



Introducción al Deep Learning

Dr. Ing. Gabriel Hermosilla Vigneau

Deep Learning

- El Deep Learning o aprendizaje profundo es un subcampo dentro del Machine Learning, el cuál utiliza distintas estructuras de redes neuronales para lograr el **aprendizaje** de sucesivas *capas de representaciones* cada vez más significativas de los datos (imágenes, video, audio, etc).
- Lo profundo o deep en Deep Learning hace referencia a la cantidad de capas de representaciones que se utilizan en el modelo; en general se suelen utilizar decenas o incluso cientos de capas de representación, las cuales aprenden automáticamente a medida que el modelo es entrenado con los datos.

Deep Learning

ARTIFICIAL INTELLIGENCE

Any technique that enables computers to mimic human behavior



MACHINE LEARNING

Ability to learn without explicitly being programmed



DEEP LEARNING

Learn underlying features in data using neural networks

3 1 3 4 7 2

Deep Learning

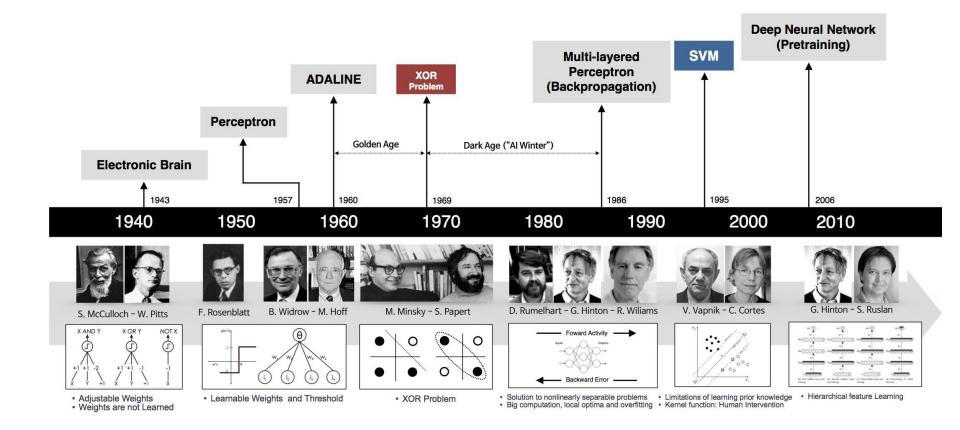
- Los algoritmos basados en Deep Learning (como las redes feed forward) tienen en teoría la capacidad de aproximar cualquier función.
- Esa propiedad teórica indica que existe una red óptima, pero no implica que el proceso de aprendizaje sea capaz de alcanzar la configuración óptima.
- En la práctica las redes no funcionaban bien y los algoritmos basados en Deep Learning fueron abandonados (< 2008).
- Durante bastante tiempo dominaban los algoritmos que extraen características (ingeniería de características feature extraction).
- El desempeño (performance) era limitado y dependía del modelo.
- Requieren conocimiento experto.
- En los últimos años las redes Deep han revivido producto de mejoras en los algoritmos y al uso de GPUs.

Historia del Deep Learning

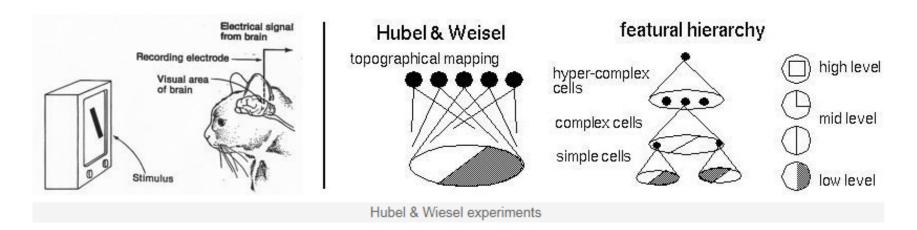
- McCulloch-Pitts, Neuron [1943]. Pesos configurados a mano
- Rosemblatt, perceptron [1957]. Clasificador, aprendía de ejemplos
- Widrow and Hoff, ADALINE (adaptivelinear element) [1960]. Regresor, usaba descenso estocástico de gradiente.
- Minsky and Papert, XOR problema [1969]. Los modelos lineales tienen limitaciones
- Fukushima, Neocognitron[1980]. Campos receptivos, precursor de CNN
- Backpropagation + Multilayer perceptron. Tenía ideas ya establecidas pero rescatadas por Rumelhart, Hintony Willians [1986] para resolver problemas complicados.
- LeCun, LeNet-5 [1998]. Utiliza Convolutional Neural Networks para reconocer patrones visuales (dígitos) a partir de píxeles.

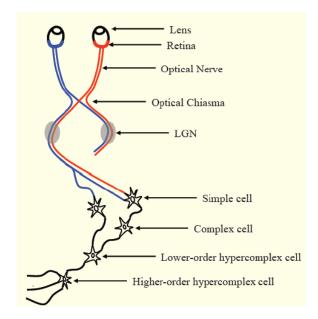
• ...

Historia del Deep Learning

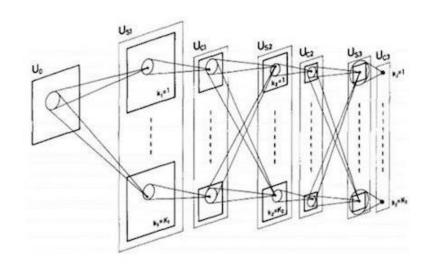


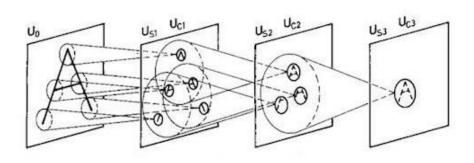
- Experimentos de Hubel y Wiesel (años 60)
- Durante la década de 1960, los neurofisiólogos David Hubel y Torsten Wiesel realizaron una serie de experimentos para investigar las propiedades de las cortezas visuales de los animales. En uno de los experimentos más notables, midieron las respuestas eléctricas del cerebro de un gato mientras lo estimulaban con patrones simples en una pantalla de televisión.
- Lo que encontraron fue que las neuronas en la corteza visual primaria se organizan de manera jerárquica, donde las primeras células conectadas a las retinas del gato son responsables de detectar patrones simples como bordes y barras, seguidos de capas posteriores que responden a patrones más complejos combinando las actividades neuronales primarias.

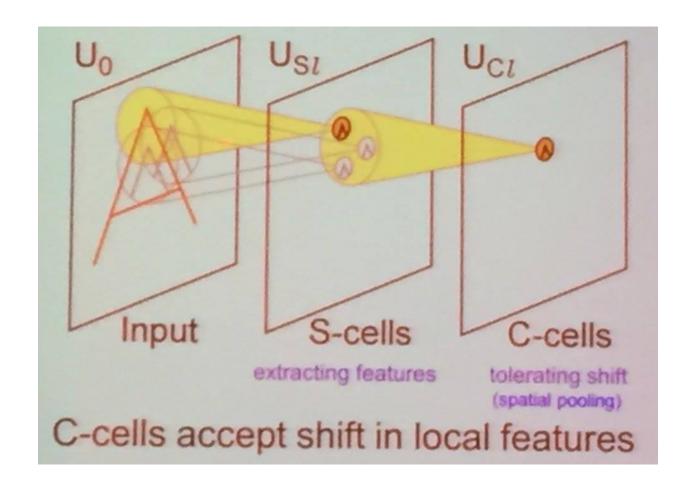




- Neocognitron (1982)
- El neocognitron es una red neuronal artificial jerárquica y multicapa propuesta por Kunihiko Fukushima en la década de 1980. Se ha utilizado para el reconocimiento de caracteres a mano y otras tareas de reconocimiento de patrones, y ha servido de inspiración para las redes neuronales convolucionales.
- El neocognitron se inspiró en el modelo propuesto por Hubel & Wiesel en 1959. Encontraron dos tipos de células en la corteza primaria visual, llamadas células simples y células complejas, y también propusieron un modelo en cascada de estos dos tipos de células para usar en el reconocimiento de patrones.
- El neocognitron fue la primera arquitectura de red neuronal en usar capas jerárquicas en las que cada capa es responsable de detectar un patrón a partir de la salida del anterior, utilizando un filtro deslizante para ubicarlo en cualquier lugar de la imagen.





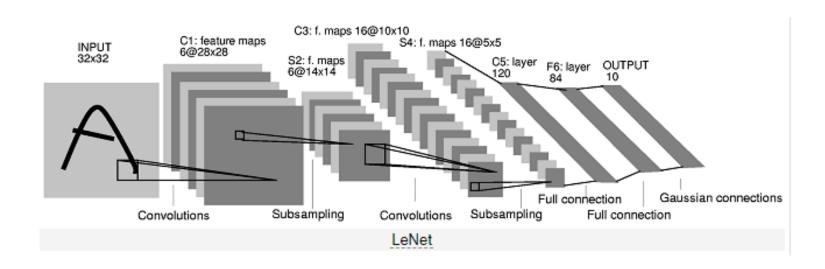


Nota: Foto tomada en la conferencia de Kunihiko Fukushima, ICONIP 2016, Kioto, Japón.

• LeNet (1998)

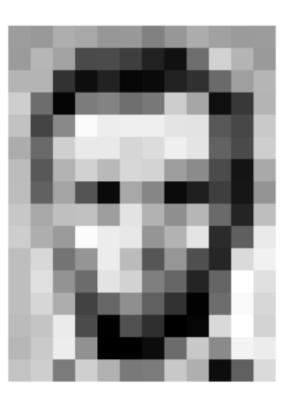
- A fines de la década de 1980, Geoffrey Hinton et. al. lograron aplicar backpropagation al entrenamiento de redes neuronales. Durante la década de 1990, un equipo de AT&T Labs dirigido por el ex estudiante postdoctoral de Hinton, Yann LeCun, formó una red convolucional, apodada "LeNet", para clasificar imágenes de dígitos escritos a mano con una precisión del 99.3%.
- Su sistema se utilizó durante un tiempo para leer automáticamente los números en el 10-20% de los cheques impresos en los Estados Unidos. LeNet tenía 7 capas, incluidas dos capas convolucionales, con la arquitectura resumida en la siguiente figura.

• LeNet (1998)



Objetivos en Deep Learning

• Las imágenes son números

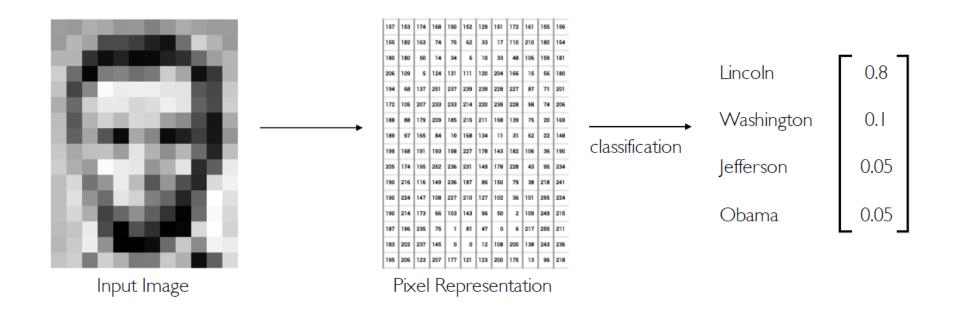


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	6	124	131	111	120	204	166	15	56	180
194	68	197	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
206	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	٥	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	n	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	96	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

Objetivos en Deep Learning

- Las tareas que debe realizar son:
- Regresión: la variable de salida toma un valor continuo.
- Clasificación: la variable de salida toma la etiqueta de una clase. Además puede producir la probabilidad perteneciente a una clase en particular.



Detección de características de alto nivel

• Identifiquemos las características clave en cada categoría de imagen



Nariz, ojos, boca



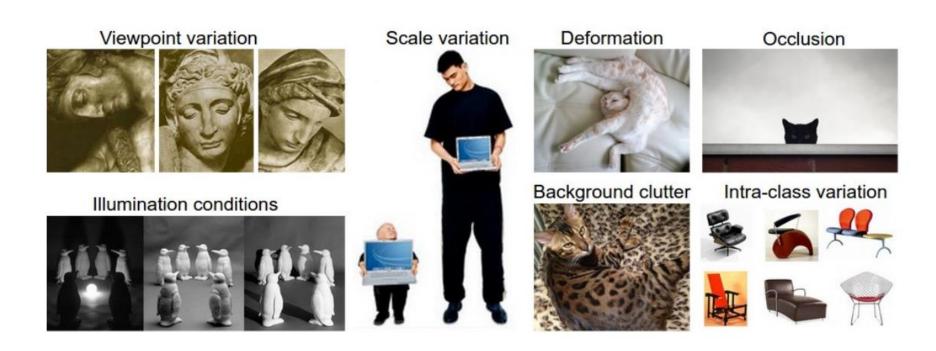
Ruedas, patente, focos



puerta, ventanas, escala

Detección de características de alto nivel

• Existen problemas al tratar de detectar esas características de manera manual.



¿Porqué el Deep Learning ahora?

- La obtención de características realizadas a mano requieren de mucho consumo de tiempo y no son escalables en la práctica.
- ¿Podemos aprender las características subyacentes directamente de los datos?

¿Porqué el Deep Learning ahora?

• Las redes neuronales provienen de décadas atrás, porque su reaparición ahora?

Stochastic Gradient
Descent

Perceptron
Learnable Weights

Backpropagation
Multi-Layer Perceptron

Deep Convolutional NN
Digit Recognition

•

I. Big Data

- Larger Datasets
- EasierCollection &Storage







2. Hardware

- Graphics Processing Units (GPUs)
- Massively Parallelizable



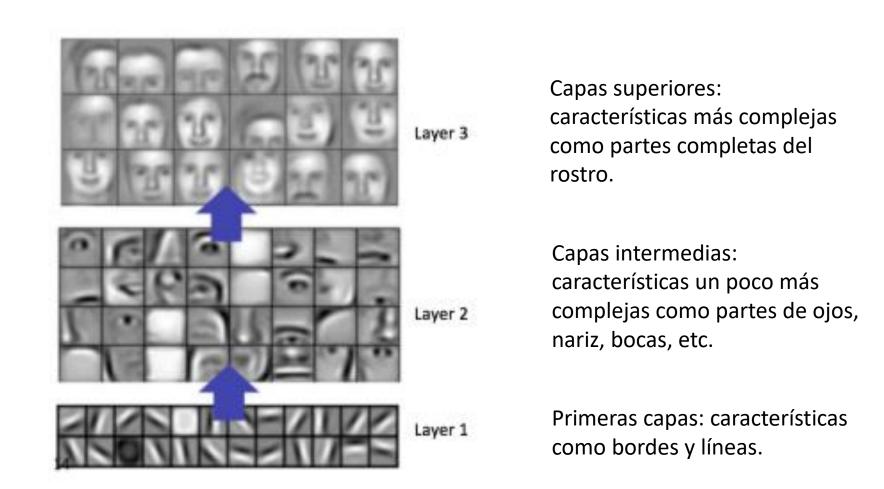
3. Software

- Improved Techniques
- New Models
- Toolboxes



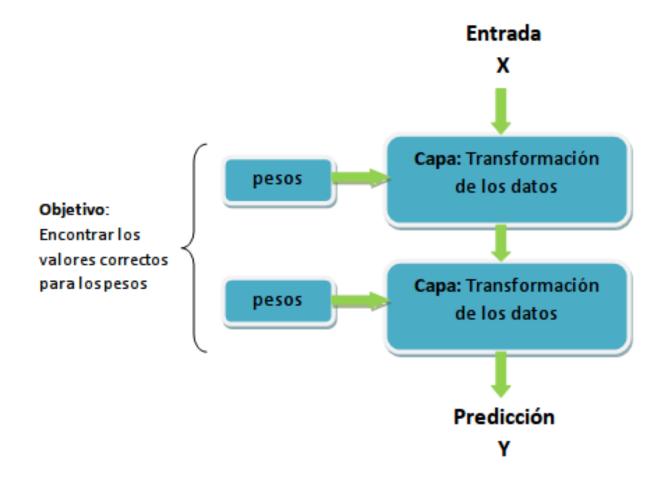
- En general, cualquier técnica de Machine Learning trata de realizar la asignación de entradas (por ejemplo imágenes) a salidas objetivos (por ejemplo, la etiqueta "gato"), mediante la observación de un gran número de ejemplos de entradas y salidas (bases de datos).
- El Deep Learning realiza este mapeo de entrada-a-objetivo por medio de una red neuronal artificial que está compuesta de un número grande de *capas* dispuestas en forma de jerarquía.
- La red aprende algo simple en la capa inicial de la jerarquía y luego envía esta información a la siguiente capa. La siguiente capa toma esta información simple, lo combina en algo que es un poco más complejo, y lo pasa a la tercer capa. Este proceso continúa de forma tal que cada capa de la jerarquía construye algo más complejo de la entrada que recibió de la capa anterior. De esta forma, la red irá aprendiendo por medio de la exposición a los datos de ejemplo.

Ejemplo de pesos aprendidos para una red para Face Recognition

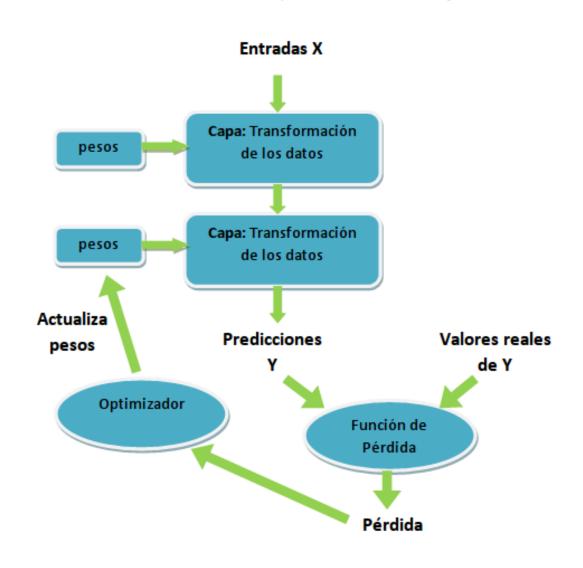


- Para que la red aprenda debemos encontrar los *pesos* de todas las *capas* de forma tal que la red realice un mapeo perfecto entre los ejemplos de entrada con sus respectivas salidas objetivo.
- El problema reside en que una red de Deep Learning puede tener millones de *parámetros*, por lo que encontrar el valor correcto de todos ellos puede ser una tarea realmente muy difícil, especialmente si la modificación del valor de uno de ellos afecta a todos los demás.
- Pero ¿¿¿cómo entrenábamos la red????
- Recordar: Funciones de pérdida, gradiente descendiente y backpropagation.

Recordemos un poco el proceso de entrenamiento de las redes neuronales.



- Para poder controlar algo, en primer lugar debemos poder observarlo. En este sentido, para controlar la salida de la red neuronal, deberíamos poder medir cuan lejos está la salida que obtuvimos de la que se esperaba obtener (ejemplos de entrenamiento).
- Este es el trabajo de la *función de pérdida* (loss function) de la red. Esta función toma las predicciones que realiza el modelo y los valores objetivos (lo que realmente esperamos que la red produzca), y calcula cuán lejos estamos de ese valor, de esta manera, podemos capturar que tan bien está funcionando el modelo para el ejemplo especificado.
- El truco fundamental del Deep Learning es utilizar el valor que nos devuelve esta función de pérdida para retroalimentar la red y ajustar los pesos en la dirección que vayan reduciendo la pérdida (error) del modelo para cada ejemplo. Este ajuste, es el trabajo del optimizador, el cuál implementa la propagación hacia atrás.

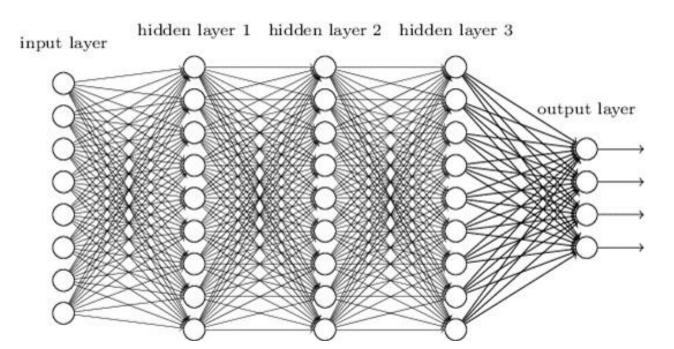


- Resumiendo, el funcionamiento sería el siguiente: inicialmente, los *pesos* de cada *capa* son asignados en forma aleatoria, por lo que la red simplemente implementa una serie de transformaciones aleatorias. En este primer paso, obviamente la salida del modelo dista bastante del ideal que deseamos obtener, por lo que el valor de la *función de pérdida* va a ser bastante alto.
- A medida que la red va procesando nuevos casos, los *pesos* se van ajustando de forma tal de ir reduciendo cada vez más el valor de la *función de pérdida*. Este proceso es el que se conoce como *entrenamiento* de la red, el cual repetido una suficiente cantidad de veces, generalmente 10 iteraciones de miles de ejemplos, logra que los *pesos* se ajusten a los que minimizan la *función de pérdida*. Una red que ha minimizado la *pérdida* es la que logra los resultados que mejor se ajustan a las salidas objetivo, es decir, que el modelo se encuentra *entrenado*.

Redes Deep Feedforward

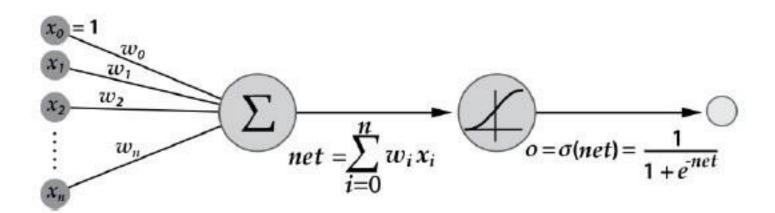
- Son redes que son capaces de aprender relaciones entrada-salida a partir de una gran cantidad de ejemplos.
- Están formadas por muchas neuronas simples interconectadas entre si.
- Deep => muchas capas

Deep neural network



Neuronas

- La neurona consiste de un modelo lineal seguido por una no-linealidad
- Cada neurona tiene varias entradas y una salida
- Cada entrada es multiplicada por un peso sináptico
- Neurona determinada por pesos y no-linealidad



Manejo de los datos

- El sistema se debe entrenar y probar con un conjunto de datos (database).
- Se debe dividir el conjunto de datos en tres:

Conjunto de entrenamiento (60-70%)

Conjunto de validación (40-30%)

Conjunto de prueba o test (nuevos datos)

Conjunto de entrenamiento (train):

Permite obtener los pesos de una red dada

Ajuste de parámetros

• Conjunto de validación (validation):

Permite comparar el desempeño de distintas arquitecturas de red

Ajuste de hiperparámetros

Conjunto de prueba (test):

Permite estimar el desempeño de la red en datos futuros. El mejor modelo se debe probar acá.

Ajuste de la red (Fit)

• Inicialización de los pesos.

- Antes de comenzar el entrenamiento, es necesario elegir los pesos iniciales.
- Los pesos para cada neurona dependen del número de entradas
- Se usa una distribución uniforme para inicializar aleatoriamente los pesos.

Entrenamiento

- El entrenamiento consiste en minimizar una función de perdida min L(x, y, w)
- La función de pérdida puede consistir por ejemplo en el error de aproximación:

$$L(x, y, w) = E[(y-f(x, w))^2]$$

- La función de pérdida depende del problema.
- En el caso de las redes feed-forward, se suele usar un método del gradiente estocástico para resolver el problema.

$$w_t = w_{t-1} - \lambda * \frac{dL(x, y, w_{t-1})}{dw}$$

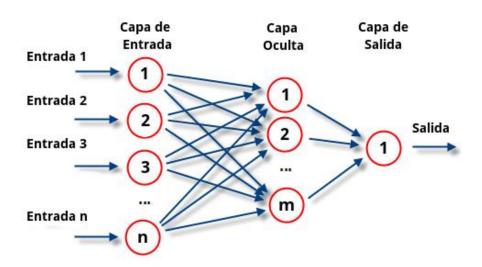
- El hiperparámetro λ se denomina tasa de aprendizaje.
- Se usan pocos ejemplos en cada iteración.

Arquitecturas de Deep Learning

- La estructura de datos fundamental de una red neuronal está vagamente inspirada en el cerebro humano. Cada una de nuestras células cerebrales (neuronas) está conectada a muchas otras neuronas por sinapsis.
- A medida que experimentamos e interactuamos con el mundo, nuestro cerebro crea nuevas conexiones, refuerza algunas conexiones y debilita a los demás. De esta forma, en nuestro cerebro se desarrollan ciertas regiones que se especializan en el procesamiento de determinadas *entradas*. Así vamos a tener un área especializada en la visión, otra que se especializa en la audición, otra para el lenguaje, etc.
- De forma similar, dependiendo del tipo de *entradas* con las que trabajemos, van a existir distintas *arquitecturas* de redes neuronales que mejor se adaptan para procesar esa información. Algunas de las arquitecturas más populares son:

Redes neuronales clásicas

 Las redes neuronales clásicas fueron las primeras que se desarrollaron y son el modelo más sencillo. En estas redes la información se mueve en una sola dirección: hacia adelante. Los principales exponentes de este tipo de arquitectura son el perceptrón y el perceptrón multicapa. Se suelen utilizar en problemas de clasificación simples.

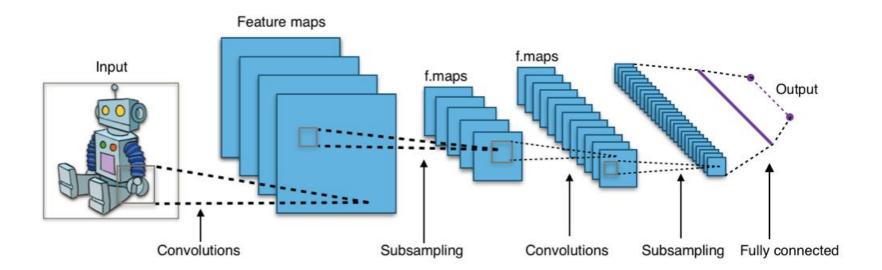


Redes Neuronales Convolucionales

- Las redes neuronales convolucionales (CNN) son muy similares a las redes neuronales ordinarias como el perceptrón multicapa; se componen de neuronas que tienen *pesos* y *sesgos* que pueden aprender.
- Cada neurona recibe algunas entradas, realiza un producto escalar y luego aplica una función de activación. Al igual que en el perceptrón multicapa también vamos a tener una función de pérdida o costo sobre la última capa, la cual estará totalmente conectada.
- Lo que diferencia a las redes neuronales convolucionales es que suponen explícitamente que las entradas son imágenes, lo que nos permite codificar ciertas propiedades en la arquitectura; permitiendo ganar en eficiencia y reducir la cantidad de parámetros en la red.

Redes Neuronales Convolucionales

Arquitectura típica de una CNN

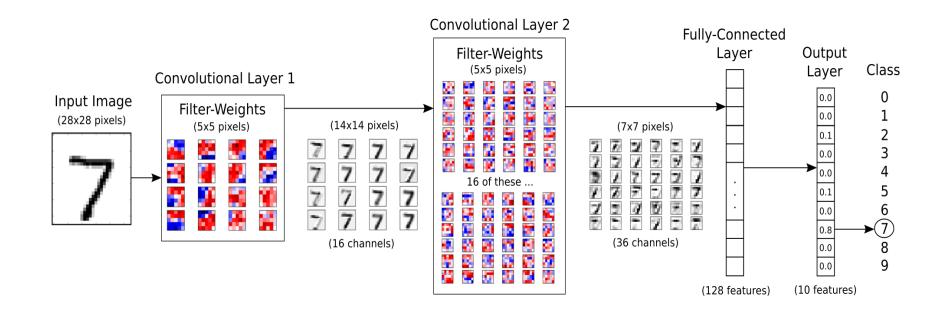


Redes Neuronales Convolucionales

- En general, las redes neuronales convolucionales van a estar construidas con una estructura que contendrá 3 tipos distintos de capas:
 - 1. Una capa convolucional (convolutional layer), que es la que le da le nombre a la red (usa filtros de convolución).
 - 2. Una capa de reducción o de pooling (pooling layer), la cual va a reducir la cantidad de parámetros al quedarse con las características más comunes.
 - 3. Una capa clasificadora totalmente conectada (fully connected layer), la cual nos va dar el resultado final de la red, nuestra clasificación.

Redes neuronales convolucionales

 Algunas implementaciones específicas que podemos encontrar sobre este tipo de redes son: Inception v3, ResNet, VGG16 y Xception, entre otras. Todas ellas han logrado excelentes resultados.

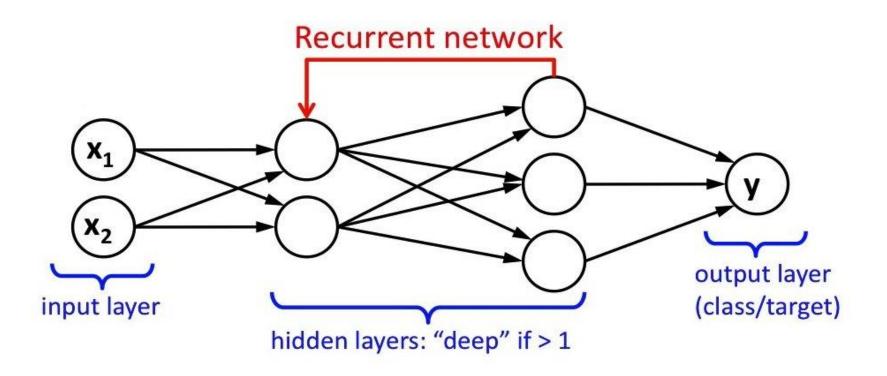


Redes neuronales recurrentes

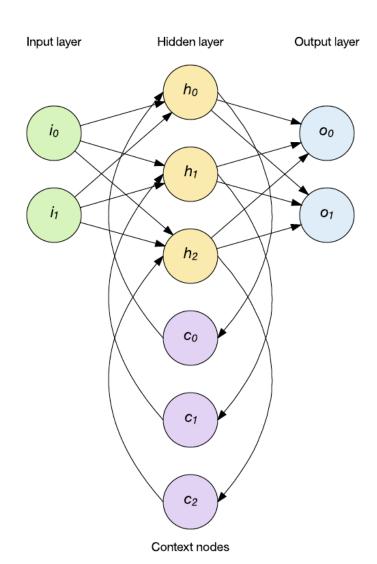
- Los seres humanos no comenzamos nuestro pensamiento desde cero cada segundo, sino que los mismos tienen una persistencia. Las redes neuronales tradicionales no cuentan con esta persistencia, y esto parece una deficiencia importante. Las redes neuronales recurrentes abordan este problema, son redes con bucles de retroalimentación que permiten que la información persista (poseen memoria).
- Una red neural recurrente puede ser pensada como una red con múltiples copias de ella misma, en las que cada una de ellas pasa un mensaje a su sucesor. Esta naturaleza en forma de cadena revela que las redes neuronales recurrentes están íntimamente relacionadas con las secuencias y listas; por lo que son ideales para trabajar con este tipo de datos. En los últimos años ha habido un éxito increíble aplicando redes neurales recurrentes a una variedad de problemas como: reconocimiento de voz, modelado de lenguaje, traducción, subtítulos de imágenes, etc.

Redes neuronales recurrentes

Arquitectura básica

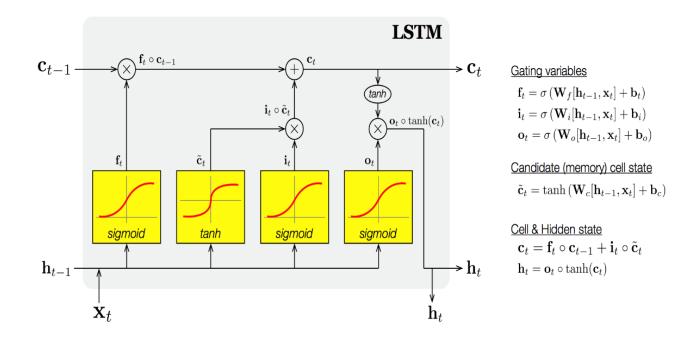


Redes neuronales recurrentes



Redes LSTM

• Las redes de memoria de largo plazo y corto plazo - generalmente llamadas LSTMs - son un tipo especial de redes neuronales recurrentes, capaces de aprender dependencias a largo plazo. Ellas también tienen una estructura como cadena, pero el módulo de repetición tiene una estructura diferente. En lugar de tener una sola capa de red neuronal, tiene cuatro, que interactúan de una manera especial permitiendo tener una memoria a más largo plazo.



Logros del Deep Learning

- En los últimos años el Deep Learning ha producido toda una revolución en el campo del Machine Learning, con resultados notables en todos los problemas de percepción, como ver y escuchar, problemas que implican habilidades que parecen muy naturales e intuitivas para los seres humanos, pero que desde hace tiempo se han mostrado difíciles para las máquinas.
- En particular, el Deep Learning ha logrado los siguientes avances, todos ellos en áreas históricamente difíciles del Machine Learning.
- Algunos logros se pueden describir a continuación:

Logros del Deep Learning

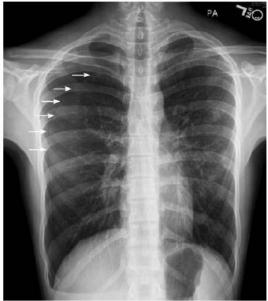
- Un nivel casi humano para la clasificación de imágenes.
- Un nivel casi humano para el reconocimiento del lenguaje hablado.
- Un nivel casi humano en el reconocimiento de escritura.
- Grandes mejoras en traducciones de lenguas.
- Grandes mejoras en conversaciones text-to-speech.
- Asistentes digitales como Google Now o Siri.
- Un nivel casi humano en autos autónomos.
- Mejores resultados de búsqueda en la web.
- Grandes mejoras para responder preguntas en lenguaje natural.
- Alcanzado nivel maestro (superior al humano) en varios juegos.

Image Recognition



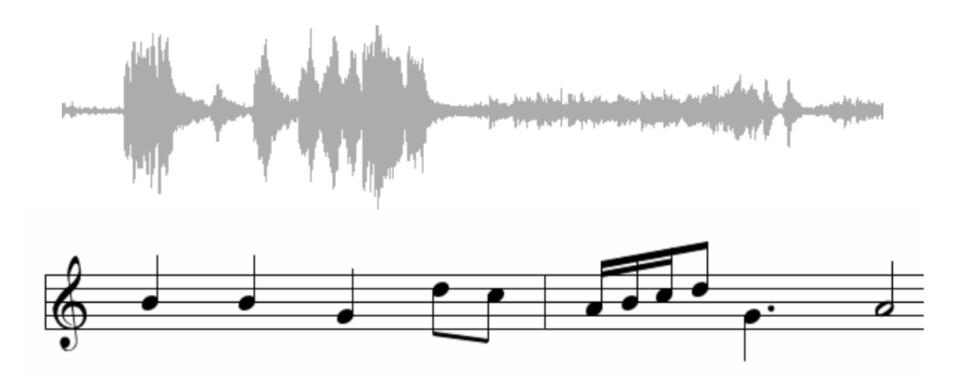
Detect pneumothorax in real X-Ray scans



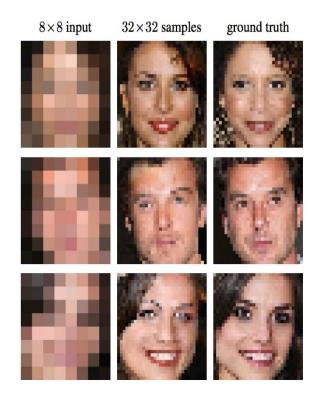




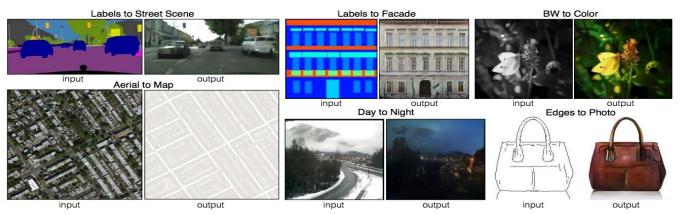
Music Generation



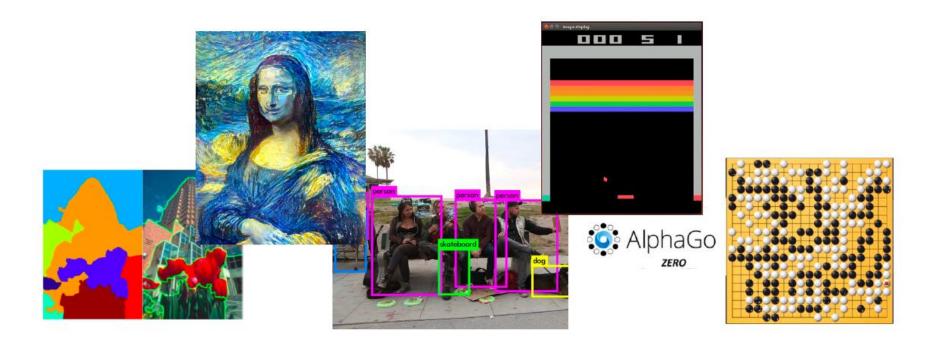
Generación de imágenes y detección







Transferencia de estilo y detección de objetos



Nuevos agentes en aprendizaje reforzado