Deep Learning

Deeper, Better, _____ , Stronger than Machine Learning

Valentin Barriere

Universidad de Chile – DCC

CC6204, Primavera 2025

Cross-entropy Loss

Softmax

Outline : Softmax

Outline : Recordatorios

Recordatorios de geometría: hiperplano I

Definición

En un espacio de dimensión d, como $\mathbb{R}^d$ un hiperplano $\mathcal{H}$ es el conjunto de puntos $\mathbf{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$ que satisfacen una ecuación del tipo:

$$\mathcal{H} : w_1 x_1 + w_2 x_2 + ... + w_d x_d + w_0 = 0, \quad \text{donde} \quad \forall i \quad w_i \in \mathbb{R}$$

Un hiperplano divide el espacio en 2 partes (es un espacio de dimensión $d-1$). En $\mathbb{R}^2$ (plano) es de dimensión 1 (es ??), en $\mathbb{R}^3$ es de dimensión 2 (es ??)...

El signo de $w_1 x_1 + w_2 x_2 + ... + w_d x_d + w_0$ define de qué lado del espacio separado por $\mathcal{H}$ se encuentra $\mathbf{X}$: si > 0 entonces $\mathbf{X}$ está de un lado del hiperplano, sino está del otro lado.

Recordatorios de geometría: hiperplano I

Definición

En un espacio de dimensión d, como $\mathbb{R}^d$ un hiperplano $\mathcal{H}$ es el conjunto de puntos $\mathbf{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$ que satisfacen una ecuación del tipo:

$$\mathcal{H} : w_1 x_1 + w_2 x_2 + ... + w_d x_d + w_0 = 0, \quad \text{donde} \quad \forall i \quad w_i \in \mathbb{R}$$

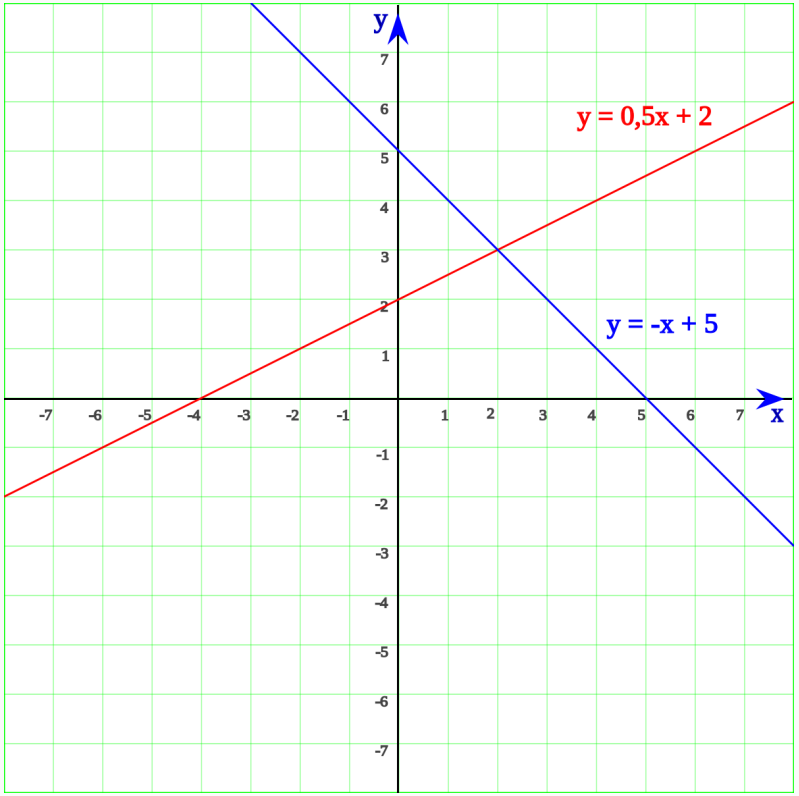
Un hiperplano divide el espacio en 2 partes (es un espacio de dimensión $d-1$). En $\mathbb{R}^2$ (plano) es de dimensión 1 (es una recta), en $\mathbb{R}^3$ es de dimensión 2 (es un plano)...

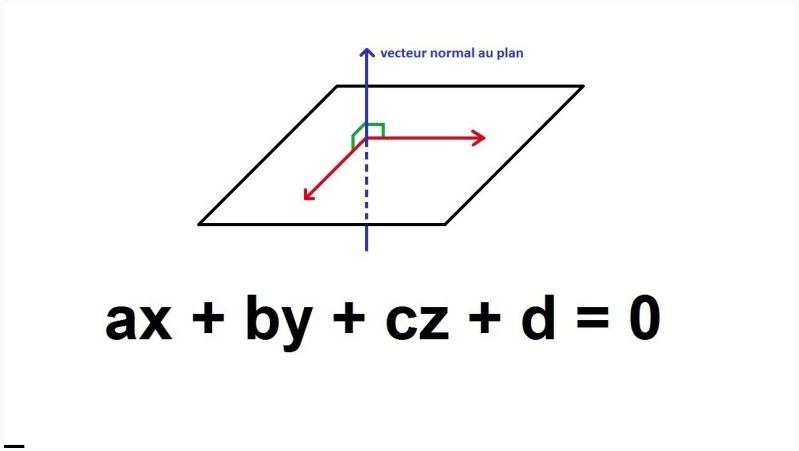
El signo de $w_1 x_1 + w_2 x_2 + ... + w_d x_d + w_0$ define de qué lado del espacio separado por $\mathcal{H}$ se encuentra $\mathbf{X}$: si > 0 entonces $\mathbf{X}$ está de un lado del hiperplano, sino está del otro lado.

Recordatorios de geometría: hiperplano II

Un hiperplano en 2D (= ???) definido por un vector normal $\vec{n} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ y un sesgo w_0.

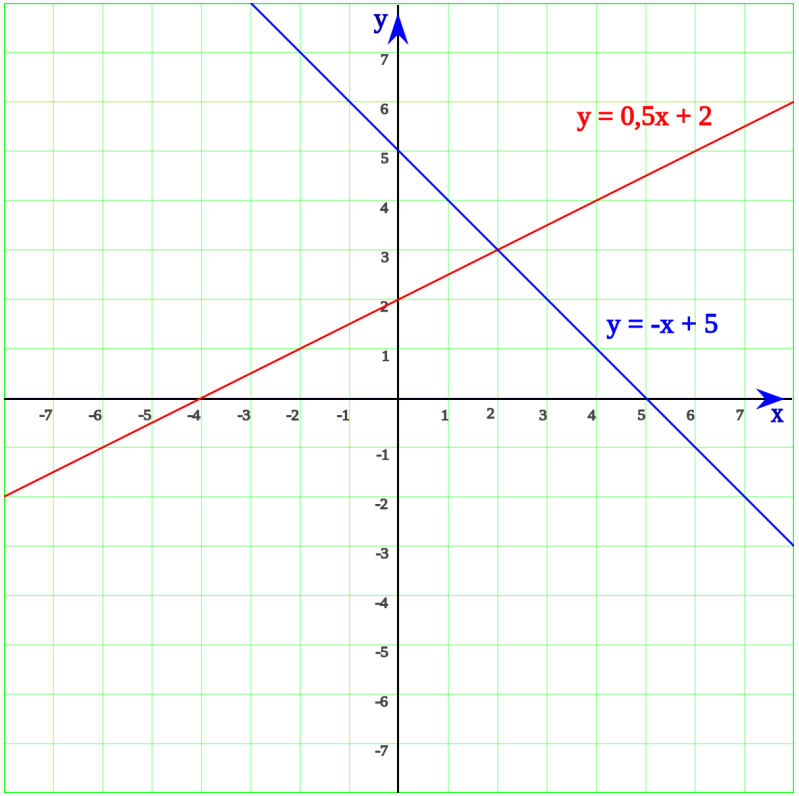
Un hiperplano en 3D (= ???) definido por un vector normal $\vec{n} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$ y un sesgo d



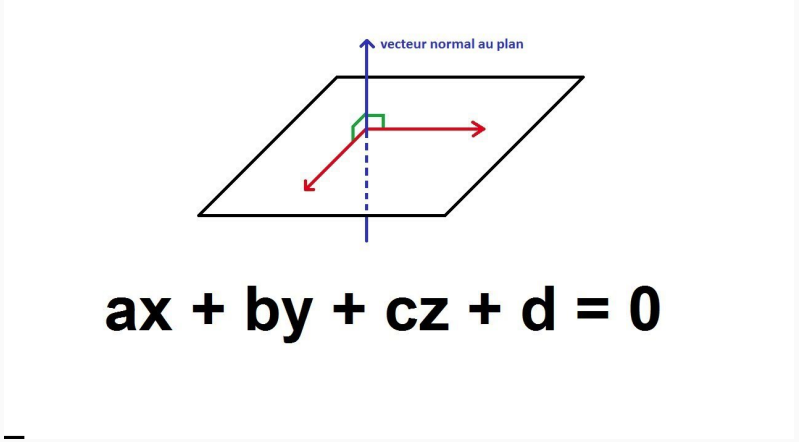


Recordatorios de geometría: hiperplano II

Un hiperplano en 2D (= Recta) definido por un vector normal $\vec{n} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ y un sesgo w_0.



Un hiperplano en 3D (= Plano) definido por un vector normal $\vec{n} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$ y un sesgo d

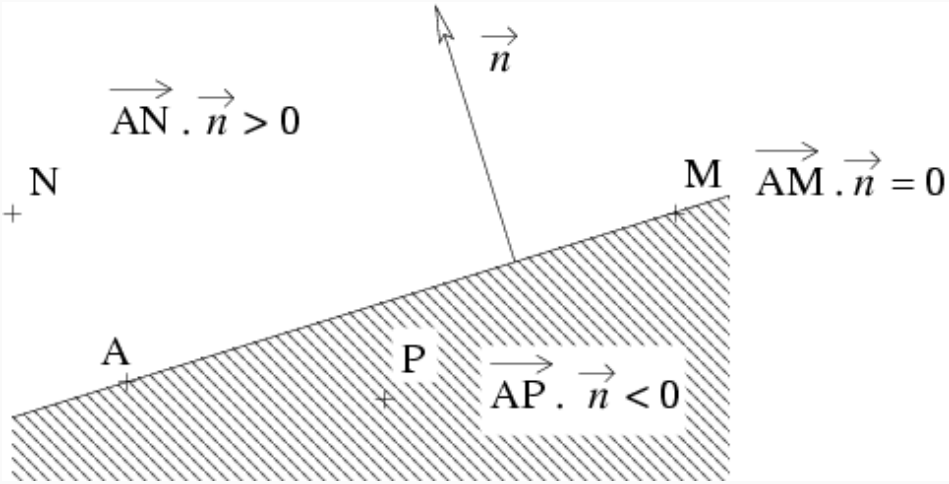


Recordatorios de geometría: hiperplano III

Ecuación

Para cualquier punto $\mathbf{X} \in \mathbb{R}^d$, el signo de $w_1 x_1 + ... w_d x_d$ delimita de qué lado del espacio separado por el hiperplano se encuentra el punto $\mathbf{X}$

Tenemos lo que está de un lado del plano y lo que está del otro lado del plano según el valor del producto escalar con la normal $\vec{n}$



Recordatorios de geometría: hiperplano IV

Con $\mathbf{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \\ 1 \end{pmatrix}$ y $\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_d \\ \theta_0 \end{pmatrix}$ entonces se puede reducir la ecuación anterior a:

$$\mathcal{H} : w_1 x_1 + w_2 x_2 + ... + w_d x_d + w_0 = w_0 + \sum_{i=1}^{d} w_i x_i = \langle \theta | \mathbf{X} \rangle = 0$$

Donde $\langle \theta | \mathbf{X} \rangle$ es el producto escalar entre θ y $\mathbf{X}$: una operación lineal.

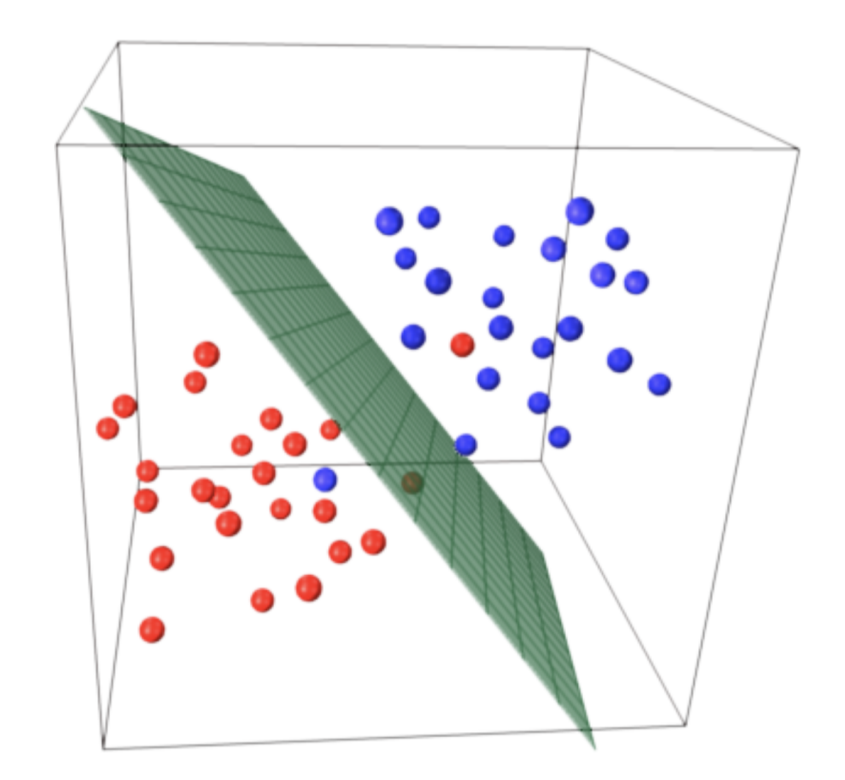
Aumento del espacio para linealidad

Al aumentar el tamaño del espacio, se puede representar un hiperplano afín mediante la ecuación de un hiperplano lineal.

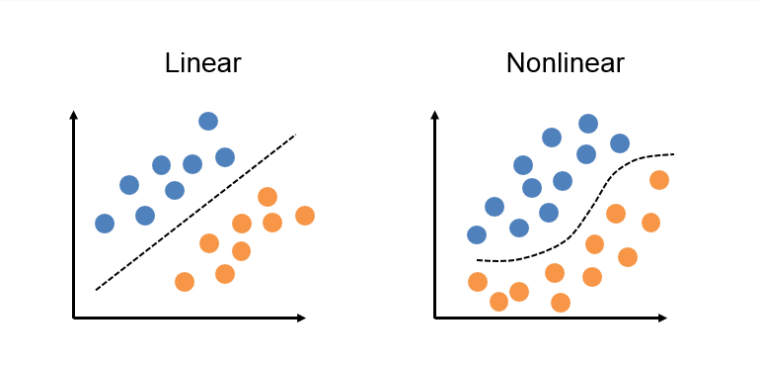
Clasificador lineal: resumen

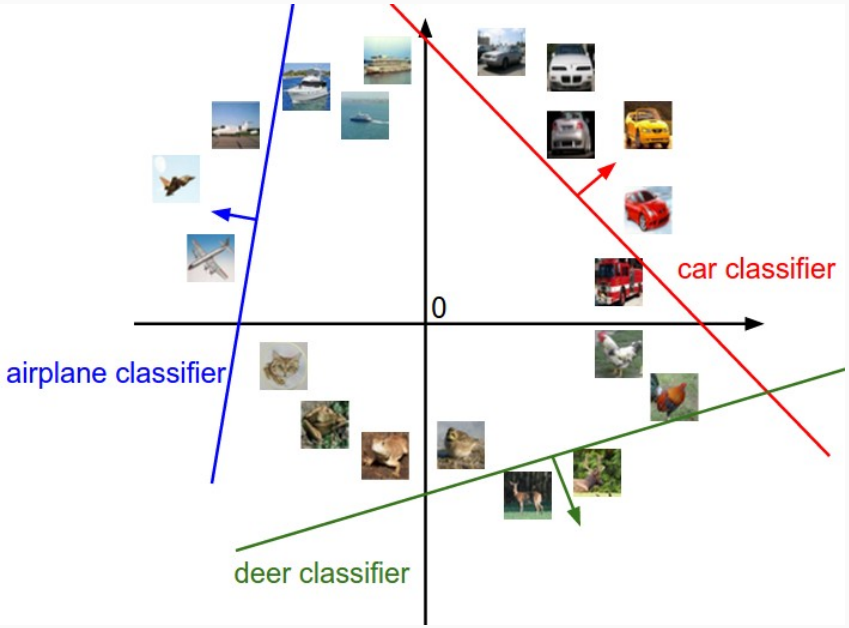
Resumen

- $\mathbf{X} \in \mathbb{R}^d$ es el vector de descriptores
- La ecuación $\mathbf{W}^T\mathbf{X} + b = 0$ define un hiperplano en $\mathbb{R}^d$
- $f_{\mathbf{W},b}(\mathbf{X}) = signo(\mathbf{W}^T\mathbf{X} + b)$ da la clase de $\mathbf{X}$



Clasificador lineal

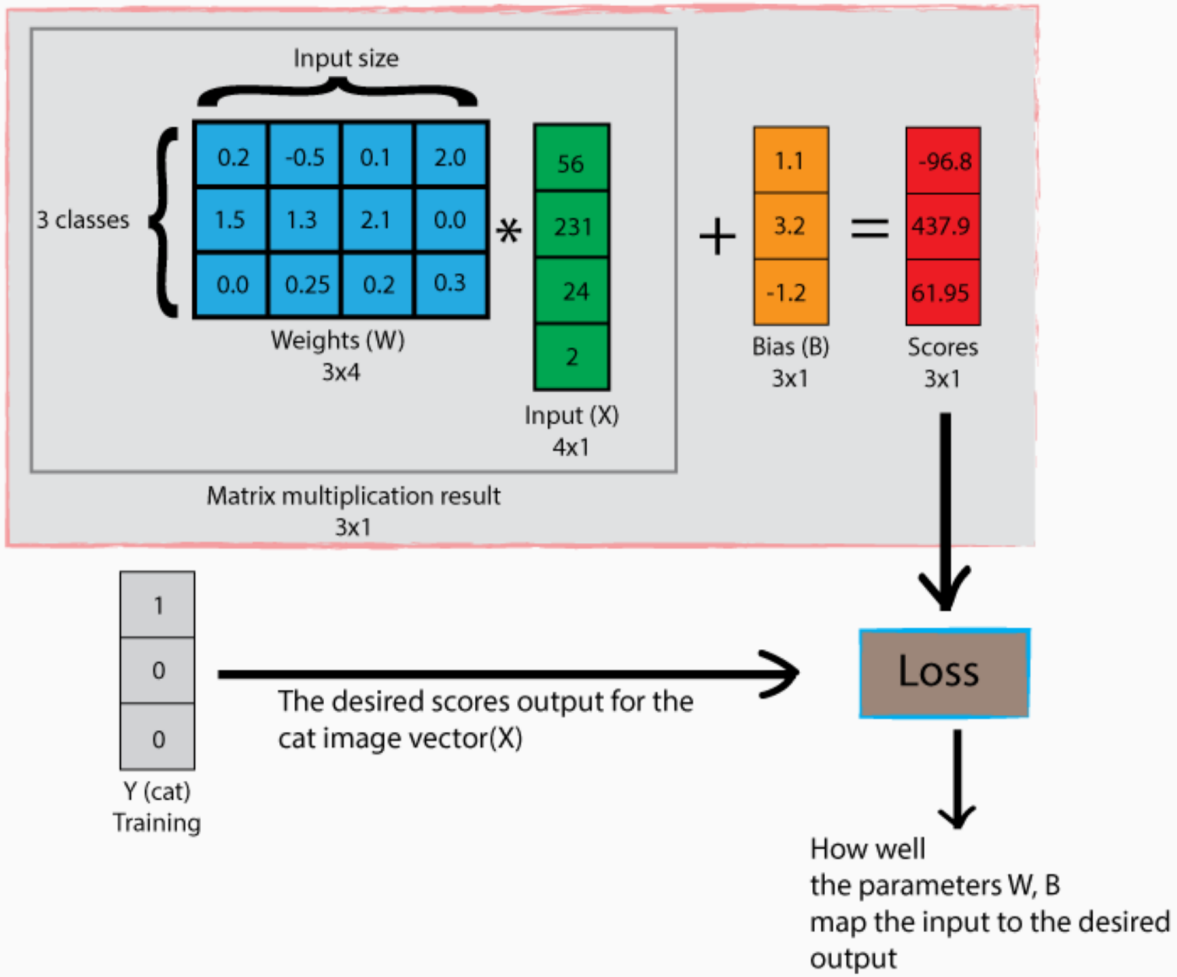




Linealidad

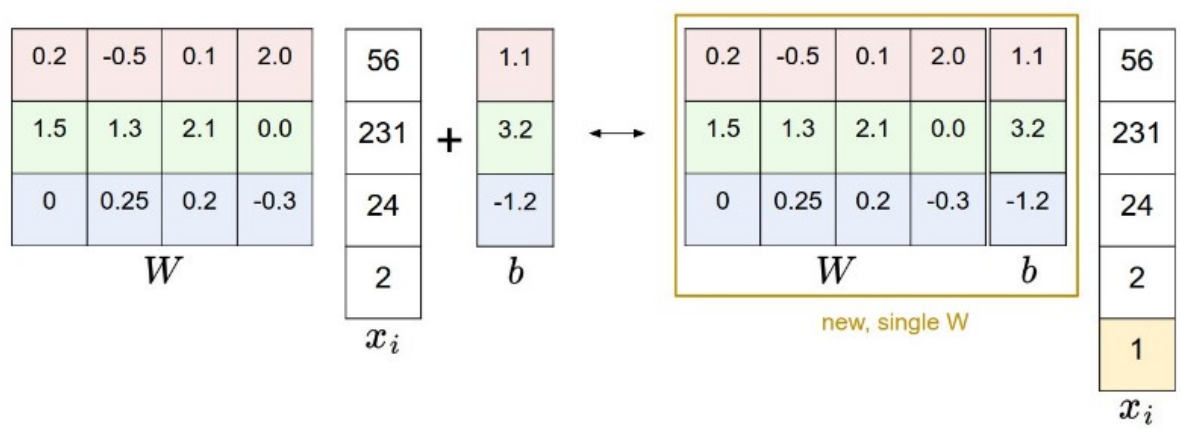
- Separa el espacio con un hiperplano
- Clasificador de la forma $f_{\mathbf{W},b}(\mathbf{X}) = signo(\mathbf{W}^T\mathbf{X} + b)$
- Dos ejemplos anteriores: binario (*2 clases*) y multinomial (*n clases*)
- Operación matricial lineal

Clasificador lineal multiclasses : operación matricial



Ya no tenemos un simple vector de parámetros $\mathbf{W} = (w_1 \cdots w_D)$ como antes, sino una matriz $\mathbf{W} = (w_{cd})_{c=1..C,d=1..D}$

Clasificador lineal : Integración del sesgo



Linealidad

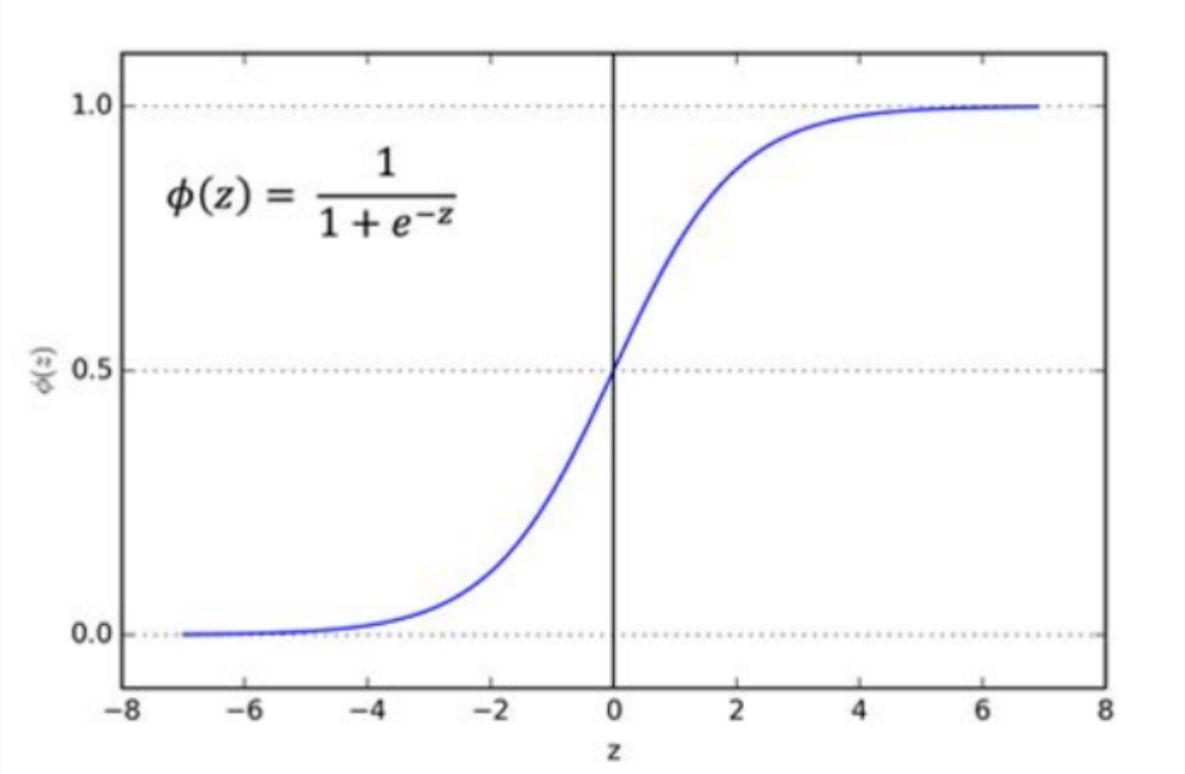
Al aumentar la dimensión y concatenar un simple vector de 1 al final de los descriptores, obtenemos una operación lineal

- Separa el espacio con un hiperplano
- Clasificador de la forma $f_{\mathbf{W},b}(\mathbf{X}) = \arg\max_{clases}(\mathbf{W}^T\mathbf{X} + b)$
- Si solo hay 2 clases, el argmax puede reemplazarse por una función de signo

Regresión Logística

Para transformar las "distancias" $\mathbf{W}^T\mathbf{X} + b$ entre los vectores y el plan en probabilidades de classes tenemos que utilizar una función de normalización, la funcion sigmoidal o *softmax*:

$$\Phi(\text{dist}) = \frac{1}{1 + e^{-\text{dist}}}$$



Regresión Logística Multiclase: Softmax

- Clasificador lineal binario y función logística:

$$P(Y=1) = \frac{1}{1+\exp^{-<\theta|\mathbf{X}>}} = \frac{\exp^{<\theta|\mathbf{X}>}}{\exp^{<\theta|\mathbf{X}>}+1} = \frac{\exp^{<\theta^{(1)}-\theta^{(-1)}|\mathbf{X}>}}{\exp^{<\theta^{(1)}-\theta^{(-1)}|\mathbf{X}>}+1}$$

$$= \frac{\exp^{<\theta^{(1)}|\mathbf{X}>}}{\exp^{<\theta^{(1)}|\mathbf{X}>} + \exp^{<\theta^{(-1)}|\mathbf{X}>}}$$

- RL para varias clases: $P(Y=c) = \frac{\exp^{<\theta^{(c)}|\mathbf{X}>}}{\sum_{j=1}^{C} \exp^{<\theta^{(j)}|\mathbf{X}>}}$, con :

$$\theta = \begin{pmatrix} | & | & | & | \\ \theta^{(1)} & \theta^{(2)} & \dots & \theta^{(C)} \\ | & | & | & | \end{pmatrix}$$

- Clase final: $\hat{c} = \arg\max_c P(Y=c)$

Regresión Logística Multiclase: Softmax

Softmax

$$Softmax : \begin{array}{c} \mathbb{R}^d \to]0;1[^d \\ (x_k)_k \mapsto (\frac{e^{x_k}}{\sum_j e^{x_j}})_k \end{array}$$

La funcion Softmax transforma un set de valores en probabilidades:

- Tienen valores entre 0 y 1
- La suma iguale 1:

$$\sum_k (\frac{e^{x_k}}{\sum_j e^{x_j}})_k = 1$$

Capa final

Por eso, se puede usar un Softmax en capa final, con una loss cross-entropia que necesita probabilidades.

Cross-Entropía

Outline : Cross-Entropía

Outline : Entropía y Información

Información y Sorpresa

Para un único evento x con probabilidad $P(x)$, la auto-información (o sorpresa) se define como:

$$I(x) = \log P(x)$$

Propiedades

- Si $P(x) = 1$ (cierto), entonces $I(x) = 0$: no hay nueva información.
- Si $P(x) \to 0$, entonces $I(x) \to \infty$: eventos muy sorprendentes conllevan mucha información.

Entropía de una distribución

- **Un evento**: Self-información
- **Muchos eventos**: La entropía mide el inverso de la cantidad promedio de información producida por una fuente estocástica de datos.

Para una variable aleatoria discreta X con distribución de probabilidad $P(x)$, la entropía de Shannon es:

$$H(X) = \mathbb{E}[I(x)] = -\sum_{x} P(x) \log P(x)$$

Este es el número esperado de bits necesarios para codificar los resultados de X usando el esquema de codificación óptimo.

H es máximo cuando $p(x) = \frac{1}{|X|}$, es decir, p es uniforme: sin estructura/información.

Entropía de una distribución

- **Un evento**: Self-información
- **Muchos eventos**: La entropía mide el inverso de la cantidad promedio de información producida por una fuente estocástica de datos.

Para una variable aleatoria discreta X con distribución de probabilidad $P(x)$, la entropía de Shannon es:

$$H(X) = \mathbb{E}[I(x)] = -\sum_x P(x) \log P(x)$$

Este es el número esperado de bits necesarios para codificar los resultados de X usando el esquema de codificación óptimo.

H es máximo cuando $p(x) = \frac{1}{|X|}$, es decir, p es uniforme: sin estructura/información.

Otra notación: La entropía de una distribución se define como:

$$H(p) = \sum_x p(x) \log \frac{1}{p(x)}.$$

Temperatura

Supongamos que muestreamos de una distribución p. Si introducimos una variable T esta cambiará la distribución a $p^{\frac{1}{T}}$.

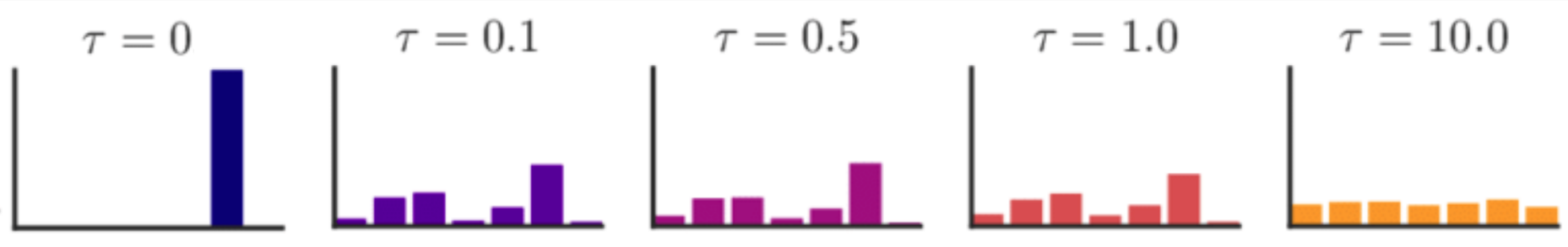


Figure 1: Misma distribución con diferentes temperaturas

$T \geq 0$ es un parámetro de **temperatura** que controla cuánta aleatoriedad queremos al muestrear de la distribución.

Temperatura y Modelos de Lenguaje Grandes

Supongamos que muestreamos de una distribución p. Si introducimos una variable T esta cambiará la distribución a $p^{\frac{1}{T}}$.

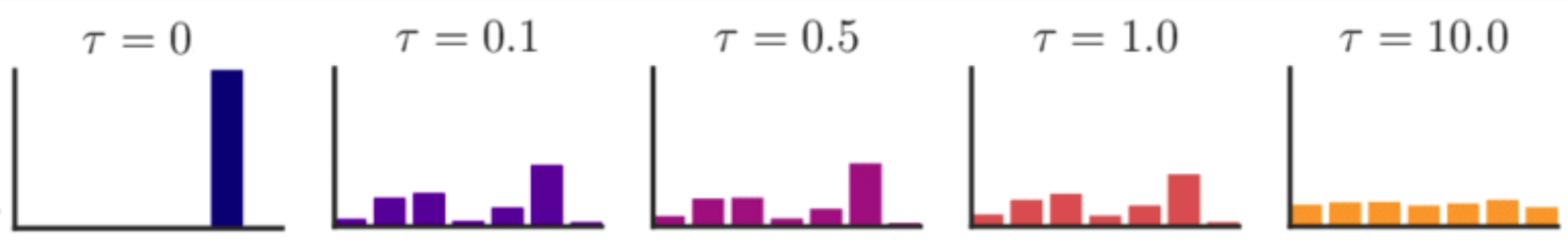


Figure 1: Misma distribución con diferentes temperaturas

$T \geq 0$ es un parámetro de **temperatura** que controla cuánta aleatoriedad queremos al muestrear de la distribución. Si es usado por un LLM:

- $T = 0$: elige determinísticamente el token más probable x_i en cada posición
- $T = 1$: muestrea "normalmente" del modelo de lenguaje puro
- $T = \infty$: muestrea de una **distribución uniforme** sobre todo el vocabulario $\mathcal{V}$

Termodinámica y Entropía

- Una medida del **grado de aleatoriedad** de la energía en un sistema
- Cuanto menor es la entropía, más ordenado y menos aleatorio es

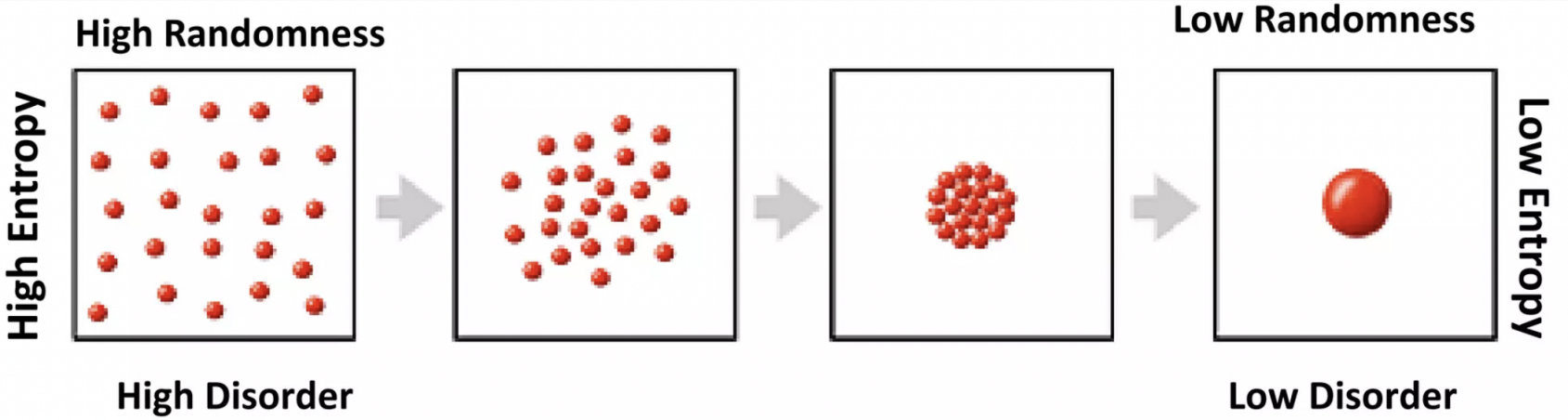


Figure 2: Entropía en termodinámica

Con temperatura y LM

Cuanto mayor es la temperatura de la sala, más entropía se crea.

Usando $T = \infty$ obtenemos una distribución uniforme de las proba sobre $\mathcal{V}$, que es lo más aleatorio posible.

Termodinámica y Entropía

- Una medida del **grado de aleatoriedad** de la energía en un sistema
- Cuanto menor es la entropía, más ordenado y menos aleatorio es

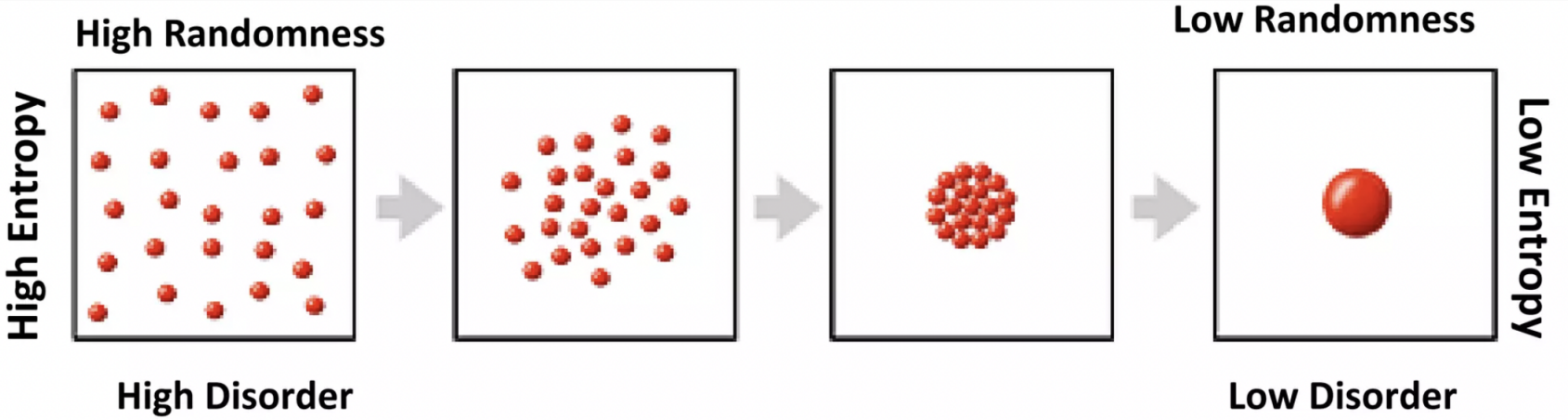


Figure 2: Entropía en termodinámica

Con temperatura y LM

Cuanto mayor es la temperatura de la sala, más entropía se crea. Usando $T = \infty$ obtenemos una distribución uniforme de las proba sobre $\mathcal{V}$, que es lo más aleatorio posible. **Una distribución uniforme contiene poca información, ya que todos los eventos tienen la misma probabilidad.**

Entropía e Información en Bits

La entropía mide el **número mínimo promedio de bits** necesarios para codificar sin pérdidas muestras de p. En otras palabras, si un evento es muy probable, debe codificarse óptimamente con menos bits para ahorrar espacio.

Intuición del porqué

- Cada bit divide el espacio de posibilidades en dos. Para aislar un evento que ocurre con probabilidad $p(x)$, se necesitan suficientes bits para distinguirlo entre $1/p(x)$ eventos igualmente probables — lo que requiere $\log_2(1/p(x))$ bits.
- Por ejemplo, si $p(x) = 1/8$, entonces x recibe un código de 3 bits (p. ej., `110`), ya que $\log_2(8) = 3$. Hay exactamente 8 de esos códigos de 3 bits — uno para cada resultado.
- Los códigos óptimos asignan longitudes de bits proporcionales a $-\log_2 p(x)$. La entropía es el límite teórico de compresión impuesto por la información.

Outline : Intuición de la Cross-Entropía

Cross-Entropía

Cross-Entropía

$$H(p, q) = \sum_x p(x) \log \frac{1}{q(x)},$$

mide el número esperado de bits necesarios para codificar una muestra $x \sim p$ usando el esquema de compresión dado por el modelo q (representando x con un código de longitud $\frac{1}{q(x)}$).

Una propiedad crucial es que la cross-entropía $H(p, q)$ acota superiormente la entropía $H(p)$:

$$H(p, q) \geq H(p),$$

Si queremos minimizar la entropía, necesitamos encontrar un q (nuestra predicción) lo más cercano posible a p.

Lo que nos da al final

Con los redes neuronales

- La x-entropía sirve a comparar distribuciones de probabilidades
- Despues de un softmax tenemos una pdf
- Un encoding one-hot de las etiquetas es tambien una pdf

Cross-entropía calcula una divergencia entre las distribuciones de probabilidad.

$$\ell(y, \hat{y}) = -\sum y_i \log(\hat{y}_i) = -\sum y_i \log(p(y_i)) = -\log(p(\hat{y}_k))$$

Donde $y = (\mathbb{1}_k)_i$ un encoding one-hot de la etiqueta real y_k.

Questions?

References i