

## CAPÍTULO 20. MODELOS GENERATIVOS PROFUNDOS

---

de notas musicales utilizadas para componer canciones.[Boulanger-Lewandowski et al.](#)(2012) introdujo la**RNN-RBM**modelo de secuencia y lo aplicó a esta tarea. El RNN-RBM es un modelo generativo de una secuencia de tramas $X_t$ consistente en un RNN que emite los parámetros RBM para cada paso de tiempo. A diferencia de los enfoques anteriores en los que solo los parámetros de sesgo del RBM variaban de un paso de tiempo al siguiente, el RNN-RBM usa el RNN para emitir todos los parámetros del RBM, incluidos los pesos. Para entrenar el modelo, debemos poder propagar hacia atrás el gradiente de la función de pérdida a través de la RNN. La función de pérdida no se aplica directamente a las salidas RNN. En cambio, se aplica a la RBM. Esto significa que debemos diferenciar aproximadamente la pérdida con respecto a los parámetros RBM usando divergencia contrastiva o un algoritmo relacionado. Este gradiente aproximado puede entonces propagarse hacia atrás a través de la RNN utilizando el algoritmo habitual de propagación hacia atrás a través del tiempo.

## 20.8 Otras máquinas de Boltzmann

Son posibles muchas otras variantes de las máquinas Boltzmann.

Las máquinas de Boltzmann pueden ampliarse con diferentes criterios de formación. Nos hemos centrado en máquinas Boltzmann entrenadas para maximizar aproximadamente el criterio generativoregistros  $p(v)$ . También es posible entrenar RBM discriminativos que apuntan a maximizar registros  $p(y|v)$ en cambio ([Larochelle y Bengio,2008](#)). Este enfoque suele funcionar mejor cuando se utiliza una combinación lineal de los criterios generativo y discriminativo. Desafortunadamente, los RBM no parecen ser aprendices supervisados tan poderosos como los MLP, al menos utilizando la metodología existente.

La mayoría de las máquinas de Boltzmann utilizadas en la práctica solo tienen interacciones de segundo orden en sus funciones de energía, lo que significa que sus funciones de energía son la suma de muchos términos y cada término individual solo incluye el producto entre dos variables aleatorias. Un ejemplo de tal término es  $v_i W_{yo,j} h_j$ . También es posible entrenar máquinas Boltzmann de orden superior ([Sejnowski,1987](#)) cuyos términos de función de energía involucran los productos entre muchas variables. Las interacciones de tres vías entre una unidad oculta y dos imágenes diferentes pueden modelar transformaciones espaciales de un cuadro de video al siguiente ([Memisevic y Hinton,2007,2010](#)). La multiplicación por una variable de clase única puede cambiar la relación entre las unidades visibles y ocultas según la clase presente ([Nair y Hinton,2009](#)). Un ejemplo reciente del uso de interacciones de orden superior es una máquina de Boltzmann con dos grupos de unidades ocultas, con un grupo de unidades ocultas que interactúan con las unidades visibles y la etiqueta de clase y, y otro grupo de unidades ocultas que interactúan solo con el valores de entrada ([Luo et al.,2011](#)). Esto puede interpretarse como un estímulo

algunas unidades ocultas para aprender a modelar la entrada utilizando funciones que son relevantes para la clase, pero también para aprender unidades ocultas adicionales que explican los detalles molestos que son necesarios para las muestras de  $v$  para ser realistas pero no determinan la clase del ejemplo. Otro uso de las interacciones de orden superior es activar algunas funciones. [Sohn et al.\(2013\)](#) introdujo una máquina de Boltzmann con interacciones de tercer orden con variables de máscara binaria asociadas con cada unidad visible. Cuando estas variables de enmascaramiento se establecen en cero, eliminan la influencia de una unidad visible en las unidades ocultas. Esto permite que las unidades visibles que no son relevantes para el problema de clasificación se eliminen de la ruta de inferencia que estima la clase.

De manera más general, el marco de la máquina de Boltzmann es un espacio rico en modelos que permite muchas más estructuras de modelos de las que se han explorado hasta ahora. Desarrollar una nueva forma de máquina de Boltzmann requiere algo más de cuidado y creatividad que desarrollar una nueva capa de red neuronal, porque a menudo es difícil encontrar una función de energía que mantenga la manejabilidad de todas las diferentes distribuciones condicionales necesarias para usar la máquina de Boltzmann, pero a pesar de este esfuerzo requerido el campo permanece abierto a la innovación.

## 20.9 Propagación hacia atrás a través de operaciones aleatorias

Las redes neuronales tradicionales implementan una transformación determinista de algunas variables de entrada  $X$ . Al desarrollar modelos generativos, a menudo deseamos extender redes neuronales para implementar transformaciones estocásticas de  $X$ . Una forma sencilla de hacer esto es aumentar la red neuronal con entradas adicionales  $z$  que se muestran a partir de una distribución de probabilidad simple, como una distribución uniforme o gaussiana. La red neuronal puede continuar realizando cálculos deterministas internamente, pero la función  $f(x, z)$  aparecerá estocástico para un observador que no tiene acceso a  $z$ . Siempre que  $f$  es continua y diferenciable, entonces podemos calcular los gradientes necesarios para el entrenamiento utilizando la propagación hacia atrás como de costumbre.

Como ejemplo, consideremos la operación que consiste en extraer muestras y de una distribución gaussiana con media  $\mu$  y varianza  $\sigma^2$ :

$$y \sim \text{norte}(\mu, \sigma^2). \quad (20.54)$$

Debido a que una muestra individual de  $y$  no es producida por una función, sino por un proceso de muestreo cuyo resultado cambia cada vez que lo consultamos, puede parecer contradictorio tomar las derivadas de  $y$  con respecto a los parámetros de su distribución,  $\mu$  y  $\sigma^2$ . Sin embargo, podemos reescribir el proceso de muestreo como

transformando un valor aleatorio subyacente  $z \sim \text{norte}(z, 0, 1)$  para obtener una muestra de la distribución deseada:

$$y = \mu + \sigma z \quad (20.55)$$

Ahora podemos propagar hacia atrás a través de la operación de muestreo, considerándola como una operación determinista con una entrada adicional  $z$ . Fundamentalmente, la entrada extra es una variable aleatoria cuya distribución no es una función de ninguna de las variables cuyas derivadas queremos calcular. El resultado nos dice cómo un cambio infinitesimal en  $\mu$  o  $\sigma$  cambiaría la salida si pudiéramos repetir la operación de muestreo nuevamente con el mismo valor de  $z$ .

Ser capaz de propagar hacia atrás a través de esta operación de muestreo nos permite incorporarlo en un gráfico más grande. Podemos construir elementos del gráfico sobre la salida de la distribución de muestreo. Por ejemplo, podemos calcular las derivadas de alguna función de pérdida  $J(y)$ . También podemos construir elementos del gráfico cuyas salidas sean las entradas o los parámetros de la operación de muestreo. Por ejemplo, podríamos construir un gráfico más grande con  $\mu = f(X; \theta)$  y  $\sigma = g(\omega; X; \theta)$ . En este gráfico aumentado, podemos usar la propagación hacia atrás a través de estas funciones para derivar  $\nabla_{\theta} J(y)$ .

El principio utilizado en este ejemplo de muestreo gaussiano es de aplicación más general. Podemos expresar cualquier distribución de probabilidad de la forma  $p_{\text{ag}}(y; \theta) = p_{\text{ag}}(y | \omega)$  como  $p_{\text{ag}}(y | \omega)$ , donde  $\omega$  es una variable que contiene ambos parámetros  $\theta$ , y en su caso, las entradas  $X$ . dado un valor y muestreado de distribución  $p_{\text{ag}}(y | \omega)$ , donde  $\omega$  puede a su vez ser una función de otras variables, podemos reescribir

$$y \sim p_{\text{ag}}(y | \omega) \quad (20.56)$$

como

$$y = f(z, \omega), \quad (20.57)$$

dónde  $z$  es una fuente de aleatoriedad. Entonces podemos calcular las derivadas de  $y$  con respecto a  $\omega$  utilizando herramientas tradicionales como el algoritmo de retropropagación aplicado a  $f$ , siempre y cuando  $f$  sea continua y diferenciable en casi todas partes. Crucialmente,  $\omega$  no debe ser una función de  $z$ , y  $z$  no debe ser una función de  $\omega$ . Esta técnica a menudo se denomina **truco de reparametrización, Propagación inversa estocástica o análisis de perturbaciones**.

El requisito de que  $f$  sea continuo y diferenciable por supuesto requiere  $y$  a ser continuo. Si deseamos propagar hacia atrás a través de un proceso de muestreo que produce muestras de valores discretos, aún puede ser posible estimar un gradiente en  $\omega$ , utilizando algoritmos de aprendizaje por refuerzo como variantes del algoritmo REINFORCE ([Williams, 1992](#)), discutido en la sección [20.9.1](#).

En las aplicaciones de redes neuronales, normalmente elegimos extraerse de alguna distribución simple, como una unidad uniforme o una distribución gaussiana unitaria, y lograr distribuciones más complejas al permitir que la porción determinista de la red remodele su entrada.

La idea de propagar gradientes u optimizar mediante operaciones estocásticas se remonta a mediados del siglo XX (Presto, 1958; Capó, 1964) y se utilizó por primera vez para el aprendizaje automático en el contexto del aprendizaje por refuerzo (Williams, 1992). Más recientemente, se ha aplicado a aproximaciones variacionales (Opper y Archambeau, 2009) y redes neuronales estocásticas o generativas (Bengio et al., 2013b; Reyma, 2013; Kingma y Welling, 2014b, a; Rezende et al., 2014; Buen compañero et al., 2014c). Muchas redes, como los codificadores automáticos de eliminación de ruido o las redes regularizadas con abandono, también están naturalmente diseñadas para tomar el ruido como entrada sin requerir ninguna reparametrización especial para independizar el ruido del modelo.

### 20.9.1 Propagación hacia atrás a través de operaciones estocásticas discretas

Cuando un modelo emite una variable discreta  $y$ , el truco de la reparametrización no es aplicable. Supongamos que el modelo toma entradas  $x$  y parámetros  $\theta$ , ambos encapsulados en el vector  $\omega$ , y los combina con ruido aleatorio  $z$  para producir  $y$ :

$$y = f(z; \omega). \quad (20.58)$$

Porque  $y$  es discreto,  $f$  debe ser una función escalonada. Las derivadas de una función escalonada no son útiles en ningún punto. Justo en el límite de cada paso, las derivadas no están definidas, pero eso es un pequeño problema. El gran problema es que las derivadas son cero en casi todas partes, en las regiones entre los límites de los pasos. Las derivadas de cualquier función de costo  $J(y)$  por lo tanto, no brinde ninguna información sobre cómo actualizar los parámetros del modelo  $\theta$ .

El algoritmo REFORZAR (Incremento de RECOMPENSA = Factor no negativo  $\times$  Refuerzo compensado  $\times$  Elegibilidad característica) proporciona un marco que define una familia de soluciones simples pero poderosas (Williams, 1992). La idea central es que, aunque  $f(z; \omega)$  es una función escalonada con derivadas inútiles, el costo esperado  $\sum_{z \sim p(z)} J(f(z; \omega))$  es a menudo una función suave susceptible de descenso de gradiente. Aunque esa expectativa normalmente no es manejable cuando  $y$  es de alta dimensión (o es el resultado de la composición de muchas decisiones estocásticas discretas), se puede estimar sin sesgo utilizando un promedio de Monte Carlo. La estimación estocástica del gradiente se puede utilizar con SGD u otras técnicas de optimización estocástica basadas en gradientes.

La versión más simple de REINFORCE se puede derivar simplemente diferenciando el costo esperado:

$$miz[j(y)] = \sum_y j(y) pag(y) \quad (20.59)$$

$$\frac{\partial MI[j(y)]}{\partial \omega} = \sum_y j(y) \frac{\partial p(y)}{\partial \omega} \quad (20.60)$$

$$= \sum_y j(y) pag(y) \frac{\partial \text{registro} pag(y)}{\partial \omega} \quad (20.61)$$

$$\approx \frac{1}{\text{metro}} \sum_{y(i) \sim pag(y), i=1}^{\text{metro}} j(y) \frac{\partial \text{registro} pag(y)}{\partial \omega}. \quad (20.62)$$

Ecuación 20.60 se basa en la suposición de que  $\omega$  no hace referencia directamente. Es trivial extender el enfoque para relajar esta suposición. Ecuación 20.61 explota la regla de la derivada para el logaritmo,  $\frac{\partial \text{registro} pag(y)}{\partial \omega} = \frac{1}{pag(y)} \frac{\partial p(y)}{\partial \omega}$ . Ecuación 20.62 da un estimador de Monte Carlo imparcial del gradiente.

Donde sea que escribamos  $pag(y)$  en esta sección, uno podría igualmente escribir  $pag(y / X)$ . Esto es porque  $pag(y)$  está parametrizado por  $\omega$ , y  $w$  contiene ambos  $\theta$  y  $X$ , si  $X$  está presente.

Un problema con el estimador REINFORCE simple anterior es que tiene una varianza muy alta, por lo que muchas muestras de  $y$  debe dibujarse para obtener un buen estimador del gradiente, o de manera equivalente, si solo se extrae una muestra, SGD convergerá muy lentamente y requerirá una tasa de aprendizaje menor. Es posible reducir considerablemente la varianza de ese estimador usando **reducción de varianza** métodos (Wilson, 1984; L'Écuyer, 1994). La idea es modificar el estimador para que su valor esperado permanezca sin cambios pero su varianza se reduzca. En el contexto de REINFORCE, los métodos de reducción de la varianza propuestos involucran el cálculo de un **base** que se usa para compensar  $j(y)$ . Tenga en cuenta que cualquier compensación  $b(\omega)$  eso no depende de  $y$  no cambiaría la expectativa del gradiente estimado porque

$$mipag(y) \frac{\partial \text{registro} pag(y)}{\partial \omega} = \sum_y pag(y) \frac{\partial \text{registro} pag(y)}{\partial \omega} \quad (20.63)$$

$$= \sum_y -\frac{\partial p(y)}{\partial \omega} \quad (20.64)$$

$$= \frac{\partial}{\partial \omega} \sum_y pag(y) = \frac{\partial}{\partial \omega} 1 = 0, \quad (20.65)$$

Lo que significa que

$$mipag(y)(\hat{J}(y) - b(\omega)) \frac{\partial \text{registro}_{\text{pag}}(y)}{\partial \omega} = mipag(y) \hat{J}(y) \frac{\partial \text{registro}_{\text{pag}}(y)}{\partial \omega} - b(\omega) mipag(y) \frac{\partial \text{registro}_{\text{pag}}(y)}{\partial \omega} \quad (20.66)$$

$$= mipag(y) \hat{J}(y) \frac{\partial \text{registro}_{\text{pag}}(y)}{\partial \omega}. \quad (20.67)$$

Además, podemos obtener el óptimo  $b(\omega)$  calculando la varianza de  $(\hat{J}(y) - b(\omega)) \frac{\partial \text{registro}_{\text{pag}}(y)}{\partial \omega}$  bajo  $\text{pag}(y)$  y minimizando con respecto a  $b(\omega)$ . Lo que encontramos es que esta línea de base óptima  $b^*(\omega)$  es diferente para cada elemento  $\omega_i$  del vector  $\omega$ :

$$b^*(\omega) = \frac{mipag(y) \hat{J}(y) \frac{\partial \text{registro}_{\text{pag}}(y)}{\partial \omega_i}}{mipag(y) \frac{\partial \text{registro}_{\text{pag}}(y)^2}{\partial \omega_i}} \quad (20.68)$$

El estimador de gradiente con respecto a  $\omega$  entonces se convierte

$$(\hat{J}(y) - b(\omega))_i \frac{\partial \text{registro}_{\text{pag}}(y)}{\partial \omega_i} \quad (20.69)$$

dónde  $b(\omega)$  estima lo anterior  $b^*(\omega)_i$ . El estimado  $b$  generalmente se obtiene agregando salidas adicionales a la red neuronal y entrenando las nuevas salidas para estimar  $mipag(y)[\hat{J}(y) \frac{\partial \text{registro}_{\text{pag}}(y)}{\partial \omega_i}]^2$  y  $mipag(y) \frac{\partial \text{registro}_{\text{pag}}(y)^2}{\partial \omega_i}$  para cada elemento de  $\omega$ . Estos adicionales las salidas se pueden entrenar con el objetivo de error cuadrático medio, usando respectivamente  $\hat{J}(y) \frac{\partial \text{registro}_{\text{pag}}(y)^2}{\partial \omega_i}$  y  $\frac{\partial \text{registro}_{\text{pag}}(y)^2}{\partial \omega_i}$  como objetivos cuando se muestrean  $\text{pag}(y)$ , para una dada  $\omega$ . El estimado  $b$  luego puede recuperarse sustituyendo estas estimaciones en la ecuación 20.68. Mnih y Gregor (2014) prefirieron usar una sola salida compartida (en todos los elementos  $i$  de  $\omega$ ) entrenado con el objetivo  $\hat{J}(y)$ , usando como base  $b(\omega) \approx mipag(y)[\hat{J}(y)]$ .

Los métodos de reducción de la varianza se han introducido en el contexto del aprendizaje por refuerzo (Sutton et al., 2000; Tejedor y Tao, 2001), generalizando trabajos previos sobre el caso de la recompensa binaria por Dayán (1990). Verbengio et al. (2013b), Mnih y Gregor (2014), Licenciado en Letras et al. (2014), Mnih et al. (2014), oxuet al. (2015) para ver ejemplos de usos modernos del algoritmo REINFORCE con varianza reducida en el contexto del aprendizaje profundo. Además del uso de una línea de base dependiente de la entrada  $b(\omega)$ , Mnih y Gregor (2014) encontró que la escala de  $(\hat{J}(y) - b(\omega))$  podría ajustarse durante el entrenamiento dividiéndolo por su desviación estándar estimada por un promedio móvil durante el entrenamiento, como una especie de tasa de aprendizaje adaptativo, para contrarrestar el efecto de variaciones importantes que ocurren durante el curso del entrenamiento en el

magnitud de esta cantidad. Mnih y Gregor(2014) llamó a esto heurística **normalización de la varianza**.

Se puede entender que los estimadores basados en REINFORCE estiman el gradiente mediante la correlación de opciones de  $y$  con los valores correspondientes de  $\hat{y}(y)$ . Si un buen valor de  $y$  es poco probable con la parametrización actual, podría llevar mucho tiempo obtenerlo por casualidad y obtener la señal requerida de que esta configuración debe reforzarse.

## 20.10 Redes Generativas Dirigidas

Como se discutió en el capítulo dieciséis Los modelos gráficos dirigidos constituyen una clase destacada de modelos gráficos. Si bien los modelos gráficos dirigidos han sido muy populares dentro de la gran comunidad de aprendizaje automático, dentro de la comunidad más pequeña de aprendizaje profundo han sido eclipsados hasta aproximadamente 2013 por modelos no dirigidos como el RBM.

En esta sección, revisamos algunos de los modelos gráficos dirigidos estándar que tradicionalmente se han asociado con la comunidad de aprendizaje profundo.

Ya hemos descrito las redes de creencias profundas, que son un modelo parcialmente dirigido. También hemos descrito modelos de codificación dispersa, que pueden considerarse como modelos generativos dirigidos poco profundos. A menudo se utilizan como aprendices de funciones en el contexto del aprendizaje profundo, aunque tienden a tener un desempeño deficiente en la generación de muestras y la estimación de la densidad. Ahora describimos una variedad de modelos profundos y completamente dirigidos.

### 20.10.1 Redes de creencias sigmoideas

Redes de creencias sigmoideas (Neal, 1990) son una forma simple de modelo gráfico dirigido con un tipo específico de distribución de probabilidad condicional. En general, podemos pensar en una red de creencias sigmoidea como si tuviera un vector de estados binarios  $s$ , con cada elemento del estado influenciado por sus antepasados:

$$p_{\text{ag}}(s_i) = \sigma^{-} W_{j \leq i} s_j + b - i. \quad (20.70)$$

La estructura más común de la red de creencias sigmoidea es una que se divide en muchas capas, con un muestreo ancestral que procede a través de una serie de muchas capas ocultas y finalmente genera la capa visible. Esta estructura es muy similar a la red de creencias profundas, excepto que las unidades al comienzo de

el proceso de muestreo son independientes entre sí, en lugar de tomar muestras de una máquina Boltzmann restringida. Tal estructura es interesante por una variedad de razones. Una razón es que la estructura es un aproximador universal de las distribuciones de probabilidad sobre las unidades visibles, en el sentido de que puede aproximar arbitrariamente bien cualquier distribución de probabilidad sobre variables binarias, dada la profundidad suficiente, incluso si el ancho de las capas individuales está restringido al dimensionalidad de la capa visible ([Sutskever y Hinton, 2008](#)).

Mientras que generar una muestra de las unidades visibles es muy eficiente en una red de creencias sigmoideas, la mayoría de las otras operaciones no lo son. La inferencia sobre las unidades ocultas dadas las unidades visibles es intratable. La inferencia del campo medio también es intratable porque el límite inferior variacional implica tomar expectativas de camarillas que abarcan capas enteras. Este problema sigue siendo lo suficientemente difícil como para restringir la popularidad de las redes discretas dirigidas.

Un enfoque para realizar inferencias en una red de creencias sigmoideas es construir un límite inferior diferente que esté especializado para las redes de creencias sigmoideas ([Saúl et al., 1996](#)). Este enfoque solo se ha aplicado a redes muy pequeñas. Otro enfoque es usar mecanismos de inferencia aprendidos como se describe en la sección [19.5](#). La máquina de Helmholtz ([Dayán et al., 1995; Dayan y Hinton, 1996](#)) es una red de creencias sigmoidea combinada con una red de inferencia que predice los parámetros de la distribución del campo medio sobre las unidades ocultas. Enfoques modernos ([gregoret al., 2014; Mnih y Gregor, 2014](#)) a las redes de creencias sigmoideas todavía usan este enfoque de red de inferencia. Estas técnicas siguen siendo difíciles debido a la naturaleza discreta de las variables latentes. Uno no puede simplemente propagar hacia atrás a través de la salida de la red de inferencia, sino que debe usar la maquinaria relativamente poco confiable para propagar hacia atrás a través de procesos de muestreo discretos, descritos en la sección [20.9.1](#). Enfoques recientes basados en muestreo de importancia, vigilia-sueño reponderado ([Bornstein y Bengio, 2015](#)) y máquinas bidireccionales de Helmholtz ([Bornstein et al., 2015](#)) permiten entrenar rápidamente las redes de creencias sigmoideas y alcanzar un rendimiento de última generación en tareas de referencia.

Un caso especial de las redes de creencias sigmoideas es el caso en el que no hay variables latentes. El aprendizaje en este caso es eficiente, porque no hay necesidad de marginar las variables latentes de la probabilidad. Una familia de modelos llamados redes autorregresivas generaliza esta red de creencias completamente visible a otros tipos de variables además de las variables binarias y otras estructuras de distribuciones condicionales además de las relaciones loglineales. Las redes autorregresivas se describen más adelante, en la sección [20.10.7](#).

### 20.10.2 Redes de Generadores Diferenciables

Muchos modelos generativos se basan en la idea de utilizar un diferenciable **red generadora**. El modelo transforma muestras de variables latentes  $z$  a distribuciones sobre muestras  $X$  utilizando una función diferenciable  $gramo(z, \theta_{gramo})$  que normalmente se representa mediante una red neuronal. Esta clase de modelo incluye codificadores automáticos variacionales, que emparejan la red generadora con una red de inferencia, redes antagónicas generativas, que emparejan la red generadora con una red discriminadora, y técnicas que entranen redes generadoras de forma aislada.

Las redes de generadores son esencialmente solo procedimientos computacionales parametrizados para generar muestras, donde la arquitectura proporciona la familia de posibles distribuciones para muestrear y los parámetros seleccionan una distribución dentro de esa familia.

Como ejemplo, el procedimiento estándar para extraer muestras de una distribución normal con media  $\mu$  y la covarianza  $\Sigma$  es para alimentar muestras  $z$  de una distribución normal con media cero y covarianza de identidad a una red generadora muy simple. Esta red generadora contiene solo una capa afín:

$$X = gromo(z) = \mu + LZ \quad (20.71)$$

dónde  $L$  viene dada por la descomposición de Cholesky de  $\Sigma$ .

Los generadores de números pseudoaleatorios también pueden usar transformaciones no lineales de distribuciones simples. Por ejemplo, **muestreo por transformada inversa** (Devroye, 2013) dibuja un escalar  $z$  de  $tu(0, 1)$  y aplica una transformación no lineal a un escalar  $X$ . En este caso  $gramo(z)$  viene dada por la inversa de la función de distribución acumulada  $F(X) = \int p_{\text{ag}}(v) dv$ . Si somos capaces de especificar  $p_{\text{ag}}(X)$ , integrar sobre  $X$ , e invertir la función resultante, podemos tomar muestras de  $p_{\text{ag}}(X)$  sin utilizar el aprendizaje automático.

Para generar muestras a partir de distribuciones más complicadas que son difíciles de especificar directamente, difíciles de integrar o cuyas integrales resultantes son difíciles de invertir, usamos una red feedforward para representar una familia paramétrica de funciones no lineales  $gramo$  y use los datos de entrenamiento para inferir los parámetros seleccionando la función deseada.

podemos pensar en  $gramo$  como proporcionando un cambio no lineal de variables que transforma la distribución sobre  $z$  en la distribución deseada sobre  $X$ .

Recuperar de la ecuación 3.47 que, para invertible, diferenciable, continuo  $gramo$ ,

$$p_{\text{ag}}(z) = p_{\text{ag}}(gramo(z)) - \det\left(\frac{\partial g}{\partial z}\right)^{-1} \quad (20.72)$$

Esto impone implícitamente una distribución de probabilidad sobre  $X$ :

$$pag(X) = \frac{pag(gramo^{-1}(X))}{\det(\frac{\partial g}{\partial z})} \quad (20.73)$$

Por supuesto, esta fórmula puede ser difícil de evaluar, dependiendo de la elección de  $gramo$ , por lo que a menudo usamos medios indirectos de aprendizaje  $gramo$ , en lugar de tratar de maximizar registro  $pag(X)$  directamente.

En algunos casos, en lugar de usar  $gramo$  para proporcionar una muestra de  $X$  directamente, usamos  $gramo$  para definir una distribución condicional sobre  $X$ . Por ejemplo, podríamos usar una red generadora cuya capa final consta de salidas sigmoideas para proporcionar los parámetros medios de las distribuciones de Bernoulli:

$$pag(X=1/z) = gramo(z). \quad (20.74)$$

En este caso, cuando usamos  $gramo$  para definir  $pag(x/z)$ , imponemos una distribución sobre  $X$  al marginar  $z$ .

$$pag(X) = \int_z pag(x/z). \quad (20.75)$$

Ambos enfoques definen una distribución  $pag(gramo(X))$  y permitirnos entrenar diversos criterios de  $pag$  usando el truco de reparametrización de la sección 20.9.

Los dos enfoques diferentes para formular redes de generadores (emisión de los parámetros de una distribución condicional versus muestras de emisión directa) tienen fortalezas y debilidades complementarias. Cuando la red generadora define una distribución condicional sobre  $X$ , es capaz de generar datos discretos así como datos continuos. Cuando la red del generador proporciona muestras directamente, es capaz de generar solo datos continuos (podríamos introducir la discretización en la propagación hacia adelante, pero hacerlo significaría que el modelo ya no podría entrenarse usando la propagación hacia atrás). La ventaja del muestreo directo es que ya no estamos obligados a utilizar distribuciones condicionales cuya forma puede ser fácilmente escrita y manipulada algebraicamente por un diseñador humano.

Los enfoques basados en redes generadoras diferenciables están motivados por el éxito del descenso de gradiente aplicado a las redes feedforward diferenciables para la clasificación. En el contexto del aprendizaje supervisado, las redes de retroalimentación profunda entrenadas con aprendizaje basado en gradientes parecen prácticamente garantizados para tener éxito dadas suficientes unidades ocultas y suficientes datos de entrenamiento. ¿Puede esta misma receta para el éxito transferirse al modelado generativo?

El modelado generativo parece ser más difícil que la clasificación o la regresión porque el proceso de aprendizaje requiere optimizar criterios intratables. En el contexto

de redes generadoras diferenciables, los criterios son intratables porque los datos no especifican tanto las entradas  $z$  y las salidas  $x$  de la red del generador. En el caso del aprendizaje supervisado, tanto las entradas  $x$  y las salidas  $y$  se dieron, y el procedimiento de optimización solo necesita aprender cómo producir el mapeo especificado. En el caso del modelado generativo, el procedimiento de aprendizaje debe determinar cómo organizar el espacio de una manera útil y, además, cómo mapear desde  $z$  a  $x$ .

[Dosovitskiy et al. \(2015\)](#) estudiaron un problema simplificado, donde la correspondencia entre  $z$  y  $x$  es dado. Específicamente, los datos de entrenamiento son imágenes de sillas renderizadas por computadora. Las variables latentes  $z$  son parámetros proporcionados al motor de renderizado que describen la elección de qué modelo de silla usar, la posición de la silla y otros detalles de configuración que afectan la renderización de la imagen. Usando estos datos generados sintéticamente, una red convolucional puede aprender a mapear  $z$  descripciones del contenido de una imagen para  $x$  aproximaciones de imágenes renderizadas. Esto sugiere que las redes de generadores diferenciables contemporáneas tienen suficiente capacidad de modelo para ser buenos modelos generativos, y que los algoritmos de optimización contemporáneos tienen la capacidad de adaptarse a ellos. La dificultad radica en determinar cómo entrenar redes de generadores cuando el valor de  $z$  para cada  $x$  no es fijo y conocido de antemano cada vez.

Las siguientes secciones describen varios enfoques para entrenar redes de generadores diferenciables dados solo muestras de entrenamiento de  $x$ .

### 20.10.3 Codificadores automáticos variacionales

El **codificador automático variacional** (VAE) ([Reyma, 2013; Rezende et al., 2014](#)) es un modelo dirigido que utiliza la inferencia aproximada aprendida y se puede entrenar únicamente con métodos basados en gradientes.

Para generar una muestra del modelo, el VAE primero extrae una muestra  $z$  de la distribución de código  $p_{\text{modelo}}(z)$ . Luego, la muestra se ejecuta a través de una red generadora diferenciable  $g_{\text{modelo}}(z)$ . Finalmente,  $x$  se muestra de una distribución  $p_{\text{modelo}}(x; g_{\text{modelo}}(z)) = p_{\text{modelo}}(x | z)$ . Sin embargo, durante el entrenamiento, la red de inferencia aproximada (o codificador)  $q(z | x)$  se utiliza para obtener  $z$  de  $p_{\text{modelo}}(x | z)$  entonces se ve como una red decodificadora.

La idea clave detrás de los codificadores automáticos variacionales es que pueden entrenarse maximizando el límite inferior variacional  $L(q)$  asociado con el punto de datos  $x$ :

$$L(q) = \mathbb{E}_{z \sim q(z|x)} \text{registro } p_{\text{modelo}}(z, x) + H(q(z|x)) \quad (20.76)$$

$$= \mathbb{E}_{z \sim q(z|x)} \text{registro } p_{\text{modelo}}(x | z) - D_{\text{KL}}(q(z|x) || p_{\text{modelo}}(z)) \quad (20.77)$$

$$\leq \text{registro } p_{\text{modelo}}(x). \quad (20.78)$$

en ecuacion20.76, reconocemos el primer término como la verosimilitud logarítmica conjunta de las variables visibles y ocultas bajo el posterior aproximado sobre las variables latentes (al igual que con EM, excepto que usamos un posterior aproximado en lugar del posterior exacto).

Reconocemos también un segundo término, la entropía del posterior aproximado. Cuando  $q$  se elige para que sea una distribución gaussiana, con ruido agregado a un valor medio predicho, maximizar este término de entropía fomenta el aumento de la desviación estándar de este ruido. De manera más general, este término de entropía alienta al posterior variacional a colocar una masa de alta probabilidad en muchos valores que podrían haber generado  $X$ , en lugar de reducirse a una sola estimación puntual del valor más probable. en ecuacion20.77, reconocemos el primer término como el registro de probabilidad de reconstrucción que se encuentra en otros codificadores automáticos. El segundo término trata de hacer la distribución posterior aproximada  $q(z|X)$  y el modelo anterior  $p_{\text{modelo}}(z)$  acercarnos unos a otros.

Los enfoques tradicionales de la inferencia variacional y el aprendizaje infieren  $q$  a través de un algoritmo de optimización, normalmente ecuaciones iteradas de punto fijo (sección19.4). Estos enfoques son lentos y a menudo requieren la capacidad de calcular  $\text{míz}_{\sim q} \text{registro } p_{\text{modelo}}(z, x)$  en forma cerrada. La idea principal detrás del codificador automático variacional es entrenar un codificador paramétrico (a veces también llamado red de inferencia o modelo de reconocimiento) que produce los parámetros de  $q$ . Siempre y cuando  $z$  sea una variable continua, entonces podemos propagar hacia atrás a través de muestras de  $z$  trazada desde  $q(z|X) = q(z, X; \theta)$  para obtener un gradiente con respecto a  $\theta$ . El aprendizaje consiste entonces únicamente en maximizar  $L$  con respecto a los parámetros del codificador y decodificador. Todas las expectativas en  $L$  puede ser aproximado por muestreo de Monte Carlo.

El enfoque del codificador automático variacional es elegante, teóricamente agradable y fácil de implementar. También obtiene excelentes resultados y se encuentra entre los enfoques más avanzados para el modelado generativo. Su principal inconveniente es que las muestras de codificadores automáticos variacionales entrenados en imágenes tienden a ser algo borrosas. Las causas de este fenómeno aún no se conocen. Una posibilidad es que la borrosidad sea un efecto intrínseco de máxima verosimilitud, que minimiza  $D_{KL}(p_{\text{datos}} - p_{\text{modelo}})$ . Como se ilustra en la figura3.6, esto significa que el modelo asignará una alta probabilidad a los puntos que ocurren en el conjunto de entrenamiento, pero también puede asignar una alta probabilidad a otros puntos. Estos otros puntos pueden incluir imágenes borrosas. Parte de la razón por la que el modelo elegiría poner masa de probabilidad en imágenes borrosas en lugar de alguna otra parte del espacio es que los codificadores automáticos variacionales utilizados en la práctica suelen tener una distribución gaussiana para  $p_{\text{modelo}}(X|g_{\text{romo}}(z))$ . Maximizar un límite inferior en la probabilidad de tal distribución es similar a entrenar un codificador automático tradicional con error cuadrático medio, en el sentido de que tiende a ignorar las características de la entrada que ocupan pocos píxeles o que causan solo un pequeño cambio en el brillo de los píxeles que ocupan. Este problema no es específico de VAE y

se comparte con modelos generativos que optimizan un log-verosimilitud, o de manera equivalente,  $D_{KL}(p_{\text{datos}}||p_{\text{modelo}})$ , como sostiene [Teis et al. \(2015\)](#) y por [Húszár \(2015\)](#). Otro problema preocupante con los modelos VAE contemporáneos es que tienden a usar solo un pequeño subconjunto de las dimensiones  $z$ , como si el codificador no pudiera transformar suficientes direcciones locales en el espacio de entrada a un espacio donde la distribución marginal coincida con la anterior factorizada.

El marco VAE es muy sencillo de extender a una amplia gama de arquitecturas modelo. Esta es una ventaja clave sobre las máquinas Boltzmann, que requieren un diseño de modelo extremadamente cuidadoso para mantener la manejabilidad. Los VAE funcionan muy bien con una familia diversa de operadores diferenciables. Un VAE particularmente sofisticado es el **escritor de atención recurrente profundo** DIBUJAR modelo ([gregoret al., 2015](#)). DRAW utiliza un codificador recurrente y un decodificador recurrente combinados con un mecanismo de atención. El proceso de generación del modelo DRAW consiste en visitar secuencialmente diferentes parches de imagen pequeños y dibujar los valores de los píxeles en esos puntos. Los VAE también se pueden extender para generar secuencias definiendo RNN variacionales ([Chung et al., 2015b](#)) mediante el uso de un codificador y decodificador recurrente dentro del marco VAE. La generación de una muestra a partir de un RNN tradicional implica solo operaciones no deterministas en el espacio de salida. Los RNN variacionales también tienen una variabilidad aleatoria en el nivel potencialmente más abstracto capturado por las variables latentes de VAE.

El marco VAE se ha ampliado para maximizar no solo el límite inferior variacional tradicional, sino también el **codificador automático ponderado por importancia** ([Burda et al., 2015](#)) objetivo:

$$L_k(X, q) = \min_{z^{(1)}, \dots, z^{(k)} \sim q(z/X)} \text{registro} \left[ \frac{1}{k} \sum_{i=1}^k \frac{p_{\text{modelo}}(x, z^{(i)})}{q(z^{(i)}|X)} \right]. \quad (20.79)$$

Este nuevo objetivo es equivalente al límite inferior tradicional  $L$  cuando  $k=1$ . Sin embargo, también puede interpretarse como una estimación de la verdadera registro  $p_{\text{modelo}}(X)$  usando muestreo de importancia de  $z$  de la distribución de propuestas  $q(z|X)$ . El objetivo del codificador automático ponderado por importancia también es un límite inferior en registro  $p_{\text{modelo}}(X)$  y se vuelve más apretado como  $k$  aumenta.

Los codificadores automáticos variacionales tienen algunas conexiones interesantes con el MP-DBM y otros enfoques que implican la retropropagación a través del gráfico de inferencia aproximada ([Buen compañero et al., 2013b; Stoyanov et al., 2011; freno et al., 2013](#)). Estos enfoques anteriores requerían un procedimiento de inferencia como ecuaciones de punto fijo de campo medio para proporcionar el gráfico computacional. El codificador automático variacional se define para gráficos computacionales arbitrarios, lo que lo hace aplicable a una gama más amplia de familias de modelos probabilísticos porque no hay necesidad de restringir la elección.

de modelos a aquellos con ecuaciones de punto fijo de campo medio manejables. El codificador automático variacional también tiene la ventaja de que aumenta un límite en la verosimilitud logarítmica del modelo, mientras que los criterios para el MP-DBM y los modelos relacionados son más heurísticos y tienen poca interpretación probabilística más allá de hacer que los resultados de la inferencia aproximada sean precisos. Una desventaja del codificador automático variacional es que aprende una red de inferencia para un solo problema, infiriendo  $dado X$ . Los métodos más antiguos pueden realizar inferencias aproximadas sobre cualquier subconjunto de variables dado cualquier otro subconjunto de variables, porque las ecuaciones de punto fijo de campo medio especifican cómo compartir parámetros entre los gráficos computacionales para todos estos problemas diferentes.

Una propiedad muy agradable del codificador automático variacional es que entrenar simultáneamente un codificador paramétrico en combinación con la red del generador obliga al modelo a aprender un sistema de coordenadas predecible que el codificador puede capturar. Esto lo convierte en un excelente algoritmo de aprendizaje múltiple. Ver figura 20.6 para ver ejemplos de variedades de baja dimensión aprendidas por el codificador automático variacional. En uno de los casos demostrados en la figura, el algoritmo descubrió dos factores independientes de variación presentes en las imágenes de rostros: el ángulo de rotación y la expresión emocional.

#### 20.10.4 Redes adversarias generativas

Redes antagónicas generativas o GAN ([Buen compañero et al., 2014c](#)) son otro enfoque de modelado generativo basado en redes de generadores diferenciables.

Las redes adversarias generativas se basan en un escenario de teoría de juegos en el que la red generadora debe competir contra un adversario. La red de generadores produce muestras directamente  $X = g_{\text{gramo}}(z; \theta_{\text{gramo}})$ . Su adversario, el **red discriminatoria**, intenta distinguir entre muestras extraídas de los datos de entrenamiento y muestras extraídas del generador. El discriminador emite un valor de probabilidad dado por  $d(X; \theta_d)$ , indicando la probabilidad de que  $X$  es un ejemplo de entrenamiento real en lugar de una muestra falsa extraída del modelo.

La forma más sencilla de formular el aprendizaje en redes antagónicas generativas es como un juego de suma cero, en el que una función  $\mathcal{V}(\theta_{\text{gramo}}, \theta_d)$  determina el pago del discriminador. El generador recibe  $-\mathcal{V}(\theta_{\text{gramo}}, \theta_d)$  como su propia recompensa. Durante el aprendizaje, cada jugador intenta maximizar su propio pago, de modo que en la convergencia

$$\underset{\text{gramo}}{\arg\min} \mathcal{V}(g, d) \quad (20.80)$$

La opción predeterminada para  $\mathcal{V}$

$$\mathcal{V}(\theta_{\text{gramo}}, \theta_d) = \text{mix}_{\sim \text{pag}} \text{registro } d(X) + \text{mix}_{\sim \text{pag}} \text{registro } (1 - d(X)). \quad (20.81)$$

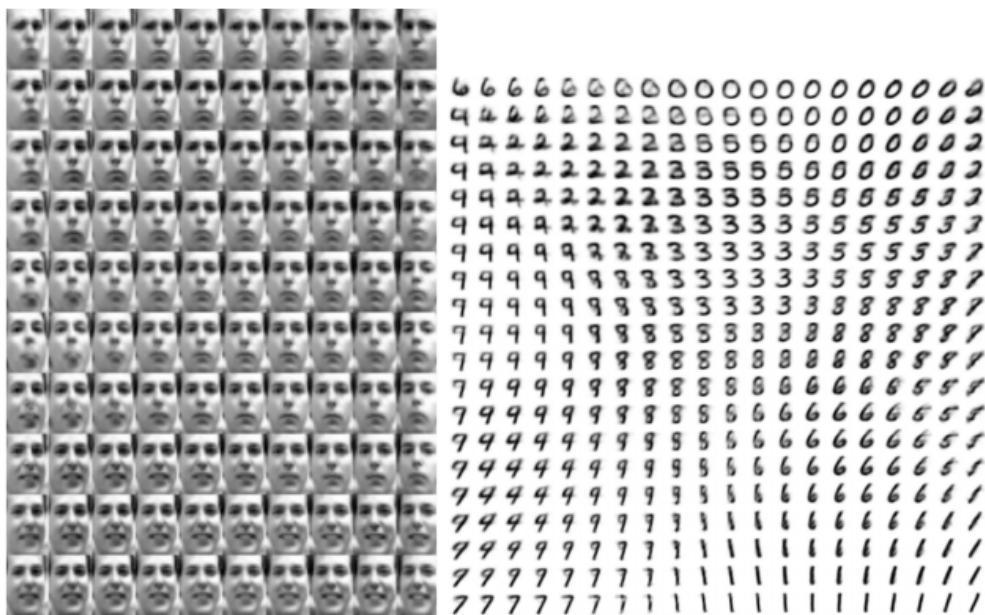


Figura 20.6: Ejemplos de sistemas de coordenadas bidimensionales para variedades de alta dimensión, aprendidos por un autocodificador variacional (Kingma y Welling, 2014a). Se pueden trazar dos dimensiones directamente en la página para su visualización, por lo que podemos obtener una comprensión de cómo funciona el modelo entrenando un modelo con un código latente 2-D, incluso si creemos que la dimensionalidad intrínseca de la variedad de datos es mucho mayor. Las imágenes que se muestran no son ejemplos del conjunto de entrenamiento sino imágenes realmente generadas por el modelo  $p_{\text{gen}}(x | z)$ , simplemente cambiando el "código" 2-D ( $z$ ). Cada imagen corresponde a una elección diferente de "código" en una cuadrícula uniforme 2-D. (Izquierda) El mapa bidimensional de las caras de Frey es múltiple. Una dimensión que se ha descubierto (horizontal) corresponde mayoritariamente a una rotación del rostro, mientras que la otra (vertical) corresponde a la expresión emocional. (Derecha) El mapa bidimensional de la variedad MNIST.

Esto lleva al discriminador a intentar aprender a clasificar correctamente las muestras como reales o falsas. Simultáneamente, el generador intenta engañar al clasificador haciéndole creer que sus muestras son reales. En la convergencia, las muestras del generador son indistinguibles. de datos reales, y las salidas del discriminador en todos los lados. el discriminador luego puede descartarse.

La principal motivación para el diseño de GAN es que el proceso de aprendizaje no requiere inferencia aproximada ni aproximación de un gradiente de función de partición. En el caso de que el  $\max_d \mathcal{V}(g, d)$  sea convexo en  $\theta_{\text{disc}}$  (como el caso donde la optimización se realiza directamente en el espacio de las funciones de densidad de probabilidad), se garantiza que el procedimiento convergerá y es asintóticamente consistente.

Desafortunadamente, el aprendizaje en GAN puede ser difícil en la práctica cuando  $g$  y  $d$  están representados por redes neuronales y  $\max_d \mathcal{V}(g, d)$  no es convexo. Buen compañero

(2014) identificó la falta de convergencia como un problema que puede hacer que las GAN no se ajusten bien. En general, no se garantiza que el descenso de gradiente simultáneo en los costos de dos jugadores alcance un equilibrio. Considere, por ejemplo, la función de valor  $v(a, b) = abdомinales$ , donde un jugador controla  $a$  e incurre en costos  $abdомinales$ , mientras que el otro jugador controla  $b$  y recibe un costo  $-abdомinales$ . Si modelamos a cada jugador haciendo pasos de gradiente infinitesimalmente pequeños, cada jugador reduciendo su propio costo a expensas del otro jugador, entonces  $a$  y  $b$  entrarán en una órbita circular estable, en lugar de llegar al punto de equilibrio en el origen. Tenga en cuenta que los equilibrios para un juego minimax no son mínimos locales de  $v$ . En cambio, son puntos que son simultáneamente mínimos para los costos de ambos jugadores. Esto significa que son puntos de silla de  $v$  que son mínimos locales con respecto a los parámetros del primer jugador y máximos locales con respecto a los parámetros del segundo jugador. Es posible que los dos jugadores se turnen para aumentar y luego disminuir  $v$  para siempre, en lugar de aterrizar exactamente en el punto de silla donde ningún jugador es capaz de reducir su costo. No se sabe en qué medida este problema de no convergencia afecta a las GAN.

Buen compañero(2014) identificaron una formulación alternativa de los pagos, en la que el juego ya no es de suma cero, que tiene el mismo gradiente esperado que el aprendizaje de máxima verosimilitud siempre que el discriminador sea óptimo. Debido a que el entrenamiento de máxima verosimilitud converge, esta reformulación del juego GAN también debería converger, dadas suficientes muestras. Desafortunadamente, esta formulación alternativa no parece mejorar la convergencia en la práctica, posiblemente debido a la subóptima calidad del discriminador, o posiblemente debido a la alta variación alrededor del gradiente esperado.

En experimentos realistas, la formulación de mejor rendimiento del juego GAN es una formulación diferente que no es de suma cero ni equivalente a la máxima verosimilitud, introducida porBuen compañeroet al.(2014c) con una motivación heurística. En esta formulación de mejor rendimiento, el generador tiene como objetivo aumentar la probabilidad logarítmica de que el discriminador cometa un error, en lugar de intentar disminuir la probabilidad logarítmica de que el discriminador haga la predicción correcta. Esta reformulación está motivada únicamente por la observación de que hace que la derivada de la función de costo del generador con respecto a los logits del discriminador permanezca grande incluso en la situación en la que el discriminador rechaza con confianza todas las muestras del generador.

La estabilización del aprendizaje GAN sigue siendo un problema abierto. Afortunadamente, el aprendizaje GAN funciona bien cuando la arquitectura del modelo y los hiperparámetros se seleccionan cuidadosamente.Radfordet al.(2015) elaboró una GAN convolucional profunda (DCGAN) que funciona muy bien para tareas de síntesis de imágenes y demostró que su espacio de representación latente captura importantes factores de variación, como se muestra en la figura15.9. Ver figura20.7para ver ejemplos de imágenes generadas por un generador DCGAN.

El problema de aprendizaje de GAN también se puede simplificar rompiendo la generación



Figura 20.7: Imágenes generadas por GAN entrenados en el conjunto de datos LSUN.(Izquierda) Imágenes de dormitorios generadas por un modelo DCGAN, reproducidas con autorización de [Radford et al.\(2015\)](#).(Bien)Imágenes de iglesias generadas por un modelo LAPGAN, reproducidas con permiso de[Dentónet al.\(2015\)](#).

proceso en muchos niveles de detalle. Es posible entrenar GAN condicionales ([Mirza y Osindero, 2014](#)) que aprenden a muestrear a partir de una distribución  $p_{\text{ag}}(x | y)$  en lugar de simplemente tomar muestras de una distribución marginal  $p_{\text{ag}}(X)$ .[Dentónet al.\(2015\)](#) mostró que se puede entrenar una serie de GAN condicionales para generar primero una versión de muy baja resolución de una imagen y luego agregar detalles a la imagen de manera incremental. Esta técnica se denomina modelo LAPGAN, debido al uso de una pirámide laplaciana para generar imágenes que contienen diferentes niveles de detalle. Los generadores LAPGAN pueden engañar no solo a las redes discriminadoras sino también a los observadores humanos, con sujetos experimentales que identifican hasta el 40% de las salidas de la red como datos reales. Ver figura 20.7 para ver ejemplos de imágenes generadas por un generador LAPGAN.

Una capacidad inusual del procedimiento de entrenamiento GAN es que puede adaptarse a distribuciones de probabilidad que asignan probabilidad cero a los puntos de entrenamiento. En lugar de maximizar la probabilidad logarítmica de puntos específicos, la red generadora aprende a trazar una variedad cuyos puntos se asemejan a los puntos de entrenamiento de alguna manera. De manera un tanto paradójica, esto significa que el modelo puede asignar una probabilidad logarítmica de infinito negativo al conjunto de prueba, al mismo tiempo que representa una variedad que un observador humano juzga para capturar la esencia de la tarea de generación. Esto no es claramente una ventaja o una desventaja, y también se puede garantizar que la red del generador asigne una probabilidad distinta de cero a todos los puntos simplemente haciendo que la última capa de la red del generador agregue ruido gaussiano a todos los valores generados.

Distribución gaussiana.

La deserción parece ser importante en la red discriminatoria. En particular, las unidades deben descartarse estocásticamente mientras se calcula el gradiente a seguir por la red del generador. Seguir el gradiente de la versión determinista del discriminador con sus pesos divididos por dos no parece ser tan efectivo. Del mismo modo, nunca usar la deserción parece arrojar resultados deficientes.

Si bien el marco GAN está diseñado para redes de generadores diferenciables, se pueden usar principios similares para entrenar otros tipos de modelos. Por ejemplo, **refuerzo autosupervisado** puede utilizar para entrenar un generador RBM para engañar a un discriminador de regresión logística ([brotando et al., 2002](#)).

## 20.10.5 Redes generativas de coincidencia de momentos

**Redes generativas de coincidencia de momentos** ([li et al., 2015](#); [Dziugaite et al., 2015](#)) son otra forma de modelo generativo basado en redes generadoras diferenciables. A diferencia de los VAE y las GAN, no necesitan emparejar la red del generador con ninguna otra red, ni una red de inferencia como la que se usa con los VAE ni una red discriminatoria como la que se usa con las GAN.

Estas redes están entrenadas con una técnica llamada **coincidencia de momentos**. La idea básica detrás de la coincidencia de momentos es entrenar al generador de tal manera que muchas de las estadísticas de las muestras generadas por el modelo sean lo más similares posible a las estadísticas de los ejemplos en el conjunto de entrenamiento. En este contexto, un **momento** es una expectativa de diferentes potencias de una variable aleatoria. Por ejemplo, el primer momento es la media, el segundo momento es la media de los valores al cuadrado, y así sucesivamente. En múltiples dimensiones, cada elemento del vector aleatorio puede elevarse a diferentes potencias, de modo que un momento puede ser cualquier cantidad de la forma

$$\text{mix} \prod_i X_i^{\text{norte}_i} \quad (20.82)$$

dónde  $\text{norte} = [\text{norte}_1, \text{norte}_2, \dots, \text{norte}_d]$  es un vector de enteros no negativos.

Tras un primer examen, este enfoque parece ser computacionalmente inviable. Por ejemplo, si queremos hacer coincidir todos los momentos de la forma  $X_i X_j$ , entonces necesitamos minimizar la diferencia entre un número de valores que es cuadrático en la dimensión de  $X$ . Además, incluso hacer coincidir todos los momentos primero y segundo solo sería suficiente para ajustar una distribución gaussiana multivariada, que captura solo relaciones lineales entre valores. Nuestras ambiciones para las redes neuronales son capturar relaciones no lineales complejas, lo que requeriría muchos más momentos. Las GAN evitan este problema de enumerar exhaustivamente todos los momentos mediante el uso de un

discriminador actualizado dinámicamente que enfoca automáticamente su atención en cualquier estadística que la red del generador coincida con la menor eficacia.

En cambio, las redes generativas de coincidencia de momentos se pueden entrenar minimizando una función de costo llamada **máxima discrepancia media** (Scholkopf y Smola, 2002; Gretton et al., 2012) o DMM. Esta función de costo mide el error en los primeros momentos en un espacio de dimensión infinita, utilizando un mapeo implícito al espacio de características definido por una función kernel para hacer que los cálculos en vectores de dimensión infinita sean manejables. El costo de MMD es cero si y solo si las dos distribuciones que se comparan son iguales.

Visualmente, las muestras de las redes generativas de coincidencia de momentos son algo decepcionantes. Afortunadamente, se pueden mejorar combinando la red del generador con un autocodificador. Primero, se entrena un codificador automático para reconstruir el conjunto de entrenamiento. A continuación, el codificador del codificador automático se utiliza para transformar todo el conjunto de entrenamiento en espacio de código. Luego, la red del generador se entrena para generar muestras de código, que pueden mapearse en muestras visualmente agradables a través del decodificador.

A diferencia de las GAN, la función de costo se define solo con respecto a un lote de ejemplos tanto del conjunto de entrenamiento como de la red del generador. No es posible realizar una actualización de entrenamiento en función de un solo ejemplo de entrenamiento o solo una muestra de la red del generador. Esto se debe a que los momentos deben calcularse como un promedio empírico de muchas muestras. Cuando el tamaño del lote es demasiado pequeño, MMD puede subestimar la verdadera cantidad de variación en las distribuciones que se muestran. Ningún tamaño de lote finito es lo suficientemente grande para eliminar este problema por completo, pero los lotes más grandes reducen la cantidad de subestimación. Cuando el tamaño del lote es demasiado grande, el procedimiento de entrenamiento se vuelve inviablemente lento, porque se deben procesar muchos ejemplos para calcular un solo paso de gradiente pequeño.

Al igual que con las GAN, es posible entrenar una red generadora usando MMD incluso si esa red generadora asigna probabilidad cero a los puntos de entrenamiento.

## 20.10.6 Redes generativas convolucionales

Al generar imágenes, a menudo es útil usar una red generadora que incluya una estructura convolucional (ver por ejemplo Buen compañero et al. (2014c) o Dosovitskiy et al. (2015)). Para hacerlo, usamos la "transposición" del operador de convolución, descrito en la sección 9.5. Este enfoque a menudo produce imágenes más realistas y lo hace usando menos parámetros que usando capas completamente conectadas sin compartir parámetros.

Las redes convolucionales para tareas de reconocimiento tienen un flujo de información desde la imagen hasta alguna capa de resumen en la parte superior de la red, a menudo una etiqueta de clase.

A medida que esta imagen fluye hacia arriba a través de la red, la información se descarta a medida que la representación de la imagen se vuelve más invariable a las transformaciones molestas. En una red de generadores, ocurre lo contrario. Se deben agregar detalles ricos a medida que la representación de la imagen que se generará se propaga a través de la red, culminando en la representación final de la imagen, que por supuesto es la imagen misma, en todo su esplendor detallado, con posiciones de objetos, poses y texturas. e iluminación El mecanismo principal para descartar información en una red de reconocimiento convolucional es la capa de agrupación. La red de generadores parece necesitar agregar información. No podemos poner el inverso de una capa de agrupación en la red del generador porque la mayoría de las funciones de agrupación no son invertibles. Una operación más simple es simplemente aumentar el tamaño espacial de la representación. Un enfoque que parece funcionar aceptablemente es usar una "desagrupación" como la introducida por [Dosovitskiy et al. \(2015\)](#). Esta capa corresponde a la inversa de la operación de agrupación máxima bajo ciertas condiciones de simplificación. En primer lugar, la zancada de la operación de agrupación máxima está restringida para que sea igual al ancho de la región de agrupación. En segundo lugar, se supone que la entrada máxima dentro de cada región de agrupación es la entrada en la esquina superior izquierda. Finalmente, se supone que todas las entradas no máximas dentro de cada región de agrupación son cero. Estas son suposiciones muy fuertes y poco realistas, pero permiten invertir el operador de agrupación máxima. La operación de desagrupación inversa asigna un tensor de ceros, luego copia cada valor de la coordenada espacial de la entrada a la coordenada espacial  $i \times k$  de la salida. El valor entero  $k$  define el tamaño de la región de agrupación. Aunque las suposiciones que motivan la definición del operador de desagrupamiento no son realistas, las capas posteriores pueden aprender a compensar su salida inusual, por lo que las muestras generadas por el modelo en su conjunto son visualmente agradables.

## 20.10.7 Redes autorregresivas

Las redes autorregresivas son modelos probabilísticos dirigidos sin variables aleatorias latentes. Las distribuciones de probabilidad condicional en estos modelos están representadas por redes neuronales (a veces, redes neuronales extremadamente simples, como la regresión logística). La estructura gráfica de estos modelos es el gráfico completo. Descomponen una probabilidad conjunta sobre las variables observadas utilizando la regla de la cadena de probabilidad para obtener un producto de condicionales de la forma  $PAG(X_d | X_{d-1}, \dots, X_1)$ . Tales modelos han sido llamados **redes bayesianas totalmente visibles** (FVBN) y se usó con éxito en muchas formas, primero con regresión logística para cada distribución condicional ([frey, 1998](#)) y luego con redes neuronales con unidades ocultas ([Bengio y Bengio, 2000b; Larochelle y Murray, 2011](#)). En algunas formas de redes autorregresivas, como NADE ([Larochelle y Murray, 2011](#)), descrito

en la sección 20.10.10 a continuación, podemos introducir una forma de compartir parámetros que brinda tanto una ventaja estadística (menos parámetros únicos) como una ventaja computacional (menos cómputo). Este es un ejemplo más del motivo recurrente de aprendizaje profundo de *reutilización de características*.

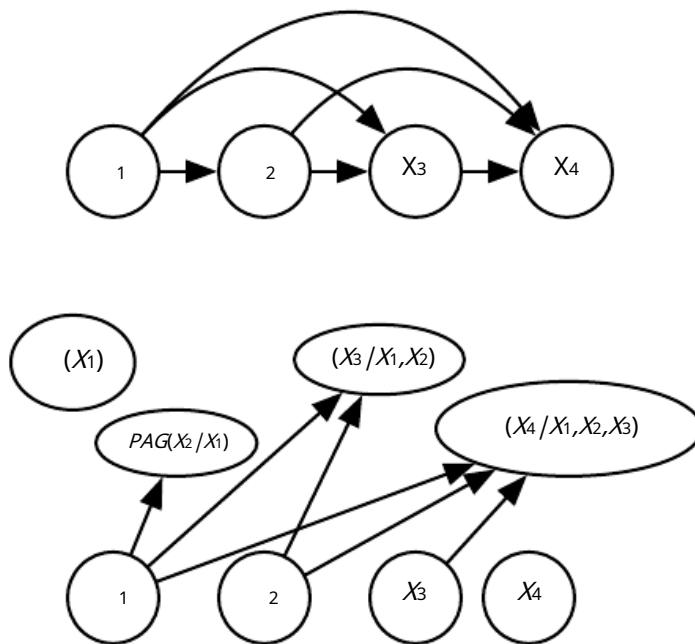


Figura 20.8: Una red de creencias totalmente visible predice la  $i$ -ésima variable de layo -1 los anteriores.(Arriba) El modelo gráfico dirigido para un FVBN.(Abajo) Gráfica computacional correspondiente, en el caso del FVBN logístico, donde cada predicción es realizada por un predictor lineal.

## 20.10.8 Redes autorregresivas lineales

La forma más simple de red autorregresiva no tiene unidades ocultas y no comparte parámetros o características. Cada  $PAG(X_i | X_{i-1}, \dots, X_1)$  está parametrizado como un modelo lineal (regresión lineal para datos con valores reales, regresión logística para datos binarios, regresión softmax para datos discretos). Este modelo fue introducido por frey(1998) y tiene  $\mathcal{O}(d^2)$  parámetros cuando hay  $d$  variables a modelar. se ilustra en la figura 20.8.

Si las variables son continuas, un modelo autorregresivo lineal es simplemente otra forma de formular una distribución gaussiana multivariante, capturando interacciones lineales por pares entre las variables observadas.

Las redes autorregresivas lineales son esencialmente la generalización de los métodos de clasificación lineal al modelado generativo. Por lo tanto, tienen la misma

ventajas y desventajas como clasificadores lineales. Al igual que los clasificadores lineales, pueden entrenarse con funciones de pérdida convexas y, a veces, admiten soluciones de forma cerrada (como en el caso de Gauss). Al igual que los clasificadores lineales, el modelo en sí no ofrece una forma de aumentar su capacidad, por lo que la capacidad debe aumentarse utilizando técnicas como expansiones de base de la entrada o el truco del núcleo.

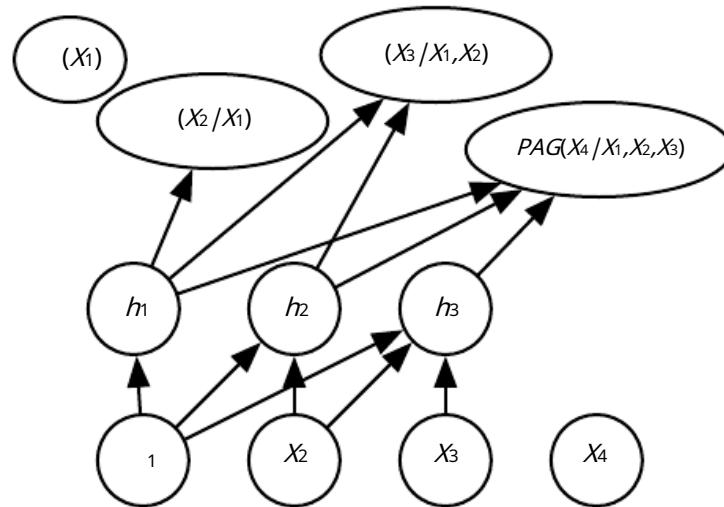


Figura 20.9: Una red neuronal autorregresiva predice la  $i$ -ésima variable  $X_i$  desde los  $i-1$  anteriores, pero está parametrizado de modo que las características (grupos de unidades ocultas denotadas  $h_j$ ) que son funciones de  $X_1, \dots, X_{i-1}$  puede reutilizarse para predecir todas las variables subsiguientes  $X_{i+1}, X_{i+2}, \dots, X_d$ .

## 20.10.9 Redes neuronales autorregresivas

Redes neuronales autorregresivas ([Bengio y Bengio, 2000a,b](#)) tienen el mismo modelo gráfico de izquierda a derecha que las redes autorregresivas logísticas (figura [20.8](#)) pero emplean una parametrización diferente de las distribuciones condicionales dentro de esa estructura de modelo gráfico. La nueva parametrización es más potente en el sentido de que se puede aumentar su capacidad tanto como sea necesario, permitiendo la aproximación de cualquier distribución conjunta. La nueva parametrización también puede mejorar la generalización mediante la introducción de un principio de uso compartido de parámetros y características común al aprendizaje profundo en general. Los modelos fueron motivados por el objetivo de evitar la maldición de la dimensionalidad que surge de los modelos gráficos tabulares tradicionales, compartiendo la misma estructura que la figura [20.8](#). En los modelos tabulares probabilísticos discretos, cada distribución condicional se representa mediante una tabla de probabilidades, con una entrada y un parámetro para cada posible configuración de las variables involucradas. Al utilizar una red neuronal en su lugar, se obtienen dos ventajas:

1. La parametrización de cada  $PAG(X_i | X_{yo-1}, \dots, X_1)$  por una red neuronal con  $(yo - 1) \times k$  entradas y  $k$  salidas (si las variables son discretas y toman  $k$  valores, codificado one-hot) permite estimar la probabilidad condicional sin requerir un número exponencial de parámetros (y ejemplos), pero aún así es capaz de capturar dependencias de alto orden entre las variables aleatorias.
2. En lugar de tener una red neuronal diferente para la predicción de cada  $X_i$ , *de izquierda a derecha* conectividad ilustrada en la figura 20.9 permite fusionar todas las redes neuronales en una sola. De manera equivalente, significa que las características de la capa oculta calculadas para predecir  $X_i$  se puede reutilizar para predecir  $X_{i+k}$  ( $k > 0$ ). Las unidades ocultas se organizan así en grupos que tienen la particularidad de que todas las unidades del  $i$ -th grupo solo depende de los valores de entrada  $X_1, \dots, X_i$ . Los parámetros utilizados para calcular estas unidades ocultas se optimizan conjuntamente para mejorar la predicción de todas las variables de la secuencia. Esta es una instancia de la *principio de reutilización* que se repite a lo largo del aprendizaje profundo en escenarios que van desde arquitecturas de red recurrentes y convolucionales hasta aprendizaje de transferencia y multitarea.

Cada  $PAG(X_i | X_{yo-1}, \dots, X_1)$  puede representar una distribución condicional al hacer que las salidas de la red neuronal predigan parámetros de la distribución condicional de  $X_i$ , como se discutió en la sección 6.2.1.1. Aunque las redes neuronales autorregresivas originales se evaluaron inicialmente en el contexto de datos multivariados puramente discretos (con una salida sigmoidea para una variable Bernoulli o una salida softmax para una variable multinomial), es natural extender tales modelos a variables continuas o distribuciones conjuntas que involucran variables discretas y continuas.

### 20.10.10 NADE

El **estimador de densidad autorregresiva neuronal** (NADE) es una forma reciente muy exitosa de red neuronal autorregresiva (Larochelle y Murray, 2011). La conectividad es la misma que para la red autorregresiva neuronal original de Bengio y Bengio (2000b) pero NADE introduce un esquema de intercambio de parámetros adicional, como se ilustra en la figura 20.10. Los parámetros de las unidades ocultas de diferentes grupos. *son compartidas*.

Los pesos  $W_{j, k, yo}$  desde el  $j$ -ésima entrada  $X_j$  hacia el  $k$ -ésimo elemento de la  $j$ -ésima grupo de unidad oculta  $h_{yo}$  ( $j \geq i$ ) se comparten entre los grupos:

$$W_{j, k, yo} = W_{k, yo}. \quad (20.83)$$

Los pesos restantes, donde  $j < yo$ , son cero.

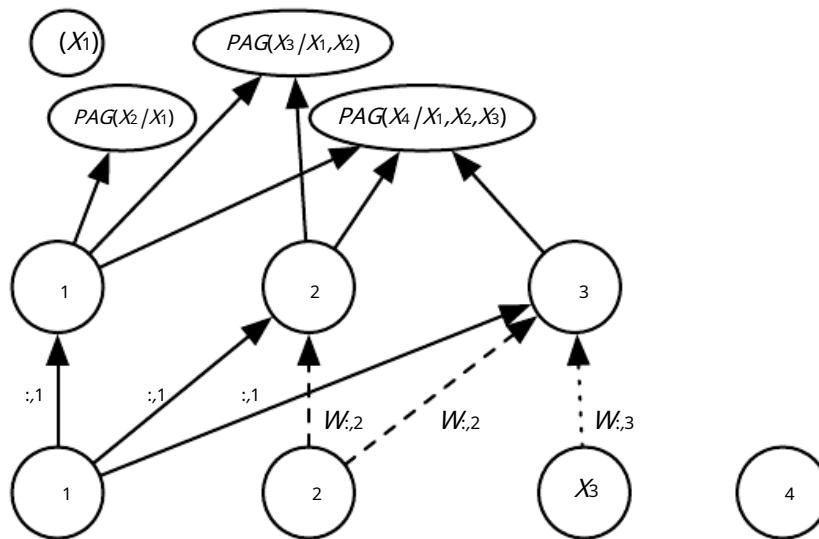


Figura 20.10: Una ilustración del estimador de densidad autorregresivo neuronal (NADE). Las unidades ocultas están organizadas en grupos.  $h_{ij}$  para que solo las entradas  $X_1, \dots, X_j$  participar en la informática  $h_{ij}$  y predecir  $PAG(X_j/X_{j-1}, \dots, X_1)$ , para  $j > yo$ . NADE se diferencia de las redes neuronales autorregresivas anteriores por el uso de un reparto de peso particular patrón:  $W_{-j, k, yo} = W_{k, yo}$  se comparte (indicado en la figura por el uso del mismo patrón de línea para cada instancia de un peso replicado) para todos los pesos que salen de  $X_j$  hacia la  $k$ -ésima unidad de cualquier grupo  $j \geq i$ . Recuerde que el vector  $(W_{1,i}, W_{2,i}, \dots, W_{n,yo})$  se denota  $W_{.,i}$ .

[Larochelle y Murray\(2011\)](#) eligió este esquema de compartición para que la propagación directa en un modelo NADE se asemeje vagamente a los cálculos realizados en la inferencia de campo medio para completar las entradas que faltan en un RBM. Esta inferencia de campo medio corresponde a ejecutar una red recurrente con pesos compartidos y el primer paso de esa inferencia es el mismo que en NADE. La única diferencia es que con NADE, los pesos de salida que conectan las unidades ocultas con la salida se parametrizan independientemente de los pesos que conectan las unidades de entrada con las unidades ocultas. En RBM, los pesos de oculto a salida son la transposición de los pesos de entrada a ocultos. La arquitectura NADE se puede extender para imitar no solo un paso de tiempo de la inferencia recurrente del campo medio, sino para imitar  $k$  pasos. Este enfoque se llama NADE- $k$ ([Raiko et al., 2014](#)).

Como se mencionó anteriormente, las redes autorregresivas pueden extenderse para procesar datos de valor continuo. Una forma particularmente poderosa y genérica de parametrizar una densidad continua es como una mezcla gaussiana (presentada en la sección [3.9.6](#)) con pesos de mezcla  $\alpha_i$  (el coeficiente o probabilidad previa para el componente  $i$ ), por media condicional componente  $\mu_i$  y varianza condicional por componente  $\sigma^2_i$ . A modelo denominado RNADE ([Uria et al., 2013](#)) utiliza esta parametrización para extender NADE a valores reales. Al igual que con otras redes de mezcla de densidad, los parámetros de esta

distribución son salidas de la red, con las probabilidades de peso de la mezcla producidas por una unidad softmax, y las varianzas parametrizadas para que sean positivas. El descenso del gradiente estocástico puede tener un mal comportamiento numérico debido a las interacciones entre las medias condicionales  $\mu$  y las varianzas condicionales  $\sigma^2$ . Para reducir esta dificultad, Uria et al. (2013) usan un pseudo-gradiente que reemplaza el gradiente en la media, en la fase de retropropagación.

Otra extensión muy interesante de las arquitecturas autorregresivas neuronales elimina la necesidad de elegir un orden arbitrario para las variables observadas (Murray y Larochelle, 2014). En las redes autorregresivas, la idea es entrenar a la red para que pueda hacer frente a cualquier orden mediante un muestreo aleatorio de las órdenes y proporcionando la información a las unidades ocultas especificando cuáles de las entradas se observan (en el lado derecho de la barra de condicionamiento) y que se deben predecir y, por lo tanto, se consideran faltantes (en el lado izquierdo de la barra de acondicionamiento). Esto es bueno porque permite usar una red autorregresiva entrenada para *realizar cualquier problema de inferencia* (es decir, predecir o tomar muestras de la distribución de probabilidad sobre cualquier subconjunto de variables dado cualquier subconjunto) de manera extremadamente eficiente. Finalmente, dado que muchos órdenes de variables son posibles (*no*  $t$  para *n* variables) y cada orden  $o$  de variables produce una diferencia  $pag(X/o)$ , podemos formar un conjunto de modelos para muchos valores de  $o$ :

$$pag_{\text{conjunto}}(x) = \frac{1}{k} \sum_{i=1}^k pag(x/\alpha_i). \quad (20.84)$$

Este modelo de conjunto generalmente generaliza mejor y asigna una mayor probabilidad al conjunto de prueba que un modelo individual definido por un solo orden.

En el mismo artículo, los autores proponen versiones profundas de la arquitectura, pero desafortunadamente eso hace que el cálculo sea tan costoso como en la red neuronal autorregresiva neuronal original (Bengio y Bengio, 2000b). La primera capa y la capa de salida todavía se pueden calcular en  $O(Nueva Hampshire)$  operaciones de suma y multiplicación, como en el NADE regular, donde  $h$  es el número de unidades ocultas (el tamaño de los grupos  $h$ , en cifras 20, 10 y 20, 9), mientras que es  $O(norte_2 h)$  en Bengio y Bengio (2000b). Sin embargo, para las otras capas ocultas, el cálculo es  $O(norte_2 h)$  si cada grupo "anterior" en la capa  $y$  participa en la predicción del "siguiente" grupo en la capa  $y+1$ , asumiendo *n* grupos de  $h$  unidades ocultas en cada capa. Haciendo el  $i$ -ésimo grupo en la capa  $y+1$  solo depende de la  $i$ -ésimo grupo, como en Murray y Larochelle (2014) en la capa  $y$  lo reduce a  $O(Nueva Hampshire)$ , que sigue siendo *h* veces peor que el NADE regular.

## 20.11 Extraer muestras de codificadores automáticos

en el capítulo 14, vimos que muchos tipos de codificadores automáticos aprenden la distribución de datos. Existen conexiones estrechas entre la coincidencia de puntajes, los codificadores automáticos que eliminan el ruido y los codificadores automáticos contractivos. Estas conexiones demuestran que algunos tipos de codificadores automáticos aprenden la distribución de datos de alguna manera. Todavía no hemos visto cómo extraer muestras de tales modelos.

Algunos tipos de codificadores automáticos, como el codificador automático variacional, representan explícitamente una distribución de probabilidad y admiten un muestreo ancestral directo. La mayoría de los otros tipos de codificadores automáticos requieren muestreo MCMC.

Los codificadores automáticos contractivos están diseñados para recuperar una estimación del plano tangente de la variedad de datos. Esto significa que la codificación y decodificación repetidas con ruido inyectado inducirán una caminata aleatoria a lo largo de la superficie de la variedad ([rifai et al., 2012; Mesnil et al., 2012](#)). Esta técnica de difusión múltiple es una especie de cadena de Markov.

También hay una cadena de Markov más general que puede muestrear desde cualquier codificador automático de eliminación de ruido.

### 20.11.1 Cadena de Markov asociada con cualquier codificador automático de eliminación de ruido

La discusión anterior dejó abierta la pregunta de qué ruido inyectar y dónde, para obtener una cadena de Markov que generaría a partir de la distribución estimada por el autocodificador. [bengio et al. \(2013c\)](#) mostró cómo construir tal cadena de Markov para **codificadores automáticos de eliminación de ruido generalizados**. Los codificadores automáticos de eliminación de ruido generalizados se especifican mediante una distribución de eliminación de ruido para muestrear una estimación de la entrada limpia dada la entrada corrupta.

Cada paso de la cadena de Markov que genera a partir de la distribución estimada consta de los siguientes subpasos, ilustrados en la figura 20.11:

1. A partir del estado anterior  $X$ , inyectar ruido de corrupción, muestreo  $x$  de  $C(x / X)$ .
2. Codificar  $x$  en  $h = F(X)$ .
3. Decodificar  $h$  para obtener los parámetros  $\omega = \text{gramo}(h)$  de  $\text{pag}(X / \omega = \text{gramo}(h)) = \text{pag}(X / X)$ .
4. Pruebe el siguiente estado  $X$  de  $\text{pag}(X / \omega = \text{gramo}(h)) = \text{pag}(X / X)$ .

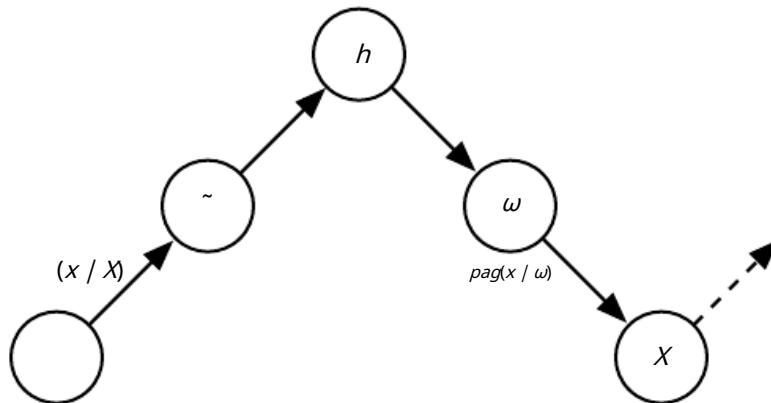


Figura 20.11: Cada paso de la cadena de Markov asociado con un autocodificador de eliminación de ruido entrenado, que genera las muestras del modelo probabilístico entrenado implícitamente por el criterio de verosimilitud logarítmica de eliminación de ruido. Cada paso consiste en (a) inyectar ruido a través del proceso de corrupción  $C$  en estado  $X$ , dando  $\tilde{X}$ , (b) codificarlo con la función  $F$ , dando  $h = F(\tilde{X})$ , (c) decodificando el resultado con la función  $gramo$ , dando parámetros  $\omega$  para la distribución de reconstrucción, y (d) dada  $\omega$ , muestreando un nuevo estado de la distribución de reconstrucción  $pag(X / \omega = gram(h))$ . En el típico caso de error de reconstrucción al cuadrado,  $gram(h) = X$ , que estima  $MI[x / X]$ , la corrupción consiste en agregar ruido gaussiano y muestreo de  $pag(X / \omega)$  consiste en añadir ruido gaussiano, por segunda vez, a la reconstrucción  $X$ . Este último nivel de ruido debería corresponder al error cuadrático medio de las reconstrucciones, mientras que el ruido inyectado es un hiperparámetro que controla la velocidad de mezcla, así como la medida en que el estimador suaviza la distribución empírica (Vicente, 2011). En el ejemplo ilustrado aquí, sólo los pasos condicionales son estocásticos ( $F$  y  $gramo$  son cálculos deterministas), aunque también se puede inyectar ruido dentro del codificador automático, como en las redes estocásticas generativas (Bengio et al., 2014).

bengio et al.(2014) mostró que si el codificador automático  $pag(X/X)$  forma un estimador consistente de la distribución condicional verdadera correspondiente, entonces la distribución estacionaria de la cadena de Markov anterior forma un estimador consistente (aunque implícito) de la distribución de generación de datos de  $X$ .

## 20.11.2 Sujeción y muestreo condicional

De manera similar a las máquinas de Boltzmann, los codificadores automáticos de eliminación de ruido y sus generalizaciones (como los GSN, que se describen a continuación) se pueden usar para tomar muestras de una distribución condicional.  $pag(X_F/X_0)$ , simplemente sujetando el *observado* unidades  $X_0$  y solo remuestreando el *gratis* unidades  $X_0$  dado  $X_0$  y las variables latentes muestreadas (si las hay). Por ejemplo, los MP-DBM se pueden interpretar como una forma de codificador automático de eliminación de ruido y pueden muestrear entradas faltantes. Posteriormente, los GSN generalizaron algunas de las ideas presentes en los MP-DBM para realizar la misma operación (bengio et al., 2014). alain et al.(2015) identificó una condición faltante de la Proposición 1 de bengio et al.(2014), que es que el operador de transición (definido por el mapeo estocástico que va de un estado de la cadena al siguiente) debe satisfacer una propiedad llamada **saldo detallado**, que especifica que una Cadena de Markov en equilibrio permanecerá en equilibrio ya sea que el operador de transición se ejecute hacia adelante o hacia atrás.

En la figura se muestra un experimento para sujetar la mitad de los píxeles (la parte derecha de la imagen) y ejecutar la cadena de Markov en la otra mitad. 20.12.



Figura 20.12: Ilustración de sujetar la mitad derecha de la imagen y ejecutar la Cadena de Markov volviendo a muestrear solo la mitad izquierda en cada paso. Estas muestras provienen de un GSN entrenado para reconstruir dígitos MNIST en cada paso de tiempo utilizando el procedimiento de retroceso.

### 20.11.3 Procedimiento de entrenamiento de regreso

El procedimiento de entrenamiento de caminata de regreso fue propuesto por [bengio et al. \(2013c\)](#) como una forma de acelerar la convergencia del entrenamiento generativo de autocodificadores de eliminación de ruido. En lugar de realizar una reconstrucción de codificación-descodificación de un solo paso, este procedimiento consiste en varios pasos alternativos de codificación-descodificación estocástica (como en la cadena generativa de Markov) inicializados en un ejemplo de entrenamiento (al igual que con el algoritmo de divergencia contrastiva, descrito en la sección [18.2](#)) y penalizando las últimas reconstrucciones probabilísticas (o todas las reconstrucciones en el camino).

Entrenamiento con  $k$  pasos es equivalente (en el sentido de lograr la misma distribución estacionaria) que entrenar con un paso, pero prácticamente tiene la ventaja de que los modos espurios más alejados de los datos se pueden eliminar de manera más eficiente.

## 20.12 Redes Estocásticas Generativas

**Redes estocásticas generativas** o GSN ([bengio et al., 2014](#)) son generalizaciones de codificadores automáticos de eliminación de ruido que incluyen variables latentes en el generativo

cadena de Markov, además de las variables visibles (generalmente denotadas  $X$ ).

Un GSN está parametrizado por dos distribuciones de probabilidad condicional que especifican un paso de la cadena de Markov:

1.  $pag(X_{(k)} / h_{(k)})$  dice cómo generar la siguiente variable visible dado el estado latente actual.

Esta "distribución de reconstrucción" también se encuentra en la eliminación de ruido de codificadores automáticos, RBM, DBN y DBM.

2.  $pag(h_{(k)} / h_{(k-1)}, X_{(k-1)})$  dice cómo actualizar la variable de estado latente, dado el estado latente anterior y la variable visible.

Los autocodificadores de eliminación de ruido y los GSN difieren de los modelos probabilísticos clásicos (dirigidos o no dirigidos) en que parametrizan el proceso generativo en sí mismo en lugar de la especificación matemática de la distribución conjunta de variables visibles y latentes. En cambio, este último se define *implícitamente, si existiera*, como la distribución estacionaria de la cadena generativa de Markov. Las condiciones para la existencia de la distribución estacionaria son leves y son las mismas condiciones requeridas por los métodos estándar de MCMC (consulte la sección 17.3). Estas condiciones son necesarias para garantizar que la cadena se mezcle, pero pueden ser violadas por algunas elecciones de las distribuciones de transición (por ejemplo, si fueran deterministas).

Uno podría imaginar diferentes criterios de formación para los GSN. El propuesto y evaluado por [bengio et al. \(2014\)](#) es simplemente una probabilidad logarítmica de reconstrucción en las unidades visibles, al igual que para eliminar el ruido de los codificadores automáticos. Esto se logra sujetando  $X_{(0)} = X$  al ejemplo observado y maximizando la probabilidad de generar  $X$  en algunos pasos de tiempo posteriores, es decir, maximizando el registro  $pag(X_{(k)} = x / h_{(k)})$ , donde  $h_{(k)}$  se muestra de la cadena, dado  $X_{(0)} = X$ . Para estimar el gradiente de registro  $pag(X_{(k)} = x / h_{(k)})$  con respecto a las demás piezas del modelo, [bengio et al. \(2014\)](#) utilice el truco de la reparametrización, presentado en la sección 20.9.

El **trenamiento de regresión** protocolo (descrito en la sección 20.11.3) se utilizó ([bengio et al., 2014](#)) para mejorar la convergencia de formación de los GSN.

### 20.12.1 GSN discriminantes

La formulación original de los GSN ([bengio et al., 2014](#)) estaba destinado al aprendizaje no supervisado y al modelado implícito  $pag(X)$  para datos observados  $X$ , pero es posible modificar el marco para optimizar  $pag(y / X)$ .

Por ejemplo, [Zhou y Troyanskaya \(2014\)](#) generalizan los GSN de esta manera, solo propagando hacia atrás la probabilidad logarítmica de reconstrucción sobre las variables de salida, manteniendo fijas las variables de entrada. Aplicaron esto con éxito para modelar secuencias.

(estructura secundaria de la proteína) e introdujo una estructura convolucional (unidimensional) en el operador de transición de la cadena de Markov. Es importante recordar que, para cada paso de la cadena de Markov, se genera una nueva secuencia para cada capa, y esa secuencia es la entrada para calcular los valores de otras capas (por ejemplo, el de abajo y el de arriba) en el siguiente paso de tiempo. .

Por lo tanto, la cadena de Markov está realmente sobre la variable de salida (y las capas ocultas de nivel superior asociadas), y la secuencia de entrada solo sirve para condicionar esa cadena, con la propagación hacia atrás que permite aprender cómo la secuencia de entrada puede condicionar la distribución de salida representada implícitamente por el Markov cadena. Por lo tanto, se trata de utilizar el GSN en el contexto de productos estructurados.

Zöhrer y Pernkopf(2014) introdujo un modelo híbrido que combina un objetivo supervisado (como en el trabajo anterior) y un objetivo no supervisado (como en el trabajo original de GSN), simplemente agregando (con un peso diferente) los costos supervisados y no supervisados, es decir, el registro de reconstrucción. probabilidades deyyXrespectivamente. Este criterio híbrido había sido introducido previamente para los mecanismos de gestión basada en los resultados porLarochelle y Bengio(2008). Muestran un rendimiento de clasificación mejorado usando este esquema.

## 20.13 Otros Esquemas de Generación

Los métodos que hemos descrito hasta ahora usan muestreo MCMC, muestreo ancestral o alguna combinación de los dos para generar muestras. Si bien estos son los enfoques más populares para el modelado generativo, de ninguna manera son los únicos enfoques.

Sohl-Dicksteinet al.(2015) desarrollado una **inversión de difusión**esquema de entrenamiento para el aprendizaje de un modelo generativo, basado en termodinámica de no equilibrio. El enfoque se basa en la idea de que las distribuciones de probabilidad de las que deseamos muestrear tienen estructura. Esta estructura puede destruirse gradualmente mediante un proceso de difusión que cambia gradualmente la distribución de probabilidad para tener más entropía. Para formar un modelo generativo, podemos ejecutar el proceso a la inversa, entrenando un modelo que restaura gradualmente la estructura a una distribución no estructurada. Mediante la aplicación iterativa de un proceso que acerque una distribución a la objetivo, podemos acercarnos gradualmente a esa distribución objetivo. Este enfoque se asemeja a los métodos MCMC en el sentido de que involucra muchas iteraciones para producir una muestra. Sin embargo, el modelo se define como la distribución de probabilidad producida por el paso final de la cadena. En este sentido, no hay aproximación inducida por el procedimiento iterativo. El enfoque introducido porSohl-Dicksteinet al.(2015) también está muy cerca de la interpretación generativa del codificador automático de eliminación de ruido

(sección 20.11.1). Al igual que con el codificador automático de eliminación de ruido, la inversión de difusión entrena a un operador de transición que intenta deshacer probabilísticamente el efecto de agregar algo de ruido. La diferencia es que la inversión de difusión requiere deshacer solo un paso del proceso de difusión, en lugar de viajar todo el camino de regreso a un punto de datos limpio. Esto aborda el siguiente dilema presente con el objetivo de probabilidad logarítmica de reconstrucción ordinaria de eliminar el ruido de los codificadores automáticos: con niveles pequeños de ruido, el alumno solo ve configuraciones cerca de los puntos de datos, mientras que con niveles grandes de ruido se le pide que haga un trabajo casi imposible (porque la distribución de eliminación de ruido es muy compleja y multimodal). Con el objetivo de inversión de difusión,

Otro enfoque para la generación de muestras es el **cálculo bayesiano aproximado** (ABC) marco (Frota et al., 1984). En este enfoque, las muestras se rechazan o modifican para que los momentos de las funciones seleccionadas de las muestras coincidan con los de la distribución deseada. Si bien esta idea utiliza los momentos de las muestras como en la coincidencia de momentos, es diferente de la coincidencia de momentos porque modifica las muestras mismas, en lugar de entrenar al modelo para que emita automáticamente muestras con los momentos correctos. Bachman y Precopa (2015) mostró cómo usar ideas de ABC en el contexto del aprendizaje profundo, usando ABC para dar forma a las trayectorias MCMC de GSN.

Esperamos que muchos otros enfoques posibles para el modelado generativo estén a la espera de ser descubiertos.

## 20.14 Evaluación de modelos generativos

Los investigadores que estudian modelos generativos a menudo necesitan comparar un modelo generativo con otro, generalmente para demostrar que un modelo generativo recién inventado es mejor para capturar alguna distribución que los modelos preexistentes.

Esto puede ser una tarea difícil y sutil. En muchos casos, en realidad no podemos evaluar la probabilidad logarítmica de los datos bajo el modelo, sino solo una aproximación. En estos casos, es importante pensar y comunicar con claridad exactamente lo que se está midiendo. Por ejemplo, supongamos que podemos evaluar una estimación estocástica de la probabilidad logarítmica del modelo A y un límite inferior determinista de la probabilidad logarítmica del modelo B. Si el modelo A obtiene una puntuación más alta que el modelo B, ¿cuál es mejor? Si nos preocupamos por determinar qué modelo tiene una mejor representación interna de la distribución, en realidad no podemos decirlo, a menos que tengamos alguna forma de determinar cuán flexible es el límite para el modelo B. Sin embargo, si nos importa qué tan bien podemos usar el modelo en la práctica, por ejemplo, para realizar la detección de anomalías, entonces es justo

decir que es preferible un modelo basado en un criterio específico de la tarea práctica de interés, por ejemplo, basado en ejemplos de prueba de clasificación y criterios de clasificación como precisión y memoria.

Otra sutileza de evaluar modelos generativos es que las métricas de evaluación a menudo son problemas de investigación difíciles en sí mismos. Puede ser muy difícil establecer que los modelos se comparan de manera justa. Por ejemplo, supongamos que usamos AIS para estimar  $\text{registro}_Z$  para calcular  $\text{registro}_{\text{pag}}(X)$  -  $\text{registro}_Z$  para un nuevo modelo que acabamos de inventar. Una implementación computacionalmente económica de AIS puede fallar en encontrar varios modos de distribución del modelo y subestimar  $Z$ , lo que hará que sobreestimemos  $\text{registro}_{\text{pag}}(X)$ . Por lo tanto, puede ser difícil saber si una estimación de alta probabilidad se debe a un buen modelo o a una mala implementación del AIS.

Otros campos del aprendizaje automático suelen permitir alguna variación en el preprocesamiento de los datos. Por ejemplo, al comparar la precisión de los algoritmos de reconocimiento de objetos, generalmente es aceptable preprocesar las imágenes de entrada de forma ligeramente diferente para cada algoritmo según el tipo de requisitos de entrada que tenga. El modelado generativo es diferente porque los cambios en el preprocesamiento, incluso los más pequeños y sutiles, son completamente inaceptables. Cualquier cambio en los datos de entrada cambia la distribución a capturar y altera fundamentalmente la tarea. Por ejemplo, multiplicar la entrada por 0,1 aumentará artificialmente la probabilidad por un factor de 10.

Los problemas con el preprocesamiento suelen surgir cuando se comparan modelos generativos en el conjunto de datos MNIST, uno de los puntos de referencia de modelado generativo más populares. MNIST consta de imágenes en escala de grises. Algunos modelos tratan las imágenes MNIST como puntos en un espacio vectorial real, mientras que otros las tratan como binarias. Sin embargo, otros tratan los valores de la escala de grises como probabilidades para muestras binarias. Es esencial comparar modelos de valor real solo con otros modelos de valor real y modelos de valor binario solo con otros modelos de valor binario. De lo contrario, las probabilidades medidas no están en el mismo espacio. Para los modelos con valores binarios, la probabilidad logarítmica puede ser como máximo cero, mientras que para los modelos con valores reales puede ser arbitrariamente alta, ya que es la medida de una densidad. Entre los modelos binarios, es importante comparar modelos usando exactamente el mismo tipo de binarización. Por ejemplo, podríamos binarizar un píxel gris a 0 o 1 mediante un umbral de 0,5, o dibujando una muestra aleatoria cuya probabilidad de ser 1 viene dada por la intensidad del píxel gris. Si usamos la binarización aleatoria, podríamos binarizar todo el conjunto de datos una vez, o podríamos dibujar un ejemplo aleatorio diferente para cada paso del entrenamiento y luego extraer varias muestras para su evaluación. Cada uno de estos tres esquemas produce números de probabilidad muy diferentes, y cuando se comparan diferentes modelos, es importante que ambos modelos usen el mismo esquema de binarización para el entrenamiento y la evaluación. De hecho, los investigadores que aplican un único método aleatorio o podemos dibujar un ejemplo aleatorio diferente para cada paso del entrenamiento y luego sacar varias muestras para evaluar. Cada uno de estos tres esquemas produce números de probabilidad muy diferentes, y cuando se comparan diferentes modelos, es importante que ambos modelos usen el mismo esquema de binarización para el entrenamiento y la evaluación. De hecho, los investigadores que aplican un único método aleatorio o podemos dibujar un ejemplo aleatorio diferente para cada paso del entrenamiento y luego sacar varias muestras para evaluar. Cada uno de estos tres esquemas produce números de probabilidad muy diferentes, y cuando se comparan diferentes modelos, es importante que ambos modelos usen el mismo esquema de binarización para el entrenamiento y la evaluación. De hecho, los investigadores que aplican un único método aleatorio

El paso de binarización comparte un archivo que contiene los resultados de la binarización aleatoria, de modo que no haya diferencias en los resultados según los diferentes resultados del paso de binarización.

Dado que uno de los objetivos de un modelo generativo es poder generar muestras realistas a partir de la distribución de datos, los profesionales suelen evaluar los modelos generativos mediante la inspección visual de las muestras. En el mejor de los casos, esto no lo hacen los propios investigadores, sino los sujetos experimentales que no conocen el origen de las muestras ([Dentón et al., 2015](#)). Desafortunadamente, es posible que un modelo probabilístico muy pobre produzca muestras muy buenas. Una práctica común para verificar si el modelo solo copia algunos de los ejemplos de entrenamiento se ilustra en la figura [16.1](#). La idea es mostrar para algunas de las muestras generadas su vecino más cercano en el conjunto de entrenamiento, según la distancia euclíadiana en el espacio de  $X$ . Esta prueba está destinada a detectar el caso en el que el modelo se ajusta demasiado al conjunto de entrenamiento y solo reproduce las instancias de entrenamiento. Incluso es posible adaptar y sobreajustar simultáneamente y aún así producir muestras que se vean bien individualmente. Imagine un modelo generativo entrenado con imágenes de perros y gatos que simplemente aprende a reproducir las imágenes de entrenamiento de los perros. Tal modelo claramente tiene un sobreajuste, porque no produce imágenes que no estaban en el conjunto de entrenamiento, pero también tiene un ajuste insuficiente, porque no asigna probabilidad a las imágenes de entrenamiento de los gatos. Sin embargo, un observador humano juzgaría que cada imagen individual de un perro es de alta calidad. En este ejemplo simple, sería fácil para un observador humano que puede inspeccionar muchas muestras determinar que los gatos están ausentes. En escenarios más realistas,

Dado que la calidad visual de las muestras no es una guía confiable, a menudo también evaluamos la probabilidad logarítmica que el modelo asigna a los datos de prueba, cuando esto es computacionalmente factible. Desafortunadamente, en algunos casos la probabilidad parece no medir ningún atributo del modelo que realmente nos interese. Por ejemplo, los modelos de valor real de MNIST pueden obtener una probabilidad arbitrariamente alta al asignar una varianza arbitrariamente baja a los píxeles de fondo que nunca cambian. Los modelos y algoritmos que detectan estas características constantes pueden obtener recompensas ilimitadas, aunque esto no es algo muy útil. El potencial para lograr un costo que se acerque al infinito negativo está presente para cualquier tipo de problema de máxima verosimilitud con valores reales, pero es especialmente problemático para los modelos generativos de MNIST porque muchos de los valores de salida son triviales de predecir.

[Teiset et al. \(2015\)](#) revisan muchos de los temas involucrados en la evaluación generativa

modelos, incluyendo muchas de las ideas descritas anteriormente. Destacan el hecho de que hay muchos usos diferentes de los modelos generativos y que la elección de la métrica debe coincidir con el uso previsto del modelo. Por ejemplo, algunos modelos generativos son mejores para asignar una alta probabilidad a los puntos más realistas, mientras que otros modelos generativos son mejores para asignar raramente una alta probabilidad a los puntos poco realistas. Estas diferencias pueden resultar de si un modelo generativo está diseñado para minimizar  $D_{KL}(p_{\text{datos}} \text{-} p_{\text{modelo}})$  o  $D_{KL}(p_{\text{modelo}} \text{-} p_{\text{datos}})$ , como se ilustra en la figura 3.6. Desafortunadamente, incluso cuando restringimos el uso de cada métrica a la tarea para la que es más adecuada, todas las métricas actualmente en uso continúan teniendo serias debilidades. Por lo tanto, uno de los temas de investigación más importantes en el modelado generativo no es solo cómo mejorar los modelos generativos, sino, de hecho, diseñar nuevas técnicas para medir nuestro progreso.

## 20.15 Conclusión

Entrenar modelos generativos con unidades ocultas es una forma poderosa de hacer que los modelos entiendan el mundo representado en los datos de entrenamiento dados. Aprendiendo un modelo  $p_{\text{modelo}}(X)$  y una representación  $p_{\text{modelo}}(h / X)$ , un modelo generativo puede proporcionar respuestas a muchos problemas de inferencia sobre las relaciones entre las variables de entrada en  $X$  y puede proporcionar muchas maneras diferentes de representar  $X$  tomando expectativas de  $h$  en diferentes capas de la jerarquía. Los modelos generativos prometen proporcionar a los sistemas de IA un marco para todos los diferentes conceptos intuitivos que necesitan comprender y la capacidad de razonar sobre estos conceptos frente a la incertidumbre. Esperamos que nuestros lectores encuentren nuevas formas de hacer que estos enfoques sean más poderosos y continúen el viaje hacia la comprensión de los principios que subyacen al aprendizaje y la inteligencia.

# Bibliografía

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, GS, Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. y Zheng, X. (2015). TensorFlow: aprendizaje automático a gran escala en sistemas heterogéneos. Software disponible en tensorflow.org.[25, 214,446](#)

Ackley, DH, Hinton, GE y Sejnowski, TJ (1985). Un algoritmo de aprendizaje para Máquinas de Boltzmann. *Ciencia cognitiva*,[9](#), 147-169.[570,654](#)

Alain, G. y Bengio, Y. (2013). Qué aprenden los codificadores automáticos regularizados de los datos generación de distribución. En *ICLR'2013, arXiv:1211.4246*.[507,513,514,521](#)

Alain, G., Bengio, Y., Yao, L., Éric Thibodeau-Laufer, Yosinski, J. y Vincent, P. (2015). GSNs: Redes estocásticas generativas. *arXiv:1503.05571*.[510,713](#)

Anderson, E. (1935). Los lirios de la península de Gaspé. *Boletín del Iris Americano Sociedad*,[59](#), 2-5.[21](#)

Ba, J., Mnih, V. y Kavukcuoglu, K. (2014). Reconocimiento de objetos múltiples con visual atención. *arXiv:1412.7755*.[691](#)

Bachman, P. y Precup, D. (2015). Redes estocásticas generativas variacionales con formación colaborativa. En *Actas de la 32.ª Conferencia Internacional sobre Aprendizaje Automático, ICML 2015, Lille, Francia, 6-11 de julio de 2015*, páginas 1964-1972.[717](#)

Bacon, P.-L., Bengio, E., Pineau, J. y Precup, D. (2015). Cálculo condicional en redes neuronales utilizando un enfoque de teoría de la decisión. En *II Jornadas Multidisciplinares de Aprendizaje por Refuerzo y Toma de Decisiones (RLDM 2015)*.[450](#)

Bagnell, JA y Bradley, DM (2009). Codificación dispersa diferenciable. En D. Koller, D. Schuurmans, Y. Bengio y L. Bottou, editores, *Avances en sistemas de procesamiento de información neuronal 21 (NIPS'08)*, páginas 113-120.[498](#)

- Bahdanau, D., Cho, K. y Bengio, Y. (2015). Traducción automática neuronal por jointly aprender a alinear y traducir. En *ICLR'2015, arXiv:1409.0473*.[25,101,397,418,420, 465, 475,476](#)
- Bahl, LR, Brown, P., de Souza, PV y Mercer, RL (1987). Reconocimiento de voz con modelos de Markov ocultos de parámetros continuos. *Informática, Habla y Lenguaje*,[2, 219–234.458](#)
- Baldi, P. y Hornik, K. (1989). Redes neuronales y análisis de componentes principales: Aprendiendo de ejemplos sin mínimos locales. *Redes neuronales*,[2, 53–58.286](#)
- Baldi, P., Brunak, S., Frasconi, P., Soda, G. y Pollastri, G. (1999). explotando el Pasado y futuro en la predicción de estructuras secundarias de proteínas. *Bioinformática*,[15\(11\), 937–946.395](#)
- Baldi, P., Sadowski, P. y Whiteson, D. (2014). Buscando partículas exóticas en física de alta energía con aprendizaje profundo. *Comunicaciones de la naturaleza*,[5.26](#)
- Ballard, DH, Hinton, GE y Sejnowski, TJ (1983). Cálculo de visión paralela. *Naturaleza*,[452](#)
- Barlow, HB (1989). Aprendizaje sin supervisión. *Computación neuronal*,[1, 295–311.147](#)
- Barrón, AE (1993). Límites de aproximación universales para superposiciones de un sigmoidal función. *Trans. IEEE. sobre teoría de la información*,[39, 930–945.199](#)
- Bartolomé, DJ (1987). *Modelos de variables latentes y análisis factorial*. Universidad de Oxford Prensa.[490](#)
- Basilevsky, A. (1994). *Análisis Factorial Estadístico y Métodos Relacionados: Teoría y Aplicaciones*. Wiley.[490](#)
- Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, IJ, Bergeron, A., Bouchard, N. y Bengio, Y. (2012). Theano: nuevas funciones y mejoras de velocidad. Taller de NIPS 2012 sobre aprendizaje profundo y aprendizaje de funciones no supervisado.[25,82,214, 222,446](#)
- Basu, S. y Christensen, J. (2013). Enseñando los límites de clasificación a los humanos. En *AAAI'2013*.[329](#)
- Baxter, J. (1995). Aprendizaje de representaciones internas. En *Actas de la 8ª Internacional Conferencia sobre Teoría del Aprendizaje Computacional (COLT'95)*, páginas 311–320, Santa Cruz, California. Prensa ACM.[245](#)
- Bayer, J. y Osendorfer, C. (2014). Aprendizaje de redes recurrentes estocásticas. *ArXiv impresiones electrónicas*.[265](#)
- Becker, S. y Hinton, G. (1992). Una red neuronal autoorganizada que descubre superficies en estereogramas de puntos aleatorios. *Naturaleza*,[355, 161–163.541](#)

- Behnke, S. (2001). Aprendizaje de la reconstrucción iterativa de imágenes en la abstracción neuronal pirámide. En *t. J. Inteligencia Computacional y Aplicaciones*, **1**(4), 427–438.[515](#)
- Beiu, V., Quintana, JM y Avedillo, MJ (2003). Implementaciones VLSI de umbral logic-una encuesta comprensiva. *Redes neuronales, transacciones IEEE en*, **14**(5), 1217–1243.[451](#)
- Belkin, M. y Niyogi, P. (2002). Automapas laplacianos y técnicas espectrales para incrustación y agrupamiento. En T. Dietterich, S. Becker y Z. Ghahramani, editores, *Avances en sistemas de procesamiento de información neuronal 14 (NIPS'01)*, Cambridge, MA. Prensa del MIT.[244](#)
- Belkin, M. y Niyogi, P. (2003). Automapas laplacianos para reducción de dimensionalidad y representación de datos. *Computación neuronal*, **15**(6), 1373–1396.[164,518](#)
- Bengio, E., Bacon, P.-L., Pineau, J. y Precup, D. (2015a). Cálculo condicional en redes neuronales para modelos más rápidos. arXiv:1511.06297.[450](#)
- Bengio, S. y Bengio, Y. (2000a). Asumiendo la maldición de la dimensionalidad en conjunto distribuciones utilizando redes neuronales. *IEEE Transactions on Neural Networks, número especial sobre minería de datos y descubrimiento de conocimientos*, **11**(3), 550–557.[707](#)
- Bengio, S., Vinyals, O., Jaitly, N. y Shazeer, N. (2015b). Muestreo programado para predicción de secuencias con redes neuronales recurrentes. Informe técnico, arXiv:1506.03099. [384](#)
- Bengio, Y. (1991). *Redes Neuronales Artificiales y su Aplicación al Reconocimiento de Secuencias*. Doctor. tesis, Universidad McGill, (Ciencias de la Computación), Montreal, Canadá.[407](#)
- Bengio, Y. (2000). Optimización basada en gradiente de hiperparámetros. *Computación neuronal*, **12**(8), 1889–1900.[435](#)
- Bengio, Y. (2002). Nuevos modelos de lenguaje probabilístico distribuido. Informe Técnico 1215, Departamento IRO, Universidad de Montreal.[467](#)
- Bengio, Y. (2009). *Aprendizaje de arquitecturas profundas para IA*. Ahora Editores.[201,622](#)
- Bengio, Y. (2013). Aprendizaje profundo de representaciones: mirar hacia adelante. En *Estadístico Procesamiento del lenguaje y el habla*, tomo 7978 de *Apuntes de clase en informática*, páginas 1–37. Springer, también en arXiv en <http://arxiv.org/abs/1305.0445>.[448](#)
- Bengio, Y. (2015). La inferencia temprana en modelos basados en energía se aproxima a la retropropagación. Informe técnico arXiv:1510.02777, Universidad de Montreal.[656](#)
- Bengio, Y. y Bengio, S. (2000b). Modelado de datos discretos de alta dimensión con multi-redes neuronales de capa. En *PINZAS 12*, páginas 400–406. Prensa del MIT.[705,707,708,710](#)
- Bengio, Y. y Delalleau, O. (2009). Justificación y generalización de la divergencia contrastiva. *Computación neuronal*, **21**(6), 1601–1621.[513,611](#)

- Bengio, Y. y Grandvalet, Y. (2004). Sin estimador imparcial de la varianza de k-fold validación cruzada. En S. Thrun, L. Saul y B. Schölkopf, editores, *Avances en sistemas de procesamiento de información neuronal 16 (NIPS'03)*, Cambridge, MA. MIT Press, Cambridge. 122
- Bengio, Y. y LeCun, Y. (2007). Escalando los algoritmos de aprendizaje hacia la IA. En *Gran escala Máquinas de núcleo*. 19
- Bengio, Y. y Monperrus, M. (2005). Aprendizaje múltiple tangente no local. En L. Saúl, Y. Weiss y L. Bottou, editores, *Avances en sistemas de procesamiento de información neuronal 17 (NIPS'04)*, páginas 129–136. Prensa del MIT. 160, 519
- Bengio, Y. y Sénécal, J.-S. (2003). Entrenamiento rápido de redes neuronales probabilísticas por muestreo de importancia. En *Actas de AISTATS 2003*. 470
- Bengio, Y. y Sénécal, J.-S. (2008). Muestreo de importancia adaptativo para acelerar el entrenamiento de un modelo de lenguaje neuronal probabilístico. *Trans. IEEE. Redes neuronales*, 19(4), 713–722. 470
- Bengio, Y., De Mori, R., Flammia, G. y Kompe, R. (1991). Motivado fonéticamente parámetros acústicos para el reconocimiento continuo de voz utilizando redes neuronales artificiales. En *Actas de EuroSpeech'91*. 27, 459
- Bengio, Y., De Mori, R., Flammia, G. y Kompe, R. (1992). Red neuronal-Gaussiana mezcla híbrida para reconocimiento de voz o estimación de densidad. En *PINZAS 4*, páginas 175–182. Morgan Kaufman. 459
- Bengio, Y., Frasconi, P. y Simard, P. (1993). El problema del aprendizaje a largo plazo dependencias en redes recurrentes. En *Conferencia internacional IEEE sobre redes neuronales*, páginas 1183–1195, San Francisco. Prensa IEEE. (papel invitado). 403
- Bengio, Y., Simard, P. y Frasconi, P. (1994). Aprender dependencias a largo plazo con el descenso de gradiente es difícil. *IEEE Tr. Redes neuronales*. 18, 401, 403, 411
- Bengio, Y., Latendresse, S. y Dugas, C. (1999). Aprendizaje basado en gradientes de hiper-parámetros Conferencia de aprendizaje, Snowbird. 435
- Bengio, Y., Ducharme, R. y Vincent, P. (2001). Un modelo de lenguaje probabilístico neural. En TK Leen, TG Dietterich y V. Tresp, editores, *NIPS'2000*, páginas 932–938. Prensa del MIT. 18, 447, 464, 466, 472, 477, 482
- Bengio, Y., Ducharme, R., Vincent, P. y Jauvin, C. (2003). Una probabilidad neural modelo de lenguaje/*MLR*, 3, 1137–1155. 466, 472
- Bengio, Y., Le Roux, N., Vincent, P., Delalleau, O. y Marcotte, P. (2006a). Convexo Redes neuronales. En *NIPS'2005*, páginas 123–130. 258
- Bengio, Y., Delalleau, O. y Le Roux, N. (2006b). La maldición de las funciones altamente variables para máquinas con núcleo local. En *NIPS'2005*. 158

- Bengio, Y., Larochelle, H. y Vincent, P. (2006c). Ventanas Parzen múltiples no locales. En *NIPS'2005*. Prensa del MIT.[160,520](#)
- Bengio, Y., Lamblin, P., Popovici, D. y Larochelle, H. (2007). Codicioso por capas formación de redes profundas. En *NIPS'2006*.[14,19,201,323,324,528,530](#)
- Bengio, Y., Louradour, J., Collobert, R. y Weston, J. (2009). Aprendizaje del currículo. En *ICML '09*.[328](#)
- Bengio, Y., Mesnil, G., Dauphin, Y. y Rifai, S. (2013a). Mejor mezcla a través de deep representaciones. En *ICML'2013*.[604](#)
- Bengio, Y., Léonard, N. y Courville, A. (2013b). Estimación o propagación de gradientes a través de neuronas estocásticas para el cálculo condicional. arXiv:1308.3432.[448,450,689,691](#)
- Bengio, Y., Yao, L., Alain, G. y Vincent, P. (2013c). Eliminación automática de ruido generalizada codificadores como modelos generativos. En *NIPS'2013*.[507,711,714](#)
- Bengio, Y., Courville, A. y Vincent, P. (2013d). Aprendizaje de representaciones: una revisión y nuevas perspectivas. *Trans. IEEE. Análisis de patrones e inteligencia artificial (PAMI)*, **35**(8), 1798–1828.[555](#)
- Bengio, Y., Thibodeau-Laufer, E., Alain, G. y Yosinski, J. (2014). generativo profundo redes estocásticas entrenables por backprop. En *ICML'2014*.[711,712,713,714,715](#)
- Bennett, C. (1976). Estimación eficiente de las diferencias de energía libre a partir de datos de Monte Carlo. *Revista de Física Computacional*, **22**(2), 245–268.[628](#)
- Bennett, J. y Lanning, S. (2007). El premio Netflix.[479](#)
- Berger, AL, Della Pietra, VJ y Della Pietra, SA (1996). Una entropía máxima Acercamiento al procesamiento del lenguaje natural. *Ligüística computacional*, **22**, 39–71.[473](#)
- Berglund, M. y Raiko, T. (2013). Gradiante estocástico estimar la varianza en contraste divergencia y divergencia contrastiva persistente. *CoRR,abs/1312.6002*.[614](#)
- Bergstra, J. (2011). *Incorporación de células complejas en redes neuronales para patrones Clasificación*. Doctor. tesis, Université de Montreal.[255](#)
- Bergstra, J. y Bengio, Y. (2009). Rasgos lentos y descorrelacionados para el complejo de preentrenamiento Redes similares a células. En *NIPS'2009*.[494](#)
- Bergstra, J. y Bengio, Y. (2012). Búsqueda aleatoria para la optimización de hiperparámetros./ *Resolución de aprendizaje automático*, **13**, 281–305.[433,434,435](#)
- Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D. y Bengio, Y. (2010). Theano: un compilador de expresiones matemáticas de CPU y GPU. En *proc. SciPy*.[25,82,214,222,446](#)

- Bergstra, J., Bardenet, R., Bengio, Y. y Kégl, B. (2011). Algoritmos para hiperparámetro mejoramiento. En *NIPS'2011*.[436](#)
- Berkes, P. y Wiskott, L. (2005). El análisis lento de características produce un rico repertorio de complejas propiedades de la celda. *Diario de la visión*, **5**(6), 579–602.[495](#)
- Bertsekas, DP y Tsitsiklis, J. (1996). *Programación Neuro-Dinámica*. Atenea Científica. [106](#)
- Besag, J. (1975). Análisis estadístico de datos no reticulares. *el estadístico*, **24**(3), 179–195. [615](#)
- Obispo, CM (1994). Redes de densidad de mezcla.[189](#)
- Obispo, CM (1995a). Regularización y control de complejidad en redes feed-forward. En *Actas de la Conferencia Internacional sobre Redes Neuronales Artificiales ICANN'95*, volumen 1, página 141–148.[242,250](#)
- Obispo, CM (1995b). El entrenamiento con ruido es equivalente a la regularización de Tikhonov. *Computación neuronal*, **7**(1), 108–116.[242](#)
- Obispo, CM (2006). *Reconocimiento de patrones y aprendizaje automático*. Saltador. [98,146](#)
- Blum, AL y Rivest, RL (1992). El entrenamiento de una red neuronal de 3 nodos es NP-completo. [293](#)
- Blumer, A., Ehrenfeucht, A., Haussler, D. y Warmuth, MK (1989). Capacidad de aprendizaje y la dimensión Vapnik-Chervonenkis. *Diario de la ACM*, **36**(4), 929–865.[114](#)
- Bonnet, G. (1964). Transformations des signaux aleatoires à travers les systèmes non Lineaires sans mémoire. *Annales des Télécommunications*, **19**(9–10), 203–220.[689](#)
- Bordes, A., Weston, J., Collobert, R. y Bengio, Y. (2011). Aprendizaje estructurado incrustaciones de bases de conocimiento. En *AAAI 2011*.[484](#)
- Bordes, A., Glorot, X., Weston, J. y Bengio, Y. (2012). Aprendizaje conjunto de palabras y representaciones de significado para el análisis semántico de texto abierto. *AISTATS'2012*.[401,484,485](#)
- Bordes, A., Glorot, X., Weston, J. y Bengio, Y. (2013a). Una energía de coincidencia semántica Función para el aprendizaje con datos multirelacionales. *Aprendizaje automático: número especial sobre la semántica del aprendizaje*.[483](#)
- Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. y Yakhnenko, O. (2013b). Traducir incrustaciones para modelar datos multirelacionales. En C. Burges, L. Bottou, M. Welling, Z. Ghahramani y K. Weinberger, editores, *Avances en los sistemas de procesamiento de información neuronal* 26, páginas 2787–2795. Curran Associates, Inc.[484](#)
- Bornstein, J. y Bengio, Y. (2015). *arXiv:1406.2751*.[693](#) Despertar-dormir reponderado. En *ICLR'2015*,

- Bornstein, J., Shabanian, S., Fischer, A. y Bengio, Y. (2015). Entrenamiento bidireccional máquinas Helmholtz. Informe técnico, arXiv:1506.03877.[693](#)
- Boser, BE, Guyon, IM y Vapnik, VN (1992). Un algoritmo de entrenamiento para optimizadores de margen mal. En *COLT '92: Actas del quinto taller anual sobre teoría del aprendizaje computacional*, páginas 144–152, Nueva York, NY, EE. UU. ACM.[18,141](#)
- Bottou, L. (1998). Algoritmos en línea y aproximaciones estocásticas. En D. Saad, editor, *Aprendizaje en línea en redes neuronales*. Prensa de la Universidad de Cambridge, Cambridge, Reino Unido.[296](#)
- Bottou, L. (2011). Del aprendizaje automático al razonamiento automático. Reporte técnico, arXiv.1102.1808.[401](#)
- Bottou, L. (2015). Redes neuronales multicapa. Escuela de verano de aprendizaje profundo.[440](#)
- Bottou, L. y Bousquet, O. (2008). Las ventajas y desventajas del aprendizaje a gran escala. En *NIPS'2008*.  
[282,295](#)
- Boulanger-Lewandowski, N., Bengio, Y. y Vincent, P. (2012). Modelado temporal dependencias en secuencias de alta dimensión: aplicación a la generación y transcripción de música polifónica. En *ICML '12*.[685,686](#)
- Boureau, Y., Ponce, J. y LeCun, Y. (2010). Un análisis teórico de la agrupación de características en algoritmos de visión. En *proc. Conferencia Internacional sobre Aprendizaje Automático (ICML'10)*.[345](#)
- Boureau, Y., Le Roux, N., Bach, F., Ponce, J. y LeCun, Y. (2011). Pregunta a los lugareños: agrupación local multidireccional para el reconocimiento de imágenes. En *proc. Congreso Internacional de Visión por Computador (ICCV'11)*. IEEE.[345](#)
- Bourlard, H. y Kamp, Y. (1988). Autoasociación por perceptrones multicapa y valor singular de descomposición. *Cibernética Biológica*,[59](#), 291–294.[502](#)
- Bourlard, H. y Wellekens, C. (1989). Discriminación de patrones de voz y multicapa. perceptrones. *Habla y lenguaje informático*,[3](#), 1–19.[459](#)
- Boyd, S. y Vandenberghe, L. (2004). *Optimización convexa*. Universidad de Cambridge Press, Nueva York, NY, Estados Unidos.[93](#)
- Brady, ML, Raghavan, R. y Slawny, J. (1989). La retropropagación no logra separarse donde los perceptrones tienen éxito. *Transacciones IEEE en circuitos y sistemas*,[36](#), 665–674. [284](#)
- Brakel, P., Stroobandt, D. y Schrauwen, B. (2013). Entrenamiento de modelos basados en energía para imputación de series de tiempo. *Revista de investigación de aprendizaje automático*,[14](#), 2771–2797.[674, 698](#)
- Marca, M. (2003). Graficando una variedad. En *NIPS'2002*, páginas 961–968. Prensa del MIT.[164, 518](#)

- Breiman, L. (1994). Predictores de embolsado. *Aprendizaje automático*, **24**(2), 123–140. [256](#)
- Breiman, L., Friedman, JH, Olshen, RA y Stone, CJ (1984). *Clasificación y Árboles de regresión*. Grupo internacional de Wadsworth, Belmont, CA. [146](#)
- Brida, JS (1990). Alphanets: una arquitectura de red 'neuronal' recurrente con un Interpretación del modelo de Markov. *Comunicación del habla*, **9**(1), 83–92. [186](#)
- Briggman, K., Denk, W., Seung, S., Helmstaedter, MN y Turaga, SC (2009). Maximin affinity learning of image segmentation. En *NIPS'2009*, páginas 1865–1873. [360](#)
- Brown, PF, Cocke, J., Pietra, SAD, Pietra, VJD, Jelinek, F., Lafferty, JD, Mercer, RL y Roossin, PS (1990). Un enfoque estadístico para la traducción automática. *Ligüística computacional*, **dieciséis**(2), 79–85. [21](#)
- Brown, PF, Pietra, VJD, DeSouza, PV, Lai, JC y Mercer, RL (1992). Clase-basedonorte-Modelos gramaticales del lenguaje natural. *Ligüística computacional*, **18**, 467–479. [463](#)
- Bryson, A. y Ho, Y. (1969). *Control óptimo aplicado: optimización, estimación y control*. Pub Blaisdell. Co. [225](#)
- Bryson, Jr., AE y Denham, WF (1961). Un método de ascenso más empinado para resolver Problemas de programación óptima. Informe técnico BR-1303, Raytheon Company, División de Misiles y Espacio. [225](#)
- Buciluă, C., Caruana, R. y Niculescu-Mizil, A. (2006). Modelo de compresión. En *Actas de la 12.ª conferencia internacional ACM SIGKDD sobre descubrimiento de conocimiento y minería de datos*, páginas 535–541. ACM. [448](#)
- Burda, Y., Grosse, R. y Salakhutdinov, R. (2015). Codificadores automáticos ponderados por importancia. *preimpresión de arXiv arXiv:1509.00519*. [698](#)
- Cai, M., Shi, Y. y Liu, J. (2013). Redes neuronales maxout profundas para el reconocimiento de voz. En *Reconocimiento y comprensión automáticos del habla (ASRU), Taller IEEE 2013 sobre*, páginas 291–296. IEEE. [194](#)
- Carreira-Perpiñan, MA y Hinton, GE (2005). Sobre el aprendizaje divergente contrastivo. En RG Cowell y Z. Ghahramani, editores, *Actas del Décimo Taller Internacional sobre Inteligencia Artificial y Estadísticas (AISTATS'05)*, páginas 33–40. Sociedad de Inteligencia Artificial y Estadística. [611](#)
- Caruana, R. (1993). Aprendizaje conexionista multitarea. En *proc. 1993 Modelos conexionistas Escuela de Verano*, páginas 372–379. [244](#)
- Cauchy, A. (1847). Méthode générale pour la résolution de systèmes d'équations simultanées. En *Compte rendu des seances de l'académie des sciences*, páginas 536–538. [83](#), [225](#)

- Cayton, L. (2005). Algoritmos para el aprendizaje múltiple. Informe técnico CS2008-0923, UCSD.[164](#)
- Chandola, V., Banerjee, A. y Kumar, V. (2009). Detección de anomalías: una encuesta. *MCA encuestas informáticas (CSUR)*, **41**(3), 15.[102](#)
- Chapelle, O., Weston, J. y Schölkopf, B. (2003). Núcleos de clúster para semi-supervisados aprendiendo. En S. Becker, S. Thrun y K. Obermayer, editores, *Avances en sistemas de procesamiento de información neuronal 15 (NIPS'02)*, páginas 585–592, Cambridge, MA. Prensa del MIT.[244](#)
- Chapelle, O., Schölkopf, B. y Zien, A., editores (2006). *Aprendizaje semisupervisado*. MIT Prensa, Cambridge, MA.[244,541](#)
- Chellapilla, K., Puri, S. y Simard, P. (2006). Neural convolucional de alto rendimiento Redes para el Tratamiento de Documentos. En Guy Lorette, editor, *Décimo Taller Internacional sobre Fronteras en el Reconocimiento de Escritura*, La Baule (Francia). Universidad de Rennes 1, Suvisoft. <http://www.suvisoft.com>.[24,27,445](#)
- Chen, B., Ting, J.-A., Marlin, BM y de Freitas, N. (2010). Aprendizaje profundo de invariantes características espacio-temporales del video. NIPS\*2010 Taller de aprendizaje profundo y aprendizaje de funciones no supervisado.[360](#)
- Chen, SF y Goodman, JT (1999). Un estudio empírico de técnicas de suavizado para modelado del lenguaje. *Informática, Habla y Lenguaje*, **13**(4), 359–393.[462,463,473](#)
- Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y. y Temam, O. (2014a). Diana Nao: Un acelerador de alto rendimiento y tamaño reducido para el aprendizaje automático ubicuo. En *Actas de la 19ª conferencia internacional sobre soporte arquitectónico para lenguajes de programación y sistemas operativos*, páginas 269–284. ACM.[451](#)
- Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., y Zhang, Z. (2015). MXNet: una biblioteca de aprendizaje automático flexible y eficiente para sistemas distribuidos heterogéneos. *preimpresión de arXiv arXiv:1512.01274*.[25](#)
- Chen, Y., Luo, T., Liu, S., Zhang, S., He, L., Wang, J., Li, L., Chen, T., Xu, Z., Sun, N., et al.(2014b). DaDianNao: una supercomputadora de aprendizaje automático. En *Microarquitectura (MICRO), 2014 47º Simposio Internacional Anual IEEE/ACM sobre*, páginas 609–622. IEEE.[451](#)
- Chilimbi, T., Suzue, Y., Apacible, J. y Kalyanaraman, K. (2014). Proyecto Adán: Construyendo un sistema de entrenamiento de aprendizaje profundo eficiente y escalable. En *11º Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos (OSDI'14)*.[447](#)
- Cho, K., Raiko, T. e Ilin, A. (2010). El templado paralelo es eficiente para el aprendizaje restringido Máquinas de Boltzmann. En *IJCNN'2010*.[603,614](#)

- Cho, K., Raiko, T. e Ilin, A. (2011). Gradiente mejorado y tasa de aprendizaje adaptativo para Máquinas de Boltzmann con restricciones de entrenamiento. En *ICML'2011*, páginas 105–112.[674](#)
- Cho, K., van Merriënboer, B., Gulcehre, C., Bougares, F., Schwenk, H. y Bengio, Y. (2014a). Aprendizaje de representaciones de frases utilizando el codificador-decodificador RNN para la traducción automática estadística. En *Actas de los métodos empíricos en el procesamiento del lenguaje natural (EMNLP 2014)*.[397,474,475](#)
- Cho, K., Van Merriënboer, B., Bahdanau, D. y Bengio, Y. (2014b). en el properties de la traducción automática neuronal: enfoques de codificador-decodificador. *Impresiones electrónicas ArXiv*, [abs/1409.1259](#).[412](#)
- Choromanska, A., Henaff, M., Mathieu, M., Arous, GB y LeCun, Y. (2014). El superficie de pérdida de redes multicapa.[285,286](#)
- Chorowski, J., Bahdanau, D., Cho, K. y Bengio, Y. (2014). Continuo de extremo a extremo Reconocimiento de voz mediante NN recurrente basado en la atención: Primeros resultados. arXiv:1412.1602. [461](#)
- Christianson, B. (1992). Arpilleras automáticas por acumulación inversa. *Revista IMA de Análisis numérico*, **12**(2), 135–150.[224](#)
- Chrupala, G., Kadar, A. y Alishahi, A. (2015). Aprendizaje de idiomas a través de imágenes. arXiv 1506.03694.[412](#)
- Chung, J., Gulcehre, C., Cho, K. y Bengio, Y. (2014). Evaluación empírica de gated redes neuronales recurrentes en el modelado de secuencias. Taller de aprendizaje profundo NIPS'2014, arXiv 1412.3555.[412,460](#)
- Chung, J., Gülc̄ehre, Ç., Cho, K. y Bengio, Y. (2015a). Retroalimentación cerrada recurrente Redes neuronales. En *ICML'15*.[412](#)
- Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. y Bengio, Y. (2015b). A modelo de variable latente recurrente para datos secuenciales. En *NIPS'2015*.[698](#)
- Ciresan, D., Meier, U., Masci, J. y Schmidhuber, J. (2012). Neural profundo multicolumna Red de clasificación de señales de tráfico. *Redes neuronales*, **32**, 333–338.[23,201](#)
- Ciresan, DC, Meier, U., Gambardella, LM y Schmidhuber, J. (2010). profundo grande redes neuronales simples para el reconocimiento de dígitos escritos a mano. *Computación neuronal*, **22**, 1–14. [24,27,446](#)
- Coates, A. y Ng, AY (2011). La importancia de la codificación frente al entrenamiento con escasa codificación y cuantificación vectorial. En *ICML'2011*.[27,256,498](#)
- Coates, A., Lee, H. y Ng, AY (2011). Un análisis de las redes de una sola capa en aprendizaje de características no supervisado. En *Actas de la Decimotercera Conferencia Internacional sobre Inteligencia Artificial y Estadísticas (AISTATS 2011)*.[363,364,455](#)

- Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B. y Andrew, N. (2013). Aprendizaje profundo con sistemas COTS HPC. En S. Dasgupta y D. McAllester, editores, *Actas de la 30.ª Conferencia Internacional sobre Aprendizaje Automático (ICML-13)*, volumen 28 (3), páginas 1337–1345. Actas de talleres y conferencias de la JMLR.[24,27,364,447](#)
- Cohen, N., Sharir, O. y Shashua, A. (2015). Sobre el poder expresivo del aprendizaje profundo: Un análisis tensorial. arXiv:1509.05009.[554](#)
- Collobert, R. (2004). *Aprendizaje automático a gran escala*. Doctor. tesis, Université de Paris VI, LIP6.[197](#)
- Collobert, R. (2011). Aprendizaje profundo para un análisis discriminativo eficiente. En *AISTATS'2011*.[101,477](#)
- Collobert, R. y Weston, J. (2008a). Una arquitectura unificada para el procesamiento del lenguaje natural: Redes neuronales profundas con aprendizaje multitarea. En *ICML'2008*.[471,477](#)
- Collobert, R. y Weston, J. (2008b). Una arquitectura unificada para el lenguaje natural Procesamiento: redes neuronales profundas con aprendizaje multitarea. En *ICML'2008*.[535](#)
- Collobert, R., Bengio, S. y Bengio, Y. (2001). Una mezcla paralela de SVM para muy problemas a gran escala. Informe Técnico IDIAP-RR-01-12, IDIAP.[450](#)
- Collobert, R., Bengio, S. y Bengio, Y. (2002). Mezcla paralela de SVM para muy grandes problemas de escala. *Computación neuronal*, **14**(5), 1105–1114.[450](#)
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. y Kuksa, P. (2011a). Procesamiento del lenguaje natural (casi) desde cero. *El diario de investigación de aprendizaje automático*, **12**, 2493–2537.[328,477,535,536](#)
- Collobert, R., Kavukcuoglu, K. y Farabet, C. (2011b). Torch7: un entorno tipo Matlab para el aprendizaje automático. En *BigLearn, taller de NIPS*.[25,214,446](#)
- Comon, P. (1994). Análisis de componentes independientes: ¿un nuevo concepto? *Procesamiento de la señal*, **36**, 287–314.[491](#)
- Cortés, C. y Vapnik, V. (1995). Admite redes vectoriales. *Aprendizaje automático*, **20**, 273–297.[18,141](#)
- Couprise, C., Farabet, C., Najman, L. y LeCun, Y. (2013). Segmentación semántica interior utilizando información de profundidad. En *Conferencia Internacional sobre Representaciones de Aprendizaje (ICLR2013)*.[23,201](#)
- Courbariaux, M., Bengio, Y. y David, J.-P. (2015). Aritmética de baja precisión para profundidad aprendiendo. En *Arxiv:1412.7024, Taller ICLR'2015*.[452](#)
- Courville, A., Bergstra, J. y Bengio, Y. (2011). Modelos no supervisados de imágenes por RBM de punta y losa. En *ICML'11*.[561,681](#)

- Courville, A., Desjardins, G., Bergstra, J. y Bengio, Y. (2014). La espiga y losa RBM y extensiones para distribuciones de datos discretos y dispersos. *Análisis de patrones e inteligencia artificial, transacciones IEEE en*, **36**(9), 1874–1887.[682](#)
- Cubierta, TM y Thomas, JA (2006). *Elementos de la teoría de la información, 2ª edición.* Wiley-Interscience.[73](#)
- Cox, D. y Pinto, N. (2011). Más allá de las características simples: una búsqueda de características a gran escala enfoque para el reconocimiento facial sin restricciones. En *Reconocimiento automático de rostros y gestos y talleres (FG 2011), Conferencia internacional IEEE 2011 sobre*, páginas 8–15. IEEE. [363](#)
- Cramér, H. (1946). *Métodos matemáticos de estadística..* Prensa de la Universidad de Princeton.[135](#), [295](#)
- Crick, FHC y Mitchison, G. (1983). La función del sueño del sueño. *Naturaleza*, **304**, 111–114.[609](#)
- Cybenko, G. (1989). Aproximación por superposiciones de una función sigmoidal. *Matemáticas de Control, Señales y Sistemas*, **2**, 303–314.[198](#)
- Dahl, GE, Ranzato, M., Mohamed, A. y Hinton, GE (2010). Reconocimiento de teléfono con la máquina de Boltzmann restringida por covarianza media. En *NIPS'2010*.[23](#)
- Dahl, GE, Yu, D., Deng, L. y Acero, A. (2012). Profundo preentrenado dependiente del contexto Redes neuronales para el reconocimiento de voz de gran vocabulario. *Transacciones IEEE sobre procesamiento de audio, voz y lenguaje*, **20**(1), 33–42.[459](#)
- Dahl, GE, Sainath, TN y Hinton, GE (2013). Mejorando las redes neuronales profundas para LVCSR utilizando unidades lineales rectificadas y dropout. En *ICASSP'2013*.[460](#)
- Dahl, GE, Jaitly, N. y Salakhutdinov, R. (2014). Redes neuronales multitarea para Predicciones QSAR. arXiv:1406.1231.[26](#)
- Dauphin, Y. y Bengio, Y. (2013). Coincidencia de relación estocástica de RBM para escasa entradas de alta dimensión. En *NIPS26*. Fundación NIP.[619](#)
- Dauphin, Y., Glorot, X. y Bengio, Y. (2011). Aprendizaje a gran escala de incrustaciones con muestreo de reconstrucción. En *ICML'2011*.[471](#)
- Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S. y Bengio, Y. (2014). Identificar y atacar el problema del punto de silla en la optimización no convexa de alta dimensión. En *NIPS'2014*.[285](#),[286](#),[288](#)
- Davis, A., Rubinstein, M., Wadhwa, N., Mysore, G., Durand, F. y Freeman, WT (2014). El micrófono visual: Recuperación pasiva del sonido del video. *Transacciones ACM en Gráficos (Proc. SIGGRAPH)*, **33**(4), 79:1–79:10.[452](#)

- Dayán, P. (1990). Comparación de refuerzos. En *Modelos conexionistas: Procedimientos de la Escuela de Verano Conexiónista de 1990*, San Mateo, CA.[691](#)
- Dayan, P. y Hinton, GE (1996). Variedades de máquina Helmholtz. *Redes neuronales*, **9**(8), 1385–1403.[693](#)
- Dayan, P., Hinton, GE, Neal, RM y Zemel, RS (1995). La máquina de Helmholtz. *Computación neuronal*, **7**(5), 889–904.[693](#)
- Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Le, Q., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K. y Ng, AY (2012). Redes profundas distribuidas a gran escala. En *NIPS'2012*.[25,447](#)
- Dean, T. y Kanazawa, K. (1989). Un modelo para razonar sobre la persistencia y la causalidad. *Inteligencia Computacional*, **5**(3), 142–150.[662](#)
- Deerwester, S., Dumais, ST, Furnas, GW, Landauer, TK y Harshman, R. (1990). Indexación por análisis semántico latente. *Revista de la Sociedad Estadounidense de Ciencias de la Información*, **41**(6), 391–407.[477,482](#)
- Delalleau, O. y Bengio, Y. (2011). Redes de suma-producto superficiales vs. profundas. En *PINZAS*.[19,554](#)
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. y Fei-Fei, L. (2009). ImageNet: A Base de datos de imágenes jerárquicas a gran escala. En *CVPR09*.[21](#)
- Deng, J., Berg, AC, Li, K. y Fei-Fei, L. (2010a). ¿Qué significa clasificar más de 10.000 categorías de imágenes nos dicen? En *Actas de la 11.ª Conferencia Europea sobre Visión por Computador: Parte V, ECCV'10*, páginas 71–84, Berlín, Heidelberg. Springer-Verlag. [21](#)
- Deng, L. y Yu, D. (2014). Aprendizaje profundo: métodos y aplicaciones. *Fundaciones y Tendencias en el procesamiento de señales*.[460](#)
- Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed, A. y Hinton, G. (2010b). Binario codificación de espectrogramas de voz utilizando un codificador automático profundo. En *Interdiscurso 2010*, Makuhari, Chiba, Japón.[23](#)
- Denil, M., Bazzani, L., Larochelle, H. y de Freitas, N. (2012). Saber dónde asistir con arquitecturas profundas para el seguimiento de imágenes. *Computación neuronal*, **24**(8), 2151–2184.[367](#)
- Denton, E., Chintala, S., Szlam, A. y Fergus, R. (2015). Modelos de imágenes generativas profundas utilizando una pirámide laplaciana de redes adversarias. *PINZAS*.[702,719](#)
- Desjardins, G. y Bengio, Y. (2008). Evaluación empírica de RBM convolucionales para visión. Informe técnico 1327, Departamento de Informática y de Investigación Operativa, Universidad de Montreal.[683](#)

- Desjardins, G., Courville, AC, Bengio, Y., Vincent, P. y Delalleau, O. (2010). Cadena Markov templada Monte Carlo para entrenamiento de máquinas Boltzmann restringidas. En *Congreso Internacional de Inteligencia Artificial y Estadística*, páginas 145–152.[603, 614](#)
- Desjardins, G., Courville, A. y Bengio, Y. (2011). Sobre el seguimiento de la función de partición. En *NIPS'2011*.[629](#)
- Desjardins, G., Simonyan, K., Pascanu, R., et al.(2015). Redes neuronales naturales. En *Avances en sistemas de procesamiento de información neuronal*, páginas 2062–2070.[320](#)
- Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. y Makhoul, J. (2014). Rápido y robustos modelos conjuntos de redes neuronales para la traducción automática estadística. En *proc. ACL'2014*.[473](#)
- Devroye, L. (2013). *Generación de variantes aleatorias no uniformes*. SpringerLink: Bücher. Springer Nueva York.[694](#)
- DiCarlo, JJ (2013). Mecanismos subyacentes al reconocimiento visual de objetos: humanos vs. neuronas vs máquinas. Tutorial NIPS.[26,366](#)
- Dinh, L., Krueger, D. y Bengio, Y. (2014). NIZA: componentes independientes no lineales Estimacion. arXiv:1410.8516.[493](#)
- Donahue, J., Hendricks, LA, Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. y Darrell, T. (2014). Redes convolucionales recurrentes a largo plazo para reconocimiento y descripción visual. arXiv:1411.4389.[102](#)
- Donoho, DL y Grimes, C. (2003). Automapas hessianos: nueva incrustación localmente lineal Técnicas para datos de alta dimensión. Informe Técnico 2003-08, Departamento de Estadística, Universidad de Stanford.[164,519](#)
- Dosovitskiy, A., Springenberg, JT y Brox, T. (2015). Aprendiendo a generar sillas con Redes neuronales convolucionales. En *Actas de la Conferencia IEEE sobre visión artificial y reconocimiento de patrones*, páginas 1538–1546.[696,704,705](#)
- Doya, K. (1993). Bifurcaciones de redes neuronales recurrentes en el aprendizaje por descenso de gradientes. *Transacciones IEEE en redes neuronales*, **1**, 75–80.[401,403](#)
- Dreyfus, SE (1962). La solución numérica de problemas variacionales. *Diario de Análisis Matemático y Aplicaciones*, **5(1)**, 30–45.[225](#)
- Dreyfus, SE (1973). La solución computacional de problemas de control óptimo con el tiempo retraso. *Transacciones IEEE en control automático*, **18(4)**, 383–385.[225](#)
- Drucker, H. y LeCun, Y. (1992). Mejorar el rendimiento de la generalización usando doble retropropagación. *Transacciones IEEE en redes neuronales*, **3(6)**, 991–997.[271](#)

- Duchi, J., Hazan, E. y Singer, Y. (2011). Métodos adaptativos de subgradiente para online. aprendizaje y optimización estocástica. *Revista de investigación de aprendizaje automático*.[307](#)
- Dudik, M., Langford, J. y Li, L. (2011). Evaluación y aprendizaje de políticas doblemente sólidos. En *Actas de la 28.ª Conferencia Internacional sobre Aprendizaje Automático, ICML '11*. [482](#)
- Dugas, C., Bengio, Y., Bélisle, F. y Nadeau, C. (2001). Incorporación de segundo orden conocimiento funcional para una mejor fijación de precios de opciones. En T. Leen, T. Dietterich y V. Tresp, editores, *Avances en Sistemas de Procesamiento de Información Neural 13 (NIPS'00)*, páginas 472-478. Prensa del MIT.[68,197](#)
- Dziugaite, GK, Roy, DM y Ghahramani, Z. (2015). Entrenamiento de redes neuronales generativas funciona a través de la optimización de discrepancia media máxima. *preimpresión de arXiv arXiv:1505.03906*. [703](#)
- El Hihi, S. y Bengio, Y. (1996). Redes neuronales recurrentes jerárquicas a largo plazo dependencias En *NIPS'1995*.[398,407,408](#)
- Elkahky, AM, Song, Y. y He, X. (2015). Un enfoque de aprendizaje profundo multivista para modelado de usuarios de dominios cruzados en sistemas de recomendación. En *Actas de la 24.ª Conferencia Internacional sobre World Wide Web*, páginas 278-288.[480](#)
- Elman, JL (1993). Aprendizaje y desarrollo en redes neuronales: La importancia de empezando pequeño. *Cognición*,[48](#), 781-799.[328](#)
- Erhan, D., Manzagol, P.-A., Bengio, Y., Bengio, S. y Vincent, P. (2009). La dificultad del entrenamiento de arquitecturas profundas y el efecto del pre-entrenamiento no supervisado. En *Actas de AISTATS'2009*.[201](#)
- Erhan, D., Bengio, Y., Courville, A., Manzagol, P., Vincent, P. y Bengio, S. (2010). ¿Por qué el entrenamiento previo no supervisado ayuda al aprendizaje profundo?. *Aprendizaje automático Res.* [529,533,534](#)
- Fahlman, SE, Hinton, GE y Sejnowski, TJ (1983). Arquitecturas masivamente paralelas para IA: máquinas NETL, thistle y Boltzmann. En *Actas del Congreso Nacional de Inteligencia Artificial AAAI-83*.[570,654](#)
- Fang, H., Gupta, S., Iandola, F., Srivastava, R., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, JC, Zitnick, CL y Zweig, G. (2015). Desde subtítulos hasta conceptos visuales y viceversa. *arXiv:1411.4952*.[102](#)
- Farabet, C., LeCun, Y., Kavukcuoglu, K., Culurciello, E., Martini, B., Akselrod, P. y Talay, S. (2011). Redes convolucionales basadas en FPGA a gran escala. En R. Bekkerman, M. Bilenko y J. Langford, editores, *Ampliación del aprendizaje automático: enfoques paralelos y distribuidos*. Prensa de la Universidad de Cambridge.[523](#)

- Farabet, C., Couprie, C., Najman, L. y LeCun, Y. (2013). Aprendizaje de características jerárquicas para el etiquetado de escenas. *Transacciones IEEE sobre análisis de patrones e inteligencia artificial*, **35**(8), 1915–1929.[23,201,360](#)
- Fei-Fei, L., Fergus, R. y Perona, P. (2006). Aprendizaje de una sola vez de categorías de objetos. *Transacciones IEEE sobre análisis de patrones e inteligencia artificial*, **28**(4), 594–611.[538](#)
- Finn, C., Tan, XY, Duan, Y., Darrell, T., Levine, S. y Abbeel, P. (2015). Aprendiendo Espacios de características visuales para la manipulación robótica con codificadores automáticos espaciales profundos. *pre impresión de arXiv arXiv:1509.06113*.[25](#)
- Fischer, RA (1936). El uso de múltiples mediciones en problemas taxonómicos. *Anales de la eugenesia*, **7**, 179–188.[21,105](#)
- Földiák, P. (1989). Red adaptativa para una extracción óptima de características lineales. En *Internacional Conferencia Conjunta sobre Redes Neuronales (IJCNN)*, volumen 1, páginas 401–405, Washington 1989. IEEE, Nueva York.[494](#)
- Franzius, M., Sprekeler, H. y Wiskott, L. (2007). La lentitud y la escasez conducen al lugar, células de dirección de la cabeza y de visión espacial.[495](#)
- Franzius, M., Wilbert, N. y Wiskott, L. (2008). Reconocimiento invariable de objetos con lento análisis de características En *Redes Neuronales Artificiales-ICANN 2008*, páginas 961–970. Saltador. [496](#)
- Frasconi, P., Gori, M. y Sperduti, A. (1997). Sobre la clasificación eficiente de los datos estructuras por redes neuronales. En *proc. En t. Conf. conjunta sobre Inteligencia Artificial*.[401](#)
- Frasconi, P., Gori, M. y Sperduti, A. (1998). Un marco general para la adaptación procesamiento de estructuras de datos. *Transacciones IEEE en redes neuronales*, **9**(5), 768–786. [401](#)
- Freund, Y. y Schapire, RE (1996a). Experimentos con un nuevo algoritmo de impulso. En *Aprendizaje automático: actas de la decimotercera conferencia internacional*, páginas 148–156, EE. UU. ACM.[258](#)
- Freund, Y. y Schapire, RE (1996b). Teoría de juegos, predicción on-line y boosting. En *Actas de la Novena Conferencia Anual sobre Teoría del Aprendizaje Computacional*, páginas 325–332.[258](#)
- Frey, BJ (1998). *Modelos gráficos para aprendizaje automático y comunicación digital*. Prensa del MIT.[705,706](#)
- Frey, BJ, Hinton, GE y Dayan, P. (1996). ¿Aprende bien el algoritmo de despertar-dormir? estimadores de densidad? En D. Touretzky, M. Mozer y M. Hasselmo, editores, *Avances en sistemas de procesamiento de información neuronal 8 (NIPS'95)*, páginas 661–670. Prensa del MIT, Cambridge, MA.[651](#)

- Frobenius, G. (1908). Über matrizen aus positiven elementen, s.B. *Preuss. Akad. sabio Berlín, Alemania.*[597](#)
- Fukushima, K. (1975). Cognitron: una red neuronal multicapa autoorganizada. *Biológico Cibernetica*, **20**, 121–136.[dieciséis,226,528](#)
- Fukushima, K. (1980). Neocognitron: un modelo de red neuronal autoorganizada para un mecanismo de reconocimiento de patrones no afectado por el cambio de posición. *Cibernetica Biológica*, **36**, 193–202.[dieciséis,24,27,226,367](#)
- Gal, Y. y Ghahramani, Z. (2015). Redes neuronales convolucionales bayesianas con Bernoulli inferencia variacional aproximada. *preimpresión de arXiv arXiv:1506.02158.*[264](#)
- Gallinari, P., LeCun, Y., Thiria, S. y Fogelman-Soulie, F. (1987). Memorias asociativas distribuidos En *Actas de COGNITIVA 87*, París, La Villette.[515](#)
- García-Durán, A., Bordes, A., Usunier, N. y Grandvalet, Y. (2015). combinando dos y modelos de incrustaciones de tres vías para la predicción de enlaces en bases de conocimiento. *preimpresión de arXiv arXiv:1506.00999.*[484](#)
- Garofolo, JS, Lamel, LF, Fisher, WM, Fiscus, JG y Pallett, DS (1993). Darpa timit cd-rom acústico-fonético de corpus de habla continua. disco de voz nista 1-1.1. *Informe técnico de STI/Recon de la NASA N*, **93**, 27403.[459](#)
- Garson, J. (1900). El sistema métrico de identificación de criminales, como se usa en Gran Gran Bretaña e Irlanda. *Revista del Instituto Antropológico de Gran Bretaña e Irlanda*, (2), 177–227.[21](#)
- Gers, FA, Schmidhuber, J. y Cummins, F. (2000). Aprendiendo a olvidar: Continuo predicción con LSTM. *Computación neuronal*, **12**(10), 2451–2471.[410,412](#)
- Ghahramani, Z. y Hinton, GE (1996). El algoritmo EM para mezclas de factor analizadores Informe Técnico CRG-TR-96-1, Dpto. de Comp. Sci., Univ. de toronto[489](#)
- Gillick, D., Brunk, C., Vinyals, O. y Subramanya, A. (2015). Idioma multilingüe procesamiento a partir de bytes. *preimpresión de arXiv arXiv:1512.00103.*[477](#)
- Girshick, R., Donahue, J., Darrell, T. y Malik, J. (2015). Convolucional basado en regiones Redes para la detección y segmentación precisa de objetos.[426](#)
- Giudice, MD, Manera, V. y Keysers, C. (2009). ¿Programado para aprender? la ontogenia de neuronas espejo. *desarrollo ciencia*, **12**(2), 350–363.[656](#)
- Glorot, X. y Bengio, Y. (2010). Comprender la dificultad de entrenar feedforward profundo Redes neuronales. En *AISTATS'2010.*[303](#)
- Glorot, X., Bordes, A. y Bengio, Y. (2011a). Redes neuronales de rectificador disperso profundo. En *AISTATS'2011.*[dieciséis,174,197,226,227](#)

- Glorot, X., Bordes, A. y Bengio, Y. (2011b). Adaptación de dominios a gran escala clasificación de sentimientos: un enfoque de aprendizaje profundo. En *ICML'2011*.[507,537](#)
- Goldberger, J., Roweis, S., Hinton, GE y Salakhutdinov, R. (2005). Vecindario análisis de componentes. En L. Saul, Y. Weiss y L. Bottou, editores, *Avances en sistemas de procesamiento de información neuronal 17 (NIPS'04)*. Prensa del MIT.[115](#)
- Gong, S., McKenna, S. y Psarrou, A. (2000). *Visión dinámica: de las imágenes al rostro Reconocimiento*. Prensa del Colegio Imperial.[165,519](#)
- Goodfellow, I., Le, Q., Saxe, A. y Ng, A. (2009). Midiendo las invariancias en profundidad redes En *NIPS'2009*, páginas 646–654.[255](#)
- Goodfellow, I., Koenig, N., Muja, M., Pantofaru, C., Sorokin, A. y Takayama, L. (2010). Ayúdame a ayudarte: Interfaces para robots personales. En *proc. de Interacción Humano-Robot (HRI)*, Osaka, Japón. Prensa ACM, Prensa ACM.[100](#)
- Goodfellow, IJ (2010). Informe técnico: convolución multidimensional con resolución reducida para autocodificadores. Informe técnico, Université de Montréal.[357](#)
- Goodfellow, IJ (2014). Sobre los criterios de distinguibilidad para la estimación de modelos generativos. En *Conferencia Internacional sobre Representaciones de Aprendizaje, Ciclo de Talleres*.[622,700, 701](#)
- Goodfellow, IJ, Courville, A. y Bengio, Y. (2011). Codificación dispersa de punta y losa para el descubrimiento de funciones sin supervisión. En *Taller NIPS sobre desafíos en el aprendizaje de modelos jerárquicos*.[532,538](#)
- Goodfellow, IJ, Warde-Farley, D., Mirza, M., Courville, A. y Bengio, Y. (2013a). Maximizar redes. En S. Dasgupta y D. McAllester, editores, *ICML'13*, páginas 1319–1327.[193,264,344,365,455](#)
- Goodfellow, IJ, Mirza, M., Courville, A. y Bengio, Y. (2013b). Predicción múltiple profunda Máquinas de Boltzmann. En *NIPS26*. Fundación NIP.[100,617,671,672,673,674,675, 698](#)
- Goodfellow, IJ, Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F. y Bengio, Y. (2013c). PyLearn2: una biblioteca de investigación de aprendizaje automático. *pre impresión de arXiv arXiv:1308.4214*.[25,446](#)
- Goodfellow, IJ, Courville, A. y Bengio, Y. (2013d). Ampliación de modelos de espiga y losa para el aprendizaje de características no supervisado. *Transacciones IEEE sobre análisis de patrones e inteligencia artificial*.[35\(8\), 1902–1914.](#)[497,498,499,650,683](#)
- Goodfellow, IJ, Mirza, M., Xiao, D., Courville, A. y Bengio, Y. (2014a). un empírico investigación del olvido catastrófico en redes neuronales basadas en gradientes. En *ICLR'2014*.[194](#)

- Goodfellow, IJ, Shlens, J. y Szegedy, C. (2014b). Explicar y aprovechar la publicidad ejemplos sariales. *CoRR,abs/1412.6572.268,269,271,555,556*
- Goodfellow, IJ, Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. y Bengio, Y. (2014c). Redes adversarias generativas. En *NIPS'2014*. *544, 689,699,701,704*
- Goodfellow, IJ, Bulatov, Y., Ibarz, J., Arnoud, S. y Shet, V. (2014d). de varios dígitos reconocimiento de números a partir de imágenes de Street View utilizando redes neuronales convolucionales profundas. En *Conferencia Internacional sobre Representaciones de Aprendizaje*. *25,101,201,202,203,391, 422, 449*
- Goodfellow, IJ, Vinyals, O. y Saxe, AM (2015). Caracterización cualitativa de los nervios problemas de optimización de redes. En *Conferencia Internacional sobre Representaciones de Aprendizaje*. *285,286,287,291*
- Goodman, J. (2001). Clases para un entrenamiento rápido de máxima entropía. En *Internacional Conferencia sobre Acústica, Procesamiento de Señales y Habla (ICASSP)*, Utah. *467*
- Gori, M. y Tesi, A. (1992). Sobre el problema de los mínimos locales en retropropagación. *IEEE Transacciones sobre análisis de patrones e inteligencia artificial, PAMI-14*(1), 76–86. *284*
- Gosset, WS (1908). El error probable de una media. *Biometrika*, *6*(1), 1–25. Originalmente publicado bajo el seudónimo de “Estudiante”. *21*
- Gouws, S., Bengio, Y. y Corrado, G. (2014). BilBOWA: Rápido bilingüe distribuido representaciones sin alineación de palabras. Informe técnico, arXiv:1410.2455. *476,539*
- Graf, HP y Jackel, LD (1989). Circuitos de redes neuronales electrónicas analógicas. *Circuitos y revista de dispositivos, IEEE*, *5*(4), 44–49. *451*
- Tumbas, A. (2011). Inferencia variacional práctica para redes neuronales. En *NIPS'2011*. *242*
- Graves, A. (2012). *Etiquetado de secuencias supervisadas con redes neuronales recurrentes*. Estudios en Inteligencia Computacional. Saltador. *374,395,411,460*
- Graves, A. (2013). Generación de secuencias con redes neuronales recurrentes. Reporte técnico, arXiv:1308.0850. *190,410,415,420*
- Graves, A. y Jaitly, N. (2014). Hacia el reconocimiento de voz de extremo a extremo con recurrente Redes neuronales. En *ICML'2014*. *410*
- Graves, A. y Schmidhuber, J. (2005). Clasificación de fonemas por fotogramas con bidirección LSTM opcional y otras arquitecturas de redes neuronales. *Redes neuronales*, *18*(5), 602–610. *395*
- Graves, A. y Schmidhuber, J. (2009). Reconocimiento de escritura a mano sin conexión con redes neuronales recurrentes mensionales. En D. Koller, D. Schuurmans, Y. Bengio y L. Bottou, editores, *NIPS'2008*, páginas 545–552. *395*

- Graves, A., Fernández, S., Gómez, F. y Schmidhuber, J. (2006). Conexiónista temporal clasificación: Etiquetado de datos de secuencias no segmentadas con redes neuronales recurrentes. En *ICML'2006*, páginas 369–376, Pittsburgh, EE. UU.[460](#)
- Graves, A., Liwicki, M., Bunke, H., Schmidhuber, J. y Fernández, S. (2008). Descon- Reconocimiento de escritura en línea forzado con redes neuronales recurrentes. En J. Platt, D. Koller, Y. Singer y S. Roweis, editores, *NIPS'2007*, páginas 577–584.[395](#)
- Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H. y Schmidhuber, J. (2009). Un novedoso sistema conexiónista para el reconocimiento de escritura a mano sin restricciones. *Análisis de patrones e inteligencia artificial, transacciones IEEE en*,**31**(5), 855–868.[410](#)
- Graves, A., Mohamed, A. y Hinton, G. (2013). Reconocimiento de voz con profunda recurrencia Redes neuronales. En *ICASSP'2013*, páginas 6645–6649.[395,398,410,411,460](#)
- Graves, A., Wayne, G. y Danihelka, I. (2014a). Máquinas neurales de Turing. arXiv:[1410.5401.25](#)
- Graves, A., Wayne, G. y Danihelka, I. (2014b). Máquinas neurales de Turing. *preimpresión de arXiv arXiv:1410.5401.418*
- Grefenstette, E., Hermann, KM, Suleyman, M. y Blunsom, P. (2015). Aprendiendo a transducir con memoria ilimitada. En *NIPS'2015*.[418](#)
- Greff, K., Srivastava, RK, Koutník, J., Steunebrink, BR y Schmidhuber, J. (2015). LSTM: una odisea espacial de búsqueda. *preimpresión de arXiv arXiv:1503.04069.412*
- Gregor, K. y LeCun, Y. (2010a). Aparición de células complejas en un producto temporal red con campos receptivos locales. Informe técnico, arXiv:[1006.0448.352](#)
- Gregor, K. y LeCun, Y. (2010b). Aprendizaje de aproximaciones rápidas de codificación escasa. En L. Bottou y M. Littman, editores, *Actas de la Vigésimoséptima Conferencia Internacional sobre Aprendizaje Automático (ICML-10)*. ACM.[652](#)
- Gregor, K., Danihelka, I., Mnih, A., Blundell, C. y Wierstra, D. (2014). Profundo Redes autorregresivas. En *Conferencia Internacional sobre Aprendizaje Automático (ICML'2014)*.[693](#)
- Gregor, K., Danihelka, I., Graves, A. y Wierstra, D. (2015). DIBUJO: Una neural recurrente red para la generación de imágenes. *preimpresión de arXiv arXiv:1502.04623.698*
- Gretton, A., Borgwardt, KM, Rasch, MJ, Schölkopf, B. y Smola, A. (2012). A Prueba de núcleo de dos muestras. *El diario de investigación de aprendizaje automático*,**13**(1), 723–773. [704](#)
- Gülçehre, Ç. y Bengio, Y. (2013). El conocimiento importa: Importancia de la información previa para la optimización. En *Conferencia Internacional sobre Representaciones de Aprendizaje (ICLR'2013)*.[25](#)

- Guo, H. y Gelfand, SB (1992). Árboles de clasificación con función de red neuronal extracción. *Redes neuronales, transacciones IEEE en*, 3(6), 923–933. [450](#)
- Gupta, S., Agrawal, A., Gopalakrishnan, K. y Narayanan, P. (2015). Aprendizaje profundo con precisión numérica limitada. *CoRR,abs/1502.02551*. [452](#)
- Gutmann, M. y Hyvarinen, A. (2010). Estimación con contraste de ruido: una nueva estimación principio de ción para modelos estadísticos no normalizados. En *Actas de la Decimotercera Conferencia Internacional sobre Inteligencia Artificial y Estadísticas (AISTATS'10)*. [620](#)
- Hadsell, R., Sermanet, P., Ben, J., Erkan, A., Han, J., Muller, U. y LeCun, Y. (2007). Aprendizaje en línea para robots todoterreno: propagación de etiquetas espaciales para aprender la transitabilidad de largo alcance. En *Actas de robótica: ciencia y sistemas*, Atlanta, GA, Estados Unidos. [453](#)
- Hajnal, A., Maass, W., Pudlak, P., Szegedy, M. y Turan, G. (1993). Circuitos de umbral de profundidad acotada. *J. Cómputo. Sistema. ciencia*, 46, 129–154. [199](#)
- Håstad, J. (1986). Límites inferiores casi óptimos para circuitos de pequeña profundidad. En *Actas del 18º Simposio anual de ACM sobre Teoría de la Computación*, páginas 6–20, Berkeley, California. Prensa ACM. [199](#)
- Håstad, J. y Goldmann, M. (1991). Sobre el poder de los circuitos de umbral de pequeña profundidad. *Complejidad computacional*, 1, 113–129. [199](#)
- Hastie, T., Tibshirani, R. y Friedman, J. (2001). *Los elementos del aprendizaje estadístico: minería de datos, inferencia y predicción*. Serie Springer en Estadística. Springer Verlag. [146](#)
- Él, K., Zhang, X., Ren, S. y Sun, J. (2015). Profundizando en los rectificadores: Superando rendimiento a nivel humano en la clasificación de ImageNet. *preimpresión de arXiv arXiv:1502.01852*. [28](#), [193](#)
- Hebb, DO (1949). *La organización del comportamiento*. Wiley, Nueva York. [14](#), [17](#), [656](#)
- Henaff, M., Jarrett, K., Kavukcuoglu, K. y LeCun, Y. (2011). Aprendizaje sin supervisión de características escasas para la clasificación de audio escalable. En *ISMIR'11*. [523](#)
- Henderson, J. (2003). Inducir representaciones históricas para estadísticas de amplia cobertura analizando En *HLT-NAACL*, páginas 103–110. [477](#)
- Henderson, J. (2004). Entrenamiento discriminativo de un analizador estadístico de redes neuronales. En *Actas de la 42.ª Reunión Anual de la Asociación de Lingüística Computacional*, página 95. [477](#)
- Henniges, M., Puertas, G., Bornschein, J., Eggert, J. y Lücke, J. (2010). Binario escaso codificación. En *Análisis de variables latentes y separación de señales*, páginas 450–457. Saltador. [640](#)

- Herault, J. y Ans, B. (1984). Circuits neuronaux à synapses modifiables: Décodage de mensajes compuestos por aprendizaje no supervisado. *Comptes Rendus de l'Académie des Sciences*, **299** (III-13), 525–528.[491](#)
- Hinton, G. (2012). Redes neuronales para el aprendizaje automático. Coursera, videoconferencias.[307](#)
- Hinton, G., Deng, L., Dahl, GE, Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. y Kingsbury, B. (2012a). Redes neuronales profundas para modelado acústico en reconocimiento de voz. *Revista de procesamiento de señales IEEE*, **29**(6), 82–97.[23](#), [460](#)
- Hinton, G., Vinyals, O. y Dean, J. (2015). Destilando el conocimiento en una red neuronal. *preimpresión de arXiv arXiv:1503.02531*.[448](#)
- Hinton, GE (1989). Procedimientos de aprendizaje conexiónista. *Inteligencia artificial*, **40**, 185–234.[494](#)
- Hinton, GE (1990). Mapeo de jerarquías parte-todo en redes conexiónistas. *Artificial Intelligence*, **46**(1), 47–75.[418](#)
- Hinton, GE (1999). Productos de expertos. En *ICANN'1999*.[571](#)
- Hinton, GE (2000). Productos de entrenamiento de expertos minimizando la divergencia contrastiva. Informe técnico GCNU TR 2000-004, Unidad Gatsby, University College London.[610](#), [676](#)
- Hinton, GE (2006). Para reconocer formas, primero aprende a generar imágenes. Reporte técnico UTML TR 2006-003, Universidad de Toronto.[528](#), [595](#)
- Hinton, GE (2007a). Cómo hacer backpropagation en un cerebro. Charla invitada en el Taller de Aprendizaje Profundo NIPS'2007.[656](#)
- Hinton, GE (2007b). Aprendizaje de múltiples capas de representación. *Tendencias en lo cognitivo ciencias*, **11**(10), 428–434.[660](#)
- Hinton, GE (2010). Una guía práctica para entrenar máquinas Boltzmann restringidas. Informe técnico UTML TR 2010-003, Departamento de Ciencias de la Computación, Universidad de Toronto.[610](#)
- Hinton, GE y Ghahramani, Z. (1997). Modelos generativos para descubrir la dispersión. representaciones distribuidas. *Transacciones filosóficas de la Royal Society de Londres*. [147](#)
- Hinton, GE y McClelland, JL (1988). Aprendizaje de representaciones por recirculación. En *NIPS'1987*, páginas 358–366.[502](#)
- Hinton, GE y Roweis, S. (2003). Incrustación de vecinos estocásticos. En *NIPS'2002*.[519](#)

- Hinton, GE y Salakhutdinov, R. (2006). Reducir la dimensionalidad de los datos con Redes neuronales. *Ciencia*, **313**(5786), 504–507. [509, 524, 528, 529, 534](#)
- Hinton, GE y Sejnowski, TJ (1986). Aprendizaje y reaprendizaje en máquinas de Boltzmann. En DE Rumelhart y JL McClelland, editores, *Procesamiento distribuido en paralelo*, volumen 1, capítulo 7, páginas 282–317. MIT Press, Cambridge. [570, 654](#)
- Hinton, GE y Sejnowski, TJ (1999). *Aprendizaje no supervisado: fundamentos de neural cálculo*. Prensa del MIT. [541](#)
- Hinton, GE y Shallice, T. (1991). Lesión de una red atractora: investigaciones de dislexia adquirida. *revisión psicológica*, **98**(1), 74. [13](#)
- Hinton, GE y Zemel, RS (1994). Codificadores automáticos, longitud mínima de descripción y Energía libre de Helmholtz. En *NIPS'1993*. [502](#)
- Hinton, GE, Sejnowski, TJ y Ackley, DH (1984). Máquinas de Boltzmann: Restricción Redes de satisfacción que aprenden. Informe técnico TR-CMU-CS-84-119, Universidad Carnegie-Mellon, Departamento de Ciencias de la Computación. [570, 654](#)
- Hinton, GE, McClelland, J. y Rumelhart, D. (1986). Representaciones distribuidas. En DE Rumelhart y JL McClelland, editores, *Procesamiento distribuido en paralelo: exploraciones en la microestructura de la cognición*, volumen 1, páginas 77–109. MIT Press, Cambridge. [17, 225, 526](#)
- Hinton, GE, Revow, M. y Dayan, P. (1995a). Reconocimiento de dígitos escritos a mano usando mezclas de modelos lineales. En G. Tesauro, D. Touretzky y T. Leen, editores, *Avances en sistemas de procesamiento de información neuronal 7 (NIPS'94)*, páginas 1015–1022. Prensa del MIT, Cambridge, MA. [489](#)
- Hinton, GE, Dayan, P., Frey, BJ y Neal, RM (1995b). El algoritmo de despertar-dormir para redes neuronales no supervisadas. *Ciencia*, **268**, 1558–1161. [504, 651](#)
- Hinton, GE, Dayan, P. y Revow, M. (1997). Modelado de las variedades de imágenes de dígitos escritos a mano. *Transacciones IEEE en redes neuronales*, **8**, 65–74. [499](#)
- Hinton, GE, Welling, M., Teh, YW y Osindero, S. (2001). Una nueva visión de ICA. En *Actas de la 3.ª Conferencia Internacional sobre Análisis de Componentes Independientes y Separación Ciega de Señales (ICA'01)*, páginas 746–751, San Diego, CA. [491](#)
- Hinton, GE, Osindero, S. y Teh, Y. (2006). Un algoritmo de aprendizaje rápido para una creencia profunda redes. *Computación neuronal*, **18**, 1527–1554. [14, 19, 27, 143, 528, 529, 660, 661](#)
- Hinton, GE, Deng, L., Yu, D., Dahl, GE, Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, TN y Kingsbury, B. (2012b). Redes neuronales profundas para el modelado acústico en el reconocimiento de voz: las opiniones compartidas de cuatro grupos de investigación. *Proceso de señal IEEE. revista*, **29**(6), 82–97. [101](#)

- Hinton, GE, Srivastava, N., Krizhevsky, A., Sutskever, I. y Salakhutdinov, R. (2012c). Mejora de las redes neuronales al evitar la coadaptación de los detectores de características. Informe técnico, arXiv:1207.0580.[238,263,267](#)
- Hinton, GE, Vinyals, O. y Dean, J. (2014). Conocimiento oscuro. Charla invitada en el Simposio de aprendizaje automático BayLearn Bay Area.[448](#)
- Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, TU München.[18,401,403](#)
- Hochreiter, S. y Schmidhuber, J. (1995). Simplificando las redes neuronales al descubrir planos mínimos. En *Avances en Sistemas de Procesamiento de Información Neural* 7, páginas 529–536. Prensa del MIT.[243](#)
- Hochreiter, S. y Schmidhuber, J. (1997). Memoria a corto plazo. *Computación neuronal*, 9(8), 1735–1780.[18,410,411](#)
- Hochreiter, S., Bengio, Y. y Frasconi, P. (2001). Flujo de gradiente en redes recurrentes: el Dificultad para aprender dependencias a largo plazo. En J. Kolen y S. Kremer, editores, *Guía de campo para redes recurrentes dinámicas*. Prensa IEEE.[411](#)
- Holi, JL y Hwang, J.-N. (1993). Análisis de error de precisión finita de la red neuronal implementaciones de hardware. *Computadoras, Transacciones IEEE en*, 42(3), 281–290.[451](#)
- Holt, JL y Baker, TE (1991). Simulaciones de retropropagación usando precisión limitada cálculos de sion. En *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*, volumen 2, páginas 121–126. IEEE.[451](#)
- Hornik, K., Stinchcombe, M. y White, H. (1989). Las redes feedforward multicapa son aproximadores universales. *Redes neuronales*, 2, 359–366.[198](#)
- Hornik, K., Stinchcombe, M. y White, H. (1990). Aproximación universal de un mapeo desconocido y sus derivados utilizando redes feedforward multicapa. *Redes neuronales*, 3(5), 551–560.[198](#)
- Hsu, F.-H. (2002). *Detrás de Deep Blue: construyendo la computadora que derrotó al mundo campeón de ajedrez*. Princeton University Press, Princeton, Nueva Jersey, EE. UU.[2](#)
- Huang, F. y Ogata, Y. (2002). Estimaciones generalizadas de pseudoverosimilitud para Markov campos aleatorios en la red. *Anales del Instituto de Matemática Estadística*, 54(1), 1–18. [616](#)
- Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A. y Heck, L. (2013). aprendizaje profundo modelos semánticos estructurados para la búsqueda web utilizando datos de clics. En *Actas de la 22.ª conferencia internacional ACM sobre la gestión de la información y el conocimiento*, páginas 2333–2338. ACM.[480](#)
- Hubel, D. y Wiesel, T. (1968). Campos receptivos y arquitectura funcional del mono. corteza estriada. *Journal of Physiology (Londres)*, 195, 215–243.[364](#)

- Hubel, DH y Wiesel, TN (1959). Campos receptivos de neuronas individuales en el gato. corteza estriada. *Revista de fisiología*, **148**, 574–591.[364](#)
- Hubel, DH y Wiesel, TN (1962). Campos receptivos, interacción binocular y arquitectura funcional en la corteza visual del gato. *Journal of Physiology (Londres)*, **160**, 106–154.[364](#)
- Húszar, F. (2015). Cómo (no) entrenar su modelo generativo: programación de muestreo, probabilidad, ¿adversario? *arXiv:1511.05101*.[698](#)
- Hutter, F., Hoos, H. y Leyton-Brown, K. (2011). Optimización basada en modelos secuenciales para la configuración general del algoritmo. En *LEÓN-5*. Versión extendida como informe UBC Tech TR-2010-10.[436](#)
- Hyotyniemi, H. (1996). Las máquinas de Turing son redes neuronales recurrentes. En *PASO'96*, páginas 13–24.[379](#)
- Hyvärinen, A. (1999). Encuesta sobre análisis de componentes independientes. *Computación neuronal Encuestas*, **2**, 94–128.[491](#)
- Hyvärinen, A. (2005). Estimación de modelos estadísticos no normalizados mediante emparejamiento de puntuaciones. *Revista de investigación de aprendizaje automático*, **6**, 695–709.[513,617](#)
- Hyvärinen, A. (2007a). Conexiones entre puntuación coincidente, divergencia contrastiva, y pseudoverosimilitud para variables de valor continuo. *Transacciones IEEE en redes neuronales*, **18**, 1529–1531.[618](#)
- Hyvärinen, A. (2007b). Algunas extensiones de coincidencia de puntuación. *Estadística computacional y Análisis de los datos*, **51**, 2499–2512.[618](#)
- Hyvärinen, A. y Hoyer, PO (1999). Aparición de topografía y celda compleja. propiedades de imágenes naturales usando extensiones de ica. En *PINZAS*, páginas 827–833.[493](#)
- Hyvärinen, A. y Pajunen, P. (1999). Análisis de componentes independientes no lineales: Resultados de existencia y unicidad. *Redes neuronales*, **12**(3), 429–439.[493](#)
- Hyvärinen, A., Karhunen, J. y Oja, E. (2001a). *Análisis de componentes independientes*. Wiley-Interscience.[491](#)
- Hyvärinen, A., Hoyer, PO e Inki, MO (2001b). Componente topográfico independiente análisis. *Computación neuronal*, **13**(7), 1527–1558.[493](#)
- Hyvärinen, A., Hurri, J. y Hoyer, PO (2009). *Estadísticas de imagen natural: una prueba probabilística aproximación a la visión computacional temprana*. Springer-Verlag.[370](#)
- Iba, Y. (2001). Conjunto ampliado Montecarlo. *Revista internacional de física moderna*, **C12**, 623–656.[603](#)

- Inayoshi, H. y Kurita, T. (2005). Generalización mejorada al agregar tanto auto-asociación y ruido de capa oculta a clasificadores basados en redes neuronales. En *Taller IEEE sobre aprendizaje automático para el procesamiento de señales*, páginas 141—146.[515](#)
- Ioffe, S. y Szegedy, C. (2015). Normalización de lotes: aceleración del entrenamiento profundo de redes reduciendo el cambio de covariable interno.[100,317,320](#)
- Jacobs, RA (1988). Mayores tasas de convergencia a través de la adaptación de la tasa de aprendizaje. *Redes neuronales*,[1\(4\)](#), 295–307.[307](#)
- Jacobs, RA, Jordan, MI, Nowlan, SJ y Hinton, GE (1991). Mezclas adaptativas de expertos locales. *Computación neuronal*,[3](#), 79–87.[189,450](#)
- Jaeger, H. (2003). Identificación adaptativa de sistemas no lineales con redes de estado de eco. En *Avances en los sistemas de procesamiento de información neuronal* 15.[404](#)
- Jaeger, H. (2007a). Descubriendo características dinámicas multiescala con estado de eco jerárquico redes Informe técnico, Universidad de Jacobs.[398](#)
- Jaeger, H. (2007b). Red estatal de eco. *Scholarpedia*,[2\(9\)](#), 2330.[404](#)
- Jaeger, H. (2012). Memoria a corto plazo en redes de estados de eco: detalles de una simulación estudiar. Informe técnico, Informe técnico, Universidad Jacobs de Bremen.[405](#)
- Jaeger, H. y Haas, H. (2004). Aprovechar la no linealidad: predecir sistemas caóticos y ahorro de energía en la comunicación inalámbrica. *Ciencia*,[304](#)(5667), 78–80.[27,404](#)
- Jaeger, H., Lukosevicius, M., Popovici, D. y Siewert, U. (2007). Optimización y aplicaciones de redes de estado de eco con neuronas integradoras con fugas. *Redes neuronales*,[20](#) (3), 335–352.[407](#)
- Jain, V., Murray, JF, Roth, F., Turaga, S., Zhigulin, V., Briggman, KL, Helmstaedter, MN, Denk, W. y Seung, HS (2007). Aprendizaje supervisado de restauración de imágenes con redes convolucionales. En *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, páginas 1–8. IEEE.[359](#)
- Jaityl, N. y Hinton, G. (2011). Aprender una mejor representación de las ondas sonoras del habla usando máquinas Boltzmann restringidas. En *Procesamiento de Acústica, Habla y Señal (ICASSP), Conferencia Internacional IEEE 2011 sobre*, páginas 5884–5887. IEEE.[458](#)
- Jaityl, N. y Hinton, GE (2013). La perturbación de la longitud del tracto vocal (VTLP) mejora reconocimiento de voz. En *ICML'2013*.[241](#)
- Jarrett, K., Kavukcuoglu, K., Ranzato, M. y LeCun, Y. (2009). Que es lo mejor arquitectura de varias etapas para el reconocimiento de objetos? En *ICCV'09. dieciséis*,[24,27,174,193,226,363,364,523](#)
- Jarzynski, C. (1997). Igualdad de no equilibrio para diferencias de energía libre. *física Rev. Lett.*,[78](#), 2690–2693.[625,628](#)

Jaynes, ET (2003). *Teoría de la probabilidad: la lógica de la ciencia*. Universidad de Cambridge Prensa.[53](#)

Jean, S., Cho, K., Memisevic, R. y Bengio, Y. (2014). Sobre el uso de un objetivo muy grande Vocabulario para la traducción automática neuronal. arXiv:1412.2007.[474,475](#)

Jelinek, F. y Mercer, RL (1980). Estimación interpolada de los parámetros de fuente de Markov a partir de datos escasos. En ES Gelsema y LN Kanal, editores, *Reconocimiento de patrones en la práctica*. Holanda Septentrional, Ámsterdam.[462,473](#)

Jia, Y. (2013). Caffe: una arquitectura convolucional de código abierto para la incorporación rápida de características. <http://caffe.berkeleyvision.org/>[25,214](#)

Jia, Y., Huang, C. y Darrell, T. (2012). Más allá de las pirámides espaciales: campo receptivo aprendizaje para características de imágenes agrupadas. En *Visión artificial y reconocimiento de patrones (CVPR), Conferencia IEEE 2012 sobre*, páginas 3370–3377. IEEE.[345](#)

Jim, K.-C., Giles, CL y Horne, BG (1996). Un análisis del ruido en las neuronas recurrentes Redes: convergencia y generalización. *Transacciones IEEE en redes neuronales*, 7(6), 1424–1438.[242](#)

Jordán, MI (1998). *Aprendizaje en modelos gráficos*. Kluwer, Dordrecht, Países Bajos.[18](#)

Joulin, A. y Mikolov, T. (2015). Inferir patrones algorítmicos con pila aumentada redes recurrentes. *pre impresión de arXiv arXiv:1503.01007*.[418](#)

Jozefowicz, R., Zaremba, W. y Sutskever, I. (2015). Una evaluación empírica de recurrente arquitecturas de red. En *ICML'2015*.[306,412](#)

Judd, JS (1989). *Diseño de redes neuronales y la complejidad del aprendizaje*. Prensa del MIT. [293](#)

Jutten, C. y Herault, J. (1991). Separación ciega de fuentes, parte I: una adaptación algoritmo basado en arquitectura neuromimética. *Procesamiento de la señal*, 24, 1-10.[491](#)

Kahou, SE, Pal, C., Bouthillier, X., Froumenty, P., Gülçehre, c., Memisevic, R., Vincent, P., Courville, A., Bengio, Y., Ferrari, RC, Mirza, M., Jean, S., Carrier, PL, Dauphin, Y., Boulanger-Lewandowski, N., Aggarwal, A., Zumer, J., Lamblin, P., Raymond, J.-P., Desjardins, G., Pascanu, R., Warde-Farley, D., Torabi, A., Sharma, A., Bengio, E., Côté, M., Konda, KR y Wu, Z. (2013). Combina redes neuronales profundas específicas de modalidad para el reconocimiento de emociones en video. En *Actas de la 15ª ACM sobre la Conferencia Internacional sobre Interacción Multimodal*.[201](#)

Kalchbrenner, N. y Blunsom, P. (2013). Modelos recurrentes de traducción continua. En *EMNLP'2013*.[474,475](#)

Kalchbrenner, N., Danihelka, I. y Graves, A. (2015). Cuadrícula de memoria a largo plazo. *pre impresión de arXiv arXiv:1507.01526*.[395](#)

- Kamyshanska, H. y Memisevic, R. (2015). La energía potencial de un autocodificador. *Transacciones IEEE sobre análisis de patrones e inteligencia artificial*.[515](#)
- Karpathy, A. y Li, F.-F. (2015). Profundos alineamientos visual-semánticos para generar imágenes descripciones En *CVPR'2015*. arXiv:[1412.2306](#).[102](#)
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. y Fei-Fei, L. (2014). Clasificación de video a gran escala con redes neuronales convolucionales. En *CVPR*.[21](#)
- Karush, W. (1939). *Mínimos de funciones de varias variables con desigualdades como lado Restricciones*. Tesis de Maestría, Dpto. de Matemáticas, Univ. de Chicago.[95](#)
- Katz, SM (1987). Estimación de probabilidades a partir de datos escasos para el modelo de lenguaje componente de un reconocedor de voz. *Transacciones IEEE sobre acústica, voz y procesamiento de señales, ASSP-35*(3), 400–401.[462,473](#)
- Kavukcuoglu, K., Ranzato, M. y LeCun, Y. (2008). Inferencia rápida en codificación escasa algoritmos con aplicaciones al reconocimiento de objetos. Informe técnico, Computational and Biological Learning Lab, Courant Institute, NYU. Informe técnico CBLL-TR-2008-12-01. [523](#)
- Kavukcuoglu, K., Ranzato, M.-A., Fergus, R. y LeCun, Y. (2009). aprendizaje invariante características a través de mapas de filtros topográficos. En *CVPR'2009*.[523](#)
- Kavukcuoglu, K., Sermanet, P., Boureau, Y.-L., Gregor, K., Mathieu, M. y LeCun, Y. (2010). Aprendizaje de jerarquías de características convolucionales para el reconocimiento visual. En *NIPS'2010*.[364](#), [523](#)
- Kelley, HJ (1960). Teoría del gradiente de trayectorias de vuelo óptimas. *Diario ARS*,[30](#)(10), 947–954.[225](#)
- Khan, F., Zhu, X. y Mutlu, B. (2011). Cómo enseñan los humanos: Sobre el aprendizaje curricular y dimensión didáctica. En *Avances en sistemas de procesamiento de información neuronal 24 (NIPS'11)*, páginas 1449–1457.[328](#)
- Kim, SK, McAfee, LC, McMahon, PL y Olukotun, K. (2009). altamente escalable Implementación restringida de FPGA de máquina Boltzmann. En *Field Programmable Logic and Applications, 2009. FPL 2009. Conferencia Internacional sobre*, páginas 367–372. IEEE. [451](#)
- Kindermann, R. (1980). *Campos aleatorios de Markov y sus aplicaciones (Contemporáneo Matemáticas ; V 1)*. Sociedad Matemática Americana.[566](#)
- Kingma, D. y Ba, J. (2014). Adam: Un método para la optimización estocástica. *arXiv preimpresión arXiv:1412.6980*.[308](#)
- Kingma, D. y LeCun, Y. (2010). Estimación regularizada de estadísticas de imagen por puntuación pareo. En *NIPS'2010*.[513,620](#)

Kingma, D., Rezende, D., Mohamed, S. y Welling, M. (2014). Aprendizaje semisupervisado con profundos modelos generativos. En *NIPS'2014*.<sup>426</sup>

Kingma, DP (2013). Inferencia rápida basada en gradientes con variable latente continua modelos en forma auxiliar. Informe técnico, arxiv:1306.0733.<sup>652,689,696</sup>

Kingma, DP y Welling, M. (2014a). Bayes variacional de codificación automática. En *Actas de la Conferencia Internacional sobre Representaciones de Aprendizaje (ICLR)*.<sup>689,700</sup>

Kingma, DP y Welling, M. (2014b). Inferencia eficiente basada en gradientes a través de Transformaciones entre redes bayesianas y redes neuronales. Informe técnico, arxiv:1402.0480.<sup>689</sup>

Kirkpatrick, S., Jr., CDG, y Vecchi, MP (1983). Optimización por simulación recocido. *Ciencia*,<sup>220</sup>, 671–680.<sup>327</sup>

Kiros, R., Salakhutdinov, R. y Zemel, R. (2014a). Modelos de lenguaje neuronal multimodal. En *ICML'2014*.<sup>102</sup>

Kiros, R., Salakhutdinov, R. y Zemel, R. (2014b). Unificación de incrustaciones visual-semánticas con modelos de lenguaje neuronal multimodal. *arXiv:1411.2539 [cs.LG]*.<sup>102,410</sup>

Klementiev, A., Titov, I. y Bhattachari, B. (2012). Inducir la distribución translingual representaciones de palabras. En *Actas de COLING 2012*.<sup>476,539</sup>

Knowles-Barley, S., Jones, TR, Morgan, J., Lee, D., Kasthuri, N., Lichtman, JW y Pfister, H. (2014). Aprendizaje profundo para el conectoma. *Conferencia de tecnología GPU*.<sup>26</sup>

Koller, D. y Friedman, N. (2009). *Modelos gráficos probabilísticos: principios y Técnicas*. Prensa del MIT.<sup>583,595,645</sup>

Konig, Y., Bourlard, H. y Morgan, N. (1996). REMAP: Estimación recursiva y maximización de probabilidades a posteriori: aplicación al reconocimiento de voz conexiónista basado en transiciones. En D. Touretzky, M. Mozer y M. Hasselmo, editores, *Avances en sistemas de procesamiento de información neuronal 8 (NIPS'95)*. Prensa del MIT, Cambridge, MA.<sup>459</sup>

Koren, Y. (2009). La solución de BellKor al gran premio de Netflix.<sup>258,480</sup>

Kotzias, D., Denil, M., de Freitas, N. y Smyth, P. (2015). Del grupo al individuo etiquetas usando características profundas. En *ACM SIGKDD*.<sup>106</sup>

Koutnik, J., Greff, K., Gómez, F. y Schmidhuber, J. (2014). Un reloj RNN. En *ICML'2014*.<sup>408</sup>

Kočiský, T., Hermann, KM y Blunsom, P. (2014). Aprendizaje de la representación bilingüe de palabras sentencias por Alineaciones Marginalizantes. En *Procedimientos de ACL*.<sup>476</sup>

Krause, O., Fischer, A., Glasmachers, T. e Igel, C. (2013). Propiedades de aproximación de DBN con unidades ocultas binarias y unidades visibles de valor real. En *ICML'2013*.<sup>553</sup>

- Krizhevsky, A. (2010). Redes convolucionales de creencias profundas en CIFAR-10. Reporte técnico, Universidad de Toronto. Manuscrito no publicado: <http://www.cs.utoronto.ca/kriz/convifar10-aug2010.pdf>.<sup>446</sup>
- Krizhevsky, A. y Hinton, G. (2009). Aprendiendo múltiples capas de funciones desde pequeños imágenes Informe técnico, Universidad de Toronto.<sup>21,561</sup>
- Krizhevsky, A. y Hinton, GE (2011). Uso de codificadores automáticos muy profundos para contenido recuperación de imágenes En *ESANN*.<sup>525</sup>
- Krizhevsky, A., Sutskever, I. y Hinton, G. (2012). Clasificación ImageNet con profundidad Redes neuronales convolucionales. En *NIPS'2012*.<sup>23,24,27,100,201,371,454,458</sup>
- Krueger, KA y Dayan, P. (2009). Modelado flexible: cómo ayuda el aprendizaje en pequeños pasos. *Cognición*, **110**, 380–394.<sup>328</sup>
- Kuhn, HW y Tucker, AW (1951). Programación no lineal. En *Actas de la Segundo Simposio de Berkeley sobre Estadística Matemática y Probabilidad*, páginas 481–492, Berkeley, California. University of California Press.<sup>95</sup>
- Kumar, A., Irsoy, O., Su, J., Bradbury, J., English, R., Pierce, B., Ondruska, P., Iyyer, M., Gulrajani, I. y Socher, R. (2015). Pregúntame cualquier cosa: redes de memoria dinámicas para el procesamiento del lenguaje natural. *arXiv:1506.07285*.<sup>418,485</sup>
- Kumar, MP, Packer, B. y Koller, D. (2010). Aprendizaje a su propio ritmo para la variable latente modelos En *NIPS'2010*.<sup>328</sup>
- Lang, KJ y Hinton, GE (1988). El desarrollo de la red neuronal de retardo de tiempo Arquitectura para el reconocimiento de voz. Informe Técnico CMU-CS-88-152, Universidad Carnegie-Mellon.<sup>367,374,407</sup>
- Lang, KJ, Waibel, AH y Hinton, GE (1990). Una red neuronal de retardo de tiempo arquitectura para el reconocimiento de palabras aisladas. *Redes neuronales*, **3**(1), 23–43.<sup>374</sup>
- Langford, J. y Zhang, T. (2008). El algoritmo codicioso de época para múltiples brazos contextuales Bandidos. En *NIPS'2008*, páginas 1096–1103.<sup>480</sup>
- Lappalainen, H., Giannakopoulos, X., Honkela, A. y Karhunen, J. (2000). no lineal análisis de componentes independientes usando aprendizaje conjunto: Experimentos y discusión. En *proc. ICA*. Citeseer.<sup>493</sup>
- Larochelle, H. y Bengio, Y. (2008). Clasificación usando discriminativo restringido Máquinas de Boltzmann. En *ICML'2008*.<sup>244,255,530,686,716</sup>
- Larochelle, H. y Hinton, GE (2010). Aprender a combinar destellos foveales con un Máquina de Boltzmann de tercer orden. En *Avances en los sistemas de procesamiento de información neuronal* 23, páginas 1243–1251.<sup>367</sup>

- Larochelle, H. y Murray, I. (2011). El estimador de distribución autorregresivo neuronal. En *AISTATS'2011*.[705,708,709](#)
- Larochelle, H., Erhan, D. y Bengio, Y. (2008). Aprendizaje de datos cero de nuevas tareas. En *Conferencia AAAI sobre Inteligencia Artificial*.[539](#)
- Larochelle, H., Bengio, Y., Louradour, J. y Lamblin, P. (2009). Explorar estrategias para entrenar redes neuronales profundas. *Revista de investigación de aprendizaje automático*, **10**, 1-40.[535](#)
- Lasserre, JA, Bishop, CM y Minka, TP (2006). Híbridos basados en principios de generativo y modelos discriminativos. En *Actas de la Conferencia de reconocimiento de patrones y visión artificial (CVPR'06)*, páginas 87–94, Washington, DC, EE. UU. Sociedad de Computación IEEE. [244,253](#)
- Le, Q., Ngiam, J., Chen, Z., hao Chia, DJ, Koh, PW y Ng, A. (2010). embaldosado Redes neuronales convolucionales. En J. Lafferty, CKI Williams, J. Shawe-Taylor, R. Zemel y A. Culotta, editores, *Avances en sistemas de procesamiento de información neuronal 23 (NIPS'10)*, páginas 1279–1287.[352](#)
- Le, Q., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B. y Ng, A. (2011). Sobre la optimización Métodos para el aprendizaje profundo. En *proc. ICML'2011. ACM*.[316](#)
- Le, Q., Ranzato, M., Monga, R., Devin, M., Corrado, G., Chen, K., Dean, J. y Ng, A. (2012). Creación de características de alto nivel utilizando aprendizaje no supervisado a gran escala. En *ICML'2012*.[24,27](#)
- Le Roux, N. y Bengio, Y. (2008). Poder representacional de Boltzmann restringido máquinas y profundas redes de creencias. *Computación neuronal*, **20**(6), 1631–1649.[553,655](#)
- Le Roux, N. y Bengio, Y. (2010). Las redes de creencias profundas son compactas y universales aproximadamente. matores *Computación neuronal*, **22**(8), 2192–2207.[553](#)
- Le Cun, Y. (1985). Une procédure d'apprentissage pour Réseau à seuil assymétrique. En *Cognitiva 85: A la Frontière de l'Intelligence Artificielle, des Sciences de la Connaissance et des Neurosciences*, páginas 599–604, París 1985. CESTA, París.[225](#)
- Le Cun, Y. (1986). Procesos de aprendizaje en una red umbral asimétrica. En F. Fogelman-Soulie, E. Bienenstock y G. Weisbuch, editores, *Sistemas Desordenados y Organización Biológica*, páginas 233–240. Springer-Verlag, Les Houches, Francia.[352](#)
- Le Cun, Y. (1987). *Modèles connexionnistes de l'apprentissage*. Doctor. tesis, Universidad de París VI.[18,502,515](#)
- Le Cun, Y. (1989). Estrategias de generalización y diseño de redes. Reporte técnico CRG-TR-89-4, Universidad de Toronto.[330,352](#)

- LeCun, Y., Jackel, LD, Boser, B., Denker, JS, Graf, HP, Guyon, I., Henderson, D., Howard, RE y Hubbard, W. (1989). Reconocimiento de dígitos escritos a mano: aplicaciones de chips de redes neuronales y aprendizaje automático. *Revista de comunicaciones IEEE*, **27**(11), 41–46.[368](#)
- LeCun, Y., Bottou, L., Orr, GB y Müller, K.-R. (1998a). Apoyo trasero eficiente. En *Redes neuronales, trucos del oficio*, Lecture Notes in Computer Science LNCS 1524. Springer Verlag.[310,429](#)
- LeCun, Y., Bottou, L., Bengio, Y. y Haffner, P. (1998b). Aprendizaje basado en gradientes aplicado al reconocimiento de documentos. *proc. IEEE*.[dieciséis,18,21,27,371,458,460](#)
- LeCun, Y., Kavukcuoglu, K. y Farabet, C. (2010). Redes convolucionales y Aplicaciones en la visión. En *Circuitos y Sistemas (ISCAS), Actas del Simposio Internacional IEEE 2010 sobre*, páginas 253–256. IEEE.[371](#)
- L'Ecuyer, P. (1994). Mejora de la eficiencia y reducción de la varianza. En *Procedimientos de la Conferencia de simulación de invierno de 1994*, páginas 122–132.[690](#)
- Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z. y Tu, Z. (2014). Redes profundamente supervisadas. *preimpresión de arXiv arXiv:1409.5185*.[326](#)
- Lee, H., Battle, A., Raina, R. y Ng, A. (2007). Algoritmos de codificación dispersa eficientes. En B. Schölkopf, J. Platt y T. Hoffman, editores, *Avances en Sistemas de Procesamiento de Información Neural 19 (NIPS'06)*, páginas 801–808. Prensa del MIT.[637](#)
- Lee, H., Ekanadham, C. y Ng, A. (2008). Modelo de red de creencias profundas dispersas para el área visual V2. En *NIPS'07*.[255](#)
- Lee, H., Grosse, R., Ranganath, R. y Ng, AY (2009). Creencia profunda convolucional redes para el aprendizaje escalable no supervisado de representaciones jerárquicas. En L. Bottou y M. Littman, editores, *Actas de la Vigésima Sexta Conferencia Internacional sobre Aprendizaje Automático (ICML'09)*. ACM, Montreal, Canadá.[363,683,684](#)
- Lee, YJ y Grauman, K. (2011). Aprender primero las cosas fáciles: visualización a su propio ritmo descubrimiento de categorías. En *CVPR'2011*.[328](#)
- Leibniz, GW (1676). Memorias usando la regla de la cadena. (Citado en TMME 7:2&3 p 321-332, 2010).[225](#)
- Lenat, DB y Guha, RV (1989). *Construir grandes sistemas basados en el conocimiento; representación e inferencia en el proyecto Cyc*. Addison-Wesley Longman Publishing Co., Inc. [2](#)
- Leshno, M., Lin, VY, Pinkus, A. y Schocken, S. (1993). realimentación multicapa las redes con una función de activación no polinomial pueden aproximarse a cualquier función. *Redes neuronales*, **6**, 861–867.[198,199](#)

- Levenberg, K. (1944). Un método para la solución de ciertos problemas no lineales en al menos cuadrícula. *Revista trimestral de matemáticas aplicadas*, **Y**o(2), 164–168.[312](#)
- L'Hôpital, GFA (1696). *Analyse des infiniment petits, pour l'intelligence des lignes courbes*. París: L'Imprimerie Royale.[225](#)
- Li, Y., Swersky, K. y Zemel, RS (2015). Redes generativas de coincidencia de momentos. *CoRR*, **abs/1502.02761**.[703](#)
- Lin, T., Horne, BG, Tino, P. y Giles, CL (1996). Aprender dependencias a largo plazo no es tan difícil con las redes neuronales recurrentes NARX. *Transacciones IEEE en redes neuronales*, **7**(6), 1329–1338.[407](#)
- Lin, Y., Liu, Z., Sun, M., Liu, Y. y Zhu, X. (2015). Entidad de aprendizaje y relación incrustaciones para completar gráficos de conocimiento. *Enproc. AAAI'15*.[484](#)
- Linde, N. (1992). La máquina que cambió el mundo, episodio 3. Miniserie documental. [2](#)
- Lindsey, C. y Lindblad, T. (1994). Revisión de redes neuronales de hardware: la de un usuario perspectiva. *Enproc. Tercer Taller de Redes Neuronales: De la Biología a la Física de Altas Energías*, páginas 195–202, Isola d'Elba, Italia.[451](#)
- Linnainmaa, S. (1976). Expansión de Taylor del error de redondeo acumulado. *POCO Matemáticas numéricas*, **dieciséis**(2), 146–160.[225](#)
- LISA (2008). Tutoriales de aprendizaje profundo: máquinas Boltzmann restringidas. Reporte técnico, LISA Lab, Universidad de Montreal.[589](#)
- Largo, PM y Servedio, RA (2010). Las máquinas Boltzmann restringidas son difíciles de evaluar o simular aproximadamente. *EnActas de la 27.ª Conferencia Internacional sobre Aprendizaje Automático (ICML'10)*.[658](#)
- Lotter, W., Kreiman, G. y Cox, D. (2015). Aprendizaje no supervisado de la estructura visual usando redes generativas predictivas. *preimpresión de arXiv arXiv:1511.06380*.[544,545](#)
- Lovelace, A. (1842). Apuntes sobre el “Sketch of the Analytical Engine” de LF Menabrea inventado por Charles Babbage”.[1](#)
- Lu, L., Zhang, X., Cho, K. y Renals, S. (2015). Un estudio de la red neuronal recurrente codificador-decodificador para reconocimiento de voz de vocabulario amplio. *Enproc. interdiscurso*.[461](#)
- Lu, T., Pál, D. y Pál, M. (2010). Bandidos multi-armados contextuales. *EnInternacional Jornada sobre Inteligencia Artificial y Estadística*, páginas 485–492.[480](#)
- Luenberger, DG (1984). *Programación lineal y no lineal*. AddisonWesley.[316](#)
- Lukoševičius, M. y Jaeger, H. (2009). Enfoques de computación de yacimientos para recurrentes entrenamiento de redes neuronales. *Revisión de ciencias de la computación*, **3**(3), 127–149.[404](#)

- Luo, H., Shen, R., Niu, C. y Ullrich, C. (2011). Aprendizaje de características relevantes para la clase y características irrelevantes para la clase a través de un RBM híbrido de tercer orden. En *Congreso Internacional de Inteligencia Artificial y Estadística*, páginas 470–478.[686](#)
- Luo, H., Carrier, PL, Courville, A. y Bengio, Y. (2013). Modelado de texturas con RBM convolucionales de punta y losa y extensiones profundas. En *AISTATS'2013*.[102](#)
- Lyu, S. (2009). Interpretación y generalización del emparejamiento de puntuaciones. En *Actas de la XXIV Jornadas de Incertidumbre en Inteligencia Artificial (UAI'09)*.[618](#)
- Ma, J., Sheridan, RP, Liaw, A., Dahl, GE y Svetnik, V. (2015). redes neuronales profundas como método para relaciones cuantitativas estructura – actividad. *J. Información y modelos químicos*.[530](#)
- Maas, AL, Hannun, AY y Ng, AY (2013). Las no linealidades del rectificador mejoran las neuronas modelos acústicos de red. En *Taller ICML sobre aprendizaje profundo para procesamiento de audio, voz y lenguaje*.[193](#)
- Maass, W. (1992). Límites para el poder computacional y la complejidad de aprendizaje de los analógicos redes neuronales (resumen extendido). En *proc. del 25º Simposio ACM. Teoría de la Computación*, páginas 335–344.[199](#)
- Maass, W., Schnitger, G. y Sontag, ED (1994). Una comparación de los cálculos potencia de los circuitos de umbral sigmoide y booleano. *Avances teóricos en computación neuronal y aprendizaje*, páginas 127–151.[199](#)
- Maass, W., Natschlaeger, T. y Markram, H. (2002). Cómputo en tiempo real sin Estados estables: un nuevo marco para el cálculo neuronal basado en perturbaciones. *Computación neuronal*, **14**(11), 2531–2560.[404](#)
- Mackay, D. (2003). *Teoría de la Información, Inferencia y Algoritmos de Aprendizaje*. Cambridge Prensa Universitaria.[73](#)
- MacLaurin, D., Duvenaud, D. y Adams, RP (2015). Hiperparámetro basado en gradiente optimización a través del aprendizaje reversible. *preimpresión de arXiv arXiv:1502.03492*.[435](#)
- Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z. y Yuille, AL (2015). Subtítulos profundos con redes neuronales recurrentes multimodales. En *ICLR'2015*. arXiv:1410.1090.[102](#)
- Marcotte, P. y Savard, G. (1992). Nuevas aproximaciones al problema de la discriminación. *Zeitschrift für Operations Research (Teoría)*, **36**, 517–545.[276](#)
- Marlin, B. y de Freitas, N. (2011). Eficiencia asintótica de estimadores deterministas para Modelos discretos basados en energía: Ratio matching y pseudoverosimilitud. En *AUI'2011*.[617](#), **619**

- Marlin, B., Swersky, K., Chen, B. y de Freitas, N. (2010). Principios inductivos para Aprendizaje automático de Boltzmann restringido. En *Actas de la Decimotercera Conferencia Internacional sobre Inteligencia Artificial y Estadísticas (AISTATS'10)*, volumen 9, páginas 509–516. [613,618,619](#)
- Marquardt, DW (1963). Un algoritmo para la estimación por mínimos cuadrados de parámetros no lineales éteres. *Revista de la Sociedad de Matemáticas Industriales y Aplicadas*, **11**(2), 431–441. [312](#)
- Marr, D. y Poggio, T. (1976). Cálculo cooperativo de la disparidad estéreo. *Ciencia*, **194**. [367](#)
- Martens, J. (2010). Aprendizaje profundo a través de la optimización sin Hessian. En L. Bottou y M. Littman, editores, *Actas de la Vigésimoséptima Conferencia Internacional sobre Aprendizaje Automático (ICML-10)*, páginas 735–742. ACM. [304](#)
- Martens, J. y Medabalimi, V. (2014). Sobre la eficiencia expresiva de la suma del producto redesarXiv:[1411.7717](#). [554](#)
- Martens, J. y Sutskever, I. (2011). Aprendizaje de redes neuronales recurrentes con Hessian-free mejoramiento. En *proc. ICML'2011*. ACM. [413](#)
- Mase, S. (1995). Consistencia del estimador de máxima pseudoverosimilitud de continuo Procesos gibbsianos en el espacio de estados. *Los Anales de la Probabilidad Aplicada*, **5**(3), págs. 603–612. [616](#)
- McClelland, J., Rumelhart, D. y Hinton, G. (1995). El atractivo de la distribución paralela Procesando. En *Computación e inteligencia*, páginas 305–341. Asociación Americana de Inteligencia Artificial. [17](#)
- McCulloch, WS y Pitts, W. (1943). Un cálculo lógico de ideas inmanentes en nervioso actividad. *Boletín de Biofísica Matemática*, **5**, 115–133. [14,15](#)
- Mead, C. e Ismail, M. (2012). *Implementación analógica VLSI de sistemas neuronales*, volumen 80. Springer Science & Business Media. [451](#)
- Melchior, J., Fischer, A. y Wiskott, L. (2013). Cómo centrar binario profundo Boltzmann máquinas. preimpresión de arXiv arXiv:[1311.1354](#). [674](#)
- Memisevic, R. y Hinton, GE (2007). Aprendizaje no supervisado de transformaciones de imágenes. En *Actas de la Conferencia de reconocimiento de patrones y visión artificial (CVPR'07)*. [686](#)
- Memisevic, R. y Hinton, GE (2010). Aprender a representar transformaciones espaciales con máquinas Boltzmann factorizadas de orden superior. *Computación neuronal*, **22**(6), 1473–1492. [686](#)

- Mesnil, G., Dauphin, Y., Glorot, X., Rifai, S., Bengio, Y., Goodfellow, I., Lavoie, E., Muller, X., Desjardins, G., Warde-Farley, D., Vincent, P., Courville, A. y Bergstra, J. (2011). Desafío de aprendizaje no supervisado y de transferencia: un enfoque de aprendizaje profundo. En *JMLR W&CP: Proc. Aprendizaje no supervisado y de transferencia*, volumen 7. [201,532,538](#)
- Mesnil, G., Rifai, S., Dauphin, Y., Bengio, Y. y Vincent, P. (2012). surfeando en el colector. Taller de Aprendizaje, Snowbird. [711](#)
- Miikkulainen, R. y Dyer, MG (1991). Procesamiento de lenguaje natural con modular Redes PDP y léxico distribuido. *Ciencia cognitiva*, **15**, 343–399. [477](#)
- Mikolov, T. (2012). *Modelos de lenguaje estadístico basados en redes neuronales*. Doctor. tesis, Universidad Tecnológica de Brno. [414](#)
- Mikolov, T., Deoras, A., Kombrink, S., Burget, L. y Cernocky, J. (2011a). Empírico evaluación y combinación de técnicas avanzadas de modelado del lenguaje. En *proc. 12<sup>a</sup> conferencia anual de la asociación internacional de comunicación del habla (INTERSPEECH 2011)*. [472](#)
- Mikolov, T., Deoras, A., Povey, D., Burget, L. y Cernocky, J. (2011b). Estrategias para entrenar modelos de lenguaje de redes neuronales a gran escala. En *proc. ASRU'2011*. [328,472](#)
- Mikolov, T., Chen, K., Corrado, G. y Dean, J. (2013a). Estimación eficiente de la representación de palabras resentaciones en el espacio vectorial. En *Conferencia Internacional sobre Representaciones de Aprendizaje: Ciclo de Talleres*. [536](#)
- Mikolov, T., Le, QV y Sutskever, I. (2013b). Explotar similitudes entre idiomas para la traducción automática. Informe técnico, arXiv:1309.4168. [539](#)
- Minka, T. (2005). Medidas de divergencia y paso de mensajes. *Investigación de Microsoft Cambridge Representante técnico del Reino Unido MSRTR2005173*, **72**(TR-2005-173). [625](#)
- Minsky, ML y Papert, SA (1969). *perceptrones*. MIT Press, Cambridge. [15](#)
- Mirza, M. y Osindero, S. (2014). Redes adversarias generativas condicionales. *preimpresión de arXiv arXiv:1411.1784*. [702](#)
- Mishkin, D. y Matas, J. (2015). Todo lo que necesitas es un buen init. *preimpresión de arXiv arXiv:1511.06422*. [305](#)
- Misra, J. y Saha, I. (2010). Redes neuronales artificiales en hardware: una encuesta de dos décadas de progreso. *neurocomputación*, **74**(1), 239–255. [451](#)
- Mitchell, TM (1997). *Aprendizaje automático*. McGraw-Hill, Nueva York. [99](#)
- Miyato, T., Maeda, S., Koyama, M., Nakae, K. e Ishii, S. (2015). Distribucional Suavizado con entrenamiento adversarial virtual. En *ICLR*. Preimpresión: arXiv:1507.00677. [269](#)

- Mnih, A. y Gregor, K. (2014). Inferencia variacional neuronal y aprendizaje en la creencia de redes En *ICML'2014*.[691, 692, 693](#)
- Mnih, A. y Hinton, GE (2007). Tres nuevos modelos gráficos para el lenguaje estadístico modelado. En Z. Ghahramani, editor, *Actas de la Vigésima Cuarta Conferencia Internacional sobre Aprendizaje Automático (ICML'07)*, páginas 641–648. ACM.[465](#)
- Mnih, A. y Hinton, GE (2009). Un modelo de lenguaje distribuido jerárquico escalable. En D. Koller, D. Schuurmans, Y. Bengio y L. Bottou, editores, *Avances en sistemas de procesamiento de información neuronal 21 (NIPS'08)*, páginas 1081–1088.[467](#)
- Mnih, A. y Kavukcuoglu, K. (2013). Aprender incrustaciones de palabras de manera eficiente con ruido. estimación contrastiva. En C. Burges, L. Bottou, M. Welling, Z. Ghahramani y K. Weinberger, editores, *Avances en los sistemas de procesamiento de información neuronal 26*, páginas 2265–2273. Curran Associates, Inc.[472, 622](#)
- Mnih, A. y Teh, YW (2012). Un algoritmo rápido y simple para entrenar neural modelos de lenguaje probabilístico. En *ICML'2012*, páginas 1751–1758.[472](#)
- Mnih, V. y Hinton, G. (2010). Aprender a detectar carreteras en imágenes aéreas de alta resolución. En *Actas de la 11.ª Conferencia Europea sobre Visión por Computador (ECCV)*.[102](#)
- Mnih, V., Larochelle, H. y Hinton, G. (2011). Boltzmann condicional restringido máquinas para la predicción de la salida de la estructura. En *proc. Conf. sobre Incertidumbre en Inteligencia Artificial (UAI)*.[685](#)
- Mnih, V., Kavukcuoglo, K., Silver, D., Graves, A., Antonoglou, I. y Wierstra, D. (2013). Jugar a Atari con aprendizaje de refuerzo profundo. Informe técnico, arXiv:1312.5602.[106](#)
- Mnih, V., Heess, N., Graves, A. y Kavukcuoglu, K. (2014). Modelos recurrentes de visual atención. En Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence y K. Weinberger, editores, *NIPS'2014*, páginas 2204–2212.[691](#)
- Mnih, V., Kavukcuoglo, K., Silver, D., Rusu, AA, Veness, J., Bellemare, MG, Graves, A., Riedmiller, M., Fidgeland, AK, Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. y Hassabis, D. (2015). Control a nivel humano a través del aprendizaje de refuerzo profundo. *Naturaleza*, **518**, 529–533.[25](#)
- Mobahi, H. y Fisher, III, JW (2015). Un análisis teórico de la optimización por Continuación gaussiana. En *AAAI'2015*.[327](#)
- Mobahi, H., Collobert, R. y Weston, J. (2009). Aprendizaje profundo a partir de la coherencia temporal en vídeo En L. Bottou y M. Littman, editores, *Actas de la 26ª Conferencia Internacional sobre Aprendizaje Automático*, páginas 737–744, Montreal. Omnipress.[494](#)
- Mohamed, A., Dahl, G. y Hinton, G. (2009). Redes de creencias profundas para el reconocimiento de teléfonos.  
[459](#)

- Mohamed, A., Sainath, TN, Dahl, G., Ramabhadran, B., Hinton, GE y Picheny, MA (2011). Redes de creencias profundas que utilizan funciones discriminatorias para el reconocimiento de teléfonos. En *Procesamiento de Acústica, Habla y Señal (ICASSP), Conferencia Internacional IEEE 2011 sobre*, páginas 5060–5063. IEEE.[459](#)
- Mohamed, A., Dahl, G. y Hinton, G. (2012a). Modelado acústico usando creencias profundas redes *Trans. IEEE. sobre procesamiento de audio, habla y lenguaje*,**20**(1), 14–22.[459](#)
- Mohamed, A., Hinton, G. y Penn, G. (2012b). Comprender cómo las redes de creencias profundas realizar modelado acústico. En *Procesamiento de Acústica, Habla y Señal (ICASSP), Conferencia Internacional IEEE 2012 sobre*, páginas 4273–4276. IEEE.[459](#)
- Moller, MF (1993). Un algoritmo de gradiente conjugado escalado para un aprendizaje supervisado rápido. *Redes neuronales*,**6**, 525–533.[316](#)
- Montavon, G. y Muller, K.-R. (2012). Máquinas Deep Boltzmann y el centrado truco. En G. Montavon, G. Orr y K.-R. Muller, editores, *Redes neuronales: trucos del oficio*, volumen 7700 de *Apuntes de clase en informática*, páginas 621–637. Preimpresión: <http://arxiv.org/abs/1203.3783>.[673](#)
- Montúfar, G. (2014). Profundidad de aproximación universal y errores de redes de creencias estrechas con unidades discretas. *Computación neuronal*,**26**.[553](#)
- Montúfar, G. y Ay, N. (2011). Refinamientos de los resultados de la aproximación universal para profundas redes de creencias y máquinas restringidas de Boltzmann. *Computación neuronal*,**23**(5), 1306–1319.[553](#)
- Montufar, GF, Pascanu, R., Cho, K. y Bengio, Y. (2014). Sobre el número de lineales regiones de redes neuronales profundas. En *NIPS'2014*.[19,199,200](#)
- Mor-Yosef, S., Samueloff, A., Modan, B., Navot, D. y Schenker, JG (1990). Clasificación los factores de riesgo de cesárea: análisis de regresión logística de un estudio a nivel nacional. *obstetricia ginecológica*,**75**(6), 944–7.[3](#)
- Morin, F. y Bengio, Y. (2005). Lenguaje de red neuronal probabilística jerárquica modelo. En *AISTATS'2005*.[467,469](#)
- Mozer, MC (1992). La inducción de la estructura temporal multiescala. En JMS Hanson y R. Lippmann, editores, *Avances en sistemas de procesamiento de información neuronal 4 (NIPS'91)*, páginas 275–282, San Mateo, CA. Morgan Kaufman.[407,408](#)
- Murphy, KP (2012). *Aprendizaje automático: una perspectiva probabilística*. prensa del MIT, Cambridge, MA, Estados Unidos.[62,98,146](#)
- Murray, BUI y Larochelle, H. (2014). Un estimador de densidad profundo y manejable. En *ICML'2014*.[190,710](#)
- Nair, V. y Hinton, G. (2010). Las unidades lineales rectificadas mejoran Boltzmann restringido máquinas. En *ICML'2010*.[dieciséis,174,197](#)

- Nair, V. y Hinton, GE (2009). Reconocimiento de objetos 3d con redes de creencias profundas. En Y. Bengio, D. Schuurmans, JD Lafferty, CKI Williams y A. Culotta, editores, *Avances en los sistemas de procesamiento de información neuronal* 22, páginas 1339–1347. Curran Associates, Inc. **686**
- Narayanan, H. y Mitter, S. (2010). Muestra la complejidad de probar la hipótesis múltiple. En *NIPS'2010*. **164**
- Naumann, U. (2008). La acumulación jacobiana óptima es NP-completa. *Matemático Programación*, **112**(2), 427–441. **222**
- Navigli, R. y Velardi, P. (2005). Interconexiones semánticas estructurales: un conocimiento- enfoque basado en la desambiguación del sentido de las palabras. *Trans. IEEE. Análisis de patrones e inteligencia artificial*, **27**(7), 1075–1086. **485**
- Neal, R. y Hinton, G. (1999). Una vista del algoritmo EM que justifica incrementos, escaso, y otras variantes. En MI Jordan, editor, *Aprendizaje en modelos gráficos*. Prensa del MIT, Cambridge, MA. **634**
- Neal, RM (1990). Aprendizaje de redes feedforward estocásticas. Reporte técnico. **692**
- Neal, RM (1993). Inferencia probabilística usando métodos Monte-Carlo de cadenas de Markov. Informe técnico CRG-TR-93-1, Departamento de Ciencias de la Computación, Universidad de Toronto. **680**
- Neal, RM (1994). Muestreo de distribuciones multimodales usando transiciones temperadas. Informe Técnico 9421, Departamento de Estadística, Universidad de Toronto. **603**
- Neal, RM (1996). *Aprendizaje bayesiano para redes neuronales*. Apuntes de clase en estadística. Saltador. **265**
- Neal, RM (2001). Muestreo de importancia recocido. *Estadística y Computación*, **11**(2), 125–139. **625, 627, 628**
- Neal, RM (2005). Estimación de proporciones de constantes de normalización utilizando la importancia vinculada muestreo. **629**
- Nésterov, Y. (1983). Un método para resolver un problema de programación convexa con convergencia tasa  $O(1/k)$ . *Doklady de matemáticas soviéticas*, **27**, 372–376. **300**
- Nésterov, Y. (2004). *Conferencias introductorias sobre optimización convexa: un curso básico*. Aplicado mejoramiento. Kluwer Academic Publ., Boston, Dordrecht, Londres. **300**
- Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. y Ng, AY (2011). Lectura dígitos en imágenes naturales con aprendizaje de funciones no supervisado. Taller de Aprendizaje Profundo y Aprendizaje de Funciones No Supervisado, NIPS. **21**
- Ney, H. y Kneser, R. (1993). Técnicas de agrupamiento mejoradas para estadísticas basadas en clases. modelado del lenguaje. En *Conferencia europea sobre comunicación y tecnología del habla (Eurospeech)*, páginas 973–976, Berlín. **463**

- ng, R. (2015). Consejo para aplicar máquina aprendiendo. <https://see.stanford.edu/materials/aimlcs229/ML-advice.pdf>.<sup>421</sup>
- Niesler, TR, Whittaker, EWD y Woodland, PC (1998). Comparación de parte de voz y modelos de lenguaje basados en categorías derivados automáticamente para el reconocimiento de voz. En *Conferencia Internacional sobre Acústica, Habla y Procesamiento de Señales (ICASSP)*, páginas 177–180.<sup>463</sup>
- Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L. y Barbano, PE (2005). Hacia el fenotipado automático de embriones en desarrollo a partir de videos. *Procesamiento de imágenes, transacciones IEEE en*, **14**(9), 1360–1371.<sup>360</sup>
- Nocedal, J. y Wright, S. (2006). *Optimización Numérica*. Saltador.<sup>92,96</sup>
- Norouzi, M. y Fleet, DJ (2011). Hashing de pérdida mínima para códigos binarios compactos. En *ICML'2011*.<sup>525</sup>
- Nowlan, SJ (1990). Expertos en competencia: una investigación experimental de asociaciones modelos de mezcla Informe técnico CRG-TR-90-5, Universidad de Toronto.<sup>450</sup>
- Nowlan, SJ y Hinton, GE (1992). Simplificación de las redes neuronales mediante el reparto de peso suave. *Computación neuronal*, **4**(4), 473–493.<sup>139</sup>
- Olshausen, B. y Field, DJ (2005). ¿Qué tan cerca estamos de entender V1? *Neural Cálculo*, **17**, 1665–1699.<sup>dieciséis</sup>
- Olshausen, BA y Field, DJ (1996). Aparición de propiedades de campo receptivo de células simples aprendiendo un código escaso para imágenes naturales. *Naturaleza*, **381**, 607–609.<sup>147,255,370,496</sup>
- Olshausen, BA, Anderson, CH y Van Essen, DC (1993). Un neurobiológico modelo de atención visual y reconocimiento de patrones invariantes basado en enrutamiento dinámico de información. *J. Neurosci.*, **13**(11), 4700–4719.<sup>450</sup>
- Opper, M. y Archambeau, C. (2009). Revisión de la aproximación variacional de Gauss. *Computación neuronal*, **21**(3), 786–792.<sup>689</sup>
- Oquab, M., Bottou, L., Laptev, I. y Sivic, J. (2014). Aprendizaje y transferencia de nivel medio y representaciones de imágenes utilizando redes neuronales convolucionales. En *Visión artificial y reconocimiento de patrones (CVPR), Conferencia IEEE 2014 sobre*, páginas 1717–1724. IEEE.<sup>536</sup>
- Osindero, S. y Hinton, GE (2008). Modelado de parches de imagen con una jerarquía dirigida de campos aleatorios de Markov. En J. Platt, D. Koller, Y. Singer y S. Roweis, editores, *Avances en sistemas de procesamiento de información neuronal 20 (NIPS'07)*, páginas 1121–1128, Cambridge, MA. Prensa del MIT.<sup>632</sup>
- Ovidio y Martín, C. (2004). *Metamorfosis*. W. W. Norton.<sup>1</sup>

- Paccanaro, A. y Hinton, GE (2000). Extracción de representaciones distribuidas de conceptos y relaciones a partir de proposiciones positivas y negativas. En *Conferencia Internacional Conjunta sobre Redes Neuronales (IJCNN)*, Como, Italia. IEEE, Nueva York.[484](#)
- Paine, TL, Khorrami, P., Han, W. y Huang, TS (2014). Un análisis de no supervisado pre-entrenamiento a la luz de los avances recientes. *pre impresión de arXiv arXiv:1412.6597*.[532](#)
- Palatucci, M., Pomerleau, D., Hinton, GE y Mitchell, TM (2009). Disparo cero aprendizaje con códigos de salida semánticos. En Y. Bengio, D. Schuurmans, JD Lafferty, CKI Williams y A. Culotta, editores, *Avances en los sistemas de procesamiento de información neuronal* 22, páginas 1410–1418. Curran Associates, Inc.[539](#)
- Parker, DB (1985). Lógica de aprendizaje. Informe Técnico TR-47, Centro de Comp. Investigación en Ciencias Económicas y Administrativas, MIT.[225](#)
- Pascanu, R., Mikolov, T. y Bengio, Y. (2013). Sobre la dificultad del entrenamiento recurrente Redes neuronales. En *ICML'2013*.[289,402,403,408,414,416](#)
- Pascanu, R., Gülcéhre, Ç., Cho, K. y Bengio, Y. (2014a). Cómo construir profundo redes neuronales recurrentes. En *ICLR'2014*.[19,265,398,399,410,460](#)
- Pascanu, R., Montufar, G. y Bengio, Y. (2014b). Sobre el número de regiones de inferencia de redes feed-forward profundas con activaciones lineales por partes. En *ICLR'2014*.[550](#)
- Pati, Y., Rezaifar, R. y Krishnaprasad, P. (1993). Persecución de emparejamiento ortogonal: Aproximación de funciones recursivas con aplicaciones a la descomposición wavelet. En *Actas de la 27.ª Conferencia Anual de Asilomar sobre Señales, Sistemas y Computadores*, páginas 40–44.[255](#)
- Perla, J. (1985). Redes bayesianas: un modelo de memoria autoactivada para pruebas razonamiento. En *Actas de la 7.ª Conferencia de la Sociedad de Ciencias Cognitivas, Universidad de California, Irvine*, páginas 329–334.[563](#)
- Perla, J. (1988). *Razonamiento Probabilístico en Sistemas Inteligentes: Redes de Plausibles Inferencia*. Morgan Kaufman.[54](#)
- Perron, O. (1907). Zur theorie der matrix. *Anales matemáticos*, **64**(2), 248–263.[597](#)
- Petersen, KB y Pedersen, MS (2006). El libro de cocina matriz. Versión 20051003.[31](#)
- Peterson, GB (2004). Un día de gran iluminación: el descubrimiento de la conformación de BF Skinner. *Revista de Análisis Experimental del Comportamiento*, **82**(3), 317–328.[328](#)
- Pham, D.-T., Garat, P. y Jutten, C. (1992). Separación de una mezcla de independientes. fuentes a través de un enfoque de máxima verosimilitud. En *EUSIPCO*, páginas 771–774.[491](#)

- Pham, P.-H., Jelaca, D., Farabet, C., Martini, B., LeCun, Y. y Culurciello, E. (2012). NeuFlow: sistema de procesamiento de visión de flujo de datos en un chip. En *Circuitos y sistemas (MWS-CAS), 2012 IEEE 55th International Midwest Symposium on*, páginas 1044-1047. IEEE. [451](#)
- Pinheiro, OPS y Collobert, R. (2014). Redes neuronales convolucionales recurrentes para rotulación de escenas. En *ICML'2014*. [359](#)
- Pinheiro, OPS y Collobert, R. (2015). Del etiquetado a nivel de imagen a nivel de píxel con redes convolucionales. En *Conferencia sobre Visión por Computador y Reconocimiento de Patrones (CVPR)*. [359](#)
- Pinto, N., Cox, DD y DiCarlo, JJ (2008). ¿Por qué el reconocimiento de objetos visuales del mundo real ¿duro? *PLoS Comput Biol*, **4**. [456](#)
- Pinto, N., Stone, Z., Zickler, T. y Cox, D. (2011). Escalando inspirado biológicamente visión por computadora: un estudio de caso en el reconocimiento facial sin restricciones en facebook. En *Talleres de visión por computadora y reconocimiento de patrones (CVPRW), Conferencia de la Sociedad de Computación IEEE 2011 sobre*, páginas 35-42. IEEE. [363](#)
- Pollack, JB (1990). Representaciones distribuidas recursivas. *Inteligencia artificial*, **46**(1), 77-105. [401](#)
- Polyak, B. y Juditsky, A. (1992). Aceleración de la aproximación estocástica por promedio. *SIAM J. Control y Optimización*, **30**(4), 838-855. [322](#)
- Polyak, BT (1964). Algunos métodos para acelerar la convergencia de los métodos de iteración. *URSS Matemática Computacional y Física Matemática*, **4**(5), 1-17. [296](#)
- Poole, B., Sohl-Dickstein, J. y Ganguli, S. (2014). Análisis de ruido en codificadores automáticos y redes profundas. *CoRR,abs/1406.1831*. [241](#)
- Poon, H. y Domingos, P. (2011). Redes de suma-producto: una nueva arquitectura profunda. En *Actas de las XXII Jornadas de Incertidumbre en Inteligencia Artificial (UAI), Barcelona, España*. [554](#)
- Presley, RK y Haggard, RL (1994). Una implementación de punto fijo de la backpropagación algoritmo de aprendizaje de gación. En *Sudestecon'94. Transferencia de tecnología creativa: un asunto global. Actas del IEEE de 1994*, páginas 136-138. IEEE. [451](#)
- Precio, R. (1958). Un teorema útil para dispositivos no lineales con entradas gaussianas. *IEEE Transacciones sobre teoría de la información*, **4**(2), 69-72. [689](#)
- Quiroga, RQ, Reddy, L., Kreiman, G., Koch, C. y Fried, I. (2005). visual invariant representation por neuronas individuales en el cerebro humano. *Naturaleza*, **435**(7045), 1102-1107. [366](#)

- Radford, A., Metz, L. y Chintala, S. (2015). Aprendizaje de representación no supervisado con redes antagónicas generativas convolucionales profundas. *preimpresión de arXiv arXiv:1511.06434*. 552, 701,702
- Raiko, T., Yao, L., Cho, K. y Bengio, Y. (2014). Autorregresivo neural iterativo estimador de distribución (NADE-k). Informe técnico, arXiv:1406.1485. 676,709
- Raina, R., Madhavan, A. y Ng, AY (2009). Aprendizaje profundo no supervisado a gran escala utilizando procesadores gráficos. En L. Bottou y M. Littman, editores, *Actas de la Vigésima Sexta Conferencia Internacional sobre Aprendizaje Automático (ICML'09)*, páginas 873–880, Nueva York, NY, EE. UU. ACM. 27,446
- Ramsey, FP (1926). Verdad y probabilidad. En RB Braithwaite, editor, *Los cimientos de Matemáticas y otros Ensayos Lógicos*, capítulo 7, páginas 156–198. Archivo de la Universidad McMaster para la Historia del Pensamiento Económico. 56
- Ranzato, M. y Hinton, GH (2010). Modelado de medias de píxeles y covarianzas usando Máquinas de Boltzmann de tercer orden factorizadas. En *CVPR'2010*, páginas 2551–2558. 680
- Ranzato, M., Poultney, C., Chopra, S. y LeCun, Y. (2007a). Aprendizaje eficiente de escasos representaciones con un modelo basado en la energía. En *NIPS'2006*. 14,19,507,528,530
- Ranzato, M., Huang, F., Boureau, Y. y LeCun, Y. (2007b). Aprendizaje no supervisado de jerarquías de características invariantes con aplicaciones para el reconocimiento de objetos. En *Actas de la Conferencia de reconocimiento de patrones y visión artificial (CVPR'07)*. Prensa IEEE. 364
- Ranzato, M., Boureau, Y. y LeCun, Y. (2008). Aprendizaje de características escasas para una creencia profunda redes En *NIPS'2007*. 507
- Ranzato, M., Krizhevsky, A. y Hinton, GE (2010a). Restringido de 3 vías factorizado Máquinas de Boltzmann para el modelado de imágenes naturales. En *Actas de AISTATS 2010*. 678, 679
- Ranzato, M., Mnih, V. y Hinton, G. (2010b). Generando imágenes más realistas usando MRF cerrados. En *NIPS'2010*. 680
- Rao, C. (1945). La información y la precisión alcanzable en la estimación de estadísticas parámetros *Boletín de la Sociedad Matemática de Calcuta*, 37, 81–89. 135,295
- Rasmus, A., Valpola, H., Honkala, M., Berglund, M. y Raiko, T. (2015). Semi-supervisado aprendizaje con red de escalera. *preimpresión de arXiv arXiv:1507.02672*. 426,530
- Recht, B., Re, C., Wright, S. y Niu, F. (2011). Hogwild: un enfoque sin bloqueo para descenso de gradiente estocástico paralelizado. En *NIPS'2011*. 447
- Reichert, DP, Seriès, P. y Storkey, AJ (2011). Adaptación neuronal para muestreo-inferencia probabilística basada en la biestabilidad perceptiva. En *Avances en sistemas de procesamiento de información neuronal*, páginas 2357–2365. 666

- Rezende, DJ, Mohamed, S. y Wierstra, D. (2014). Retropropagación estocástica e inferencia aproximada en modelos generativos profundos. En *ICML'2014*. Preimpresión: arXiv:1401.4082.[652,689,696](#)
- Rifai, S., Vincent, P., Muller, X., Glorot, X. y Bengio, Y. (2011a). Contractivo codificadores automáticos: invariancia explícita durante la extracción de características. En *ICML'2011*.[521,522, 523](#)
- Rifai, S., Mesnil, G., Vincent, P., Muller, X., Bengio, Y., Dauphin, Y. y Glorot, X. (2011b). Codificador automático contractivo de orden superior. En *ECML PKDD*.[521,522](#)
- Rifai, S., Dauphin, Y., Vincent, P., Bengio, Y. y Muller, X. (2011c). el múltiple clasificador tangente. En *NIPS'2011*.[271,272,523](#)
- Rifai, S., Bengio, Y., Dauphin, Y. y Vincent, P. (2012). Un proceso generativo de Muestreo de codificadores automáticos contractivos. En *ICML'2012*.[711](#)
- Ringach, D. y Shapley, R. (2004). Correlación inversa en neurofisiología. *Cognitivo Ciencia*, **28**(2), 147–166.[368](#)
- Roberts, S. y Everson, R. (2001). *Análisis de componentes independientes: principios y práctica*. Prensa de la Universidad de Cambridge.[493](#)
- Robinson, AJ y Fallside, F. (1991). Un discurso de red de propagación de errores recurrente sistema de reconocimiento *Habla y lenguaje informático*, **5**(3), 259–274.[27,459](#)
- Rockafellar, RT (1997). Análisis convexo. Puntos de referencia de princeton en matemáticas.[93](#)
- Romero, A., Ballas, N., Ebrahimi Kahou, S., Chassang, A., Gatta, C. y Bengio, Y. (2015). Redes ajustadas: Consejos para redes delgadas y profundas. En *ICLR'2015*, arXiv:1412.6550.[325](#)
- Rosen, JB (1960). El método de proyección de gradiente para programación no lineal. parte i. restricciones lineales. *Revista de la Sociedad de Matemáticas Industriales y Aplicadas*, **8**(1), págs. 181–217.[93](#)
- Rosenblatt, F. (1958). El perceptrón: Un modelo probabilístico para el almacenamiento de información y organización en el cerebro. *Revisión psicológica*, **sesenta y cinco**, 386–408.[14,15,27](#)
- Rosenblatt, F. (1962). *Principios de la neurodinámica*. Espartano, Nueva York.[15,27](#)
- Roweis, S. y Saul, LK (2000). Reducción de dimensionalidad no lineal por localmente lineal incrustación *Ciencia*, **290**(5500).[164,518](#)
- Roweis, S., Saul, L. y Hinton, G. (2002). Coordinación global de modelos lineales locales. En T. Dietterich, S. Becker y Z. Ghahramani, editores, *Avances en sistemas de procesamiento de información neuronal 14 (NIPS'01)*, Cambridge, MA. Prensa del MIT.[489](#)
- Rubín, DB et al.(1984). Cálculos de frecuencia relevantes y justificables bayesianamente para la estadística aplicada. *Los Anales de Estadística*, **12**(4), 1151-1172.[717](#)

Rumelhart, D., Hinton, G. y Williams, R. (1986a). Aprendizaje de representaciones por errores de retropropagación. *Naturaleza*, **323**, 533–536.[14](#),[18](#),[23](#),[204](#),[225](#),[373](#),[476](#),[482](#)

Rumelhart, DE, Hinton, GE y Williams, RJ (1986b). Aprendizaje de representación interna taciones por propagación de errores. En DE Rumelhart y JL McClelland, editores, *Procesamiento distribuido en paralelo*, volumen 1, capítulo 8, páginas 318–362. MIT Press, Cambridge.[21](#),[27](#),[225](#)

Rumelhart, DE, McClelland, JL y el Grupo de Investigación PDP (1986c). *Paralelo Procesamiento distribuido: exploraciones en la microestructura de la cognición*. MIT Press, Cambridge.[17](#)

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, AC y Fei-Fei, L. (2014a). Desafío de reconocimiento visual a gran escala de ImageNet.[21](#)

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.(2014b). Desafío de reconocimiento visual a gran escala Imagenet. *pre impresión de arXiv arXiv:1409.0575*.[28](#)

Russell, SJ y Norvig, P. (2003). *Inteligencia artificial: un enfoque moderno*. Aprendiz Sala.[86](#)

Rust, N., Schwartz, O., Movshon, JA y Simoncelli, E. (2005). Espaciotemporal elementos de los campos receptivos V1 de macaco. *Neurona*, **46**(6), 945–956.[367](#)

Sainath, T., Mohamed, A., Kingsbury, B. y Ramabhadran, B. (2013). profundo convolu-redes neuronales nacionales para LVCSR. En *ICASSP 2013*.[460](#)

Salakhutdinov, R. (2010). Aprendizaje en campos aleatorios de Markov utilizando transiciones temperadas. En Y. Bengio, D. Schuurmans, C. Williams, J. Lafferty y A. Culotta, editores, *Avances en sistemas de procesamiento de información neuronal 22 (NIPS'09)*.[603](#)

Salakhutdinov, R. y Hinton, G. (2009a). Máquinas profundas de Boltzmann. En *Procedimientos de la Conferencia Internacional sobre Inteligencia Artificial y Estadística*, volumen 5, páginas 448–455.[24](#),[27](#),[529](#),[663](#),[666](#),[671](#),[672](#)

Salakhutdinov, R. y Hinton, G. (2009b). Hashing semántico. En *Revista Internacional de Razonamiento aproximado*.[525](#)

Salakhutdinov, R. y Hinton, GE (2007a). Aprendiendo una incrustación no lineal por preservando la estructura de barrio de clases. En *Actas de la Undécima Conferencia Internacional sobre Inteligencia Artificial y Estadísticas (AISTATS'07)*, San Juan, Puerto Rico. Omnipress.[527](#)

Salakhutdinov, R. y Hinton, GE (2007b). Hashing semántico. En *SIGIR'2007*.[525](#)

- Salakhutdinov, R. y Hinton, GE (2008). Uso de redes de creencias profundas para aprender covarianza Núcleos para procesos gaussianos. En J. Platt, D. Koller, Y. Singer y S. Roweis, editores, *Avances en sistemas de procesamiento de información neuronal 20 (NIPS'07)*, páginas 1249–1256, Cambridge, MA. Prensa del MIT.[244](#)
- Salakhutdinov, R. y Larochelle, H. (2010). Aprendizaje eficiente de máquinas profundas de Boltzmann. En *Actas de la Decimotercera Conferencia Internacional sobre Inteligencia Artificial y Estadísticas (AISTATS 2010), JMLR W&CP*, volumen 9, páginas 693–700.[652](#)
- Salakhutdinov, R. y Mnih, A. (2008). Factorización de matrices probabilísticas. En *NIPS'2008*.  
[480](#)
- Salakhutdinov, R. y Murray, I. (2008). Sobre el análisis cuantitativo de la creencia profunda redes En WW Cohen, A. McCallum y ST Roweis, editores, *Actas de la Vigésimoquinta Conferencia Internacional sobre Aprendizaje Automático (ICML'08)*, volumen 25, páginas 872–879. ACM.[628,662](#)
- Salakhutdinov, R., Mnih, A. y Hinton, G. (2007). Máquinas Boltzmann restringidas para filtración colaborativa. En *ICML*.[480](#)
- Sanger, TD (1994). Control de aprendizaje de redes neuronales de manipuladores de robots usando aumentando gradualmente la dificultad de la tarea. *Transacciones IEEE sobre robótica y automatización*, **10**(3).[328](#)
- Saúl, LK y Jordan, MI (1996). Explotación de subestructuras tratables en intratables redes En D. Touretzky, M. Mozer y M. Hasselmo, editores, *Avances en sistemas de procesamiento de información neuronal 8 (NIPS'95)*. Prensa del MIT, Cambridge, MA.[638](#)
- Saul, LK, Jaakkola, T. y Jordan, MI (1996). Teoría del campo medio para la creencia sigmoidea redes *Revista de investigación de inteligencia artificial*, **4**, 61–76.[27,693](#)
- Savich, AW, Moussa, M. y Areibi, S. (2007). El impacto de la representación aritmética sobre la implementación de mlp-bp en fpgas: un estudio. *Redes neuronales, transacciones IEEE en*, **18**(1), 240–252.[451](#)
- Saxe, AM, Koh, PW, Chen, Z., Bhand, M., Suresh, B. y Ng, A. (2011). al azar pesos y aprendizaje de funciones no supervisado. En *proc. ICML'2011*. ACM.[363](#)
- Saxe, AM, McClelland, JL y Ganguli, S. (2013). Soluciones exactas a la no lineal. Dinámica del aprendizaje en redes neuronales lineales profundas. En *ICLR*.[285,286,303](#)
- Schaul, T., Antonoglou, I. y Silver, D. (2014). Pruebas unitarias para optimización estocástica. En *Conferencia Internacional sobre Representaciones de Aprendizaje*.[309](#)
- Schmidhuber, J. (1992). Aprender secuencias complejas y extendidas usando el principio de compresión de la historia. *Computación neuronal*, **4**(2), 234–242.[398](#)
- Schmidhuber, J. (1996). Compresión de texto neuronal secuencial. *Transacciones IEEE en Neural Redes*, **7**(1), 142–146.[477](#)

- Schmidhuber, J. (2012). Redes neuronales autodelimitantes. *preimpresión de arXiv arXiv:1210.0118*.  
**390**
- Schölkopf, B. y Smola, AJ (2002). *Aprendizaje con núcleos: máquinas de vectores de soporte, regularización, optimización y más allá*. Prensa del MIT.  
**704**
- Schölkopf, B., Smola, A. y Müller, K.-R. (1998). Análisis de componentes no lineales como Problema de valores propios del núcleo. *Computación neuronal*, **10**, 1299–1319.  
**164,518**
- Schölkopf, B., Burges, CJC y Smola, AJ (1999). *Avances en los métodos del núcleo — Aprendizaje de vectores de soporte*. Prensa del MIT, Cambridge, MA.  
**18,142**
- Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K. y Mooij, J. (2012). En Aprendizaje causal y anticausal. En *ICML'2012*, páginas 1255–1262.  
**545**
- Schuster, M. (1999). Sobre el aprendizaje supervisado a partir de datos secuenciales con aplicaciones para reconocimiento de voz.  
**190**
- Schuster, M. y Paliwal, K. (1997). Redes neuronales recurrentes bidireccionales. *IEEE Transacciones en procesamiento de señales*, **45**(11), 2673–2681.  
**395**
- Schwenk, H. (2007). Modelos de lenguaje espacial continuo. *Habla y lenguaje informático*, **21**, 492–518.  
**466**
- Schwenk, H. (2010). Modelos de lenguaje espacial continuo para traducción automática estadística. *El Boletín de Praga de Lingüística Matemática*, **93**, 137–146.  
**473**
- Schwenk, H. (2014). Subconjunto limpio del conjunto de datos WMT '14.  
**21**
- Schwenk, H. y Bengio, Y. (1998). Métodos de entrenamiento para el refuerzo adaptativo de redes neuronales. obras. En M. Jordan, M. Kearns y S. Solla, editores, *Avances en sistemas de procesamiento de información neuronal 10 (NIPS'97)*, páginas 647–653. Prensa del MIT.  
**258**
- Schwenk, H. y Gauvain, J.-L. (2002). Modelado del lenguaje conexionista para grandes reconocimiento de voz continuo de vocabulario. En *Conferencia Internacional sobre Acústica, Habla y Procesamiento de Señales (ICASSP)*, páginas 765–768, Orlando, Florida.  
**466**
- Schwenk, H., Costa-jussà, MR y Fonollosa, JAR (2006). espacio continuo modelos de lenguaje para la tarea IWSLT 2006. En *Taller Internacional de Traducción de Lenguas Habladas*, páginas 166–173.  
**473**
- Seide, F., Li, G. y Yu, D. (2011). Transcripción del habla conversacional usando context-redes neuronales profundas dependientes. En *Interdiscurso 2011*, páginas 437–440.  
**23**
- Sejnowski, T. (1987). Máquinas Boltzmann de orden superior. En *Actas de la conferencia AIP 151 sobre redes neuronales para computación*, páginas 398–403. Instituto Americano de Física Inc.  
**686**

- Series, P., Reichert, DP y Storkey, AJ (2010). Alucinaciones en Charles Bonnet síndrome inducido por la homeostasis: un modelo profundo de máquina de Boltzmann. En *Avances en sistemas de procesamiento de información neuronal*, páginas 2020–2028.[666](#)
- Sermanet, P., Chintala, S. y LeCun, Y. (2012). Redes neuronales convolucionales aplicadas a la clasificación de los dígitos de los números de las casas. *CoRR*,[abs/1204.3968](#).[457](#)
- Sermanet, P., Kavukcuoglu, K., Chintala, S. y LeCun, Y. (2013). Detección de peatones con aprendizaje de características de múltiples etapas no supervisado. En *proc. Congreso Internacional de Visión por Computador y Reconocimiento de Patrones (CVPR'13)*. IEEE.[23,201](#)
- Shilov, G. (1977). *Álgebra lineal*. Libros de Dover sobre series de matemáticas. Publicaciones de Dover.  
[31](#)
- Siegelmann, H. (1995). Cálculo más allá del límite de Turing. *Ciencia*,[268](#)(5210), 545–548.[379](#)
- Siegelmann, H. y Sontag, E. (1991). Turing computabilidad con redes neuronales. *Aplicado Letras de Matemáticas*,[4](#)(6), 77–80.[379](#)
- Siegelmann, HT y Sontag, ED (1995). Sobre el poder computacional de las redes neuronales. *Revista de Ciencias de la Computación y Sistemas*,[50](#)(1), 132–150.[379,403](#)
- Sietsma, J. y Dow, R. (1991). Crear redes neuronales artificiales que generalicen. *Neural Redes*,[4](#)(1), 67–79.[241](#)
- Simard, D., Steinkraus, PY y Platt, JC (2003). Mejores prácticas para convolucional Redes neuronales. En *ICDAR'2003*.[371](#)
- Simard, P. y Graf, HP (1994). Retropropagación sin multiplicación. En *Avances en Sistemas de Procesamiento de Información Neural*, páginas 232–239.[451](#)
- Simard, P., Victorri, B., LeCun, Y. y Denker, J. (1992). Puntal tangente - Un formalismo para especificar invariancias seleccionadas en una red adaptativa. En *NIPS'1991*.[270,271,272](#), [356](#)
- Simard, PY, LeCun, Y. y Denker, J. (1993). Reconocimiento eficiente de patrones usando un nueva distancia de transformación. En *NIPS'92*.[270](#)
- Simard, PY, LeCun, YA, Denker, JS y Victorri, B. (1998). Transformación invariancia en el reconocimiento de patrones: distancia tangente y propagación tangente. *Apuntes de clase en informática*,[1524](#).[270](#)
- Simons, DJ y Levin, DT (1998). No detectar cambios en las personas durante un interacción del mundo real. *Boletín psiconómico y revisión*,[5](#)(4), 644–649.[543](#)
- Simonyan, K. y Zisserman, A. (2015). Redes convolucionales muy profundas para aplicaciones a gran escala. reconocimiento de imagen. En *ICLR*.[323](#)

- Sjöberg, J. y Ljung, L. (1995). Sobreentrenamiento, regularización y búsqueda de un mínimo, con aplicación a redes neuronales. *Revista Internacional de Control*, **62**(6), 1391–1407. [250](#)
- Skinner, BF (1958). Refuerzo hoy. *psicólogo estadounidense*, **13**, 94–99. [328](#)
- Smolensky, P. (1986). Procesamiento de información en sistemas dinámicos: Fundamentos de teoría de la armonía. En DE Rumelhart y JL McClelland, editores, *Procesamiento distribuido en paralelo*, volumen 1, capítulo 6, páginas 194–281. MIT Press, Cambridge. [571,587,656](#)
- Snoek, J., Larochelle, H. y Adams, RP (2012). Práctica optimización bayesiana de algoritmos de aprendizaje automático. En *NIPS'2012*. [436](#)
- Socher, R., Huang, EH, Pennington, J., Ng, AY y Manning, CD (2011a). Dinámica agrupación y despliegue de codificadores automáticos recursivos para la detección de paráfrasis. En *NIPS'2011*. [401](#)
- Socher, R., Manning, C. y Ng, AY (2011b). Análisis de escenas naturales y lenguaje natural. calibre con redes neuronales recursivas. En *Actas de la Vigésima Octava Conferencia Internacional sobre Aprendizaje Automático (ICML'2011)*. [401](#)
- Socher, R., Pennington, J., Huang, EH, Ng, AY y Manning, CD (2011c). Codificadores automáticos recursivos semisupervisados para predecir distribuciones de sentimientos. En *EMNLP'2011*. [401](#)
- Socher, R., Perelygin, A., Wu, JY, Chuang, J., Manning, CD, Ng, AY y Potts, C. (2013a). Modelos profundos recursivos para la composicionalidad semántica sobre un banco de árboles de sentimientos. En *EMNLP'2013*. [401](#)
- Socher, R., Ganjoo, M., Manning, CD y Ng, AY (2013b). Aprendizaje de tiro cero a través de transferencia intermodal. En *27.ª Conferencia anual sobre sistemas de procesamiento de información neuronal (NIPS 2013)*. [539](#)
- Sohl-Dickstein, J., Weiss, EA, Maheswaranathan, N. y Ganguli, S. (2015). Profundo aprendizaje no supervisado usando termodinámica de no equilibrio. [716](#)
- Sohn, K., Zhou, G. y Lee, H. (2013). Aprendizaje y selección de características junto con Máquinas de Boltzmann con compuertas puntuales. En *ICML'2013*. [687](#)
- Solomonoff, RJ (1989). Un sistema de aprendizaje incremental basado en probabilística algorítmica habilidad [328](#)
- Sontag, ED (1998). Dimensión VC de las redes neuronales. *Computadora NATO ASI Serie F y Ciencias de Sistemas*, **168**, 69–96. [547,551](#)
- Sontag, ED y Sussman, HJ (1989). La retropropagación puede dar lugar a errores locales mínimos incluso para redes sin capas ocultas. *Sistemas complejos*, **3**, 91–106. [284](#)

- Sparkes, B. (1996). *El rojo y el negro: estudios en cerámica griega*. Routledge.<sup>1</sup>
- Spitkovsky, VI, Alshawi, H. y Jurafsky, D. (2010). De los pasos de bebé al salto de rana: cómo “menos es más” en el análisis de dependencia no supervisado. En *HLT'10*.<sup>328</sup>
- Squire, W. y Trapp, G. (1998). Uso de variables complejas para estimar derivadas de valores reales funciones *SIAM Rev.*, **40**(1), 110–112.<sup>439</sup>
- Srebro, N. y Shraibman, A. (2005). Rango, norma de seguimiento y norma máxima. En *Procedimientos de la 18ª Conferencia Anual sobre Teoría del Aprendizaje*, páginas 545–560. Springer-Verlag.<sup>238</sup>
- Srivastava, N. (2013). *Mejora de las redes neuronales con abandono*. tesis de maestría, u. toronto<sup>535</sup>
- Srivastava, N. y Salakhutdinov, R. (2012). Aprendizaje multimodal con Boltzmann profundo máquinas. En *NIPS'2012*.<sup>541</sup>
- Srivastava, N., Salakhutdinov, RR y Hinton, GE (2013). Modelado de documentos con máquinas profundas de Boltzmann. *pre impresión de arXiv arXiv:1309.6865*.<sup>663</sup>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. y Salakhutdinov, R. (2014). Abandono: una forma sencilla de evitar el sobreajuste de las redes neuronales. *Revista de investigación de aprendizaje automático*, **15**, 1929–1958.<sup>258,265,267,672</sup>
- Srivastava, RK, Greff, K. y Schmidhuber, J. (2015). Redes de carreteras. *arXiv:1505.00387*.<sup>326</sup>
- Steinkrau, D., Simard, PY y Buck, I. (2005). Uso de GPU para el aprendizaje automático algoritmos *2013 12ª Conferencia Internacional sobre Análisis y Reconocimiento de Documentos*, **0**, 1115–1119.<sup>445</sup>
- Stoyanov, V., Ropson, A. y Eisner, J. (2011). Minimización empírica del riesgo de gráficos parámetros del modelo dada la inferencia aproximada, la decodificación y la estructura del modelo. En *Actas de la 14ª Conferencia Internacional sobre Inteligencia Artificial y Estadísticas (AISTATS)*, volumen 15 de *Actas del taller y la conferencia de la JMLR*, páginas 725–733, Fort Lauderdale. Material complementario (4 páginas) también disponible.<sup>674,698</sup>
- Sukhbaatar, S., Szlam, A., Weston, J. y Fergus, R. (2015). Memoria débilmente supervisada redes *pre impresión de arXiv arXiv:1503.08895*.<sup>418</sup>
- Supancic, J. y Ramanan, D. (2013). Aprendizaje a su propio ritmo para el seguimiento a largo plazo. En *CVPR'2013*.<sup>328</sup>
- Sussillo, D. (2014). Paseos aleatorios: entrenamiento de redes feed-forward no lineales muy profundas con inicialización inteligente. *CoRR,abs/1412.6558*.<sup>290,303,305,403</sup>
- Sutskever, I. (2012). *Entrenamiento de redes neuronales recurrentes*. Doctor. tesis, Departamento de informática, Universidad de Toronto.<sup>406,413</sup>

- Sutskever, I. y Hinton, GE (2008). Las redes profundas y estrechas de creencias sigmoideas son universales aproximadores. *Computación neuronal*, **20**(11), 2629–2636. [693](#)
- Sutskever, I. y Tieleman, T. (2010). Sobre las propiedades de convergencia del contrastivo Divergencia. En YW Teh y M. Titterington, editores, *proc. de la Conferencia Internacional sobre Inteligencia Artificial y Estadística (AISTATS)*, volumen 9, páginas 789–795. [612](#)
- Sutskever, I., Hinton, G. y Taylor, G. (2009). El temporal restringido recurrente Máquina de Boltzmann. En *NIPS'2008*. [685](#)
- Sutskever, I., Martens, J. y Hinton, GE (2011). Generando texto con recurrente Redes neuronales. En *ICML'2011*, páginas 1017–1024. [477](#)
- Sutskever, I., Martens, J., Dahl, G. y Hinton, G. (2013). Sobre la importancia de Inicialización e impulso en el aprendizaje profundo. En *ICML*. [300,406,413](#)
- Sutskever, I., Vinyals, O. y Le, QV (2014). Secuencia a secuencia de aprendizaje con Redes neuronales. En *NIPS'2014, arXiv:1409.3215.25,101,397,410,411,474,475*
- Sutton, R. y Barto, A. (1998). *Aprendizaje por refuerzo: una introducción*. Prensa del MIT. [106](#)
- Sutton, RS, Mcallester, D., Singh, S. y Mansour, Y. (2000). Métodos de gradiente de políticas para el aprendizaje por refuerzo con aproximación de funciones. En *NIPS'1999*, páginas 1057– – 1063. Prensa del MIT. [691](#)
- Swersky, K., Ranzato, M., Buchman, D., Marlin, B. y de Freitas, N. (2011). En codificadores automáticos y coincidencia de puntuación para modelos basados en energía. En *ICML'2011. ACM*. [513](#)
- Swersky, K., Snoek, J. y Adams, RP (2014). Optimización bayesiana de congelación y descongelación. *preimpresión de arXiv arXiv:1406.3896.436*
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. y Rabinovich, A. (2014a). Profundizando con las circunvoluciones. Informe técnico, *arXiv:1409.4842.24,27,201,258,269,326,347*
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, IJ y Fergus, R. (2014b). Propiedades intrigantes de las redes neuronales. *ICLR,abs/1312.6199. 268, 271*
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. y Wojna, Z. (2015). Repensar el Arquitectura inicial para visión artificial. *Impresiones electrónicas ArXiv:243,322*
- Taigman, Y., Yang, M., Ranzato, M. y Wolf, L. (2014). DeepFace: Cerrando la brecha para rendimiento a nivel humano en la verificación facial. En *CVPR'2014*. [100](#)
- Tandy, DW (1997). *Works and Days: A Translation and Commentary for the Social Ciencias*. Prensa de la Universidad de California. [1](#)

- Tang, Y. y Eliasmith, C. (2010). Redes profundas para un reconocimiento visual robusto. En *Actas de la 27.<sup>a</sup> Conferencia Internacional sobre Aprendizaje Automático, 21-24 de junio de 2010, Haifa, Israel.*241
- Tang, Y., Salakhutdinov, R. y Hinton, G. (2012). Mezclas profundas de analizadores factoriales. *preimpresión de arXiv arXiv:1206.4635.*489
- Taylor, G. y Hinton, G. (2009). Máquinas de Boltzmann restringidas condicionales factorizadas para modelar el estilo de movimiento. En L. Bottou y M. Littman, editores, *Actas de la Vigésima Sexta Conferencia Internacional sobre Aprendizaje Automático (ICML'09)*, páginas 1025–1032, Montreal, Quebec, Canadá. ACM.685
- Taylor, G., Hinton, GE y Roweis, S. (2007). Modelado del movimiento humano usando binario variables latentes. En B. Schölkopf, J. Platt y T. Hoffman, editores, *Avances en Sistemas de Procesamiento de Información Neural 19 (NIPS'06)*, páginas 1345–1352. Prensa del MIT, Cambridge, MA.685
- Teh, Y., Welling, M., Osindero, S. y Hinton, GE (2003). Modelos basados en energía para representaciones escasas y sobrecompletas. *Revista de investigación de aprendizaje automático*,4, 1235–1260.491
- Tenenbaum, J., de Silva, V. y Langford, JC (2000). Un marco geométrico global para reducción de dimensionalidad no lineal. *Ciencia*,290(5500), 2319–2323.164,518,533
- Theis, L., van den Oord, A. y Bethge, M. (2015). Una nota sobre la evaluación de la generación modelos arXiv:1511.01844.698,719
- Thompson, J., Jain, A., LeCun, Y. y Bregler, C. (2014). Entrenamiento conjunto de un convolucional red y un modelo gráfico para la estimación de la pose humana. En *NIPS'2014*.360
- Thrun, S. (1995). Aprendiendo a jugar el juego de ajedrez. En *NIPS'1994*.271
- Tibshirani, RJ (1995). Encogimiento de regresión y selección a través del lazo. *Diario de la Sociedad Real de Estadística B*,58, 267–288.236
- Tieleman, T. (2008). Entrenamiento de máquinas Boltzmann restringidas usando aproximaciones a el gradiente de probabilidad. En WW Cohen, A. McCallum y ST Roweis, editores, *Actas de la Vigesimoquinta Conferencia Internacional sobre Aprendizaje Automático (ICML'08)*, páginas 1064–1071. ACM.612
- Tieleman, T. y Hinton, G. (2009). Uso de pesos rápidos para mejorar el contraste persistente divergencia. En L. Bottou y M. Littman, editores, *Actas de la Vigésima Sexta Conferencia Internacional sobre Aprendizaje Automático (ICML'09)*, páginas 1033–1040. ACM. 614
- Tipping, ME y Bishop, CM (1999). Análisis probabilístico de componentes principales. *Revista de la Real Sociedad Estadística B*,61(3), 611–622.491

- Torralba, A., Fergus, R. y Weiss, Y. (2008). Pequeños códigos y grandes bases de datos para reconocimiento. En *Actas de la Conferencia de reconocimiento de patrones y visión artificial (CVPR'08)*, páginas 1–8.[525](#)
- Touretzky, DS y Minton, GE (1985). Símbolos entre las neuronas: Detalles de una arquitectura de inferencia conexionista. En *Actas de la 9ª Conferencia Internacional Conjunta sobre Inteligencia Artificial - Volumen 1, IJCAI'85*, páginas 238–243, San Francisco, CA, EE. UU. Morgan Kaufmann Publishers Inc.[17](#)
- Tu, K. y Honavar, V. (2011). Sobre la utilidad de los currículos en el aprendizaje no supervisado de gramáticas probabilísticas. En *IJCAI'2011*.[328](#)
- Turaga, SC, Murray, JF, Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W. y Seung, HS (2010). Las redes convolucionales pueden aprender a generar gráficos de afinidad para la segmentación de imágenes. *Computación neuronal*, **22**(2), 511–538.[360](#)
- Turian, J., Ratinov, L. y Bengio, Y. (2010). Representaciones de palabras: Una forma simple y método general para el aprendizaje semi-supervisado. En *proc. ACL'2010*, páginas 384–394.[535](#)
- Töscher, A., Jahrer, M. y Bell, RM (2009). La solución BigChaos a Netflix gran Premio.[480](#)
- Uriel, B., Murray, I. y Larochelle, H. (2013). Rnade: la autorregresión neuronal de valor real estimador de densidad sive. En *NIPS'2013*.[709,710](#)
- van den Oord, A., Dieleman, S. y Schrauwen, B. (2013). Música profunda basada en contenido recomendación. En *NIPS'2013*.[480](#)
- van der Maaten, L. y Hinton, GE (2008). Visualización de datos usando t-SNE. *J máquina Resolución de aprendizaje*, **9**.[477,519](#)
- Vanhoutte, V., Senior, A. y Mao, MZ (2011). Mejorando la velocidad de las redes neuronales en las CPU. En *proc. Taller NIPS de aprendizaje profundo y aprendizaje de funciones no supervisado*.[444, 452](#)
- Vapnik, VN (1982). *Estimación de Dependencias Basada en Datos Empíricos*. Saltador-Verlag, Berlín.[114](#)
- Vapnik, VN (1995). *La naturaleza de la teoría del aprendizaje estadístico*. Springer, Nueva York.[114](#)
- Vapnik, VN y Chervonenkis, AY (1971). Sobre la convergencia uniforme de los relativos frecuencias de eventos a sus probabilidades. *Teoría de la probabilidad y sus aplicaciones*, **dieciséis**, 264–280.[114](#)
- Vicente, P. (2011). Una conexión entre la coincidencia de puntuación y los codificadores automáticos de eliminación de ruido. *Computación neuronal*, **23**(7).[513,515,712](#)

- Vicente, P. y Bengio, Y. (2003). Múltiples ventanas Parzen. En *NIPS'2002*. Prensa del MIT. **520**
- Vincent, P., Larochelle, H., Bengio, Y. y Manzagol, P.-A. (2008). Extracción y componer características robustas con codificadores automáticos que eliminan el ruido. En *ICML 2008*.**241,515**
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. y Manzagol, P.-A. (2010). apilado codificadores automáticos de eliminación de ruido: aprendizaje de representaciones útiles en una red profunda con un criterio de eliminación de ruido local. *J. Aprendizaje automático Res.*,**11,515**
- Vincent, P., de Brébisson, A. y Bouthillier, X. (2015). Actualización de gradiente exacta eficiente para entrenar redes profundas con objetivos dispersos muy grandes. En C. Cortes, ND Lawrence, DD Lee, M. Sugiyama y R. Garnett, editores, *Avances en los sistemas de procesamiento de información neuronal* 28, páginas 1108–1116. Curran Associates, Inc.**466**
- Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I. y Hinton, G. (2014a). La gramática como lengua extranjera. Informe técnico, arXiv:1412.7449.**410**
- Vinyals, O., Toshev, A., Bengio, S. y Erhan, D. (2014b). Mostrar y contar: una imagen neuronal generador de subtítulos arXiv 1411.4555.**410**
- Vinyals, O., Fortunato, M. y Jaitly, N. (2015a). Redes de punteros. *preimpresión de arXiv arXiv:1506.03134*.**418**
- Vinyals, O., Toshev, A., Bengio, S. y Erhan, D. (2015b). Mostrar y contar: una imagen neuronal generador de subtítulos En *CVPR'2015*. arXiv:1411.4555.**102**
- Viola, P. y Jones, M. (2001). Robusta detección de objetos en tiempo real. En *Internacional Revista de visión artificial*.**449**
- Visin, F., Kastner, K., Cho, K., Matteucci, M., Courville, A. y Bengio, Y. (2015). ReNet: una alternativa basada en redes neuronales recurrentes a las redes convolucionales. *preimpresión de arXiv arXiv:1505.00393*.**395**
- Von Melchner, L., Pallas, SL y Sur, M. (2000). Comportamiento visual mediado por la retina. proyecciones dirigidas a la vía auditiva. *Naturaleza*,**404**(6780), 871–876.**dieciséis**
- Apuesta, S., Wang, S. y Liang, P. (2013). El abandono del entrenamiento como regularización adaptativa. En *Avances en los sistemas de procesamiento de información neuronal* 26, páginas 351–359.**265**
- Waibel, A., Hanazawa, T., Hinton, GE, Shikano, K. y Lang, K. (1989). Fonema Reconocimiento mediante redes neuronales de retardo de tiempo. *Transacciones IEEE sobre acústica, voz y procesamiento de señales*,**37**, 328–339.**374,453,459**
- Wan, L., Zeiler, M., Zhang, S., LeCun, Y. y Fergus, R. (2013). Regularización de neural Redes usando dropconnect. En *ICML'2013*.**266**
- Wang, S. y Manning, C. (2013). Entrenamiento de abandono rápido. En *ICML'2013*.**266**

- Wang, Z., Zhang, J., Feng, J. y Chen, Z. (2014a). Gráfico de conocimiento y texto juntos incrustación Enproc. *EMNLP'2014*.[484](#)
- Wang, Z., Zhang, J., Feng, J. y Chen, Z. (2014b). Gráfico de conocimiento incrustado por traduciendo en hiperplanos. Enproc. *AAAI'2014*.[484](#)
- Warde-Farley, D., Goodfellow, IJ, Courville, A. y Bengio, Y. (2014). un empírico análisis de abandono en redes lineales por partes. En*ICLR'2014*.[262,266,267](#)
- Wawrzynek, J., Asanovic, K., Kingsbury, B., Johnson, D., Beck, J. y Morgan, N. (1996). Spert-II: Un sistema de microprocesador vectorial. *Computadora*,[29](#)(3), 79–86.[451](#)
- Weaver, L. y Tao, N. (2001). La línea de base de recompensa óptima para el refuerzo basado en gradientes aprendizaje mental. Enproc. *AUI'2001*, páginas 538–545.[691](#)
- Weinberger, KQ y Saúl, LK (2004). Aprendizaje no supervisado de múltiples imágenes por programación semidefinida. En*CVPR'2004*, páginas 988–995.[164,519](#)
- Weiss, Y., Torralba, A. y Fergus, R. (2008). Hashing espectral. En*PINZAS*, paginas 1753-1760.[525](#)
- Welling, M., Zemel, RS y Hinton, GE (2002). Impulso auto supervisado. En*Avances en Sistemas de Procesamiento de Información Neural*, páginas 665–672.[703](#)
- Welling, M., Hinton, GE y Osindero, S. (2003a). Aprendizaje topográfico disperso representaciones con productos de distribuciones Student-t. En*NIPS'2002*.[680](#)
- Welling, M., Zemel, R. y Hinton, GE (2003b). Refuerzo autosupervisado. En S. Becker, S. Thrun y K. Obermayer, editores, *Avances en sistemas de procesamiento de información neuronal 15 (NIPS'02)*, páginas 665–672. Prensa del MIT.[622](#)
- Welling, M., Rosen-Zvi, M. y Hinton, GE (2005). Armonios familiares exponenciales con una aplicación para la recuperación de información. En L. Saul, Y. Weiss y L. Bottou, editores, *Avances en sistemas de procesamiento de información neuronal 17 (NIPS'04)*, volumen 17, Cambridge, MA. Prensa del MIT.[676](#)
- Werbos, PJ (1981). Aplicaciones de los avances en análisis de sensibilidad no lineal. En *Actas de la 10ª Conferencia IFIP, 31.8 - 4.9, NYC*, páginas 762–770.[225](#)
- Weston, J., Bengio, S. y Usunier, N. (2010). Anotación de imágenes a gran escala: aprender a clasificar con incrustaciones conjuntas de palabras e imágenes. *Aprendizaje automático*,[81](#)(1), 21–35.[401](#)
- Weston, J., Chopra, S. y Bordes, A. (2014). Redes de memoria. *preimpresión de arXiv arXiv:1410.3916*.[418,485](#)
- Widrow, B. y Hoff, ME (1960). Circuitos de conmutación adaptativos. En *1960 IRE WESCON Registro de la convención*, volumen 4, páginas 96–104. IRE, Nueva York.[15,21,24,27](#)

- Wikipedia (2015). Lista de animales por número de neuronas —Wikipedia, la enciclopedia libre.  
[En línea; consultado el 4 de marzo de 2015].[24,27](#)
- Williams, CKI y Agakov, FV (2002). Productos de Gaussianas y Probabilísticas Análisis de Componentes Menores. *Computación neuronal*, **14**(5), 1169–1182.[682](#)
- Williams, CKI y Rasmussen, CE (1996). Procesos gaussianos para la regresión. En D. Touretzky, M. Mozer y M. Hasselmo, editores, *Avances en sistemas de procesamiento de información neuronal 8 (NIPS'95)*, páginas 514–520. Prensa del MIT, Cambridge, MA.[142](#)
- Williams, RJ (1992). Conexiónista de algoritmos estadísticos simples de seguimiento de gradientes aprendizaje reforzado. *Aprendizaje automático*, **8**, 229–256.[688,689](#)
- Williams, RJ y Zipser, D. (1989). Un algoritmo de aprendizaje para ejecutar continuamente completamente redes neuronales recurrentes. *Computación neuronal*, **1**, 270–280.[223](#)
- Wilson, DR y Martínez, TR (2003). La ineficiencia general del entrenamiento por lotes para Aprendizaje de gradiente descendente. *Redes neuronales*, **dieciséis**(10), 1429–1451.[279](#)
- Wilson, JR (1984). Técnicas de reducción de varianza para simulación digital. *Americano Revista de Ciencias Matemáticas y de Gestión*, **4**(3), 277–312.[690](#)
- Wiskott, L. y Sejnowski, TJ (2002). Análisis de características lentas: aprendizaje no supervisado de invariantes. *Computación neuronal*, **14**(4), 715–770.[494](#)
- Wolpert, D. y MacReady, W. (1997). No hay teoremas de almuerzo gratis para la optimización. *IEEE Transacciones en computación evolutiva*, **1**, 67–82.[293](#)
- Wolpert, DH (1996). La falta de distinción a priori entre algoritmos de aprendizaje. *Neural Cálculo*, **8**(7), 1341–1390.[116](#)
- Wu, R., Yan, S., Shan, Y., Dang, Q. y Sun, G. (2015). Imagen profunda: Ampliación de la imagen reconocimiento. *arXiv:1501.02876*.[447](#)
- Wu, Z. (1997). Continuación global para problemas de geometría de distancias. *Revista SIAM de Mejoramiento*, **7**, 814–836.[327](#)
- Xiong, HY, Barash, Y. y Frey, BJ (2011). Predicción bayesiana de tejidos regulados empalme utilizando la secuencia de ARN y el contexto celular. *Bioinformática*, **27**(18), 2554–2562. [265](#)
- Xu, K., Ba, JL, Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, RS y Bengio, Y. (2015). Mostrar, atender y contar: generación de leyendas de imágenes neuronales con atención visual. En *ICML'2015*, *arXiv:1502.03044*.[102,410,691](#)
- Yildiz, IB, Jaeger, H. y Kiebel, SJ (2012). Revisitando la propiedad de estado de eco. *Redes neuronales*, **35**, 1–9.[405](#)

- Yosinski, J., Clune, J., Bengio, Y. y Lipson, H. (2014). ¿Qué tan transferibles son las características en redes neuronales profundas? En *NIPS'2014*.[325,536](#)
- Younes, L. (1998). Sobre la convergencia de los algoritmos estocásticos markovianos con disminución de las tasas de ergodicidad. En *Modelos estocásticos y estocásticos*, páginas 177-228.[612](#)
- Yu, D., Wang, S. y Deng, L. (2010). Etiquetado secuencial usando estructuras profundas campos aleatorios condicionales. *Revista IEEE de temas seleccionados en procesamiento de señales*.[323](#)
- Zaremba, W. y Sutskever, I. (2014). Aprendiendo a ejecutar. arXiv 1410.4615.[329](#)
- Zaremba, W. y Sutskever, I. (2015). Máquinas de Turing neuronales de aprendizaje por refuerzo. *arXiv:1505.00521*.[419](#)
- Zaslavsky, T. (1975). *Enfrentando arreglos: fórmulas de conteo de caras para particiones del espacio por hiperplanos*. número 154 en Memorias de la Sociedad Matemática Americana. Sociedad Matemática Americana.[550](#)
- Zeiler, MD y Fergus, R. (2014). Visualización y comprensión de redes convolucionales. En *ECCV'14*.[6](#)
- Zeiler, MD, Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J. y Hinton, GE (2013). Sobre unidades lineales rectificadas para procesamiento de voz. En *ICASSP 2013*.[460](#)
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. y Torralba, A. (2015). detectores de objetos emergen en la escena profunda las CNNs. *ICLR'2015*, arXiv:1412.6856.[551](#)
- Zhou, J. y Troyanskaya, OG (2014). Generativo convolucional y supervisado profundo Red estocástica para la predicción de estructura secundaria de proteínas. En *ICML'2014*.[715](#)
- Zhou, Y. y Chellappa, R. (1988). Cálculo del flujo óptico utilizando una red neuronal. En *Neural Networks, 1988., Conferencia internacional IEEE sobre*, páginas 71-78. IEEE.[339](#)
- Zöhrer, M. y Pernkopf, F. (2014). Redes estocásticas generales para clasificación. En *NIPS'2014*.[716](#)

# Índice

- 0-1 derrota, 102, 274
- Rectificación de valor absoluto, 191
- Exactitud, 420
- función de activación, 169
- restricción activa, 94
- adagrad, 305
- adalina, ver elemento lineal adaptativo adán, 307, 422
- elemento lineal adaptativo, 15, 23, 26
- Ejemplo adversario, 265
- entrenamiento adversario, 266, 268, 526 afín, 109
- AIS, ver muestreo de importancia recocido
- Casi en todas partes, 70
- Convergencia casi segura, 128
- Muestreo ancestral, 576, 591
- ANA, ver Red neuronal artificial
- Muestreo de importancia recocido, 621, 662, 711
- Cálculo bayesiano aproximado, 710
- inferencia aproximada, 579
- Inteligencia artificial, 1
- Red neuronal artificial, ver Red neuronal: Burn-in, 593
- trabajar
- RAS, ver reconocimiento automático de voz
- Asintóticamente imparcial, 123
- Audio, 101, 357, 455
- codificador automático, 4, 353, 498
- Reconocimiento automático de voz, 455
- retropropagación, 201
- Retropropagación a través del tiempo, 381
- backprop, ver retropropagación
- Saco de palabras, 467
- Harpillera, 252
- normalización de lotes, 264, 422
- error de Bayes, 116
- regla de Bayes, 69
- Optimización de hiperparámetros bayesianos, 433
- red bayesiana, ver gráfico dirigido
- modelo
- probabilidad bayesiana, 54
- estadísticas bayesianas, 134
- red de creencias, ver modelo gráfico dirigido
- distribución de Bernoulli, 61
- BFGS, 314
- Inclinación, 123, 227
- parámetro de sesgo, 109
- muestreo de importancia sesgada, 589
- bigrama, 458
- relación binaria, 478
- Muestreo de bloque Gibbs, 595
- distribución de Boltzmann, 566
- máquina de Boltzmann, 566, 648
- BPTT, ver retropropagación a través del tiempo
- Radiodifusión, 33
- CAE, ver autocodificador contractivo
- Cálculo de variaciones, 178
- distribución categórica, ver enfermedad multinomial tributo
- CD, ver truco de centrado de
- divergencia contrastiva (DBM), 667
- Teorema del límite central, 63
- Regla de la cadena (cálculo), 203
- regla de la cadena de probabilidad, 58

- Ajedrez,[2](#)  
Acorde,[575](#)  
gráfico de cuerdas,[575](#)  
Modelos de lenguaje basados en clases,[460](#) sistema dinámico clásico, [372](#) Clasificación,[99](#)  
potencial camarilla, *ver* factor (modelo gráfico)  
CNN, *ver* filtrado colaborativo de redes neuronales convolucionales,[474](#) colisionador, *ver* explicando imágenes en color,[357](#)
- célula compleja,[362](#)  
gráfico computacional,[202](#)  
Visión por computador,[449](#)  
Deriva del concepto,[533](#)  
número de condición,[277](#)  
Cálculo condicional, *ver* restructura dinámica  
independencia condicional,[XIII,59](#) La probabilidad condicional,[58](#) RBM condicional,[679](#)  
conexionismo,[17,440](#) clasificación temporal connexionista,[457](#) Consistencia,[128,509](#)  
Optimización con restricciones,[92,235](#)  
Direccionamiento basado en contenido,[416](#)  
Sistemas de recomendación basados en contenido,[475](#) Independencia específica del contexto,[569](#) bandidos contextuales,[476](#) métodos de continuación,[324](#) autocodificador contractivo, [516](#) Contraste,[451](#)
- divergencia contrastiva,[289,606,666](#)  
Optimización convexa,[140](#) Circunvolución, [327,677](#) red convolucional, *dieciséis* red neuronal convolucional,[250,327,422,456](#)  
Descenso coordinado,[319,665](#)  
Correlación,[60](#)  
función de costo, *ver* función objetivo covarianza,[XIII,60](#) Matriz de covarianza,[61](#)  
Cobertura,[421](#)
- Temperatura crítica,[599](#)  
correlación cruzada,[329](#)  
entropía cruzada,[74,131](#)  
Validación cruzada,[121](#)  
CTC, *ver* clasificación temporal conexiónista aprendizaje del currículo,[326](#)  
Maldición de dimensionalidad,[153](#)  
ciclo,[2](#)
- D-separación,[568](#)  
DAE, *ver* denoising autoencoder Distribución de generación de datos,[110,130](#) Proceso de generación de datos,[110](#) paralelismo de datos, [444](#)  
conjunto de datos,[103](#)  
Aumento de conjuntos de datos,[268,454](#)  
DBM, *ver* maquina profunda de boltzmann dcgan,[547,548,695](#) Árbol de decisión,[144,544](#) Descifrador,[4](#)
- red de creencias profundas,[26,525,626,651,654,678,686](#)  
Azul profundo,[2](#)  
máquina profunda de Boltzmann,[23,26,525,626,647,651,657,666,678](#)  
red de retroalimentación profunda,[166,422](#)  
Aprendizaje profundo,[2,5](#) Codificador automático de eliminación de ruido,[506,683](#) Coincidencia de puntuación de eliminación de ruido,[615](#) Estimación de densidad,[102](#) Derivado,[XIII,82](#) matriz de diseño, [105](#)
- capa detectora,[336](#)  
Determinante,[xi](#)  
Matriz diagonal,[40](#)  
entropía diferencial,[73,641](#)  
función delta de Dirac,[64](#)  
modelo gráfico dirigido,[76,503,559,685](#)  
Derivado direccional,[84](#) Ajuste fino discriminativo, *versus* supervisado sintonía FINA  
RBM discriminatorio,[680](#) representación distribuida,[17,149,542](#) adaptación de dominio,[532](#)

- producto punto,[33](#),[139](#)  
backprop doble,[268](#)  
Matriz circulante de doble bloque,[330](#)  
sueño sueño,[605](#),[647](#) DropConnect,[263](#)  
Abandonar,[255](#),[422](#),[427](#),[428](#),[666](#),[683](#)  
estructura dinámica,[445](#)
- paso E,[629](#)  
Parada temprana,[244](#),[246](#),[270](#),[271](#),[422](#)  
EBM, ver modelo basado en energía Echo state network,[23](#),[26](#),[401](#) capacidad efectiva,[113](#) Descomposición propia,[41](#)
- valor propio,[41](#)  
vector propio,[41](#)  
ELBO, ver límite inferior de evidencia Producto de elemento sabio, ver producto Hadamard producto, ver producto Hadamard  
EM, ver maximización de expectativas  
incrustación,[512](#)  
distribución empírica,[sesenta y cinco](#) riesgo empírico,[274](#)  
Minimización empírica del riesgo,[274](#)  
codificador,[4](#)  
función de energía,[565](#)  
Modelo basado en energía,[565](#),[591](#),[648](#),[657](#)  
métodos de conjunto,[252](#)  
Época,[244](#)  
Restricción de igualdad,[93](#)  
Equivarianza,[335](#)  
función de error, ver función objetivo  
ESN, ver echo state network norma euclíadiana,[38](#)  
ecuación de Euler-Lagrange,[641](#)  
Evidencia límite inferior,[628](#),[655](#)  
Ejemplo,[98](#)  
Expectativa,[59](#)  
maximización de expectativas,[629](#) Valor esperado, ver expectativa explicando lejos,[570](#),[626](#),[639](#) Explotación,[477](#)  
Exploración,[477](#)  
Distribución exponencial,[64](#)
- puntuación F,[420](#)  
Factor (modelo gráfico),[563](#)  
Análisis factorial,[486](#)  
gráfica de factores,[575](#)  
Factores de variación,[4](#)  
Característica,[98](#)  
Selección de características,[234](#)  
red neuronal feedforward,[166](#)  
Sintonía FINA,[321](#)  
diferencias finitas,[436](#)  
Olvídate de la puerta,[304](#)  
Propagación hacia adelante,[201](#)  
Transformada de Fourier,[357](#),[359](#)  
fóvea,[363](#)  
FPCD,[610](#)  
Energía gratis,[567](#),[674](#)  
base libre,[479](#)  
probabilidad frecuentista,[54](#)  
estadísticas frecuentistas,[134](#)  
norma de frobenius,[45](#)  
Red bayesiana totalmente visible,[699](#)  
derivados funcionales,[640](#) FVBN, ver red bayesiana totalmente visible  
función de Gabor,[365](#)  
GAN, ver redes antagónicas generativas Unidad recurrente cerrada,[422](#)  
Distribución gaussiana, ver distribución normal  
núcleo gaussiano,[140](#)  
mezcla gaussiana,[66](#),[187](#)  
GCN, ver normalización de contraste global  
Ontología de genes,[479](#)  
Generalización,[109](#)  
Función de Lagrange generalizada, ver generalizado Lagrangiano izado  
Lagrangiano generalizado,[93](#) Redes antagónicas generativas,[683](#),[693](#) Redes generativas de coincidencia de momentos,[696](#) red de generadores,[687](#)  
distribución de Gibbs,[564](#) muestreo de gibbs,[577](#),[595](#) Normalización de contraste global,[451](#) GPU, ver unidad de procesamiento de gráficos gradiente,[83](#)

- recorte de degradado,[287](#),[411](#)  
Descenso de gradiente,[82](#),[84](#)  
Grafico,[xi](#)  
modelo gráfico,*ver*probabilidad estructurada  
    modelo de tic  
Unidad de procesamiento gráfico,[441](#)  
Algoritmo codicioso,[321](#)  
Preentrenamiento no supervisado codicioso por capas,  
    [524](#)  
Preentrenamiento supervisado codicioso,[321](#)  
búsqueda de cuadrícula,[429](#)
- producto Hadamard,[xi](#),[33](#)  
DuroTanh,[195](#)  
Armonio,*ver*Boltzmann ma-  
    sierra  
teoría de la armonía,[567](#)  
energía libre de Helmholtz,*ver*evidencia más baja  
    atado  
Arpillera,[221](#)  
Matriz Hessiana,[XIII](#),[86](#)  
heteroscedástico,[186](#)  
capa oculta,[6](#),[166](#)  
Montañismo,[85](#)  
optimización de hiperparámetros,[429](#)  
hiperparámetros,[119](#),[427](#) espacio de  
hipótesis,[111](#),[117](#)
- supuestos iid,[110](#),[121](#),[265](#)  
Matriz de identidad,[35](#)  
ILSVRC,*ver*Visual a gran escala de ImageNet  
    Desafío de reconocimiento  
Reconocimiento visual a gran escala de ImageNet  
    Desafío,[22](#)
- Inmoralidad,[573](#)  
muestreo de importancia,[588](#),[620](#),[691](#) Codificador  
automático ponderado por importancia,[691](#)  
Independencia,[XIII](#),[59](#)  
Independientes e idénticamente distribuidos,*ver*  
    supuestos iid  
Análisis de componentes independientes,[487](#)  
Análisis subespacial independiente,[489](#)  
Restricción de desigualdad,[93](#)  
Inferencia,[558](#),[579](#),[626](#),[628](#),[630](#),[633](#),[643](#),  
    [646](#)
- Recuperación de información,  
    [520](#) inicialización,[298](#)  
Integral,[XIII](#)  
invariancia,[339](#)  
isotrópico,[64](#)
- matriz jacobiana,[XIII](#),[71](#),[85](#)  
Probabilidad conjunta,[56](#)
- k*-medio,[361](#),[542](#)  
*k*-vecinos más cercanos,[141](#),[544](#) condiciones  
de Karush-Kuhn-Tucker,[94](#),[235](#) Karush-  
Kuhn-Tucker,[93](#)  
Núcleo (convolución),[328](#),[329](#)  
máquina de núcleo,[544](#)  
truco del núcleo,[139](#)  
KKT,*ver*Condiciones de Karush-Kuhn-  
Tucker KKT,*ver*Karush-Kuhn-Tucker  
    condiciones
- divergencia KL,*ver*Buceador Kullback-Leibler-  
    gencia  
Base de conocimientos,[2](#),  
    [479](#) métodos de krylov,[222](#)  
divergencia Kullback-Leibler,[XIII](#),[73](#)
- Suavizado de etiquetas,[241](#)  
multiplicadores de Lagrange,[93](#),[641](#)  
lagrangiano,*ver*Lagrangiano generalizado  
LAPGAN,[695](#)  
distribución de Laplace,[64](#),[492](#)  
Variable latente,[66](#)  
Capa (red neuronal),[166](#) lcn,*ver*  
normalización de contraste local  
Leaky ReLU,[191](#)  
Unidades con fugas,[404](#)  
Tasa de aprendizaje,[84](#)  
búsqueda de línea,[84](#),[85](#),  
    [92](#) Combinación lineal,[36](#)  
dependencia lineal,[37](#)  
Modelos de factores lineales,[485](#)  
regresión lineal,[106](#),[109](#),[138](#) predicción  
de enlaces,[480](#)  
constante de Lipschitz,[91](#)  
Lipschitz continuo,[91](#) máquina  
de estado líquido,[401](#)

- Distribución de probabilidad condicional local, 560  
Normalización del contraste local, 452 Regresión logística, 3, 138, 139  
sigmoide logístico, 7, 66  
Memoria a corto plazo, 18, 24, 304, 407, 422  
Bucle, 575  
Propagación de creencias locas, 581 Función de pérdida, ver función objetiva  $L_{\text{pagnorma}}$ , 38
- LSTM, ver memoria a corto plazo  
paso m, 629  
Aprendizaje automático, 2  
Máquina traductora, 100  
Diagonal principal, 32  
Colector, 159  
Hipótesis múltiple, 160  
aprendizaje múltiple, 160  
clasificador de tangente múltiple, 268 aproximación MAP, 137, 501  
Probabilidad marginal, 57 cadena de Markov, 591  
Cadena de Markov Montecarlo, 591 red de Markov, ver modelo no dirigido de campo aleatorio de Markov, ver modelo no dirigido matriz,  $x_i, x_j, \dots, x_1$  matriz inversa, 35
- producto matriz, 33  
norma máxima, 39  
agrupación máxima, 336  
Máxima verosimilitud, 130  
Máximo fuera, 191, 422  
MCMC, ver Cadena de Markov Monte Carlo  
Campo medio, 633, 634, 666 Error medio cuadrado, 107 Teoría de la medida, 70
- medir cero, 70  
red de memoria, 413, 415  
Método de descenso más empinado, ver degradado descendencia  
minilote, 277  
entradas faltantes, 99  
Mezcla (cadena de Markov), 597
- Redes de densidad de mezcla, 187  
Distribución de la mezcla, sesenta y cinco modelo de mezcla, 187, 506  
Mezcla de expertos, 446, 544 MLP, ver percepción multicapa MNIST, 20, 21, 666 promedio del modelo, 252
- compresión del modelo, 444  
Identificabilidad del modelo, 282  
modelo de paralelismo, 444  
Coincidencia de momentos, 696  
Pseudoinversa de Moore-Penrose, 44, 237  
gráfico moralizado, 573  
MP-DBM, ver multipredicción DBM MRF (campo aleatorio de Markov), ver undimodelo rectificado  
MSE, ver error cuadrático medio  
Aprendizaje multimodal, 535  
DBM de predicción múltiple, 668  
Aprendizaje multitarea, 242, 533  
percepción multicapa, 5  
perceptrón multicapa, 26  
Distribución multinomial, 61  
distribución multinomial, 61
- norte-gramo, 458  
NADE, 702  
ingenuo bayes, 3  
Nat, 72  
imagen natural, 555  
Procesamiento natural del lenguaje, 457  
Regresión del vecino más cercano, 114  
Negativo definido, 88  
fase negativa, 466, 602, 604  
neocognitrón, dieciséis, 23, 26, 364 impulso de Nésterov, 298  
gran premio netflix, 255, 475 modelo de lenguaje neuronal, 460, 472 red neuronal, 13  
maquina neural de turing, 415  
neurociencia, 15  
método de newton, 88, 309 BNL, ver modelo de lenguaje neural PNL, ver Procesamiento del lenguaje natural No hay teorema de almuerzo gratis, 115

- Estimación de contraste de ruido,  
**616** modelo no paramétrico, **113**  
Norma, **xiv,38**  
Distribución normal, **62,63,124** ecuaciones normales, **108,108,111,232** Inicialización normalizada, **301** Diferenciación numérica, *verfi* noche diferente-  
encias  
detección de objetos, **449**  
Reconocimiento de objetos,  
**449** Función objetiva, **81**  
OMP-*k*, *ver* Persecución de emparejamiento ortogonal Aprendizaje de un disparo, **534**  
Operación, **202**  
Mejoramiento, **79,81**  
estadísticas ortodoxas, *ver* estadísticas frequentistas Persecución de emparejamiento ortogonal, **26,252**  
matriz ortogonal, **41**  
ortogonalidad, **40**  
capa de salida, **166**  
  
Procesamiento distribuido en paralelo, **17** Inicialización de parámetros, **298,403** intercambio de parámetros, **249,332,370,372,386** vinculación de parámetros, *ver* Intercambio de parámetros Modelo paramétrico, **113**  
  
ReLU paramétrico, **191**  
Derivada parcial, **83**  
Función de partición, **564,601,663** PCA, *ver* análisis de componentes principales PCD, *ver* máxima verosimilitud estocástica perceptrón, **15,26**  
divergencia contrastiva persistente, *ver* stochas- máxima probabilidad de tic  
Análisis de perturbaciones, *ver* reparametrización truco  
estimador puntual, **121**  
Política, **476**  
puesta en común, **327,677**  
Positivo definitivo, **88**  
fase positiva, **466,602,604,650,662**  
Precisión, **420**  
Precisión (de una distribución normal), **62,64**  
Descomposición escasa predictiva, **519**  
preprocesamiento, **450**  
Pre-entrenamiento, **320,524**  
corteza visual primaria, **362**  
Análisis de componentes principales, **47,145,146,486,626**  
Distribución de probabilidad previa, **134**  
Agrupación máxima probabilística, **677** PCA probabilístico, **486,487,627** función de densidad de probabilidad, **57** Distribución de probabilidad, **55**  
Función de probabilidad, **55** Estimación de la función de masa de probabilidad, **102** Producto de expertos, **566**  
  
regla de probabilidad del producto, *ver* cadena de reglas de probabilidad  
PSD, *ver* descomposición escasa predictiva pseudoverosimilitud, **611**  
  
par en cuadratura, **366**  
Métodos quasi-Newton, **314**  
  
Función de base radial, **195**  
Búsqueda aleatoria, **431**  
Variable aleatoria, **55**  
coincidencia de proporciones, **614**  
RBF, **195**  
RBM, *ver* retirada restringida de la máquina Boltzmann, **420**  
Campo receptivo, **334**  
sistemas de recomendación, **474** Unidad lineal rectificada, **170,191,422,503** red recurrente, **26**  
red neuronal recurrente, **375**  
regresión, **99**  
regularización, **119,119,176,226,427**  
regularizador, **118**  
REFORZARSE, **683**  
Aprendizaje reforzado, **24,105,476,683** Base de datos relacional, **479** Relaciones, **478**  
  
truco de reparametrización, **682** aprendizaje de la representación, **3** capacidad representativa, **113**  
Máquina de Boltzmann restringida, **353,456,**  
**475,583,626,650,651,666,670,**

- 672,674,677
- Regresión de cresta,*ver*riesgo de caída de peso,273
- RNN-RBM,679
- puntos de silla,283
- Muestra promedio,124
- Escalar,*xi*,*xi*,30 Coincidencia de puntuación,509,613 Segunda derivada,85
- Prueba de la segunda derivada,88 Autoinformación,72
- hash semántico,521
- Aprendizaje semisupervisado,241 convolución separable,359 Separación (modelado probabilístico),568 Colocar,*xi*
- USD,*ver*descenso de gradiente estocástico entropía de Shannon,XIII,73 preselección,462
- Sigmoideo,*xiv*,*ver*lógica sigmoide sigmoide red de creencias,26 celda sencilla,362
- valor singular,*ver*valor singular descomposición posición
- Valor singular de descomposición,43,146,475
- vector singular,*ver*valor singular descomposición
- Análisis de características lentes,489
- SML,*ver*máxima verosimilitud estocástica
- Softmax,182,415,446 Softplus,*xiv*,67,195 detección de correo no deseado,3
- Codificación escasa,319,353,492,626,686
- Inicialización escasa,302,403
- Representación escasa,145,224,251,501,552
- Menta verde,433
- radio espectral,401
- Reconocimiento de voz,*ver*voz automática reconocimiento
- esferas,*ver*blanqueo
- Spike y losa restringida Boltzmann ma- Error de entrenamiento,109 sierra,674
- SPN,*ver*red de suma-producto
- Matriz cuadrada,37
- ssRBM,*ver*pernos y losas restringidas Boltz- máquina humana
- Desviación Estándar,60
- Error estándar,126
- Error standar de la media,126,276
- Estadística,121
- teoría del aprendizaje estadístico,109 Descenso más empinado,*ver*descenso de gradiente Propagación estocástica hacia atrás,*ver*reparametriza- truco de ción
- Descenso de gradiente estocástico,15,149,277,292,666
- Máxima verosimilitud estocástica,608,666 agrupación estocástica,263 estructurar el aprendizaje,578 salida estructurada,100,679
- Modelo probabilístico estructurado,76,554
- Suma regla de probabilidad,57 red suma- producto,549 Puesta a punto supervisada,525,656 Aprendizaje supervisado,104 Máquinas de vectores soporte,139 Función de pérdida sustituta,274 SVD,*ver*descomposición en valor singular Matriz simétrica,40,42
- distancia tangente,267
- Plano de la tangente,511
- puntal tangente,267
- TDNN,*ver*red neuronal de retardo de tiempo
- Forzamiento del maestro,379,380 templado,599
- Comparación de plantillas,140
- Tensor,*xi*,*xi*,32 Equipo de prueba,109
- regularización de Tikhonov,*ver*decaimiento de peso
- Convolución en mosaico,349
- red neuronal de retardo de tiempo,364,371
- matriz de Toeplitz,330
- ICA topográfica,489
- operador de rastreo,45
- Transcripción,100
- Transferir el aprendizaje,532

- Transponer,[xi,32](#) Desigualdad triangular,  
[38](#) gráfico triangulado, *ver*gráfico cordal  
trígrama,[458](#)
- Aprendizaje de datos cero, *ver*aprendizaje de disparo cero  
aprendizaje de disparo cero,[534](#)
- Imparcial,[123](#)  
modelo gráfico no dirigido,[76,503](#)  
modelo no dirigido,[562](#)  
Distribución uniforme,[56](#)  
unígrama,[458](#)  
norma unitaria,[40](#)  
Vector unitario,[40](#)  
teorema de aproximación universal,[196](#)  
aproximador universal,[549](#) Distribución de  
probabilidad no normalizada,[563](#) Aprendizaje sin  
supervisión,[104,144](#) Preentrenamiento no  
supervisado,[456,524](#)
- estructura en V, *ver*explicando  
V1,[362](#)  
vae, *ver*Codificador automático variacional Dimensión  
Vapnik-Chervonenkis,[113](#) Diferencia,[XIII,60,227](#)  
codificador automático variacional,[683,690](#) derivadas  
variacionales, *ver*derivación funcional
- tivos  
energía libre variacional, *ver*evidencia más baja  
atado  
dimensión de CV, *ver*Vapnik-Chervonenkis di-  
mención
- Vector,[xi,xi,31](#)  
Ejemplos de adversarios virtuales,  
[266](#) capa visible,[6](#)  
datos volumétricos,[357](#)
- Despertar-dormir,[646,655](#)  
Decaimiento de peso,[117,176,229,](#)  
[428](#) simetría del espacio de peso,  
[282](#) pesos,[15,106](#) Blanqueo,[452](#)
- wikibase,[479](#)  
wikibase,[479](#)  
incrustación de palabras,[460](#)  
Desambiguación del sentido de las palabras,[480](#)  
red de palabras,[479](#)