

N° Ordre : /2010-M/EL

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE DES SCIENCES ET DE LA THECHNOLOGIE HOUARI BOUMEDIENE
FACULTE D'ELECTRONIQUE ET INFORMATIQUE



Mémoire présenté par :

SELMANE Fares

Pour l'obtention du diplôme de :

MAGISTER EN ELECTRONIQUE

Spécialité : Microondes et Radiofréquences

THEME

**DÉVELOPPEMENT D'UN OUTIL DE CAO POUR
MODÉLISATION PAR RÉSEAUX DE NEURONES**

Soutenu publiquement, le 21/12/2010, devant le jury composé de :

M. B. Haddad	Professeur	Faculté E-I., USTHB	Président
M. M.C.E. Yagoub	Professeur	Université d'Ottawa	Rapporteur
M. A. Amrouche	Maître de Conférences	Faculté E-I., USTHB	Examineur
Mme N. Achour	Maître de Conférences	Faculté E-I., USTHB	Examinatrice
M. S. Gaoua	Docteur	Faculté E-I., USTHB	Invité

DEDICACES

Je dédie ce modeste travail à :

Ma famille,

Mes amis.

REMERCIEMENTS

Le travail présenté dans ce mémoire est réalisé au Laboratoire d'instrumentation de la Faculté d'Electronique et d'Informatique (FEI) de l'USTHB. Il s'inscrit dans le cadre de la préparation d'un Magister en Radiofréquence et Microonde.

Je tiens en premier lieu à remercier sincèrement mon responsable de mémoire de Magister en l'occurrence Monsieur le Professeur M.C.E. Yagoub, pour avoir proposé et dirigé ce travail, pour m'avoir accueilli dans son équipe et donné les moyens d'effectuer mon travail de recherche dans de très bonnes conditions et pour son soutien et ses encouragements pendant toute la durée de ce mémoire.

Je tiens à exprimer ma reconnaissance à Monsieur B. HADDAD, professeur à l'USTHB, pour l'honneur qu'il me fait en acceptant de présider le jury de ce mémoire.

J'exprime mes vifs remerciements à Monsieur Mr A. Amrouche, Maître de conférences à USTHB, Madame N. Achour, Maître de Conférences à l'USTHB, et Monsieur S. Gaoua, Docteur à l'USTHB pour avoir accepté de faire partie du jury de ce mémoire.

Mes remerciements vont également à tous les membres du laboratoire d'instrumentation de la FEI de l'USTHB, pour leur sympathie et la bonne ambiance permanente au laboratoire.

Sommaire

Liste Des Figures	3
Liste d'abréviations	4
Introduction Générale.....	5
Outils de modélisation.....	7
1. Introduction.....	7
2. Procédure de conception	7
3. Techniques de conception des circuits	9
3.1. Procédure de conception conventionnelle	9
3.2. L'approche CAO.....	10
4. Utilisation des réseaux de neurones dans la conception des circuits Micro-ondes	18
5. Conclusion.....	19
Théorie sur les réseaux de neurones.....	20
1. Introduction.....	20
1.1. Définitions	20
1.2. Évolution des réseaux de neurones artificiels	21
2. Structure du réseau de neurones.....	23
2.1. Modèle de neurone.....	23
2.2. Différents types de réseaux de neurones	23
3. Apprentissage des réseaux de neurones	33
3.1. Algorithmes de minimisation.....	34
3.2. Choix de la fonction de coût	36
3.3. Le problème de sous- et sur-apprentissage	38
4. Test du modèle neuronal.....	40
5. Conclusion.....	41
Développement de l'outil neuronal.....	43
1. Introduction.....	43
2. Développement du modèle de réseau de neurone	43
2.1. Génération de données	43

2.2. Choix de la structure	47
2.3. Effectuer l'apprentissage	47
2.4. Test du modèle neuronal.....	51
3. Conclusion.....	52
Applications de l'outil neuronal.....	53
1. Introduction.....	53
2. Description de l'outil neuronal développé	54
3. Résultats Obtenus.....	57
3.1. Modélisation de composants micro-ondes	57
3.2. Modélisation de circuits micro-ondes	70
4. Conclusion.....	76
Conclusion et perspectives.....	77
Bibliographie.....	80

Liste Des Figures

FIG. 1.1. L'ORDRE DES ETAPES D'UN PROCÉDE DE CONCEPTION	8
FIG. 1.2. REPRESENTATION BOITE NOIRE D'UN SYSTEME.	17
FIG. 2.1. MODELE DE NEURONE ARTIFICIEL.....	23
FIG. 2.2. RESEAU FEEDFORWARD A UNE SEULE COUCHE	24
FIG. 2.3. ILLUSTRATION DE LA STRUCTURE PERCEPTRONS MULTICOUCHE FEEDFORWARD (MLP).	27
FIG. 2.4. ILLUSTRATION DE LA STRUCTURE KBNN.	30
FIG. 2.5. RESEAU RECURRENT AVEC NEURONES CACHEES.	32
FIG. 3.1. ORGANIGRAMME DES ETAPES CONVENTIONNELLES NECESSAIRES POUR LE DEVELOPPEMENT DU MODELE NEURAL	44
FIG. 3.1. (A) REDIMENSIONNEMENT D'ENTREE-SORTIE DANS LE DEVELOPPEMENT DU MODELE NEURONAL ET (B) DANS L'UTILISATION DU MODELE NEURONAL	46
FIG. 4.1. FICHIER DE DONNEES AVEC N+M COLONNES POUR N ENTREES ET M SORTIES	55
FIG. 4.2. CHOIX DE LA STRUCTURE DU RESEAU DE NEURONES	55
FIG. 4.3. RESEAU MLP A TROIS COUCHES OU MLP3 (UNE COUCHE CACHEE).....	56
FIG. 4.4. CHOIX DE LA FONCTION D'ACTIVATION.	56
FIG. 4.5. CREATION DU RESEAU ET DEFINITION DES PARAMETRES D'APPRENTISSAGE.	57
FIG. 4.6. EVOLUTION DE L'ERREUR D'APPRENTISSAGE EN FONCTION DU NOMBRE D'ITERATION.....	57
FIG. 4.7. A- STRUCTURE PHYSIQUE 3-D D'UNE INDUCTANCE ENCASTREE. LA FREQUENCE EST LA SEULE VARIABLE ADJUSTABLE DANS LA CONCEPTION DE CE COMPOSANT MULTICOUCHE.	58
FIG.4.8.A RESULTATS D'APPRENTISSAGE POUR L'AMPLITUDE ET LA PHASE DU PARAMETRE S_{11} D'UNE INDUCTANCE AVEC 67 ECHANTILLONS, 200 ITERATIONS.....	61
FIG. 4.8.B. RESULTATS D'APPRENTISSAGE POUR L'AMPLITUDE ET LA PHASE DU PARAMETRE S_{12} D'UNE INDUCTANCE AVEC 67 ECHANTILLONS, 200 ITERATIONS.....	62
FIG. 4.9. A- STRUCTURE PHYSIQUE 3-D D'UNE RESISTANCE ENCASTREE. FREQUENCE, LONGUEUR, LARGEUR, PERMITTIVITE ET RESISTANCE SONT DES VARIABLES ADJUSTABLE DANS LA CONCEPTION DE CE COMPOSANT MULTICOUCHE. B- STRUCTURE DU MODELE DE RESEAUX DE NEURONES	63
FIG. 4.10. RESULTATS D'APPRENTISSAGE POUR LES PARTIES REELLES DES PARAMETRES S_{11} ET S_{12} D'UNE RESISTANCE AVEC 14750 ECHANTILLONS, 200 ITERATIONS	64
FIG. 4.11.A. RESULTATS D'APPRENTISSAGE POUR L'AMPLITUDE ET LA PHASE DU PARAMETRE S_{11} D'UN TRANSISTOR AVEC 121 ECHANTILLONS, 200 ITERATIONS.....	66
FIG. 4.11.B. RESULTATS D'APPRENTISSAGE POUR L'AMPLITUDE ET LA PHASE DU PARAMETRE S_{12} D'UN TRANSISTOR AVEC 121 ECHANTILLONS, 200 ITERATIONS.....	67
FIG. 4.11.C. RESULTATS D'APPRENTISSAGE POUR L'AMPLITUDE ET LA PHASE DU PARAMETRE S_{21} D'UN TRANSISTOR AVEC 121 ECHANTILLONS, 200 ITERATIONS.....	68
FIG. 4.11.D. RESULTATS D'APPRENTISSAGE POUR L'AMPLITUDE ET LA PHASE DU PARAMETRE S_{22} D'UN TRANSISTOR AVEC 121 ECHANTILLONS, 200 ITERATIONS.....	69
FIG. 4.12. SCHEMA DE L'AMPLIFICATEUR DE PUISSANCE UTILISE.	70
FIG. 4.13.A. RESULTATS D'APPRENTISSAGE POUR LA PARTIE REELLE DES PARAMETRES S_{11} ET S_{22} D'UN AMPLIFICATEUR DE PUISSANCE AVEC 91 ECHANTILLONS ET 15 NEURONES,	72
FIG. 4.13.B. RESULTATS D'APPRENTISSAGE POUR LA PARTIE REELLE DES PARAMETRES S_{21} ET S_{12} D'UN AMPLIFICATEUR DE PUISSANCE AVEC 91 ECHANTILLONS ET 15 NEURONES,	73
FIG. 4. 13.C. RESULTATS D'APPRENTISSAGE POUR LA PARTIE REELLE DES PARAMETRES S_{11} ET S_{22} D'UN AMPLIFICATEUR DE PUISSANCE AVEC 91 ECHANTILLONS ET 1000 ITERATIONS, EN FAISANT VARIER LE NOMBRE DE NEURONES.....	74
FIG. 4. 13.D. RESULTATS D'APPRENTISSAGE POUR LA PARTIE REELLE DES PARAMETRES S_{21} ET S_{12} D'UN AMPLIFICATEUR DE PUISSANCE AVEC 91 ECHANTILLONS ET 1000 ITERATIONS, EN FAISANT VARIER LE NOMBRE DE NEURONES.....	75

Liste d'abréviations

CPU : Central Processing Unit.

RNA: Réseau de Neurones Artificiels.

MLP: Multi Layer Perceptron.

MLP3: Multi Layer Perceptron à trois couches.

RBF: Radial Based Function.

KBNN: Knowledge Based Neural Network.

PKI: Prior Knowledge Input.

SOM: Self Organizing Method.

RNR: Réseau de Neurones Récurent.

HFSS: high frequency structural simulator.

ADS: Advanced Design System.

Introduction Générale

De nos jours, la demande en dispositifs de communication radiofréquences (RF) et micro-ondes est de plus en plus importante tant en qualité qu'en quantité. Cet engouement pour les technologies de l'information amène les concepteurs de ces dispositifs à innover et à proposer des systèmes de plus en plus performants, rapidement disponibles sur le marché et à un prix compétitif. Or une telle démarche implique une conception de dispositifs de plus en plus complexe, rigoureuse et en un minimum de temps. Malheureusement ces objectifs ne peuvent être atteints sans disposer de modèles rapidement simulables, fiables, pouvant être générés rapidement. Delà, l'étape de modélisation prend tout son sens. Or, la modélisation classique de processus RF/micro-ondes est souvent difficile voire impossible par exemple lorsque la technologie est nouvelle et non encore maîtrisée. En effet, il n'est pas possible d'utiliser des équations régissant le comportement du dispositif (lesquelles utiliser ?), ni le modèle de circuit équivalent (technologie nouvelle donc représentation électrique non encore connue) ...

L'intérêt pour un type de modélisation rapide, fiable, valable quelques soit la technologie utilisée, sans avoir à maîtriser le comportement du dispositif ni son circuit équivalent interne ... devient de plus en plus évident auprès des concepteurs.

Les réseaux de neurones remplissent ces conditions. Les réseaux de neurones possèdent en effet l'avantage incontesté de déterminer avec une grande précision les réponses de composants et circuits et la capacité de leur généralisation. Une fois son apprentissage terminé, le réseau de neurones peut être aisément implanté dans un simulateur de circuits, ce qui accélère de manière significative le processus de conception et d'optimisation.

Pour ces raisons, les réseaux de neurones ont été utilisés de manière extensive pour modéliser une variété de composants actifs et passifs micro-ondes [1]-[30].

L'objectif de cette thèse est de mettre en œuvre un outil de modélisation des dispositifs/circuits RF et micro-ondes basé sur les réseaux de neurones.

Ce mémoire de thèse est organisé en quatre chapitres. Le premier chapitre introduit les outils de modélisation, les techniques de modélisation ainsi que les méthodes de modélisation et de simulation (physique, numérique, analytique, ...) utilisés aux fréquences RF et micro-ondes. Il introduit aussi l'impact des réseaux de neurones sur la modélisation de dispositifs RF/micro-ondes.

Le deuxième chapitre est consacré à la théorie des réseaux de neurones, à expliquer leurs principes, les structures de base telles que le MLP et les structures plus avancées (KBNN, PKI et Space Mapping entre autres) ainsi que les différents algorithmes d'apprentissage. Le problème des minima locaux est introduit et quelques techniques pour y remédier y sont mentionnées.

Des techniques pour mieux utiliser les réseaux de neurones sont présentées au chapitre III, tel que la distribution des échantillons, l'intervalle de variations des variables d'entrées/sorties et la normalisation de données, ainsi le problème du choix de la structure du réseau neuronal et l'algorithme d'apprentissage adéquat.

Afin de pouvoir modéliser des dispositifs, un outil de développement basé sur les réseaux de neurones a été mis au point et présenté au quatrième chapitre. Il a aussi été testé à travers différents exemples.

Enfin, la conclusion présente le résumé des différents résultats obtenus dans cette thèse et les perspectives et développement futur de ce travail.

Outils de modélisation

1. Introduction

Les concepteurs de circuits micro-ondes sont souvent astreints à composer avec des modèles empiriques, approximatifs et/ou simplifiés à cause du coût de calcul associé à la simulation de modèles plus précis. Ces imperfections peuvent être au niveau de la modélisation, du fonctionnement des composants électroniques constituant le circuit ou au niveau de la modélisation des phénomènes physiques ou électriques mis en jeu dans un dispositif micro-ondes (couplage parasite, rayonnement, dispersion, pertes ...). Afin d'y pallier tout en augmentant les performances des outils de conception assistée par ordinateur (CAO), de très nombreuses recherches ont porté sur la génération de modèles fiables et rapides. Cependant, il faut considérer qu'il ne sera pas possible d'obtenir un modèle qui reproduit exactement le comportement d'un système réel. En effet, certains phénomènes électromagnétiques par exemple restent à ce jour non caractérisables ou tout au moins non maîtrisés.

2. Procédure de conception

L'ordre des diverses étapes dans une procédure de conception RF et micro-onde typique est résumé à la figure 1.1.

La procédure de conception commence par l'identification du problème. Cette phase est concernée par la détermination du besoin d'un produit. Un produit doit ainsi être identifié, des ressources allouées et des utilisateurs potentiels ciblés. L'étape suivante est d'élaborer les caractéristiques de conception du produit qui décrit les spécifications du besoin et les performances du produit.

Ceci est suivi par une étape de génération de concept, ou des décisions préliminaires de la conception sont faites. Plusieurs alternatives seront normalement considérées.

Les décisions faites à ce stade déterminent la configuration générale du produit et ont ainsi d'énormes implications pour le reste du procédé de conception. Pour chacune de ces étapes de conception, il y a habituellement un besoin de retourner aux étapes précédentes pour faire des retouches si besoin.

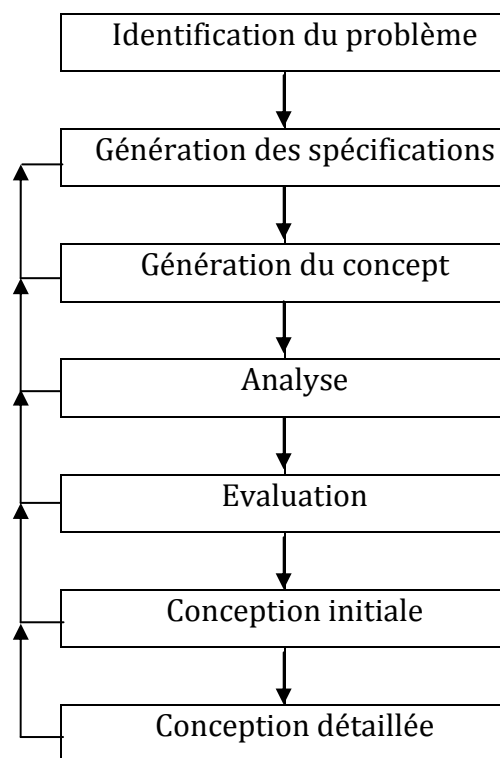


Fig. 1. 1. L'ordre des étapes d'un procédé de conception [31]

La performance du plan d'étude est testée pour sa compatibilité avec les caractéristiques de conception et sa réaction aux entrées externes. Ces étapes mènent à une première conception.

Les étapes à partir de la conception initiale jusqu'à la conception finale concernent la modélisation, l'analyse assistée par ordinateur, et souvent l'optimisation.

Les outils de CAO actuellement disponibles pour la conception RF et micro-onde s'adressent principalement à cette étape seulement. L'examen de la séquence des étapes de la figure 1.1 montre que l'analyse de la conception d'un composant, circuit ou antenne est nécessaire au niveau de deux étapes différentes du procédé de conception. Une fois le concept représentant la configuration atteint, il y a un besoin d'analyser et d'évaluer les performances potentielles de la conception expérimentale (si elle existe). À ce stade, les méthodes d'analyse approximatives (comme celles basées sur des modèles approximatifs) fournissent une approche suffisamment précise. La deuxième étape du procédé de conception où une analyse est nécessaire est dans la dernière étape qui convertit la conception initiale en une conception détaillée optimisée. La précision du procédé d'analyse doit être traduite directement sous la forme d'une comparaison entre les performances de conception et ses spécifications. A ce stade, nous avons besoin d'une analyse assistée par ordinateur basée sur des modèles plus précis pour les différents composants. [31]

3. Techniques de conception des circuits

Il y a deux philosophies de conception qui peuvent être appliquées pour les circuits RF et micro-ondes, antennes et autres systèmes. Ce sont :

- Procédure de conception conventionnelle.
- Approche de CAO.
 - Modèle physique
 - Modèle circuit équivalent
 - Modèle boîte noire

3.1. Procédure de conception conventionnelle

La procédure de conception conventionnelle était la méthode utilisée par les concepteurs avant que les méthodes de CAO et que les logiciels ne soient développés. Elle consiste à construire le modèle initial, puis faire les mesures et évaluer ses caractéristiques.

Les performances expérimentales sont comparées avec les spécifications désirées, et si les spécifications données ne sont pas appropriées, la conception sera modifiée. Des mécanismes de réglage incorporés à la conception sont utilisés pour faire des modifications. Des mesures sont refaites et les résultats sont comparés avec les spécifications désirées. La séquence de modifications, mesure, et de comparaison est effectuée itérativement jusqu'à ce que les spécifications désirées soient atteintes. La configuration finale du produit obtenu est alors envoyée pour fabriquer le prototype.

Cette procédure avait été utilisée pour la conception des circuits et antennes RF et microondes pendant une longue période. Cependant, il est devenu de plus en plus difficile d'utiliser cette méthode itérative expérimentale parce que les systèmes modernes deviennent plus complexes et que la miniaturisation des circuits a rendu la modification très difficile à faire, sans parler des coûts élevés liés à cette approche.

3.2. L'approche CAO

Dans la Conception Assistée par Ordinateur, les différents dispositifs sont modélisés à l'aide de leurs caractéristiques physiques ou empiriques. Les modèles physiques décrivent le dispositif en termes de sa structure physique et prévoient ses performances au moyen d'équations électromagnétiques (équations de Maxwell) et/ou d'équations de transport de charge. En principe, le comportement du dispositif peut être prévu *à priori*, sans besoin de fabriquer réellement le dispositif lui-même. Dans la pratique, quelques paramètres doivent être ajustés *à posteriori*, parce que tous les phénomènes réels ayant lieu à l'intérieur d'un dispositif ne sont pas tous connus, et aussi en raison des tolérances de fabrication. En outre, les équations physiques sont habituellement simplifiées pour gagner en temps de calcul et simplicité d'utilisation ; par conséquent, quelques paramètres empiriques doivent remplacer les termes manquants dans les équations. De toutes les façons, les modèles physiques exigent un long temps de calcul et des espaces mémoire prohibitifs.

Leur utilisation se base essentiellement sur la possibilité d'optimiser un dispositif avant sa fabrication, au moins préalablement, et dans une meilleure compréhension

du comportement de ce dispositif et des causes possibles de ses défauts de fonctionnement.

Ils sont également utiles pour l'optimisation quand leur coût de calcul reste raisonnable.

3.2.1. Modèles physiques

Les modèles physiques couvrent une large gamme de dispositifs actifs et diffèrent très largement pour la formulation des équations sur lesquels ils sont basés. Ainsi prenons comme exemple la modélisation des semi-conducteurs. En partant des phénomènes physiques, l'équation de mouvement pour les particules chargées (des électrons et/ou des trous) peut être l'équation quantique de Schrödinger ou l'équation du Boltzmann semi-classique, selon l'échelle des phénomènes microscopiques. Dans des dispositifs électroniques, le champ électrique apparaît toujours parmi les causes du mouvement des particules à l'intérieur d'un semi-conducteur; pour cette raison, l'équation de Poisson doit généralement être considérée puisqu'elle associe le champ électrique et les particules chargées à l'intérieur du semi-conducteur. Aucun dispositif réel n'a été jusqu'ici démontré où l'équation de Poisson doit être remplacée par les équations de Maxwell, bien que pleins de modèles électromagnétiques/transport aient été développés [32].

Les équations sont écrites en trois dimensions pour une description complète des effets des dispositifs réels (modèles 3D). Cependant, seulement une section du dispositif peut être considérée lorsque le dispositif est uniforme dans la direction transversale, pour économiser beaucoup d'efforts de calcul (modèles 2D). Dans certains cas, seule la direction principale de mouvement des particules chargées est considérée (par exemple de la source au drain, ou de l'émetteur au collecteur), ayant pour résultat un modèle unidimensionnel ; dans ce cas-ci, les résultats sont encore moins précis mais l'effort de calcul est considérablement réduit (modèles 1D).

Quelques fois, la deuxième direction est incluse sous forme d'équations supplémentaires, fournissant un modèle quasi-2D.

La description mathématique dépend fortement de la formulation du modèle. Parfois, le dispositif est décrit en termes d'équations de Poisson et de Schrödinger seulement.

Plus souvent, les effets quantiques ont lieu seulement dans une partie limitée du dispositif, par exemple, dans une hétérojonction : le reste du dispositif est alors décrit par des équations semi-classiques. Dans d'autres cas, seule une équation semi-classique de mouvement et l'équation de Poisson sont requises pour une précision suffisante. L'équation du Boltzmann semi-classique est souvent augmentée en moments, par l'intégration appropriée dans l'espace des moments, et alors seulement les premiers sont maintenus. Le premier moment est l'équation de conservation des particules ou l'équation de continuité ; le second est l'équation de conservation des moments ou équation de densité de courant ; le troisième est l'équation de conservation d'énergie, et ainsi de suite.

Une fois que le type de formulation est défini, les forces principales pilotant le mouvement des particules chargées doivent être décrites par des équations de telle manière que seuls les effets physiques significatifs soient maintenus et que la solution demeure raisonnablement simple et précise. Il n'est souvent pas facile de décrire les forces avec suffisamment de détail et de précision, donner leur formulation mathématiquement compliquée et l'incertitude des paramètres liés au procédé de fabrication. Un compromis doit être recherché, parfois au moyen de paramètres semi-empiriques dérivés des mesures ou de l'évidence pratique.

Toutes les équations sont différentielles par rapport au temps et à l'espace. Dans le cas particulier des modèles équilibrés, la dépendance au temps est enlevée, et dans les modèles plus simples, la dépendance de l'espace peut être exprimée de façon à être non différentielle; dans tous les cas, les équations exigent des conditions aux limites. Pour l'équation de Poisson, dans la plupart des cas, les conditions aux limites sont données par la tension externe appliquée.

Pour l'équation de transport, elles doivent être un état physiquement significatif: par exemple, condition de neutralité ou d'équilibre très loin de la jonction ou du

canal. La solution des équations différentielles exige alors habituellement une solution numérique et le modèle est dit numérique. La méthode de solution peut être stochastique ou déterministe, c.-à-d., comportant ou pas les propriétés statistiques du comportement microscopique des particules dans un semi-conducteur.

Quand la formulation mathématique est assez simple pour tenir compte une solution analytique, explicite en termes de tensions externes et de courants, le modèle est dit analytique. Seulement les modèles très simplifiés appartiennent à cette catégorie, et leurs utilités se basent essentiellement sur leur simplicité et clarté de description; leur précision est souvent au-dessous des limites acceptables pour la simulation pratique de circuit

Les modèles physiques sont normalement employés pour prévoir le comportement de la partie intrinsèque d'un dispositif actif ; pour une évaluation réaliste de ses performances, il est souvent nécessaire d'ajouter des éléments parasites comme les résistances de contact, capacités de boîtier ou inductances de ligne; celles-ci peuvent à peine être théoriquement prévues, et sont habituellement évaluées au moyen d'expressions empiriques ou semi-empiriques. L'évaluation des effets dus au layout du dispositif est également difficile et a été récemment adressée en couplant une analyse électromagnétique des parties connectés du dispositif à l'étude physique de la cloison intrinsèque. Cette approche va probablement gagner en importance à mesure que la fréquence de fonctionnement augmente dans la région d'onde millimétrique et au-delà. [41]

3.2.1.1. Equations de Base

Toujours pour illustrer notre propos, prenons les équations de base à une dimension pour plus de simplicité ; Ainsi, par exemple, l'équation dépendant du temps de Schrödinger est [33]

$$i \frac{d\varphi}{dt} = -\frac{\hbar^2}{2m} \frac{d^2\varphi}{dx^2} - q(V - V_0) \varphi \quad (1.1)$$

où $\Psi(x, t)$ est la fonction de probabilité ou fonction d'onde, h est la constante de Planck et V le potentiel électrique. La dérivée de la fonction d'onde par rapport au temps correspond à l'énergie de la particule. La dérivée de la fonction d'onde par rapport à l'espace correspond au moment classique. L'équation montre que l'énergie de la particule égale la somme du carré du moment (l'énergie cinétique) et de l'énergie potentielle.

Si le potentiel est invariable dans le temps, l'équation de Schrödinger à temps-invariable est obtenue comme étant [33]

$$\omega = -\left(\frac{h}{2\pi}\right)^2 \frac{d^2\varphi}{dx^2} - q(V - V_0) \quad (1.2)$$

où ω est l'énergie à temps-invariable de la particule et sa solution est un état quantique invariable dans le temps, ou de niveau, pour l'électron. Une solution dépendant du temps de l'équation de Schrödinger (1.2) est obtenue comme superposition linéaire des solutions ou des états temps-invariables.

3.2.1.2. Modèle numérique

Quand les équations ainsi que leurs conditions aux limites ne peuvent pas être résolues explicitement, et qu'aucune solution analytique n'est disponible, un plan de solution numérique doit être mis en application. Les plans de solution appartiennent à deux catégories principales : déterministe et stochastique. Dans la première approche, les particules chargées dans le semi-conducteur sont rapprochées par une seule particule de gaz, où le mouvement de chaque particule est piloté par les forces déterministes qui sont les mêmes pour chaque particule dans le gaz; toutes les particules dans le gaz sont identiques. Dans certains cas, les particules chargées sont rapprochées par deux gaz, un pour les particules de grande énergie et un pour les particules à énergie réduite. Les équations sont résolues par des calculs exacts (dans le sens numérique), et les quantités macroscopiques d'intérêt sont calculées pour le gaz entier. Dans la deuxième approche, les forces et les événements qui déterminent le mouvement d'une particule maintiennent leur nature statistique de sorte que la trajectoire de chaque particule soit différente de celle de tout autre dans le gaz.

Dans cette deuxième approche, à la base des modèles de Monte Carlo, des quantités macroscopiques comme le courant ou la température de gaz sont calculées comme moyenne du comportement de la population du gaz ; un grand nombre de particules doivent être simulées afin d'obtenir statistiquement des résultats significatifs.

Les équations sont discrétisées dans le domaine espace et dans le domaine temps, si la dépendance du temps est incluse, de sorte que le système des équations aux dérivées partielles soit remplacé par un système d'équation aux différences finies [34]. Le système est habituellement résolu dans le domaine du temps, avec des systèmes tels que l'Équilibrage Harmonique [35], et la Simulation d'Enveloppe [36].

3.2.1.3. Modèle analytique

Les modèles analytiques sont très simples et rapides, mais leur exactitude est limitée ; comme déjà mentionné, leur fonctionnalité la plus utile est la possibilité de prévoir et optimiser le rendement d'un circuit, une fois connues les tolérances de fabrication des paramètres physiques et leurs propriétés. En fait, un modèle analytique peut également être traité comme un modèle d'équivalent-circuit, où les paramètres ne sont pas directement les valeurs des éléments, mais plutôt les paramètres physiques dans la formulation analytique, qui doivent pour cette raison être destinés comme paramètres équivalents [37], [38]. Les données ajustées aux données mesurées s'avèrent presque aussi bonnes que dans le cas du circuit équivalent, mais elles sont habituellement plus stables, et les valeurs extraites affichent une meilleure corrélation au procédé technologique. Néanmoins, l'exactitude n'est parfois pas suffisante pour une conception de circuit précise

3.2.2. Modèle de circuit équivalent

Les modèles empiriques sont une solution de choix pour une conception pratique et relativement précise des circuits non linéaires. L'effort exigé par de longues mesures et procédures parfois ennuyeuses d'extraction éponge en termes de flexibilité et précision des applications de CAO.

Les modèles de circuit équivalent sont les modèles empiriques les plus utilisés pour les applications de CAO. Le dispositif actif est modélisé dans le cadre du programme de simulation à l'aide d'un circuit électrique, dont les éléments sont habituellement localisés, avec des valeurs indépendantes de la fréquence mais dépendantes de la tension. Les circuits équivalents d'éléments distribués ne se sont pas encore imposés pour la conception non linéaire pour deux raisons principales: d'abord, les éléments non linéaires distribués ne sont pas facilement traités par les algorithmes de CAO non linéaires; en second lieu, les dispositifs actifs sont habituellement fabriqués de telle manière que les effets distribués dans le dispositif intrinsèque soient négligeables dans la bande de fréquence utilisée. Naturellement, les éléments d'accès distribués ne sont pas un problème, au moins pour des méthodes d'analyse dans le domaine fréquentiel ou mixte (temps-fréquence); néanmoins, ils sont habituellement remplacés par des éléments localisés sans perte significative de précision, étant donnée les petites dimensions des dispositifs. Les éléments non linéaires dépendants de la fréquence posent des problèmes aux algorithmes d'analyse non linéaires, à moins qu'une approche purement dans le domaine fréquentiel soit adoptée ; si c'est le cas, alors ces modèles s'avèrent être tout à fait généraux et pratiques.

Tous les modèles de circuit équivalent tels que ceux modélisant les transistors sont faits de deux réseaux complémentaires :

- **Extrinsèque et linéaire** correspondant aux éléments parasites comme les résistances de contact, capacités de boîtier ou inductances dues aux lignes d'accès ou fils de connections,
- **Intrinsèques et linéaires ou non linéaires** (selon le signal appliqué et les conditions de fonctionnement) correspondant au dispositif intérieur qui set le cœur du dispositif.

La topologie des deux réseaux devrait être *à priori* connue avant l'extraction, bien qu'elle puisse être ajustée pendant l'extraction si les mesures la suggèrent. C'est donc un grand handicap si le système n'est pas bien défini.

3.2.3. Modèle Boîte Noire

La modélisation *boîte noire*, aussi appelée modélisation comportementale, est une approche mathématique qui relie les entrées et sorties du système à modéliser en faisant abstraction des connaissances physiques sur la dynamique interne du système. [39]

Pour obtenir la représentation boîte noire d'un système donné (Figure 1.2), il faut déterminer la sortie $y(t)$ en fonction du signal d'entrée $x(t)$ sans le passage par des représentations topologiques intermédiaires.



Fig. 1.2. Représentation boîte noire d'un système. [39]

Cette caractérisation du système fait ensuite l'objet d'une introduction dans les simulateurs dans le but d'analyser le fonctionnement du système soumis à des excitations plus ou moins arbitraires.

Les modèles comportementaux sont basés sur l'hypothèse qu'aucune information sur la structure interne du dispositif n'est connue. Le comportement du dispositif ou du sous-système est mesuré, et alors il est approché au moyen de fonctions d'ajustement appropriées, habituellement de la classe des réseaux neuronaux artificiels (RNA). Cette supposition limite sérieusement les propriétés prédictives du modèle, qui sont valides seulement dans un voisinage des conditions de mesure. Cependant, dans certains cas ce voisinage inclut les situations très intéressantes (utilisées) en lesquelles la grande épargne en termes de temps de simulation est obtenue [41].

De plus, lorsque la mise au point d'un modèle est très longue et nécessite beaucoup de moyens informatiques et humains ou lorsque le comportement du dispositif est très difficile voire impossible à évaluer à travers des modèles physiques ou électriques (circuits équivalents) comme cela peut être le cas pour des dispositifs issus de nouvelles technologies, ce type de modélisation reste le seul utilisable. Nous allons nous consacrer à ce type de modèles à travers une classe importante de modèles comportementaux que sont les modèles neuronaux.

4. Utilisation des réseaux de neurones dans la conception des circuits Micro-ondes

Récemment, l'approche basée sur des réseaux neuronaux a été introduite pour la modélisation de nombreux composants passifs et actifs micro-ondes [12] - [42] et pour la conception de circuits micro-ondes [42], [17], [43], [44]. Un réseau neuronal peut être développé par l'apprentissage des données micro-ondes, un processus appelé donc apprentissage. Une fois déterminé, le réseau neuronal peut alors être employé pendant la conception micro-onde pour apporter des réponses instantanées à la tâche qu'il a apprise [42]. Les travaux récents ont démontré la capacité des réseaux neuronaux d'apprendre et de modéliser une série de composants micro-onde, tels que les circuits planaires [12], [45], [46], les Vias [45], les inductances [5], les transistors [12], [47], les guide d'ondes coplanaires (CPW) [17], les interconnexions [48], etc. Des réseaux neuronaux ont été également utilisés dans la simulation et l'optimisation de circuits [42], [47], [49], la conception de circuits micro-ruban [50], la modélisation de circuits intégrés (IC) [51], la modélisation de processus [52], la synthèse de circuits [43], l'utilisation de l'abaque de Smith [44], l'adaptation d'impédances [52], etc. Les modèles neuronaux sont beaucoup plus rapides que les modèles électromagnétiques/physiques détaillés, plus précis que les modèles polynômiaux ou empiriques (puisque'ils s'appuient sur des données le plus souvent issues de mesures), et sont plus faciles à développer quand un nouveau dispositif/technologie est introduit [49]. Les coûts pour développer les modèles neuronaux sont principalement la collecte d'informations et l'apprentissage du réseau neuronal [12], [42].

5. Conclusion

Dans ce premier chapitre, nous avons introduit les différentes phases de conception des circuits RF et micro-ondes, en commençant par l'identification du problème jusqu'à la conception détaillée.

Nous avons discuté des différentes techniques de modélisation des circuits RF et micro-ondes, à savoir l'ancienne méthode conventionnelle et les méthodes de CAO. Puis, nous avons discuté des différents modèles de circuits.

Pour le modèle *boite noire*, nous avons introduit les réseaux de neurones et leur utilité dans la modélisation des circuits RF et micro-ondes.

Dans le chapitre suivant, nous allons entamer la théorie des réseaux de neurones, les différentes structures ainsi que les méthodes d'apprentissage.

Théorie sur les réseaux de neurones

1. Introduction

Les calculateurs numériques modernes surpassent les humains dans les tâches basées sur des opérations arithmétiques précises et rapides. Cependant, les humains sont plus aptes à résoudre des problèmes perceptuels complexes tels qu'identifier des images, même en présence de perturbations. En outre, les humains peuvent exécuter des opérations complexes avec précision, même en présence de perturbations et peuvent généraliser à partir d'expériences antérieures [53]. Par conséquent c'est le désir de concevoir des systèmes qui peuvent un tant soit peu atteindre les performances des systèmes biologiques qui a mené à l'étude du réseau de neurones artificiels.

1.1. Définitions

Les réseaux neuronaux artificiels sont des modèles mathématiques et des algorithmes qui émulent certains aspects du traitement des données et des méthodes de rassemblement de données de système nerveux humain [54]. L'utilisation des réseaux neuronaux artificiels pour exécuter des calculs est inspirée du système neurobiologique dans lequel les messages sont transférés entre plusieurs neurones en utilisant les neurotransmetteurs spécifiques. Les réseaux neuronaux artificiels utilisent des neurones et des poids afin d'exécuter un traitement parallèle pour déterminer le rapport entre les données d'entrées et les valeurs de sorties désirées [55].

Un réseau neuronal artificiel peut exécuter des opérations complexes sur des données non linéairement dépendantes en impliquant des relations entre paramètres d'entrée et paramètres de sortie. Il peut, en principe, généraliser d'une quantité limitée de données d'apprentissage à la représentation du modèle complet.

1.2. Évolution des réseaux de neurones artificiels

Les simulations de réseau de neurones semblent être un développement récent. Cependant, ce champ a été établi avant l'arrivée des ordinateurs. Il a commencé par la modélisation des fonctions d'un cerveau humain par McCulloch et de Pitts en 1943, qui ont édité un document décrivant le calcul logique des réseaux neurologiques [57]. Le développement principal des réseaux neurologiques est venu en 1949 avec la publication du livre de Hebb « The Organization of Behavior » [58], dans lequel un rapport explicite d'une règle d'étude physiologique pour la modification synaptique a été présenté pour la première fois.

Le livre de Hebb a été une source d'inspiration pour le développement des méthodes d'apprentissage et du système adaptatif. En 1954, Minsky a écrit une thèse sur « le réseau de neurones » dans son étude de doctorat [59]. En 1961, il a écrit un article sur l'intelligence artificielle intitulé « Steps Towards Artificial Intelligence » [60]. En 1954, Gabor a proposé l'idée du filtre adaptatif non linéaire [61]. Il a essayé de construire une machine, en laquelle l'apprentissage a été accompli en introduisant des échantillons d'un procédé stochastique dans la machine, ainsi que la fonction cible qui était prévisible. En 1958, Rosenbatt a proposé une nouvelle approche au problème de reconnaissance de forme [62]. Les travaux de Rosenbatt sont connus sous l'appellation du théorème de convergence des perceptrons. En 1960, Widrow et Hoff ont introduit l'algorithme des moindres carrés (LMS) [63]. Un des premiers réseaux de neurones à couches entraînable avec multiples éléments adaptatifs était la structure Madaline (multiple-adaline) proposée par Widrow (1962) [64]. Le problème majeur des premières recherches sur le réseau de neurones était en partie technologique et en partie financier. En l'absence d'ordinateur personnel, la conception et la formation de réseau de neurones ont dû être faites avec des circuits analogiques. Par exemple, Gabor et son équipe de recherche ont travaillé plus de six années pour construire le filtre avec les unités analogiques. Cependant, dans les années 80, avec le développement des micros ordinateurs, il y a eu un regain d'intérêt envers les réseaux neuronaux.

Dans les années 80, des contributions importantes à la théorie et la conception des réseaux neuronaux ont été faites sur plusieurs fronts. Grossberg (1980) a établi un nouveau principe d'organisation autonome qui combine le filtrage adaptatif et le perfectionnement de contraste dans la mémoire à court terme avec couplage descendant et stabilisation d'indicatif d'apprentissage. Ainsi, si la configuration du modèle d'entrée et le contrôle par retour de l'information instruite s'assortissent, un état dynamique appelé résonance adaptative a lieu. Ce phénomène constitue la base de la nouvelle classe de réseaux neuronaux connue sous le nom de la théorie de résonance adaptative [64]. En 1982, Hopfield a développé une nouvelle classe de réseaux neuronaux avec le contrôle par retour de l'information, qui est bien connu comme réseaux de Hopfield [65].

Un autre développement important en 1982 a été fait par Kohonen. Il a développé une carte à organisation autonome « Self-Organizing Map (SOM) » en utilisant un ou deux structures de réseau [66]. En 1983, Kirkpatrick, Gallat et Vecchi ont décrit une nouvelle marche à suivre appelée « Simulated Annealing », pour résoudre le problème d'optimisation combinatoire [67]. En 1985, Hinton et Sejnowski ont développé un algorithme d'apprentissage appelé apprentissage de Boltzmann qui utilise la distribution de Boltzmann [68]. En 1986, Rumelhart, Hinton et Williams ont développé un algorithme d'apprentissage appelé algorithme de rétro-propagation ou « Back-Propagation » [69]. Leurs publications « Parallel Distributed Processing » et « Explorations in the Microstructures of Cognition » ont eu une influence importante dans l'utilisation de la rétro-propagation pour l'apprentissage du réseau neuronal, qui a émergé comme algorithme d'apprentissage le plus populaire pour l'apprentissage du perceptron multicouches [56]. Plus tard, l'algorithme de rétro-propagation a été modifié par beaucoup de chercheurs pour augmenter la vitesse d'apprentissage [5], [70].

Broomhead et Lowe, en 1988, ont décrit une marche à suivre pour concevoir des réseaux à couches feedforward d'alimentation en utilisant des fonctions de base radiales qui fournit une alternative aux perceptrons multicouches [71].

Aujourd'hui, les réseaux de neurones se sont établis comme sujet interdisciplinaire avec des racines profondes en neurologie, psychologie, mathématiques et en sciences et techniques entre autres. Aujourd'hui, des réseaux de neurones sont employés avec succès pour résoudre beaucoup de problèmes complexes du monde réel.

2. Structure du réseau de neurones

2.1. Modèle de neurone

Le neurone est la cellule fondamentale d'un réseau de neurones artificiel. Par analogie avec le neurone biologique, le neurone artificiel doit être apte à accomplir les tâches suivantes : collecter, traiter les données qui viennent des neurones émetteurs et transmettre les messages aux autres neurones, puis vers ceux de sortie.

La structure d'un neurone artificiel est représentée sur la figure 2.1. Le neurone calcule la somme de ses entrées puis cette valeur passe à travers la fonction d'activation pour produire sa sortie.

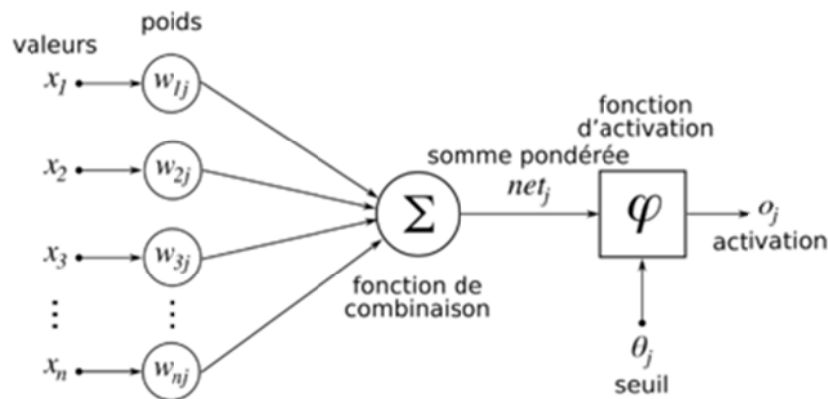


Fig. 2. 1. Modèle de neurone artificiel [56].

2.2. Différents types de réseaux de neurones

Il y a une large variété de réseaux de neurones et d'architectures. Les types de réseaux de neurones s'étendent des réseaux booléens simples (perceptions) aux réseaux SOM complexes (réseaux de Kohonen). Il y a également beaucoup d'autres

types de réseaux comme les réseaux de Hopfield, d'impulsion, de fonctions radiales (RBF), de Boltzmann, etc.

Bien que l'architecture des réseaux de neurones ne puisse pas être liée par des règles bien définies, il y a quelques architectures de réseaux dites standards.

2.2.1. Réseau feedforward à une seule couche

C'est le type le plus simple de réseau qui se compose d'une couche de neurones d'entrée source qui se projette directement sur les neurones de sortie. Il n'y a aucune couche cachée dans cette architecture. Il est strictement de type feedforward, car il n'y a aucun contrôle de l'information par retour de la couche de sortie. La figure 2.2 affiche un réseau feedforward à une seule couche.

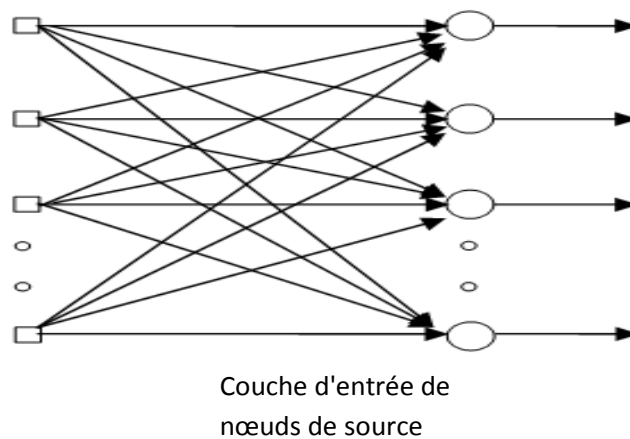


Fig. 2. 2. Réseau feedforward à une seule couche [72]

2.2.2. Réseau feedforward multicouche

Un réseau feedforward multicouche se distingue par la présence d'une ou plusieurs couches cachées. Les neurones cachés dans des couches cachées interviennent entre les entrées externes et les sorties du réseau. L'ajout des couches cachées dans le réseau augmente sa capacité d'extraire des statistiques évoluées.

Cette caractéristique augmente sa capacité de traitement d'un niveau important de non-linéarité et des situations complexes. Le réseau feedforward multicouche peut être entièrement connecté ou partiellement connecté.

2.2.2.1. Réseau feedforward multicouche entièrement connecté

a. Perceptron multicouche

Une classe importante des réseaux de neurones feedforward, qui est un type fondamental de réseaux de neurones capables de rapprocher des fonctions continues et intégrables génériques, est le perceptron multicouche (MLP). C'est le réseau le plus populaire et le plus utilisé. Typiquement, le réseau de neurones MLP se compose d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie, suivant les indications de la figure 2.3.

Supposons que le nombre total de couches soit L . La couche d'entrée est la couche 1, la couche de sortie est la couche L et les couches cachées sont notées $2, 3, \dots, L-1$. Soit N_l le nombre de neurones dans la couche l . Soit w_{ij}^l le poids du lien entre le j^{ieme} neurone de la couche cachée $l-1$ et le i^{ieme} neurone de la couche cachée l , et soit θ_i^l le biais du i^{ieme} neurone de la couche cachée l . Soit x_i le paramètre d'entrée du MLP. Soit \hat{y}_i^l la sortie du i^{ieme} neurone de la couche cachée l , qui peut être calculé selon les formules standards de MLP comme suit [73]:

$$\hat{y}_i^l = \sigma\left(\sum_{j=1}^{N_{l-1}} w_{ij}^l \cdot \hat{y}_j^{l-1} + \theta_i^l\right), \quad i = 1, 2, \dots, N_l, \quad l = 1, 2, \dots, N_h \quad (2.1)$$

$$\hat{y}_i^1 = x_i, \quad i=1,2,\dots,N_1 \quad (2.2)$$

où $\sigma(\cdot)$ représente la fonction d'activation des neurones cachés et N_1 le nombre d'entrées du modèle. Supposant que v_{ki} représente le poids du lien entre le i^{ieme} neurone de la couche cachée N_l et le k^{ieme} neurone de la couche de sortie, et β_k le biais du k^{ieme} neurone de sortie ; Les sorties du MLP peuvent être calculés par [73]

$$\hat{y}_k = \sum_{i=1}^{N_{N_h}} v_{ki} \cdot \hat{y}_i^{N_h} + \beta_k, \quad k=1,2,\dots, N_y \quad (2.3)$$

Pour l'approximation des fonctions, la fonction d'activation des neurones de sortie peut être représentée par la fonction linéaire (2.3).

La fonction d'activation la plus utilisée pour les neurones cachés est la fonction sigmoïde logistique donnée par [73]

$$\sigma(\gamma) = \frac{1}{(1 + e^{-\gamma})} \quad (2.4)$$

qui présente la propriété suivante

$$\sigma(\gamma) \rightarrow \begin{cases} 1 & \text{as } \gamma \rightarrow +\infty \\ 0 & \text{as } \gamma \rightarrow -\infty \end{cases} . \quad (2.5)$$

D'autres candidats possibles pour $\sigma(\cdot)$ sont la fonction d'arc tangente

$$\sigma(\gamma) = \left(\frac{2}{\pi} \right) \arctan(\gamma) \quad (2.6)$$

et la fonction tangente hyperbolique

$$\sigma(\gamma) = \frac{(e^{\gamma} - e^{-\gamma})}{(e^{\gamma} + e^{-\gamma})} \quad (2.7)$$

Toutes ces fonctions sont bornées, continues, monotones et continuellement différentiables.

Le théorème d'approximation universel déclare qu'il existe toujours un MLP à trois couches qui peut approcher n'importe quel modèle désiré pour n'importe quelle fonction non linéaire, continue, multidimensionnelle arbitraire [74], [75]. Cependant, le théorème d'approximation universel n'indique pas le nombre de neurones requis. Par conséquent, l'échec de développer un modèle neuronal précis peut être attribuée à un apprentissage insuffisant, un nombre insuffisant de neurones cachés, ou à la présence d'une relation stochastique plutôt que déterministe entre les entrées et les sorties [76]. Dans la pratique, le nombre de neurones cachés dépend du degré de non-linéarités et des dimensions du problème initial. Mais trop de neurones cachés mèneront à un sur-apprentissage (*over-Learning*).

Les réseaux de neurones avec une ou deux couches cachées (MLP à trois ou quatre couches) sont plus fréquemment utilisés et sont habituellement appropriés aux applications RF/Micro-ondes [73]. Les performances du réseau de neurones peuvent être évaluées en termes de capacité de généralisation et capacité de mappage. Dans [77], Tamura a expliqué que les MLP à 3 couches sont préférés dans l'approximation de fonctions où la capacité de généralisation est un souci important.

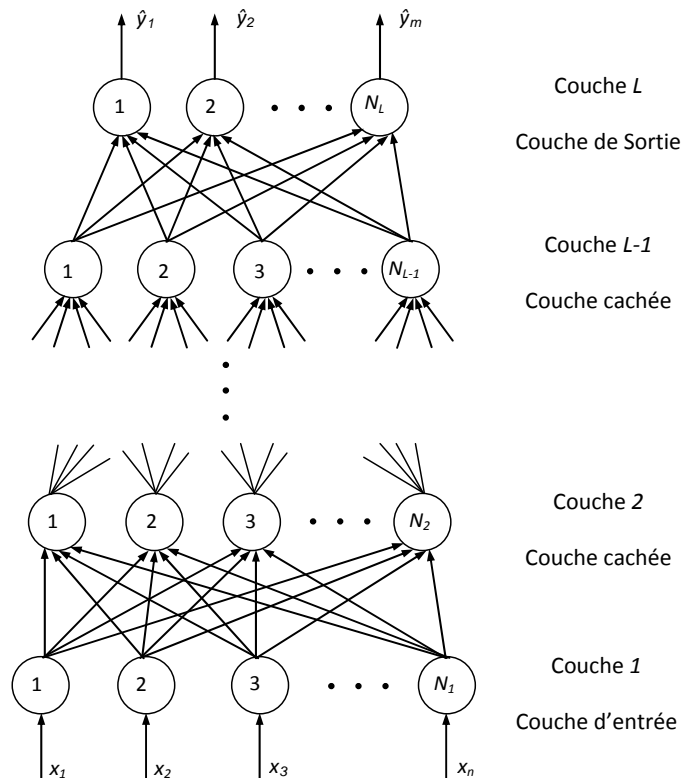


Fig. 2. 3. Illustration de la structure perceptrons multicouche feedforward (MLP) [31].

Intuitivement, les MLP à 4 couches donnent plus de performances dans les problèmes non linéaires dans lesquels les composantes comportementales localisés existent à plusieurs reprises dans différentes régions de l'espace du problème.

b. Fonctions Radiales

Les fonctions radiales ont été introduites par [78] dans le cadre de l'interpolation, c'est-à-dire de la recherche de fonctions passant exactement par un nombre fini de points (dits points de collocation).

Dans ce contexte, la fonction recherchée est une combinaison linéaire de fonctions de base, en nombre égal au nombre de points de collocation ; une fonction de base $\Phi_n(x)$, relative au point de collocation x_n , est dite radiale si elle ne dépend que de la distance du point courant x au point de collocation x_n . On peut utiliser diverses fonctions radiales, notamment des fonctions localisées (qui tendent vers zéro dans toutes les directions de l'espace des variables) telles que des gaussiennes centrées aux points de collocation. Bien entendu, la recherche d'une fonction passant exactement par les points n'a de sens que si ces points ne sont pas entachés de bruit.

Ces réseaux ont été utilisés comme outils de modélisation sous forme de "boîte noire" dans le domaine de l'automatique. On les trouve à base de modèles entrées-sorties [79] et aussi de modèles d'état [80]. Certaines spécificités de ces réseaux permettent de les utiliser pour la synthèse de lois de commande adaptatives stables [81]. Le fait que ces réseaux permettent de garantir la stabilité des correcteurs qu'ils réalisent les rend plus intéressants que les réseaux de neurones pour la résolution des problèmes de commande non linéaire. En revanche, cette propriété se fait au détriment de la parcimonie du réseau.

2.2.2.2. Réseau feedforward multicouche partiellement connecté :

a. Structure "Knowledge-Based Neural Network" (KBNN) [31]

Un grand nombre de données d'apprentissage est habituellement nécessaire pour assurer l'exactitude du modèle. Cependant, produire un grand nombre de données d'apprentissage pourrait être très cher du point de vue temps de simulation pour des problèmes d'hyperfréquences. Par exemple, la simulation EM pourrait être très chère pour générer beaucoup de points dans l'espace de paramètres du modèle. De plus, les problèmes RF et micro-ondes sont souvent modélisés par des relations empiriques à défaut d'expressions analytiques.

Or ces relations sont le plus souvent approximatives et ne tiennent pas compte de tous les phénomènes électromagnétiques (EM) présents dans les dispositifs à modéliser.

La connaissance micro-onde existante peut fournir des informations complémentaires du problème initial qui ne peut être adéquatement représenté par les caractéristiques limitées d'apprentissage.

Le réseau de neurones KBNN (Knowledge-Based Neural Network), peut aider à établir le lien entre le modèle empirique (modèle brut ou approximatif) et les solutions EM (solution plus précise) en se servant du modèle empirique comme information complémentaire entre entrées et sorties, ce qui ne peut qu'accélérer son apprentissage en exigeant moins de données

La structure des réseaux de neurones basés sur la connaissance [12] est illustrée à la figure 2.4.

Il y a six couches qui ne sont pas entièrement connectés entre eux, à savoir la couche d'entrée X , la couche de connaissance Z , la couche limite B , la couche de région R , la couche de région normalisé R et la couche de sortie \hat{Y} [31]. La connaissance préalable (relation empirique) est incluse dans le réseau de neurones (dans la couche de connaissance).

Comparé aux structures pures du réseau de neurones, la connaissance préalable fournit au réseau neuronal plus d'informations sur le problème micro-onde initial, sans compter l'information incluse dans les données d'apprentissage. En conséquence, les modèles KBNN ont une meilleure fiabilité quand les données d'apprentissage sont limitées ou quand le modèle est utilisé au-delà de l'intervalle d'apprentissage.

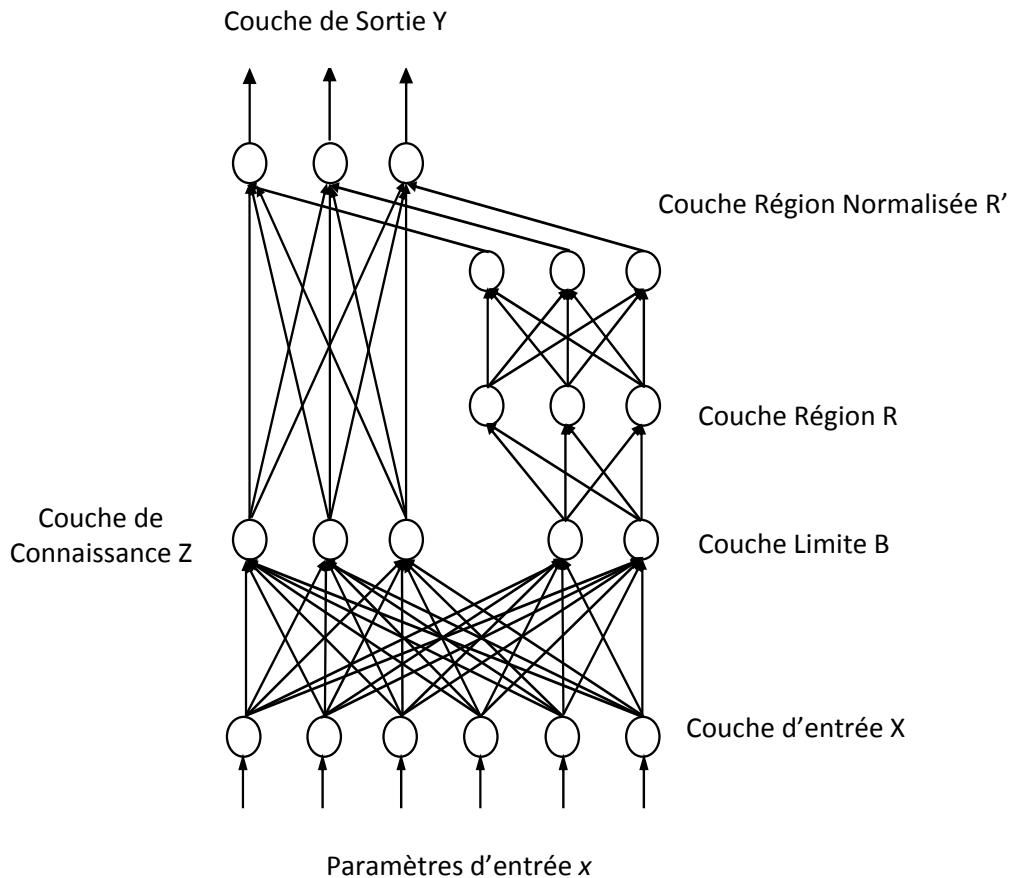


Fig. 2. 4. Illustration de la structure KBNN. [31]

b. Structure "Prior Knowledge Input" (PKI)

Les modèles empiriques existants ne donnent pas la précision demandée sur l'intervalle désiré [31]. Dans les structures "Prior Knowledge Input", les sorties du modèle empirique sont utilisés comme entrées au modèle du réseau de neurone, en plus des entrées du problème original. Dans ce cas, le mappage d'entrée-sortie à apprendre par le réseau de neurones est celui entre les sorties du modèle approximatif existant et le problème original (très utile dans les cas de généralisation des problèmes). Pour le cas où les sorties cibles sont les mêmes que les sorties du modèle approximatif, le problème d'apprentissage est réduit à un mappage un-à-un. [31].

c. Le réseau de neurone à mappage d'espace (SOM) [31]

Ce modèle combine l'efficacité de calcul des modèles bruts avec l'exactitude des bons modèles [31]. Les modèles bruts sont des fonctions empiriques approximatives ou des modèles de circuits équivalents, qui sont typiquement très efficaces du point de vue temps de calcul, mais peu précis en haute fréquence ou si le problème exige une solution plus élaborée (plus précise et/ou plus détaillée).

Ainsi, de tels modèles sont souvent valides seulement dans un intervalle limité de l'espace d'entrée (par exemple pour une certaine gamme de fréquences ou pour un intervalle restreint de valeurs des variables physiques ou électriques d'entrée) au-delà duquel les prévisions des modèles deviennent imprécises. D'autre part, des modèles détaillés ou fins peuvent être fournis par un simulateur électromagnétique ou par des mesures, les rendant ainsi très précis, mais leur génération peut être couteuse au regard du temps de calcul ou de l'équipement expérimental requis. La technique SM établit un lien mathématique entre le modèle brut et le bon modèle et dirige la partie des calculs CPU intensive vers le modèle brut, en préservant l'exactitude offerte par le modèle fin [31].

2.2.3. Réseaux récurrents

Appelés aussi réseaux bouclés ou dynamiques, ce sont des réseaux dans lesquels il y a un retour en arrière de l'information, ainsi des connexions apparaissent entre la sortie du réseau et les neurones qui se trouvent dans les couches amonts (Figure 2.5).

Les réseaux de neurones artificiels récurrents (RNR), qui possèdent une mémoire interne grâce aux cycles dans leur graphe d'interconnexion, ont des capacités d'approximation universelle pour les problèmes temporels [38] comparables à celles des réseaux de type Feedforward pour les problèmes statiques. Il convient d'insister sur le fait que le temps ne joue aucun rôle fonctionnel dans un réseau de neurones multicouche de type Feedforward : si les entrées sont constantes, les sorties le sont également.

Pour cette raison, les réseaux multicouches sont souvent appelés « réseaux statiques », par opposition aux réseaux bouclés ou « dynamiques » [82]. Les réseaux

récurrents se sont illustrés avec succès dans le traitement des signaux temporel, la commande adaptative, la prédiction des suites chronologiques, la reconnaissance de la parole, l'identification de système [83].

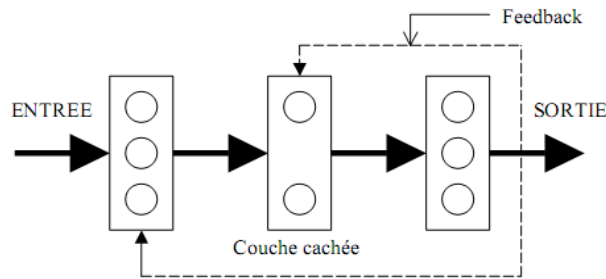


Fig. 2. 5. Réseau récurrent avec neurones cachées [84].

2.2.4. Structures Lattice

Une structure Lattice est différente des autres types d'architecture de réseaux de neurones par ses arrangements des neurones et de leurs connexions. Elle se compose d'un réseau de neurones unidimensionnel, bidimensionnel ou multidimensionnel avec un ensemble correspondant de nœuds de source qui fournissent les signaux d'entrée au réseau. La figure 2.6 dépeint un réseau bidimensionnel de 3 neurones by-3 alimentés d'une couche de puissance d'entrée de trois nœuds de source. Il est assimilé au réseau de réaction avec les neurones de sortie disposés dans les lignes et les colonnes. [85]

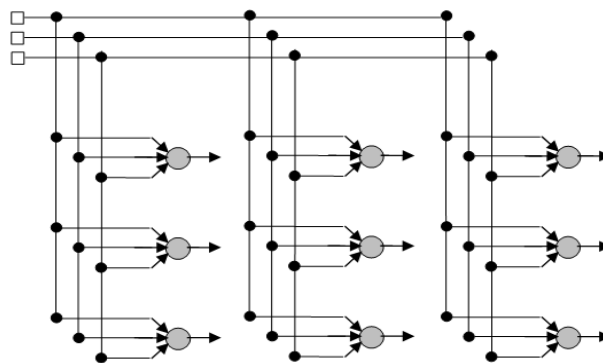


Fig. 2.6. Structure Lattice bidimensionnel de 3 X 3 neurones [85].

3. Apprentissage des réseaux de neurones

Une fois l'architecture d'un réseau de neurones choisie (en fonction du problème à modéliser), il est nécessaire d'effectuer un apprentissage.

Cette étape permet de déterminer les valeurs des poids permettant à la sortie du réseau de neurones d'être aussi proche que possible de l'objectif fixé (données de sorties fournies au réseau lors de son apprentissage). Dans le cas d'un problème de régression, il s'agit d'approcher une fonction continue alors que dans le cas d'un problème de classification supervisée, il s'agit de déterminer une surface de séparation.

Cet apprentissage s'effectue grâce à la minimisation d'une fonction, appelée *fonction de coût*, calculée à partir des exemples de la base d'apprentissage et de la sortie du réseau de neurones; cette fonction détermine l'objectif à atteindre (minimiser l'écart entre réponses du modèle et données de sorties fournies par l'utilisateur).

Dans la suite de ce chapitre, nous utiliserons les notations suivantes: N est le nombre d'exemples de la base d'apprentissage (nombre de jeux de données entrées-sorties); à chaque exemple i est associée sa classe t_i (codée 1 ou 0); chaque exemple est représenté par un vecteur x_i de dimension n ; les poids du réseau sont représentés par un vecteur w ; la sortie du réseau associée au vecteur d'entrée x_i est notée y_i .

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissages. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie associée) alors que l'on ne dispose que des valeurs (Entrée) pour l'apprentissage non supervisé. Remarquons cependant que les modèles à apprentissage non supervisé nécessitent avant la phase d'utilisation une étape de labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision.

3.1. Algorithmes de minimisation

3.1.1. Principe des algorithmes

Soit $J(w)$ la fonction de coût, les algorithmes utilisés pour la minimiser nécessitent que $J(w)$ soit dérivable par rapport aux poids. Le principe de ces méthodes est de se placer en un point initial, de trouver une direction de descente du coût dans l'espace des paramètres w , puis de se déplacer d'un pas dans cette direction. On atteint un nouveau point et l'on itère la procédure jusqu'à satisfaction d'un critère d'arrêt. Ainsi, à l'itération k , on calcule [86]

$$w_k = w_{k-1} + \alpha_{k-1} \cdot d_{k-1} \quad (2.8)$$

α_k est le pas de la descente et d_k la direction de descente : les différents algorithmes se distinguent par le choix de ces deux quantités.

3.1.2. Descente du gradient

L'algorithme le plus simple consiste à choisir comme direction de descente l'opposé du gradient de la fonction de coût ($d_k = -\nabla J(w) = -\text{Grad}(J(w_k))$).

Cette méthode est efficace, loin du minimum et permet uniquement de s'en approcher. Pour cette raison, la détermination du pas n'est pas cruciale : loin du minimum, il faut seulement vérifier que le pas n'est ni trop petit ni trop grand. En pratique, on utilise, selon les cas, deux méthodes:

Soit un asservissement par la norme du gradient :

$$\alpha_k = \frac{\alpha_0}{1 + \|\nabla J(w_k)\|} \quad (2.9)$$

où α_0 est une constante qui vaut typiquement 0,01.

Soit la méthode de Goldstein [87] pour laquelle le pas est adapté afin de satisfaire deux conditions :

$$J(w_k + \alpha_k d_k) < J(w_k) + m_1 \alpha_k \nabla J^T(w_k) d_k \quad (2.10)$$

$$J(w_k + \alpha_k d_k) > J(w_k) + m_2 \alpha_k \nabla J^T(w_k) d_k \quad (2.11)$$

La première condition s'assure que le pas choisi n'est pas trop grand (sinon l'algorithme risque d'avoir un comportement oscillatoire), alors que la deuxième s'assure qu'il n'est pas trop petit (sinon l'algorithme a une convergence très lente). Les valeurs habituelles pour les deux paramètres m_1 et m_2 sont respectivement de 0,1 et 0,7. Ces méthodes de recherche du pas sont "économiques", car elles ne demandent pas de calculs inutiles de gradient (seul celui dans la direction de descente).

3.1.3. La méthode de Newton

La méthode de Newton utilise la courbure (dérivée seconde) de la fonction de coût pour atteindre le minimum. La modification des paramètres s'écrit ainsi

$$\omega_k = \omega_{k-1} - H_{k-1}^{-1} \cdot \nabla J(\omega_{k-1}) \quad (2.12)$$

La direction de descente est $-H_{k-1}^{-1} \cdot \nabla J(\omega_{k-1})$ et H_{k-1}^{-1} est l'inverse du hessien de la fonction de coût, et le pas est constant et fixé à un. Cet algorithme converge en une seule itération pour une fonction quadratique. C'est donc un algorithme qui est inefficace loin du minimum de la fonction et très efficace près du minimum.

Dans la pratique, le calcul du hessien et surtout de son inverse est à la fois complexe et source d'instabilités numériques; on utilise de préférence une méthode de type "quasi-Newton".

3.1.4. La méthode quasi-Newton

Les méthodes quasi-Newton consistent à approcher l'inverse du hessien plutôt que de calculer sa valeur exacte. La modification des paramètres s'écrit:

$$\omega_k = \omega_{k-1} - M_{k-1} \cdot \nabla J(\omega_{k-1}) \quad (2.13)$$

La suite M_k est construite de façon à converger vers l'inverse du hessien avec M_0 égale à la matrice identité. Cette suite est construite grâce à la méthode Broyden-Fletcher-Goldfarb-Shanno ou BFGS [88] - [90], [91], dont la vitesse de convergence est beaucoup plus grande que celle de la méthode du gradient. De plus, elle est relativement insensible au choix du pas, qui peut être déterminé économiquement par la méthode de Goldstein.

3.1.5. Problème des minima locaux

Les minima trouvés par les algorithmes précédents peuvent être des minima locaux. Le minimum trouvé dépend du point de départ de la recherche c'est-à-dire de l'initialisation des poids. En pratique, il faut effectuer plusieurs minimisations avec des initialisations différentes pour trouver plusieurs minima et retenir le "meilleur". Il est néanmoins impossible et généralement inutile de s'assurer que le minimum choisi est le minimum global. Les réseaux de neurones à couches présentent des symétries, si bien que l'on peut montrer que pour une architecture avec N_c neurones cachés, il existe $2^{N_c} N_c!$ minima équivalents [92].

3.2. Choix de la fonction de coût [42]

Le choix de la fonction de coût (l'erreur qu'il faut minimiser entre les données et les valeurs calculées par le modèle) est conditionné par l'objectif à atteindre.

3.2.1. Erreur quadratique

Pour les problèmes de régression, l'ensemble d'apprentissage est constitué d'exemples pour lesquels la sortie désirée t est une variable continue. La fonction de coût la plus utilisée est l'erreur quadratique sur la base d'apprentissage : elle consiste à minimiser la somme des carrés des erreurs entre la sortie du réseau et la valeur réelle de la sortie [93].

$$J(\omega_k) = \frac{1}{2} \sum_{i=0}^N (y_i(\omega) - t_i)^2 \quad (2.14)$$

Cette fonction de coût est issue du principe de maximum de vraisemblance avec une hypothèse gaussienne sur la distribution des sorties. Pour les problèmes de classification à deux classes (C_0 et C_1), la sortie désirée est une variable binaire codée 1 ou 0 selon que l'exemple appartient respectivement à C_1 ou C_0 . L'hypothèse gaussienne sur la distribution des sorties n'est alors clairement plus vérifiée. Cependant, si l'apprentissage est effectué en minimisant l'erreur quadratique, la sortie du réseau de neurones peut être interprétée comme la probabilité a posteriori, au sens du théorème de Bayes, d'appartenance à la classe C_1 .

3.2.2. Entropie croisée

L'entropie croisée, comme l'erreur quadratique moyenne, est issue du principe du maximum de vraisemblance. Comme l'hypothèse sous-jacente pour l'utilisation de l'erreur quadratique est erronée pour les problèmes de classification, un autre modèle est construit pour tenir compte de la spécificité du codage utilisé dans ces problèmes. Considérons un problème de classification à deux classes C_1 et C_0 où les sorties t sont codées 1 ou 0. Afin que la sortie du réseau de neurones approche la probabilité a posteriori d'appartenir à la classe C_1 , considérons tout d'abord la probabilité d'observer l'une ou l'autre des valeurs de la sortie t en un point de l'espace x si la sortie du modèle est y :

$$p(t|x) = y^t \cdot (1 - y)^{1-t} \quad (2.15)$$

La probabilité d'observer l'ensemble d'apprentissage en supposant que les données sont indépendantes s'écrit :

$$\prod_{i=1}^N (y_i)^{t_i} (1 - y_i)^{1-t_i} \quad (2.16)$$

Pour maximiser cette fonction, on préfère minimiser l'opposé de son logarithme. La fonction de coût utilisée est donc finalement :

$$J(w) = - \sum_{i=1}^N \{ t_i \cdot \ln y_i(w) + (1 - t_i) \cdot \ln(1 - y_i(w)) \} \quad (2.17)$$

Cette fonction, appelée entropie croisée, atteint son minimum lorsque $t_i = y_i$ pour tout i . Par construction, la sortie du réseau est interprétée comme la probabilité a posteriori d'appartenir à la classe C_1 .

3.2.3. Calcul du gradient de la fonction de coût

Les méthodes de minimisation nécessitent généralement le calcul du gradient de la fonction de coût par rapport aux poids du réseau. Les fonctions de coût présentées étant additives, le gradient total est la somme de tous les gradients partiels calculés pour chacun des exemples de la base d'apprentissage.

Ainsi :

$$\nabla J(\mathbf{w}) = \sum_{i=1}^N \nabla J^i(\mathbf{w}) \quad (2.18)$$

Pour chaque exemple, le gradient partiel $\nabla J^i(\mathbf{w})$ est effectué de manière économique grâce à l'algorithme de rétro propagation [22]. La mise en œuvre de cet algorithme, nécessite l'expression analytique de la quantité $\frac{J^i(\mathbf{w})}{y_i(\mathbf{w})}$ où $y_i(\mathbf{w})$ est la sortie du réseau pour l'exemple i .

Si la fonction de coût est l'erreur quadratique, alors

$$\frac{J^i(\mathbf{w})}{y_i} = (y_i - t_i) = e_i \quad (2.19)$$

est la différence entre la sortie du réseau et la sortie désirée, c'est-à-dire l'erreur de modélisation ; on parle alors de "rétro propagation de l'erreur".

- Si la fonction de coût est l'entropie croisée, alors

$$\frac{J^i(\mathbf{w})}{y_i} = \frac{y_i}{y_i(1 - y_i)} \quad (2.20)$$

La modification des poids peut être effectuée, soit après chaque calcul de gradient partiel, soit après le calcul du gradient total.

3.3. Le problème de sous- et sur-apprentissage

3.3.1. Définition du sous- et sur-apprentissage

Si l'on considère un ensemble d'apprentissage et une fonction de coût quadratique, en vertu de la propriété d'approximation universelle [74], il est toujours possible d'obtenir une fonction de coût aussi petite que l'on veut sur l'ensemble d'apprentissage, à condition de mettre suffisamment de neurones cachés.

Cependant, le but de l'apprentissage n'est pas d'apprendre exactement la base d'apprentissage, mais le modèle sous-jacent qui a servi à engendrer les données.

Or, si la fonction apprise par le réseau de neurones est ajustée trop finement aux données, elle apprend les particularités de la base d'apprentissage au détriment du modèle sous-jacent: le réseau de neurones est *sur appris* (trop de données par rapport aux dimensions du problème). Au contraire, si le nombre de données est insuffisant pour bien modéliser le problème, on parle de *sous-apprentissage*.

3.3.2. Biais et variance des modèles

Les phénomènes de sous- et sur-apprentissage sont souvent expliqués grâce aux concepts de biais et de variance introduits dans la communauté des réseaux de neurones [94].

Si l'on considère plusieurs ensembles d'apprentissage, le biais rend compte de la différence moyenne entre les modèles et l'espérance mathématique de la grandeur à modéliser. Le biais est donc lié à la valeur du bruit du processus que l'on cherche à modéliser. La variance rend compte des différences entre les modèles selon la base d'apprentissage utilisée.

On parle souvent de compromis entre le biais et la variance. Si un modèle est trop simple par rapport au processus à modéliser (sous-apprentissage), alors son biais est élevé, mais sa variance est faible puisqu'il est peu influencé par les données. Si un modèle est trop complexe, son biais est faible puisqu'il est capable de s'ajuster exactement à la base d'apprentissage, mais sa variance est élevée puisqu'une nouvelle base avec une réalisation différente du bruit peut entraîner un modèle très différent : c'est le cas du sur-apprentissage.

Ainsi, la complexité du modèle doit être ajustée pour trouver un compromis entre le biais et la variance. Dans leur article [94] on contrôle la complexité du modèle et donc le sur-apprentissage en limitant le nombre de neurones cachés. Cependant il a été montré [95] que cette vision théorique avait des limites pour un réseau à couches dont l'apprentissage était effectué avec une base d'apprentissage comprenant peu d'exemples. En étudiant différentes architectures pour un problème de régression, ils ont montré que le biais et la variance n'évoluent pas nécessairement en sens contraire lorsque le nombre de neurones cachés augmente.

Dans leur cas, un modèle avec quinze neurones cachés à une variance plus élevée qu'un modèle avec soixante neurones cachés.

En résumé, le sur-apprentissage ne s'explique pas seulement par le compromis biais-variance, notamment lorsque le nombre d'exemples est faible. De plus, l'interprétation du sur-apprentissage en ces termes a été développée pour les problèmes de régression et ne se transpose pas simplement aux problèmes de classification. Il en est de même pour le cas du sous-apprentissage.

3.3.3. Les méthodes pour limiter le sous- ou sur-apprentissage

On distingue deux familles de méthodes pour prévenir le sous- ou sur-apprentissage : les méthodes passives et les méthodes actives. Les philosophies de ces deux familles de méthodes sont différentes.

Les méthodes passives essaient de détecter le sous- ou sur-apprentissage *à posteriori* pour supprimer les mauvais modèles. Parmi les méthodes les plus classiques figurent l'utilisation d'une base de validation pendant l'apprentissage, et les mesures de critère d'information.

Les méthodes actives interviennent pendant la phase d'apprentissage pour empêcher le modèle de faire du sous- ou sur-apprentissage. Les méthodes de régularisation comme l'arrêt prématuré ou la pénalisation entrent dans ce cadre.

4. Test du modèle neuronal [73]

C'est l'étape finale de la génération du modèle. Cette étape est exécutée après que les modèles soient générés. Cette étape assure la qualité du modèle produit. Selon la qualité, le modèle doit être réappris si l'erreur de test est trop haute. On a besoin pour cela d'un ensemble indépendant de données connues sous le nom de données de test.

Une quantité δ_{pk} est définie par [73]:

$$\delta_{pk} = \frac{y_k(x_p, w) - d_{pk}}{d_k^{\max} - d_k^{\min}}, \quad k=1 \dots N, \quad p \in T_E \quad (2.21)$$

Un critère de qualité basé sur la $q^{\text{ième}}$ norme est alors défini par

$$M_q = \left[\sum_{p \in TE} \sum_{k=1}^{N_y} |\delta p_k|^q \right]^{\frac{1}{q}} \quad (2.22)$$

Quand $q = 1$, l'erreur moyenne de test peut être calculé directement à partir de M_1 comme :

$$\text{Erreur Moyenne de Test} = \frac{M_1}{N_{TE} N_y} \quad (2.23)$$

où N_{TE} est le nombre d'échantillons dans l'ensemble de données de test et N_y est le nombre de sorties du modèle neuronal. Quand $q = 2$, la norme est la distance euclidienne entre la prévision du modèle et les données de test. Quand $q = \text{infini}$, la mesure de la $q^{\text{ième}}$ norme est l'erreur maximale de test.

5. Conclusion

Les réseaux de neurones formels, tels que nous les avons définis, possèdent une propriété remarquable qui est à l'origine de leur intérêt pratique dans des domaines très divers : ce sont des approximateurs universels parcimonieux.

La propriété d'approximation peut être énoncée de la manière suivante : toute fonction bornée suffisamment régulière peut être approchée avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire.

Le choix le plus populaire est le MLP, depuis que la structure et son apprentissage sont bien établis, et le modèle a une bonne capacité de généralisation. Les réseaux de RBF peuvent être utilisés quand le comportement de problème contient des phénomènes fortement non linéaires ou de fortes variations. Dans ces cas-ci, la nature localisée du RBF le facilite pour apprendre et réaliser une précision respectable du modèle. Cependant, il est nécessaire d'avoir un nombre suffisant de données d'apprentissage.

Une des directions les plus importantes de recherches dans la zone des structures de réseau neuronal orientées micro-onde est la combinaison d'information RF/micro-ondes empirique et des modèles existants de circuit équivalent avec des réseaux neuronaux. Ceci a mené à une classe avancée des structures connues sous le nom de réseaux neuronaux artificiels basés sur la connaissance. [31]

Dans la suite nous allons développer une application basée sur les réseaux de neurones afin de faire des tests sur leur efficacité d'approximation.

Développement de l'outil neuronal

1. Introduction

Le procédé de développement des modèles neuronaux n'est pas trivial et nécessite des étapes critiques telles que la génération de données, le redimensionnement, etc. Bien que certains de ces aspects soient largement décrits dans la littérature, la modélisation d'un réseau neuronal reste relativement nouvelle aux concepteurs micro-ondes. Il faut donc passer en revue les points les plus fondamentaux au développement du modèle neuronal applicable au domaine micro-onde.

2. Développement du modèle de réseau de neurone

Le développement d'un modèle neuronal conventionnel nécessite diverses étapes telles que la collection des données d'apprentissage, la sélection de la structure du réseau neuronal, l'apprentissage du réseau neuronal et le test du modèle neuronal [8]. La clé pour obtenir un modèle précis reste une bonne collection des données d'apprentissage (bien distribuées dans le domaine de travail, suffisantes pour bien englober les variations du phénomène à modéliser, ...). Un organigramme séquentiel récapitulant les diverses étapes nécessaires au développement d'un modèle neuronal est illustré à la figure 3.1.

2.1. Génération de données

L'étape préliminaire dans le développement d'un modèle de réseau de neurones est l'identification des entrées/sorties du modèle. Une fois les entrées/sorties identifiées, les données du dispositif/circuit micro-ondes doivent être rassemblées/produites. Pour des applications micro-ondes, il y a deux types de générateurs de données, à savoir les mesures et les simulations.

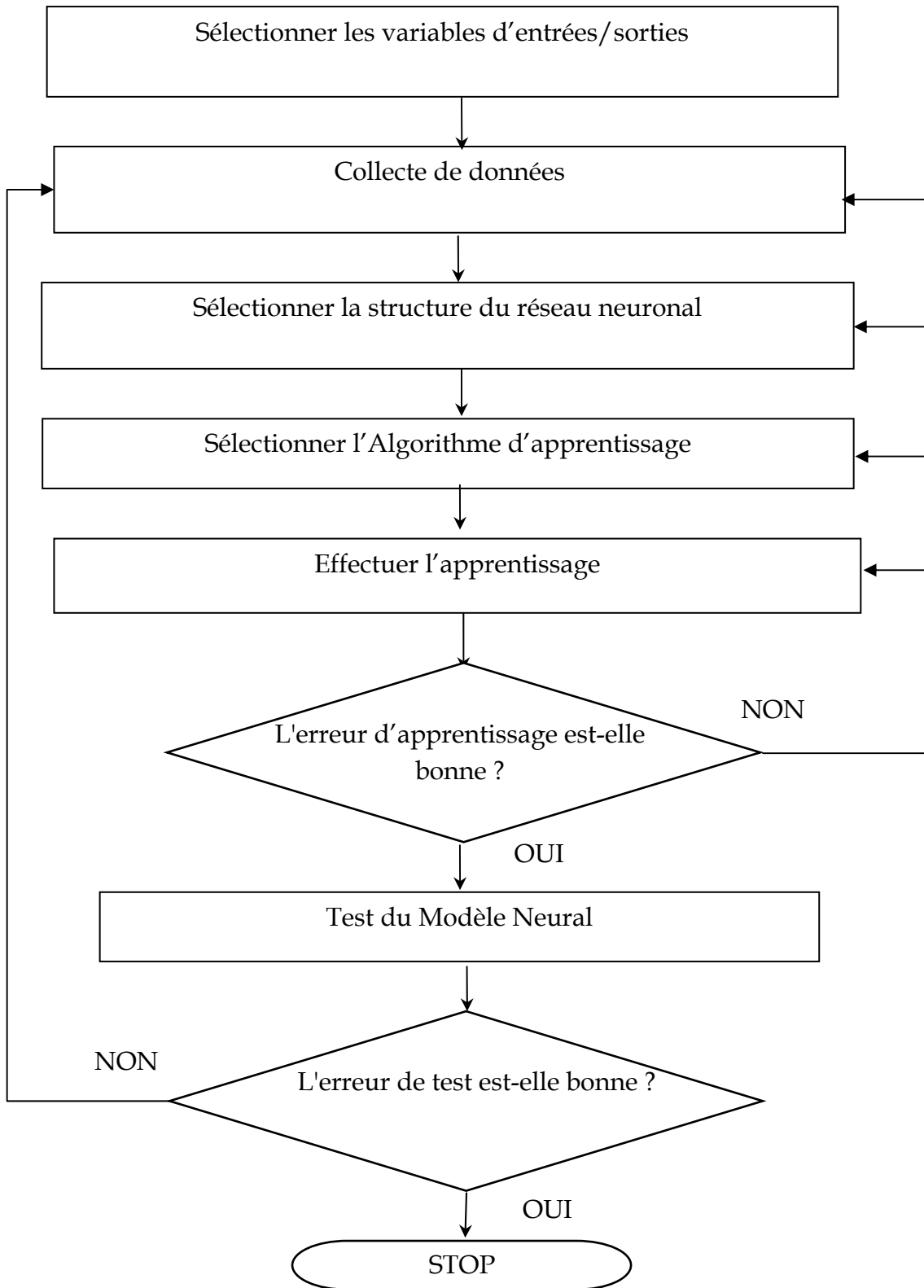


Fig. 3. 1. Organigramme des étapes conventionnelles nécessaires pour le développement du modèle neural [73]

Généralement la génération de données signifie l'utilisation d'un générateur de données pour obtenir la sortie d_p , pour chaque échantillon d'entrée x_p . Le nombre d'échantillons à produire pour un problème micro-onde donné est choisi tel que le modèle neuronal développé représente exactement le problème initial.

2.1.1. Intervalle et distribution des échantillons dans l'espace d'entrée des paramètres du modèle

Une fois toutes les entrées et sorties déterminées, la prochaine étape est de définir l'intervalle d'apprentissage, du test et de la distribution des échantillons de données dans cet intervalle (domaine de définition ou intervalle de variations des variables d'entrées). Notons par $[x_{min}, x_{max}]$ l'intervalle d'une entrée dans lequel le modèle de réseau neuronal serait utilisé après l'apprentissage. Les données d'apprentissage sont généralement échantillonnées légèrement au-delà de l'intervalle du modèle neuronal d'utilisation, c.-à-d., $[x_{min} - \Delta, x_{max} + \Delta]$, afin d'assurer la fiabilité du modèle neuronal aux bornes de cet intervalle. Le test est produit dans l'intervalle $[x_{min}, x_{max}]$. Pour que l'apprentissage soit mené à bonne fin, une distribution des données d'entrées doit être choisie. Les distributions d'échantillons utilisées généralement incluent des distributions uniformes et non-uniformes [8]. Dans la distribution uniforme, chaque paramètre d'entrée x_i est échantillonné à des intervalles égaux. Ce type de distribution est généralement choisi par défaut si aucune information n'est disponible sur le comportement du phénomène à modéliser. Dans la distribution non-uniforme, chaque paramètre d'entrée est échantillonné à des intervalles inégaux. Pour un problème avec forte non linéarité dans certaines sous-régions, il faut un échantillonnage plus dense dans ces sous-régions. Dans ce cas-ci une distribution non-uniforme fonctionne habituellement mieux qu'une distribution uniforme.

2.1.2. Normalisation de données

La normalisation de données est une étape essentielle pour améliorer l'apprentissage des réseaux neuronaux.

L'ordre de grandeur de valeurs d'entrée-sortie dans les applications micro-onde peut être très différent d'un paramètre à un autre (fréquences en GHz, dimensions en mm, ...) et même quelque fois pour un même paramètre (gamme de fréquence de 100 MHz à 10 GHz par exemple) contrairement à la situation binaire (0 et 1) dans des applications typiques de reconnaissance des formes. Il faut donc normaliser toutes les données (entrées et sorties) dans un intervalle $[-1, 1]$ pour que toutes aient le même poids lors de l'apprentissage (figure 3.2).

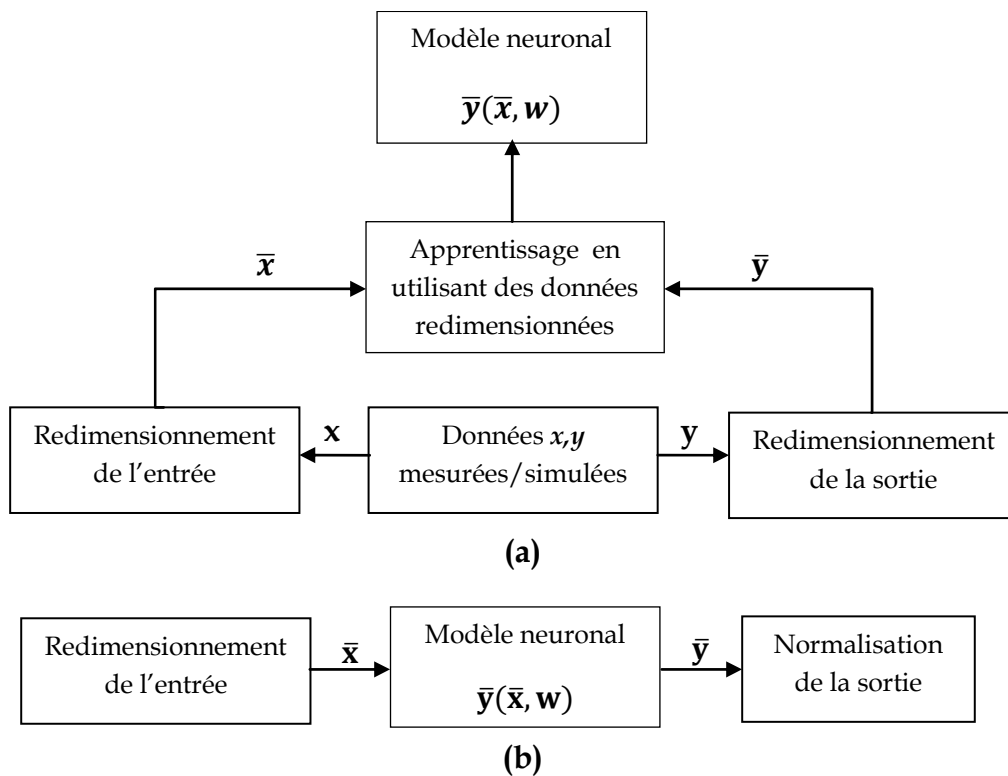


Fig. 3. 2. (a) Redimensionnement d'entrée-sortie dans le développement du modèle neuronal et (b) dans l'utilisation du modèle neuronal [8].

Les types de redimensionnement ou normalisation généralement suggérés sont le redimensionnement linéaire, le redimensionnement logarithmique et le redimensionnement logarithmique bilatéral. Ainsi, soient x_{min} et x_{max} les éléments génériques des vecteurs \mathbf{x}_{min} et \mathbf{x}_{max} des données initiales représentant les valeurs minimales et maximales du vecteur d'entrée \mathbf{x} . Soient \bar{x}_{min} et \bar{x}_{max} les éléments

génériques des vecteurs normalisés \bar{x}_{min} et \bar{x}_{max} des données redimensionnées \bar{x} . Le redimensionnement linéaire est donnée par :

$$\bar{x} = \bar{x}_{min} + \frac{x - x_{min}}{(x_{max} - x_{min})(\bar{x}_{max} - \bar{x}_{min})} \quad (3.1)$$

Le redimensionnement logarithmique est donné par :

$$\bar{x} = \ln(x - x_{min} + 1) \quad (3.2)$$

Généralement le redimensionnement d'entrée-sortie fait que problème est mieux conditionné pour l'apprentissage.

2.2. Choix de la structure

La collecte des informations d'apprentissage peut être très chère pour un problème micro-onde (coût en moyens humains, financiers et en temps). Réduire au minimum le nombre de données d'apprentissage exige une sélection soignée de la structure du réseau neuronal qui aiderait à obtenir une précision satisfaisante avec peu de données d'apprentissage [42]. Par exemple, un réseau neuronal feedforward avec des fonctions d'activation relativement simples dans la couche cachée est adéquat pour la modélisation de fonctions faiblement non linéaires, alors qu'un réseau neuronal feedforward avec des fonctions gaussiennes dans la couche cachée pourrait être plus pertinent pour la modélisation des fonctions fortement non linéaires. En outre, pour les KBNN, on a besoin d'avoir moins de données d'apprentissage pour réaliser une bonne précision [8].

2.3. Effectuer l'apprentissage

Même avec des données d'apprentissage précises et un réseau neuronal approprié, un modèle de réseau neuronal ne peut pas encore garantir pouvoir approcher avec précision le phénomène à modéliser à moins qu'il se soit entraîné par l'algorithme d'apprentissage approprié. L'apprentissage est une étape essentielle dans le développement du modèle de réseau neuronal. Un bon algorithme d'apprentissage peut accélérer l'apprentissage et parvenir à une précision acceptable pour l'utilisateur.

Le procédé d'apprentissage des réseaux de neurones peut être classé selon un apprentissage échantillon par échantillon ou un apprentissage en lots.

Dans l'apprentissage échantillon par échantillon, également appelée apprentissage en ligne, le réseau est mis à jour chaque fois qu'un échantillon d'apprentissage est présenté au réseau.

Dans l'apprentissage en lots, également connu sous le nom de l'apprentissage hors ligne, le réseau est mis à jour après chaque itération, où une itération est requise pour mettre à jour les poids lors du procédé d'apprentissage après chaque lecture et utilisation de toutes les données.

2.3.1. Initialisation des poids du réseau neuronal

Afin de fournir un bon point de départ pour l'apprentissage (optimisation), les poids du réseau neuronal doivent être initialisés. La méthode de génération aléatoire des poids est la stratégie la plus employée pour l'initialisation des poids.

Les poids sont généralement initialisés avec de petites valeurs aléatoires, par exemple, dans l'intervalle $[-0.5, 0.5]$.

2.3.2. Exemple d'algorithme [84]

Soit un réseau neuronal multicouche de type Feedforward, formé de p neurones en entrée, q neurones en sortie, r couches cachées et h_n neurones dans la couche cachée numéro n .

Le but de cet algorithme est de minimiser une erreur quadratique qui est le résultat de la comparaison entre les sorties désirées et celles obtenues (du modèle) en appliquant un exemple de la base d'apprentissage à l'entrée du réseau.

L'erreur obtenue sera redistribuée sur les différentes couches cachées en partant de la couche de sortie pour arriver à la couche d'entrée, d'où le terme rétro-propagation du gradient.

Les différentes étapes de l'algorithme sont les suivantes :

1. Initialiser tous les poids des connexions w_{ij} et les biais $\theta_{i,j}$ aléatoirement par des petites valeurs distribuées uniformément dans un intervalle bien défini. L'indice j représente un neurone appartenant à une couche amont et l'indice i un neurone appartenant à une couche aval.
2. $k=1$ (k représente le jeu de données entrées-sorties. Le fichier de données d'apprentissage est formé de N jeux de données.)
3. Présenter un vecteur d'entrée $\mathbf{X}^k = (x_1^k, x_2^k, x_3^k \dots x_p^k)$ aux neurones de la couche d'entrée et un vecteur de sortie $\mathbf{S}^k = (s_1^k, s_2^k, s_3^k \dots s_p^k)$ aux neurones de la couche de sortie de manière à calculer l'erreur commise par le réseau. Dans notre application, notons que \mathbf{X}^k est le vecteur formé de p paramètres géométriques et physiques à optimiser et \mathbf{S}^k est la réponse du dispositif à modéliser aux valeurs d'entrées (Paramètres S, champs électromagnétiques, gains d'amplificateurs, ...).
4. Calculer les sorties de neurones de toutes les couches (couches cachées et couche de sortie) du réseau. La sortie d'un neurone i est donnée par:

$$o_i = f_i \left(\sum_{j=1}^m w_{ij} \cdot o_j + \theta_i \right) \quad (3.3) \quad \left\{ \begin{array}{l} \text{- Si le neurone est en couche d'entrée :} \\ \quad \mathbf{o}_i = \mathbf{x}_i^k \\ \text{- Si le neurone } i \text{ est en couche CC1, alors :} \\ \quad m = p \text{ et } 1 \leq i \leq h_1 \\ \text{- Si le neurone } i \text{ est en couche CCn, alors :} \\ \quad m = h_{n-1} \text{ et } 1 \leq i \leq h_n \\ \text{- Si le neurone } i \text{ est en couche de sortie :} \\ \quad m = h_r \text{ et } 1 \leq i \leq q \\ \text{- Si le neurone est en couche de sortie :} \\ \quad o_i = y_i^k \end{array} \right.$$

f_i est la fonction d'activation (par exemple une fonction sigmoïde de type

$$f_i = \frac{1}{1+e^{-x}})$$

5. Modifier les poids de connexions w_{ij} et les biais $\theta_{i,j}$

$$\Delta w_{i,j}(k) = w_{i,j}(k+1) - w_{i,j}(k) = -\eta \frac{\partial E_k}{\partial w_{i,j}} = \eta \delta_i o_j \quad (3.4)$$

$$\theta_i(k+1) = \theta_i(k) + \eta \delta_i \quad (3.5)$$

$$\theta_i = w_{i0} \quad (3.6)$$

où :

- η est le coefficient d'apprentissage.
- δ_i est l'erreur locale (gradient):
 - o Si le neurone i appartient à la couche de sortie.

$$\delta_i = (s_i^k - y_i^k) f_i'(A_i) \quad (3.7)$$

- o Si la fonction f_i est une fonction sigmoïde simplifiée alors :

$$\delta_i = (s_i^k - y_i^k) y_i^k (1 - y_i^k) \quad (3.8)$$

- o Si le neurone i étant un neurone caché. l représente les neurones de la couche aval auxquels le neurone i est connecté.

$$\delta_i = f_i'(A_i) \sum_l w_{il} \delta_l, \quad (3.9)$$

$$A_i = \sum_j w_{ij} o_j + \theta_i \text{ est le biais du neurone } i.$$

- o_j est la sortie du neurone j .

6. Calculer l'erreur quadratique:

$$E_k = \frac{1}{2} \left[\sum_{l=0}^q (s_l^k - y_l^k)^2 \right] \quad (3.10)$$

7. Poser $k = k+1$, si $k \leq N$ revenir en (3).

8. Calculer l'erreur totale moyenne :

$$MSE = \frac{1}{2} \sum_{k=0}^N E_k \quad (3.11)$$

Si $MSE \leq \epsilon$ alors arrêter l'apprentissage, sinon revenir en (2) jusqu'à la stabilisation du réseau (avec $\epsilon > 0$ donné).

2.3.3. Sur, sous-apprentissage

La capacité d'un réseau neuronal d'estimer une sortie y pour un ensemble d'entrées x jamais utilisé en apprentissage s'appelle la capacité de généralisation du modèle neuronal. Le sur-apprentissage est observé quand le réseau neuronal mémorise les données d'apprentissage, mais ne peut pas bien les généraliser (surdimensionnement du problème : trop de neurones par exemple). Le sous apprentissage intervient lorsque qu'il n'y a pas suffisamment de données ou de neurones pour bien modéliser le phénomène. Notons par E_{TR} l'erreur d'apprentissage

$$E_{TR}(w) = \sum_{p \in TR} (\epsilon_p(w))^q \quad (3.12)$$

où q représente la q^{eme} norme et $\epsilon_p(w)$ est l'erreur du p^{eme} échantillon, elle est donnée par :

$$\epsilon_p(w) = \left[\frac{1}{q} \sum_{k=1}^{N_y} |y_k(x_p, w) - d_{pk}|^q \right]^{1/q} \quad (3.13)$$

où d_{pk} est le k^{eme} élément de d_p et $y_k(x_p, w)$ est la k^{eme} sortie du réseau neuronal pour l'échantillon x_p

2.4. Test du modèle neuronal

L'étape finale est l'étape de test, c'est-à-dire un ensemble indépendant d'échantillons d'entrée-sortie jamais utilisées en apprentissage et appelées données de test. Elles servent à mesurer la précision du modèle neuronal et sa capacité de généralisation. Au cas où un modèle de réseau neuronal ne répond pas à l'exigence, une action appropriée doit être faite. On pourrait produire plus de données, sélectionner une structure différente de réseau neuronal ou utiliser un algorithme d'optimisation différent.

3. Conclusion

Pour développer une modélisation basée sur les réseaux de neurones, il est nécessaire de générer les données selon des critères précis (distribution des échantillons, intervalle de variations des variables d'entrées/sorties), de les configurer (normalisation des données), de les séparer en données d'apprentissage et données de test (généralement de 50%-50% à 80%-20% selon le problème à modéliser (nombre d'entrées-sorties, degré de non linéarité et/ou discontinuités possibles des relations entre données d'entrées et de sortie ...)) et le nombre des données dont on dispose.

Ainsi, il faut sélectionner la bonne structure de réseau et le bon algorithme d'apprentissage.

Dans le prochain chapitre, nous allons présenter différentes applications en illustrant ces fonctionnalités.

Applications de l'outil neuronal

1. Introduction

Dans ce chapitre, nous allons présenter l'outil développé puis montrer des exemples de simulation réalisés avec cet outil.

Les réseaux de neurones artificiels ont été récemment introduits avec succès pour la modélisation, simulation et optimisation des circuits RF et microondes [18], [47]. Des modèles de réseaux de neurones rapides, précis et fiables peuvent être développés à partir des données mesurées/simulées. Une fois développé, ces modèles peuvent être utilisés à la place des modèles physiques ou électromagnétiques (EM) des composants microondes pour accélérer la conception et l'optimisation.

Nous avons introduit précédemment plusieurs types de structures neuronales. Néanmoins, le perceptron multicouche (MLP) reste, de loin, le plus utilisé [18]. Cette popularité est certes due à sa facilité de mise en forme et de génération de modèles neuronaux, à sa faculté de modéliser des problèmes complexes sans connaissances a priori, mais le plus important point justifiant son emploi intensif reste le théorème d'approximation universel [74], [75] qui forme une base théorique pour l'emploi des réseaux MLP. En effet, en se basant sur ce théorème, un MLP3 (MLP avec une seule couche cachée) peut, théoriquement, modéliser n'importe quel jeu d'entrées-sorties à n'importe quelle précision désirée. Cependant, ce théorème ne spécifie pas quel sera le nombre optimal de données d'apprentissage, le nombre de neurones dans la couche cachée, l'intervalle de définition des variables d'entrées, le type de normalisation, le pas de variation des variables d'entrées ... à sélectionner pour obtenir un modèle neuronal avec une précision donnée.

La détermination des données d'apprentissage et la structure du MLP pour le modèle à entraîner sont les parties les plus difficiles dans le développement d'un modèle de réseau de neurones.

En pratique, si le modèle de réseau de neurones MLP qui a été construit ne donne pas de bons résultats après un premier apprentissage, des étapes additionnelles sont requises. Le concepteur peut ajouter des données d'apprentissages, modifier le nombre de neurones dans la couche cachée, changer la répartition des données dans l'ensemble des données d'entrées utilisées en apprentissage ou choisir une structure de réseau de neurones différente (par exemple un réseau neuronal de type Knowledge Based Neural Network (KBNN) ou Prior Knowledge Input PKI)).

Dans ce travail, nous avons développé un outil neuronal à base de MLP. À travers une série d'exemples, Nous allons montrer son aptitude à générer des modèles neuronaux, mais aussi ses limites dans la génération de modèles afin de donner une vue complète de ses avantages et inconvénients pour de potentiels utilisateurs.

2. Description de l'outil neuronal développé

Dans notre projet, nous avons choisi le langage de programmation Java pour développer l'application.

Nous commençons par importer les données d'apprentissage (figure 4.1) qui se présentent sous la forme de P jeux de valeurs de N entrées et M sorties, soit P ensembles renfermant chacun $N+M$ valeurs $(x_1, \dots, x_N, y_1, \dots, y_M)$.

Puis, nous choisissons la structure de réseau de neurones (figure 4.2).

Entrée 1	Entrée 2	Sortie 1
0.0	0.76525	1.0
0.011111	0.78643	0.98249
0.022222	0.81136	0.96013
0.033333	0.84047	0.93094
0.044444	0.87491	0.8919
0.055556	0.91151	0.83851
0.066667	0.95076	0.76408
0.077778	0.98535	0.65941
0.088889	1.0	0.51456
0.1	0.96449	0.32805
0.11111	0.83767	0.13
0.12222	0.60737	0.0
0.13333	0.34213	0.012001
0.14444	0.14088	0.14343
0.15556	0.035454	0.31178
0.16667	0.0	0.46307
0.17778	0.0018203	0.57928
0.18889	0.019987	0.66647
0.2	0.043805	0.73076
0.21111	0.068359	0.77846
0.22222	0.091384	0.81432
0.23333	0.11274	0.84164
0.24444	0.13167	0.86276
0.25556	0.14849	0.87926
0.26667	0.16339	0.89231
0.27778	0.17659	0.90271
0.28889	0.1883	0.91105
0.3	0.19871	0.91777
0.31111	0.20799	0.92321
0.32222	0.21628	0.92761
0.33333	0.22371	0.93117
0.34444	0.23036	0.93384

Fig. 4.1. Fichier de données avec N+M colonnes pour N entrées et M sorties (dans cet exemple, N = 2, M = 1).

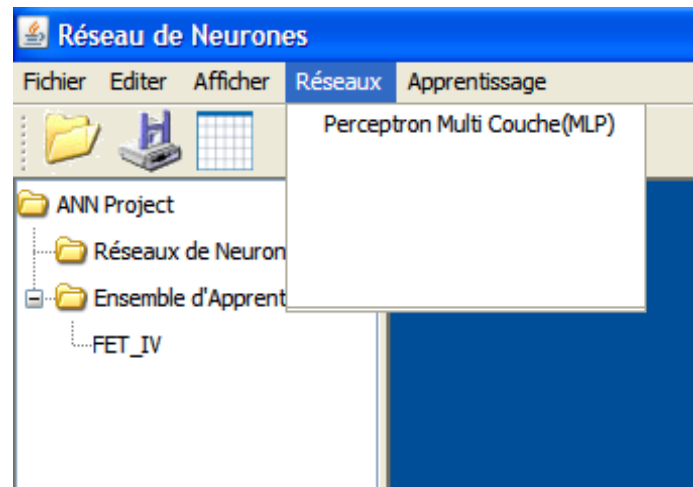


Fig. 4.2. Choix de la structure du réseau de neurones

Etant le plus largement utilisé, nous nous sommes concentrés sur le réseau de neurones perceptron multicouches MLP (figure 4.3). On introduit alors la hiérarchie du réseau (nombre de neurones dans chaque couche : entrée, cachée(s) et sortie) ainsi que la fonction d'activation (Figure 4.4).

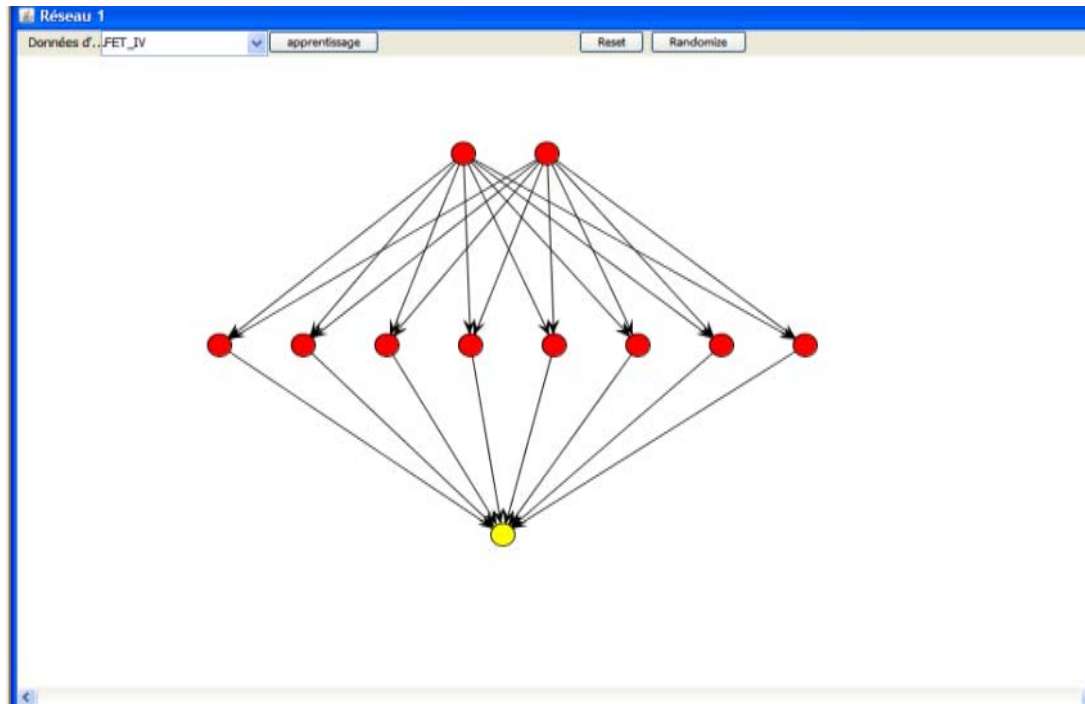


Fig. 4.3. Réseau MLP à trois couches ou MLP3 (une couche cachée).

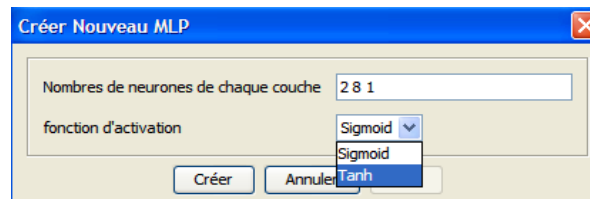


Fig. 4.4. Choix de la fonction d'activation.

Finalement le réseau est créé et prêt à commencer l'apprentissage. Pour plus de simplicité et en accord avec le théorème d'approximation universel, nous nous sommes servis exclusivement de réseaux MLP3 (une seule couche cachée). Pour cela il faut d'abord définir les paramètres de l'apprentissage (erreur maximale désirée, nombre maximal d'itérations, intervalle des paramètres d'entrée, pas de variation ...) (Figure 4.5). On procède alors à l'apprentissage en évaluant l'erreur d'apprentissage (Figure 4.6). L'algorithme d'apprentissage de descente de gradient a été utilisé en premier, puis lorsque l'algorithme est proche de la solution, la méthode quasi-Newton a été utilisée.

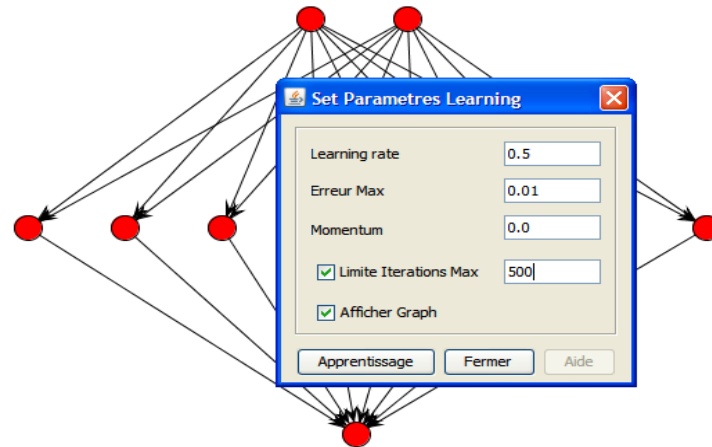


Fig. 4.5. Création du réseau et définition des paramètres d'apprentissage.

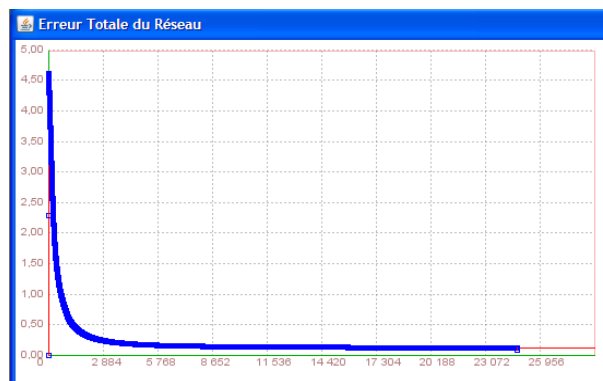


Fig. 4.6. Evolution de l'erreur d'apprentissage en fonction du nombre d'itération

3. Résultats Obtenus

Nous avons utilisé l'outil que nous avons développé pour modéliser différents types de composants ainsi que des circuits dont les données d'apprentissage sont extraites de simulations sur des logiciels commerciaux, en l'occurrence Ansoft-HFSS [96] pour les composants et Agilent-ADS [97] pour les circuits.

3.1. Modélisation de composants micro-ondes

1- Inductance

Le premier composant à être modélisé est une inductance utilisée dans les modules multi-puces (*multichip modules*) (Figure 4.7).

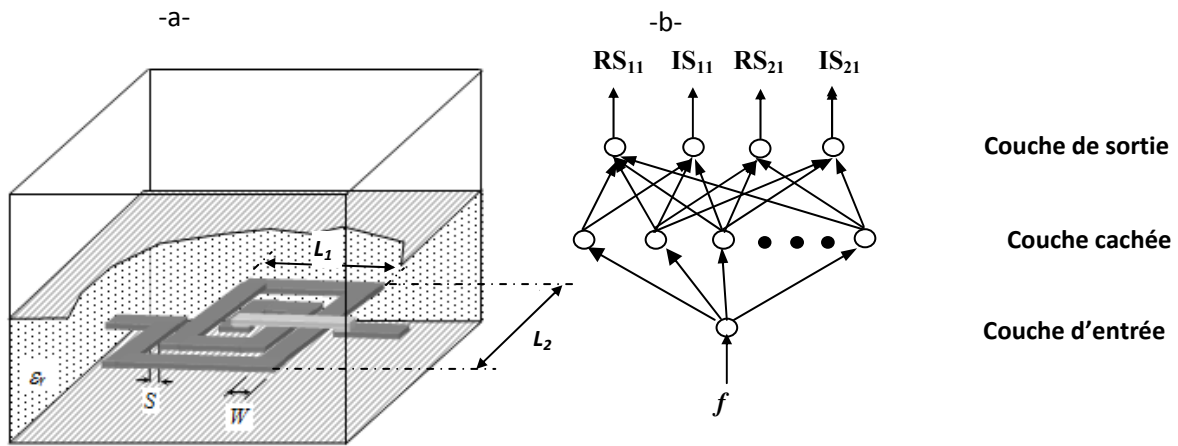


Fig. 4.7. a- Structure physique 3-D d'une inductance encastrée. La fréquence est la seule variable ajustable dans la conception de ce composant multicouche.

b- Structure du modèle de réseaux de neurones.

La modélisation de tels composants passifs s'appuie généralement soit sur l'utilisation de circuits électriques équivalents [98], [99], soit sur une formulation intégral-différentielle des équations de Maxwell et leur résolution par différentes méthodes numériques pour quantifier le champ électromagnétique (EM) au niveau de la structure étudiée [98] - [102].

Si la première pêche par une précision quelque peu relative en haute fréquence et large bande, la seconde approche a certes montré son efficacité du point de vue précision, mais elle reste largement dominée par un temps de calcul très long et des capacités mémoire conséquentes [9].

Ainsi, les cycles d'optimisation ou de type Monte-Carlo, indispensables dans tout processus de conception, limitent de manière drastique leur implantation dans les simulateurs de circuits. De plus, une fois insérés dans ces simulateurs, les modèles électromagnétiques sont figés car les paramètres électriques ne sont valables que pour une topologie donnée, donc que pour un jeu préétabli de paramètres électriques et géométriques des composants passifs et des lignes de connexions qui les relient.

De là, le développement de modèles électromagnétiques micro-ondes de composants passifs pour les simulateurs de circuits, à la fois précis, rapides et dont les performances électriques soient continûment variables en fonction de leurs paramètres électriques et géométriques de base, demeure une tâche colossale autant qu'un stimulant défi pour les chercheurs.

De ce fait, la modélisation par réseau de neurones reste la plus efficace du point de vue rapidité et précision. Dans ce travail, elle se base sur des données de simulation issues d'un logiciel EM-3D.

Pour ce premier exemple, nous allons modéliser une seule entrée, la fréquence. Les données du réseau sont alors :

- 1 entrée : fréquence [0.1-10 GHz, pas aléatoire].
- 5 neurones cachés
- 4 sorties : amplitude et phase de S_{11} et S_{12}
- Nombre de données d'apprentissage/test : 67/33

Les caractéristiques de l'inductance à trois tours sont :

L_1 : Longueur du segment à l'extérieur $762\mu\text{m}$

L_2 : Longueur du second segment à l'extérieur, $508\mu\text{m}$

W : largeur du Conducteur, $25.4\mu\text{m}$

S : espacement entre conducteurs, $25.4\mu\text{m}$

T : Epaisseur du conducteur $5.08\mu\text{m}$

Résultats :

- Nombre d'itérations : 200
- Erreur moyenne totale du réseau : 0.012913

On constate que les données d'apprentissage et les résultats obtenus par le modèle neuronal sont proches l'un de l'autre (Figures 4.8). Cet exemple montre que même avec peu de données, il est possible d'obtenir un modèle acceptable (erreur moyenne de l'ordre de 1.3 % par rapport aux données expérimentales).

Ainsi, lorsque les fonctions entrées-sorties ne sont pas très complexes et que le nombre de variables d'entrée est faible, on peut utiliser peu de données d'apprentissage et de neurones cachés.

Néanmoins, la précision du modèle peut être améliorée en augmentant le nombre de données (ce qui peut être difficile en pratique car chaque simulation dans un logiciel électromagnétique 3D peut prendre d'une à quatre heures) ou en ayant un pas plus uniforme. En effet, le pas aléatoire est certes pratique lorsque nous ne connaissons pas la tendance de variation du composant à modéliser, mais avec aussi peu de données, il est malheureusement hautement probable que plusieurs paquets de données soient concentrés dans des zones bien limitées du domaine de variation de la donnée d'entrée alors qu'il faudrait des données bien réparties sur tout le domaine de variation.

Avec une telle approche, le résultat aurait pu être meilleur. Cette remarque devient cruciale lorsque le nombre de données augmente. C'est le cas du second exemple ci-dessous.

2- Résistance

Le second composant à être modélisé est une résistance utilisée dans les modules multi-puces (*multichip modules*) (Figure 4.9). Nous allons utiliser plusieurs entrées et aussi plus de données/entrée comparativement à l'exemple précédent :

- 5 entrées: fréquence [6-20 GHz, pas de 2], largeur [6-20 μm , pas de 2], longueur [2-7 μm , pas de 1], permittivité [10, 50-200, pas de 50] et résistivité [1-10 Ω , pas de 1].
- 20 neurones cachés.
- 2 sorties: partie réelle de S_{11} et S_{12} .
- Nombre de données d'apprentissage/test : 14750/2950

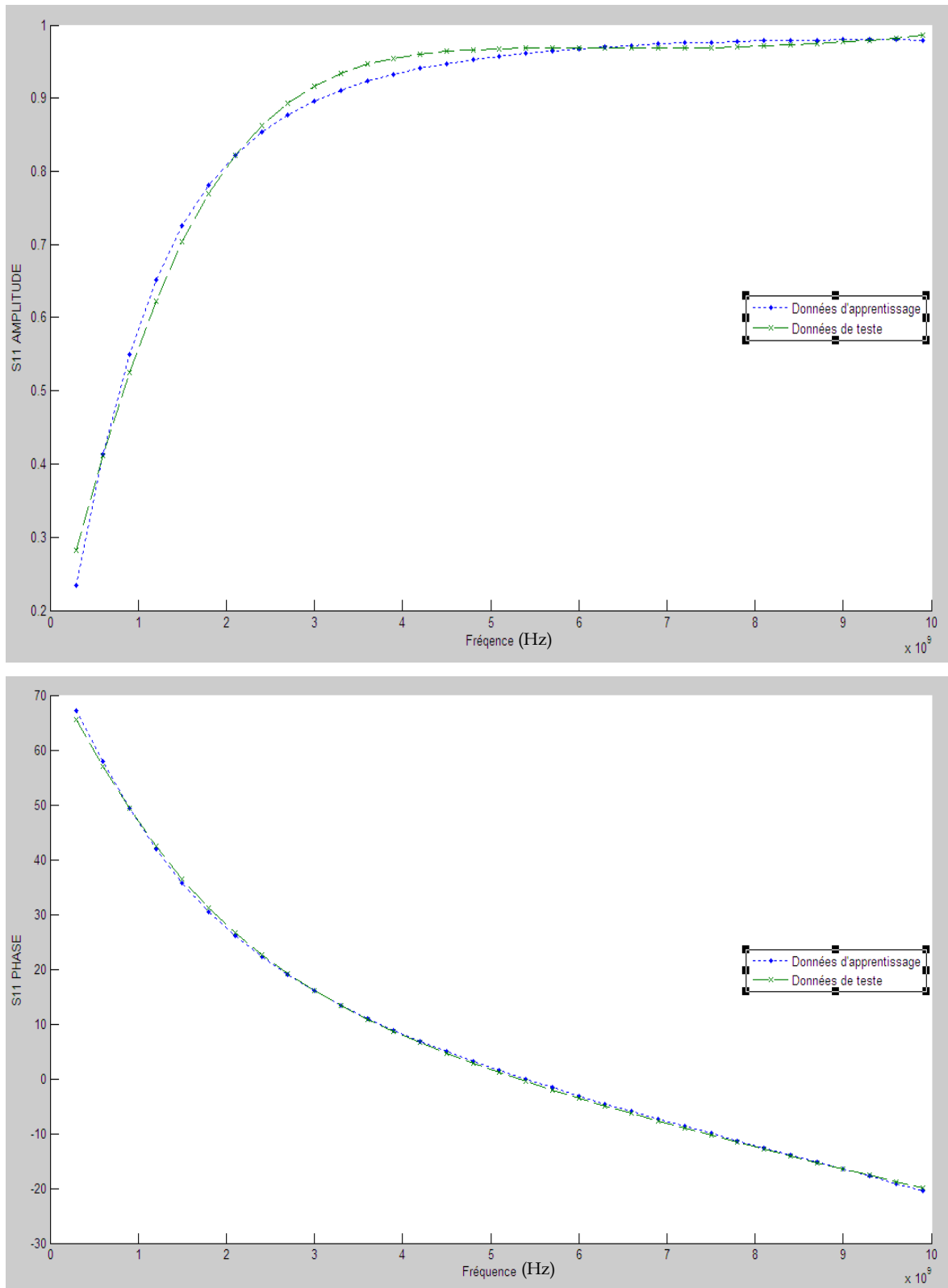


Fig.4.8.a Résultats d'apprentissage pour l'amplitude et la phase du paramètre S_{11} d'une inductance avec 67 échantillons, 200 itérations (comparaison avec 33 échantillons de test).

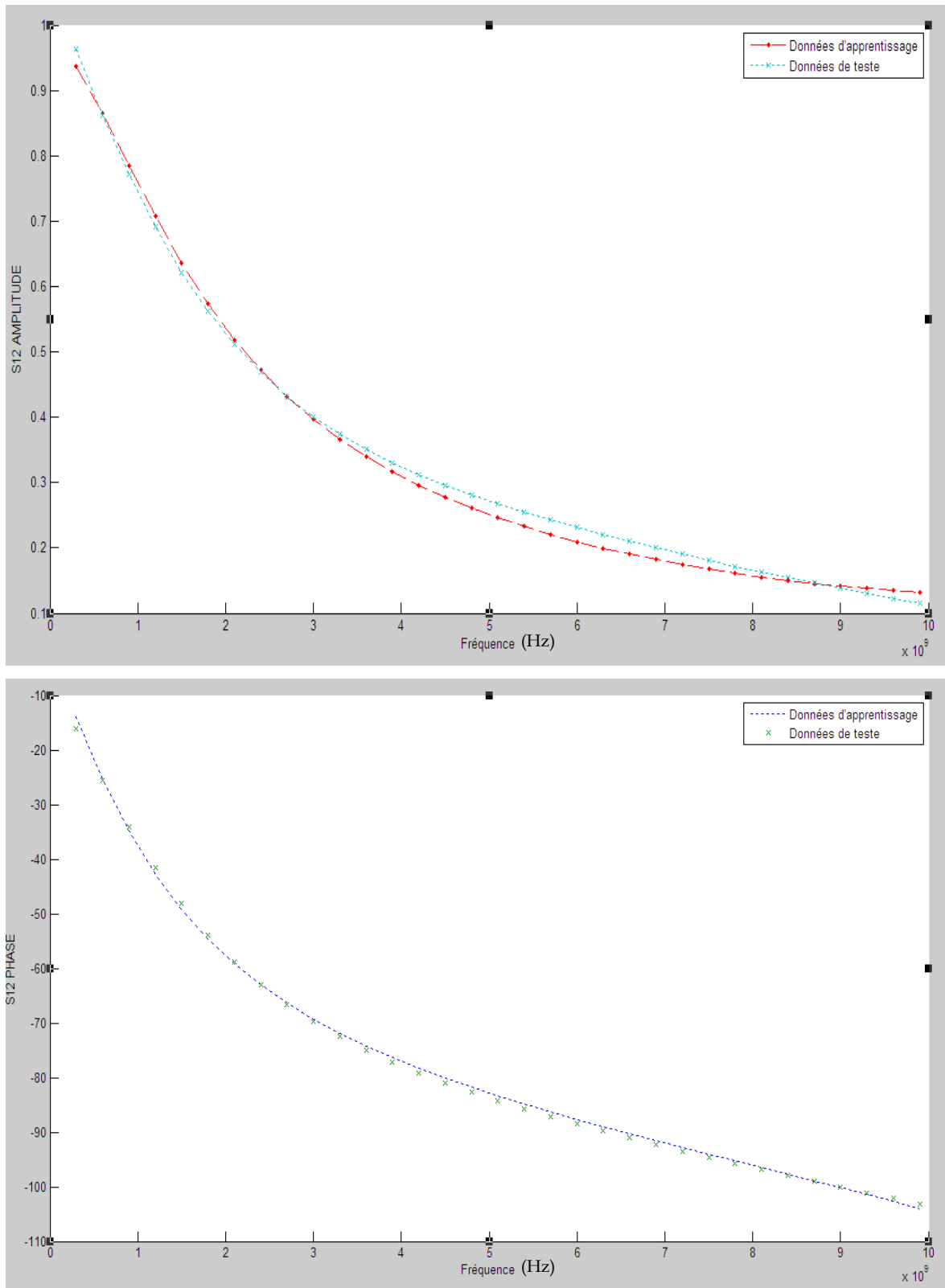


Fig. 4.8-b. Résultats d'apprentissage pour l'amplitude et la phase du paramètre S_{12} d'une inductance avec 67 échantillons, 200 itérations (comparaison avec 33 échantillons de test).

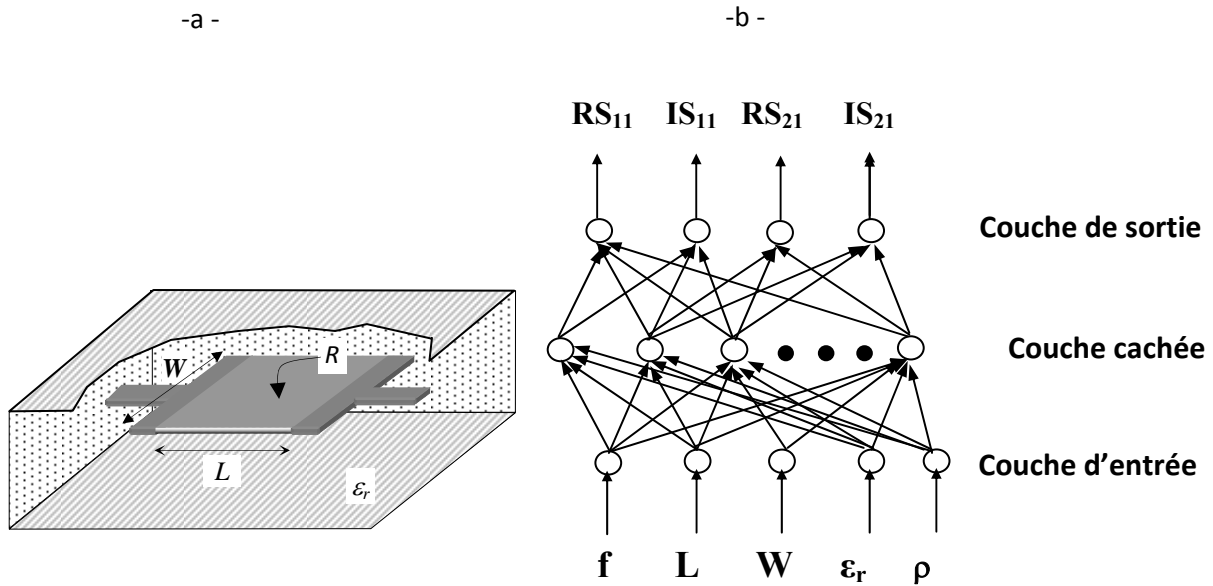


Fig. 4.9. a- structure physique 3-D d'une résistance encadrée. Fréquence, Longueur, Largeur, Permittivité et Résistance sont des variables ajustable dans la conception de ce composant multicouche. b- Structure du modèle de réseaux de neurones

Résultats

- Nombre d'itérations : 200
- Erreur totale moyenne du réseau : 0.024678

On constate que les données d'apprentissage et les résultats obtenus par le modèle basé sur les réseaux de neurones sont proches l'un de l'autre (Figures 4.10), bien que le processus de génération du modèle neuronal soit plus complexe à cause du très grand nombre d'échantillons.

De plus, même si l'erreur moyenne dans ce deuxième exemple de composant est supérieure à celle obtenue pour le premier composant, soit 2.5% versus 1.3%, elle reste acceptable pour un concepteur comme le confirme les figures 4.10. Ainsi, si le nombre d'échantillons est important, le réseau de neurones peut mieux approcher le modèle à approximer.

Cela signifie que l'erreur du pire-cas est nettement plus faible dans le deuxième exemple que dans le premier.

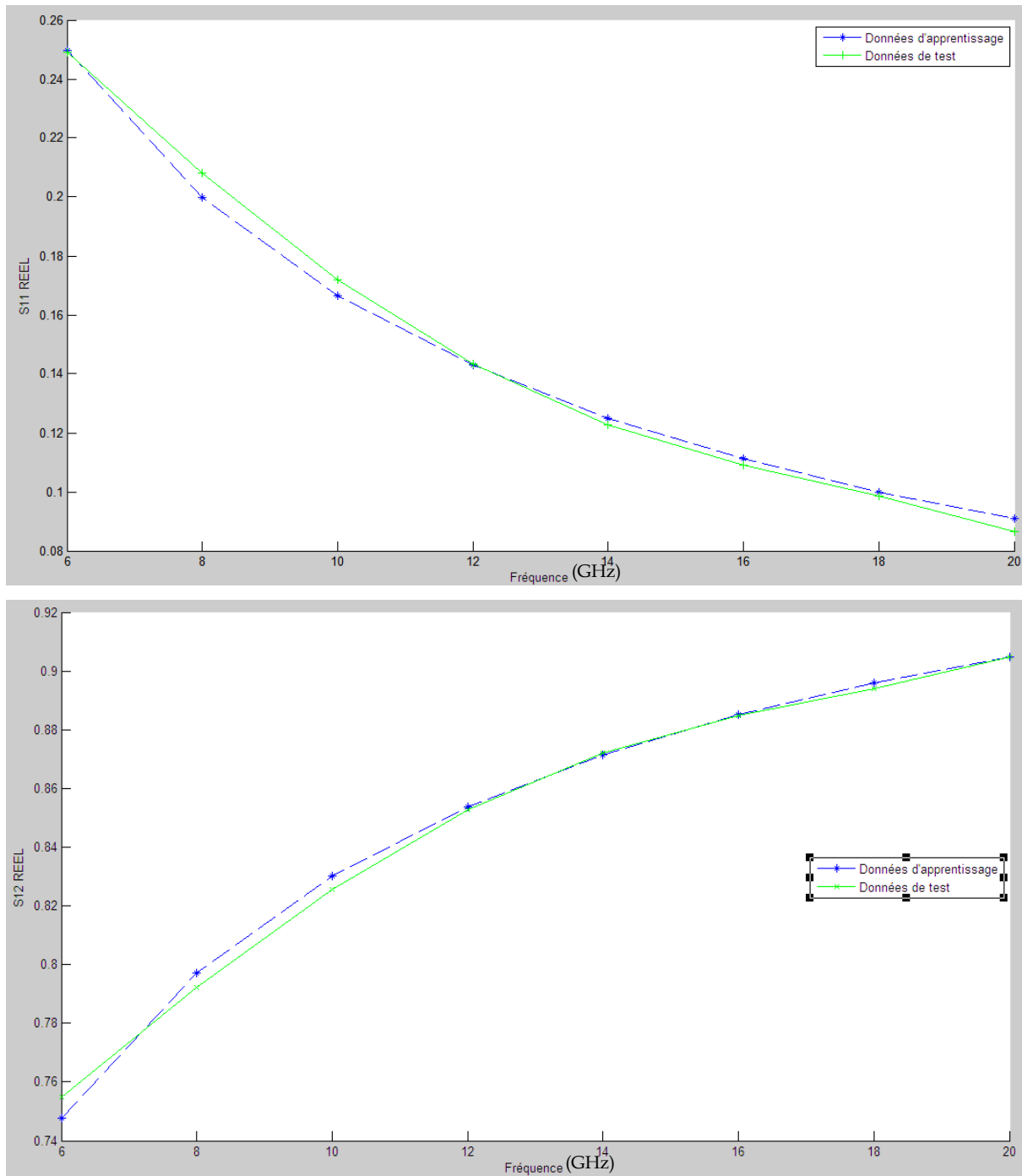


Fig. 4.10. Résultats d'apprentissage pour les parties réelles des paramètres S_{11} et S_{12} d'une résistance avec 14750 échantillons, 200 itérations (comparaison avec 2950 échantillons de test).

3- Transistor HEMT

Pour ce troisième exemple, nous avons modélisé un composant actif, un transistor HEMT dont les caractéristiques sont données à la référence [18]. Nous allons retenir une seule entrée, toujours la fréquence, mais cette fois-ci avec un pas fixe. De plus, nous avons limité au minimum le nombre de données d'apprentissage tout en augmentant le nombre de sorties. Les données du réseau sont alors :

- 1 entrée: fréquence [3-18 GHz, pas de 0.1].
- 5 neurones cachés
- 8 sorties: amplitude et phase de S_{11} , S_{12} , S_{21} et S_{22}
- Nombre de données d'apprentissage/test : 121/30

Résultats :

- Nombre d'itérations : 200
- Erreur totale moyenne du réseau : 0.011909

On constate que les données d'apprentissage et les résultats obtenus par le modèle basé sur les réseaux de neurones sont très proches l'un de l'autre, bien que le nombre de données soit nettement inférieur au deuxième exemple et que le nombre de sorties est nettement plus élevé (Figures 4.11).

Ces exemples montrent bien la flexibilité des réseaux de neurones et leur grande adaptation selon le problème à modéliser. Ainsi, la précision obtenue est très acceptable (1.2%) même avec peu de données. Il faut donc adapter le nombre de données avec le problème à modéliser (si l'utilisateur a une idée sur la tendance de variation des paramètres qu'il veut modéliser).

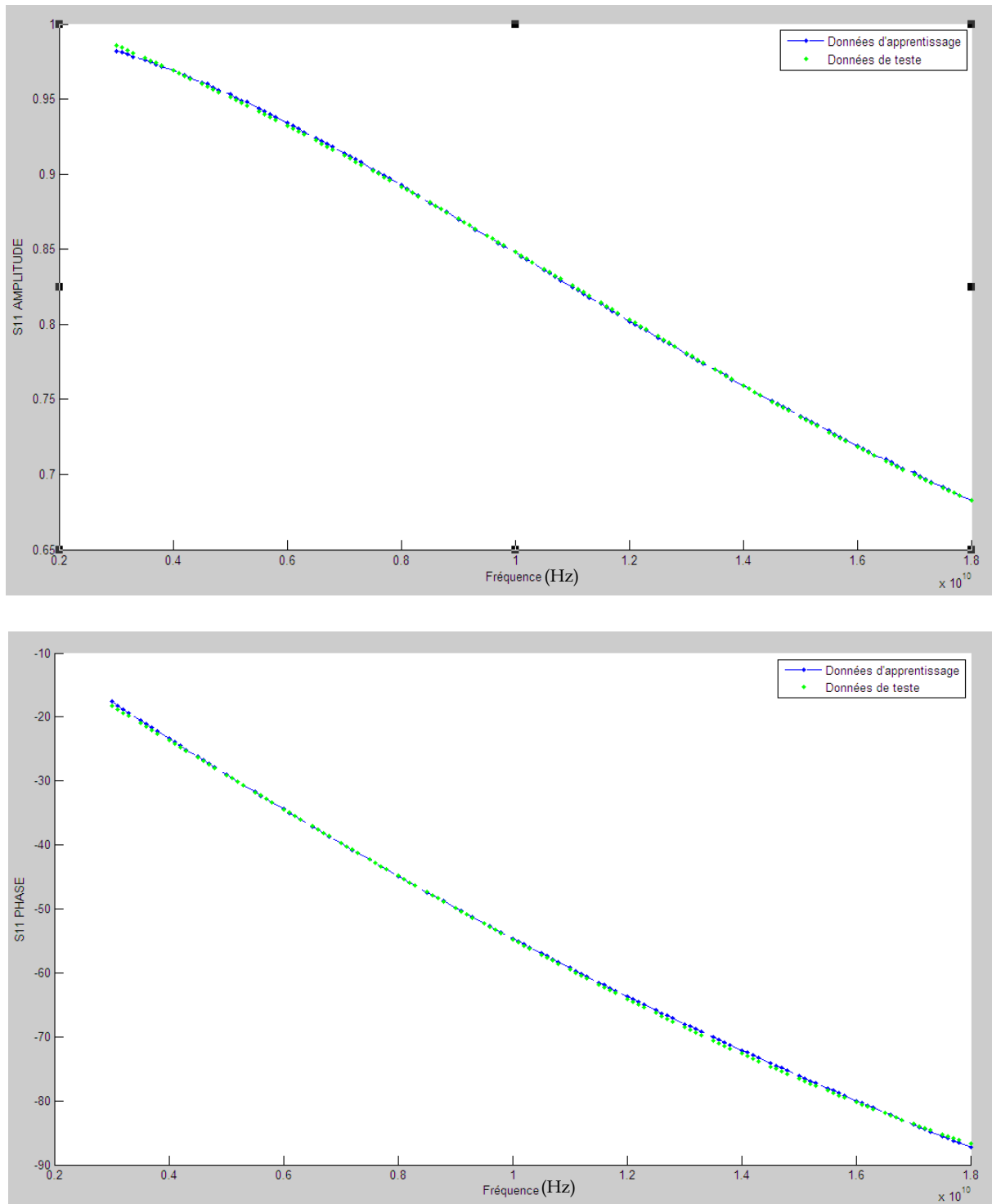


Fig. 4.11-a. Résultats d'apprentissage pour l'amplitude et la phase du paramètre S_{11} d'un transistor avec 121 échantillons, 200 itérations (comparaison avec 30 échantillons de test).

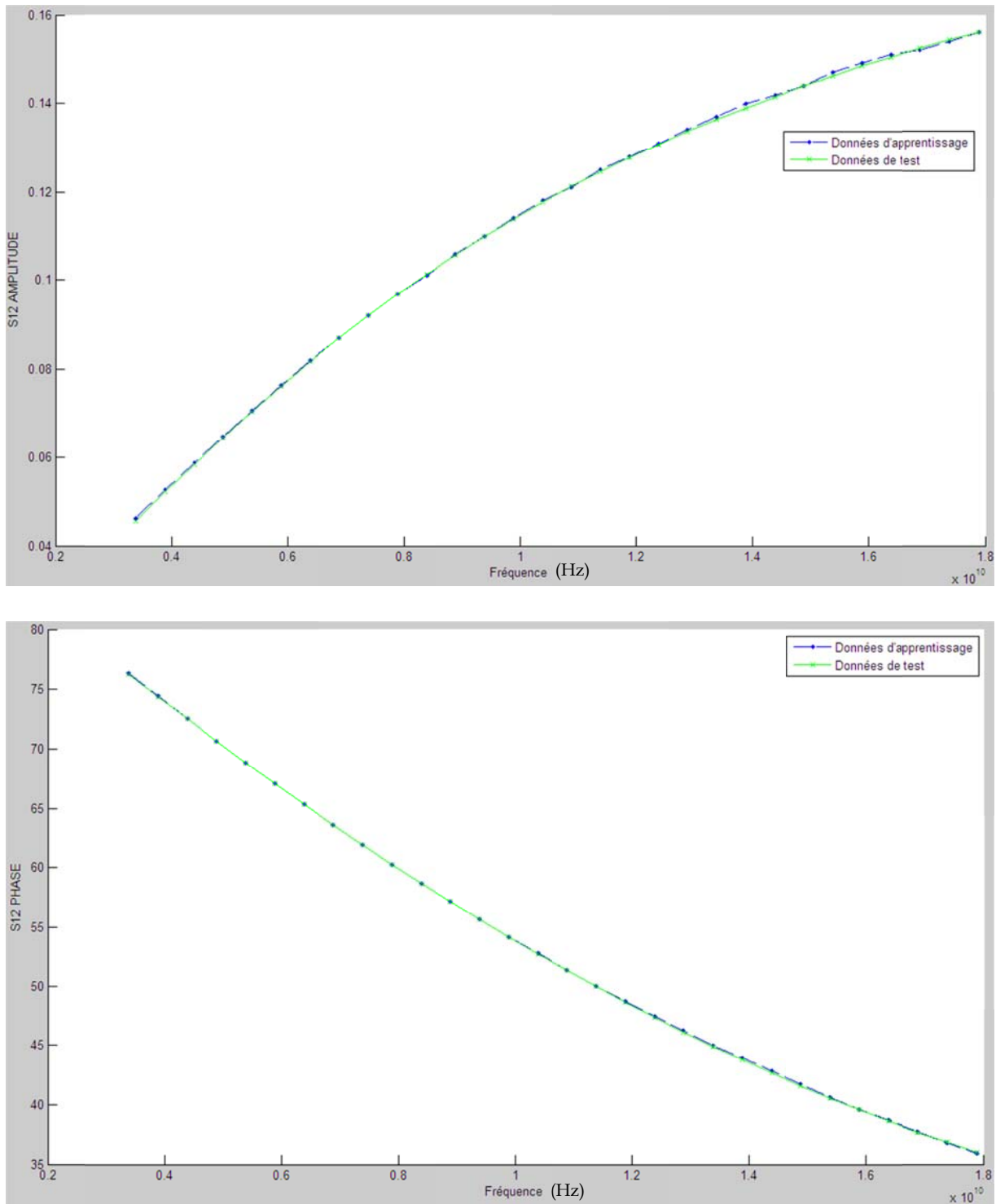


Fig. 4.11.-b. Résultats d'apprentissage pour l'amplitude et la phase du paramètre S_{12} d'un transistor avec 121 échantillons, 200 itérations (comparaison avec 30 échantillons de test).

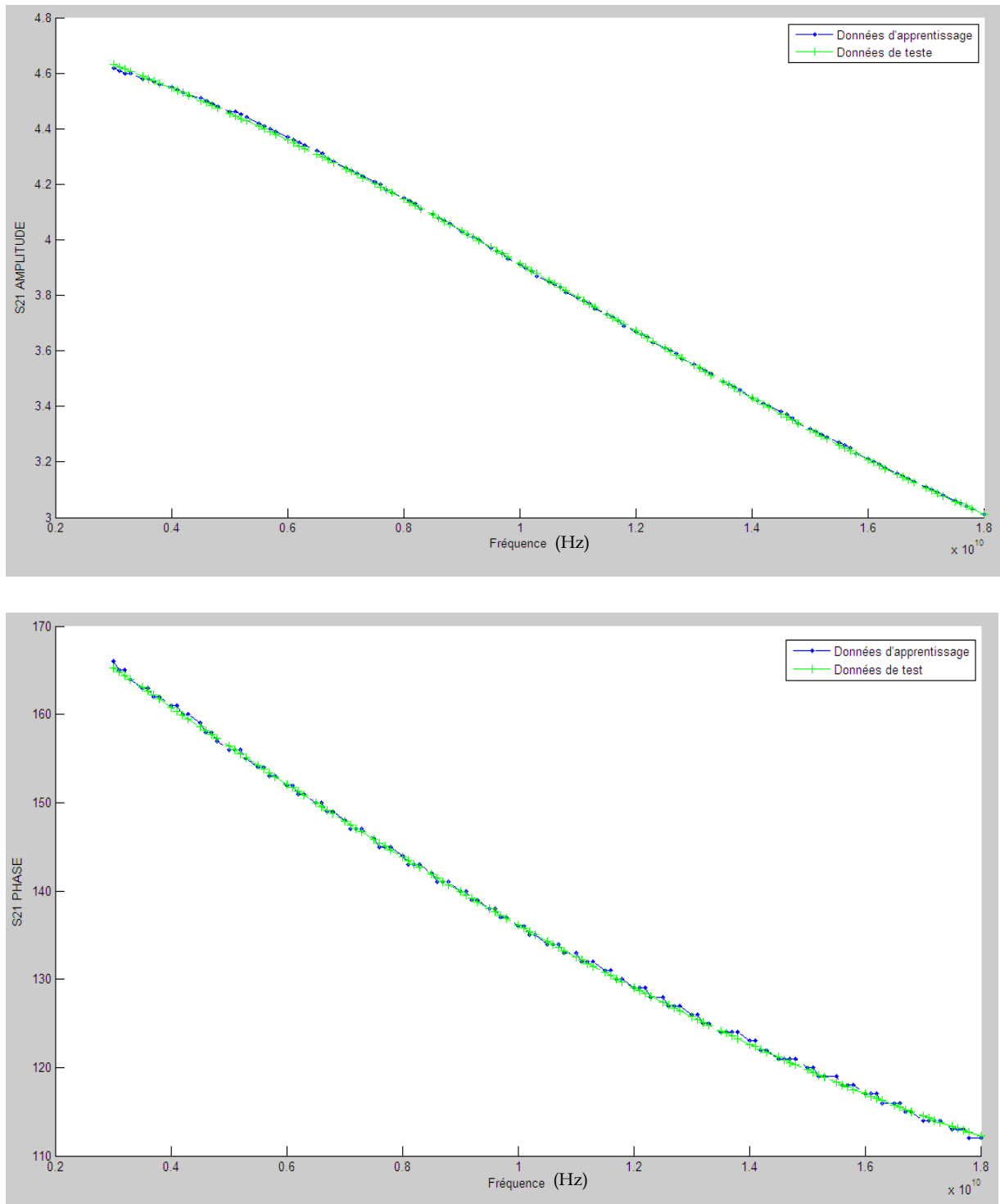


Fig. 4.11.-c. Résultats d'apprentissage pour l'amplitude et la phase du paramètre S_{21} d'un transistor avec 121 échantillons, 200 itérations (comparaison avec 30 échantillons de test).

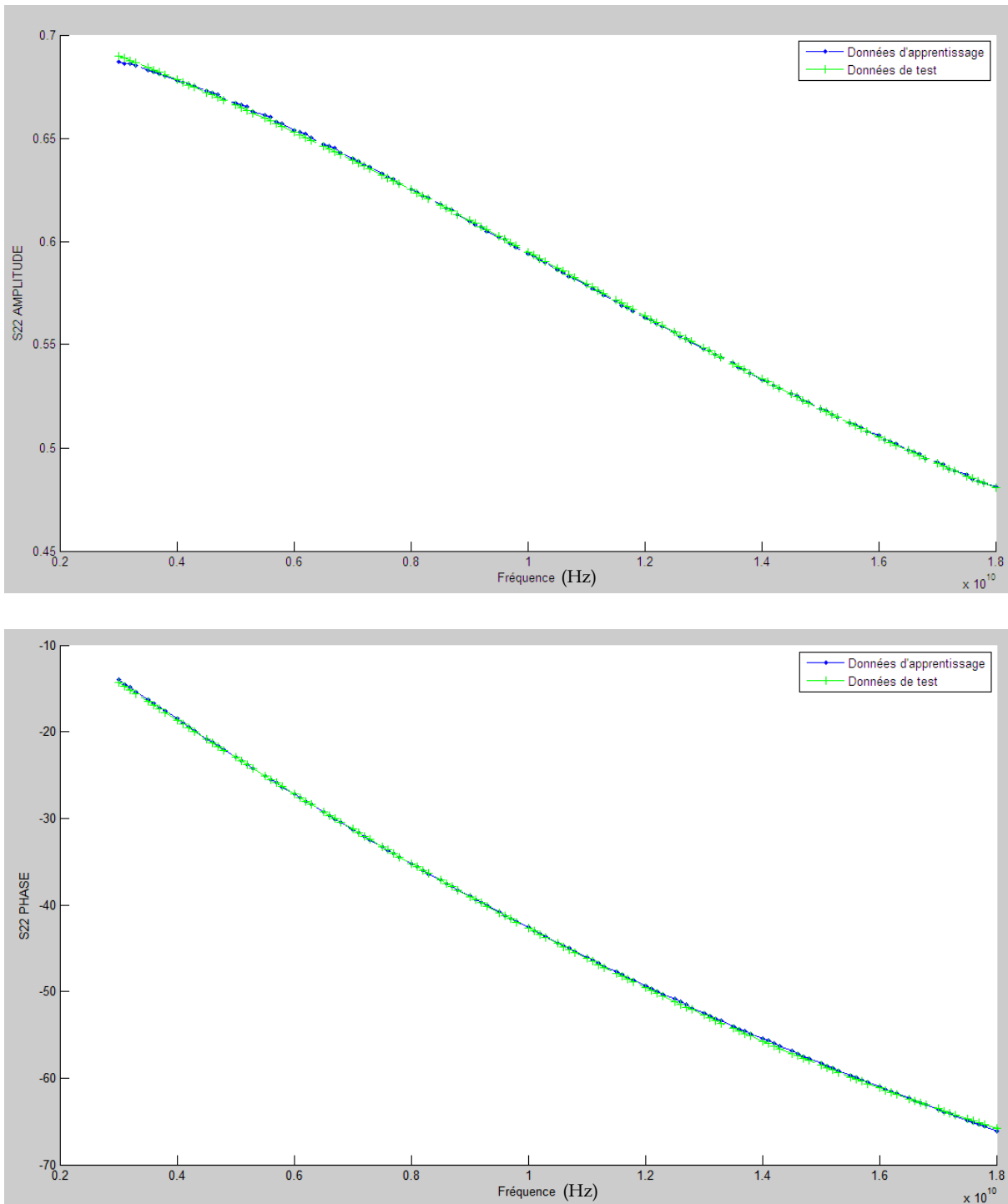


Fig. 4.11-d. Résultats d'apprentissage pour l'amplitude et la phase du paramètre S_{22} d'un transistor avec 121 échantillons, 200 itérations (comparaison avec 30 échantillons de test).

3.2. Modélisation de circuits micro-ondes

Nous avons retenu un amplificateur de puissance pour tester notre outil sur la modélisation de circuits micro-ondes. Le circuit de cet amplificateur est illustré à la figure 4.12. C'est un amplificateur de puissance de la bibliothèque du logiciel commercial Agilent-ADS à base de huit transistors BJT et utilisant deux types de transistors à savoir le Q34MOD et Q37MOD de Motorola, cet amplificateur est polarisé à 5.8V au port supérieur et à 2V au port inférieur.

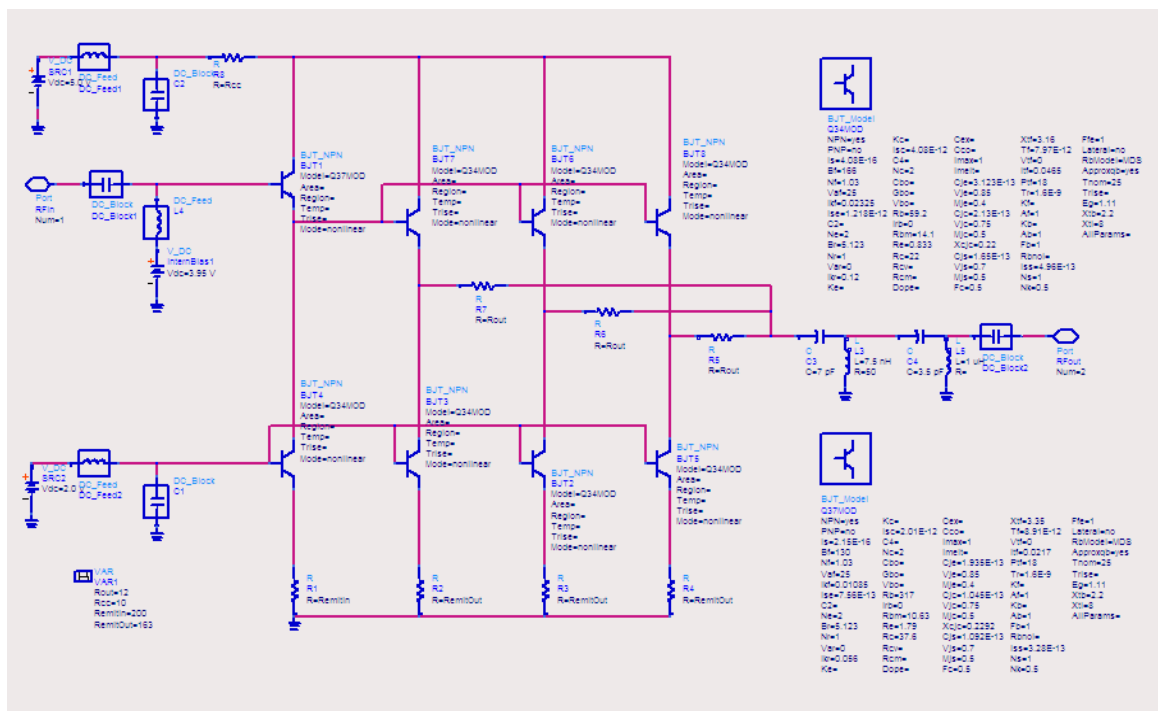


Fig. 4.12.. Schéma de l'amplificateur de puissance utilisé.

Nous avons considéré plusieurs cas de figure pour sa modélisation en changeant cette fois-ci le nombre d'itérations et de neurones dans la couche cachée et avec

- 1 entrée: fréquence [1-10 GHz, pas de 0.1].
- 4 sorties: partie réelle de S_{11} , S_{12} , S_{21} et S_{22}
- Nombre de données d'apprentissage/test : 91/37

Premier cas :

- Nombre d'itérations : 100
- Nombre de neurones dans la couche cachée : 10
- Erreur moyenne totale du réseau : 0.240132

Deuxième cas :

- Nombre d'itérations : 1000
- Nombre de neurones dans la couche cachée : 10
- Erreur moyenne totale du réseau : 0.090735

Troisième cas :

- Nombre d'itérations : 100
- Nombre de neurones dans la couche cachée : 15
- Erreur moyenne totale du réseau : 0.145008

Quatrième cas :

- Nombre d'itérations : 1000
- Nombre de neurones dans la couche cachée : 15
- Erreur moyenne totale du réseau : 0.026382

De ces résultats (Figures 4.13), nous pouvons conclure que le modèle extrait après 100 itérations est trop imprécis : pas suffisamment d'itérations. Tandis que le modèle extrait après 1000 itérations se rapproche nettement plus des données originales, ce qui était prévisible. Ce nombre est à comparer avec les 200 itérations utilisées lors des exemples précédents.

On peut facilement constater l'effet du nombre d'itérations sur la précision du modèle. Il faut donc optimiser ce paramètre en fonction de la taille du problème.

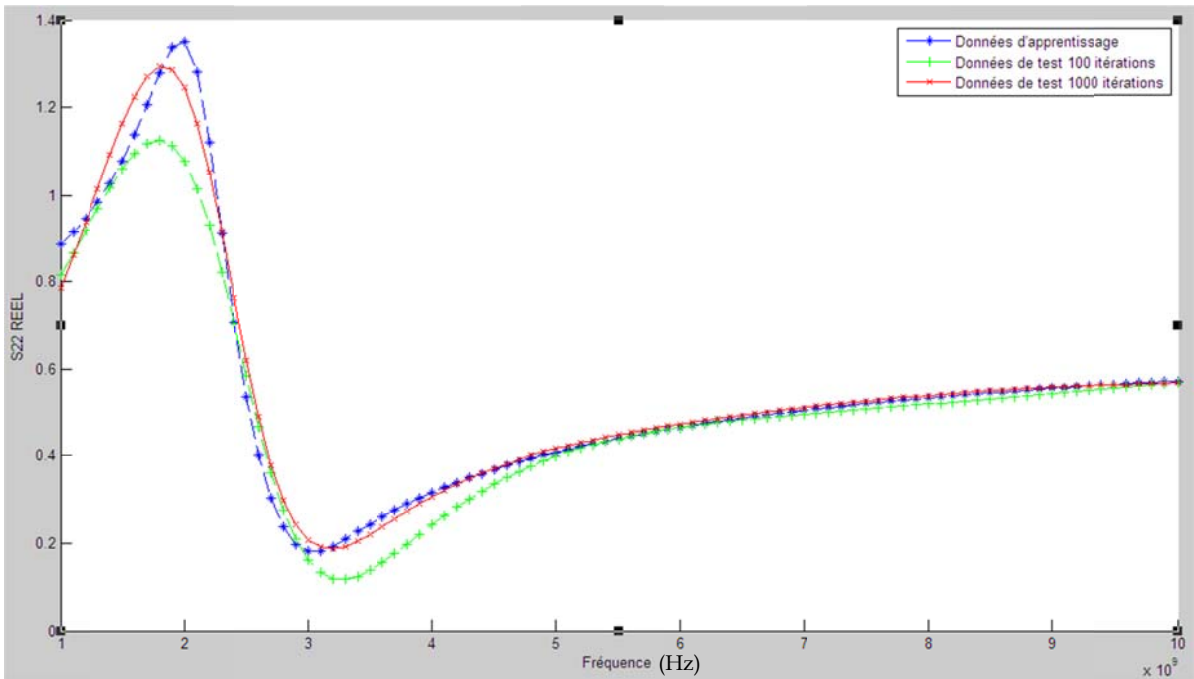
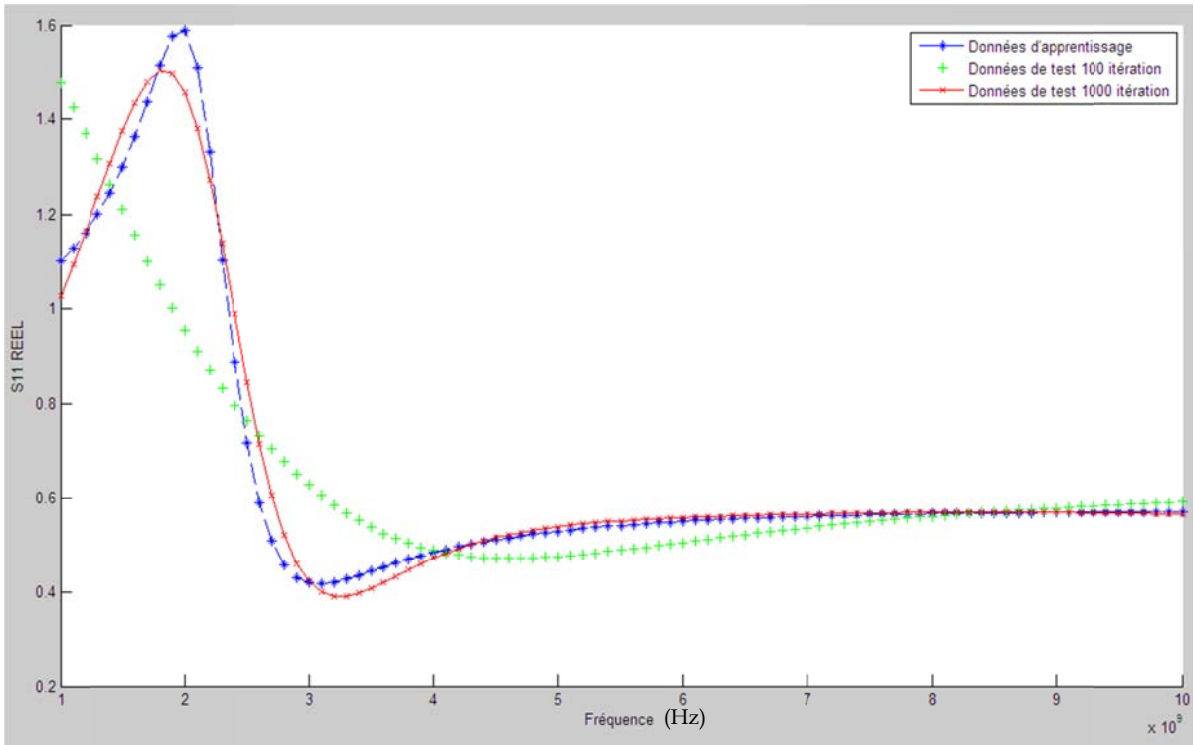


Fig. 4.13.-a. Résultats d'apprentissage pour la partie réelle des paramètres S_{11} et S_{22} d'un amplificateur de puissance avec 91 échantillons et 15 neurones, en faisant varier le nombre d'itérations (comparaison avec 37 échantillons de test).

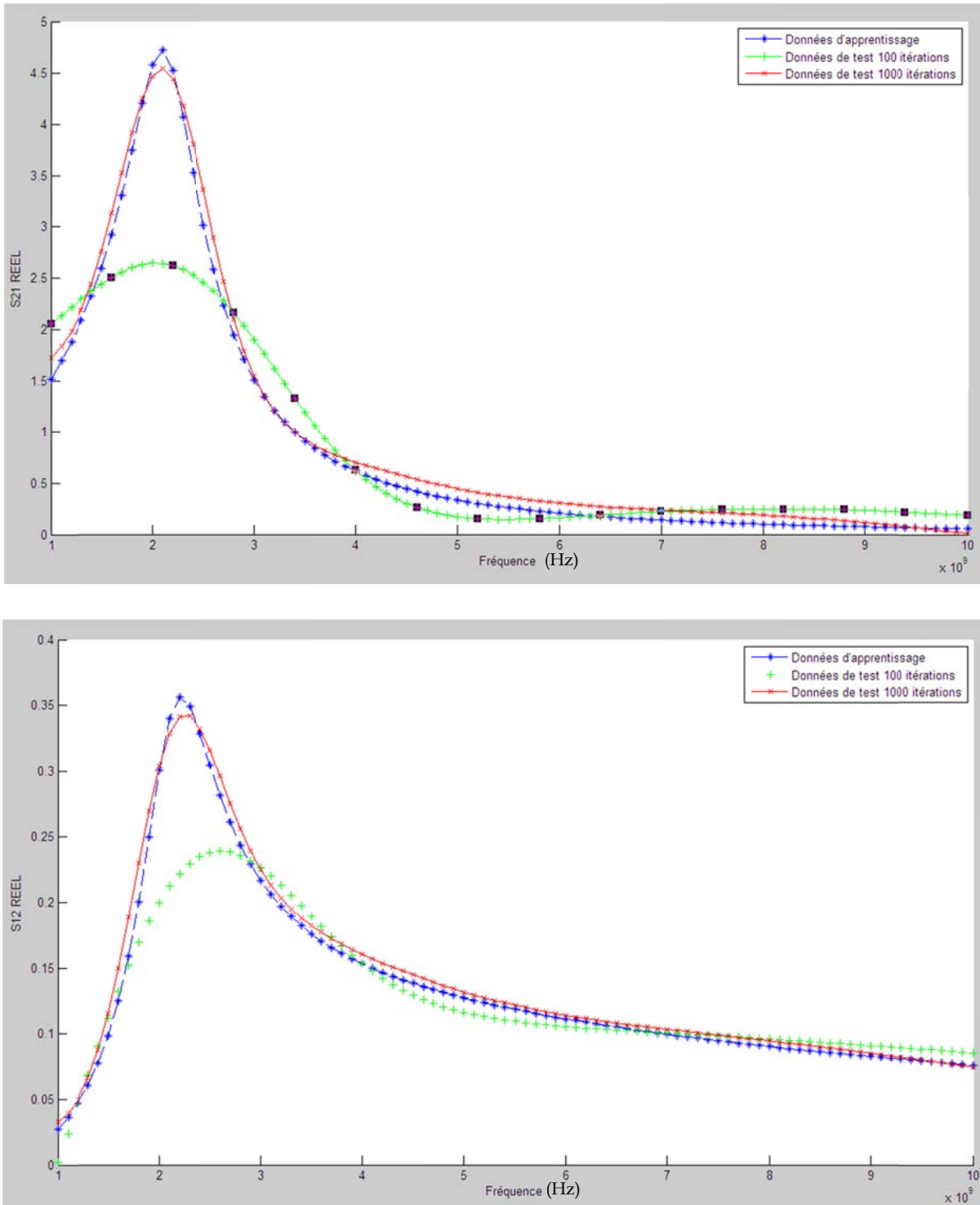


Fig. 4.13.-b. Résultats d'apprentissage pour la partie réelle des paramètres S_{21} et S_{12} d'un amplificateur de puissance avec 91 échantillons et 15 neurones, en faisant varier le nombre d'itérations (comparaison avec 37 échantillons de test).

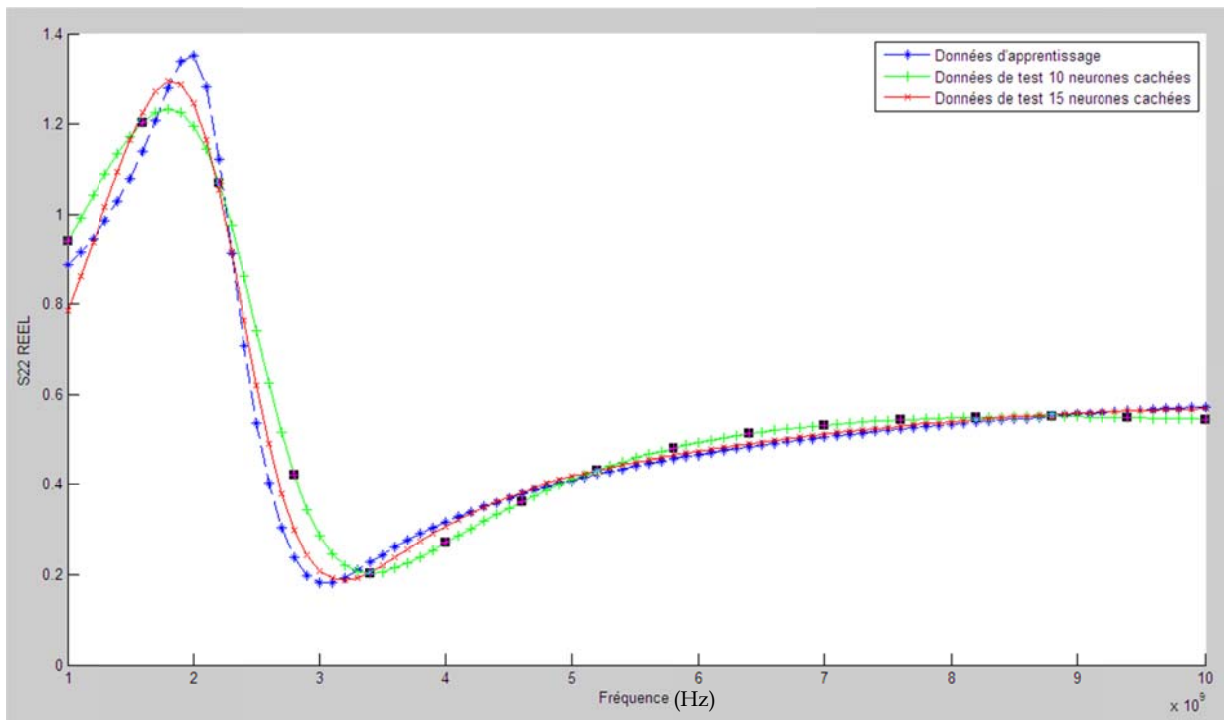
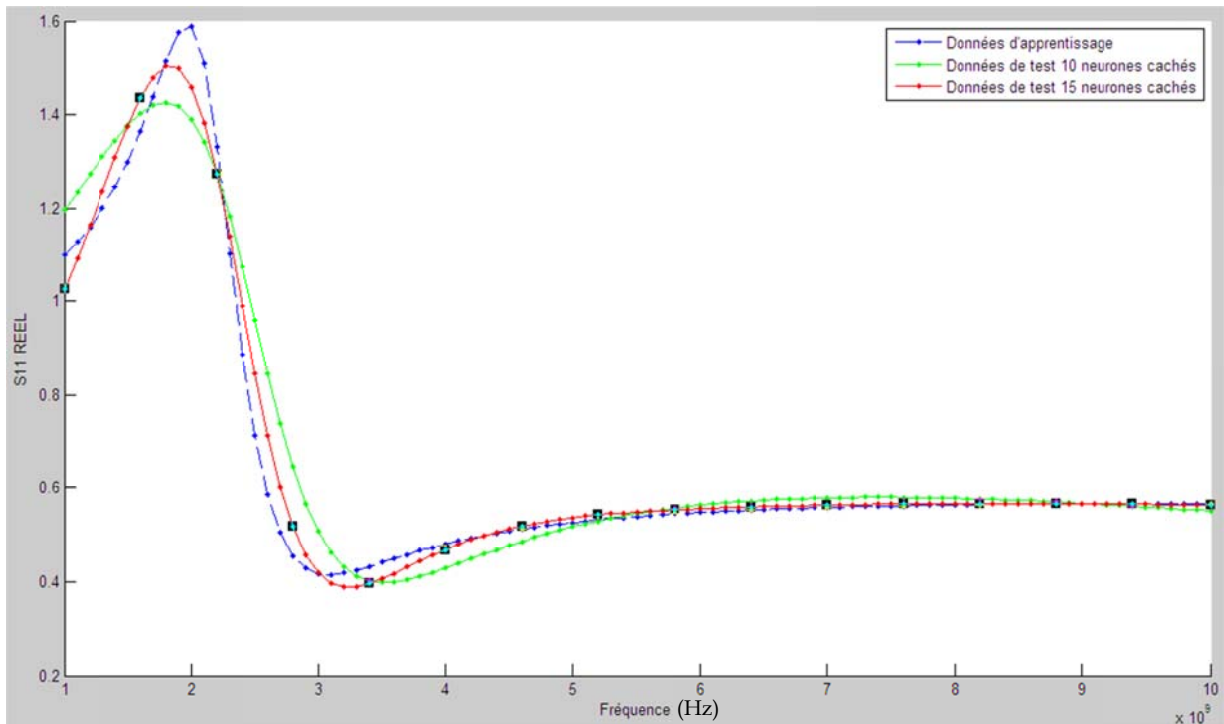


Fig. 4. 13.-c. Résultats d'apprentissage pour la partie réelle des paramètres S_{11} et S_{22} d'un amplificateur de puissance avec 91 échantillons et 1000 itérations, en faisant varier le nombre de neurones (comparaison avec 37 échantillons de test).

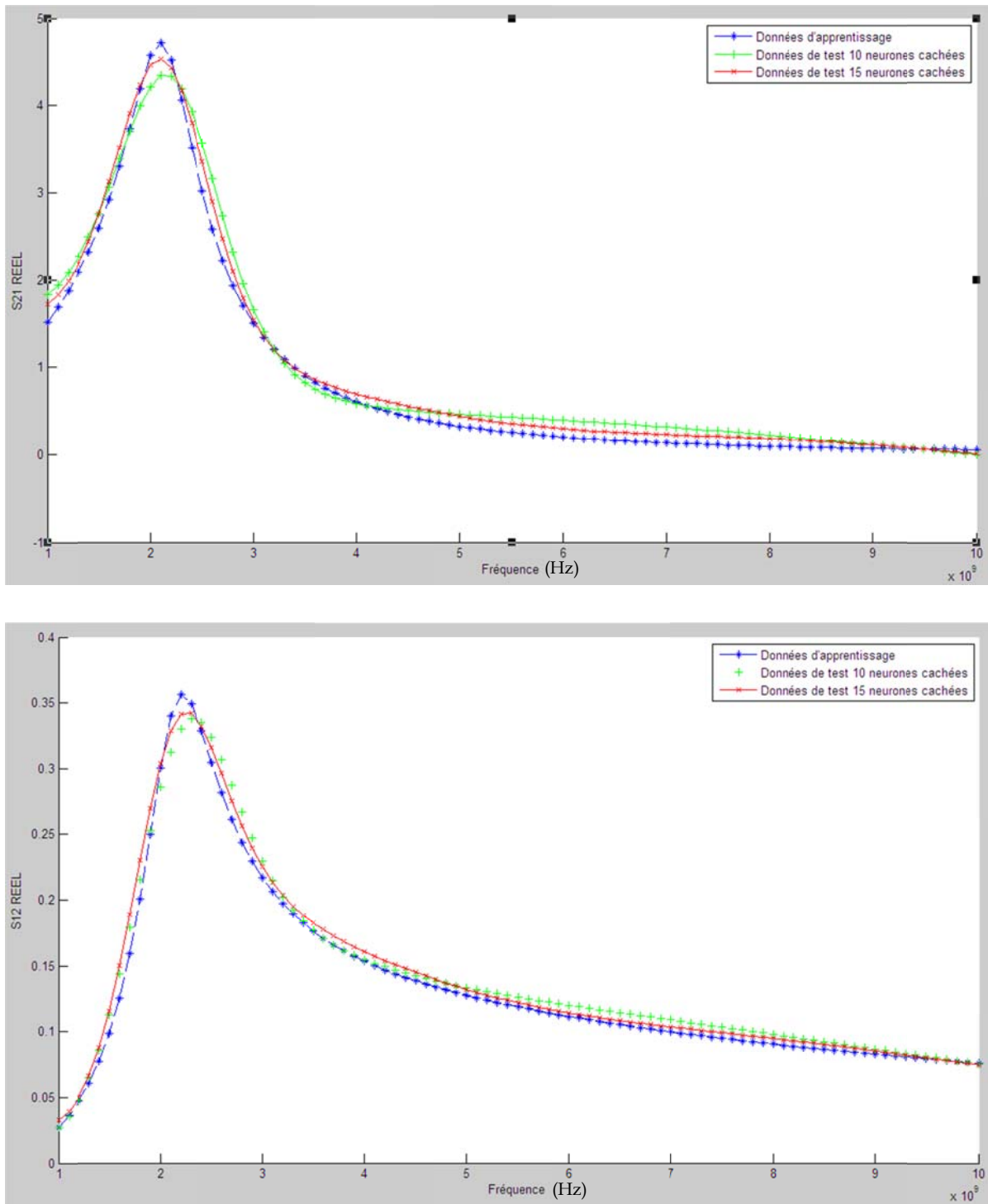


Fig. 4. 13.-d. Résultats d'apprentissage pour la partie réelle des paramètres S_{21} et S_{12} d'un amplificateur de puissance avec 91 échantillons et 1000 itérations, en faisant varier le nombre de neurones (comparaison avec 37 échantillons de test).

Le même constat est à faire pour le nombre de neurones. On pourrait aussi conclure qu'il faut augmenter encore plus le nombre de neurones pour avoir une meilleure précision que le 2.6% obtenu.

Néanmoins, en augmentant ce nombre, on constate que la précision ne change pas de manière sensible et même diminue lorsque l'on continue d'augmenter encore plus ce nombre : c'est l'effet de sur-apprentissage.

4. Conclusion

La modélisation des composants et circuits RF et microondes à l'aide des réseaux de neurones est une opération efficace et réduit considérablement le temps de calcul pour la simulation des circuits et systèmes RF et microondes complexes, mais elle a des contraintes spécifiques concernant le choix de la structure ainsi que le nombre de neurones cachés, le nombre d'entrées-sorties et le nombre de données d'apprentissage.

L'utilisation de réseaux de neurones plus évolués (de type KBNN, PKI, RBF, SOM ...) devient incontournable. C'est une perspective très intéressante dans le cadre de futurs travaux dans le domaine des réseaux de neurones.

Conclusion et perspectives

Les concepteurs de circuits micro-ondes sont souvent astreints à composer avec des modèles empiriques, approximatifs et/ou simplifiés à cause du coût associé au développement de modèles plus fiables. Les imperfections qui en résultent peuvent se situer au niveau de la précision du modèle, de son fonctionnement dans un cycle de conception (analyse, simulation et/ou optimisation) ou au niveau de la modélisation des phénomènes physiques ou électriques mis en jeu dans un dispositif micro-ondes (effets négligés, approximés, voir même purement et simplement ignorés tels que le couplage parasite, le rayonnement dû à la dispersion, ...). Afin d'y pallier tout en augmentant les performances des outils de conception assistée par ordinateur (CAO), de très nombreuses recherches ont porté sur la génération de modèles fiables et rapides.

Ces dernières années, une approche de CAO basée sur les réseaux de neurones est en train de s'imposer de plus en plus dans le domaine des micro-ondes. C'est ainsi que nous avons développé un outil de modélisation basé sur les réseaux de neurones, outil qui pourra être ultérieurement utilisé par tout chercheur potentiel. Les réseaux de neurones sont en effet une technologie nouvelle dans le domaine des radiofréquences (RF) et micro-ondes, utilisée pour la modélisation, simulation, optimisation et conception. Le développement efficace d'un modèle neuronal adéquat nécessite le bon choix des données d'apprentissage et de test, la structure de réseau de neurones approprié et l'algorithme d'apprentissage convenable. Ces aspects sont importants pour une utilisation optimale des réseaux de neurones pour résoudre les problèmes de conception micro-onde.

Les objectifs clés que nous nous sommes fixés dans ce travail sont alors :

- L'étude des différentes approches de modélisation des composants et circuits micro-ondes,
- Le développement d'un outil de gestion de réseaux de neurones ainsi que de leurs algorithmes d'apprentissage. En partant du double constat que les réseaux de neurones de type perceptron multicouche (MLP) restent les plus largement utilisés et qu'en vertu du théorème d'approximation universel les MLP à trois couches (MLP3) peuvent approximer n'importe quel système, un outil de modélisation a été développé en utilisant les MLP3.
- L'application de cet outil à la modélisation de composants et circuits micro-ondes.

Ce travail présente en effet un outil de modélisation basé sur les réseaux de neurones. Après avoir introduit quelques différents types de réseaux de neurones comme les MLP, KBNN, RBF, SOM ..., nous avons retenu les réseaux de neurones feedforward MLP qui restent les plus utilisés dans les applications RF et micro-ondes. Nous avons également discuté des algorithmes d'apprentissage (minimisation) comme la méthode de descente de gradient, la méthode de Newton, la méthode quasi-Newton,

Dans nos exemples, nous avons utilisé les réseaux MLP à trois couches en nous basant sur le théorème d'approximation universel. L'algorithme d'apprentissage de descente de gradient a été utilisé en premier, puis lorsque l'algorithme est proche de la solution, la méthode quasi-Newton a été utilisée.

Quoiqu'avec un succès largement prouvé dans plusieurs cas où d'autres approches traditionnelles ont échoué, l'utilisation des réseaux de neurones soulève encore beaucoup de questions. Ainsi, le problème du choix des paramètres du réseau de neurones comme le nombre d'échantillons de données d'apprentissage, l'initialisation des poids, la meilleure structure de réseau à utiliser, le nombre de neurones, le nombre de couches, reste entier. C'est ce que nous avons essayé de démontrer à travers nos différents exemples.

Conclusion Générale

Nous avons en effet montré les effets du nombre de données d'apprentissage, du nombre d'itérations ainsi que du nombre de neurones dans la couche cachée sur l'efficacité du réseau et ce, *pour sensibiliser les potentiels utilisateurs aux avantages certes indéniables des réseaux de neurones, mais aussi à leurs limites.*

Dans les travaux futures, des structures neuronales plus évoluées telles que les KBNN, RBF et autres SOM, devront être développées pour compenser les limites des MLP et ainsi créer des modèles de réseaux de neurones plus robustes et plus efficaces.

Bibliographie

- [1] G. Antonini, A. Orlandi, "Gradient evaluation for neural-network-based electromagnetic procedures", *IEEE Trans. Microwave Theory Tech.*, 48, pp. 874-876, 2000.
- [2] J. Bandler, M. Ismail, J. Rayas-Sanchez, Q.J. Zhang, "New directions in model development for RF/microwave components utilizing artificial neural networks and space mapping", *IEEE APS Int. Symp.*, pp. 2572-2575, 1999.
- [3] B. Chattaraj, M.C.E. Yagoub, X. Ding, Q.J. Zhang, "EM Based optimization of microwave circuits by neural models and their application to power distribution and decoupling optimization", *European Microwave Conf.*, pp. 125-128, 2001.
- [4] C. Cho, K. Gupta, "EM-ANN modeling of overlapping open-ends in multiplayer microstrip lines for design of bandpass filter", *IEEE APS Int. Symp.*, pp. 2592-2595, 1999.
- [5] E. Fernandes, P.Silva, A.Gomez, M.Barbosa, "A neural network modeling of microwave circuits on PBG structures," *IEEE MTT-S Digest*, pp. 181-184, 2003.
- [6] G.L. Creech, J.M. Zurada, "Neural network modeling of GaAs IC material and MESFET device characteristics", *Int. J. RF and Microwave CAE*, 9, pp. 241-253, 1999.
- [7] V.K. Devabhaktuni, B. Chattaraj, M.C.E. Yagoub, Q.J. Zhang, "Advanced microwave modeling framework exploiting automatic model generation, knowledge neural networks and space mapping," *IEEE MTT Int. Microwave Symp*, pp. 1098-1100, 2002.
- [8] V.K. Devabhaktuni, M.C.E. Yagoub, Y. Fang, J.J. Xu, Q.J. Zhang, "Neural networks for microwave modeling: model development issues and nonlinear techniques," *Int. J. RF Microwave CAE*, 11, pp. 4-21, 2001.

- [9] X. Ding, V.K. Devabhaktuni, B. Chattaraj, M.C.E. Yagoub, M. Deo, J.J. Xu, Q.J. Zhang, "Neural-Network approaches to electromagnetic-based modeling of passive components and their applications to high-frequency and high-speed nonlinear circuit optimization," *IEEE Trans. Microwave Theory Tech.*, 52, pp. 436-449, 2004.
- [10] J.A. Garcia, A.T. Puente, A.M. Sanchez, I. Santamaria, M. Lazaro, C.J. Pantaleon, C.J. Pedro, "Modeling MESFETs and HEMTs intermodulation distortion behavior using generalized radial basis function network," *Int. J. RF and Microwave CAE*, 9, pp. 261-276, 1999.
- [11] R.L. Mahajan, "Design and optimization through physical/neural network models," *IEEE MTT Int. Microwave Symp., Workshop on Appl. of ANN to Microwave Design*, pp. 1-16, 1997.
- [12] M. Nakhla, "Application of Neural networks to simulation and optimization of high speed interconnects," *IEEE MTT Int. Microwave Symp., Workshop on Appl. of ANN to Microwave Design*, 1997, pp. 49-66.
- [13] A. Patnaik, R.K. Mishra, "ANN techniques in microwave engineering," *IEEE Microwave Magazine*, 1, pp. 55-60, 2000.
- [14] J. Rousset, Y. Harkouss, J.M. Collantes, M. Campovecchio, "An accurate neural network model of FET intermodulation and power analysis," *European Microwave Conf.*, 1996, pp. 16-19.
- [15] K. Shirakawa, M. Shimiz, N. Okubo, Y. Daido, "A large-signal characterization of an HEMT using a multilayered network," *IEEE Trans. Microwave Theory Tech.*, 45, pp. 1630-1633, 1997.
- [16] M. Vai, S. Prasad, "Neural networks in microwave circuit design - beyond black box models," *Int. J. of RF and Microwave CAE*, 9, pp. 187-197, 1999.
- [17] P.M. Watson, K.C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Trans. Microwave Theory Tech.*, 45, pp. 2515-2523, 1997.
- [18] V.K. Devabhaktuni, C. Xi, Q.J. Zhang, "A neural network approach to the modeling of heterojunction bipolar transistors from S-parameter data," *European Microwave Conf.*, pp. 306-311, 1998.

- [19] J.J. Xu, M.C.E. Yagoub, R.T. Ding, Q.J. Zhang, "Robust neural based microwave modeling and design using advanced model extrapolation," *IEEE MTT-S Int. Microwave Symp.*, pp. 1549-1552, 2004.
- [20] J.J. Xu, M.C.E. Yagoub, R.T. Ding, Q.J. Zhang, "Exact adjoint sensitivity analysis for neural based microwave modeling and design," *IEEE Trans. Microwave Theory Tech.*, 51, pp. 26- 237, 2003.
- [21] J. Gao, L. Zhang, J. Xu, Q.J. Zhang, "Nonlinear HEMT modeling using artificial neural network technique," *IEEE MTT-S Int. Microwave Symp.*, pp. 469-472, 2005.
- [22] J. Gao, L. Zhang, J. Xu, Q.J. Zhang, "Microwave noise modeling for PHEMT using artificial neural network technique," *Microwave Conference, Asia-Pacific*, pp. 185-188, 2006
- [23] F. Wang, V.K.Devabhaktuni, Q.J. Zhang, "A Hierarchical Neural Network Approach to the Development of a Library of Neural Models for Microwave Design," *IEEE MTT Int. Microwave Symp*, pp. 2391-2403, 1998.
- [24] J.P. Garcia, F.Q. Pereira, D.C. Rebenaque, J.L.G. Tornero, A.A. Melcon, "A neural network method for the analysis of multilayered shielded microwave circuits," *IEEE Trans. Microwave Theory Tech.*, 54, pp. 309-320, 2006.
- [25] Y. Cao, L. Simonovich, Q.J. Zhang, "A broadband and parametric model of differential via holes using space-mapping neural network", *IEEE Microwave And Wireless Components Lett.*, 19, Sept. 2009
- [26] Y. Kim, S. Keely, J. Ghosh, H. Ling, "Application of artificial neural networks to broadband antenna design based on a parametric frequency model," *IEEE Trans. Antennas Propagat.*, 55, pp. 669-674, 2007.
- [27] H. Kabir, Y. Wang, M. Yu, Q.J. Zhang, "Neural network inverse modeling and applications to microwave filter design," *IEEE Trans. Microwave Theory Tech.*, 56, pp. 867-879, 2008.
- [28] B. Gao, Y. Xin, "A coplanar waveguide power divider and it's neural network model", *Int. Conf. Data Storage and Data Engineering (DSDE)*, pp. 292-295, 2010.

- [29] S. Caylar, G. Dural, K. Leblebicioglu, "Neural network method for direction of arrival estimation with uniform cylindrical microstrip patch array", *IET Microw. Antennas Propag.*, 4, pp. 153-161, 2010.
- [30] J. Meng, S. Koziel, J.W. Bandler, M.H. Bakr, Q.S. Cheng, "Tuning space mapping: A novel technique for engineering design optimization," *IEEE MTT-S Int. Microwave Symp.*, pp. 991-994, 2008.
- [31] Q.J. Zhang, K.C. Gupta, *Neural Networks for RF and Microwave Design*, Artech House, Norwood, MA, 2000.
- [32] M.A. Alsunaidi, S.M.S. Imtiaz, S.M. El-Ghazaly, "Electromagnetic wave effects on microwave transistors using a full-wave time domain model", *IEEE Trans. Microwave Theory Tech.*, 44, pp. 799-807, 1996.
- [33] S. Datta, "Electronic Transport in Mesoscopic Systems", Cambridge University Press, Cambridge, UK, 1995.
- [34] S. Selberherr, "Analysis and Simulation of Semiconductor Devices", Springer-Verlag, Wien, New York, NY, 1984.
- [35] B. Troyanovsky, Z.-P. Yu, R.W. Dutton, "Physics-based simulation of nonlinear distortion in semiconductor devices using the harmonic balance method", *Comput. Methods Appl. Mech. Eng.*, 181, pp. 467-482, 2000.
- [36] E. Ngoya, R. Larchevêque, "Envelope transient analysis: a new method for the transient and steady state analysis of microwave communication circuits and systems", *IEEE MTT-S Int. Microwave Symp.*, pp. 1365-1368, 1996.
- [37] J.W. Bandler, R.M. Biernacki, Q. Cai, S.H. Chen, S. Ye, Q.-J. Zhang, "Integrated physics-oriented statistical modeling, simulation and optimization", *IEEE Trans. Microwave Theory Tech.*, 40, pp. 1374-1400, 1992.
- [38] M.Y. Frankel, D. Pavlidis, "An analysis of the large-signal characteristics of AlGaAs7GaAs heterojunction bipolar transistor", *IEEE Trans. Microwave Theory Tech.*, 40, pp. 465-474, 1992.
- [39] H. Chehade, "Modélisation des composants microélectroniques non linéaires par séries de Volterra a noyaux dynamiques, pour la CAO des circuits RF et micro-ondes", *Thèse de doctorat*, Limoges, 1998.

- [40] V. Rizzoli, A. Neri, "State of the art and present trends in non linear microwave CAD techniques", *IEEE Trans. Circuits and Systems*, 27, pp. 161-175, 1980.
- [41] G. Leuzzi, F. Giannini, "Nonlinear Microwave Circuit Design", Wiley & Sons, 2004.
- [42] H. Zaabab, Q.J. Zhang, M.S. Nakhla, "Neural network modeling approach to circuit optimization and statistical design", *IEEE Trans Microwave Theory Tech.*, 43, pp. 1349-1358, 1995.
- [43] M. Vai, S. Wu, B. Li, S. Prasad, "Creating neural network based microwave circuit models for analysis and synthesis", *Asia Pacific Microwave Conf.*, Hong Kong, pp. 853-856, 1997.
- [44] M. Vai, S. Prasad, "Microwave circuit analysis and design by a massively distributed computing network", *IEEE Trans Microwave Theory Tech.*, 43, pp. 1087-1094, 1995.
- [45] P.M. Watson, K.C. Gupta, "EM-ANN models for microstrip vias and interconnects in dataset circuits", *IEEE Trans. Microwave Theory Tech.*, 44, pp. 2495-2503, 1996.
- [46] Q.J. Zhang, M.S. Nakhla, "Signal integrity analysis and optimization of VLSI interconnects using neural network models", *IEEE Int. Symp. Circuits Systems*, London, pp. 459-462, 1994.
- [47] P.M. Watson, K.C. Gupta, R.L. Mahajan, "Development of knowledge based artificial neural network models for microwave components", *IEEE MTT-S Int. Microwave Symp.*, pp. 9-12, 1998.
- [48] G.L. Creech, B.J. Paul, C.D. Lesniak, T.J. Jenkins, M.C. Calcaterra, "Artificial neural networks for fast and accurate EM-CAD of microwave circuits", *IEEE Trans Microwave Theory Tech.*, 45, pp. 794-802, 1997.
- [49] V.B. Litovski, J.I. Radjenovic, Z.M. Mrcarica, S.L. Milenkovic, "MOS transistor modeling using neural network", *Electron Lett.*, 28, pp. 1766-1768, 1992.
- [50] A.H. Zaabab, Q.J. Zhang, M.S. Nakhla, "Analysis and optimization of microwave circuits and devices using neural network models", *IEEE MTT-S Int. Microwave Symp.*, pp. 393-396, 1994.

- [51] G. Kothapalli, "Artificial neural networks as aids in circuit design", *Microelectron J.*, 26, pp. 569-578, 1995.
- [52] A. Veluswami, M.S. Nakhla, Q.J. Zhang, "The application of neural networks to EM based simulation and optimization of interconnects in high speed VLSI circuits", *IEEE Trans. Microwave Theory Tech.*, 45, pp. 712-723, 1997.
- [53] K.L. Moor, "artificial neural network," *IEEE potentials*, 11, pp. 23-28, 1992.
- [54] P.R. Reddy, M. Chakraborty, "Prediction of grain size of Al-Si Alloy by neural networks", *Materials Science and Engineering*, 391, pp. 131-140, 2005.
- [55] H. Simon, *Neural Networks*, Macmillan College Publishing Co., New Jersey, 1994.
- [56] S. Haykin, *Neural Network: A Comprehensive Foundation*, Macmillan College Publishing Co., New York, 1994.
- [57] W.S. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, 5, pp. 115-133, 1943.
- [58] D.O. Hebb, *The Organization of Behavior*, Wiley, New York, 1949.
- [59] M. Minsky, "Neural nets and the brain model problem", *Ph.D. dissertation*, Princeton University, 1954.
- [60] M. Minsky "Steps toward artificial intelligence," *Computers and Thought*, pp. 406-450, 1961.
- [61] D. Gabor, "Communication theory and cybernetics", *IRE Trans. Circuit Theory*, 1, pp. 19-31, 1954.
- [62] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review*, 65, pp. 386-408, 1958.
- [63] B. Widrow, M.E. Hoff, Jr., "Adaptive switching circuits", *IRE Wescon Convention Record*, pp. 96-104, 1960.
- [64] S. Grossberg, "Intracellular mechanisms of adaptation and self-regulation in self-organizing networks: The role of chemical transducers," *Bulletin of Mathematical Biology*, 42, pp. 365-396, 1980.
- [65] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *National Academy of Sciences of the USA*, 9, pp. 2554, 1982.

- [66] T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, New York, 1984.
- [67] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, "Optimization by Simulated Annealing", *Cognitive Science*, 220, pp. 671-680, 1983.
- [68] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, "Learning algorithm for Boltzmann machines", *Cognitive Science*, 9, pp. 147-169, 1985.
- [69] G.E. Hinton, T.J. Sejnowski, "Learning and relearning in Boltzmann machines J. McClelland, D. Rumelhart, Eds., in *Explorations in the Microstructure of Cognition 1: Foundations*, MIT Press, Cambridge, pp. 282-317, 1986.
- [70] P. Watson, G. Creech, K. Gupta, "Knowledge based EM-ANN models for the design of wide bandwidth CPW patch/slot antennas," *IEEE APS Int. Symp.*, Orlando, FL, pp. 2588-2591, 1999.
- [71] D.S. Broomhead, D. Lowe, "Multi-variable functional interpolation and adaptive networks", *Complex Syst.*, 2, pp. 327-355, 1988.
- [72] G.-B. Huang, Y.-Q. Chen, H.A. Babri, "Classification ability of single hidden layer feedforward neural networks", *IEEE Trans. Neural Networks*, pp. 799-801, 2000.
- [73] L. Ton, "Automatic EM Based Modeling of Embedded Passive by Neural Network," *Master of Applied Science*, Carleton U., 2004.
- [74] C. Cybenko, "Approximation by superposition of a sigmoid function", *Math. Control Signals Syst.*, 2, pp. 303-314, 1989.
- [75] K. Hornik, M. Stinchcombe, H. White, "Multilayer feedforward networks are universal approximators", *IEEE Trans. Neural Networks*, 2, pp. 359-366, 1989.
- [76] J. de Villiers, E. Barnard, "Backpropagation neural nets with one and two hidden layers", *IEEE Trans. Neural Networks*, 4, pp. 136-141, 1992.
- [77] S. Tamura, M. Tateishi, "Capabilities of a four-layered feedforward neural network: Four layer versus three", *IEEE Trans. Neural Networks*, 8, pp. 251-255, 1997.
- [78] M.J.D. Powell, "Radial Basis Functions for Multi-variable Interpolation: A Review," *IMA Conf. Algorithms for the Approximation of Functions and Data*, RMCS Shrivenham, UK, 1985.

- [79] S. Chen, A. Billings, C.F.N. Cowan, P.M. Grant "Practical identification of NARMAX models using radial basis functions." *Int. Journal of Control*, 52, 6, pp. 1327-1350, 1990.
- [80] S. Elanayar, Y.C. Shin, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic system," *IEEE Trans. Neural Networks*, 5, pp. 594-603, 1994.
- [81] L. Behara, M. Gopal, S. Chaudhury, "Inversion of RBF networks and applications to adaptive control of systems," *IEE Proc., Control Theory Appl.*, 142, pp. 617-624, 1995.
- [82] G. Dreyfus, J.-M. Martinez, M. Samuelides, M.B. Gordon, F. Badran, S. Thiria, L. Herault, *Réseaux de neurones*, Eyrolles, Paris, 2002.
- [83] Y. Bennani, *Apprentissage connexionniste*, Lavoisier, 2006.
- [84] M. El Zoghbi, "Analyse électromagnétique et outils de modélisation couplés. Application à la conception hybride de composants et modules hyperfréquences", *Thèse de doctorat*, Limoges, 2008.
- [85] R. Haque, "Transmission loss allocation using artificial neural networks", *Master of Science Thesis*, University of Saskatchewan, 2006.
- [86] M. Stricker, "Réseaux de neurones pour le traitement automatique du langage : conception et réalisation de filtres d'informations", *Thèse de Doctorat*, Université Pierre et Marie Curie - Paris VI, 2000.
- [87] M. Minoux, *Programmation Mathématique, Théorie et Algorithmes*, Dunod, Tome 1, 1983.
- [88] C.G. Broyden, "The convergence of a class of double-rank minimization algorithms 2: the new algorithm," *J. Institute of Mathematics and its Applications*, 6, pp. 222-231, 1970.
- [89] R. Fletcher. "A New Approach to Variable Metric Algorithms," *The Computer J.*, 13, pp. 317-322, 1970.
- [90] D. Goldfarb, "A family of variable metric methods derived by variational means," *Mathematics of Computation*, 24, pp. 23-26, 1970.
- [91] D.F. Shanno, "Conjugate gradient methods with inexact searches," *Mathematics of operations Research*, 3, pp. 244-256, 1978.

- [92] C.M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.
- [93] M.D. Richard, R. P. Lippmann. "Neural Network classifiers estimate Bayesian a-posteriori probabilities," *Neural Computation*, 3, pp. 461-483, 1991.
- [94] S. Geman, E. Bienenstock, R. Doursat. "Neural network and the bias/variance dilemma," *Neural Computation*, 4, pp. 1-58, 1992.
- [95] P. Gallinari, T. Cibas. "Practical complexity control in multilayer perceptrons". *Signal Processing*, 74, pp. 29-46, 1999.
- [96] ADS v.2004, *Agilent Technologies*, Palo Alto, CA, USA.
- [97] Ansoft HFSS v.8.0, 2002, *Ansoft Corporation*, Pittsburg, PA, USA.
- [98] S.R. Kythakyapuzha, W.B. Kuhn, "Modeling of inductors and transformers", *IEEE MTT Int. Symp.*, pp. 587-590, 2001.
- [99] S.S. Mohan, M. Hershenson, S.P. Boyd, T.H. Lee, "Simple accurate expressions for planar spiral inductances," *IEEE J. of Solid-State Circuits*, 34, pp. 1419-1424, 1999.
- [100] M. Yamaguchi, M. Baba, K.I. Arai, "Sandwich type ferromagnetic RF integrated inductor," *IEEE MTT Int. Microwave Symp.*, pp. 185-188, 2001.
- [101] E. Yamashita, R. Mittra, "Variational method for the analysis of microstrip lines", *IEEE Trans. Microwave Theory Tech.*, 16, pp. 251-256, 1968.
- [102] E. Martini, G. Pelosi, S. Selleri, "A hybrid finite-element-modal-expansion method with a new type of curvilinear mapping for the analysis of microwave passive devices," *IEEE Trans. Microwave Theory Tech.*, 51, pp. 1712-1717, 2003.

" DÉVELOPPEMENT D'UN OUTIL DE CAO POUR MODÉLISATION PAR RÉSEAUX DE NEURONES " *

Par

Mr FARES SELMANE

Laboratoire d'Instrumentation.

Faculté d'Electronique et d'Informatique, U.S.T.H.B

Résumé :

Dans ce siècle de l'innovation, la conception assistée par ordinateur (CAO) est devenue une étape incontournable du processus de mise sur le marché de nouveaux produits avec l'objectif de produire le plus rapidement possible des dispositifs de plus en plus performants avec des rapports qualité/prix toujours plus compétitifs. Cependant, si tout logiciel de CAO doit reposer avant tout sur des modèles rapides et fiables, les modèles existants dans les simulateurs commerciaux ne répondent que partiellement à toutes les attentes des concepteurs. De ce fait, le besoin en modèles plus performants, notamment du point de vue intégrabilité, rapidité de réponse, fiabilité et fonctionnalité, est une réalité incontournable particulièrement en haute fréquence avec l'engouement sans cesse croissant du public pour des innovations technologiques toujours plus élaborées.

Les concepteurs de systèmes micro-ondes sont souvent astreints à composer avec des modèles empiriques, approximatifs et/ou simplifiés à cause du coût associé au développement de modèles plus fiables. Les imperfections qui en résultent peuvent se situer au niveau de la précision du modèle, de son fonctionnement dans un cycle de conception (analyse, simulation et/ou optimisation) ou au niveau de la modélisation des phénomènes physiques ou électriques mis en jeu dans un dispositif micro-ondes (effets négligés, approximés, voir même ignorés tels que couplage parasite, rayonnement, dispersion, ...). Afin d'y pallier tout en augmentant les performances des outils de conception assistée par ordinateur (CAO), de nombreuses recherches ont porté sur la génération de modèles fiables et rapides.

Ces dernières années, une approche de CAO basée sur les réseaux de neurones est en train de s'imposer de plus en plus dans le domaine des micro-ondes. C'est ainsi que nous avons développé un outil de modélisation basé sur les réseaux de neurones, outil qui pourra être ultérieurement utilisé par tout chercheur potentiel. Les objectifs clés de cette thèse sont :

- L'étude des différentes approches de modélisation des composants et circuits micro-ondes,
- Le développement d'un outil de gestion de réseaux de neurones ainsi que de leurs algorithmes d'apprentissage. En partant du double constat que les réseaux de neurones de type perceptron multicouche (MLP) restent les plus largement utilisés et qu'en vertu du théorème d'approximation universel, les MLP à trois couches (MLP3) peuvent approximer n'importe quel système, un outil de modélisation a été développé en utilisant les MLP3.
- L'application de cet outil à la modélisation de composants et circuits micro-ondes.

Quoiqu'avec un succès largement prouvé dans plusieurs cas où d'autres approches traditionnelles ont échoué, l'utilisation des réseaux de neurones soulève encore beaucoup de questions. Ainsi, le problème du choix des paramètres du réseau de neurones comme le nombre d'échantillons de données d'apprentissage, l'initialisation des poids, la meilleure structure de réseau à utiliser, le nombre de neurones, le nombre de couches, ..., reste entier. C'est ce que nous avons essayé de démontrer dans nos différents exemples.

* Thèse de Magister

Directeur de thèse: Mustapha C.E Yagoub, Professeur, Université d'Ottawa

Mots-clés: CAO, circuits micro-ondes, MLP, modélisation, réseaux de neurones.