Deep Learning: Current State

J. Salas, *Member, IEEE*, F. Vidal, and J. Martínez-Trinidad

***Abstract*—Deep learning, a derived from machine learning, has grown into widespread usage with applications as diverse as cancer detection, elephant spotting, and game development. The number of published studies shows an increasing interest by researchers because of its demonstrated ability to achieve high performance in the solution of complex problems, the wide availability of data and computing resources, and the groundbreaking development of effective algorithms. This paper reviews the current state of deep learning. It includes a revision of basic concepts, such as the operations of feed forward and backpropagation, the use of convolution to extract features, the role of the loss function, and the optimization and learning processes; the survey of main stream techniques, in particular convolutional, recurrent, recursive, deep belief, deep generative, generative adversarial, and variational auto-enconder neural networks; the description of an ample array of applications organized by the type of technique employed; and the discussion of some of its most intriguing open problems.**

***Index Terms*—Applications of Deep Learning, Convolutional Neural Networks, Deep Generative Networks, Recursive Neural Networks, Recurrent Neural Networks.**

I. Introducción

DOMINGOS [1] clasifica los principales esfuerzos para desarrollar inteligencia artificial en evolutivos [2], bayesianos [3], simbólicos [4], analógicos [5] y conexionistas [6]. Una aplicación de la inteligencia artificial es el aprendizaje automático, el cual se ha vuelto cada vez más popular y actualmente se utiliza en aplicaciones que incluyen detección temprana del cáncer [7], desarrollo de juegos [8], detección de objetos [9], sistemas de recomendación [10], reconocimiento de voz [11], análisis de redes sociales [12], reconocimiento de acciones en video [13] y minería de texto [14]. Entre las técnicas de aprendizaje automático, el aprendizaje profundo (una aproximación conexionista) se destaca pues permite la extracción automática de características. Lecun *et al.* [15] y Goodfellow *et al.* [16] definen el aprendizaje profundo como un conjunto de métodos de aprendizaje multi-nivel, obtenidos mediante la composición de módulos simples no lineales que abstraen progresivamente la representación de los datos de entrada. Interesantemente, algunos investigadores han observado que los métodos de aprendizaje profundo tienen una mayor semejanza con programación de computadoras que con neurobiología, y han sugerido rebautizarla como *programación diferencial* [17] o *software 2.0* [18].

Las primeras ideas que pudieran tomarse como antecedentes del aprendizaje profundo podrían atribuirse a Aristóteles que, en el año 300 a. de C., propuso el *asociacionismo* para describir la forma en que funciona el cerebro. Sin embargo, es hasta 1943 que McCulloch y Pitts [19] introdujeron el primer modelo de red neuronal artificial, imitando la función de la neocorteza en el cerebro humano, cuando el aprendizaje profundo comenzó a evolucionar. Entonces, Rosenblatt [20] dio otro paso significativo a través de la implementación de la teoría hebbiana en un dispositivo electrónico llamado *perceptron* [20]. Más tarde, Werbos [21] introdujo el proceso de entrenamiento de redes neuronales artificiales a través de la retro-propagación de errores. En 1980, Fukushima introdujo el *neocognitron* [22], que a su vez inspiró el desarrollo de las redes neuronales convolucionales (CNN, *Convolutional Neural Networks*) [23] y posteriormente de las redes neuronales recurrentes (RNN, *Recurrent Neural Networks*) [24]. Luego, en 1998, Lecun *et al.* [23] implementó *LeNet*, la primera red neuronal profunda. En el 2006, las redes de creencia profunda (DBN, *Deep Belief Networks*) [25], junto con un pre-entrenamiento en capas, impulsaron el uso generalizado del aprendizaje profundo. La idea detrás de estas redes consiste en entrenar un modelo no supervisado simple de dos capas como una Máquina de Boltzmann Restringida (RBM, *Restricted Boltzmann Machine*) [26], donde los parámetros se fijan, se agrega una nueva capa en la parte superior y sólo los parámetros para la nueva capa son entrenados. Esta técnica permitió incrementar el número de capas que hasta ese momento se habían logrado entrenar.

Este artículo describe el estado actual de la ciencia, las técnicas y las aplicaciones del aprendizaje profundo. También incluye conceptos básicos clave y una discusión de los problemas abiertos, dando una idea general del campo. El manuscrito está organizado de la siguiente manera: la Sección II revisa los conceptos de aprendizaje profundo, incluidas las técnicas de convolución como extractores de características, las funciones de pérdida como directoras de la optimización, y la retro-propagación para la optimización. Luego, la Sección III presenta algunas Redes Neuronales Profundas (DNN, *Deep Neural Networks*), incluyendo las CNN, las RNN y las Redes Generativas Profundas (DGN, *Deep Generative Networks*). A continuación, la Sección IV revisa diferentes aplicaciones de aprendizaje profundo. Finalmente, este artículo concluye discutiendo los desafíos y las líneas futuras de investigación en el campo.

II. Conceptos Básicos

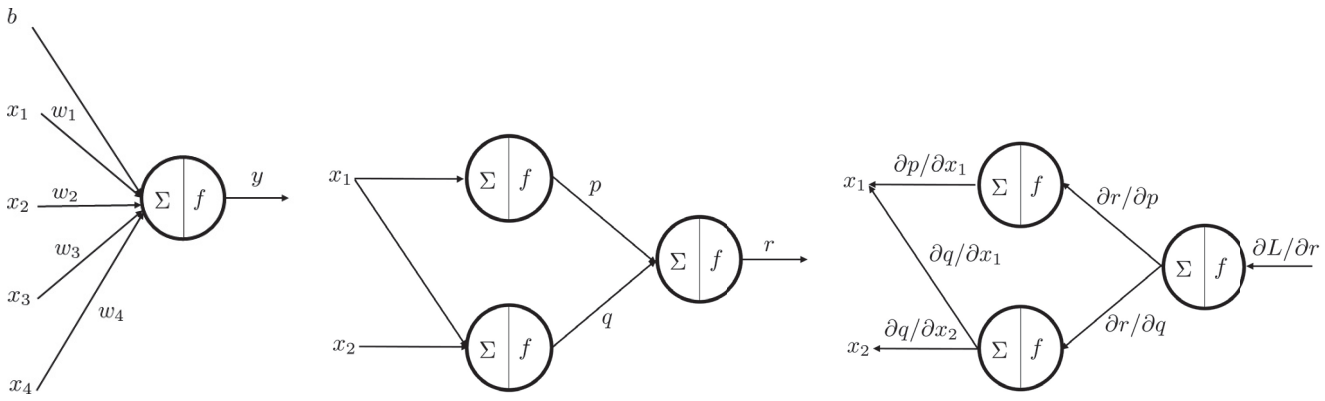
Existe cierta controversia sobre si las DNN son mejores aproximadores de funciones que las redes neuronales poco profundas (aquellas con sólo una capa oculta), con algunos investigadores que prefieren las primeras [27] y otros las segundas [28]. En todo caso, existe un consenso sobre el

Joaquín Salas was partially supported by IPN under grant SIP2019.
Send your correspondence to Joaquín Salas, Cerro Blanco 141, Colinas del Cimatario, Querétaro, México, 76090. salas@ieee.org.

Joaquín Salas is with the Instituto Politécnico Nacional, México.

Flavio B. Vidal is with the University of Brasilia, Brazil.

José Fco. Martínez-Trinidad is with the Instituto Nacional de Astrofísica, Óptica y Electrónica, México.



(a) Neurona artificial (b) *feedforward* (c) *retro-propagación*

Fig. 1. *Etapa de Estimación.* En cada neurona artificial (a) se calcula la posición en un plano con parámetros llamados pesos $\mathbf{w}^T = (w_1, \ldots, w_n)$ y sesgos b de las entradas $\mathbf{x}^T = (x_1, \ldots, x_n)$ mediante la operación $z = \mathbf{w}^T\mathbf{x} + b$. El valor de z es proyectado hacia un espacio no lineal mediante la operación $y = f(z)$. La ejecución de este proceso para todas las neuronas a través de las capas de la red es conocido como *feedforward*, donde se calcula el valor de la función para una cierta entrada (b). En entrenamiento, *retro-propagación* nos permite calcular el valor del gradiente de la función respecto a las entradas, mediante la regla de la cadena (c).

carácter innovador de las DNN. Por lo general, uno organiza las DNN como flujos de capas secuenciales, donde tiene lugar la composición de funciones. Si el flujo tiene ciclos, tenemos RNN [29]. De lo contrario, tenemos una red neuronal *feed-forward*, *i.e.*, un perceptrón de múltiples capas que incluyen una de entrada, al menos una oculta y una de salida. En esta sección, revisamos algunos conceptos en DNN, enfatizando el estado actual de la técnica.

A. Etapa de Estimación

El término *profundo* (en inglés *deep*) en el acrónimo DNN es debido a su capacidad para aprender características jerárquicamente, más que a su número de capas [15]. En las DNN, a medida que el flujo del procesamiento de datos avanza, capa tras capa, se obtienen características más generales. En cada etapa, para cada neurona se calcula una suma ponderada de las entradas. Luego, se proyecta el resultado en un espacio no lineal utilizando una función de activación (ver Figura 1(a)-(b)). Al principio, los investigadores utilizaron sigmoides (como la función logística y la tangente hiperbólica) como funciones de activación, ya que son continuos, derivables y, por lo tanto, adecuados para la retro-propagación. Sin embargo, tanto para valores positivos como negativos, el desvanecimiento de su derivada impide el aprendizaje. Como respuesta a ése problema, Nair y Hinton [30] propusieron la función de activación de la unidad lineal rectificada (ReLU, *Rectified Linear Unit*), que genera y pone a cero su argumento para valores positivos y negativos, respectivamente. En la literatura, ha habido un esfuerzo considerable para aliviar el problema de las entradas negativas de ReLU yéndose a cero para sus gradientes [31].

B. Extracción de Características

La inclusión de convoluciones para extraer características fue la innovación primordial introducida por las DNN. Antes de las DNN, uno tenía que extraer ingeniosamente las características relevantes para el aprendizaje. Una convolución corresponde a la respuesta lineal de una señal a un operador. Es importante destacar que las DNN aprenden los operadores durante el entrenamiento, y la definición de las convoluciones minimiza el valor de la función de pérdida [15]. Además, dado que las convoluciones operan localmente, comparten los parámetros a través del dominio de la entrada, pero a medida que avanza el procesamiento de los datos, resultan características cada vez más abstractas.

A pesar de su éxito, existe cierta preocupación sobre la confiabilidad de las DNN en operaciones críticas. Considere a Su *et al.* [32], quienes introducen un método para generar perturbaciones que con sólo un píxel reducen sustancialmente el rendimiento de los clasificadores de imágenes. Como consecuencia, los investigadores han realizado un esfuerzo considerable en aumentar la transparencia de las DNN. Estos esfuerzos se han conjuntado con la creación de: a) mapas de prominencia para facilitar la interpretabilidad [33]; interfaces para comprender el funcionamiento interno de las arquitecturas de aprendizaje profundo [34]; o herramientas didácticas para aumentar el nivel de entendimiento de los usuarios sobre la tecnología [35]. Aunque hemos avanzado en metodologías para comprender la dinámica del aprendizaje y el diseño [36], aún hay un amplio espacio para avances teóricos que permitan analizar el funcionamiento de las DNN.

C. Función de Pérdida

Las DNN cuantifican el concepto de aprendizaje como la minimización de la función de pérdida. Para la regresión, las funciones de pérdida típicas incluyen las normas $\mathcal{L}_1$ (la suma de los valores absolutos) y $\mathcal{L}_2$ (la raíz cuadrada de la suma de los valores al cuadrado), mientras que para la clasificación, las funciones de pérdida más extendidas incluyen la función de pérdida denominada *hinge* (que ganó popularidad con el advenimiento del clasificador llamado *Support Vector Machine* [37]) y la función de pérdida de entropía cruzada (estrechamente relacionada con la entropía de Shannon para la teoría de la información [38]).

Hoy en día hay consenso sobre la importancia de definir funciones de pérdida específicas para el problema a resolver. Algunos ejemplos incluyen los trabajos de Zhao *et al.* [39] quienes usan el índice de similitud estructural [40] para la restauración de imágenes, Yao *et al.* [41] quienes proponen incorporar costos de pérdida parcial para la re-identificación de una persona, y Ward *et al.* [42] quienes introducen una función de pérdida para la localización basada en imágenes, que considera el acoplamiento entre rotación y traslación para definir la posición de la cámara. De hecho, un área activa de investigación es la definición de funciones de pérdida que promueven la eficiencia de cómputo. Por ejemplo, considere el problema de aprendizaje de distancias métricas profundas, que tiene como objetivo aprender características eficientes para espacios embebidos. En este problema, Do *et al.* [43] mejoraron la eficiencia en el cálculo de la función de pérdida mediante la introducción de centroides virtuales incrustados, lo que resultó en la reducción de la complejidad algorítmica de la solución al problema.

D. Retro-Propagación (Backpropagation)

El grafo de las DNN que ilustra el flujo de procesamiento es una representación conveniente de la derivada de la función de pérdida con respecto a los datos de entrada, a través de la *regla de la cadena* [44] (ver Figura 1(c)). Por ello, una opción natural para minimizar la función de pérdida es el *descenso por gradiente*, y la piedra angular para obtener iterativamente los parámetros óptimos es el procedimiento de *retro-propagación*. Utilizando la retro-propagación, se calcula la nueva estimación de los parámetros considerando su valor actual y una forma de la regla de la cadena ponderada (a través de una *tasa de aprendizaje*) del gradiente.

Las DNN operan en estructuras de datos densas. Sin embargo, las neuronas de un cerebro biológico parecen funcionar mediante impulsos, inspirando a los investigadores a buscar soluciones comparables [45]. En esa dirección, las arquitecturas exhibirían una alta dispersión temporal y espacial, lo que daría como resultado sistemas neuromórficos con alta eficiencia energética, donde los impulsos se comunicarían en forma asincrónica. Actualmente, esta aproximación ha generado métodos para integrar la retro-propagación en una red neuronal con impulsos mediante el uso de acumuladores (neuronas que integran y disparan, IF, *Integrate and Fire*) para la propagación de errores, discretizando el error en la forma de impulsos [46]. Lee *et al.* [47] extendieron la noción de impulsos a las DNN. Además, para mejorar la velocidad de entrenamiento, Pu y Wang [48] introdujeron una red neuronal de retro-propagación de orden fraccional entrenada con un método de descenso más pronunciado, también de orden fraccional.

Para la búsqueda de los parámetros óptimos durante el entrenamiento de una RNN, se utiliza la retro-propagación a través del tiempo (BPTT, *BackPropagation Through Time*) en lugar de la versión común de simple retro-propagación [49]. Una posible forma de comprender la analogía es utilizar una representación desplegada en el tiempo de las RNN y aplicar la retro-propagación a esta representación. Así, es posible apreciar que el problema es difícil porque la misma neurona tiene que reconstruir el historial de activación para resolver un *problema de asignación temporal*. Las investigaciones sobre las RNN se centran actualmente en evitar la explosión o el desvanecimiento de gradientes. Por lo general, esto se logra a través del diseño de la red (a través de la memoria a corto y largo plazo, LSTM (*Long-Short Term Memory*) o unidades recurrentes activadas, GRU (*Gated Recurrent Unit*) [16]) u optimización (recorte de gradiente a cierto valor de umbral). De hecho, la propagación hacia atrás es biológicamente inverosímil, *i.e.*, para aprender, uno tendría que transmitir señales de error hacia atrás en el tiempo. Para solventar esto, Bellec *et al.* [50] fusionan información disponible localmente y muestran que esto proporciona una excelente aproximación a la BPTT.

E. Optimización

La función de pérdida dirige el proceso de optimización. Debido a su simplicidad, al gran corpus de datos comúnmente asociado con el aprendizaje profundo y la representación de gradiente de la regla de la cadena definida por el grafo de computación, los practicantes han preferido el uso de la técnica de descenso por gradiente, o sus derivados, para la optimización [51]. En la mayoría de los casos (tal vez excepto cuando se aplica *Curriculum Learning* [52]), uno estima el gradiente a partir de una muestra aleatoria, lo cual da pie al descenso del gradiente estocástico [53]. Algunas mejoras que buscan vincular las actualizaciones al gradiente de la función de pérdida incluyen la incorporación de *momentum* [54] (para acelerar la búsqueda en la dirección cuyo gradiente es consistente) y el cálculo del gradiente con respecto al valor futuro de los parámetros (gradiente acelerado de Nesterov, NAG, *Nesterov's Accelerated Gradient*) [55]. Otras mejoras incluyen restringir las actualizaciones de cada parámetro. Por ejemplo, Adagrad (el algoritmo de gradiente adaptativo, *Adaptive Gradient Algorithm*) [56] normaliza el gradiente por la raíz cuadrada de la acumulación del cuadrado de los gradientes. Dado que esta operación da como resultado la pérdida de la capacidad para aprender, Zeiler propuso Adadelta [57], que restringe la acumulación a una ventana fija que toma en cuenta el promedio del cuadrado de gradientes anteriores o pasados. Una alternativa a Adadelta es RMSProp (*Root Mean Square Propagation*) [58], que utiliza un promedio exponencialmente decreciente de gradientes pasados. Por su parte, Adam (*Adaptive Moment Estimation)* [59] es el resultado de combinar RMSProp y NAG. Adam utiliza la versión con correción del sesgo de los momentos de primer y segundo orden del gradiente para modificar el valor de los parámetros. Del mismo modo, una combinación de Adam y NAG da como resultado Nadam [60], donde el *momentum* se aplica una vez y no dos veces (para actualizar el gradiente y actualizar los parámetros). Adamax [59] desciende de Adam y generaliza la actualización de la norma $\mathcal{L}_\infty$. Finalmente, se puede observar que en Adadelta el promedio móvil exponencial de los gradientes cuadrados pasados dificulta la generalización. Por ello, AMSGrad [61] usa el máximo de los gradientes pasados. Recientemente, Zhang *et al.* [62] introdujeron *Lookahead*,

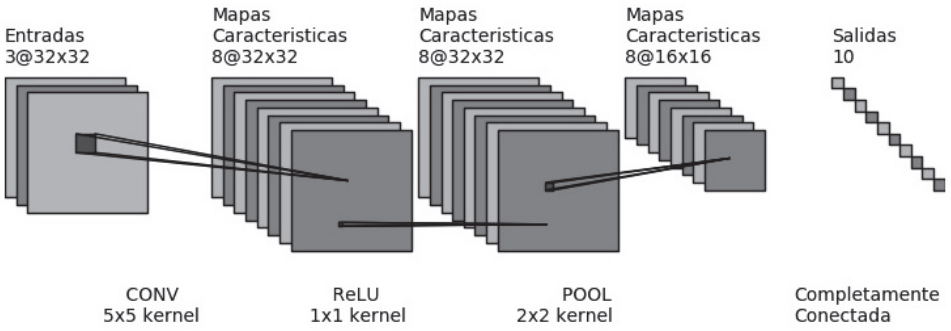


Fig. 2. Red Neuronal Convolucional. Una red neuronal establece el mapeo entre *entradas* y *salidas*. Para ello se construye una representación jerárquica porque incrementa su abstracción al avanzar a través de las capas. Lo que hace interesante a una *red neuronal convolucional* es su capacidad para extraer las mapas de características a partir de los datos. En cada capa el resultado de la convolución se proyecta hacia un espacio no lineal; por ejemplo, mediante la función ReLU. Se ha encontrado útil incluir operaciones intermedias; por ejemplo, sumarizar usando *pooling*, aunque no se incluyen otras comunes, como *normalización de lotes de datos (batch normalization)*.

un optimizador que asume la existencia de parámetros de red *lentos* y *rápidos*. En *Lookahead*, se usa un optimizador estándar varias veces en los parámetros *rápidos* por cada vez que se usa el optimizador en los parámetros *lentos*, lo que da al primero una mejor oportunidad de realizar una actualización correcta.

Un problema que ocurre durante la optimización es que el espacio de búsqueda para los mejores parámetros puede ser enorme. Por lo tanto, las soluciones de las DNN son propensas al sobreajuste. Para abordar este problema, uno puede aplicar varios enfoques de regularización. Una de ellos consiste en introducir la regularización $\mathcal{L}_p$ en forma de una restricción de suavizado para la función de pérdida, generalmente como un término que arroja un costo en la magnitud de los parámetros [63]. Otros esquemas incluyen *dropout* [64], donde algunas de las aristas en el grafo no se consideran durante la retro-propagación, y paro temprano [65], donde si la tasa de entrenamiento sigue descendiendo y la tasa de validación comienza a aumentar, el aprendizaje se detiene.

F. Aprendizaje

Uno podría clasificar las técnicas de aprendizaje entre supervisadas (existen etiquetas), sin supervisión (no existen etiquetas), semi-supervisadas (existe un pequeño número de etiquetas), débilmente supervisadas (existen etiquetas muy generales) y aprendizaje por refuerzo (RL, *Reinforcement Learning*) [66]. En *aprendizaje supervisado*, tenemos pares de características y valores de referencia (*ground truth*). Igualmente, podemos distinguir un proceso de aprendizaje entre si comparamos el modelo resultante con otros, o no. En el primer caso, uno divide los datos en conjuntos de validación-entrenamiento, y prueba [67]. Luego, se divide el conjunto de validación-entrenamiento en conjuntos de entrenamiento y validación. En el último caso, dividimos los datos en los conjuntos de entrenamiento y prueba. Uno llama validación cruzada (CV, *Cross-Validation*) a una iteración en la selección de los conjuntos de entrenamiento y el otro conjunto (validación para configuraciones comparadas y prueba para las no comparadas). Afrendas y Markatou [68] muestran que para la CV, el tamaño de muestra óptimo para el conjunto de entrenamiento es la mitad del tamaño de la muestra. La interacción entre el conjunto de entrenamiento y el otro conjunto (validación o prueba, dependiendo de la configuración) permite inferir las propiedades de generalización del sistema y evaluar la posibilidad de sobreajuste. Desafortunadamente, los datos etiquetados abundantes y de alta calidad son escasos o caros. Por tanto, los investigadores han recurrido a la extracción de información mediante aprendizaje por auto-supervisión [69].

Por su parte, la idea de RL Profundo (DRL, *Deep Reinforcement Learning*) es que el sistema aprende de su interacción con el entorno [70]. Un agente interactúa con su entorno a través de una serie de acciones definidas por una política, modificando su entorno y, por lo tanto, sus observaciones posteriores. Por lo general, uno enmarca RL como un proceso de decisión de Markov [71] donde uno se mueve entre estados dependiendo de las acciones. En DRL [72] uno no diseña el estado explícitamente, sino que el agente actúa siguiendo las políticas de transición entre estados para los que corresponde una recompensa.

El uso del aprendizaje profundo en varios problemas prácticos ha motivado desafíos que actualmente están bajo investigación. Por ejemplo, el problema de aprendizaje no supervisado [73], el aprendizaje en línea para la transmisión de datos [74], el desarrollo de nuevas técnicas de optimización para el entrenamiento de redes neuronales profundas [75], la creación de técnicas de aprendizaje profundo distribuidas para acelerar el proceso de entrenamiento [76], y el uso de aprendizaje profundo multimodal [77].

III. Redes Neuronales Profundas

En esta sección revisamos las arquitecturas de DNN que han resistido la prueba del tiempo, así como las que han aparecido recientemente en la literatura.

A. Redes Neuronales Convolucionales

De manera similar a las redes neuronales poco profundas, las CNN están inspiradas en las neuronas de los cerebros de animales [78], pero tienen como ventaja que comparten parámetros, interacciones escasas y representaciones equivalentes [16]. Las CNN (ver Figura 2), a diferencia de las capas completamente conectadas, utilizan conexiones locales

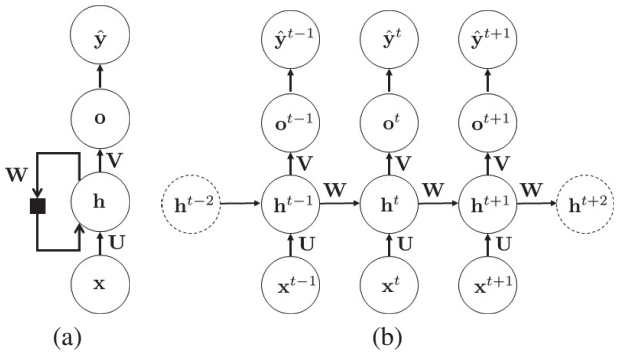


Fig. 3. *Redes Neuronales Recurrentes.* Una red recurrente tiene ciclos (a). Para ganar intuición, uno pudiera considerarlas una red regular que se desarrolla en el tiempo, conservando los pesos ($\mathbf{W}$, $\mathbf{U}$, $\mathbf{V}$) en cada nueva iteración (b). El problema de la inestabilidad numérica se resuelve mediante compuertas en las arquitecturas tipo LSTM o GRU.

y pesos compartidos, lo que resulta en redes más rápidas y con menos parámetros que son más fáciles de entrenar. En las CNN las funciones de activación no lineal y las capas de *pooling* (sumarización) siguen a las capas convolucionales, mientras que se usan capas completamente conectadas al final del flujo de procesamiento. Estas capas toman características de bajo y medio nivel para generar abstracción de alto nivel de los datos. La última capa (mediante una función *softmax* o *hinge*) se usa para calcular los puntajes de clasificación, es decir, una medición de la pertenencia de una instancia a la clase.

B. Redes Neuronales Recurrentes

Las RNN tienen conexiones que permiten que la información se mueva hacia adelante, a la misma o a una capa anterior (ver Figura 3). Gracias a estas conexiones recurrentes, estas redes pueden tener en cuenta el historial acumulado y, en consecuencia, a menudo se usan para procesar datos temporales. Esta propiedad es esencial en muchas aplicaciones donde la estructura embebida en la secuencia de los datos transmite conocimientos útiles. Por ejemplo, para comprender una palabra en una oración, es necesario conocer el contexto. Por ende, uno puede considerar a las RNN como unidades de memoria a corto plazo que incluyen la capa de entrada, la capa oculta (que puede incluir su estado) y la capa de salida. Razvan *et al.* [79] presentan tres enfoques de RNN profundas; a saber, *Input-to-Hidden*, *Hidden-to-Output*, y *Hidden-to-Hidden*, que aprovechan la profundidad de las RNN y reducen la dificultad del aprendizaje. Un problema importante en las RNN es su sensibilidad a la desaparición o explosión de los gradientes [80]. En otras palabras, los gradientes pueden decaer o crecer exponencialmente durante el entrenamiento debido a la multiplicación de muchas derivadas con valores pequeños o grandes. Como respuesta, se introdujo LSTM y GRU [81], [82] proporcionando bloques de memoria a sus conexiones recurrentes. Cada bloque de memoria contiene una o más celdas de memoria auto-conectadas y compuertas multiplicativas para controlar el flujo de información. Además, según He *et al.* [83] las conexiones residuales en redes muy profundas pueden aliviar significativamente el problema de desvanecimiento del gradiente.

C. Redes Neuronales Recursivas (RvNN)

Las RvNN son modelos adaptativos no lineales que aprenden información profunda y estructurada. La memoria recursiva auto-asociativa (RAAM) [84], una arquitectura creada para procesar objetos estructurados en forma arbitraria, como árboles o grafos, inspiró el desarrollo de las RvNN. El enfoque consiste en tomar una estructura de datos recursiva de tamaño variable y generar una representación distribuida de ancho fijo. El esquema de aprendizaje *Backpropagation Through Structure* (BTS) se introdujo para entrenar a la red [84]. BTS es similar al algoritmo de retro-propagación estándar, pero admite un grafo de estructura en forma de árbol. La red se entrena mediante auto-asociación para reproducir los datos de la capa de entrada en la capa de salida.

D. Redes Generativas Profundas (DGN)

Las DGN aprenden modelos de la distribución actual de los datos del conjunto de entrenamiento mediante la generación de nuevas muestras, con algunas variaciones. Esto permite obtener una distribución lo más similar posible a la distribución de datos real. A continuación, revisamos los enfoques más utilizados y eficientes.

1) Redes de Creencia Profundas y Redes de Boltzmann Profundas: Las DBN [85] son modelos generativos probabilísticos híbridos en los que las dos capas superiores forman una RBM típica con conexiones no dirigidas, y las capas inferiores usan conexiones dirigidas para recibir entradas de la capa superior. La capa más baja, que es la capa visible, representa los estados de las unidades de entrada como un vector de datos. Las DBN aprenden a reconstruir probabilísticamente sus entradas en un enfoque auto supervisado, mientras que sus capas actúan como detectores de características en las entradas. Además, un proceso adicional de entrenamiento supervisado da a las DBN la capacidad de realizar tareas de clasificación. Uno puede considerar a las DBN como compuestas de varias RBMs [26], donde la capa oculta de cada subred se puede ver como la capa de entrada para la siguiente subred. Las DBN incluyen un algoritmo voraz para mejorar el modelo generativo, al permitir que cada subred reciba secuencialmente diferentes representaciones de los datos, ya que, de manera ideal, una RBM no podrá modelar los datos originales. Una vez que la red aprende los pesos iniciales, los datos pueden asignarse a través de la matriz de ponderación transpuesta para crear los datos de nivel superior para la siguiente capa. Hinton *et al.* [25] muestran que el logaritmo de la probabilidad de cada vector de datos de entrada está acotado por su distribución aproximada. Además, cada vez que se agrega una nueva capa a la DBN, los límites de variación en la capa más profunda, que inicializa el nuevo bloque RBM, se mejoran en comparación con la capa anterior. Por otra parte, las DBM [86] difieren de las DBN en que tienen una RBM no dirigida en las capas inferiores. El algoritmo de entrenamiento voraz en capas para las DBM se puede calcular fácilmente modificando el procedimiento de las DBN. Una aproximación factorial a la probabilidad posterior puede tomar el resultado de la primera RBM o la probabilidad de la segunda capa. Otra opción interesante es

tomar la media geométrica de estas dos distribuciones para equilibrar las aproximaciones a la probabilidad posterior.

2) Redes Generativas Adversariales (GAN): Las GAN [87] constan de un modelo generativo G y un modelo discriminativo D. Mientras G captura localmente la distribución p_g sobre los datos reales t, D intenta diferenciar una muestra que proviene de los datos de modelado m, representada por la distribución p_m, en lugar de p_g. En cada iteración de la retro-propagación, el generador y el discriminador compiten entre sí. Mientras el generador produce datos más realistas para engañar al discriminador, este último trata de distinguir los datos reales de los falsos generados por G. El modelo se considera estable cuando ambos alcanzan el punto en que ninguno de ellos puede mejorarse, cuando $p_g = p_m$. Es decir, el discriminador ya no puede distinguir la fuente de datos.

3) Auto-codificadores Variacionales (VAE): Los VAE [88] utilizan una red neuronal profunda para aprender representaciones de datos complejos mediante auto-supervisión. Los VAE incluyen un codificador y un decodificador, los cuales son redes neuronales. El codificador aprende variables latentes de la entrada, y el decodificador genera una salida basada en muestras de las variables latentes. Dados suficientes datos de entrenamiento, el codificador y el decodificador se pueden entrenar por completo al minimizar la pérdida de reconstrucción y la divergencia de *Kullback-Leibler* entre las distribuciones de variables latentes y las distribuciones normales independientes. Los VAE condicionales [89] son versiones extendidas de los VAE donde el codificador y el decodificador pueden tomar una variable auxiliar como entrada adicional.

IV. Aplicaciones

Podemos encontrar aplicaciones de aprendizaje profundo en áreas como finanzas [90], medicina [91], deportes [92], educación [93], contaminación del aire [94], análisis de documentos [95], biología [96], geografía [97], manufactura [98], transporte [99], robótica [100] y electrónica [101], entre otras. Uno podría agrupar estas aplicaciones en términos del tipo de red neuronal profunda empleada, como a continuación detallamos.

A. Redes Neuronales Convolucionales

Los desarrolladores han creado una amplia gama de aplicaciones de CNN que van desde el reconocimiento de imágenes en grandes bases de datos [102] hasta el reconocimiento facial en 3D a alta frecuencia [103]. Aunque la mayoría de las aplicaciones están orientadas a imágenes [104], hay muchos modelos que utilizan otras fuentes de datos. Por ejemplo, Wang *et al.* [105] proponen un enfoque basado en ondoletas para la predicción probabilística de la energía eólica. Otros ejemplos incluyen la clasificación de objetos 3D a partir de nubes de puntos [106], el reconocimiento de enfermedades de las plantas [107], y reconocimiento de la actividad humana en ambientes de interiores [108].

B. Redes Neuronales Recurrentes

El reconocimiento de voz es la aplicación arquetípica para los modelos basados en las RNN, donde los resultados experimentales muestran una reducción de la tasa de error de reconocimiento en conjuntos de datos de dominio público [109], [110]. Recientemente, la Corte Suprema de Brasil ha estado utilizando un modelo LSTM bidireccional para clasificar los casos de demandas a partir de documentos escaneados, con niveles muy variados de calidad de imagen [111]. Existen varias extensiones a RNN. Por ejemplo, las *Redes Inside-Outside* (ION, Inside-Outside Networks) [112] son RNN espaciales para la detección de objetos de contexto. Las ION se han utilizado en aplicaciones que incluyen subtitulado denso en video [113], métodos de diagnóstico asistidos por computadora [114] y detección de rostros [115]. Otras aplicaciones para las RNN incluyen compresión de imágenes [116], reidentificación biométrica de personas [117], predicción de incendios [118], modelado acústico [119], predicción de trayectoria [120], análisis de correlación [121], incrustación de oraciones [122] [123], y clasificación de textos cortos [124].

C. Redes Neuronales Recursivas

Socher *et al.* [125] introdujeron un modelo RvNN para la detección de sentimientos en espacios semánticos de palabras, una tarea que exige capacitación supervisada, recursos de evaluación y potentes modelos de composición. Por su parte, Chandra *et al.* [126] ampliaron esta investigación introduciendo un modelo para predecir el ciber-acoso en Twitter. En el modelo propuesto se demuestra la identificación de texto en tiempo real, en datos estructurados y no estructurados en línea. Otras aplicaciones incluyen el trabajo de Dinh *et al.* [127], quienes introdujeron un método para la detección de peatones.

D. Redes Generativas Profundas

Cuando los investigadores comenzaron a aplicar DGN en la detección de objetos [128] se abrieron oportunidades para los modelos que requieren pocas muestras de entrenamiento, con la capacidad de incorporar información previa. Posteriormente, otras variantes generativas mostraron un potencial creciente. Por ejemplo, el modelo GAN de Goodfellow *et al.* [87] ha demostrado ser efectivo en la generación de imágenes sintéticas que los humanos no distinguen de las reales [129]. La efectividad de GAN incluso ha llevado a los investigadores a estudiar su aplicación en ciberseguridad [130]. Otros usos de GAN incluyen la generación de imágenes para entrenar robots para captar objetos [131], descubrir fármacos [132], adaptación de dominio [133], y aplicaciones de GAN en redes inalámbricas y móviles [134].

Entre otras aplicaciones, los investigadores han usado las DBN para extraer características fonéticas [135], predecir la emoción humana [136] y clasificar tumores cerebrales [137]. Por su lado, las DBM se han aplicado con éxito para modelar patrones de compra del consumidor [138], modelar rostros [139], clasificar el área minera [140] y estimar la presión arterial [141].

Las aplicaciones de VAE se centran principalmente en problemas de datos faltantes. Por ejemplo, McCoy *et al.* [142] utilizaron las VAE para la imputación de datos faltantes en imágenes, comparando un conjunto de imágenes provenientes de circuitos de fresado simulado, que incluye perturbaciones

de proceso, ruido de medición y control de retroalimentación. Además, Marivate *et al.* [143] utilizaron VAE para estimar los datos faltantes en bases de datos del Virus de Deficiencia Humana (VIH). Otras aplicaciones incluyen el monitoreo de procesos [144], la detección de anomalías [145], [146], la detección de intrusiones [147], la predicción de sentimientos [148], la representación de documentos de texto [149], la manipulación de atributos faciales [150], la identificación de anomalías cerebrales [151] y la detección de movimiento lineal [152].

V. Conclusión

El aprendizaje profundo es un vibrante subcampo del aprendizaje automático, cuyos desarrollos actuales lo convierten en un tema de discusión esencial en educación, investigación y desarrollo. Fundado en la piedra angular de la aproximación de funciones universales y el aprendizaje de parámetros a través de la retro-propagación, el poder del aprendizaje profundo es su capacidad para extraer características cada vez más abstractas a partir únicamente de los datos. Aunque continuamente surgen nuevas ideas, algunas redes neuronales profundas, como CNN, RNN y GAN, se han consolidado y están listas para su uso en aplicaciones.

Algunos de los desafíos en el campo del aprendizaje profundo incluyen: 1. La comprensión del espacio de parámetros para mejorar las técnicas de optimización. Esta tarea es relevante dado que una red de aprendizaje profundo puede tener millones de parámetros, por lo que encontrar un valor adecuado para ellos es una tarea difícil, especialmente al considerar su co-dependencia. 2. La escasez de datos. Actualmente, el aprendizaje profundo se basa en grandes conjuntos de datos y el desempeño de los modelos decrece con su tamaño. Algunas alternativas incluyen la aumentación de datos, la transferencia de aprendizaje, así como la generación de datos sintéticos. Sin embargo, se requieren investigaciones para buscar mejores soluciones. 3. Retro-propagación y memoria. Con respecto al primer concepto, se requiere el desarrollo de alternativas a la retro-propagación para generar estrategias de aprendizaje biológicamente plausibles. Con respecto al segundo concepto, se necesita desarrollar mecanismos para la introducción de conocimiento previo y memoria a largo plazo, para minimizar el tiempo de entrenamiento y el error de generalización. En conjunto, esto permitiría mejorar el aprendizaje en las redes neuronales profundas. 4. Aprendizaje autosupervisado. El objetivo de reducir la dependencia en grandes conjuntos de datos etiquetados ha motivado la adopción de enfoques con arquitecturas de redes profundas que establecen sus propios objetivos, refinando su desempeño mediante auto-corrección. 5. Aprendizaje por refuerzo. La inspiración extraída de la neurociencia cognitiva y la psicología del desarrollo para descifrar el patrón de comportamiento humano, se replica para construir modelos capaces de aprender tareas mediante el establecimiento de políticas de acción basadas en premios y castigos, al tiempo que se modifica el entorno.

Este documento sugiere que el aprendizaje profundo es un campo amplio y dinámico, donde se han logrado desarrollos impresionantes. Nuestro artículo también muestra que el aprendizaje profundo es una disciplina fértil, llena de ideas, nuevas, emocionantes, retos y oportunidades.

Referencias

[1] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine will Remake our World*. Basic Books, 2015.

[2] J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press, 1992.

[3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, 2014.

[4] S. Muggleton and L. De Raedt, "Inductive Logic Programming: Theory and Methods," *The Journal of Logic Programming*, vol. 19, pp. 629–679, 1994.

[5] D. Hofstadter, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic books, 1995.

[6] D. Rumelhart, G. Hinton, and R. Williams, "Learning Internal Representations by Error Propagation," California University, Tech. Rep., 1985.

[7] A. Yala, C. Lehman, T. Schuster, T. Portnoi, and R. Barzilay, "A Deep Learning Mammography-based Model for Improved Breast Cancer Risk Prediction," *Radiology*, p. 182716, 2019.

[8] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, and T. Graepel, "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[9] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[10] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning based Recommender System: A Survey and New Perspectives," *ACM CSUR*, vol. 52, no. 1, p. 5, 2019.

[11] P. Univaso, J. Ale, and J. Gurlekian, "Data Mining applied to Forensic Speaker Identification," *IEEE Latin America Transactions*, vol. 13, no. 4, pp. 1098–1111, April 2015.

[12] J. Jimenez, I. Gonzalez, and J. Lopez, "Challenges And Opportunities In Analytic-Predictive Environments Of Big Data And Natural Language Processing For Social Network Rating Systems," *IEEE Latin America Transactions*, vol. 16, no. 2, pp. 592–597, Feb 2018.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," in *CVPR*, 2014.

[14] Min-Ling Zhang and Zhi-Hua Zhou, "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, Oct 2006.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[17] A. Shaikhha, A. Fitzgibbon, D. Vytiniotis, and S. Peyton Jones, "Efficient differentiable programming in a functional array-processing language," *Proceedings of the ACM on Programming Languages*, vol. 3, no. ICFP, p. 97, 2019.

[18] A. Karpathy, "Software 2.0," 2017.

[19] H. Landahl, W. Mcculloch, and W. Pitts, "A Statistical Consequence of the Logical Calculus of Nervous Nets," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, p. 135–137, 1943.

[20] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, vol. 65, no. 6, p. 386–408, 1958.

[21] P. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Ph.D. dissertation, Harvard University, 1974.

[22] K. Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics*, vol. 36, no. 4, p. 193–202, 1980.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, p. 2278–2324, 1998.

[24] M. Jordan, "Serial Order: A Parallel Distributed Processing Approach," in *Advances in Psychology*. Elsevier, 1997, vol. 121, pp. 471–495.

[25] G. Hinton, S. Osindero, and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, p. 1527–1554, 2006.

[26] P. Smolensky, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," D. Rumelhart and J. McClelland, Eds. MIT Press, 1986, ch. Information Processing in Dynamical Systems: Foundations of Harmony Theory, pp. 194–281.

[27] H. Mhaskar and T. Poggio, "Function Approximation by Deep Networks," *arXiv:1905.12882*, 2019.

[28] P. Baldi and R. Vershynin, "The Capacity of Feed Forward Neural Networks," *Neural Networks*, vol. 116, pp. 288–311, 2019.

[29] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[30] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *International Conference on Machine Learning*, 2010, pp. 807–814.

[31] S. Roy, S. Manna, S. Dubey, and B. Chaudhuri, "LiSHT: Non-Parametric Linearly Scaled Hyperbolic Tangent Activation Function for Neural Networks," *arXiv:1901.05894*, 2019.

[32] J. Su, D. Vargas, and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks," *IEEE Transactions on Evolutionary Computation*, 2019.

[33] H. Li, Y. Tian, K. Mueller, and X. Chen, "Beyond Saliency: Understanding Convolutional Neural Networks from Saliency Prediction on Layer-Wise Relevance Propagation," *Image and Vision Computing*, vol. 83, pp. 70–86, 2019.

[34] M. Alber, S. Lapuschkin, P. Seegerer, M. Hägele, K. Schütt, G. Montavon, W. Samek, K. Müller, S. Dähne, and P. Kindermans, "iNNvestigate neural networks!" *Journal of Machine Learning Research*, vol. 20, no. 93, pp. 1–8, 2019.

[35] M. Kahng, N. Thorat, D. Chau, F. Viégas, and M. Wattenberg, "GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1–11, 2018.

[36] S. Yu and J. Principe, "Understanding Autoencoders with Information Theoretic Concepts," *Neural Networks*, vol. 117, pp. 104–123, 2019.

[37] N. Cristianini and J. Shawe, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[38] C. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[39] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss Functions for Neural Networks for Image Processing," *arXiv:1511.08861*, 2015.

[40] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[41] H. Yao, S. Zhang, R. Hong, Y. Zhang, C. Xu, and Q. Tian, "Deep Representation Learning with Part Loss for Person Re-Identification," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2860–2871, 2019.

[42] I. Ward, M. Jalwana, and M. Bennamoun, "Improving Image-Based Localization with Deep Learning: The Impact of the Loss Function," *arXiv:1905.03692*, 2019.

[43] T. Do, T. Tran, I. Reid, V. Kumar, T. Hoang, and G. Carneiro, "A Theoretically Sound Upper Bound on the Triplet Loss for Improving the Efficiency of Deep Distance Metric Learning," in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 404–10 413.

[44] A. Baydin, B. Pearlmutter, A. Radul, and J. Siskind, "Automatic Differentiation in Machine Learning: A Survey," *Journal of Machine Learning Research*, vol. 18, no. 153, 2018.

[45] W. Maass, "Networks of Spiking Neurons: The Third Generation of Neural Network Models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[46] J. Thiele, O. Bichler, and A. Dupret, "SpikeGrad: An ANN-equivalent Computation Model for Implementing Backpropagation with Spikes," *arXiv:1906.00851*, 2019.

[47] C. Lee, S. Sarwar, and K. Roy, "Enabling Spike-based Backpropagation in State-of-the-Art Deep Neural Network Architectures," *arXiv:1903.06379*, 2019.

[48] Y. Pu and J. Wang, "Fractional-Order Backpropagation Neural Networks: Modified Fractional-order Steepest Descent Method for Family of Backpropagation Neural Networks," *arXiv:1906.09524*, 2019.

[49] T. Lillicrap and A. Santoro, "Backpropagation through Time and the Brain," *Current Opinion in Neurobiology*, vol. 55, pp. 82–89, 2019.

[50] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "Biologically Inspired Alternatives to Backpropagation Through Time for Learning in Recurrent Neural Nets," *arXiv:1901.09049*, 2019.

[51] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *arXiv:1609.04747*, 2016.

[52] G. Hacohen and D. Weinshall, "On the Power of Curriculum Learning in Training Deep Networks," *arXiv:1904.03626*, 2019.

[53] P. Netrapalli, "Stochastic Gradient Descent and its Variants in Machine Learning," *Journal of the Indian Institute of Science*, pp. 1–13, 2019.

[54] N. Qian, "On the Momentum Term in Gradient Descent Learning Algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[55] Y. Nesterov, "A Method for Unconstrained Convex Minimization Problem with the Rate of Convergence $O(1/k^2)$," in *Doklady AN USSR*, vol. 269, 1983, pp. 543–547.

[56] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[57] M. Zeiler, "AdaDelta: An Adaptive Learning Rate Method," *arXiv:1212.5701*, 2012.

[58] G. Hinton, "Neural Networks for Machine Learning." [Online]. Available: https://www.coursera.org/learn/neural-networks-deep-learning

[59] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, 2014.

[60] T. Dozat, "Incorporating Nesterov Momentum into Adam," 2016.

[61] S. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," *arXiv:1904.09237*, 2019.

[62] M. Zhang, J. Lucas, G. Hinton, and J. Ba, "Lookahead Optimizer: k Steps Forward, 1 Step Back," *arXiv:1907.08610*, 2019.

[63] A. Ng, "Feature Selection, L_1 vs. L_2 Regularization, and Rotational Invariance," in *International Conference on Machine Learning*. ACM, 2004, p. 78.

[64] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[65] M. Li, M. Soltanolkotabi, and S. Oymak, "Gradient Descent with Early Stopping is Provably Robust to Label Noise for Overparameterized Neural Networks," *arXiv:1903.11680*, 2019.

[66] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[67] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *IJCAI*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[68] G. Afendras and M. Markatou, "Optimality of Training/Test Size and Resampling Effectiveness in Cross-Validation," *Journal of Statistical Planning and Inference*, vol. 199, pp. 286–301, 2019.

[69] L. Jing and Y. Tian, "Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey," *arXiv:1902.06162*, 2019.

[70] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[71] M. Puterman, *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[72] V. François-Lavet, P. Henderson, R. Islam, M. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," *Foundations and Trends in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[73] Q. Le, "Building High-Level Features using Large Scale Unsupervised Learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8595–8598.

[74] M. Choy, D. Srinivasan, and R. Cheu, "Neural Networks for Continuous Online Learning and Control," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1511–1531, Nov 2006.

[75] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the Importance of Initialization and Momentum in Deep Learning," in *International Conference on Machine Learning*. JMLR, 2013, pp. 1139–1147.

[76] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large Scale Distributed Deep Networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1223–1231. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999271

[77] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng, "Multimodal Deep Learning," in *International Conference on International Conference on Machine Learning*, 2011, pp. 689–696.

[78] D. Hubel and T. Wiesel, "Receptive Fields, Binocular Interaction and Functional Architecture in the Cats Visual Cortex," *The Journal of Physiology*, vol. 160, no. 1, p. 106–154, 1962.

[79] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks," Apr 2014.

[80] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *International Conference on Artificial Intelligence and Statistics*, 2010.
[81] X. Li and X. Wu, "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition," *CoRR*, vol. 1410.4281, 2014.
[82] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv:1406.1078*, 2014.
[83] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, jun 2016, pp. 770–778.
[84] C. Goller and A. Kuchler, "Learning Task-dependent Distributed Representations by Backpropagation through Structure," in *International Conference on Neural Networks*, vol. 1, 1996, pp. 347–352.
[85] G. Hinton, "Deep Belief Networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.
[86] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.
[87] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Conference on Neural Information Processing Systems*, 2014, pp. 2672–2680.
[88] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes," p. arXiv:1312.6114, Dec 2013.
[89] D. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised Learning with Deep Generative Models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589. [Online]. Available: http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf
[90] G. Jeong and H. Kim, "Improving Financial Trading Decisions using Deep Q-Learning: Predicting the Number of Shares, Action Strategies, and Transfer Learning," *Expert Systems with Applications*, vol. 117, pp. 125 – 138, 2019.
[91] S. Khan and T. Yairi, "A Review on the Application of Deep Learning in System Health Management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241 – 265, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327017306064
[92] E. Cust, A. Sweeting, K. Ball, and S. Robertson, "Machine and Deep Learning for Sport-Specific Movement Recognition: A Systematic Review of Model Development and Performance," *Journal of Sports Sciences*, vol. 37, no. 5, pp. 568–600, 2019.
[93] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, "A Systematic Review of Deep Learning Approaches to Educational Data Mining," *Complexity*, vol. 2019, 2019.
[94] Y. Ayturan, Z. Ayturan, and H. Altun, "Air Pollution Modelling with Deep Learning: A Review," *International Journal of Environmental Pollution and Environmental Modelling*, vol. 1, no. 3, pp. 58–62, 2018.
[95] L. Aristodemou and F. Tietze, "The State-of-the-Art on Intellectual Property Analytics (IPA): A Literature Review on Artificial Intelligence, Machine Learning and Deep Learning Methods for Analysing Intellectual Property (IP) Data," *World Patent Information*, vol. 55, pp. 37–51, 2018.
[96] H. Wang, S. Shang, L. Long, R. Hu, Y. Wu, N. Chen, S. Zhang, F. Cong, and S. Lin, "Biological Image Analysis using Deep Learning-based Methods: Literature Review," *Digital Medicine*, vol. 4, no. 4, p. 157, 2018.
[97] G. Grekousis, "Artificial Neural Networks and Deep Learning in Urban Geography: A Systematic Review and Meta-Analysis," *Computers, Environment and Urban Systems*, 2018.
[98] M. Borg, C. Englund, K. Wnuk, B. Duran, C. Levandowski, S. Gao, Y. Tan, H. Kaijser, H. Lönn, and J. Törnqvist, "Safety Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry," *arXiv:1812.05389*, 2018.
[99] K. Gopalakrishnan, "Deep Learning in Data-driven Pavement Image Analysis and Automated Distress Detection: A Review," *Data*, vol. 3, no. 3, p. 28, 2018.
[100] Y. Li, Q. Lei, C. Cheng, G. Zhang, W. Wang, and Z. Xu, "A Review: Machine Learning on Robotic Grasping," in *International Conference on Machine Vision*, vol. 11041, 2019.
[101] P. Andersen, M. Goodwin, and O. Granmo, "Deep RTS: A Game Environment for Deep Reinforcement Learning in Real-Time Strategy Games," in *Conference on Computational Intelligence and Games*, 2018, pp. 1–8.
[102] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-Scale Long-Tailed Recognition in an Open World," in *Computer Vision and Pattern Recognition*, 2019.
[103] Y. Zhou, J. Deng, I. Kotsia, and S. Zafeiriou, "Dense 3D Face Decoding Over 2500FPS: Joint Texture and Shape Convolutional Mesh Decoders," in *Computer Vision and Pattern Recognition*, 2019.
[104] R. Singh, A. Mittal, and R. Bhatia, "3D Convolutional Neural Network for Object Recognition: A Review," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 15 951–15 995, 2019.
[105] H. Wang, G. Li, G. Wang, J. Peng, H. Jiang, and Y. Liu, "Deep Learning based Ensemble approach for Probabilistic Wind Power Forecasting," *Applied Energy*, vol. 188, pp. 56 – 70, 2017.
[106] H. Lei, N. Akhtar, and A. Mian, "Octree Guided CNN With Spherical Kernels for 3D Point Clouds," in *Computer Vision and Pattern Recognition*, 2019.
[107] A. da Silva, A. de Almeida, and F. Vidal, "Plant Diseases Recognition from Digital Images using Multichannel Convolutional Neural Networks," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019, pp. 450–458.
[108] B. Vandersmissen, N. Knudde, A. Jalalvand, I. Couckuyt, T. Dhaene, and W. De Neve, "Indoor Human Activity Recognition using High-Dimensional Sensors and Deep Neural Networks," *Neural Computing and Applications*, Aug 2019.
[109] T. Yang, T. Tseng, and C. Chen, "Recurrent Neural Network-based Language Models with Variation in Net Topology, Language, and Granularity," in *International Conference on Asian Language Processing*, Nov 2016, pp. 71–74.
[110] X. Chen, X. Liu, Y. Wang, A. Ragni, J. Wong, and M. Gales, "Exploiting Future Word Contexts in Neural Network Language Models for Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1444–1454, 2019.
[111] F. Braz, N. Silva, T. Emidio, F. Borges, M. Ferreira, P. Inazawa, V. Coelho, P. Sukiennik, P. Goncalves, F. Vidal, D. Alves, D. Gusmao, G. Ziegler, R. Fernandes, R. Zumblick, and F. Peixoto, "Document Classification using a Bi-LSTM to Unclog Brazil's Supreme Court," in *NIPS Workshop on Machine Learning for the Developing World*, 2018.
[112] S. Bell, C. Zitnick, K. Bala, and R. Girshick, "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks," in *Computer Vision and Pattern Recognition*, 2016, pp. 2874–2883.
[113] L. Yang, K. Tang, J. Yang, and L. Li, "Dense Captioning with Joint Inference and Visual Context," in *Computer Vision and Pattern Recognition*, 2017, pp. 1978–1987.
[114] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang, "MDNet: A Semantically and Visually Interpretable Medical Image Diagnosis Network," in *Computer Vision and Pattern Recognition*, 2017, pp. 3549–3557.
[115] Y. Bai and B. Ghanem, "Multi-scale Fully Convolutional Network for Face Detection in the Wild," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 2078–2087.
[116] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Image Compressed Sensing using Convolutional Neural Network," *IEEE Transactions on Image Processing*, pp. 1–1, 2019.
[117] S. Zhai, S. Liu, X. Wang, and J. Tang, "FMT: Fusing Multi-task Convolutional Neural Network for Person Search," *Multimedia Tools and Applications*, Jul 2019.
[118] J. Hodges, B. Lattimer, and K. Luxbacher, "Compartment Fire Predictions using Transpose Convolutional Neural Networks," *Fire Safety Journal*, p. 102854, 2019.
[119] T. Zia and U. Zahid, "Long Short-Term Memory Recurrent Neural Network Architectures for Urdu Acoustic Modeling," *International Journal of Speech Technology*, vol. 22, no. 1, pp. 21–30, 2019.
[120] D. Duives, G. Wang, and J. Kim, "Forecasting Pedestrian Movements Using Recurrent Neural Networks: An Application of Crowd Monitoring Data," *Sensors*, vol. 19, p. 382, 2019.
[121] Y. Yu, S. Tang, F. Raposo, and L. Chen, "Deep Cross-Modal Correlation Learning for Audio and Lyrics in Music Retrieval," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 1, pp. 20:1–20:16, 2019.
[122] J. Maillard, S. Clark, and D. Yogatama, "Jointly Learning Sentence Embeddings and Syntax with Unsupervised Tree-LSTMs," *Natural Language Engineering*, vol. 25, no. 4, p. 433–449, 2019.
[123] J. Kang, H. Choi, and H. Lee, "Deep Recurrent Convolutional Networks for Inferring user Interests from Social Media," *Journal of Intelligent Information Systems*, vol. 52, no. 1, pp. 191–209, Feb 2019.
[124] J. Xu and Y. Cai, "Incorporating Context-Relevant Knowledge into Convolutional Neural Networks for Short Text Classification," *AAAI*

Conference on Artificial Intelligence, vol. 33, no. 01, pp. 10 067–10 068, Jul. 2019.

[125] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank," *EMNLP*, vol. 1631, pp. 1631–1642, 01 2013.

[126] N. Chandra, S. Khatri, and S. Som, "Cyberbullying Detection using Recursive Neural Network through Offline Repository," in *International Conference on Reliability, Infocom Technologies and Optimization*, Aug 2018, pp. 748–754.

[127] T. Dinh, N. Vinh, and J. Wook, "Robust Pedestrian Detection via a Recursive Convolution Neural Network," in *International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2018, pp. 281–286.

[128] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59 – 70, 2007.

[129] N. Caporusso, K. Zhang, G. Carlson, D. Jachetta, D. Patchin, S. Romeiser, N. Vaughn, and A. Walters, "User Discrimination of Content Produced by Generative Adversarial Networks," in *Human Interaction and Emerging Technologies*, T. Ahram, R. Taiar, S. Colson, and A. Choplin, Eds., Cham, 2020, pp. 725–730.

[130] C. Yinka and O. Ugot, "A Review of Generative Adversarial Networks and its Application in Cybersecurity," *Artificial Intelligence Review*, 2019.

[131] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-To-Real via Sim-To-Sim: Data-Efficient Robotic Grasping via Randomized-To-Canonical Adaptation Networks," in *IEEE Computer Vision and Pattern Recognition*, 2019.

[132] X. Yang, Y. Wang, R. Byrne, G. Schneider, and S. Yang, "Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery," *Chemical Reviews*, 2019.

[133] C. Ruan, W. Wang, H. Hu, and D. Chen, "Category-Level Adversaries for Semantic Domain Adaptation," *IEEE Access*, vol. 7, pp. 83 198–83 208, 2019.

[134] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

[135] Y. Seddiq, Y. Alotaibi, S. Selouani, and A. Meftah, "Distinctive Phonetic Features Modeling and Extraction using Deep Neural Networks," *IEEE Access*, vol. 7, pp. 81 382–81 396, 2019.

[136] M. Hassan, G. Alam, Z. Uddin, S. Huda, A. Almogren, and G. Fortino, "Human Emotion Recognition using Deep Belief Network Architecture," *Information Fusion*, vol. 51, pp. 10 – 18, 2019.

[137] A. Kharrat and M. Néji, "Classification of Brain Tumors using Personalized Deep Belief Networks on MRImages: PDBN-MRI," in *International Conference on Machine Vision*, vol. 11041, 2019.

[138] F. Xia, R. Chatterjee, and J. May, "Using Conditional Restricted Boltzmann Machines to Model Complex Consumer Shopping Patterns," *Marketing Science*, vol. 38, no. 4, pp. 711–727, 2019.

[139] C. Duong, K. Luu, K. Quach, and T. Bui, "Deep Appearance Models: A Deep Boltzmann Machine Approach for Face Modeling," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 437–455, 2019.

[140] K. Tan, F. Wu, Q. Du, P. Du, and Y. Chen, "A Parallel Gaussian–Bernoulli Restricted Boltzmann Machine for Mining Area Classification With Hyperspectral Imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 2, pp. 627–636, 2019.

[141] S. Lee and J. Chang, "Dempster–Shafer Fusion Based on a Deep Boltzmann Machine for Blood Pressure Estimation," *Applied Sciences*, vol. 9, no. 1, p. 96, 2019.

[142] J. McCoy, S. Kroon, and L. Auret, "Variational Autoencoders for Missing Data Imputation with Application to a Simulated Milling Circuit," *IFAC*, vol. 51, no. 21, pp. 141 – 146, 2018.

[143] N. Vukosi, V. Fulufhelo, and T. Marwala, "Investigation into the use of Autoencoder Neural Networks, Principal Component Analysis and Support Vector Regression in Estimating Missing HIV Data," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 682 – 689, 2008.

[144] S. Lee, M. Kwak, K. Tsui, and S. Kim, "Process Monitoring using Variational Autoencoder for High-Dimensional Nonlinear Processes," *Engineering Applications of Artificial Intelligence*, vol. 83, pp. 13 – 27, 2019.

[145] M. Nicolau and J. McDermott, "Learning Neural Representations for Network Anomaly Detection," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3074–3087, 2018.

[146] T. Wang, M. Qiao, Z. Lin, C. Li, H. Snoussi, Z. Liu, and C. Choi, "Generative Neural Networks for Anomaly Detection in Crowded Scenes," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1390–1399, 2018.

[147] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network," *Sensors*, vol. 19, no. 11, p. 2528, 2019.

[148] C. Wu, F. Wu, S. Wu, Z. Yuan, J. Liu, and Y. Huang, "Semi-Supervised Dimensional Sentiment Analysis with Variational Autoencoder," *Knowledge-Based Systems*, vol. 165, pp. 30–39, 2019.

[149] S. Wang, J. Cai, Q. Lin, and W. Guo, "An Overview of Unsupervised Deep Feature Representation for Text Categorization," *IEEE Transactions on Computational Social Systems*, 2019.

[150] X. Hou, K. Sun, L. Shen, and G. Qiu, "Improving Variational Autoencoder with Deep Feature Consistent and Generative Adversarial Training," *Neurocomputing*, vol. 341, pp. 183–194, 2019.

[151] H. Choi, S. Ha, H. Kang, H. Lee, and D. S. Lee, "Deep Learning Only by Normal Brain PET Identify Unheralded Brain Anomalies," *EBioMedicine*, vol. 43, pp. 447–453, 2019.

[152] M. S. Kim, J. P. Yun, S. Lee, and P. Park, "Unsupervised Anomaly Detection of LM Guide Using Variational Autoencoder," in *International Symposium on Advanced Topics in Electrical Engineering*, 2019, pp. 1–5.

Deep Learning: Current State

J. Salas, *Member, IEEE*, F. Vidal, and J. Martínez-Trinidad

***Abstract*—Deep learning, a derived from machine learning, has grown into widespread usage with applications as diverse as cancer detection, elephant spotting, and game development. The number of published studies shows an increasing interest by researchers because of its demonstrated ability to achieve high performance in the solution of complex problems, the wide availability of data and computing resources, and the groundbreaking development of effective algorithms. This paper reviews the current state of deep learning. It includes a revision of basic concepts, such as the operations of feed forward and backpropagation, the use of convolution to extract features, the role of the loss function, and the optimization and learning processes; the survey of main stream techniques, in particular convolutional, recurrent, recursive, deep belief, deep generative, generative adversarial, and variational auto-enconder neural networks; the description of an ample array of applications organized by the type of technique employed; and the discussion of some of its most intriguing open problems.**

***Index Terms*—Applications of Deep Learning, Convolutional Neural Networks, Deep Generative Networks, Recursive Neural Networks, Recurrent Neural Networks.**

I. Introdução

EM Domingos [1] são apresentados os principais esforços para o melhor desenvolvimento de uma inteligência artificial com característica evolucionária [2], bayesiana [3], simbolista [4], analizadora [5] e conexionista [6]. Nesse contexto, uma importante aplicação da inteligência artificial é o aprendizado de máquina, se tornando cada vez mais popular e sendo atualmente empregado em aplicações como detecção precoce do câncer [7], jogos eletrônicos [8], detecção de objetos [9], recomendação de vídeo por contexto [10], reconhecimento de fala [11], análise de redes sociais [12], reconhecimento de ação em vídeos [13] e mineração de texto [14]. Entre as técnicas de aprendizado de máquina, o aprendizado profundo, (uma abordagem conexionista) destaca-se como uma das de maior relevância, pois permite a extração automática de características para sua utilização. Lecun *et al.* [15] e Goodfellow *et al.* [16] definem como aprendizagem profunda o conjunto de métodos com múltiplos níveis de aprendizagem, obtidos pela composição a partir de módulos simples, não lineares, que alteram progressivamente a representação dos dados de entrada em sua forma bruta, para um nível abstrato. É interessante notar, que alguns pesquisadores definiram que os métodos de aprendizado profundo possuem uma semelhança maior com a programação de computadores do que com a neurobiologia e, sugeriram renomeá-lo como *programação diferenciável* (do inglês *differentiable programming* [17] ou *Software 2.0* [18].

As primeiras ideias sobre aprendizado profundo poderiam ser atribuídas a Aristóteles, que no ano 300 A.C., propôs o associativismo para descrever o caminho para o entendimento do funcionamento do cérebro humano. No entanto, foi em 1943 que McCulloch e Pitts [19] introduziram o primeiro modelo de rede neural artificial imitando a função do neocórtex do cérebro humano, quando as técnicas de aprendizado profundo começaram a evoluir. Então, Rosenblatt [20] deu outro passo significativo na implementação da Teoria *hebbiana* no primeiro dispositivo eletrônico chamado *Perceptron* [20]. Posteriormente, Werbos [21] introduziu o processo de treinamento de redes neurais artificiais utilizando a retro-propagação do erro. Em 1980, Fukushima introduziu o *Neocognitron* [22], que por sua vez inspirou o desenvolvimento das redes neurais convolucionais (do inglês *Convolutional Neural Network* - CNN) [23] e das redes neurais Recorrentes (do inglês *Recurrent Neural Network* - RNN) [24]. Então, em 1998, Lecun *et al.* [23] implementou a *LeNet*, a primeira rede neural profunda. Em 2006, as Redes de Crenças Profundas (do inglês *Deep Belief Network* - DBN) [25], juntamente com o pré-treinamento em camadas, levaram a um amplo uso do aprendizado profundo. A ideia por trás dessas redes consistia em treinar um modelo simples não supervisionado de duas camadas, *e.g. Máquinas Restritas de Boltzmann* (do inglês *Restricted Boltzmann Machine* - RBM) [26], em que os parâmetros são fixos, uma nova camada na parte superior é adicionada e apenas os parâmetros para a nova camada são treinados. Essa técnica permitiu aumentar o número de camadas que, até aquele momento, haviam sido treinadas.

Este artigo tem por objetivo descrever o estado atual da ciência, técnicas e aplicações utilizando aprendizado profundo. Também estão incluídos os conceitos chaves básicos e a discussão de problemas ainda em aberto, apresentando uma visão geral desta área de pesquisa. O manuscrito é organizado como: Na Seção II são revisados os principais conceitos de aprendizado profundo, incluindo técnicas de convoluções como extratores de características, funções de perda melhorando o processo de otimização e a retro-propagação como otimizador. Então, na Seção III são visitados de forma geral algumas redes neurais profundas (do inglês *Deep Neural Network* - DNN), incluindo as CNN, RNN e as redes generativas profundas (do inglês *Deep Generative Network* - DGN). Na Seção IV são apresentadas diferentes aplicações recentes de aprendizado profundo. Finalmente, este artigo é concluído discutindo os desafios e as direções futuras das pesquisas na área.

II. Conceitos Básicos

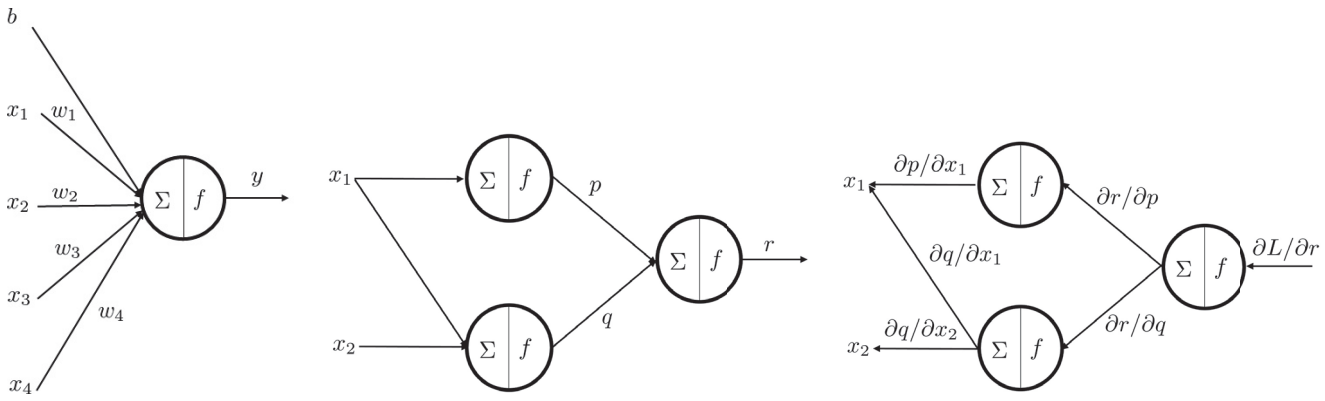
Existem algumas controvérsias sobre se as DNNs são melhores aproximadores de função, do que as redes neurais

Joaquín Salas was partially supported by IPN under grant SIP2019.
Send your correspondence to Joaquín Salas, Cerro Blanco 141, Colinas del Cimatario, Querétaro, México, 76090. salas@ieee.org.

Joaquín Salas is with the Instituto Politécnico Nacional, México.

Flavio B. Vidal is with the University of Brasilia, Brazil.

José Fco. Martínez-Trinidad is with the Instituto Nacional de Astrofísica, Óptica y Electrónica, México.



(a) Neurona artificial (b) *feedforward* (c) *retro-propagação*

Fig. 1. *Etapa de Estimação.* Em cada neurônio artificial (a) calcula-se a posição em um plano com os parâmetros definidos como pesos $\mathbf{w}^T = (w_1, \ldots, w_n)$ e desvios b das entradas $\mathbf{x}^T = (x_1, \ldots, x_n)$ a partir da operação $z = \mathbf{w}^T\mathbf{x} + b$. O valor de z é projetado em um espaço não-linear a partir da operação $y = f(z)$. Este processo é realizado para todos os neurônios em todas as camadas da rede. Durante a etapa de alimentação direta (do inglês *feedforward*) é obtido o valor da função para uma determinada entrada (a). No treinamento, a retro-propagação (do inglês, *backpropagation*) permite calcular o valor do gradiende da função de custo das entradas, a partir do uso da Regra da Cadeia.

superficiais (redes neurais artificiais com somente uma camada oculta), com alguns pesquisadores preferindo a primeira [27], e outros, a segunda [28]. De qualquer forma, existe um consenso sobre a natureza inovadora das DNNs. Geralmente, organiza-se as DNNs como fluxos de camadas sequenciais, onde a composição das funções ocorre. Se o fluxo tiver ciclos, temos uma RNN [29]. Caso contrário, teremos uma rede neural *perceptron* multicamada que inclui uma entrada, pelo menos uma camada oculta e uma saída. Nesta Seção, são revisados alguns conceitos sobre as DNNs, enfatizando o atual estado da arte.

A. Etapa de Estimação

O termo *profundo* (do inglês *deep*) no acrônimo das DNNs é devido à sua capacidade de aprender as características dos dados hierarquicamente, e não devido ao seu número de camadas [15]. Em uma DNN, na medida que avançamos no fluxo de dados, camada após camada, obtém-se as características mais gerais. Em cada estágio, para cada neurônio, calcula-se uma soma ponderada das entradas. Em seguida, projeta-se o resultado em um espaço não-linear usando uma função de ativação (ver Figuras 1(a)-(b)). Logo no seu início, os pesquisadores utilizaram funções sigmoides (como a logística e a tangente hiperbólica) como funções de ativação, uma vez que são funções contínuas, deriváveis e, portanto, adequadas para a retro-propagação. No entanto, tanto para valores positivos quanto negativos, sua derivada desaparece amortecida pela aprendizagem. Como resposta a esse problema, Nair e Hinton [30] propuseram a função de ativação por unidade linear retificada (do inglês *Rectified Linear Unit* - ReLU), que produz e zera seu argumento para valores positivos e negativos, respectivamente. Na literatura, houve um esforço considerável com o objetivo de reduzir os efeitos das entradas negativas da ReLU para quando os valores dos gradientes tendem a zero [31], [32].

B. Extração de Características

A inclusão de convoluções para extrair características foi a inovação apresentada nas DNNs. Antes das DNNs, era preciso recursos de engenharia consideráveis para o processo de aprendizagem. Uma convolução corresponde à resposta linear de um sinal a um operador. Neste caso o que importa é que as DNNs aprendam os operadores durante o treinamento, e a definição das convoluções minimiza o valor da função de perda [15]. Além disso, como as convoluções operam localmente, estas compartilham os parâmetros em todo o domínio da entrada, mas à medida que o fluxo de dados do processo avança, resultam em características mais abstratas.

Apesar do sucesso, existem preocupações com a confiabilidade de uma DNN em operações críticas. Considerando como exemplo o trabalho de Su *et al.* [33], que introduziu o método para gerar perturbações adversas em um pixel, reduzindo substancialmente o desempenho dos classificadores de imagem. Como consequência, os pesquisadores direcionaram esforços consideráveis para aumentar a transparência das DNNs. Esses esforços vão ao longo da criação de: mapas de saliência para facilitar a interpretabilidade [34]; interfaces para entender o funcionamento interno das arquiteturas de aprendizagem profundas [35]; ou ferramentas educacionais para aumentar a conscientização dos usuários [36]. Embora tenha-se avançado em metodologias para entender a dinâmica do aprendizado e do projeto [37], existe ainda amplo espaço para avanços teóricos para analisar as DNNs.

C. Função de Perda

Uma DNN quantifica o conceito de aprendizado como a minimização de um valor de função de perda. Em uma regressão, as funções típicas de perda incluem as normas $\mathcal{L}_1$ (soma dos valores absolutos), $\mathcal{L}_2$ (raiz quadrada da soma dos valores ao quadrado), enquanto que na classificação, as funções de perda mais comuns incluem a função de perda tipo *dobradiça* (que ganhou popularidade com o advento do classificador Máquinas de Vetores de Suporte [38]) e a perda de entropia cruzada (diretamente relacionada à entropia de Shannon para a teoria da informação [39]).

Atualmente, existe o consenso sobre a importância em se definir funções de perda específicas para cada tarefa. Por

exemplo, em Zhao *et al.* [40] é usado o índice de similaridade estrutural [41] para a restauração de imagens de forma perceptiva, Yao *et al.* [42] propõe incorporar parte da perda no processo re-identificação de pessoas e Ward *et al.* [43] introduz uma função de perda para localização baseada em imagens que considera o acoplamento entre rotação e translação para definir a posição da câmera. Além disso, uma área ativa de pesquisa é a definição de funções de perda que promovem a eficiência computacional. Considere o problema do aprendizado por métrica de distância, que visa aprender a incorporação eficiente de características. Nesse problema, Do *et al.* [44] apresentam uma melhoria no cálculo da linearização da função de perda, a partir da introdução de centroides virtuais incorporados, resultando em ganhos na eficiência computacional.

D. Retro-Propagação (Backpropagation)

O grafo de uma DNN que representa o passo sem realimentação é uma representação conveniente para a derivada da função de perda em relação à entrada, a partir da aplicação da *Regra da Cadeia* [45] (ver Figura 1(c)). Assim, uma escolha natural para minimizar a função de perda é o *Gradiente Descendente* sendo o procedimento fundamental para se obter iterativamente os parâmetros ideais da retro-propagação. Utilizando a retro-propagação, calcula-se a nova estimativa dos parâmetros considerando seu valor atual por uma *Regra da Cadeia Ponderada* (a partir da *Taxa de Aprendizagem*) do gradiente.

As DNNs operam em estruturas de dados densas. No entanto, os neurônios do cérebro humano real funcionam em picos de sinais. Esse fato inspirou os pesquisadores a procurarem soluções comparáveis à forma de funcionamento do cérebro [46]. Potencialmente, as arquiteturas exibem alta esparsividade temporal e espacial, resultando em sistemas neuromórficos energeticamente eficientes, onde os picos se comunicam de forma assíncrona. Atualmente, essas soluções fornecem um método para integrar a retro-propagação em uma rede neural pulsada usando um acumulador (neurônios *integrador-e-disparador*, do inglês *Integrate and Fire* - IFs) para propagação dos erros, discretizando os erros em pulsos [47], [48]. Lee *et al.* [49] estendeu a noção de picos para redes neurais profundas. Além disso, para melhorar a velocidade do treinamento, Pu e Wang [50] introduziram uma rede neural de retro-propagação com ordem fracionária, esta treinada com o método de descida mais íngreme.

Para a busca de parâmetros ótimos, durante o treinamento das RNNs, os profissionais usam o modo de *Retro-propagação Através do Tempo* (do inglês *backpropagation-through-time* - BPTT) para aprender ao invés da retro-propagação tradicional [51]. Uma forma possível de entender esta analogia é utilizar uma representação da RNN de forma ”desenrolada” e aplicar a retro-propagação nessa arquitetura ”desenrolada”. Ainda assim, o problema é complexo pois o mesmo neurônio precisa reconstruir o histórico de ativação para resolver um problema de atribuição de *Crédito Temporal*. A pesquisa fundamental da RNN concentra-se em evitar gradientes com valores elevados ou dissipados. Normalmente, isso é obtido a partir do projeto da rede (via Memória de Longo Prazo, do inglês *Long-Short Term Memory* - LSTM, ou Unidades Recorrentes Fechadas do inglês *Gated Recurrent Units*, GRU [16]) ou otimização (utilizando o recorte do gradiente para algum valor limítrofe). De fato, a retro-propagação é biologicamente implausível, isto é, para aprender, seria necessário transmitir sinais de erro para o passado temporal. Bellec *et al.* [52] mesclou as informações disponíveis localmente e mostrou que estas fornecem uma excelente aproximação para o valor do BPTT.

E. Otimização

A função de perda conduz o processo de otimização. Devido à sua simplicidade, a grande massa de dados comumente associada à aprendizagem profunda e a representação do gradiente utilizando a *Regra da Cadeia* definida pelo grafo de computação, levou os usuários desenvolvedores na preferência pelo uso do gradiente descendente para otimização [53]. Na maioria dos casos (talvez exceto quando se aplica o *Curriculum Learning* [54]), estima-se o gradiente de uma amostra aleatória originando a descida do gradiente estocástico [55]. As melhorias que visam ligar as atualizações à inclinação da função de perda incluem a incorporação do momento [56] (para acelerar a procura na direção cujos pontos de gradiente permanecem na mesma direção), e o cálculo do gradiente em relação ao valor futuro dos parâmetros (*Nesterov Accelerated Gradient* - NAG) [57]. Outras melhorias visam as atualizações das restrições para cada parâmetro. Por exemplo, *Adagrad* (algoritmo do gradiente adaptativo, do inglês *Adaptive Gradient Algorithm*) [58] normaliza o gradiente pela raiz quadrada da soma do quadrado dos gradientes. Como essa operação resulta em uma possibilidade de extrapolação da capacidade de aprendizado, Zeiler propôs o *Adadelta* [59] (do inglês *Adaptive Learning Rated Method*), que restringe o acúmulo a uma janela fixa, levando em conta a média dos gradientes no passado. Uma alternativa para o *Adadelta* é a Raiz Quadrada da Média da Propagação (do inglês *Root Mean Square Progation - RMSProp*) [60] que usa uma média exponencial decrescente dos gradientes passados. A Estimação Adaptativa do Momento (do inglês *Adaptive Moment Estimation - Adam* [61]) resulta da combinação do *RMSProp* e *NAG*. Este usa a versão corrigida da tendência dos momentos de primeira e segunda ordens do gradiente na atualização do valor dos parâmetros. Da mesma forma, uma combinação da *Adam* e *NAG* resulta em *Nadam* [62], em que o momento é aplicado uma vez e não duas vezes (para atualizar o gradiente e atualizar os parâmetros). O *Adamax* [61] é derivado do *Adam*. Este generaliza a atualização para a norma $\mathcal{L}_\infty$. Finalmente, pode-se observar que, no *Adadelta*, a média móvel exponencial dos gradientes passados ao quadrado impede a generalização. Assim, ao invés disso, o *AMSGrad* [63] usa o máximo dos gradientes anteriores. Recentemente, em Zhang *et al.* [64] foi introduzido o *Lookahead*, um otimizador que assume a existência dos parâmetros da rede *slow* e *fast*. No *Lookahead*, usa-se o otimizador padrão várias vezes nos parâmetros *fast* para cada vez que o otimizador é usado nos parâmetros *slow*, permitindo ao primeiro a oportunidade para melhor atribuir os valores da atualização correta para os parâmetros.

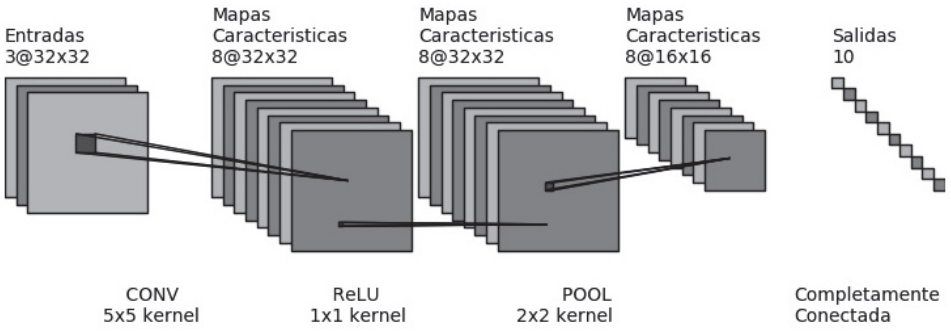


Fig. 2. Rede Neural Convolucional. Uma rede neural estabelece o mapeamento entre as entradas e as saídas. Para isso, uma representação hierárquica é construída aumentando sua abstração à medida em que se avança pelas camadas. O que torna uma rede neural convolucional interessante é sua capacidade de extrair mapas de características dos dados. Em cada camada, o resultado da convolução é projetado em um espaço não-linear, por exemplo, a partir do uso da função ReLU. Faz-se útil a inclusão de operações intermediárias, por exemplo, o uso do *pooling*, embora outras operações comuns não sejam incluídas, como a normalização em lote.

Um problema que ocorre durante a otimização é que o espaço de busca para os melhores parâmetros pode ser enorme. Assim, as soluções com DNNs são propensas a sobreajustes. Para resolver este problema, pode-se aplicar várias abordagens de regularização. Uma delas consiste em introduzir a regularização $\mathcal{L}_p$ como uma restrição de suavização para a função de perda, geralmente como um termo que gera um custo na magnitude dos parâmetros [65]. Outros esquemas incluem o *Dropout* [66], em que algumas das arestas no grafo não são consideradas durante a retro-propagação, e a parada antecipada [67], onde interrompe-se o aprendizado quando a taxa de treinamento continua decrescente e a taxa de validação começa a aumentar.

F. Aprendizagem

Pode-se classificar as técnicas de aprendizado como supervisionadas (os rótulos para cada classe existem), não supervisionadas (rótulos inexistentes), semi-supervisionadas (existe um pequeno número de rótulos), fracamente supervisionadas (rótulos com granulação grossa) e aprendizado por reforço (do inglês *Reinforcement Learning* - RL) [68]. No *aprendizado supervisionado*, temos as características e os valores de referência reais(*Ground Truth*). Pode-se distinguir o desenvolvimento de um processo de aprendizado, se comparamos ou não, o modelo resultante com outros. No primeiro caso, divide-se os dados em conjuntos de treinamento-validação e testes [69]. Em seguida, subdivide-se o conjunto de dados treinamento-validação em conjuntos de treinamento e validação. Neste último caso, divide-se os dados em conjuntos de treinamento e conjuntos de testes. Um realiza a Validação Cruzada (do inglês *Cross Validation* - CV) para a iteração em que a seleção é do conjunto de treinamento e o outro conjunto realiza a validação para configurações comparativas e testes para valores não-referenciados. Afrendas e Markatou [70] mostram que, para o uso da CV, o tamanho ideal da amostra para o conjunto de treinamento é metade do tamanho total de amostras disponíveis. A interação entre o conjunto de treinamento e o conjunto de testes, permite inferir as propriedades de generalização do sistema e avaliar a possibilidade de superajuste. Infelizmente, dados rotulados de alta qualidade e abundantes são intangíveis ou difíceis em se obter. Assim, os pesquisadores se voltaram para extrair informações usando aprendizado auto-supervisionado [71]. A ideia do aprendizado por reforço profundo (*Deep* RL ou DRL) é que o sistema aprende com a sua interação com o ambiente [72]. Um agente interage com seu ambiente, a partir de uma série de ações definidas por uma política, modificando seu ambiente e, portanto, suas observações. Normalmente, uma das estruturas do RL é definida como um processo de *Decisão de Markov* [73] em que se move entre os estados, dependendo das ações. Em um DRL [74], não se pode projetar o estado explicitamente. O agente age seguindo as políticas de transição entre estados aos quais corresponde a uma recompensa.

O uso do aprendizado profundo em vários problemas práticos motivou os desafios que estão atualmente sob investigação. Por exemplo, o problema de aprendizado não supervisionado na aprendizagem profunda [75], o aprendizado *on-line* para fluxos de dados [76], o desenvolvimento de novas técnicas de otimização para o treinamento de redes neurais profundas [77], a criação de técnicas de aprendizagem profunda distribuídas para acelerar ainda mais o processo de treinamento [78], e o uso do aprendizado multimodal profundo [79].

III. Redes Neurais Profundas

Nesta Seção serão revisadas as arquiteturas de Redes Neurais Profundas(DNNs) que sobreviveram ao teste da passagem de tempo, ou seja mesmo sendo criadas e propostas em décadas passadas, ainda possuem importância atualmente, bem como as que surgiram recentemente na literatura.

A. Redes Neurais Convolucionais (CNNs)

Semelhante às redes neurais tradicionais, as CNNs são inspiradas pelos neurônios dos cérebros de animais, com a vantagem de utilizarem o compartilhamento de parâmetros, interações esparsas e representações equivalentes [16]. Uma CNN (ver Figura 2), utiliza conexões locais e pesos compartilhados, ao contrário de camadas totalmente conectadas, resultando em redes mais rápidas com menos parâmetros que são mais fáceis e rápidas de treinar. Na CNN, as funções de

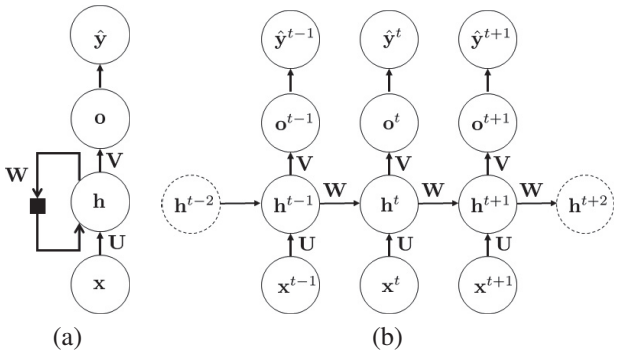


Fig. 3. *Redes Neurais Recorrentes.* Uma Rede Recorrente possui ciclos (a). Para ganhar intuição, pode-se considerá-las uma rede regular que se desenvolve ao longo do tempo, mantendo os pesos ($\mathbf{W}, \mathbf{U}, \mathbf{V}$) em cada nova iteração (b). O problema da instabilidade numérica é resolvido por meio de arquiteturas do tipo LSTM ou GRU.

ativações não lineares e as camadas de subamostragem (*pooling*) são localizadas após as camadas convolucionais, enquanto são utilizadas camadas totalmente conectadas no final do fluxo de processamento. Essas camadas utilizam características de baixo e médio níveis para gerar uma abstração de alto nível a partir dos dados de entrada. Sendo assim, é utilizado na última camada uma função do tipo *softmax* ou uma função tipo *Dobradiça*, por exemplo, para calcular os índices de classificação, isto é, uma medida da associação da instância à classe.

B. Redes Neurais Recorrentes (RNNs)

As RNNs possuem conexões que permitem as informações se moverem para frente, para a mesma camada ou para uma camada anterior (ver Figura 3). Graças a essas conexões recorrentes, essas redes podem levar em consideração o histórico acumulado e, portanto, são frequentemente usadas no processamento de dados temporais. Essa propriedade é essencial em muitas aplicações em que essa estrutura incorporada na sequência de dados transmite conhecimento útil à rede durante o treinamento. Por exemplo, para entender uma palavra em uma frase, se faz necessário conhecer o contexto. Portanto, pode-se considerar a RNN como unidades de memória de curto prazo que incluem camada de entrada, camada oculta(estado) e camada de saída. Em Razvan *et al.* [80] são apresentadas três abordagens de RNNs profundas, incluindo *Input-to-Hidden*, *Hidden-to-Output* e *Hidden-to-Hidden*, todas profundas que tiram vantagem de uma RNN mais profunda e reduzem a dificuldade do processo de aprendizagem. Uma dificuldade da RNN é sua sensibilidade à dissipação e elevados valores do gradiente [81]. Em outras palavras, os gradientes podem decair ou elevar seus valores exponencialmente devido às multiplicações de pequenos ou grandes lotes de derivadas durante o treinamento. Como resposta, as redes utilizando uma abordagem LSTM [82] foram introduzidas, fornecendo blocos de memória para suas conexões recorrentes. Cada bloco de memória contém uma ou mais células de memória auto-conectadas e três portas multiplicativas para controlar o fluxo de informações. Além disso, de acordo com He *et al.* [83], as conexões residuais em redes muito profundas podem aliviar significativamente o problema da dissipação do gradiente.

C. Redes Neurais Recursivas (RvNNs)

RvNNs são modelos adaptativos não lineares que aprendem informações profundas e estruturadas. O uso de Memória auto-associativa recursiva (do inglês *Recursive Auto-Associative Memory* - RAAM) [84], a qual é uma arquitetura criada para processar objetos estruturados de forma arbitrária, como árvores ou grafos e inspirou o desenvolvimento das RvNNs. Esta abordagem consiste em obter uma estrutura de dados recursiva de tamanho variável e gerar uma representação distribuída de largura fixa. A proposta de aprendizado tipo retro-propagação a partir da estrutura (do inglês *Backpropagation Through Structure* - BTS) foi introduzida para treinar a rede [84]. O BTS é semelhante ao algoritmo padrão de retro-propagação, mas suporta uma árvore tipo um grafo estruturado. A rede é treinada por associação automática para reproduzir o padrão da camada de entrada na camada de saída.

D. Redes Generativas Profundas (DGNs)

As DGNs aprendem por modelos de distribuição de dados do conjunto de treinamento gerando novas amostras dos dados com algumas variações. Isso permite obter uma distribuição que seja tão semelhante quanto possível à distribuição de dados real original. A seguir, são apresentadas as abordagens mais utilizadas e eficientes das DGNs.

1) Redes de Crenças Profundas (DBNs) e Redes Profundas de Boltzmann (DBMs): DBNs [85] são modelos generativos probabilísticos híbridos nos quais uma típica RBM com conexões não direcionadas é formada nas duas camadas superiores, e as camadas inferiores usam conexões direcionadas para receber as entradas da camada acima. A camada mais baixa, que é a camada visível, representa os estados das unidades de entrada como um vetor de dados. Uma DBN aprende a reconstruir probabilisticamente suas entradas em uma abordagem auto-supervisionada, enquanto as camadas atuam como detectores de características nas entradas. Além disso, um processo de treinamento supervisionado dá à DBN a capacidade de realizar tarefas de classificação. Pode-se considerar a DBN como composta de várias RBMs [26], onde por sua vez a camada oculta de cada sub-rede pode ser vista como uma camada visível para a próxima sub-rede. A DBN inclui um algoritmo guloso para melhorar o modelo generativo, permitindo que cada sub-rede receba sequencialmente diferentes representações dos dados, uma vez que a RBM não será capaz de modelar os dados originais de forma ideal. Quando a rede aprende os pesos iniciais, os dados podem ser mapeados a partir da matriz de ponderação transposta para criar os dados de nível superior para a próxima camada. Em Hinton *et al.* [25], demonstra-se que o logaritmo da probabilidade de cada vetor de dados de entrada é limitado sob a distribuição aproximada. Além disso, cada nova camada na DBN melhora os limites variacionais na camada mais profunda em comparação com a camada anterior, que inicializa o novo bloco de RBM na direção correta. As DBMs [86] diferem das DBNs em que estas possuem uma RBM não direcionada nas camadas inferiores. Calcula-se o algoritmo de treinamento guloso em cada uma das camadas para uma DBM, modificando o procedimento em uma DBN. Uma aproximação

fatorial *a posteriori* pode pegar o resultado da primeira RBM ou a probabilidade da segunda camada. Utilização de uma média geométrica dessas duas distribuições pode ser uma melhor ideia para equilibrar as aproximações *a posteriori*.

2) Redes Generativas Adversariais (GANs): As GANs [87] consistem em um modelo generativo G e um modelo discriminativo D. Enquanto G captura a distribuição p_g sobre os dados reais t, D tenta diferenciar uma amostra que vem dos dados modelados m, representados como p_m, ao invés de p_g. Em cada iteração de retro-propagação, o gerador e o discriminador competem entre si. Enquanto o gerador produz dados mais realistas para enganar o discriminador, este último tenta distinguir os dados reais dos falsos que foram gerados por G. O modelo é considerado estável quando ambos atingem o ponto em que nenhum deles pode ser melhorado, como $p_g = p_m$. Ou seja, o discriminador não consegue mais distinguir a fonte dos dados.

3) Autocodificadores Variacionais (VAEs): Os VAEs [88] usam uma rede neural profunda para aprender representações a partir de dados complexos sem supervisão. Um VAE inclui um codificador e um decodificador, sendo ambos redes neurais. O codificador aprende variáveis latentes da entrada e o decodificador gera uma saída com base em amostras das variáveis latentes. Com a quantidade de dados suficientes de treinamento, o codificador e o decodificador podem ser treinados por meio da minimização da perda de reconstrução e da divergência entre as distribuições de variáveis latentes e distribuições normais independentes. Os VAEs Condicionais [89] são versões estendidas de um VAE onde as redes de codificação e decodificação podem ter uma variável auxiliar como uma entrada adicional.

IV. Aplicações

Aplicações que utilizam aprendizagem profunda podem ser encontradas em áreas como finanças [90], medicina [91], esportes [92], educação [93], análise da qualidade do ar [94], análise de documentos [95], biologia [96], geografia [97], manufatura [98], transportes [99], robótica [100] e jogos eletrônicos [101], entre tantas outras. A seguir são agrupadas essas aplicações de acordo com o tipo de rede neural profunda empregada.

A. Redes Neurais Convolucionais

Desenvolvedores criaram uma ampla gama de aplicações utilizando CNNs, que vão desde reconhecimento de imagem em bancos de dados de grande porte [102] até decodificação de faces 3D com alta taxa de quadros [103]. Embora a maioria das aplicações sejam orientadas para imagens [104], existem muitos modelos usando outras fontes de dados. Por exemplo, em Wang *et al.* [105] é proposta uma abordagem baseada em *wavelets* para a previsão probabilística de energia eólica. Outros exemplos incluem a classificação de objetos 3D a partir de nuvens de pontos [106], reconhecimento de doenças em plantas [107] e reconhecimento da atividade humana em ambientes fechados [108].

B. Redes Neurais Recorrentes

O reconhecimento de fala é a aplicação clássica para modelos RNNs, em que os resultados experimentais mostram uma redução da taxa de erro no reconhecimento em conjuntos de dados de domínio público [109], [110]. Recentemente, o Supremo Tribunal Federal Brasileiro tem usado um modelo LSTM bidirecional para classificar casos de processos judiciais a partir de documentos digitalizados com níveis variados de qualidade de imagem deste documentos [111]. Existem várias formas estendidas de RNNs. Por exemplo, as *Inside-Outside Nets* (IONs) [112] são RNNs espaciais para detecção de objetos por contexto. Elas têm sido usadas em aplicações que vão desde legendas de vídeos densos [113], métodos de diagnóstico auxiliados por computador [114] e detecção de faces [115]. Outras aplicações para as RNNs incluem a compactação de imagem [116], identificação biométrica de indivíduos [117], previsão de incêndio [118], modelagem acústica [119], [120], previsão de trajetória [121], [122], análise de correlação [123], incorporação de sentenças textuais [124], [125] e classificação de textos curtos [126].

C. Redes Neurais Recursivas

Em Socher *et al.* [127] é introduzido um modelo de RvNN para detecção de sentimentos em espaços semânticos de palavras, uma tarefa que exige treinamento supervisionado, recursos de avaliação e poderosos modelos composicionais. Já em Chandra *et al.* [128] ampliou-se esta pesquisa introduzindo um modelo para prever o *cyberbullying* no *Twitter*. No modelo proposto é demonstrada a identificação em tempo real do texto a partir de dados estruturados e não estruturados. Outras aplicações incluem o trabalho de Dinh*et al.* [129], que introduziu uma abordagem de detecção de pedestres treinada com propostas de regiões cuja disparidade predominante é orientada perpendicularmente aos planos verticais.

D. Redes Generativas Profundas

Quando os pesquisadores começaram a aplicar as DGNs na categorização de objetos [130], surgiram novas oportunidades para modelos que requerem poucas amostras de treinamento e informações prévias, onde estes poderiam ser incorporados. Depois, outras variantes generativas mostraram um potencial crescente. Por exemplo, o modelo GAN apresentado em [87] provou ser eficaz na geração de imagens artificiais em que os humanos não as distinguem das imagens reais [131]. A eficácia dos modelos GANs levou os pesquisadores a estudarem sua aplicação em segurança cibernética [132], [133]. Outros usos das GANs incluem a geração de imagens para treinar robôs para manipulação de objetos [134], descoberta de drogas assistida por computador [135], adaptação de domínios [136] e redes sem fio e móveis [137].

Em outras aplicações, os pesquisadores usaram as DBNs para extrair características fonéticas [138], prever a emoção humana [139] e a classificação de tumores cerebrais [140]. Enquanto que as DBMs foram aplicadas com sucesso para modelar padrões de compras do consumidor [141], modelagem de faces [142], classificação de área de mineração [143] e estimativa de pressão arterial [144].

Aplicações dos VAEs são principalmente focadas em problemas de dados ausentes (ou falta de dados). Por exemplo, em McCoy*et al.* [145] foi usado um VAE para a imputação de dados perdidos em dados de imagens, comparando um conjunto de imagens do circuito de fresagem sintético, não-linear e simulado, incluindo distúrbios de processo, ruído de medição e controle de realimentação. Além disso, no trabalho de Marivate *et al.* [146] foi usado o VAE para estimar dados perdidos do vírus da imunodeficiência humana (HIV). Outras aplicações incluem monitoramento de processos [147], detecção de anomalias [148], [149], detecção de intrusões [150], previsão de sentimentos [151], representação de documentos de texto [152], manipulação de atributos faciais [153], identificação de anomalias cerebrais [154] e detecção de movimento linear [155].

V. Conclusões

Aprendizagem profunda é um sub-campo vibrante da área de aprendizado de máquina em que os desenvolvimentos atuais fazem dela um tópico de discussão essencial em educação, pesquisa e aplicações em diversas áreas do conhecimento. Fundada com sua base a partir da aproximação universal de funções e da aprendizagem de parâmetros via retro-propagação, o poder da aprendizagem profunda é sua capacidade de extrair recursos cada vez mais abstratos apenas dos dados de entrada. Embora novas ideias estejam surgindo continuamente, algumas redes neurais profundas, incluindo CNNs, RNNs e GANs, solidificaram e estão maduras para seu uso em aplicações reais.

Alguns dos desafios no campo da aprendizagem profunda incluem: a) Compreensão do espaço de parâmetros para a melhoria das técnicas de otimização. Essa tarefa é de extrema relevância, pois em uma rede de aprendizado profundo pode ter milhões de parâmetros; portanto, encontrar um valor adequado para eles é uma tarefa difícil, especialmente quando se considera sua co-dependência. b) A falta de dados. Atualmente, o aprendizado profundo é baseado em grandes conjuntos de dados e o desempenho dos modelos tende a diminuir com o tamanho dos conjuntos de dados. Algumas alternativas incluem o aumento de dados (do inglês *Data Augmentation*), transferência de aprendizado (do inglês *Transfer Learning*) e geração de dados sintéticos. No entanto, se faz necessária uma investigação minuciosa para encontrar melhores soluções. c) Retro-propagação e memória. Com relação ao primeiro conceito, o desenvolvimento de alternativas para a retro-propagação para gerar estratégias de aprendizado biologicamente plausíveis, se faz necessário. Com relação ao segundo conceito, mecanismos para a introdução de conhecimento prévio e memória de longo prazo precisam ser desenvolvidos para minimizar o tempo de treinamento e o erro de generalização. Juntos, isso permitiria melhorar o aprendizado em redes neurais profundas, abordando a maneira pela qual o aprendizado humano ocorre, ou seja com poucos exemplos. d) Aprendizagem auto-supervisionada. O objetivo de reduzir a dependência de grandes conjuntos de dados rotulados motivou a adoção de abordagens com arquiteturas profundas de redes, que definem seus próprios objetivos, refinando seu desempenho por meio da autocorreção. e) Aprendizagem por reforço. A inspiração na neurociência cognitiva e da psicologia do desenvolvimento para decifrar o padrão do comportamento humano é replicada para construir modelos capazes de aprender tarefas, estabelecendo políticas de ação baseadas em recompensas e punições enquanto muda o ambiente.

Este trabalho sugere que a aprendizagem profunda é um campo amplo e dinâmico, onde foram alcançados desenvolvimentos impressionantes. Neste manuscrito, também é mostrado que o aprendizado profundo é uma disciplina fértil, cheia de ideias, estas novas e empolgantes, além de desafios e oportunidades.

Referências

[1] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine will Remake our World*. Basic Books, 2015.

[2] J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press, 1992.

[3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, 2014.

[4] S. Muggleton and L. De Raedt, "Inductive Logic Programming: Theory and Methods," *The Journal of Logic Programming*, vol. 19, pp. 629–679, 1994.

[5] D. Hofstadter, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic books, 1995.

[6] D. Rumelhart, G. Hinton, and R. Williams, "Learning Internal Representations by Error Propagation," California University, Tech. Rep., 1985.

[7] A. Yala, C. Lehman, T. Schuster, T. Portnoi, and R. Barzilay, "A Deep Learning Mammography-based Model for Improved Breast Cancer Risk Prediction," *Radiology*, p. 182716, 2019.

[8] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, and T. Graepel, "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[9] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[10] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning based Recommender System: A Survey and New Perspectives," *ACM CSUR*, vol. 52, no. 1, p. 5, 2019.

[11] P. Univaso, J. Ale, and J. Gurlekian, "Data Mining applied to Forensic Speaker Identification," *IEEE Latin America Transactions*, vol. 13, no. 4, pp. 1098–1111, April 2015.

[12] J. Jimenez, I. Gonzalez, and J. Lopez, "Challenges And Opportunities In Analytic-Predictive Environments Of Big Data And Natural Language Processing For Social Network Rating Systems," *IEEE Latin America Transactions*, vol. 16, no. 2, pp. 592–597, Feb 2018.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," in *CVPR*, 2014.

[14] Min-Ling Zhang and Zhi-Hua Zhou, "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, Oct 2006.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[17] A. Shaikhha, A. Fitzgibbon, D. Vytiniotis, and S. Peyton Jones, "Efficient differentiable programming in a functional array-processing language," *Proceedings of the ACM on Programming Languages*, vol. 3, no. ICFP, p. 97, 2019.

[18] A. Karpathy, "Software 2.0," 2017.

[19] H. Landahl, W. Mcculloch, and W. Pitts, "A Statistical Consequence of the Logical Calculus of Nervous Nets," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, p. 135–137, 1943.

[20] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, vol. 65, no. 6, p. 386–408, 1958.

[21] P. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Ph.D. dissertation, Harvard University, 1974.

[22] K. Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics*, vol. 36, no. 4, p. 193–202, 1980.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, p. 2278–2324, 1998.

[24] M. Jordan, "Serial Order: A Parallel Distributed Processing Approach," in *Advances in Psychology*. Elsevier, 1997, vol. 121, pp. 471–495.

[25] G. Hinton, S. Osindero, and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, p. 1527–1554, 2006.

[26] P. Smolensky, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," D. Rumelhart and J. McClelland, Eds. MIT Press, 1986, ch. Information Processing in Dynamical Systems: Foundations of Harmony Theory, pp. 194–281.

[27] H. Mhaskar and T. Poggio, "Function Approximation by Deep Networks," *arXiv:1905.12882*, 2019.

[28] P. Baldi and R. Vershynin, "The Capacity of Feed Forward Neural Networks," *Neural Networks*, vol. 116, pp. 288–311, 2019.

[29] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[30] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *International Conference on Machine Learning*, 2010, pp. 807–814.

[31] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," *arXiv:1811.03378*, 2018.

[32] S. Roy, S. Manna, S. Dubey, and B. Chaudhuri, "LiSHT: Non-Parametric Linearly Scaled Hyperbolic Tangent Activation Function for Neural Networks," *arXiv:1901.05894*, 2019.

[33] J. Su, D. Vargas, and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks," *IEEE Transactions on Evolutionary Computation*, 2019.

[34] H. Li, Y. Tian, K. Mueller, and X. Chen, "Beyond Saliency: Understanding Convolutional Neural Networks from Saliency Prediction on Layer-Wise Relevance Propagation," *Image and Vision Computing*, vol. 83, pp. 70–86, 2019.

[35] M. Alber, S. Lapuschkin, P. Seegerer, M. Hägele, K. Schütt, G. Montavon, W. Samek, K. Müller, S. Dähne, and P. Kindermans, "iNNvestigate neural networks!" *Journal of Machine Learning Research*, vol. 20, no. 93, pp. 1–8, 2019.

[36] M. Kahng, N. Thorat, D. Chau, F. Viégas, and M. Wattenberg, "GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1–11, 2018.

[37] S. Yu and J. Principe, "Understanding Autoencoders with Information Theoretic Concepts," *Neural Networks*, vol. 117, pp. 104–123, 2019.

[38] N. Cristianini and J. Shawe, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[39] C. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[40] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss Functions for Neural Networks for Image Processing," *arXiv:1511.08861*, 2015.

[41] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[42] H. Yao, S. Zhang, R. Hong, Y. Zhang, C. Xu, and Q. Tian, "Deep Representation Learning with Part Loss for Person Re-Identification," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2860–2871, 2019.

[43] I. Ward, M. Jalwana, and M. Bennamoun, "Improving Image-Based Localization with Deep Learning: The Impact of the Loss Function," *arXiv:1905.03692*, 2019.

[44] T. Do, T. Tran, I. Reid, V. Kumar, T. Hoang, and G. Carneiro, "A Theoretically Sound Upper Bound on the Triplet Loss for Improving the Efficiency of Deep Distance Metric Learning," in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 404–10 413.

[45] A. Baydin, B. Pearlmutter, A. Radul, and J. Siskind, "Automatic Differentiation in Machine Learning: A Survey," *Journal of Machine Learning Research*, vol. 18, no. 153, 2018.

[46] W. Maass, "Networks of Spiking Neurons: The Third Generation of Neural Network Models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[47] J. Thiele, O. Bichler, and A. Dupret, "SpikeGrad: An ANN-equivalent Computation Model for Implementing Backpropagation with Spikes," *arXiv:1906.00851*, 2019.

[48] A. Tavanaei and A. Maida, "BP-STDP: Approximating Backpropagation using Spike Timing Dependent Plasticity," *Neurocomputing*, vol. 330, pp. 39–47, 2019.

[49] C. Lee, S. Sarwar, and K. Roy, "Enabling Spike-based Backpropagation in State-of-the-Art Deep Neural Network Architectures," *arXiv:1903.06379*, 2019.

[50] Y. Pu and J. Wang, "Fractional-Order Backpropagation Neural Networks: Modified Fractional-order Steepest Descent Method for Family of Backpropagation Neural Networks," *arXiv:1906.09524*, 2019.

[51] T. Lillicrap and A. Santoro, "Backpropagation through Time and the Brain," *Current Opinion in Neurobiology*, vol. 55, pp. 82–89, 2019.

[52] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "Biologically Inspired Alternatives to Backpropagation Through Time for Learning in Recurrent Neural Nets," *arXiv:1901.09049*, 2019.

[53] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *arXiv:1609.04747*, 2016.

[54] G. Hacohen and D. Weinshall, "On the Power of Curriculum Learning in Training Deep Networks," *arXiv:1904.03626*, 2019.

[55] P. Netrapalli, "Stochastic Gradient Descent and its Variants in Machine Learning," *Journal of the Indian Institute of Science*, pp. 1–13, 2019.

[56] N. Qian, "On the Momentum Term in Gradient Descent Learning Algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[57] Y. Nesterov, "A Method for Unconstrained Convex Minimization Problem with the Rate of Convergence $O(1/k^2)$," in *Doklady AN USSR*, vol. 269, 1983, pp. 543–547.

[58] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[59] M. Zeiler, "AdaDelta: An Adaptive Learning Rate Method," *arXiv:1212.5701*, 2012.

[60] G. Hinton, "Neural Networks for Machine Learning." [Online]. Available: https://www.coursera.org/learn/neural-networks-deep-learning

[61] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, 2014.

[62] T. Dozat, "Incorporating Nesterov Momentum into Adam," 2016.

[63] S. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," *arXiv:1904.09237*, 2019.

[64] M. Zhang, J. Lucas, G. Hinton, and J. Ba, "Lookahead Optimizer: k Steps Forward, 1 Step Back," *arXiv:1907.08610*, 2019.

[65] A. Ng, "Feature Selection, L_1 vs. L_2 Regularization, and Rotational Invariance," in *International Conference on Machine Learning*. ACM, 2004, p. 78.

[66] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[67] M. Li, M. Soltanolkotabi, and S. Oymak, "Gradient Descent with Early Stopping is Provably Robust to Label Noise for Overparameterized Neural Networks," *arXiv:1903.11680*, 2019.

[68] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[69] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *IJCAI*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[70] G. Afendras and M. Markatou, "Optimality of Training/Test Size and Resampling Effectiveness in Cross-Validation," *Journal of Statistical Planning and Inference*, vol. 199, pp. 286–301, 2019.

[71] L. Jing and Y. Tian, "Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey," *arXiv:1902.06162*, 2019.

[72] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[73] M. Puterman, *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[74] V. François-Lavet, P. Henderson, R. Islam, M. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," *Foundations and Trends in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[75] Q. Le, "Building High-Level Features using Large Scale Unsupervised Learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8595–8598.

[76] M. Choy, D. Srinivasan, and R. Cheu, "Neural Networks for Continuous Online Learning and Control," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1511–1531, Nov 2006.
[77] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the Importance of Initialization and Momentum in Deep Learning," in *International Conference on Machine Learning*. JMLR, 2013, pp. 1139–1147.
[78] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large Scale Distributed Deep Networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1223–1231. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999271
[79] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng, "Multimodal Deep Learning," in *International Conference on International Conference on Machine Learning*, 2011, pp. 689–696.
[80] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks," Apr 2014.
[81] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *International Conference on Artificial Intelligence and Statistics*, 2010.
[82] X. Li and X. Wu, "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition," *CoRR*, vol. 1410.4281, 2014.
[83] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, jun 2016, pp. 770–778.
[84] C. Goller and A. Kuchler, "Learning Task-dependent Distributed Representations by Backpropagation through Structure," in *International Conference on Neural Networks*, vol. 1, 1996, pp. 347–352.
[85] G. Hinton, "Deep Belief Networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.
[86] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.
[87] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Conference on Neural Information Processing Systems*, 2014, pp. 2672–2680.
[88] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes," p. arXiv:1312.6114, Dec 2013.
[89] D. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised Learning with Deep Generative Models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589. [Online]. Available: http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf
[90] G. Jeong and H. Kim, "Improving Financial Trading Decisions using Deep Q-Learning: Predicting the Number of Shares, Action Strategies, and Transfer Learning," *Expert Systems with Applications*, vol. 117, pp. 125 – 138, 2019.
[91] S. Khan and T. Yairi, "A Review on the Application of Deep Learning in System Health Management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241 – 265, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327017306064
[92] E. Cust, A. Sweeting, K. Ball, and S. Robertson, "Machine and Deep Learning for Sport-Specific Movement Recognition: A Systematic Review of Model Development and Performance," *Journal of Sports Sciences*, vol. 37, no. 5, pp. 568–600, 2019.
[93] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, "A Systematic Review of Deep Learning Approaches to Educational Data Mining," *Complexity*, vol. 2019, 2019.
[94] Y. Ayturan, Z. Ayturan, and H. Altun, "Air Pollution Modelling with Deep Learning: A Review," *International Journal of Environmental Pollution and Environmental Modelling*, vol. 1, no. 3, pp. 58–62, 2018.
[95] L. Aristodemou and F. Tietze, "The State-of-the-Art on Intellectual Property Analytics (IPA): A Literature Review on Artificial Intelligence, Machine Learning and Deep Learning Methods for Analysing Intellectual Property (IP) Data," *World Patent Information*, vol. 55, pp. 37–51, 2018.
[96] H. Wang, S. Shang, L. Long, R. Hu, Y. Wu, N. Chen, S. Zhang, F. Cong, and S. Lin, "Biological Image Analysis using Deep Learning-based Methods: Literature Review," *Digital Medicine*, vol. 4, no. 4, p. 157, 2018.
[97] G. Grekousis, "Artificial Neural Networks and Deep Learning in Urban Geography: A Systematic Review and Meta-Analysis," *Computers, Environment and Urban Systems*, 2018.
[98] M. Borg, C. Englund, K. Wnuk, B. Duran, C. Levandowski, S. Gao, Y. Tan, H. Kaijser, H. Lönn, and J. Törnqvist, "Safety Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry," *arXiv:1812.05389*, 2018.
[99] K. Gopalakrishnan, "Deep Learning in Data-driven Pavement Image Analysis and Automated Distress Detection: A Review," *Data*, vol. 3, no. 3, p. 28, 2018.
[100] Y. Li, Q. Lei, C. Cheng, G. Zhang, W. Wang, and Z. Xu, "A Review: Machine Learning on Robotic Grasping," in *International Conference on Machine Vision*, vol. 11041, 2019.
[101] P. Andersen, M. Goodwin, and O. Granmo, "Deep RTS: A Game Environment for Deep Reinforcement Learning in Real-Time Strategy Games," in *Conference on Computational Intelligence and Games*, 2018, pp. 1–8.
[102] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-Scale Long-Tailed Recognition in an Open World," in *Computer Vision and Pattern Recognition*, 2019.
[103] Y. Zhou, J. Deng, I. Kotsia, and S. Zafeiriou, "Dense 3D Face Decoding Over 2500FPS: Joint Texture and Shape Convolutional Mesh Decoders," in *Computer Vision and Pattern Recognition*, 2019.
[104] R. Singh, A. Mittal, and R. Bhatia, "3D Convolutional Neural Network for Object Recognition: A Review," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 15 951–15 995, 2019.
[105] H. Wang, G. Li, G. Wang, J. Peng, H. Jiang, and Y. Liu, "Deep Learning based Ensemble approach for Probabilistic Wind Power Forecasting," *Applied Energy*, vol. 188, pp. 56 – 70, 2017.
[106] H. Lei, N. Akhtar, and A. Mian, "Octree Guided CNN With Spherical Kernels for 3D Point Clouds," in *Computer Vision and Pattern Recognition*, 2019.
[107] A. da Silva, A. de Almeida, and F. Vidal, "Plant Diseases Recognition from Digital Images using Multichannel Convolutional Neural Networks," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019, pp. 450–458.
[108] B. Vandersmissen, N. Knudde, A. Jalalvand, I. Couckuyt, T. Dhaene, and W. De Neve, "Indoor Human Activity Recognition using High-Dimensional Sensors and Deep Neural Networks," *Neural Computing and Applications*, Aug 2019.
[109] T. Yang, T. Tseng, and C. Chen, "Recurrent Neural Network-based Language Models with Variation in Net Topology, Language, and Granularity," in *International Conference on Asian Language Processing*, Nov 2016, pp. 71–74.
[110] X. Chen, X. Liu, Y. Wang, A. Ragni, J. Wong, and M. Gales, "Exploiting Future Word Contexts in Neural Network Language Models for Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1444–1454, 2019.
[111] F. Braz, N. Silva, T. Emidio, F. Borges, M. Ferreira, P. Inazawa, V. Coelho, P. Sukiennik, P. Goncalves, F. Vidal, D. Alves, D. Gusmao, G. Ziegler, R. Fernandes, R. Zumblick, and F. Peixoto, "Document Classification using a Bi-LSTM to Unclog Brazil's Supreme Court," in *NIPS Workshop on Machine Learning for the Developing World*, 2018.
[112] S. Bell, C. Zitnick, K. Bala, and R. Girshick, "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks," in *Computer Vision and Pattern Recognition*, 2016, pp. 2874–2883.
[113] L. Yang, K. Tang, J. Yang, and L. Li, "Dense Captioning with Joint Inference and Visual Context," in *Computer Vision and Pattern Recognition*, 2017, pp. 1978–1987.
[114] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang, "MDNet: A Semantically and Visually Interpretable Medical Image Diagnosis Network," in *Computer Vision and Pattern Recognition*, 2017, pp. 3549–3557.
[115] Y. Bai and B. Ghanem, "Multi-scale Fully Convolutional Network for Face Detection in the Wild," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 2078–2087.
[116] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Image Compressed Sensing using Convolutional Neural Network," *IEEE Transactions on Image Processing*, pp. 1–1, 2019.
[117] S. Zhai, S. Liu, X. Wang, and J. Tang, "FMT: Fusing Multi-task Convolutional Neural Network for Person Search," *Multimedia Tools and Applications*, Jul 2019.
[118] J. Hodges, B. Lattimer, and K. Luxbacher, "Compartment Fire Predictions using Transpose Convolutional Neural Networks," *Fire Safety Journal*, p. 102854, 2019.
[119] N. Kim, K. Jeon, and H. Kim, "Convolutional Recurrent Neural Network-Based Event Detection in Tunnels Using Multiple Microphones," *Sensors*, vol. 19, no. 12, 2019.
[120] T. Zia and U. Zahid, "Long Short-Term Memory Recurrent Neural Network Architectures for Urdu Acoustic Modeling," *International Journal of Speech Technology*, vol. 22, no. 1, pp. 21–30, 2019.

[121] S. Sun, J. Chen, and J. Sun, "Traffic Congestion Prediction based on GPS Trajectory Data," *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, 2019.

[122] D. Duives, G. Wang, and J. Kim, "Forecasting Pedestrian Movements Using Recurrent Neural Networks: An Application of Crowd Monitoring Data," *Sensors*, vol. 19, p. 382, 2019.

[123] Y. Yu, S. Tang, F. Raposo, and L. Chen, "Deep Cross-Modal Correlation Learning for Audio and Lyrics in Music Retrieval," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 1, pp. 20:1–20:16, 2019.

[124] J. Maillard, S. Clark, and D. Yogatama, "Jointly Learning Sentence Embeddings and Syntax with Unsupervised Tree-LSTMs," *Natural Language Engineering*, vol. 25, no. 4, p. 433–449, 2019.

[125] J. Kang, H. Choi, and H. Lee, "Deep Recurrent Convolutional Networks for Inferring user Interests from Social Media," *Journal of Intelligent Information Systems*, vol. 52, no. 1, pp. 191–209, Feb 2019.

[126] J. Xu and Y. Cai, "Incorporating Context-Relevant Knowledge into Convolutional Neural Networks for Short Text Classification," *AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 10 067–10 068, Jul. 2019.

[127] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank," *EMNLP*, vol. 1631, pp. 1631–1642, 01 2013.

[128] N. Chandra, S. Khatri, and S. Som, "Cyberbullying Detection using Recursive Neural Network through Offline Repository," in *International Conference on Reliability, Infocom Technologies and Optimization*, Aug 2018, pp. 748–754.

[129] T. Dinh, N. Vinh, and J. Wook, "Robust Pedestrian Detection via a Recursive Convolution Neural Network," in *International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2018, pp. 281–286.

[130] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59 – 70, 2007.

[131] N. Caporusso, K. Zhang, G. Carlson, D. Jachetta, D. Patchin, S. Romeiser, N. Vaughn, and A. Walters, "User Discrimination of Content Produced by Generative Adversarial Networks," in *Human Interaction and Emerging Technologies*, T. Ahram, R. Taiar, S. Colson, and A. Choplin, Eds., Cham, 2020, pp. 725–730.

[132] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–20, 2019.

[133] C. Yinka and O. Ugot, "A Review of Generative Adversarial Networks and its Application in Cybersecurity," *Artificial Intelligence Review*, 2019.

[134] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-To-Real via Sim-To-Sim: Data-Efficient Robotic Grasping via Randomized-To-Canonical Adaptation Networks," in *IEEE Computer Vision and Pattern Recognition*, 2019.

[135] X. Yang, Y. Wang, R. Byrne, G. Schneider, and S. Yang, "Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery," *Chemical Reviews*, 2019.

[136] C. Ruan, W. Wang, H. Hu, and D. Chen, "Category-Level Adversaries for Semantic Domain Adaptation," *IEEE Access*, vol. 7, pp. 83 198–83 208, 2019.

[137] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

[138] Y. Seddiq, Y. Alotaibi, S. Selouani, and A. Meftah, "Distinctive Phonetic Features Modeling and Extraction using Deep Neural Networks," *IEEE Access*, vol. 7, pp. 81 382–81 396, 2019.

[139] M. Hassan, G. Alam, Z. Uddin, S. Huda, A. Almogren, and G. Fortino, "Human Emotion Recognition using Deep Belief Network Architecture," *Information Fusion*, vol. 51, pp. 10 – 18, 2019.

[140] A. Kharrat and M. Néji, "Classification of Brain Tumors using Personalized Deep Belief Networks on MRImages: PDBN-MRI," in *International Conference on Machine Vision*, vol. 11041, 2019.

[141] F. Xia, R. Chatterjee, and J. May, "Using Conditional Restricted Boltzmann Machines to Model Complex Consumer Shopping Patterns," *Marketing Science*, vol. 38, no. 4, pp. 711–727, 2019.

[142] C. Duong, K. Luu, K. Quach, and T. Bui, "Deep Appearance Models: A Deep Boltzmann Machine Approach for Face Modeling," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 437–455, 2019.

[143] K. Tan, F. Wu, Q. Du, P. Du, and Y. Chen, "A Parallel Gaussian–Bernoulli Restricted Boltzmann Machine for Mining Area Classification With Hyperspectral Imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 2, pp. 627–636, 2019.

[144] S. Lee and J. Chang, "Dempster–Shafer Fusion Based on a Deep Boltzmann Machine for Blood Pressure Estimation," *Applied Sciences*, vol. 9, no. 1, p. 96, 2019.

[145] J. McCoy, S. Kroon, and L. Auret, "Variational Autoencoders for Missing Data Imputation with Application to a Simulated Milling Circuit," *IFAC*, vol. 51, no. 21, pp. 141 – 146, 2018.

[146] N. Vukosi, V. Fulufhelo, and T. Marwala, "Investigation into the use of Autoencoder Neural Networks, Principal Component Analysis and Support Vector Regression in Estimating Missing HIV Data," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 682 – 689, 2008.

[147] S. Lee, M. Kwak, K. Tsui, and S. Kim, "Process Monitoring using Variational Autoencoder for High-Dimensional Nonlinear Processes," *Engineering Applications of Artificial Intelligence*, vol. 83, pp. 13 – 27, 2019.

[148] M. Nicolau and J. McDermott, "Learning Neural Representations for Network Anomaly Detection," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3074–3087, 2018.

[149] T. Wang, M. Qiao, Z. Lin, C. Li, H. Snoussi, Z. Liu, and C. Choi, "Generative Neural Networks for Anomaly Detection in Crowded Scenes," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1390–1399, 2018.

[150] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network," *Sensors*, vol. 19, no. 11, p. 2528, 2019.

[151] C. Wu, F. Wu, S. Wu, Z. Yuan, J. Liu, and Y. Huang, "Semi-Supervised Dimensional Sentiment Analysis with Variational Autoencoder," *Knowledge-Based Systems*, vol. 165, pp. 30–39, 2019.

[152] S. Wang, J. Cai, Q. Lin, and W. Guo, "An Overview of Unsupervised Deep Feature Representation for Text Categorization," *IEEE Transactions on Computational Social Systems*, 2019.

[153] X. Hou, K. Sun, L. Shen, and G. Qiu, "Improving Variational Autoencoder with Deep Feature Consistent and Generative Adversarial Training," *Neurocomputing*, vol. 341, pp. 183–194, 2019.

[154] H. Choi, S. Ha, H. Kang, H. Lee, and D. S. Lee, "Deep Learning Only by Normal Brain PET Identify Unheralded Brain Anomalies," *EBioMedicine*, vol. 43, pp. 447–453, 2019.

[155] M. S. Kim, J. P. Yun, S. Lee, and P. Park, "Unsupervised Anomaly Detection of LM Guide Using Variational Autoencoder," in *International Symposium on Advanced Topics in Electrical Engineering*, 2019, pp. 1–5.

Joaquín Salas es profesor en el campo de la visión por computadora en el Instituto Politécnico Nacional. Miembro del Sistema Nacional de Investigación de México, sus intereses de investigación incluyen el monitoreo de sistemas naturales utilizando la percepción visual y plataformas aéreas. Salas recibió un doctorado en informática del ITESM, México. Ha sido profesor visitante en la Universidad de Stanford, la Universidad de Duke, la Universidad Estatal de Oregón, el Xerox PARC, el Centro de Visión por Computador y la Escuela Nacional Superior de Telecomunicaciones de Bretaña. Se ha desempeñado como copresidente de la Conferencia Mexicana para el Reconocimiento de Patter tres veces. Salas fue becario Fulbright para el Departamento de Estado de los Estados Unidos. Ha sido editor invitado de Elsevier Pattern Recognition y Pattern Recognition Letters. Por sus servicios en el Instituto Politécnico Nacional, recibió la medalla it Lázaro Cárdenas del Presidente de México. Póngase en contacto con él en salas@ieee.org.

Joaquín Salas é professor na área de visão computacional do Instituto Politécnico Nacional. Membro do Sistema Nacional de Pesquisa do México, seus interesses de pesquisa incluem o monitoramento de sistemas naturais usando percepção visual e plataformas aéreas. Salas recebeu um Ph.D. em ciência da computação pelo ITESM, México. Foi pesquisador visitante na Universidade de Stanford, Duke University, Oregon State University, Xerox PARC, no Computer Vision Center e na École Nationale Supérieure des Télécommunications de Bretagne. Ele atuou como co-presidente da Conferência Mexicana de Reconhecimento de Patter três vezes. Salas foi estudioso da Fulbright no Departamento de Estado dos EUA. Ele foi editor convidado de Elsevier Pattern Recognition e Pattern Recognition Letters. Por seus serviços no Instituto Politécnico Nacional, recebeu a medalha Lázaro Cárdenas do Presidente do México. Entre em contato com ele em salas@ieee.org.

Flavio de Barros Vidal se graduó en Ingeniería Eléctrica de la Universidad Federal de Juiz de Fora (UFJF), Juiz de Fora, Brasil, en 2002. En 2005, recibió una Maestría . en Ingeniería Eléctrica de la Universidad de Brasilia (UnB), Brasilia, Brasil. En 2009, recibió un doctorado en Ingeniería Eléctrica de la Universidad de Brasilia, Brasilia, Brasil. Actualmente es profesor asociado de arquitectura de computadoras y visión por computadora en el Departamento de Ciencias de la Computación de la Universidad de Brasilia, Brasil. Sus intereses de investigación actuales incluyen análisis forense, biometría, aprendizaje profundo y visión por computadora. Es el líder del Grupo de Biometría y Tecnologías (BiTGroup) y miembro del grupo de investigación de Imagen, Señal y Acústica (LISA), ambos en la Universidad de Brasilia, Brasil. Póngase en contacto con él en url fbvidal@unb.br.

Flavio de Barros Vidal recebeu um B.Sc. em Engenharia Elétrica pela Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora, Brasil, em 2002. Em 2005, ele recebeu um mestrado. em Engenharia Elétrica pela Universidade de Brasília (UnB), Brasília, Brasil. Em 2009, ele recebeu um Ph.D. em Engenharia Elétrica pela Universidade de Brasília, Brasília, Brasil. Atualmente, é professor associado de arquitetura e visão computacional no Departamento de Ciência da Computação da Universidade de Brasília, Brasil. Seus interesses atuais de pesquisa incluem forense, biometria, aprendizado profundo e visão computacional. Ele é o líder do Grupo de Biometria e Tecnologias (BiTGroup) e membro do grupo de pesquisa Imagem, Sinal e Acústica (LISA), ambos da Universidade de Brasília, Brasil. Entre em contato com ele pelo e-mail fbvidal@unb.br.

José Fco. Martínez-Trinidad se graduó como licenciado en ciencias de la computación por la Facultad de Física y Matemáticas de la Universidad Autónoma de Puebla (BUAP), México, en 1995. Obtuvo una maestría en Ciencias de la Computación de la Facultad de Ciencias de la Computación de la Universidad Autónoma de Puebla, México, en 1997. Y el doctorado por el Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional (IPN), México, en 2000. Actualmente es miembro de la Coordinación de Ciencias Computacionales, en el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Mexico. Sus intereses de investigación incluyen reconocimiento de patrones lógico-combinatorios, clasificación supervisada y no supervisada, selección de características, selección de instancias y métodos de reconocimiento de patrones para la minería de datos. Ha coeditado 13 libros de actas de congresos en la serie Lecture Notes publicada por Springer y ha publicado alrededor de 150 artículos en revistas y documentos de conferencias sobre temas relacionados con el reconocimiento de patrones. Póngase en contacto con él en url fmartine@inaoep.mx.

José Fco. Martínez-Trinidad recebeu o B.S. Bacharel em Ciência da Computação pela Escola de Física e Matemática da Universidade Autônoma de Puebla (BUAP), México, em 1995, Mestrado em Ciência da Computação pela Faculdade de Ciências da Computação da Universidade Autônoma de Puebla, México , em 1997, e o Ph.D. formado pelo Centro de Investigação em Computação (CIC), Instituto Politécnico Nacional (IPN), México, em 2000. Atualmente é membro da Coordenação de Ciências Computacionais, no Instituto Nacional de Astrofísica, Óptica e Eletrônica (INAOE), México. Seus interesses atuais de pesquisa incluem reconhecimento de padrões lógico-combinatórios, classificação supervisionada e não supervisionada, seleção de recursos, seleção de instâncias e métodos de reconhecimento de padrões para mineração de dados. Ele co-editou 13 livros de anais de conferências da série Lecture Notes, publicados pela Springer, e publicou cerca de 150 jornais e conferências sobre assuntos relacionados ao reconhecimento de padrões. Entre em contato com ele pelo e-mail fmartine@inaoep.mx.