A curious case of the symmetric binary perceptron model. Algorithms and algorithmic barriers

David Gamarnik (MIT)

DIMACS Workshop on Modeling Randomness in Neural Network Training

June, 2024

Joint work with Eren Kizildag, Will Perkins & Changji Xu

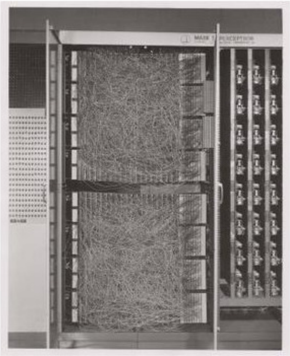
History

History

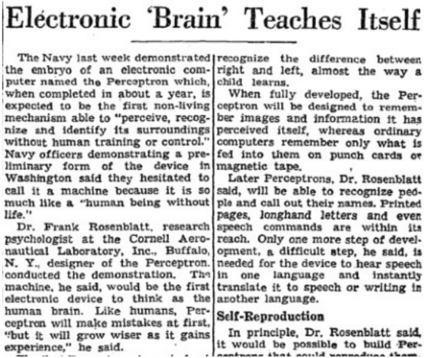
- Warren McCulloch & Walter Pitts [43]

History

- Warren McCulloch & Walter Pitts [43]
- Frank Rosenblatt [58]



Mark I Perceptron



Electronic 'Brain' Teaches Itself

The Navy last week demonstrated the embryo of an electronic computer named the Perceptron which, when completed in about a year, is expected to be the first non-living mechanism able to "perceive, recognize and identify its surroundings without human training or control." Navy officers demonstrating a preliminary form of the device in Washington said they hesitated to call it a machine because it is so much like a "human being without life."

Dr. Frank Rosenblatt, research psychologist at the Cornell Aeronautical Laboratory, Inc., Buffalo, N. Y., designer of the Perceptron, conducted the demonstration. The machine, he said, would be the first electronic device to think as the human brain. Like humans, Perceptron will make mistakes at first, "but it will grow wiser as it gains experience," he said.

... recognize the difference between right and left, almost the way a child learns.

When fully developed, the Perceptron will be designed to remember images and information it has perceived itself, whereas ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons, Dr. Rosenblatt said, will be able to recognize people and call out their names. Printed pages, longhand letters and even speech commands are within its reach. Only one more step of development, a difficult step, he said, is needed for the device to hear speech in one language and instantly translate it to speech or writing in another language.

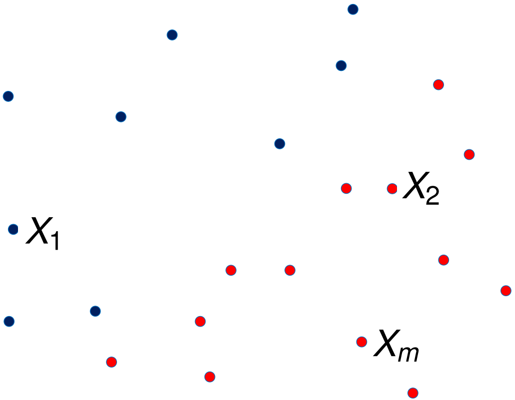
Self-Reproduction

In principle, Dr. Rosenblatt said, it would be possible to build Per-

New York Times, 1958

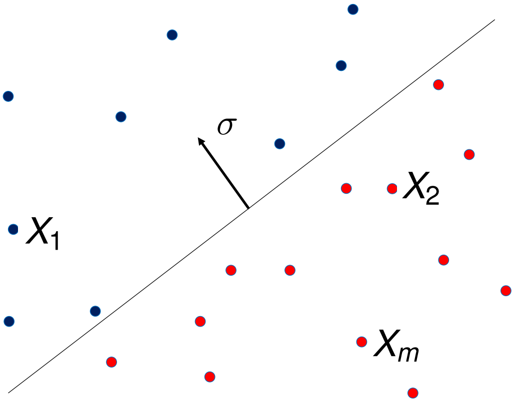
Perceptron

Given labeled data



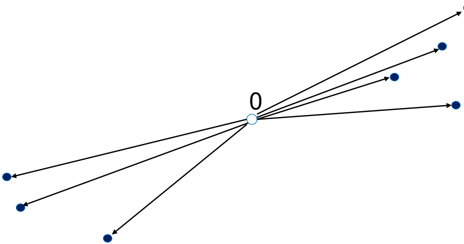
Perceptron

Find a classifier

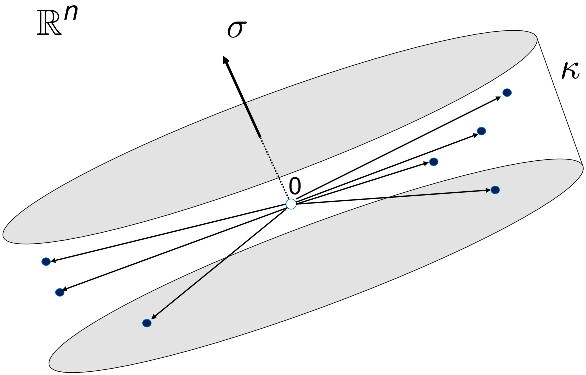


Symmetric perceptron

$\mathbb{R}^n$



Symmetric perceptron



Perceptron

Perceptron

- Fix a subset $K \subset \mathbb{R}$. Given data $X_1, \ldots, X_m \in \mathbb{R}^n$ find a "classifier" σ from the set of classifiers Σ such that $\langle X_i, \sigma \rangle \in K$ for all i.

Perceptron

- Fix a subset $K \subset \mathbb{R}$. Given data $X_1, \ldots, X_m \in \mathbb{R}^n$ find a "classifier" σ from the set of classifiers Σ such that $\langle X_i, \sigma \rangle \in K$ for all i.
- For example, $\Sigma = \{\pm 1\}^n$, $\langle X_i, \sigma \rangle \in (-\kappa, \kappa) = K$

Perceptron

- Fix a subset $K \subset \mathbb{R}$. Given data $X_1, \ldots, X_m \in \mathbb{R}^n$ find a "classifier" σ from the set of classifiers Σ such that $\langle X_i, \sigma \rangle \in K$ for all i.
- For example, $\Sigma = \{\pm 1\}^n$, $\langle X_i, \sigma \rangle \in (-\kappa, \kappa) = K$ – Version we consider

Random symmetric binary perceptron model

Random symmetric binary perceptron model

- $A \in \mathbb{R}^{m \times n}$, $A_{ij} \stackrel{d}{=} N(0, 1/n)$. $\kappa > 0$.

Random symmetric binary perceptron model

- $A \in \mathbb{R}^{m \times n}$, $A_{ij} \overset{d}{=} N(0, 1/n)$. $\kappa > 0$.
- Find $\sigma \in \{\pm 1\}^n$: $A\sigma \in (-\kappa, \kappa)^n$.

$$\begin{bmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \dots & A_{mn} \end{bmatrix} \times \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{bmatrix} \in (-\kappa, \kappa)^n.$$

Random symmetric binary perceptron model

- $A \in \mathbb{R}^{m \times n}$, $A_{ij} \stackrel{d}{=} N(0, 1/n)$. $\kappa > 0$.
- Find $\sigma \in \{\pm 1\}^n$: $A\sigma \in (-\kappa, \kappa)^n$.

$$\begin{bmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \dots & A_{mn} \end{bmatrix} \times \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{bmatrix} \in (-\kappa, \kappa)^n.$$

- $m = \alpha n$ (proportional regime)

Random symmetric binary perceptron model

- $A \in \mathbb{R}^{m \times n}$, $A_{ij} \stackrel{d}{=} N(0, 1/n)$. $\kappa > 0$.
- Find $\sigma \in \{\pm 1\}^n$: $A\sigma \in (-\kappa, \kappa)^n$.

$$\begin{bmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \dots & A_{mn} \end{bmatrix} \times \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{bmatrix} \in (-\kappa, \kappa)^n.$$

- $m = \alpha n$ (proportional regime)
- Set of solutions $\Sigma_{\mathrm{SAT}} = \{\sigma : A\sigma \in (-\kappa, \kappa)^n\}$

Random symmetric binary perceptron model

- $A \in \mathbb{R}^{m \times n}$, $A_{ij} \overset{d}{=} N(0, 1/n)$. $\kappa > 0$.
- Find $\sigma \in \{\pm 1\}^n$: $A\sigma \in (-\kappa, \kappa)^n$.

$$\begin{bmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \dots & A_{mn} \end{bmatrix} \times \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{bmatrix} \in (-\kappa, \kappa)^n.$$

- $m = \alpha n$ (proportional regime)
- Set of solutions $\Sigma_{\mathrm{SAT}} = \{\sigma : A\sigma \in (-\kappa, \kappa)^n\}$
- **Questions:**
  (a) When do solutions exist $\Sigma_{\mathrm{SAT}} \neq \emptyset$?
  (b) Can we find them algorithmically given A, κ?

Existence

Existence

Theorem (Abbe, Li & Sly [20], Xu & Perkins [20])

$Z \overset{d}{=} N(0,1)$. *W.h.p. solutions exist iff*

$$\alpha < \alpha_{\mathrm{SAT}}(\kappa) \triangleq \frac{\log 2}{-\log \mathbb{P}(|Z| \le \kappa)}.$$

Existence

Theorem (Abbe, Li & Sly [20], Xu & Perkins [20])

$Z \overset{d}{=} N(0,1)$. W.h.p. solutions exist iff

$$\alpha < \alpha_{\mathrm{SAT}}(\kappa) \triangleq \frac{\log 2}{-\log \mathbb{P}(|Z| \le \kappa)}.$$

- Intuition: for a fixed σ,

$$\mathbb{P}(A\sigma \in (-\kappa,\kappa)^n) = (\mathbb{P}(-\kappa < Z < \kappa))^{\alpha n}.$$

Existence

Theorem (Abbe, Li & Sly [20], Xu & Perkins [20])

$Z \overset{d}{=} N(0,1)$. W.h.p. solutions exist iff

$$\alpha < \alpha_{\mathrm{SAT}}(\kappa) \triangleq \frac{\log 2}{-\log \mathbb{P}(|Z| \le \kappa)}.$$

- Intuition: for a fixed σ,

$$\mathbb{P}(A\sigma \in (-\kappa,\kappa)^n) = (\mathbb{P}(-\kappa < Z < \kappa))^{\alpha n}.$$

- Expected number of solutions: $2^n\,(\mathbb{P}(-\kappa < Z < \kappa))^{\alpha n}$. Goes to 0 as $n \to \infty$ if $\alpha > \alpha_{\mathrm{SAT}}(\kappa)$.

Algorithms

Algorithms

- For any $\alpha < \alpha_{\mathrm{SAT}}(\kappa)$ the solution space is mostly clustered.

Algorithms

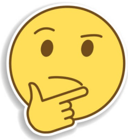
- For any $\alpha < \alpha_{\mathrm{SAT}}(\kappa)$ the solution space is mostly clustered.
- Based on other similar problems this suggests fast (poly-time) algorithms should not exist for any $\alpha > 0$.

Algorithms

- For any $\alpha < \alpha_{\mathrm{SAT}}(\kappa)$ the solution space is mostly clustered.
- Based on other similar problems this suggests fast (poly-time) algorithms should not exist for any $\alpha > 0$.
- Case in point: random K-SAT. Solutions are not clustered in $(0, \alpha_{\mathrm{ALG}})$ and algorithms exist, and are clustered in $(\alpha_{\mathrm{ALG}}, \alpha_{\mathrm{SAT}})$ and algorithms are not known.

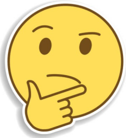
Algorithms

- For any $\alpha < \alpha_{\text{SAT}}(\kappa)$ the solution space is mostly clustered.
- Based on other similar problems this suggests fast (poly-time) algorithms should not exist for any $\alpha > 0$.
- Case in point: random K-SAT. Solutions are not clustered in $(0, \alpha_{\text{ALG}})$ and algorithms exist, and are clustered in $(\alpha_{\text{ALG}}, \alpha_{\text{SAT}})$ and algorithms are not known.
- But solutions **do** exist for small enough α for perceptron! Kim & Roche [98], Bansal & Spencer [20]



Algorithms

- For any $\alpha < \alpha_{\mathrm{SAT}}(\kappa)$ the solution space is mostly clustered.
- Based on other similar problems this suggests fast (poly-time) algorithms should not exist for any $\alpha > 0$.
- Case in point: random K-SAT. Solutions are not clustered in $(0, \alpha_{\mathrm{ALG}})$ and algorithms exist, and are clustered in $(\alpha_{\mathrm{ALG}}, \alpha_{\mathrm{SAT}})$ and algorithms are not known.
- But solutions **do** exist for small enough α for perceptron! Kim & Roche [98], Bansal & Spencer [20]



- Clustering can't be the right answer.

Overlap gap property (OGP)

Overlap gap property (OGP)

- $\kappa = 1$. $\alpha_{\mathrm{SAT}}(1) = 1.82...$

Overlap gap property (OGP)

- $\kappa = 1$. $\alpha_{\rm SAT}(1) = 1.82...$

Theorem (G, Kizildag, Perkins & Xu [22])

For $\alpha \in (1.71.., 1.82..)$ the following holds. There exists $0 < \nu_1 < \nu_2 < 1$ such that for every two $\sigma, \tau \in \Sigma_{\rm SAT}$,

$$n^{-1} d(\sigma, \tau) \in [0, \nu_1] \cup [\nu_2, 1].$$

Namely it exhibits gap in the overlaps.

Overlap gap property (OGP)

- $\kappa = 1$. $\alpha_{\mathrm{SAT}}(1) = 1.82...$

Theorem (G, Kizildag, Perkins & Xu [22])

For $\alpha \in (1.71.., 1.82..)$ the following holds. There exists $0 < \nu_1 < \nu_2 < 1$ such that for every two $\sigma, \tau \in \Sigma_{\mathrm{SAT}}$,

$$n^{-1} d(\sigma, \tau) \in [0, \nu_1] \cup [\nu_2, 1].$$

Namely it exhibits gap in the overlaps.

- OGP is a barrier to classes of algorithms. Surveys G [21], G, Moore & Zdeborova [22], Marino [23]

Overlap gap property (OGP)

- $\kappa = 1$. $\alpha_{\text{SAT}}(1) = 1.82...$

Theorem (G, Kizildag, Perkins & Xu [22])

For $\alpha \in (1.71.., 1.82..)$ the following holds. There exists $0 < \nu_1 < \nu_2 < 1$ such that for every two $\sigma, \tau \in \Sigma_{\text{SAT}}$,

$$n^{-1} d(\sigma, \tau) \in [0, \nu_1] \cup [\nu_2, 1].$$

Namely it exhibits gap in the overlaps.

- OGP is a barrier to classes of algorithms. Surveys G [21], G, Moore & Zdeborova [22], Marino [23]
- But... known algorithms work for α much smaller than 1.71...

Overlap gap property (OGP)

- $\kappa = 1$. $\alpha_{\mathrm{SAT}}(1) = 1.82...$

Theorem (G, Kizildag, Perkins & Xu [22])

For $\alpha \in (1.71.., 1.82..)$ *the following holds. There exists* $0 < \nu_1 < \nu_2 < 1$ *such that for every two* $\sigma, \tau \in \Sigma_{\mathrm{SAT}}$,

$$n^{-1} d(\sigma, \tau) \in [0, \nu_1] \cup [\nu_2, 1].$$

Namely it exhibits gap in the overlaps.

- OGP is a barrier to classes of algorithms. Surveys G [21], G, Moore & Zdeborova [22], Marino [23]
- But... known algorithms work for α much smaller than 1.71...
- Can we get matching bounds?

ensemble-OGP

ensemble-OGP

- Create an independent copy $\tilde{A}$ of A.
  $A(t) = \sqrt{1-t}A + \sqrt{t}\tilde{A}$. Still i.i.d. $N(0, 1/n)$ entries.

ensemble-OGP

- Create an independent copy $\tilde{A}$ of A.
  $A(t) = \sqrt{1-t}A + \sqrt{t}\tilde{A}$. Still i.i.d. $N(0, 1/n)$ entries.

Theorem (G, Kizildag, Perkins & Xu [22])

For $\alpha \in (1.71.., 1.82..)$ and the same $0 < \nu_1 < \nu_2 < 1$, the following holds: for every $0 < s < t < 1$ and every $\sigma \in \Sigma_{\mathrm{SAT}}(s), \tau \in \Sigma_{\mathrm{SAT}}(t)$,

$$n^{-1}d(\sigma, \tau) \in [0, \nu_1] \cup [\nu_2, 1].$$

Namely the gap holds across all instances $A(t)$.

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

- $\kappa \to 0$. m independent copies $A_1, \ldots, A_m$. m interpolation paths $\sqrt{1-t}A + \sqrt{t}\tilde{A}_k, k = 1, \ldots, m$.

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

- $\kappa \to 0$. m independent copies $A_1, \ldots, A_m$. m interpolation paths $\sqrt{1-t}A + \sqrt{t}\tilde{A}_k, k = 1, \ldots, m$.

Theorem (G, Kizildag, Perkins & Xu [22])

For every sufficiently small κ and every

$$10\kappa^2 \log\left(\frac{1}{\kappa}\right) < \alpha < \frac{\log 2}{\log\left(\frac{1}{\kappa}\right)}$$

there exist $0 < \nu_1 < \nu_2 < 1$ and m such that the following holds: for every $0 < s_1 < s_2 < \cdots < s_m < 1$ and every $\sigma_1 \in \Sigma_{\text{SAT}}(A(s_1)), \ldots, \sigma_m \in \Sigma_{\text{SAT}}(A(s_m))$,

$$n^{-1} d(\sigma_k, \sigma_\ell) \in [0, \nu_1] \cup [\nu_2, 1],$$

for at least one pair $1 \le k, l \le m$.

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

Theorem (GKPX)

Informally, for small κ, and

$$\alpha_{\mathrm{OGP}} = 10\kappa^2 \log\left(\frac{1}{\kappa}\right) < \alpha < \alpha_{\mathrm{SAT}}(\kappa) = \frac{\log 2}{\log\left(\frac{1}{\kappa}\right)}$$

there are no tuples of solutions with intermediate pair-wise distances, across all m interpolation paths.

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

- The best algorithm Bansal & Spencer [20] works when $\alpha = O(\kappa^2) \approx O(\kappa^2 \log(1/\kappa)) = \alpha_{\text{OGP}}$.

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

- The best algorithm Bansal & Spencer [20] works when $\alpha = O(\kappa^2) \approx O(\kappa^2 \log(1/\kappa)) = \alpha_{\mathrm{OGP}}$.
- A suspect is found – α_{OGP} !

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

- The best algorithm Bansal & Spencer [20] works when $\alpha = O(\kappa^2) \approx O(\kappa^2 \log(1/\kappa)) = \alpha_{\text{OGP}}$.
- A suspect is found – α_{OGP} !

ensemble-multi-OGP. Asymptotics $\kappa \to 0$

- The best algorithm Bansal & Spencer [20] works when $\alpha = O(\kappa^2) \approx O(\kappa^2 \log(1/\kappa)) = \alpha_{\text{OGP}}$.
- A suspect is found – α_{OGP} !

- But can we prove guilt beyond reasonable doubt?

What can be ruled out by OGP?

What can be ruled out by OGP?

- What classes of algorithms can be ruled out with OGP?

What can be ruled out by OGP?

- What classes of algorithms can be ruled out with OGP?
- Stable algorithms: small change in A results in a small change in $\mathcal{ALG}(A)$.

What can be ruled out by OGP?

- What classes of algorithms can be ruled out with OGP?
- Stable algorithms: small change in A results in a small change in $\mathcal{ALG}(A)$.

Theorem (GKPX)

Kim & Roche [93] algorithm is stable: if $t = \frac{1}{n^{0.02}}$ then whp

$$d\left(ALG(A(t))\mathcal{ALG}(A(0))\right) = o(n)$$

I.e. changing input by $1/n^{0.02}$ changes only sublinearly many entries in the solution.

OGP is a barrier to stable algorithms

OGP is a barrier to stable algorithms

But what about Banal & Spencer algorithm? Is it stable?

OGP is a barrier to stable algorithms

But what about Banal & Spencer algorithm? Is it stable? Most likely not...

OGP is a barrier to *online* algorithms

OGP is a barrier to *online* algorithms

Bansal & Spencer algorithm is online: σ_k is set to ± 1 depending *only* on columns $1, \ldots, k$ of the matrix A.

OGP is a barrier to *online* algorithms

Bansal & Spencer algorithm is online: σ_k is set to ± 1 depending *only* on columns $1, \ldots, k$ of the matrix A.

Theorem (GKPX)

Every ***online*** *algorithm can be implemented on interpolated instances* $A(t), t \in [0, 1]$ *to create m-tuples ruled out by e-m-OGP. That is OGP is a barrier to online algorithms.*

OGP is a barrier to *online* algorithms

Bansal & Spencer algorithm is online: σ_k is set to ± 1 depending *only* on columns $1, \ldots, k$ of the matrix A.

Theorem (GKPX)

*Every **online** algorithm can be implemented on interpolated instances $A(t), t \in [0, 1]$ to create m-tuples ruled out by e-m-OGP. That is OGP is a barrier to online algorithms.*

Thank you.