Ethik in der KI

KI soll **verantwortlich** sein - idealerweise sollten wir wissen, wer oder was von einem KI-Modell getroffene Entscheidung verursacht hat.

KI soll **Fair** sein - Frei von Verurteilen.

KI soll **Mensch-zentriert** sein - sie sollte menschliche Bedürfnisse in sozialen, politischen und emotionalen Kontexten beachten und respektieren.

→ Wir können diese Anforderungen zu erfüllen, indem wir KI erklärbar machen.

→ Ein möglicher Ansatz ist die Verwendung von symbolischer KI.

- Ein besonderer Ansatz ist die Verwendung von symbolischen Vektorarchitekturen, auch bekannt als 'hyperdimensional computing' (HDC)

HDC

Binding:

$$Wein = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \quad Rot = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \quad Weiß = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

← Element-wise multiplikation

$$Rotwein = Wein \otimes Rot$$

$$= \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

$$Weißwein = Wein \otimes weiß$$

$$= \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

Bundling:

Gegeben - ein Menge von HDVs $\{\vec{v}_1, \vec{v}_2, \dots \vec{v}_n\}$

use majority vote sum to bundle them:

$$s = \sum_{k=1}^{n} v_n$$

Then take the sign of s:

$$\text{sign}(s) = \begin{cases} +1 & s \geq 0 \\ -1 & s < 0 \end{cases}$$

Weinkeller = Rotwein $\oplus$ Weißwein

$$\Rightarrow \quad s = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \\ -2 \end{bmatrix}$$

$$\text{Weinkeller} = \text{sign}(s) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

Unbinding

Problem: Wir haben ein unbekanntes Getränk mit einer rosa Farbe - eine Mischung aus Rot und Weiß.

Lösung: Wir können den gebündelten Vektor Weinkeller abfragen ob es Wein ist.

$$\text{Rosa} = \text{Rot} \oplus \text{Weiß}$$

$$= \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \oplus \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

Unbinding:

$$\text{Rosa} \otimes \text{Weinkeller} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

Ergebniss $\neq$ Wein

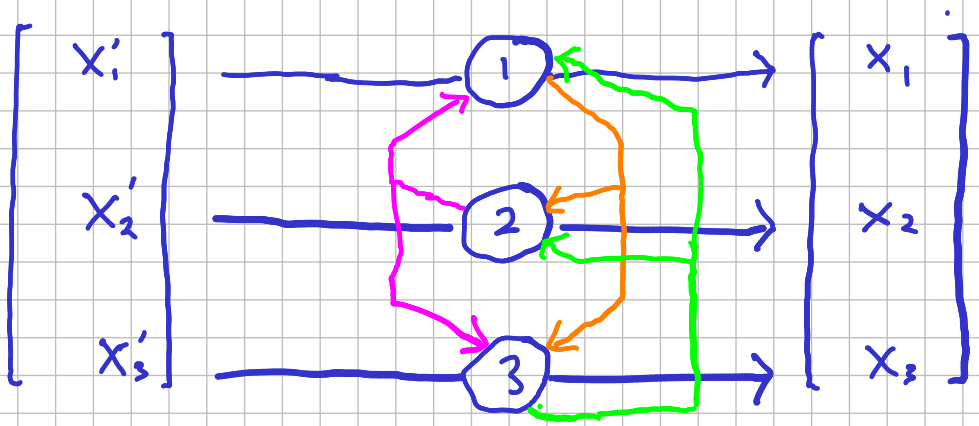
Ergebniss ist eine verrauschte Version von Wein

Frage: Wie können wir die ursprüngliche Repräsentation von Wein wiederherstellen?

Hopfield Network (HF)

Ein Hopfield-Netz ist ein vollständig verbundenes neuronales Netz, das einen verrauschten oder unvollständigen Vektor aufnimmt und die am besten passende gelernte Repräsentation ausgibt.

$\vec{x}'$ $\Rightarrow$ verrauschten $\vec{x}$



- No 'self-loops'
- Gewichte $W \in \mathbb{R}^{N \times N}$ $\Rightarrow$ hier 3×3

$$W = \begin{bmatrix} 0 & w_{12} & w_{13} \\ w_{21} & 0 & w_{23} \\ w_{31} & w_{32} & 0 \end{bmatrix}$$

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} x_i^{(\mu)} x_j^{(\mu)}$$

Beispiel: Wir können die Muster von Wein, Rot und Weiß lernen.

$P = 3$

$N = 3$

$$p_1 = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \qquad p_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \qquad p_3 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

Wein Rot Weiß

$$w_{12} = \frac{1}{3}[(-1)(-1) + (1)(1) + (-1)(1)] = \frac{1}{3}(1) = \frac{1}{3}$$

$$w_{13} = \frac{1}{3}[(-1)(1) + (1)(-1) + (-1)(-1)] = \frac{1}{3}(-1) = -\frac{1}{3}$$

$$w_{21} = w_{12} = \frac{1}{3}$$

$$w_{23} = \frac{1}{3}[(-1)(1) + (1)(-1) + (1)(-1)] = \frac{1}{3}(-3) = -1$$

$$w_{31} = w_{13} = -\frac{1}{3}$$

$$w_{32} = w_{23} = -1$$

$$W = \begin{bmatrix} 0 & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & 0 & -1 \\ -\frac{1}{3} & -1 & 0 \end{bmatrix}$$

Frage: Wie können wir die ursprüngliche Repräsentation von Wein wiederherstellen?

$\vec{x}$ ausgabe für $\vec{x}'$ eingabe

$$x_i = \text{sign}\left(\sum_j w_{ij} x'_j\right)$$

$$\text{Wein} + \text{Noise (Rauschen)} = \vec{x}' = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$$x_1 = \text{sign}\left[w_{12} x'_2 + w_{13} x'_3\right] = \text{sign}\left[\tfrac{1}{3}(-1) + (-\tfrac{1}{3})(1)\right]$$
$$= \text{sign}(-\tfrac{2}{3}) = \boxed{-1}$$

$$x_2 = \text{sign}\left[w_{21} x'_1 + w_{23} x'_3\right] = \text{sign}\left[\tfrac{1}{3}(1) + (-1)(1)\right]$$
$$= \text{sign}(-\tfrac{2}{3}) = \boxed{-1}$$

$$x_3 = \text{sign}\left[w_{31} x'_1 + w_{32} x'_2\right] = \text{sign}\left[(-\tfrac{1}{3})(1) + (-1)(-1)\right]$$
$$= \text{sign}(\tfrac{2}{3}) = \boxed{1}$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} = w_{ein} \quad \checkmark$$