Introducción a las redes neuronales: Redes generativas

M. Sc. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Escuela de Computación, bachillerato en Ingeniería en Computación,
PAttern Recongition and MAchine Learning Group (PARMA-Group)

29 de mayo de 2019

Resumen

Este material está basado en el capítulo de redes generativas, del libro *Deep Learning* de Ian Goodfellow et. all[1].

1 Teoría de juegos y modelos conexionistas

Las redes generativas combinan ingeniosamente la **teoría de juegos** (rama de las probabilidades) con los **modelos conexionistas** o redes neuronales, originalmente propuestas en el artículo *Generative Adversarial Nets, GoodFellow et. all, 2014.* Los autores proponen el siguiente escenario de juego: Un agente **generador** de muestras sintéticas

$$\overrightarrow{x}_k = g\left(\overrightarrow{z}, \overrightarrow{w}^g\right)$$

debe competir contra un agente **adversario o discriminador**

$$\widetilde{p}_j = d\left(\overrightarrow{x}_j, \overrightarrow{w}^d\right)$$

donde:

- g es el modelo correspondiente al **generador de muestras**, según lo propuesto por los autores, una red neuronal con parámetros a ajustar $\overrightarrow{w}^g$ y entrada $\overrightarrow{z}$, la cual corresponde a una generadora de entropía con una distribución específica, la cual le permite al modelo generar las muestras con tal distribución. La salida del modelo consiste entonces en la muestra artificial o falsa $\overrightarrow{x}_k^f$ para la cual sabemos no es real, lo que se representa en la etiqueta $r_k = 0$.
- d corresponde al modelo **adversario o discriminador** con parámetros $\overrightarrow{w}^d$ (también una red neuronal según los autores), el cual tiene por objetivo

estimar la probabilidad $\widetilde{p}_j$ de que la muestra $\overrightarrow{x}_j$ sea real, representado en la **etiqueta de veracidad** $r_j = 1$ cuando es real, y $r_j = 0$ de otro modo. Basado en ello, podemos construir una función de densidad binomial

$$p\left(\overrightarrow{x}_j, r_j\right) = \left(1 - \widetilde{p}_j\right)^{1-r_j} \widetilde{p}_j^{r_j}$$

la cual describe la probabilidad de que la muestra $\overrightarrow{x}_j$ sea real o artificial. Para un conjunto de muestras $X = \{\overrightarrow{x}_1, \overrightarrow{x}_2, \ldots, \overrightarrow{x}_N\}$ y sus respectivo conjunto de etiquetas de veracidad $R = \{r_1, \ldots, r_N\}$. El discriminador buscará entonces maximizar al siguiente función de verosimilitud:

$$P\left(X|R\right) = \prod_{i=1}^{N} p\left(\overrightarrow{x}_i, r_i\right) \tag{1}$$

para $r_i \in R$ y $\overrightarrow{x}_i \in X$.

El objetivo del agente generador de muestras es generar las muestras lo más reales posibles, para **engañar** al agente discriminador, mientras que por el contrario, el agente discriminador tiene por meta **discriminar** de la forma más acertada entre las muestras creadas por el agente generador y las muestras reales representadas en la matriz

$$X^r = \begin{bmatrix} | & | & | & | \\ \vec{x}_1^r & \vec{x}_2^r & \ldots & \vec{x}_N^r \\ | & | & | & | \end{bmatrix} \qquad R^r = \left[1 \ldots 1_N\right],$$

con su correspondiente matriz de veracidad R^r.

Lo anterior se expresa en lo que los autores llaman un juego de suma nula, con la función $v\left(g, d\right)$ la cual determina la **recompensa** al discriminador. Entre **más alta la recompensa, mejor trabajo estará haciendo el discriminador** en adivinar qué muestras son artificiales o son reales (y por ende, pertenecen al conjunto de muestras X). El objetivo entonces del **generador al ser engañar al discriminador, es entonces disminuir la recompensa al discriminador**, lo cual se expresa entonces en el siguiente problema de minimización/maximización:

$$\arg \min_g \max_d v\left(g, d\right)$$

Existen muchas formas de elegir la función de recompensa al discriminador, una de ellas es utilizando el **logaritmo de la función de verosimilitud, para un conjunto de muestras** $X = \left[X^r, X^f\right]$ y sus respectivas etiquetas de veracidad $R = \left[R^r, R^f\right]$, donde la matriz

$$X^f = \begin{bmatrix} | & | & | & | \\ \vec{x}_1^f & \vec{x}_2^f & \ldots & \vec{x}_N^f \\ | & | & | & | \end{bmatrix}$$

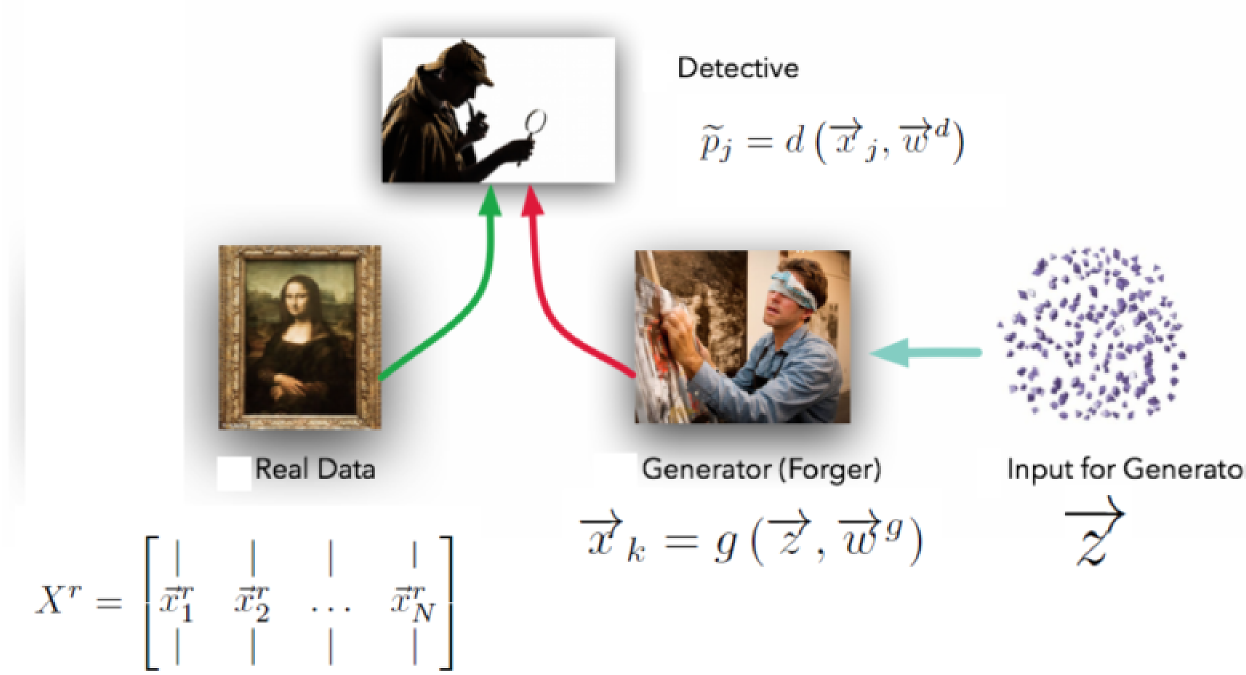

Figura 1: Aprendizaje adversario generativo. Tomado de `https://bit.ly/2ttFZtv`.

contiene muestras construidas por el generador por lo que $\vec{x}_k^f = g\left(\overrightarrow{z}, \overrightarrow{w}^g\right)$ y $R^f = [0 \dots 0_N]$. Tal logaritmo de la función de verosimilitud se basa entonces en la Ecuación 1:

$$v\left(g,d\right) = \log\left(P\left(X|R\right)\right) = \sum_{i=1}^{N} \log\left(\left(1-\widetilde{p}_j\right)^{1-r_j} \widetilde{p}_j^{r_j}\right)$$

$$\Rightarrow v\left(g,d\right) = \frac{1}{N}\sum_{i=1}^{N} \left(1-r_j\right) \log\left(1-d\left(\overrightarrow{x}_i, \overrightarrow{w}^d\right)\right) + r_j \log\left(d\left(\overrightarrow{x}_i, \overrightarrow{w}^d\right)\right)$$

La Figura 1 ilustra el concepto de aprendizaje adversario generativo.

Referencias

[1] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.