Explicabilidad de Redes Neuronales Profundas

Andrea Quintanilla Carranza

11 de diciembre de 2021

1. Motivación

Las Redes neuronales artificiales profundas (RNP's) son una familia de funciones que se contruyeron con inspiración en las redes neuronales naturales para realizar tareas de decisión automática, de clasificación o de regresión.

Las RNPs han mostrado un gran poder de predicción (exactitud) en distintas áreas industriales y científicas; su uso se ha extendido fuertemente a la par de la disponibilidad de grandes conjuntos de datos y de avances en el poder de procesamiento de las computadoras.

Aunque han mostrado un desempeño excepcional en su exactitud y actualmente ese criterio es el más utilizado para cuantificar el desempeño de modelos; existen otros atributos igualmente importantes para evaluarlos, como son: la interpretabilidad, la robustez frente a perturbaciones, existencia de intervalos de confianza para las predicciones, aspectos de seguridad, requerimientos legales, impacto ecológico, entre otros. Y aunque dichos aspectos son cruciales en las aplicaciones, muchas RNP's carecen de ellos. En este trabajo siguiendo la exposición en [Sam+21] y en [Rud19], daremos un panorama parcial de los esfuerzos que se han hecho para subsanar su falta de interpretabilidad y sobre algunos de los obstáculos vigentes en esos esfuerzos.

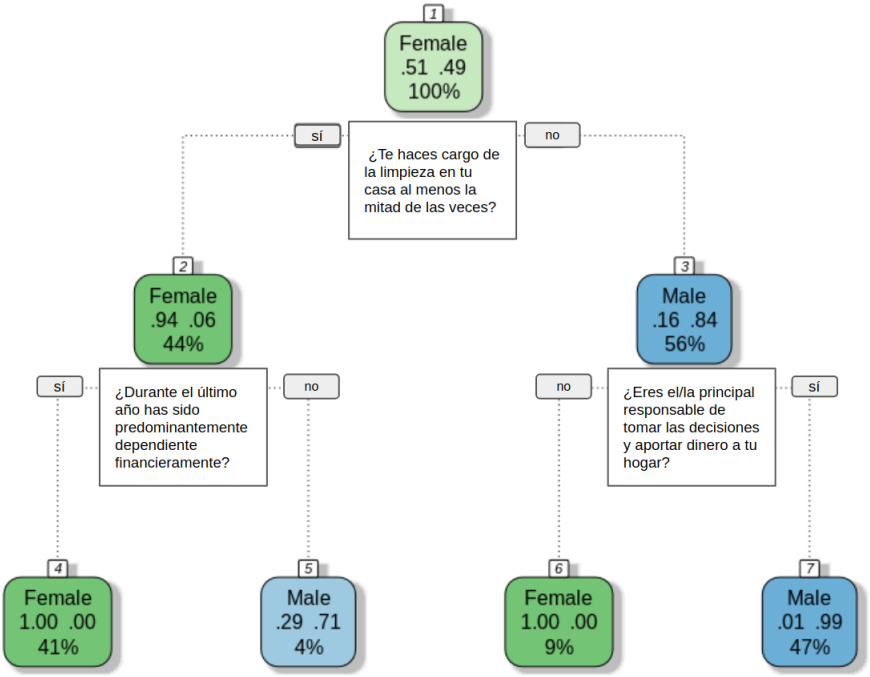
2. Interpretabilidad

La interpretabilidad o la capacidad de comprender un modelo, es una noción específica del dominio de aplicación y de los intereses del usuario. Por eso es difícil acordar una definición totalitaria. Hay un espectro amplio entre modelos totalmente trasparentes, en los que entendemos cómo se relacionan todas las variables de entrada entre sí y con la variable de salida; y hay otros que tienen sólo ciertas restricciones que ayudan a comprenderlos mejor. Esas restricciones pueden ser para simplificar el modelo (como selección de características) o para sólo admitir valores que tenga una interpretación en el fenómeno real que se quiere modelar.

Ejemplos más específicos de características que pueden llevar a que un modelo sea interpretable, son: ser susceptible de representarse con una estructura gráfica o textual; involucrar sólo algunos predictores relevantes o detallar la contribución a la predicción de cada uno; indicar una jerarquía de importancia entre los predictores que se utilizan; evitar redundancia entre criterios de decisión, entre otras. Ejemplos de modelos interpretables pueden ser los modelos lineales o los árboles de decisión. En ellos, el valor de los coeficientes y un orden jerárquico de los predictores, respectivamente, indican la importancia de cada predictor para que el modelo produzca una determinada salida (decisión/clasificación/predicción) a partir de cierta entrada (un conjunto de predictores).

En la figura 1, observamos un árbol de decisión. Éste predice si un grupo de personas está conformado por mujeres o por hombres, dependiendo del porcentaje de respuestas afirmativas a una serie de preguntas. Cada nodo representa un corte en el espacio y dependiendo a que nodo terminal pertenezca un grupo, se le asigna una clase: femenino (verde) o masculino (azul).

Figura 1: Ejemplo de árbol de decisión.



Así, podemos dar sentido a la clasificación que está realizando el árbol. Notamos que una forma en que distingue entre hombres y mujeres es que los hombres tienden a dedicar menos tiempo que mujeres a tareas domésticas y que también tienden a tener mayor independencia económica que las mujeres.

En general, entender un modelo da confianza a sus usuarios. Existen ya varios casos en los que se han utilizado modelos **no** interpretables, conocidos como "Modelos de caja negra", y que han tomado decisiones equivocadas con consecuencias graves. Pueden ser cajas negras porque (a) la función es muy complicada, o porque (b) son propiedad intelectual. Una rama en donde surgen consecuencias graves al ocuparlos, es en medicina. Por ejemplo, se encontró que una red neuronal que recibía radiografías para detectar neumonía, estaba detectando la palabra "portable", que definía el equipo con el que fue tomado la radiografía, en lugar de centrarse en la imagen del tejido. Otros ejemplos dañinos se han dado en modelos que miden la inocuidad del aire, y en decisiones automáticas para despido de profesores. ([Rud19])

Además de que un modelo no sea interpretable puede tener consecuencias negativas en la vida de las personas, existen beneficios directos cuando son interpretables, por ejemplo ([AHG21], [Sam+21]):

- Pueden señalar relaciones interesantes entre las variables (y sugerir causalidad, correlación);
- por lo anterior, pueden ser utilizados para ir mejorando el modelo iterativamente;
- necesarios cuando se requiere transparencia de las decisiones que se basan el modelo;

- ayudar a revisar equidad, es decir, que las decisiones que afecten a personas no estén tomando como criterio una variable protegida como raza o género;
- también puede señalar problemas con el conjunto de datos utilizado (por ejemplo, correlaciones espurias).
- mayor confianza en su capacidad de generalización (de su desempeño en datos fuera del conjunto de entrenamiento).

Como ejemplo de correlaciones espurias y de confianza en la capacidad de generalización, en la siguiente figura se muestra una red neuronal que ganó un concurso (The PASCAL Visual Object Classes Challenge), identificando marcas de agua asociadas a caballos en el conjunto de imágenes de entrenamiento, en lugar de identificando la forma del caballo. Esa asociación se detectó tiempo después del concurso, cuando se le aplicó LRP -un algoritmo que se revisará más adelante- para entenderla.

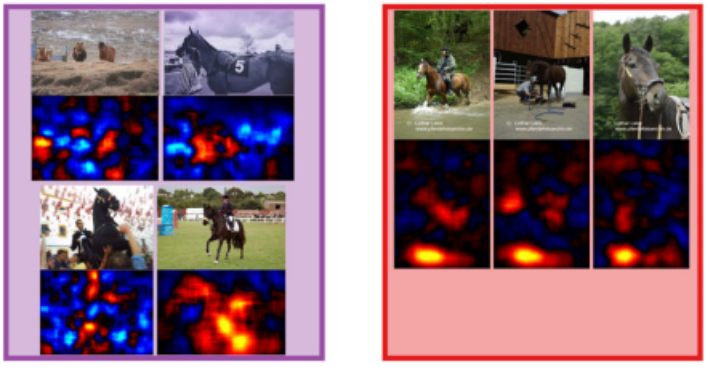


Figura 2: LRP aplicado a una red neuronal que etiqueta imágenes. Conjunto de imágenes donde las atribuciones se centran en la figura del caballo (izquierda) y conjunto de imágenes donde las atribuciones se centran en una marca de agua (derecha). Tomado de [GH21]

Luego, puede haber casos en donde entender las relaciones entre las variables no es de interés para el usuario pero aún así, entender los patrones que está aprendiendo el modelo da mayor confianza en su capacidad de generalización.

Por otro lado, los modelos interpretables pueden conllevar mucho más trabajo tanto en términos de complejidad computacional, como en necesidad de expertos en la materia. Esto es debido a que mejorarlos muchas veces se traduce en añadir varias restricciones a un proceso de optimización. Resolver problemas de optimización con varias restricciones es complejo, y proponer restricciones adecuadas implica un conocimiento profundo en la materia. [Rud19]

Para concretar lo anterior, regresemos al ejemplo de árboles de decisión, para que un clasificador de este tipo fuera interpretable, además de elegirlo en esa familia (que nos ofrece una jerarquía visual de los predictores según su importancia; figura 1), desearíamos que ese árbol no fuera muy amplio, para que efectivamente, podamos entender las razones de la clasificación. Para medir qué tan amplio es un árbol podemos considerar distintas propiedades, por ejemplo su número de hojas.

Dado un conjunto de observaciones de entrenamiento indexadas por $i = 1, .., n$, el problema de optimización asociado sería entonces, encontrar un árbol en la familia $\mathcal{F}$, de árboles binarios, que minimice tanto el número de observaciones mal clasificadas como el número de hojas del árbol:

$$\min_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^{n} 1_{[\text{observación } i \text{ es mal clasificada por } f]} + \lambda \times \text{número de hojas de } (f) \right)$$

El parámetro λ se puede interpretar como el error de clasificación que se estaría dispuesto a sacrificar para que el árbol tuviera menos hojas. Si tenemos un árbol A con un porcentaje $10\,\%$ de error, con un número h de hojas, y fijamos λ is 0.01, para que otro árbol con $h+1$ hojas tenga un valor menor en la función objetivo del problema de arriba, debería de tener un error de clasificación menor a $9\,\%$. De otra manera, A tendría un desemepeño mejor.

Un obstáculo con los problemas de optimización de esa forma, donde $\mathcal{F}$ puede ser otra familia de modelos interpretables, como cualquier modelo lógico (clasificadores que se pueden leer como enunciados lógicos), es que generalmente son NP-duros, es decir, no se pueden resolver en tiempos razonables computacionalmente.

3. Notación

En general, supondremos que la RNP ya está entrenada, es decir, tiene unos parámetros (pesos e interceptos) θ fijos, y corresponde a una función:

$$f : \mathbb{R}^p \to \mathbb{R}$$

que recibe p predictores de entrada, $x = (x_1, \ldots, x_d)$, y devuelve un valor real. El resultado $f(x)$ en algunos casos será una probabilidad en el que una decisión o clasificación es basado, y en otros la solución de un problema de regresión. Se especificará de acuerdo a los métodos y ejemplos correspondientes.

4. Nociones generales de algoritmos de XAI

Buscando no renunciar a los logros que se han obtenido con las RNP y subsanar su falta de interpretabilidad, ha crecido en los últimos años una rama denominada "Explicabilidad", conocida como XAI. Aunque se le ha puesto atención ya desde los 90's, ha crecido especialmente desde la introducción de regulaciones de datos en EUA en el 2018 ([Rud19]). La Explicabilidad se distingue de la interpretabilidad porque los modelos a los que se les busca aplicar son intrínsecamente no interpretables; así, trata de entender tendencias en las estrategias decisión de un modelo, de manera indirecta. La idea de la Explicabilidad es construir **otro** algoritmo que dada una entrada, cuantifique la importancia que da la caja negra a cada variable de la entrada para determinar la predicción correspondiente.

Con mayor detalle, si:

$$f : \mathbb{R}^p \to \mathbb{R}$$

es una caja negra -un modelo no interpretable (ya entrenado)-, que recibirá como entrada un vector $x = (x_1, \ldots, x_p)$ y como salida se produce un valor real $f(x)$. Por medio de algoritmos de explicabilidad, que revisaremos con detenimiento a continuación, se buscará determinar cuáles predictores x_i contribuyen a favor o en contra del valor $f(x)$. Es decir, en general, los algoritmos de explicabilidad tratarán de rastrear (1) cuáles variables x_i tienen una responsabilidad en el valor $f(x)$; y (2) -entre las que sí tengan un efecto en $f(x)$- se busca distinguir cuáles aumentan el valor de $f(x)$ (aportan evidencia en favor de su pertenencia a la clase) y cuáles contribuyen a que decrezca (contra-evidencia).

Así, un algoritmo explicativo recibirá una caja negra $f : \mathbb{R}^p \to \mathbb{R}$ y generará un vector de atribuciones $(a_1, \ldots, a_p)^t \in \mathbb{R}^p$, donde se buscará que la norma de la entrada a_i sea proporcional a la importancia de la dimensión i en el cálculo de $f(x)$; mientras que su signo, indicará si trabaja como evidencia o como contra-evidencia.

Para ejemplificar uno de los tipos de resultados que se busca, en la figura 3 se aplicó un algoritmo de explicabilidad, llamado Gradiente integrado (GI), a una red neuronal que etiqueta imágenes como pertenecientes a una de 10 clases: los dígitos del 0 al 9. En este caso, una imagen es una entrada x, cada pixel corresponde a un predictor x_i con valores del 0 al 256, estos valores representan una escala de grises. En la figura de la izquierda se muestra la imagen original.

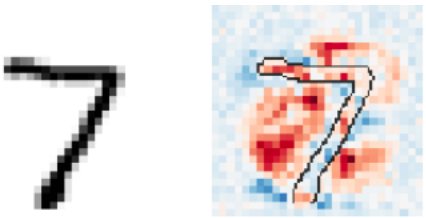


Figura 3: Gradiente integrado (GI) aplicado a una red neuronal aplicada en MNIST.
Imagen original (izquierda) y atribuciones asignadas por GI (derecha)

El algoritmo de GI opera sobre la red neuronal y la imagen de entrada, a cada pixel asigna un valor real negativo o positivo. En la imagen de la derecha, los valores positivos -coloreados en rojo- representan evidencia a favor de que el número pertenezca a la clase del dígito 7, mientras que los pixeles a los que GI asignó números negativos -coloreados en azul- representan evidencia en contra. La intensidad del coloreado es proporcional a la magnitud del valor asignado por GI y dicha magnitud representa el nivel de importancia de dicho pixel. Por ejemplo, observamos que el triángulo interno pareciera ser un distintivo importante del 7 frente a otras clases, pues está coloreado en rojo. Mientras que la punta inferior izquierda, está coloreada de azul intenso, es decir, el que esa región esté vacía disminuye la probabilidad $f(x)$. Esto último podría estar indicando que quizá en el conjunto de entrenamiento, la mayoría de los sietes tendían a tener una punta más extendida.

Otra característica de las atribuciones que toman en cuenta algunos algoritmos de XAI, es que (3) el conjunto no nulo de atribuciones a_i sea minimal (*feature selection*), puesto que si el número de predictores es amplio y no pedimos esta propiedad, sería igualmente complicado darles sentido a las atribuciones.

Cabe señalar que aunque la abstracción anterior engloba de cierta manera a la mayoría de los algoritmos de explicabilidad que revisaremos en las siguientes secciones, no todos siguen exactamente esa estructura. Por ejemplo, hay algunos métodos de XAI que no buscan que a_i sea minimal. Otra razón es que el sentido que se da a las atribuciones a_i, depende del dominio de aplicación.

Por ejemplo, considérese la tarea de Análisis de Sentimiento de Procesamiento de Lenguaje Natural, cuando se desean distinguir entre opiniones favorables de usuarios de algún servicio o producto a partir de sus opiniones en foros. En este problema podemos considerar las palabras que aparecen en un comentario como predictores x_i y a la salida una clasificación entre comentarios favorables o no favorables. En este caso sería importante identificar palabras de los comentarios que hacen que la frase invierta su significado (de favorable a desfavorable, o viceversa); es decir palabras que resulten en contra-evidencia de ser favorables.

Mientras que en sistemas de recomendación, como una plataforma que recomiende productos, la noción de contra-evidencia puede no tener significado directo.

Otra anotación importante es que **entender** profundamente un modelo sería controlar y desenrollar cada aspecto de una predicción del modelo: aspectos inherentes a los datos; aspectos inducidos por el modelo; impacto de las explicaciones en la audiencia ([AHG21]). Sin embargo, la forma general que están siguiendo dichos algoritmos de explicabilidad no apunta a **cómo** la información en los predictores está siendo empleada para determinar la salida. Es decir, no revela las relaciones entre los distintos predictores ni la forma de relación (lineal, exponencial, etc.) entre un predictor y la salida. En el mejor de los casos sólo apunta a **cuáles** de los predictores son más relevantes para el modelo en determinada decisión. Por ello, Rudin en [Rud19] observa que el término "Explicación", no es adecuado para referirse a las atribuciones $(a_1, \ldots, a_d)^t$. Es más franco pensarlas como "tendencias" o "resúmenes de estadísticas".

Para asignar las atribuciones se han propuesto algoritmos de explicabilidad que exploran técnicas muy distintas entre sí. Existen diferentes maneras de clasificarlos; una usual es entre atribuciones globales vs. locales. Los algoritmos de XAI, que construyen atribuciones a nivel local, se centran en encontrar las "razones" de que la RNP asigne cierto valor $f(x)$ a **una sola entrada** x. Luego, un algoritmo explicativo local recibirá una caja negra $f : \mathbb{R}^p \to \mathbb{R}$ una entrada $x \in \mathbb{R}^p$ y generará un vector de atribuciones $(a_1, \ldots, a_d)^t \in \mathbb{R}^p$. Estos son útiles cuando las decisiones automáticas están afectando a personas, como pacientes, juzgados y aplicantes a trabajos o escuelas, que necesitan saber las razones de decisión en una sola muestra (la del individuo).

Existen otras aproximaciones que intentan dar sentido a nivel global. Por ejemplo, Activación máxima (activation maximization), que construye ejemplos prototípicos como:

$$x^* = argmax_x f(x)$$

es decir, ejemplos x^* en los que la red tiene máxima confianza en su pertenencia a la clase de interés. Este método se revisará con mayor detalle en una sección posterior.

Las dos vías -la global y la local- tienen sus limitaciones. La global porque revela poco sobre las estrategias de decisión ejemplos específicos (como x cercanos a la frontera de decisión). Un análisis local es importante porque pueden haber distintos motivos que llevan a una RNP a clasificar dos datos en la misma clase. Mientras que un problema del análisis local, es que si los conjuntos de datos estudiados son amplios, es difícil identificar las estrategias del modelo a través de la observación de varios ejemplos.

Siguiendo la línea del artículo de referencia ([Sam+21]), nos centraremos en métodos locales.

Otra forma de clasificación de lo métodos de XAI, que nos será útil para distinguir sus características es según si se basan en alguna (o varias) de las siguientes ideas ([AHG21]):

- Cálculo de **gradiente**: los métodos de este tipo estudian cambios respecto a la entrada x_i, en la función de pérdida ℓ con la que se entrena el modelo, o en el modelo mismo f para determinar el valor de la atribución a_i.
- Los que se basan en **comparaciones**, proponen alguna entrada $\widetilde{x}$ neutra respecto al modelo. Es decir, si $f(x) > 0$ implica que x se clasifica en la clase positiva y si $f(x) < 0$, x se clasifica en la negativa; un elemento neutro $\widetilde{x}$ respecto al modelo sería uno tal que $f(\widetilde{x}) = 0$. Una vez que se tiene esa raíz, se calcula la atribución a_i utilizando x_i, x_i^* y su efecto en el modelo. La idea detrás de ese punto neutro se debe a que si deseamos encontrar la causa de cierto cambio, debemos partir de la ausencia de esa causa (de un punto neutro) para poder comparar las distintas dimensiones. En varias redes, pero no en todas, es posible encontrar un punto base $\widetilde{x}$ tal que

$f(\widetilde{x}) \approx 0$.

- Otro enfoque que se ha propuesto es el **axiomático**, en éste se proponen reglas que deben seguir las atribuciones a_i con respecto a f y se derivan fórmulas para su cálculo, que las haga respetar dichas reglas. En algunos casos, como en GI -método que revisaremos con mayor detalle- el algoritmo que los cumple es único, mientras que en otros, como en el caso de LRP, existen diversos algoritmos que las cumplen.

En la figura 4, se esquematizan las dos clasificaciones mencionadas anteriormente. Las terminaciones del árbol son algunos métodos que pertenecen a esas clases. Cabe destacar que algunos métodos los incluimos en distintas terminaciones porque mezclan elementos de más de una rama, como lo es GI.

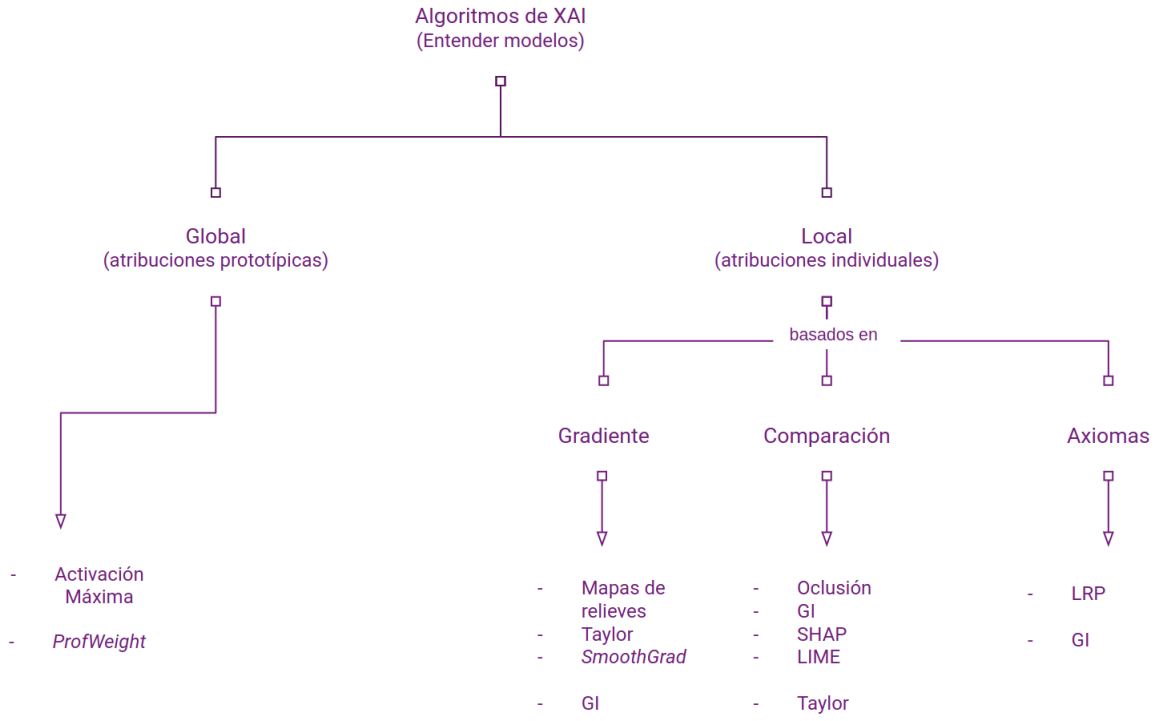


Figura 4: Un esquema de clasificación de métodos en XAI.

A continuación, describiremos algunos de los algoritmos que se mencionan en la figura 4. Se eligieron porque, como se menciona en [Sam+21], son representativos de la gama de ideas que se están explorando actualmente: Explicaciones localmente interpretables modelo-agnósticas (LIME, por sus siglas en inglés, véase [RSG16]); Propagación de relevancia por capa (LRP, también por sus siglas en inglés) y Descomposición de Taylor (ambos en [Bac+15]); Oclusión (consúltese [ZF14]); Gradiente integrado (GI, [STY17]) y, finalmente, Maximización de activación ([Erh+09]).

5. Descripción de algunos Algoritmos de XAI

5.1. Descomposición aditiva

La idea es aproximar a $f(x)$ como una suma:

$$f(x) \approx \sum_{i=1}^{d} a_i \tag{1}$$

donde cada término a_i, denominado relevancia (o atribución), dependa sólo de un predictor x_i. Se busca que dichos términos, además, tengan la interpretación: $a_i < 0$ si x_i contribuye con evidencia en contra de estructura de la clase de interés. Y $a_i > 0$ si x_i contribuye con evidencia de su presencia. Hay distintas maneras coherentes de construirlas. El nombre que original con el que se introdujo este método en [Bac+15], es *Pixel-wise descomposition*, pensando en su aplicación a imágenes. Pero se puede adaptar a otras aplicaciones de clasificación y regresión sin mayores ajustes.

5.1.1. Descomposición de Taylor

Una primera aproximación es utilizar el gradiente ya que el gradiente es una manera natural para identificar su sensibilidad a ciertos cambios. Por el teorema de Taylor sabemos que si f es diferenciable, $\widetilde{x}$ es una raíz de f, es decir, $f(\widetilde{x}) = 0$ y x es un punto cercano a $\widetilde{x}$, $f(x)$ puede ser aproximado con una función lineal:

$$f(\boldsymbol{x}) \approx \sum_{i=1}^{d} \underbrace{[\nabla f(\widetilde{\boldsymbol{x}})]_i \cdot (x_i - \widetilde{x}_i)}_{a_i}$$

Esta expansión es una suma pesada donde a_i puede ser interpretado como la contribución del i-ésimo predictor a $f(x)$. Esta interpretación se justifica si observamos que a_i será amplio solamente si sucede que:

1. $x_i \neq \widetilde{x}_i$: los predictores son diferentes que los del valor neutro $\widetilde{x}_i$ de referencia.
2. y $[\nabla f(\widetilde{x})]_i \neq 0$: f es sensible a variaciones en la variable x_i localmente.

Una explicación puede formarse entonces con los vectores de atribuciones, $(a_i)_i$, y presentarse como un histograma (para pocos predictores) o con un *heatmap* (en imágenes).

Aunque los gradientes generalmente se pueden calcular de manera automática para muchos tipos de RNP, existen varios problemas con esta aproximación. Las (1) aproximaciones lineales pueden no ser suficientes para redes complejas; también por su complejidad, (2) encontrar una raíz $\widetilde{x}$ cercana al punto x de interés puede ser complicado por los mismo y, además, (3) sufren de algo conocido como "Desintegración de gradiente" (*Shattered gradient*).

La Desintegración de gradiente, se presenta en las RNP y crece junto con la profundidad de la red y signigica que los gradientes en ellas, localmente se parecen a ruido blanco. Se ilustra en la figura 5.1.1; el modelo f es una red entrenada para predecir si aparece una barra en una imagen. Las imágenes que se muestran son una secuencia de imágenes en un video. Para todas ellas $f(x) > 0$ indica que el modelo detectó una barra; sin embargo, observamos que los cambios en el gradiente (para esta secuencia de imágenes similares) varía fuertemente. Como localmente el gradiente cambia más rápidamente que la predicción, las contrucciones basadas en él no serán consistentes.

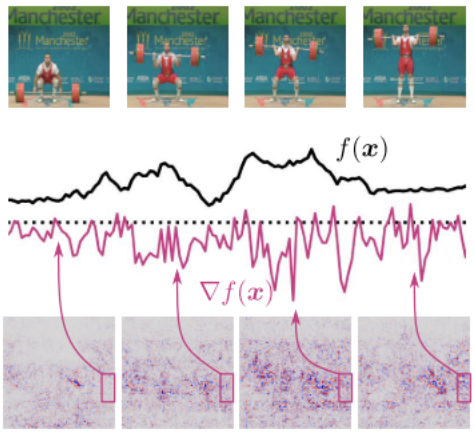


Figura 5: Desintegración de gradiente. Gradiente con alta variablidad en una red VGG-16.). Tomado de [Sam+21]

5.1.2. Propagación de relevancias por capas (LRP)

En esta segunda propuesta de la descomposición, para calcular los valores de a_i, se utiliza la estructura de las RNP y se propone un algoritmo recursivo que avanza hacia atrás y ordenadamente por capas. La primera capa son las entradas x, la última la salida $f(x)$.

Se define la relevancia $R_i^{(l)}$ como la relevancia de la i-ésima neurona de la capa l. De tal manera que la siguiente ecuación se mantenga:

$$f(x) = \cdots = \sum_{i=1}^{l+1} R_i^{(l+1)} = \sum_{i=1}^{l} R_i^{(l)} = \cdots = \sum_{i=1}^{p} R_i^{(1)}$$

Las atribuciones que nos interesan serían entonces las relevancias de la primera capa (donde cada neurona corresponde a un único predictor). La ecuación anterior se puede interpretar como una regla de conservación de la relevancia através de las capas. Entonces, la relevancia de la primera capa funcionará como una descomposición de $f(x)$ en p términos, uno por cada predictor x_i.

Una descomposición como la deseada no es única y dicha regla no se asegura que los signos de sus términos tengan el significado de evidencia y contraevidencia que deseamos. La siguiente, es una primera propuesta sencilla que cumple con la regla y se basa en la razón entre la relación local y global de las activaciones.

Para detallar este método, considérese la siguiente notación. Fijemos una neurona j de una capa $l+1$. A las neuronas de la capa l las indexamos con i; denotamos por $\hat{x}_i$ a la salida de la i-ésima neurona de la capa l; y con w_{ij} al peso conectando a la i-ésima neurona de la capa l con la j-ésima neurona de la capa $l+1$ (véase la figura 6).

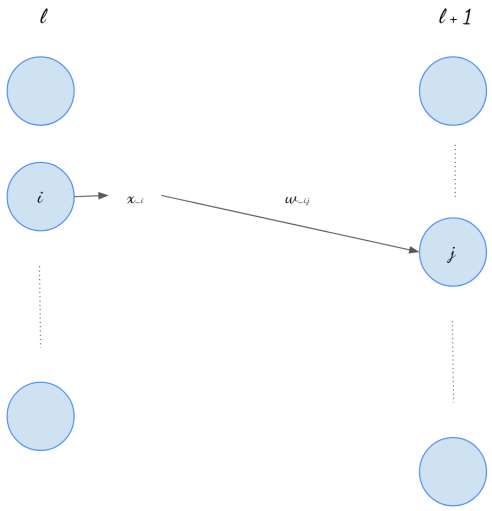


Figura 6: Notación de RNP para LRP.

También definimos:

$$z_{ij} = \hat{x}_i w_{ij}$$
$$z_j = \sum_i z_{ij} + b_j$$

así, z_{ij} es el mensaje de entrada de la i-ésima neurona a la j-ésima neurona. A la suma total de mensajes de entradas de la capa l a la j-ésima neurona de la capa $l+1$ más un intercepto b_j, lo denotamos por z_j. Y a la función de activación, con g.

También definimos como $R_{i\leftarrow j}^{(l,l+1)}$ al mensaje de **relevancia** de la neurona j de la capa $l+1$ a la neurona i de la capa l. Así, una propuesta que asegura la ley de conservación, es:

$$R_i^{(l)} = \sum_j R_{i\leftarrow j}^{(l,l+1)}$$

con:

$$R_{i\leftarrow j}^{(l,l+1)} = \frac{z_{ij}}{z_j} \cdot R_j^{(l+1)}$$

definiendo $R_k^{(l+1)} = f(x)$, donde k es el índice de la última capa. Es decir, una vez calculada la relevancia $R_j^{(l+1)}$ asociada a la capa $l+1$, ésta se distribuirá a la neurona i de la capa anterior de manera proporcional al valor z_{ij}, que conecta a la neurona j con la i. La suma de los mensajes de la capa $l+1$ a la neurona i de la capa l será la relevancia $R_i^{(l)}$. Las atribuciones serán: $a_i = R_i^{(1)}$.

En la figura que sigue se ilustra del lado izquierdo una red pequeña y en en lado derecho la manera en la que se distribuirían los mensajes de relevancia.

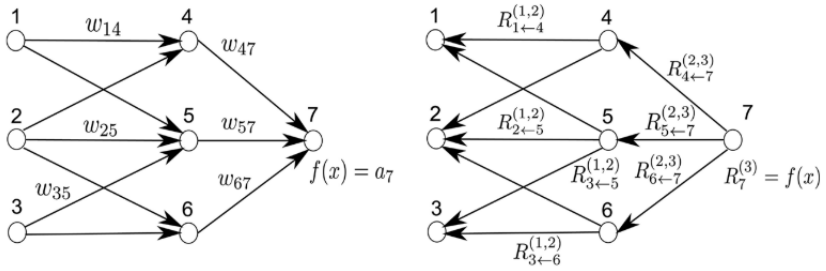


Figura 7: Esquema de algoritmo LRP. Tomado de [Lap+19].

Existen otras variantes, por ejemplo, si como función de activación se utiliza ReLU y proponemos las relevancias:

$$R^{(l,l+1)}_{i\leftarrow j} = \left(\alpha \cdot \frac{z^+_{ij}}{z^+_j} + \beta \cdot \frac{z^-_{ij}}{z^-_j}\right) \cdot R^{(l+1)}_j$$

donde los mensajes de entrada se separan como:

$$z^+_{ij} = \begin{cases} z_{ij} & ; z_{ij} > 0 \\ 0 & ; \text{ e.o.c.} \end{cases} \text{ y } z^-_{ij} = \begin{cases} z_{ij} & ; z_{ij} < 0 \\ 0 & ; \text{ e.o.c.} \end{cases}$$

y b^+ y b^- se separan de igual manera. Entonces α y β son parámetros que centran las relevancias en predictores que activan, o desactivan a la red, respectivamente. Esto debido a que con esa elección de función de activación, una neurona sólo se activará si su salida y su entrada es positiva.

Como veremos más adelante, al igual que varios de los demás métodos sólo se ha demostrado su desempeño en la práctica y visualmente. Ahondaremos también después en la necesidad de hacerlo visualmente.

Un aspecto que se debe de investigar sobre este método es que es que el método de propagación a través de la capas es *greedy* o voraz. Es decir, son heurísticas que se proponen en elegir soluciones coherentes a nivel de transmisión de capa por capa (porque la relevancia se pondera con los pesos z_{ij}); pero eso no asegura que la relevancia al propagarse desde la última hasta la primera, mantenga una ponderación coherente (como la que se tiene entre capas contiguas). Eso implica que la relevancia que se calcula en las capas cercanas a la entrada, puedan distribuirse de manera arbitraria.

5.2. GI

Si, de nuevo, la entrada de interés es $x \in \mathbb{R}^p$ y $\widetilde{x} \in \mathbb{R}^p$ es otro vector llamado base. De nuevo, $f : \mathbb{R}^p \to [0, 1]$. Una atribución o relevancias de la predicción $f(x)$, relativa a la base $\widetilde{x}$ es un vector $A_f(x, x') = (a_1, \ldots, a_p) \in \mathrm{R}^p$. Donde a_i será la contribución del predictor x_i a la salida $f(x)$ y está definida como:

$$a_i(x) := (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial f\left(\widetilde{x} + \alpha \times (x - \widetilde{x})\right)}{\partial x_i} d\alpha$$

La idea detrás de la fórmula anterior es sumar los cambios de la caja negra f respecto a la dimensión i en una trayectoria de un *punto neutro* $\widetilde{x}$, hasta el punto de interés x para asignar la responsabilidad del valor $f(x)$ al predictor x_i.

La integral puede ser aproximada por una suma de la forma:

$$a_i^{\text{approx}}(x) ::= (x_i - x_i') \times \Sigma_{k=1}^{m} \frac{\partial F\left(x' + \frac{k}{m} \times (x - x')\right)}{\partial x_i} \times \frac{1}{m}$$

El número m de pasos es el parámetro a fijar, mientras éste incrementa la aproximación mejora pero el costo computacional crece.

El otro parámetro que se debe decidir es el punto base o neutro $\widetilde{x}$. Recomiendan que también sea una raíz de f, para que las atribuciones se puedan interpretar como una función sólo de x. Sin embargo, depende del dominio de aplicación. En la figura 8, se muestran los resultados de aplicar GI a una red de clasificación de preguntas. En el artículo donde se propuso GI (en [STY17]), mencionan que obtuvieron buenos resultados utilizando como base, no una raíz, sino al vector (*embedding*) de ceros.

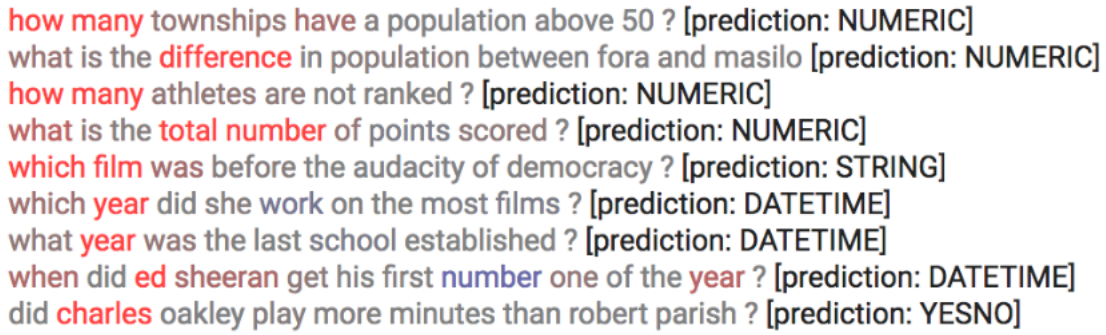


Figura 8: GI en clasificación de preguntas. El color indica la relevancia positiva de la palabra (rojo) o negativa (azul). La clase predecida por el modelo se muestra entre corchetes. Tomado de [STY17].

5.3. LIME

There once was a package called lime,
Whose models were simply sublime,
It gave explanations for their variations,
one observation at a time.
— Mara Averick

Éste busca ajustar un modelo $g : \mathbb{R}^p \to \mathbb{R}$ que sea sencillo e interpretable, para aproximar al modelo complejo $f : \mathbb{R}^p \to \mathbb{R}$, en una región local alrededor de la entrada de interés x. En general el modelo g se elige lineal y ralo. La regresión se hace con perturbaciones alrededor de la instancia x que se desea explicar. Se puede dividir en los siguientes pasos:

1. Para asegurar que g sea interpretable, se aplica una transformación al dominio $\mathbb{R}^p$ del modelo complejo f. Por ejemplo, si los predictores del modelo original son pixeles de una imagen, se puede dividir a la imagen en cuadrados. Una representación simplificada $z' \in \mathbb{R}^{d'}$ de un $z \in \mathbb{R}^p$, puede ser un vector con variables indicadoras como componentes: es decir, variables que identifiquen si z_i, la componente i de z, descansa en el mismo cuadrado que x_i.

2. (opcional) Adicionalmente, como $g \in G$ sólo es un elemento de una clase de funciones que es potencialmente interpretable, para asegurar la interpretabilidad de g, se puede agregar un procedimiento de selección de predictores (*feature selection*) y castigar la complejidad de $g : \mathbb{R}^{d'}$ con alguna medida Ω.

3. Para incorporar la fidelidad, es decir, el que el modelo g capture el comportamiento de f, el modelo g se ajusta con una muestra Z conformada por perturbaciones z' alrededor de x'. Las etiquetas de los elementos $z' \in Z$ son $f(z)$, donde z es una representación de z' en el dominio de f.

Tras los pasos anteriores obtenemos atribuciones de la forma:

$$\xi(x) = \text{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Donde $\mathcal{L}$ es una función de costo/fidelidad y π_x es una medida de proximidad para definir la región local alrededor de x. $\xi(x)$ es un modelo interpretable. Las atribuciones en este caso son los coeficientes del modelo lineal.

En la figura 9, se muestra un esquema del procedimiento.

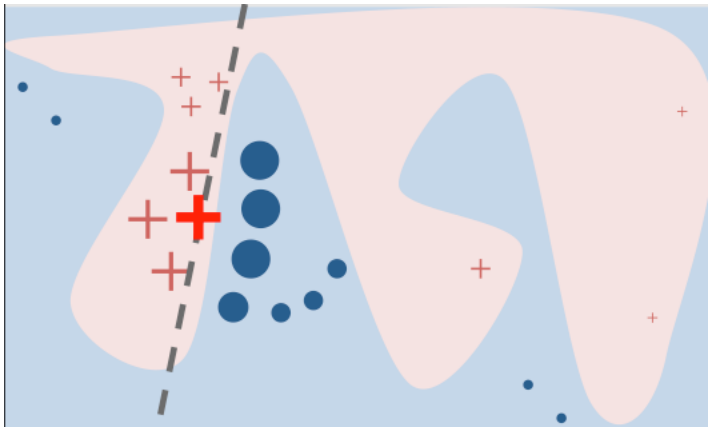


Figura 9: Esquema de LIME. La frontera de decisión (compleja) de la RNP se simboliza con los colores rosa/azul. La cruz amplia roja es la muestra $\widetilde{x}$ explicada. Las demás cruces simbolizan las perturbaciones z' con las que se entrenará el modelo lineal. Este último se representa por la línea punteada. Tomado de [RSG16]

Observemos que al igual que en los demás métodos, las atribuciones construidas se ajustan en una vecindad y no en todo el dominio. Eso es una necesidad de las explicaciones porque sino serían tan complejas como la caja negra y dejarían de ser interpretables:

... En aquel Imperio, el Arte de la Cartografía logró tal Perfección que el mapa de una sola Provincia ocupaba toda una Ciudad, y el mapa del Imperio, toda una Provincia. Con el tiempo, esos Mapas Desmesurados no satisficieron y . . . levantaron un Mapa del Imperio, que tenía el tamaño del Imperio y coincidía puntualmente con él. . . las Generaciones Siguientes entendieron que ese dilatado Mapa era Inútil y lo entregaron a las Inclemencias del Sol y de los Inviernos . . .

— Jorge Luis Borges

Un problema con este método es que se debe de elegir una noción de localidad para establecer a cuáles observaciones se les dará más peso al ajustar el modelo interpretable. En la figura 10, hay un ejemplo de donde LIME propone dos modelos ξ muy diferentes, al variar el tamaño de la vecindad.

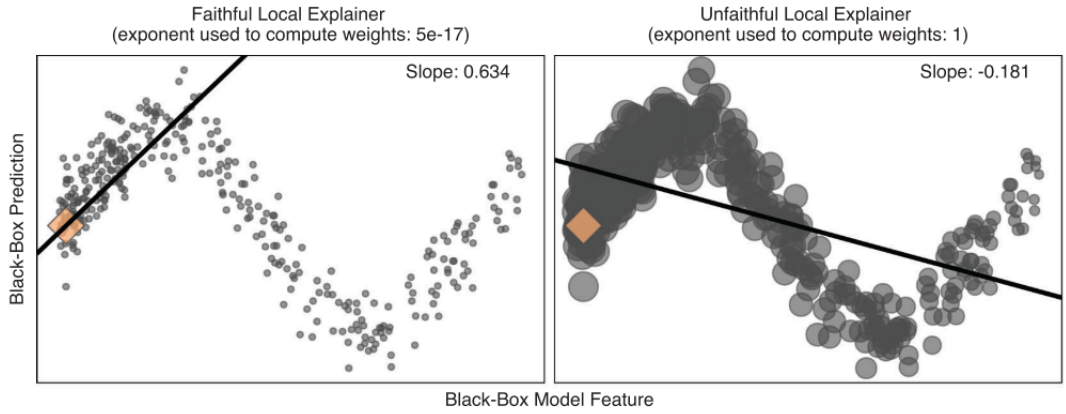


Figura 10: Dos modelos lineales (línea sólida) construidos con LIME sobre el mismo conjunto de datos, pero variando lo que se considera como vecindad del punto de interés (en naranja). Los puntos negros son las observaciones, su tamaño depende de la importancia en el ajuste de los modelos lineales. Tomada de [GH21].

El modelo explicativo (en la figura 10 la línea recta) es una aproximación del modelo complejo y no una interpretación directa. Luego, las atribuciones calculadas están sujetas a la calidad de la aproximación. Ese es el concepto central en el área de XAI, se llama **fidelidad**. Podemos evaluar visualmente que la de la izquierda es fiel al modelo localmente, pero la de la derecha no. Por supuesto, en dimensiones más altas este tipo de evaluación no es directa. En [GH21], Goode, construye otras herramientas visuales para evaluar LIME y concluyen que es muy sensible al parámetro de vecindad.

5.4. Oclusión

Este método se basa en perturbar las entradas para asignar las atribuciones. Se utiliza en imágenes. Se realiza una partición en regiones de la imagen x y sistemáticamente se cubren diferentes porciones de la imagen con color gris (el valor medio de la escala), generando una imagen similar $\widetilde{x}$. Después, se monitorean las salidas del clasificador. Las porciones que al ser cubiertas, presenten una mayor diferencia en $f(\widetilde{x})$ respecto a $f(x)$, se les asignará una atribución mayor. A cada píxel/predictor x_i se le asigna como atribución la misma que las de la región la que pertenece.

En la figura 11, mostramos un forma en la que evaluaron visualmente el método en el artículo en que se introdujo ([ZF14]).

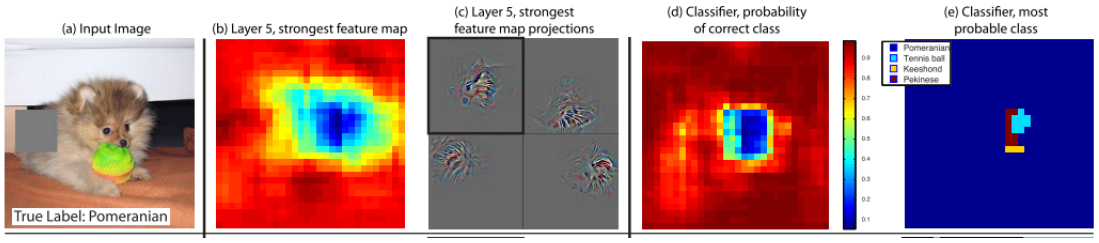


Figura 11: Disminución de $f(x)$ obstruyendo regiones importantes. De izquierda a derecha: (a) una imagen de test; (b) los mapeos la última capa de una CNN al evaluar la imagen obstruyendo una región importante; (c) proyección de los mapeos en la imagen de entrada (marco negro) y en otras imágenes; (d) Tomado de [ZF14].

Este método sólo se ocupa en imágenes. No hay una forma directa de adaptarlo para otro tipo de predictores además de pixeles, en donde no haya un orden espacial.

5.5. Maximización de activación

Éste, a diferencia de los anteriores es un método global. Es decir, produce una atribución por RNP, y no una por cada entrada x.

La idea inicial fue, dado un modelo entrenado f, buscar entre un conjunto de imágenes, aquella imagen x^* que maximizara el valor de f. Sin embargo, la idea puede generalizarse de dos maneras: (1) al no restringirse a un conjunto de imágenes y dejar que x^* pueda tomar todos los valores permitidos para una imagen; y (2) no limitarse a buscar el valor que maximiza f, sino maximizar cualquier función de salida de la red, asociada a cualquier neurona. (1) implica que no tengamos imágenes realista, pero que pueden señalar a aspectos importantes para la red y (2) permite dar ideas de aspectos importantes para cada unidad básica del modelo.

La manera de hacerlo es traduciendo el problema a uno de optimización. Defimos θ como los parámetros de la red (pesos e interceptos), $g_{ij}(\theta, x)$ será la función de activación de una neurona i de una capa j. Si suponemos que la red ya fue entrenada y que θ es fijo, buscamos resolver:

$$x^* = \arg \max_{x \text{ tal que } \|x\|=\rho} g_{ij}(\theta, x)$$

como suele ser un problema no convexo, se resuelve con métodos iterativos como ascenso de gradiente, asegurándonos tener sólo un máximo local. Por lo mismo, se pueden obtener distintas soluciones dependiendo de la inicialización y de los hiper-parámetros del ascenso en gradiente. Se puede realizar en cualquier red donde se puedan calcular los gradientes de la función objetivo g_{ij}, respecto de la imagen de entrada x.

Cuando se obtiene más de un máximo con distintas inicializaciones, se pueden promediar los resultados, elegir al que los maximiza o mostrarlos todos. En la figura se muestran los resultado del trabajo en donde se propuso ([Erh+09]). En ellos las imágenes que lo maximizaban fueron únicas.

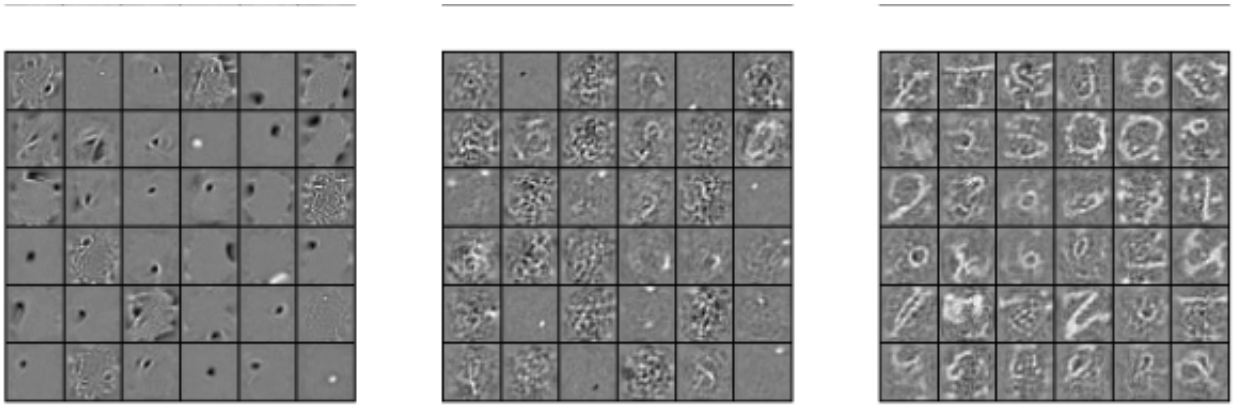


Figura 12: Activation maximization aplicado a una red SDAE en MNIST. Cada mosaico corresponde a una de las 3 capas escondidas de la red. Cada capa tiene 36 neuronas. Tomado de [Erh+09]

Como mencionamos en la introducción, con éste como en los demás métodos -en el caso en el que sean fieles- sabemos en qué predictores se están concentrando los modelos, pero no sabemos cómo se están usando.

6. SpRAy

Finalmente, para mostrar un poco de los métodos de XAI en conjuntos que no son imágenes, aplicaremos algunos de ellos en una red para detección de diabetes (*Pima Indians diabetes dataset*) y en otra red que predice el precio de casas (*Boston Housing Dataset*).

6.1. LIME en *Pima Indians diabetes dataset*

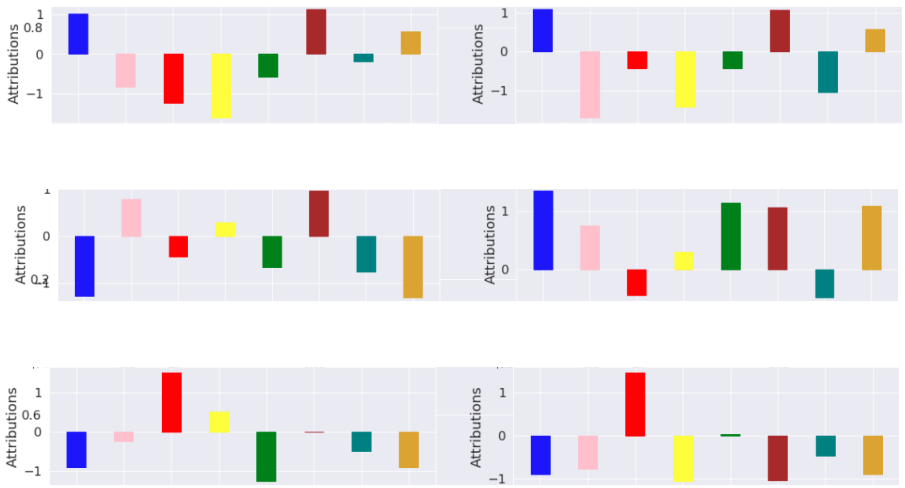
La primera red recibe varias medidas de diagnóstico de una mujer y predice si padece diabetes o no. Las medidas son: niveles de glucosa, de presión arterial e insulina, además del grosor de su piel, el IMC, edad, cantidad de embarazos y una función llamada "Pedigree de diabetes" que es una medida de la influencia de diabetes hereditaria en la persona, a partir de la presencia de familiares con diabetes y su cercanía a ellos.

Este ejemplo es de juguete pues se pueden lograr niveles de exactitud similares con un modelo que sí es interpretable. Pero se propone para mostrar otras formas de ganar confianza métodos de XAI: (a) comparando las atribuciones con modelos interpretables en conjuntos donde se puedan aplicar los dos. Y (b) viendo si las atribuciones de la clase positiva se distinguen claramente de las negativas.

La red no es profunda; consiste en dos capas escondidas de 12 y 8 neuronas, respectivamente. Como función de activación se utilizó ReLU. En la capa de salida sólo hay una neurona y utiliza como función de activación a la sigmoide.

La red obtuvo 79 % de exactitud en el conjunto de prueba. Para identificar **tendencias en su manera de predecir**, aplicamos LIME a algunos datos x y a la red. Para aplicar LIME se debe de elegir una familia de modelos que se optimizará en una vecindad de x, la familia utilizada para este ejemplo fue de funciones lineales. Asimismo, se debe de elegir un método para construir las muestras y una distancia para determinar la noción de vecindad alrededor de x. Para generar las muestras se hizo una partición del dominio en hipercubos equiespaciados y la muestra se tomó uniformemente. La distancia utilizada fue la de Gower. Para la implementación se ocupó la librería LIME en Python.

Las barras de las gráficas que siguen (figura 13) se formaron con los valores de las entradas de los vectores de atribuciones, $a_x = (a_{embarazos}, a_{glucosa}, a_{presión}, \ldots)$. Observemos que para poder comparar tendencias, las atribuciones están normalizadas.



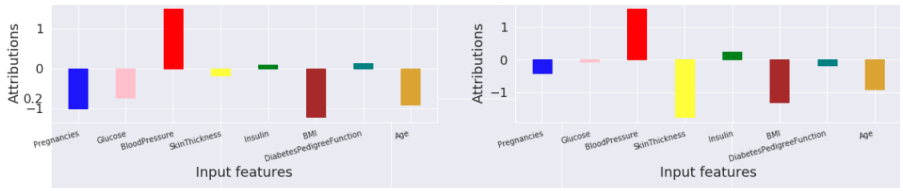


Figura 13: Visualización de vectores de atribuciones. Cada gráfica de barras corresponde a un vector de atribuciones a_x. Las dos filas de arriba pertenece a observaciones x clasificadas con diabetes y las filas de abajo clasificadas como negativas.

En la figura 13, algunas tendencias que se pueden identificar son que:

- Los clasificados como positivos tienen atribuciones a IMC (color café) con valores positivos mientras que en la clase de no-diabéticos tienen valores negativos.
- La norma del de glucosa (en rosa) tiende a tener norma alta en los diabéticos -aunque hay valores positivos y negativos- a comparación de la de los no-diabéticos.
- En los casos no-diabéticos, las atribuciones a presión (rojo) son más altas que en los positivos.
- En embarazos (azul) y en edad (amarillo), todos los no-diabéticos tienen valores negativos y bajos.
- La función de pedigree (esmeralda) las normas de los diabéticos suelen ser más altas.

mientras que no es clara una tendencia en el predictor de grosor de piel ni en el de insulina, donde hay ejemplos variados en las dos clases.

Por otro lado, a decir verdad, los vectores de atribuciones mostrados en la figura 13 fueron seleccionados con el propósito de mostrar tendencias. Sin embargo, en una muestra más grande, señalar tendencias es una tarea más difícil: en la figura 14 se observa una muestra (elegida aleatoriamente) de los diabéticos. Para lidiar con la multitud de atribuciones de los métodos locales de XAI, se introduce una técnica en la siguiente sección.

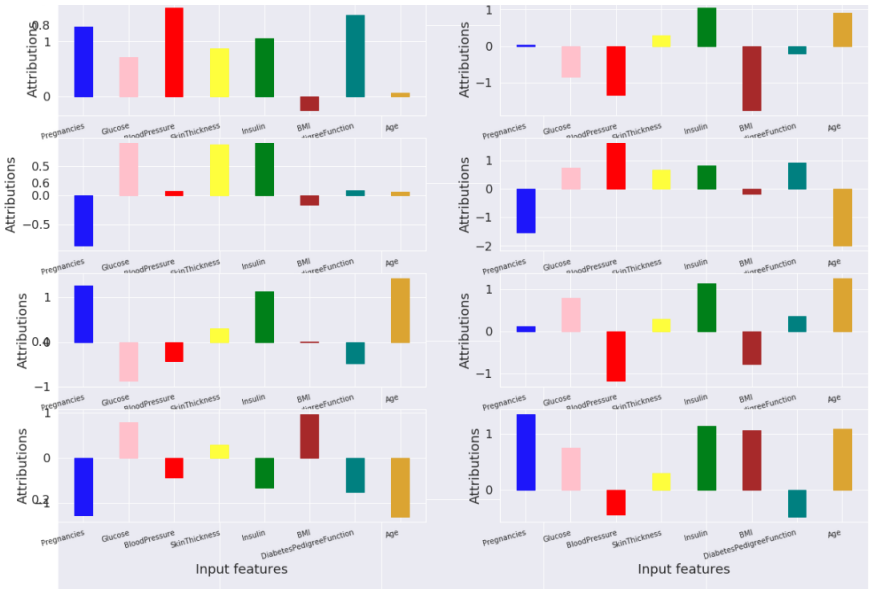


Figura 14: Visualización de vectores de atribuciones. Cada gráfica de barras corresponde a un vector de atribuciones a_x. Todas corresponden a observaciones x clasificadas con diabetes.

6.2. SpRAy en *Boston Housing Dataset*

Para lidiar con la confusión y sistematizar la búsqueda de tendencias, se propuso un método llamado Análisis de Relevancia Espectral (SpRAy, por sus siglas en inglés, véase [Lap+19]). Éste ofrece una alternativa para investigar eficientemente el comportamiento de una caja negra en conjuntos de datos amplios, en lugar de revisar varias atribuciones individuales. Consiste en pasos sencillos:

i) Primero se utiliza un método explicativo (como LRP, GI o LIME) para generar un vector $a_x \in \mathbb{R}^p$ de atribuciones por cada instancia $x \in \mathbb{R}^p$, del conjunto de datos de interés.

ii) Se aplica un algoritmo de agrupamiento en el conjunto de explicaciones $\{a_x : x \in \mathbb{R}^p\}$. Con la intención de observar si cada grupo identificado corresponde a una estrategia de decisión.

iii) Finalmente, podemos proyectar en dimensiones bajas ($\mathbb{R}^2$ o $\mathbb{R}^3$) para visualizar los grupos y ejemplos de atribuciones en ellos.

Éste se aplicó en el conjunto de datos *Boston Housing Dataset*, que, como se mencionó anteriormente se utiliza para predecir el precio de casas. El conjunto está conformado por indicadores de la localidad de una casa y predice el precio de la casa. Entre los indicadores están: la tasa de crímenes per capita en la localidad, proporción de edificios residenciales en la zona, promedio de cuartos por vivienda, accesibilidad a vías rápidas, tasa de maestros/alumnos en el área, proporción de personas afroamericanas en la zona, entre otros.

Se utilizó una RN con 2 capas ocultas de 8 neuronas por capa, dando un total de 193 parámetros entrenables, como función de activación en las capas ocultas de nuevo se utilizó ReLU. En la figura 15 observamos imágenes generadas con SpRAy aplicados a la red. En este caso se utilizó GI para generar los vectores de atribuciones. El número m de pasos nuestro caso se utilizó $m = 100$. Para la implementación se ocupó la librería DeepExplain en Python, en ella están otros métodos de XAI como LRP y Oclusión. Mientras que para aplicar SpRAy se eligió Agrupamiento Espectral (véase [Von07]) para construir grupos relativamente homogéneos entre los vectores de atribuciones. En este se debe de elegir de antemano un número de grupos a formar; para este ejemplo, basandónos en proyecciones formadas

con t-SNE, se eligieron 4 grupos. Asimismo, para la visualización de los vectores de atribuciones de la figura 15, se utilizó t-SNE; este último método permite reducir dimensiones y también necesita fijar un parámetro de perplejidad, en nuestro caso se eligió una perplejidad de 20.

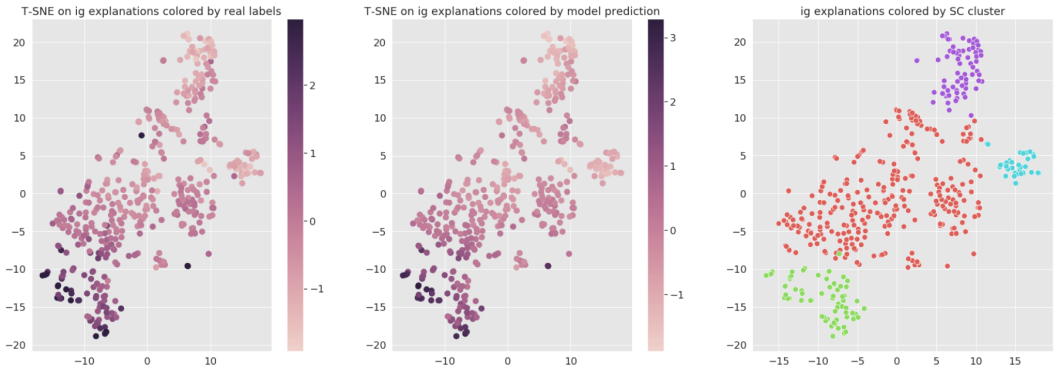


Figura 15: Las tres gráficas son los vectores de atribuciones $\{a_x\}_x$, generados con GI y proyectados con t-SNE en dos dimensiones. Sólo cambia el criterio para colorearlas. En las primeras dos, la intensidad del color es proporcional al precio de la casa correspondiente a x. En la del extremo izquierdo el precio es el real y en la del centro el precio es asignado por la RNP. Cada color de la gráfica del extremo derecho, corresponde a un grupo generado con Agrupamiento Espectral.

En la figura 15 podemos notar que el grupo rojo corresponde a viviendas con precio medio, los morados y azules a viviendas con precios bajos y el verde a viviendas con precios más altos.

En la figura 16 retomamos la gráfica del extremo derecho de la figura anterior (figura 15) pero desplegamos algunos representantes de cada grupo, encima del grupo correspondiente.

Figura 16: Vectores de atribuciones a_x, generados con GI y proyectados en dos dimensiones. Cada color corresponde a un grupo generado con Agrupamiento Espectral. Sobre cada grupo se dibujaron algunos vectores de atribuciones pertenecientes al grupo como gráficas de barras.

Viendo los ejemplos de la figura 16, notamos que una diferencia importante entre el grupo con precios más altos y el grupo con precios más bajos es la tasa de crímenes per capita (color rojo); en el primer grupo se observa siempre como valores negativos y en el segundo con valores positivos. Otra diferencia importante es el promedio de cuartos por vivienda (color verde): cuando el precio es alto son valores positivos y grandes, cuando el precio es bajo son valores negativos y grandes.

En efecto, con estos agrupamientos es más sencillo identificar tendencias en las predicciones. Existen varias diferencias de interés que se pueden estudiar, por mencionar una más, el que se hayan construido dos grupos en viviendas con precios bajos -si LIME es fiel a la RNP- señala que hay dos criterios distintos por los cuáles la RNP está prediciendo valores bajos. Observando las muestras con más detenimiento (figura 17), resulta que la accesibilidad a vías rápidas (barra azul) y la tasa de estudiantes por maestro (barra morada) son factores de decisión más importantes para el grupo morado, mientras que la tasa de crímenes per capita (barra roja) y la proporción de afroamericanos en la zona (barra rosa), lo son para el grupo azul.

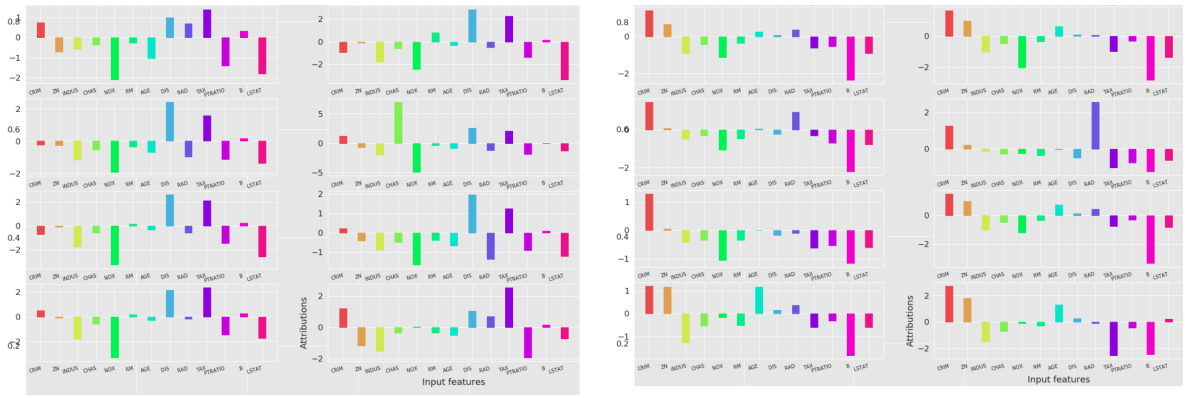


Figura 17: Muestra de atribuciones del grupo morado y azul (izquierda y derecha, respectivamente). Ambos grupos representan precios relativamente bajos. El grupo azul tiende a tener más bajos.

7. Discusión

Todos los métodos explicativos comparten algunos problemas. Uno es que aún, su fidelidad con el modelo en el que se está aplicando no se puede cuantificar directamente porque no existen atribuciones ideales contra los que se puedan comparar sus resultados. Y sin evaluar los algoritmos de XAI estamos descansando en otra caja negra para entender a una caja negra. Sin embargo, no hay certidumbre de que el modelo de XAI trabaje de la misma manera en distintos modelos. Para compensar esa falta se certidumbre se ocupan los métodos axiomáticos (como con el que se propuso en GI). También están desarrollándose teorías en ese mismo para evaluarlos como en [AHG21]. Otras alternativas que se han propuesto involucran pruebas visuales para diagnosticarlos (véase ejemplos en LIME en [GH21]).

Otra limitación es que en todos los métodos hay parámetros que se deben de elegir y variarlo deriva en explicaciones distintas. Como el problema que se observó con las vecindades en LIME.

Otro problema es que es difícil distinguir mal desempeño de la RNP donde se está aplicando, del mal desempeño del algoritmo de XAI. Se ha experimentado haciendo mínimos cambios a los predictores en distintos algoritmos de explicabilidad, obteniendo vectores de atribuciones distintos (conocidos como ejemplos adversarios). Esto se puede deber a que la misma RNP sea susceptible a esas pruebas (un ejemplo en la figura 18) o a que únicamente el algoritmo de XAI lo sea.

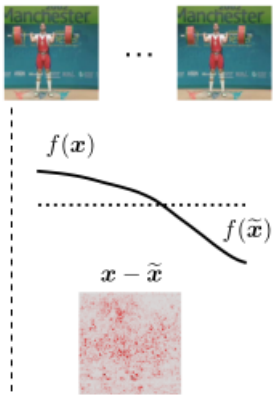


Figura 18: Ejemplo adversario: cambios mínimos en la entrada salidas distintos del modelo. Tomada de [Sam+21].

Así, aunque los algoritmos de XAI se construyen para dar claridad, en realidad agregan otro grado de opacidad pues necesitan ser evaluados. Como ejemplo, Rudin en [Rud19], propone el siguiente ejemplo. Supongamos que se tiene una caja negra que modela la probabilidad de que una persona cometa un crimen después de ser liberada de encarcelamiento -aplicación extendida actualmente en Estados Unidos-. Muchos modelos se basan en esas e historia criminal pero no explícitamente en variables como raza. Sin embargo, como los primeros dos predictores están correlacionados con la raza, un algoritmo de explicación nos podría indicar que la predicción e una persona se está basando es su raza, y simular de una manera muy similar a la de la caja negra. Pero eso no implica que la caja negra esté esté basándose en el criterio de raza. Luego, para asegurar justicia en aplicaciones de este tipo, antes de lidiar con las interpretaciones y la explicabilidad hay que lidiar con la transparencia.

Siguiendo la línea sobre decisiones sensibles para la vida de personas, como se sostiene también en [Rud19], utilizarlas en esos casos no es adecuado por otras razones. Por ejemplo, en situaciones donde hay información fuera de la base de datos que se toma en cuenta para decidir la evaluación (como para libertad condicional). Si el modelo es de caja negra, la manera de calibrar los resultados del modelo con las variables que no toma en cuenta no es claro. Es decir, aunque la rama de XAI ha recibido un impulso fuerte debido a legislaciones para proteger a personas de decisiones arbitrarias, el estado actual de la disciplina no asegura que dichas explicaciones logren decisiones más justas. Por lo que por el momento no es recomendable el aplicar XAI a decisiones que involucren decisiones sensibles en la vida de las personas.

Su uso aún así puede (y se ha logrado) orientar al proceso de descubrimiento de conocimiento: señalar posibles relaciones o defectos de los datos. También la exactitud lograda por RNP se puede usar como un indicador de posibles exactitudes que se puedan lograr ajustando suficientemente modelos que sí sean interpretables en decisiones de alto riesgo.

Bibliografía

[AHG21] Darius Afchar, Romain Hennequin y Vincent Guigue. "Towards Rigorous Interpretations: a Formalisation of Feature Attribution". En: (2021). arXiv: 2104.12437. URL: http://arxiv.org/abs/2104.12437.

[Bac+15] Sebastian Bach y col. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". En: *PLoS ONE* 10.7 (2015), págs. 1-46. ISSN: 19326203. DOI: 10.1371/journal.pone.0130140.

[Erh+09] Dumitru Erhan y col. "Visualizing higher-layer features of a deep network". En: *Bernoulli* 1341 (2009), págs. 1-13. URL: http://igva2012.wikispaces.asu.edu/file/view/Erhan+2009+Visualizing+higher+layer+features+of+a+deep+network.pdf.

[GH21] Katherine Goode y Heike Hofmann. "Visual diagnostics of an explainer model: Tools for the assessment of LIME explanations". En: *Statistical Analysis and Data Mining* 14.2 (2021), págs. 185-200. ISSN: 19321872. DOI: 10.1002/sam.11500.

[Lap+19] Sebastian Lapuschkin y col. "Unmasking Clever Hans predictors and assessing what machines really learn". En: *Nature Communications* 10.1 (2019), págs. 1-8. ISSN: 20411723. DOI: 10.1038/s41467-019-08987-4. arXiv: 1902.10178. URL: http://dx.doi.org/10.1038/s41467-019-08987-4.

[RSG16] Marco Tulio Ribeiro, Sameer Singh y Carlos Guestrin. ""Why should i trust you?.Explaining the predictions of any classifier". En: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-Augu (2016), págs. 1135-1144. DOI: 10.1145/2939672.2939778. arXiv: 1602.04938.

[Rud19] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". En: *Nature Machine Intelligence* 1.5 (2019), págs. 206-215. ISSN: 25225839. DOI: 10.1038/s42256-019-0048-x. arXiv: 1811.10154.

[Sam+21] Wojciech Samek y col. "Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications". En: *Proceedings of the IEEE* 109.3 (2021), págs. 247-278. ISSN: 15582256. DOI: 10.1109/JPROC.2021.3060483. arXiv: 2003.07631.

[STY17] Mukund Sundararajan, Ankur Taly y Qiqi Yan. "Axiomatic attribution for deep networks". En: *34th International Conference on Machine Learning, ICML 2017* 7 (2017), págs. 5109-5118. arXiv: 1703.01365.

[Von07] Ulrike Von Luxburg. "A tutorial on spectral clustering". En: *Statistics and Computing* 17.4 (2007), págs. 395-416. ISSN: 09603174. DOI: 10.1007/s11222-007-9033-z. arXiv: 0711.0189.

[ZF14] Matthew D. Zeiler y Rob Fergus. "Visualizing and understanding convolutional networks". En: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8689 LNCS.PART 1 (2014), págs. 818-833. ISSN: 16113349. DOI: 10.1007/978-3-319-10590-1_53. arXiv: 1311.2901.