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

| Table de matière | 2 |
|------------------|----|
| Expérience A1 | 3 |
| Expérience A2 | 8 |
| Expérience A3 | 13 |
| Expérience A4 | 18 |
| Expérience B1 | 22 |
| Expérience B3 | 26 |
| Expérience C1 | 30 |
| Expérience C2 | 34 |
| Expérience C3 | 38 |
| Expérience C4 | 42 |
| Expérience D1 | 45 |
| Expérience D2 | 49 |
| Expérience D3 | 53 |
| Expérience D4 | 57 |

Expérience A1

Résumé

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

But

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

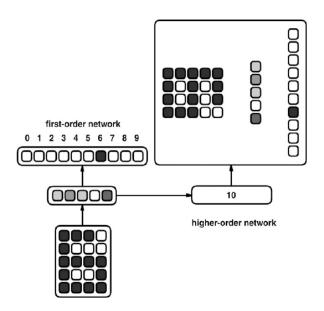
Architecture

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 - poids initialisés sur [-0.25; 0.25]

– 10 chiffres différents présentés

taux d'apprentissage constant
entrées valent 0 ou 1

– apprentissage 10 (formes) x 1000 (époques)

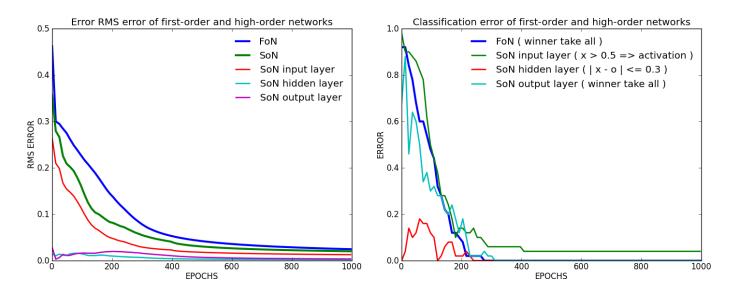
- sigmoïde à température 1

- utilisation de biais

EXPÉRIENCE A1 Résultats

Résultats

Principaux Analyse des performances



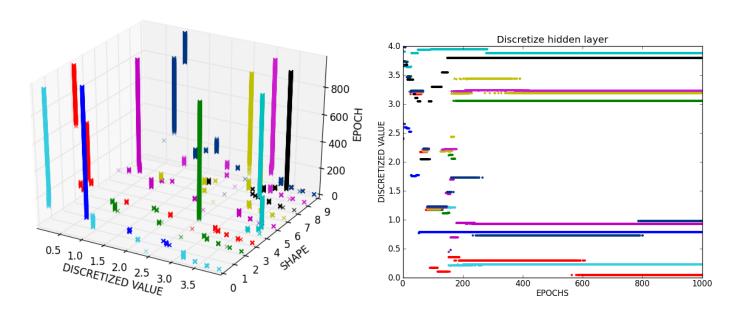
Notes

- 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 Discrétisation de la couche cachée du premier réseau



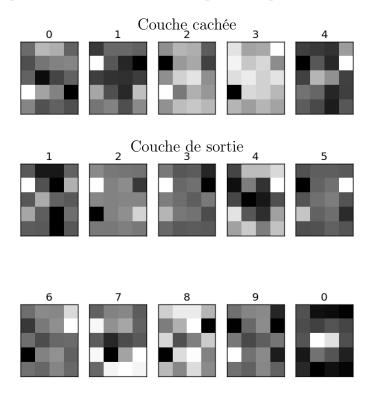
EXPÉRIENCE A1 Résultats

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^{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



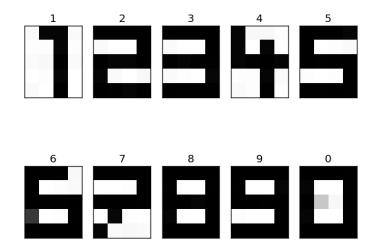
Notes

- 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

EXPÉRIENCE A1 Conclusion



Notes

_

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

EXPÉRIENCE A1 Formules

Formules

 \mathbf{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:

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

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

Résumé

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

But

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

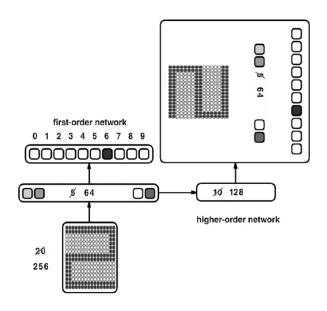
Architecture

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

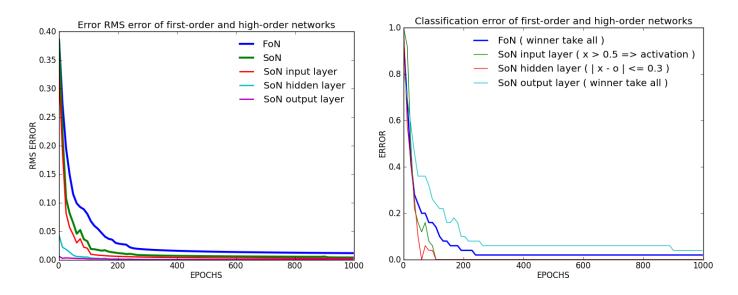
- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau poids initialisés sur [-0.25; 0.25]
- 10 chiffres différents présentés
- apprentissage 10 1000 (formes) \mathbf{X} (époques)
- utilisation de biais

- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

EXPÉRIENCE A2 Résultats

Résultats

Principaux Analyse des performances



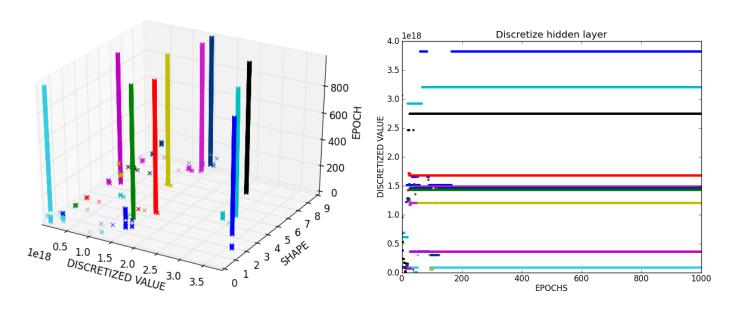
Notes

- 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 Discrétisation de la couche cachée du premier réseau



Notes

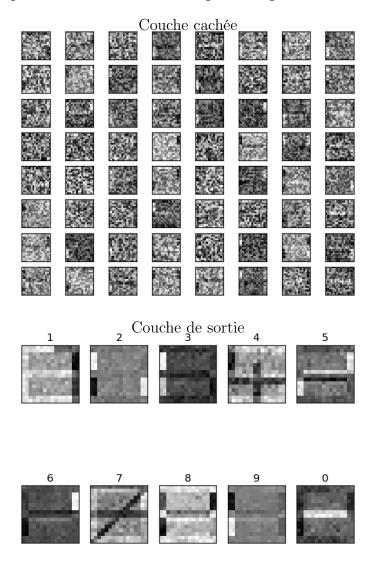
- une couleur équivaut à un chiffre présenté

EXPÉRIENCE A2 Résultats

– 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^{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

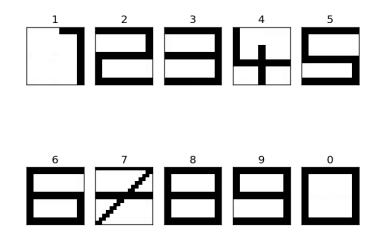


Notes

- 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

_

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

EXPÉRIENCE A2 Formules

Formules

 \mathbf{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:

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

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

Résumé

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

But

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

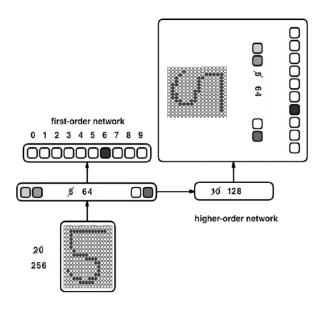
Architecture

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

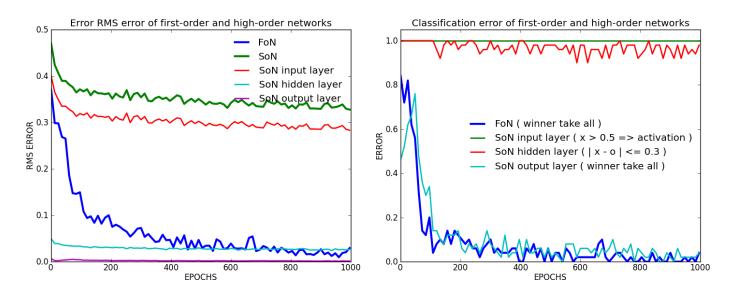
- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau poids initialisés sur [-0.25; 0.25]
- 10 chiffres différents présentés
- apprentissage 10 (formes) 1000 \mathbf{X} (époques)
- utilisation de biais

- taux d'apprentissage constant
- entrées valent 0 ou 1
- sigmoïde à température 1

EXPÉRIENCE A3 Résultats

Résultats

Principaux Analyse des performances



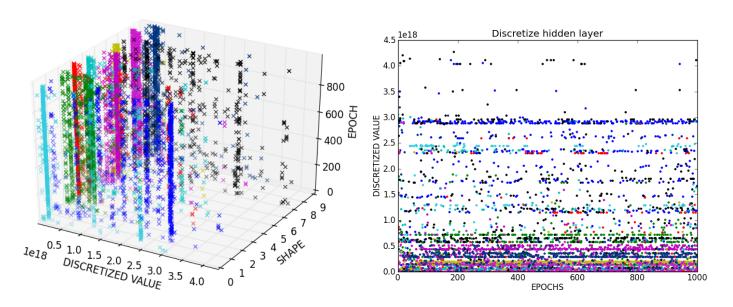
Notes

- 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 Discrétisation de la couche cachée du premier réseau



Notes

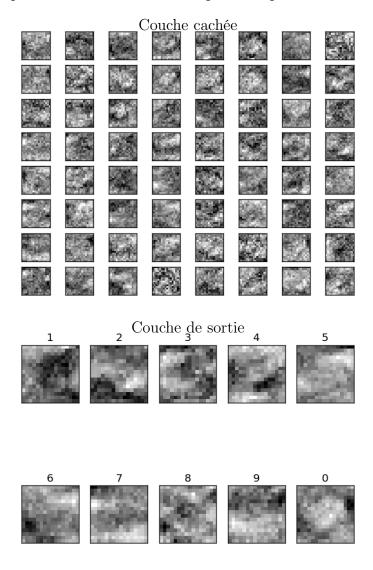
- une couleur équivaut à un chiffre présenté

EXPÉRIENCE A3 Résultats

– 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^{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

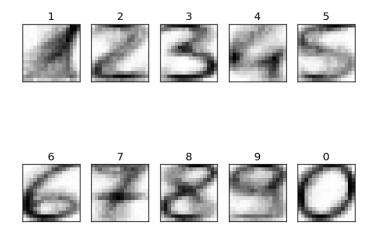


Notes

- 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

_

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

EXPÉRIENCE A3 Formules

Formules

 \mathbf{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:

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

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 A4

Résumé

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

But

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

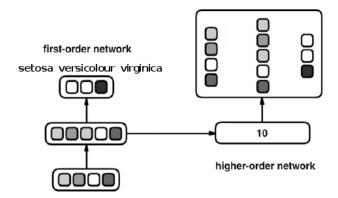
Architecture

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 poids initialisés sur [-0.25; 0.25]
- 10 chiffres différents présentés
 - e 10 (formes) x 1000 entrées valent 0 ou 1
- apprentissage 10 (formes) x 10 (époques)
- sigmoïde à température 1

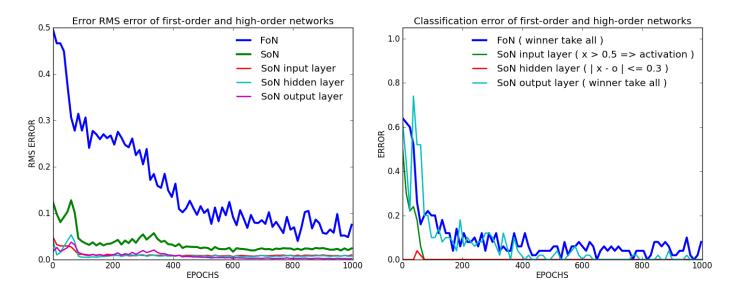
- taux d'apprentissage constant

utilisation de biais

EXPÉRIENCE A4 Résultats

Résultats

Principaux Analyse des performances



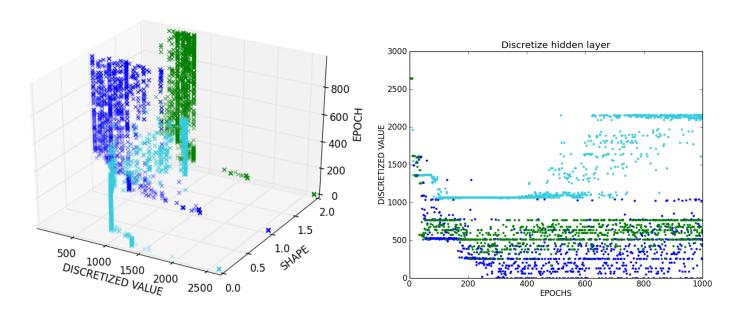
Notes

- 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 Discrétisation de la couche cachée du premier réseau



EXPÉRIENCE A4 Conclusion

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^{ième} époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

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

EXPÉRIENCE A4 Formules

Formules

 \mathbf{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:

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

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 B1

Résumé

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

But

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

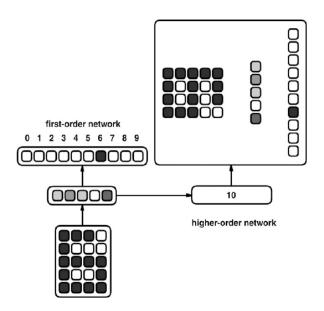
Architecture

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 - poids initialisés sur [-0.25; 0.25]

- 10 chiffres différents présentés - taux d'apprentissage constant

– apprentissage 10 (formes) x 1000 – entrées valent 0 ou 1

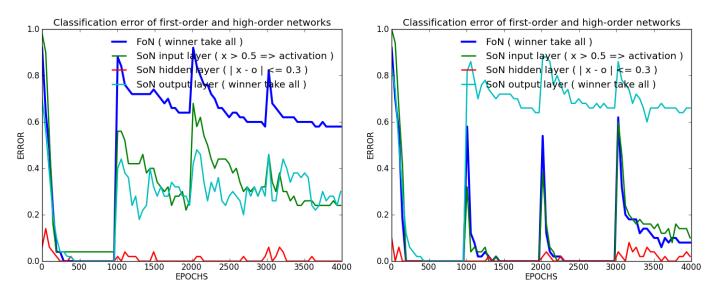
(époques) – sigmoïde à température 1

- utilisation de biais

EXPÉRIENCE B1 Résultats

Résultats

Principaux Analyse des performances



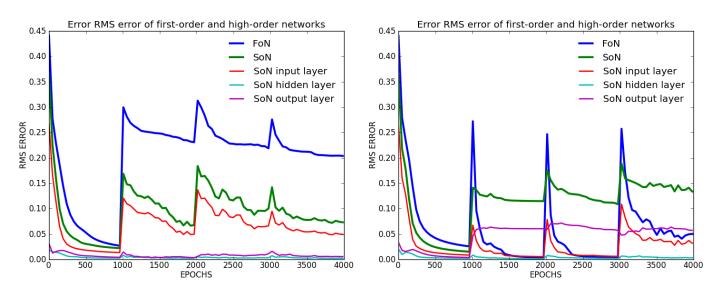
Notes

- 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



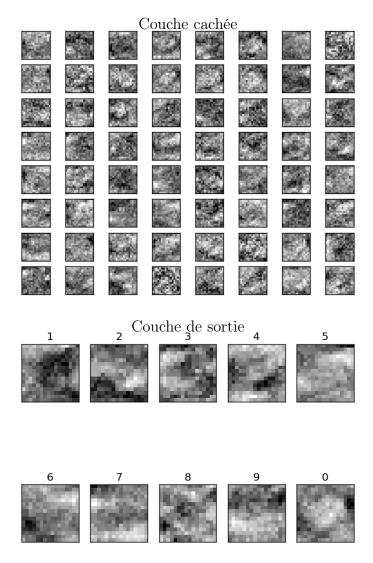
Notes

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

EXPÉRIENCE B1 Conclusion

Conclusion Les neurones se stabilisent très rapidement (autour de la 50^{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.

EXPÉRIENCE B1 Formules

Formules

 \mathbf{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:

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

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

Résumé

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

But

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

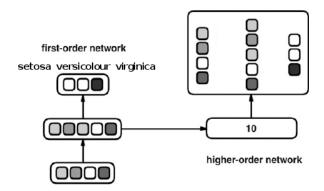
Architecture

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

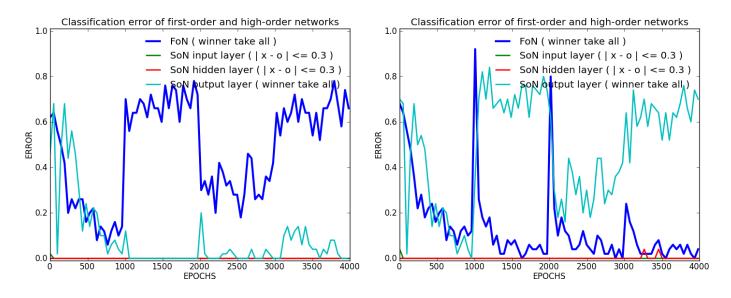
- momentum : 0.9 sur les 2 réseau
- taux d'apprentissage : 0.1 sur les 2 réseau poids initialisés sur [-0.25; 0.25]
- 10 chiffres différents présentés
- taux d'apprentissage constant
- apprentissage 10 (formes) (époques)
- 1000 entrées valent 0 ou 1 - sigmoïde à température 1

- utilisation de biais

EXPÉRIENCE B3 Résultats

Résultats

Principaux Analyse des performances



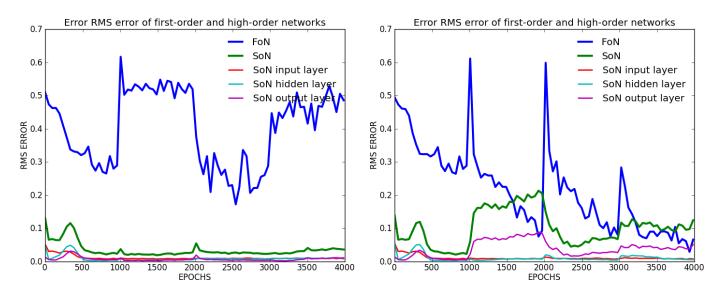
Notes

- 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

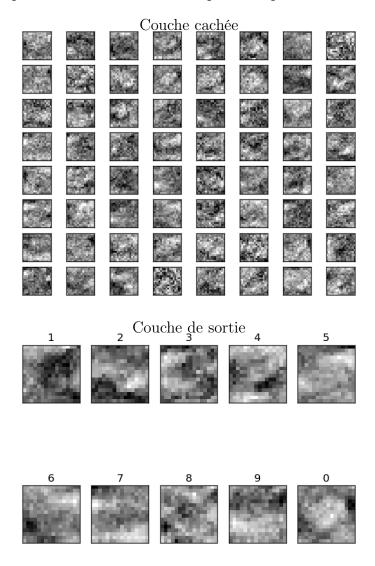
– une couleur équivaut à un chiffre présenté

EXPÉRIENCE B3 Conclusion

– 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^{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.

EXPÉRIENCE B3 Formules

Formules

 \mathbf{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:

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

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

Résumé

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

But

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

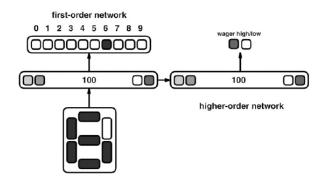
Architecture

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 poids initialisés sur [-0.25; 0.25]
- 10 chiffres différents présentés
- taux d'apprentissage constant
- apprentissage 10 (formes) x 1000
- entrées valent 0 ou 1

(époques)

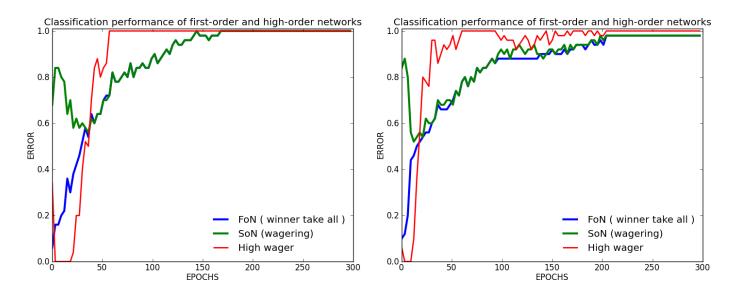
sigmoïde à température 1

- utilisation de biais

EXPÉRIENCE C1 Résultats

Résultats

Principaux Analyse des performances



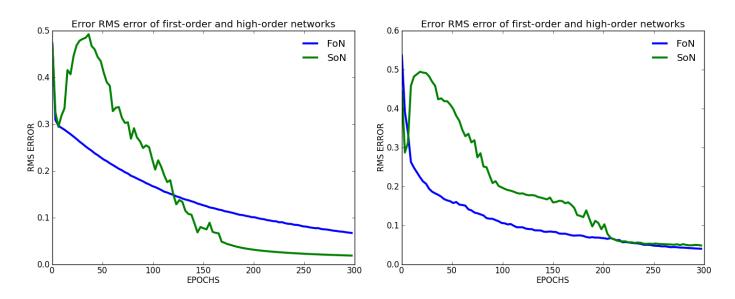
Notes

- 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

EXPÉRIENCE C1 Conclusion

– 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^{ième} époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

EXPÉRIENCE C1 Formules

Formules

 \mathbf{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:

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

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

Résumé

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

But

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

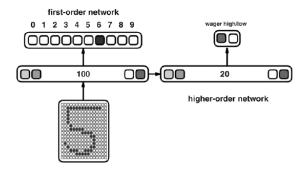
Architecture

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
- poids initialisés sur [-0.25; 0.25]
- 10 chiffres différents présentés
- taux d'apprentissage constant
- apprentissage 10 (formes) x 1000 (époques)
- sigmoïde à température 1

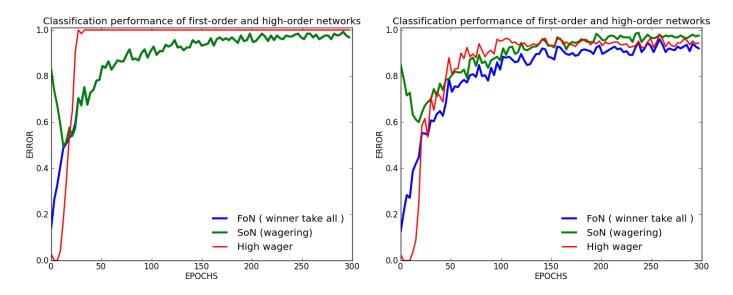
– entrées valent 0 ou 1

- utilisation de biais

EXPÉRIENCE C2 Résultats

Résultats

Principaux Analyse des performances



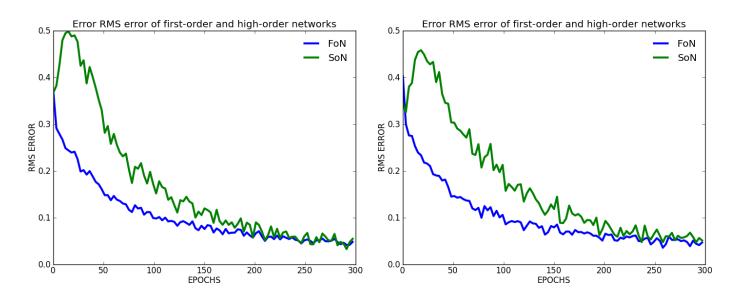
Notes

- 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

EXPÉRIENCE C2 Conclusion

– 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.

EXPÉRIENCE C2 Formules

Formules

 \mathbf{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:

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

References

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

Expérience C3

Résumé

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

But

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

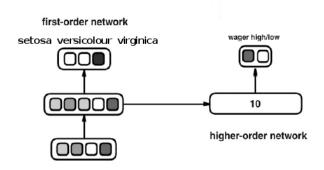
Architecture

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



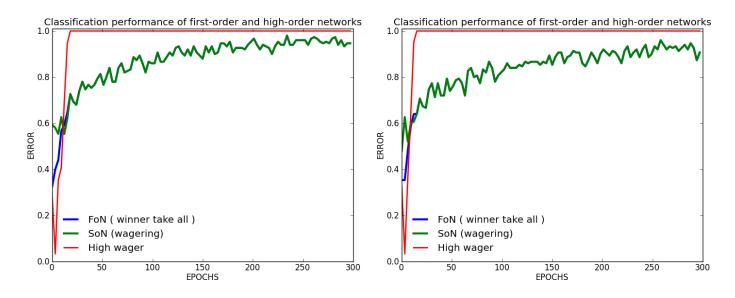
Paramètres

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

EXPÉRIENCE C3 Résultats

Résultats

Principaux Analyse des performances



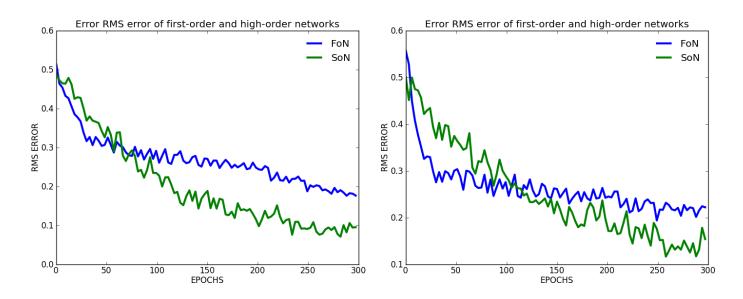
Notes

- 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

EXPÉRIENCE C3 Conclusion

– 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^{ième} époque en moyenne), le tout permettant au second réseau d'avoir des entrées très peu variables, favorisant son apprentissage.

EXPÉRIENCE C3 Formules

Formules

 \mathbf{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:

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

References

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

Expérience C4

Résumé

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

But

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

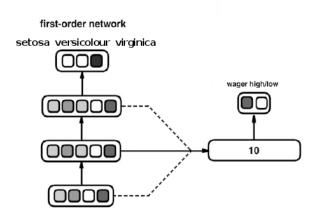
Architecture

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



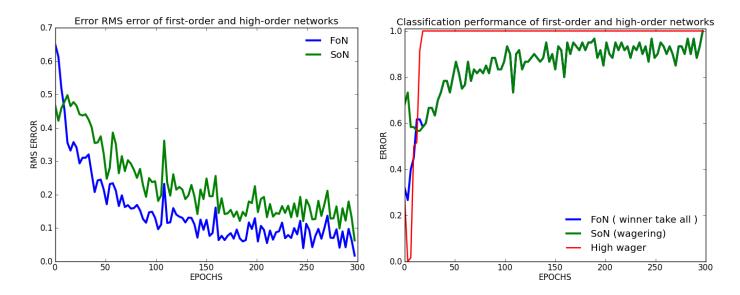
Paramètres

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

EXPÉRIENCE C4 Résultats

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

EXPÉRIENCE C4 Formules

Formules

 \mathbf{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:

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

References

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

Expérience D1

Résumé

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

But

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

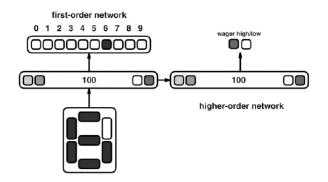
Architecture

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 poids initialisés sur [-0.25; 0.25]
- 10 chiffres différents présentés
- taux d'apprentissage constant
- apprentissage 10 (formes) x 1000 (époques)
- entrées valent 0 ou 1

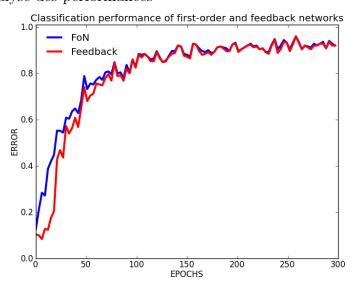
- sigmoïde à température 1

- utilisation de biais

EXPÉRIENCE D1 Résultats

Résultats

Principaux Analyse des performances



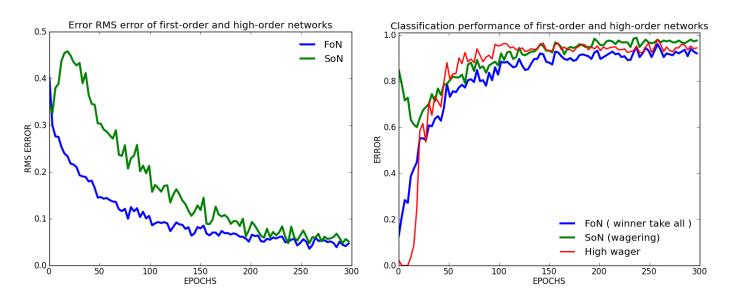
Notes

- 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)

EXPÉRIENCE D1 Conclusion

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

EXPÉRIENCE D1 Formules

Formules

 \mathbf{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:

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

References

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

Expérience D2

Résumé

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

But

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

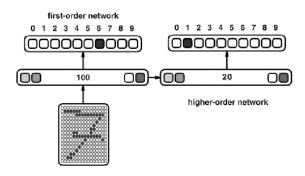
Architecture

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 poids initialisés sur [-0.25; 0.25]
- 10 chiffres différents présentés
- taux d'apprentissage constant
- apprentissage 10 (formes) 1000 \mathbf{X} (époques)
- sigmoïde à température 1

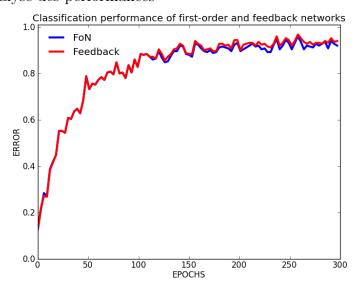
– entrées valent 0 ou 1

- utilisation de biais

EXPÉRIENCE D2 Résultats

Résultats

Principaux Analyse des performances



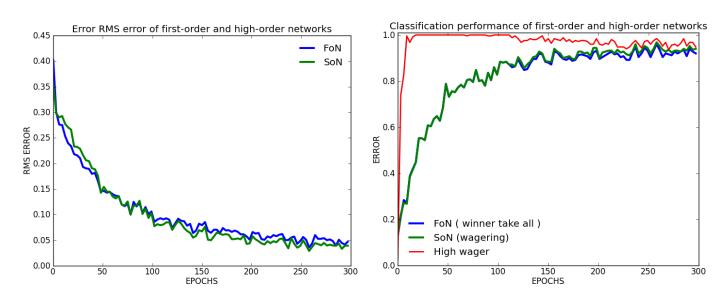
Notes

- 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)

EXPÉRIENCE D2 Conclusion

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

EXPÉRIENCE D2 Formules

Formules

 \mathbf{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:

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

References

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

Expérience D3

Résumé

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

But

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

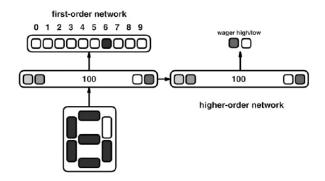
Architecture

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



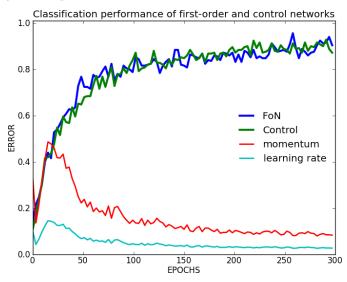
Paramètres

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

EXPÉRIENCE D3 Résultats

Résultats

Principaux Analyse des performances



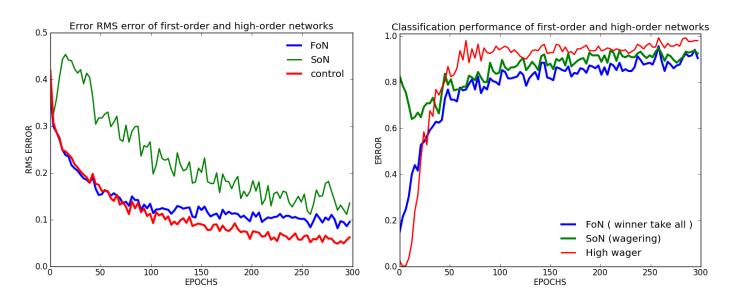
Notes

- 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)

EXPÉRIENCE D3 Conclusion

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

EXPÉRIENCE D3 Formules

Formules

 \mathbf{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:

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

References

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

Expérience D4

Résumé

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

But

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

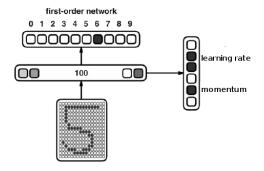
Architecture

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
- poids initialisés sur $[-0.25\,;\,0.25]$
- 10 chiffres différents présentés
- taux d'apprentissage constant
- apprentissage 10 (formes) x 1000 (époques)
- sigmoïde à température 1

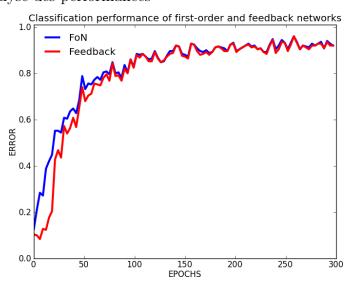
– entrées valent 0 ou 1

- utilisation de biais

EXPÉRIENCE D4 Résultats

Résultats

Principaux Analyse des performances



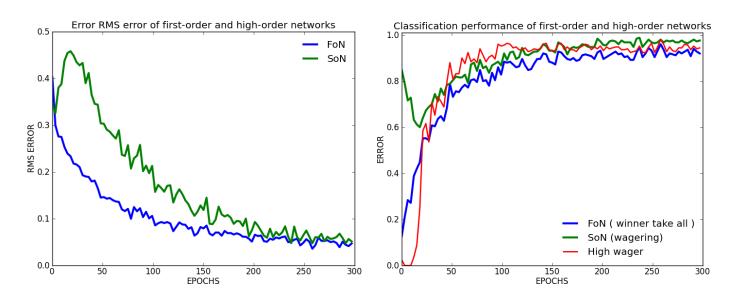
Notes

- 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)

EXPÉRIENCE D4 Conclusion

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

EXPÉRIENCE D4 Formules

Formules

 \mathbf{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:

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

References

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