

## INTRODUCTION

Afin de diagnostiquer l'état de surface de l'isolateur F160D/146DC pollué sous tension alternative des essais ont été effectuée au laboratoire de Haute Tension de l'Université de Tiaret par A. Rahouani et A. SMAIL [42]. Ce type d'isolateur a été choisi car il est utilisé en Algérie par la SONELGAZ pour les lignes de haute tension (400 kV) entre Bechar et Naama.

Dans ce chapitre , nous présentons au premier lieu les résultats expérimentaux de la tension de contournement et de courant de fuite en fonction de largeur des bandes propres et en fonction de type de pollution[42] ensuite nous utiliserons l'une des techniques d'intelligence artificielle qui est le réseau neurones pour la prédiction de contournement d'un isolateur de haute tension.

### III.1. Résultats expérimentaux

#### III.1.1. Mesure de la tension de contournement pour différentes largeurs des bandes propres

Le tableau III.1.montre les résultats de la tension de contournement mesurée pour différentes largeurs des bandes propres et une conductivité de 1mS/cm.

**Tableau III.1.** Mesures de la tension de contournement pour différentes largeurs des bandes propres

N° d'essai		Essaie N° 1	Essaie N° 1	Essaie N° 1	Moyenne
Uc (KV)	1cm×3	124	112	110	115.33
	3cm×3	128	124	132	128
	5cm×3	114	108	116	112.66

#### III.1.2. Mesure de la tension de contournement pour différents types de pollution

Le tableau III.2. montre les résultats de la tension de contournement mesurée pour différentes types de pollution et une largeur des bandes propre de 1cm.

**Tableau III.2.** Mesure de la tension de contournement pour différents types de pollution.

N° d'essai		Essaie N° 1	Essaie N° 1	Essaie N° 1	Moyenne
Uc (KV)	Sable des dunes	126	134	140	133.3
	Sable de sol	124	130	126	126.66
	Sable de mer	61	55	60	58.66

### III.1.3. Mesure de courant de fuite en fonction de la largeur des bandes propres

Les tableaux III.3, III.4 et III.5, montrent les courants de fuites mesurés pour différentes tensions appliquées et une conductivité de 1mS/cm.

**Tableau III.3.** Evolution du courant de fuite en fonction de la tension appliquée (3 bandes propres de 1cm).

$U_{APP}$ (KV)	1	2	3	4	$I_{moyenne}(\mu A)$
10	48	38	45.1	64.2	48.825
20	100	129	108	121	114.5
30	235	174	214	207	207.5
40	382	359	383	357	370.25
50	1190	1150	1150	1220	117.75

**Tableau III.4.** Evolution du courant de fuite en fonction de la tension appliquée (3 bandes propres de 3cm).

$U_{APP}$ (KV)	1	2	3	4	$I_{moyennes} (\mu A)$
10	45.3	43.2	41.1	41.7	42.825
20	64.6	64.6	72.8	64.2	66.55
30	82	89	84	89.7	86.175
40	112	118	118	115	115.75
50	145	144	142	142	143.25

**Tableau III.5.** Evolution du courant de fuite en fonction de la tension appliquée (3 bandes propres de 5cm)

$U_{APP}$ (KV)	1	2	3	4	$I_{moyennes} (\mu A)$
10	28	33.5	29	33.6	31.025
20	56.1	51.3	53.9	62	55.825
30	115	119	108	119	115.25
40	147	148	155	140	147.5
50	221	226	221	208	219

### III.2. Prédiction de contournement par réseaux de neurones

Dans cette partie de notre étude nous utilisons le NNTOOL (Neural Network Toolbox) pour prédire le contournement d'un isolateur de haute tension soumis sous les contraintes de différents types de pollutions et la variation de la largeur des bandes propres.

#### III.2.1. Boîte à outils du réseau de neurones

Neural Network Toolbox (Boîte à outils du réseau neurones) fournit des algorithmes, des modèles prédéfinis et des applications pour créer, former, visualiser et simuler des réseaux de neurones superficiels et profonds. Vous pouvez effectuer une classification, une régression, un regroupement, une réduction de la dimensionnalité, une prévision chronologique, ainsi que la modélisation et le contrôle dynamiques de systèmes. [43]

La prédiction des tensions de contournements des isolateurs par RNA pour différentes valeurs de conductivités de sable et différents bandes propres et la prédiction de courant de fuites pour différentes valeurs de tension appliquée et différents bandes propres est l'objectif de notre travail.

Pour améliorer les performances de cette prédiction la base de l'apprentissage doit couvrir un large domaine, d'où la nécessité d'avoir une base de données plus étendue. Les tableaux précédents de ce chapitre représentent les valeurs de tensions de contournement expérimentales pour différentes conductivités de sable et différentes largeurs des bandes propres, représentent aussi les valeurs de courants de fuites pour différentes tension et différentes largeurs des bandes propres. Ces valeurs ont été utilisée pour collecter la base des données utilisée pour l'apprentissage de notre RNA.

Plusieurs paramètres peuvent influés sur la tension de contournement. Dans notre étude nous intéresserons qu'aux deux : conductivités et largeur des bandes propres .Pour le courant de fuites nous avons des paramètres de la tension et largeur des band propres.

La structure générale du réseau de neurones prend la forme représentée sur la figure(III.1). Comme nous avons expliqué, pour la première simulation de réseau de neurones a comme input la les valeurs de tension et la bande propre et comme output les courantes de fuites. Et pour la deuxième simulation de réseau de neurones a comme input la les conductivités et la band propres et comme output les tensions de contournement.

Il est très important de signaler que la base de données utilisée dans notre travail que ce soit pour l'apprentissage ou pour la validation a été collecté de la littérature [42]

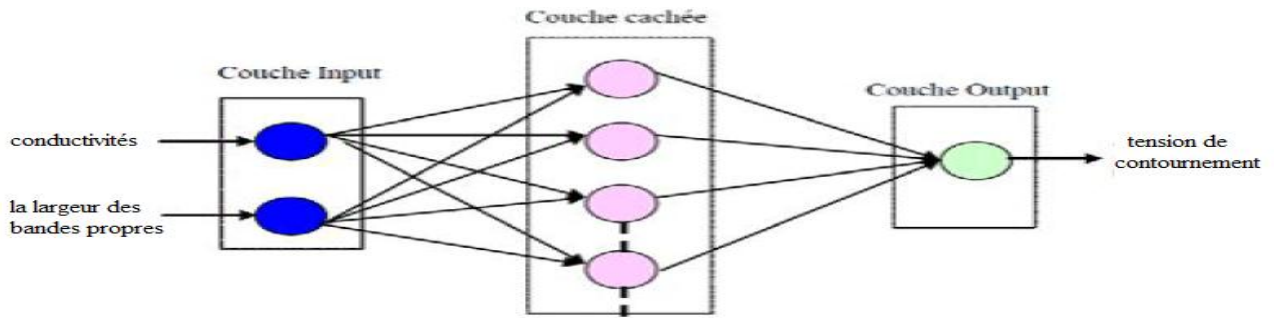


Figure III.1. Structure du réseau de neurones utilisé

### III.2.2. Processus de réseaux de neurones

Le processus d'entraînement de notre système de RNA consiste en plusieurs étapes (Figure III.1) qui peuvent être résumées comme suit :

- Le choix des variables d'entrée et de sortie basé sur la banque des données collecté de la littérature[42].
- La détermination du nombre de couches cachées, le nombre de neurones dans chaque couche cachée et le nombre d'itérations nécessaire pour atteindre une solution stable.
- Architecture du réseau : Dans notre cas d'étude, l'architecture la plus convenable est la structure du type Perceptron multicouches PML
- L'apprentissage du système de RNA basé sur la rétro-propagation "Feed-Forward Backprop" de l'erreur dans les poids synaptiques.

Le type d'apprentissage utilisé dans notre application est l'apprentissage supervisé.

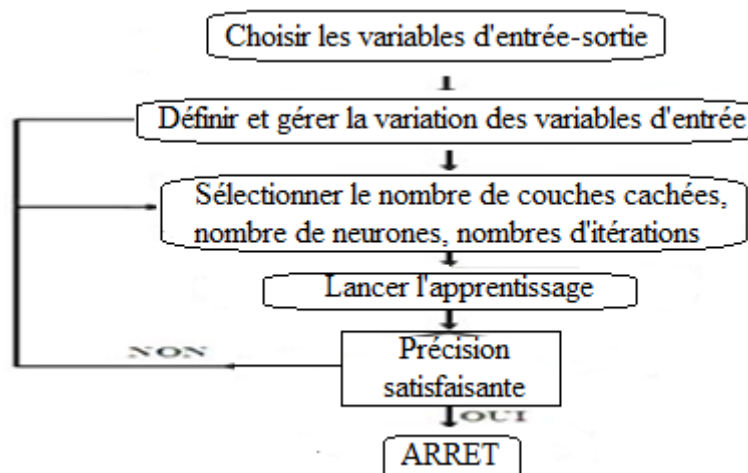


Figure III.2. Schéma de la procédure d'apprentissage de notre RNA [44] .

### III.2.2.1. Implémentation du modèle RNA

La boîte à outils utilisé Pour la simulation de RNA est « NNTOOL » sous Le logiciel Matlab.

### III.2.2.2.Choix des paramètres du réseau

Le choix des paramètres d'un réseau de neurones dépend principalement du problème à résoudre.

#### - Input data

Introduire les entrées (vecteur de deux lignes qui contient les tension et la largeur des bandes propres)( vecteur de deux lignes qui contient les conductivité et la largeur des bandes propres).

#### - Target data

Introduire la sortie qui représente vecteur de courant de fuites ou vecteur de tension de contournement.

### III.2.2.3. Architecteur

Le réseau utilisé est du type direct (propagation de l'information en avant sans liaison directes entre la sortie et l'entrée)

### III.2.2.4. fonction d'activation

La fonction de transfert de la couche cachée est la sigmoïde(logsig), puisque la rétro propagation nécessite des fonctions d'activation dérivables. Cette fonction est définie par [45]:

$$F(x) = \frac{1}{1+e^{-x}} \quad \text{III.1}$$

Fonction de transfert de la couche de sortie est linéaire(purelin), car la tension de contournement a des valeurs élevée, elle est définie par [45]:

$$F(x) = x \quad \text{III.2}$$

### III.2.2.5. Fonction d'apprentissage

Représente l'algorithme d'apprentissage, NNTOOL offre les algorithmes suivant:

- Trainlm : Apprentissage par l'algorithme Levenberg- Marquardt.
- Trainrp : Apprentissage par l'algorithme résilient propagation.
- Trainscg : Apprentissage par le gradient conjugué (scg).

Nous utiliserons Trainlm

### III.2.2.6. Adaption learning fonction

L'algorithme d'apprentissage incrémental, il peut être:

- Learngd : l'algorithme d'apprentissage est la descente de gradient avec Pas d'apprentissage fixe.
- Learngdm : l'algorithme d'apprentissage est la descente de gradient avec pas d'apprentissage et momentum fixes.

Nous appliquerons l'algorithme d'apprentissage incrémental du type Learngdm.

### III.2.2.7. Fonction de Performance

Fonction de performance peut être :

- MSE : Erreur quadratique moyenne.
- MSERG : Erreur quadratique moyenne avec pondération des poids.
- SSE : somme des carrés des erreurs.

Nous utiliserons l'erreur quadratique moyenne MSE.

Les figures III.3 et III.4 montrent la page et les différents paramètres de notre réseau de neurones créé sous NNTOOL.

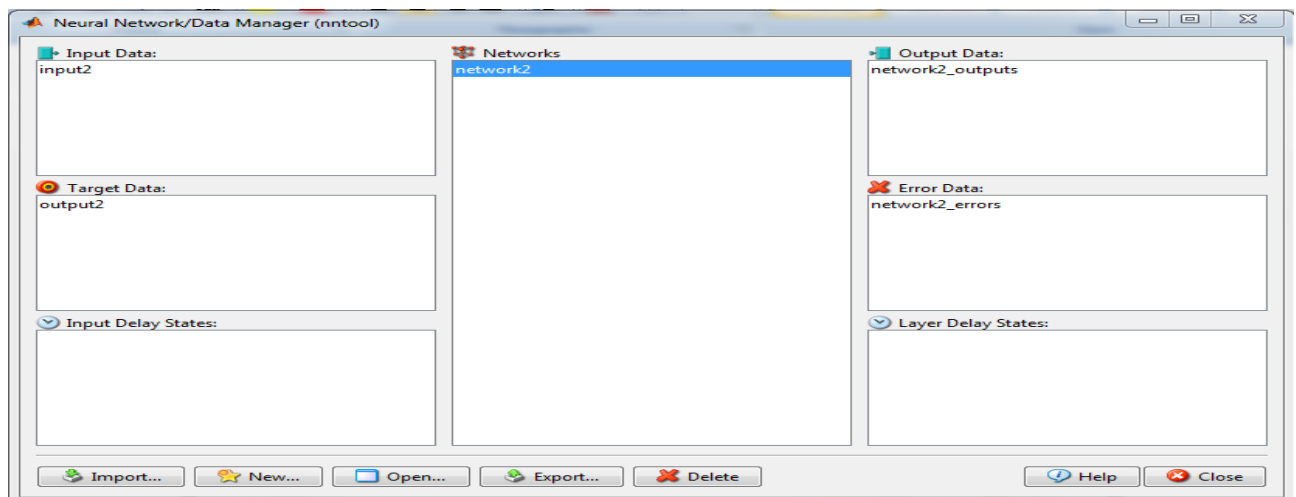


Figure III.3 Page de NNTOOL

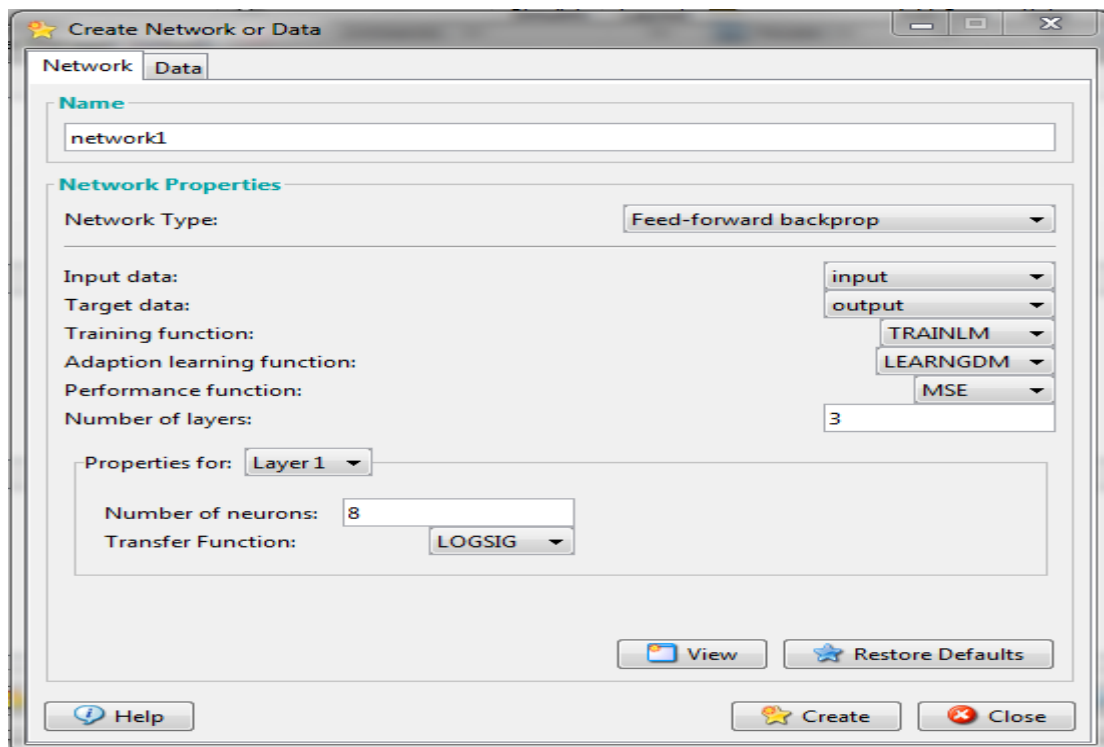


Figure.III.4. Paramètres de RNA introduits sous NNTOOL

### III.2.3. choix de nombre de couches cachées et nombre de neurones par couche.

Après la création du réseau en utilisant NNTOOL sous Matlab, il nous reste de définir le nombre de couches cachées et le nombre de neurone par couche, nous avons fait une série de tests pour le nombre de neurones cachés variant entre 2 et 10 neurones, pour des couches cachées variant entre 2 à 3 qui sont régité par des itérations de 1000 comme l'indique le tableau III.6.

La variation du nombre de neurones pour le cas de deux couches, où on remarque que l'erreur diminue en augmentant le nombre de neurones. Pour le cas de trois couches on voit une diminution significative de l'erreur entre 6 et 8 neurones.

En analysant ces résultats, alors notre réseau a la configuration suivante :

- Nombre de couche (3 couches).
- Nombre de neurone de couche (8 neurones).
- Nombre d'itération : 1000
- Erreur désirée : 0.0001

**Tableau III.6** : Détermination du nombre de couches cachée et neurone par couche

Nombre de couche	Nombre de neurone	erreur
2	2	$10^5$
	4	$10^4$
	6	$10^3$
	8	$10^3$
	10	$10^2$
3	2	$10^{-2}$
	4	$10^{-2}$
	6	$10^{-3}$
	8	$10^{-5}$
	10	$10^{-4}$

Les figures III.5 et III.6 montrent l'architecture du RNA et sa fenêtre d'apprentissage pour trois couches

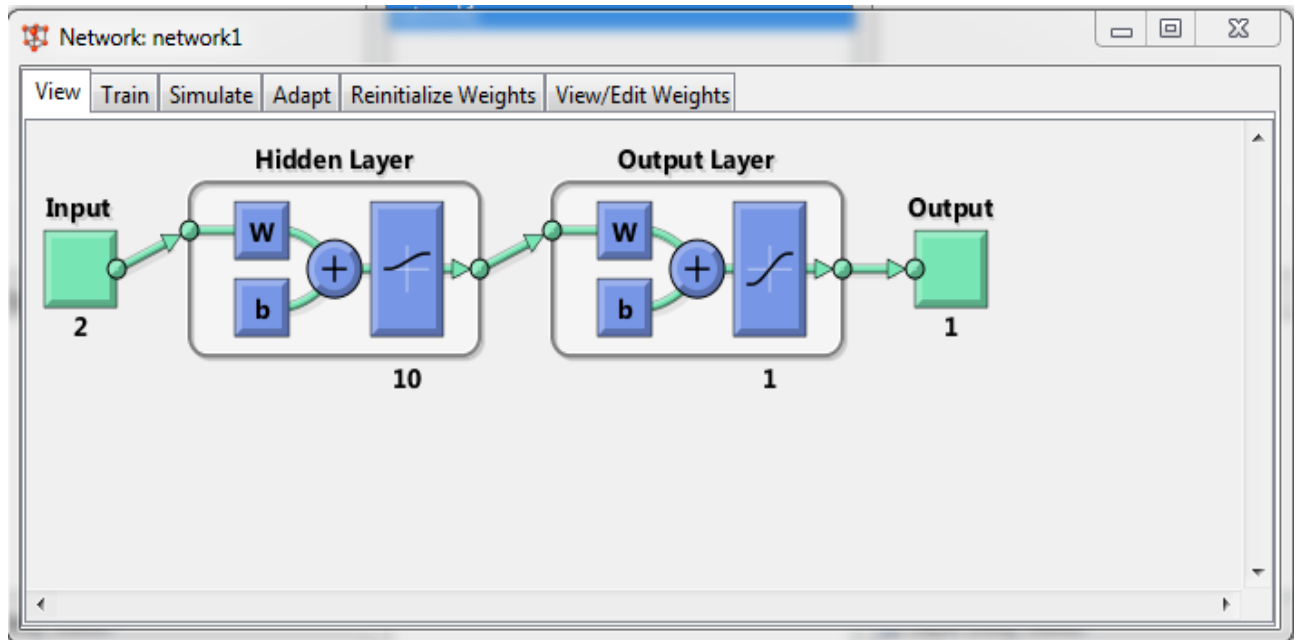


Figure.III.5.Architecture du réseau de neurones.

The screenshot shows the 'Neural Network Training (nntool)' window with the following details:

- Neural Network:** A diagram of the network architecture (2 input, 10 hidden, 1 output).
- Algorithms:**
  - Data Division: Random (dividerand)
  - Training: Levenberg-Marquardt (trainlm)
  - Performance: Mean Squared Error (mse)
  - Calculations: MEX
- Progress:**

Epoch:	0	13 iterations	1000
Time:		0:00:02	
Performance:	3.53e+04	1.62e+03	0.00
Gradient:	5.38e+04	2.50e+03	1.00e-07
Mu:	0.00100	100	1.00e+10
Validation Checks:	0	6	6
- Plots:**
  - Performance (plotperform)
  - Training State (plottrainstate)
  - Regression (plotregression)
- Plot Interval:** 1 epochs
- Status:** Validation stop. (indicated by a green checkmark)
- Buttons:** Stop Training, Cancel

Figure.III.6 Fenêtre d'apprentissage de réseau de neurones.



### III.3. Graphe de l'erreur

Après la création du réseau. L'entraînement du RNA plusieurs fois donne la caractéristique de performance suivante :

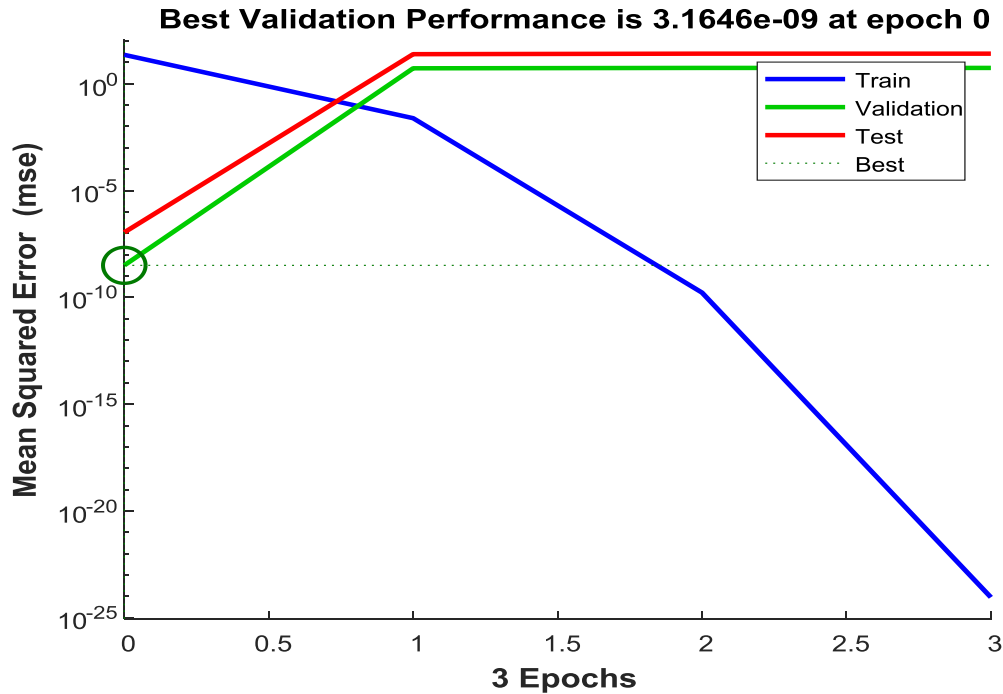


Figure.III.7 Graphe de l'erreur (d'apprentissage de test et de validation) de la 1<sup>ère</sup> simulation

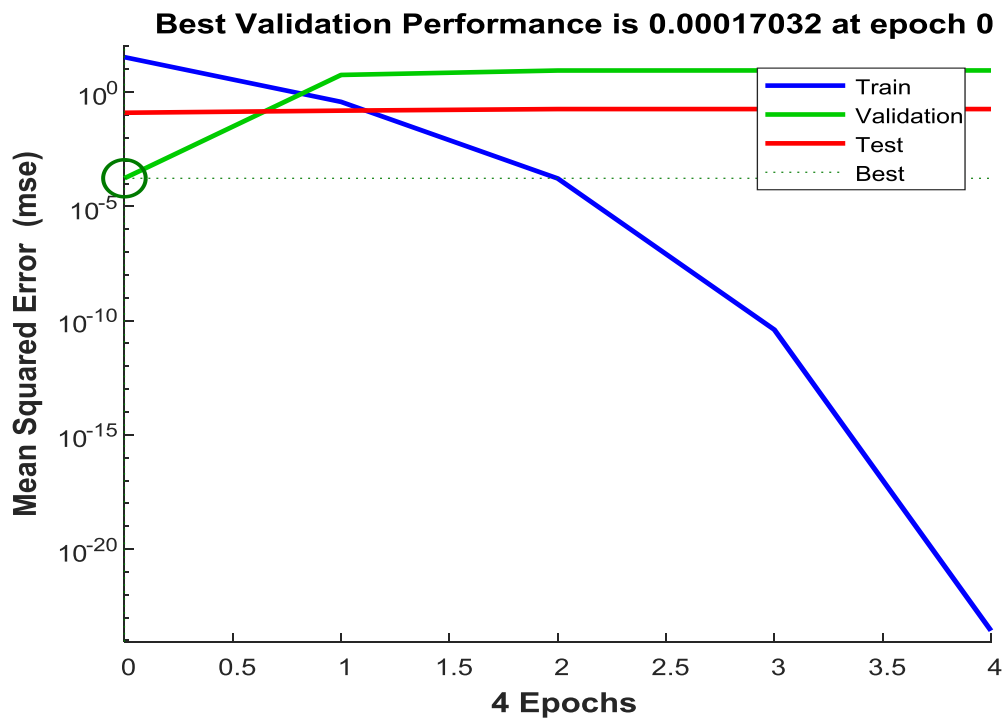


Figure.III.8 Graphe de l'erreur (d'apprentissage de test et de validation) de la 2<sup>ème</sup> simulation

Le réseau calcul les valeurs de la couche de sortie en fonction des vecteurs d'entrée. L'erreur est la différence entre la valeur de sortie calculée par le réseau de neurones et la valeur désirée. À l'itération suivante, l'algorithme ajuste les poids synaptiques dans le but de converger vers les bonnes valeurs.

L'algorithme s'arrête si l'erreur calculée est inférieure à l'erreur établie par l'utilisateur. Dans les tests de la 1<sup>ère</sup> simulation la condition d'arrêt était définie par  $\text{Erreur} < 10^{-5}$ . D'après la figure.III.7, on constate que le nombre d'itérations pour que l'apprentissage satisfasse la condition d'arrêt était l'itération 3

Pour les tests de dixième simulation la condition d'arrêt était définie par  $\text{Erreur} < 10^{-3}$ . D'après la figure.III.8 on constate que le nombre d'itérations pour que l'apprentissage satisfasse la condition d'arrêt était l'itération 4.

### **III.4. Courbes de régression**

Les courbes de régression (figure.III.9 et figure.III.10 ) permettent de valider la performance du réseau construit. Les valeurs en ordonnées représentent les sorties du réseau de neurones et les valeurs en abscisses représentent les valeurs désirées.

- Les cercles en noir représentent les valeurs désirées.
- Les droites continues représentent l'approximation faite par le réseau.
- Les droites en pointillé représentent la parfaite approximation.

#### **III.4.1. Première simulation**

Pour la première simulation, nous constatons que :

- Les données de l'apprentissage sont classées à 100%.
- Les données de validation sont classées à 99%.
- Les données de test sont classées à 100%.
- Globalement les données sont classées à 99%.

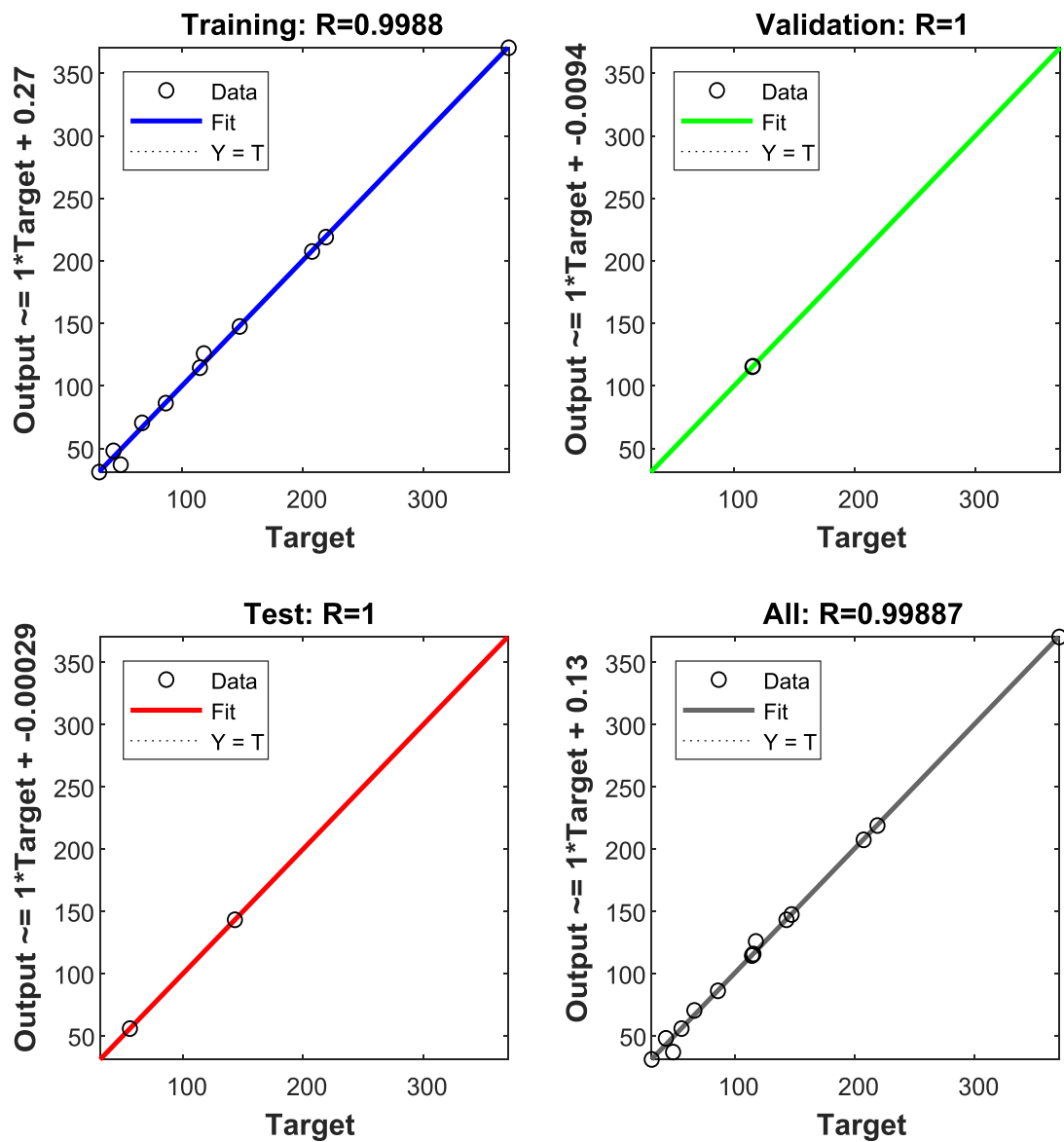


Figure.III.9 . Courbes de régression de la 1<sup>ère</sup> simulation

### III.4.2. Deuxième simulation

Pour la deuxième simulation, nous constatons que :

- Les données de l'apprentissage sont classées à 99%.
- Les données de validation sont classées à 100%.
- Les données de test sont classées à 100%.
- Globalement les données sont classées à 99%

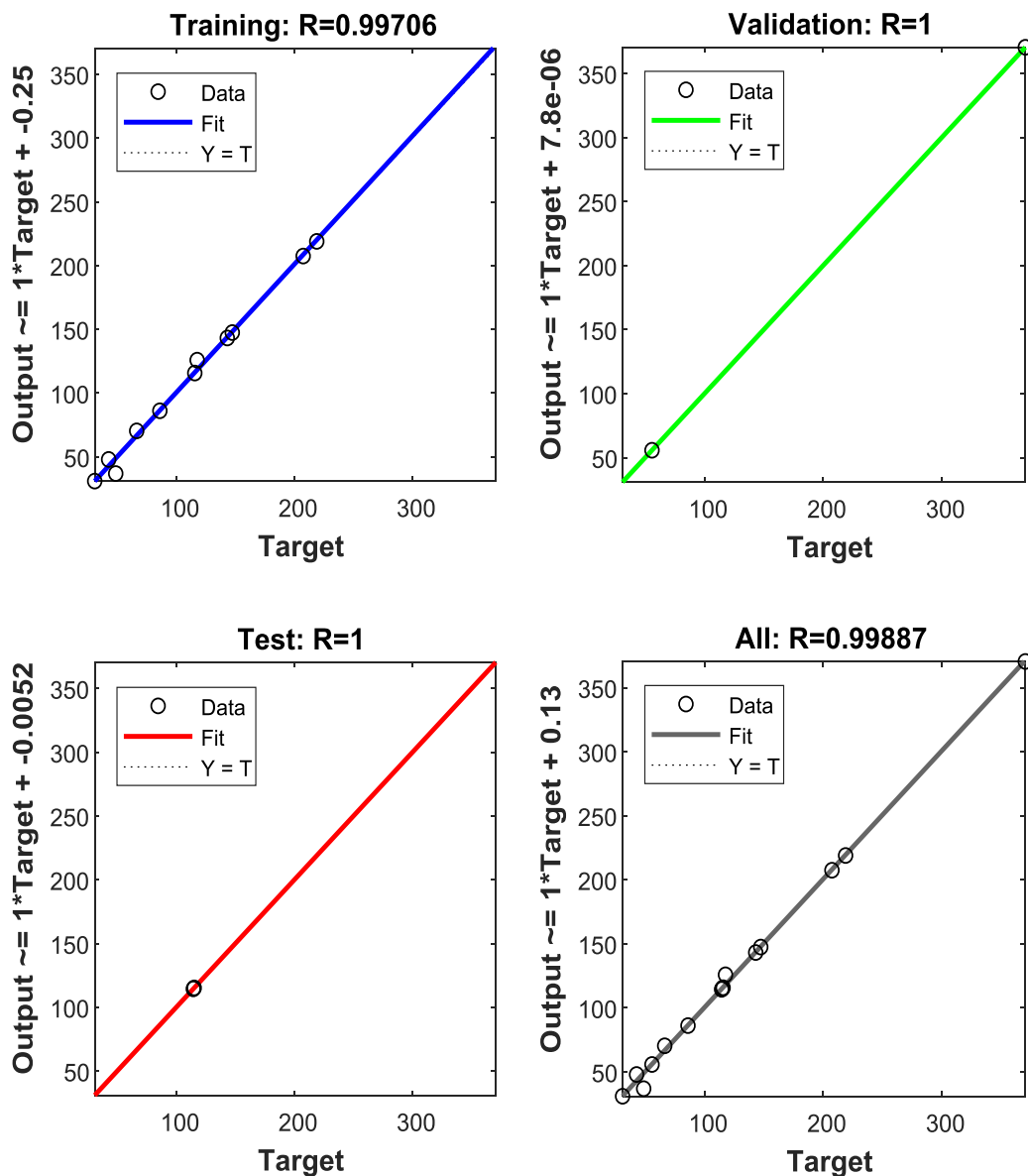


Figure.III.10 . Courbes de régression de la 2<sup>ème</sup> simulation.

### III.5. Comparaison des résultats

Après la détermination des caractéristiques du réseau, nous présentons au réseau des nouveaux exemples. Pour cela, on utilise la fonction simulation définie comme suit :

$$V = \text{sim}(\text{RNA}, ([E1 \ E2]^T))$$

Avec  $E1$  : la conductivité.

$E2$  : la largeur des bandes propres.

$V$  : la sortie calculé par réseau.

### III.5.1. Création de la fonction « sim »

Pour appliquer la fonction sim, il faut exporter le réseau de neurones vers workspace du MATLAB en sélectionnant l'onglet (Export) comme montré sur la figure suivante :

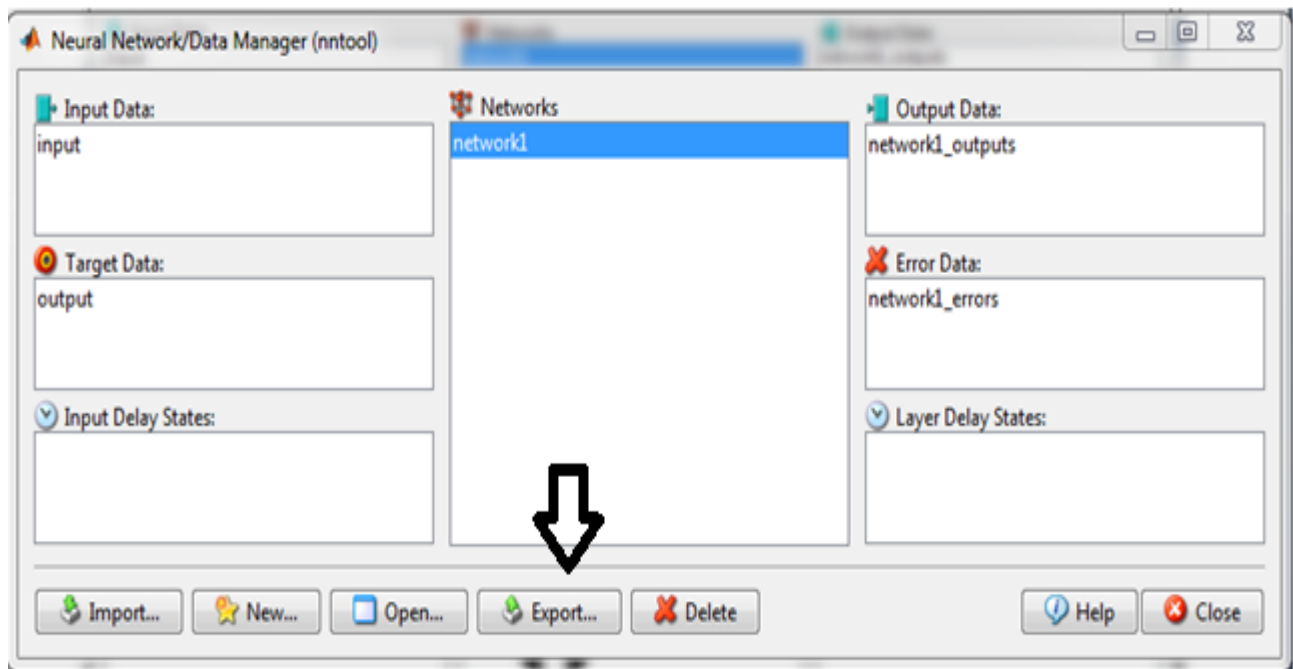


Figure III.11. Exportation de notre réseau nommé RNA vers l'espace de travail du MATLAB

### III.5.2. Résultats

Nous présentons ci-après deux tableaux, le premier tableau contient les tensions de contournement des isolateurs désirées (pratique) et celles calculées (prédites) par RNA pour différentes valeurs de conductivités et largeur des bandes propres. Le deuxième tableau contient les courant de fuites désirés (pratique) et ceux calculés (prédites) par RNA pour différentes valeurs des tension appliquées et largeur des bandes propres. Les valeurs prédites sont calculées en utilisant la fonction Sim (RNA, ([E1 E2]')) et les valeurs désirées correspondantes sont issues de la référence [42].

Après la formation de réseau de neurones on peut obtenir différentes sorties pour la même entrée. Afin de garantir l'obtention d'une bonne précision de réseau de neurones, nous devons le recycler plusieurs fois. Dans cette étude, nous avons utilisé l'erreur quadratique moyenne (MSE) pour vérifier les performances de la classification. Quand la formation a été achevée, nous avons vérifié la performance du réseau. Les résultats obtenus sont donnés dans le tableau III.7 et le tableau III.8.

Ces résultats montrent une grande précision de prédiction de courant de fuites et de la tension de contournement donnée par notre RNA. Toutes les prédictions effectuées sont justes sauf une, ce qui donne un taux de réussite de prédiction de 90%. Ce qui prouve l'efficacité du RNA dans la prédiction de la tension de contournement et de courant de fuites.

**Tableau III.7** Tableau des résultats de la première simulation

Largeur des bandes propres	Tension Appliquée	Courant de fuite calculé par (RNA)	Courant de fuite désiré	Validé
3*1cm	10	46.765	48,825	×
3*1cm	20	114,5	114,5	✓
3*1cm	30	207,5	207,5	✓
3*1cm	40	370,5	370,5	✓
3*1cm	50	117,75	117,75	✓
3*3cm	10	42,825	42,825	✓
3*3cm	20	66,55	66,55	✓
3*3cm	30	86,175	86,175	✓
3*3cm	40	115,75	115,75	✓
3*3cm	50	143,25	143,25	✓
3*5cm	10	31,025	31,025	✓
3*5cm	20	55,825	55,825	✓
3*5cm	30	115,25	115,25	✓
3*5cm	40	147,5	147,5	✓
3*5cm	50	219	219	✓

**Tableau III.8** Tableau de résultats de deuxième simulation

Conductivité	Largeur des bandes propres	Tension de Contournement calculée par (RNA)	Tension de Contournement désirée	Validé
1mS/cm	3*1cm	115.3431	115.33	✓
1mS/cm	3*3cm	128.0001	128	✓
1mS/cm	3*5cm	100.9898	112.66	×
Sable de mer	3*1cm	58.5369	58.66	✓
Sable se sol	3*1cm	126.6680	126.66	✓
Sable des dunes	3*1cm	132.9456	133.33	✓

**Tableau.III. 9** Comparaisons des résultats pratiques de la prédiction de la tension de contournement de l'isolateur trouvé par FL et RNA

conductivité	largeur des bandes propres	tension de contournement pratique	Test pratique	Etat d'isolateur prédire par LF	Tension de contournement prédire par réseau neurones		
					Tension calculée	Erreur % absolu	Etat d'isolateur
1mS/cm	3*1cm	115.33	contournement	contournement	115.3431	0.0001	contournement
1mS/cm	3*3cm	128	contournement	contournement	128.0001	10 <sup>-6</sup>	contournement
1mS/cm	3*5cm	112.66	contournement	contournement	100.9898	0.116	contournement
Sable de mer	3*1cm	58.66	contournement	contournement	58.5369	0.0013	contournement
Sable se sol	3*1cm	126.66	contournement	contournement	126.6680	8*10 <sup>-5</sup>	contournement
Sable de dune	3*1cm	133.33	contournement	contournement	132.9456	0.0039	contournement

Il existe une similitude suffisante entre les résultats obtenus en utilisant le concept RNA, les résultats pratiques et les résultats de logique flou (LF) [46]. Ceci montre l'efficacité du type RNA et des paramètres d'entraînement choisi dans cette étude.

### Conclusion

A travers, ce chapitre, nous avons présenté l'utilisation des réseaux de neurones artificiels pour la prédiction de la tension de contournement des isolateurs et les courants de fuites traversant la surface des isolateurs.

D'après cette étude, nous constatons l'efficacité du RNA dans la prédiction de courant de fuites en considérant que deux paramètres influençant sur le courant de fuite qui sont : la tension appliquée et la largeur des bandes propres.

Un tableau de résultats est présenté vers la fin de ce chapitre. Ce tableau montre une très bonne capacité du RNA de prédire les tensions de contournement en comparant avec les résultats expérimentaux et ceux donnés par LF pour différentes valeurs de conductivités et largeur des bandes propres. Ce qui donne une importance à notre technique de prédiction basée sur les réseaux de neurones artificiels.

L'utilitaire NNTOOL a montré une aptitude indiscutable dans la création et l'apprentissage de notre RNA simulé sous l'environnement MATLAB.