

Resumen de Tesis

Medidas de Invarianza y Equivarianza a Transformaciones en Redes Neuronales Convolucionales

Aplicaciones al Reconocimiento de Formas de Mano

Facundo Manuel Quiroga

Laboratorio de Investigación en Informática III-LIDI,
Facultad de Informática, Universidad Nacional de La Plata

Directora Dra. Laura Cristina Lanzarini
Carrera Doctorado en Ciencias Informáticas
Facultad de Informática, Universidad Nacional de La Plata
Fecha Defendida el 13 de marzo de 2020
Contacto {fquiroga}@lidi.info.unlp.edu.ar
Documento completo <http://sedici.unlp.edu.ar/handle/10915/90903>

1. Introducción

Las Redes Neuronales son los modelos de aprendizaje automático con mejor desempeño en la actualidad en una gran variedad de problemas. Son modelos generales y aproximadores universales. Con algoritmos de optimización basados en descenso de gradiente, pueden optimizar miles o millones de parámetros en base a una función de error. Se distinguen de otros modelos en que no requieren un diseño manual de características de los datos para funcionar. Las mismas se aprenden automáticamente mediante el proceso de optimización, también llamado entrenamiento. Su diseño se organiza en capas que determinan su arquitectura. En los últimos años, se ha conseguido entrenar Redes Neuronales con múltiples capas mediante un conjunto de técnicas que suelen denominarse Aprendizaje Profundo (Deep Learning)[GBC16].

En particular, las Redes Convolucionales, es decir, Redes Neuronales que utilizan capas convolucionales, son el estado del arte en la mayoría de los problemas de visión por computadora, incluyendo la clasificación de imágenes. Las capas convolucionales permiten aplicar convoluciones con filtros aprendidos para un mejor desempeño y eficiencia[GBC16].

Muchos de los problemas para los cuales las Redes Convolucionales son el estado del arte requieren que los modelos se comporten de cierta manera ante transformaciones de su entrada. Existen tres propiedades fundamentales que capturan dicho requerimiento: la invarianza, la auto-equivarianza y la equivarianza.

Una red f es invariante a un conjunto de transformaciones si su salida $f(x)$ no cambia cuando una de esas transformaciones se aplica a su entrada x . La auto-equivarianza, un concepto relacionado, nos permite cuantificar que tan equivalente es transformar la entrada x o la salida $f(x)$. Finalmente, la equivarianza generaliza ambos conceptos, y nos permite establecer relaciones entre una transformación de la entrada x y su salida $f(x)$. La medición de estas tres propiedades nos permite comprender mejor la forma en que se comporta una red [GBC16; LV15].

Dotar con invarianza o equivarianza a los modelos tiene utilidades en varios dominios, como la clasificación de imágenes de galaxias, imágenes de microscopios o formas de mano. En particular,

el reconocimiento de formas de mano en imágenes es una de las etapas más importantes de los sistemas de reconocimiento de lenguas de señas o gestos mediante imágenes o video. En muchos casos, la rotación, traslación o escalado de la mano en la imagen no afectan a su forma, y por ende se requiere dotar de invarianza a la red para mejorar el desempeño del sistema [SG18].

La comprensión de la invarianza o equivarianza de una red o cualquier sistema puede ayudar a mejorar su desempeño y robustez. Dado el reciente éxito de las Redes Neuronales, y que son modelos tradicionalmente entendidos como de caja negra, resulta necesario avanzar en una mayor comprensión de su funcionamiento. Este requerimiento tiene especial importancia en el contexto del surgimiento de directivas gubernamentales como la de la Unión Europea que exigen que los modelos de inteligencia artificial puedan explicar sus decisiones [GBC16].

Existen dos opciones principales para otorgar invarianza o equivarianza a un modelo de red neuronal. La tradicional ha sido modificar el modelo para dotarlo de esas propiedades [CW16; Jad+15]. La otra opción es entrenarlo con aumentación de datos utilizando como transformaciones el mismo conjunto al que se desea la invarianza o equivarianza [SG18].

No está claro cómo se adquieren dichas propiedades con aumentación de datos, así como cuál es la mejor estrategia para aumentar los datos con este fin [SG18]. Tampoco está claro como las modificaciones de los modelos afectan la eficiencia y el poder de representación de los mismos.

Con este fin, pueden definirse métricas de invarianza y equivarianza para analizar a las redes neuronales desde una perspectiva transformacional [Goo+09; LV15]. Dichas métricas permiten, por ejemplo, entender como se comporta un sistema de reconocimiento de rostros ante transformaciones afines de una imagen del rostro. Para los sistemas que analizan el riesgo crediticio, dada una transformación del género del solicitante, permiten evaluar si dicho atributo influye en la decisión final. Para los sistemas que analizan texto permite evaluar si el significado predicho varía en base a una transformación que, por ejemplo, reemplace palabras por sus sinónimos.

La invarianza y equivarianza pueden estimarse midiendo los cambios en la red en base a las transformaciones realizadas a su entrada. Las metodologías actuales de evaluación y comprensión de la invarianza y equivarianza se enfocan solamente en los cambios detectados en la salida de la red. No obstante, para poder comprender como se codifican estas propiedades, el análisis debe realizarse en base a toda la red, es decir, considerando las representaciones intermedias. No obstante, las métricas de este tipo presentadas previamente han sido inadecuadas para evaluar estas propiedades [Goo+09; LV15].

En resumen, si bien hay varios métodos propuestos para otorgar equivarianzas a las redes, y en consecuencia aumentar su robustez, no existe ninguna comparación exhaustiva entre los distintos métodos, y lo que es aún más acrucante, existen pocos análisis en profundidad sobre como dichos métodos codifican las equivarianzas. Comprender el funcionamiento transformacional de las redes ayuda a mejorar los sistemas existentes. No obstante, para ello se requieren herramientas para analizar las redes, como las métricas de invarianza y equivarianza. De estas interrogantes surgen los objetivos de investigación de esta tesis.

2. Objetivos, contribuciones y publicaciones

2.1. Objetivos

Nuestro objetivo general en esta tesis es contribuir al entendimiento y mejora de la equivarianza de los modelos de redes neuronales, en particular aplicados a la clasificación de formas de mano para la lengua de seña y otros tipos de gestos mediante modelos de redes convolucionales.

Para ello, establecimos los siguientes objetivos particulares:

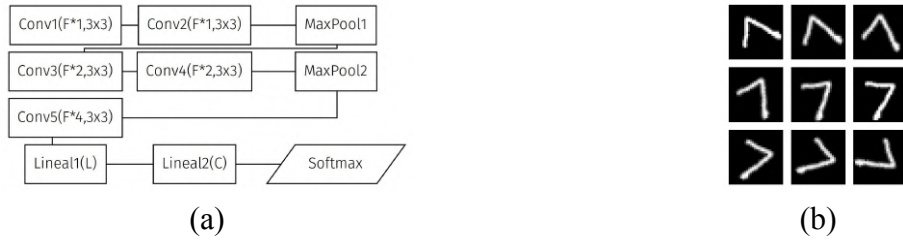


Figura 1: (a) Modelo SimpleConv (b) Ejemplos rotados de la base de datos MNIST.

1. Analizar los modelos específicos para equivarianza en CNNs. Comparar los modelos específicos y la aumentación de datos para obtener equivarianza. Evaluar estrategias de transferencia de aprendizaje para obtener modelos equivariantes a partir de modelos que no lo son.
2. Desarrollar métricas de equivarianza para las activaciones o representaciones internas de las redes neuronales. Implementar las métricas en una librería de código abierto. Analizar el comportamiento de las métricas. Comparar con las métricas existentes.
3. Caracterizar modelos de CNN para la clasificación de imágenes en términos de su equivarianza con las métricas propuestas. Comparar modelos de CNN, con y sin equivarianza, para la clasificación de formas de mano.

Debido a la existencia de múltiples métodos para lograr equivarianza, y a la falta de comparaciones rigurosas y profundas de los mismos, el alcance de esta tesis se limita al análisis y no propone nuevos modelos equivariantes de redes.

2.2. Publicaciones

Las siguientes publicaciones están relacionadas directamente con las contribuciones que presentamos en esta tesis: [Qui+17; Qui+18; Qui+19]. Además, esta línea de investigación fue motivada por trabajos relacionados realizados en la primera etapa del doctorado [Ron+15; Qui+16a; Qui+16c; Qui+16b].

Adicionalmente, durante el transcurso de esta tesis creamos **Métricas transformacionales**¹, una librería de código abierto para computar las métricas propuestas. También creamos una librería de código abierto llamada **Handshape Datasets**² para facilitar la descarga y el preprocesamiento de conjuntos de datos de formas de mano.

3. Resumen de contribuciones

En esta sección, presentamos de forma resumida las contribuciones principales de la tesis. En términos de los experimentos, si bien hemos utilizado diversos modelos, transformaciones y bases de datos, **dadas las limitaciones de página** sólo mostraremos resultados para el modelo SimpleConv, la base de datos MNIST y rotaciones de 0° a 360° como conjunto de transformaciones (Figura 1). Por el mismo motivo, tampoco presentaremos el trabajo realizado sobre la clasificación de formas de mano, el tercer aporte de la tesis.

¹https://github.com/facundoq/transformational_measures

²https://github.com/midusi/handshape_datasets

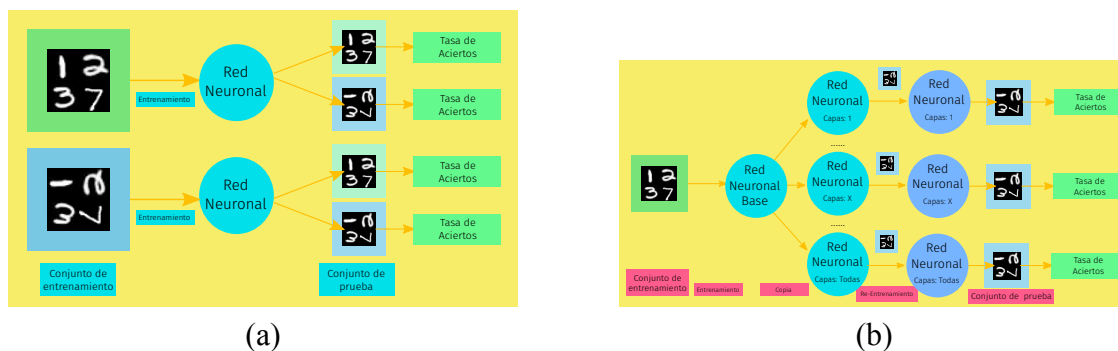


Figura 2: (a) Metodología del experimento de comparación de modelos. (b) Metodología del experimento de reentrenamiento de capas específicas.

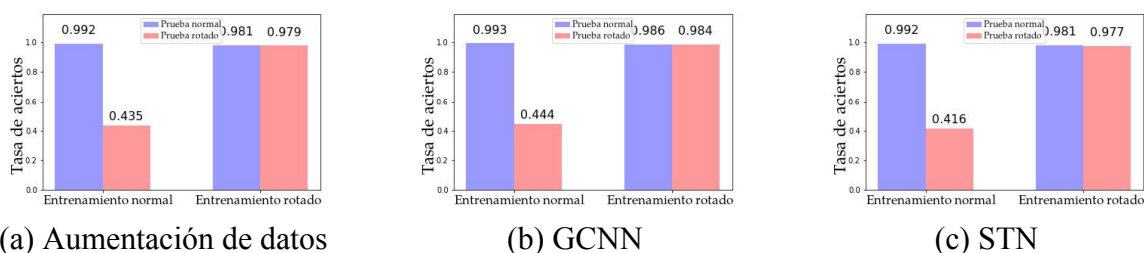


Figura 3: Resultados del experimento para el modelo base SimpleConv en la base de datos MNIST

3.1. Comparación de Modelos Invariantes con Aumentación de Datos

En el **primer aporte** de esta tesis, analizamos diversas estrategias para obtener invarianza en modelos de clasificación de imágenes con redes neuronales convolucionales.

3.1.1. Comparación de modelos invariantes

Comparamos los modelos tradicionales de CNN con aumentación de datos contra los modelos especializados Group CNN (GCNN) [CW16] y Spatial Transformer Networks (STN) [Jad+15] para determinar su desempeño, utilizando la base de datos MNIST.

Para ello, entrenamos un modelo con y sin aumentación de datos, y luego lo evaluamos también con y sin aumentación de datos, dando lugar a 4 combinaciones independientes (Figura 2 (a)).

Los resultados (Figura 3) indican que aún con ingeniosas modificaciones de las redes convolucionales, la aumentación de datos sigue siendo necesaria para obtener un desempeño similar al de los modelos no invariantes. Más aún, en varios casos la aumentación de datos por si sola puede proveer un desempeño similar al de los modelos especializados, siendo al mismo tiempo más simples de entrenar y comprender.

3.1.2. Análisis del re-entrenamiento de capas específicas para obtener invarianza

La utilización de redes pre-entrenadas es una técnica usual para mejorar el desempeño de los modelos. No obstante, al reentrenar un modelo para obtener invarianza no está claro a priori si toda la red puede o necesita ser re-entrenada, o si solo algunas capas requieren adaptarse.

Para determinarlo, entrenamos un modelo base con una BD sin rotar, y luego hacemos copias de la red. En cada copia re-entrenamos algún subconjunto de sus capas con el conjunto de entrenamiento y aumentación de datos. Luego, evaluamos el desempeño de la red con alguna de sus capas re-entrenadas (Figura 2 (b)).

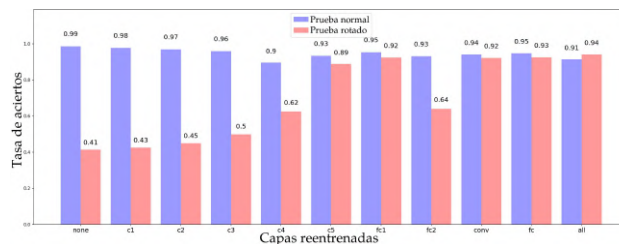


Figura 4: Tasa de acierto del modelo luego de re-entrenar una capa o conjunto de capas, para la versión rotada y la versión original de la BD. El eje x indica que capas re-entrenamos. Las etiqueta *conv* y *fc* denotan todas las capas convolucionales y lineales, respectivamente.

La Figura 4 muestra las tasas de aciertos obtenidas al re-entrenar algunos subconjuntos de capas. El re-entrenamiento para obtener invarianza muestra una dinámica similar a re-entrenamiento para hacer transferencia de aprendizaje: las capas finales, con características de alto nivel, son una mejor elección para re-entrenar de forma individual dado que están más cerca de la salida y pueden afectarla más. Esto indica que hay una redundancia de información en las capas anteriores, debido a que las versiones rotadas de las imágenes pueden codificarse con éxito a partir de la salida de dichas capas. Sorprendentemente, se puede también re-entrenar *solamente* las capas convolucionales para obtener invarianza a niveles originales. De esta forma, estos experimentos revelan que la codificación de la invarianza puede darse de distintas formas. Esto sugiere que es importante entender como se codifica la invarianza en las distintas capas, para lo cual desarrollamos las métricas de invarianza y equivarianza que describimos a continuación.

3.2. Métricas de Invarianza y Equivarianza

Como **segundo y principal aporte**, proponemos un nuevo conjunto de métricas para medir la invarianza y auto-equivarianza de las redes. Dichas métricas permiten cuantificar estas propiedades de forma empírica y no solo en la salida de la red sino también en sus activaciones o representaciones internas. De esta forma, podemos visualizar y cuantificar que tan invariante o auto-equivariante es una red, ya sea en su totalidad, por capas, o por activaciones individuales. Las métricas son aplicables a cualquier red neuronal, sin importar su diseño o arquitectura, así como a cualquier conjunto de transformaciones. Todas las métricas han sido implementadas y están disponibles públicamente en https://github.com/facundoq/transformational_measures. Para más detalles, consultar [Qui20].

3.2.1. Definiciones básicas

Dada una entrada x , un modelo de redes neuronales computa varios valores intermedios u ocultos, denotados como $a_1(x), \dots, a_k(x)$. Por simplicidad, llamaremos a estos valores $a_i(x)$ *activaciones* de la red f (Figura 5 (a)). Para medir la equivarianza de un modelo f , medimos la equivarianza de las activaciones individuales $a_1(x) \dots a_k(x)$. Dado que la métrica puede ser definida para una activación de forma independiente del resto, en lo siguiente nos referiremos a una activación simplemente como $a(x)$, sin su índice.

Dado un conjunto de n muestras $X = [x_1 \dots x_n]$ y un conjunto de m transformaciones $T = [t_1 \dots t_m]$ definidas sobre X , podemos calcular todas las posibles transformaciones de todas las muestras, y organizarlas en una matriz de muestras-transformaciones llamada $\mathbf{MT}(a, X, T)$ (Figura 5 (b)).

$$\mathbf{MT}(a, X, T)_{i,j} = a(t_j(x_i)) \quad [1]$$

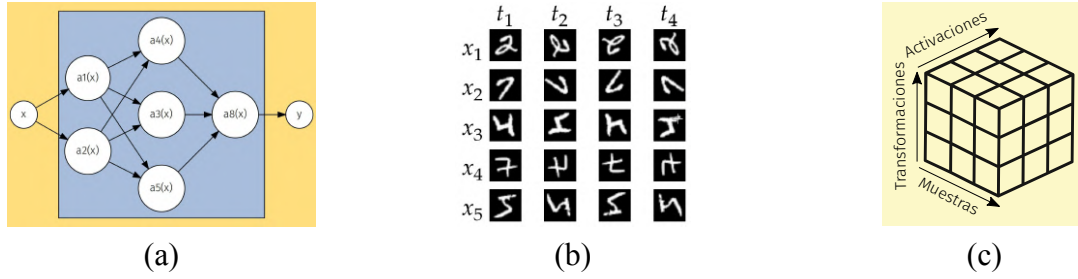


Figura 5: (a) Activaciones de un modelo. (b) Muestras y transformaciones. (c) Tensor de activaciones, muestras y transformaciones.

3.2.2. Métricas de Invarianza basadas en Varianza

Podemos cuantificar la invarianza de una activación a midiendo su varianza. Debido a que existen 2 fuentes de variabilidad, transformaciones y muestras, medimos la varianza transformacional y muestral.

Varianza Transformacional Si una activación es invariante a t , entonces $f(t(x)) = t(x)$. Por ende, para un conjunto de transformaciones invariante $T = [t_1, \dots, t_m]$, tenemos $f(t_1(x)) = f(t_2(x)) = \dots = f(t_m(x))$. Podemos calcular las desviaciones de la invarianza midiendo la varianza de los valores $f(t_1(x)), f(t_2(x)), \dots, f(t_m(x))$

La métrica de Varianza Transformacional (VT) se define entonces según la Ecuación [2], y su cómputo se ejemplifica con la Figura 6. Esta métrica mide la varianza en las activaciones con respecto a todas las transformaciones, para cada ejemplo, y luego promedia esas varianzas con respecto a todos los ejemplos. Si $VT(a) = 0$, entonces la activación a no varía con las transformaciones, y por ende es invariante.

$$VT(a) = Media \left(\begin{bmatrix} Var(\mathbf{MT}(a)[1, :]) \\ \dots \\ Var(\mathbf{MT}(a)[n, :]) \end{bmatrix} \right) \quad [2]$$

Varianza Muestral Los valores de la métrica VT pueden observarse en la Figura 7 (a). Allí podemos observar que las unidades de la métrica VT dependen de la activación y capa. Por ende, utilizaremos la Varianza Muestral (VM) para normalizar la VT . La métrica VM es la transpuesta conceptual de la VT y se define de forma análoga (Ecuación [3]).

$$VM(A) = Media \left([Var(\mathbf{MT}[:, 1]) \quad \dots \quad Var(\mathbf{MT}(a)[:, m])] \right) \quad [3]$$

Varianza Normalizada Combinando las Ecuaciones [2] y [3], podemos definir la métrica de Invarianza basada en Varianza Normalizada (VN), como la relación entre las otras dos métricas (Ecuación [4]).

$$VN(a) = \frac{VT(a)}{VM(a)} \quad [4]$$

La figura Figura 7 (c) muestra el resultado de la VN. Podemos ver que la escala difiere significativamente entre VT y VM , y por ende es necesaria la normalización que ofrece VN .

$$\begin{array}{c} \rightarrow (1) \text{ Var} \\ (2) \text{ Media} \leftarrow \end{array} \begin{bmatrix} a(t_1(x_1)) & a(t_2(x_1)) & a(t_3(x_1)) & a(t_4(x_1)) \\ a(t_1(x_2)) & a(t_2(x_2)) & a(t_3(x_2)) & a(t_4(x_2)) \\ a(t_1(x_3)) & a(t_2(x_3)) & a(t_3(x_3)) & a(t_4(x_3)) \\ a(t_1(x_4)) & a(t_2(x_4)) & a(t_3(x_4)) & a(t_4(x_4)) \\ a(t_1(x_5)) & a(t_2(x_5)) & a(t_3(x_5)) & a(t_4(x_5)) \end{bmatrix} \Rightarrow \text{Media} \left(\begin{array}{c} \text{Var}([a(t_1(x_1)) \ a(t_2(x_1)) \ a(t_3(x_1)) \ a(t_4(x_1))]) \\ \text{Var}([a(t_1(x_2)) \ a(t_2(x_2)) \ a(t_3(x_2)) \ a(t_4(x_2))]) \\ \text{Var}([a(t_1(x_3)) \ a(t_2(x_3)) \ a(t_3(x_3)) \ a(t_4(x_3))]) \\ \text{Var}([a(t_1(x_4)) \ a(t_2(x_4)) \ a(t_3(x_4)) \ a(t_4(x_4))]) \\ \text{Var}([a(t_1(x_5)) \ a(t_2(x_5)) \ a(t_3(x_5)) \ a(t_4(x_5))]) \end{array} \right)$$

Figura 6: Cálculo de la Varianza Transformacional (VT).

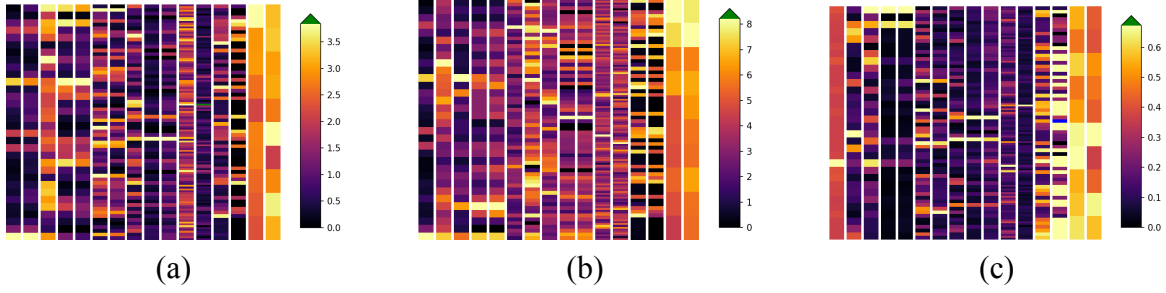


Figura 7: (a) VT, (b) VM y (c) VN de cada activación de un modelo SimpleConv, evaluado con rotaciones en MNIST.

Casos especiales En los casos donde tanto $VT(a) = 0$ como $VM(a) = 0$, definimos $VN(a) = 1$. Este caso corresponde a la definición de una activación *muerta*, que no responde a ningún patrón y por ende no tiene ningún uso en la red. En los casos donde $VT(a) > 0$ y $VM(a) = 0$, definimos $VN(a) = +\infty$. Este caso es similar al primero, pero ahora las transformaciones *si* causan variabilidad en la activación. Esto posiblemente se deba a que generan muestras muy por fuera de la distribución original de los datos.

Valores de la Varianza Normalizada Podemos analizar los valores posibles de VN y su interpretación. Si $VN(a) = 0$, entonces $VT(a) = 0$ y la activación es claramente invariante. Si $VN(a) < 1$, la varianza debido a las transformaciones es menor que la varianza debido a las muestras, y por eso podemos considerar aproximadamente invariante a la activación. Si $VN(a) > 1$ se aplica el mismo razonamiento pero con la conclusión opuesta. Si $VN(a) \simeq 1$, entonces ambas varianzas están en equilibrio.

3.2.3. Métricas de Auto-Equivarianza basada en Varianza

La auto-equivarianza es un caso especial de la equivarianza que nos permite analizar si al aplicar una transformación a las entradas, las activaciones se transforman de la misma manera: formalmente, $t(f(x)) = f(t(x))$ (Figura 8 (a)). Si asumimos que t tiene una inversa, entonces esto es equivalente a $f(x) = t^{-1}(f(t(x)))$ (Figura 8 (a)). Para un conjunto de transformaciones $T = [t_1, \dots, t_m]$, esto implica que $t_1^{-1}(f(t_1(x))) = \dots = t_m^{-1}(f(t_m(x)))$ (Figura 8 (c,d)).

Entonces, podemos definir las métricas de Auto-Equivarianza basadas en Varianza de manera similar a las métricas de Invarianza basadas en Varianza. Las métricas de Invarianza se basan en comparar los valores $a(t_1(x)), \dots, a(t_m(x))$. En el caso de la Auto-Equivarianza, podemos en lugar de eso comparar los valores $t_1^{-1}(a(t_1(x))), \dots, t_m^{-1}(a(t_m(x)))$. Para una definición completa de estas métricas, consultar [Qui20].

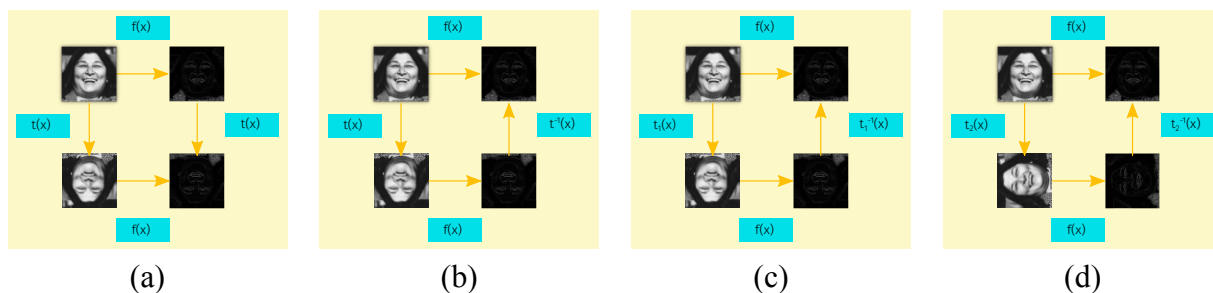


Figura 8: (a) Auto-equivarianza (b) Auto-equivarianza con t invertible. (c,d) Valores a comparar para distintas transformaciones.

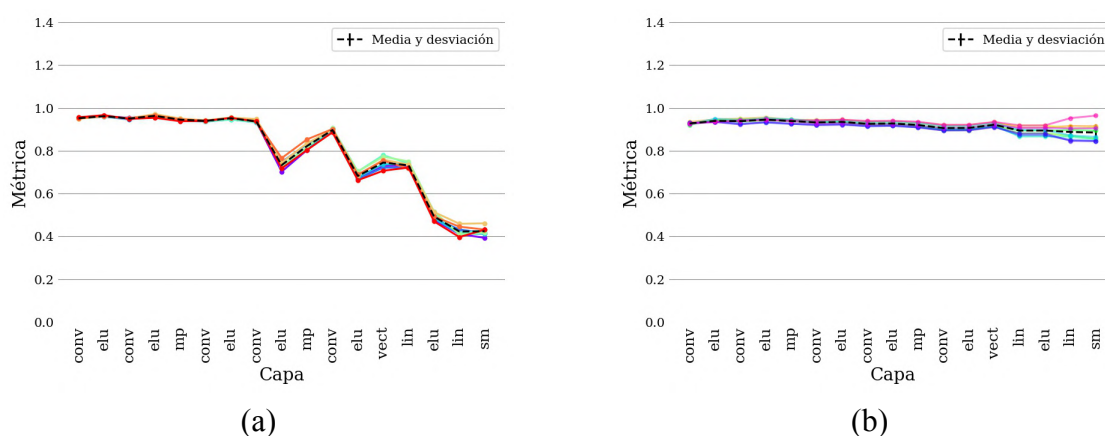


Figura 9: (a) Inicialización aleatoria (b) Pesos aleatorios.

3.3. Experimentos con Métricas de Invarianza

En esta sección, resaltamos 2 de los más de 30 experimentos³ realizados con las métricas definidas y que nos permiten caracterizar el comportamiento de las redes neuronales en términos de su invarianza y equivarianza.

Nuevamente aclaramos que sólo mostraremos los resultados para SimpleConv, MNIST, rotaciones, y la métrica de Invarianza basada en Varianza Normalizada. Para una descripción completa de estos y otros experimentos, consultar [Qui20].

Invarianza de modelos inicializados aleatoriamente Para estudiar el efecto de la inicialización aleatoria del modelo en las invarianza, entrenamos 10 instancias de cada modelo utilizando la misma arquitectura, base de datos y conjunto de transformaciones, hasta que las mismas converjan. Cada instancia del mismo modelo fue inicializada con distintos pesos aleatorios. Luego, todas las instancias fueron evaluadas con la misma métrica.

La Figura 9 (a) muestra el valor promedio de la métrica para cada capa. Los resultados sugieren que la invarianza no varía con respecto a la inicialización de sus pesos. Esto indica que la invarianza es una propiedad emergente del modelo, sin importar la inicialización aleatoria de sus pesos. Por ende, la misma depende mayormente de la arquitectura del modelo, las muestras y las transformaciones, y no el proceso de entrenamiento.

Además, la estabilidad de las métricas con respecto a la inicialización del modelo es una propiedad deseable, ya que implica que no se requiere entrenar varias instancias de un modelo y medirlos para obtener un valor de invarianza promedio del mismo.

³El código de los experimentos está disponible en https://github.com/facundoq/transformational_measures_experiments/

Invarianza de modelos con pesos aleatorios Para comprender aún mejor el origen de la estructura del invarianza de las redes, medimos dicha propiedad en modelos sin entrenar, es decir, modelos con pesos completamente aleatorios. Dado que la inicialización es aleatoria, calculamos la métrica para ocho modelos aleatorios en cada caso (Figura 9 (b)).

Nuevamente, podemos ver que la invarianza no depende del conjunto de pesos aleatorio de la red. Esto sugiere que dada una inicialización controlada, la invarianza es mayormente una propiedad de la arquitectura de la red en lugar de los valores específicos de los pesos. Por otro lado, es interesante que la invarianza de las redes sin entrenar no tiene tendencia, es decir no varía en base a las capas, a diferencia de las ya entrenadas (Figura 9 (a)).

4. Conclusiones

La invarianza y la equivarianza son conceptos clave para comprender el funcionamiento de las redes neuronales.

Presentamos una serie de métricas flexibles, eficientes e interpretables para cuantificar la invarianza y auto-equivarianza de cualquier modelo basado en Redes Neuronales. Nuestras métricas son completamente adaptables en términos de la arquitectura de la red, tipos de capas, conjuntos de transformaciones y formatos de entrada. Las métricas tienen una granularidad alta, permitiendo medir éstas propiedades en cada activación de la red de forma individual, y luego combinar dichos resultados para evaluar estructuras de más alto nivel como capas o redes enteras. A diferencia de otras métricas, son fáciles de interpretar y eficientes para calcular en redes grandes con millones de activaciones. Además, desarrollamos una librería de código abierto que provee implementaciones de referencia de las métricas desarrolladas.

Los experimentos realizados nos permiten realizar un análisis de la invarianza y la equivarianza de los modelos y sus características. Las conclusiones pueden ser utilizadas para crear mejores modelos equivariantes. Además del análisis, las comparaciones con aumentación de datos y las métricas presentan en conjunto una metodología poderosa para evaluar la equivarianza de los modelos y así comprenderlos más completamente.

4.1. Trabajos futuros

Creemos que es posible aprender más acerca de las Redes Neuronales estudiando sus equivarianzas, y así mejorar los modelos existentes para hacer posible nuevas aplicaciones. Para avanzar en esta dirección, identificamos tres áreas de trabajo.

Primero, proponemos mejorar la facilidad de uso de la librería de métricas, implementando soporte para el framework TensorFlow, aumentando la eficiencia en el cálculo de las métricas y permitiendo su ejecución en GPU o CPU de forma indistinta. Segundo, buscaremos profundizar la caracterización teórica de las métricas. Por un lado, desarrollando pruebas de hipótesis específicos para cada métrica. Por el otro, generalizando las métricas para obtener una métrica de equivarianza general. Tercero y último, buscaremos ampliar el análisis de los modelos, incluyendo más modelos invariantes (GCNN, capsules, etc), distintos tipos de modelos (recurrentes, GANs, etc), y de otros dominios, como procesamiento de texto, audio, o ejemplos adversariales.

Referencias

- [CW16] Taco S. Cohen y Max Welling. «Group Equivariant Convolutional Networks». En: *arXiv:1602.07576 [cs, stat]* (feb. de 2016). arXiv: 1602.07576. url: <http://arxiv.org/abs/1602.07576> (visitado 08-05-2018).

- [GBC16] Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep Learning*. MIT Press, 2016. url: <http://www.deeplearningbook.org>.
- [Goo+09] Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe y Andrew Y Ng. «Measuring invariances in deep networks». En: *Advances in neural information processing systems*. 2009, págs. 646-654.
- [Jad+15] Max Jaderberg, Karen Simonyan, Andrew Zisserman y Koray Kavukcuoglu. «Spatial Transformer Networks». En: *arXiv:1506.02025 [cs]* (jun. de 2015). arXiv: 1506.02025. url: <http://arxiv.org/abs/1506.02025> (visitado 08-05-2018).
- [LV15] Karel Lenc y Andrea Vedaldi. «Understanding image representations by measuring their equivariance and equivalence». En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, págs. 991-999.
- [Qui+16a] Facundo Quiroga, Franco Ronchetti, César Armando Estrebou y Laura Cristina Lanzarini. «Handshape recognition for argentinian sign language using probsom». En: *Journal of Computer Science & Technology* 16.1 (2016). issn: 1666-6038.
- [Qui+16b] Facundo Quiroga, Franco Ronchetti, César Armando Estrebou, Laura Cristina Lanzarini y Alejandro Rosete. «LSA64: An Argentinian Sign Language Dataset». En: *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*. Ed. por Springer. Red de Universidades con Carreras en Informática (RedUNCI). 2016, págs. 794-803.
- [Qui+16c] Facundo Quiroga, Franco Ronchetti, César Estrebou, Laura Lanzarini y Alejandro Rosete. «Sign language recognition without frame-sequencing constraints: A proof of concept on the argentinian sign language». En: *Ibero-American Conference on Artificial Intelligence*. Springer International Publishing. 2016, págs. 338-349.
- [Qui+17] Facundo Quiroga, Ramiro Antonio, Franco Ronchetti, Laura Cristina Lanzarini y Alejandro Rosete. «A study of convolutional architectures for handshape recognition applied to sign language». En: *XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017)*. Springer International Publishing. 2017.
- [Qui+18] Facundo Quiroga, Franco Ronchetti, Laura Lanzarini y Aurelio F Bariviera. «Revisiting Data Augmentation for Rotational Invariance in Convolutional Neural Networks». En: *International Conference on Modelling and Simulation in Management Sciences*. Springer. 2018, págs. 127-141.
- [Qui+19] Facundo Quiroga, Jordina Torrents-Barrena, Laura Lanzarini y Domenec Puig. «Measuring (in) variances in Convolutional Networks». En: *Conference on Cloud Computing and Big Data*. Springer. 2019, págs. 98-109.
- [Qui20] Facundo Quiroga. «Medidas de Invarianza y Equivarianza a Transformaciones en Redes Neuronales Convolucionales.» Tesis doct. La Plata, Buenos Aires, Argentina: Universidad Nacional de La Plata, mar. de 2020.
- [Ron+15] Franco Ronchetti, Facundo Quiroga, Laura Lanzarini y Cesar Estrebou. «Distribution of Action Movements (DAM): a Descriptor for Human Action Recognition». En: *Frontiers of Computer Science* 9.6 (2015), págs. 956-965. issn: 2095-2236. doi: 10.1007/s11704-015-4320-x. url: <http://dx.doi.org/10.1007/s11704-015-4320-x>.
- [SG18] Megha Srivastava y Kalanit Grill-Spector. «The Effect of Learning Strategy versus Inherent Architecture Properties on the Ability of Convolutional Neural Networks to Develop Transformation Invariance». En: *CoRR abs/1810.13128* (2018). arXiv: 1810.13128. url: <http://arxiv.org/abs/1810.13128>.