



Recuperación de Información Multimedia

Deep Learning

CC5213 – Recuperación de Información Multimedia

Departamento de Ciencias de la Computación

Universidad de Chile

Juan Manuel Barrios – <https://juan.cl/mir/> – 2019

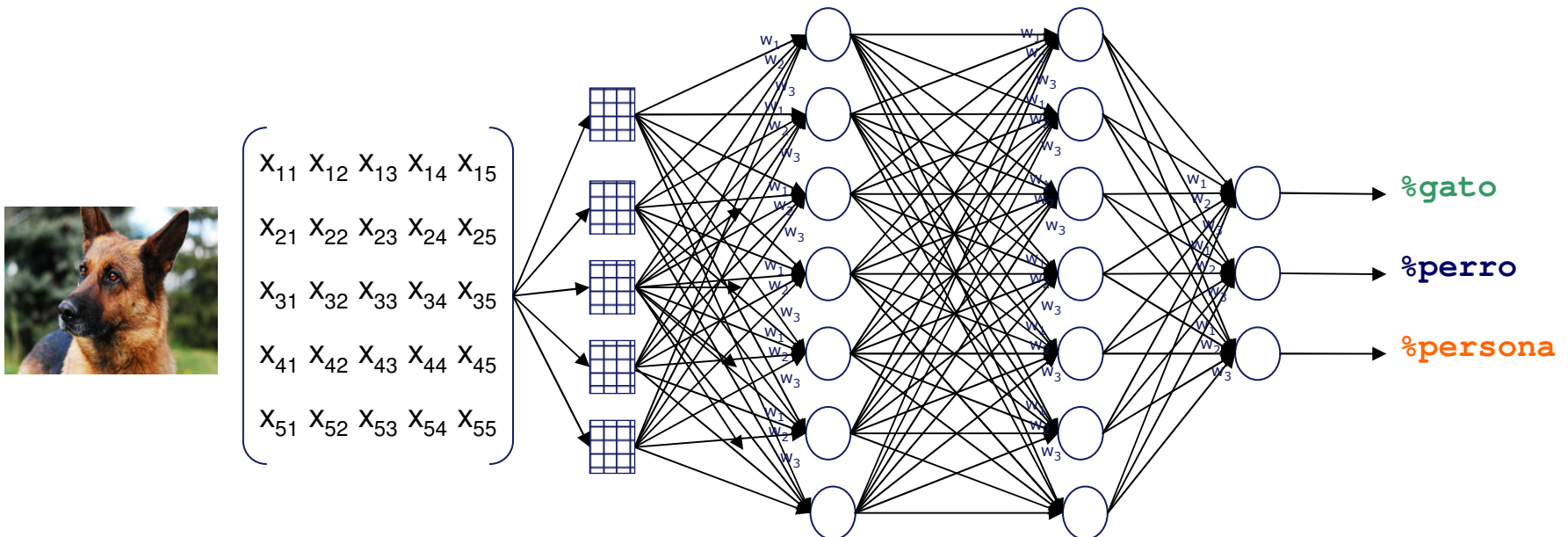


Deep Learning

- Se refiere al uso de redes neuronales “muy grandes” o “profundas”
- Se diferencia de uso de redes neuronales “tradicionales” (MLP) por:
 - Incluir el cálculo del descriptor de contenido en la misma red
 - La entrada de la red es el dato multimedia mismo
 - Las red contiene gran cantidad de parámetros (neuronas, capas, arquitecturas complejas)
 - Entrenamiento requiere gran poder computacional

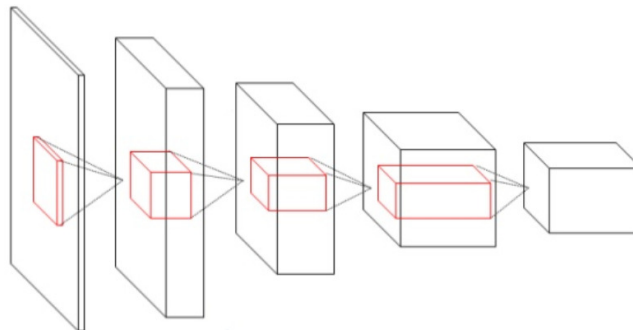
Red Neuronal Convolutiva

- Red neuronal que contiene operadores de **convolución**
 - Forma eficiente cuando se buscan patrones con localidad espacial
 - El resultado de una convolución es una imagen de (casi) el mismo tamaño
 - Los valores del filtro son parámetros a entrenar
- El operador de pooling reduce el tamaño de una imagen
- El cálculo del vector característico es parte del entrenamiento



Operador de Convolución

- Imagen de entrada de $W \times H \times D$ (alto x ancho x canales)
 - Normalizar la entrada: restar (127,127,127) u otro valor
- Tamaño del Filtro (3, 5, ...)
 - Convolución con filtros cuadrado en toda la profundidad de la entrada: tamaño 3 implica filtro de $3 \times 3 \times D$ (9D parametros)
- Cantidad de filtros (16, 32, 64, ...)
 - Un filtro produce una canal de salida
- Paso de la convolución o *stride* (1, 2, ...)
- Tamaño del borde o *zero-padding* (1, 2, ...)



Capa de Convolución

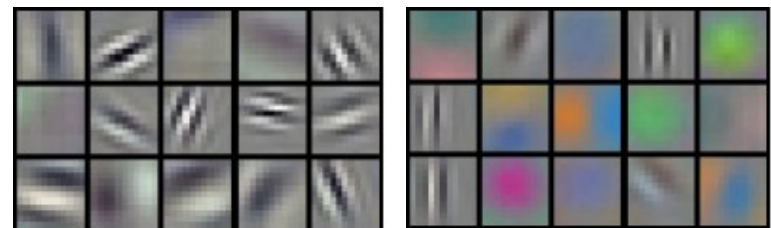
■ Salida:

- $W=H=((\text{Input} - \text{TamañoFiltro} + 2*\text{Padding}) / \text{Stride}) + 1$
- Profundidad es la cantidad de filtros
- Ej: Con $\text{stride}=1$ y $\text{padding}=(\text{TamañoFiltro} - 1)/2$ se produce una imagen con tamaño igual a la entrada

■ Cantidad de parámetros a entrenar:

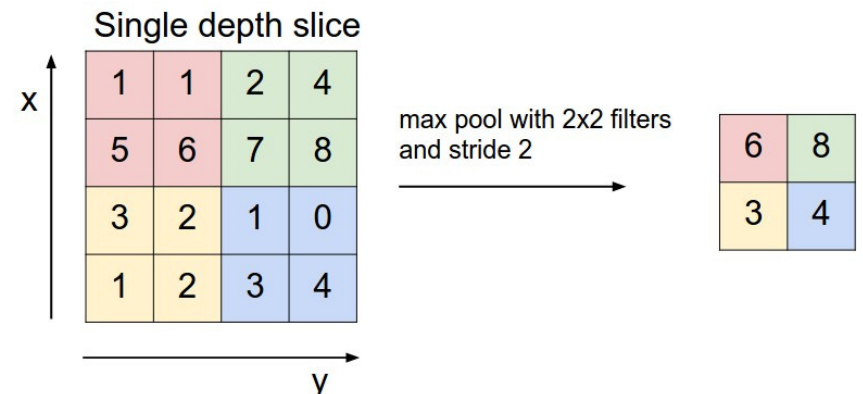
- Parámetros por capa: $\text{NumFiltros} \times (F \times F \times D + 1 \text{ bias})$
- Notar que no se entrena un filtro por pixel si no que el filtro opera sobre toda la imagen (Parameter Sharing)

■ Filtro de 1x1 se usa para reducir la profundidad



Capa de Pooling

- Operación fija en la coordenada espacial
- Entrada: $W \times H \times D$
- Tamaño de la ventana (F) usualmente 2
- Tamaño del paso o *stride* (S) usualmente 2
- Tipo de operación: MAX-Pooling
 - Se puede usar AVG-Pooling aunque los resultados muestran mejores resultados con Max (seleccionar el mayor)
- Output:
 - Ancho = $(W-F)/S + 1$
 - Alto = $(H-F)/S + 1$
 - Profundidad = D
- No hay parámetros a entrenar



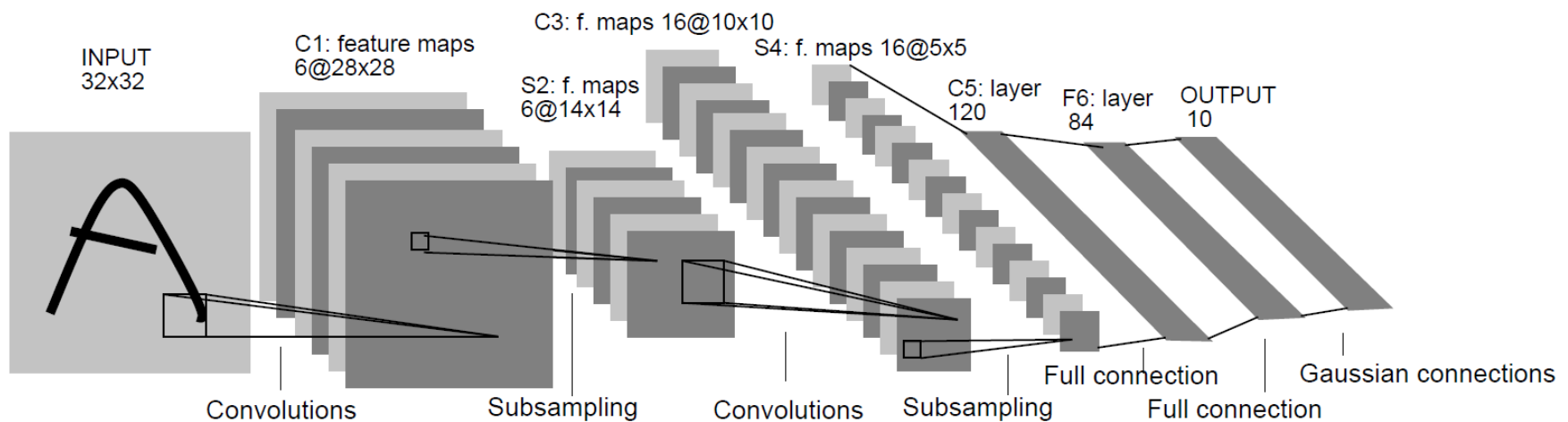


Capa Fully Connected

- Red MLP Tradicional
- Red densa, conecta todas las entradas con todas las salidas
- Simula un clasificador tradicional que opera sobre el descriptor de contenido
- Cantidad de parámetros a entrenar:
 - $\text{Entrada} \times (\text{Salida} + 1 \text{ bias})$

LeNet (1998)

- Una red convolucional (CNN) para dígitos



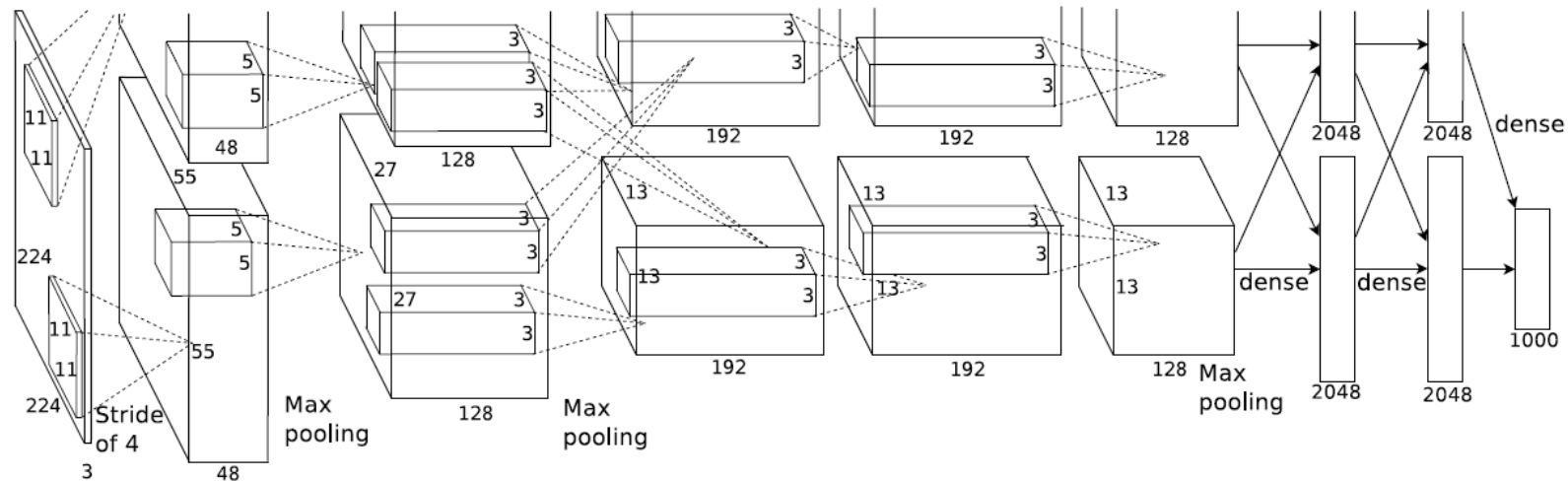


ImageNet

- ImageNet
 - 1.5 millones de imágenes de entrenamiento (ahora muchas más ~15 millones)
 - Imágenes etiquetadas en 1000 categorías
 - Uso de WordNet para agrupar significados (synsets)
- ILSVRC: ImageNet Large Scale Visual Recognition Competition, Top5 error rate:
 - 2010 28%
 - 2011 26%,
 - 2012 16% (AlexNet, 8 capas)
 - 2013 12% (ZFNet, 8 capas)
 - 2014 7% (GoogLeNet, 22 capas) 7.5% (VGG-16 VGG-19)
 - 2015 4% (ResNet, 152 capas)
 - 2016 3% (TSNet, ensemble)
 - 2017 2.2% (SENet, ensemble)

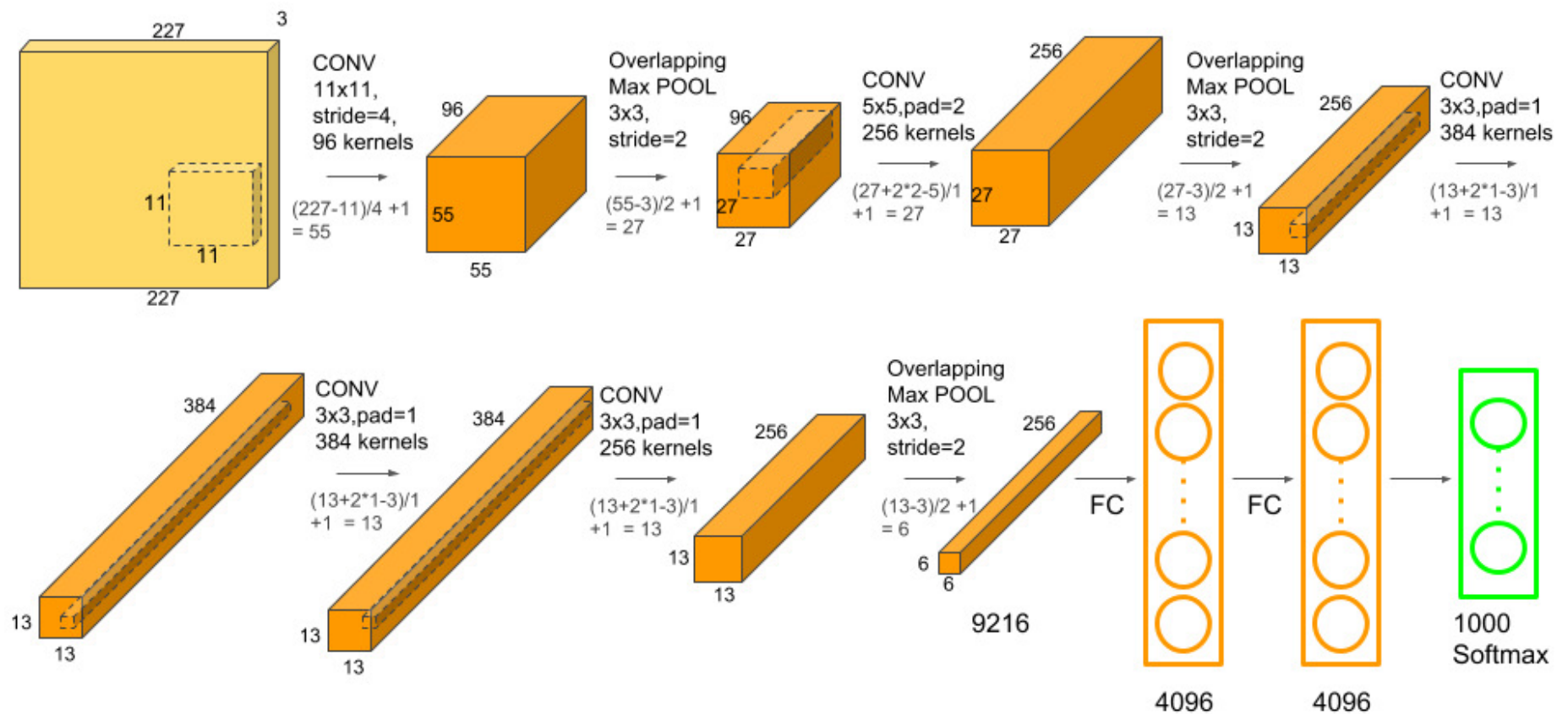
AlexNet (2012)

Krizhevsky, Sutskever, Hinton. ImageNet Classification with Deep Convolutional Neural Networks. 2012



- **Input:** 227 alto x 227 ancho x 3 canales (rgb)
 - **Convolución 1:** 96 filtros 11x11x3, stride 4 → 55x55x96 (0,03 M params)
 - **Max Pooling 1:** kernel=3, stride=2 → 27x27x96
 - **Convolución 2:** 2 grupos 128 filtros 5x5x48, padding 2 → 27x27x256 (0,31 M params, 1%)
 - **Max Pooling 2:** kernel=3, stride=2 → 13x13x256
 - **Convolución 3:** 2 grupos 192 filtros 3x3x128, padding 1 → 13x13x384 (0,44 M params, 1%)
 - **Convolución 4:** 2 grupos 192 filtros 3x3x192, padding 1 → 13x13x384 (0,66 M params, 1%)
 - **Convolución 5:** 2 grupos 128 filtros 3x3x192, padding 1 → 13x13x256 (0,44 M params, 1%)
 - **Max Pooling 5:** kernel=3, stride=2 → 6x6x256
 - **Fully Connected 6:** Input 9216 → Output 4096 (37,8 M params, 62%)
 - **Fully Connected 7:** Input 4096 → Output 4096 (16,8 M params, 28%)
 - **Fully Connected 8:** Input 4096 → Output 1000 (4,1 M params, 7%)
- (Total=60,5 M params * 4 bytes = 242 MB)

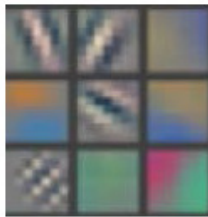
AlexNet (2012)



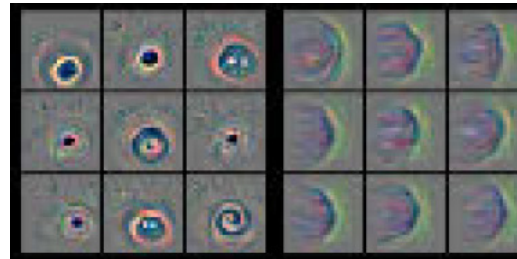
<https://www.learnopencv.com/understanding-alexnet/>



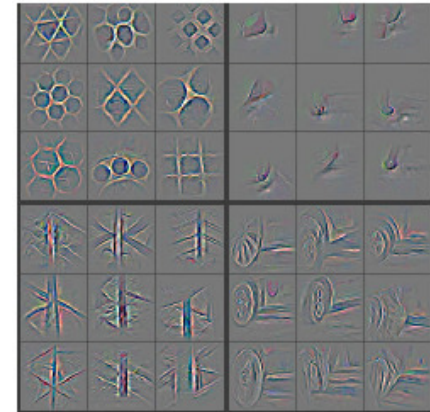
Filtros de convolución entrenados



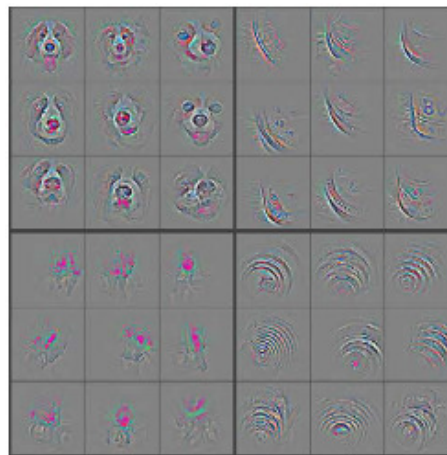
Filtros capa 1



Filtros capa 2



Filtros capa 3



Filtros capa 4

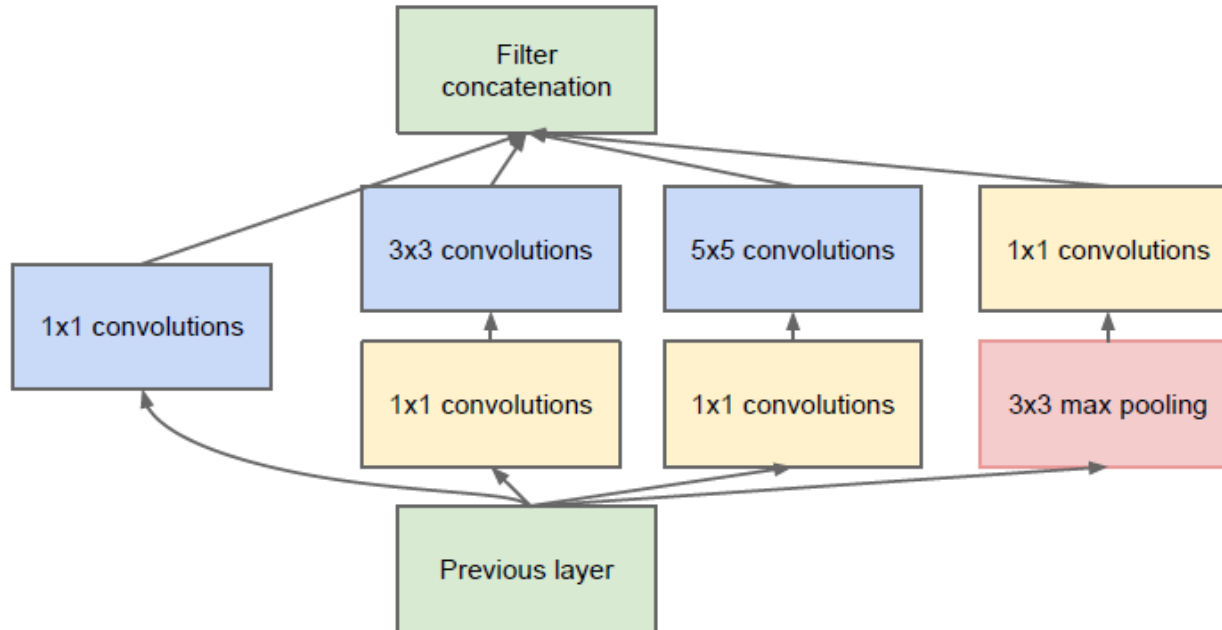


Filtros capa 5

Zeiler, Fergus. Visualizing and Understanding Convolutional Networks. 2013.

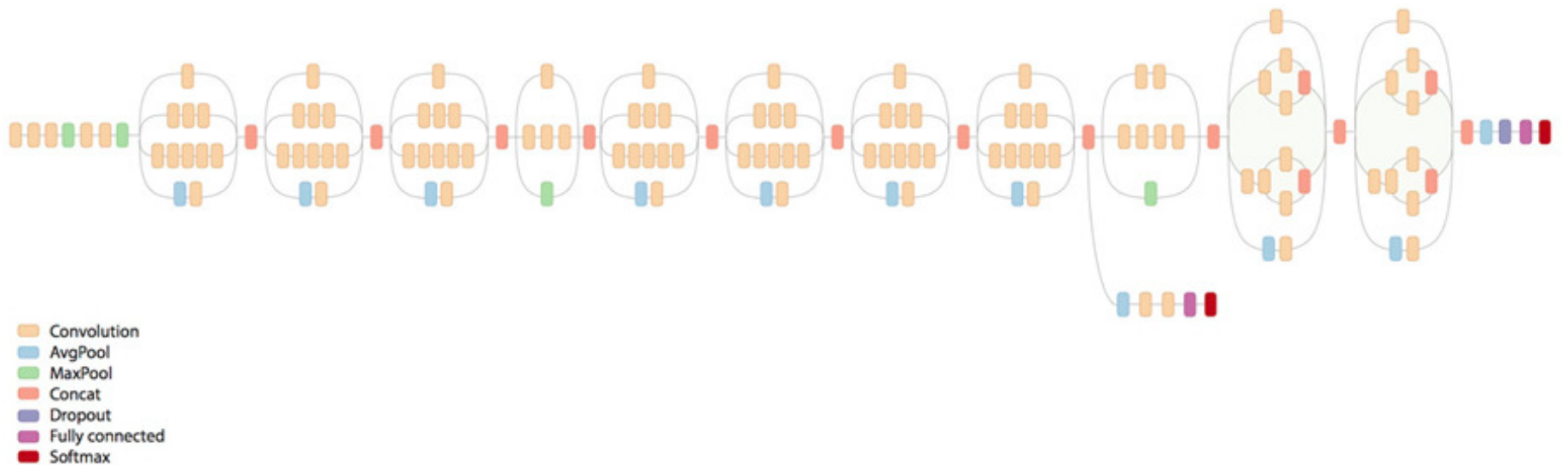
GoogleNet (2014)

- Procesar filtros de distintos tamaños en paralelo y concatenarlos (“Inception”)



GoogleNet (2014)

- No tiene Fully Connected



Szegedy et al. Going Deeper with Convolutions. 2015.

VGG (2014)

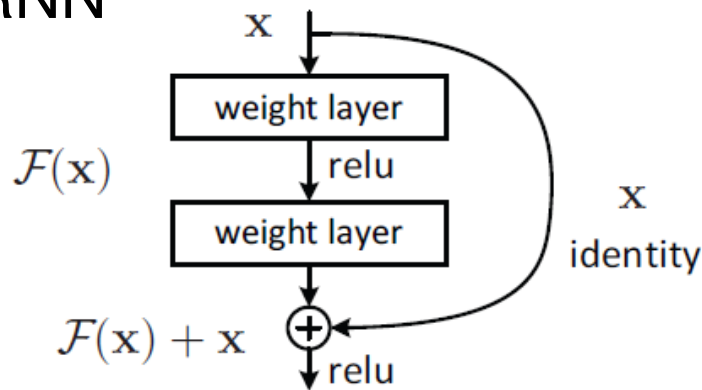
- VGG-16, VGG-19
- VGG-Face (VGG-16 entrenada con rostros)
- Muy grandes
 - Pesos requieren ~500 MB !

- Simonyan, Zisserman. Very Deep Convolutional Networks For Large-Scale Image Recognition. 2014
- http://www.robots.ox.ac.uk/~vgg/research/very_deep/
- http://www.robots.ox.ac.uk/~vgg/software/vgg_face/

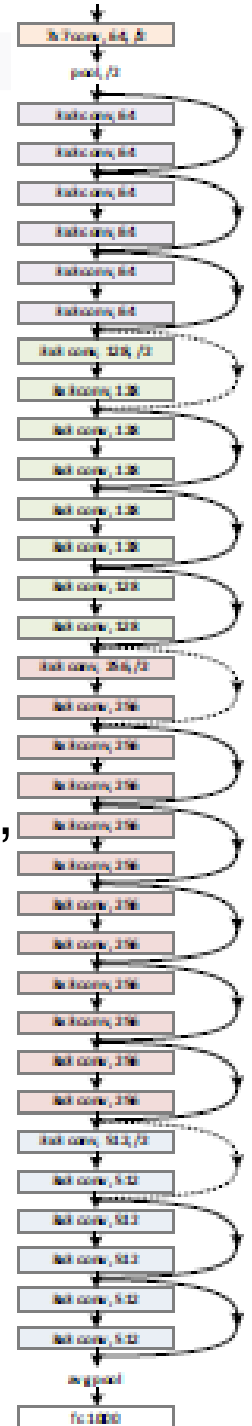


ResNet (2015)

- 152 capas
 - Una convolución de 7x7 y max-pooling, luego muchas convoluciones de 3x3...
- Bloque Residual
 - Intenta evitar el problema del “vanishing gradient”
 - Similar a una RNN



He et al. Deep Residual Learning for Image Recognition. 2015.





Deep Features

- Obtener un descriptor global de imagen usando valores de una capa previa al output de clasificación
 - Para AlexNet y VGG usualmente capa fc6 o fc7 (4096)
 - Para ResNet la capa de la última convolución (2048)
- Descriptor con propiedades similares al obtenido a través de Bag-Of-Visual-Words
- Permite buscar objetos visualmente parecidos a la imagen de consulta calculando distancias entre descriptores

Uso de Deep Features

- Deep Features son descriptores de contenido que capturan información de alto y bajo nivel
 - Comparar descriptores con distancia L_2 o L_1
 - Descriptores cercanos tienden a ser de la misma clase además de visualmente similares
- Permite buscar imágenes que contengan el objetos visualmente parecidos



Sillas		
	Sillón Ejecutivo con masajeador negro Genérico Sillas \$99990.0	0.8
	Sillón Ejecutivo Negro Asenti Sillas \$99990.0	0.8
	Sillón Ejecutivo Negro Asenti Sillas \$25990.0	0.9
Sillas de Terraza		
	Silla zero gravity Genérico Sillas de Terraza, Terraza 2016 \$49990.0	1.0
	Silla Con Brazos Plegable Metal Textil Genérico Sillas de Terraza \$16990.0	1.0



Problemas de Deep Features





- Usualmente la CNN son clasificadores entrenados con ImageNet (1000 clases)
 - No funciona muy bien cuando la imagen no contiene objetos de ImageNet
 - No funciona muy bien para buscar texturas, logos, ni patrones visuales sin semántica
 - El Deep Feature es entrenado por un clasificador y muchas clases son independientes de los colores
- Descriptores locales, como SIFT, buscan patches similares entre imágenes, sin semántica
 - Permiten encontrar logos y patrones
 - Robusto a rotaciones y oclusión parcial

Problema de Deep Features

- Deep Features no funciona muy bien buscando imágenes que comparten alguna zona (por ejemplo un mismo logo)
 - Caso productos de supermercado

Deep Features:

Similitud basada en Deep Features (como AlexNet) busca imágenes que contengan un objeto parecido (ignora el logo)

	
Otros	
	Scotch 3M Pack Cintas de Embalaje 2 Transparentes 1 Café, 3 Rollos 48 mm x 30 mts c/u. Ferretería/Automotor, Hogar \$1.890 Und \$630 x Unidad
	Stabilo Goma de Borrar 2 unidades diseño legacy Librería, Hogar \$790 Und \$395 x Unidad
	Lay's Papas Fritas Stax Original 40Grs. Cóctel, Despensa \$529 Und \$13.225 x Kilo
	Maretti Brusquette Tomate-Aceituna-Orégano Agrega 2 x \$ 990 Cóctel, Despensa \$1.199 Und \$1.199 x Unidad
	Surlat Queso Gouda Láminado Light, 250 grs. Quesos, Frescos \$2.129 Und \$8.516 x Kilo

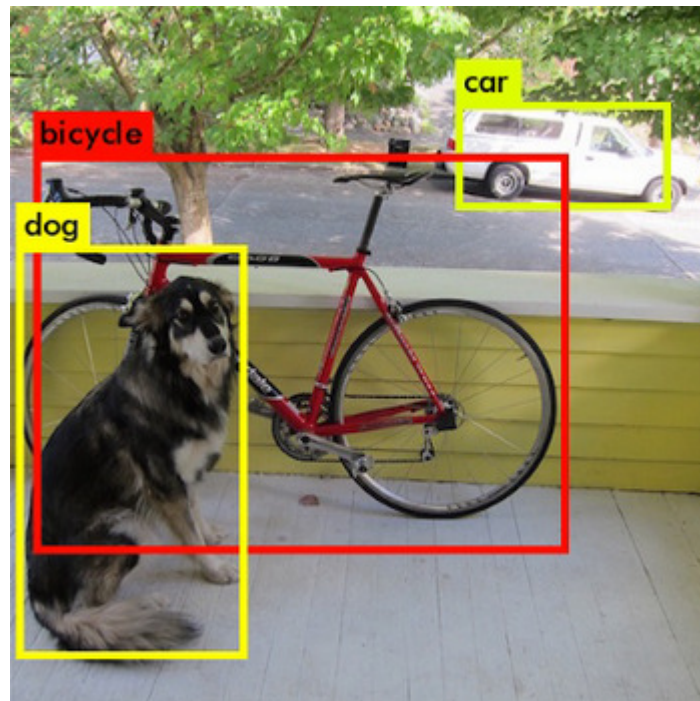
Descriptores SIFT:

Similitud basada en patches similares (encuentra todos los productos con un mismo logo y colores)

	
Otros	
	Lipton Té Blanco Blueberry y Pomegranate, Con mezcla de té verde y frut... Té y Café, Dulces \$2.890 Und \$161 x Unidad
	Lipton Té Verde Green Tea, Con trocitos de frutas, Saborizado con mand... Té y Café, Dulces \$2.629 Und \$131 x Unidad
	Lipton Té en Hojas Yellow Label. Bolsa 225 g. Té y Café, Dulces \$2.890 Und \$12.844 x Kilo
	Lipton Té Negro Vainilla Caramel Truffle Tea, Con trocitos de caramelo,... Té y Café, Dulces \$2.719 Und \$136 x Unidad
	Lipton Té Royal Ceylán, 100 Bolsas, Caja 200 grs.

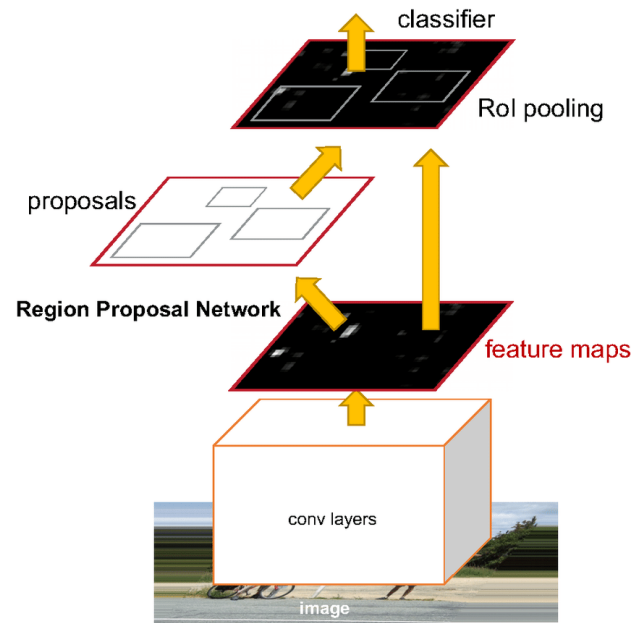
CNN con Regiones (R-CNN)

- Se desea clasificar una imagen y además dar un recuadro de la ubicación



Faster R-CNN

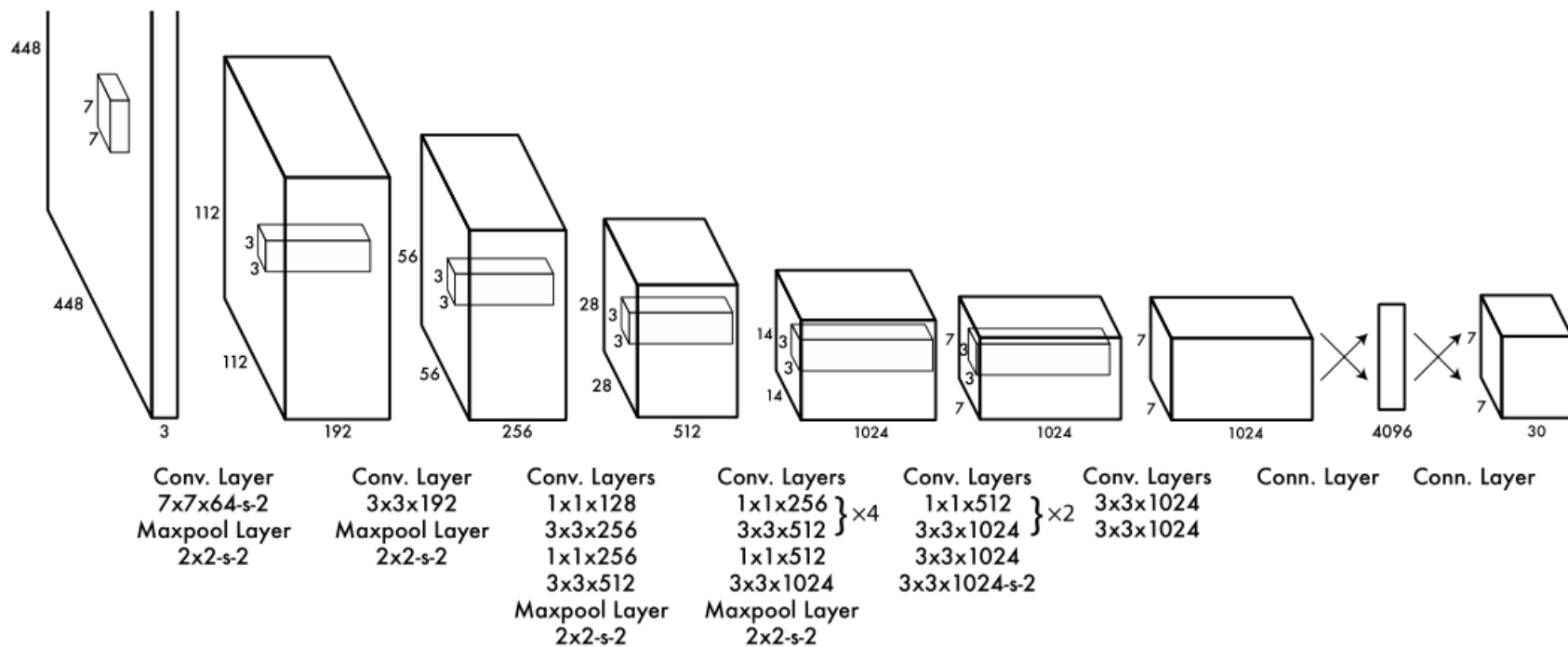
- Se usa en dos etapas: encontrar una propuesta de zonas y luego clasificar las zonas



Ren et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2015

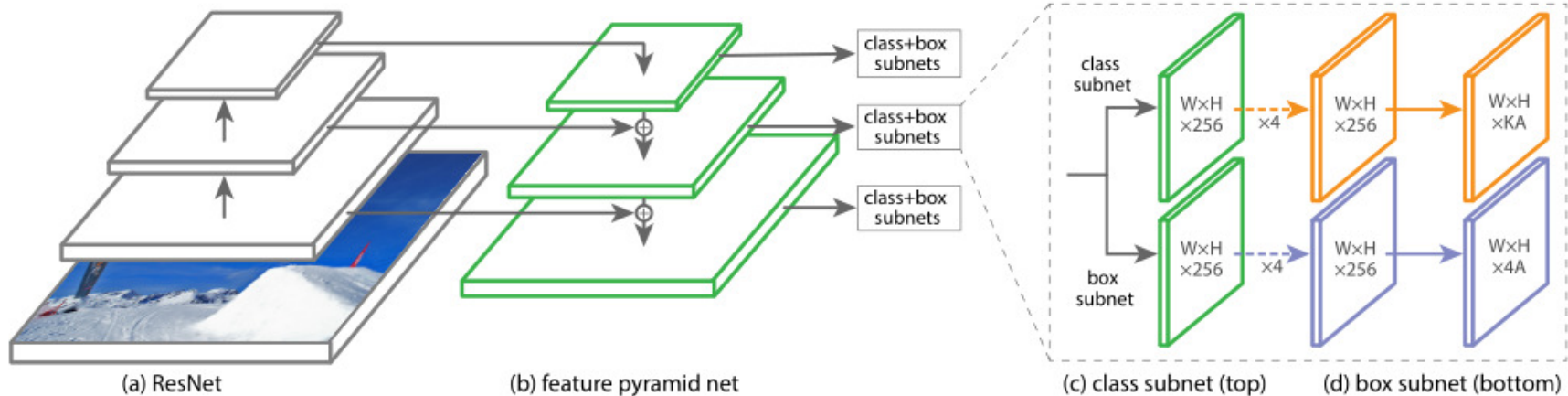
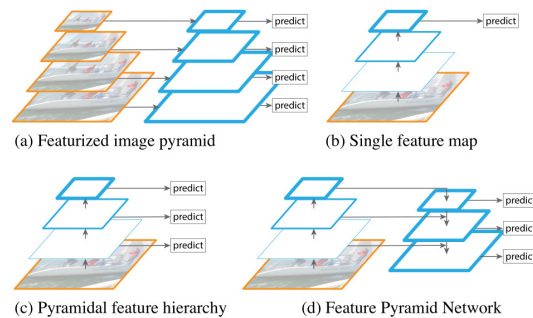
YOLO

- Entrega cajas y clases en una única regresión



Redmon et al. You Only Look Once: Unified, Real-Time Object Detection. 2016.

Feature Pyramids y RetinaNet



Lin et al. Feature Pyramid Networks for Object Detection. 2017.
Lin et al. Focal Loss for Dense Object Detection. 2018.

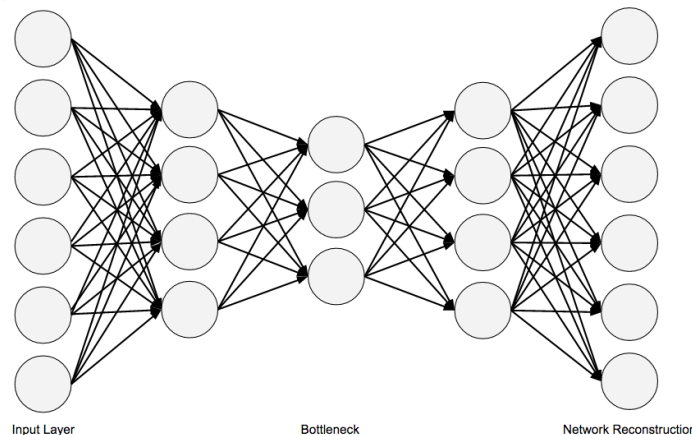


Uso de RCNN

- RCNN permite localizar uno o más objetos en la imagen y la clase de cada objeto
- Usualmente las RCNN son entrenadas con el dataset COCO Detection
 - 80 clases (bastante poco)
 - Funciona muy bien para identificar la personas
 - No funciona bien con frames de videos de baja calidad
 - Existen otros datasets recientes para entrenar RCNN
- En general, usar un RCNN para localizar regiones con objetos y luego calcular un descriptor de cada region con una CNN

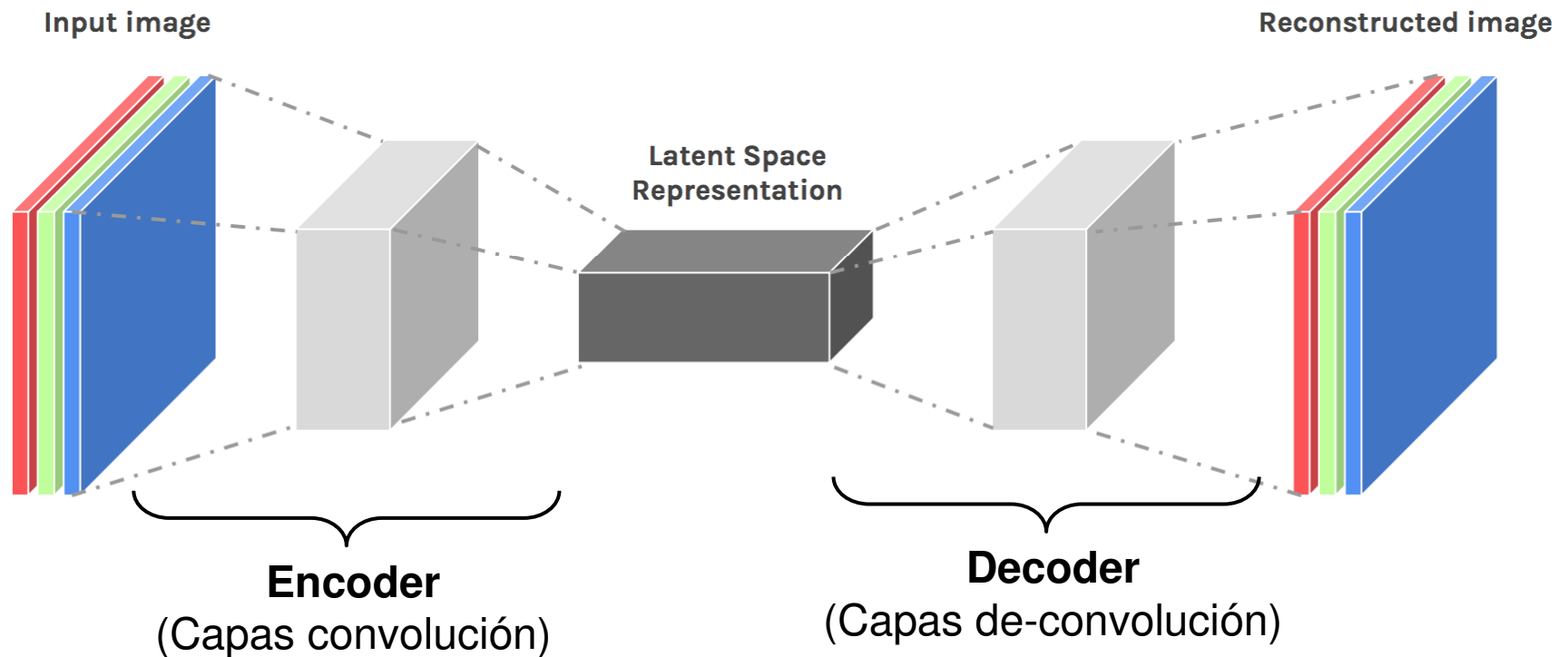
Métodos No Supervisados

- **Autoencoder:** Red neuronal donde la entrada y salida tienen el mismo tamaño y contiene una capa oculta de menor tamaño
- Se entrena para que la salida sea idéntica a la entrada
 - Debe aprender patrones comunes en el dataset para poder reconstruir exitosamente todos los vectores
 - Cada neurona se va especializando en detectar patrones típicos



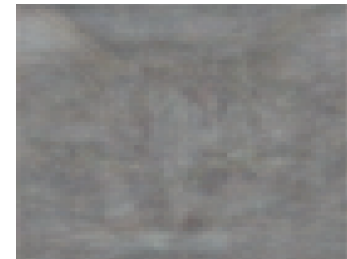
Auto-Encoders

- En la capa intermedia se obtiene una representación comprimida del objeto



Auto-Encoders

- Para autoencoders usando CNN se requiere un paso de “deconvolution” (o transposed convolution) y “unpooling”
 - Agrandar una imagen y deshacer una convolución
- Es posible entrenar un autoencoder con muchos frames de videos de Youtube.
 - Neuronas se especializan en patrones comunes, por ejemplo detectar gatos



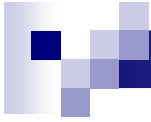
<https://blog.manash.me/implementing-pca-feedforward-and-convolutional-autoencoders-and-using-it-for-image-reconstruction-8ee44198ea55>

- Zeiler et al. Deconvolutional Networks. 2010
- Zeiler et al. Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. 2011
- Le et al. Building High-level Features Using Large Scale Unsupervised Learning. 2012.



Uso de Auto-Encoders

- Calcular descriptores de contenido:
 - Se requieren muchos datos para que el entrenamiento logre detectar patrones relevantes
- Entrenar un denoiser:
 - Entrenar el auto-encoder donde la entrada es el objeto con ruido (agregado artificialmente) y la salida es el objeto original sin ruido
 - Muy útil para limpiar ruido de fondo en audio
- Detección de anomalías:
 - Si se usa el auto-encoder con una imagen muy distinta a los datos de entrenamiento (anomalía), la salida tendrá muy mala calidad y se podrá detectar



Word Embeddings



Vectorización

- Para poder utilizar texto en una red neuronal es necesario “vectorizar” el texto
- Primero se debe segmentar el texto en palabras (Tokenizer)
 - Ver capítulo de Bag-of-Words
- Convertir cada palabra (token) en un vector:
 - One-hot encoding
 - Word embedding



One-Hot Encoding

- Primero se debe obtener el vocabulario (lista de palabras conocidas)
- Para un vocabulario de n palabras, la i -ésima palabra se codifica con un vector de n dimensiones, con un 1 en la i -ésima coordenada y 0 en el resto:

$$(0, \dots, 0, 1, 0, \dots, 0)$$

- Es una codificación sparse (muchos ceros)
- Alta dimensionalidad
- Todas las palabras son igualmente distintas



Word Embeddings

- Representar palabras con vectores cuya distancia se ajuste a su diferencia en significado
- Es una codificación densa
- Menor dimensionalidad que one-hot (ej.: 300-d)
- Similitud entre palabras se debe a similitudes en su contexto
- Se entrena una conversión desde vectores one-hot a vectores densos usando una MLP
 - Es posible usar vectores pre-entrenados para vocabularios conocidos o entrenarlos para cada problema a resolver

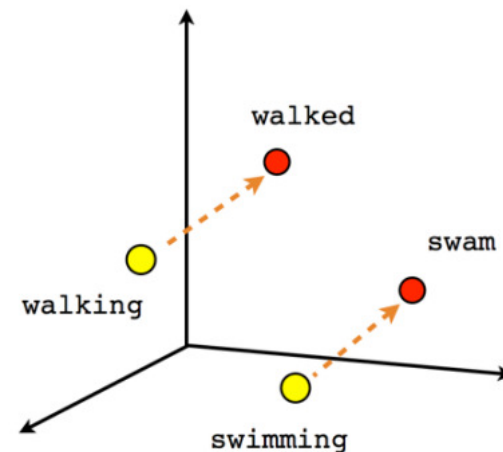
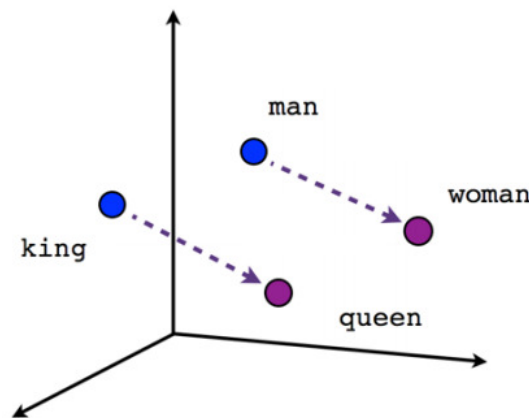


Word Embedding Space

- Se espera que el espacio de las palabras tenga propiedades como:
 - Palabras que son sinónimos estén asociadas a vectores muy cercanos entre si (distancia euclidiana cercana a cero)
 - Las direcciones en el espacio tengan algún significado de operación con las palabras:
 - singular a plural, masculino a femenino, sustantivo a adverbio, infinitivo a participio, presente a pasado, etc.
 - “el día soleado” ↔ “los días soleados”
 - “el gato negro” ↔ “la gata negra”
- El espacio de las palabras depende del idioma y también del uso (legal, técnico, popular)

Word Embedding Space

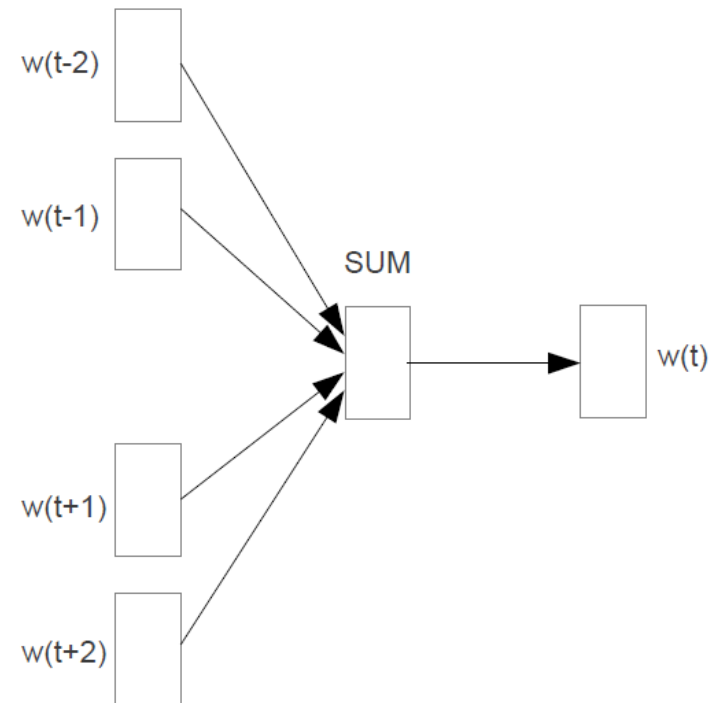
- Permiten resolver analogías:
 - (sintáctico) “Aparente” es a “Aparentemente” como “Evidente” es a ...
 - (semántico) “Atenas” es a “Grecia” como “Oslo” es a ...



Entrenamiento Word2Vec

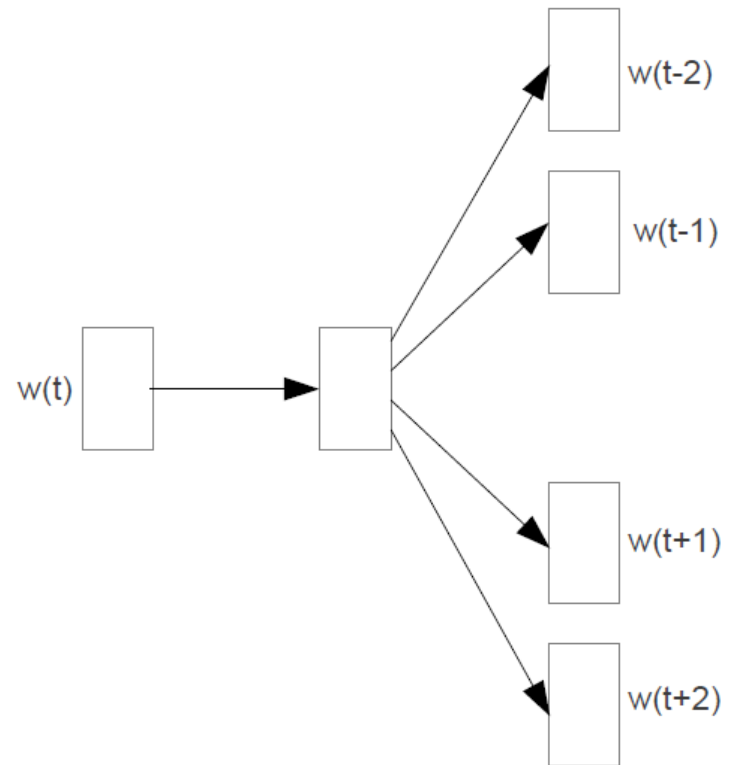
■ Modelo Continuous Bag-of-Words

- Predecir una palabra dadas sus palabras de contexto
- Produce vectores con mejor resultado en predicción sintáctica



Entrenamiento Word2Vec

- Modelo Continuous Skip-gram
 - Dada una palabra predecir sus palabras de contexto
 - Produce vectores con mejor resultado en predicción semántica





GloVe

- <https://nlp.stanford.edu/projects/glove/>
- Se basa en factorizar una matriz de co-ocurrencia de palabras
- Muy similar en idea a Latent Semantic Analysis
 - Ver capítulo de Bag-of-Words y LSA



FastText

- <https://github.com/facebookresearch/fastText>
- Calcula vectores para secuencias de caracteres y los suma para crear el vector de cada palabra
- Permite generar un vector para palabras desconocidas



Sentence Embedding

- Calcular un vector para una frase.
 - Promedio de los word vectors
 - Eliminar stop-words
 - Promedio ponderado por IDF

$$v_s = \frac{1}{|s|} \sum_{w \in s} \text{IDF}_w v_w \qquad \text{IDF}_w := \log \frac{1 + N}{1 + N_w}$$

- Doc2Vec: Entrenar word2vec incluyendo un id del documento (sentence)

Arora et al. A simple but tough-to-beat baseline for Sentence Embeddings. 2017.
Le, Mikolov. Distributed Representations of Sentences and Documents. 2018.

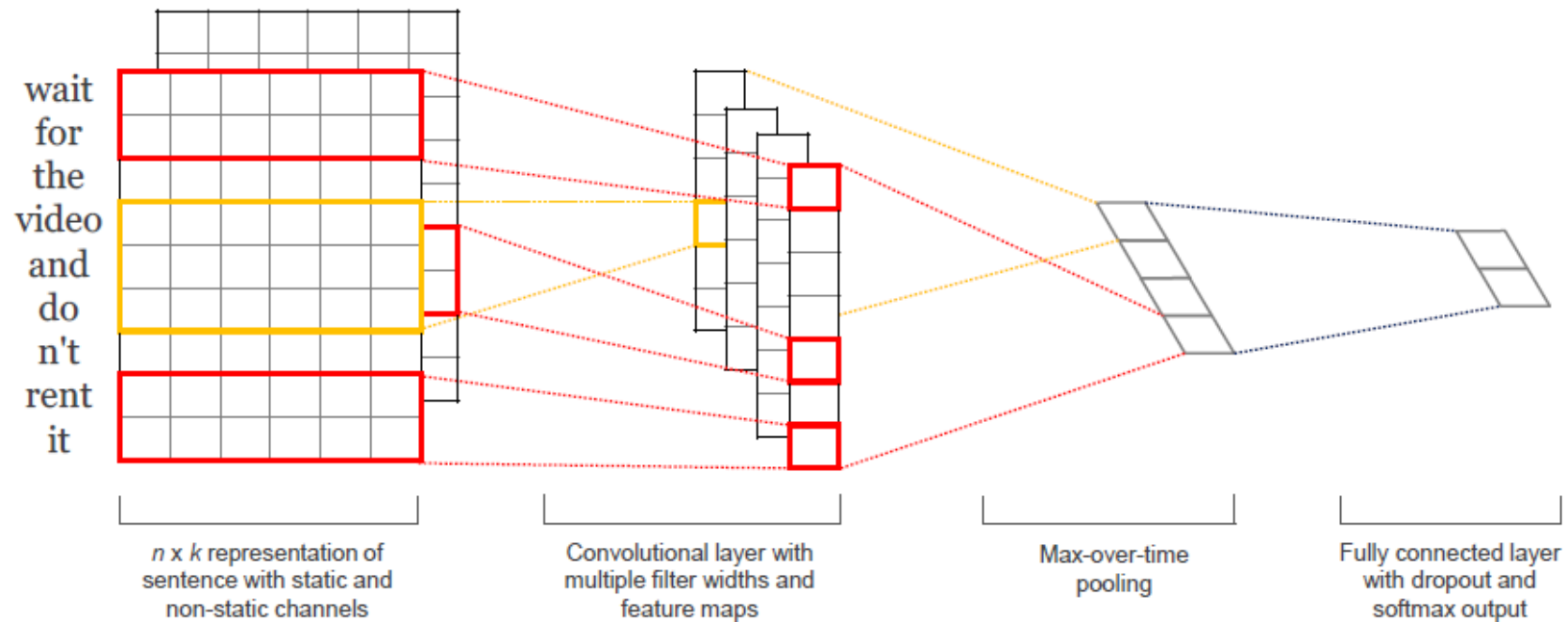


Otros Vector Embeddings

- **Node2Vec**: Vectorizar un grafo calculando un vector por nodo al medir nodos vecinos
- **Item2Vec**: Calcular vectores de ítems para sistemas recomendadores

CNN para Texto

- Conv1D es similar a N-Grams



Kim. Convolutional Neural Networks for Sentence Classification. 2014



CNN para Texto

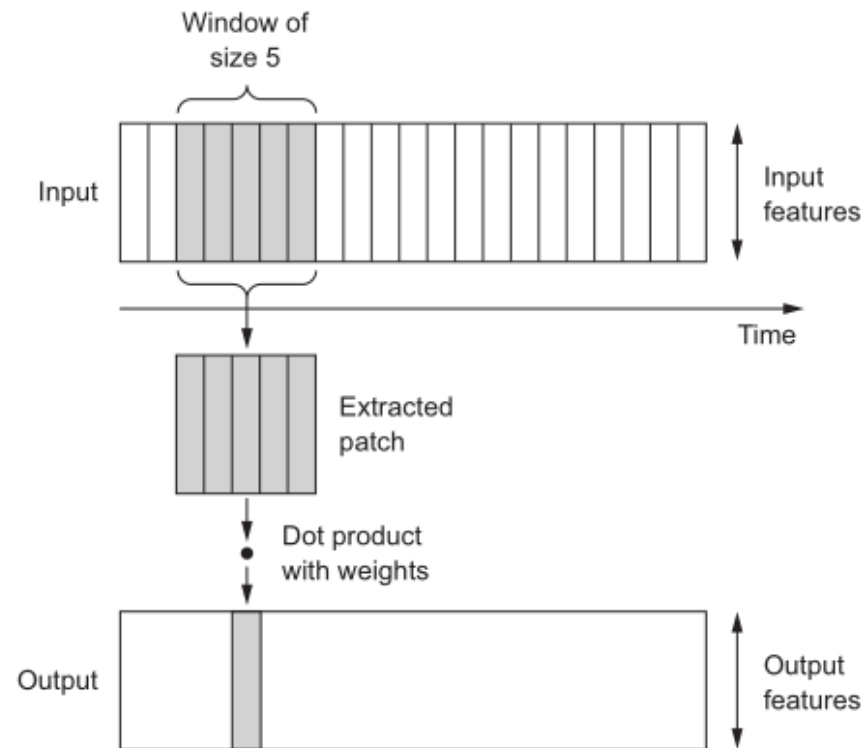
- Input: n palabras, cada palabra es un vector del word embedding de dim k (ej. $k=300$)
- Convolución: Un filtro de tamaño h corresponde un vector de $h*k$ que se usa como producto punto con una ventana de h palabras

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b) \quad \mathbf{c} = [c_1, c_2, \dots, c_{n-h+1}]$$

- Max-Pooling en el tiempo $\hat{c} = \max\{\mathbf{c}\}$
- Se concatenan varios filtros para formar un vector
- 100 filtros de tamaño 3, 4, 5

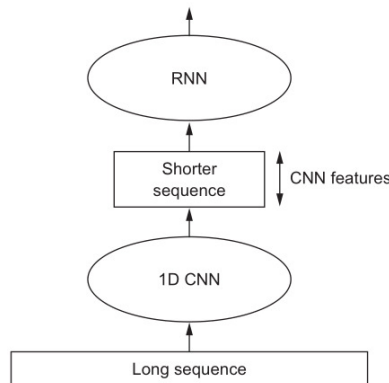
CNN para Texto

- Convolución 1D es el producto punto entre vectores



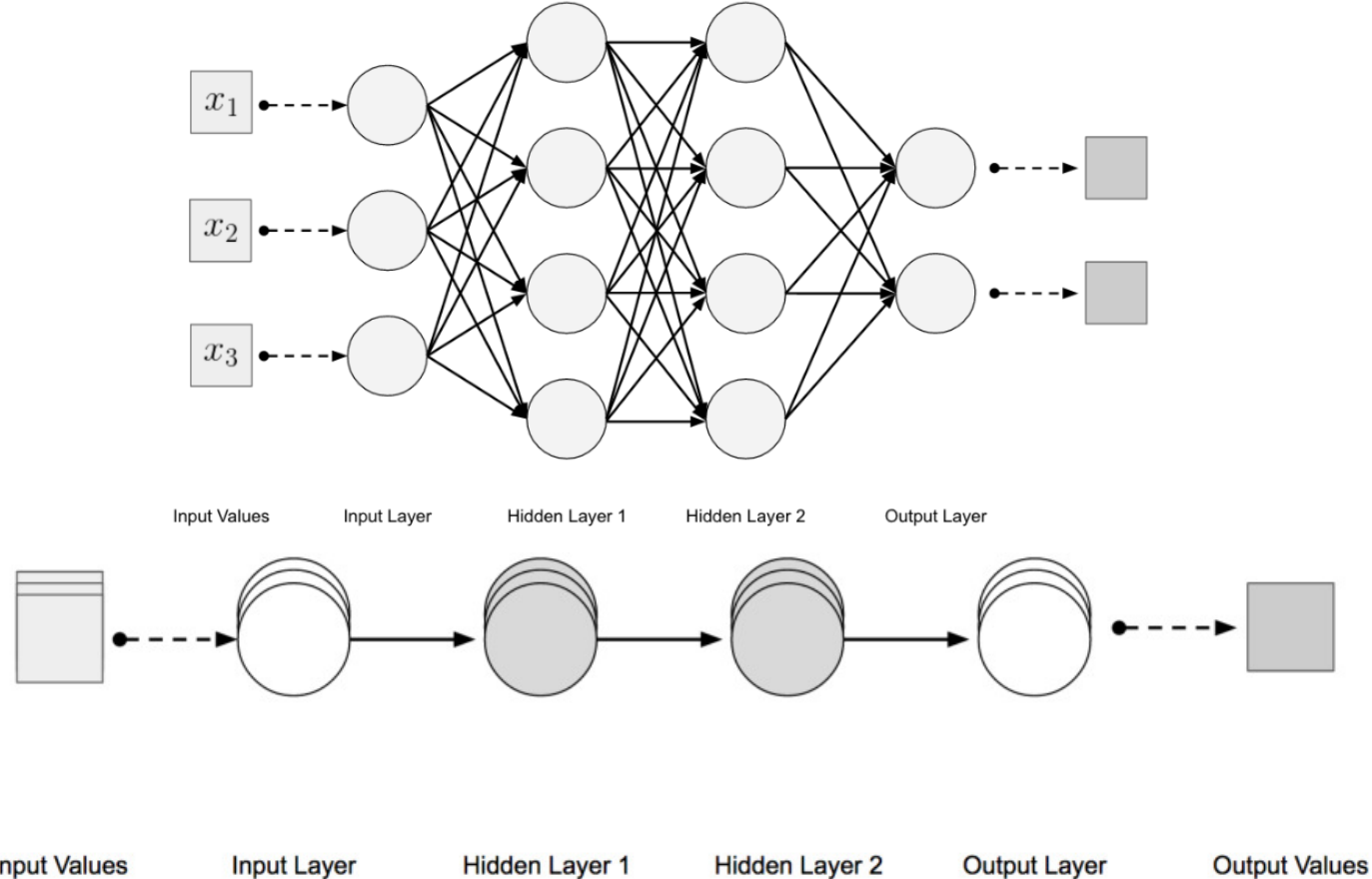
CNN para Texto

- Conv1D permiten hacer detección de grupos de palabras como n-grams
- Las convoluciones no ven la secuencias en el tiempo
- El mejor resultado se obtiene con una primera capa de Conv1D y luego una red recurrente que vea la secuencia temporal.



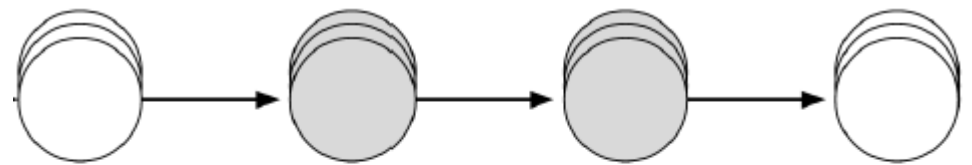
Redes Recurrentes

■ Feed-Forward vs Recurrent

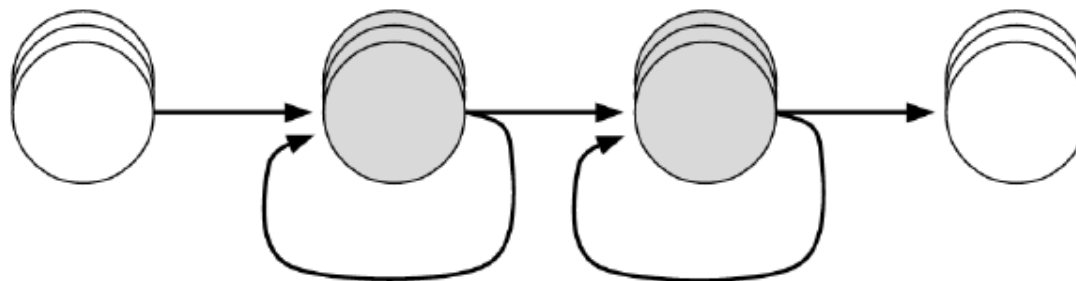


Redes Recurrentes

■ Feed-Forward vs Recurrent



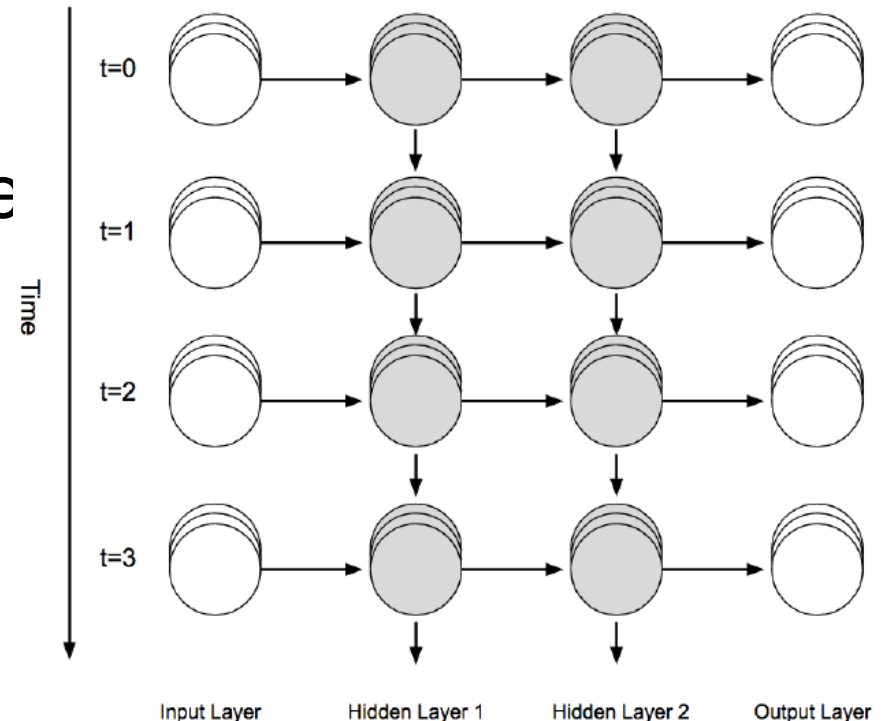
Input Layer Hidden Layer 1 Hidden Layer 2 Output Layer



Input Layer Hidden Layer 1 Hidden Layer 2 Output Layer

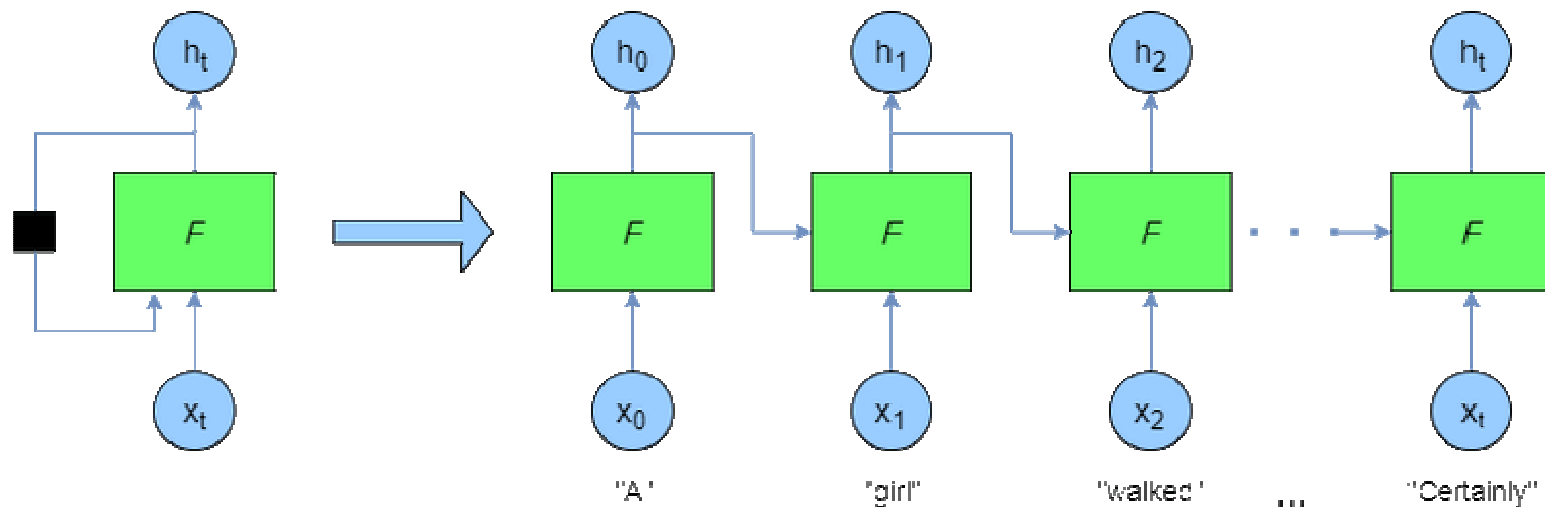
RNN

- Redes Recurrentes se usan para procesar datos con dimensión temporal, donde importa el orden de los datos
- RNN simples (“vanilla”) sufren del Vanishing Gradient para mantener información entre varias ventanas



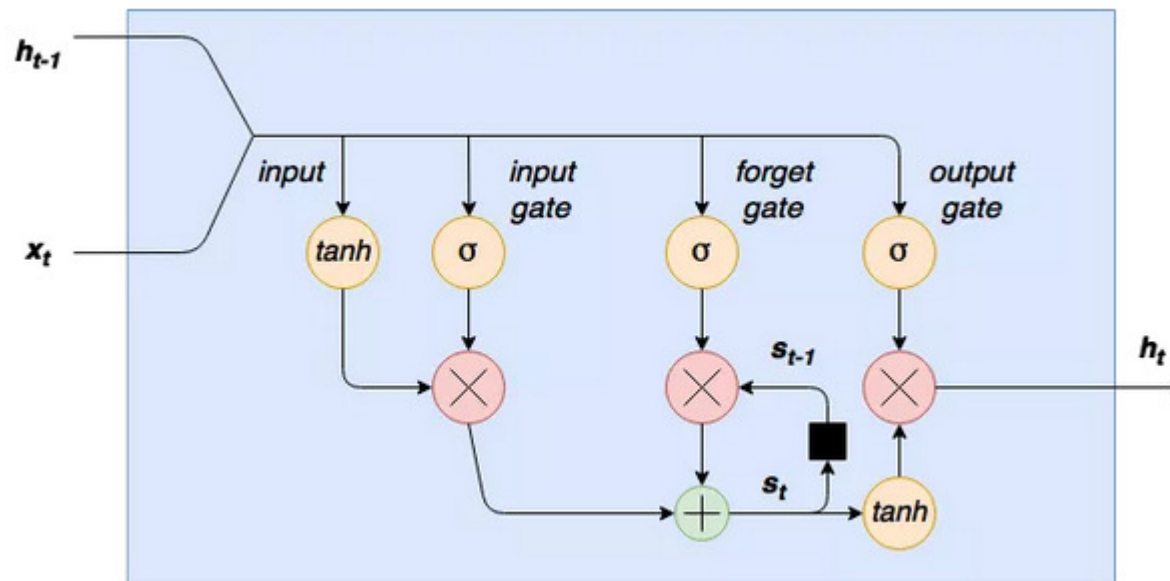
RNN

- El input de la red es una secuencia de vectores x_0 a x_t de la misma dimensión que se consumen uno a uno
- Cada entrada produce una salida intermedia h_i
- La entrada es el vector x_i junto con el estado anterior h_{i-1}
- El output final de la red es el estado final h_t



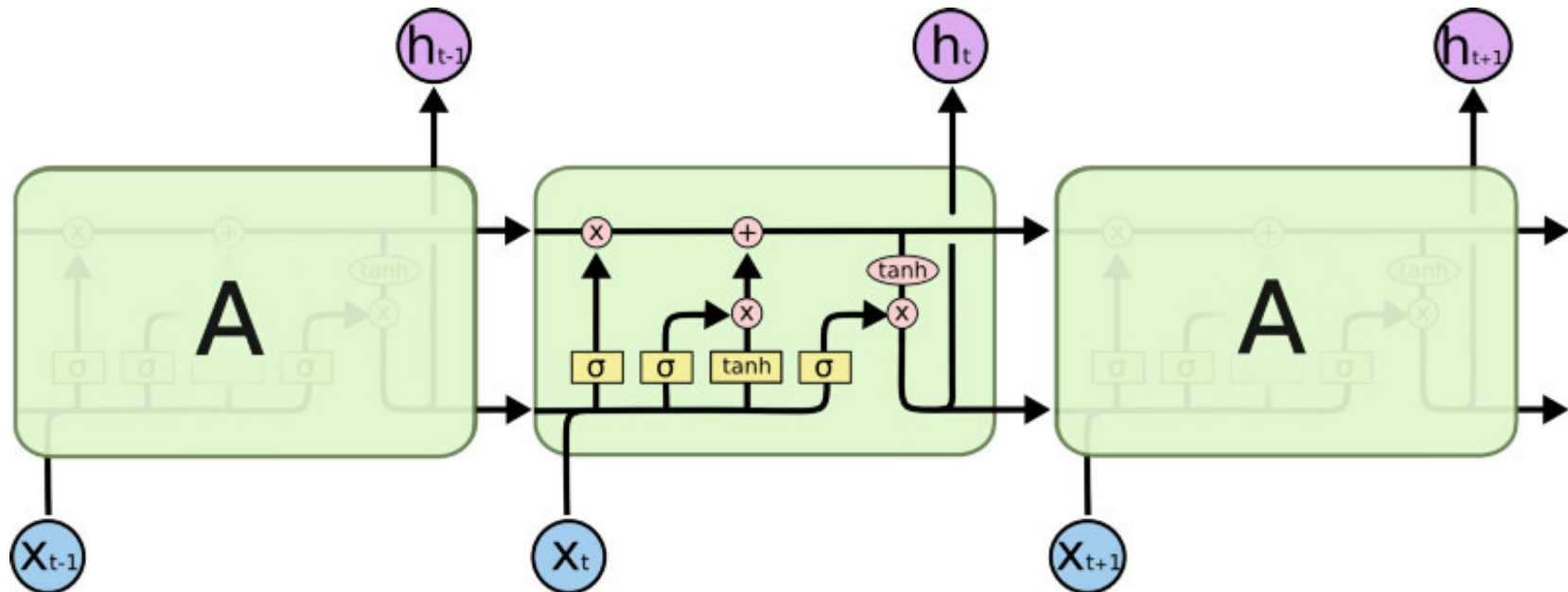
Long-Short Term Memory (LSTM)

- Contiene gates para decidir que valores se leen del estado anterior y se generan a la salida (tanh)



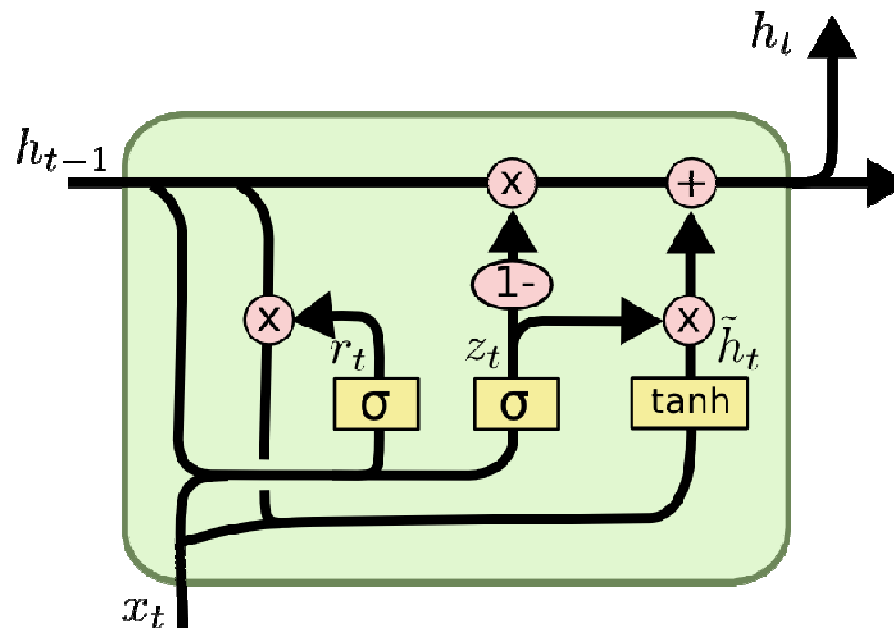
LSTM

- Long-Short Term Memory puede guardar información por periodos largos y cortos gracias a las compuertas para guardar/olvidar



RNN

- GRU (Gated Recurrent Unit)
 - Variante de LSTM con menos gates
 - Es más simple y rápida de entrenar





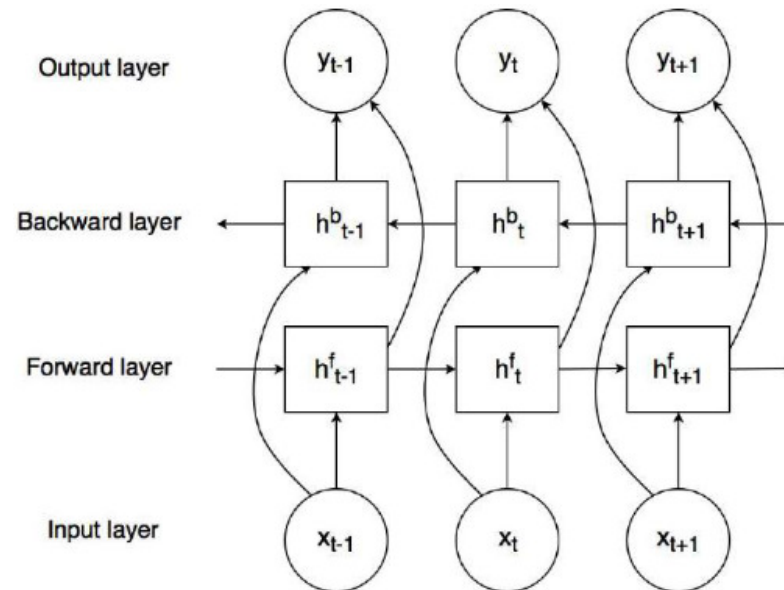
Generación de texto con RNN

- Se debe entrenar con secuencias de largo fijo con cada valor de entrada y su valor de salida correspondiente
 - Por ejemplo, para entrenar una red que genere texto se usa:

Entradas		Salidas
[puedo, escribir, los, versos]		[más]
[escribir, los, versos, más]	→	[tristes]
[los, versos, más, tristes]		[esta]
[versos, más, tristes, esta]		[noche]

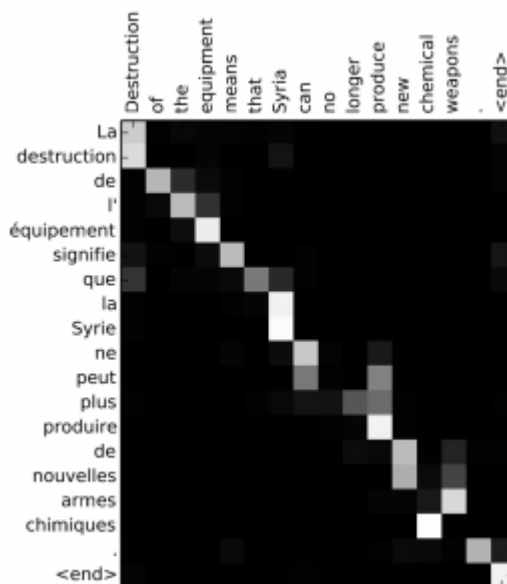
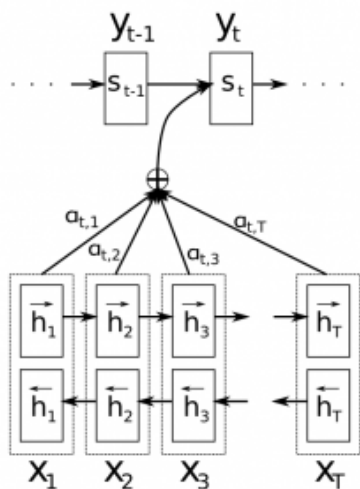
Bi-Direccional

- Para reducir la influencia de los últimos valores en el resultado final
- Se concatenan las salidas de ambas direcciones



RNN con Zona de Atención

- Crear una zona donde se guarda la relación entre inputs y outputs
- Permite encontrar la causa de una decisión



by *ent270* , *ent223* updated 9:35 am et , mon march 2 , 2015
(*ent223*) *ent63* went familial for fall at its fashion show in
ent231 on sunday , dedicating its collection to `` mamma ''
with nary a pair of `` mom jeans '' in sight . *ent164* and *ent21* ,
who are behind the *ent196* brand , sent models down the
runway in decidedly feminine dresses and skirts adorned
with roses , lace and even embroidered doodles by the
designers ' own nieces and nephews . many of the looks
featured saccharine needlework phrases like `` i love you ,
...

X dedicated their fall fashion show to moms

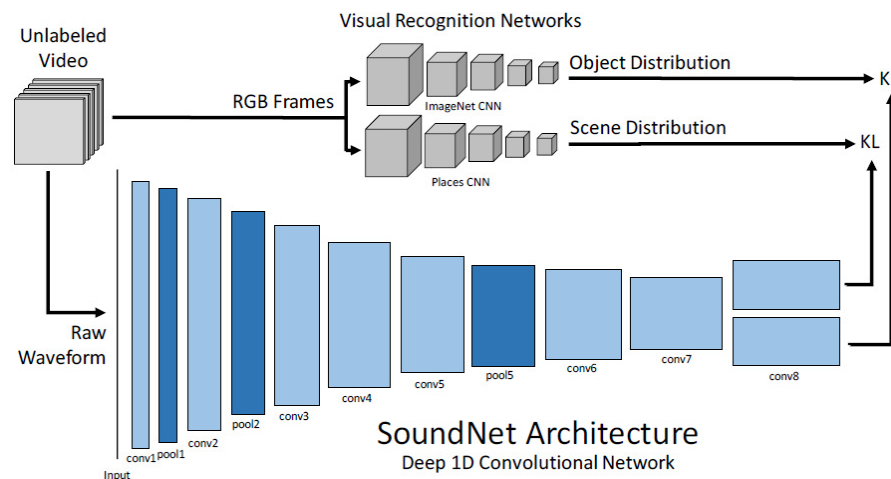


Audio

CNN para Audio

Aytar et al. SoundNet: Learning Sound Representations from Unlabeled Video. 2016
<http://soundnet.csail.mit.edu/>

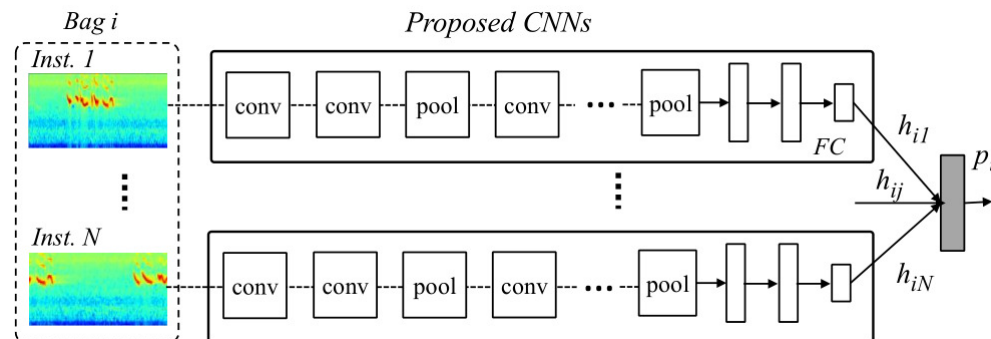
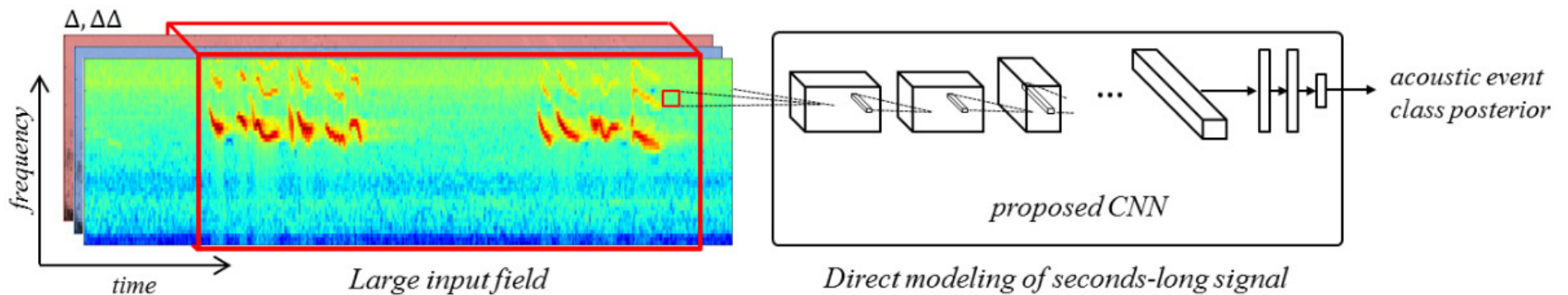
- Entrena un clasificador de audio a partir un clasificador de imagen
 - Asume que lo que es visible es lo que se está escuchando
 - Toma cada frame del video, lo ingresa a una red convolucional pre-entrenada con ImageNet y Places, obtiene la salida y entrena la red de audio para que genere la misma salida.



AENet

Takahashi et al. AENet: Learning Deep Audio Features for Video Analysis. 2017
<https://github.com/znaoya/aenet>

■ Descriptor de audio





Combinación

Combinar Texto con Imágenes

- Dataset COCO tiene ~80 mil imágenes cada una con 5 descripciones (~400 mil descripciones)



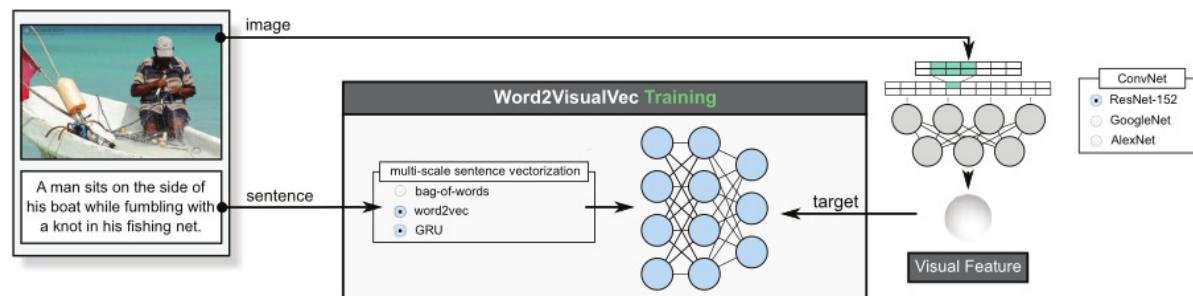
- A white lawn chair laying on top of a sandy beach
- A sun shade sitting out on the beach
- Empty beach chair under an umbrella while different boats are out in the ocean



- A group of people are surfing and swimming in the ocean
- Sunsets over a surfer and other people enjoying the ocean beach
- A child walking and watching a surfer at sunset

Buscar imágenes sin etiquetar

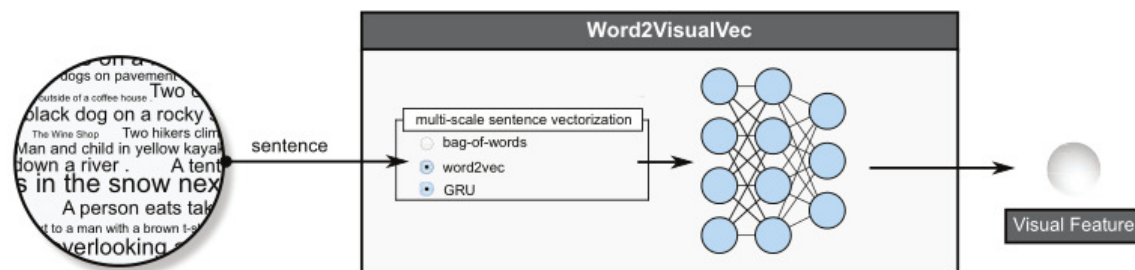
- Con datos de entrenamiento (COCO) calcular un vector visual para cada imagen y un vector textual para su descripción correspondiente
- Entrenar MLP para regresión, con entradas los vectores textuales y salidas los vectores visuales correspondientes
 - Conversión de espacios de características textual a visual (embedding)



Word2VisualVec

Dong, Li, Snoek. Predicting Visual Features from Text for Image and Video Caption Retrieval. 2018
<https://github.com/danieljf24/w2vv>

- Búsqueda de texto libre:
 - Calcular los vectores visuales de las imágenes del dataset
 - Calcular el vector textual de la consulta
 - Usar la red MLP (entrenada con COCO) y obtener su conversión a vector visual
 - Buscar los vectores visuales más cercanos en las imágenes del dataset

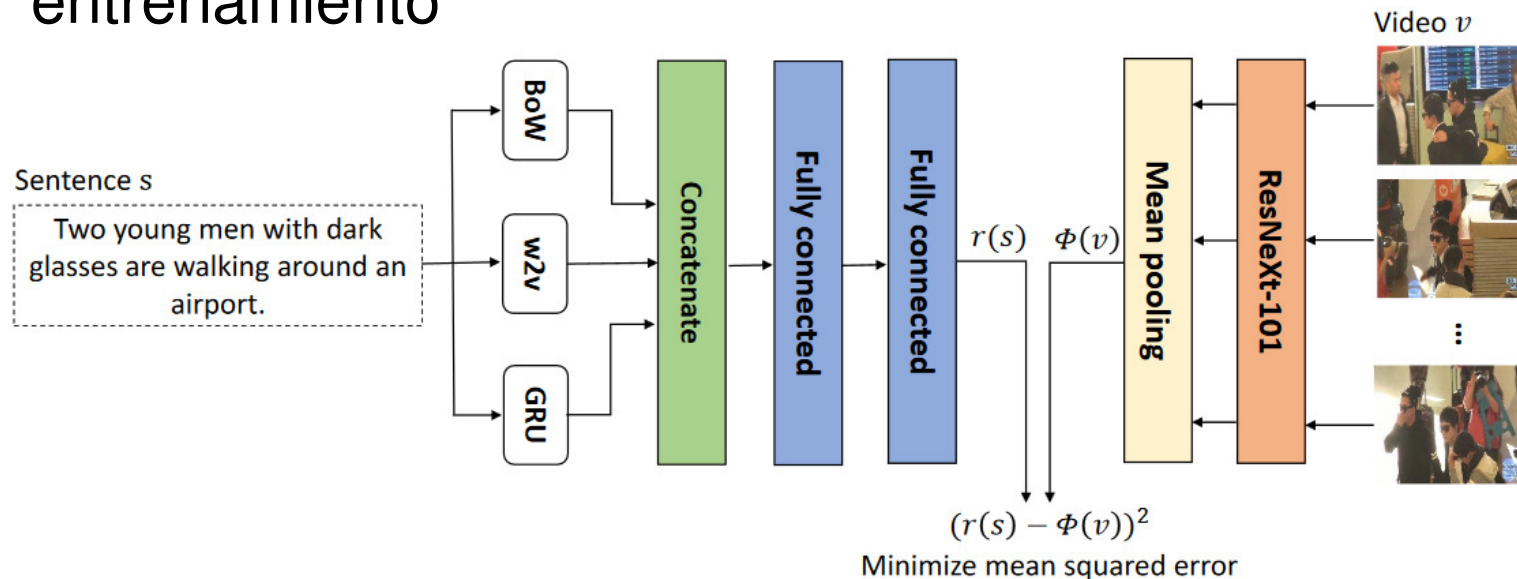


Word2VisualVec++

Li et al. Word2VisualVec++ for Ad-hoc Video Search. 2018

<https://www-nlpir.nist.gov/projects/tvpubs/tv18.slides/rucmm.avs.slides.pdf>

- En vez de convertir un tipo de descriptor en el otro, generar un Espacio Combinado al que ambos espacios se proyectan
- Se requiere una función de distancia que se desea minimizar
- Selector de los mejores pares para mejorar el entrenamiento

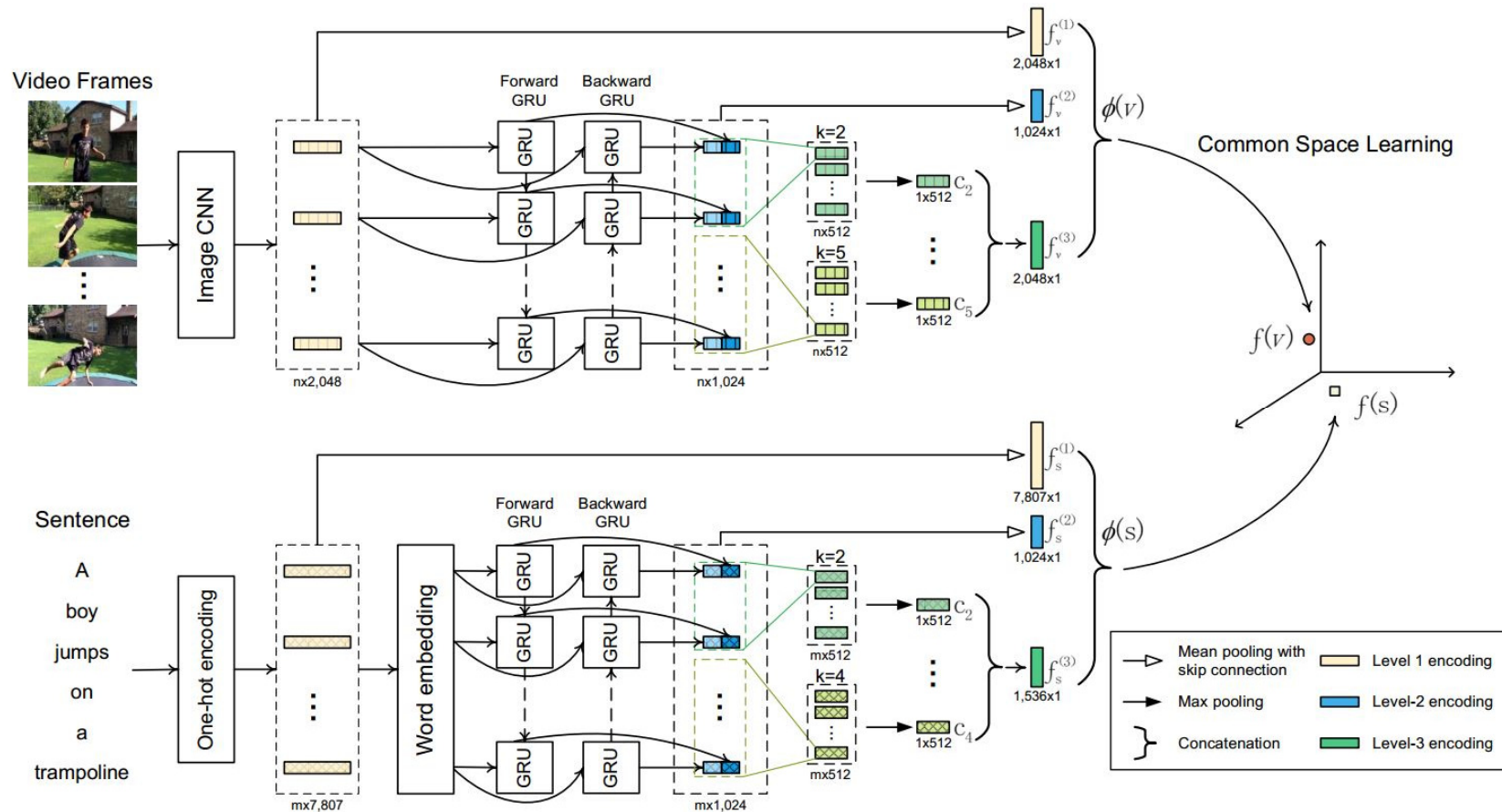


Dual Encoding

Dong et al. Dual Encoding for Zero-Example Video Retrieval. 2019

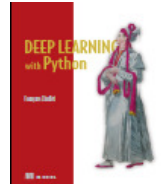
https://github.com/danieljf24/dual_encoding

- Combinar imagen y texto de forma simétrica:

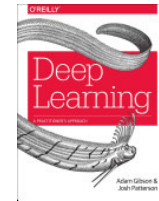


Bibliografía

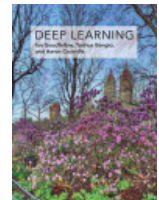
- **Deep Learning with Python.** Chollet. 2018.



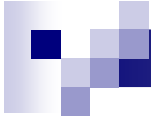
- **Deep Learning: A Practitioner's Approach.** Patterson, Gibson. 2017.



- **Deep Learning.** Goodfellow, Bengio, Courville. 2016.



- Curso de Stanford (<http://cs231n.github.io/>)
 - <http://cs231n.github.io/neural-networks-1/>
 - <http://cs231n.github.io/neural-networks-2/>



Temas para Discusión



Vehículos Autónomos

- Accidentes fatales (5 conductores y 1 peatón a 2019)
 - https://en.wikipedia.org/wiki/List_of_self-driving_car_fatalities
- Ver el historial del accidente del vehículo Uber, en Marzo 2018 en Tempe, Arizona:
 - <https://www.theverge.com/2018/3/28/17174636/uber-self-driving-crash-fatal-arizona-update>
 - En 19-nov-2019 la investigación federal culpa al conductor, al peatón, a Uber (por falta de procedimientos y monitoreo) y al Estado de Arizona (por su poca regulación)
 - En 24-mayo-2018 el reporte del accidente dice:

“At 1.3 seconds before impact, the self-driving system determined that an emergency braking maneuver was needed to mitigate a collision. According to Uber, **emergency braking maneuvers are not enabled while the vehicle is under computer control, to reduce the potential for erratic vehicle behavior.** The vehicle operator is relied on to intervene and take action.”



Altas Expectativas de la IA

- IBM has a Watson dilemma
 - <https://www.wsj.com/articles/ibm-bet-billions-that-watson-could-improve-cancer-treatment-it-hasnt-worked-1533961147>
- Layoffs at Watson health reveal IBM's problem with AI
 - <https://spectrum.ieee.org/the-human-os/robotics/artificial-intelligence/layoffs-at-watson-health-reveal-ibms-problem-with-ai>
- Why everyone is hating on IBM Watson - Including the people who helped make it
 - <https://gizmodo.com/why-everyone-is-hating-on-watson-including-the-people-w-1797510888>
- IBM Watson reportedly recommended Cancer treatments that were 'unsafe and incorrect'
 - <https://gizmodo.com/ibm-watson-reportedly-recommended-cancer-treatments-tha-1827868882>



Otros Casos

- El caso de motores Diesel de Volkswagen
 - [https://es.wikipedia.org/w/index.php?title=Esc%C3%A1ndalo de emisiones contaminantes de veh%C3%ADculos Volkswagen&oldid=120616880](https://es.wikipedia.org/w/index.php?title=Esc%C3%A1ndalo_de_emisiones_contaminantes_de_veh%C3%ADculos_Volkswagen&oldid=120616880)
- Sesgos en rostros. Amazon's face recognition misidentifies 28 members of Congress as suspected criminals:
 - <https://gizmodo.com/amazons-face-recognition-misidentifies-28-members-of-co-1827887567>
- Columna de opinión: Has AI surpassed humans at translation? Not even close!
 - https://www.skynettoday.com/editorials/state_of_nmt
- Columna de opinión: AI winter is well on its way
 - <https://blog.piekniowski.info/2018/05/28/ai-winter-is-well-on-its-way/>
- Carta de investigadores: Research Priorities for Robust and Beneficial Artificial Intelligence
 - http://futureoflife.org/misc/open_letter

Científicos piden que el desarrollo de la inteligencia artificial beneficie a la humanidad

Una agrupación de más de 700 expertos de todo el mundo firmó una carta donde se solicita que todo desarrollo debe incluir un estudio sobre cómo se beneficia a la sociedad.

AFP

Lunes, 12 de Enero de 2015, 17:49

Stephen Hawking advierte que la inteligencia artificial puede traer "el fin de la raza humana"

Los robots "podrían llegar a tomar el control y se podrían rediseñar a sí mismos" para desbancar a los humanos, afirmó el físico británico.

EFE

Martes, 2 de Diciembre de 2014, 14:38

11



Expertos en inteligencia artificial hacen llamado para frenar desarrollo de armas autónomas

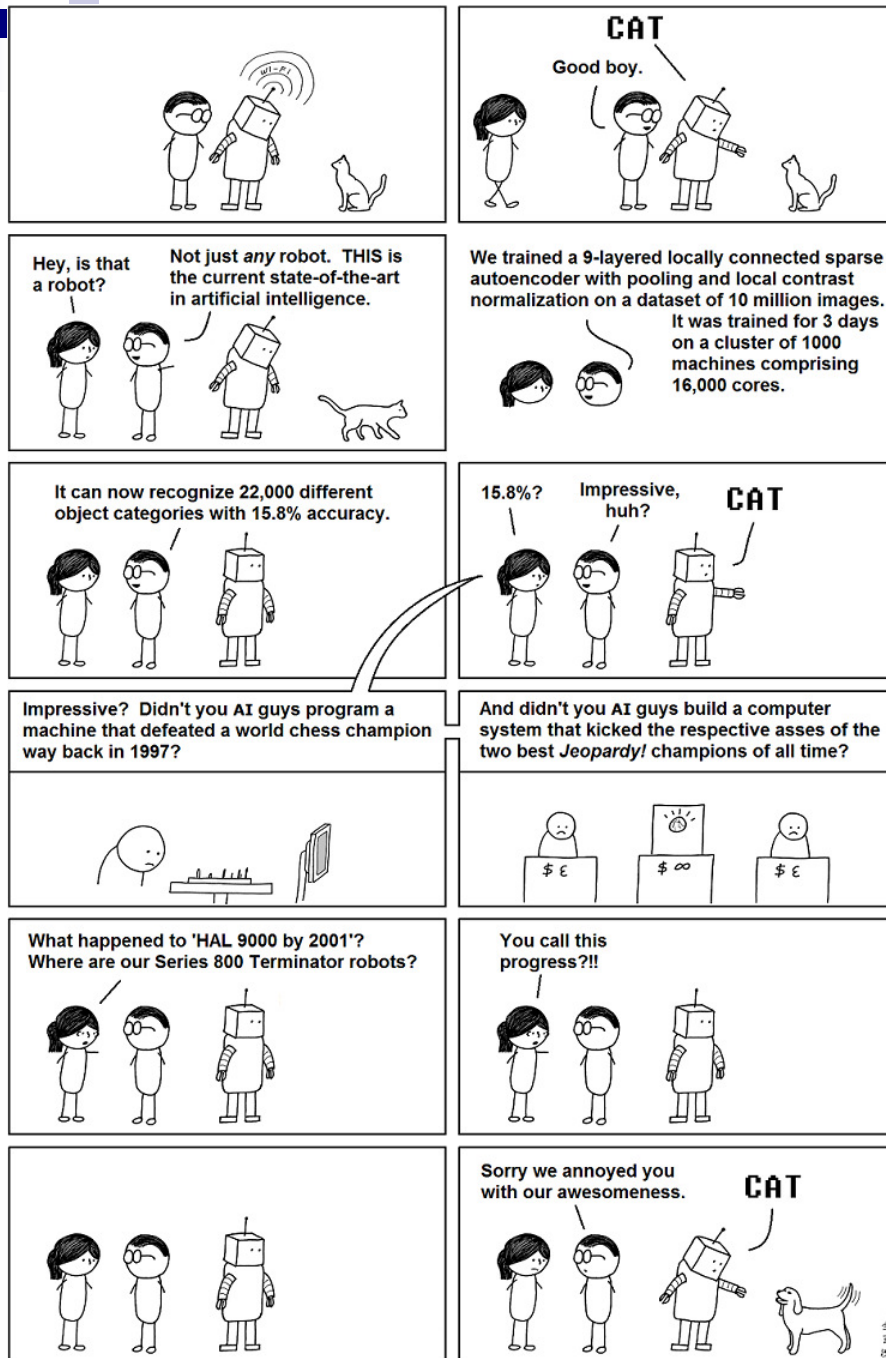
El científico Stephen Hawking y el cofundador de Apple Steve Wozniak, están en la lista de quienes firmaron una carta donde advierten sobre los peligros de este tipo de tecnología bélica.

Emol

lunes, 27 de julio de 2015 11:59

WASHINGTON.- Más de 700 científicos de todo el mundo, entre ellos el físico británico Stephen Hawking y el fundador de SpaceX Elon Musk, pidieron este fin de semana que los avances en materia de inteligencia artificial sirvan para beneficiar a la humanidad.

"Los progresos realizados en inteligencia artificial son una buena ocasión para concentrar nuestras investigaciones en aquellos trabajos que no sólo hacen de las tecnologías herramientas cada vez más poderosas, sino además más beneficiosas para la sociedad", escribieron los expertos en una carta abierta.



<http://abstrusegoose.com/496>

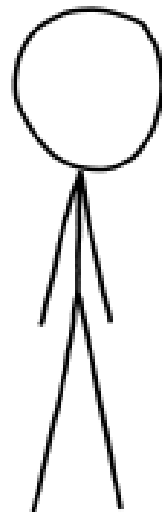
Hace referencia a:
http://research.google.com/archive/unsupervised_icml2012.html

WHEN A USER TAKES A PHOTO,
THE APP SHOULD CHECK WHETHER
THEY'RE IN A NATIONAL PARK...

SURE, EASY GIS LOOKUP.
GIMME A FEW HOURS.

... AND CHECK WHETHER
THE PHOTO IS OF A BIRD.

I'LL NEED A RESEARCH
TEAM AND FIVE YEARS.



IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

<http://xkcd.com/1425/>



<http://xkcd.com/1444/>

