Advanced machine learning models for NLP

Fabio A. González

MindLab Research Group - Universidad Nacional de Colombia

Natural Language Processing and Text Mining Course
10$^{\text{ma}}$ Cátedra Internacional de Ingeniería

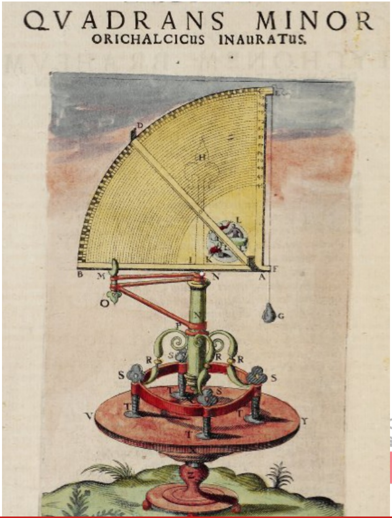
Outline

Outline

Observation and analysis

Tycho Brahe

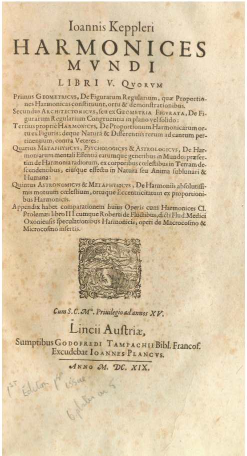

TYCHONIS BRAHE
ASTRONOMIÆ
INSTAURATÆ
MECHANICA
WANDESBURGI
ANNO


QVADRANS MINOR
ORICHALCICUS INAURATUS.

Tycho Brahe

	Date, Old Style					Longitude				Latitude			Mean Longitude			
	Year	Day	Month	H	M	D	M	S	Sign	D	M		S	D	M	S
I	1580	18	November	1	31	6	28	35	Gemeni	1	40	N.	1	25	49	31
II	1582	28	December	3	58	16	55	30	Cancer	4	6	N.	3	9	24	55
III	1585	30	January	19	14	21	36	10	Leo	4	32	N.	4	20	8	9
IV	1587	6	March	7	23	25	43	0	Virgo	3	41	N.	6	0	47	40
V	1589	14	April	6	23	4	23	0	Scorpio	1	12	N.	7	14	18	26
VI	1591	8	Jume	7	43	26	43	0	Sagitt.	4	0	S.	9	5	43	55
VII	1593	25	August	17	27	12	16	0	Pisces	6	2	S.	11	9	49	31
VIII	1595	31	October	0	39	17	31	40	Taurus	0	8	N.	1	9	55	4
IX	1597	13	December	15	44	2	28	0	Cancer	3	33	N.	2	23	11	56
X	1600	18	January	14	2	8	38	0	Leo	4	30	N.	4	4	35	50
XI	1602	20	Febuary	14	13	12	27	0	Virgo	4	10	N.	5	14	59	37
XII	1604	28	March	16	23	18	37	10	Libra	2	26	N.	6	27	0	12

Johannes Kepler



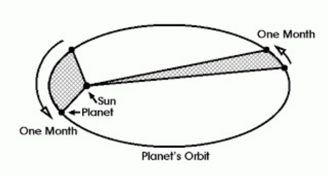
Ioannis Keppleri
HARMONICES
MVNDI
LIBRI V. QVORVM

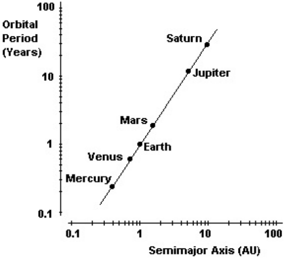
Primus GEOMETRICVS, De Figurarum Regularium, quæ Proportiones Harmonicas constituunt, ortu & demonstrationibus.
Secundus ARCHITECTONICVS, seu ex GEOMETRIA FIGVRATA, De Figurarum Regularium Congruentia in plano vel solido:
Tertius propriè HARMONICVS, De Proportionum Harmonicarum ortu ex Figuris; deque Naturâ & Differentiis rerum ad cantum pertinentium, contra Veteres:
Quartus METAPHYSICVS, PSYCHOLOGICVS & ASTROLOGICVS, De Harmoniarum mentali Essentiâ earumque generibus in Mundo; præsertim de Harmonia radiorum, ex corporibus cœlestibus in Terram descendentibus, eiusque effectu in Natura seu Anima sublunari & Humana:
Quintus ASTRONOMICVS & METAPHYSICVS, De Harmoniis absolutissimis motuum cœlestium, ortuque Eccentricitatum ex proportionibus Harmonicis.
Appendix habet comparationem huius Operis cum Harmonices Cl. Ptolemæi libro III. cumque Roberti de Fluctibus, dicti Flud. Medici Oxoniensis speculationibus Harmonicis, operi de Macrocosmo & Microcosmo insertis.

Cum S.C.M^{tis}. Priuilegio ad annos XV.
Lincii Austriæ,
Sumptibus GODOFREDI TAMPACHII Bibl. Francof.
Excudebat IOANNES PLANCVS.

ANNO M. DC. XIX.

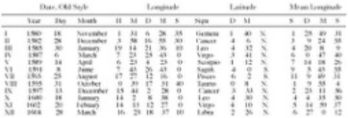
1st Edition 1st issue
6 plates on 5



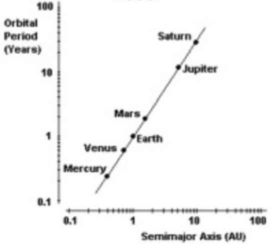


Data and models

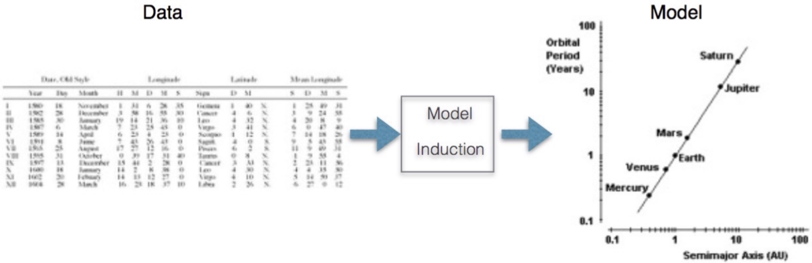
Data



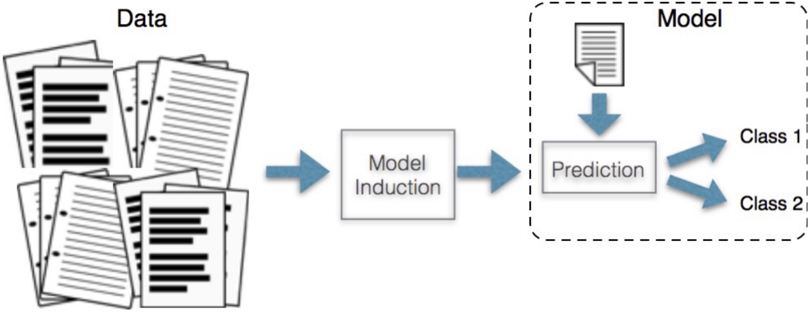
Model



Machine Learning



Machine Learning with Text Data



Outline

Machine Learning

- Construction and study of systems that can learn from data
- Main problem: to find patterns, relationships, regularities among data, which allow to build descriptive and predictive models.
- Related fields:
  - Statistics
  - Pattern recognition and computer vision
  - Data mining and knowledge discovery
  - Data analytics

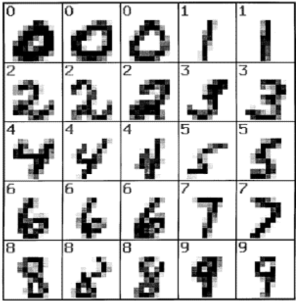
Brief history

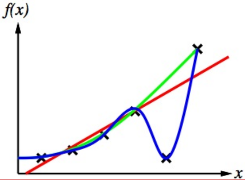
- Fisher's linear discriminant (Fisher, 1936)
- Artificial neuron model (MCCulloch and Pitts, 1943)
- Perceptron (Rosenblatt, 1957) (Minsky&Papert, 1969)
- Probably approximately correct learning (Valiant, 1984)
- Multilayer perceptron and back propagation (Rumelhart et al., 1986)
- Decision trees (Quinlan, 1987)
- Bayesian networks (Pearl, 1988)
- Support vector machines (Cortes&Vapnik, 1995)
- Efficient MLP learning, deep learning (Hinton et al., 2007)

Machine Learning in the news

Supervised learning

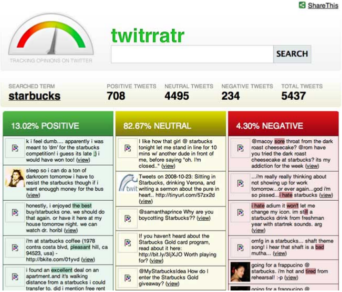
- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs
- Typical problems:
  - Classification
  - Regression





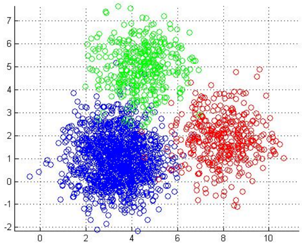
Supervised learning

- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs
- Typical problems:
  - Classification
  - Regression



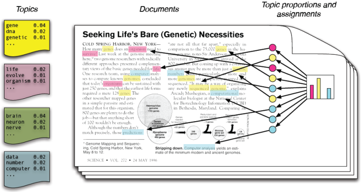
Non-supervised learning

- There are not labels for the training samples
- **Fundamental problem**: to find the subjacent structure of a training data set
- Typical problems: clustering, segmentation, dimensionality reduction, latent topic analysis
- Some samples may have labels, in that case it is called semi-supervised learning

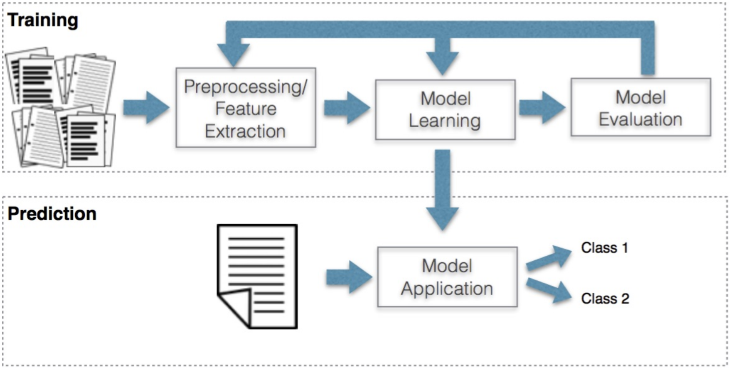


Non-supervised learning

- There are not labels for the training samples
- **Fundamental problem**: to find the subjacent structure of a training data set
- Typical problems: clustering, segmentation, dimensionality reduction, latent topic analysis
- Some samples may have labels, in that case it is called semi-supervised learning



The machine Learning process



Outline

Neural Networks

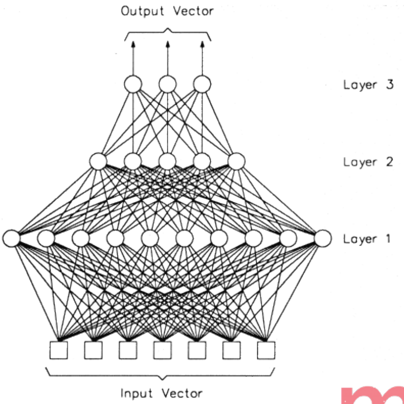
- Inspired by nature (the brain)
- Simple processing units but many of them and highly interconnected
- Distributed processing and memory
- Redundant, robust and fault tolerant
- Learn from data samples

Interactive demo

Quick and dirty introduction to neural networks

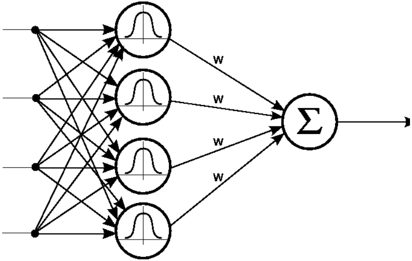
Types

- Feed-forward, multilayer perceptrons
- Radial basis function
- Recurrent
- Self-organizing maps



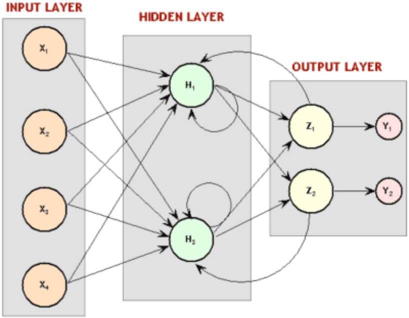
Types

- Feed-forward, multilayer perceptrons
- Radial basis function
- Recurrent
- Self-organizing maps



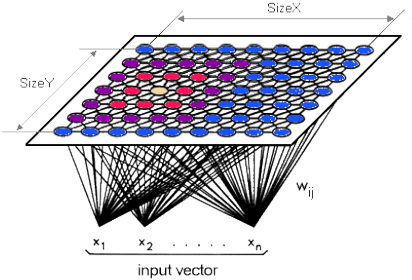
Types

- Feed-forward, multilayer perceptrons
- Radial basis function
- Recurrent
- Self-organizing maps



Types

- Feed-forward, multilayer perceptrons
- Radial basis function
- Recurrent
- Self-organizing maps



Learning as optimization

- General optimization problem:

$$\min_{f \in H} L(f, D),$$

with H: hypothesis space, D:training data, L:loss/error

- Squared error:

$$D = \{(x_1, t_1), \ldots, (x_\ell, t_\ell)\}$$

$$L(f_w, D) = E(w, D) = \sum_{i=1}^{\ell} \| f_w(x_i) - t_i \|_2^2$$

Other loss functions

- L_1 loss:

$$E(w,D) = \sum_{i=1}^{\ell} \|f_w(x_i) - t_i\|_1^2$$

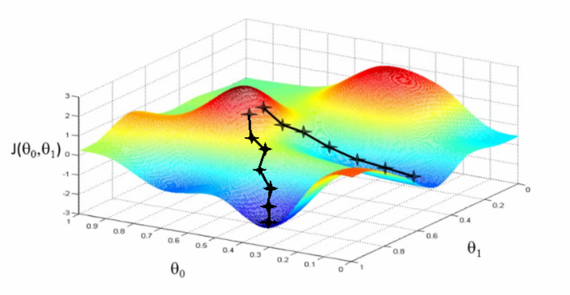
- Cross-entropy loss:

$$E(w,D) = -\ln \prod_{i=1}^{\ell} p(t_i|x_i,w) = -\sum_{i=1}^{\ell} [t_i \ln f_w(x_i) + (1-t_i)\ln(1-f$$

- Hinge loss:

$$E(w,D) = \sum_{i=1}^{\ell} \max(0, 1 - t_i f_w)$$

Optimization by Gradient descent

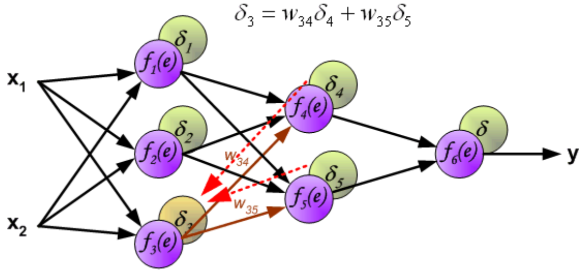


$$w^{t+1} = w^t - \eta_t \nabla_w E(w^t)$$

$$\nabla_w E(w) = \frac{\partial E(w)}{\partial w}$$

Backpropagation [Rumelhart, Hinton, 1986]

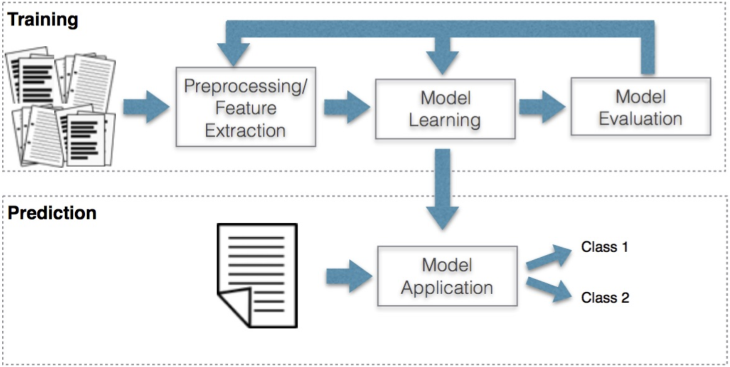
- Efficient strategy to calculate the gradient.
- Errors are back-propagated through the network to assign 'responsibility' to each neuron (δ_i)



- Gradient is calculated based on delta values.

Outline

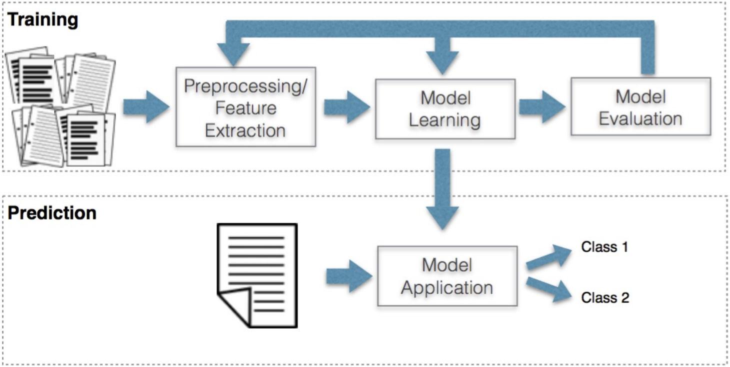
Feature extraction



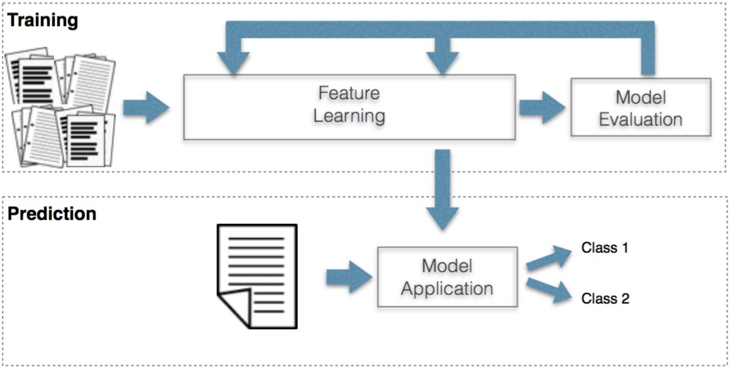
Features

- Features represent our prior knowledge of the problem
- Depend on the type of data
- Specialized features for practically any kind of data (images, video, sound, speech, text, web pages, etc)
- Medical imaging:
    - Standard computer vision features (color, shape, texture, edges, local-global, etc)
    - Specialized features tailored to the problem at hand
- New trend: learning features from data

Feature learning



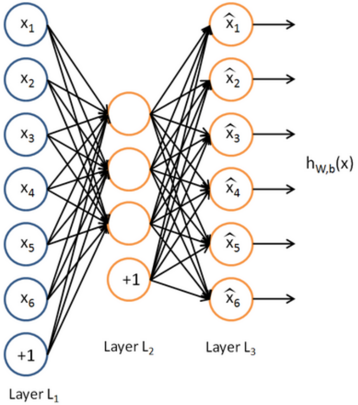
Feature learning

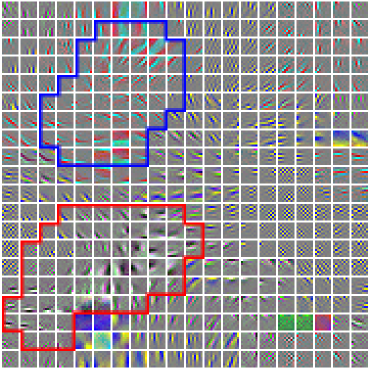


Feature learning approaches

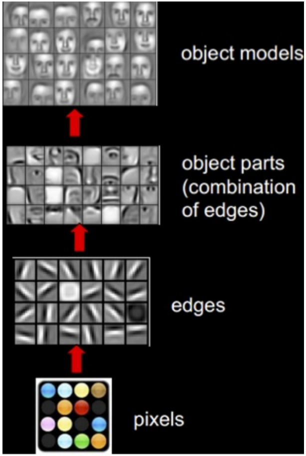
- Unsupervised feature learning
- Convolutional neural networks
- Recurrent neural networks

Unsupervised feature learning



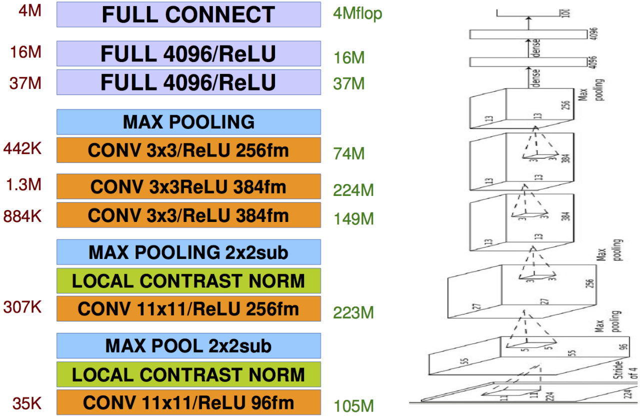


Deep feed-forward neural networks



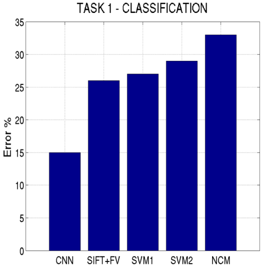
ImageNet 2012 [Krizhevsky, Sutskever, Hinton 2012]

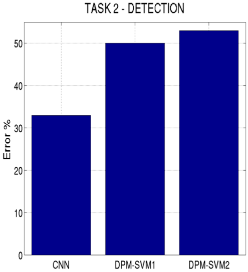
- **Won the 2012 ImageNet LSVRC. 60 Million parameters, 832M MAC ops**

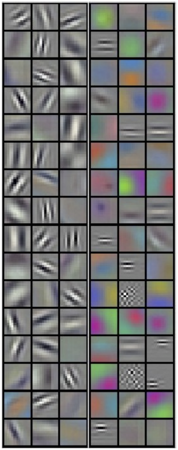


(source: ICML2013 Deep Learning Tutorial, Yan LeCun et al.)

ImageNet 2012 [Krizhevsky, Sutskever, Hinton 2012]







(source: ICML2013 Deep Learning Tutorial, Yan LeCun et al.)

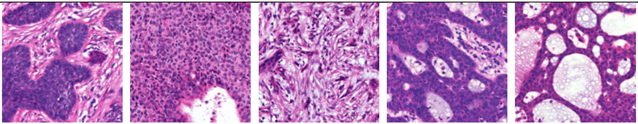
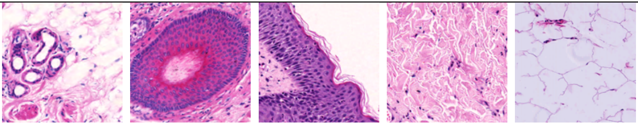
Practical considerations

- Traditional backpropagation does not work well with multiple layers
- It gets stuck in local minima
- During the last years several strategies have been developed/discovered (*tricks of the trade*):
  - Stochastic gradient descent with minibatches and adaptive learning rate
  - Logistic regression/soft max for classification
  - Normalization of input variables, shuffling of training samples
  - Regularization using L_1 and L_2 norms and dropout

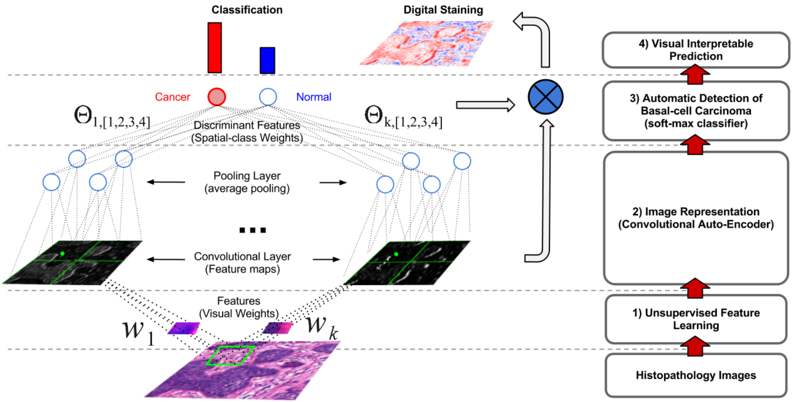
Implementation

- Use of GPUs is mandatory (speed-up > 100x)
- Sometimes combined with distributed processing
- Practically all the libraries use CUDA
- Several higher-level frameworks:
  - NVIDIA CUDA Deep Neural Network library (cuDNN)
  - Caffe
  - Torch
  - Theano
  - Blocks
  - Etc.

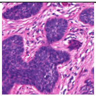
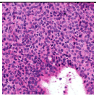
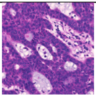
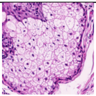
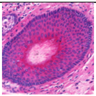
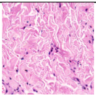
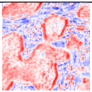
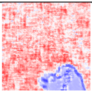
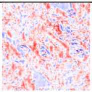
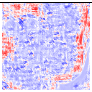
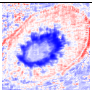
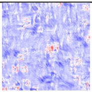
(Histopathology basal cell carcinoma

Tumor	
Non-tumor	

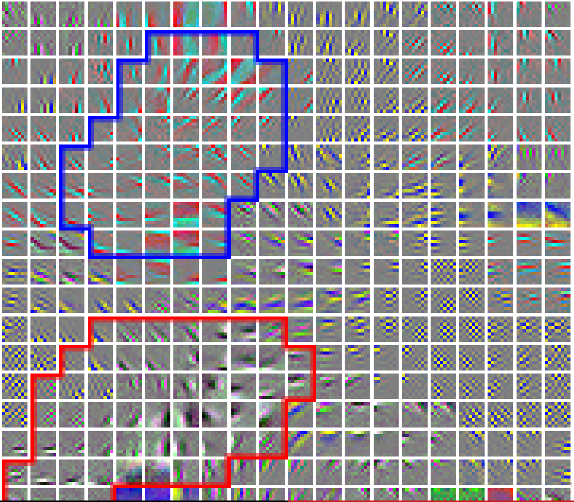
Convolutional Autoencoder for Histopathology Image Representation Learning



Digital staining results

Cancer	Cancer	Cancer	Non-cancer	Non-cancer	Non-cancer
					
Cancer	Cancer	Cancer	Non-cancer	Non-cancer	Non-cancer
0.8272	0.9604	0.7944	0.2763	0.0856	0.0303
					

TICA learned features)



Feature learning for natural language data

- But what about text?
- Neural networks are a hot topic in NLP now a days:
  - "*NN language models and word embeddings were everywhere at NAACL2015 and ACL2015*" C. Manning.
  - Many successful applications:
    - Speech recognition
    - Language modeling
    - Translation
    - Image captioning

Outline

Bag-of-words and one-hot representation

- Bag-of-words representation: a document is represented by the frequency of the words in it:

the	dog	a	cat	chases	jump	tails
1	1	0	1	1	0	0

- If we apply this representation to a word, we get a *one-hot* vector:

chases	0	0	0	0	**1**	0	0
tails	0	0	0	0	0	0	**1**

- Problem: vectors for different words are orthogonal even if the words are related

Distributed word/document representation

- Words are represented by continuous vectors:

chases	0.1	0.3	-0.3	0.0	-0.8	0.7	0.0
tails	0.2	0.3	-0.4	0.1	-0.7	0.8	0.0

- Question: how to build this kind of representation?

Distributional Hypothesis.

- "*Words that are used and occur in the same contexts tend to purport similar meanings.*"

  government debt problems turning into **banking** crises as has happened in

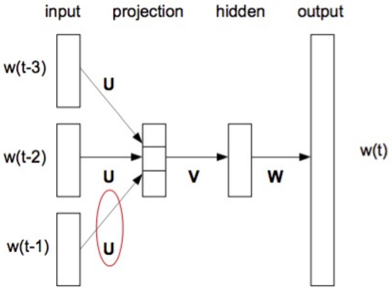
  saying that Europe needs unified **banking** regulation to replace the hodgepodge

- **Compositional distributional models**:
  the meaning of a sequence of words is represented by the combination of the vectors of the words within the sequence

$$f(\text{'the dog chases the cat'}) = f(\text{'the'}) + f(\text{'dog'}) + \cdots + f(\text{'cat'})$$

Neural Net Language Model

- Problem: predict the next word given the previous 3 words (4-gram language model)
- The matrix U corresponds to the word vector representation of the words.



Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). *A neural probabilistic language model*. The Journal of Machine Learning Research, 3, 1137-1155.

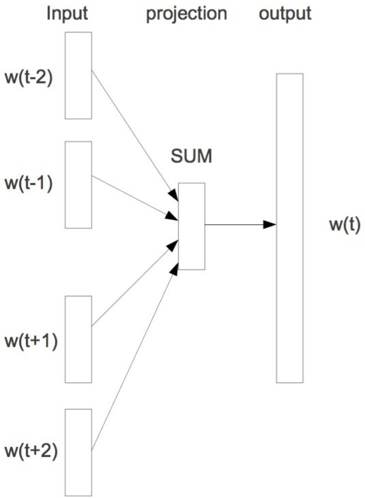
word2vec

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. In Proceedings of Workshop at ICLR, 2013.

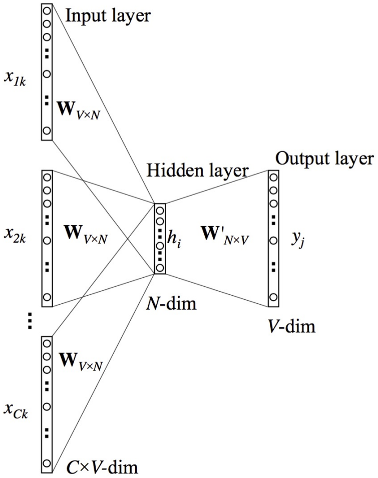
- Neural network architecture for *efficiently* computing continuous vector representations of words from very large data sets.
- Proposes two strategies:
  - Continuous bag-of-words
  - Continuous skip-gram

Continuous bag-of-words

- Problem: predict a word given its context.
- All the words in the context use the same codification.
- The representation of the words in the context are summed (compositionality).

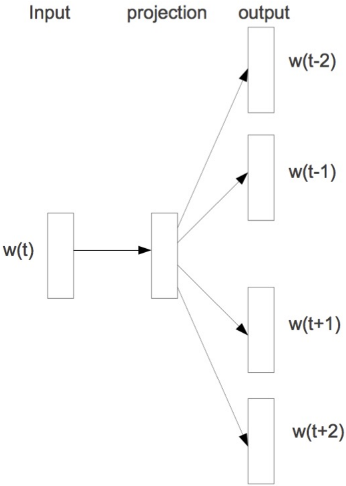


CBOW detail


Input layer
x_{1k}
$\mathbf{W}_{V\times N}$
x_{2k}
$\mathbf{W}_{V\times N}$
x_{Ck}
$\mathbf{W}_{V\times N}$
$C\times V$-dim
Hidden layer
h_i
N-dim
$\mathbf{W}'_{N\times V}$
Output layer
y_j
V-dim

Skip-gram

- Problem: predict the context given a word
- All the words in the context use the same codification.



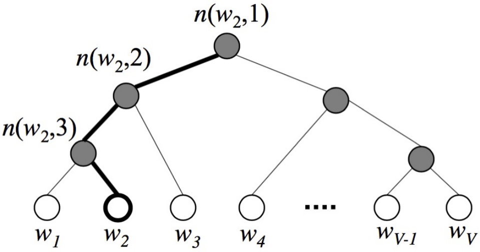
Efficient implementation

- Soft-max output:

$$y_j = P(w_j|h) = \frac{\exp(W_j'h)}{\sum_{i=1}^{n}\exp(W_i'h)}$$

- To calculate the denominator you have to add over the whole vocabulary. Very inefficient!
- Strategies:
  - Hierarchical softmax
  - Negative sampling

Hierarchical softmax



$$p(w = w_O) = \prod_{j=1}^{L(w)-1} \sigma(\llbracket n(w, j+1) = \mathrm{ch}(n(w,j)) \rrbracket \, {v'}_{n(w,j)} h)$$

Interactive demo

Playing with word2vec

Papers (1)

- Bengio, Yoshua, et al. "A neural probabilistic language model." The Journal of Machine Learning Research 3 (2003): 1137-1155.
- Bottou, Léon. "From machine learning to machine reasoning." Machine learning 94.2 (2014): 133-149.
- Turian, Joseph, Lev Ratinov, and Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning." Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010.
- Collobert, Ronan, et al. "Natural language processing (almost) from scratch." The Journal of Machine Learning Research 12 (2011): 2493-2537.
- Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." HLT-NAACL. 2013.
- Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." CoRR2013. arXiv preprint arXiv:1301.3781 (2013).

Papers (2)

- Socher, Richard, et al. "Zero-shot learning through cross-modal transfer." Advances in neural information processing systems. 2013.
- Zou, Will Y., et al. "Bilingual Word Embeddings for Phrase-Based Machine Translation." EMNLP. 2013.
- Frome, Andrea, et al. "Devise: A deep visual-semantic embedding model." Advances in Neural Information Processing Systems. 2013.
- Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12 (2014): 1532-1543.
- Soricut, Radu, and Franz Och. "Unsupervised morphology induction using word embeddings." Proc. NAACL. 2015.
- Camacho-Collados, José, Mohammad Taher Pilehvar, and Roberto Navigli. "A unified multilingual semantic representation of concepts." Proceedings of ACL, Beijing, China (2015).
- Arora, Sanjeev, et al. "Random Walks on Context Spaces: Towards an Explanation of the Mysteries of Semantic Word Embeddings." arXiv preprint arXiv:1502.03520 (2015).

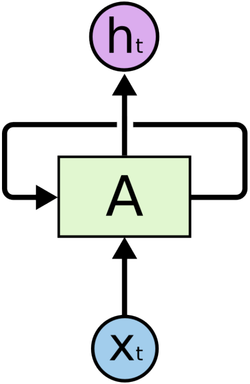
Other resources

- Blog: *Deep Learning, NLP, and Representations*, http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/
- Software: *GloVe: Global Vectors for Word Representation*, http://nlp.stanford.edu/projects/glove/
- Software: *Gensim, topic modeling for humans*, https://radimrehurek.com/gensim/
- Software: word2vec, https://code.google.com/p/word2vec/

Outline

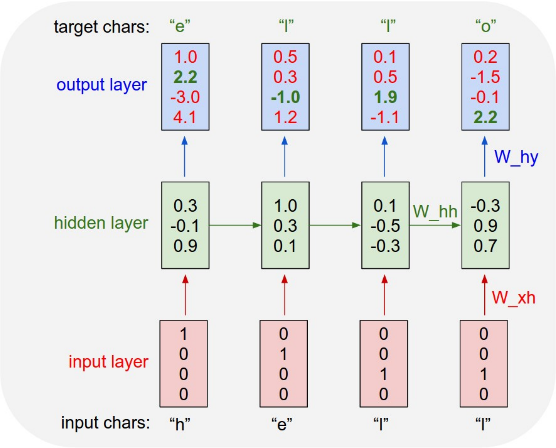
Recurrent neural network

- Neural networks with memory
- Feed-forward NN: output exclusively depends on the current input
- Recurrent NN: output depends in current and previous states
- This is accomplished through lateral/backward connections which carry information while processing a sequence of inputs



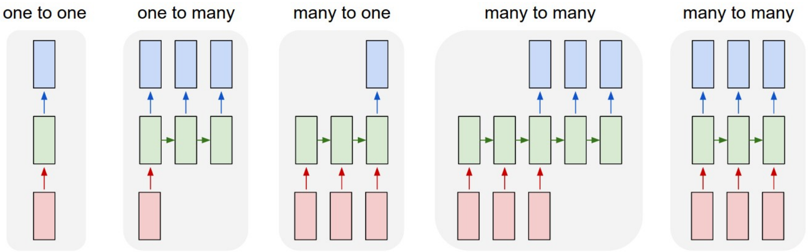
(source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Character-level language model



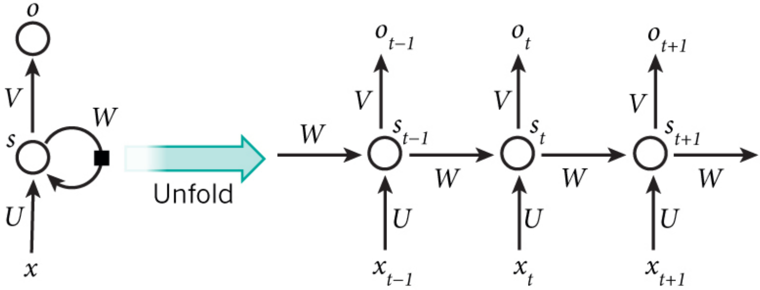
(source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Sequence learning alternatives



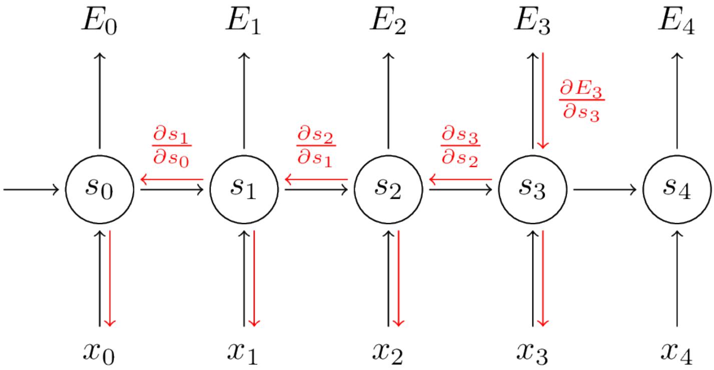
(source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Network unrolling



(source: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/)

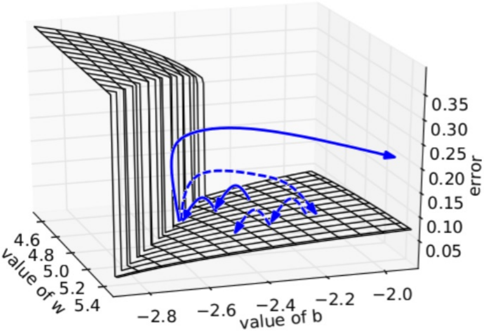
Backpropagation through time (BPTT)



(source: http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/)

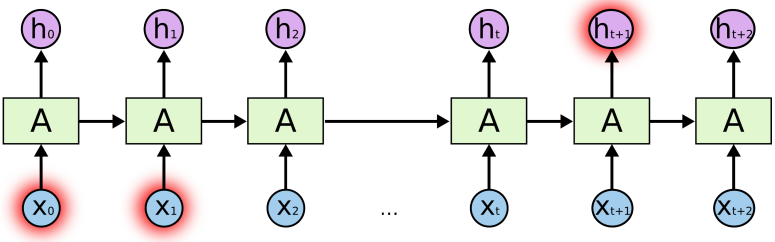
BPTT is hard

- The *vanishing* and the *exploding* gradient problem
- Gradients could vanish (or explode) when propagated several steps back
- This makes difficult to learn long-term dependencies.



Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. *On the difficulty of training Recurrent Neural Networks*. Proc. of ICML, abs/1211.5063.

Long term dependencies



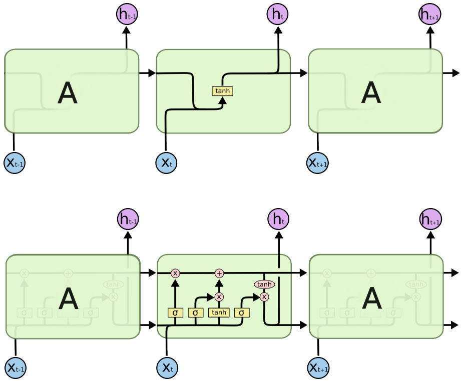
(source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Long short-term memory (LSTM)

Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.

- LSTM networks solve the problem of long-term dependency problem.
- They use *gates* that allow to keep memory through long sequences and be updated only when required.

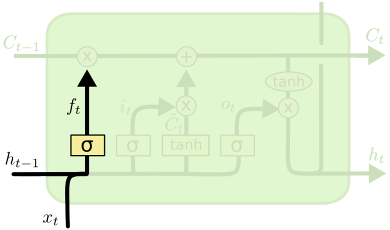
Conventional RNN vs LSTM



(image source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Forget gate

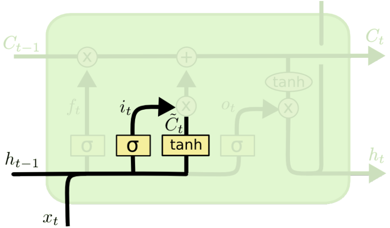
- Controls the flow of the previous internal state C_{t-1}
- $f_t = 1 \Rightarrow$ keep previous state
- $f_t = 0 \Rightarrow$ forget previous state



(image source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

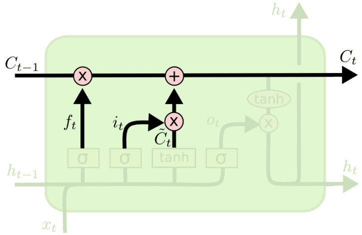
Input gate

- Controls the flow of input information (x_t)
- $i_t = 1 \Rightarrow$ take input into account
- $i_t = 0 \Rightarrow$ ignore input



(image source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Current state calculation

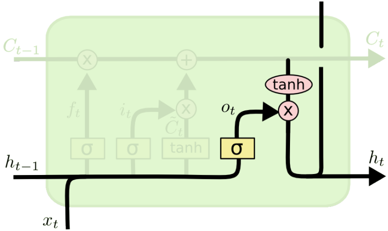


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

(image source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

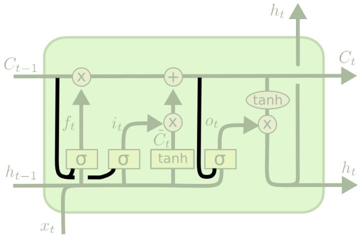
Output gate

- Controls the flow of information from the internal state (x_t) to the outside (h_t)
- $o_t = 1 \Rightarrow$ allows internal state out
- $o_t = 0 \Rightarrow$ doesn't allow internal state out



(image source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

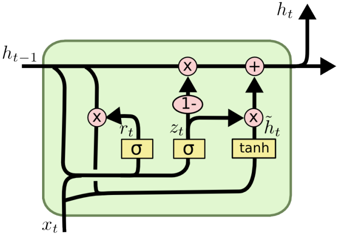
Peephole connections



$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \;+\; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \;+\; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \;+\; b_o\right)$$

Gers, F., & Schmidhuber, J. (2000). *Recurrent nets that time and count*. In Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on (Vol. 3, pp. 189-194). IEEE.
(image source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Gated recurrent units



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using rnn encoder-decoder for statistical machine translation*. arXiv preprint arXiv:1406.1078.
(image source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Interactive demo

Language modeling with LSTM

The Unreasonable Effectiveness of Recurrent Neural Networks

- Famous blog entry from Andrej Karpathy (UofS)
- Character-level language models based on multi-layer LSTMs.
- Data:
  - Shakspare plays
  - Wikipedia
  - LaTeX
  - Linux source code

Algebraic geometry book in LaTeX

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{étale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

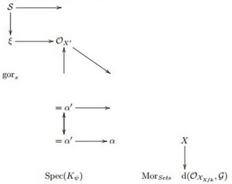
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{étale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

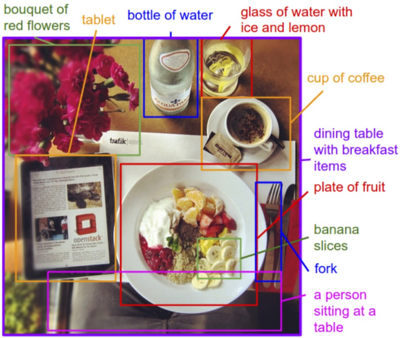
If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

(source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Linux source code

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
```

Image captioning



Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." CVPR2015. arXiv preprint arXiv:1412.2306 (2014).

Approach

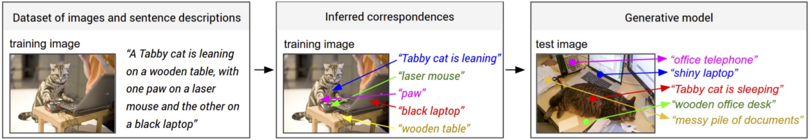

Dataset of images and sentence descriptions
training image
"A Tabby cat is leaning on a wooden table, with one paw on a laser mouse and the other on a black laptop"
Inferred correspondences
training image
"Tabby cat is leaning"
"laser mouse"
"paw"
"black laptop"
"wooden table"
Generative model
test image
"office telephone"
"shiny laptop"
"Tabby cat is sleeping"
"wooden office desk"
"messy pile of documents"

Image-sentence score model

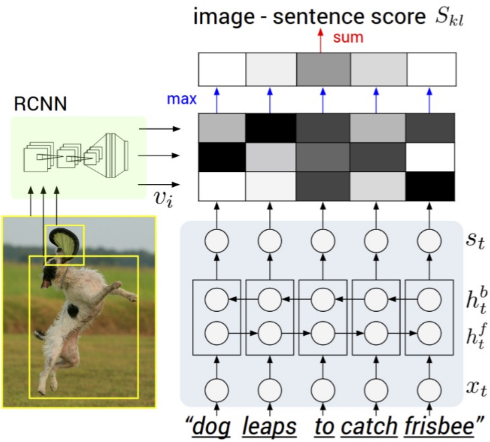


Image-sentence score model

- A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. arXiv preprint arXiv:1406.5679, 2014.

$$S_{kl} = \sum_{t \in g_l} \sum_{i \in g_k} \max(0, v_i^T s_t)$$

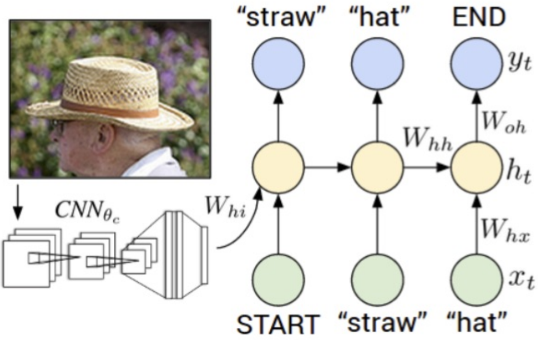
- Simplification:

$$S_{kl} = \sum_{t \in g_l} \max_{i \in g_k} v_i^T s_t$$

- Loss:

$$C(\theta) = \sum_k \left[\sum_l \max(0, S_{kl} - S_{kk} + 1) + \sum_l \max(0, S_{lk} - S_{kk} + 1) \right]$$

Multimodal RNN



Alignment results

Captioning results

man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

two young girls are playing with lego toy.

Papers (1)

- General:
  - S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735-1780, 1997. Based on TR FKI-207-95, TUM (1995).
  - J. Schmidhuber. Deep Learning in Neural Networks: An Overview. Neural Networks, Volume 61, January 2015, Pages 85-117 (DOI: 10.1016/j.neunet.2014.09.003)
- Language modeling:
  - Mikolov, Tomas, et al. "Recurrent neural network based language model." INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010. 2010.
  - Mikolov, Tomáš, et al. "Extensions of recurrent neural network language model." Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, 2011.
  - Sutskever, Ilya, James Martens, and Geoffrey E. Hinton. "Generating text with recurrent neural networks." Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011.

Papers (2)

- Machine translation:
  - Liu, Shujie, et al. "A recursive recurrent neural network for statistical machine translation." Proceedings of ACL. 2014.
  - Sutskever, Ilya, Oriol Vinyals, and Quoc VV Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
  - Auli, Michael, et al. "Joint Language and Translation Modeling with Recurrent Neural Networks." EMNLP. Vol. 3. No. 8. 2013.
- Speech recognition:
  - Graves, Alex, and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014.

Papers (3)

- Image captioning:
  - Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." CVPR2015. arXiv preprint arXiv:1412.2306 (2014).
  - Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." CVPR2015. arXiv preprint arXiv:1411.4555 (2014).
  - Chen, Xinlei, and C. Lawrence Zitnick. "Learning a recurrent visual representation for image caption generation." arXiv preprint arXiv:1411.5654 (2014).
  - Fang, Hao, et al. "From captions to visual concepts and back." CVPR2015, arXiv preprint arXiv:1411.4952 (2014).

Other resources

- Christopher Olah, Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- Denny Britz, Recurrent Neural Networks Tutorial, http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/
- Andrej Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks, http://karpathy.github.io/2015/05/21/rnn-effectiveness/
- Jürgen Schmidhuber, Recurrent Neural Networks, http://people.idsia.ch/~juergen/rnn.html

Thanks!

fagonzalezo@unal.edu.co

http://www.mindlaboratory.org