Master 2 Recherche
Apprentissage Statistique et Optimisation

François Yvon – Michèle Sebag
Alexandre Allauzen – Marc Schoenauer

http://www.limsi.fr/Individu/allauzen/wiki/index.php/TSI09 – http://tao.lri.fr/tiki-index.php

21 octobre 2010: Réseaux Neuronaux, 2

Réseaux Neuronaux Artificiels

Réseaux classiques

1. Histoire
   - Neurones logiques/non supervisés
   - Perceptron, Adaline: algorithmes
   - L'hiver des NN: limites théoriques
   - Perceptron Multi-couches: algorithme et limites algorithmiques
2. Structure du réseau et apprentissage

Types de réseaux neuronaux

Ingrédients

- Une fonction d'activation
- Un ensemble de neurones connectés (graphe orienté) feedforward ($\equiv$ DAG) ou récurrent
- Un poids sur chaque connexion

Fonction d'activation(z)

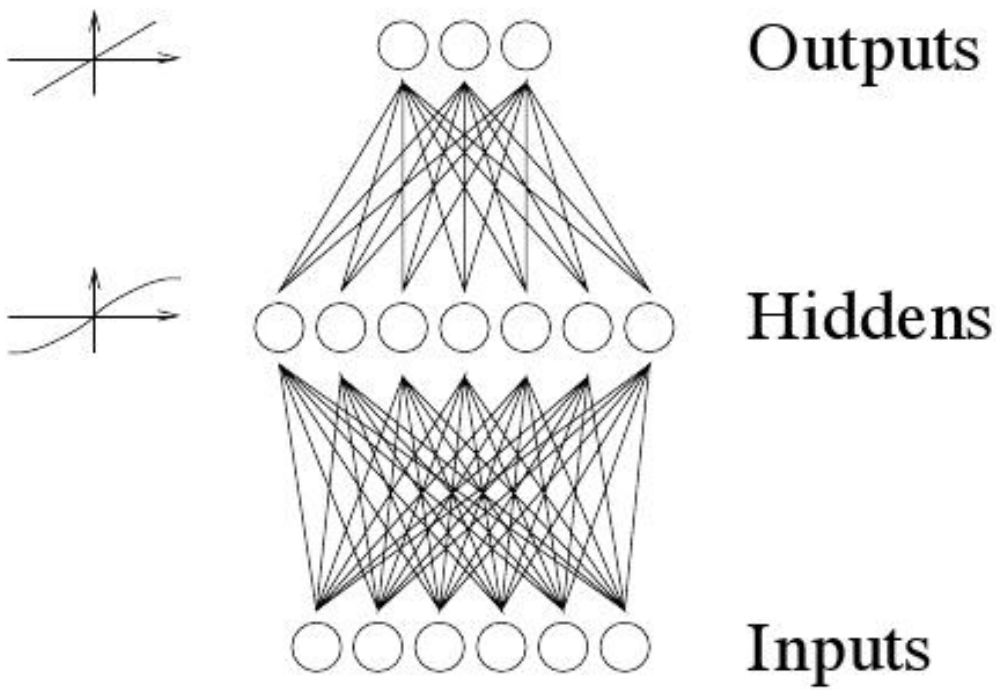
- à seuil — 0 si $z < \mathit{seuil}$, 1 sinon
- linéaire — z
- sigmoide — $1/(1+e^{-z})$
- Radius — e^{-z^2/σ^2}

Types de réseaux neuronaux, 2

Ingrédients

- Une fonction d'activation
- Un ensemble de neurones connectés (graphe orienté) feedforward (≡ DAG) ou récurrent
- Un poids sur chaque connexion

Structure feedforward



Types de réseaux neuronaux, 2

Ingrédients

- Une fonction d'activation
- Un ensemble de neurones connectés (graphe orienté) feedforward ($\equiv$ DAG) ou récurrent
- Un poids sur chaque connexion

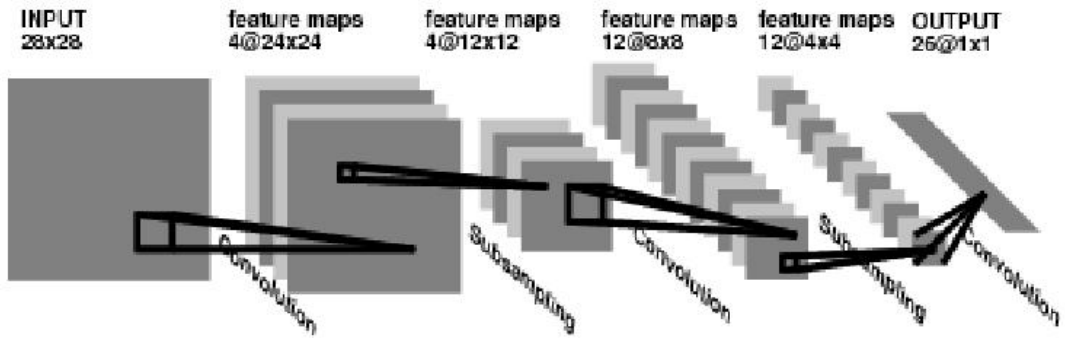
Structure récurrente

- Propager ; attendre la stabilisation
- Retro-propagation inapplicable
- Le système a une mémoire (valeur des neurones cachés)
  Hum. La mémoire s'efface exponentiellement vite
- Données dynamiques (video)

Structure / Graphe de connexion / Topologie

Connaissances a priori

- Invariance par translation, rotation,.. op
- $\longrightarrow$ Etendre $\mathcal{E}$ considérer $(op(\mathbf{x}_i), y_i)$
- ou partager les poids: Réseau convolutionnel



100,000 poids $\longrightarrow$ 2,600 paramètres

- Couche 1: detection de "features", e.g. un trait;
- Couche 2: moyennage et sous-echantillonnage: une fois qu'une feature a été détectée, sa position exacte est moins importante, tant que la position relative par rapport aux autres features est préservée.

Plus: http://yann.lecun.com/exdb/lenet/

Propriétés

Good news

- MLP, RBF: approximateurs universels

Pour toute fonction f raisonnable (de carré intégrable sur un compact), pour tout $\epsilon > 0$, il existe un MLP/RBF approchant f avec la précision ϵ.

Bad news

- Ce résultat n'est pas constructif. (i.e. il existe; so what).
- On peut avoir tout ce qu'on veut; $\rightarrow$ on peut avoir n'importe quoi (sur-apprentissage).

Questions centrales

Sélection de modèles

- Choix de nombre de neurones, de la topologie, ...
- Choix de critère overfitting

PLUS $\not\Rightarrow$ MIEUX

Choix algorithmiques un pb d'optim. difficile

- Initialisation **w** petit !
- Décroissance du pas
- Utilisation d'un momentum relaxation

$$\mathbf{w}_{neo} \leftarrow (1-\alpha)\mathbf{w}_{old} + \alpha\mathbf{w}_{neo}$$

- Critère d'arrêt

Commencer par centrer normer les données, naturellement

$$x \mapsto \frac{x - \mathit{moyenne}}{\mathit{variance}}$$

Les NN: La malédiction

- **The NIPS community has suffered of an acute convexivitis epidemic**
  - ML applications seem to have trouble moving beyond logistic regression, SVMs, and exponential-family graphical models.
  - For a new ML model, convexity is viewed as a virtue
  - Convexity is sometimes a virtue
  - But it is often a limitation

  - ML theory has essentially never moved beyond convex models
    - the same way control theory has not really moved beyond linear systems

  - Often, the price we pay for insisting on convexity is an unbearable increase in the size of the model, or the scaling properties of the optimization algorithm [O(n^2), O(n^3)...]

http://videolectures.net/eml07_lecun_wia/

Quelques bonnes adresses

Quelques liens

- un cours:
  `http://neuron.tuke.sk/math.chtf.stuba.sk/pub/vlado/NN_books_texts/Krose_Smagt_neuro-intro.pdf`
- une FAQ: `http://www.faqs.org/faqs/ai-faq/neural-nets/part1/preamble.html`
- des applets
  `http://www.lri.fr/~marc/EEAAX/Neurones/tutorial/`
- des implémentations: PDP++/Emergent (`www.cnbc.cmu.edu/PDP++/`); SNNS `http://www-ra.informatik.uni-tuebingen.de/SgNNS/`...

Voir aussi

- NEAT & HyperNEAT — Stanley, U. Texas
  When no examples available: e.g. robotics.

Réseaux Neuronaux

1. Pourquoi ?
2. Réseaux classiques
3. Quelques applications
4. Réseaux modernes: profonds, à échos, impulsionnels, liquides...
5. Pistes de recherche fraiches

Applications

1. Reconnaissance de chiffres manuscrits
2. Reconnaissance de visages, de piétons
3. Contrôle (conduite de véhicule,..)
4. Language

Philosophie

Conception, la voie royale

- Décomposer un système en modules élémentaires
- de telle sorte que chacun d'entre eux puisse être spécifié, réalisé et testé *aisément*.

Pourquoi faire autrement ?

Philosophie

Conception, la voie royale

- Décomposer un système en modules élémentaires
- de telle sorte que chacun d'entre eux puisse être spécifié, réalisé et testé *aisément*.

Pourquoi faire autrement ?

- Quand on ne peut pas faire autrement (reconnaitre un visage)
- ou quand le monde est ouvert
- ou quand ca prend trop longtemps

Preuves de principe

- speech & hand-writing recognition: avec assez de données, des algorithmes d'apprentissage peuvent construire des systèmes complexes de bonne qualité.
- hand-crafting $\longrightarrow$ learning

Reconnaissance de chiffres manuscrits

Fig. 4. Size-normalized examples from the MNIST database.

Caractéristiques

- Taille input d: quelques centaines
- $\longrightarrow$ beaucoup de poids :-(
- Connaissance a priori: la classe ne change pas par (petites) translation, rotation, homothétie, des données

Réseaux convolutionnels

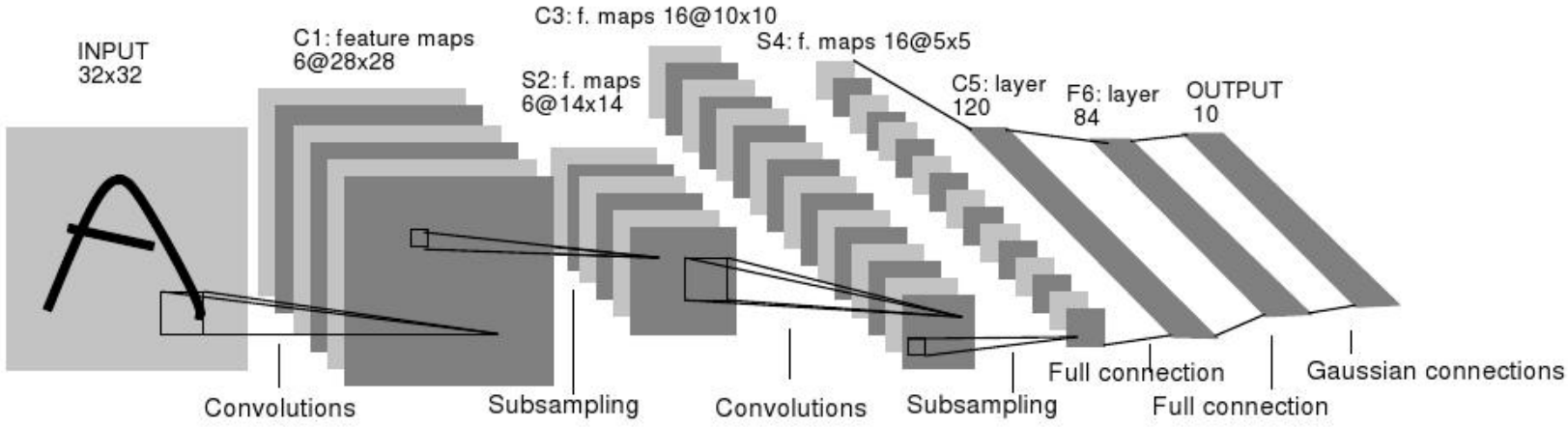
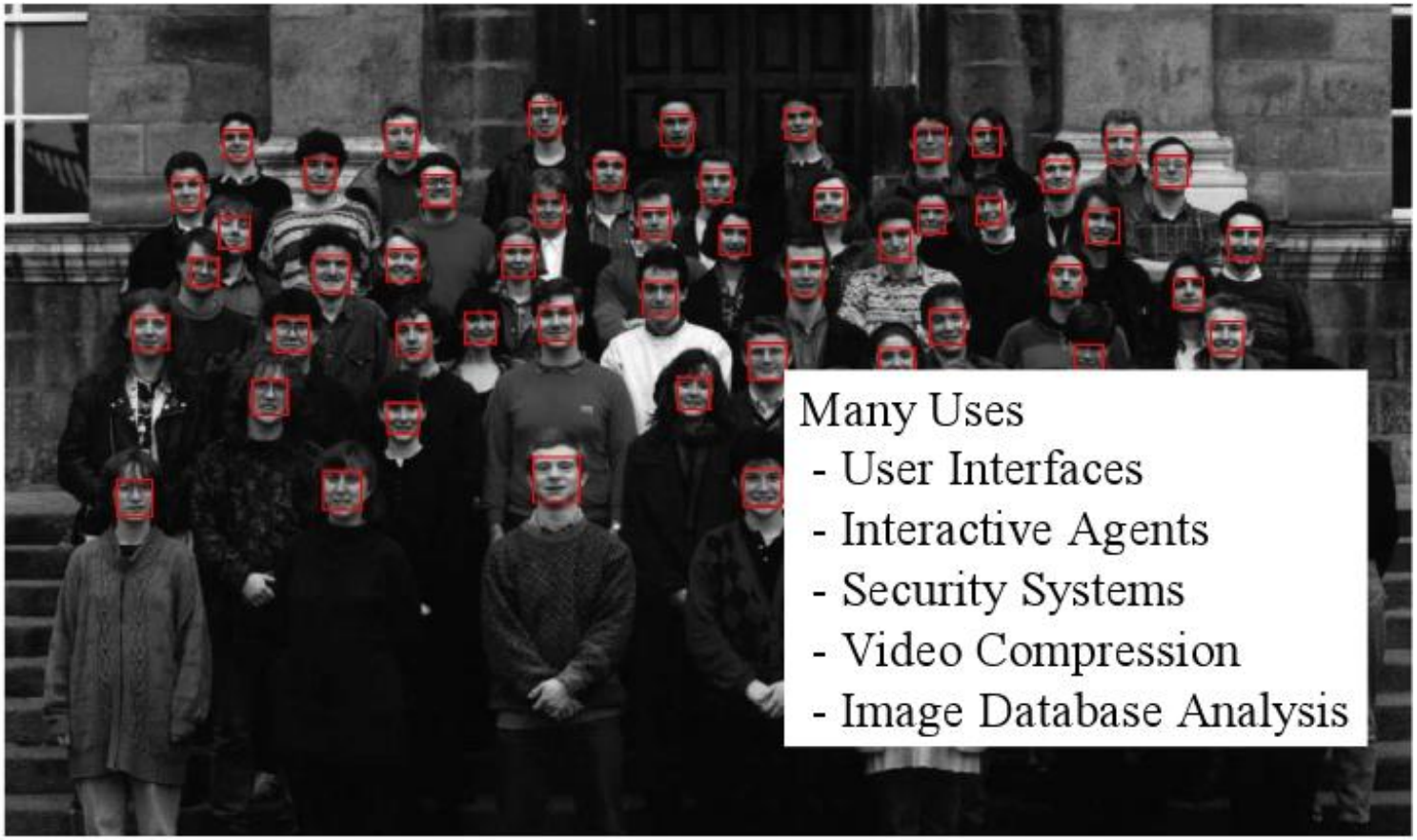


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Lecture http://yann.lecun.com/exdb/lenet/

- Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks. MIT Press, 1995.

Reconnaissance de visages



2

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Reconnaissance de visages

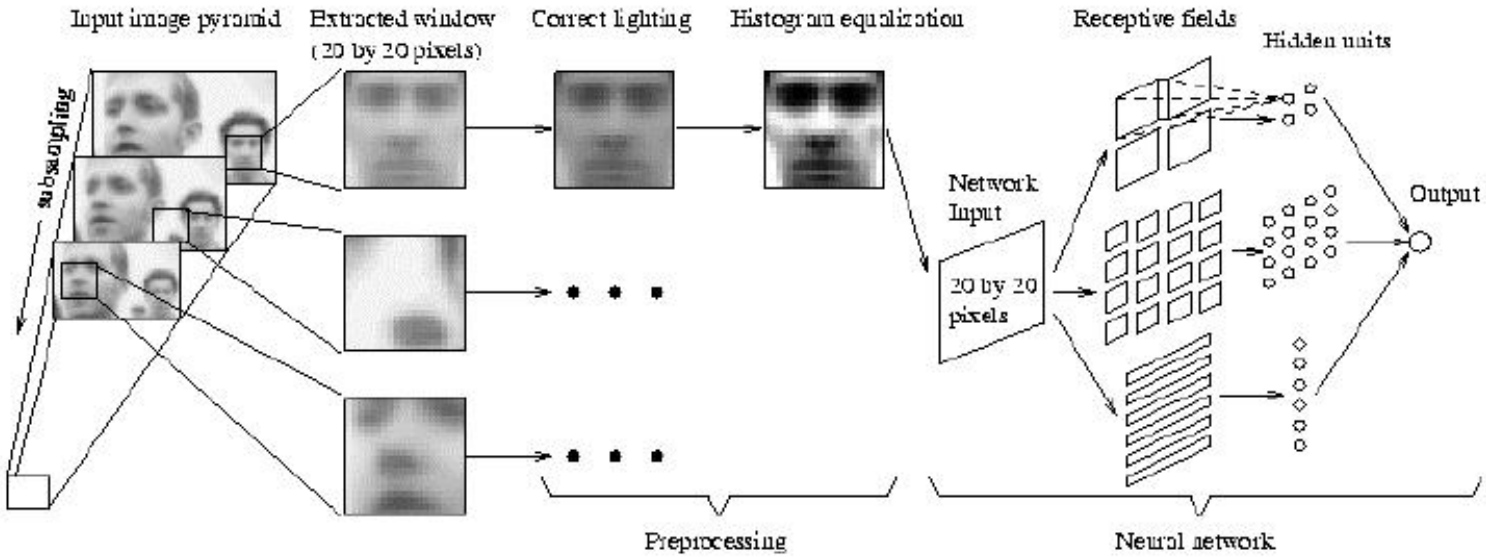
Difficultés : Variabilité

- Pose
- Elements: lunettes, barbe,...
- Eclairage
- Expression
- Orientation

Et occlusions

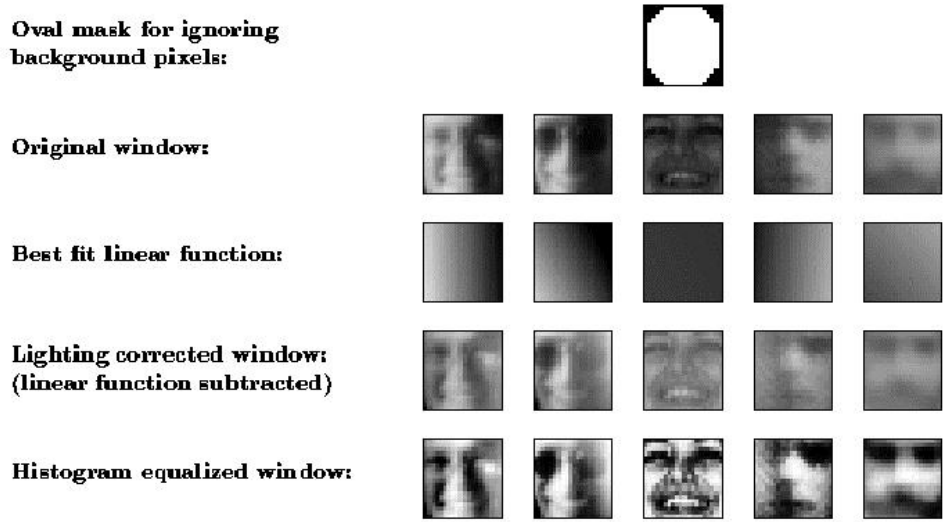
`http://www.ai.mit.edu/courses/6.891/lectnotes/lect12/lect12-slides-6up.pdf`

Reconnaissance de visages, 2

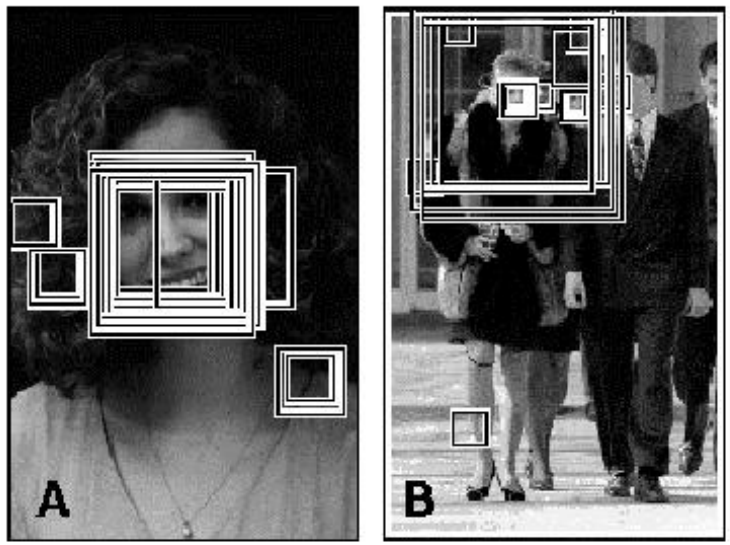


- Une équation $\longrightarrow$ 1 réseau
- NN **rapides**

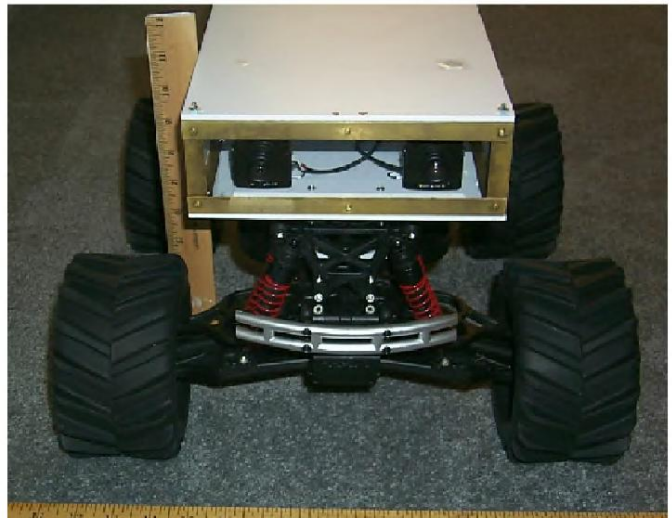
Reconnaissance de visages, 3

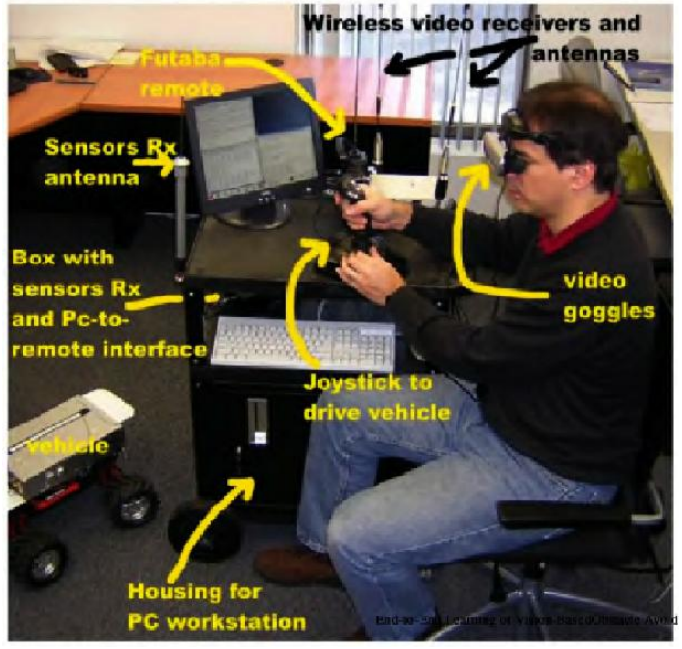


The steps in preprocessing a window. First, a linear function is fit to the intensity values in the window, and then subtracted out, correcting for some extreme lighting conditions. Then, histogram equalization is applied, to correct for different camera gains and to improve contrast. For each of these steps, the mapping is computed based on pixels inside the oval mask, while the mapping is applied to the entire window.



Controle de véhicule





Lectures, Video

http://www.cs.nyu.edu/~yann/research/dave/index.html

Modèles continus de langage

Principe

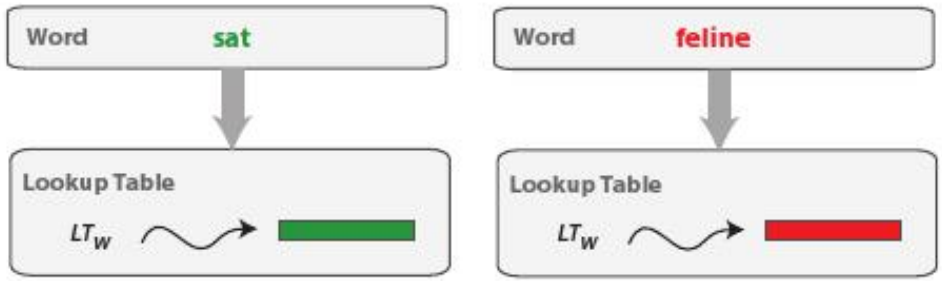
- Input: 10,000 input booleens les mots de la phrase
- Neurones cachés: 500 neurones numériques
- But: En fonction des mots d'une fenêtre $m_1, ...m_{2k+1}$, apprendre
  - Le genre grammatical du mot central m_k
  - Le mot suivant m_{2k+2}
- Rq: La première couche: $\{$ fragments de phrase $\}$ $\mapsto \mathbb{R}^{500}$

Bengio et al. 2001

Modèles continus de langage, Collobert et al. 2008

Improving Word Embedding

- Rare words are not trained properly
- Sentences with similar words should be tagged in the same way:
  - The cat sat on the mat
  - The feline sat on the mat



- Wordnet
  - pull together linked words
  - push apart other pair of words

videolectures

Modèles continus de langage, Collobert et al. 2008

Language Model: Think Massive

- Language Model: "*is a sentence actually english or not?*"
  Implicitly captures: ⋆ syntax ⋆ semantics
- Bengio & Ducharme (2001) Probability of next word given previous words. Overcomplicated – we do not need probabilities here
- English sentence windows: Wikipedia ($\sim 631M$ words)
  Non-english sentence windows: middle word randomly replaced
- Multi-class margin cost:

$$\sum_{s \in \mathcal{S}} \sum_{w \in \mathcal{D}} \max\left(0, 1 - f\left(s, w_s^\star\right) + f\left(s, w\right)\right)$$

$\mathcal{S}$: sentence windows $\mathcal{D}$: dictionary
$w_s^\star$: true middle word in s
$f\left(s, w\right)$: network score for sentence s and middle word w

Modèles continus de langage, Collobert et al. 2008

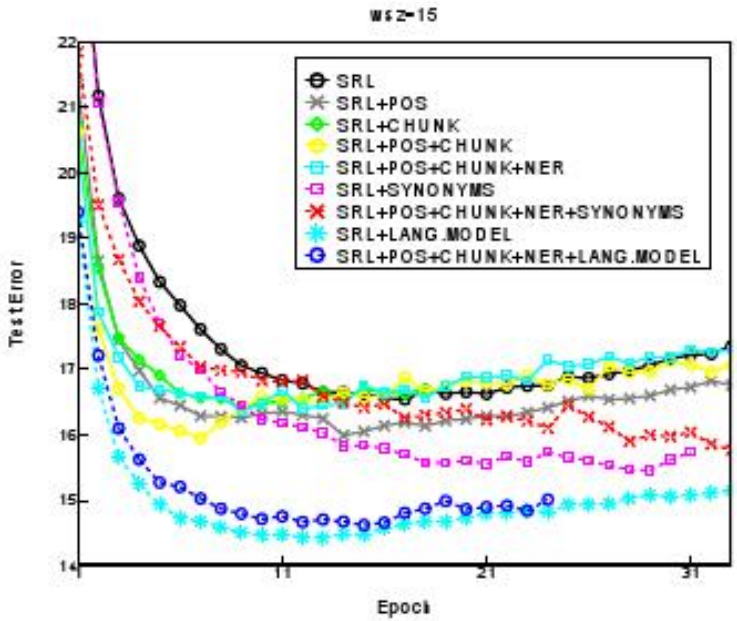
Language Model: Embedding

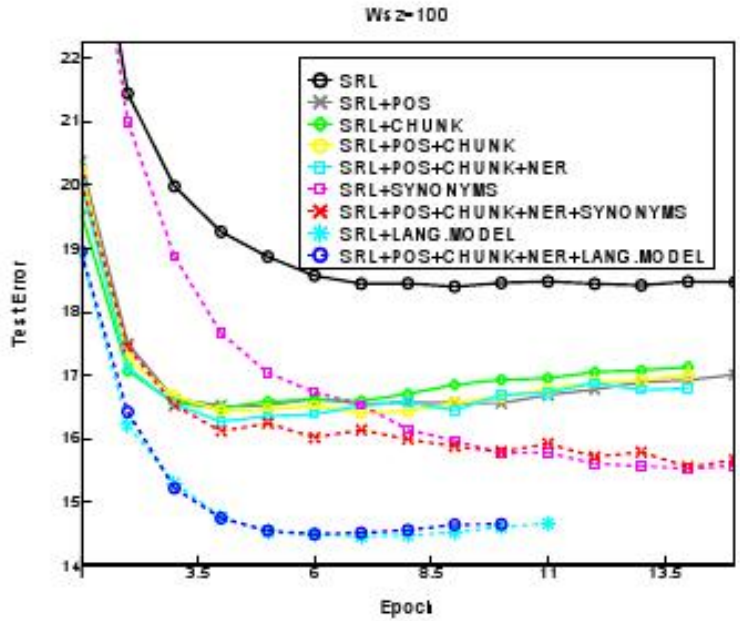
france	jesus	xbox	reddish	scratched
454	1973	6909	11724	29869
spain	christ	playstation	yellowish	smashed
italy	god	dreamcast	greenish	ripped
russia	resurrection	psNUMBER	brownish	brushed
poland	prayer	snes	bluish	hurled
england	yahweh	wii	creamy	grabbed
denmark	josephus	nes	whitish	tossed
germany	moses	nintendo	blackish	squeezed
portugal	sin	gamecube	silvery	blasted
sweden	heaven	psp	greyish	tangled
austria	salvation	amiga	paler	slashed

Dictionary size: 30,000 words. Even rare words are well embedded.

Modèles continus de langage, Collobert et al. 2008

MTL: Semantic Role Labeling





We get: 14.30%. State-of-the-art: 16.54% – Pradhan et al. (2004)

250× faster than state-of-the-art. ∼ 0.01s to label a WSJ sentence.

Modèles continus de langage, Collobert et al. 2008

MTL: Unified Network for NLP

Improved results with Multi-Task Learning (MTL)

Task	Alone	MTL
SRL	18.40%	14.30%
POS	2.95%	2.91%
Chunking – error rate	5.4%	4.9%
Chunking – F1-score	91.5%	93.6%


POS: state-of-the-art $\sim 3\%$


Chunking: Best system had 93.48% F1-score at CoNLL-2000 challenge `http://www.cnts.ua.ac.be/conll2000/chunking`. State-of-the art is 94.1%. We get 94.9% by using POS features.

Modèles continus de langage, Collobert et al. 2008

Summary


We developed a deep neural network architecture for NLP


Advantages

- General to any NLP tagging task
- State-of-the-art performance
- No hand designed features
- Joint training
- Can exploit massive unlabeled data
- Extremely fast: 0.02s for all tags of a sentence


Inconvenients

- Neural networks are a powerful tool: hard to handle


Early Impacts

- Easy to apply to other tasks or languages: extending to Japanese
- Fast: developed a semantic search system

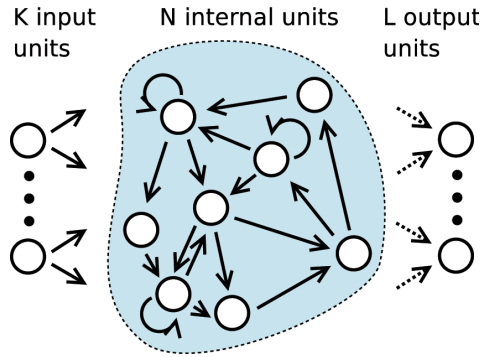
Réseaux Neuronaux

1. Pourquoi ?
2. Réseaux classiques
3. Quelques applications
4. Réseaux modernes: profonds, à échos, impulsionnels, liquides...
5. Pistes de recherche fraiches

Réseaux profonds

Bengio 2006

Echo State Networks — Jaeger 2001



NIPS 2005 Wshop, Reservoir Computing
$=$ Echo State Network, [Jaeger 2001]
$\bigcup$ Liquid State Machine, [Maas 2002]

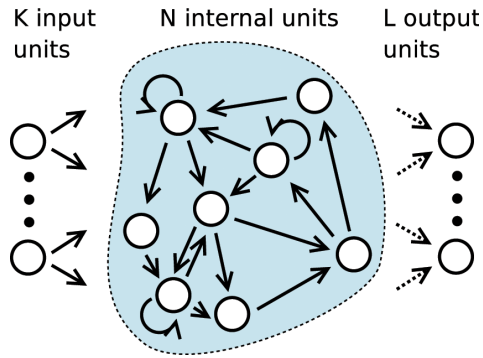
Structure

- N neurones cachés
- Connexions: aléatoires — matrice $\mathcal{G}$

$$p(x_i, x_j) \text{ connectés } = r, \; r << 1$$

- Poids: aléatoires: 1, -1, 0.
- Stabilité: max. valeur propre de $\mathcal{G}$ (damping factor) < 1
- Neurones de sortie: combinaison linéaire.

Echo State Networks, 2



A revolution

- The end of micro-management for NN (only ρ and λ)
- Training (e.g. for regression) through quadratic optimization

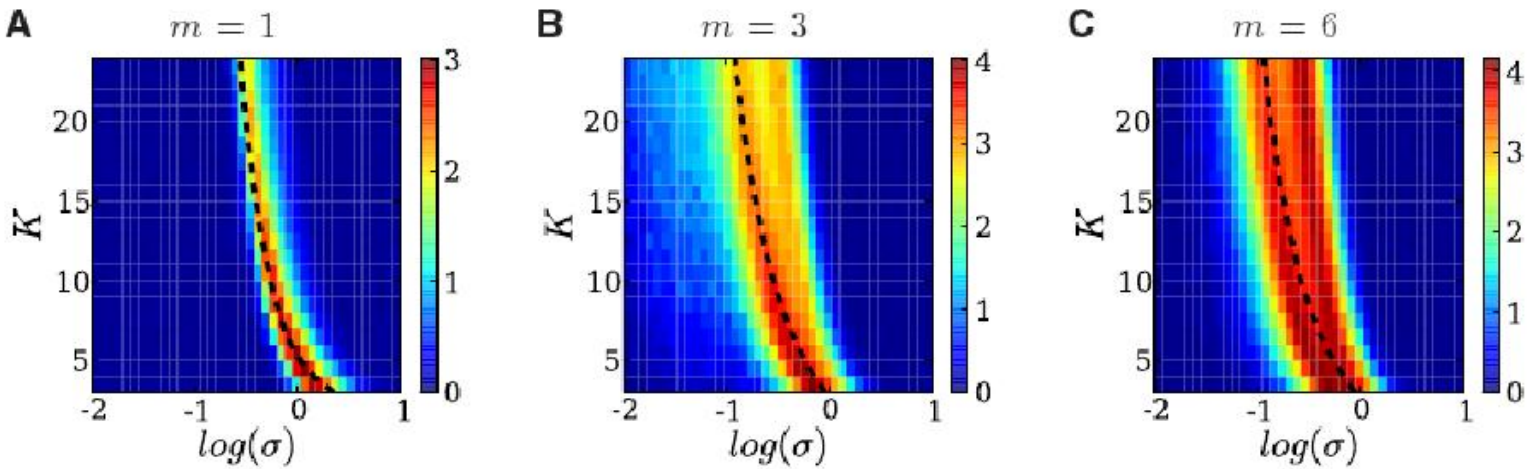
More:

On computational power and the Order Chaos: Phase Transition in Reservoir Computing, Benjamin Schrauder, Lars Büsing and Robert Legenstein, NIPS 2008.

Zone de compétence des Echo State Networks

Tâche

- Input: une séquence de bits
- Output désiré: la parité des τ derniers bits (non séparables, mémoire nécessaire)



- K: nombre de connexions d'un neurone
- σ: les poids sont gaussiens, tirés comme $\mathcal{N}(0, \sigma)$.

Réseaux Neuronaux

1. Pourquoi ?
2. Réseaux classiques
3. Quelques applications
4. Réseaux modernes: profonds, à échos, impulsionnels, liquides...
5. Pistes de recherche

Le point: On veut construire un système

Ce qu'on a compris

- Combinaison d'entités simples: arbitrairement complexe

Le point: On veut construire un système

Ce qu'on a compris

- Combinaison d'entités simples: arbitrairement complexe pourvu que les entités ne soient pas linéaires.
- Le réseau est une fonction de redescription des données.

Le point: On veut construire un système

Ce qu'on a compris

- Combinaison d'entités simples: arbitrairement complexe pourvu que les entités ne soient pas linéaires.
- Le réseau est une fonction de redescription des données.

On veut construire un système

1. Dans quel espace habite-t-il ?
   Décrit par un ensemble de poids ? par des macro-descripteurs (topologie, poids) ?
2. Comment l'entrainer ?
   Quel critère ? Quel algorithme d'optimisation ?
3. Comment coder la connaissance du domaine ?
   Dans les exemples ? Dans la structure de la solution ?