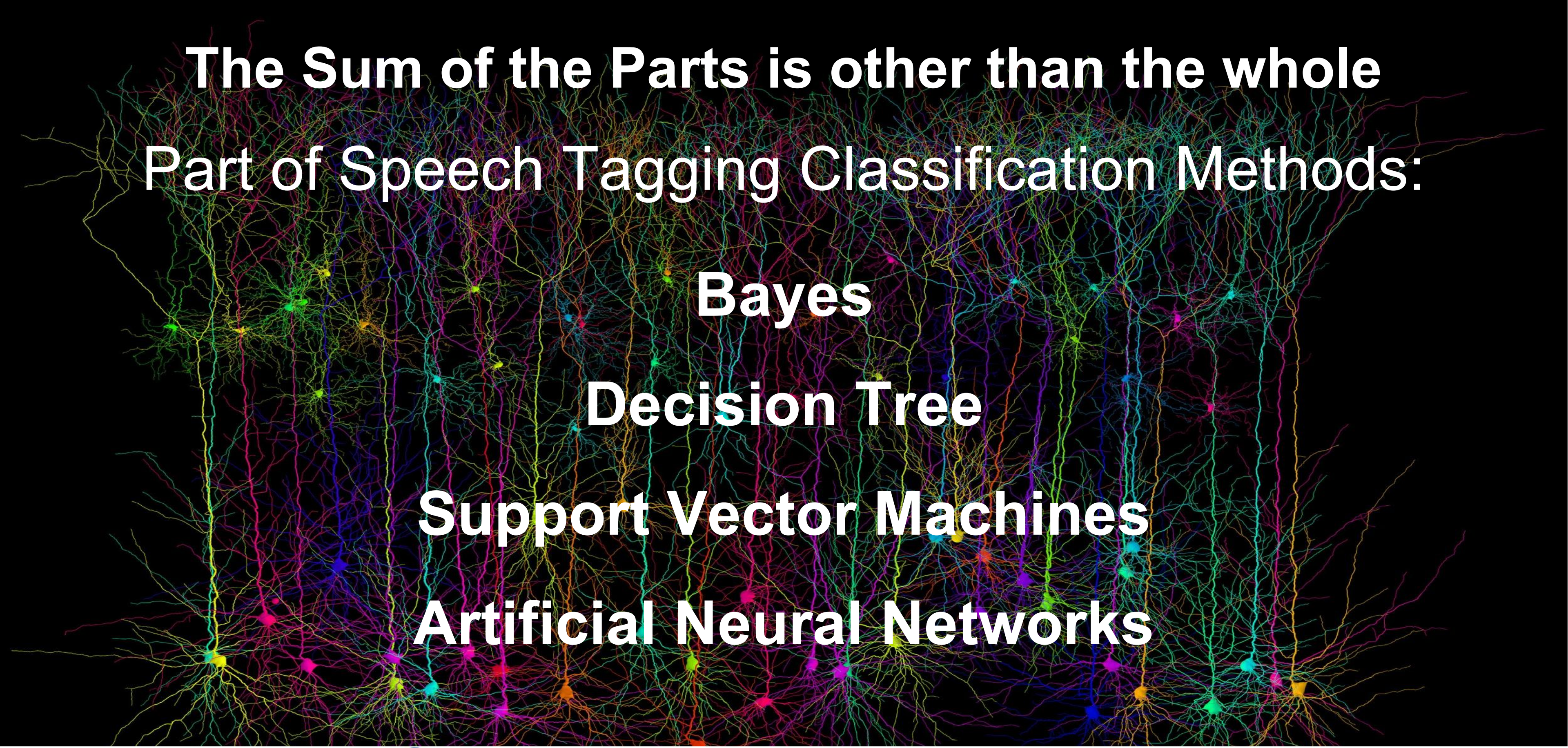
The Sum of the Parts is other than the whole

by:
Francesco Mantegna
Patrizio Bellan

Center for Mind/Brain Sciences
UNIVERSITY OF TRENTO - Italy
CIMeC



The Sum of the Parts is other than the whole

Part of Speech Tagging Classification Methods:

Bayes

Decision Tree

Support Vector Machines

Artificial Neural Networks

The Sum of the Parts is other than the whole

Walkthrough:

Bianchi D., Delmonte R. (2002), Tecniche di apprendimento applicate al problema del tagging: una prima valutazione per l'Italiano, Convegno Nazionale AI*IA, Siena, pp.20-34.

Cimino A., Dell'Orletta F. (2016) “*Building the state-of-the-art in POS tagging of Italian Tweets* “. In Proceedings of EVALITA ’16, Evaluation of NLP and Speech Tools for Italian, 7 December, Napoli, Italy.

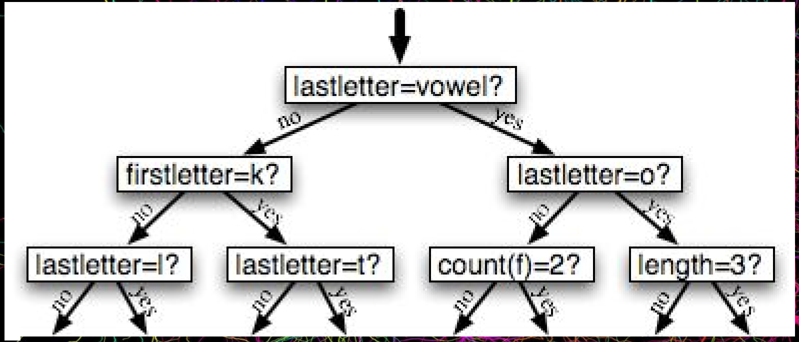
Tamburini F. (2016). (Better than) State-of-the-Art PoS-tagging for Italian Texts. In Proc. *3rd Italian Conference on Computational Linguistics - CLiC-IT 2016*, Napoli, 5-6 December 2016, 280-284.

The Sum of the Parts is other than the whole

Decision Tree

- A decision tree is a **flowchart-like structure** in which each leaf node represents a class label
- The paths from root to leaf represent **classification rules**

- The decision tree is a set of ***decision rules*** in **if-statement form.**
  if condition1 and condition2 and condition3 then outcome



$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

The Sum of the Parts is other than the whole

Decision Tree an example of decision rules

```
if final-3 == 'a': return 'PRE'
if final-3 == 'aar': return 'NOU'
if final-3 == 'aba':
if first-3 == 'ara': return 'ADJ'
if first-3 == 'end': return 'ADJ'
if first-3 == 'fia': return 'NOU'
if first-3 == 'mon': return 'ADJ'
if first-3 == 'sil': return 'VER'
if final-3 == 'abe':
        if nVowels == 3: return 'NOU'
        if nVowels == 5: return 'ADJ'
        if nVowels == 6: return 'ADJ'
if final-3 == 'abi': return 'VER'
```

The Sum of the Parts is other than the whole

D. Bianchi, R. Delmonte, 2002

Compared three different supervised learning techniques for (Italian) POS tagging classification:
Decision trees, Neural Networks, Genetic Programming

Strength:

- focus on ambiguity

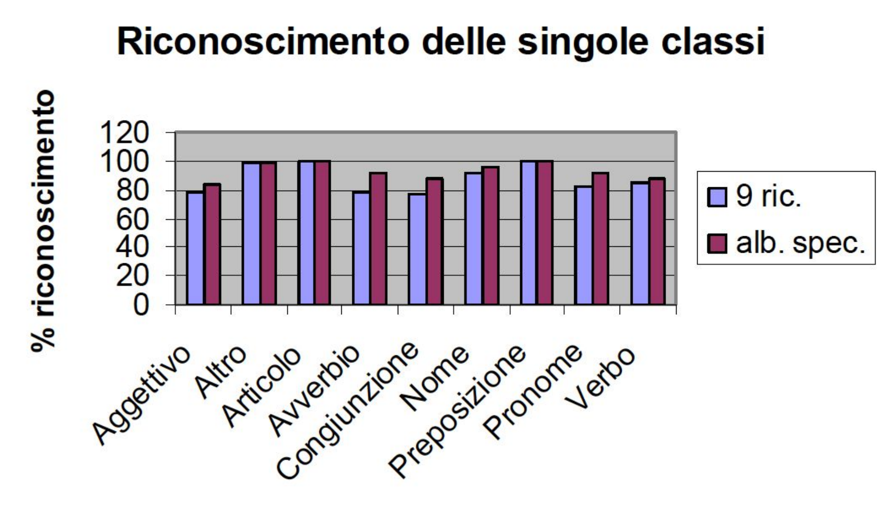
Drawbacks:

- uses only decision tree binary classifiers (e.g. VRB vs. PRN)
- separates straightforward and ambiguous cases in the training

The Sum of the Parts is other than the whole

D. Bianchi, R. Delmonte, 2002

	accuracy	type
Dec.Tree	70.0%	unique
Dec.Tree	72.9%	binary
Dec.Tree	82.9%	special.
NN	79.2%	
Genetic p.	54.6%	binary
Genetic p.	81.2%	special.

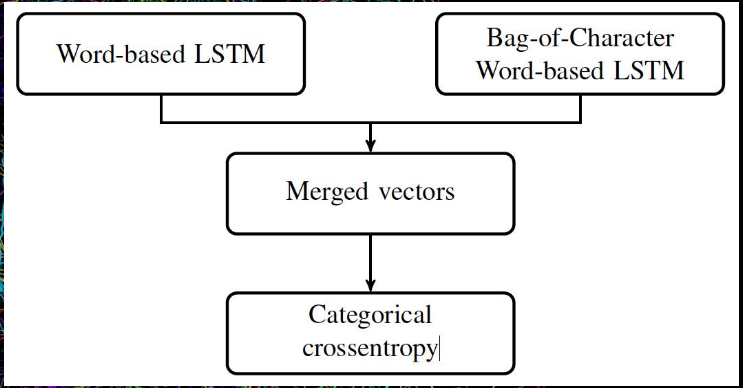


The Sum of the Parts is other than the whole

A. Cimino, F. Dell'Orletta, 2016

Developed a two-branch bidirectional Long Short Term Memory recurrent neural network.

- **Word-based bi-LSTM** : Word Embedding (i.e. word2vec, fastText), Morpho-syntactic category, spell checker, word length, URL, uppercase, capitalized, end-of-sentence
- **Bag-of-Character bi-LSTM** : Characters, lowercase characters, numbers, alphanumeric, alphabetic



The Sum of the Parts is other than the whole

Configuration	Devel	Test
Single bi-LSTM	**96.39**	**93.67**
No handcrafted features	95.22	91.99

Configuration	Devel	Test
Two-branch bi-LSTM	**96.55**	93.19
Word bi-LSTM	96.03	92.35
Bag-of-Char. Word bi-LSTM	84.47	80.77
No Morpho-syntactic lexicon	96.48	**93.54**
No spell checker	96.49	93.31
No word2vec lexicons	93.23	89.87
No fastText lexicon	95.85	92.43
No feature engineering	96.39	93.06

Table 1: Tagging accuracy (in percentage) of the different learning models on our development set and the official test set.

A. Cimino, F. Dell'Orletta, 2016

Based on model components testing:

- The Word-based bi-LSTM is clearly the best performer with respect to the Bag-of-Character one
- **Morpho-syntactic lexicon information** gives a negligible improvement on the training set and unexpectedly a slight drop on the test set.
- The spell checker do not contribute in increasing the tagging performances
- The results show that word2vec seems to be a better choice with respect to fastText (fastText was expected to be particularly useful for the analysis of non standard text such as social media ones)
- Handcrafted features yield an improvement of 1.34% and 1.68% on the training and the test sets respectively

The Sum of the Parts is other than the whole

F. Tamburini, 2016

Morphological features

Having a restricted list of possible tags for a single word-form enable the tagger to reduce the search space and force it to take reasonable decisions.

Powerful **morphological analysers** based on large lexica are invaluable resources to increase tagger accuracy.

In this paper, the word embeddings computed in a completely unsupervised way (i.e. word2vec) was extended by concatenating to them a vector containing the possible PoS-tags provided by the **AnIta analyser**.

The Sum of the Parts is other than the whole

F. Tamburini, 2016

SYSTEM	TA		Notes
	E07	E09	
MLP-256	96.45	95.57	Win=5
MLP-256	97.75	96.84	M,Win=5
2-BiLSTM-256	98.12	97.30	M,Win=5
2-BiLSTM-256	98.14	97.45	M,Seq
2-BiLSTM-256-CRF	**98.18**	**97.48**	M,Seq

Table 2: Tagging accuracies (TA) for different configurations for both datasets. ('M' marks the use of AnIta morphological information).

Two different ways of structuring the input features for processing were used:

- **Win**: based on a sliding window that starts from the beginning of each sentence and concatenates word feature vectors into one single vector.
- **Seq**: each sentence is managed as one single sequence

The information from AnIta proved to be crucial to reach such accuracy
values as well as stacked BiLSTM networks processing entire sentence sequences.

The Sum of the Parts is other than the whole

Bayes:

Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event

Bayes' theorem then links the degree of belief in a proposition before and after accounting for evidence, and it measures a "degree of belief"

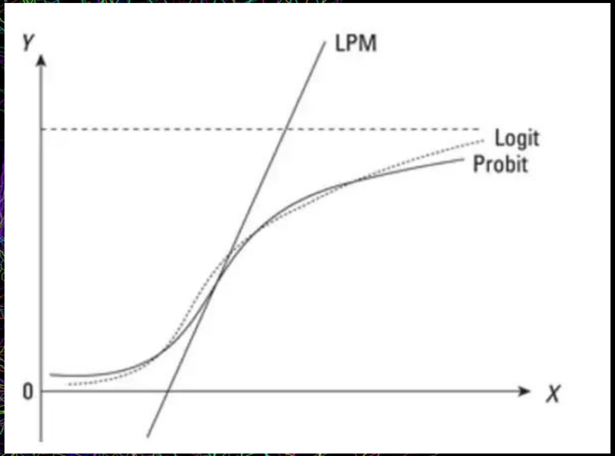
$$\Pr(A|X) = \frac{\Pr(X|A)\Pr(A)}{\Pr(X|A)\Pr(A) + \Pr(X|\text{not } A)\Pr(\text{not } A)}$$

The Sum of the Parts is other than the whole

Logistic Regression

The logistic distribution function approaches 0 and 1 asymptotically, so Y values stay within the [0,1] range.
It is used to estimate the probability of a response based on one or more predictive variables (features).

Such gradient ascent methods start with a zero weight vector and move in the direction of the gradient, LPM(w), the partial derivative of the objective function with respect to the weights.

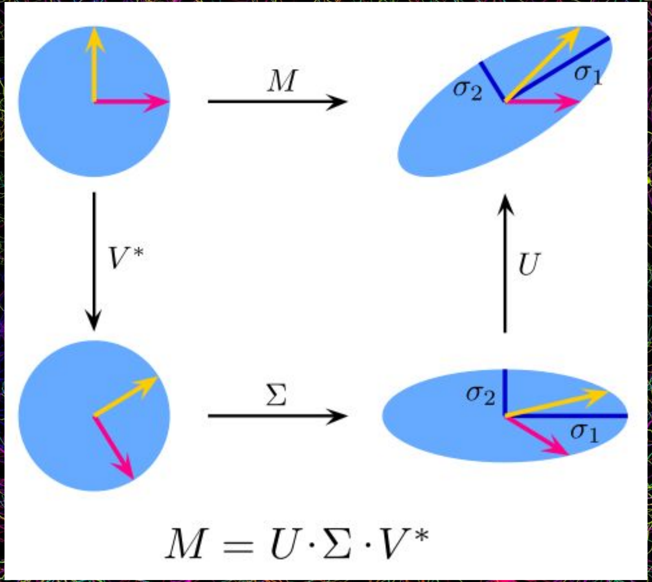


Logit: $Pr(Y=1|X)=[1+e-X'\beta]-1Pr(Y=1|X)=[1+e-X'\beta]-1$
Probit: $Pr(Y=1|X)=\Phi(X'\beta)Pr(Y=1|X)=\Phi(X'\beta)$

The Sum of the Parts is other than the whole

Singular Value Decomposition

(0, 4)	1.0
(0, 195)	1.0
(0, 329)	1.0
(0, 1606)	1.0
(0, 3661)	3.0
(1, 12)	1.0
(1, 34)	1.0
(1, 1205)	1.0
(1, 1795)	1.0
(1, 3186)	1.0
(1, 3659)	10.0
(2, 219)	1.0
(2, 343)	1.0
(2, 1630)	1.0
(2, 2025)	1.0
(2, 3659)	13.0
:	:
(266284, 1090)	1.0
(266284, 1597)	1.0
(266284, 3659)	10.0
(266284, 3660)	5.0
(266284, 3661)	5.0
(266285, 1038)	1.0
(266285, 1718)	1.0
(266285, 2632)	1.0
(266285, 3659)	11.0
(266285, 3660)	6.0
(266285, 3661)	5.0
(266286, 1606)	1.0
(266286, 1849)	1.0
(266286, 3560)	1.0
(266286, 3659)	15.0
(266286, 3660)	8.0
(266286, 3661)	7.0




```
[[  7.34181234   0.54139197]
 [ 12.32587683  -0.61286404]
 [ 15.9514927     0.57147987]
 ...,
 [ 12.22970377   0.89289463]
 [ 13.51237019   0.33121498]
 [ 18.38270152   0.62317202]]
```

The Sum of the Parts is other than the whole

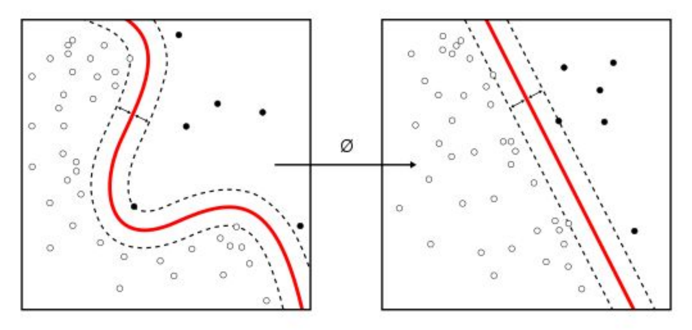
Support Vector Machine

A kernel is a similarity function

It takes inputs and returns how similar they are
computing the kernel is easy, but computing the feature vector corresponding to the kernel is really really hard

Linear kernel $K(x, y) = \langle f(x), f(y) \rangle$

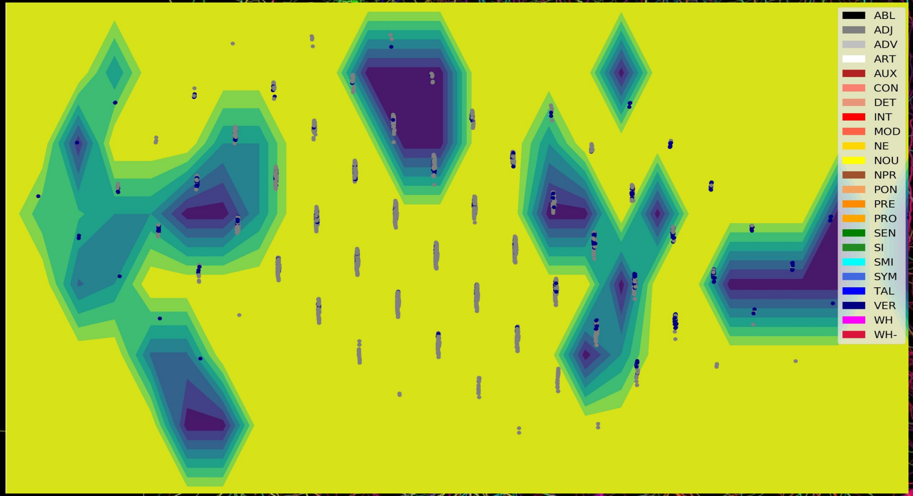
RBF kernel $(k(x,y) = \exp(-||x-y||^2)$



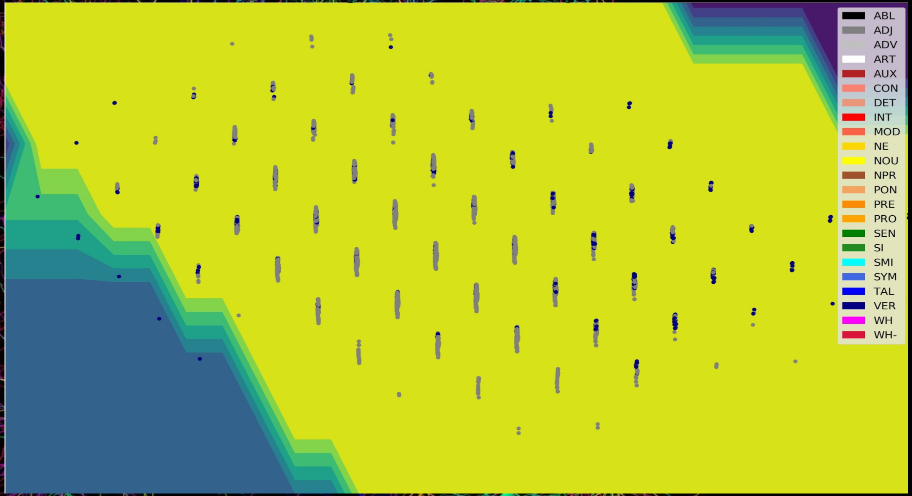
The Sum of the Parts is other than the whole

Support Vector Machine

rbf



linear



The Sum of the Parts is other than the whole

How to vectorize a word and deal with sparsity?

bigramize it!

and what about the pos tag vector?

One Hot Vector

The Sum of the Parts is other than the whole

example of the vector of the word ‘casa’

[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

and its pos tag in ‘one hot vector’ form

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

The Sum of the Parts is other than the whole

Important concepts:

Activation function

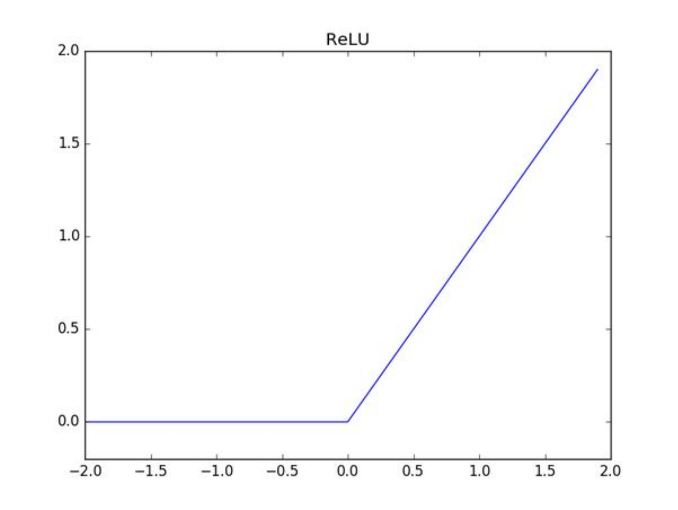
SoftMax

Loss

The Sum of the Parts is other than the whole

Rectified Linear Unit (ReLU)

activation function



The Sum of the Parts is other than the whole

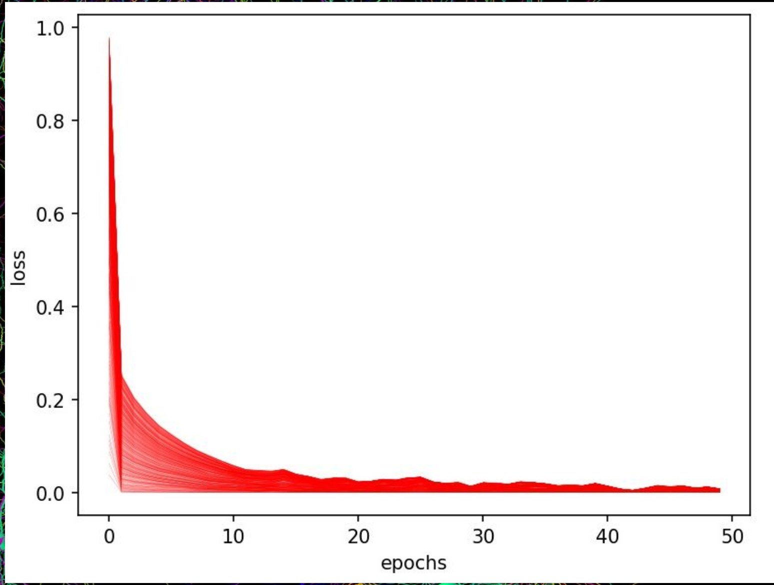
SoftMax

- *Classification problems have the advantage that the classes are mutually exclusive*
- Used in the ***final layer*** of a neural network-based classifier, they give a non-linear variant of multinomial logistic regression
- Used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables

The Sum of the Parts is other than the whole

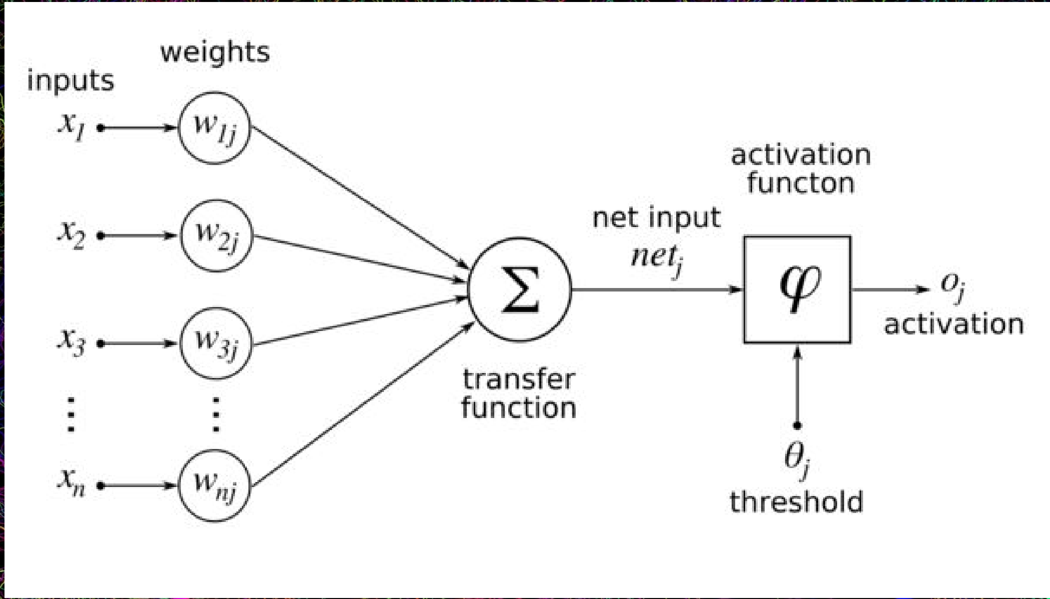
Loss

The ***"loss"*** (or cost)
is the cost associated
with the difference between the
prediction in the actual state
of the neural network and
the correct values



The Sum of the Parts is other than the whole

Artificial Neural Network



The Sum of the Parts is other than the whole

ours Results - Mathematical Models

	Accuracy
Bayes	0.888
Decision Tree	0.9004
Logistic Regression (regul.=1)	0.9146
Logistic Regression (regul.=10)	0.9074
Logistic Regression (regul.=100)	0.9036
Logistic Regression (regul.=1000)	0.9004

The Sum of the Parts is other than the whole

ours Results - Support Vector Machine

	Linear Kernel	rbf Kernel
regul = 1 Gamma = 1	0.83220	0.845949
regul = 1 Gamma = 10	0.83220	0.846673
regul = 1 Gamma = 100	0.83220	0.847878
regul = 10 Gamma = 1	0.83220	0.846190
regul = 10 Gamma = 10	0.83220	0.848360
regul = 10 Gamma = 100	0.83220	0.849083

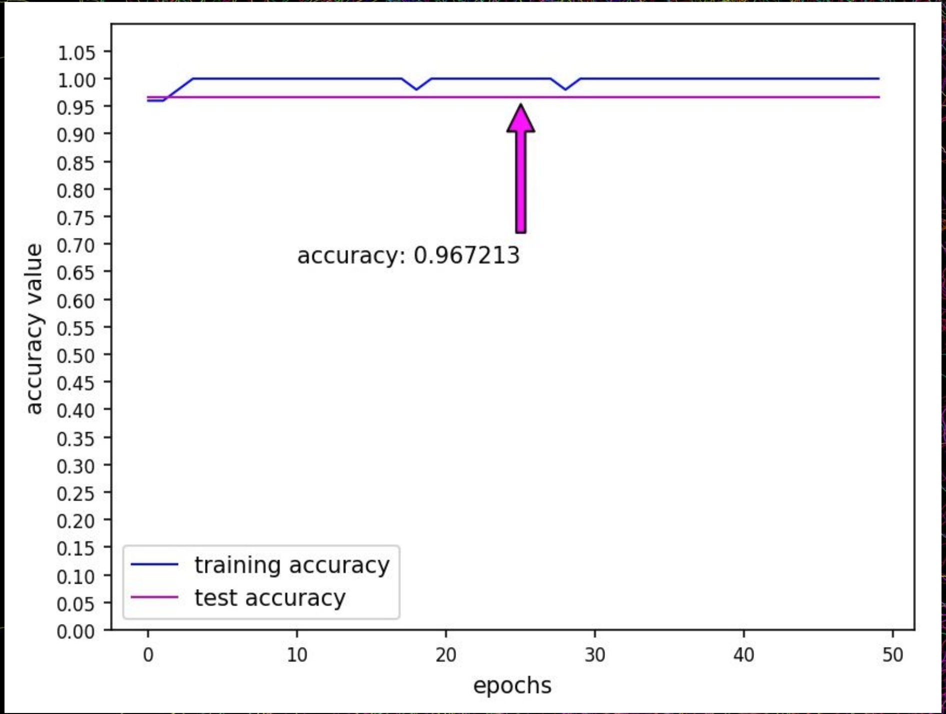
	Linear Kernel	rbf Kernel
regul = 100 Gamma = 1	0.83220	0.8457087
regul = 100 Gamma = 10	0.83220	0.8488428
regul = 100 Gamma = 100	0.83220	0.8488428
regul = 1000 Gamma = 1	0.83220	0.8478784
regul = 1000 Gamma = 10	0.83220	0.8481195
regul = 1000 Gamma = 100	0.83220	0.8459498

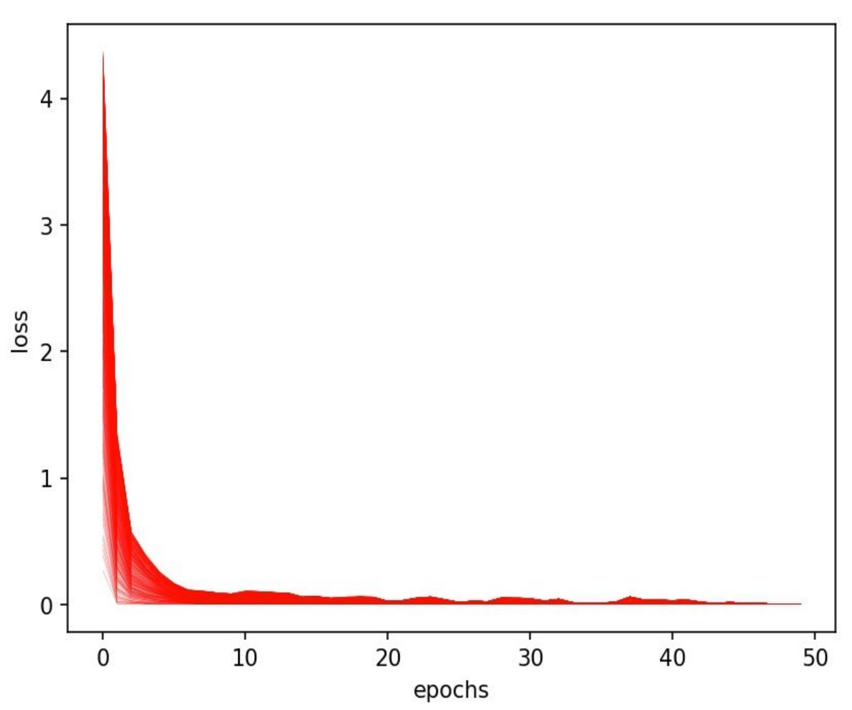
The Sum of the Parts is other than the whole

ours Results - Artificial Neural Network

	1 hidden layer	2 hidden layer	3 hidden layer
Accuracy	0.967213	0.951543	0.844021
Neurons first layer	500	250	250
Neurons second layer	0	125	125
Neurons third layer	0	0	25
starting learning rate	1e-02	e-02	e-02

The Sum of the Parts is other than the whole





The Sum of the Parts is other than the whole

Future directions

create a convolutional artificial neural network which predicts the Part of Speech feeded within a context window