

MIND 1024

Histoire de l'Informatique

Eoin Brereton Hurley

Université Grenoble Alpes

30/06/2022



- 1 Introduction
- 2 Réseau de Hopfield plus en profondeur
- 3 Simulateur
- 4 Réseaux Neuronaux Aujourd'hui
- 5 Conclusion

1 Introduction

Aperçu de la Machine

Contexte Historique

2 Réseau de Hopfield plus en profondeur

3 Simulateur

4 Réseaux Neuronaux Aujourd'hui

5 Conclusion

1 Introduction

Aperçu de la Machine

Contexte Historique

2 Réseau de Hopfield plus en profondeur

3 Simulateur

4 Réseaux Neuronaux Aujourd'hui

5 Conclusion

Machine Neuronale MIND 1024

- MIND signifie *Machine à Interaction Neuronale Démodulée*



Figure 1 – Détail 1 du MIND 1024



Figure 2 – Exemple de Carte Mère du MIND 1024



Figure 3 – Vue d'Ensemble du MIND 1024

Machine Neuronale MIND 1024

- MIND signifie *Machine à Interaction Neuronale Démodulée*
- Cette machine simule un réseau de *Hopfield*, (il s'agit d'un type spécifique du réseau de neurones récurrents)



Figure 1 – Détail 1 du MIND 1024



Figure 2 – Exemple de Carte Mère du MIND 1024



Figure 3 – Vue d'Ensemble du MIND 1024

Machine Neuronale MIND 1024

- MIND signifie *Machine à Interaction Neuronale Démodulée*
- Cette machine simule un réseau de *Hopfield*, (il s'agit d'un type spécifique du réseau de neurones récurrents)
- Le réseau consiste en 1024 neurones interconnectés mis en œuvre à l'aide de 1 048 576 synapses.



Figure 3 – Vue d'Ensemble du MIND 1024



Figure 1 – Détail 1 du MIND 1024



Figure 2 – Exemple de Carte Mère du MIND 1024

Machine Neuronale MIND 1024

- MIND signifie *Machine à Interaction Neuronale Démodulée*
- Cette machine simule un réseau de *Hopfield*, (il s'agit d'un type spécifique du réseau de neurones récurrents)
- Le réseau consiste en 1024 neurones interconnectés mis en œuvre à l'aide de 1 048 576 synapses.
- Précisions sur le réseau *Hopfield* plus tard...



Figure 3 – Vue d'Ensemble du MIND 1024



Figure 1 – Détail 1 du MIND 1024



Figure 2 – Exemple de Carte Mère du MIND 1024

Machine Neuronale MIND 1024

- MIND signifie *Machine à Interaction Neuronale Démodulée*
- Cette machine simule un réseau de *Hopfield*, (il s'agit d'un type spécifique du réseau de neurones récurrents)
- Le réseau consiste en 1024 neurones interconnectés mis en œuvre à l'aide de 1 048 576 synapses.
- Précisions sur le réseau *Hopfield* plus tard...
- **Maintenant : du contexte historique !**



Figure 3 – Vue d'Ensemble du MIND 1024



Figure 1 – Détail 1 du MIND 1024



Figure 2 – Exemple de Carte Mère du MIND 1024

1 Introduction

Aperçu de la Machine

Contexte Historique

Alan Turing

Perceptron

John Hopfield

2 Réseau de Hopfield plus en profondeur

3 Simulateur

4 Réseaux Neuronaux Aujourd'hui

5 Conclusion

Machine Neuronale MIND 1024

- Problèmes avec les ordinateurs au 20e siècle :

Machine Neuronale MIND 1024

- Problèmes avec les ordinateurs au 20e siècle :
 - Bien qu'ils fassent très bien les calculs, à l'époque ils avaient de nombreux défauts.

Machine Neuronale MIND 1024

- Problèmes avec les ordinateurs au 20e siècle :
 - Bien qu'ils fassent très bien les calculs, à l'époque ils avaient de nombreux défauts.
- Par exemple, tâches triviales pour les humains, mais difficiles pour les ordinateurs :

Machine Neuronale MIND 1024

- Problèmes avec les ordinateurs au 20e siècle :
 - Bien qu'ils fassent très bien les calculs, à l'époque ils avaient de nombreux défauts.
- Par exemple, tâches triviales pour les humains, mais difficiles pour les ordinateurs :
 - **Reconnaissance du visage,**

Machine Neuronale MIND 1024

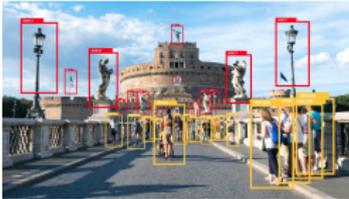
- Problèmes avec les ordinateurs au 20e siècle :
 - Bien qu'ils fassent très bien les calculs, à l'époque ils avaient de nombreux défauts.
- Par exemple, tâches triviales pour les humains, mais difficiles pour les ordinateurs :
 - Reconnaissance du visage,
 - **Identification des images,**

Machine Neuronale MIND 1024

- Problèmes avec les ordinateurs au 20e siècle :
 - Bien qu'ils fassent très bien les calculs, à l'époque ils avaient de nombreux défauts.
- Par exemple, tâches triviales pour les humains, mais difficiles pour les ordinateurs :
 - Reconnaissance du visage,
 - Identification des images,
 - Compréhension de la voix, etc.

Machine Neuronale MIND 1024

- Problèmes avec les ordinateurs au 20e siècle :
 - Bien qu'ils fassent très bien les calculs, à l'époque ils avaient de nombreux défauts.
- Par exemple, tâches triviales pour les humains, mais difficiles pour les ordinateurs :
 - Reconnaissance du visage,
 - Identification des images,
 - Compréhension de la voix, etc.



Machine Neuronale MIND 1024

- En 1950, (35 ans avant la construction du MIND 1024)...



Figure 4 – Alan Turing

Machine Neuronale MIND 1024

- En 1950, (35 ans avant la construction du MIND 1024)...
- *Alan Turing* publie un article fondateur sur l'intelligence des machines à calcul.



Figure 4 – Alan Turing

Machine Neuronale MIND 1024

- En 1950, (35 ans avant la construction du MIND 1024)...
- *Alan Turing* publie un article fondateur sur l'intelligence des machines à calcul.
- Il s'agit de l'article dans lequel il a introduit le *test de Turing*, (initialement, le *jeu d'imitation*).



Figure 4 – Alan Turing

Machine Neuronale MIND 1024



Figure 5 – Perceptron

- En 1958 : construction du *Perceptron* par *Frank Rosenblatt*.



Figure 6 – *Frank Rosenblatt* et la machine

Machine Neuronale MIND 1024

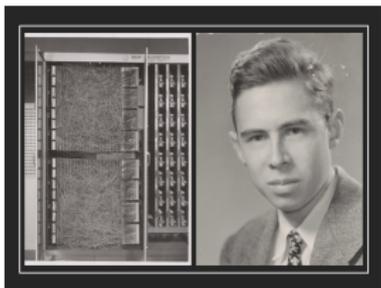


Figure 5 – Perceptron



Figure 6 – Frank Rosenblatt et la machine

- En 1958 : construction du *Perceptron* par *Frank Rosenblatt*.
- La machine visait à reconnaître des images.

Machine Neuronale MIND 1024



Figure 5 – Perceptron



Figure 6 – Frank Rosenblatt et la machine

- En 1958 : construction du *Perceptron* par *Frank Rosenblatt*.
- La machine visait à reconnaître des images.
- *Warren McCulloch et Walter Pitts* l'avait concpetualisée en 1943.

Machine Neuronale MIND 1024



Figure 5 – Perceptron



Figure 6 – Frank Rosenblatt et la machine

- En 1958 : construction du *Perceptron* par *Frank Rosenblatt*.
- La machine visait à reconnaître des images.
- *Warren McCulloch* et *Walter Pitts* l'avait conceptualisée en 1943.
- Au début, *Perceptron* avait du succès, mais cela se serait tari sous peu.

Machine Neuronale MIND 1024

- En 1982 *John Hopfield* développe un réseau de neurones récurrents.



Figure 8 – John Hopfield



Figure 9 – Le MIND 1024 à ACONIT

1 Introduction

2 Réseau de Hopfield plus en profondeur

En quoi ça consiste ?

Règle de mise à jour

Petit exemple

Règle de Hebb

3 Simulateur

4 Réseaux Neuronaux Aujourd'hui

5 Conclusion

1 Introduction

2 Réseau de Hopfield plus en profondeur

En quoi ça consiste ?

Règle de mise à jour

Petit exemple

Règle de Hebb

3 Simulateur

4 Réseaux Neuronaux Aujourd'hui

5 Conclusion

Machine Neuronale MIND 1024

- Réseau de Hopfield : composé de neurones tous interconnectés.

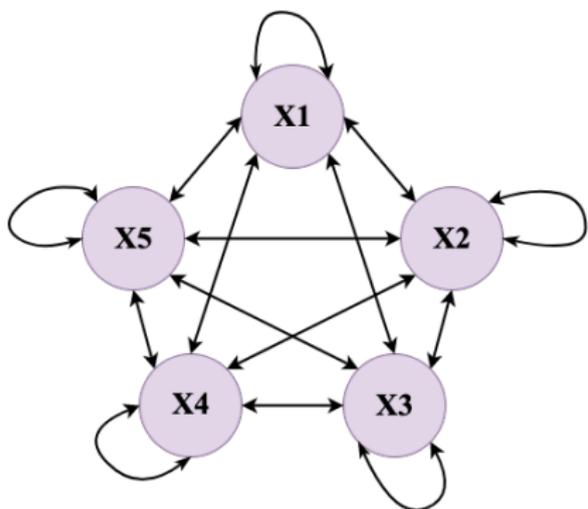


Figure 10 – Réseau de 5 neurones

Matrice de Connexions :

	<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>	<i>X5</i>
<i>X1</i>	0	w_a	w_b	w_c	w_d
<i>X2</i>	w_a	0	w_e	w_f	w_g
<i>X3</i>	w_b	w_e	0	w_h	w_i
<i>X4</i>	w_c	w_f	w_h	0	w_j
<i>X5</i>	w_d	w_g	w_i	w_j	0

$$w_{ij} = w_{ji} \in \{-1, 1\}, w_{ii} = 0$$

1 Introduction

2 Réseau de Hopfield plus en profondeur

En quoi ça consiste ?

Règle de mise à jour

Petit exemple

Règle de Hebb

3 Simulateur

4 Réseaux Neuronaux Aujourd'hui

5 Conclusion

Machine Neuronale MIND 1024

- Règle de mise à jour des états x_i :

Machine Neuronale MIND 1024

- Règle de mise à jour des états x_i :
- $x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij}x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$

Machine Neuronale MIND 1024

- Règle de mise à jour des états x_i :
- $$x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij}x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$$
- Effet de cette règle :

Machine Neuronale MIND 1024

- Règle de mise à jour des états x_i :
- $$x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij}x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$$
- Effet de cette règle :
 - Si $w_{ij} > 0$, x_i et x_j vont **converger**.

Machine Neuronale MIND 1024

- Règle de mise à jour des états x_i :
- $$x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij}x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$$
- Effet de cette règle :
 - Si $w_{ij} > 0$, x_i et x_j vont **converger**.
 - Si $w_{ij} < 0$, x_i et x_j vont **diverger**.

1 Introduction

2 Réseau de Hopfield plus en profondeur

En quoi ça consiste ?

Règle de mise à jour

Petit exemple

Règle de Hebb

3 Simulateur

4 Réseaux Neuronaux Aujourd'hui

5 Conclusion

Machine Neuronale MIND 1024

- Petit Exemple :

Machine Neuronale MIND 1024

- Petit Exemple :
- On a un motif, (c'est-à-dire un ensemble d'états).

Machine Neuronale MIND 1024

- Petit Exemple :
- On a un motif, (c'est-à-dire un ensemble d'états).
- On veut que le réseau puisse retrouver ce motif.

Machine Neuronale MIND 1024

- Petit Exemple :
- On a un motif, (c'est-à-dire un ensemble d'états).
- On veut que le réseau puisse retrouver ce motif.
- Alors, on peut fixer les valeurs des poids des connexions telles que :

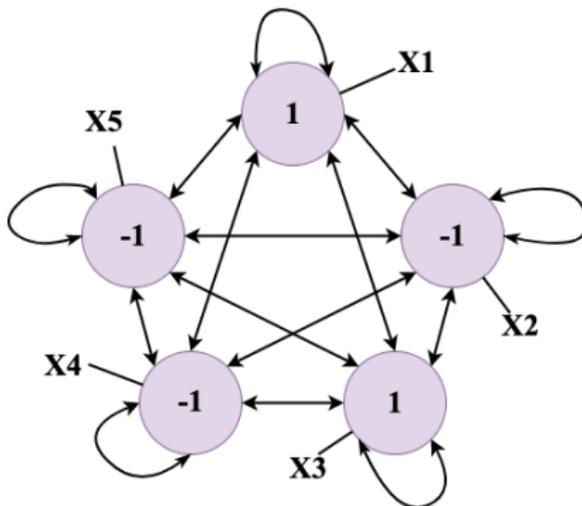
Machine Neuronale MIND 1024

- Petit Exemple :
- On a un motif, (c'est-à-dire un ensemble d'états).
- On veut que le réseau puisse retrouver ce motif.
- Alors, on peut fixer les valeurs des poids des connexions telles que :
 - $w_{ij} > 0$ pour $x_i = x_j$

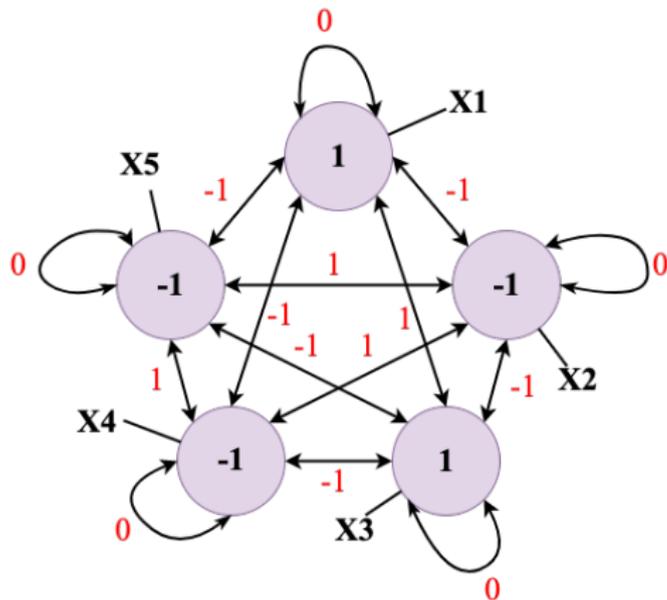
Machine Neuronale MIND 1024

- Petit Exemple :
- On a un motif, (c'est-à-dire un ensemble d'états).
- On veut que le réseau puisse retrouver ce motif.
- Alors, on peut fixer les valeurs des poids des connexions telles que :
 - $w_{ij} > 0$ pour $x_i = x_j$
 - $w_{ij} < 0$ pour $x_i \neq x_j$

Machine Neuronale MIND 1024



Machine Neuronale MIND 1024

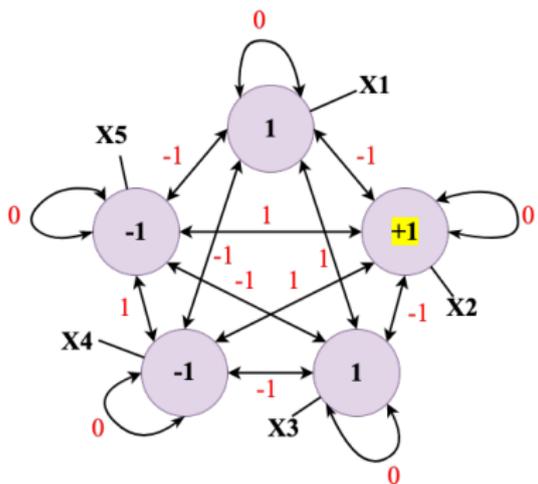


Matrice de Connexions :

	$X1$	$X2$	$X3$	$X4$	$X5$
$X1$	0	-1	1	-1	-1
$X2$	-1	0	-1	1	1
$X3$	1	-1	0	-1	-1
$X4$	-1	1	-1	0	1
$X5$	-1	1	-1	1	0

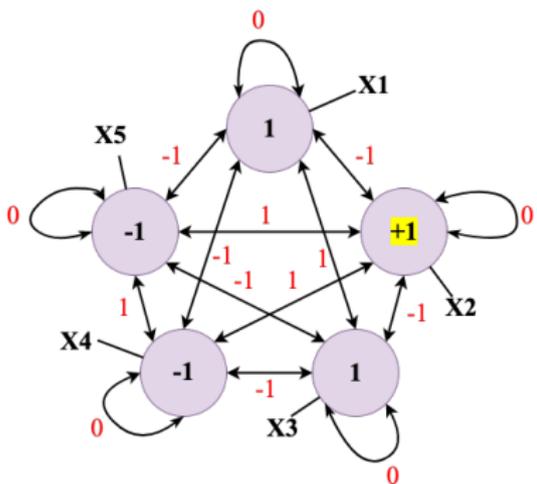
Machine Neuronale MIND 1024

- Après avoir décidé les poids, on peut changer la valeur d'un état...



Machine Neuronale MIND 1024

- Après avoir décidé les poids, on peut changer la valeur d'un état...
- ..puis, on peut appliquer la règle de tout à l'heure :

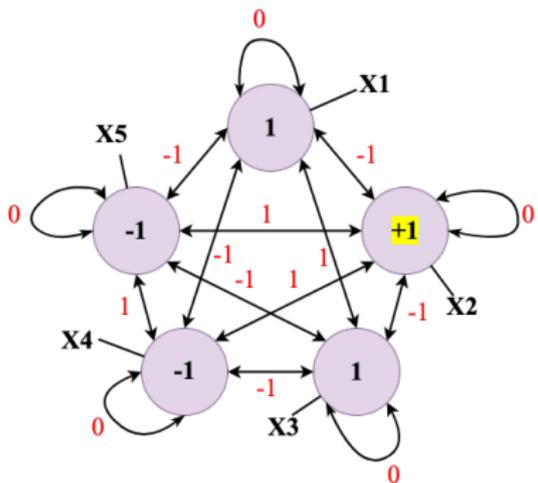


Machine Neuronale MIND 1024

- Après avoir décidé les poids, on peut changer la valeur d'un état...

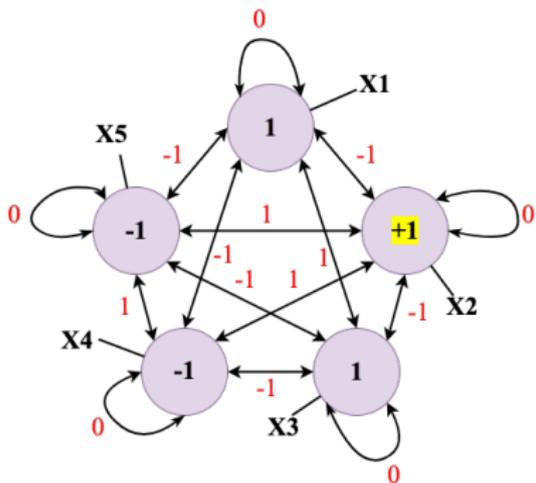
- ..puis, on peut appliquer la règle de tout à l'heure :

- $x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij}x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$



Machine Neuronale MIND 1024

- Après avoir décidé les poids, on peut changer la valeur d'un état...



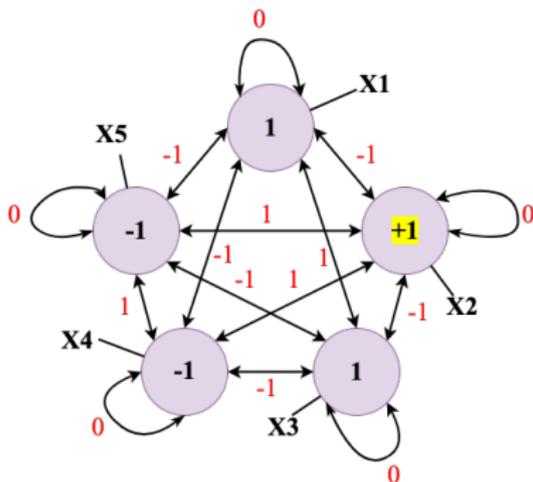
- ..puis, on peut appliquer la règle de tout à l'heure :

- $$x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij}x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$$

- En théorie, le réseau corrigera l'état différent afin de retourner au motif original.

Machine Neuronale MIND 1024

- Après avoir décidé les poids, on peut changer la valeur d'un état...



- ..puis, on peut appliquer la règle de tout à l'heure :

$$x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij}x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$$

- En théorie, le réseau corrigera l'état différent afin de retourner au motif original.

- La machine aura retrouvé le motif « dans sa mémoire ».

1 Introduction

2 Réseau de Hopfield plus en profondeur

En quoi ça consiste ?

Règle de mise à jour

Petit exemple

Règle de Hebb

3 Simulateur

4 Réseaux Neuronaux Aujourd'hui

5 Conclusion

Machine Neuronale MIND 1024

- Règle de Hebb, introduite par *Donald Hebb* en 1949 :

Machine Neuronale MIND 1024

- Règle de Hebb, introduite par *Donald Hebb* en 1949 :
- Soit n la taille de l'ensemble de configurations cibles du réseau $\{x_1, \dots, x_n\}$,

Machine Neuronale MIND 1024

- Règle de Hebb, introduite par *Donald Hebb* en 1949 :
- Soit n la taille de l'ensemble de configurations cibles du réseau $\{x_1, \dots, x_n\}$,
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^{\mu} x_j^{\mu}$, où x_i^{μ} représente bit i de motif μ .

Machine Neuronale MIND 1024

- Règle de Hebb, introduite par *Donald Hebb* en 1949 :
- Soit n la taille de l'ensemble de configurations cibles du réseau $\{x_1, \dots, x_n\}$,
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^{\mu} x_j^{\mu}$, où x_i^{μ} représente bit i de motif μ .
- $w_{ij} = 0$.

Machine Neuronale MIND 1024

- Règle de Hebb, introduite par *Donald Hebb* en 1949 :
- Soit n la taille de l'ensemble de configurations cibles du réseau $\{x_1, \dots, x_n\}$,
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^{\mu} x_j^{\mu}$, où x_i^{μ} représente bit i de motif μ .
- $w_{ij} = 0$.
- Effet de cette règle :

Machine Neuronale MIND 1024

- Règle de Hebb, introduite par *Donald Hebb* en 1949 :
- Soit n la taille de l'ensemble de configurations cibles du réseau $\{x_1, \dots, x_n\}$,
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^\mu x_j^\mu$, où x_i^μ représente bit i de motif μ .
- $w_{ij} = 0$.
- Effet de cette règle :
 - C'est le calcul de l'activation moyenne entre deux neurones voisins.

Machine Neuronale MIND 1024

- Règle de Hebb, introduite par *Donald Hebb* en 1949 :
- Soit n la taille de l'ensemble de configurations cibles du réseau $\{x_1, \dots, x_n\}$,
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^\mu x_j^\mu$, où x_i^μ représente bit i de motif μ .
- $w_{ij} = 0$.
- Effet de cette règle :
 - C'est le calcul de l'activation moyenne entre deux neurones voisins.
 - Si les bits qui correspondent aux neurones i et j sont égaux dans motif μ , alors le produit $x_i^\mu x_j^\mu$ sera positif. Par conséquent, le poids w_{ij} tend à être positif et donc les valeurs de i et j tendent à être égales.

Machine Neuronale MIND 1024

- Règle de Hebb, introduite par *Donald Hebb* en 1949 :
- Soit n la taille de l'ensemble de configurations cibles du réseau $\{x_1, \dots, x_n\}$,
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^{\mu} x_j^{\mu}$, où x_i^{μ} représente bit i de motif μ .
- $w_{ij} = 0$.
- Effet de cette règle :
 - C'est le calcul de l'activation moyenne entre deux neurones voisins.
 - Si les bits qui correspondent aux neurones i et j sont égaux dans motif μ , alors le produit $x_i^{\mu} x_j^{\mu}$ sera positif. Par conséquent, le poids w_{ij} tend à être positif et donc les valeurs de i et j tendent à être égales.
 - Si les bits qui correspondent aux neurones i et j sont différents dans motif μ , c'est l'inverse.

- 1 Introduction
- 2 Réseau de Hopfield plus en profondeur
- 3 Simulateur**
 - Code
 - Code QR
- 4 Réseaux Neuronaux Aujourd'hui
- 5 Conclusion

- 1 Introduction
- 2 Réseau de Hopfield plus en profondeur
- 3 Simulateur**
 - Code
 - Code QR
- 4 Réseaux Neuronaux Aujourd'hui
- 5 Conclusion

Machine Neuronale MIND 1024

- Pour démontrer l'utilité du MIND 1024, il faut un simulateur d'un réseau de Hopfield.

Machine Neuronale MIND 1024

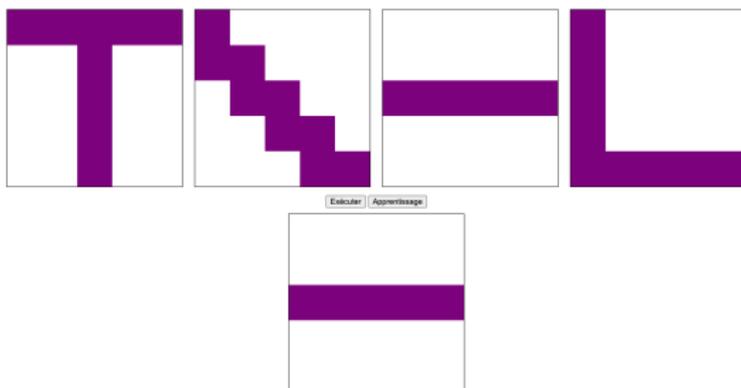
- Pour démontrer l'utilité du MIND 1024, il faut un simulateur d'un réseau de Hopfield.
- Je me suis procuré du code de base sur GitHub. Utilisateur : Alex Nichol (<https://github.com/unixpickle>).

Machine Neuronale MIND 1024

- Pour démontrer l'utilité du MIND 1024, il faut un simulateur d'un réseau de Hopfield.
- Je me suis procuré du code de base sur GitHub. Utilisateur : Alex Nichol (<https://github.com/unixpickle>).
- Il s'agit d'une application qui correspond un dessin donné à un dessin mémorisé.

Machine Neuronale MIND 1024

- Pour démontrer l'utilité du MIND 1024, il faut un simulateur d'un réseau de Hopfield.
- Je me suis procuré du code de base sur GitHub. Utilisateur : Alex Nichol (<https://github.com/unixpickle>).
- Il s'agit d'une application qui correspond un dessin donné à un dessin mémorisé.
- L'application est implémentée avec *Javascript*.



Machine Neuronale MIND 1024

Fonction 'convergeRandomNode' (calcul d'un nouvel état aléatoire) :

```
51 HopfieldNet.prototype._convergeRandomNode = function() {
52   var nodeIndex = Math.floor(Math.random() * this._values.length);
53   var sum = 0;
54   for (var j = 0, len = this._values.length; j < len; ++j) {
55     if (j === nodeIndex) {
56       continue;
57     }
58     var w = this._weights[nodeIndex][j];
59     var v = this._values[j];
60     sum += w * v;
61   }
62   if (sum > ACTIVATION_THRESHOLD) {
63     this._values[nodeIndex] = 1;
64   } else {
65     this._values[nodeIndex] = -1;
66   }
67   };
```

Machine Neuronale MIND 1024

- La fonction au-dessus converge un neurone aléatoire.

Machine Neuronale MIND 1024

- La fonction au-dessus converge un neurone aléatoire.
- Elle correspond à la règle que l'on a vue :

Machine Neuronale MIND 1024

- La fonction au-dessus converge un neurone aléatoire.
- Elle correspond à la règle que l'on a vue :
- $x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij}x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$

Machine Neuronale MIND 1024

- La fonction au-dessus converge un neurone aléatoire.
- Elle correspond à la règle que l'on a vue :
- $$x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij} x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$$
- En fait, il existe 2 types de mise à jour de l'état :

Machine Neuronale MIND 1024

- La fonction au-dessus converge un neurone aléatoire.
- Elle correspond à la règle que l'on a vue :

- $$x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij} x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$$

- En fait, il existe 2 types de mise à jour de l'état :
 - **Asynchrone** : le calcul de nouveaux états se fait *un état à la fois*.

Machine Neuronale MIND 1024

- La fonction au-dessus converge un neurone aléatoire.
- Elle correspond à la règle que l'on a vue :
- $$x_i \leftarrow \begin{cases} +1, & \text{si } \sum_j w_{ij} x_j \geq \theta_i, \\ -1, & \text{sinon} \end{cases}$$
- En fait, il existe 2 types de mise à jour de l'état :
 - **Asynchrone** : le calcul de nouveaux états se fait *un état à la fois*.
 - **Synchrone** : la mise à jour prend en compte tous les états simultanément. Cette méthode nécessite un horloge pour maintenir la synchronisation.

Machine Neuronale MIND 1024

Fonction 'train' (apprentissage) :

```
19 HopfieldNet.prototype.train = function(samples) {
20   this._resetWeights();
21   var nodeCount = this._values.length;
22   for (var i = 0; i < nodeCount; ++i) {
23     for (var j = 0; j < nodeCount; ++j) {
24       for (var k = 0, len = samples.length; k < len; ++k) {
25         if (samples[k][i] === samples[k][j]) {
26           this._weights[i][j] += 1 / len;
27         } else {
28           this._weights[i][j] -= 1 / len;
29         }
30       }
31     }
32   }
33   };
```

Machine Neuronale MIND 1024

- La fonction au-dessus fixe les valeurs des poids des connexions à partir des dessins d'entrée, (les échantillons).

Machine Neuronale MIND 1024

- La fonction au-dessus fixe les valeurs des poids des connexions à partir des dessins d'entrée, (les échantillons).
- Elle correspond à la règle de Hebb :

Machine Neuronale MIND 1024

- La fonction au-dessus fixe les valeurs des poids des connexions à partir des dessins d'entrée, (les échantillons).
- Elle correspond à la règle de Hebb :
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^{\mu} x_j^{\mu}$

Machine Neuronale MIND 1024

- La fonction au-dessus fixe les valeurs des poids des connexions à partir des dessins d'entrée, (les échantillons).
- Elle correspond à la règle de Hebb :
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^{\mu} x_j^{\mu}$
- Ici :

Machine Neuronale MIND 1024

- La fonction au-dessus fixe les valeurs des poids des connexions à partir des dessins d'entrée, (les échantillons).
- Elle correspond à la règle de Hebb :
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^{\mu} x_j^{\mu}$
- Ici :
 - Le variable *len* correspond à la taille de l'ensemble d'échantillons *n*.

Machine Neuronale MIND 1024

- La fonction au-dessus fixe les valeurs des poids des connexions à partir des dessins d'entrée, (les échantillons).
- Elle correspond à la règle de Hebb :
- $w_{ij} = \frac{1}{n} \sum_{\mu=1}^n x_i^{\mu} x_j^{\mu}$
- Ici :
 - Le variable len correspond à la taille de l'ensemble d'échantillons n .
 - Le variable k correspond à un échantillon μ .

- 1 Introduction
- 2 Réseau de Hopfield plus en profondeur
- 3 Simulateur**
 - Code
 - Code QR
- 4 Réseaux Neuronaux Aujourd'hui
- 5 Conclusion

Machine Neuronale MIND 1024

- Une image de taille n^2 produit un réseau de n^2 neurones tous interconnectés.

Machine Neuronale MIND 1024

- Une image de taille n^2 produit un réseau de n^2 neurones tous interconnectés.
- Le MIND 1024 possède 1024 et donc la taille maximale d'une image que l'on lui aurait pu fournir est de $n^2 = 1024$,
 $\therefore n = \sqrt{1024} = 32$.

Machine Neuronale MIND 1024

- Une image de taille n^2 produit un réseau de n^2 neurones tous interconnectés.
- Le MIND 1024 possède 1024 et donc la taille maximale d'une image que l'on lui aurait pu fournir est de $n^2 = 1024$,
 $\therefore n = \sqrt{1024} = 32$.
- Je voudrais essayer des codes QR comme échantillons.

Machine Neuronale MIND 1024

- Une image de taille n^2 produit un réseau de n^2 neurones tous interconnectés.
- Le MIND 1024 possède 1024 et donc la taille maximale d'une image que l'on lui aurait pu fournir est de $n^2 = 1024$,
 $\therefore n = \sqrt{1024} = 32$.
- Je voudrais essayer des codes QR comme échantillons.
- Il suffit donc d'ajuster le variable 'PICTURE_SIZE' :

```
1 (function() {  
2  
3   var PICTURE_SIZE = 29;  
4   var CANVAS_SIZE = 300;  
5  
6   function Canvas() {  
7     this._cells = [];  
8     for (var i = 0; i < PICTURE_SIZE*PICTURE_SIZE; ++i) {  
9       this._cells[i] = false;  
10    }  
11    this._element = document.createElement('canvas');  
12    this._element.width = CANVAS_SIZE;  
13    this._element.height = CANVAS_SIZE;  
14    this._element.className = 'drawing';  
15    this._registerMouseEvents();  
16  }  
}
```

Machine Neuronale MIND 1024

- On peut avoir un code QR de taille $33 * 33$ pixels, mais cela aurait créé un réseau avec trop de neurones.

Machine Neuronale MIND 1024

- On peut avoir un code QR de taille $33 * 33$ pixels, mais cela aurait créé un réseau avec trop de neurones.
- Alors, un code QR de taille $29 * 29$ pixels est aussi possible et donc j'en crée 3.

Machine Neuronale MIND 1024

- Pour le premier, je génère un code QR à partir d'une phrase courte :

Machine Neuronale MIND 1024

- Pour le premier, je génère un code QR à partir d'une phrase courte :
- « Qu'ils mangent de la brioche ! »

Machine Neuronale MIND 1024

- Pour le premier, je génère un code QR à partir d'une phrase courte :
- « Qu'ils mangent de la brioche ! »
- Il sert d'un échantillon.



Figure 11 – Code QR 1 : Échantillon 1

Machine Neuronale MIND 1024

- Pour le second, je génère un code QR à partir d'une phrase courte similaire :

Machine Neuronale MIND 1024

- Pour le second, je génère un code QR à partir d'une phrase courte similaire :
- « Qu'ils mangent du gâteau ! »

Machine Neuronale MIND 1024

- Pour le second, je génère un code QR à partir d'une phrase courte similaire :
- « Qu'ils mangent du gâteau ! »
- Il sert aussi d'un échantillon.



Figure 12 – Code QR 2 : Échantillon 2

Machine Neuronale MIND 1024

- Pour le troisième, je génère un code QR à partir d'une phrase courte similaire à la première phrase :

Machine Neuronale MIND 1024

- Pour le troisième, je génère un code QR à partir d'une phrase courte similaire à la première phrase :
- « Qu'ils mangent de la brique ! »

Machine Neuronale MIND 1024

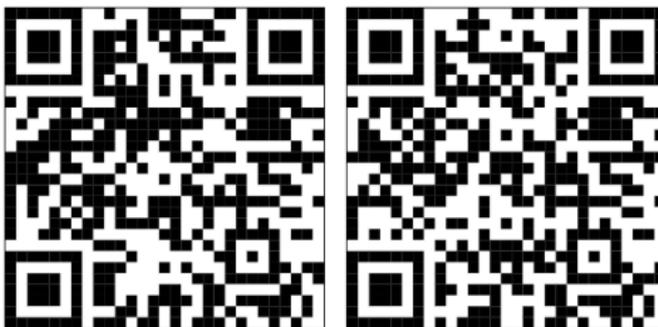
- Pour le troisième, je génère un code QR à partir d'une phrase courte similaire à la première phrase :
- « Qu'ils mangent de la brique ! »
- Il sert de la nouvelle image.



Figure 13 – Code QR 3 : Nouvelle Image

Machine Neuronale MIND 1024

Alors, je teste l'application avec ces trois codes QR...

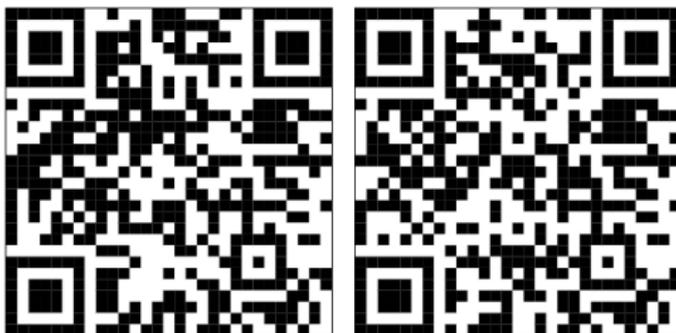


Exécuter Apprentissage



Machine Neuronale MIND 1024

...et le code QR le plus similaire, (le premier), est bien retrouvé !



Exécuter Apprentissage

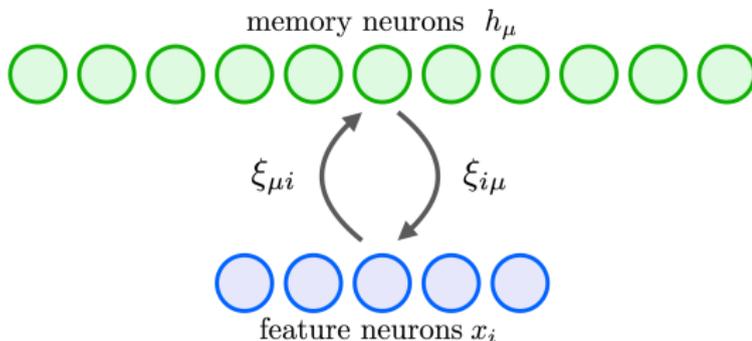


- ① Introduction
- ② Réseau de Hopfield plus en profondeur
- ③ Simulateur
- ④ Réseaux Neuronaux Aujourd'hui**
- ⑤ Conclusion

- À part le réseau de Hopfield classique, il existe aussi un réseau de Hopfield moderne.

- À part le réseau de Hopfield classique, il existe aussi un réseau de Hopfield moderne.
- Ce nouveau système s'appuie sur son prédécesseur.

- À part le réseau de Hopfield classique, il existe aussi un réseau de Hopfield moderne.
- Ce nouveau système s'appuie sur son prédécesseur.
- Il décrit comment l'activité connue de tous les neurones, (dans le passé ou dans le présent), détermine l'activité de chaque neurone à venir.



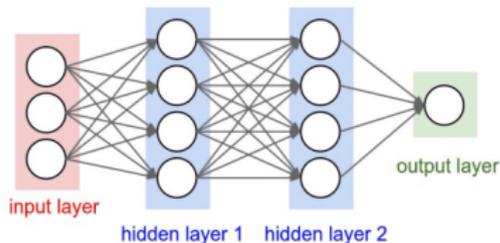
- Dans le domaine de *l'apprentissage profond* :

- Dans le domaine de *l'apprentissage profond* :
- Trois types de réseau neuronal :

- Dans le domaine de *l'apprentissage profond* :
- Trois types de réseau neuronal :
 - Réseau neuronal artificiel

- Dans le domaine de *l'apprentissage profond* :
- Trois types de réseau neuronal :
 - Réseau neuronal artificiel
 - Réseau neuronal de convolution

- Dans le domaine de *l'apprentissage profond* :
- Trois types de réseau neuronal :
 - Réseau neuronal artificiel
 - Réseau neuronal de convolution
 - **Réseau neuronal récurrent**



- ① Introduction
- ② Réseau de Hopfield plus en profondeur
- ③ Simulateur
- ④ Réseaux Neuronaux Aujourd'hui
- ⑤ Conclusion**

- Article d'*Alan Turing* très fondateur en 1950.
- Construction de *Perceptron* par *Frank Rosenblatt* en 1958.
- Réseau inventé par *John Hopfield* en 1982.
- Construction du MIND 1024 en 1985.
- Réseau de Hopfield :
 - Ce en quoi cela consiste.
 - Règle de mise à jour / calcul du nouvel état.
 - Règle de Hebb, introduite par *Donald Hebb* en 1949.
- Réseaux Neuronaux d'aujourd'hui.

Merci!