Authentification dynamique de signatures par réseaux de neurones

Dynamic Signature Authentification Using Neural Networks

Joël MINOT
Laboratoires d'Electronique Philips,
22, avenue Descartes, BP 15,
94453 Limeil-Brevannes Cedex.

Joël Minot a obtenu le diplôme d'ingénieur de l'École Nationale d'Ingénieurs Électriciens de Grenoble, spécialité traitement du signal, en 1987. Il est actuellement chargé de la réalisation d'applications dans le groupe réseaux de neurones, notamment dans les domaines de la reconnaissance de l'écriture et du traitement du signal.

Philippe GENTRIC
Laboratoires d'Electronique Philips,
22, avenue Descartes, BP 15,
94453 Limeil-Brevannes Cedex.

Philippe Gentric a obtenu le diplôme d'ingénieur de l'École Supérieure de Physique et de Chimie Industrielle en 1986, puis un doctorat en physique des semiconducteurs en 1989. Au sein du groupe réseaux de neurones, il travaille sur la reconnaissance de l'écriture.

RÉSUMÉ

Dans cet article, nous étudions le problème de l'authentification de signatures à partir des paramètres temporels de l'écriture.

Nous proposons ici une méthode qui, après normalisation des signaux, extrait des mesures de dissimilitude entre signatures. Ces mesures sont ensuite traitées par un réseau de neurones. Comme nous ne disposons pas toujours d'imitations de signatures, le problème se présente comme un problème à une classe statistique et l'utilisation d'algorithmes neuronaux classiques nous est donc interdite. Pour pallier ce type d'inconvénient, nous présentons également un nouvel algorithme d'apprentissage de réseau de neurones.

MOTS CLÉS

Authentification de signature, alignement temporel, réseaux de neurones.

ABSTRACT

This paper presents a signature authentication method based on handwriting temporal feature.

We propose here a method which, after signal normalization, extracts dissimilarity measures between signatures. Those measures are then processed by a neural network. As we do not always have forgeries, this problem is a statistical one-class problem and the use of classical learning algorithms is forbidden. This drawback is eliminated by the introduction of a new learning algorithm.

KEY WORDS

Signature authentication, time-warping, neural networks.

1. Introduction

Le besoin de vérifier l'identité d'un individu dans la société a toujours existé, et un des moyens les plus répandus est la signature écrite : il est communément utilisé dans les transactions bancaires ou les opérations d'accès à des ordinateurs.

L'identification des personnes par la signature présente deux avantages que ne possèdent pas simultanément les autres méthodes d'identification (mots de passe, codes numériques, cartes, clés, empreintes digitales, ...) [1] :

- l'identification par la signature est couramment acceptée, car les individus sont habitués à utiliser leur signature pour confirmer leur identité ;

• les signatures ne peuvent être volées, perdues ou cédées.

Les méthodes de vérification automatique de signatures sont divisées en deux groupes : l'inspection visuelle (la plus répandue) et le traitement de la dynamique de la signature. Comme l'inspection visuelle montre facilement ses limites dans la détection de contrefaçons (*fig.* 1), la méthode dynamique est une manière efficace d'obtenir des informations discriminantes. Les réseaux de neurones, grâce à leurs facultés d'adaptation et de robustesse [2], sont des outils performants pour contrecarrer les opérations frauduleuses.

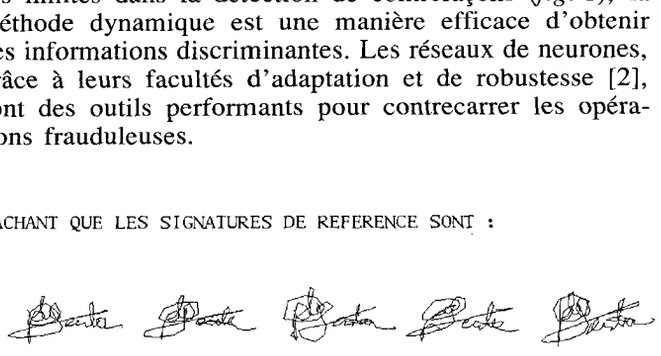

Figure 1. — L'authentification de signatures : la difficulté du problème.

Le processus d'authentification dynamique peut être divisé en quatre tâches successives :

• la position du crayon et la pression exercée sur la pointe du crayon sont enregistrées en fonction du temps ;

• la deuxième tâche est une tâche de comparaison : étant donné une nouvelle signature, comparons-la aux signatures connues et évaluons les différences ;

• les règles de décision sont données par la troisième tâche. C'est principalement un travail d'estimation de vraisemblance dont le but est de déterminer la probabilité d'authenticité d'une signature ;

• le protocole global de sécurité est la dernière tâche. Ce protocole dépend principalement de l'application (banque, accès informatique, ...).

2. Le capteur et la signature

L'étude de la signature a été abordée en termes de traitement de signal et non en termes de reconnaissance de formes. Pour cela, nous avons utilisé un capteur spécifique, conçu à LEP, qui mesure la position du crayon, $x(t)$ et $y(t)$, et la pression exercée sur la pointe du crayon, $p(t)$, en fonction du temps. Ces trois fonctions sont ensuite normalisées en taille, direction et durée. La renormalisation en durée est en fait un ré-échantillonnage temporel afin que toutes les signatures aient le même nombre d'échantillons temporels (la durée globale de chaque signature est évidemment conservée).

Les signatures ont été collectées auprès de douze personnes. Certaines personnes se sont entraînées à imiter les signatures de tierces personnes, en vue de constituer un ensemble de contrefaçons. Une signature est dite contrefaçon si le motif de cette signature ne peut être distingué visuellement du motif de l'originale. Cependant, comme il n'était pas possible de demander à chacun de signer des centaines de fois, les expériences ont été menées sur des petits ensembles et nous devrons tenir compte du biais introduit par cet effet de dimension.

3. Comparaison des signatures

Avant de mesurer une « distance » entre deux signatures (ou vecteurs de signaux), nous avons besoin de prétraiter ces vecteurs. La préparation consiste à trouver un appariement optimal entre les caractéristiques temporelles de deux signatures.

L'algorithme d'alignement temporel, combiné à la programmation dynamique (en anglais *Dynamic Time Warping* ou *DTW*), peut fournir des mesures de dissimilitudes tout en autorisant une variabilité locale du geste d'écriture, qui se traduit par des distorsions non linéaires entre échelles de temps. Cet algorithme apparaît alors comme une méthode efficace pour calculer rapidement un alignement optimal et en déduire une mesure de dissimilitudes [3], [4].

3.1. DÉTERMINATION DE LA FONCTION D'ALIGNEMENT TEMPOREL

3.1.1. L'alignement temporel

Considérons une signature de référence R et une signature de test T. Ces deux signatures sont représentées par deux séquences temporelles de paramètres caractéristiques, notées respectivement ρ et τ :

$$\rho = \{r_1, \ldots, r_i, \ldots, r_I\}$$
$$\tau = \{t_1, \ldots, t_j, \ldots, t_J\} .$$

On définit le plan i-j dans lequel ρ et τ sont développées le long des axes i et j respectivement. Les variations temporelles entre ρ et τ peuvent être décrites par une séquence de points $c = (i, j)$ de la forme :

$$F = \{c(1), \ldots, c(k), \ldots, c(K)\}$$

avec :

$$c(k) = (i(k), j(k)) .$$

Ainsi, l'instant $i(k)$ de la séquence ρ est aligné sur l'instant $j(k)$ de la séquence τ. Cette séquence F peut être considérée comme une fonction qui permet le passage de l'échelle de temps de ρ vers celle de τ. Cette fonction sera appelée chemin d'alignement.

Parmi tous les alignements possibles F, on recherche l'alignement optimal F*, donné par la minimisation d'une fonction de coût :

$$\min_{F} \sum_{k=1}^{K_F} d(c(k))$$

où d est une distance entre deux paramètres caractéristiques temporels :

$$d(c(k)) = \| r_{i(k)} - t_{j(k)} \| .$$

Pour réaliser cette minimisation, des contraintes sont introduites sur les chemins F :

• conditions sur les points de départ et d'arrivée : on impose que tous les chemins partent du début des séquences ρ et τ pour se diriger vers la fin de ces deux séquences.

$$\begin{aligned} i(1) &= 1 \\ j(1) &= 1 \\ i(K) &= I \\ j(K) &= J \end{aligned}$$

• i et j sont des fonctions monotones, croissantes et continues :

$$\begin{aligned} 0 \leqslant i(k) - i(k-1) \leqslant 1 \\ 0 \leqslant j(k) - j(k-1) \leqslant 1 \end{aligned}$$

• contrainte locale sur la pente : on veut s'interdire des pentes trop fortes ou trop faibles. En effet, rappelons qu'un des principes de l'alignement est de réduire les petites variations temporelles (pente comprise entre $\frac{1}{2}$ et 2).

$$\begin{aligned} i(k) - i(k-2) \geqslant 1 \\ j(k) - j(k-2) \geqslant 1 . \end{aligned}$$

3.1.2. Utilisation de la programmation dynamique

La programmation dynamique apporte une solution pratique et simple au problème de la recherche de l'alignement temporel optimal :

On définit :

• $d(i, j)$: la matrice des distances entre chaque paramètre de la signature R et chaque paramètre de la signature T.

$$d(i, j) = \| r_i - t_j \|$$

• $g(i, j)$: la distance cumulative le long du meilleur chemin d'alignement trouvé entre les points $c(1) = (1, 1)$ et (i, j). $g(I, J)$ correspondra au chemin optimal d'alignement F* que l'on cherche.

• $p(i, j)$: la matrice des directions locales du meilleur chemin F entre les points $c(1) = (1, 1)$ et (i, j).

Initialisation :

$$g(1, 1) = d(1, 1) .$$

Exécution pour $1 \leqslant i \leqslant I$ et pour $1 \leqslant j \leqslant J$:

On va parcourir le plan i-j pour calculer simultanément $p(i, j)$ et $g(i, j)$.

Construisons l'ensemble des points précédents du point (i, j) imposé par les contraintes de pente :

$$\begin{aligned} E_{ij} = & \{(i-1, j-1)\} \\ & \cup \left\{ \begin{array}{l} (i, j-1) \text{ si } p(i, j-1) \neq \text{HORIZONTALE} \\ \emptyset \text{ sinon} \end{array} \right\} \\ & \cup \left\{ \begin{array}{l} (i-1, j) \text{ si } p(i-1, j) \neq \text{VERTICALE} \\ 0 \text{ sinon} \end{array} \right\} \end{aligned}$$

Par exemple, on n'a pas le droit de venir du point $(i, j-1)$ si le point précédent $(i, j-1)$ était $(i, j-2)$ (direction HORIZONTALE).

$g(i, j)$, la meilleure distance cumulative, est calculée par :

$$g(i, j) = \min \{ g(k, \ell) \mid (k, \ell) \in E_{ij} \} + d(i, j) .$$

Puis on détermine la direction locale qu'aurait le chemin optimal s'il passait par le point (i, j) (en cas d'égalité des nouvelles valeurs de $g(i, j)$, on privilégie la direction DIAGONALE) :

$$p(i, j) = \left\{ \begin{array}{l} \text{DIAGONALE si } g(i, j) = g(i-1, j-1) + d(i, j) \\ \text{HORIZONTALE si } g(i, j) = g(i, j-1) + d(i, j) \\ \text{VERTICALE si } g(i, j) = g(i-1, j) + d(i, j) \end{array} \right.$$

où DIAGONALE, VERTICALE et HORIZONTALE sont des directions définies par rapport à la figure 2.

Extraction de la fonction d'alignement temporel :

Quand toutes les valeurs de $p(i, j)$ ont été calculées, on construit F* par un algorithme de chaînage arrière sur $p(i, j)$. Il suffit de suivre les directions mémorisées dans $p(i, j)$. Ceci permet de reconstruire le chemin d'alignement optimal F* en partant de (I, J) pour aller vers (1, 1) (*fig.* 2).

On remarquera, que par construction, si $\rho = \tau$, alors :

$$\begin{aligned} \forall i , \quad & d(i, i) = 0 \\ \forall (i, j) , \quad & d(i, j) = d(j, i) \\ \forall (i, j) , \quad & d(i, j) \geqslant 0 . \end{aligned}$$

Cela implique que F* est la fonction identité F_{Id}, telle que $c_{Id}(k) = (i(k), i(k))$.

3.1.3. Choix des paramètres caractéristiques de la signature

Le signal de pression a été choisi, comme paramètre caractéristique pour définir les séquences ρ et τ, pour les raisons suivantes :

1. ce paramètre contient plus d'informations que les autres (vitesse, position, énergie, ...) ;

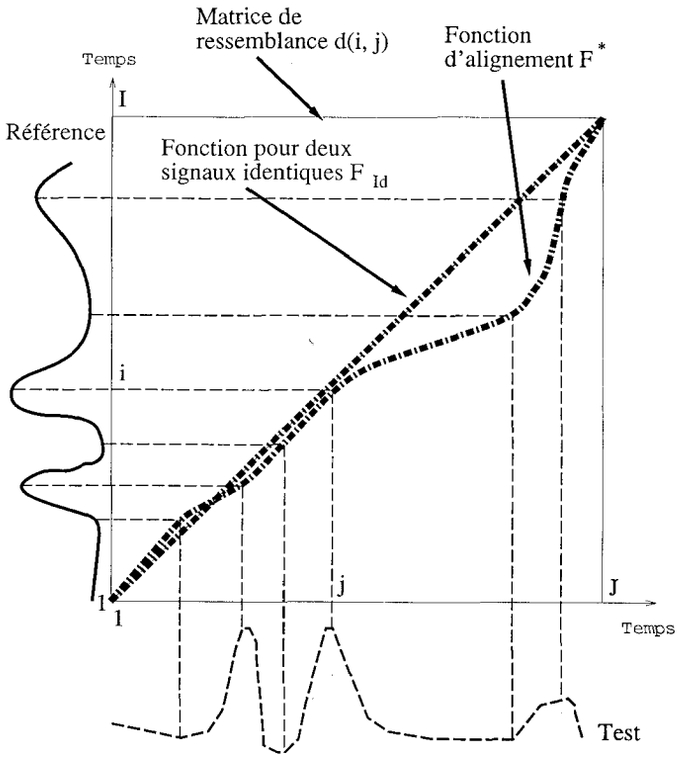

Figure 2. — Algorithme d'alignement temporel : la séquence de référence ρ (en gras), la séquence de test τ à aligner (en gras pointillé) et la matrice de ressemblance d(i, j) avec les fonctions d'alignement temporel F* et F_{Id} (en trait gras mixte).

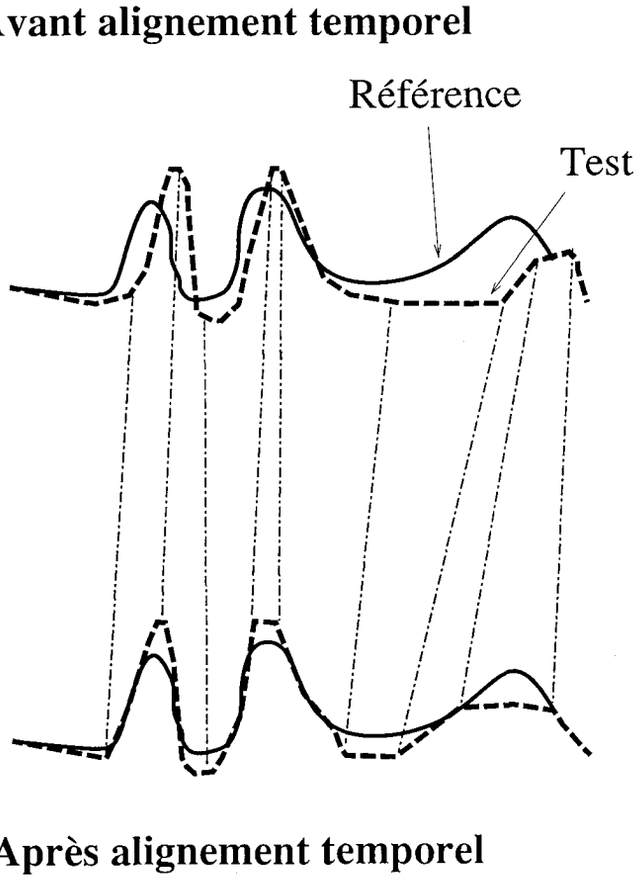

Figure 3. — Algorithme d'alignement temporel : le signal de référence (en gras) et le signal de test à aligner (en gras pointillé) avant alignement (a), après alignement (b).

2. cette information n'est jamais accessible au faussaire ;
3. ce paramètre est mono-dimensionnel. L'algorithme en est grandement simplifié.

Le chemin de ressemblance maximale F* donne une correspondance optimale entre l'ensemble de valeurs échantillonnées de la signature de référence et l'ensemble de valeurs échantillonnées de la signature à tester, en termes de minimisation des distorsions locales de pression. Quand cette fonction d'alignement temporel est appliquée sur la signature à tester T (*fig.* 3), la signature résultante T_a est appelée signature déformée et sera utilisée dans le calcul des mesures de distorsions.

3.2. MESURES DE DISTORSIONS

L'écriture est un mouvement quasi balistique [5], qui est une action rapide et automatique avec une très légère influence du système sensoriel ; ce type de mouvement dépend du système dynamique musculaire de chacun. Ainsi, un signataire ne signera jamais deux fois de la même façon (on peut même ajouter que si deux signatures sont identiques, alors l'une des deux est certainement une copie, donc une imitation). A cause des différences entre systèmes dynamiques musculaires, un faussaire peut ne pas être capable de contrefaire les caractéristiques du geste d'écriture même s'il recopie la forme de la signature originale et s'il l'écrit rapidement. De telles caractéristiques de l'écriture incitent à penser que l'on peut accroître la possibilité de détecter des imitations en extrayant les paramètres propres au geste d'écriture de chacun : pression, forme écrite, dynamique, durée totale [6].

Ces caractéristiques peuvent être extraites de manière efficace si l'on fait abstraction des distorsions temporelles par la méthode décrite précédemment.

- Notons D_p la distorsion moyenne de la pression et définissons-la comme l'erreur quadratique moyenne entre les signaux de pression de la signature de référence R et de la signature à tester T_a. La valeur mesurée D_p est égale à zéro quand les deux signaux de pression coïncident même si les signatures ont été écrites avec des dynamiques différentes.
- De même, il est possible de définir une mesure de distorsion des formes écrites, notée D_f, en calculant la distance moyenne entre la trajectoire du crayon de la signature de référence et celle de la signature déformée à tester T_a. Cette distance est l'erreur quadratique moyenne calculée à partir des coordonnées $x(t)$ et $y(t)$. D_f est égale à zéro quand les deux trajectoires coïncident même si elles ont été écrites avec des dynamiques différentes.

• Le problème est maintenant d'évaluer les différences de dynamique dans l'écriture. Rappelons que, quand les signatures ont été écrites avec la même dynamique, on a $F^* = F_{Id}$, c'est-à-dire que le chemin d'alignement est sur la diagonale de la matrice de ressemblance $d(i, j)$. Dans le cas général, les dynamiques sont différentes ($F^* \neq F_{Id}$) et donc le chemin d'alignement est représenté par une ligne brisée. Ainsi, on peut définir la différence de dynamique d'écriture D_d par l'écart quadratique moyen entre le chemin d'alignement et la diagonale.

• Enfin, la dernière mesure est plus globale que les précédentes. Elle est définie par le rapport des durées totales des deux signatures et est notée D_t.

3.3. LE CLASSIFIEUR NEURONAL

Les réseaux de neurones sont des machines parallèles qui allient robustesse et tolérance aux fautes. Quand ces réseaux sont organisés en couches, ils s'avèrent être des classifieurs très performants, qualité qui leur est conférée par les propriétés non linéaires des neurones formels. On appelle apprentissage le calcul des paramètres du réseau. Des algorithmes d'apprentissage permettent l'adaptation des réseaux à toute sorte de problème de classification et en particulier à des problèmes non modélisables par des approches classiques. Ainsi, la rétro-propagation du gradient de l'erreur [2] permet l'ajustement de surfaces de décisions entre les classes d'un problème statistique quelconque. Dans le cas de l'authentification de signatures, la modélisation de l'écriture (ou de paramètres dérivés) de chaque signataire est une tâche impensable. Ces considérations nous ont conduit à utiliser des techniques neuronales.

L'idée est de déterminer si une signature inconnue, notée S_x, est authentique ou non, étant donné que l'on possède a priori une base de données d'un signataire particulier. Si cette connaissance est constituée de N signatures authentiques $(R_i)_{i=1,\ldots,N}$, alors le vecteur suivant est formé pour chaque couple (R_i, S_x) :

$D(R_i, S_x) = \{D_f, D_d, D_p, D_t\}$. Puis, ces N vecteurs sont présentés successivement à un système neuronal qui donnera une décision.

Ce système est organisé en deux parties :

• La première partie estime, par une méthode neuronale, la vraisemblance que $D(R_i, S_x)$ appartienne au petit volume constitué par $\{D(R_k, R_j) \mid k = 1, \ldots, N ; j = k + 1, \ldots, N\}$ dans l'espace des paramètres (*fig.* 4). Cette partie est spécifique à un signataire donné.

• La deuxième partie est un réseau de neurones qui a été entraîné sur la sortie du premier réseau : il donnera une décision en fonction des N valeurs de vraisemblance correspondant aux N couples $(R_i, S_x)_{i=1,\ldots,N}$. Contrairement au premier réseau, ce réseau est commun à tous les signataires.

3.3.1. Le réseau d'estimation de vraisemblance

Principes :

La tâche de ce réseau est de rejeter les contrefaçons sans rejeter les bonnes signatures. Nous avons développé un algorithme spécial pour la construction de ce réseau. En effet, la détermination de la configuration du réseau par un algorithme d'apprentissage classique (rétro-propagation du gradient de l'erreur) nécessite un ensemble d'apprentissage comprenant des bonnes signatures et des imitations [7]. Cependant, dans un système opérationnel, il paraît impensable d'enregistrer des contrefaçons à chaque fois que l'on enrôle un signataire. Ainsi, le système devrait être capable soit d'engendrer des contrefaçons artificielles, soit de traiter des problèmes statistiques à une classe (c'est-à-dire ne pas tenir compte des contrefaçons pendant la phase d'apprentissage). La mise en œuvre de la première idée est très irréaliste : comment modéliser des contrefaçons réalistes ? Combien faut-il en créer ?... Nous avons donc étudié des réseaux neuronaux adaptés aux problèmes monoclasses.

Le mécanisme d'apprentissage étudié est fondé sur des considérations Bayésiennes [9], qui chercheront à modéliser la distribution des données dans l'espace des paramètres par des Gaussiennes multidimensionnelles. Par ailleurs, seuiller une vraisemblance sous une hypothèse

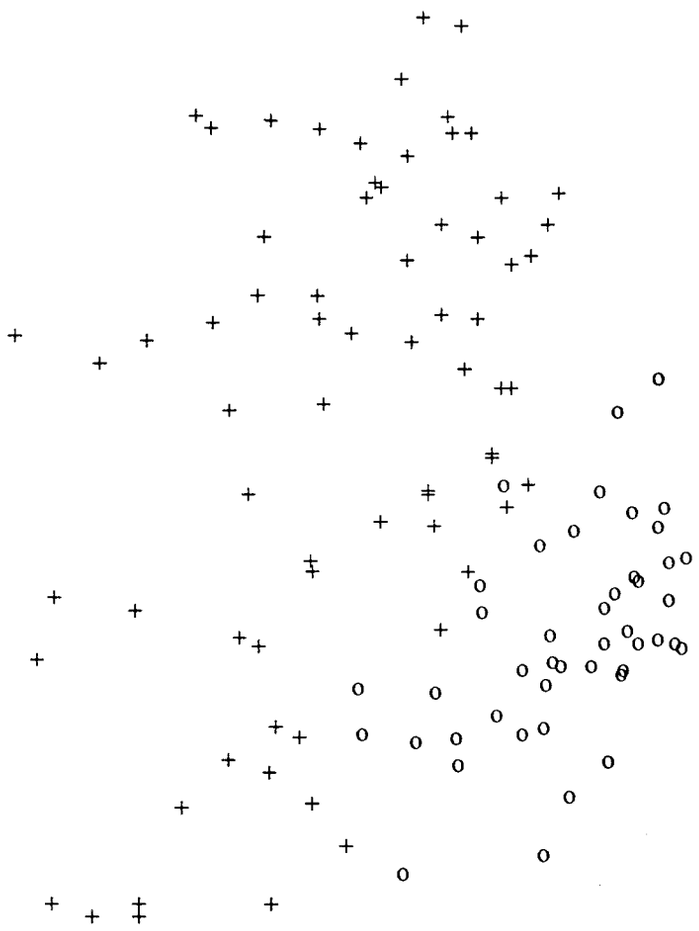

Figure 4. — Distribution des vecteurs de dissimilitude sur le plan principal après analyse en composantes principales : pour un signataire particulier, les dissimilitudes entre signatures authentiques sont marquées par un (○) et entre signatures authentiques et imitations par un (+). Les axes sont les axes principaux d'inertie.

Gaussienne conduit à l'équation d'un hyper-ellipsoïde dans l'espace des paramètres :

$$(X - \mu)\,\Gamma^{-1}(X - \mu)^T \leqslant \text{Seuil} \qquad (1)$$

$$\Leftrightarrow \sum\sum a_{ij}\,x_i\,x_j + \sum b_k\,x_k + c \geqslant 0\,. \qquad (2)$$

Notons $Y(x)$ la sommation constituant le premier membre de l'inéquation (2). Cette sommation (de type somme pondérée d'excitations), notée Y, peut être interprétée comme l'évaluation d'un neurone formel ayant pour entrée le vecteur $(-1, x_1, \ldots, x_n, x_1 x_1, \ldots, x_1 x_n, \ldots, x_n x_n)$ où le vecteur $X = (x_1, \ldots, x_n)$ représente les coordonnées d'un prototype dans l'espace des paramètres, μ représente le centre de la distribution et Γ sa matrice de covariance. La non-linéarité associée à chaque neurone formel (généralement la fonction tangente hyperbolique) permet de seuiller cette somme et « d'adoucir » les contours de l'hyper-ellipsoïde. Ainsi, pour un hyper-ellipsoïde défini uniquement par μ, Γ et Seuil, les données, la sommation Y est calculée directement et le neurone formel défini par $Y(x)$ permet de donner le degré d'appartenance d'un prototype à cet hyper-ellipsoïde :

$\tanh(Y(x)) \geqslant 0 \Leftrightarrow$ le prototype $\in$ hyper-ellipsöide (μ, Γ, Seuil)

où Seuil est un paramètre ajustant l'étendue de l'hyper-ellipsoïde.

De cette façon, si un prototype n'appartient à aucun hyper-ellipsoïde, il sera rejeté comme étant inconnu ; ou dans notre problème d'authentification de signature, il sera considéré comme une contrefaçon.

Cependant, étant donné la complexité du problème, l'hypothèse que l'ensemble des données ait une distribution Gaussienne paraît peu réaliste, aussi nous morcellerons le problème en un mélange de distributions Gaussiennes (*fig.* 5). En outre, comme la quantité de données disponibles pour ce problème d'authentification n'est pas importante, l'hypothèse faite sera encore plus souple puisqu'on considérera que chaque prototype à apprendre appartient à une nouvelle distribution Gaussienne. L'apprentissage des poids $\{a_{ij}\}$, $\{b_k\}$ et c peut alors se faire de deux façons :

- le centre de l'hyper-ellipsoïde est déterminé par chaque prototype de la base d'apprentissage, la forme de l'hyper-ellipsoïde, donnée par la matrice de covariance, restant fixée [9] ;

- le centre et la forme de l'hyper-ellipsoïde (covariances) sont adaptatifs : on considère en fait que chaque prototype et ses k plus proches voisins (selon une norme quadratique) dans l'ensemble d'apprentissage ont une distribution Gaussienne.

En ce sens notre algorithme s'apparente aux fonctions à bases radiales [8]. Ces réseaux de neurones utilisent des neurones dont la non-linéarité est une fonction à symétrie radiale (par exemple $\exp(-y)$) et dont l'entrée est une fonction quadratique (usuellement une distance). En général, de tels réseaux supposent que la matrice de covariance Γ est diagonale (problème à symétrie sphérique), voire que ses coefficients sont tous égaux. Notre approche est

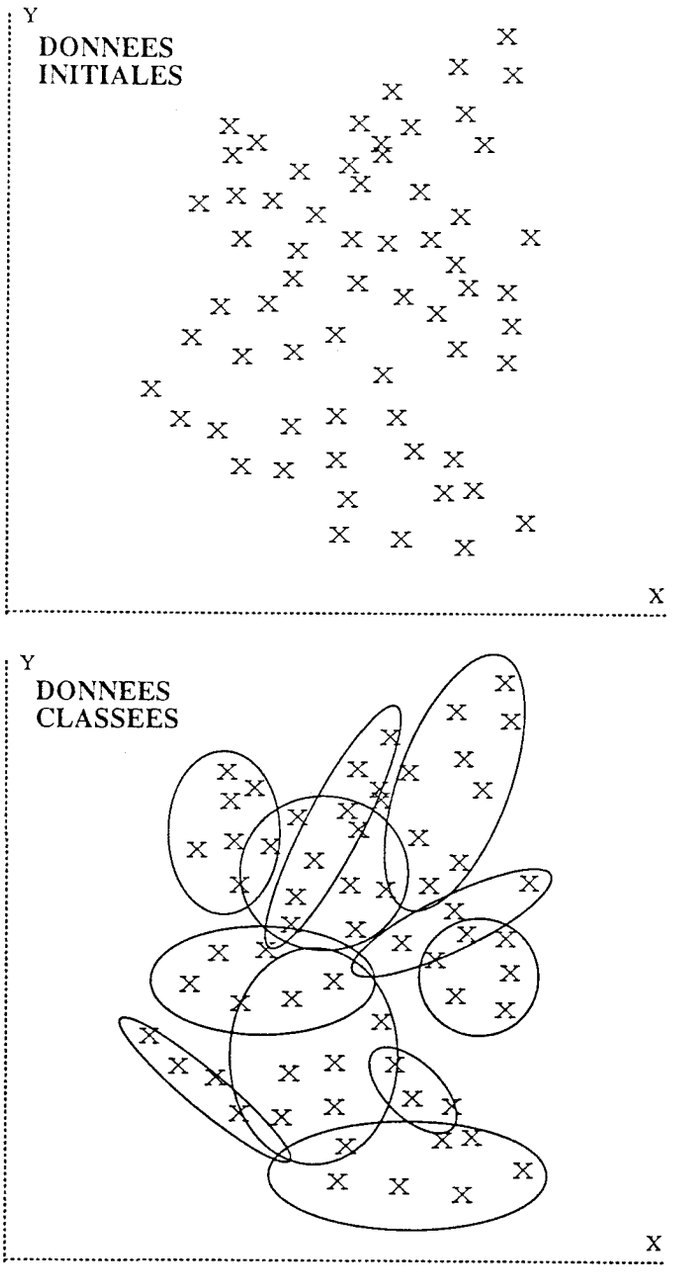

Figure 5. — Comment approximer une distribution de points par un mélange de gaussiennes ? (Les Gaussiennes ne sont pas toutes représentées.)

plus générale en ce que les hyper-ellipsoïdes ont une orientation et une forme quelconques dans l'espace.

L'apprentissage :

Pour chaque classe statistique C de l'ensemble d'apprentissage (dans l'exemple de la signature, il n'y a qu'une classe) :

1. Pour chaque prototype P de la classe C :

   On recherche les k plus proches voisins de P et appartenant à la classe C. Soit V_p l'ensemble de ces voisins.

On calcule le centroïde et la matrice de covariance des points de V_p.
On calcule les coefficients de l'hyper-ellipsoïde attaché à P en identifiant terme à terme les coefficients des inéquations 1 et 2.

2. La partie de la couche cachée du réseau correspondant à la classe C est ainsi construite (*fig.* 6). Il y a autant de neurones cachés dans cette partie qu'il a de prototypes P dans la classe C. Ces neurones sont tous connectés aux neurones d'entrée (appelés neurones quadratiques).
3. Un neurone de sortie (appelé neurone de classification de la classe C) somme les états de tous les neurones cachés créés pour la classe C et compare cette valeur cumulative à un seuil. Cela permet d'introduire le concept de vote majoritaire à l'intérieur du réseau : l'état du neurone de sortie sera actif quand un certain nombre de neurones cachés seront actifs. Ce seuil peut être fixé manuellement ou, comme dans notre application, fixé au nombre de prototypes appris pour la classe C (qui est égal au nombre de neurones hyper-ellipsoïdes de la classe C). Cependant, comme le réseau de neurones ainsi construit contient autant de neurones cachés qu'il existe d'exemples dans l'ensemble d'apprentissage, une phase d'optimisation pourrait éventuellement réduire ce nombre [9], [10].

Avantages et inconvénients :

Les avantages de cet algorithme sont les suivants :

- Problèmes à une classe résolus ;
- Généralisation aux problèmes multi-classes aisée ;
- Apprentissage très rapide (non itératif) ;
- Possibilité d'apprentissage incrémental (apprentissage de nouveautés) par l'ajout de neurones cachés ;
- L'information contenue dans les poids synaptiques est justifiable par des hypothèses statistiques maîtrisées ;
- L'information contenue dans les poids synaptiques est localisée ;
- Les structures du réseau et du neurone formel utilisé sont simples.

Les inconvénients :

- Le nombre de neurones cachés peut devenir important si la base d'apprentissage est importante ;
- On trouve en entrée du réseau des valeurs quadratiques $\{x_i x_j\}$.

3.3.2. Le réseau de décision

Ce réseau est un perceptron à une couche cachée et l'apprentissage a été réalisé par la méthode classique de rétropropagation du gradient [2], décrite ci-dessous.

En effet, à chaque présentation d'un exemple au réseau, les états des neurones sont calculés couche par couche de l'entrée vers la sortie du réseau. La sortie ainsi calculée est comparée à la sortie désirée par le professeur (apprentissage supervisé). Les poids synaptiques sont ensuite modifiés afin de minimiser le gradient de l'erreur sur les neurones de sortie. Ces modifications se font couche par couche de la sortie vers l'entrée du réseau. Cette méthode

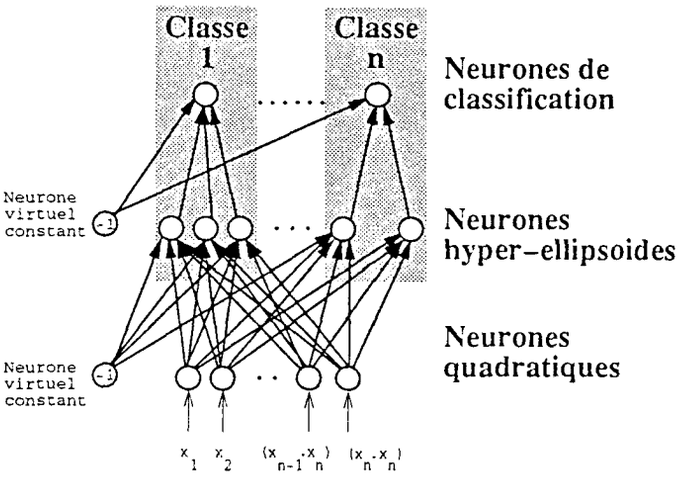

Figure 6. — Structure du réseau d'estimation de vraisemblance.

de descente en gradient se fait sur tous les exemples de l'ensemble d'apprentissage. Généralement, un certain nombre d'itérations est requis avant d'avoir une convergence du réseau (lorsque le gradient de l'erreur est quasi-nul). Cette méthode bien que très connue, doit être appliquée avec précautions : convergence locale, apprentissage par cœur, influence des facteurs aléatoires, ajustements des gains et de la structure du réseau délicats, ...

Cependant, cet algorithme donne généralement de très bons classifieurs, même pour des problèmes dont la topologie est complexe.

4. Résultats

Les méthodes exposées ci-dessus ont été appliquées à douze ensembles comportant dix signatures authentiques et quatre imitations. Ces imitations ont été obtenues en entraînant le faussaire jusqu'à ce qu'il soit capable de réaliser des faux impossibles à distinguer VISUELLEMENT des signatures authentiques. L'ensemble d'apprentissage pour chaque signataire est constitué par 5 signatures authentiques prises au hasard, les autres signatures étant rassemblées dans l'ensemble de test.

A partir des N = 5 signatures de références de chaque signataire, nous avons constitué 10 mesures de dissimilitudes qui ont permis de créer un réseau d'estimation de vraisemblance spécifique à chaque signataire. Les temps de calcul mesurés sur une station SUN SPARC 1 donnent environ 25 secondes pour la construction des vecteurs de dissimilitude et environ 1 seconde pour la construction du réseau d'estimation de vraisemblance. Le test du réseau se fait grâce aux 9 autres signatures disponibles pour chaque signataire.

Le réseau de décision nécessite la construction de tous les réseaux (un par signataire) et le calcul de la réponse de chaque réseau à ses 14 signatures propres. On obtient ainsi un ensemble de 168 vecteurs (contenant les N estimations

de vraisemblance). Cet ensemble est divisé aléatoirement en deux parties égales qui serviront respectivement à l'apprentissage et au test du réseau de décision. Compte tenu du faible nombre d'exemples, les performances se dégradent rapidement quand le nombre de neurones cachés croît [10]. L'apprentissage le meilleur (sur l'ensemble de test) a donné un réseau en couche possédant une couche cachée de 5 neurones et une couche de sortie de 2 neurones, et a été atteint en 30 itérations.

Les résultats mesurés sur les signatures de l'ensemble de test (9 signatures) et moyennés sur tous les signataires, donnent environ 5 % de fausses acceptations (FA) et 5 % de fausses rejections (FR) pour la sortie du premier réseau de neurones. Le deuxième réseau accroît la discrimination et fournit 4 % de FA pour moins de 2 % de FR.

Ces résultats peuvent être comparés à un estimateur Gaussien classique, même si le volume de notre base de données introduit un grand biais. Comparé à la sortie du premier étage du réseau, nous avons trouvé que le classifieur Gaussien donne 8 % de FA et 8 % de FR.

Évidemment, le taux de rejection sur des signatures visuellement différentes (par exemple, une signature de M. Dupond contre une signature de M. Martin) est de 100 %, car les dissimilitudes mesurées par notre méthode sont très grandes.

5. Conclusions

Les réseaux de neurones, combinés à une approche classique de traitement du signal, montrent leur puissance à travers cette application. En outre, nous envisageons les améliorations suivantes :

- les signatures évoluent avec le temps selon les conditions ou l'âge du signataire. Comment cela influe-t-il sur les performances ? Les capacités de ré-apprentissage de notre réseau de neurones spécifique seront essentielles à cet égard ;
- les performances d'un système d'authentification de signatures dépendent sévèrement du protocole de sécurité (nombre d'essais, seuils d'acceptation adaptatifs, ...) : les réseaux de neurones pourraient être utilisés pour la conception automatique de protocoles personnalisés.

Manuscrit reçu le 2 décembre 1991.

BIBLIOGRAPHIE

[1] D. F. Shaw, *Proof of identity. A review*, Proc. 1980 of Int. Conf. on security through science and engineering, Berlin (1980), 31-47.

[2] R. Lippman, *An introduction to computing with neural nets*, IEEE ASSP Mag., 3, no. 4 (1987), 4-22.

[3] R. de Mori, D. Probst, *Computer recognition of speech*, In : Handbook of Pattern Recognition and Image Processing, Academic Press, New York (1986), 499-525.

[4] H. Sakoe, S. Chiba, *Dynamic programming optimization for spoken word recognition*, IEEE Trans. Acoust. Speech Signal Process, 26 (1978), 43-49.

[5] R. Plamondon F. J. Maarse, *An evaluation of motor models for handwriting*, IEEE Trans. Syst. Man Cybern., 19 (1989), 1060-1072.

[6] Y. Sato, K. Kogure, *Online signature verification based on shape, motion and writing pressure*, Proc. 6th Int. Conf. on Pattern Recognition, IEEE Cat. CH1801-0/82 (1982), 823-826.

[7] D. A. Mighell, T. S. Wilkinson, J. W. Goodman, *Backpropagation and its application to handwritten signature verification*, In : Advances in neural information processing systems, Vol. I, Morgan Kaufmann, San Matea CA (1989), 340-347.

[8] J. Park, I. W. Sandberg, *Universal approximation using Radial-Basis-Function networks*, Neural Computation, 3 (1991), 246-257.

[9] P. A. Jokinen, *Neural networks with dynamic capacity allocation and quadratic function neurons*, Proc. 1990 of Neuronimes, Nîmes (1990), 351-362.

[10] J. Sietsma, R. Dow, *Creating artificial neural networks that generalize*, Neural Networks, 4 (1991), 67-79.