Cursos Extraordinarios

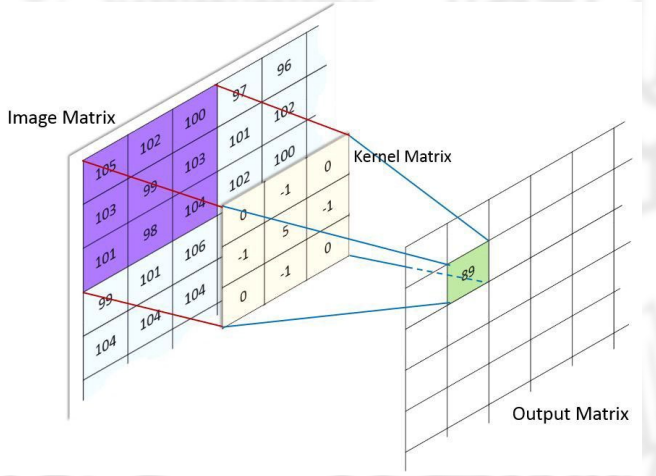
verano 2025

“Inteligencia Artificial y Grandes Modelos de Lenguaje: Funcionamiento, Componentes Clave y Aplicaciones”

Zaragoza, del 30 de junio al 02 de julio de 2025

Tipos de Redes: Arquitectura

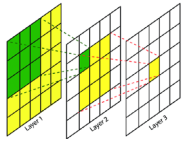
- **Redes Convolucionales (CNN)**
  - Reducción del número de grados de libertad (parámetros).
  - Reutilización de los pesos
    - Cada unidad procesa una región diferente de la entrada.
    - Todas comparten los pesos
    - Se usa conocimiento a priori para definir la topología
  - Suelen incorporan capas de pooling
    - Reducen la dimensionalidad
    - Max-pooling, avg-pooling, …



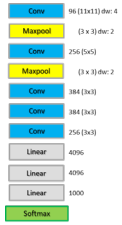
Tipos de Redes: Arquitectura

- **Redes Convolucionales CNN**

Trabajos pioneros

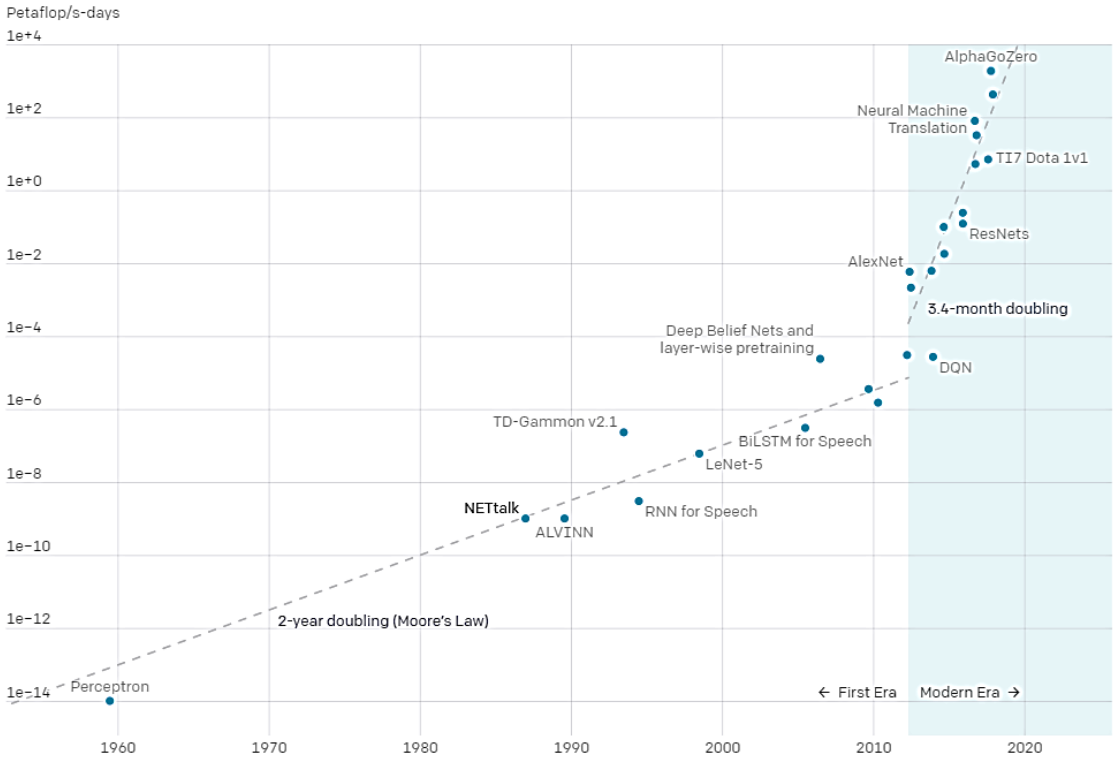


***1998**: 4 capas*



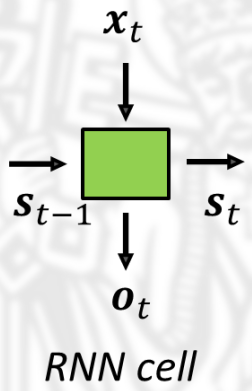
***2012**: 7 capas*
84.6 % aciertos

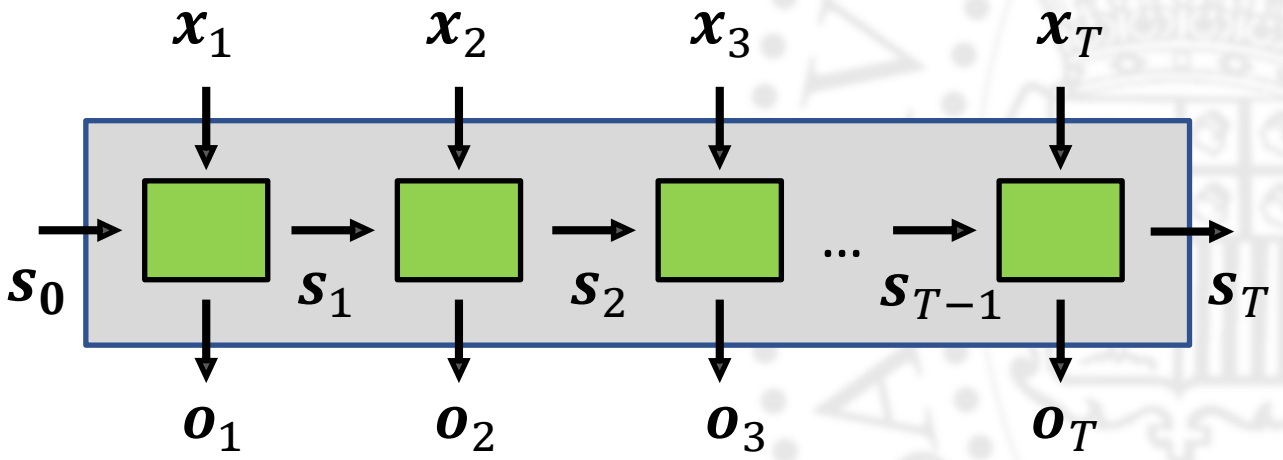
***2015**: Resnet >100 capas*
96.43% aciertos



Tipos de Redes: Arquitectura

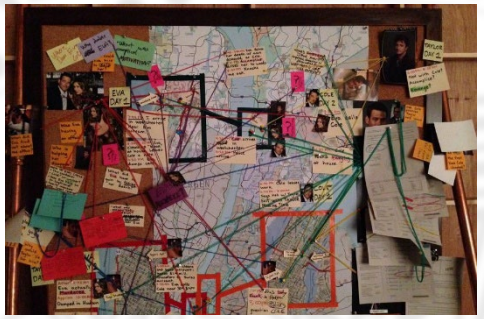
- **Redes Recurrentes:**
  - Analizan la entrada en orden (sentido temporal, orden del texto)
  - Cada celda tiene una memoria finita para recibir información de los
  - Instantes previos y escribir nueva información para el futuro
  - Entrenamiento Backpropagation-Through-Time (BTT)

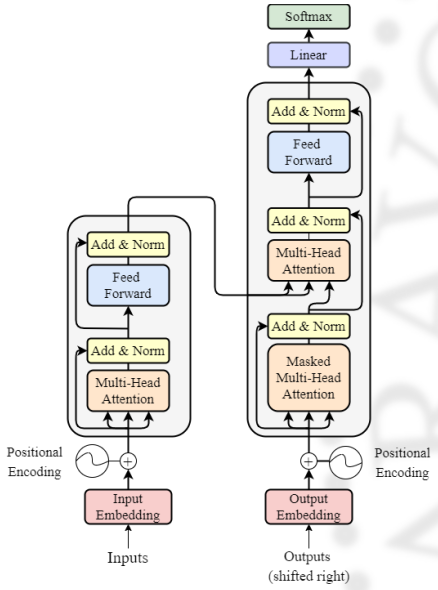




Tipos de Redes: Arquitectura

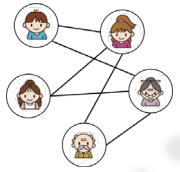
- **Transformers:**
  - Estructuras muy expresivas, eficientes y diferenciables
  - Capaces de analizar las relaciones entre todas las entradas y entre las entradas y las salidas: Mecanismos de Atención.

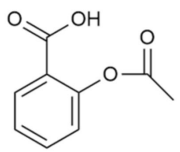


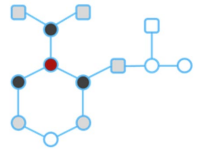


Tipos de Redes: Arquitectura

- **Redes aplicadas a grafos:**
  - Redes sociales
  - Química
  - Conceptos y mapas de palabras
  - Sistemas de recomendación
  - Objetos 3d
  - Mapas, redes de Comunicaciones, infraestrucutras
  - Tráfico, Transporte, Telemática







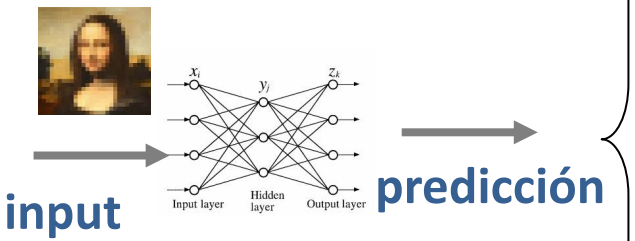
In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation
William J. Cook
Attention, learn to solve routing problems

A principled framework for diversified complementary product recommendation. Amazon 2020

https://eng.uber.com/uber-eats-graph-learning/

Tipos de Redes: entrenamiento

- **Aprendizaje supervisado:**
  - Ajuste de una función de coste mediante etiquetas

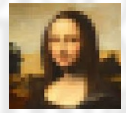


Clasificación (global): Mona Lisa

Clasificación (múltiple):

Regresión (global): Age ? -> xx years

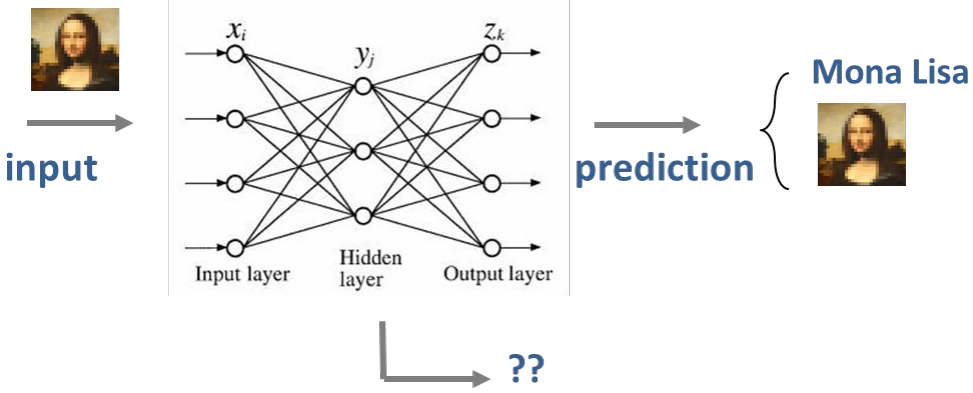
Regresión (múltiple):

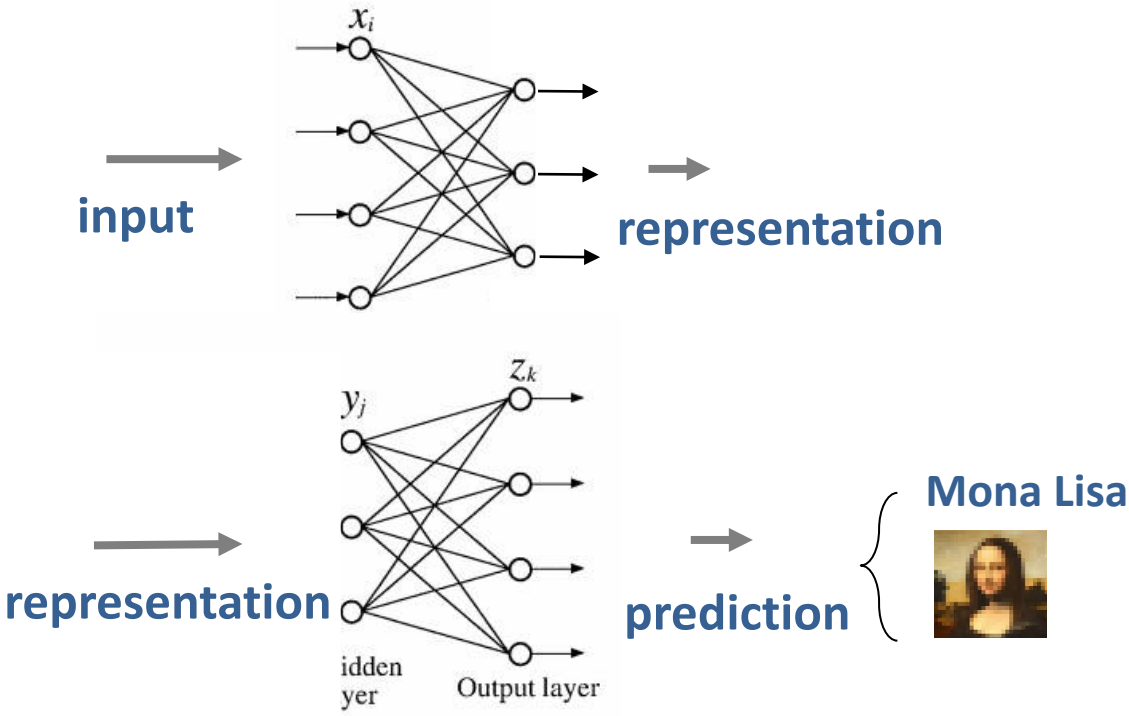


B&W to RGB
super-resolution

Tipos de Redes: entrenamiento

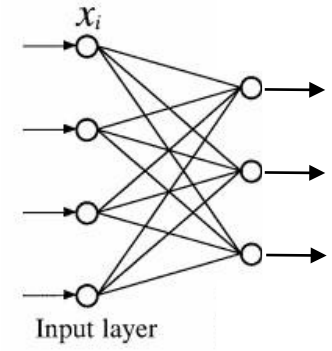
- **Representaciones:**

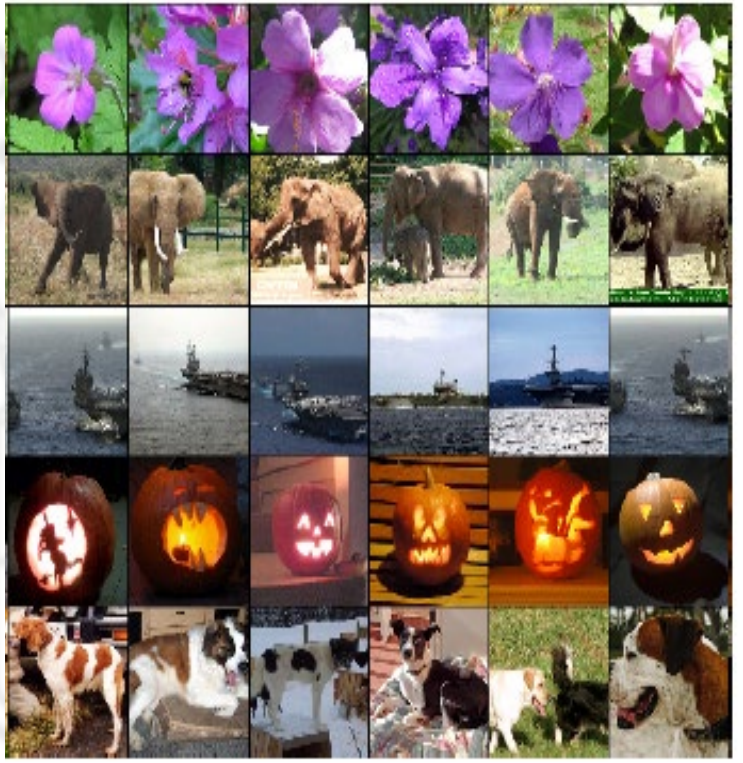




Tipos de Redes: entrenamiento

- **Representaciones:**





query

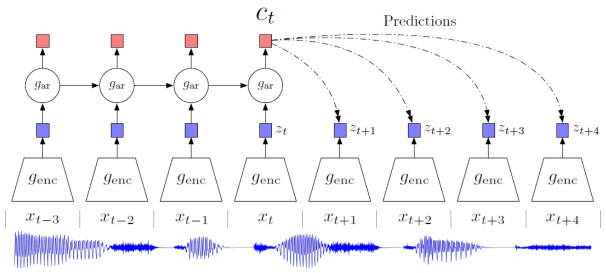
Tipos de Redes: entrenamiento

- **Aprendizaje no supervisado :**
  - Las redes aprenden la naturaleza de los datos
    - Podemos conseguir que los sistemas automáticos comprendan los datos forzando a que hagan predicciones sobre lo que no han visto

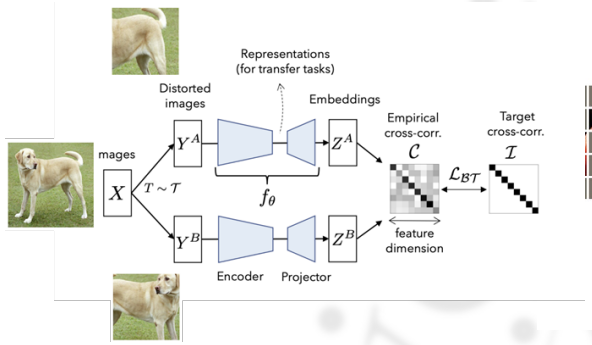
A Oord, N Kalchbrenner, O Vinyals, L Espeholt, A Graves, K Kavukcuoglu
Conditional Image Generation with PixelCNN Decoders 2016

Tipos de Redes

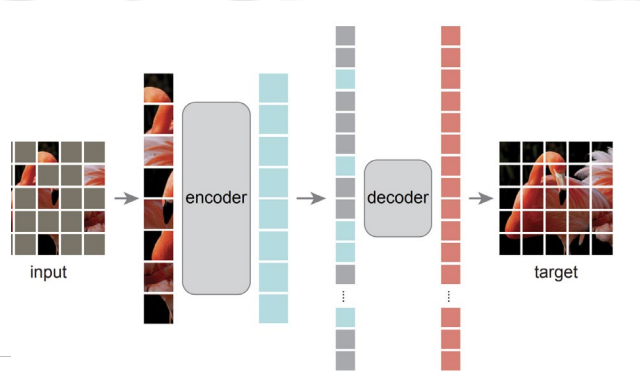
- **Self-supervised learning:**
  - Estrategias:



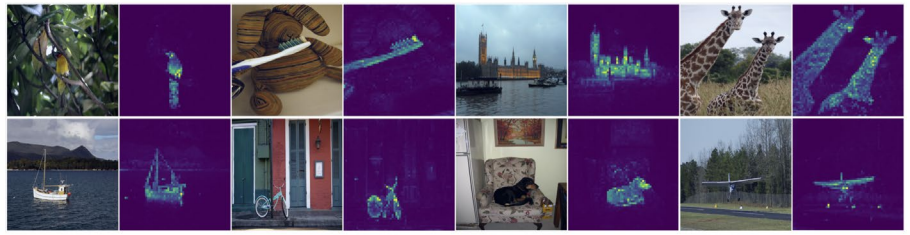
Una estrategia es dar varias opciones
como si fuera un examen

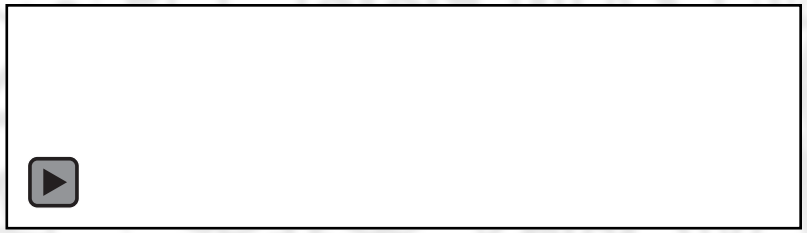


Resolver la pregunta:
¿son partes de la misma imagen ?



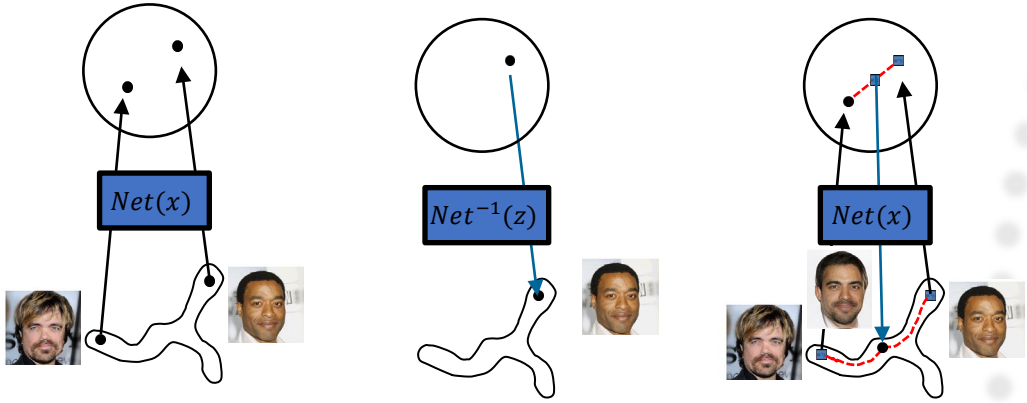
Reconstruir la imagen a partir de una con oclusiones

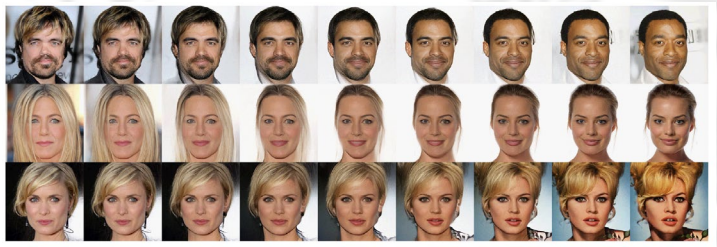




Aplicaciones

- **Generación de Contenidos:**
  - Síntesis de voz, música, imágenes, …
    - **Podemos aprender a manipular las imágenes**
      - ¿Qué pasa si cambio la representación para conseguir otra imagen distinta



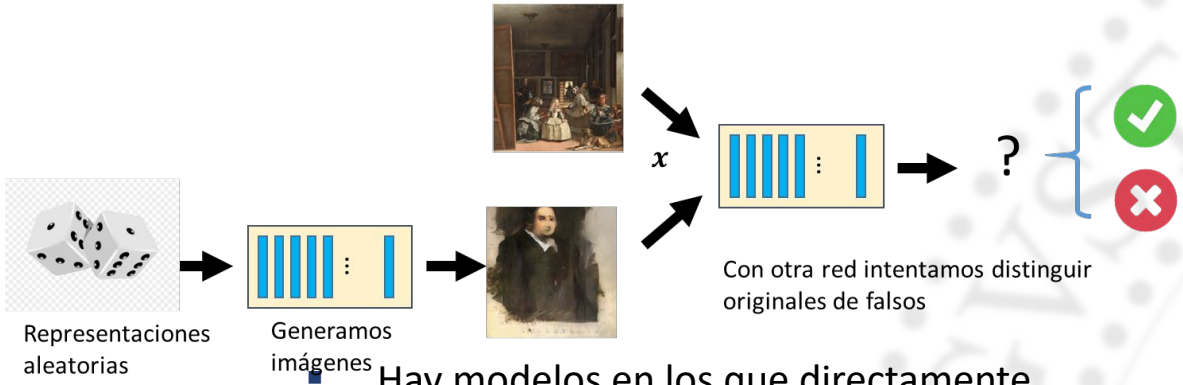


Generación de nuevas imágenes que nunca han existido

Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In Advances in neural information processing systems (pp. 10215-10224). Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In Advances in neural information processing systems (pp. 10215-10224).

Aplicaciones

- **Generación de Contenidos:**
  - Síntesis de voz, música, imágenes, …



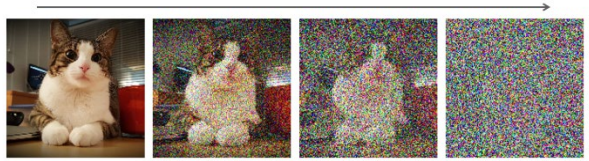
- Hay modelos en los que directamente se aprende a generar imágenes
  - **Generative adversarial network**

Este proceso se ha sofisticado mucho en menos de 10 años

Aplicaciones

- **Generación de Contenidos:**
  - Sintesis de voz, música, imágenes, …

Podemos añadir ruido hasta que no se reconozca la imagen



Con una red aprendemos a “limpiar” ese ruido

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. arXiv preprint arXiv:2006.11239

Son capaces de generar imágenes tan realistas como los GANs

the angel of air. unreal engine @arankomatsuzaki

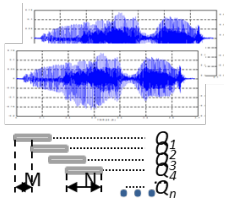
treehouse in the style of studio ghibli animation @danielrussruss

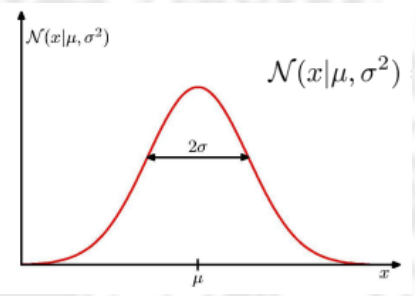
A wooden spanish laptop of 1650 found the library of El Escorial

Medieval 1230 book page illustrating monks playing basketball

Aplicaciones

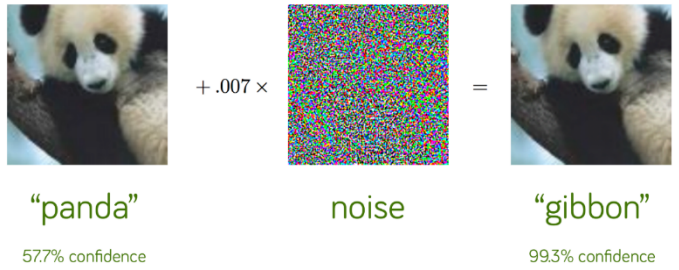
- **Redes aplicadas al modelado estadístico**
  - Tratan de definir una probabilidad o una verosimilitud con respecto a la población que se ha usado para entrenarlas
  - Aplicaciones:
    - Generar nuevos datos
    - Deteccion de ejemplos fuera de distribución
      - Fraude
      - Anomalías



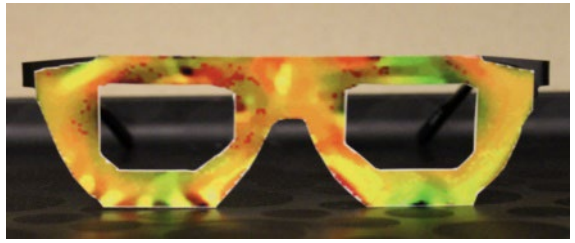


Robustez

- **Adversarial attacks**
  - Un pequeño cambio en la imagen puede hacer que cambie la predicción



Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.



Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition
Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter
ACM Conference on Computer and Communications Security (CCS 2016)

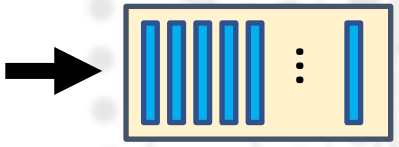
Speed Limit 80
(88% confidence)

Robust physical-world attacks on deep learning visual classification.
Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., ... & Song, D. (2018). In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1625-1634).

Augmentation

- **Data augmentation**
  - Incrementar artificialmente el dataset mediante datos transformados
    - Dependen de la tare:
      - Imagen: rotaciones, oclusiones, escalado, recortes, transformaciones
      - Audio: ruido, reverberación
  - El sistema se vuelve más robusto
    - Se expone a más variabilidad durante el entrenamiento
    - También se puede hacer al predecir (TTA test time augmentations)

paperswithcode.com



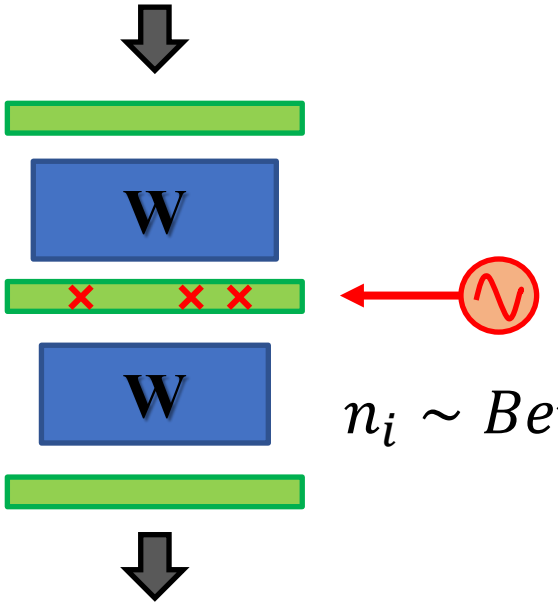
Train/Predict

Dropout

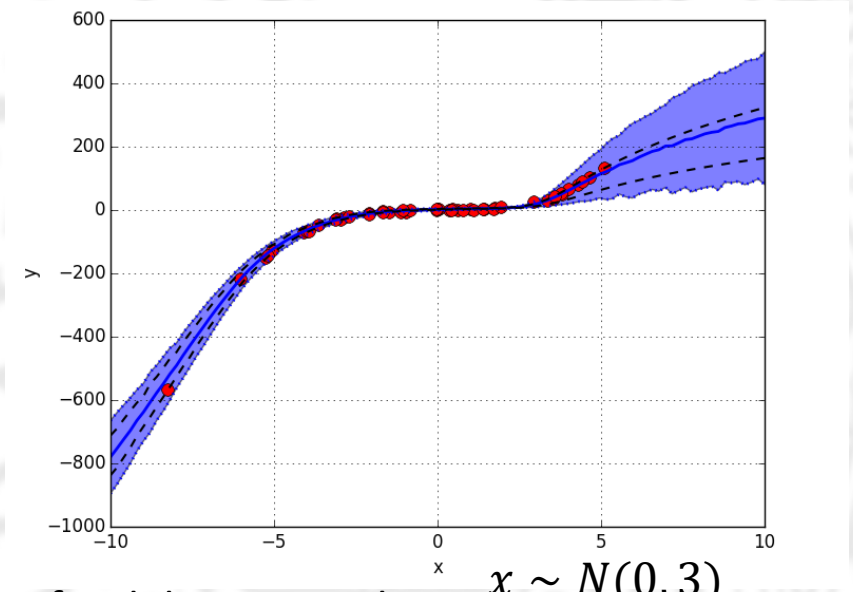
- **Dropout (Hinton 2012)**

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp.1929-1958.

  - Desactiva aleatoriamente algunas neuronas durante el entrenamiento
  - Se puede aplicar también en predicción (TTA)



$$n_i \sim Berno(1-p)$$



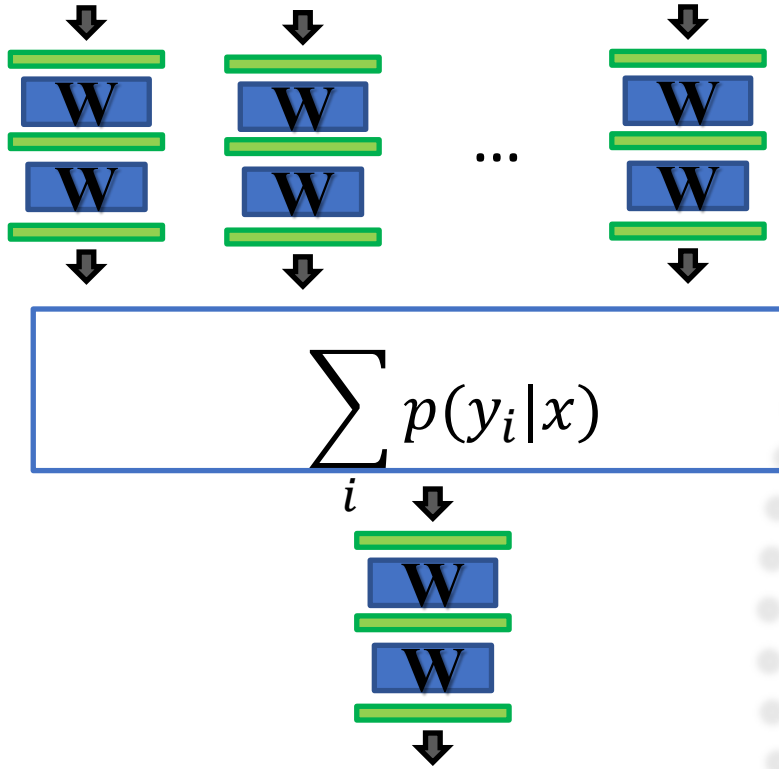
Artificial data example:
1D regression

$$x \sim N(0,3)$$
$$\varepsilon \sim N(0,3)$$
$$y = x^3 + \varepsilon$$

Bayesian dark knowledge

- **Bayesian dark knowledge(Korattikara 2015)**

Korattikara Balan, A., Rathod, V., Murphy, K. P., & Welling, M. (2015). Bayesian dark knowledge. Advances in Neural Information Processing Systems, 28, 3438-3446..



Conjunto de redes profesor

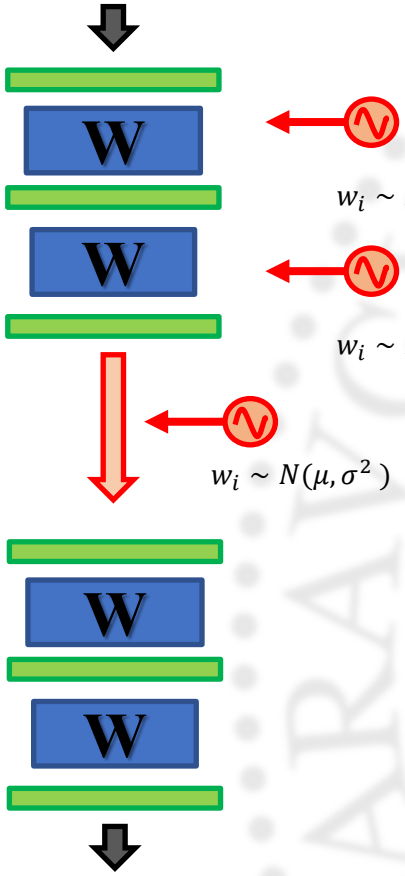
La red estudiante aprende del conjunto

Bayesian dark knowledge

- **Bayesian dark knowledge(Korattikara 2015)**

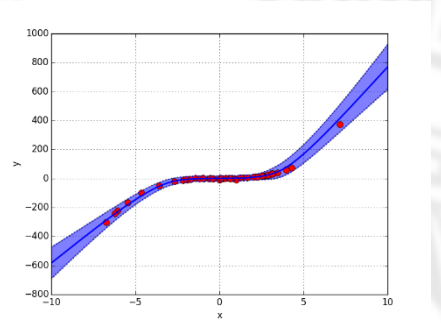
Korattikara Balan, A., Rathod, V., Murphy, K. P., & Welling, M. (2015). Bayesian dark knowledge. Advances in Neural Information Processing Systems, 28, 3438-3446..

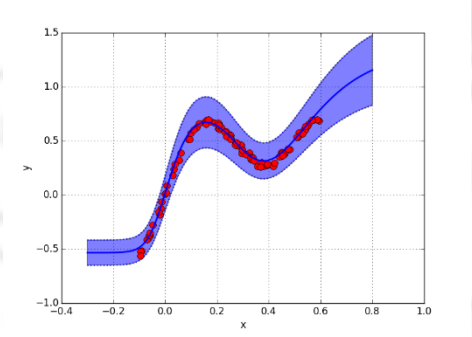
Teacher network,
muestrea



Student network,
Aprende la variabilidad

Examples regression





Ejemplos

https://colab.research.google.com/drive/1PlpIrls6XKIQYfafVkAJpysV5dzmJim7?usp=sharing

https://colab.research.google.com/drive/1sgmP9FLzK3uUncigViUR74o3caTHB5k2?usp=sharing

https://colab.research.google.com/drive/1t_bpo-lPkgP1YULkOScducSQ_f7cLlhN?usp=sharing

https://colab.research.google.com/drive/1t9ZUHjUphMhwBSbTWoPWpJWxDteWjT51?usp=sharing

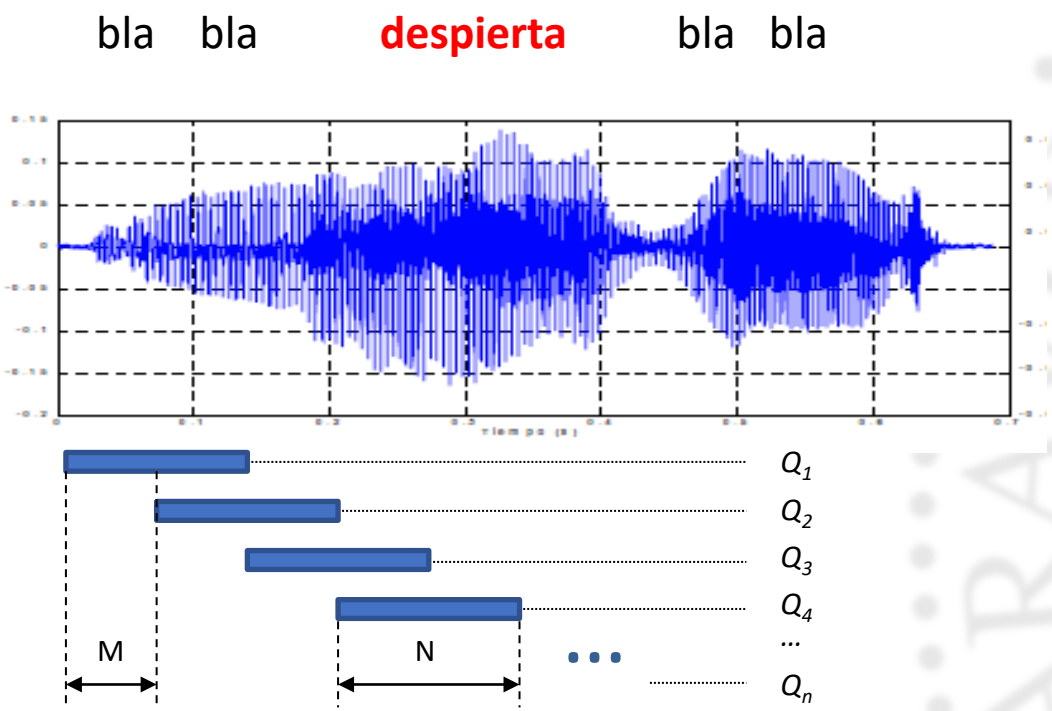
https://colab.research.google.com/drive/15GtY0NaKFpanjXDQFKXz_AaTE0022oN-?usp=sharing

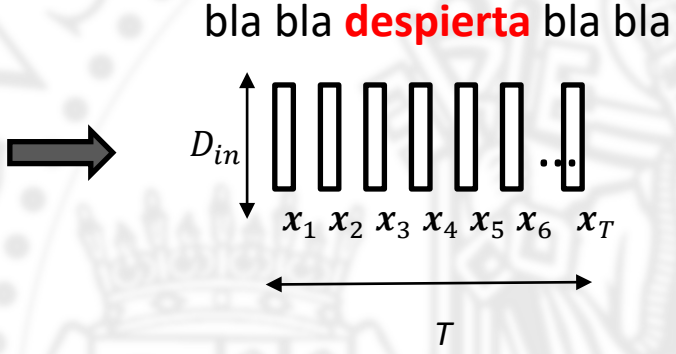
https://colab.research.google.com/drive/1qsNusOm_mzagzytwadZOeFlk6eejg4_j?usp=sharing

https://colab.research.google.com/drive/1B4nsB5vCr5jBP7cHerc-G5V3DmpvE_B-?usp=sharing

Redes convolucionales

- **Ejemplo: análisis de audio (o texto)**
  - Queremos procesar una secuencia de vectores





Secuencia de vectores

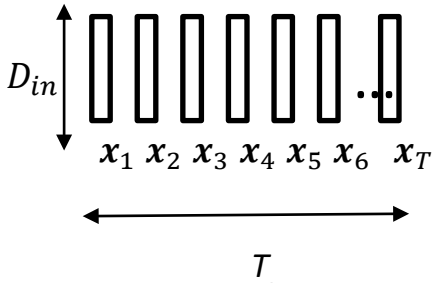
- T vectores de dimensión D_{in}

Input $(B \times T \times D_{in})$
B: batch size
T: seq. Length
D_{in}: feature dim.

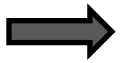
Redes convolucionales

- **Ejemplo:**
  - ¿Podemos aplicar un MLP?

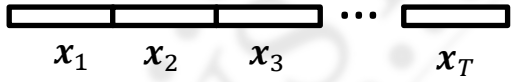
bla bla **despierta** bla bla

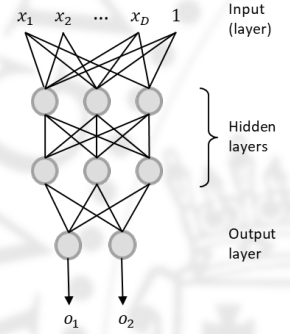


Secuencia de vectores



bla bla **despierta** bla bla





Podríamos concatenar todos los vectores

- Esto podría funcionar si T es fijo

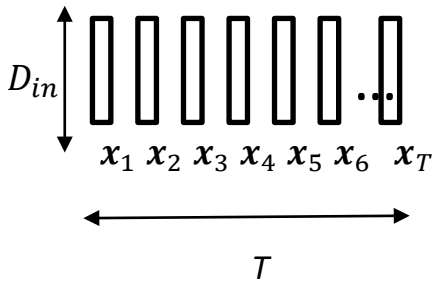
Problemas

- ¿Qué pasa si T es variable?
- ¿Qué pasa si desplazamos la secuencia?

Redes convolucionales

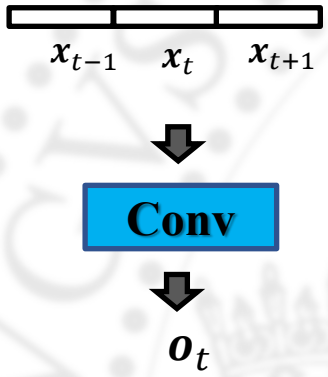
- **Ejemplo:**
  - ¿Podemos aplicar procesado deslizante?

bla bla **despierta** bla bla



Secuencia de vectores

bla bla **despierta** bla bla

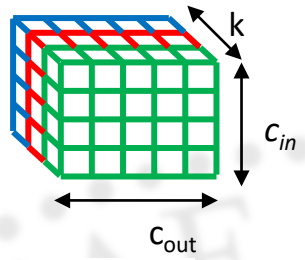


Definimos un contexto (**kernel_size**) sobre el que vamos a calcular, por ejemplo k=3 y una operación lineal:

$$o_t = W_0 x_{t-1} + W_1 x_t + W_2 x_{t+1}$$

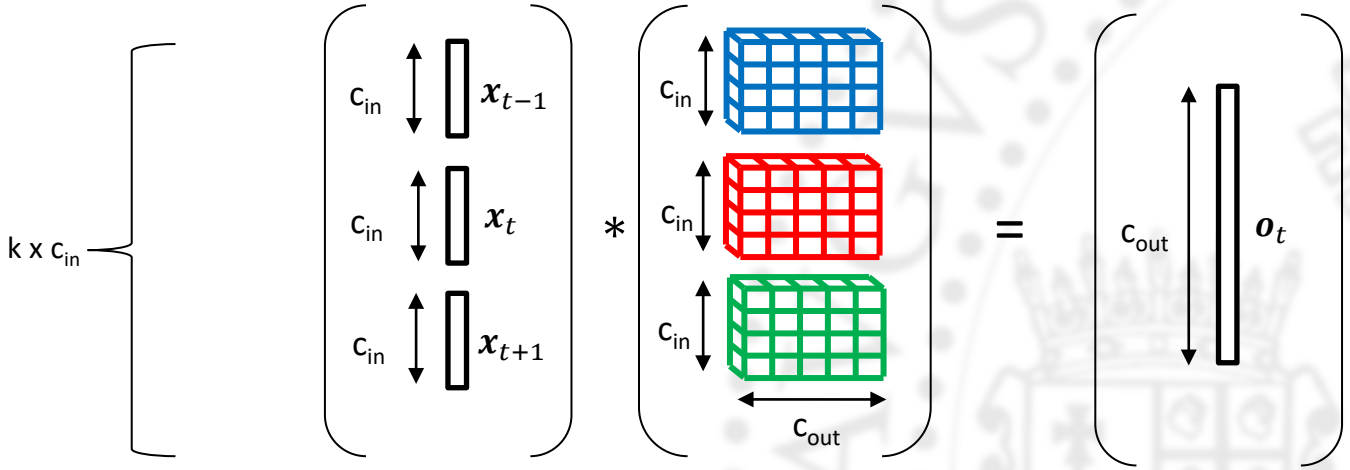
- las matrices W_i (c_in x c_out) se reutilizan en cada posición t
- Las dimensiones de entrada y salida se denominan **canales**

Redes convolucionales



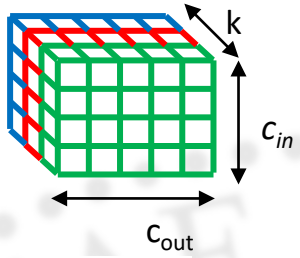
W $(c_{in} \times c_{out} \times k)$

- **Convolución 1D (DNNs)**
  - Interpretación matricial



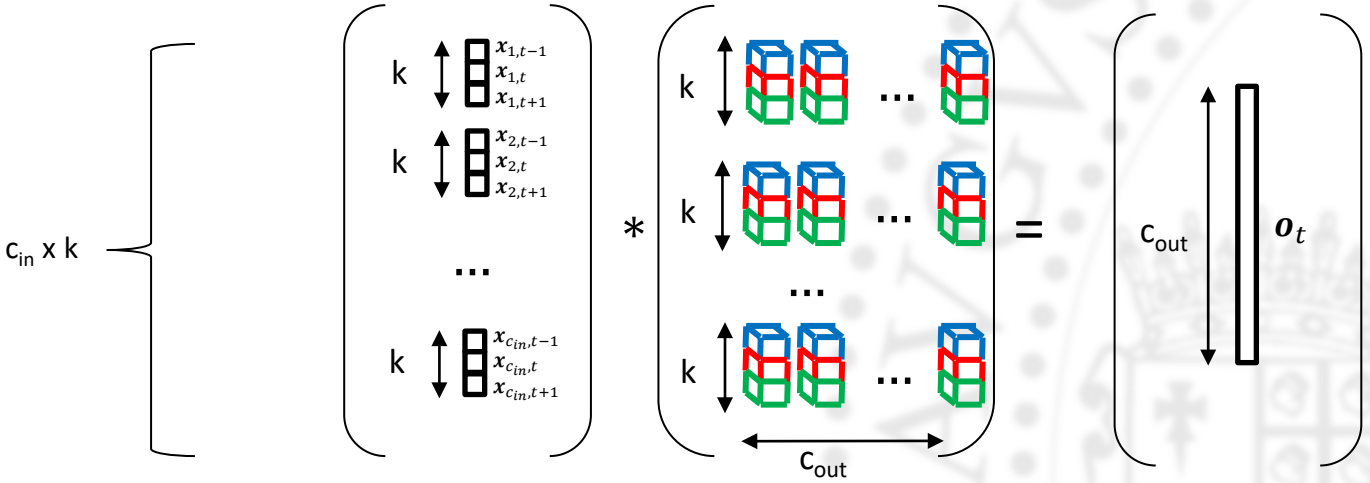
W $(c_{in} \times c_{out} \times k)$

Redes convolucionales



W $(c_{in} \times c_{out} \times k)$

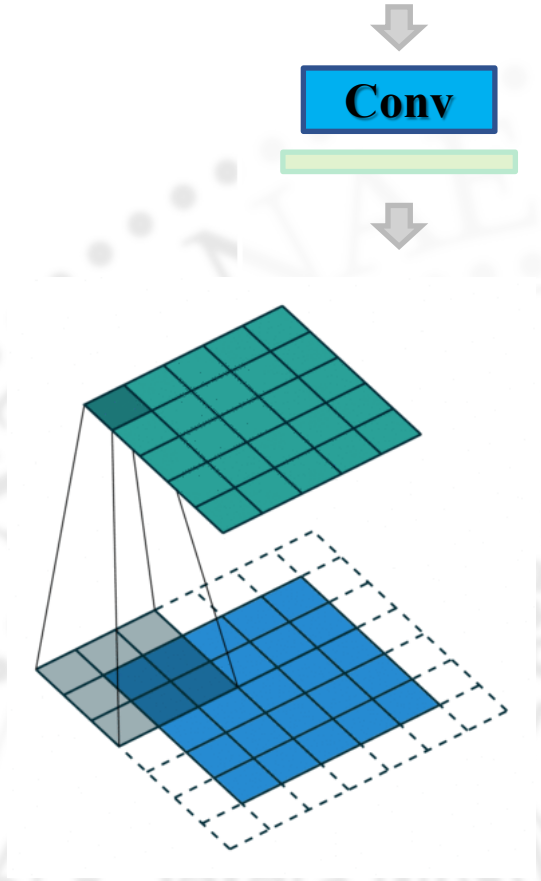
- **Convolución 1D (DNNs)**
  - Interpretación convoluciones:
    - Todos los canales de entrada x todos los canales de salida



W $(c_{in} \times c_{out} \times k)$

Redes convolucionales

- **Convolución 1D/2D/3D (CNNs),**
  - Ejemplo pytorch:
    - Convolution2d ejemplo:
      - 4 input channels (dimensión entrada)
      - 5 output channels (dimensión salida)
      - 3 kernel size
      - 1 stride (sin muestreo)
      - 1 padding (mismo tamaño de salida)



A guide to convolution arithmetic for deep learning
Vincent Dumoulin 2016 https://arxiv.org/abs/1603.07285

pytorch

```
import torch
layer = torch.nn.Conv1d(4, 5, 3, padding=1)
x = torch.randn(32, 4, 100)
print(x.shape)
o = layer(x)
print(o.shape)
```

TODO ejemplo conv2d filtros

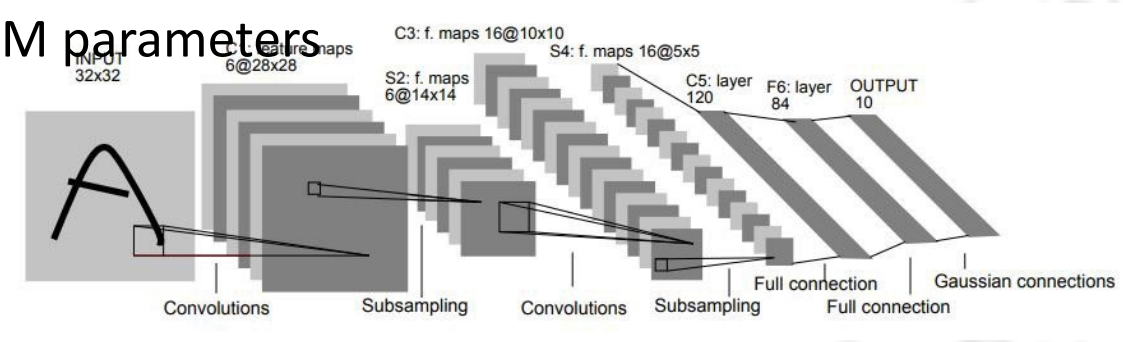
Redes convolucionales

- **LeNet5 (Yann Lecun, 1998)**

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

  - Modelo de referencia (redes convolucionales)

- **Resumen:**
  - 2 capas conv2d
  - 2 capas Pooling (diezmado)
  - Flatten/Reshape para convertir a vector
  - 2 Dense layers
  - 1.6M parameters



https://engmrk.com/lenet-5-a-classic-cnn-architecture/

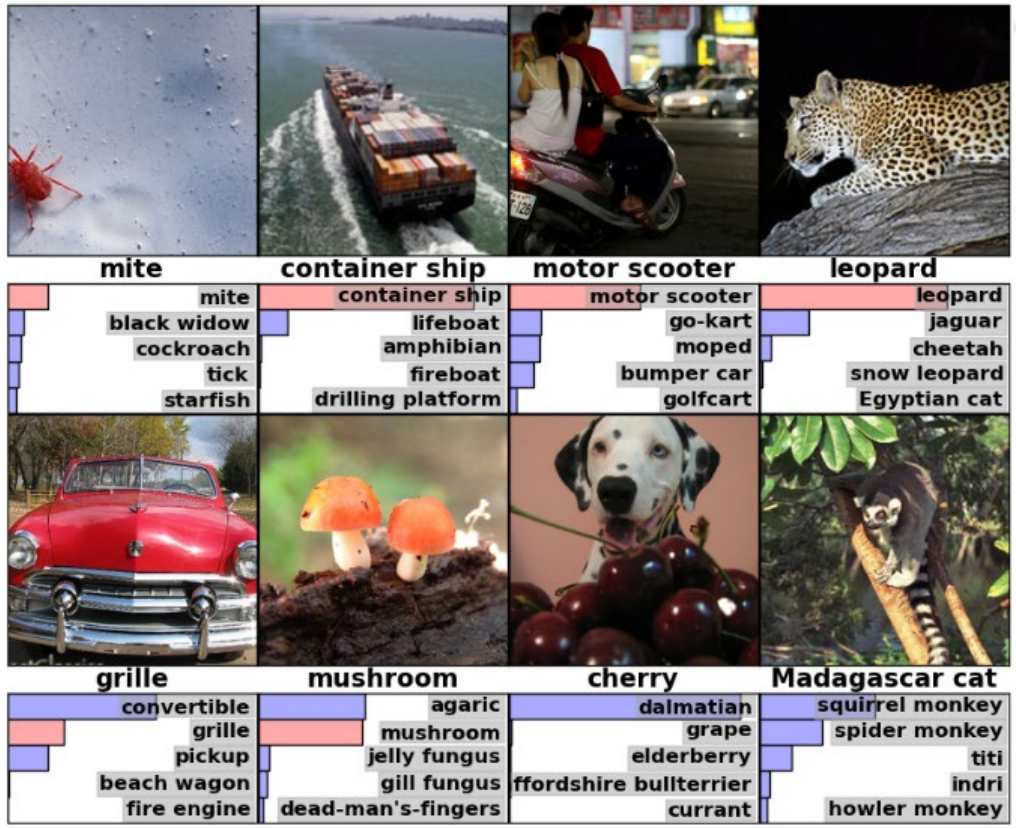
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 32, 32, 20)        1520
_________________________________________________________________
activation (Activation)      (None, 32, 32, 20)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 16, 16, 20)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 16, 16, 50)        25050
_________________________________________________________________
activation_1 (Activation)    (None, 16, 16, 50)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 50)          0
_________________________________________________________________
flatten (Flatten)            (None, 3200)              0
_________________________________________________________________
dense (Dense)                (None, 500)               1600500
_________________________________________________________________
activation_2 (Activation)    (None, 500)               0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               50100
_________________________________________________________________
activation_3 (Activation)    (None, 100)               0
=================================================================
Total params: 1,677,170
Trainable params: 1,677,170
Non-trainable params: 0
_________________________________________________________________
```

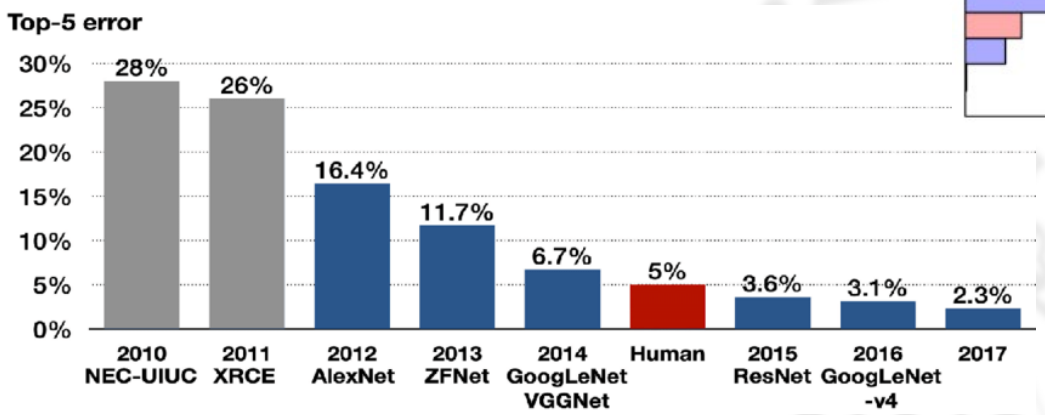
https://medium.com/ai-in-plain-english/lenet-5-the-22-year-old-neural-network-ce4af257ecd3

Redes convolucionales

- **Imagenet challenge**

- **1000 clases, imágenes RGB**
  - Hay ambigüedad en las etiquetas
  - por eso a veces se da la tasa top5
  - (error si no está entre las 5 primeras opciones)



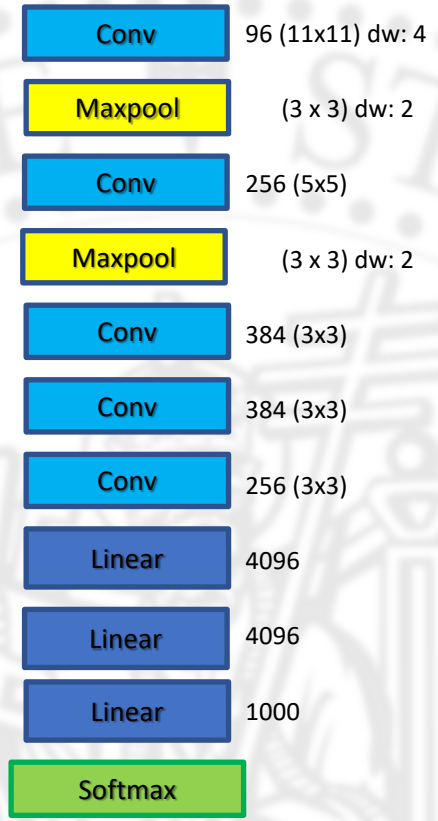


Redes convolucionales

- **Alexnet (Krizhevsky, 2012)**

Krizhevsky, A., Hinton, G. E., Sutskever, I. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1106-1114.

  - Imagenet: 84.6 % top5 accuracy
  - Aumento de profundidad con respecto a LeNet5:
    - GPUs
    - Uso de momento y weight_decay
    - Nueva técnica: dropout
    - Usa relus
    - Flatten -> modelo muy grande 62M parámetros



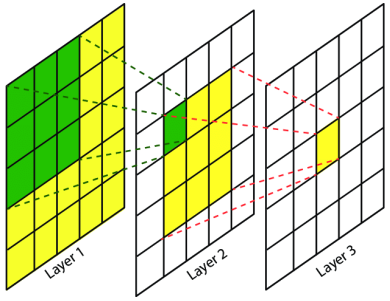
https://engmrk.com/alexnet-implementation-using-keras/

Redes convolucionales

- **VGG16 (Simonyan, 2014)**

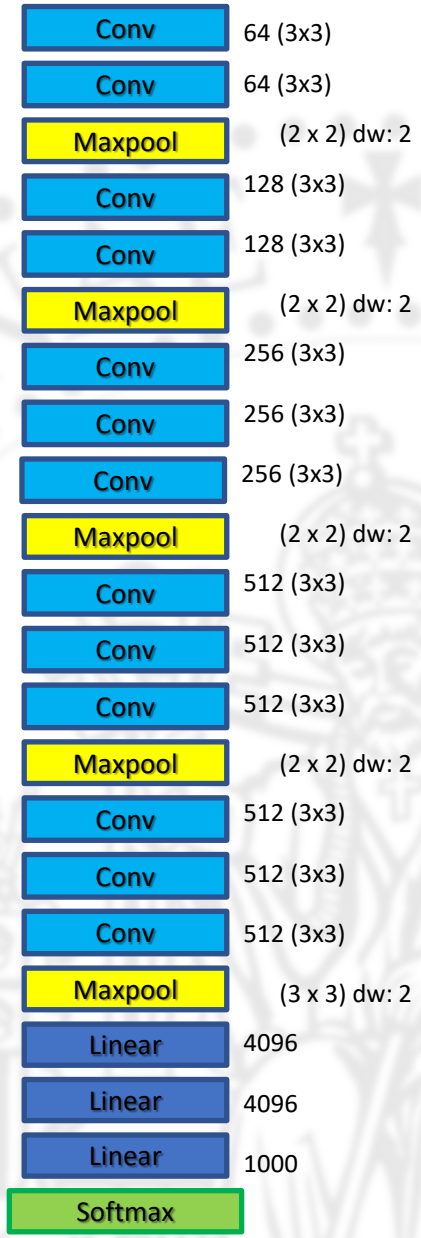
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

  - Imagenet 92.7% top-5
  - Más profundidad 16 capas 138M:



- El número de coeficientes para un tamaño 5x5 es de 25
- La composición de dos de 3x3 3*3 + 3*3 = 18

Mientras que el campo receptivo es equivalente



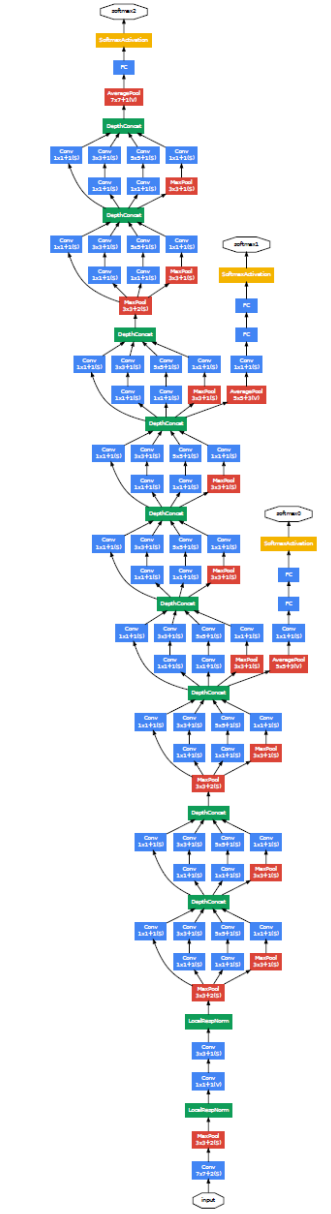
https://engmrk.com/alexnet-implementation-using-keras/

Redes convolucionales

- **GoogleLeNet Inception (Szegedy, 2014)**

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

  - Imagenet 93.3% top-5 accuracy (winner 2014)
  - Aumento de profundidad, 22 capas, 4M
  - Mean average pooling

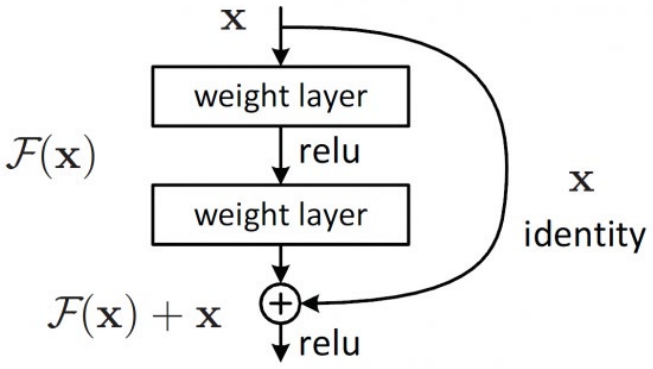


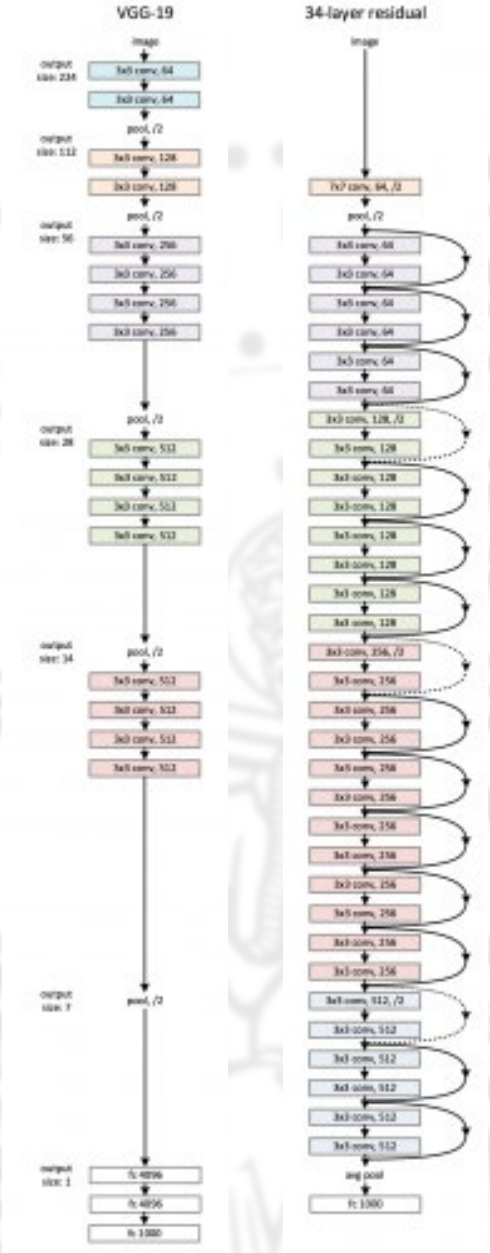
Redes convolucionales

- **Resnet (Kaiming, 2015)**

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

  - Imagenet 2015: 96.43% top-5
  - Aumento enorme de la profundidad:
    - 152 capas, 60M
    - Estructura residual

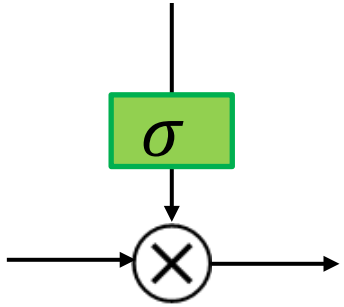


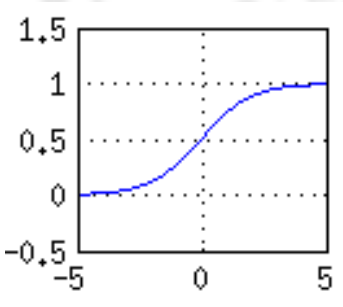


https://neurohive.io/en/popular-networks/resnet/

Redes convolucionales

- **Modelos de atención**
  - Puerta sigmoide:
    - juega el papel de una puerta lógica que deja pasar o no la información en función de su entrada





$$f(x) = \frac{1}{1 + \exp(-x)}$$

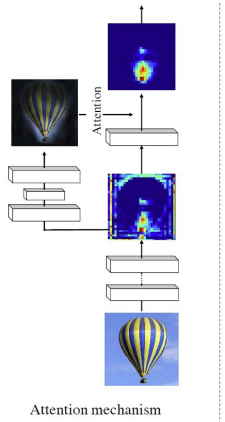
Redes convolucionales

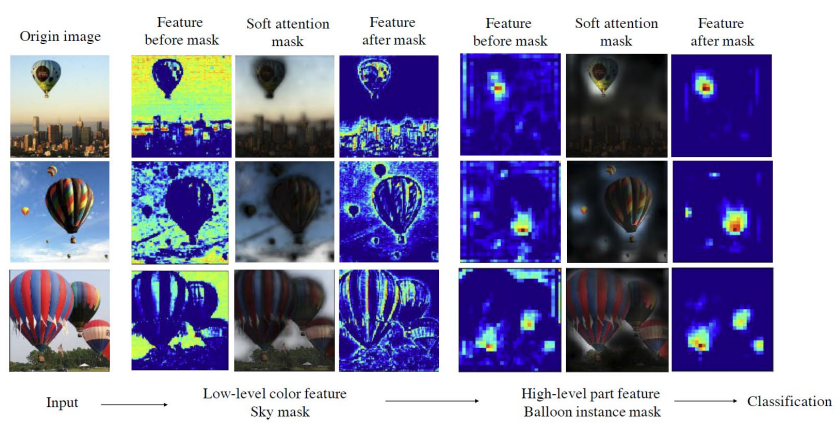
- **Modelos de atención**

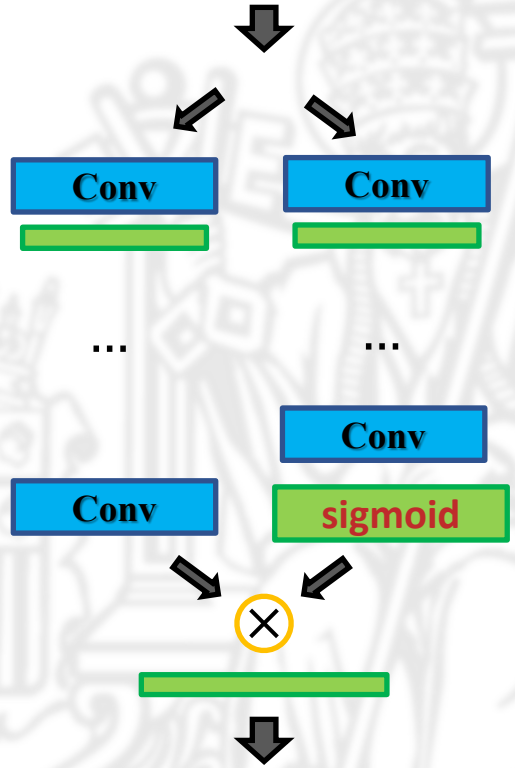
Srivastava, Rupesh Kumar, Klaus Greff, and Jürgen Schmidhuber. "Highway networks." arXiv preprint arXiv:1505.00387 (2015).

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., ... & Tang, X. (2017). Residual attention network for image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).

  - En estos modelos la red decide en la rama de atención
    - Qué información debe anular/dejar pasar





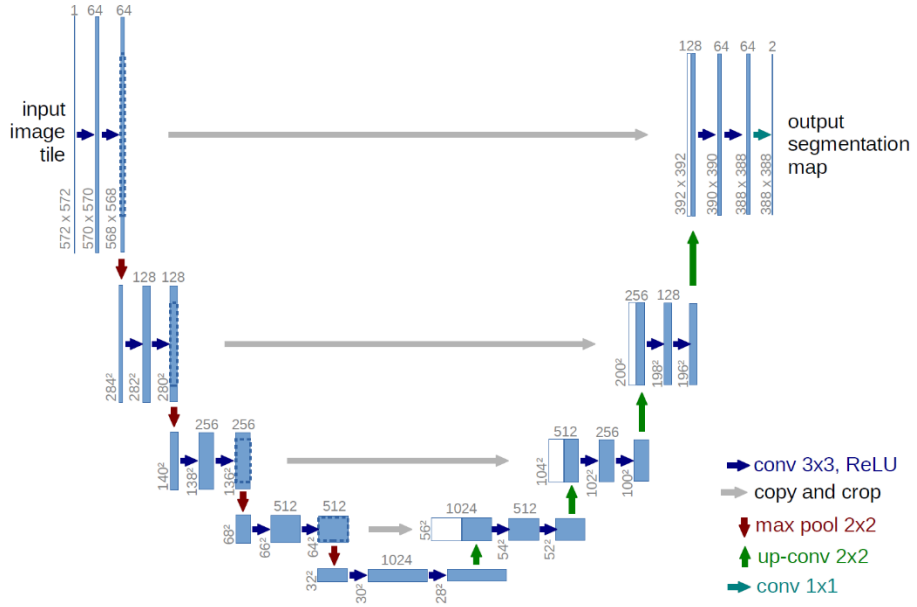


Redes convolucionales

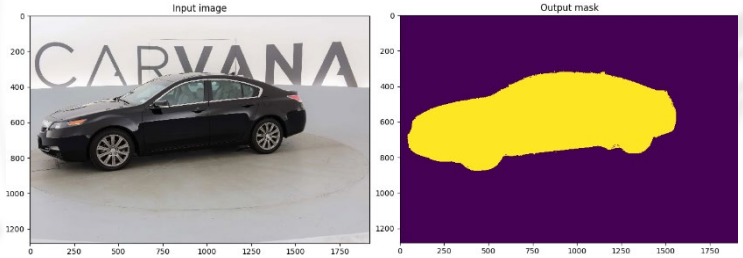
- **Unet (Ronneberger, 2015)**

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

  - Procesado multiescala
  - Los caminos bypass mejoran la resolución
  - Muy utilizada en segmentación



Multiclass segmentation

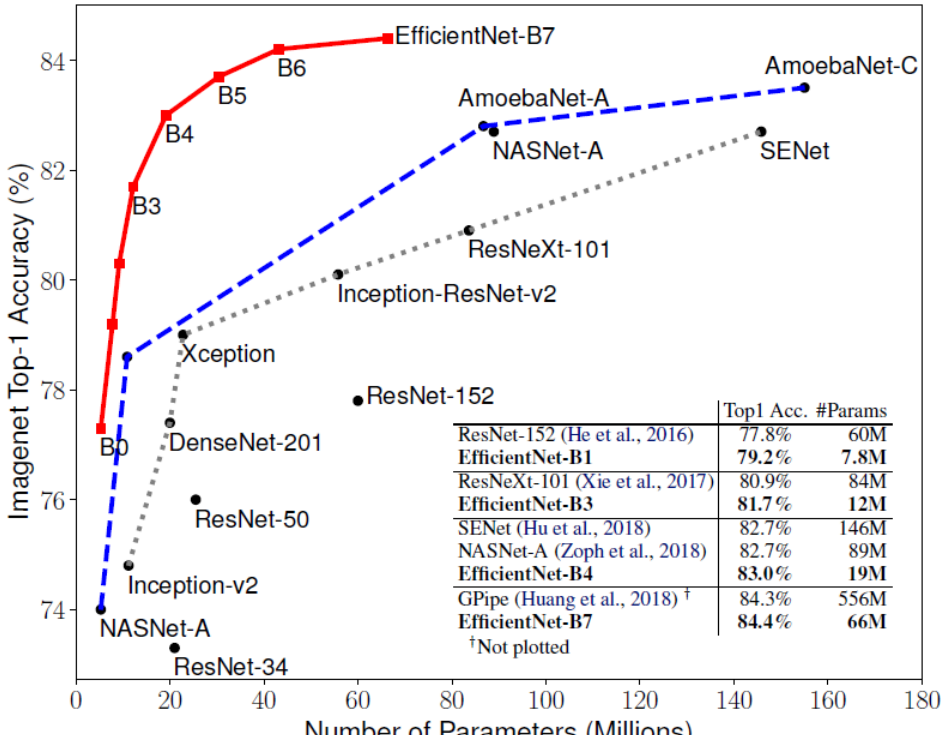


Redes convolucionales

- **Efficientnet (Tan, Quoc Le, 2019)**

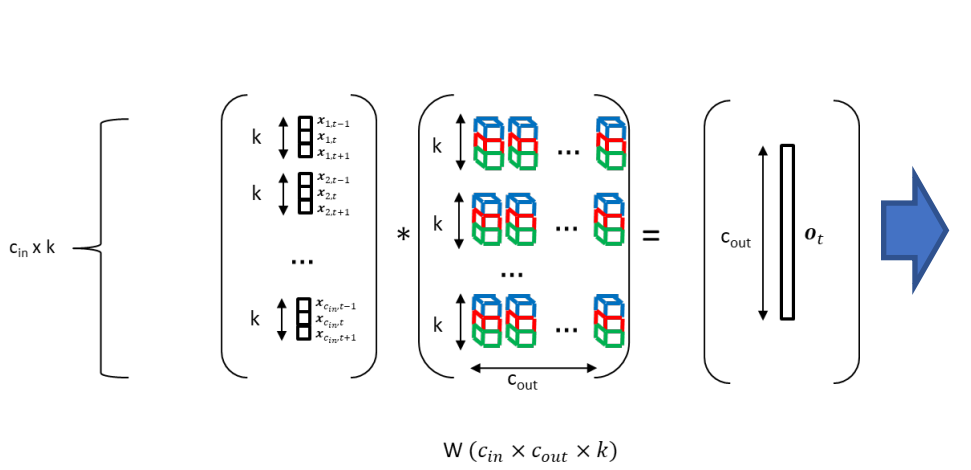
Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946.

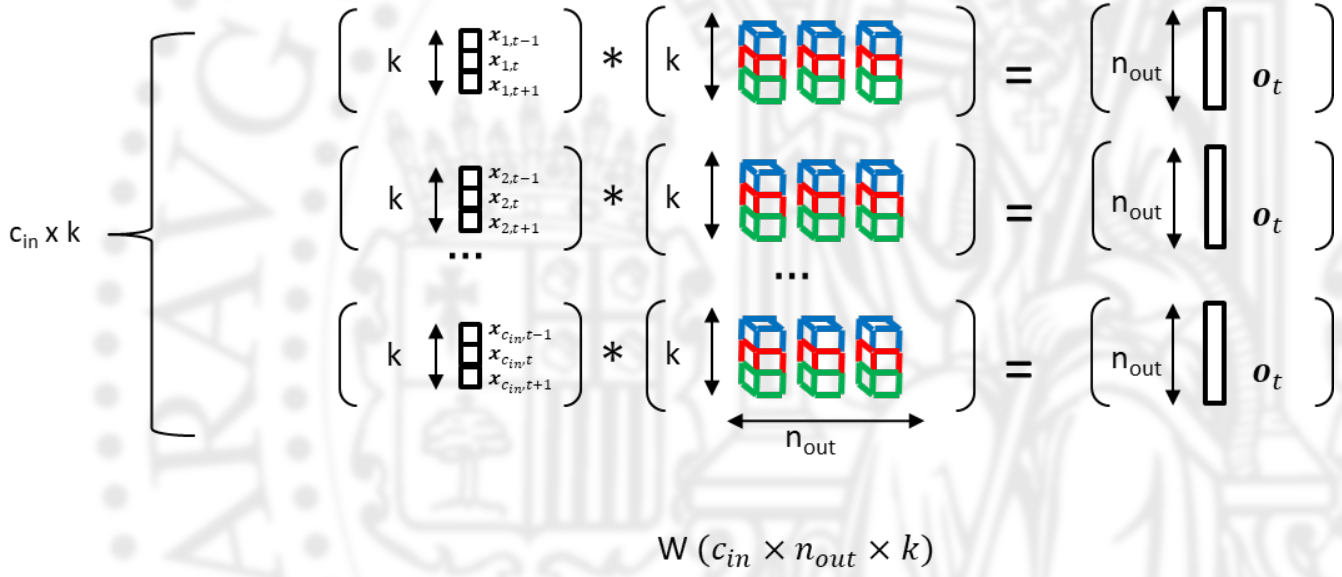
  - En los últimos años se están buscando arquitecturas que reduzcan el coste computacional/energético



Redes convolucionales

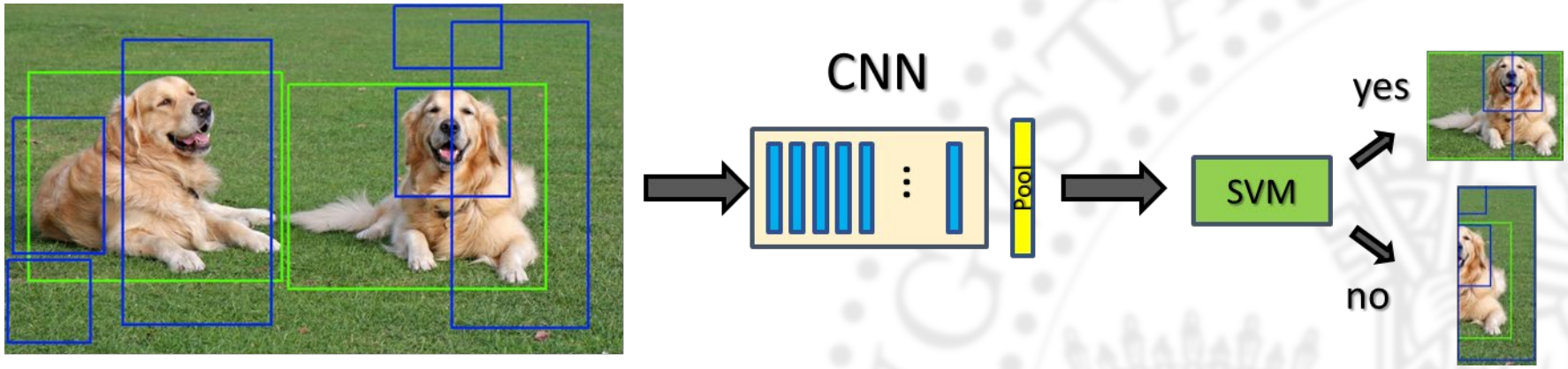
- **Efficientnet (Tan, Quoc Le, 2019)**
  - Depthwise convolutions: convoluciones ligeras
    - no multiplican todos los canales





Redes convolucionales

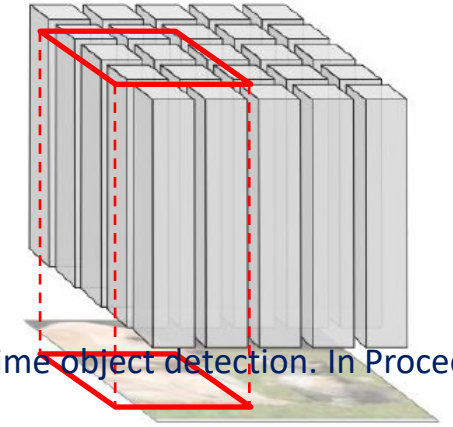
- **Detección de objetos**



- **Region CNN**
  - Se entrena una red para procesar las regiones y extraer un embedding
  - Poco eficiente porque se procesa varias veces la imagen

Redes convolucionales

- **Object detection**

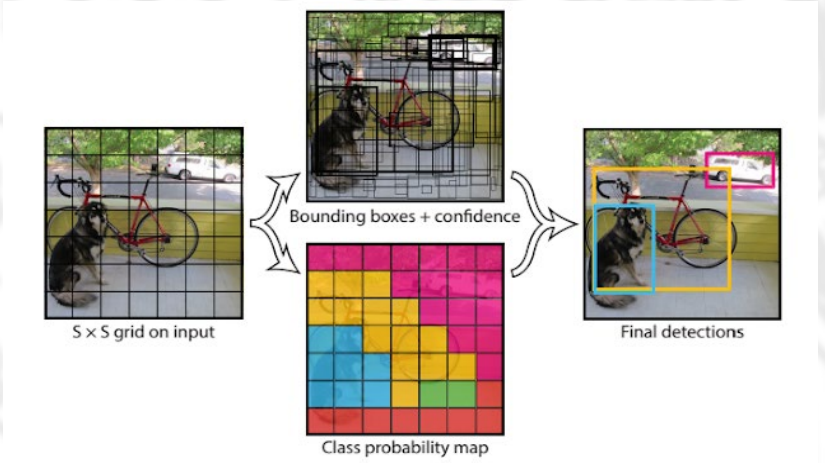


https://distill.pub/2018/building-blocks/

- **YOLO, You Only Look Once:**

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

  - La red procesa la imagen una vez
  - Por cada zona final predice los objetos que hay en esa zona
  - Muy rápido, pero puede perder objetos



Redes convolucionales

Labeled Faces in the Wild

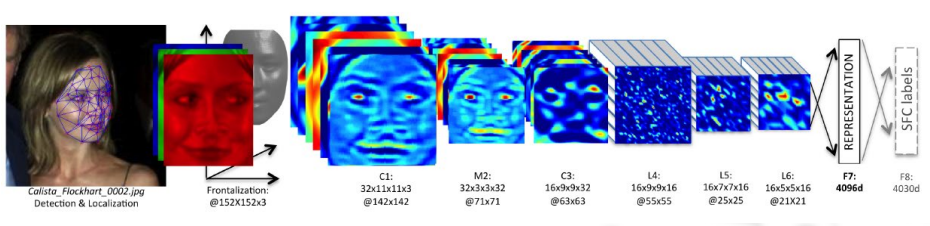
http://vis-www.cs.umass.edu/lfw/

- **Reconocimiento facial**
  - responder la pregunta
    - ¿Es la misma persona?
  - La mayoría de los enfoques usan datos de entrenamiento etiquetados (imágenes + identidad)
  - Entrenar una tarea de clasificación, función de pérdida CrossEntropy

- **Deep face 2014**

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).

Input:
Imagen cara

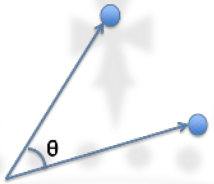


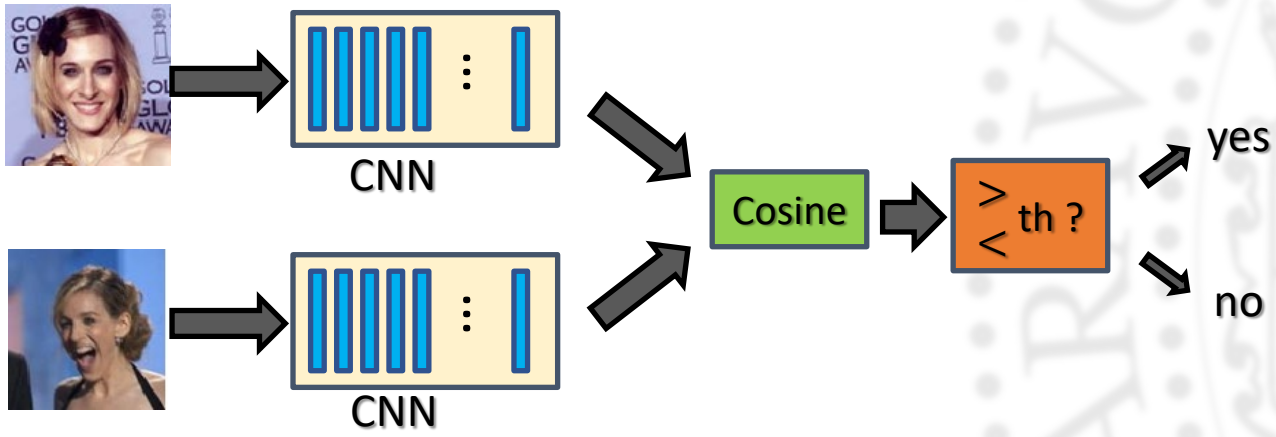
Label:
Identidad, se intenta clasificar

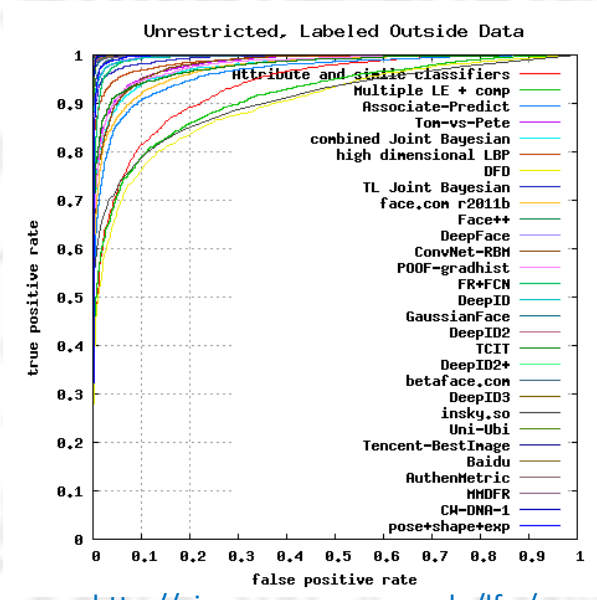
Redes convolucionales

- **Simulitud coseno**
  - Mide el ángulo de separación entre dos vectores
  - Muchos sistemas usan una red p. ej (CNN) para generar **representaciones**
    - La última capa lineal y softmax se descartan
    - Se decide mediante un umbral de parecido

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$





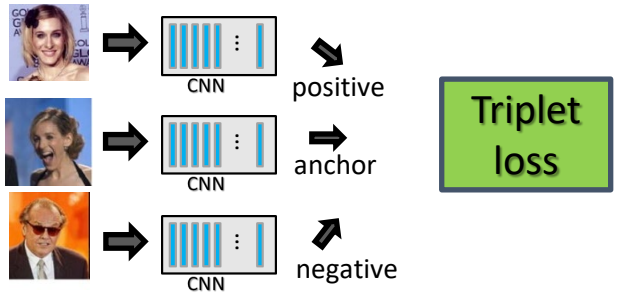


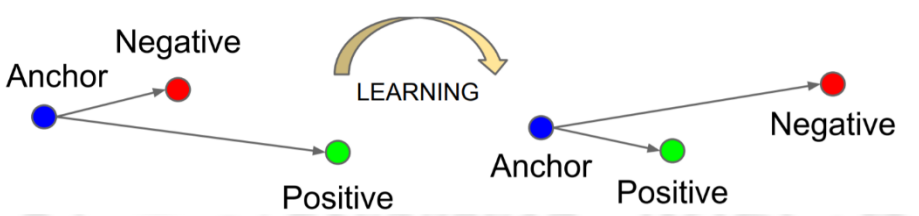
http://vis-www.cs.umass.edu/lfw/

Redes convolucionales

- **Facenet(Schroff 2015)**

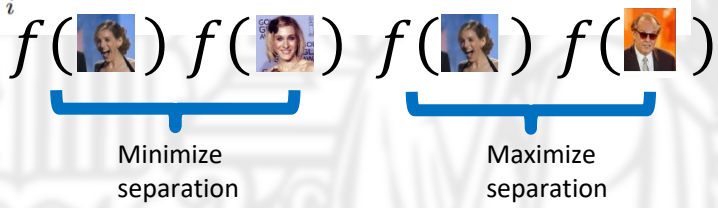
Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).





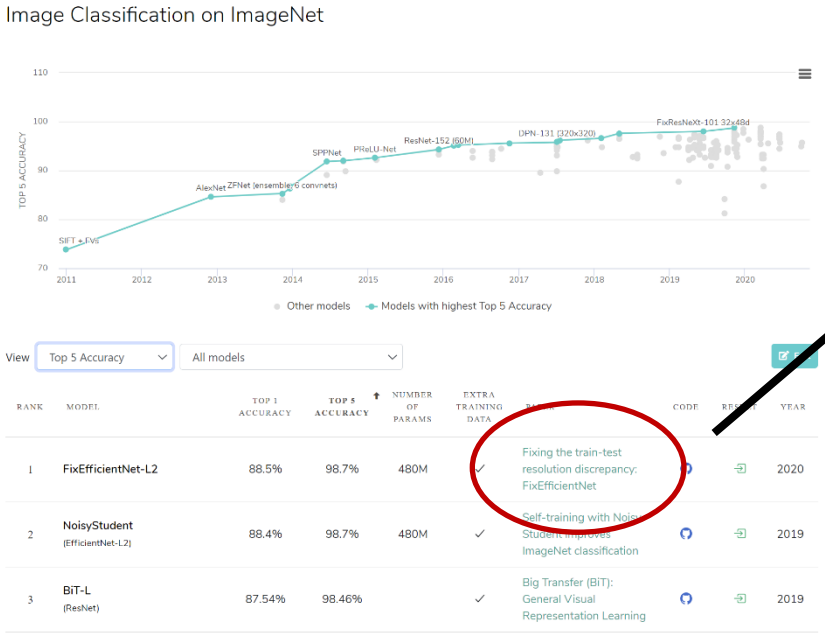
- **Triplet loss**
  - Ancla y positive misma identidad: los acerca
  - Separa negativo del ancla

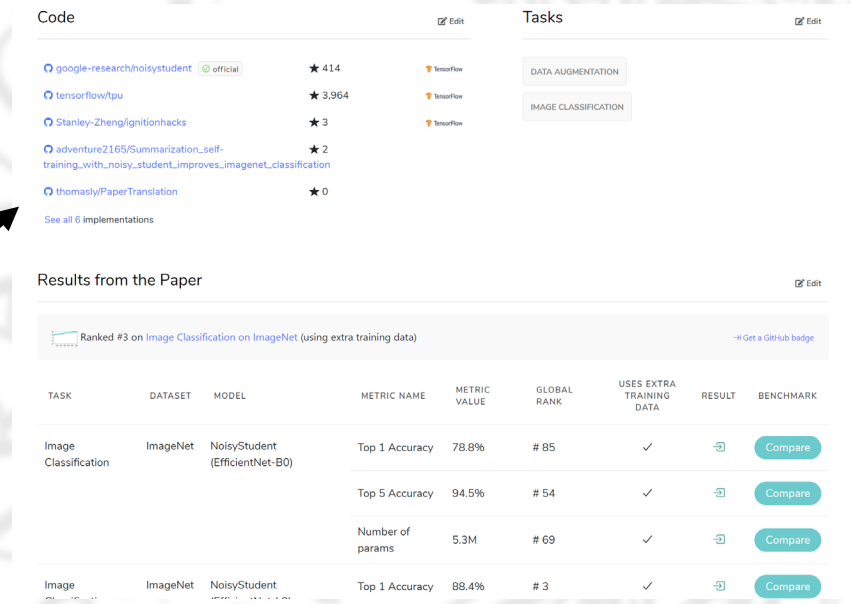
$$Triplet_{loss} = \sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]$$



Redes convolucionales

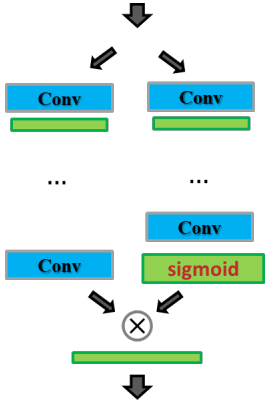
- **Papers with code:**
  - https://paperswithcode.com/sota/image-classification-on-imagenet
  - Muy útil para comprobar el estado del arte de muchas tareas
  - Además indica implementaciones en código abierto disponibles

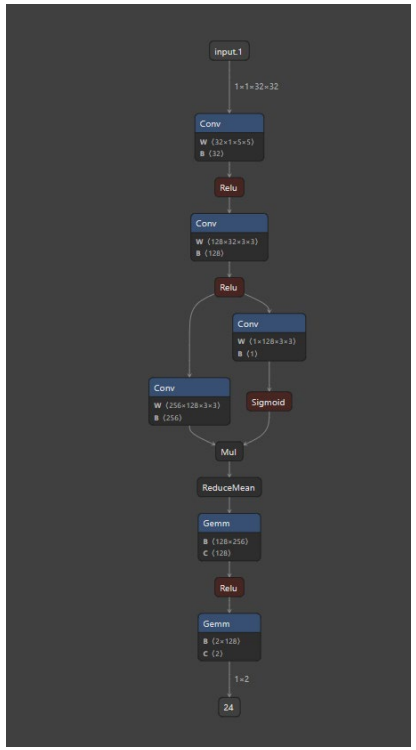


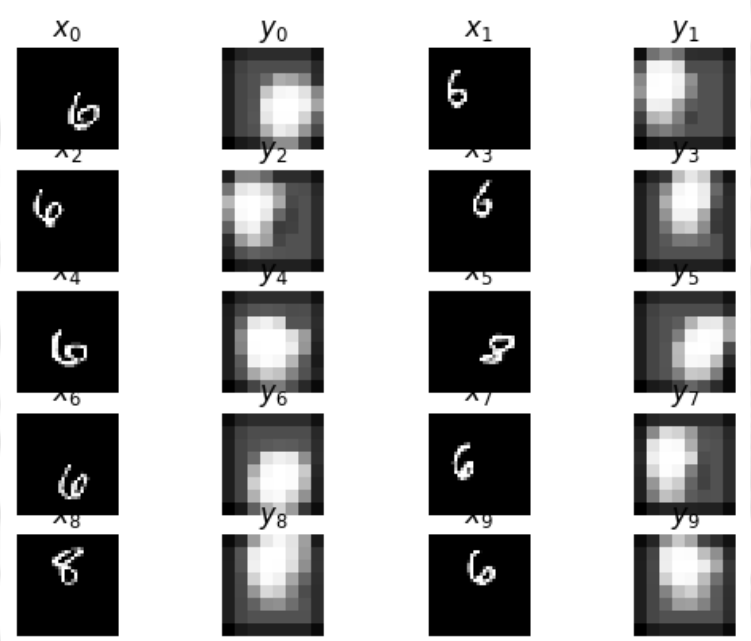


Redes convolucionales

- **Ejemplo1:** https://colab.research.google.com/drive/1Vu3HwaFlxAgDgS4KnD8gfaVwWhSaevsu?usp=sharing



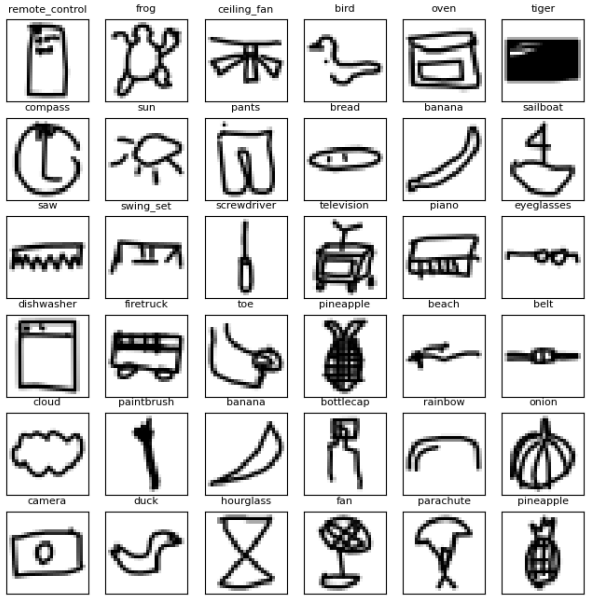




https://netron.app/

Redes convolucionales

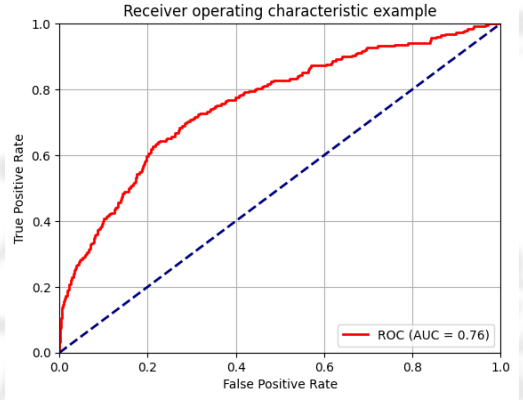
- **Ejemplo2:** https://colab.research.google.com/drive/1Zi1icw0J1eSQIGAoejjw27TYI3pVAT4q?usp=sharing

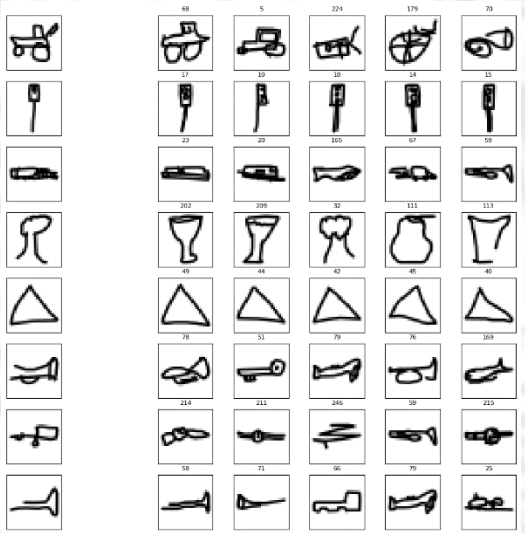


detección

embeddings

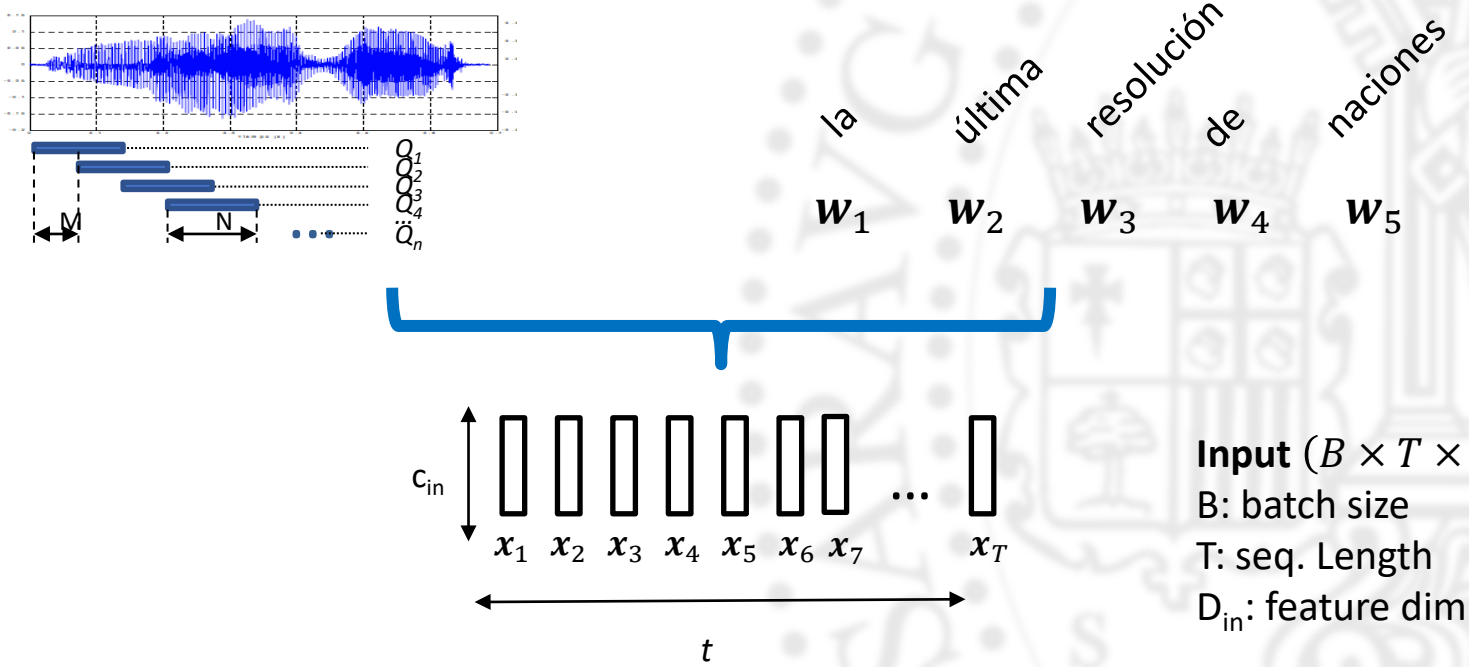
búsqueda





Redes recurrentes

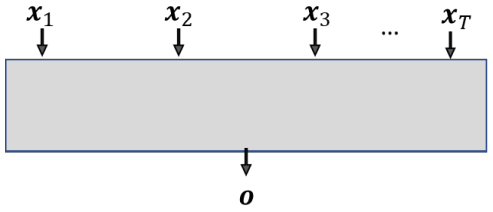
- **Modelos de secuencia**
  - Muchas aplicaciones implican secuencias (orden temporal)
  - Ejemplos
    - Audio, voz, texto, señales biológicas (ECG, EEG), señales muestreadas de sensores…



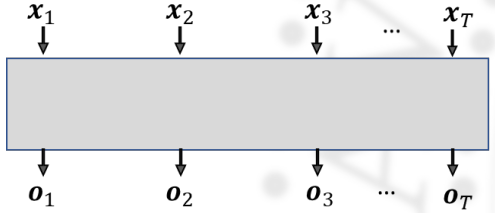
Input $(B \times T \times D_{in})$
B: batch size
T: seq. Length
D_{in}: feature dim.

Redes recurrentes

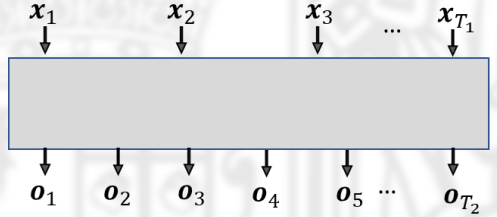
- **Modelos de secuencia**
  - Salida simple (global)
  - Salida múltiple:
  - Seq2seq
    - Diferente tamaño entrada/salida



Input $(B \times T \times D_{in})$
Output $(B \times D_{out})$

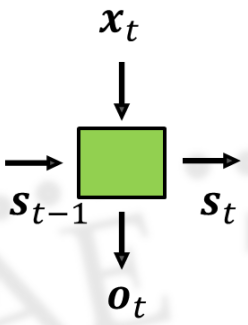


Input $(B \times T \times D_{in})$
Output $(B \times T \times D_{out})$



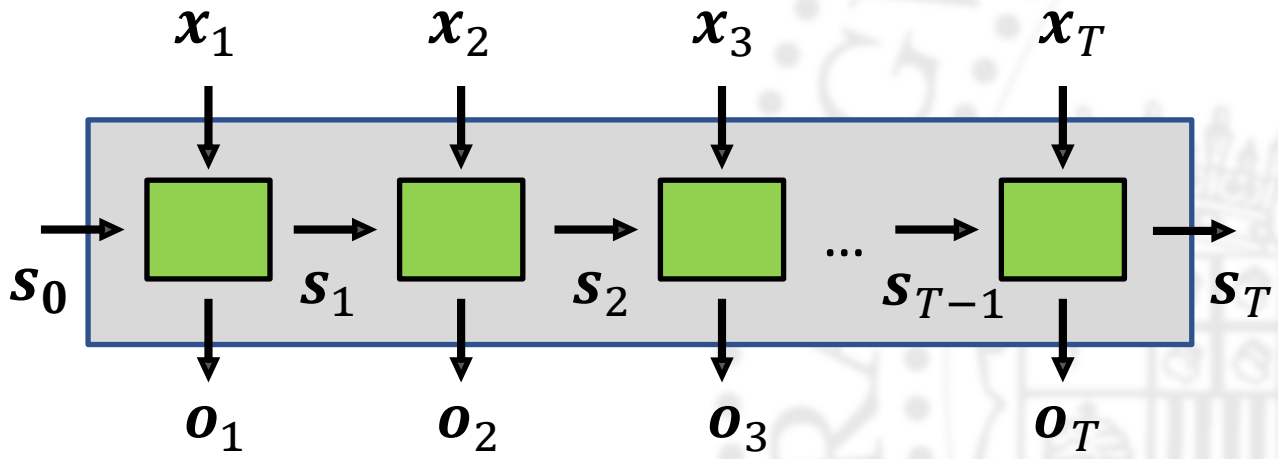
Input $(B \times T_1 \times D_{in})$
Output $(B \times T_2 \times D_{out})$
$T_1 \neq T_2$

Redes recurrentes



RNN cell

- Procesado secuencial mediante una celda básica
  - Guarda la información contexto en un estado
  - Puede ser un cuello de botella para la información del pasado: olvida
  - Es difícil de entrenar de forma robusta para secuencias largas



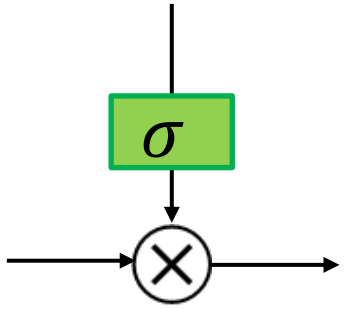
RNN

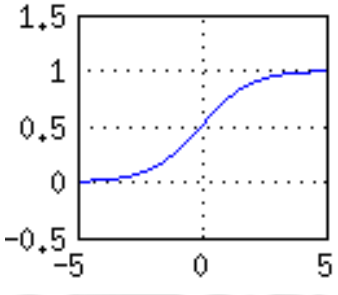
Redes recurrentes

- **LSTM**

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

  - Supuso un gran avance sobre modelos anteriores:
    - problemas de estabilidad
  - Usa puertas sigmoide como mecanismo de atención de tipo puerta

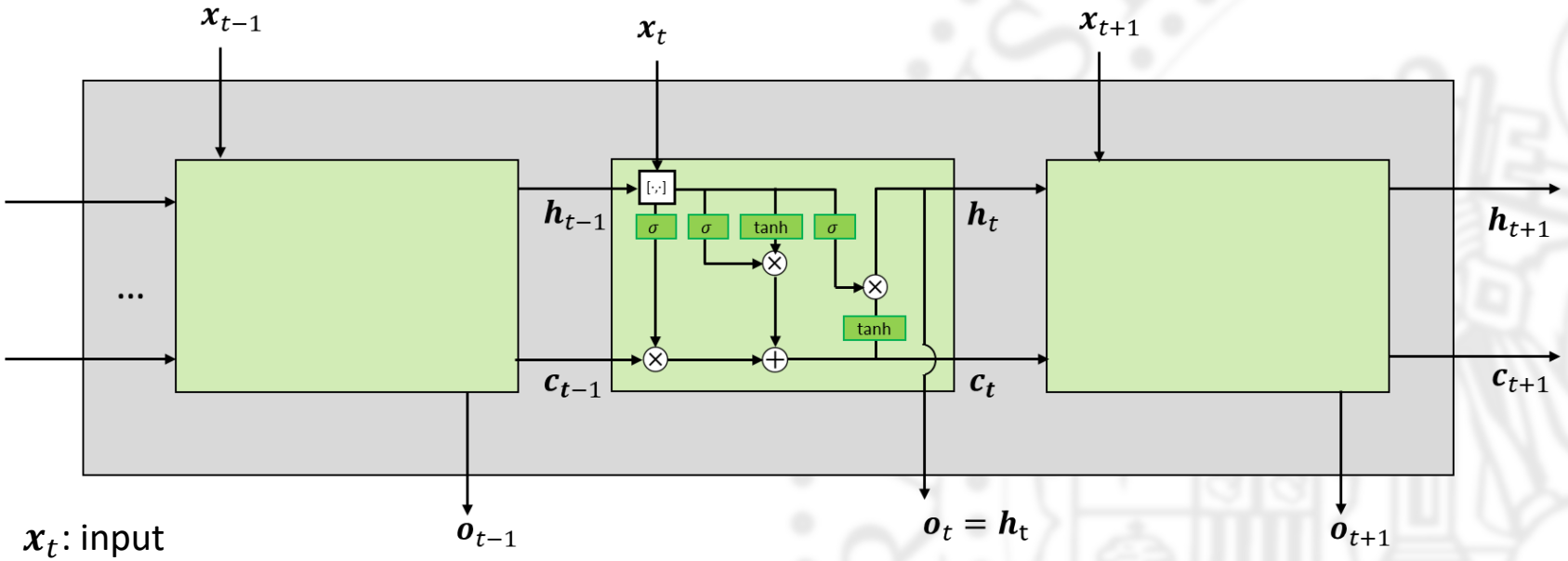




$$f(x) = \frac{1}{1 + \exp(-x)}$$

Redes recurrentes

- **LSTM**
  - La estructura interna de las celdas implica varios modelos de atención
  - La información de estado se transmite a la siguiente celda

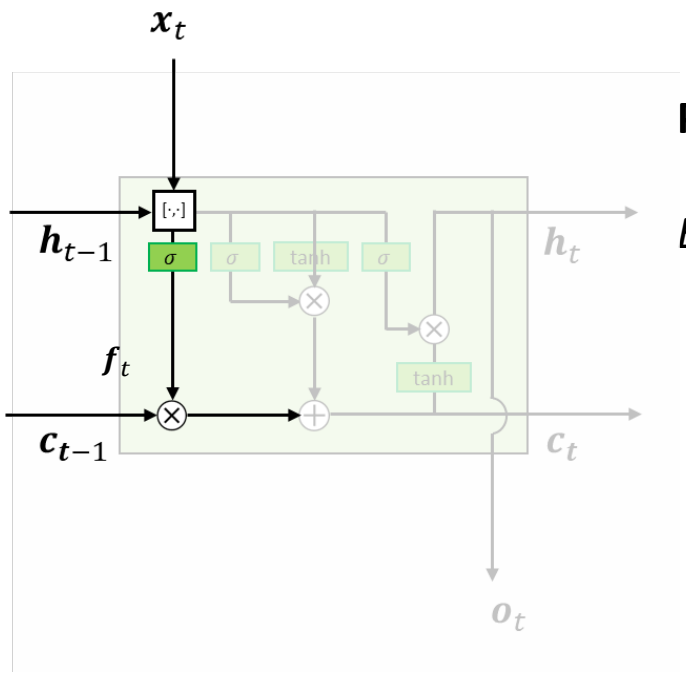


x_t: input
o_t: output
h_t: hidden state
c_t: cell state

Christopher Olah Understanding LSTM Networks 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Redes recurrentes

- **LSTM**



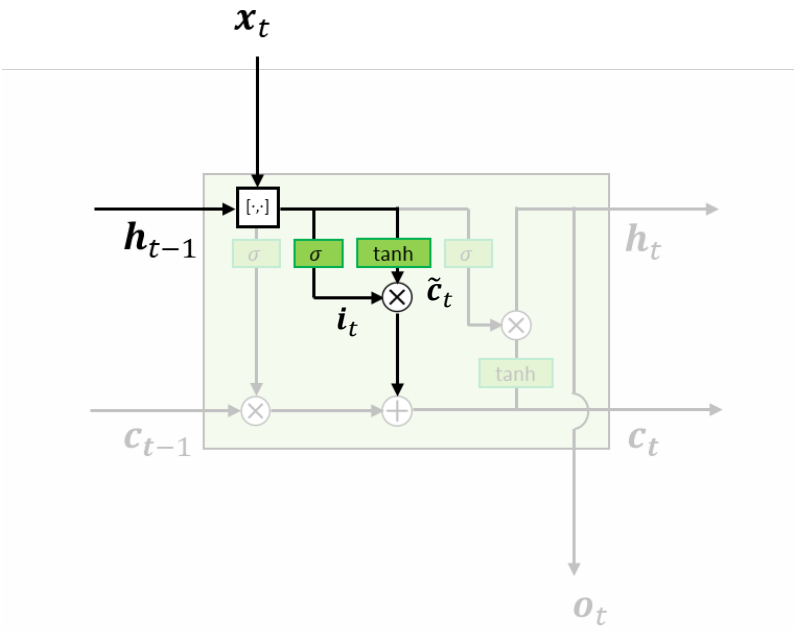
Puerta olvido

Elige mediante una capa qué información del pasado se borra

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\ ] + \boldsymbol{b}_f\right)$$

Redes recurrentes

- **LSTM**



Puerta entrada

Elige mediante una capa qué información de la entrada se introduce al modelo

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\,] + \boldsymbol{b}_i)$$

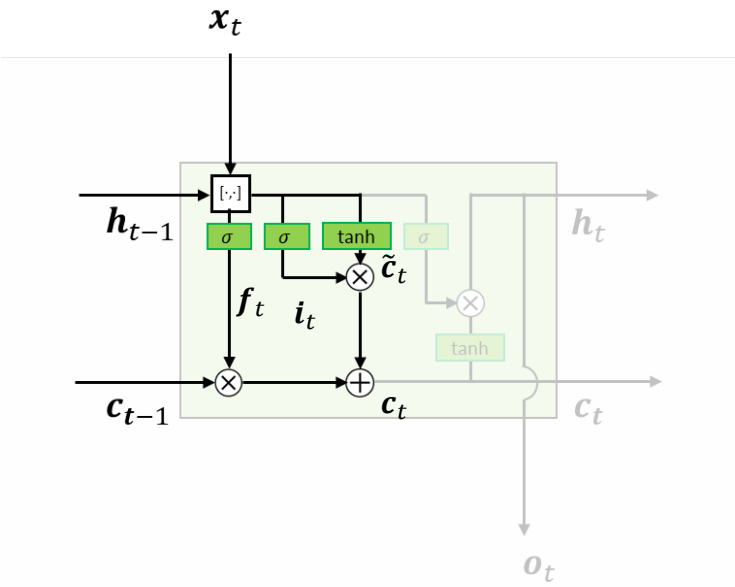
Nuevo estado

Calcula la información que se introduce al modelo

$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_c \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\,] + \boldsymbol{b}_c)$$

Redes recurrentes

- **LSTM**



Estado

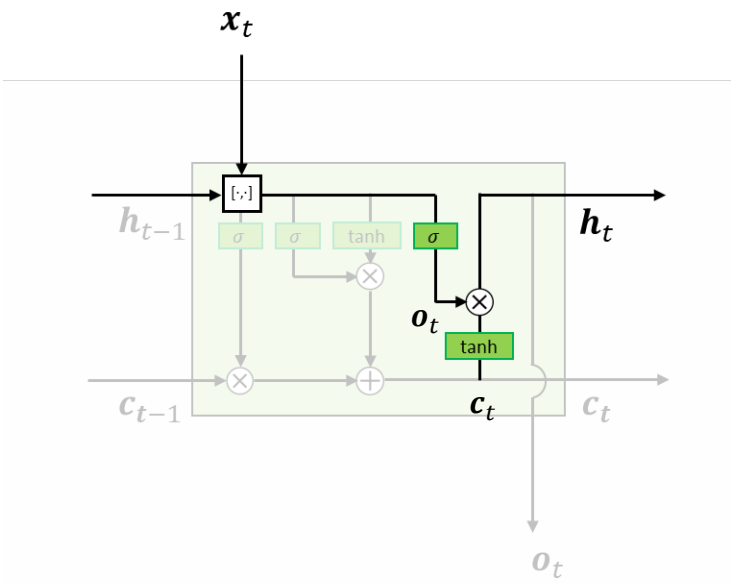
El nuevo estado se calcula con la parte que se conserva del estado anterior y la información nueva

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$

∘, Hadamard product, elementwise

Redes recurrentes

- **LSTM**



Puerta de salida

Elige qué información va a salir del modelo

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\,] + \boldsymbol{b}_o)$$

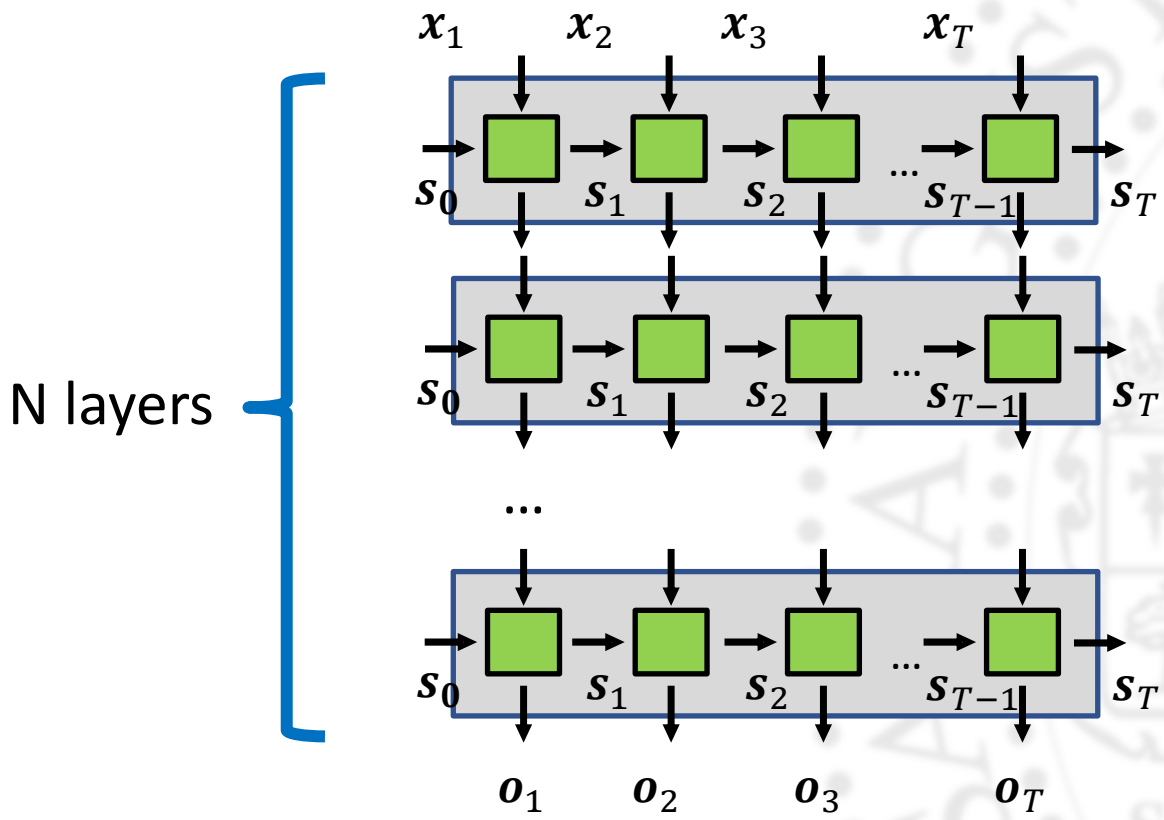
Valores de salida

Calcula las nuevas salidas

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_t)$$

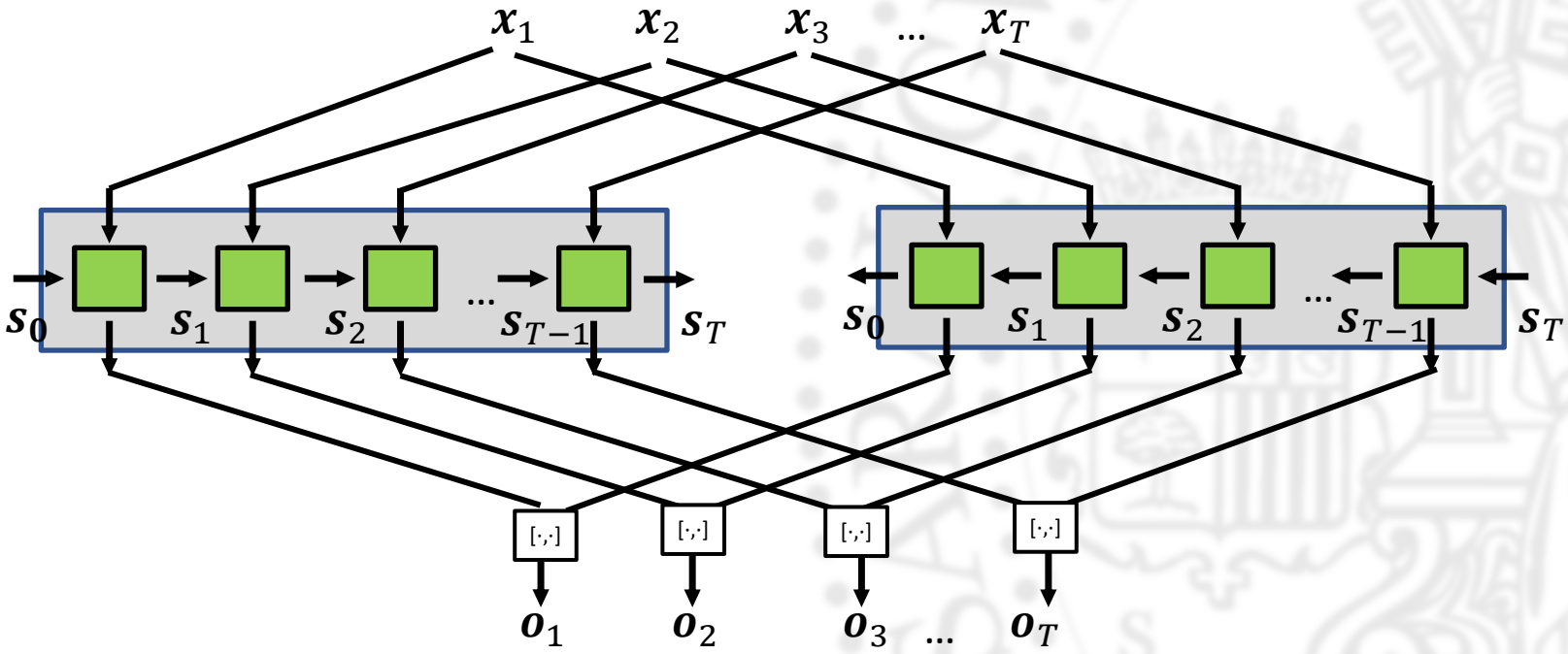
Redes recurrentes

- **Redes multicapa**
  - pueden apilarse en múltiples capas, proporcionando una mayor abstracción



Redes recurrentes

- **Redes bidireccionales**
  - Cuando la secuencia se puede analizar sin la restricción del modelado causal
    - Dos redes operando una en orden directo y la otra en inverso
    - Después de la concatenación la dimensión es $2D_h$ (hidden dimension)

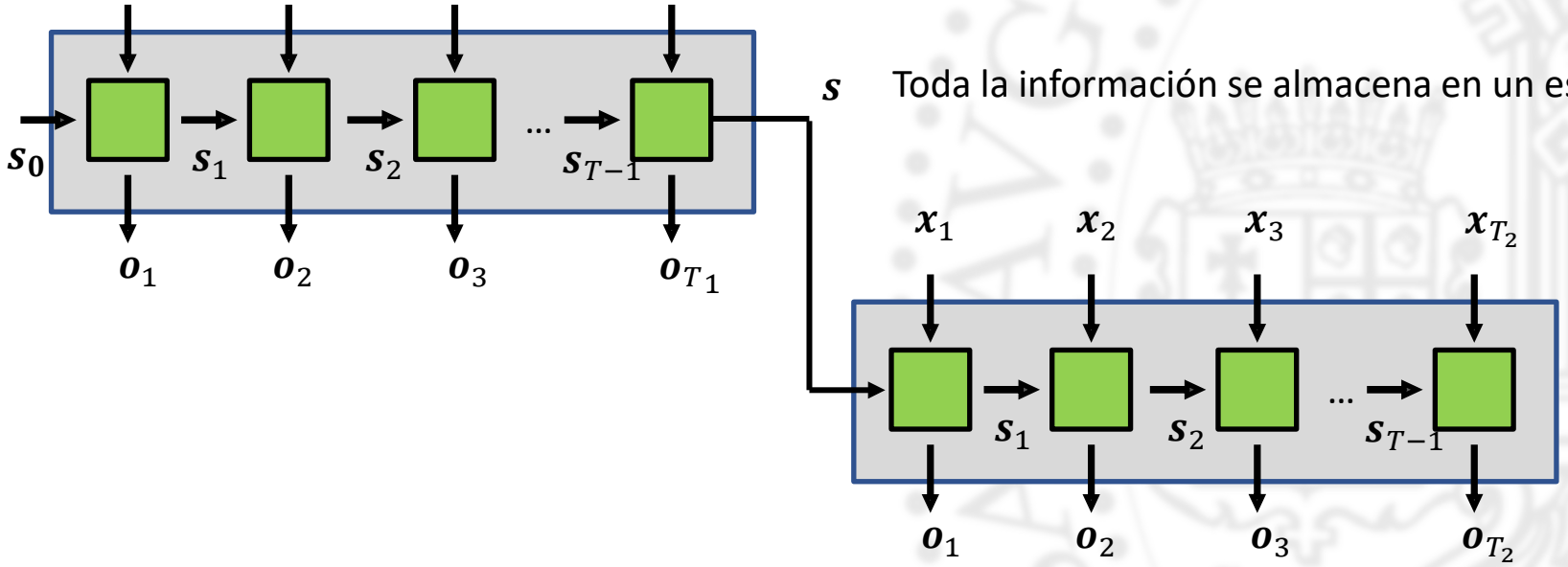


Redes recurrentes

- **Redes seq2seq (Sutskever 2014)**

Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.

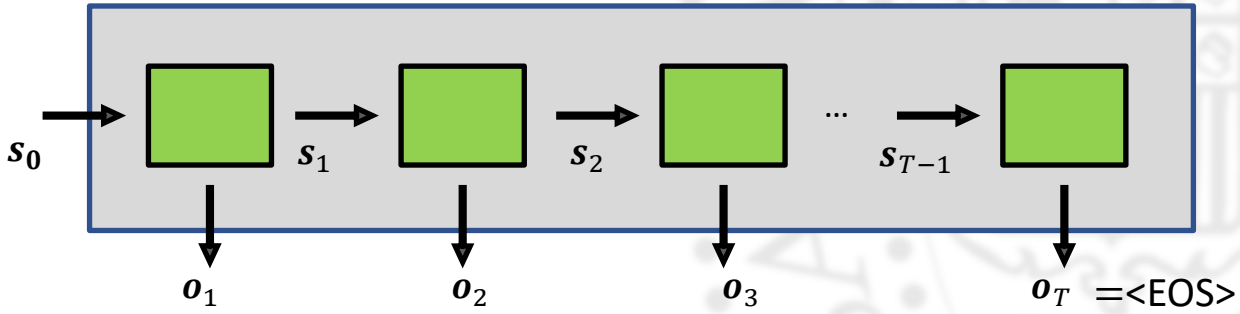
– Traduce una secuencia de entrada en una de salida
  - Primeros modelos competitivos en traducción automática de idiomas



Toda la información se almacena en un estado

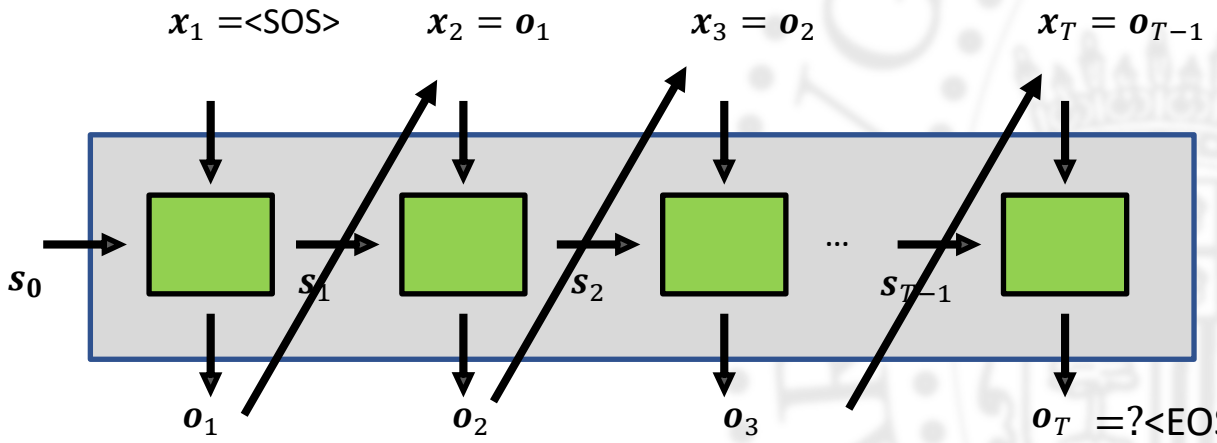
Redes recurrentes

- **Salida autorregresiva**
  - Generación texto a partir de un estado, problemas:
    - Se puede degradar la calidad para secuencias largas
    - El estado que se va pasando de unas celdas a otras puede acumular errores
  - Si intentamos generar solo a partir de el estado
    - el sistema apenas funciona en ejemplos triviales
    - ¿Qué podemos hacer?



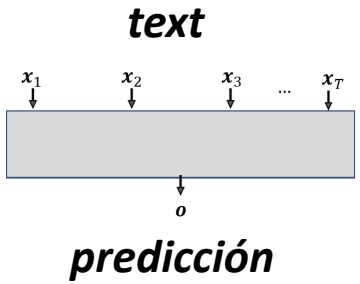
Redes recurrentes

- **Salida autorregresiva**
  - Solución: cada celda recibe la salida de la celda previa como entrada
    - Para la primera entrada se usa el símbolo especial: <SOS> start of sequence
    - La red finaliza cuando se predice el símbolo <EOS> end of sequence



Redes recurrentes

- **Ejemplos:**
  - Clasificación de texto
  - Clasificación de tema
  - Reconocimiento de autor/fraude
  - Clasificación de idioma
  - …

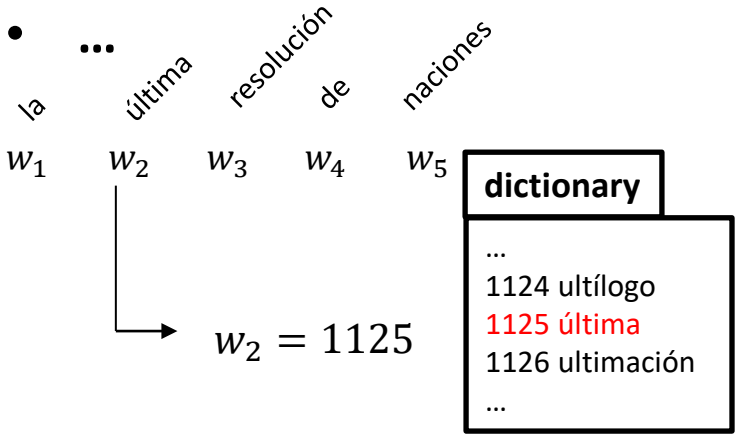


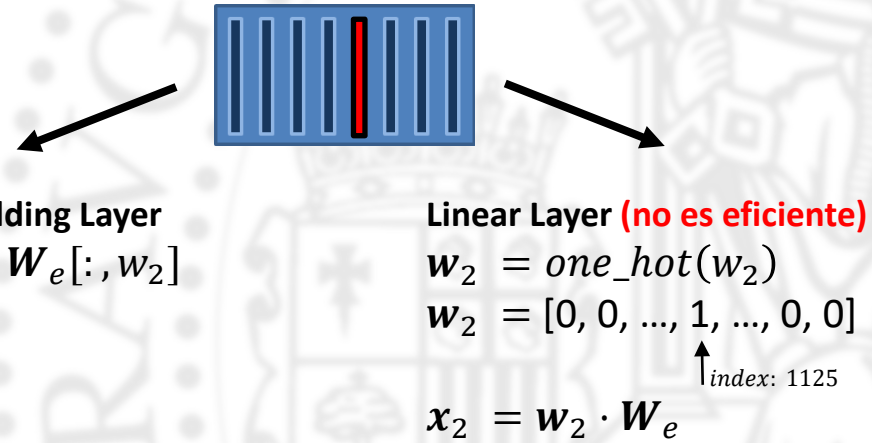
pytorch

```
import torch
layer = torch.nn.Embedding(nwords, dim)
```

keras

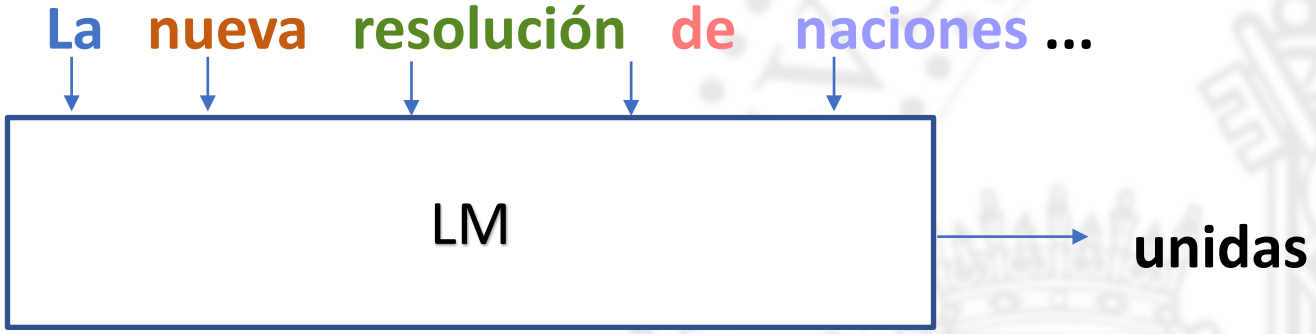
```
from tensorflow import keras
layer = keras.keras.layers.Embedding(nwords, dim)
```





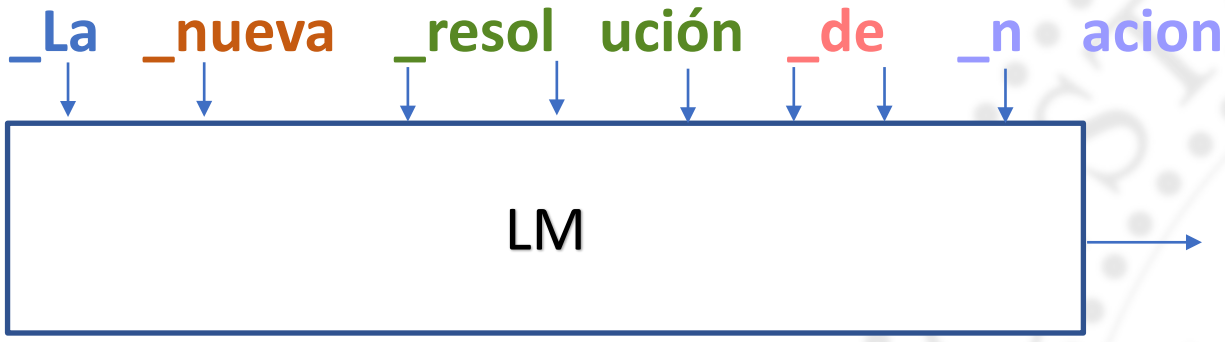
Redes recurrentes

- **Modelo de lenguaje: objetivo**
  - predecir la siguiente palabra en una secuencia de palabras (un texto)



Redes recurrentes

- **Modelo de lenguaje: partes de palabra (tokens)**



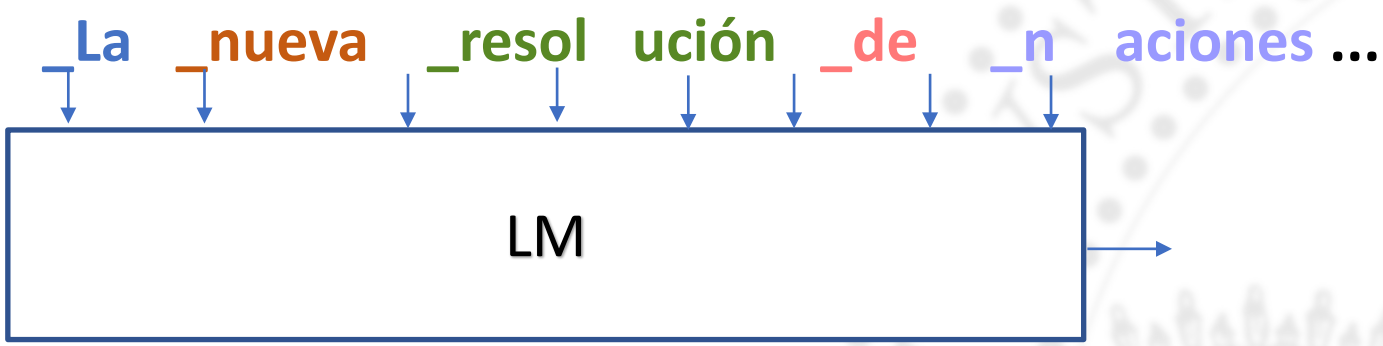
p(unidas | _La, _nueva, _resol, ución, _de, _n, aciones)

token	prob
_un	0.48
_y	0.04
_en	0.03
_Un	0.03
_que	0.02
,	0.02
…	

- Lo que obtendremos es una estimación de la probabilidad
- ¿Para qué queremos esas estimaciones?
- ¿No es suficiente con la más probable?

Redes recurrentes

- **Modelo de lenguaje: generación**




```
--------------------
_un (0.48) ->
        idas        (0.99)
        ific        (0.00)
        ip          (0.00)
        as          (0.00)
        id          (0.00)
--------------------
_y (0.04) ->
        _pue        (0.21)
        _g          (0.07)
        _raz        (0.04)
        _nacional   (0.04)
        _de         (0.04)
--------------------
_en (0.03) ->
        _la         (0.13)
        _el         (0.13)
        _conflic    (0.05)
        _las        (0.04)
        _Europa     (0.03)
```

  - Generar secuencias (la salida de los LMs generativos)
  - Controlar otros procesos: Reconocedor de automático del habla, traductores, etc.

Redes recurrentes

- **Representación de palabras**
  - Cada palabra del vocabulario se representa mediante un vector (32k por ejemplo en LLAMA)
  - Pensemos en una versión simplificada donde cada palabra
  - fuera descrita mediante una ficha de miles de preguntas (sparse vector)

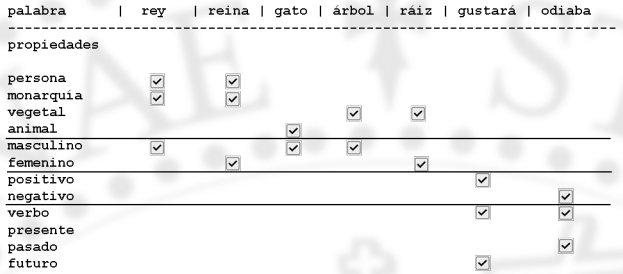
palabra	rey	reina	gato	árbol	ráiz	gustará	odiaba
propiedades							
persona	☑	☑					
monarquía	☑	☑					
vegetal				☑	☑		
animal			☑				
masculino	☑		☑	☑			
femenino		☑			☑		
positivo						☑	
negativo							☑
verbo						☑	☑
presente							
pasado							☑
futuro						☑	

Un cambio de significado puede ir asociado a cambiar uno de esos aspectos, **recuerdan más a un ideograma chino** que a una representación mediante un alfabeto fonético

木 → 本
Árbol **Raíz**

Redes recurrentes

- **Representación de palabras**



palabra	rey	reina	gato	árbol	raíz	gustará	odiaba
propiedades							
persona	✓	✓					
monarquía	✓	✓					
vegetal				✓	✓		
animal			✓				
masculino	✓		✓	✓			
femenino		✓			✓		
positivo						✓	
negativo							✓
verbo						✓	✓
presente							
pasado							✓
futuro						✓	

  - Aún siendo una simplificación podemos intuir
  - dos operaciones importantes en los modelos de lenguaje
  - El mecanismo de comparación de palabras (semántico)
  - dos palabras son parecidas si tienen muchas coincidencias (productor escalar)

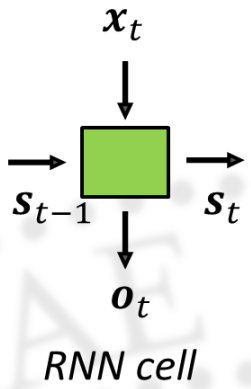
`rey - hombre + mujer -> (está cerca de) reina`

  - Podemos transformar una palabra para llegar a otra mediante manipulaciones algebraicas (aproximadamente claro):

```
tenis – raqueta + bate              -> béisbol
motocicleta – motor + pedales       -> bicicleta
amar – positivo + negativo          -> odiar
parís – francia + españa            -> madrid
```

    - Otras analogías obtenidas (word2vec):

Redes recurrentes



RNN cell

- **LSTM**

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780

  - Limitación: cada celda tiene una dimensión finita para recibir información de los
  - Instantes previos y escribir nueva información para el futuro

2015 *http://karpathy.github.io/2015/05/21/rnn-effectiveness/*

- Red recurrente (LSTM) 3 capas dimensión 512, **5.8M**
- **4.4MB** texto Shakespeare

> DUKE VINCENTIO: Well, your wit is in the care of side and that.
>
> Second Lord: They would be ruled after this chamber, and my fair nues begun out of the fact, to be conveyed, Whose noble souls I'll have the heart of the wars.
>
> Clown: Come, sir, I will make did behold your worship.
>
> VIOLA: I'll drink it.

2016 *https://medium.com/deep-writing/harry-potter-written-by-artificial-intelligence-8a9431803da6*

- Primeros 4 libros

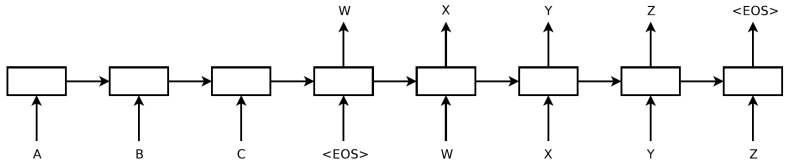
> Ron didn't even upset her little ingredients on the toilet, and a group of third-year girls last year. Highly bushy and then burst away from them quickly.
>
> "Thought you're all right?" he said.
>
> Harry grinned at Harry. "Why should she be cheerful so while you gave detentions, Moody!"

Redes recurrentes

- **Translation (Sutskever, 2014)**

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

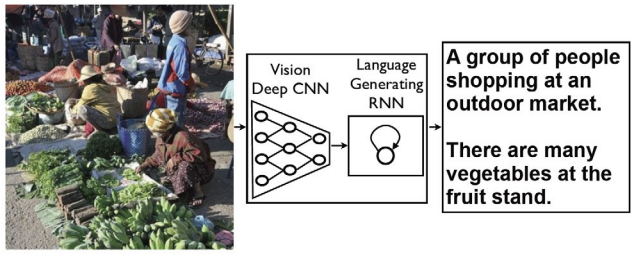
  - Se utilizó en tarea de traducción: inglés a francés

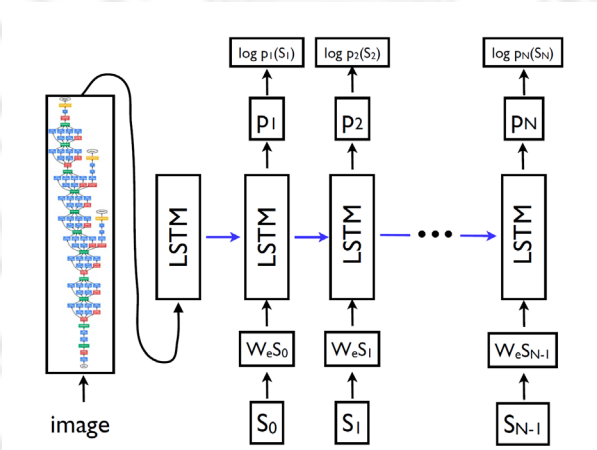


- **Image captioning (Vinyals 2014)**

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).

  - Embedding de imagen
  - Generación de texto

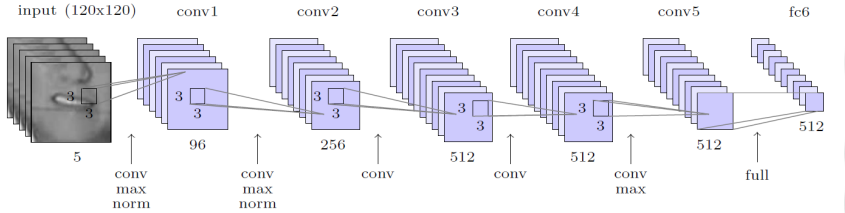




Redes recurrentes

- **Lip reading (Chung 2016) or Watch, Listen, Spell**

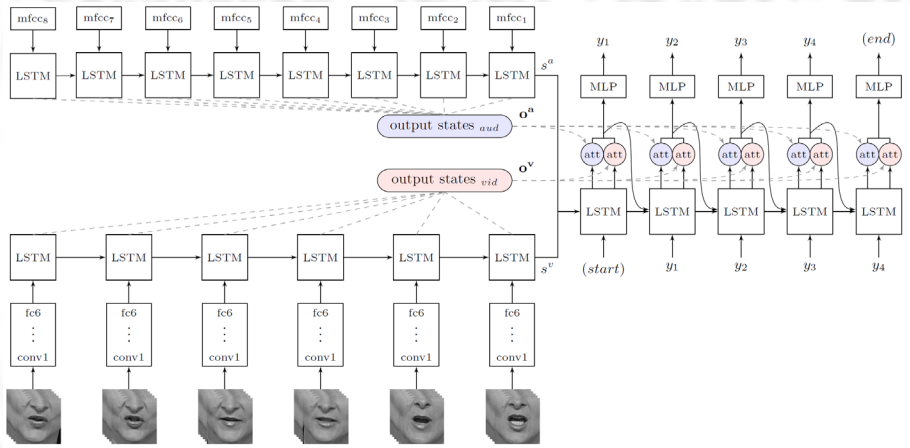
Chung, J. S., Senior, A., Vinyals, O., & Zisserman, A. (2017, July). Lip reading sentences in the wild. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3444-3453). IEEE.



– Parte visual procesado con CNN -> características

– La secuencia de embeddings de audio/video
  - se procesan mediante una LSTM cada una

– La salida se genera mediante una LSTM



Redes recurrentes

- **Ejemplos**

https://colab.research.google.com/drive/1BTRR4llqVmxA-kSGPbTfjVnTRBkl5cXn?usp=sharing

https://colab.research.google.com/drive/1S8N4jeMR49Q1e-SOtYa6mjsPe_yEOHIm?usp=sharing

Bibliografía

- Libros
  - Simon J.D. Prince, “Understanding Deep Learning”, MIT Press Dec 5th 2023.
    - https://udlbook.github.io/udlbook/
  - C. M. Bishop, H. Bishop “Deep Learning: Foundations and Concepts”, Springer 2024
    - https://www.bishopbook.com/
  - I. Goodfellow and Y. Bengio and A. Courville, “Deep Learning”, MIT Press 2016
    - https://www.deeplearningbook.org/
  - Marc P.r Deisenroth, A. Aldo Faisal, and Cheng S. On “Mathematics for Machine Learning”, Cambridge University Press 2020
    - https://mml-book.github.io/
  - Simone Scardapane. “Alice’s Adventures in a differentiable wonderland: A primer on designing neural networks (Volume I)”, arxiv 2024
    - https://www.sscardapane.it/alice-book/