term Memory Networks), RNN using gating mechanisms such as GRU (Gated Recurrent Units)

Support and materials

- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015) : 436.
- Lectures from Stanford `http://cs224d.stanford.edu/lectures/CS224d-Lecture4.pdf`
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), 2493-2537.
- Goldberg, Yoav. "A primer on neural network models for natural language processing." Journal of Artificial Intelligence Research 57 (2016) : 345-420.
- Lectures from Oxford : `https://github.com/oxford-cs-deepnlp-2017/lectures`