Cahier d'expériences

Exploration de la notion de méta-apprentissage

Dans quelle mesure un système apprenant peut prendre conscience de ses performances et altérer son comportement ?

Yann Boniface, Alain Dutech, Nicolas Rougier
Matthieu Zimmer

31 mai 2012

Table des matières

Expérience A1

Objectif

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, plus particulièrement sur des perceptrons multicouches.

Reproduction et approfondissement des résultats de la première expérience de l'article Cleeremans Alex (2007).

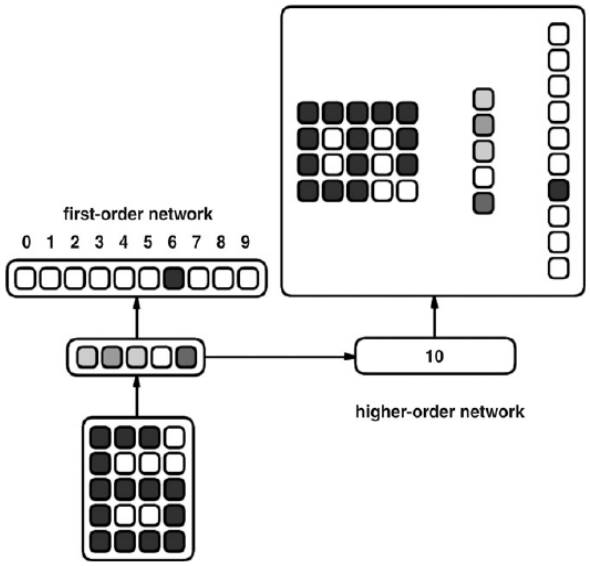
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

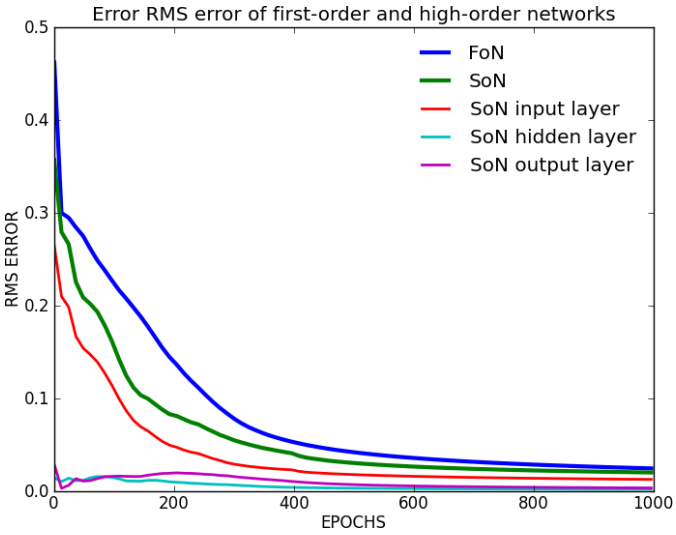


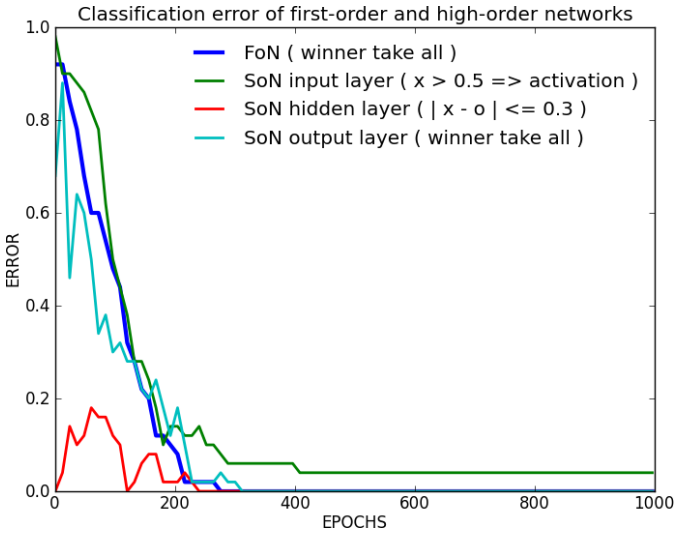
Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances





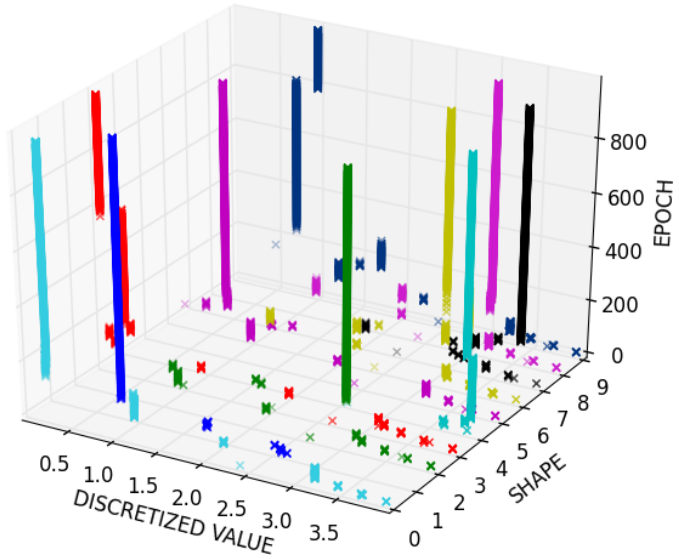
Notes

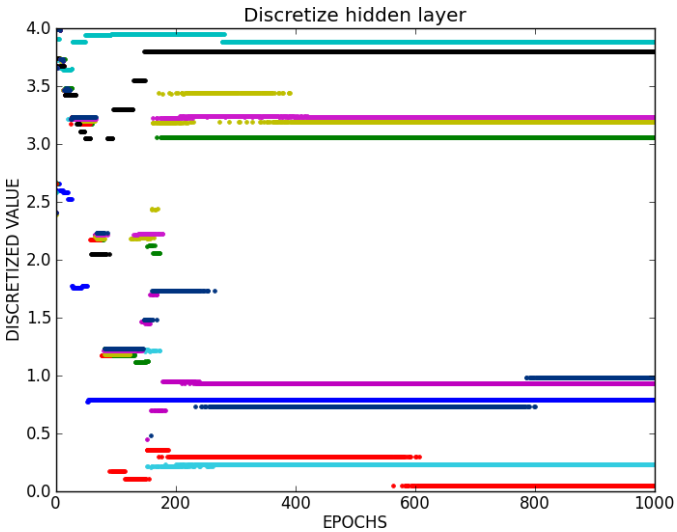
- formule utilisée pour RMS (cf. Formules RMS)
- les courbes SoN layer représentent les erreurs (du second réseaux) sur les couches à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- le premier réseau réussit à apprendre sa tâche de classification
- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Secondaires Discrétisation de la couche cachée du premier réseau

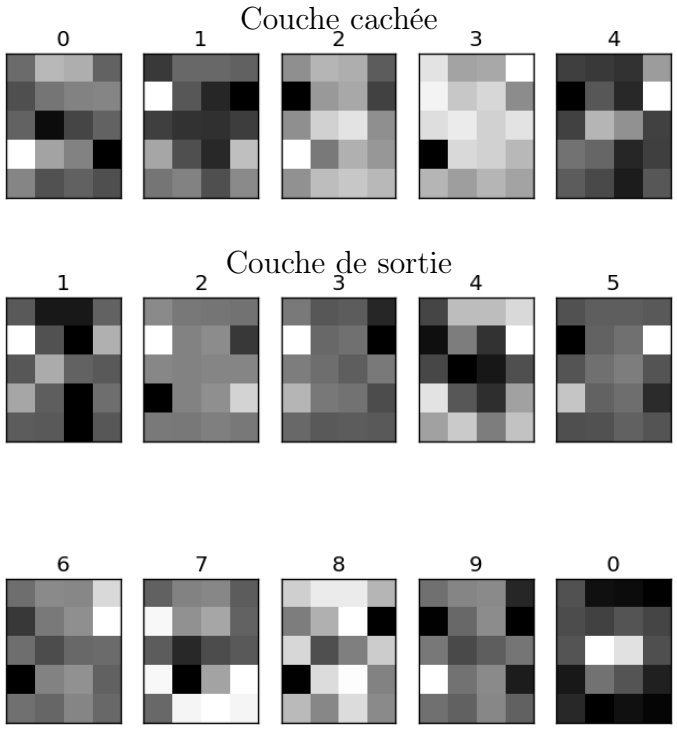




Notes
- une couleur équivaut à un chiffre présenté
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes Discrétisation)

Conclusion Les neurones se stabilisent très rapidement (autour de la 50$^{\text{ième}}$ époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

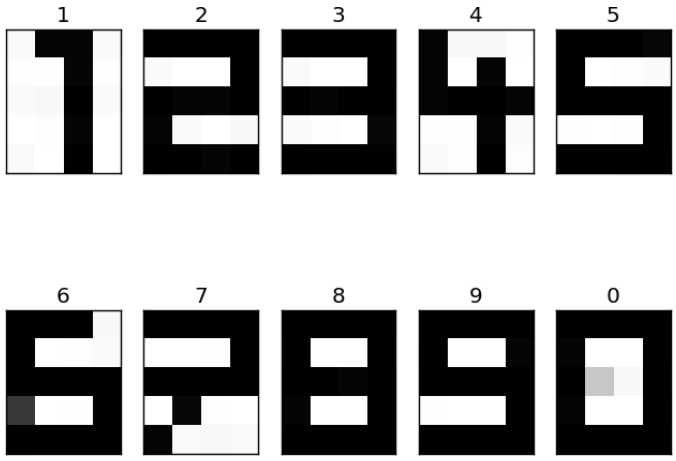
Secondaires Représentations au travers des poids du premier réseau



Notes
- plus une case est noire, plus sa présence est importante pour le chiffre en question
- plus une case est blanche, plus son absence est importante

Conclusion Il est assez difficile d'y distinquer les chiffres, mais cela semble suffisant pour le réseau qui a un taux de reconnaissance de 100%.

Secondaires Prototypes à l'intérieur de la première partie de la couche de sortie du second réseau



Notes
– Il sagit de la moyenne des réponses du second réseaux sur toutes les entrées

Conclusion Le peu d'entrées permet l'apprentissage par-coeur de chaque forme.

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience A2

Objectif

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, plus particulièrement sur des perceptrons multicouches.

Dans l'optique d'éprouver la première expérience de l'article Cleeremans Alex (2007) sur des données plus réelles, il sagit, dans un premier temps, de montrer qu'une augmentation du nombre de neurones, et qu'un simple agrandissement des chiffres en entrées, n'affecte pas fondamentalement le fonctionnement du réseau.

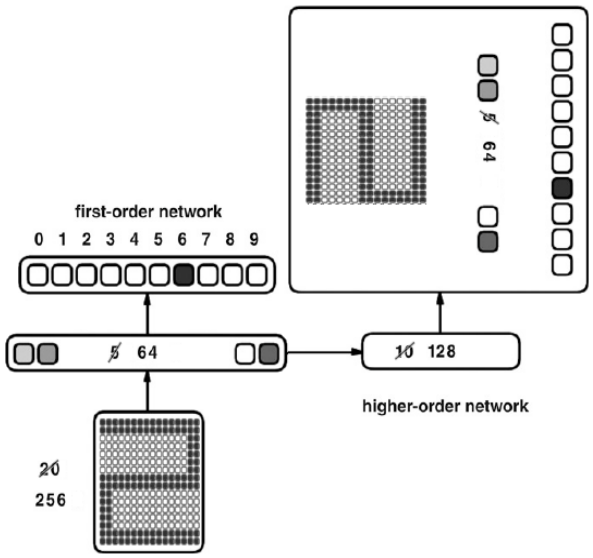
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 256 (16x16) neurones d'entrées. Il est composé d'une couche cachée de 64 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

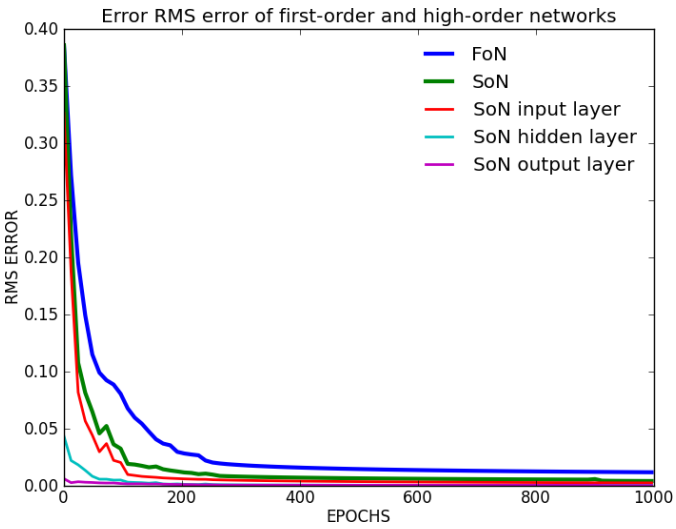


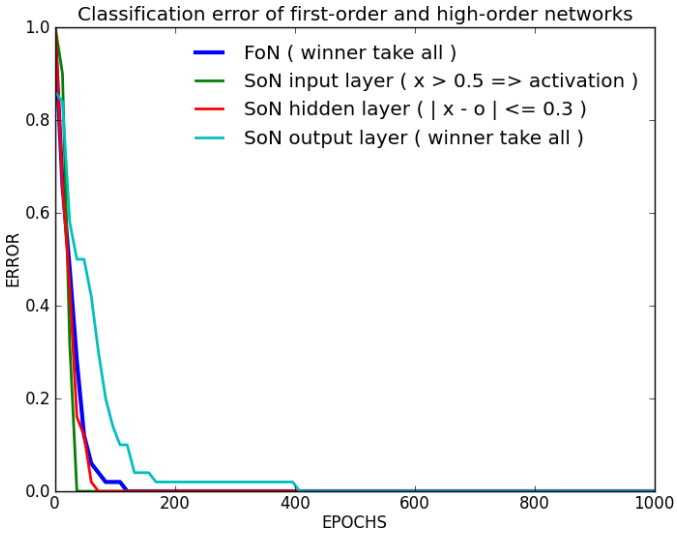
Paramètres

- momentum : 0.9 sur les 2 réseaux
- taux d'apprentissage : 0.1 sur les 2 réseaux
- **10 formes** de chiffres différents présentées (shuffle)
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances





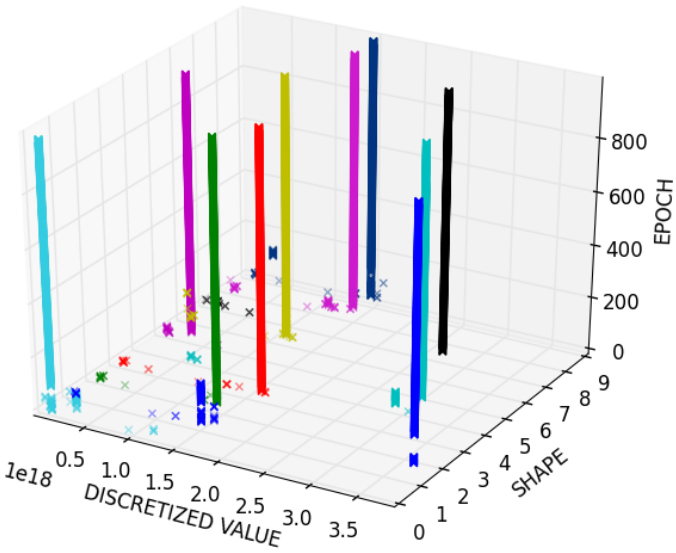
Notes

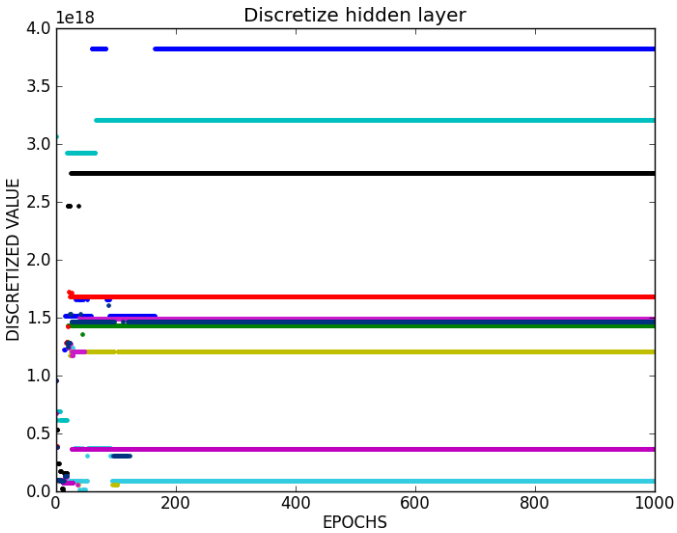
- formule utilisée pour RMS (cf. Formules RMS)
- les courbes SoN layer représentent les erreurs (du second réseaux) sur les couches à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- le premier réseau réussit à apprendre sa tâche de classification
- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier
- l'augmentation du nombre de neurone ne change pas les tendances des courbes

Secondaires Discrétisation de la couche cachée du premier réseau

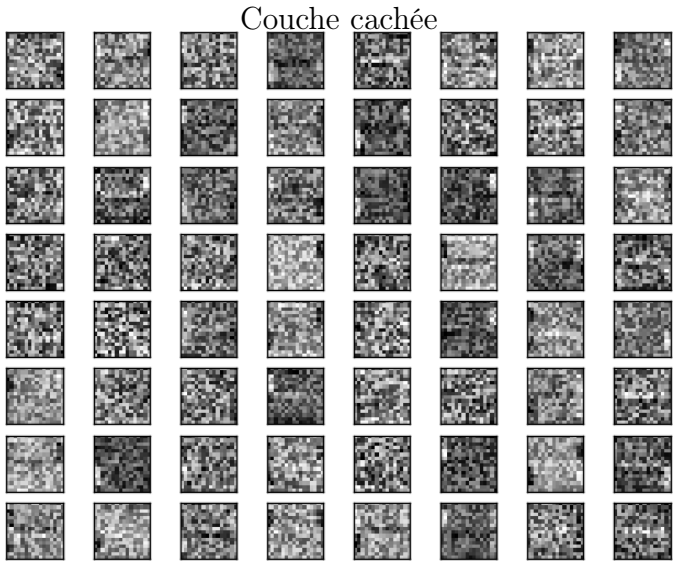


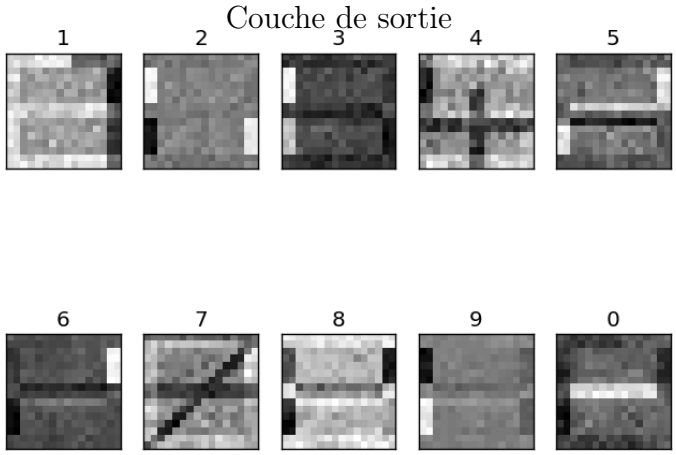


Notes
- une couleur équivaut à un chiffre présenté
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes Discrétisation)

Conclusion Les neurones se stabilisent très rapidement (autour de la $50^{\text{ième}}$ époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

Secondaires Représentations au travers des poids du premier réseau

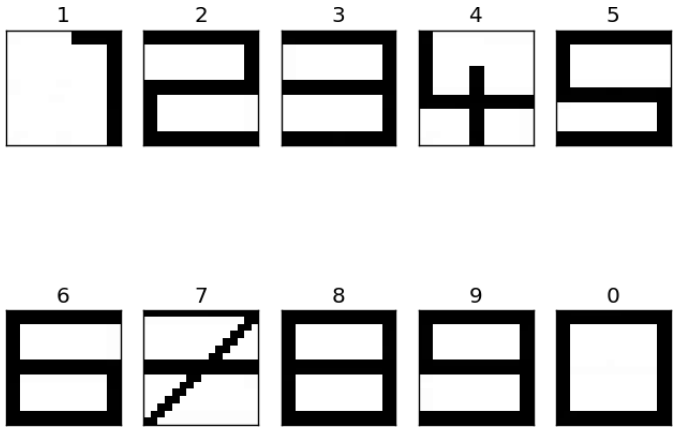




Notes
- plus une case est noire, plus sa présence est importante pour le chiffre en question
- plus une case est blanche, plus son absence est importante

Conclusion Il est intéréssant de remarquer qu'au vu du peu d'entrée, le réseau cible des neurones très précis. Pour le chiffre 2, par exemple, seul 3 lignes sont ciblées.

Secondaires Prototypes à l'intérieur de la première partie de la couche de sortie du second réseau



Notes
– Il sagit de la moyenne des réponses du second réseaux sur toutes les entrées

Conclusion Le peu d'entrées permet l'apprentissage par-coeur de chaque forme.

Conclusion

L'augmentation du nombre de neurones dans le réseau, et l'augmentation de la taille des entrées ne modifient pas l'attitude du réseau.

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience A3

Objectif

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, plus particulièrement sur des perceptrons multicouches.

Réalisation de la première expérience de l'article Cleeremans Alex (2007) sur des données réelles.

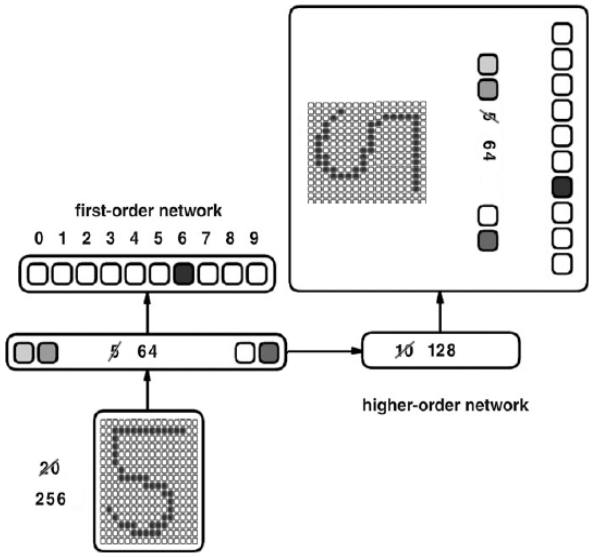
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 256 (16x16) neurones d'entrées. Il est composé d'une couche cachée de 64 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

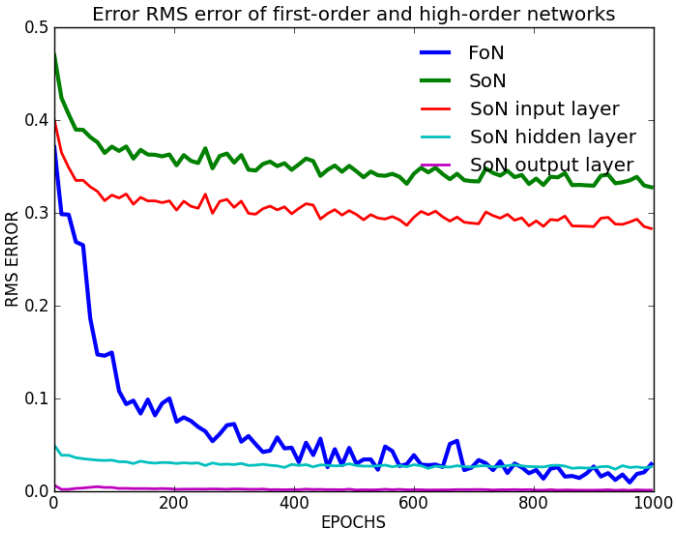


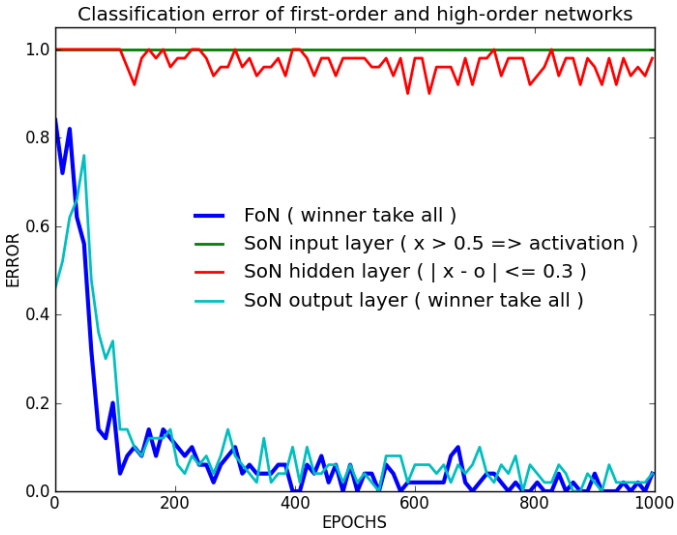
Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- **1600 formes** de chiffres différents présentées (shuffle) Semeion Research Center (1994)
- apprentissage 10 (formes) x 1000 (époques)
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1
- utilisation de biais

Résultats

Principaux Analyse des performances





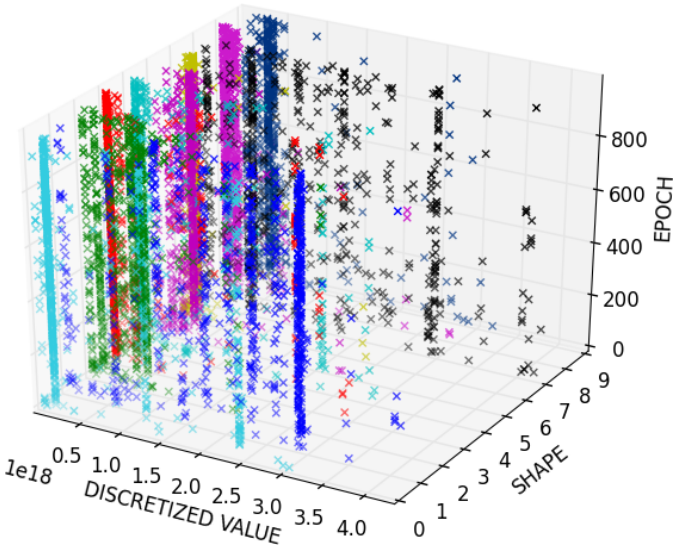
Notes

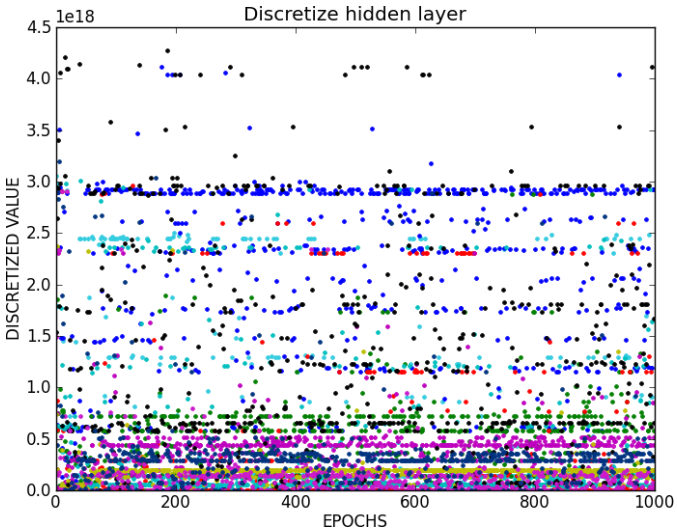
- formule utilisée pour RMS (cf. Formules RMS)
- les courbes SoN layer représentent les erreurs (du second réseaux) sur les couches à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- le premier réseau réussit à apprendre sa tâche de classification
- la couche cachée et la couche de sortie posent peu de problèmes d'apprentissage
- le second réseau a du mal à reproduire les entrées
- le second réseau apprend maintenant (vs Expérience A1, Expérience A2) moins rapidement que le premier

Secondaires Discrétisation de la couche cachée du premier réseau





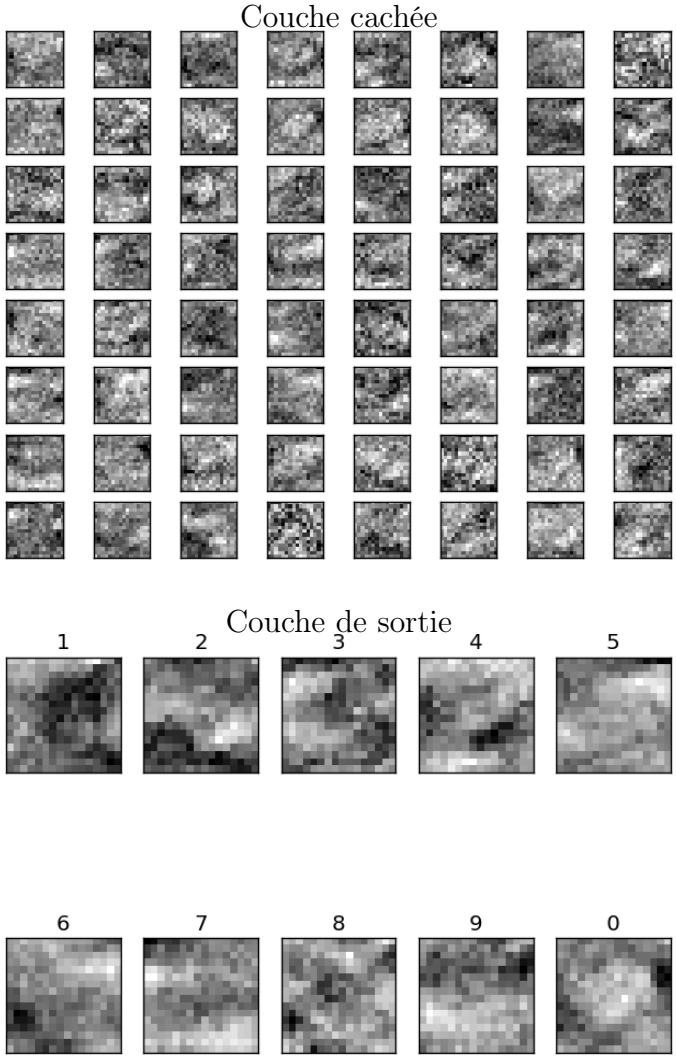
Notes
- une couleur équivaut à un chiffre présenté
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes Discrétisation)

Conclusion Nous pouvons maintenant expliquer les raisons provoquant le mauvais apprentissage des entrées par le second réseau :

Pour un seul chiffre présenté, il existe différentes valeurs de la couche cachée le représentant (factorisation dans la couche cachée du premier réseau). Le second réseau ne peut donc pas apprendre toutes les entrées.

Une même valeur discretisée peut correspondre à plusieurs couleurs (différents nombres) ce qui augmentent la difficulté d'apprentissage.

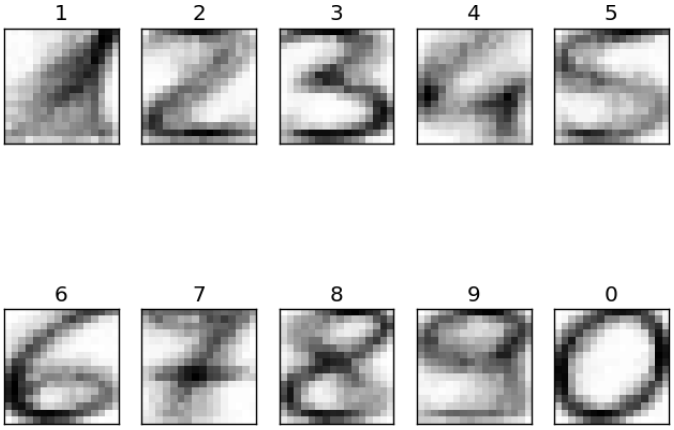
Secondaires Représentations au travers des poids du premier réseau



Notes
- plus une case est noire, plus sa présence est importante pour le chiffre en question
- plus une case est blanche, plus son absence est importante

Conclusion On arrive, sans trop de mal, à distinguer les chiffres.

Secondaires Prototypes à l'intérieur de la première partie de la couche de sortie du second réseau



Notes
– Il sagit de la moyenne des réponses du second réseaux sur toutes les entrées

Conclusion Il est intéressant de constaté que malgrès la difficulté de la tâche, ces prototypes émergent.

Et que contrairement à ce qu'indique les performances, les chiffres sont bien distincts.

On peut, par ailleurs, conjecturer que plus la couche cachée du premier réseau est petite, plus les prototypes seront factorisés (ie. personnels au réseau).

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Semeion Research Center, o. S. o. C. (1994). Semeion handwritten digit data set. 1593 handwritten digits from around 80 persons.

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience A4

Objectif

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, plus particulièrement sur des perceptrons multicouches.

Réalisation de la première expérience de l'article Cleeremans Alex (2007) sur des données réelles et non linéairement sépérables.

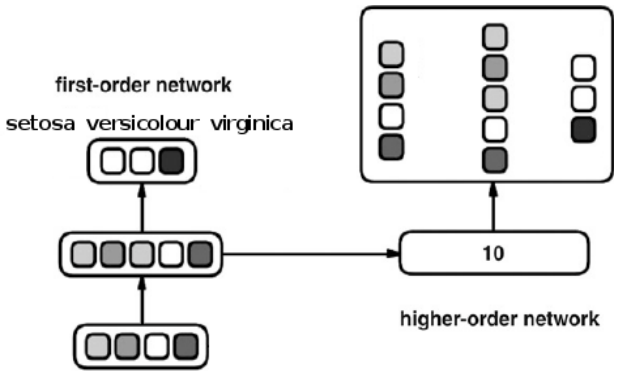
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des fleurs caractérisées par 4 neurones d'entrées qui représentent taille et largeur de la pétale et la sépale. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

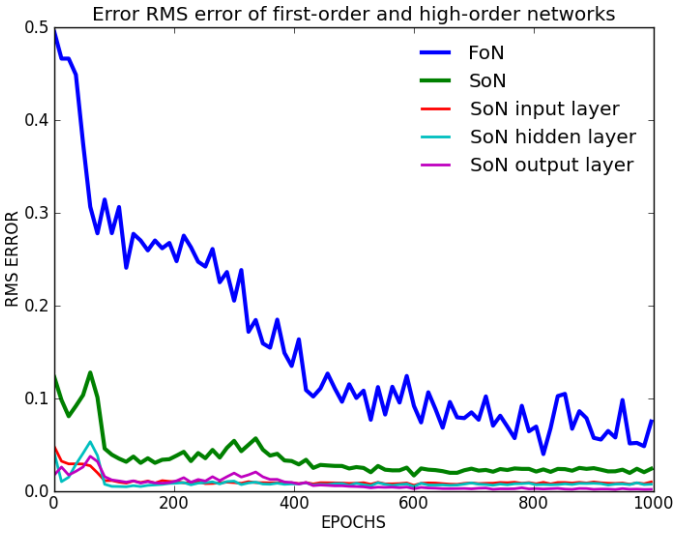


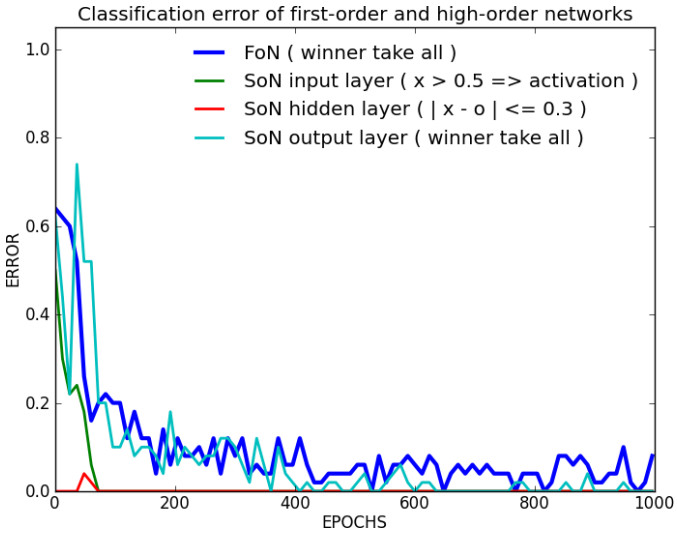
Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 150 fleurs différentes présentés Fisher (1988)
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées réelles sur [0 ; 1]
- sigmoïde à température 1

Résultats

Principaux Analyse des performances





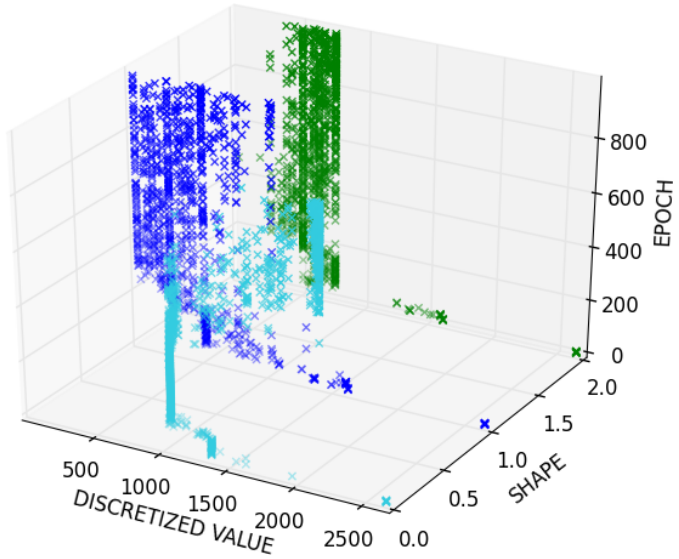
Notes

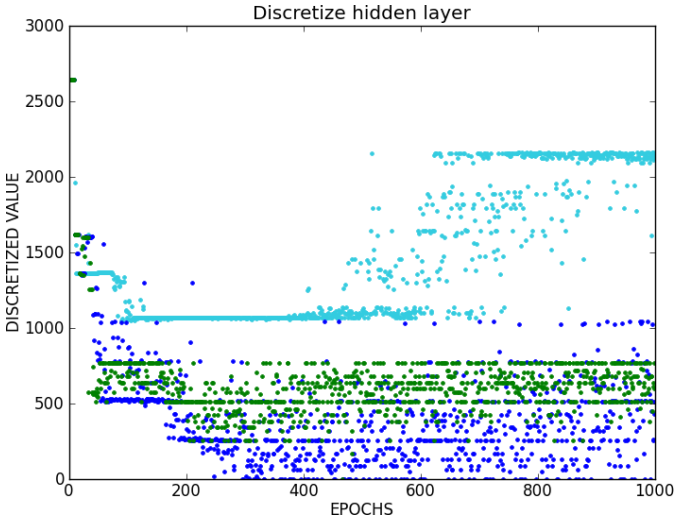
- formule utilisée pour RMS (cf. Formules RMS)
- les courbes SoN layer représentent les erreurs (du second réseaux) sur les couches à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- le premier réseau réussit à apprendre sa tâche de classification
- la couche cachée et la couche de sortie posent peu de problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Secondaires Discrétisation de la couche cachée du premier réseau





Notes
- une couleur équivaut à une fleur présentée
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes Discrétisation)

Conclusion Les données n'étant pas linéairement sépérables, leur passage dans une seule couche rend la tâche d'apprentissage du second réseau plus complexe.

On peut voir plusieurs points de différentes couleurs aux mêmes endroits.

Conclusion Le peu d'entrées permet l'apprentissage par-coeur de chaque forme.

Conclusion

Le passage sur des données non linéairement sépérables se passent très bien.

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Fisher, R. (1988). Iris plants database. 150 (50 in each of three classes).

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience B1

Objectif

Reproduction et approfondissement des résultats de la première expérience 1 dans l'article Cleeremans Alex (2007).

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, en particulier sur des perceptrons multicouches.

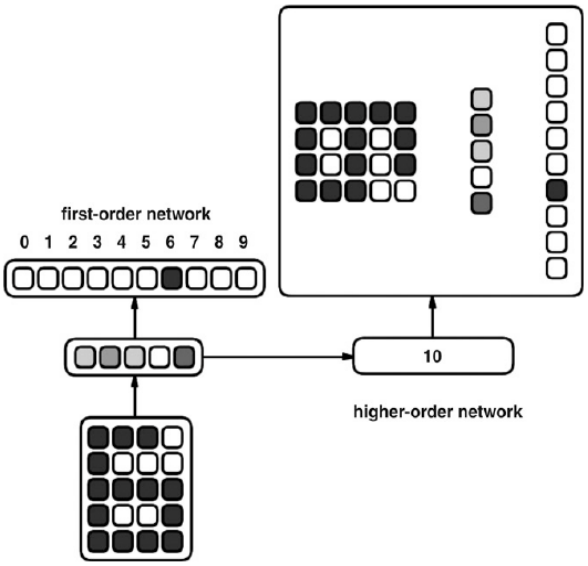
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

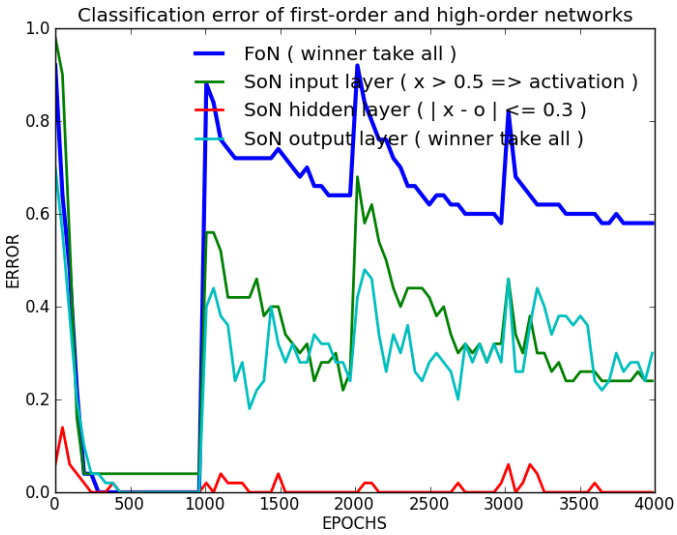


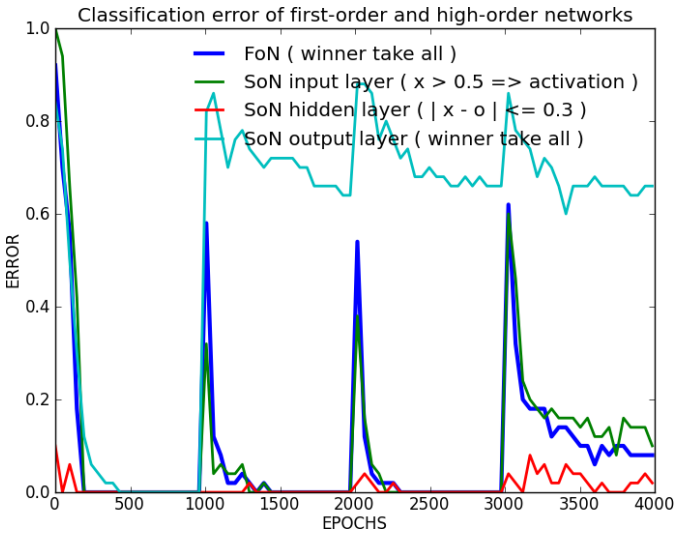
Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances

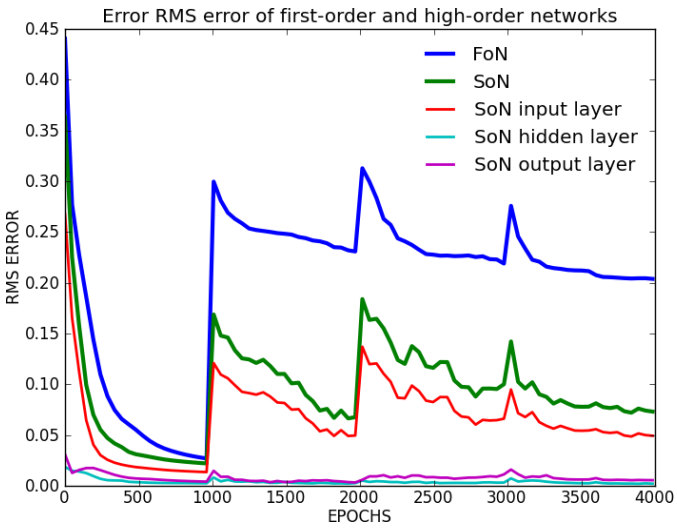


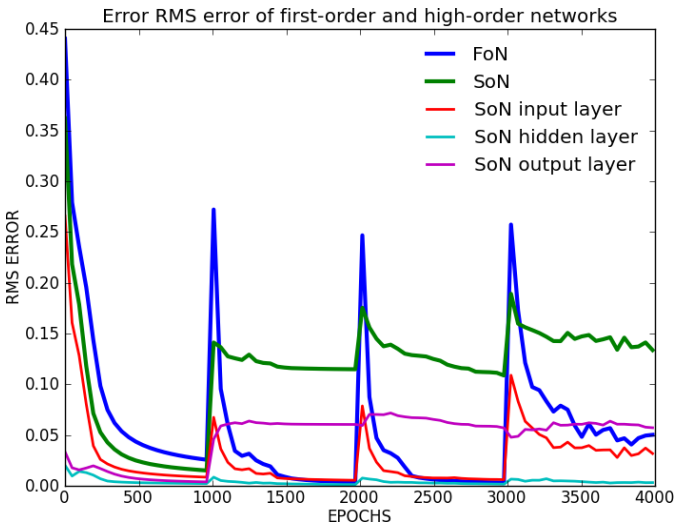


Notes
- les courbes SoN layer représentent les erreurs (du second réseaux) des couches du premier à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion
- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Secondaires RMS

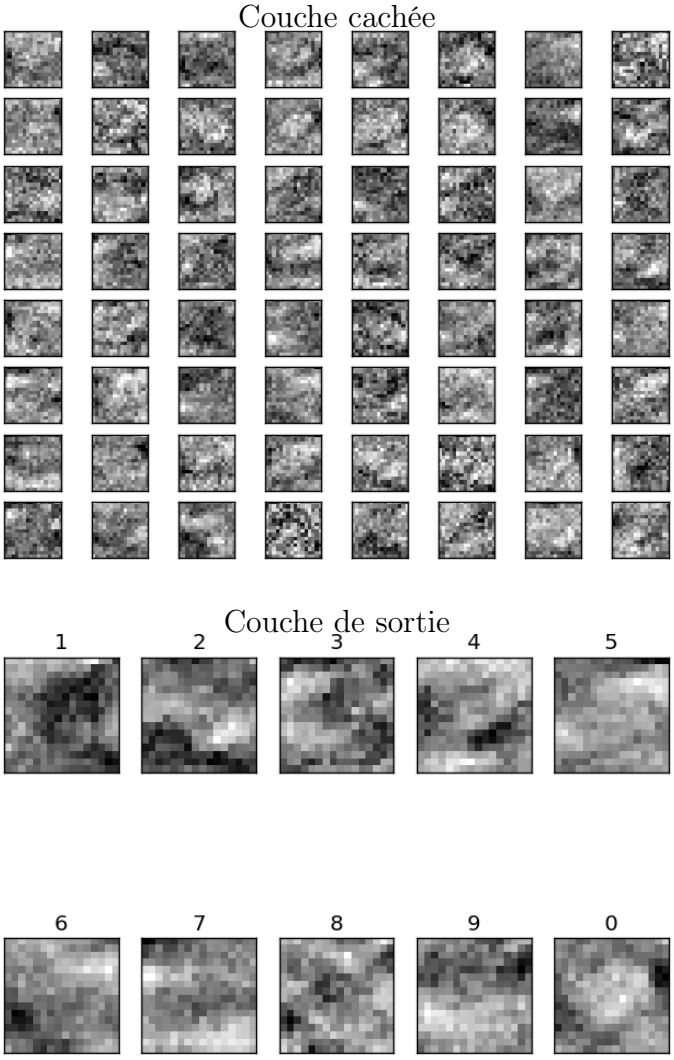




Notes
- une couleur équivaut à un chiffre présenté
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes)

Conclusion Les neurones se stabilisent très rapidement (autour de la 50$^{\text{ième}}$ époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

Secondaires Représentations au travers des poids du premier réseau



Notes
- plus une case est noire, plus sa présence est importante pour le chiffre en question
- plus une case est blanche, plus son absence est importante

Conclusion Il est assez difficile d'y distinquer les chiffres, mais cela semble suffisant pour le réseau qui a un taux de reconnaissance de 100%.

Conclusion Le peu d'entrées permet l'apprentissage par-coeur de chaque forme.

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience B3

Objectif

Reproduction et approfondissement des résultats de la première expérience 1 dans l'article Cleeremans Alex (2007).

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, en particulier sur des perceptrons multicouches.

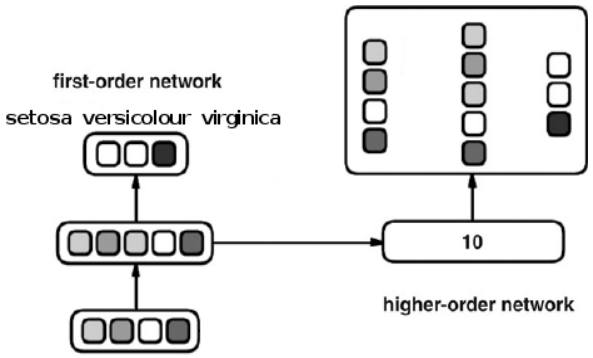
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

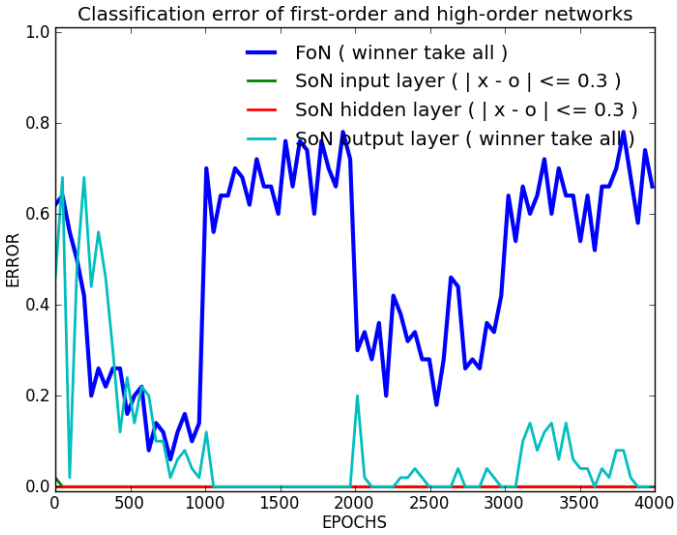


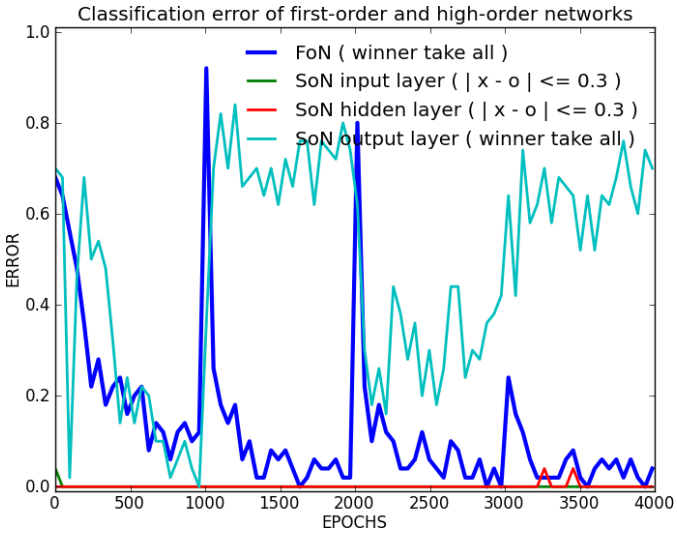
Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances





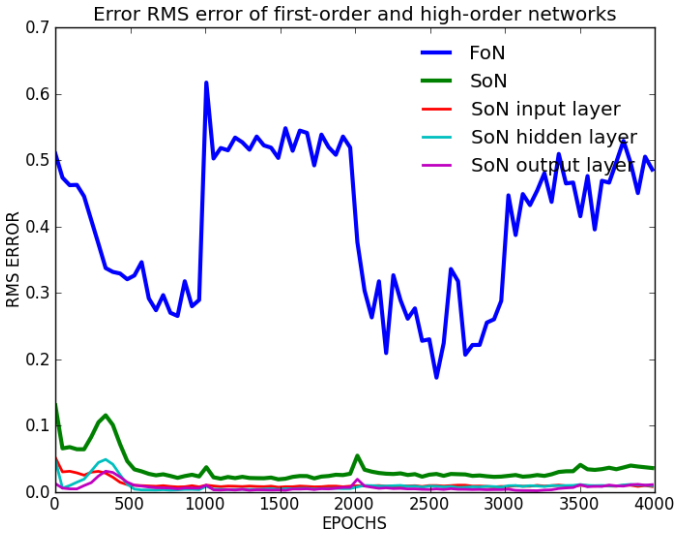
Notes

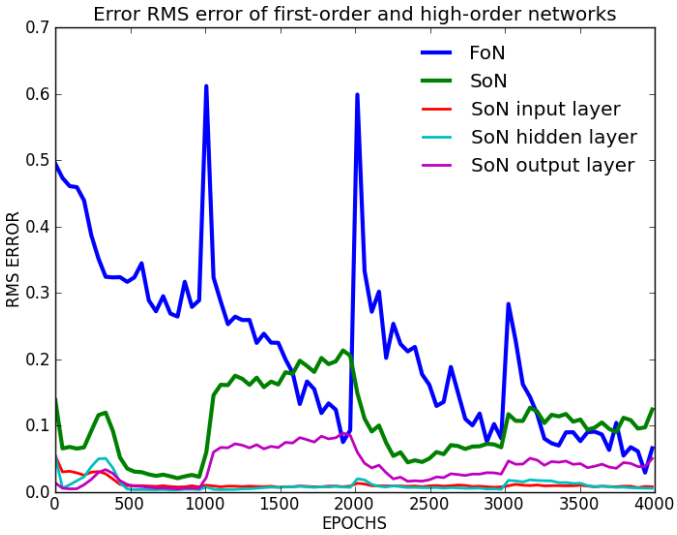
- les courbes SoN layer représentent les erreurs (du second réseaux) des couches du premier à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Secondaires RMS





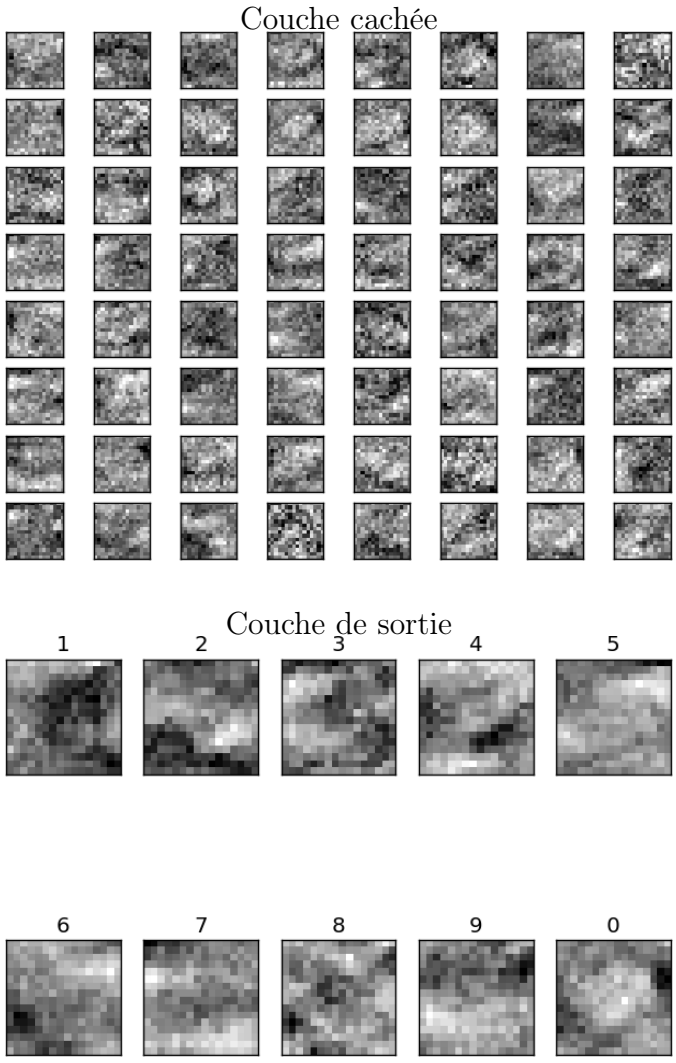
Notes

- une couleur équivaut à un chiffre présenté

– une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes)

Conclusion Les neurones se stabilisent très rapidement (autour de la $50^{\text{ième}}$ époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

Secondaires Représentations au travers des poids du premier réseau



Notes
– plus une case est noire, plus sa présence est importante pour le chiffre en question
– plus une case est blanche, plus son absence est importante

Conclusion Il est assez difficile d'y distinquer les chiffres, mais cela semble suffisant pour le réseau qui a un taux de reconnaissance de 100%.

Conclusion Le peu d'entrées permet l'apprentissage par-coeur de chaque forme.

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience C1

Objectif

Comprendre de quelles manières un réseau de neurone connexionniste peut parier sur ses propres résultats à partir de ses représentations personnelles.

Reproduction et approfondissement des résultats de la seconde expérience de l'article Cleeremans Alex (2007).

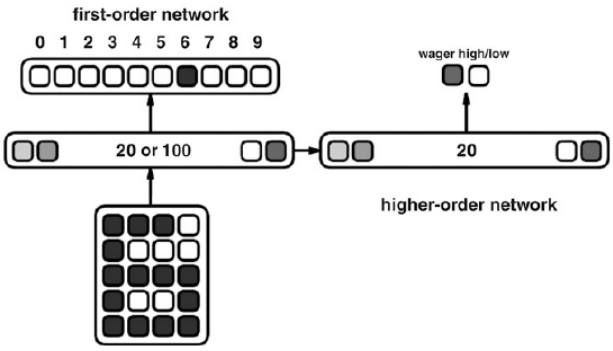
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 20 ou 100 neurones.

Un second réseau de perceptron multicouche apprend à parier sur la qualité de la réponse du premier réseau à partir de sa couche cachée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

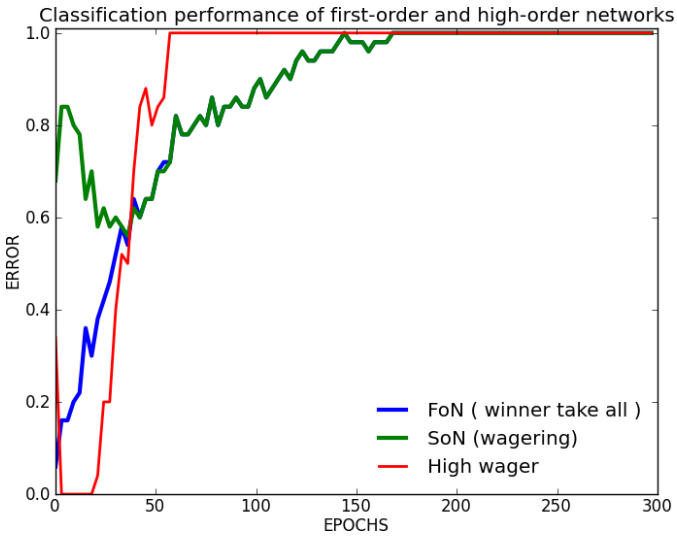


Paramètres

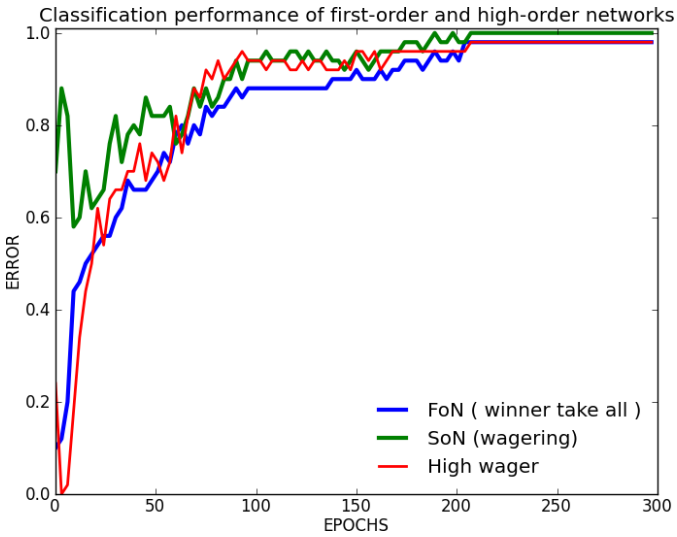
- momentum : 0.5 sur les 2 réseau
- taux d'apprentissage : 0.15 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 300 (époques)
- utilisation de biais
- sigmoïde à température 1
- poids initialisés sur [-1 ; 1] pour les 2 réseaux
- taux d'apprentissage constant
- entrées valent 0 ou 1

Résultats

Principaux Analyse des performances



20 neurones en couche cachée



100 neurones en couche cachée

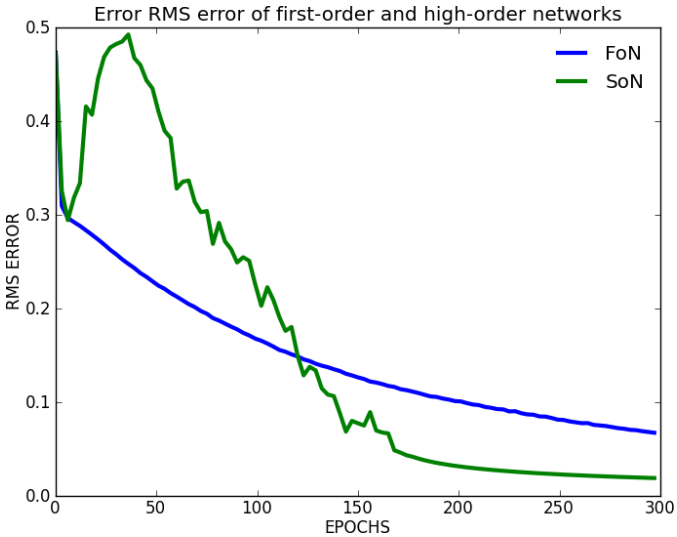
Notes

– la courbe rouge représentent le taux de paris hauts du second réseau

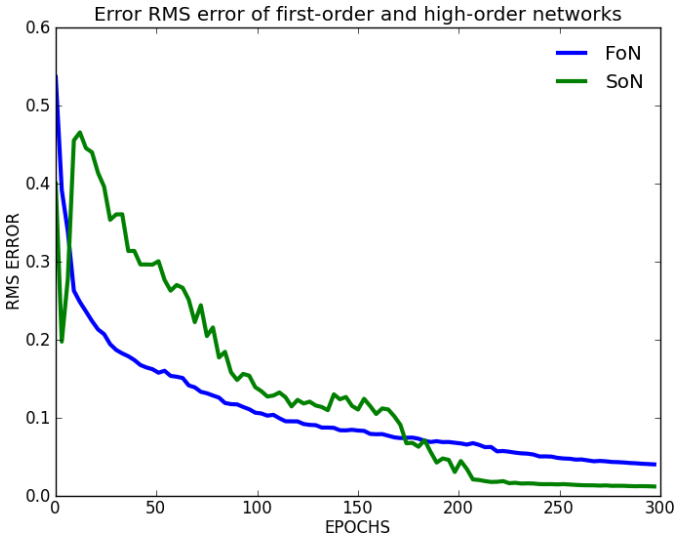
Conclusion

– dans les 2 cas, le premier réseau réussit à apprendre sa tâche de classification
– lorsque le nombre de neurones dans la couche cachée est faible, l'apprentissage de la tâche est optimal, il ne peut plus être amélioré
  Ainsi le second réseau se contente de parier haut.
– lorsque le nombre de neurones dans la couche cachée est élevé, l'apprentissage peut être amélioré, le second réseau le remarque et peut accorder son taux de paris hauts avec le taux de succès du premier réseau.
  Remarquons tout de même qu'avec l'amélioration du second réseau, on ne peut pas dépasser un réseau optimal, seulement l'égaler

Secondaires RMS



20 neurones en couche cachée



100 neurones en couche cachée

Notes
– formule utilisée pour RMS (cf. Formules RMS)

Conclusion On remarque que le second réseau apprend et désapprend. De plus, il reste toujours sous la barre des 50%.

Conclusion

Cette architecture ouvre des possibilités d'amélioration de l'apprentissage. Elle permet au réseau de détecter lui même un nombre de neurones trop important dans la couche cachée.

On pourrait, par exemple, imaginer un réseau s'autorégulant.

L'expérience suivante (Expérience C2) retente l'expérience sur des données réelles.

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience C2

Objectif

Comprendre de quelles manières un réseau de neurone connexionniste peut parier sur ses propres résultats à partir de ses représentations personnelles.

Réalisation de la seconde expérience de l'article Cleeremans Alex (2007) sur des données réelles.

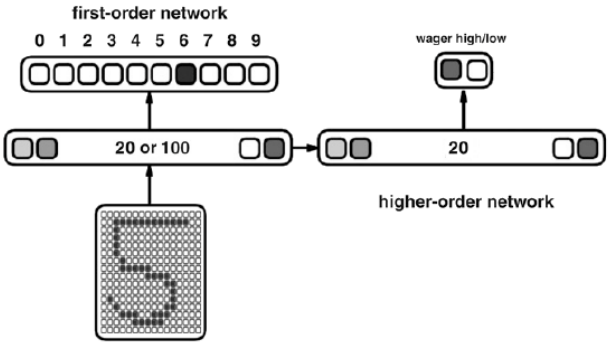
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 256 (16x16) neurones d'entrées. Il est composé d'une couche cachée de 20 ou 100 neurones.

Un second réseau de perceptron multicouche apprend à parier sur la qualité de la réponse du premier réseau à partir de sa couche cachée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

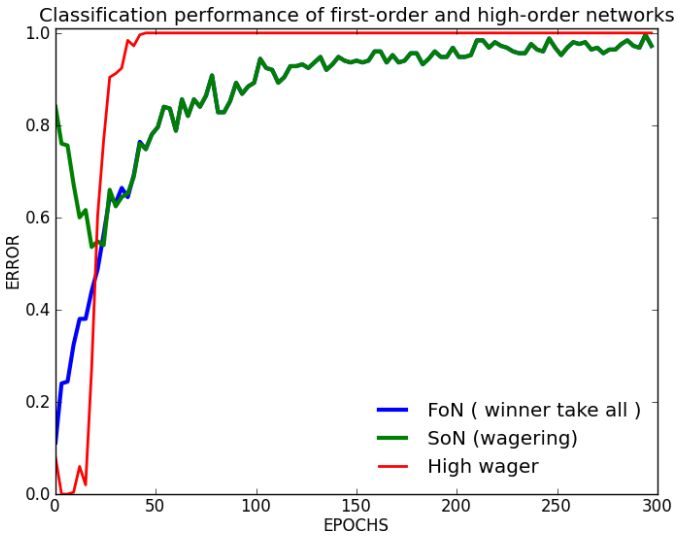


Paramètres

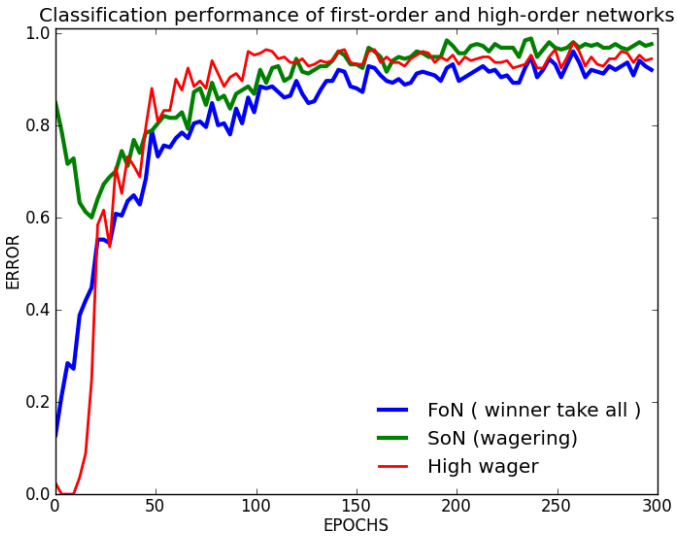
- momentum : 0.5 sur le premier réseau
- momentum : 0. sur le second réseau
- taux d'apprentissage : 0.15 sur les 2 réseau
- **1600 formes** de chiffres différents présentées (shuffle) Semeion Research Center (1994)
- apprentissage 50 (formes) x 300 (époques)
- utilisation de biais
- poids initialisés sur [-1 ; 1] pour le premier réseau
- poids initialisés sur [-0.25 ; 0.25] pour le second réseau
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances



20 neurones en couche cachée



100 neurones en couche cachée

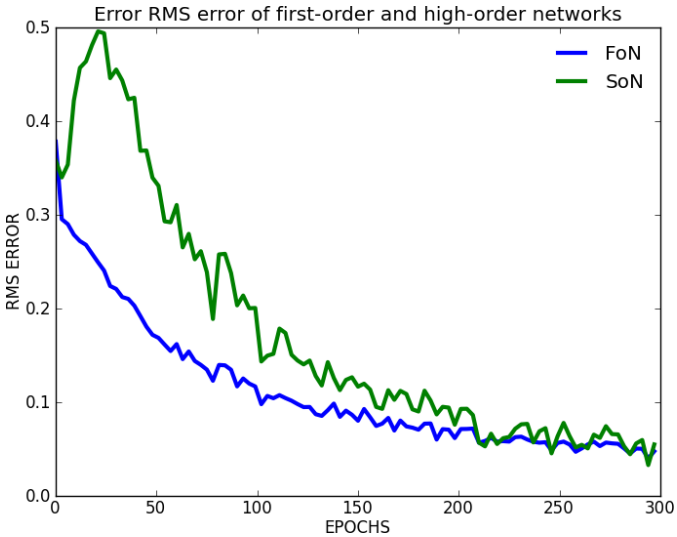
Notes

– la courbe rouge représentent le taux de paris hauts du second réseau

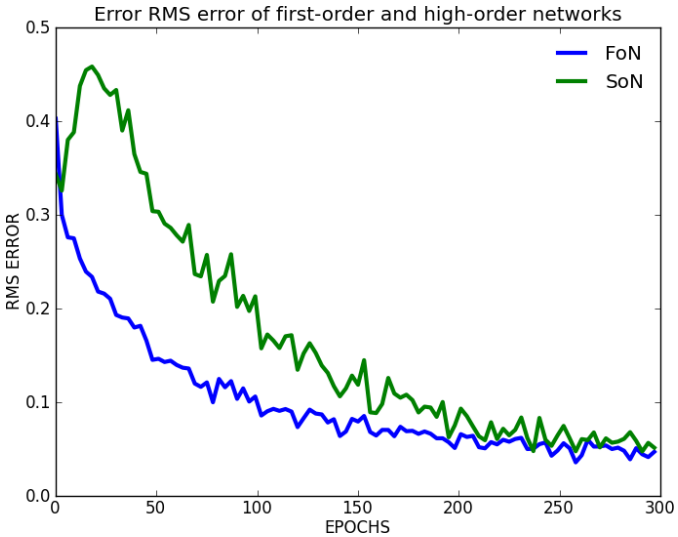
Conclusion

– dans les 2 cas, le premier réseau réussit à apprendre sa tâche de classification
– lorsque le nombre de neurones dans la couche cachée est faible, l'apprentissage de la tâche est optimal, il ne peut plus être amélioré
  Ainsi le second réseau se contente de parier haut.
– lorsque le nombre de neurones dans la couche cachée est élevé, l'apprentissage peut être amélioré, le second réseau le remarque et peut accorder son taux de paris hauts avec le taux de succès du premier réseau.
  Remarquons tout de même qu'avec l'amélioration du second réseau, on ne peut pas dépasser un réseau optimal, seulement l'égaler

Secondaires RMS



20 neurones en couche cachée



100 neurones en couche cachée

Notes
– formule utilisée pour RMS (cf. Formules RMS)

Conclusion On remarque que le second réseau apprend et désapprend. De plus, il reste toujours sous la barre des 50%.

Conclusion

Le passage sur des données réelles ne modifient pas le comportement du réseau.

Cette architecture ouvre des possibilités d'amélioration de l'apprentissage. Elle permet au réseau de détecter lui même un nombre de neurones trop important dans la couche cachée.

On pourrait, par exemple, imaginer un réseau s'autorégulant.

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Semeion Research Center, o. S. o. C. (1994). Semeion handwritten digit data set. 1593 handwritten digits from around 80 persons.

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience C3

Objectif

Comprendre de quelles manières un réseau de neurone connexionniste peut parier sur ses propres résultats à partir de ses représentations personnelles.

Réalisation de la seconde expérience de l'article Cleeremans Alex (2007) sur des données réelles non linéairement séparables.

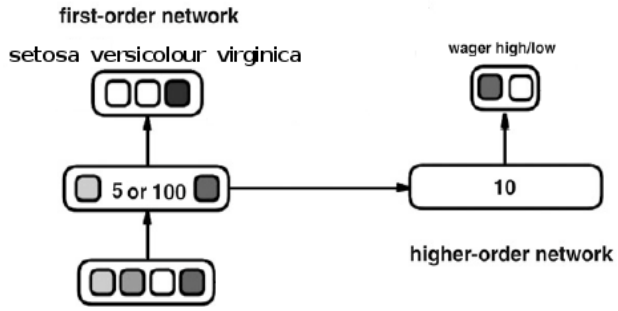
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des fleurs caractérisées par 4 neurones d'entrées qui représentent taille et largeur de la pétale et la sépale. Il est composé d'une couche cachée de 5 ou 100 neurones.

Un second réseau de perceptron multicouche apprend à parier sur la qualité de la réponse du premier réseau à partir de sa couche cachée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

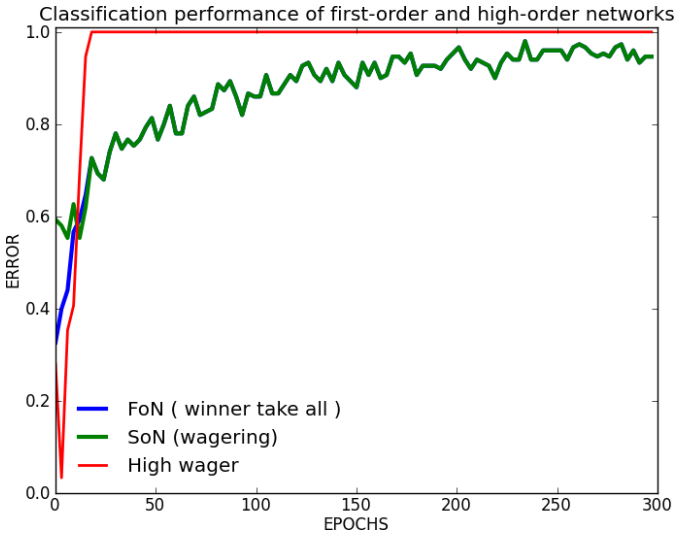


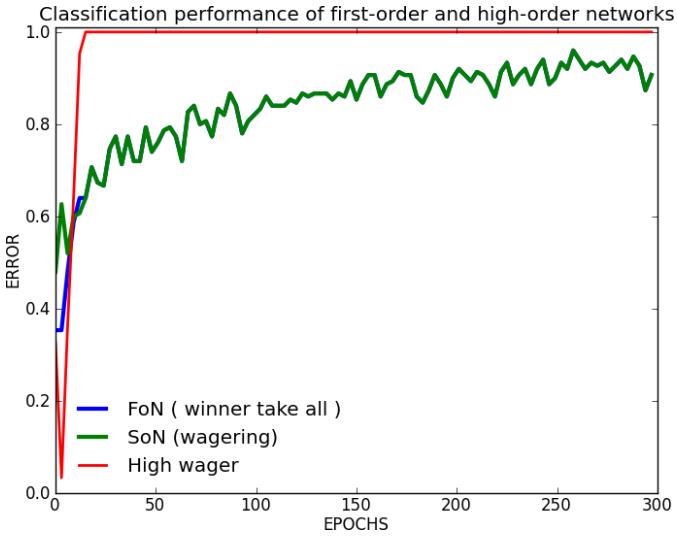
Paramètres

- momentum : 0.5 sur le premier réseau
- momentum : 0. sur le second réseau
- taux d'apprentissage : 0.15 sur les 2 réseau
- **150 formes** de fleurs différents présentées (shuffle) Fisher (1988)
- apprentissage 50 (formes) x 300 (époques)
- utilisation de biais
- poids initialisés sur [-1 ; 1] pour le premier réseau
- poids initialisés sur [-0.25 ; 0.25] pour le second réseau
- taux d'apprentissage constant
- entrées réelles sur [0 ; 1]
- sigmoïde à température 1

Résultats

Principaux Analyse des performances

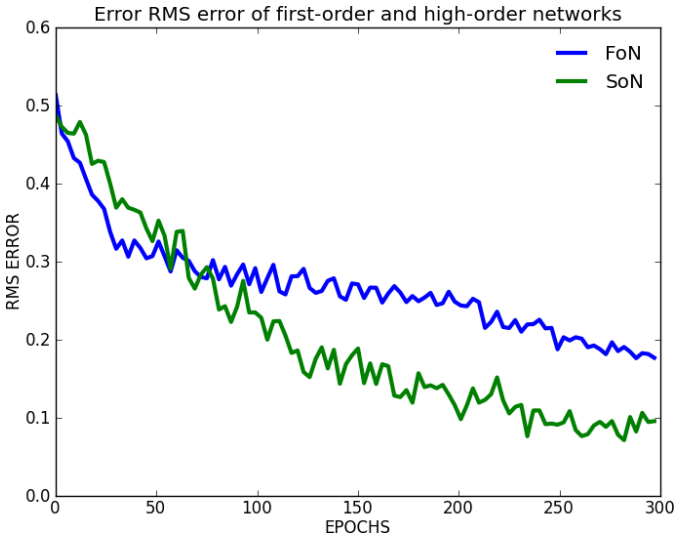


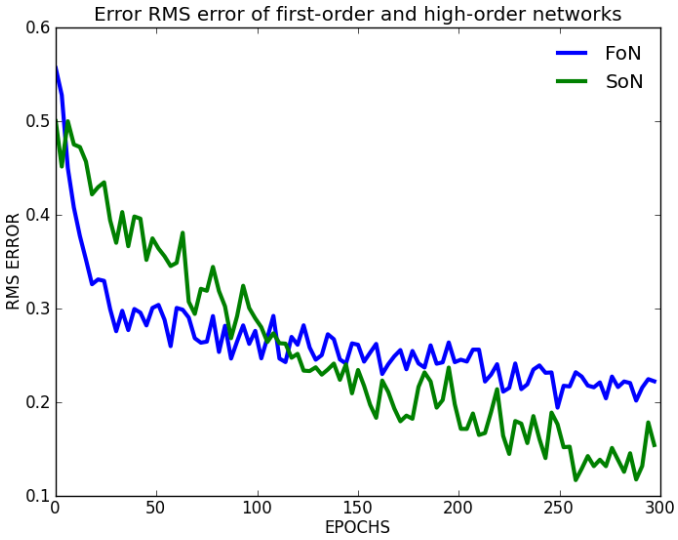


Notes
- la courbe rouge représentent le taux de paris hauts du second réseau

Conclusion
- dans les 2 cas, le premier réseau réussit à apprendre sa tâche de classification
- dans les 2 cas, le second réseau n'arrive pas à tirer parti de ses représentations, il se contente simplement de parier haut au bout d'un moment

Secondaires RMS





Notes
- formule utilisée pour RMS (cf. Formules RMS)

Conclusion Il n'y a plus de pique du second réseau comme dans les Expérience C1 et Expérience C2.

Conclusion

Lors du passage sur des données réelles non linéairement séparables, l'utilité du second réseau s'écroule, les représentations ne sont plus exploitées.

Pour tenter de résoudre ce problème, l'Expérience C4 augmente le nombre de couche du premier réseau.

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Fisher, R. (1988). Iris plants database. 150 (50 in each of three classes).

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience C4

Objectif

Reproduction et approfondissement des résultats de la première expérience 1 dans l'article Cleeremans Alex (2007).

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, en particulier sur des perceptrons multicouches.

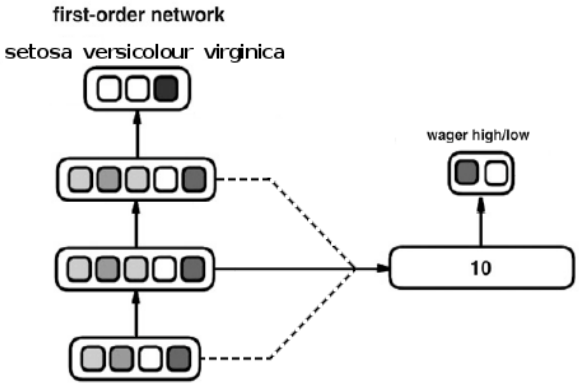
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

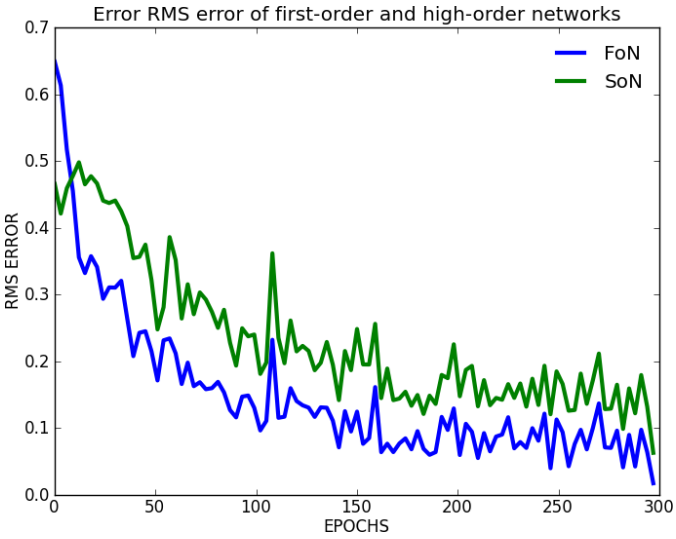


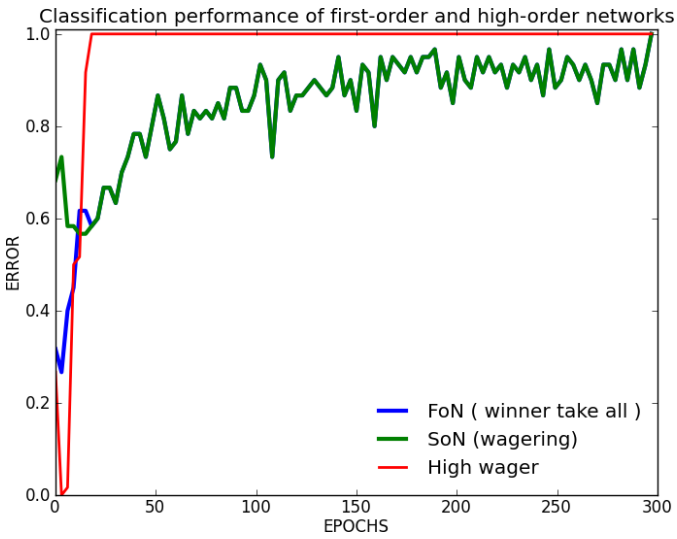
Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances





Notes

- les courbes SoN layer représentent les erreurs (du second réseaux) des couches du premier à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience D1

Objectif

Reproduction et approfondissement des résultats de la première expérience 1 dans l'article Cleeremans Alex (2007).

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, en particulier sur des perceptrons multicouches.

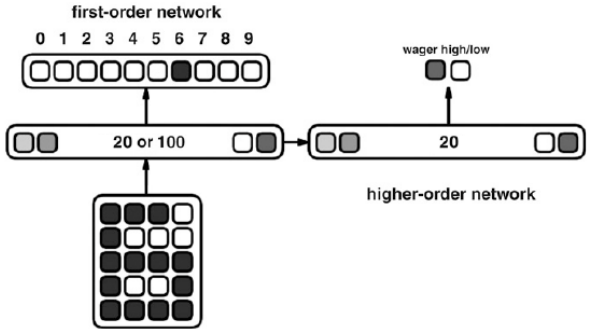
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

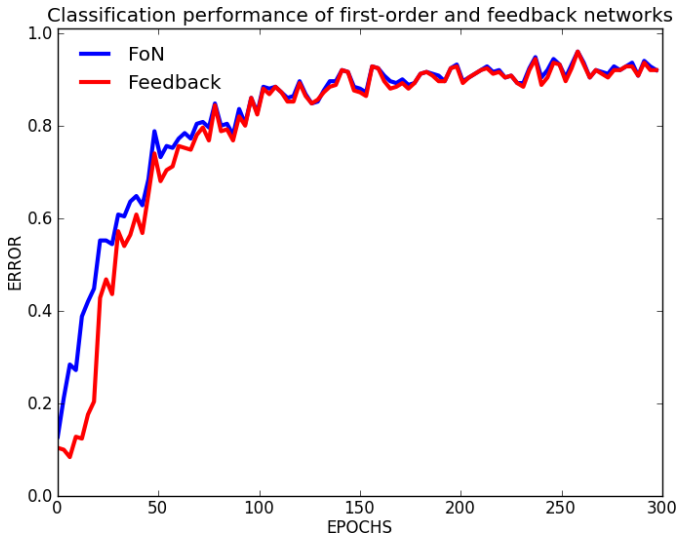


Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances



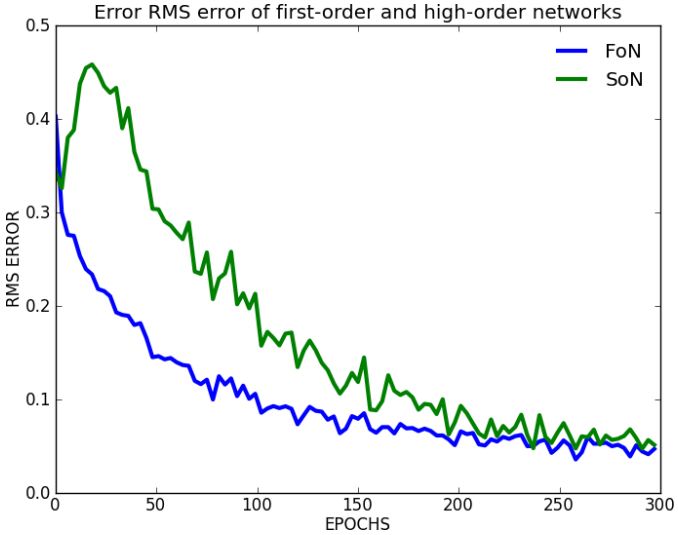
Notes

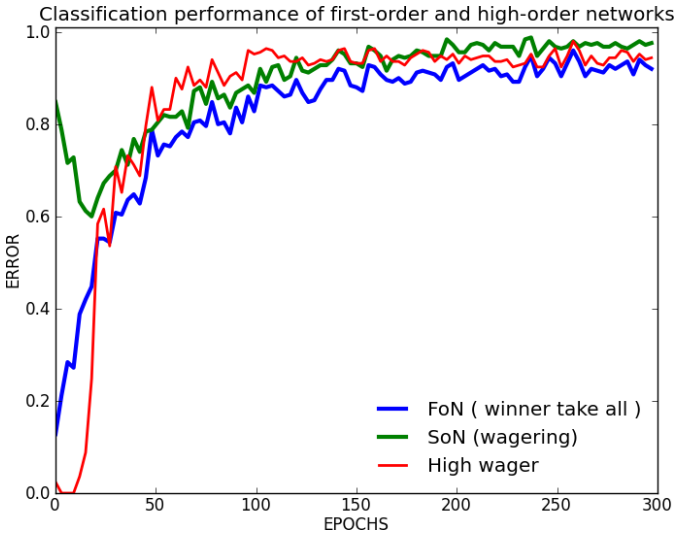
- les courbes SoN layer représentent les erreurs (du second réseaux) des couches du premier à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Secondaires RMS





Notes

- une couleur équivaut à un chiffre présenté
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes)

Conclusion Les neurones se stabilisent très rapidement (autour de la $50^{\text{ième}}$ époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience D2

Objectif

Reproduction et approfondissement des résultats de la première expérience 1 dans l'article Cleeremans Alex (2007).

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, en particulier sur des perceptrons multicouches.

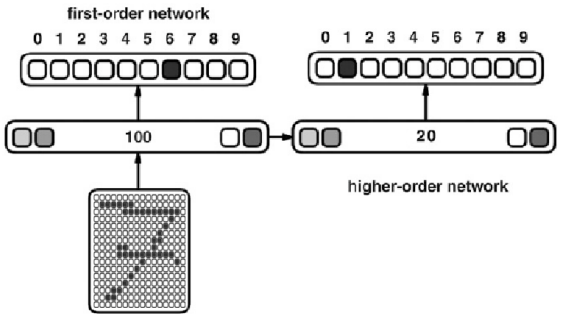
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

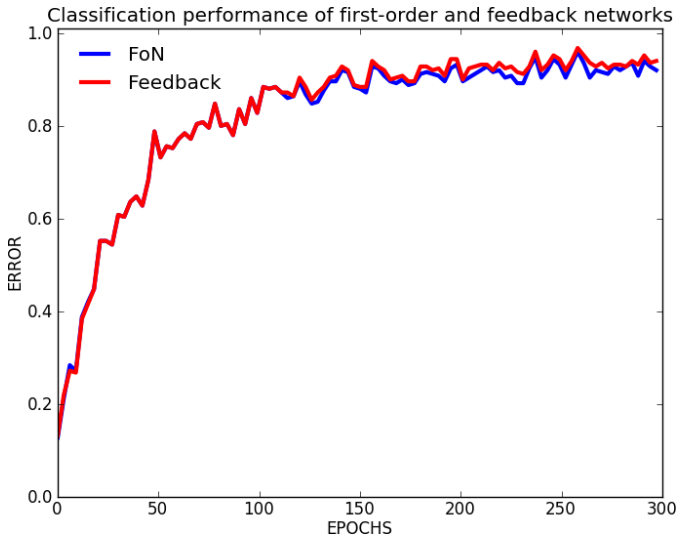


Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances



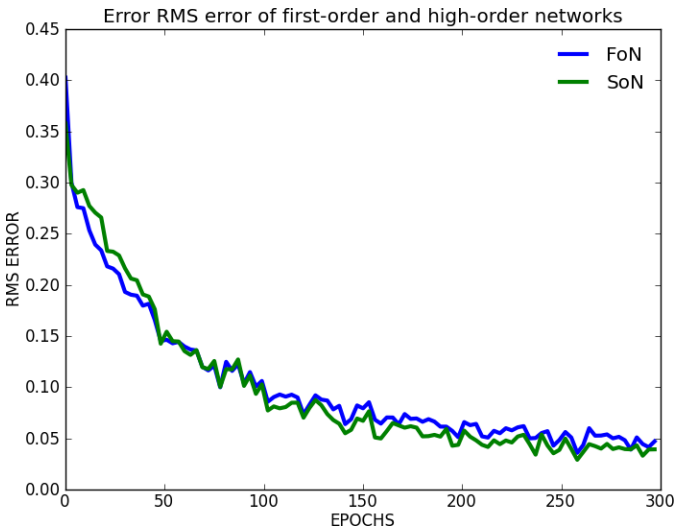
Notes

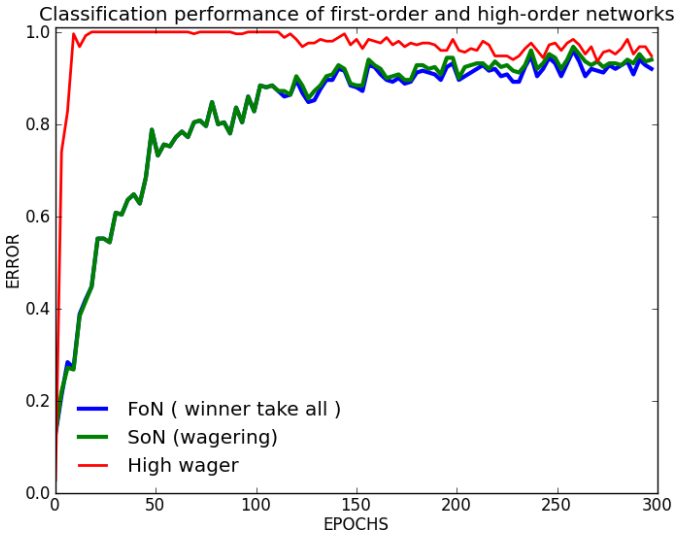
- les courbes SoN layer représentent les erreurs (du second réseaux) des couches du premier à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Secondaires RMS





Notes

- une couleur équivaut à un chiffre présenté
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes)

Conclusion Les neurones se stabilisent très rapidement (autour de la $50^{\text{ième}}$ époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience D3

Objectif

Reproduction et approfondissement des résultats de la première expérience 1 dans l'article Cleeremans Alex (2007).

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, en particulier sur des perceptrons multicouches.

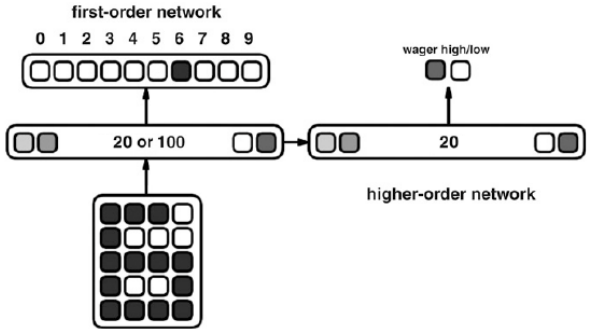
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

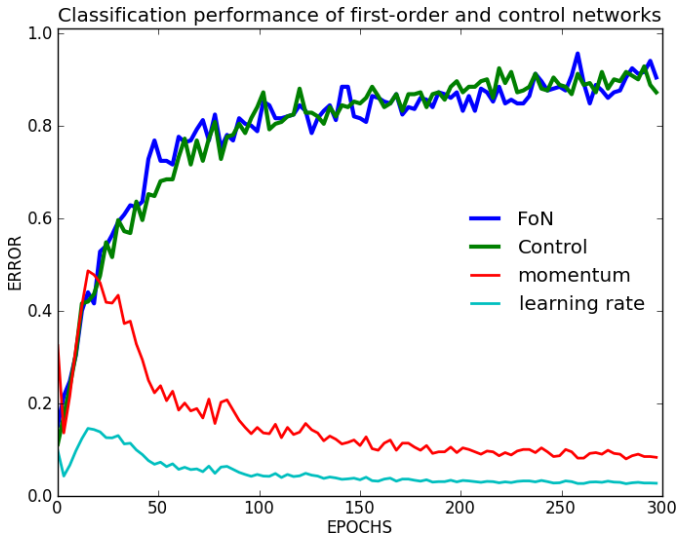


Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances



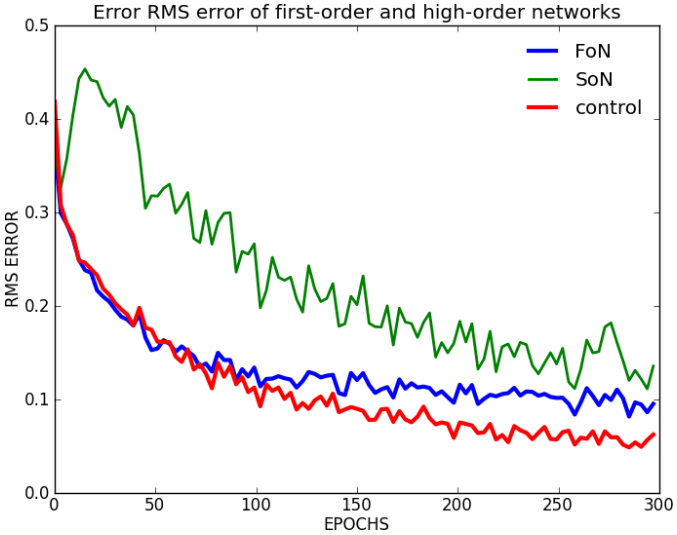
Notes

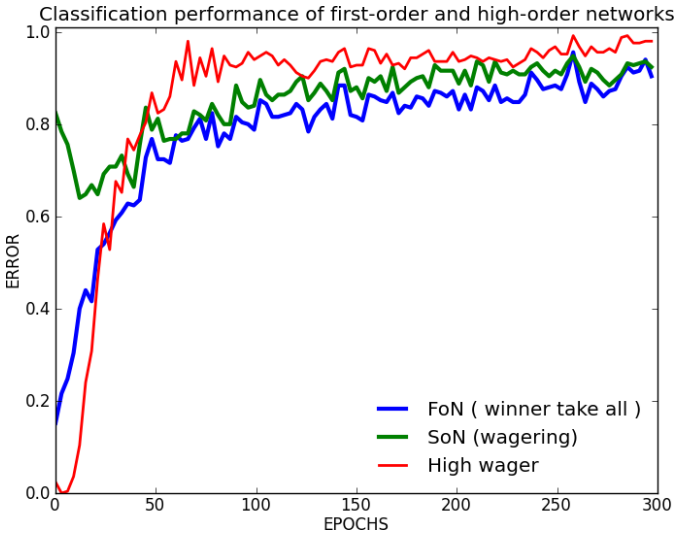
- les courbes SoN layer représentent les erreurs (du second réseaux) des couches du premier à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Secondaires RMS





Notes

- une couleur équivaut à un chiffre présenté
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes)

Conclusion Les neurones se stabilisent très rapidement (autour de la $50^{\text{ième}}$ époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.

Expérience D4

Objectif

Reproduction et approfondissement des résultats de la première expérience 1 dans l'article Cleeremans Alex (2007).

Comprendre de quelles manières peuvent émerger des représentations et méta-représentations dans un réseau de neurone connexionniste, en particulier sur des perceptrons multicouches.

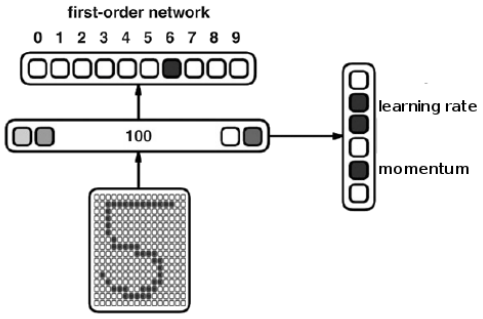
Architecture

Description Un premier réseau de perceptron multicouche apprend à discrétiser des chiffres représentés par 20 neurones d'entrées. Il est composé d'une couche cachée de 5 neurones.

Un second réseau de perceptron multicouche apprend à dupliquer toutes les couches du premier réseau en n'ayant que sa couche cachée en entrée.

L'apprentissage du second réseau, n'affecte pas les poids entre la couche d'entrée et la couche cachée du premier réseau.

Schéma

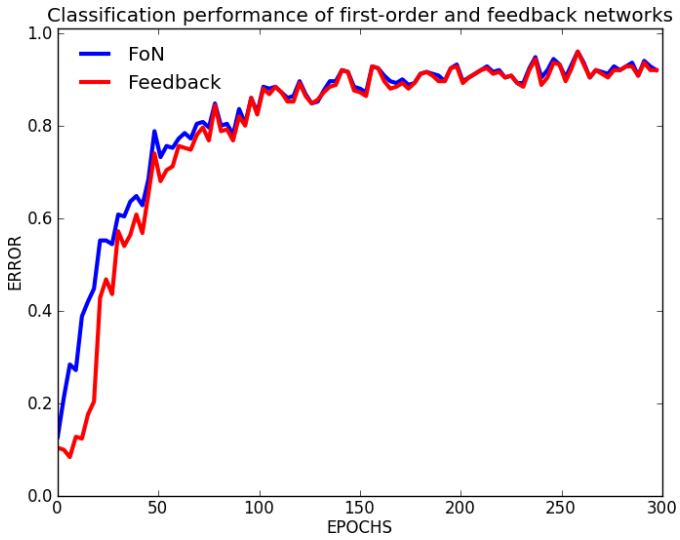


Paramètres

- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau
- 10 chiffres différents présentés
- apprentissage 10 (formes) x 1000 (époques)
- utilisation de biais
- poids initialisés sur [-0.25 ; 0.25]
- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

Résultats

Principaux Analyse des performances



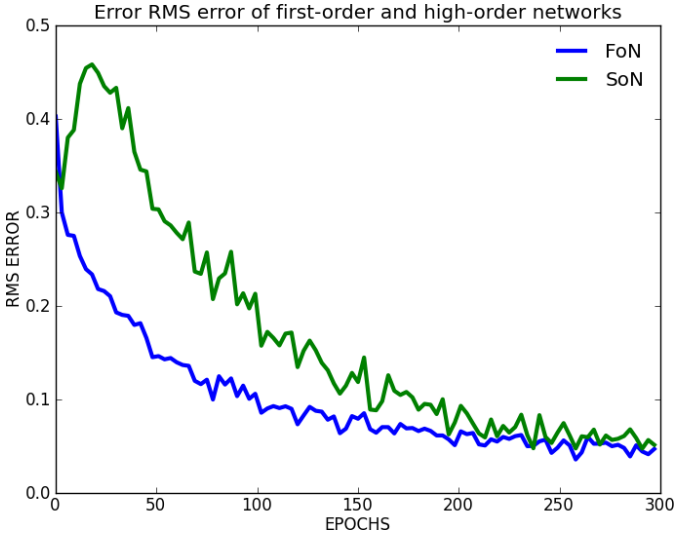
Notes

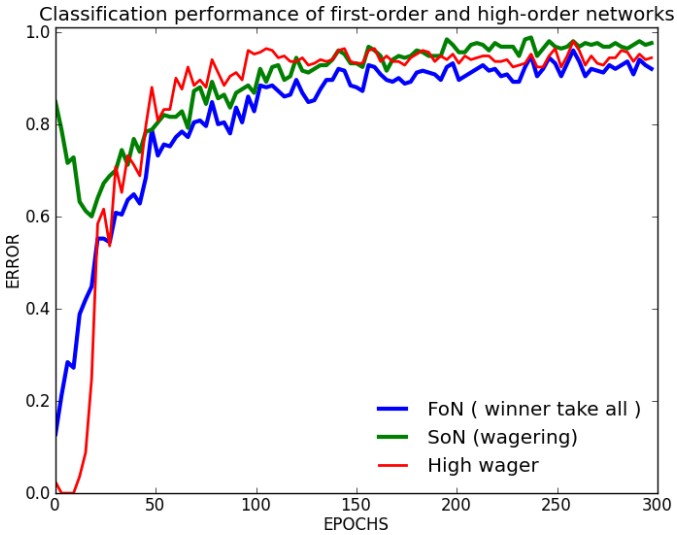
- les courbes SoN layer représentent les erreurs (du second réseaux) des couches du premier à reproduire
- la courbe RMS verte (SoN) est la somme des 3 courbes SoN layer

Conclusion

- la couche cachée et la couche de sortie ne posent aucun problèmes d'apprentissage
- les performances du second réseau dépendent principalement de sa capacité à reproduire les entrées
- le second réseau apprend plus rapidement que le premier

Secondaires RMS





Notes

- une couleur équivaut à un chiffre présenté
- une valeur discretisée correspond à un certain encodage de la couche cachée (cf Algorithmes)

Conclusion Les neurones se stabilisent très rapidement (autour de la $50^{\text{ième}}$ époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

Conclusion

Formules

RMS Pour une époque e :

$$rms_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_{i,e} - d_i)^2}$$

$$with \begin{cases} n : number\ of\ neurons\ on\ the\ output\ layer \\ o_{i,e} : value\ obtained\ for\ the\ i^{th}\ neuron\ at\ the\ e^{th}\ epoch \\ d_i : value\ desired\ for\ the\ i^{th}\ neuron \end{cases}$$

Discrétisation Pour la couche cachée $hiddenNeuron$ de n neurones, un neurone pouvant être encodé par $number\_cutting$ valeurs différentes :

$$\sum_{i=0}^{n} number\_cutting^i \times cutting(hiddenNeuron[i])$$

Exemple $400 \leftarrow [0\ ; 0,25]\ [0\ ; 0,25]\ [0,25\ ; 0,5]\ [0,5\ ; 0,75]\ [0,25\ ; 0,5]$
$400 \leftarrow 0 \times 4^0 + 0 \times 4^1 + 1 \times 4^2 + 2 \times 4^3 + 1 \times 4^4$

Descente de gradient Touzet (1992)
Construction de l'erreur :

$$y_i = f'(a_i) \times (d_i - x_i)\ si\ i\ neurone\ de\ sortie$$
$$y_i = f'(a_i) \times \sum_k (w_{ki} \times y_k)\ si\ i\ neurone\ cache$$

Mise à jour des poids :

$$w_{ij}(t+1) = w_{ij}(t) + learning\_rate \times y_i \times x_j + momentum \times (w_{ij}(t) - w_{ij}(t-1))$$

Variables :

$$\begin{cases} f : fonction\ sigmoide \\ x_i : valeur\ du\ neurone\ i \\ d_i : valeur\ desirepour\ le\ neurone\ i \\ a_i : somme\ pondere\ des\ poids\ du\ neurone\ i \end{cases}$$

References

Cleeremans Alex, Timmermans Bert, P. A. (2007). Consciousness and metarepresentation : A computational sketch. *doi :10.1016/j.neunet.2007.09.011.*

Touzet, C. (1992). Les réseaux de neurones artificiels - introduction au connexionnisme.