Réseaux Multicouches

Apprentissage et généralisation

1 - Le choix du modèle (résultats théoriques)

Plusieurs résultats théoriques généraux (biais-variance, dimension de Vapnik-Chernovenskis, …) ont montré que, l'objectif de l'apprentissage est de trouver le modèle dont la « complexité » réalise un bon « compromis » qui tient compte de la complexité du problème à traiter et de la taille de la base d'apprentissage.

La méthode adoptée consiste, considérant une suite emboîtée de L espaces de fonctions $(\boldsymbol{F}_l)_{l=1..L}$ de « richesse » croissante, dans le sens suivant :

$$\boldsymbol{F}_1 \subset \boldsymbol{F}_2 \subset \ldots\ldots\ldots \subset \boldsymbol{F}_{L-1} \subset \boldsymbol{F}_L$$

à appliquer l'algorithme « ***minimisation du risque structurel (MRS)*** » ci-dessous :

Étant donnée une même base d'apprentissage A_{pp} de taille N :

Pour tout l de 1 à L

- **A**ppliquer l'algorithme d'apprentissage sur la famille $\boldsymbol{F}_l$ ce qui permet d'obtenir une des fonctions f_l de $\boldsymbol{F}_l$. On note f_{l*} cette fonction particulière de $\boldsymbol{F}_l$ qui minimise l'erreur en apprentissage.
- **C**alculer l'erreur en test de la fonction f_{l*}.

Choisir, la fonction f_{l*} pour laquelle l'erreur en test est la plus petite.

1.2 - choix du modèle pour les réseaux multicouches

Plan

Plusieurs méthodes sont envisageables :

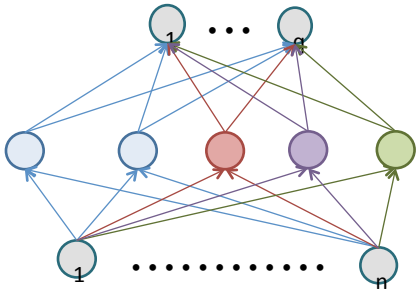
- Construction empirique de l'architecture
- Architecture à masques et poids partagés
- Régularisation de la fonction de coût
- Apprentissage avec bruit
- Suppression de poids (ou de cellules)

2 - Construction empirique de l'architecture

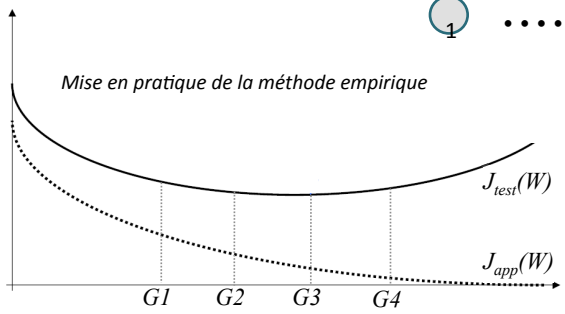
2.1- Familles de fonctions emboîtées de R^n dans R^q.

exemple ici avec une seule couche cachée et initialement 2 neurones :

$G_1 \subset G_2 \subset G_3 \subset G_4$:



Mise en pratique de la méthode empirique

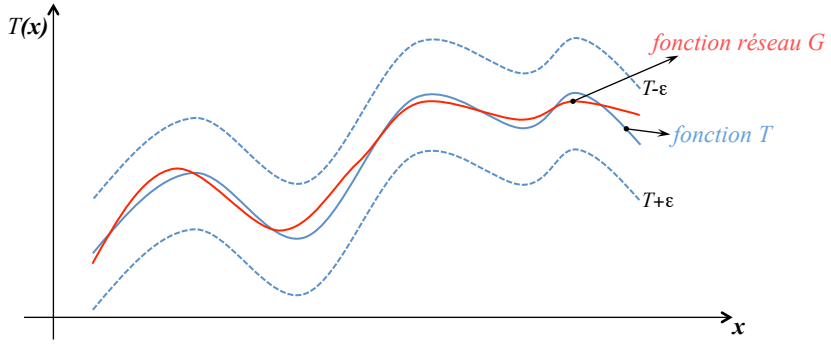


2.2 - Propriété d'approximation universelle des PMC

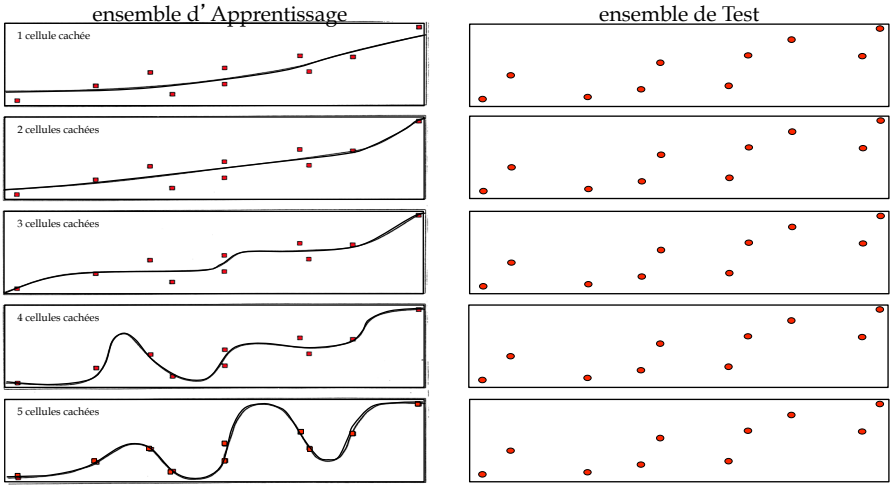
Théorème de Funahashi K.I.:

Soit $f(x)$ une fonction non constante, bornée, monotone croissante et continue. Soit K un sous ensemble compact de R^n.(« *par exemple une sphère de* R^n ») pour toute fonction T continue, $T : K \rightarrow R^q$, définie par $x = (x_1, x_2, ..., x_n) \rightarrow (T_1(x), ..., T_q(x))$, et pour tout $\varepsilon > 0$, il existe un réseau A à une couche cachée, dont la fonction d'activation est $f(x)$ pour chacune des cellules cachées et linéaire pour les cellules de sortie, tel que : $max_{x \in K}[d(T(x)\text{-}G(x,W))] < \varepsilon$

où W est la matrice des poids de connexions associées, $d(.)$ la métrique euclidienne de R^q et $G(.,W)$ la fonction générée par le PMC.

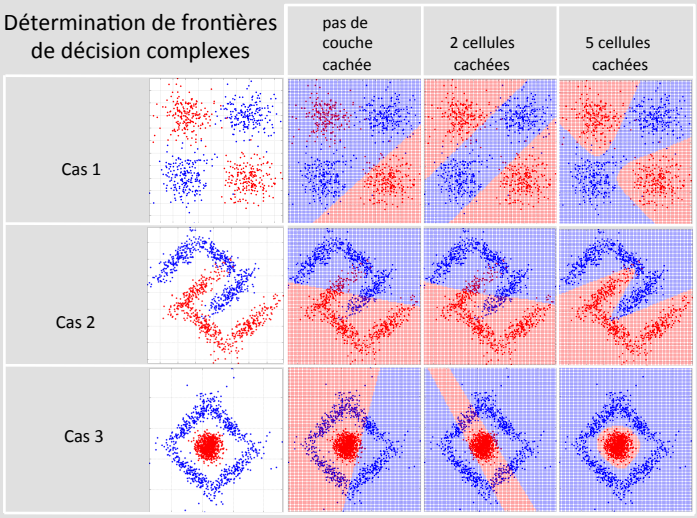


2.3 - Choix du meilleur réseau (régression)



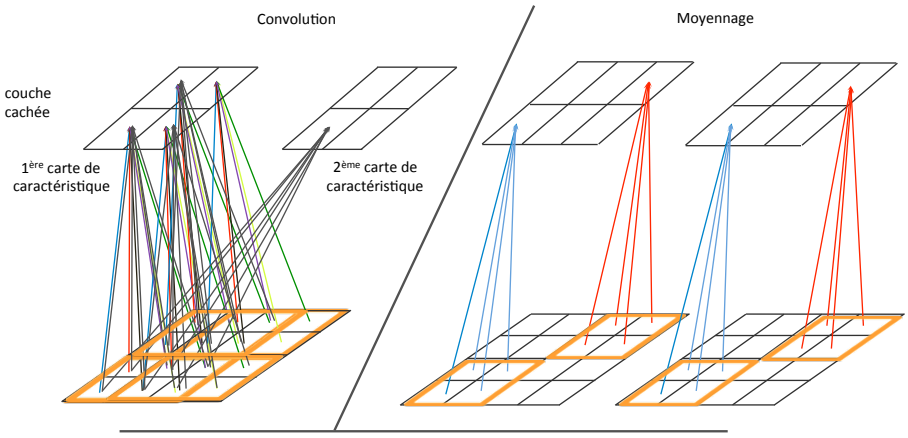
Unité cachées	E_{app}	E_{test}
1	0,012	0,0204
2	**0,0093**	**0,0147**
3	0,0077	0,0382
4	0,0043	0,0784
5	0,0019	0,1569

2.4 - Comparaison d'architectures (classification)



3 - Architectures à masques et poids partagés

3.1 - Présentation de la technique

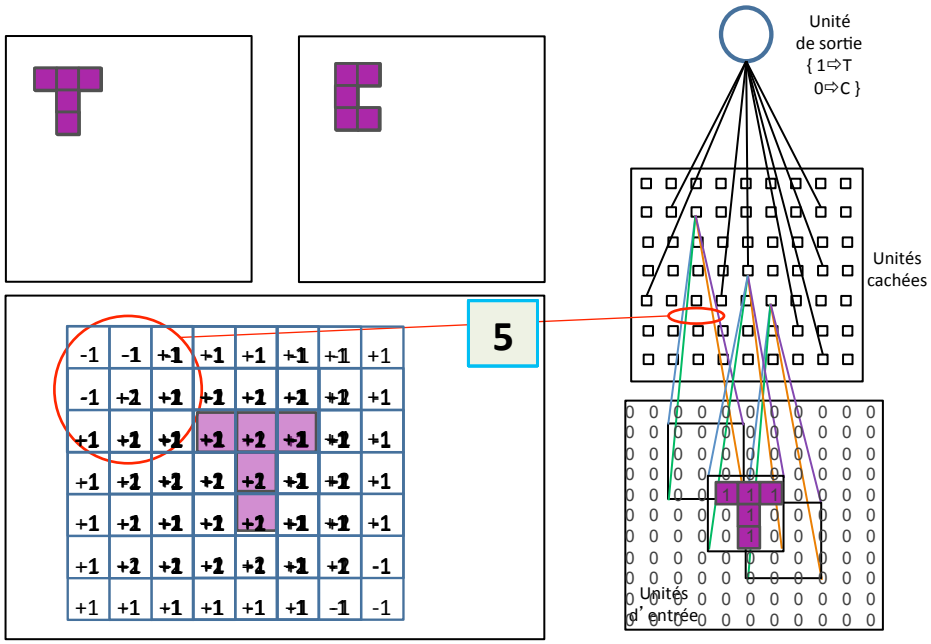


v connexions qui partagent la même valeur de poids w => : $w_1 = w_2 = \ldots = w_v = w$

Le gradient de la fonction de coût J peut s'écrire :

$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial w_1}\frac{\partial w_1}{\partial w} + \frac{\partial J}{\partial w_2}\frac{\partial w_2}{\partial w} + \ldots + \frac{\partial J}{\partial w_v}\frac{\partial w_v}{\partial w} = \sum_{i=1}^{v}\frac{\partial J}{\partial w_i} \quad \text{puisque que} \quad \frac{\partial w_i}{\partial w} = 1 \; \forall i$$

3.2 - The T-C Problem



3.3 - Exemple: Reconnaissance de chiffres manuscrits

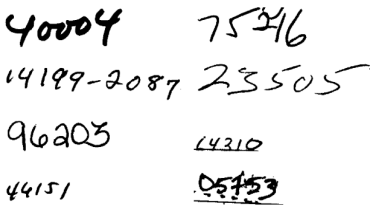


figure 1: exemples de codes postaux originaux extrait de la base d'apprentissage provenant d'une poste newyorkaise (Buffalo).

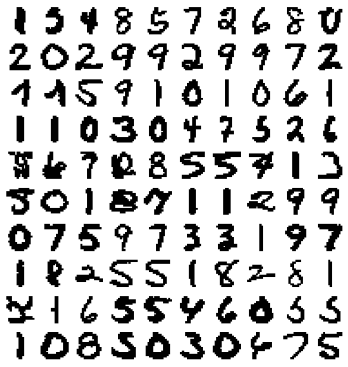
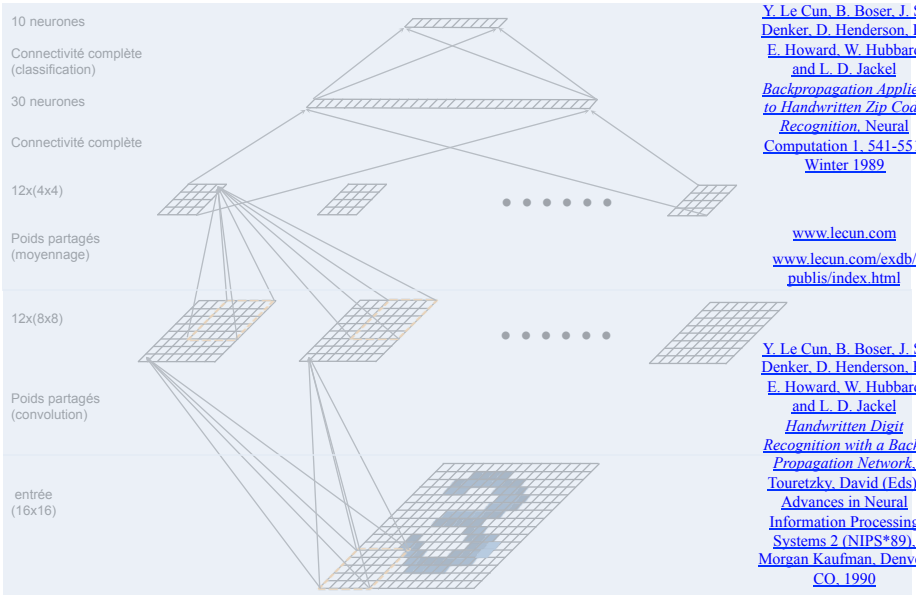


figure 2: exemples de chiffres normalisés après localisation et segmentation [Wang & Srihari, 1988]

Tiré de : Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel*Handwritten Digit Recognition with a Back-Propagation Network,* AT&T Bell Laboratories. Holmdel. N. J. 07733

commentaire

3.4 - LeNet [Y. LeCun 1989s]



Références

Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel *Backpropagation Applied to Handwritten Zip Code Recognition*, Neural Computation 1, 541-551, Winter 1989

www.lecun.com

www.lecun.com/exdb/publis/index.html

Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel *Handwritten Digit Recognition with a Back-Propagation Network*, Touretzky, David (Eds), Advances in Neural Information Processing Systems 2 (NIPS*89), Morgan Kaufman, Denver, CO, 1990

4 - Terme de pénalité dans la fonction de coût

4.1 - Principe

Soit la fonction $E_\lambda(W) = J(W) + \lambda R(W)$ à minimiser

où: - $J(W)$ l'erreur quadratique obtenue sur un ensemble d'apprentissage D

- $R(W)$ une fonction qui dépend des paramètres W et qui correspond à un terme de pénalité. On suppose $R(W) \geq 0$
- λ est un paramètre à déterminer.

Résultat 1

On démontre qu'il existe $k>0$ tel que minimiser $E_\lambda(W)$ est équivalent à :

minimiser $J(W)$ sous la contrainte $R(W) \leq k$

et donc, minimiser $E_\lambda(W)$ est équivalent à minimiser $J(W)$ dans la famille de fonctions

$$G_\lambda = \{ g \in G(.,W) \mid R(W) \leq k \}$$

Résultat 2

Soit λ_1 et λ_2 tel que $\lambda_1<\lambda_2$, alors on démontre que les 2 nombres k_1 et k_2 (qui découlent du résultat précédent) vérifient l'inégalité $k_2 \leq k_1$.

On a donc : $\lambda_1<\lambda_2 \Rightarrow G_{\lambda_2} \subset G_{\lambda_1}$

4.2 - Lien avec l'algorithme de minimisation du risque structurel

Étant donné : - une suite de paramètres $\lambda_L > \lambda_{L-1}, \ldots > \lambda_2 > \lambda_1$,
- une famille de fonctions $G(.,W)$ (définie par un PMC)
- une même base d'apprentissage A_{pp} de taille N :

l'algorithme « ***minimisation du risque structurel*** » devient :

Pour tout l de 1 à L

- Appliquer l'algorithme de minimisation de la fonction $E_{\lambda_l}(W)$ dans la famille de fonctions G, ce qui permet d'obtenir une fonction g_{l*} de G.
- Calculer l'erreur en test de la fonction g_{l*}.

Choisir, la fonction g_{l*} pour laquelle l'erreur en test est la plus petite.

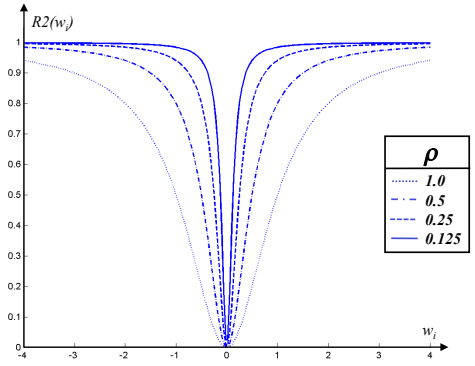
4.3 - Exemples de terme de pénalité

1) weight decay :

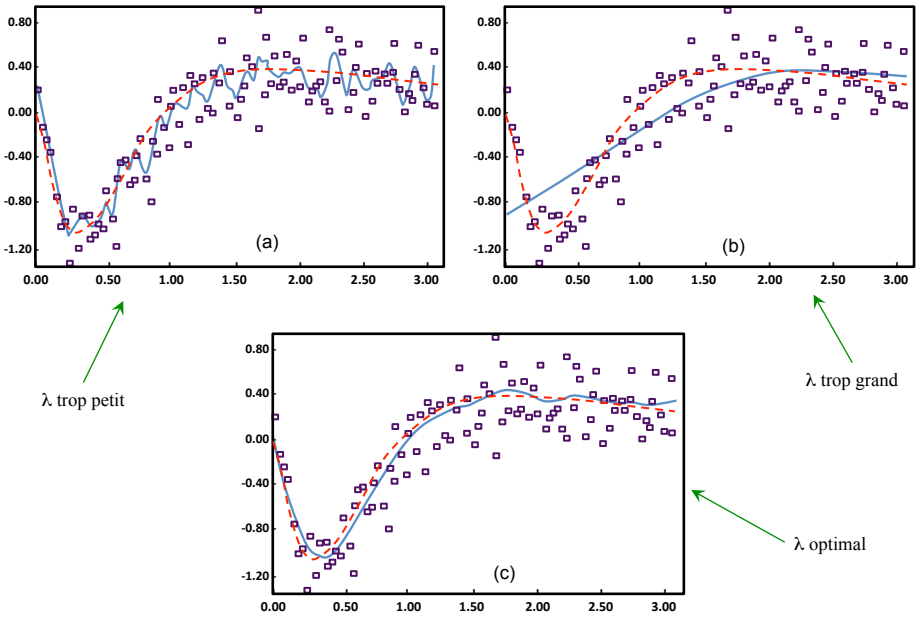
$$R1(W) = \sum_i w_i^2$$

2) :

$$R2(W) = \sum_i \left(\frac{\frac{w_i^2}{\rho^2}}{1 + \frac{w_i^2}{\rho^2}} \right)$$



4.4 - Illustration de l'impact du weight decay



commentaire

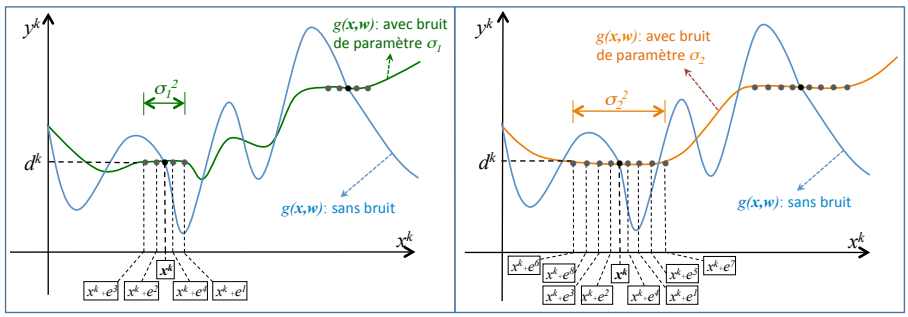
5 - Apprentissage avec bruit (sur les entrées)

- Soit une base d'apprentissage : $D = \{ (\mathbf{x}^1,\mathbf{d}^1), (\mathbf{x}^2,\mathbf{d}^2), \ldots, (\mathbf{x}^N,\mathbf{d}^N) \}$
- A chaque exemple $(\mathbf{x}^k,\mathbf{d}^k)$ on construit l autres exemples : $(\mathbf{x}^k+\mathbf{e}^i, \mathbf{d}^k)$ $i=1,2,\ldots,l$ où les e^i correspondent à des tirages indépendants suivant la loi normale $N(0,\sigma^2 I)$.
- On forme ainsi une base d'apprentissage D' ayant Nxl éléments.

⇨ On peut montrer que l'apprentissage du PMC sur D' correspond à minimiser

$$E(W) = J(W) + \sigma^2 \Omega(x) .$$

où $\Omega(x)$ est le terme de régularisation, et où σ intervient comme un paramètre de régularisation à déterminer.



2 illustrations avec $\sigma_1<\sigma_2$: En terme de généralisation, la courbe orange(σ_2) est préférable à la verte (σ_1) toutes deux étant préférables à la courbe bleue pour laquelle on a pas ajouté de bruit.

6 - Méthode de suppression de poids

6.1 - Exemple : OBD : Optimal Brain Damage

La pertinence d'un poids est estimée à partir de la formule de Taylor :

$$\delta J = J(w+\delta w) - J(w) = \sum_i \frac{\partial J}{\partial w_i}\bigg|_w \delta w_i + \frac{1}{2}\sum_i\sum_j \frac{\partial^2 J}{\partial w_i \partial w_j}\bigg|_w \delta w_i \delta w_j + o(\delta w_i^3) \quad (1)$$

En un minimum local ce premier terme est nul.

Sous l'hypothèse que la Hessienne est diagonale : $\delta J \approx \frac{1}{2}\sum_i \frac{\partial^2 J}{\partial w_i \partial w_i}\delta w_i^2$ (2)

Mesure de pertinence :

Partant d'un vecteur poids $w=(w_1, \ldots, w_k, \ldots, w_N)$ et d'une variation $\delta w=(0,\ldots,0,-w_k,0,\ldots,0)$ on obtient :

$$OBD(w_k) = \tfrac{1}{2} w_k^2 \frac{\partial^2 J}{\partial^2 w_k} \quad (3)$$

référence: Y. Lecun, J. S. Denker, S. A. Solla (1990). *Optimal Brain Damage*. In D. S. Touretsky (Ed), *Advances in Neural Information Processing Systems*, Volum 2, pp.598-605. San Mateo, CA: Morgan Kaufmann.

6.2 - Algorithme d'élimination de poids

Algorithme général

1) Faire une phase d'apprentissage complète, et calculer l'erreur en test.

2) Calculer la pertinence de chaque poids avec OBD.

3) Supprimer un (ou des) poids dont la pertinente est la plus petite.

4) Recommencer en 1) tant que l'erreur en test ne se dégrade pas.

Conclusion

Dans cette séance nous avons présenté l'algorithme de minimisation du risque structurel (MRS) qui consiste à construire des familles de fonctions emboîtées, et à retenir celle qui a la meilleure performance. Jusqu'à présent, nous avons exposé deux techniques qui permettent de mettre en œuvre cette méthode, il s'agit de :

- La construction empirique d'architectures, et des
- Architecture à masques et poids partagés.
- La régularisation de la fonction de coût
- L'apprentissage avec bruit, et
- La suppression de poids.

Ces sujets sont sous-tendus par des approches mathématiques plus complexes.