Alianza en Inteligencia Artificial

CONACYT

INTELIGENCIA ARTIFICIAL

Consorcio de Centros Públicos Conacyt

México 2021 Año de la Independencia

Variantes y retos del aprendizaje profundo

Presenta: Hugo Jair Escalante

Mayo 10, 2021

Instituto Nacional de Astrofísica, Óptica y Electrónica

Contenido

- Principales variantes de DL
  - CNNs
  - Variantes de CNNs
  - Autocodificadores
  - Modelos secuenciales
  - Modelos generativos

- Retos, oportunidades de investigación

Aprendizaje profundo
Principales variantes

DNNs

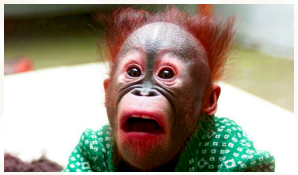
En este ejemplo la red neuronal está definida por las aristas del grafo, esto es, por matrices de pesos: **U** (5x8) y **W** (8x2)

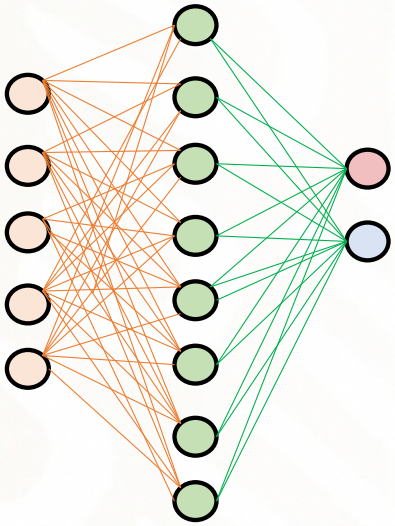
- Hasta ahora, solo hemos revisado las redes neuronales *feedforward*

Variables - U

5x8 double

	1	2	3	4	5	6	7	8	9
1	-0.7203	1.1714	-1.5029	3.3133	0.0338	-0.3660	-0.2886	-1.0836	
2	-0.5102	0.2660	0.3880	0.8725	-0.5068	-1.6706	1.5959	1.0228	
3	-1.6597	-0.5933	-0.4178	1.8309	1.0331	0.1132	-0.3820	1.0903	
4	0.3005	0.3909	-0.2304	-2.3196	3.2656	-0.0441	0.4254	0.0461	
5	0.2737	-0.6014	0.0314	0.5855	0.3692	2.1029	1.5813	-0.9019	
6									
7									
8									
9									
10									
11									





Outputs

Variables - W

8x2 double

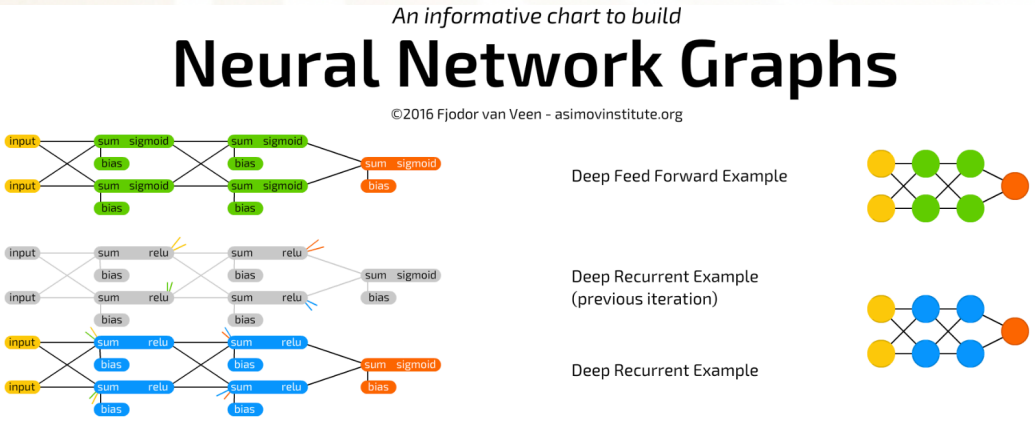
	1	2	3
1	0.5884	0.1427	
2	0.9775	0.6085	
3	0.2137	0.6286	
4	0.6291	0.0238	
5	0.7810	0.7787	
6	0.3571	0.4912	
7	0.9407	0.4116	
8	0.9616	0.7775	
9			
10			
11			
12			
13			

$$f(\mathbf{x}) = s(\mathbf{w}\phi(\mathbf{x}) + b)$$

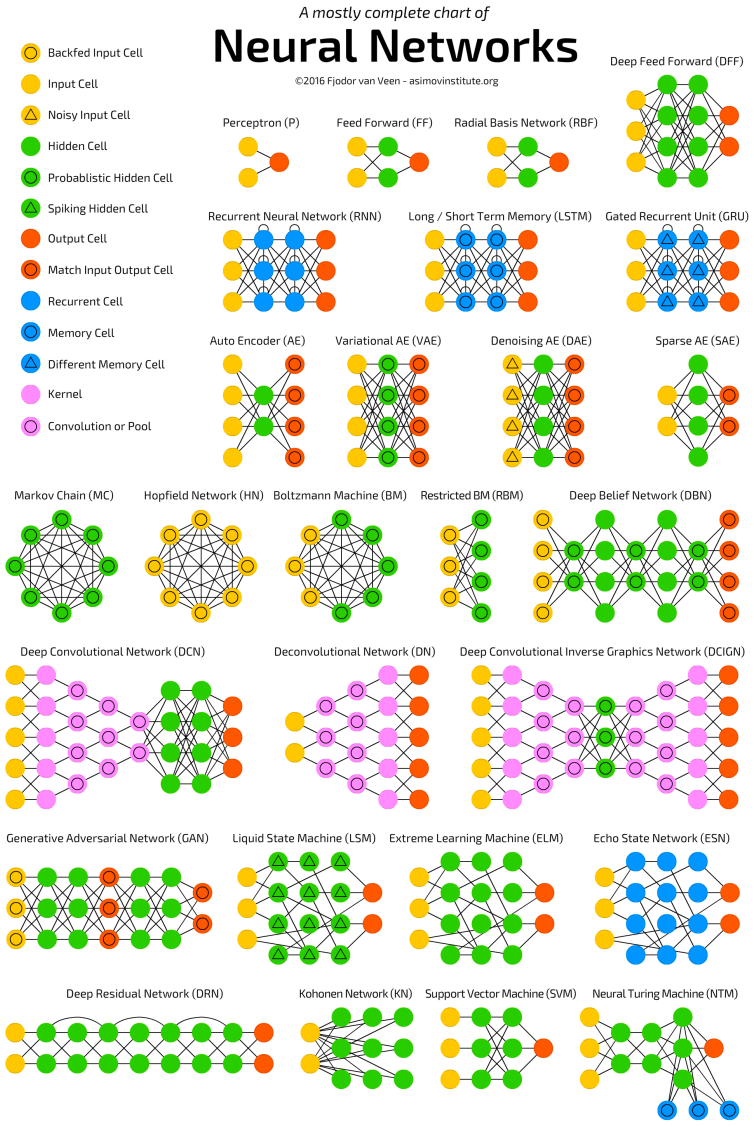
Variantes de aprendizaje profundo

- Otras variantes (modelos) de aprendizaje profundo se construyen cambiando:
  - El tipo de unidades que se utilizan
  - La forma en que se arreglan dichas unidades
  - La forma en que fluye la información
  - La tarea
  - Etc.

DNNs



https://www.asimovinstitute.org/author/fjodorvanveen/



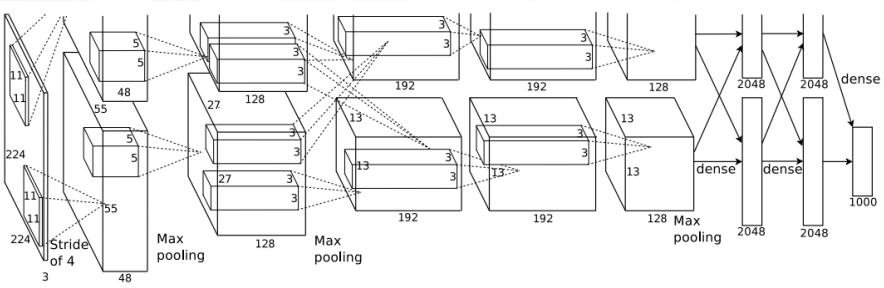
Variantes de aprendizaje profundo

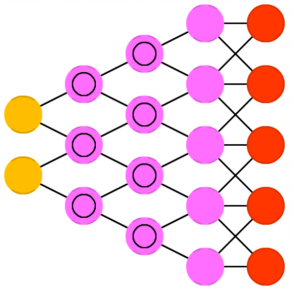
- Principales modelos de DL
  - Redes neuronales profundas (DNNs, MLPs)
  - Redes neuronales convolucionales (CNNs)
  - Autoencoders (AEs)
  - Long short term memory networks (LSTMs)

- Otros paradigmas
  - Restricted Boltzman machines
  - Deep belief networks
  - Redes residuales
  - Inception networks
  - Gated recurrent NNs
  - Generative adversarial networks
  - Transformers

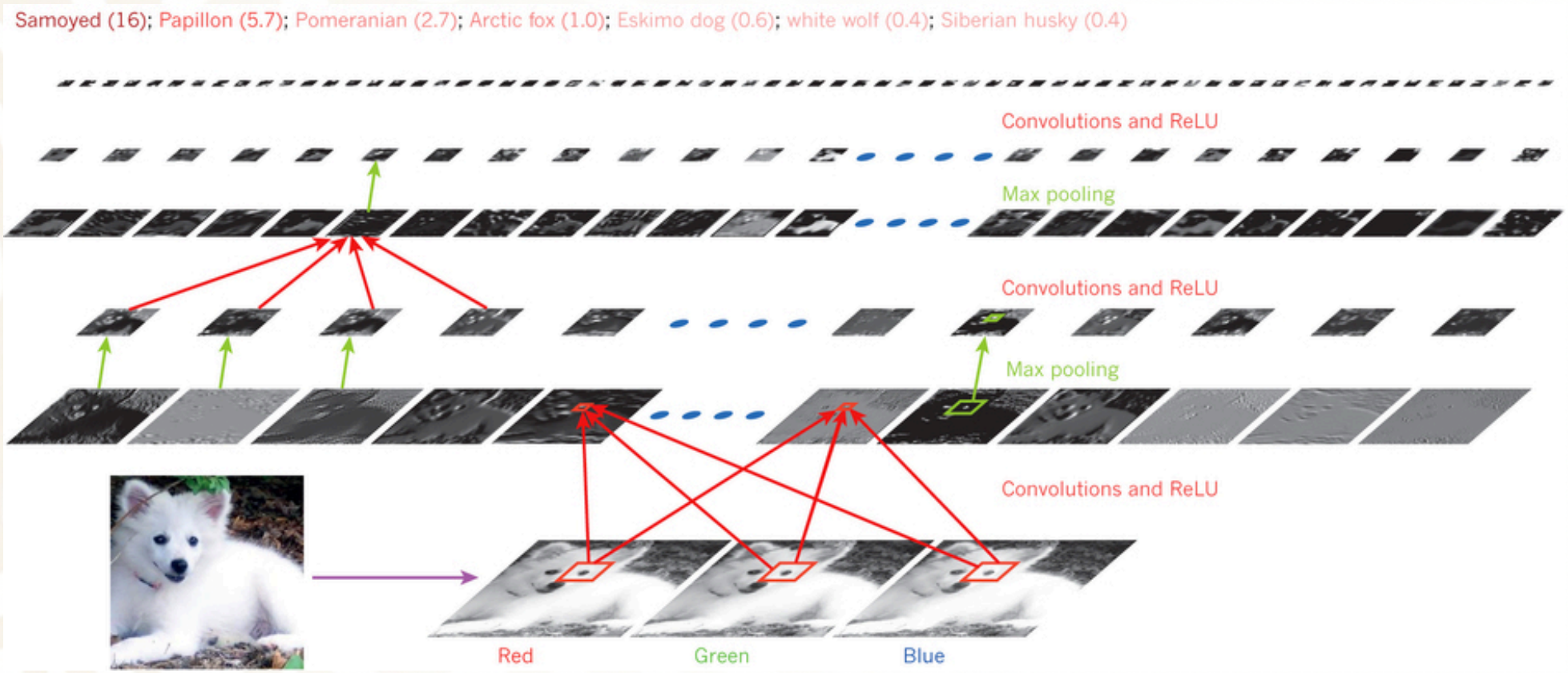
CNNs – redes neuronales convolucionales

- Tipo de red para procesar datos arreglados en un celdas
  - Series de tiempo, texto (Celda 1D)
  - Imágenes (Celda 2D)
  - Videos (Celdas 3D)
- Componentes: capas convolucionales, unidades de activación, capas de *pooling*
- Pesos se aprenden con retro-propagación



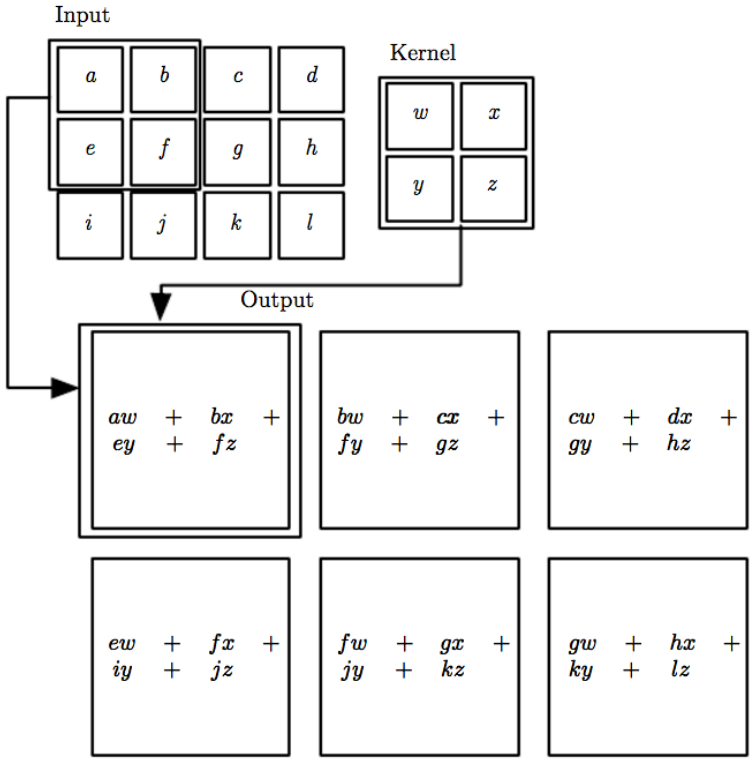


CNNs


Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)
Convolutions and ReLU
Max pooling
Convolutions and ReLU
Max pooling
Convolutions and ReLU
Red
Green
Blue

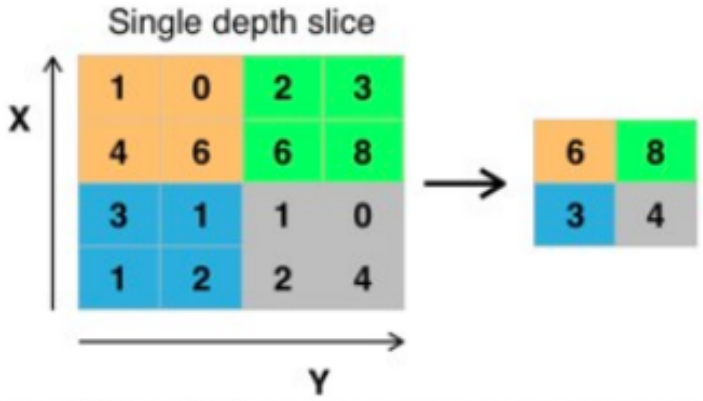
CNNs - componentes

- Convolución 2D

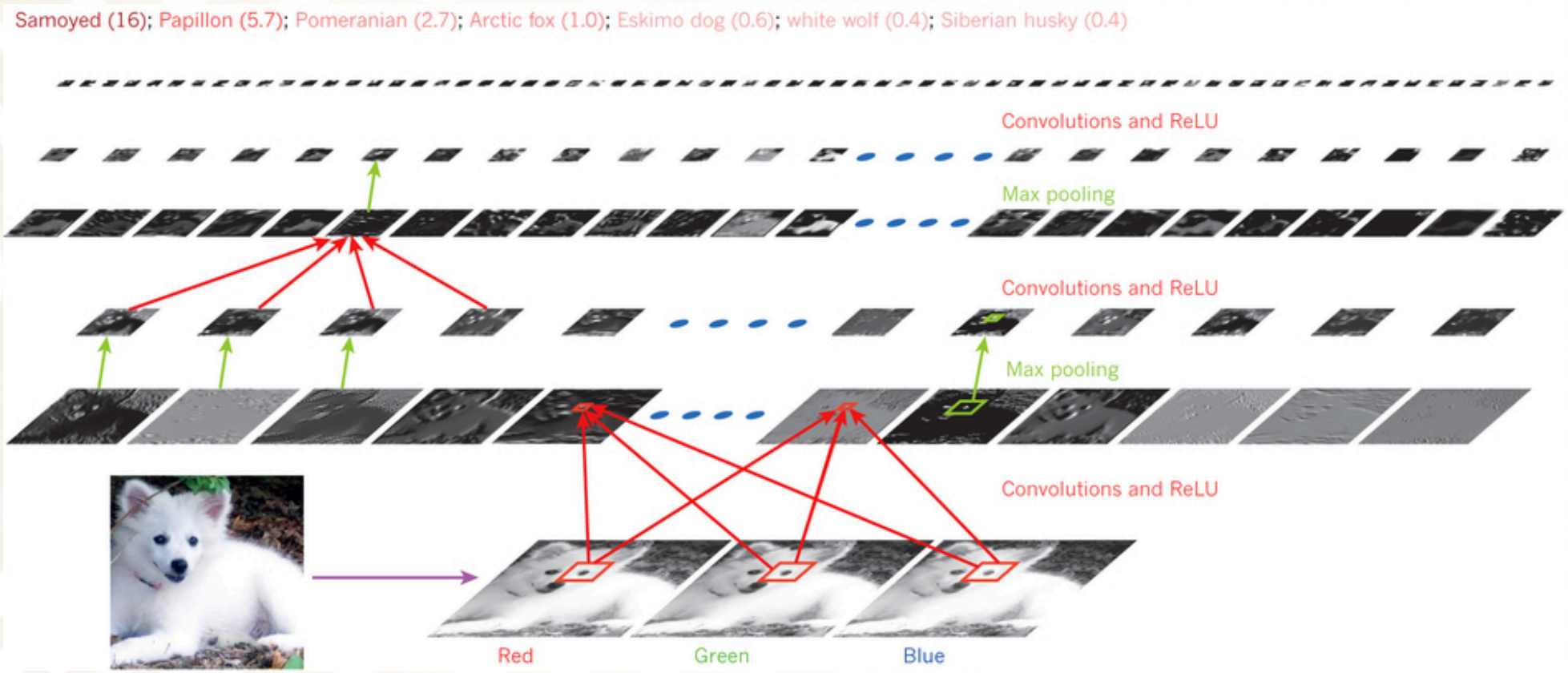


CNNs- componentes

- **Pooling:** remplaza la salida de una red en cierta posición con estadísticos que resumen la información de salidas cercanas
  - Hace a la respuesta invariante a traslación
  - Variantes: max, sub, avg.
  - Usualmente se consideran saltos



CNNs


Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)
Convolutions and ReLU
Max pooling
Convolutions and ReLU
Max pooling
Convolutions and ReLU
Red
Green
Blue

CNNs

- Redes que usan la operación de convolución
  - CNNs son NNs que usan convolución en lugar de multiplicación de matrices en al menos una de sus capas

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$
$$= \int_{-\infty}^{\infty} f(t-\tau)\, g(\tau)\, d\tau.$$

$$s(t) = \int x(a) w(t-a) da$$

$$s(t) = (x * w)(t)$$

CNNs

- Convolución en terminología de NNs
  - x – entrada
  - w –kernel o filtro
  - s –feature map

$$s(t) = (x * w)(t)$$

$$s(t) = \int x(a)w(t-a)da$$

- Convolution discreta:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

CNNs

- Convolution 2D

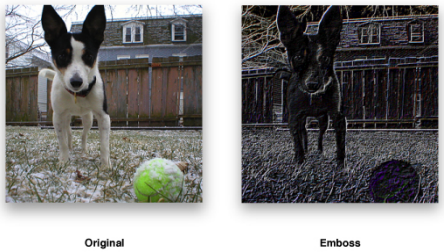
$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n).$$

- La convolución es conmutativa

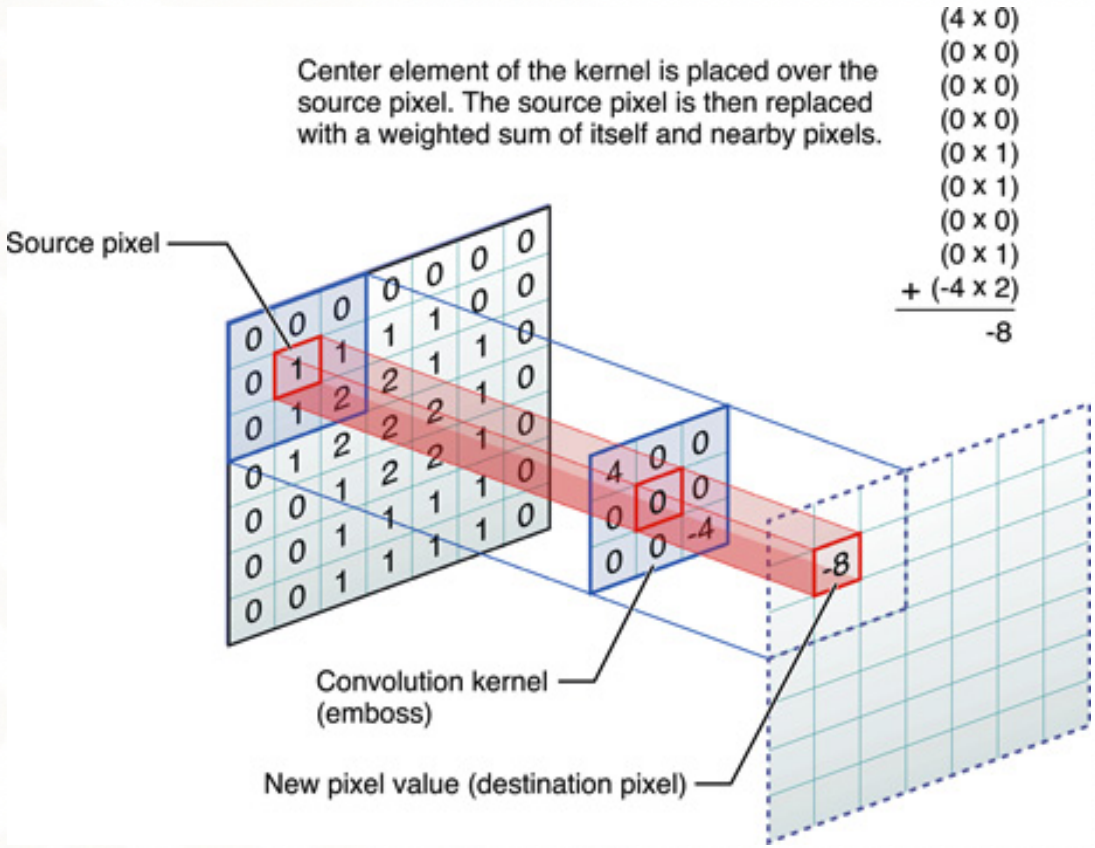
$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n).$$

CNNs

- Convolución 2D

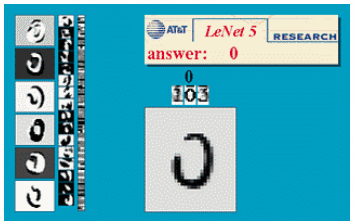


-2	-2	0
-2	6	0
0	0	0



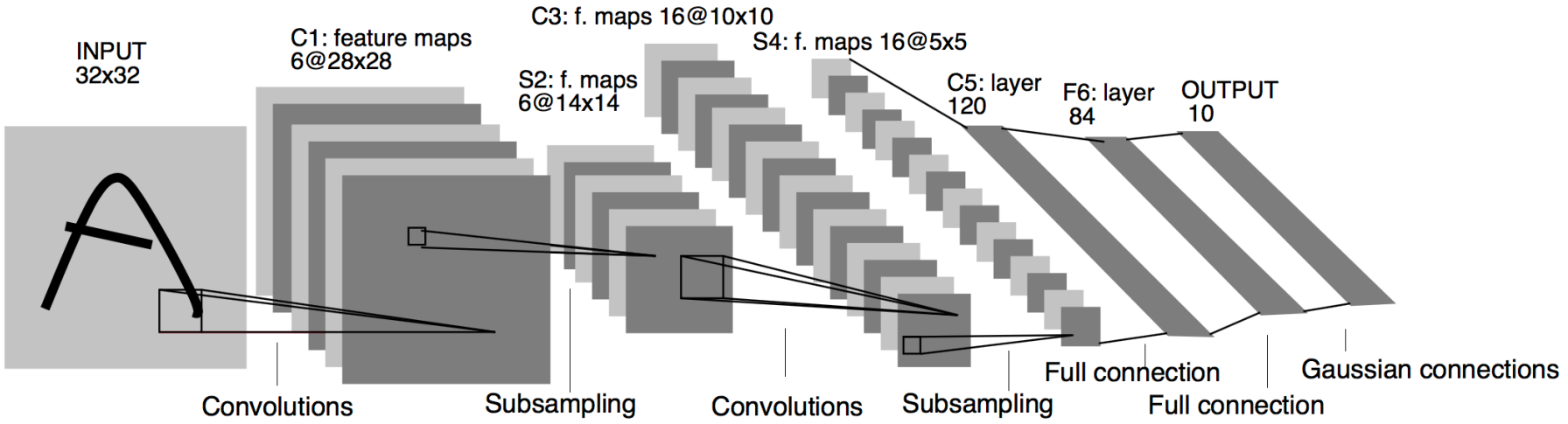
https://developer.apple.com/library/content/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html

CNNs



- Arquitectura típica I

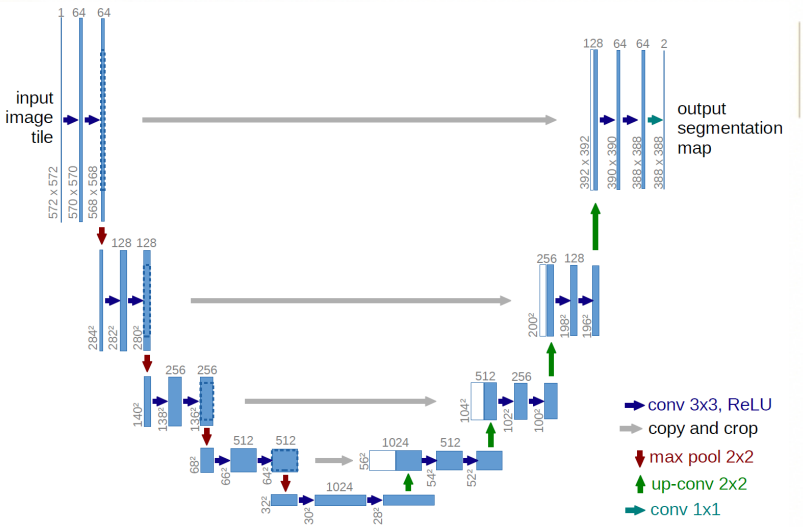
http://yann.lecun.com/exdb/lenet/

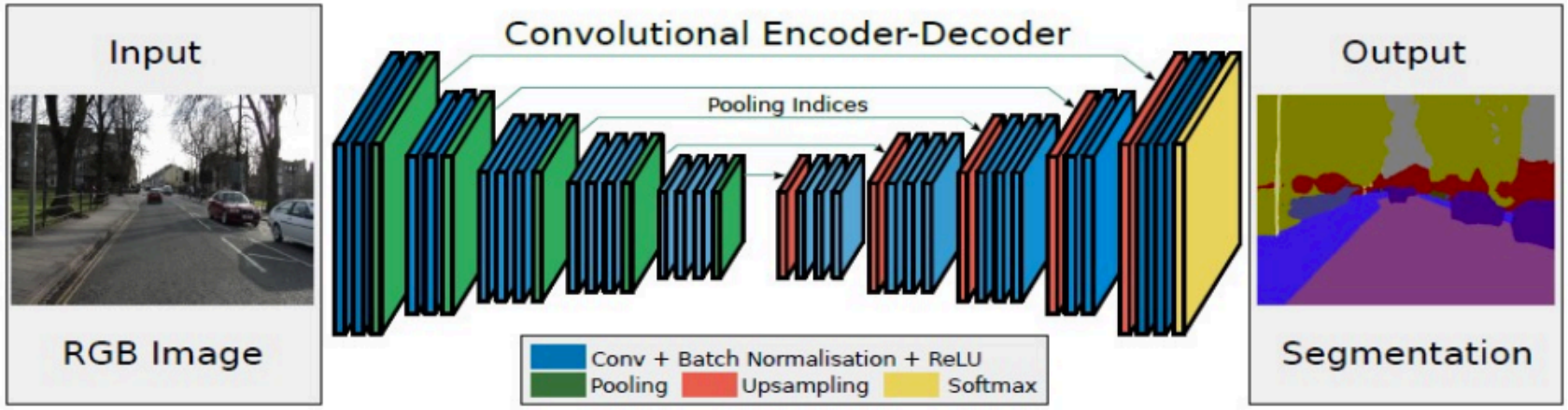


Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, november 1998.

CNNs

- Arquitectura típica II



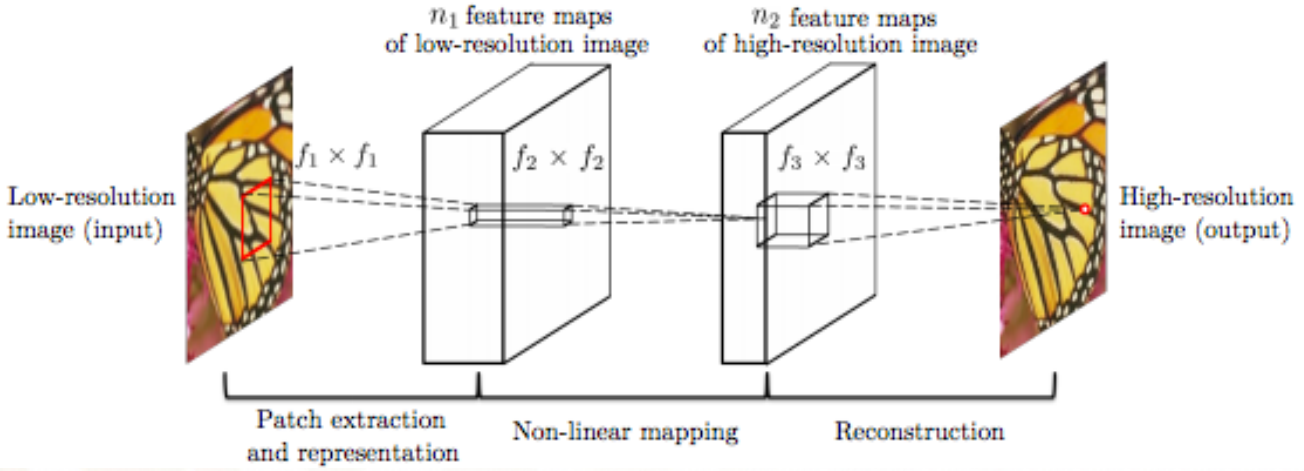


Vijay Badrinarayanan, et al. . **SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,** *ArXiv, 1511.00561, 2015*

Olaf Ronneberger et al. U-Net: **Convolutional Networks for Biomedical Image Segmentation** Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015,

CNNs

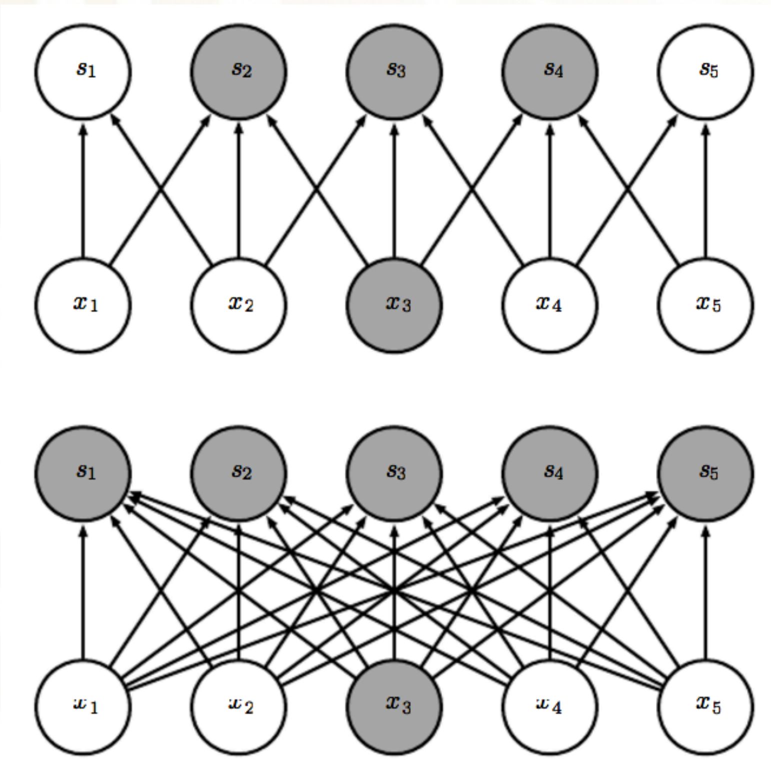
- Arquitectura típica II

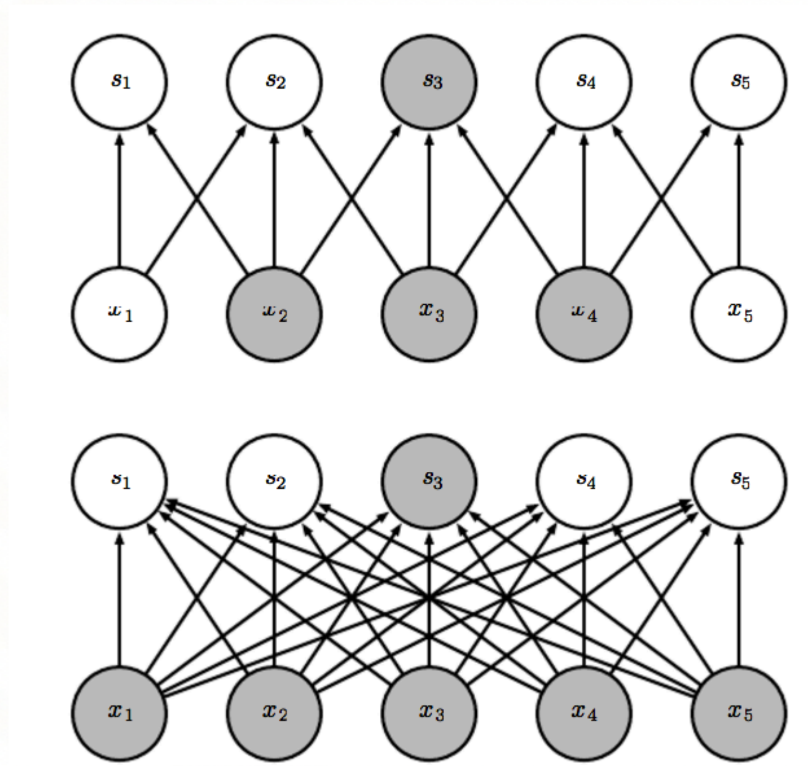


C. Dong, C. C. Loy, K. He, and X. Tang, "**Image super-resolution using deep convolutional networks**," CoRR, vol. abs/1501.00092, 2015.

CNNs

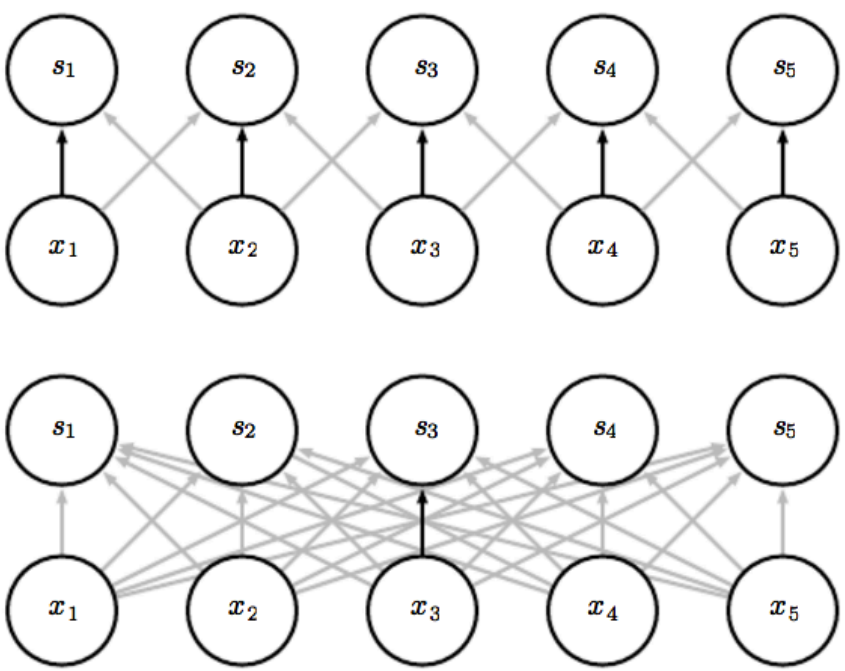
- Por qué usar capas convolucionales?
  - *Sparse connectivity*: menos pesos/parámetros





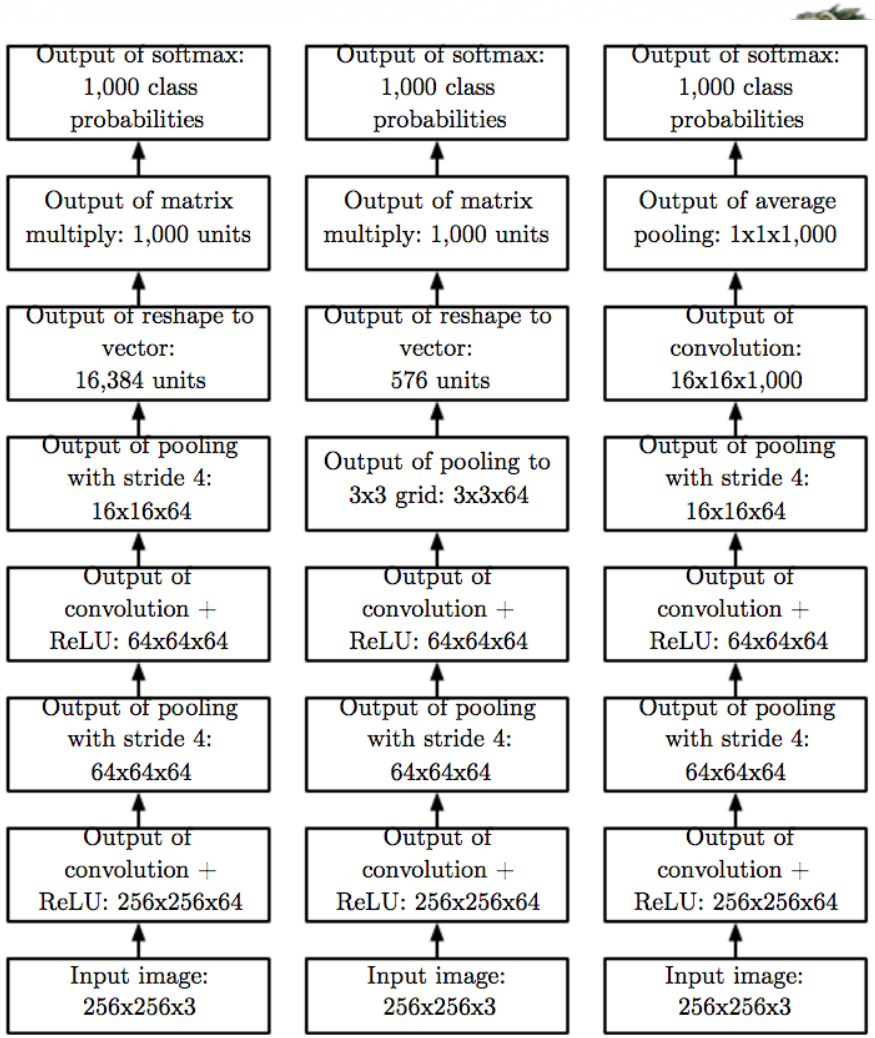
CNNs

- Por qué usar capas convolucionales?
  - *Parameter sharing: menos pesos / parámetros*



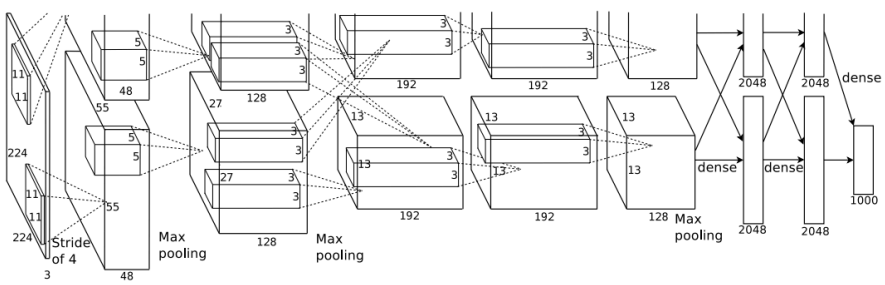
CNNs

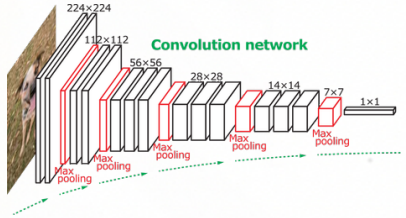
- Arquitecturas comunes



CNNs

- Qué tan profundo?

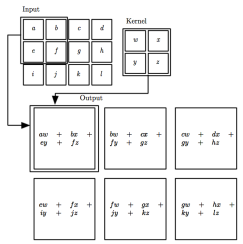


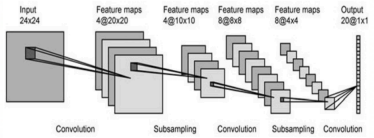


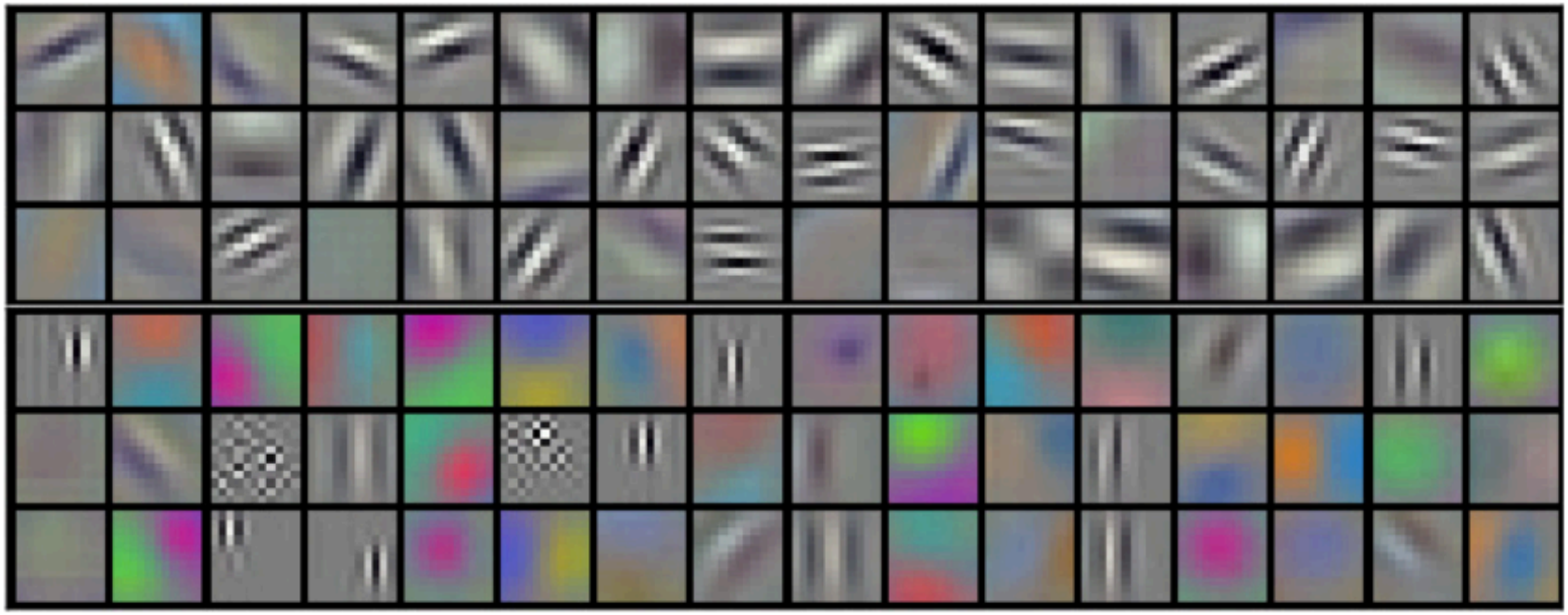
Input	: Image input
Conv	: Convolutional layer
Pool	: Max-pooling layer
FC	: Fully-connected layer
Softmax	: Softmax layer

CNNs

- Qué son los pesos aprendidos?

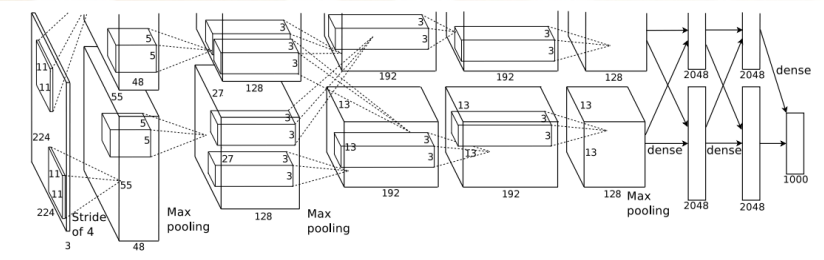


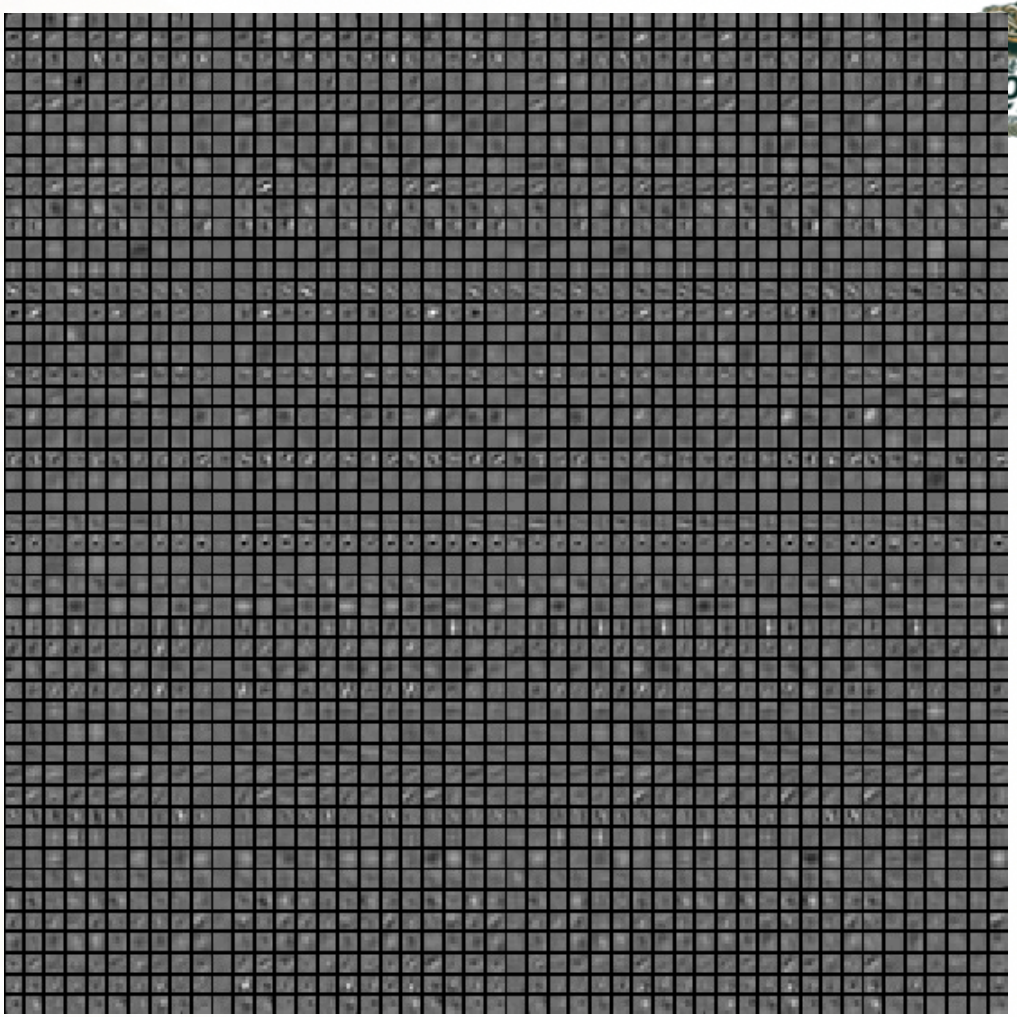




CNNs

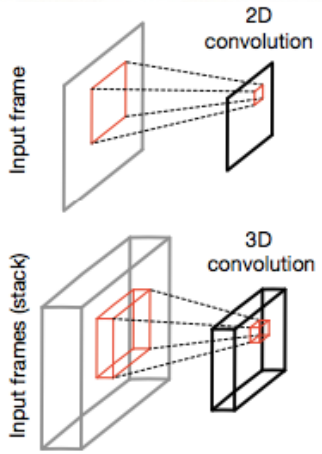
- Qué son los pesos aprendidos?

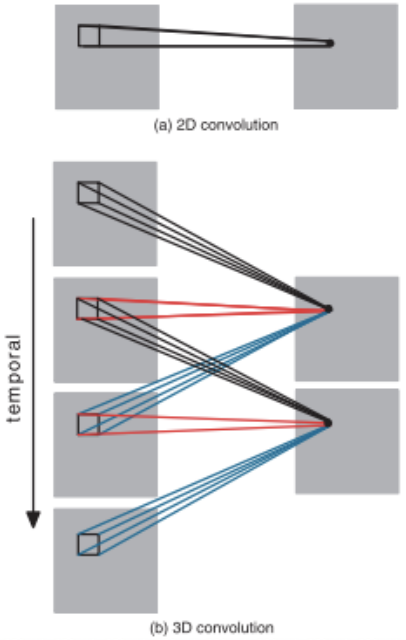




CNNs

- Qué pasa con 3D?





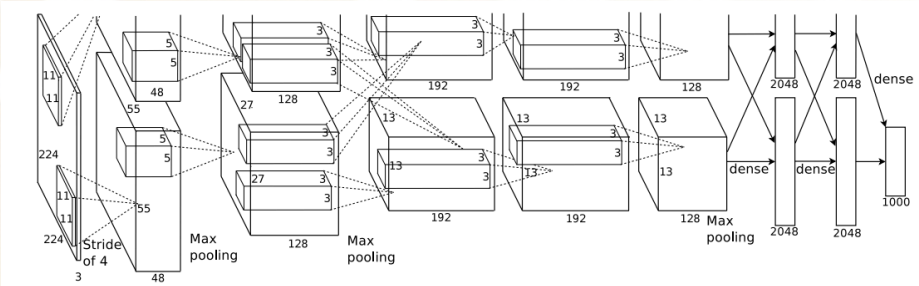
Maryam Asadi-Aghbolaghi, Albert Clapés, Marco Bellantonio, Hugo Jair Escalante, Víctor Ponce-López, Xavier Baró, Isabelle Guyon, Shohreh Kasaei, Sergio Escalera. **A survey on deep learning based approaches for action and gesture recognition in image sequences.** Proceedings of the 12th IEEE Conference on Automatic Face and Gesture Recognition, 2017

Variantes de CNNs

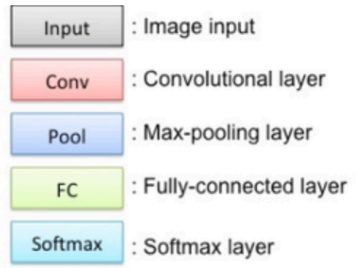
CNNs

- Dos modelos representativos

AlexNet

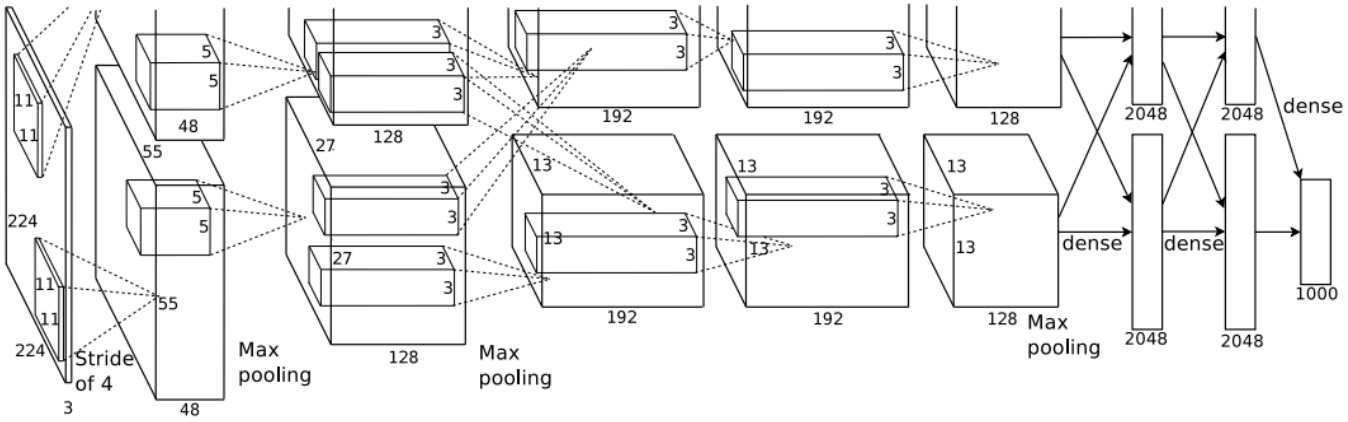


VGG



AlexNet

- En 2020, Krizhevsky et al. lograron entrenar una red convolucional usando ~1 millón de imágenes para enfrentar el concurso *ImageNET large scale classification challenge* (1000 categorías, millones de imágenes)



IMAGENET

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. **ImageNet Classification with Deep Convolutional Neural Networks.** Advances in Neural Information Processing Systems 25 (NIPS 2012)

VGG

- Arquitectura de CNN más profunda que remplaza pocas capas de filtros grandes for más capas pero de filtros más pequeños
  - Equivalencia del filtro receptor
- Muy efectivas, competitivas con modelos más complejos

K. Simonyan, A. Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition International Conference on Learning Representations, 2015

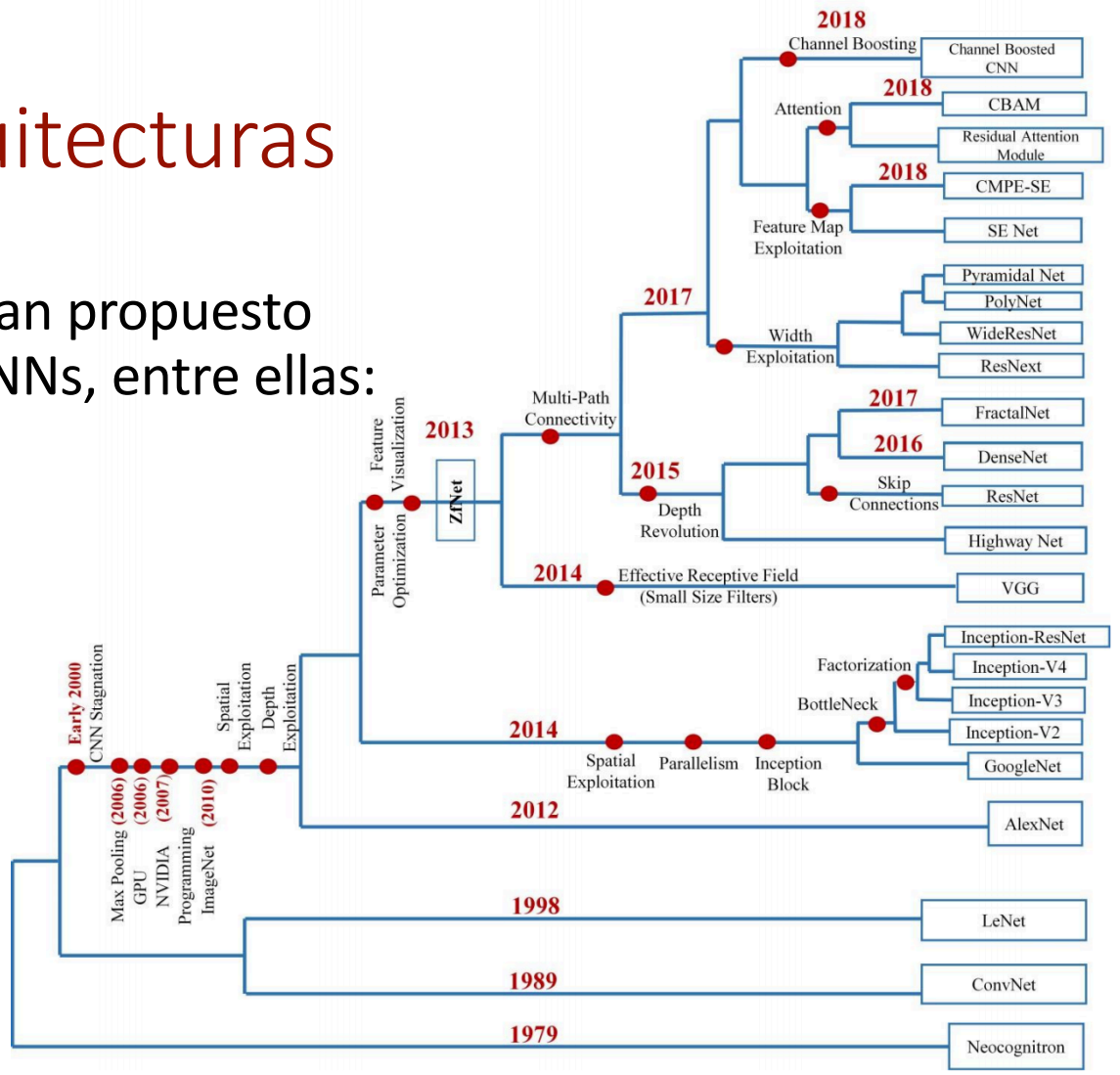
http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf

Primeras arquitecturas CNN

- Fundaron las bases para arquitecturas más modernas y complejas
- Implementaban ya los principales componentes de soluciones efectivas basadas en aprendizaje profundo
  - Big data, modelos muy complejos, uso de GPUs, entrenamiento eficiente, regularización, etc.
- AlexNet y VGG son aún los modelos pre entrenados más útilizados en aplicaciones de CNNs con imágenes
- Su desarrollo y establecimiento fue motivado en gran parte por nua competencia académica (ImageNet challenge)

Principales arquitecturas

- Además de LeNet, se han propuesto muchas variantes de CNNs, entre ellas:
  - GoogLeNet
  - DenseNet
  - ResNet
  - …



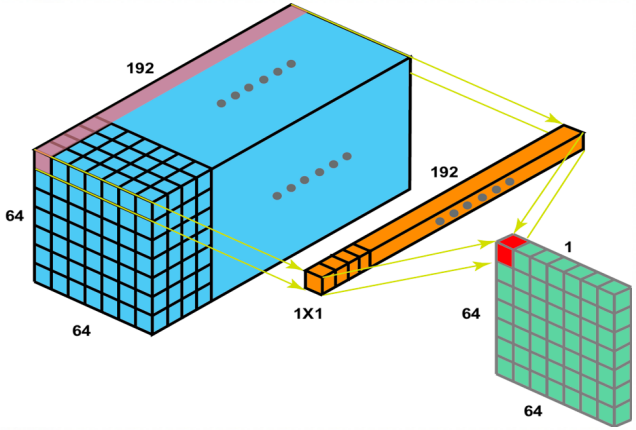
Asifullah Khan et al. A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review (2020)

Inception: networks in networks

- **Idea:** Construir redes de módulos de "incepción", cada uno de estos módulos, realizan múltiples pasos de convolución / pooling en paralelo

- Concepto clave: Convolución 1x1 con filtros de tamaño *c*
  - Activación ReLU
  - Reduce la dimensionalidad de las entradas (*c*)

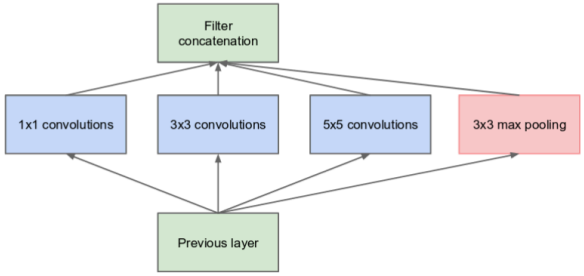
… Inception, which derives its name from the Network in network paper by Lin et al [12] in conjunction with the famous "we need to go deeper" internet meme [1]. …



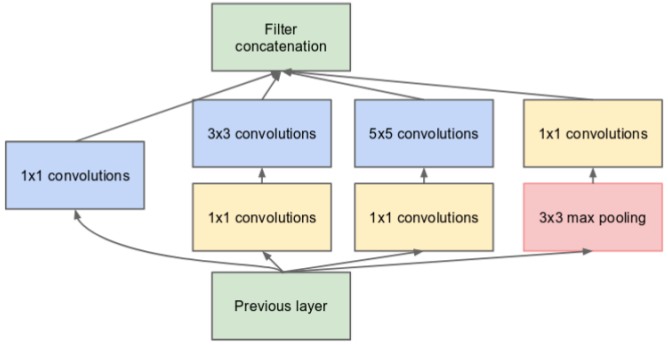
Min Lin, Qiang Chen, Shuicheng Yan. **Network In Network**. ICLR 2014
C. Szegedy et al. Going Deeper with Convolutions. CVPR 2015

Inception: networks in networks

- **Idea:** Construir redes de módulos de “incepción”, cada uno de estos módulos, realizan múltiples pasos de convolución / pooling en paralelo



(a) Inception module, naïve version



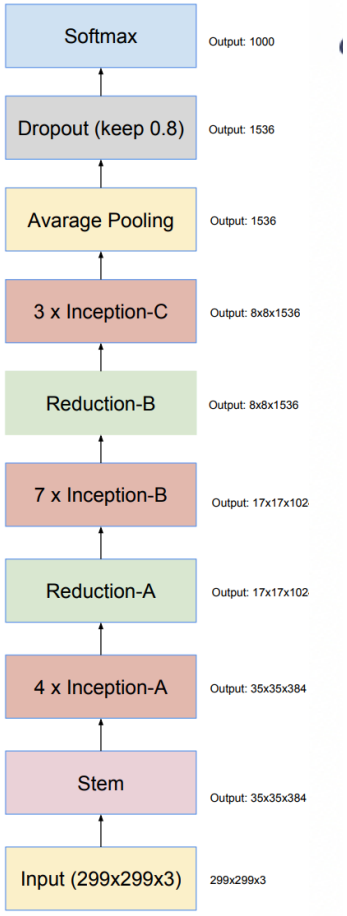
(b) Inception module with dimension reductions

https://www.youtube.com/watch?v=C86ZXvgpejM&t=1s

C. Szegedy et al. Going Deeper with Convolutions. CVPR 2015

GoogLeNet

- Varias versiones de GoogleNet están disponibles:
  - GoogLeNet or Inception-v1
    - Initial model winning the ILSVRC2014 challenge
  - GoogLeNet or Inception-v2
    - Added batch normalization
  - GoogLeNet or Inception-v3
    - Factorization of convolutions
  - GoogLeNet or Inception-v4
    - More tuning?



C. Szegedy et al. Going Deeper with Convolutions. CVPR 2015

Sergey Ioffe, Christian Szegedy . Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift http://arxiv.org/abs/1502.0

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. CVPR 20

Christian Szegedy et al. Inception-v4, inception-ResNet and the impact of residual connections on learning. AAAI'17, 2017

Redes basadas en incepción

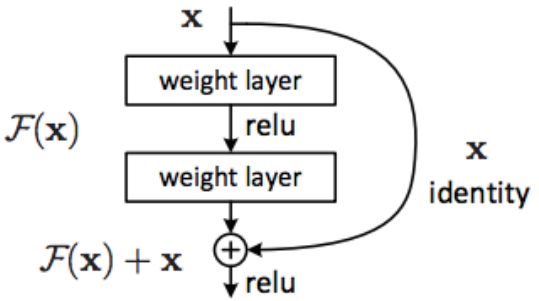
- Procesamiento paralelo
- Procesamiento eficiente, múltiples tamaños de filtros
- Nuevos hallazgos, *trucos*
- Uno de los modelos más efectivos para clasificación de imágenes
- Basado en datos, y desarrollo de prueba y error

ResNet

- **Problema:** Entrenar redes neuronales muy profundas trae varios problemas, uno de los más importantes es el problema de “vanishing/exploiting gradients”

- **Idea:** Construir redes muy profundas, con bloques residuales
  - *Skip connections*

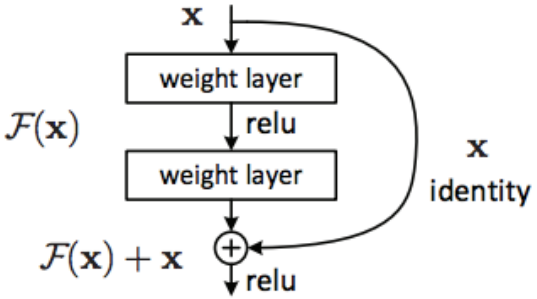
- **Motivación:** El rendimiento de redes muy profundas puede decaer cuamdo se añaden más capas, podría ayudar usar información de capas menos profundas

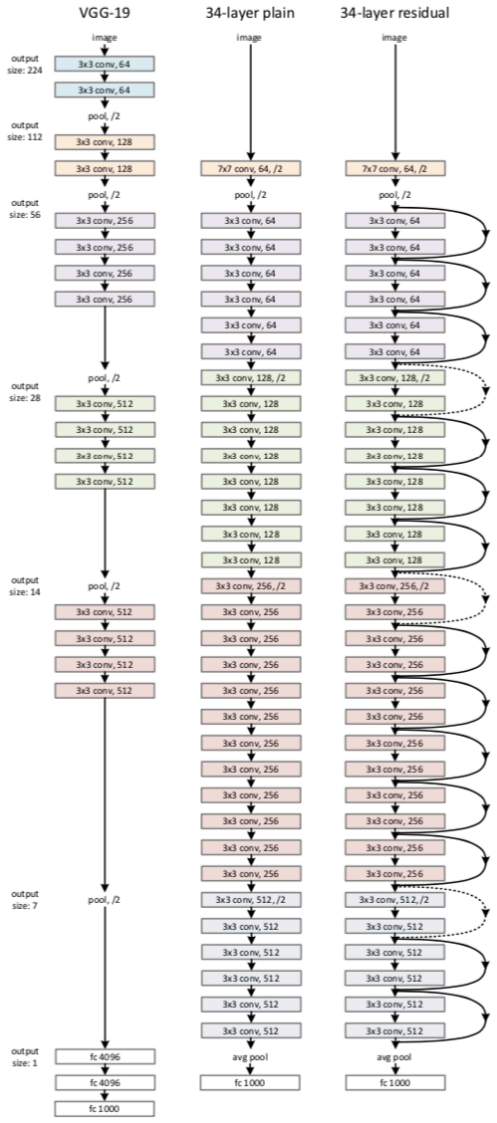


Kaiming He et al., Deep Residual Learning for Image Recognition, CVPR 2016
https://www.youtube.com/watch?v=ZILIbUvp5lk

Aprendizaje residual

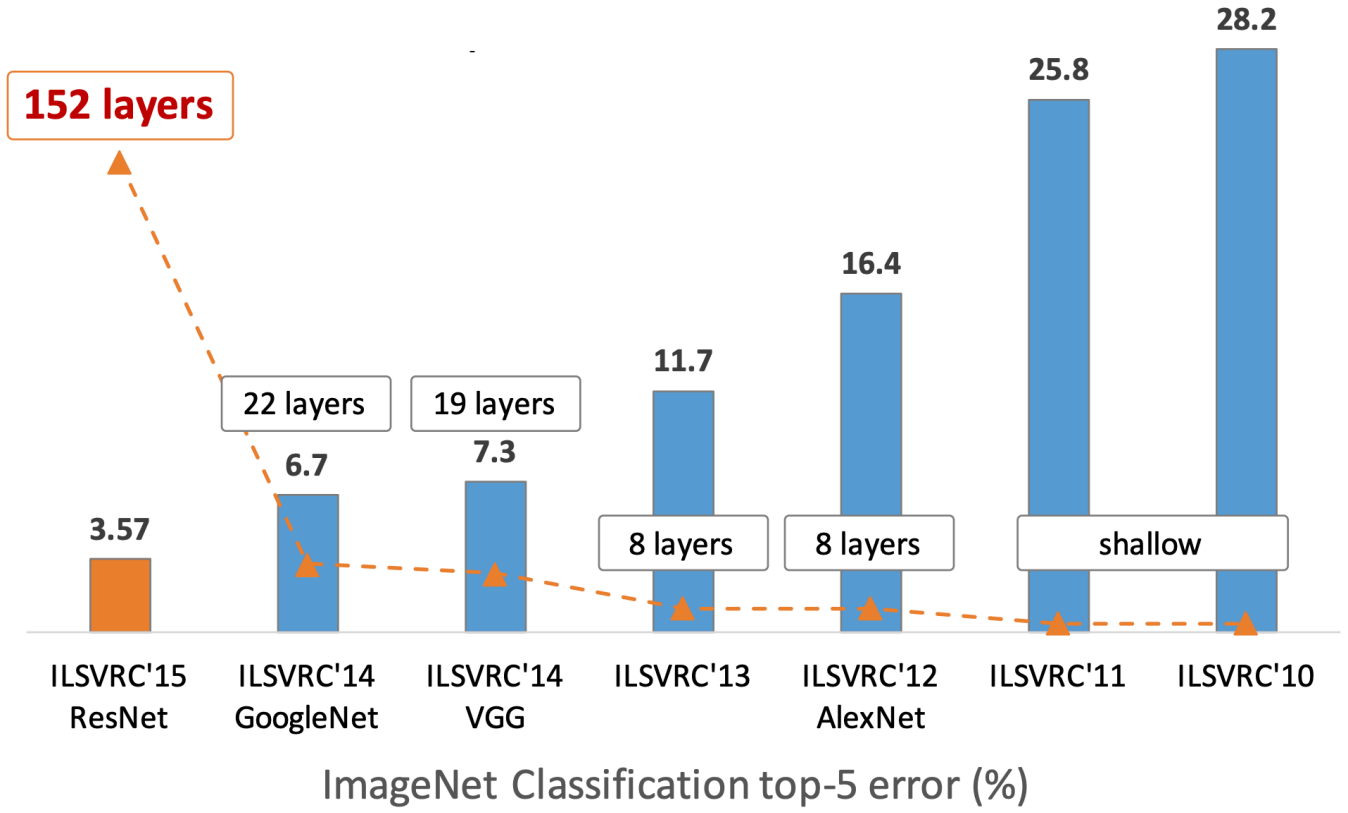
- Idea: Introducir capas que pueden usarse o no!





Kaiming He et al., Deep Residual Learning for Image Recognition, CVPR 2016

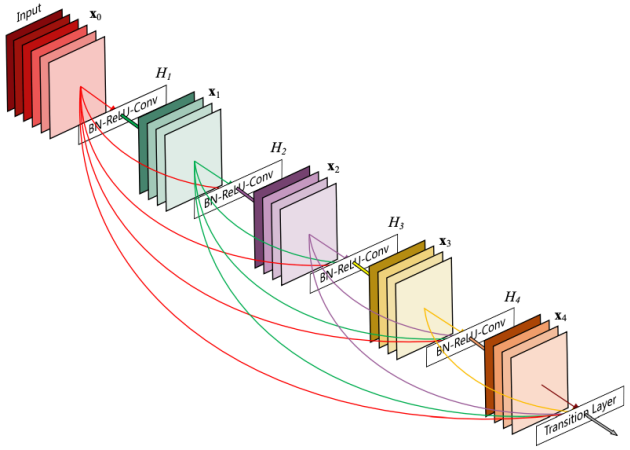
ILSVRC challenge

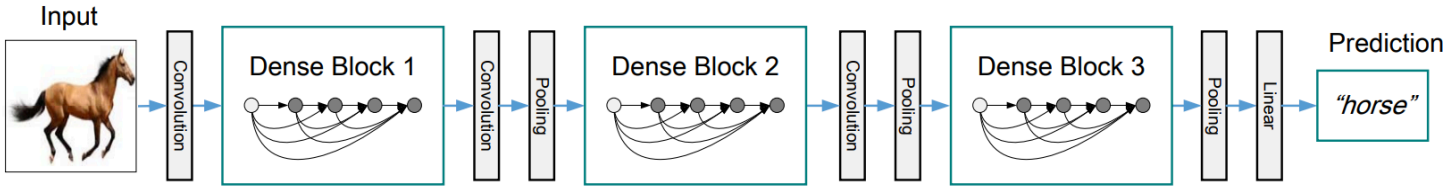


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016

DenseNet

- Conectar capas inferiores con cada una de las capas subsequentes
- La capa l-ésima recibe como entrada la concatenación de todos los feature maps previos
- Ventajas: “alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

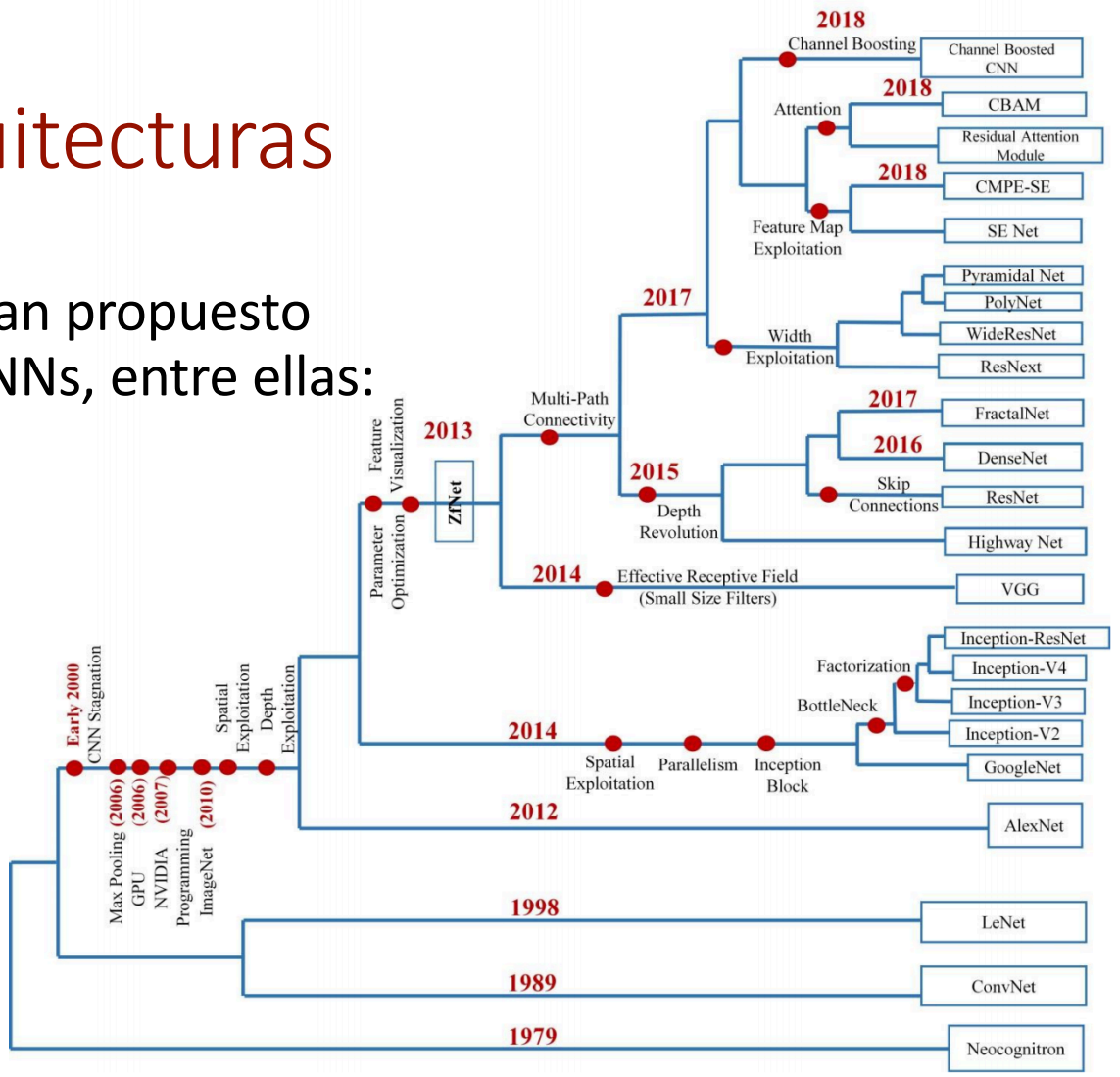




Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. **Densely Connected Convolutional Networks.** CVPR 2017

Principales arquitecturas

- Además de LeNet, se han propuesto muchas variantes de CNNs, entre ellas:
  - GoogLeNet
  - DenseNet
  - ResNet
  - …

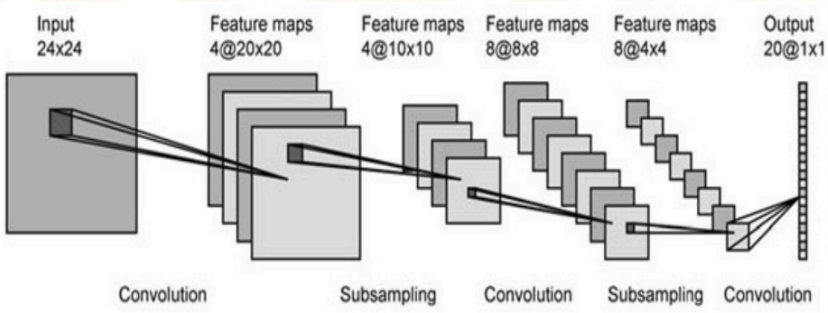


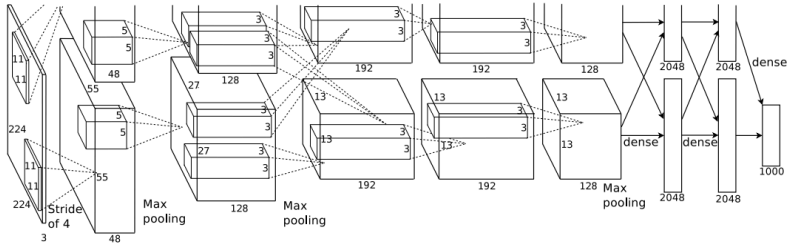
Asifullah Khan et al. A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review (2020)

Re utilizando CNNs

CNNs

- **Transferencia de conocimiento:** Es común re usar arquitecturas pre-entrenadas (pesos) de otras tareas que fueron entrenadas con millones de imágenes
  - Uso directo: usar la red como extractor de características
  - Modelos re ajustados: re usar algunas capas y realizar un ajuste fino para capas de interés

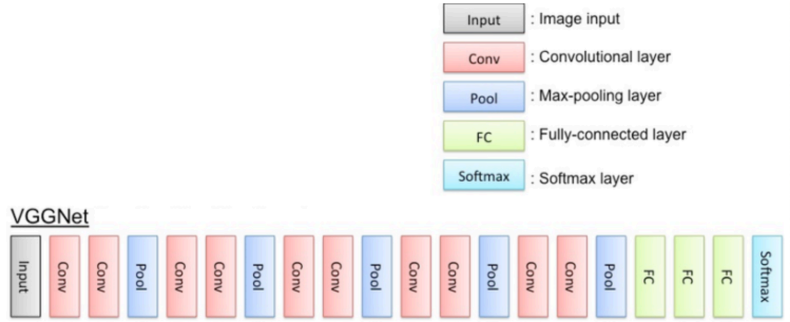


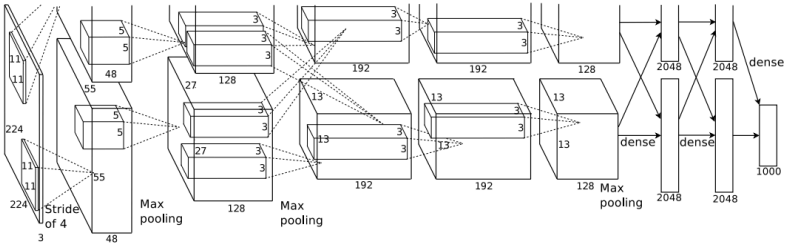


Yosinski J, Clune J, Bengio Y, and Lipson H. **How transferable are features in deep neural networks?** In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation, 2014

CNNs

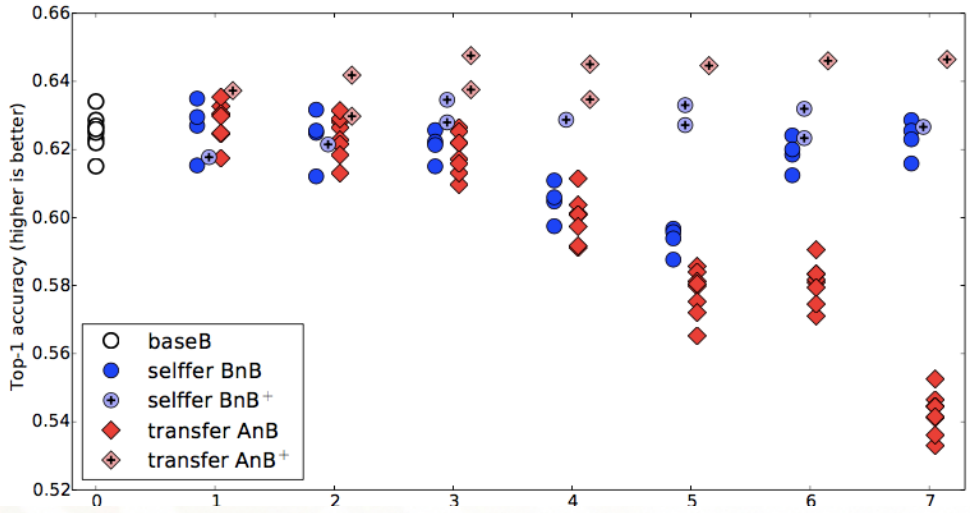
- CNNs pre-entrenadas populares:
  - AlexNet
  - VGG
  - GoogleNet
  - PlacesNet
  - FaceNet





Convolutional neural networks

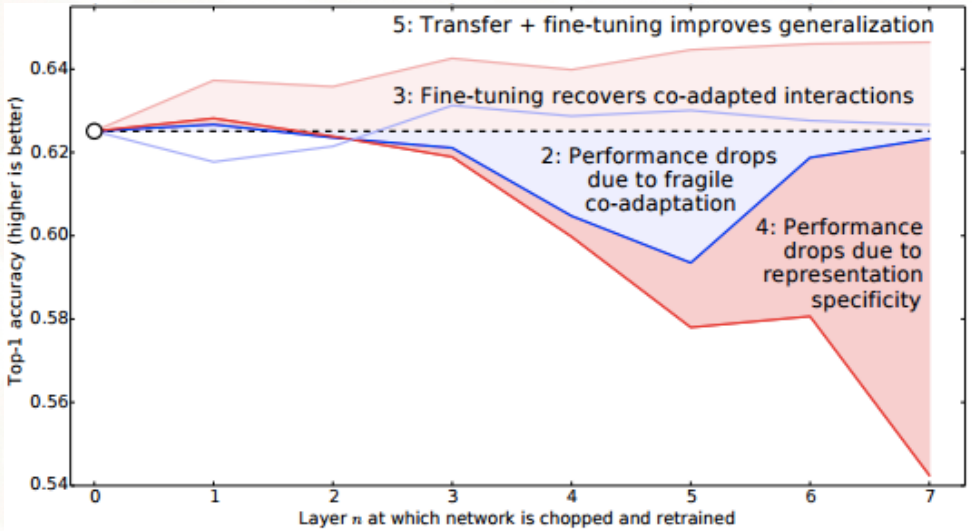
- **Transferencia de conocimiento:** Es común re usar arquitecturas pre-entrenadas (pesos) de otras tareas que fueron entrenadas con millones de imágenes



Yosinski J, Clune J, Bengio Y, and Lipson H. **How transferable are features in deep neural networks?** In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation, 2014

Convolutional neural networks

- **Transferencia de conocimiento:** Es común re usar arquitecturas pre-entrenadas (pesos) de otras tareas que fueron entrenadas con millones de imágenes



Yosinski J, Clune J, Bengio Y, and Lipson H. **How transferable are features in deep neural networks?** In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation, 2014

CNNs

- Los modelos que dominan CV & PR + NLP
  - Desde 2012, aún y cuando Lecun las uso exitosamente para reconocimiento de dígitos en los 80s
- Resultados sobresalientes en muchas tareas, dominios, datos
- El diseño de una CNN es un arte! (deep learning engineering)
- Los modelos se vuelven obsoletos muy rápido, es difícil estar actualizado

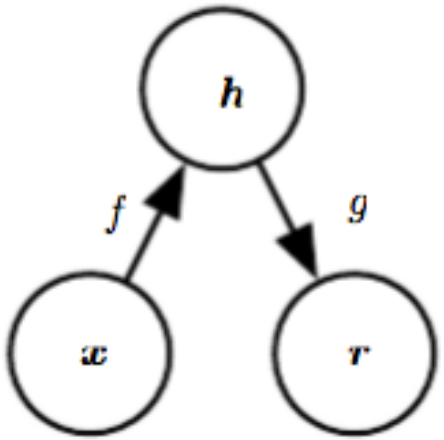
Variantes de aprendizaje profundo

- Principales modelos de DL
  - Redes neuronales profundas (DNNs, MLPs)
  - Redes neuronales convolucionales (CNNs)
  - Autoencoders (AEs)
  - Long short term memory networks (LSTMs)

- Otros paradigmas
  - Restricted Boltzman machines
  - Deep belief networks
  - Redes residuales
  - Inception networks
  - Gated recurrent NNs
  - Generative adversarial networks
  - Transformers

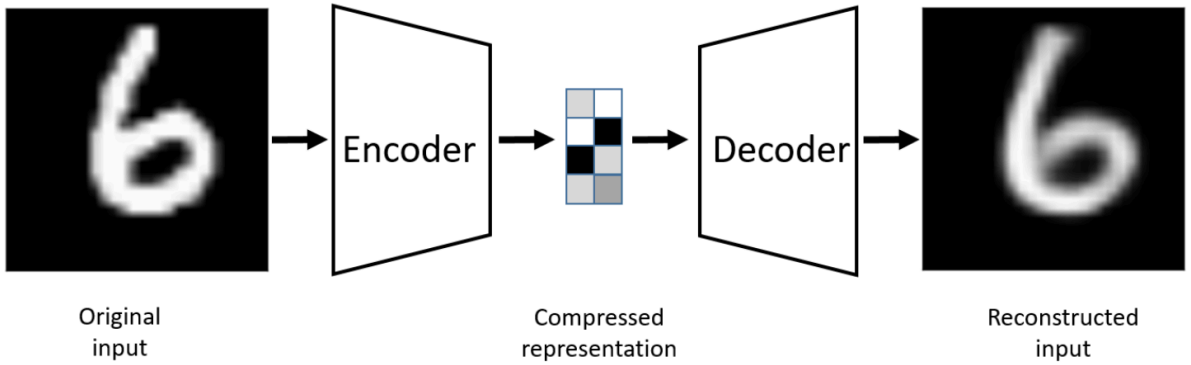
Autoencoders

- Redes neuronales que se entrenan para reproducir las entrada a través de sus salidas
- Una capa código se usa como *pivote*, donde hay capas de parámetros codificadoras y decodificadoras
  - Codificador: $\mathbf{h} = f(\mathbf{x})$
  - Decodificador: $\mathbf{r} = g(\mathbf{h})$
- Usualmente la dimensionalidad de h es menor que la de las entradas



Autoencoders

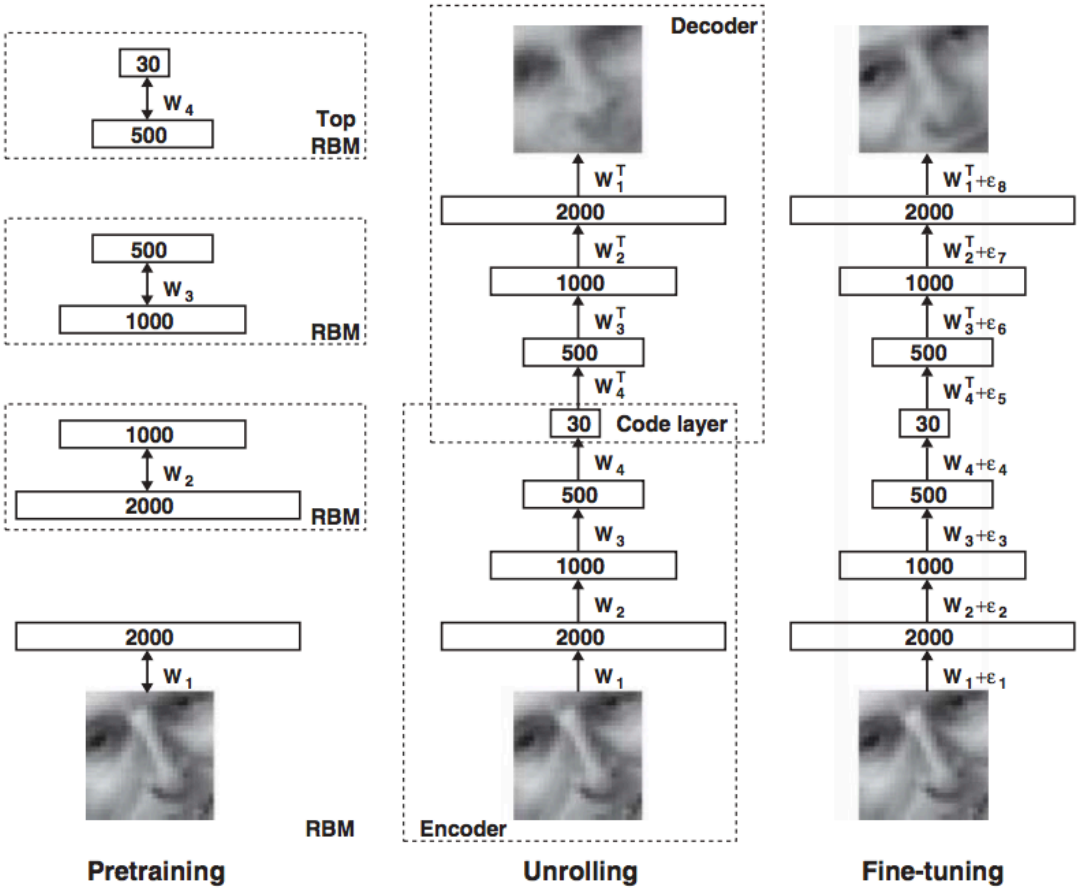
- Pueden ser lineales / no lineales, *sparse* / densos, y se pueden usar para aprendizaje de representaciones, reducción de la dimensionalidad y eliminación de ruido (*denoising*)



Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders, 2020 arxiv.org/abs/2003.05991

Autoencoders

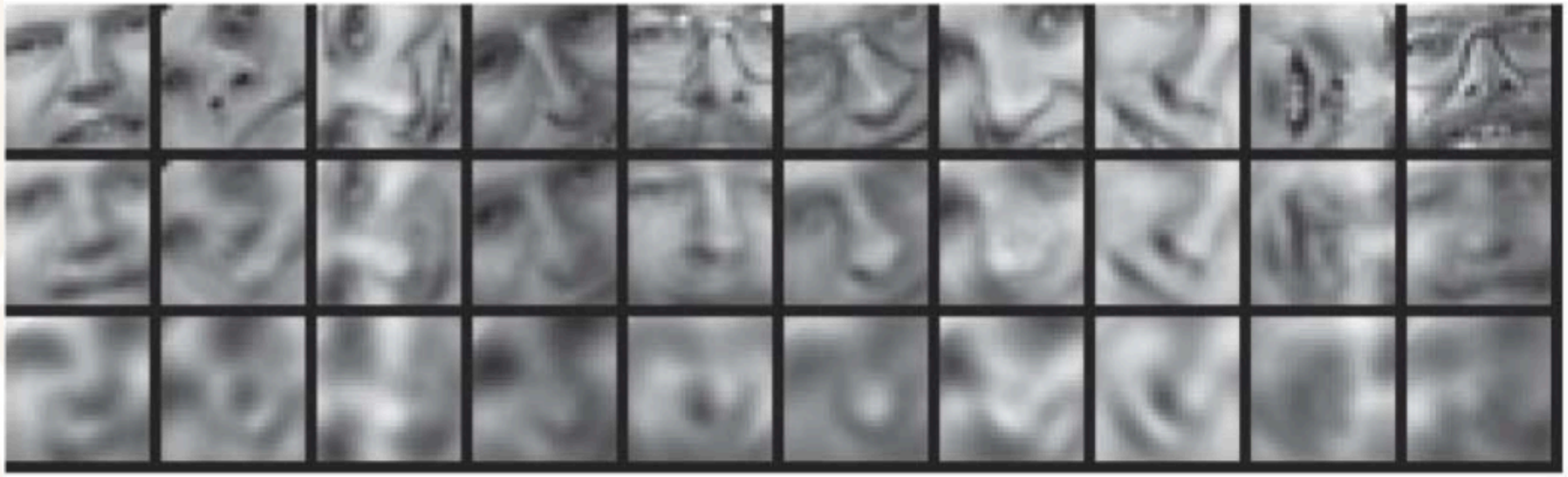
- Autoencoders profundos!
  - Pre entrenamiento de capas usando RBMs
  - Unfolding!
  - Fine tunning con backprop



G. E. Hinton, R. R. Salakhutdinov . **Reducing the dimensionality of data with neural networks.** Science 313, 2016

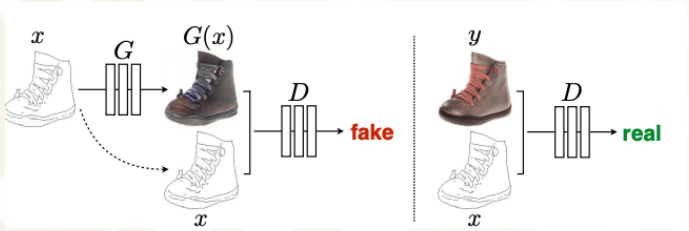
Autoencoders

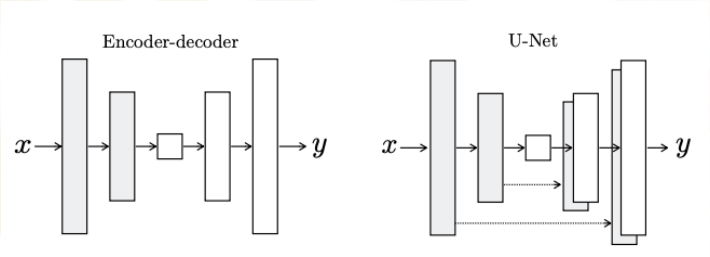
- Autoencoders profundos!



Pix to Pix

https://phillipi.github.io/pix2pix/





Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. **Image-to-Image Translation with Conditional Adversarial Networks.** https://arxiv.org/abs/1611.07004

Variantes de aprendizaje profundo

- Principales modelos de DL
  - Redes neuronales profundas (DNNs, MLPs)
  - Redes neuronales convolucionales (CNNs)
  - Autoencoders (AEs)
  - Long short term memory networks (LSTMs)

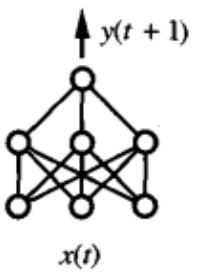
- Otros paradigmas
  - Restricted Boltzman machines
  - Deep belief networks
  - Redes residuales
  - Inception networks
  - Gated recurrent NNs
  - Generative adversarial networks
  - Transformers

Modelando datos secuenciales

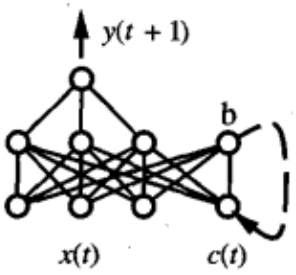
- **Redes neuronales recurrentes.** NNs que reciben como entrada, información de sus salidas

- Las unidades ocultas pueden especificarse como:

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta}),$$

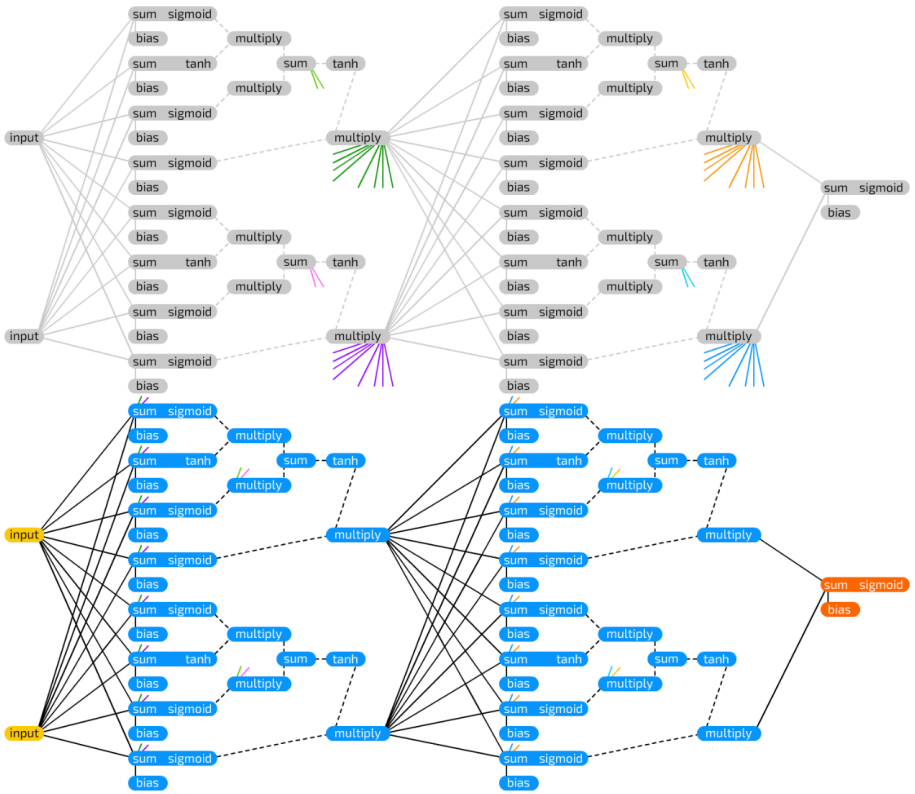


(a) Feedforward network

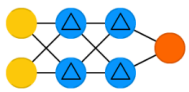


(b) Recurrent network

Modelando datos secuenciales



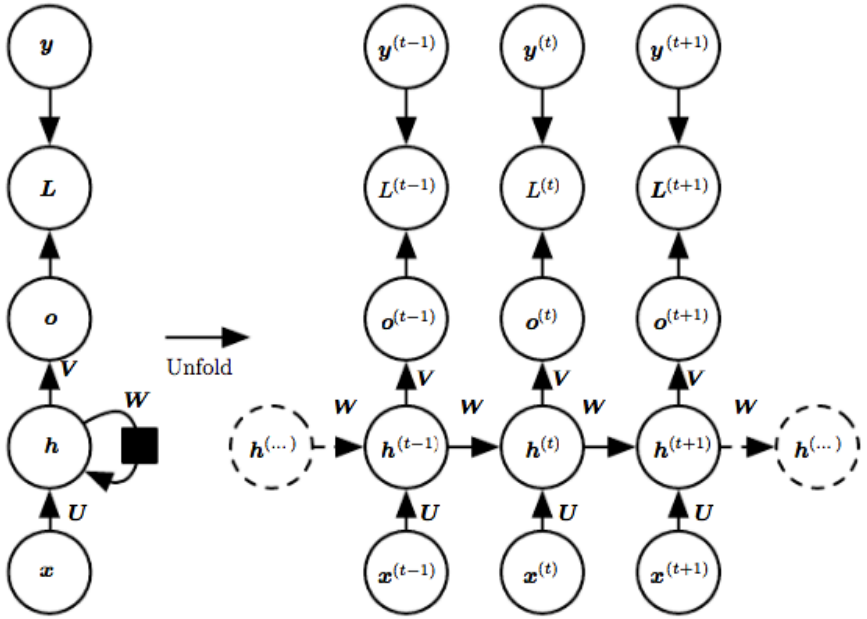
Deep LSTM Example
(previous iteration)



Deep LSTM Example

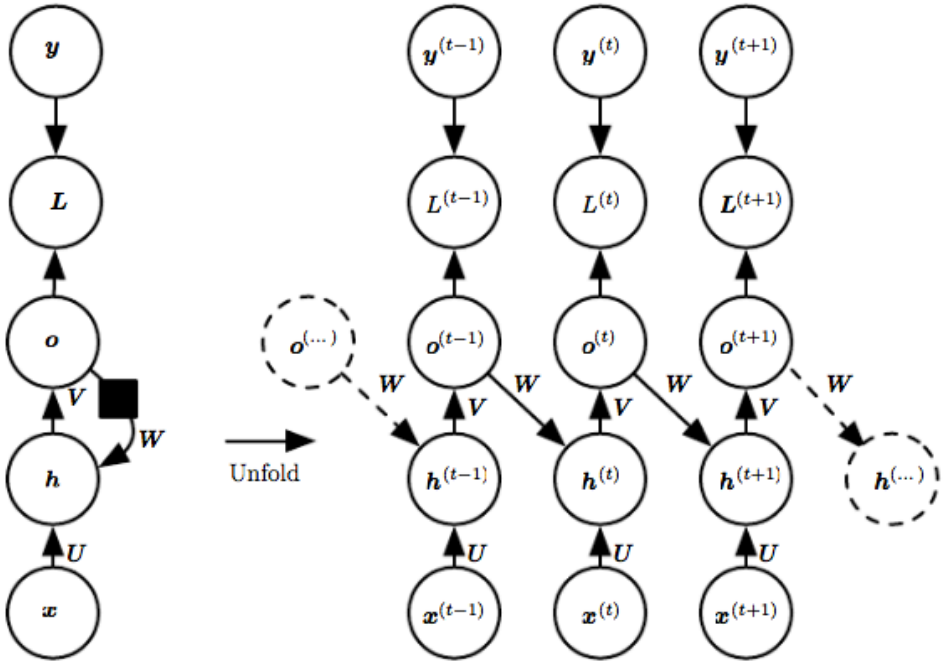
Modelando datos secuenciales

- Arquitectura típica I



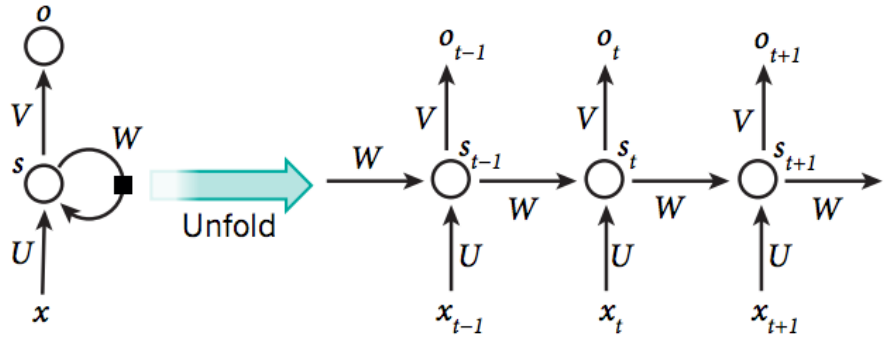
Modelando datos secuenciales

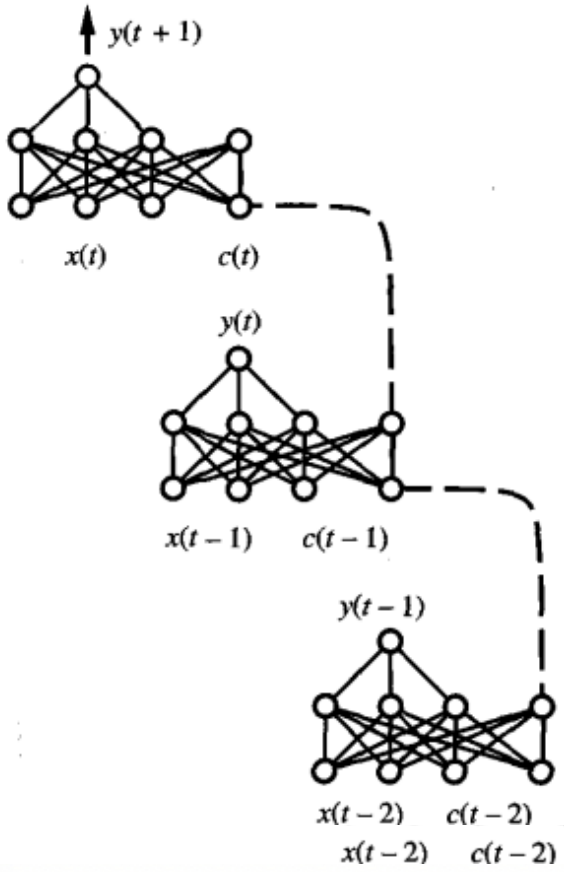
- Arquitectura típica II



Modelando datos secuenciales

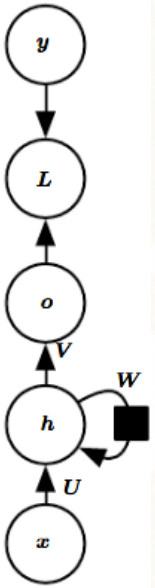
- Entrenamiento de RNNs
  - BPTT: Unfolding + backprop

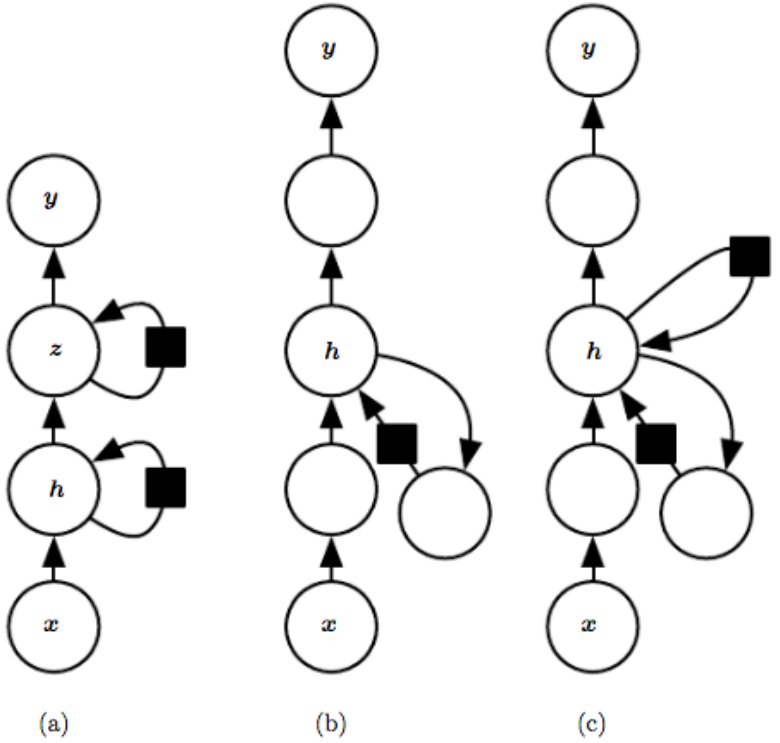




Modelando datos secuenciales

- Going deep with RNNs





Modelando datos secuenciales

- Las RNNs usan diferentes tipos de unidades para modelar información secuencial, y algoritmos de aprendizaje (ligeramente) diferentes

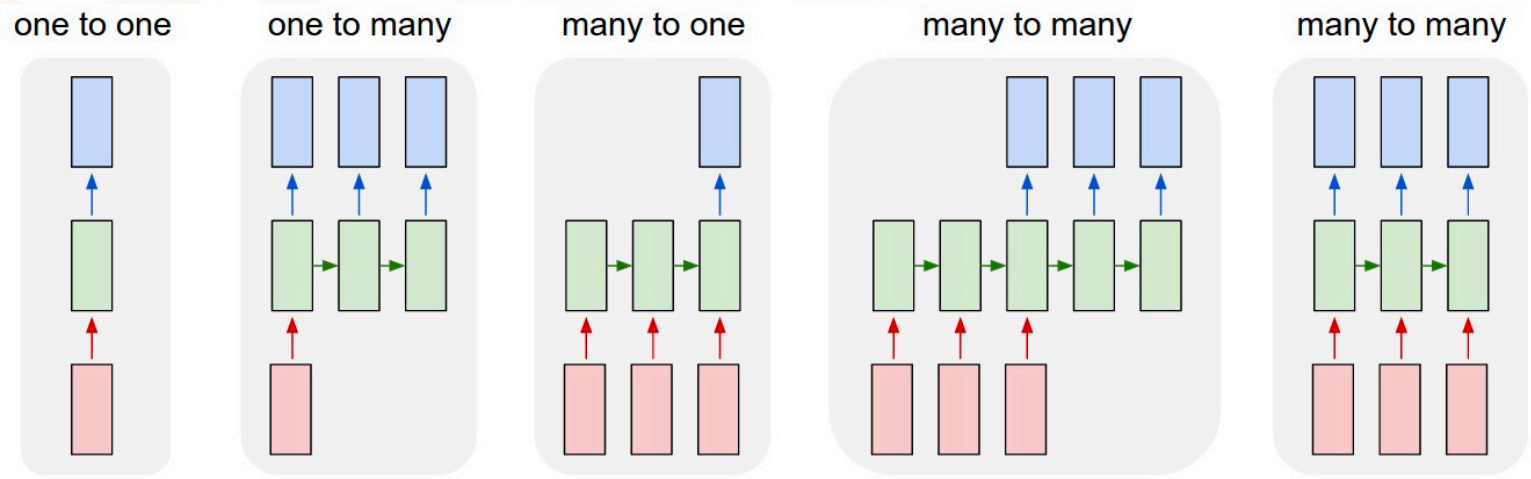
- Buscan dotar al modelo de memorio

- A cada paso, una RNN estándar recibe la entrada al tiempo t, y actualiza (o no) el estado oculto correspondiente, opcionalmente también puede realizar predicciones

Sepp hochreiter, Jurgen schmidhuber. Long Short-Term Memory, Neural Computation, 9(8):1735--1780, 1997

Modelando datos secuenciales

- Diferentes configuraciones de un escenario de modelado de datos secuencial



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

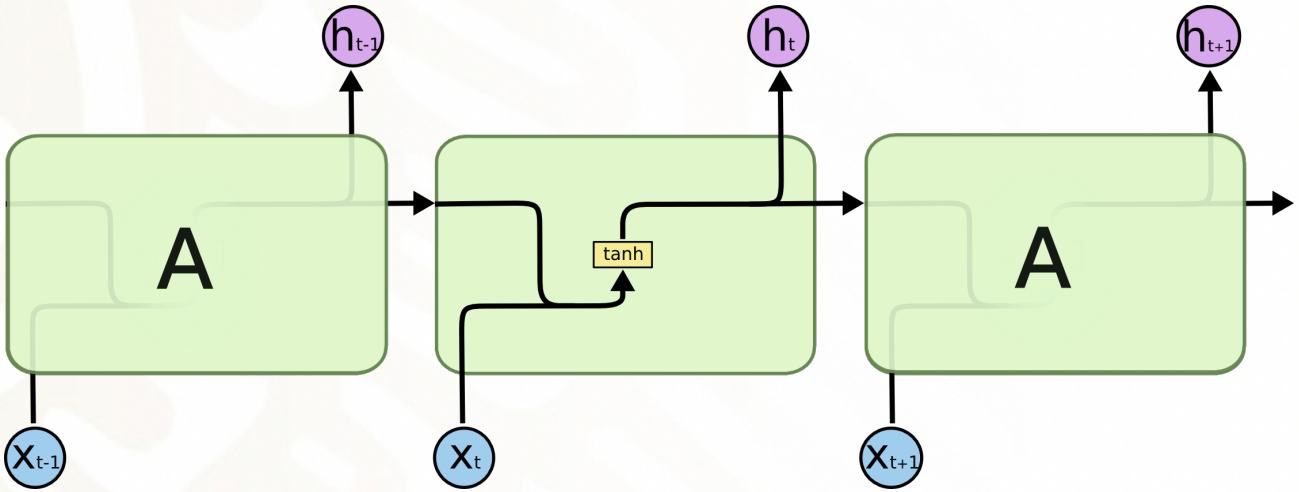
Modelando datos secuenciales

- Dentro de los modelos secuenciales más populares encontramos las redes tipo *Long short-term memory* (LSTM)
- Objetivo: construir redes de componentes tipo celda/compuerta (*gated cells*) , evitando el problema de desvanecimiento de gradientes y dotando a las redes una memoria de largo plazo
- Muy efectivas y ampliamente usadas (hasta que aparecieron los *transformers*)
- LSTM: una RNN con celdas LSTM

Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276.

Modelando datos secuenciales

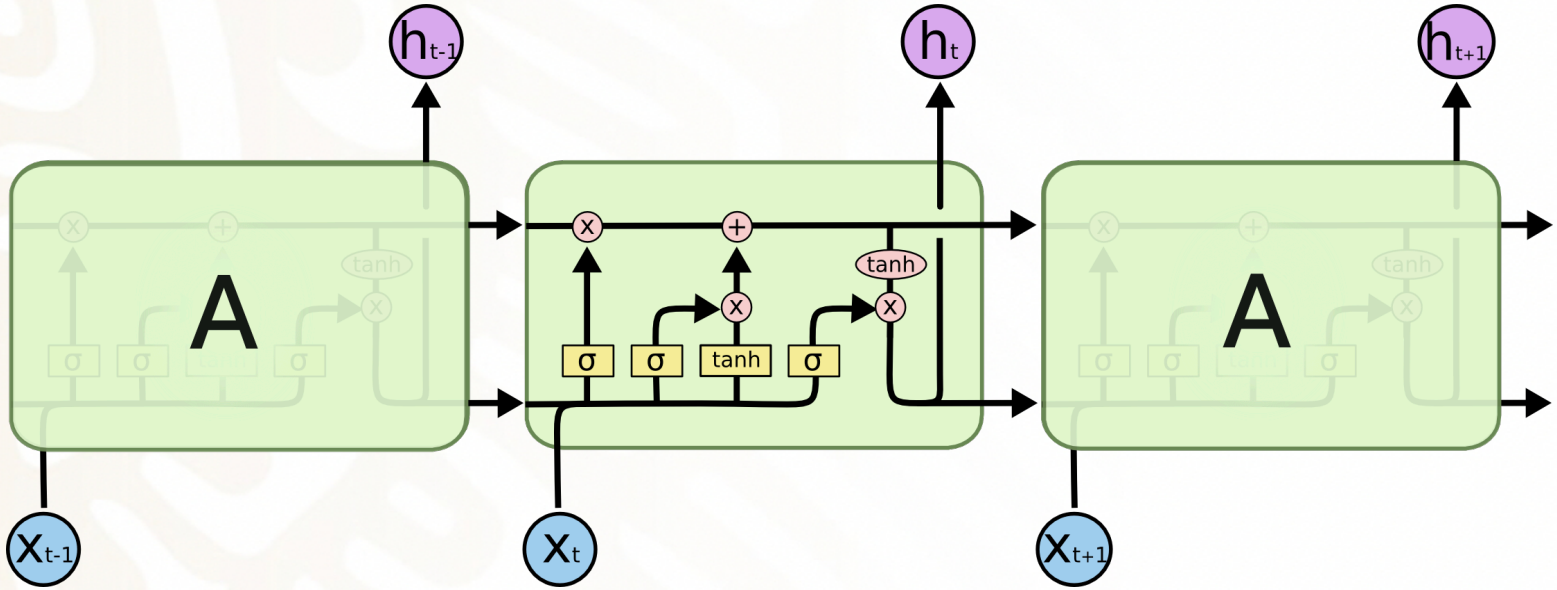
- LSTM: Long short-term memory



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Modelando datos secuenciales

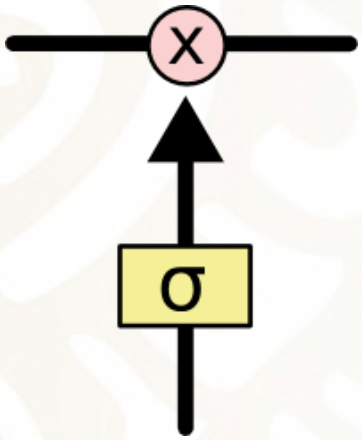
- LSTM: Long short-term memory



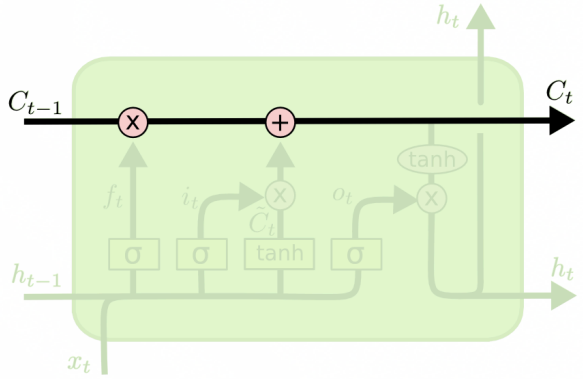
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Modelando datos secuenciales

- LSTM: Long short-term memory



Gated units

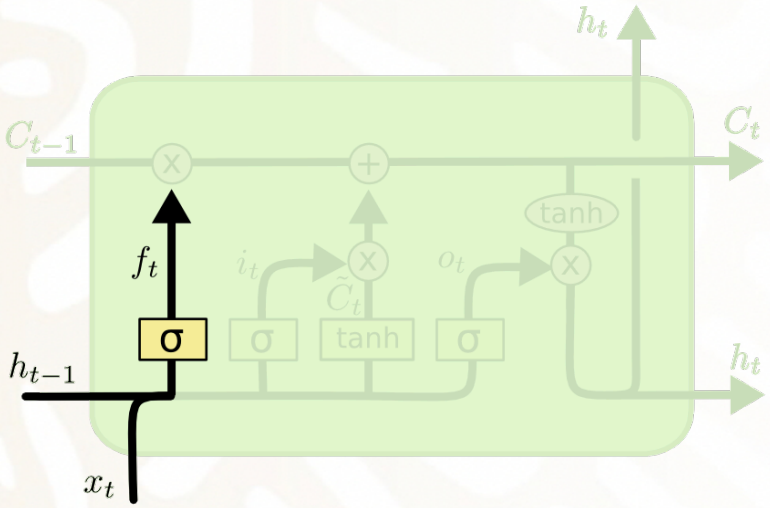


Cell state

Modelando datos secuenciales

- LSTM: Long short-term memory

Cuanta información considerar del estado previo?

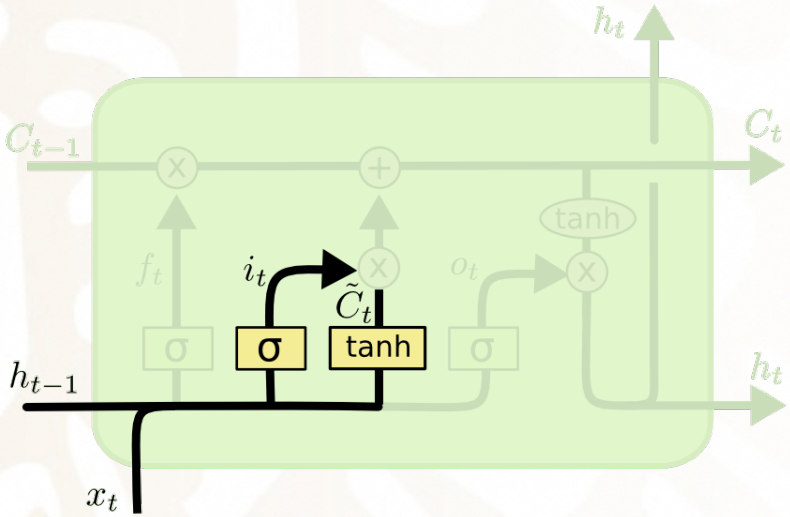


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Modelando datos secuenciales

- LSTM: Long short-term memory

Qué puede añadirse al nuevo estado, y cuánto?



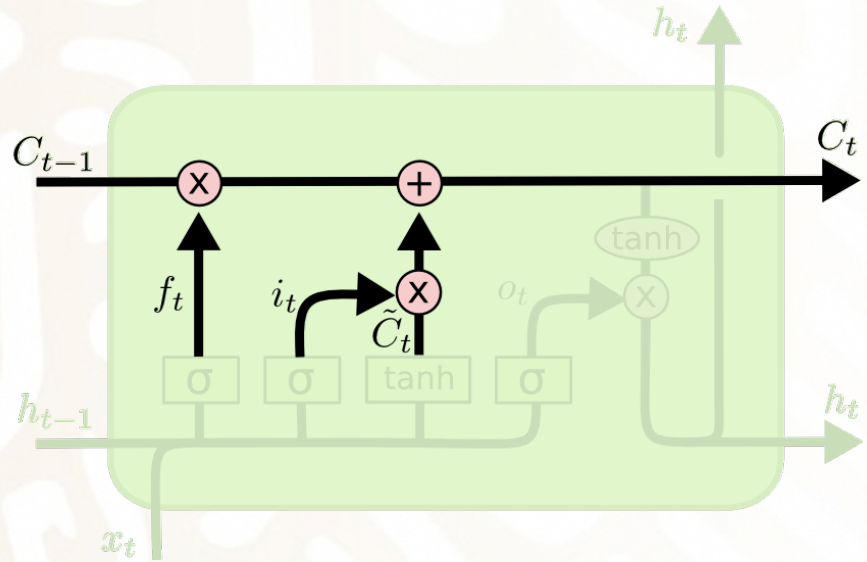
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Modelando datos secuenciales

- LSTM: Long short-term memory

Actualizar estado

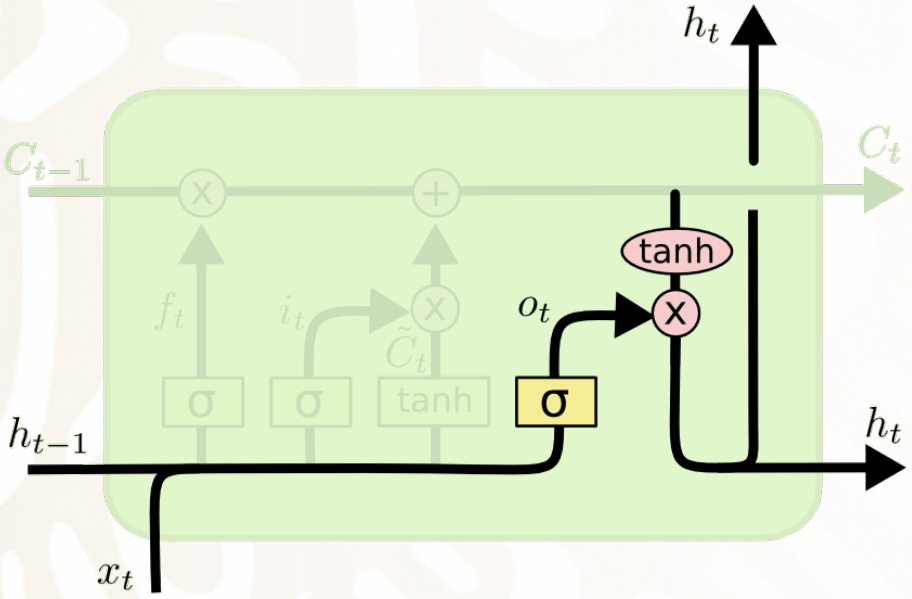


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Modelando datos secuenciales

- LSTM: Long short-term memory

Qué dar de salida?

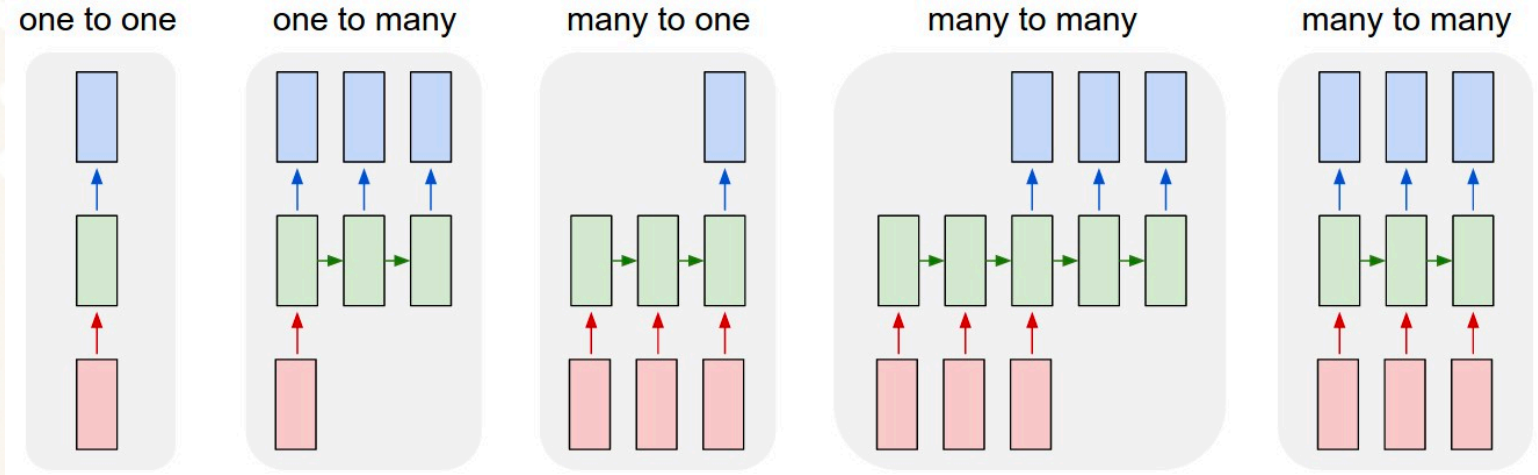


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Modelando datos secuenciales

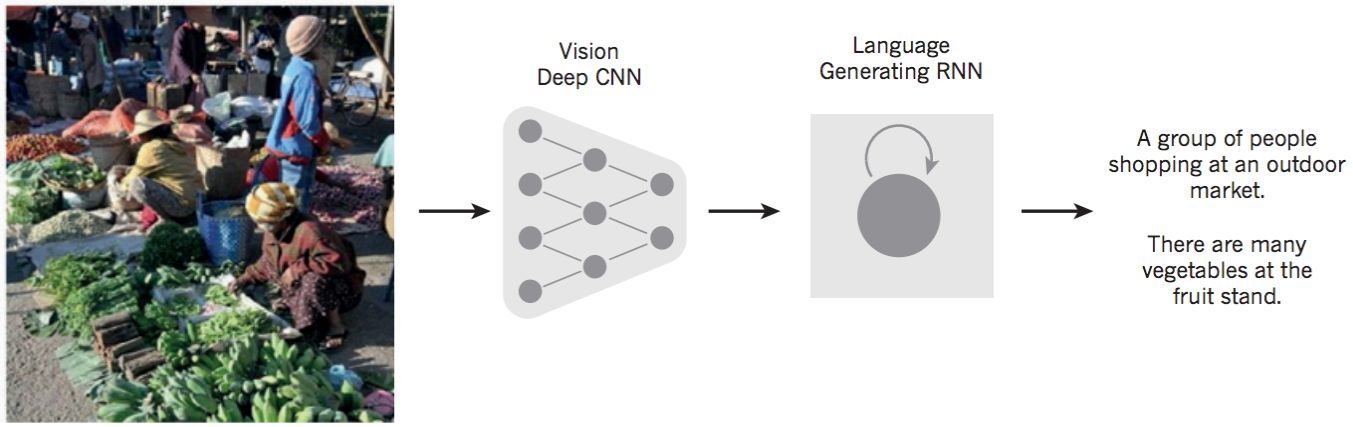
- Diferentes configuraciones de un escenario de modelado de datos secuencial



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

Modelando datos secuenciales

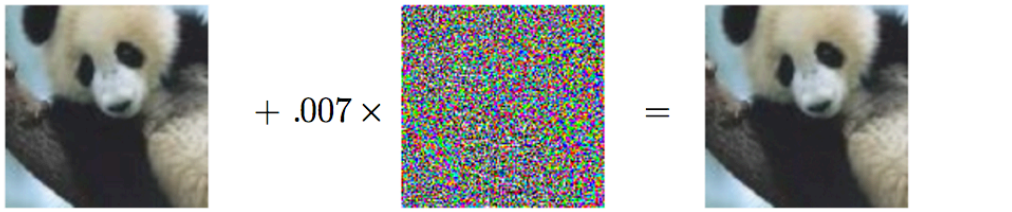
- LSTM: Resultados impresionantes en diversas tareas (procesamiento del habla, traducción), ampliamente usadas hoy en día:
  - Image captioning
  - Natural language processing
  - Multimodal information processing



Adversarial learning

Adversarial learning

- Idea: To introduce adversarial examples during the training phase
  - Adversarial example: an instance of the problem at hand that after a minor modification (even imperceptible to human eye) makes the network to fail



$\boldsymbol{x}$

$y =$"panda"
w/ 57.7%
confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
w/ 8.2%
confidence

$\boldsymbol{x} + \epsilon\,\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
w/ 99.3 %
confidence

Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy. Explaining and Harnessing Adversarial Examples ArXiv 1412.6572, 2014

Generative adversarial networks

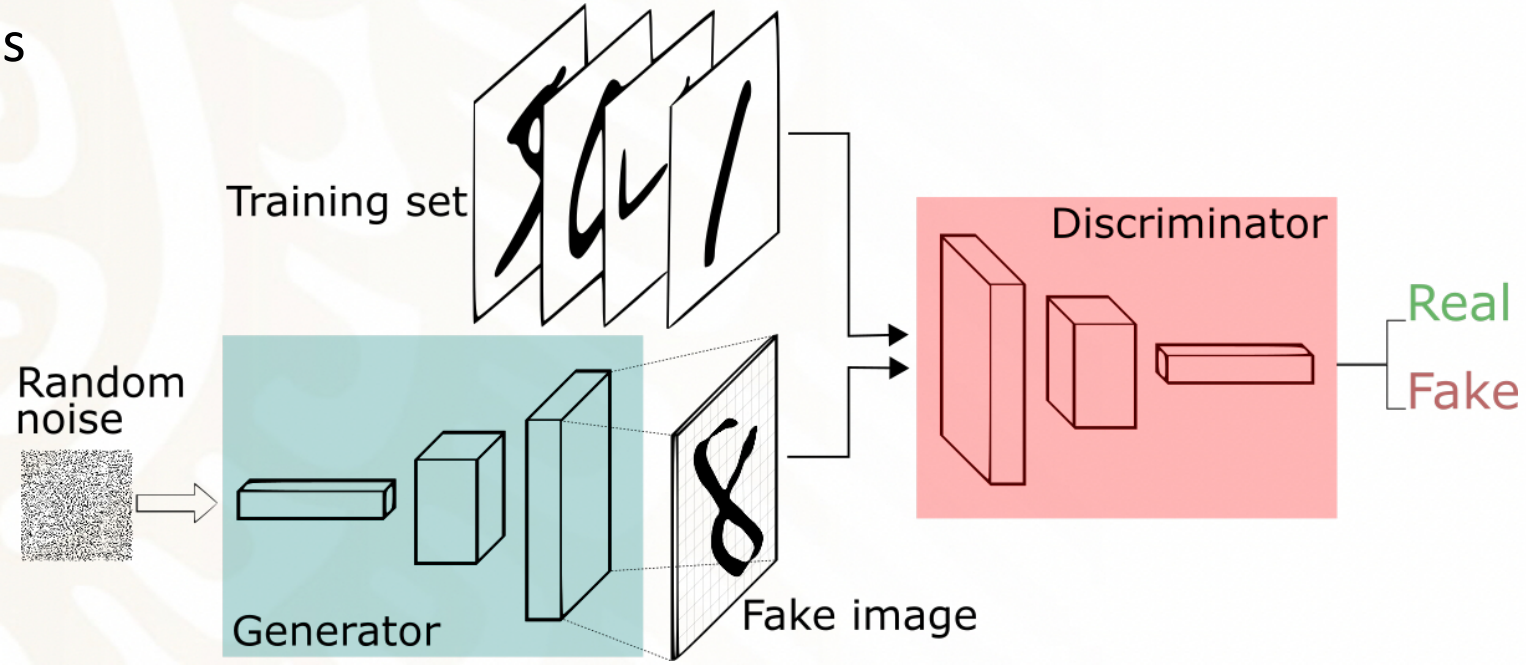
- GANs: Generative DL models that are trained jointly with a discriminator network

We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize log(1 − D(G(z))) -- or maximize (D(G(z)))

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

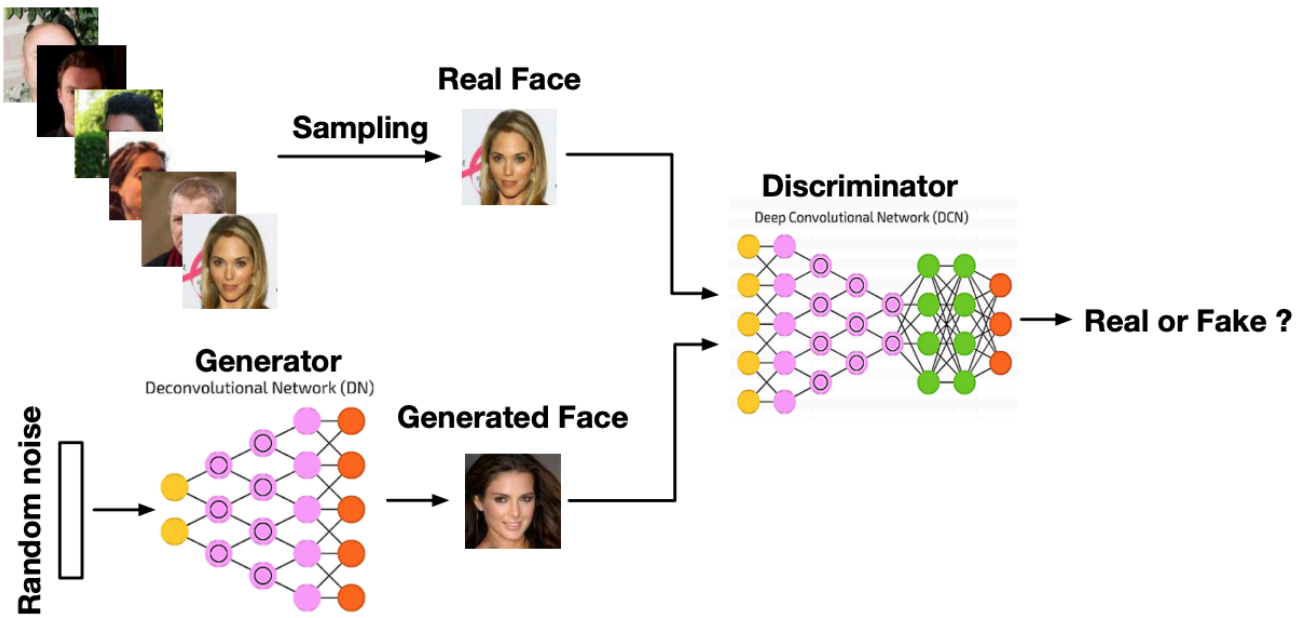
Generative adversarial networks

- GANs



https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/

Generative adversarial networks



Z. Wang et al. Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy, 2020 https://arxiv.org/pdf/1906.01529.pdf

Generative adversarial networks

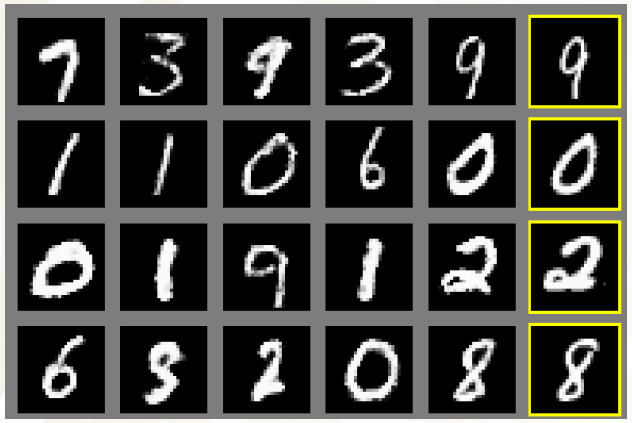
- GANs: Generative DL models that are trained jointly with a discriminator network

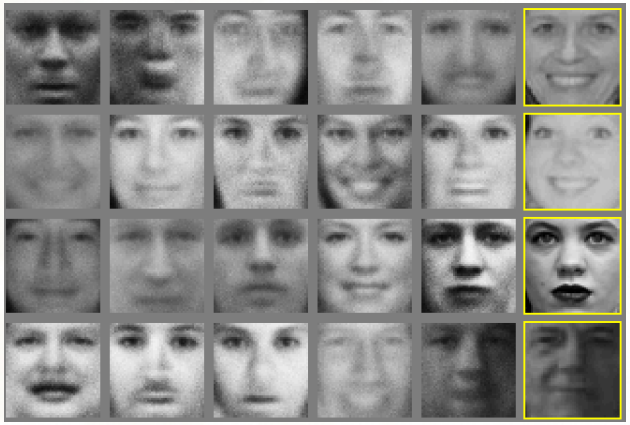
We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize log(1 − D(G(z))) -- or maximize (D(G(z)))

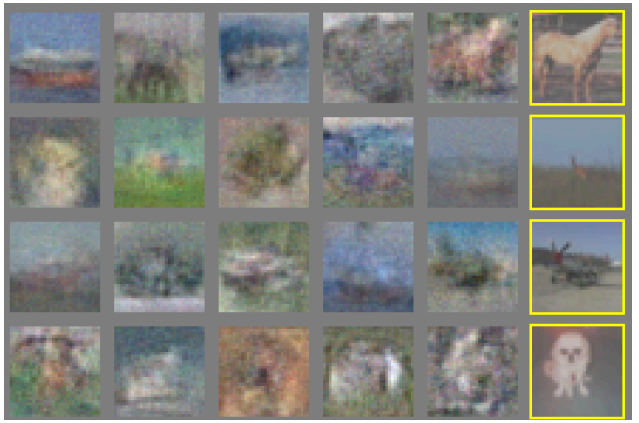
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Generative adversarial networks

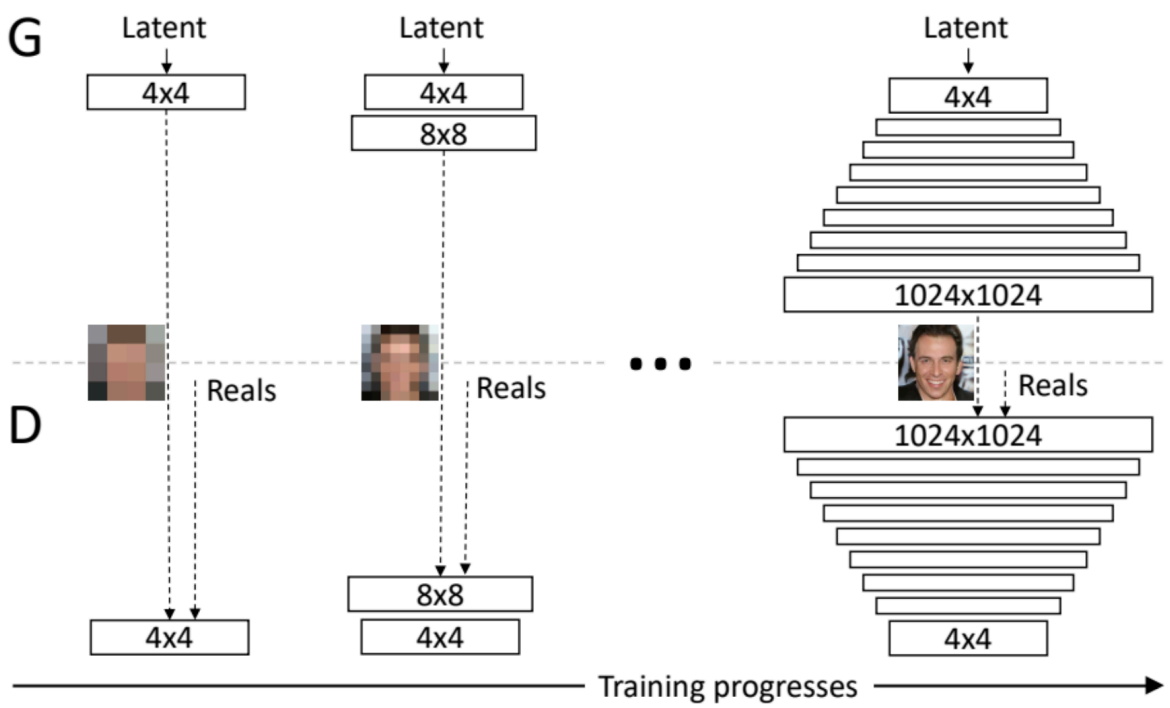
- GANs then!







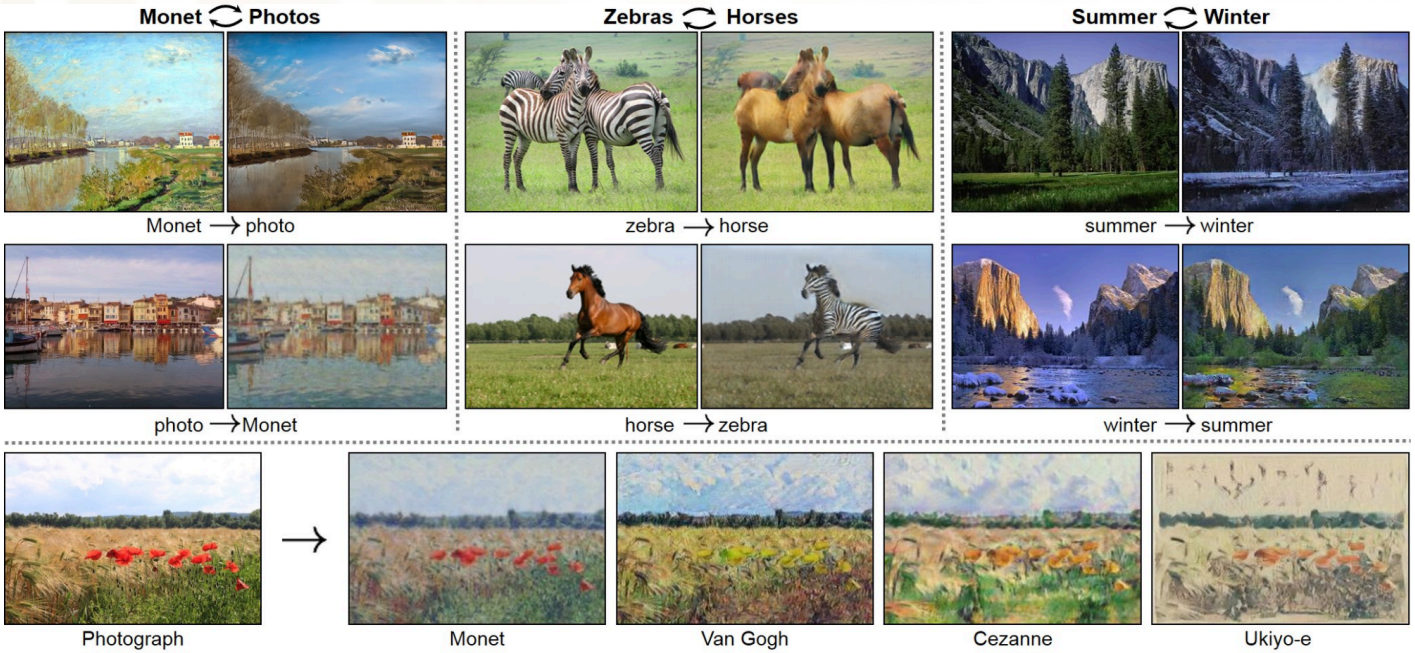
GANs: ProGAN



Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017).

GANs: CycleGAN

- Using GANs for Image-to-Image translation

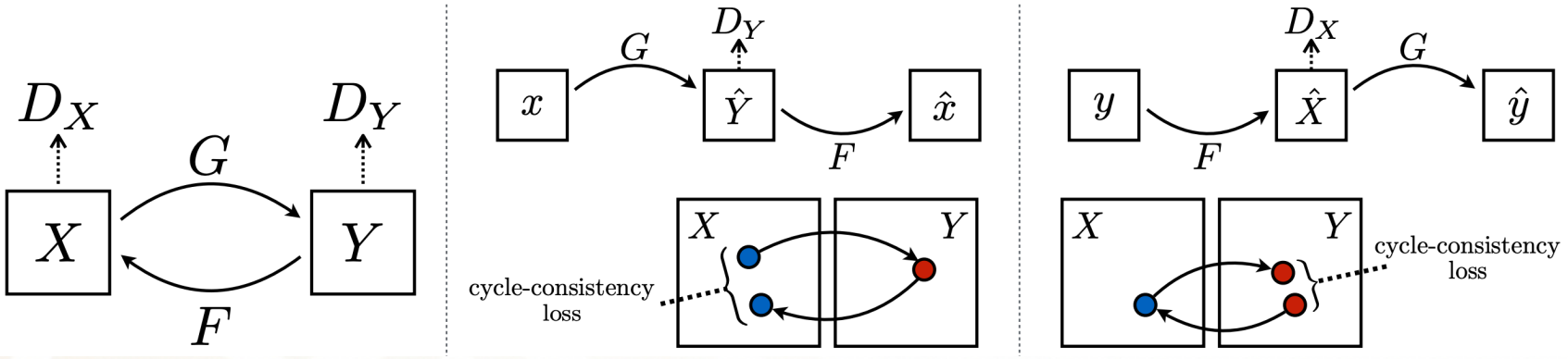


https://junyanz.github.io/CycleGAN/

Jun-Yan Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ICCV 2017

GANs: CycleGAN

- Using GANs for Image-to-Image translation



https://junyanz.github.io/CycleGAN/

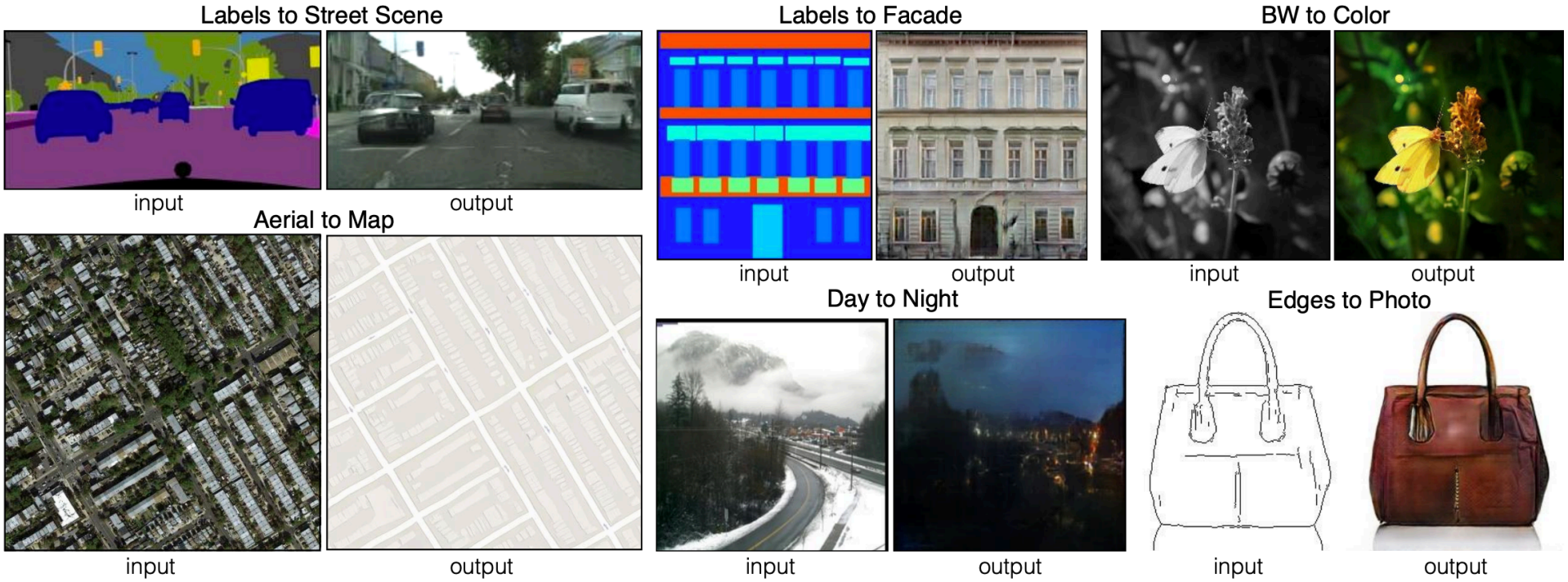
Jun-Yan Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ICCV 2017

Generative adversarial networks

- GANs now?

Pix to Pix

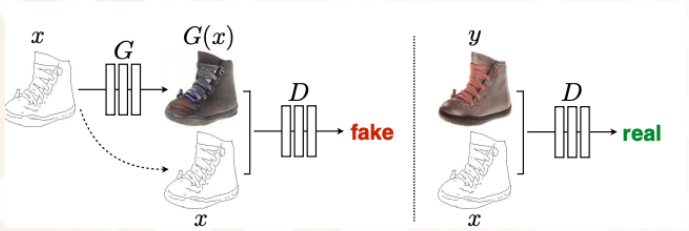
https://phillipi.github.io/pix2pix/

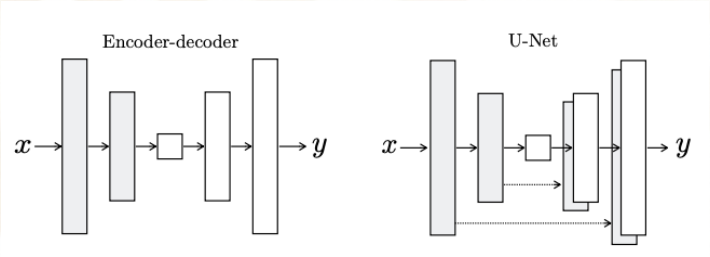


Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. **Image-to-Image Translation with Conditional Adversarial Networks.** https://arxiv.org/abs/1611.07004

Pix to Pix

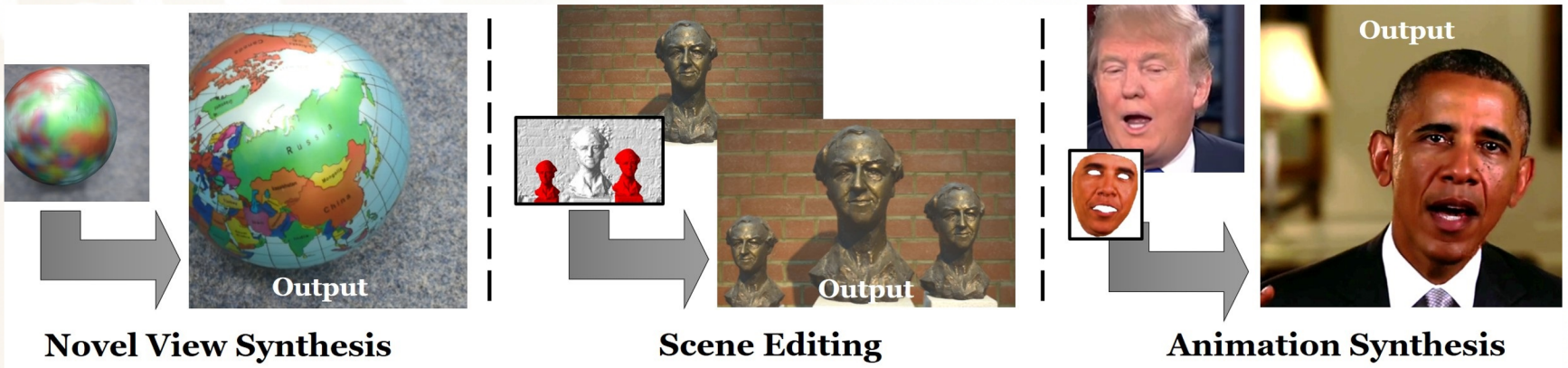
https://phillipi.github.io/pix2pix/





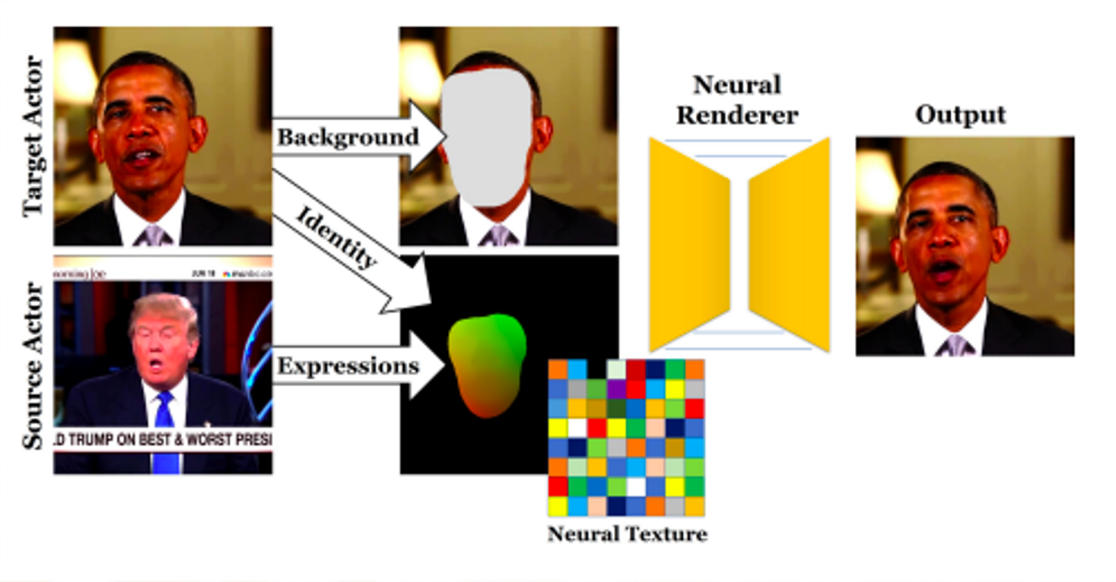
Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. **Image-to-Image Translation with Conditional Adversarial Networks.** https://arxiv.org/abs/1611.07004

Synthesis of images and videos



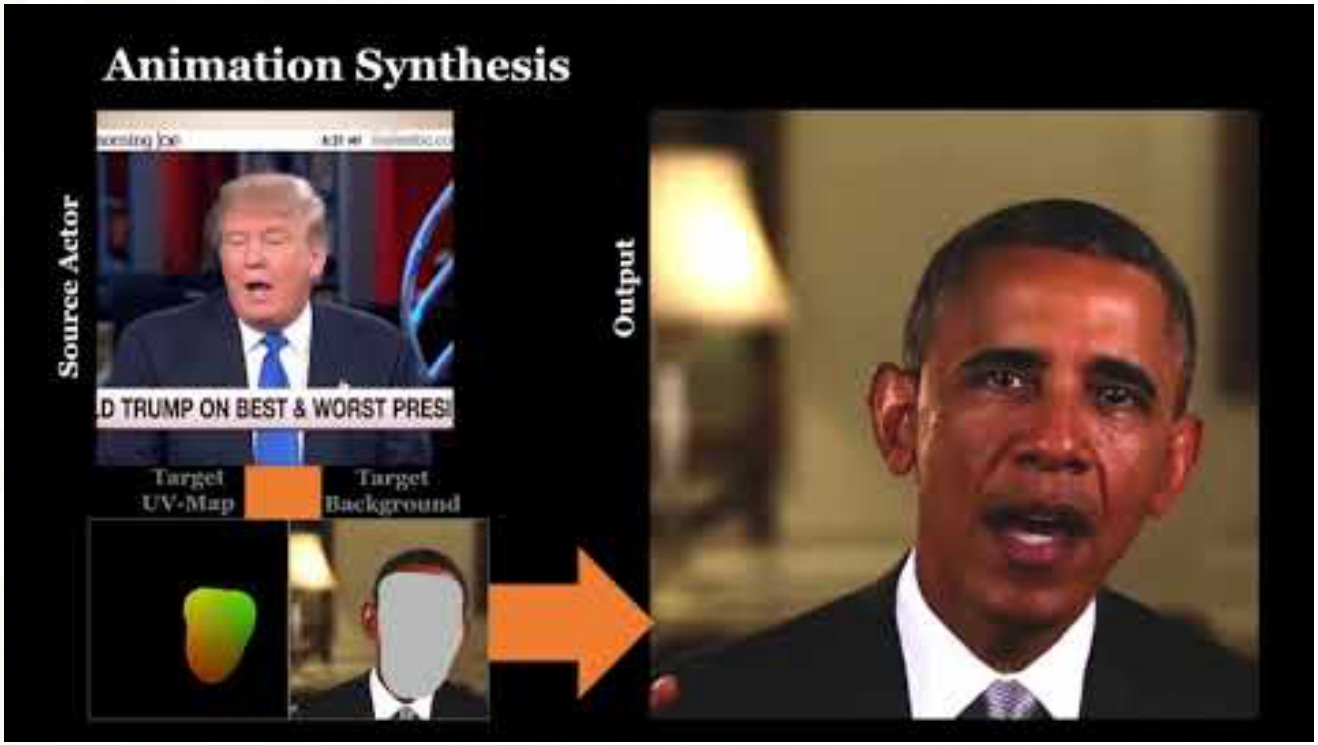
https://niessnerlab.org/projects/thies2019neural.html

Synthesis of images and videos



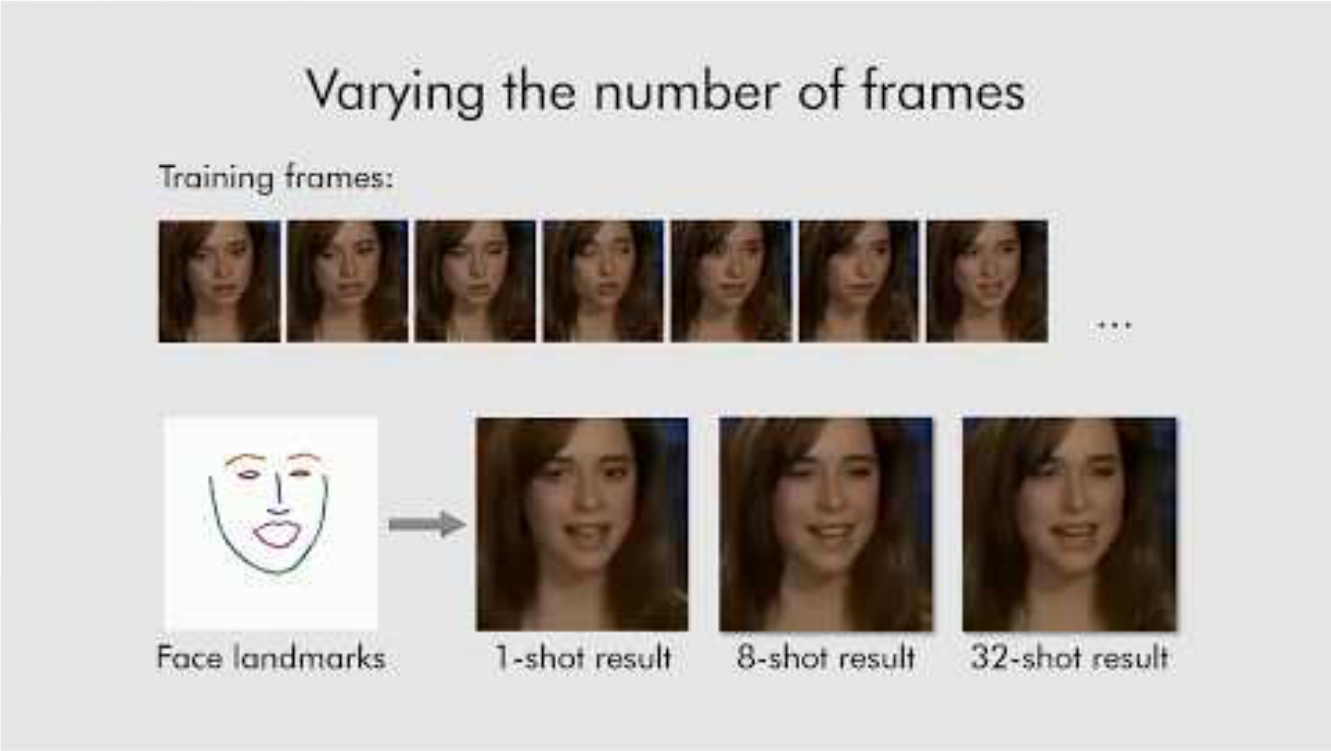
https://www.youtube.com/watch?v=z-pVip6WeyY

Synthesis of images and videos



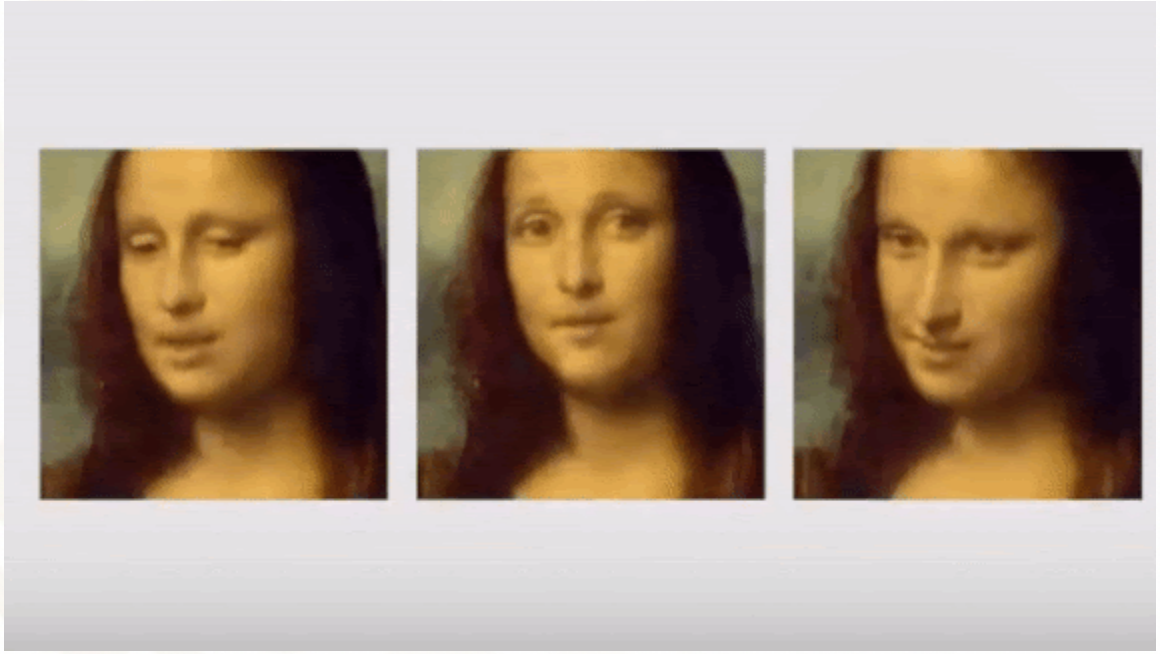
https://www.youtube.com/watch?v=z-pVip6WeyY

Synthesis of images and videos



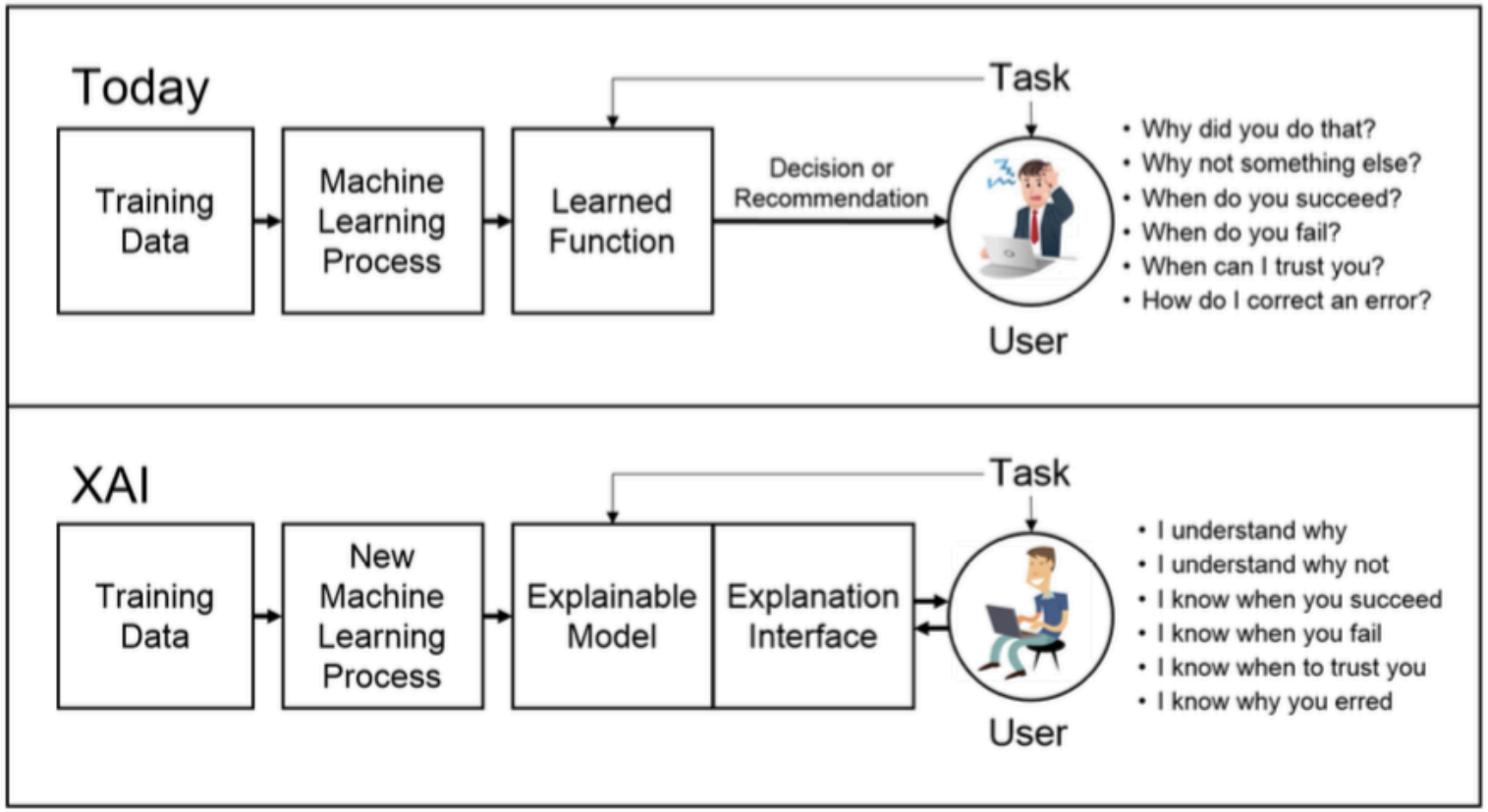
https://www.youtube.com/watch?feature=oembed&v=p1b5aiTrGzY

Synthesis of images and videos



Retos del aprendizaje profundo

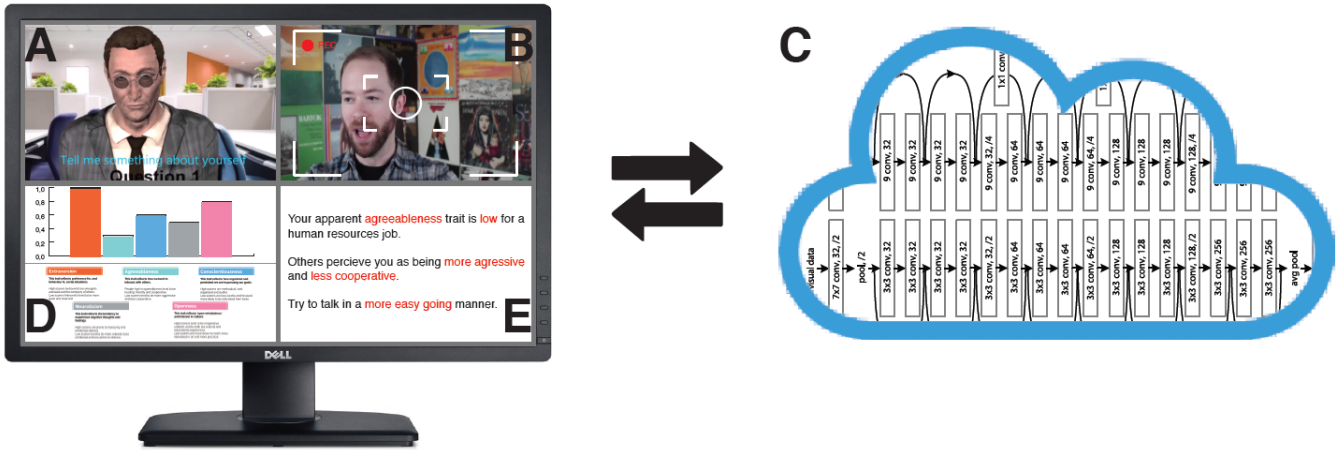
Explainability



Broad Agency Announcement Explainable Artificial Intelligence (XAI). DARPA-BAA-16-53, August 10, 2016

Explainability

- To develop interpretable models, whose predictions can be explained

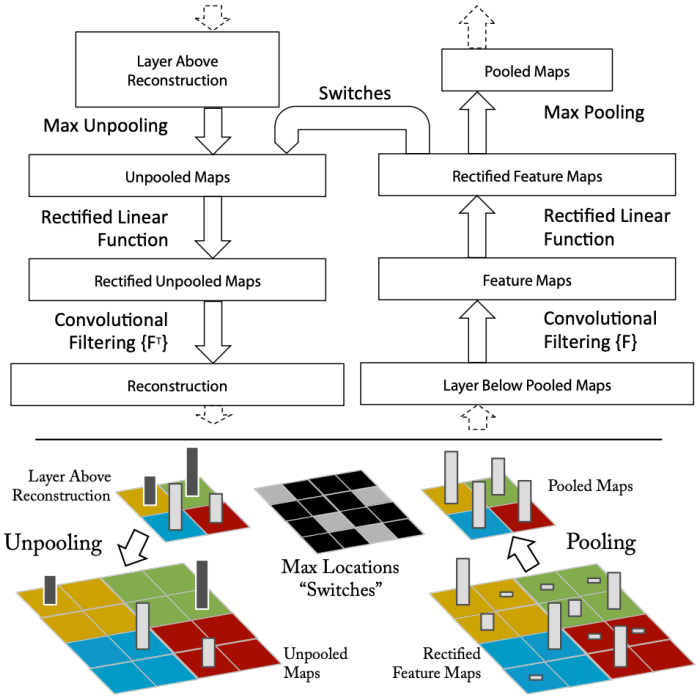


Hugo Jair Escalante et al. **Explaining First Impressions: Modeling, Recognizing, and Explaining Apparent Personality from Videos.** CoRR abs/1802.00745 (2018)

Basics of interpetability

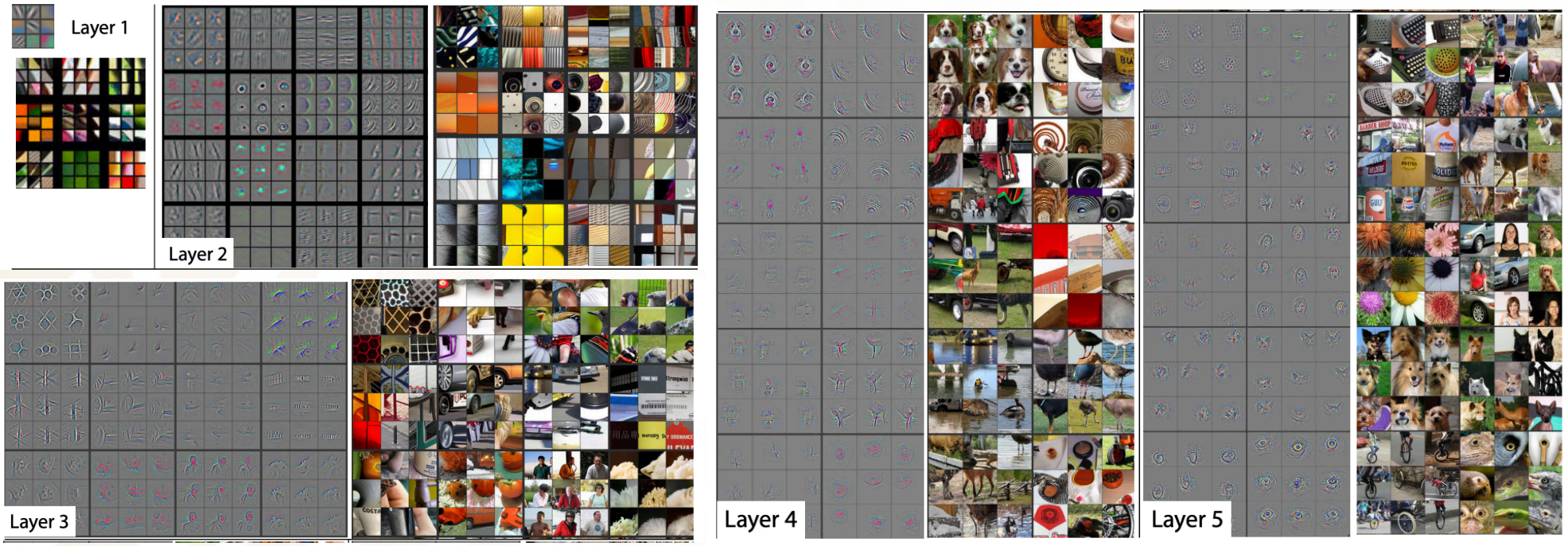
- A deconvnet is attached to every CNN layer

- To examine an activation:
  - Set all other activations in the layer to zero
  - Pass the feature maps as input to deconvnet layer
  - Successively (i) unpool, (ii) rectify and (iii) filter to reconstruct the activity in the layer beneath that gave rise to the chosen activation.
  - This is then repeated until input pixel space is reached.



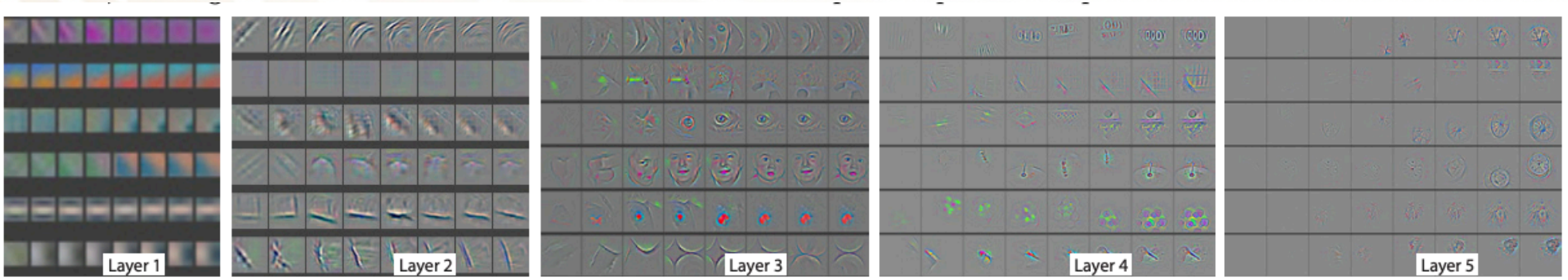
Matthew D. Zeiler, Rob Fergus: Visualizing and Understanding Convolutional Networks. ECCV (1) 2014: 818-833 https://arxiv.org/pdf/1311.2901.pdf

Visualizing the activations of layers after training



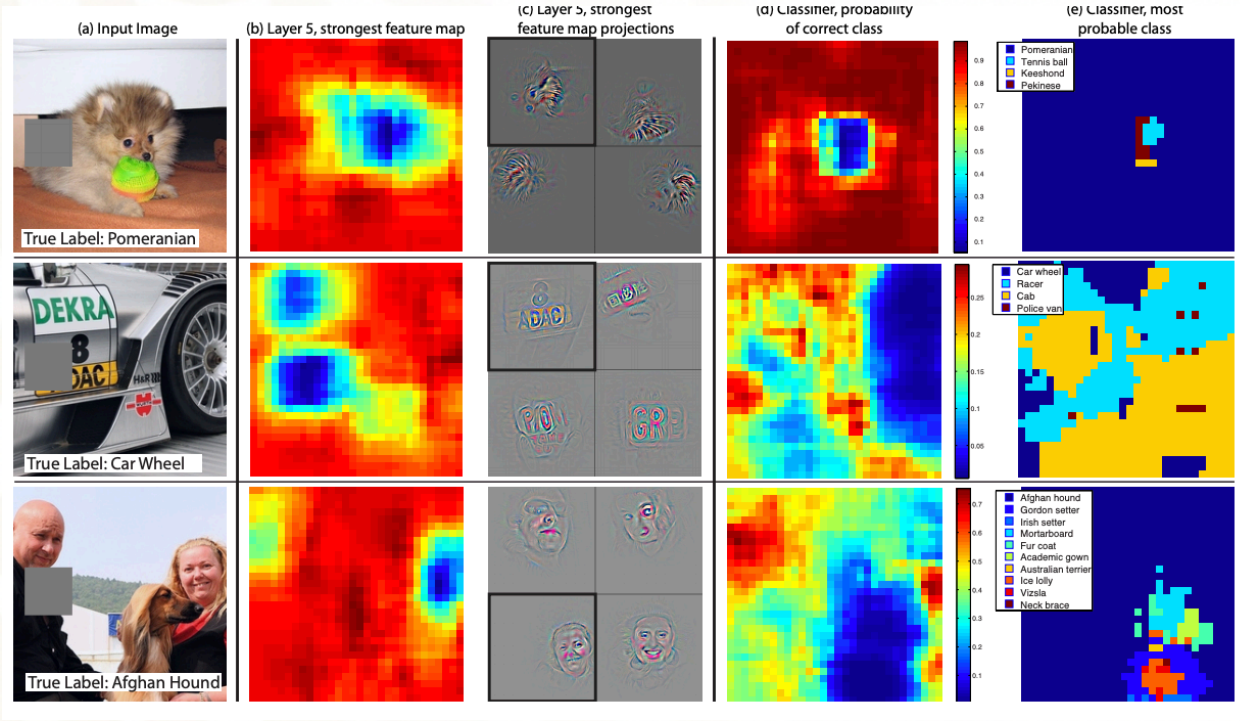
Matthew D. Zeiler, Rob Fergus: Visualizing and Understanding Convolutional Networks. ECCV (1) 2014: 818-833 https://arxiv.org/pdf/1311.2901.pdf

Visualizing the activations of layers after training



Matthew D. Zeiler, Rob Fergus: Visualizing and Understanding Convolutional Networks. ECCV (1) 2014: 818-833 https://arxiv.org/pdf/1311.2901.pdf

Detecting discriminative regions (attention)



Matthew D. Zeiler, Rob Fergus: Visualizing and Understanding Convolutional Networks. ECCV (1) 2014: 818-833 https://arxiv.org/pdf/1311.2901.pdf

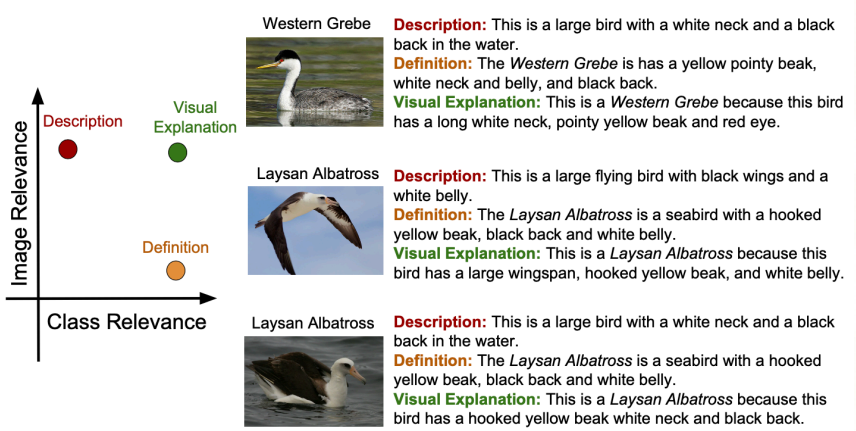
Detecting discriminative regions (attention)

- Task: apparent personality recognition

Yagmur Güçlütürk, Umut Güçlü, Marc Pérez, Hugo Jair Escalante, Xavier Baró, Carlos Andújar, Isabelle Guyon, Júlio C. S. Jacques Júnior, Meysam Madadi, Sergio Escalera, Marcel A. J. van Gerven, Rob van Lier: Visualizing Apparent Personality Analysis with Deep Residual Networks. ICCV Workshops 2017: 3101-3109

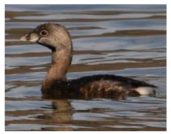
Language and vision to generate explanations

- Explain why a model recognizes certain class in an image:




This is a pine grosbeak because this bird has a red head and breast with a gray wing and white wing.


This is a Kentucky warbler because this is a yellow bird with a black cheek patch and a black crown.


This is a pied billed grebe because this is a brown bird with a long neck and a large beak.


This is an artic tern because this is a white bird with a black head and orange feet.

Akata, Zeynep (et al.) Generating Post-Hoc Rationales of Deep Visual Classification Decisions. In Explainable and Interpretable Models in Computer Vision and Machine Learning, 135-154, 2019

Language and vision to generate explanations

- Explain why a model recognizes certain class in an image:

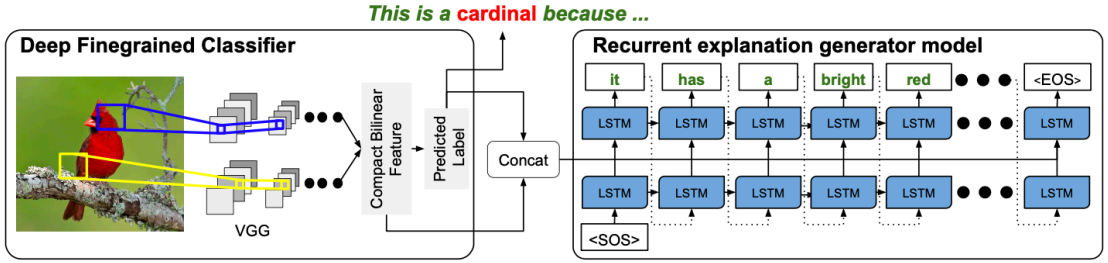
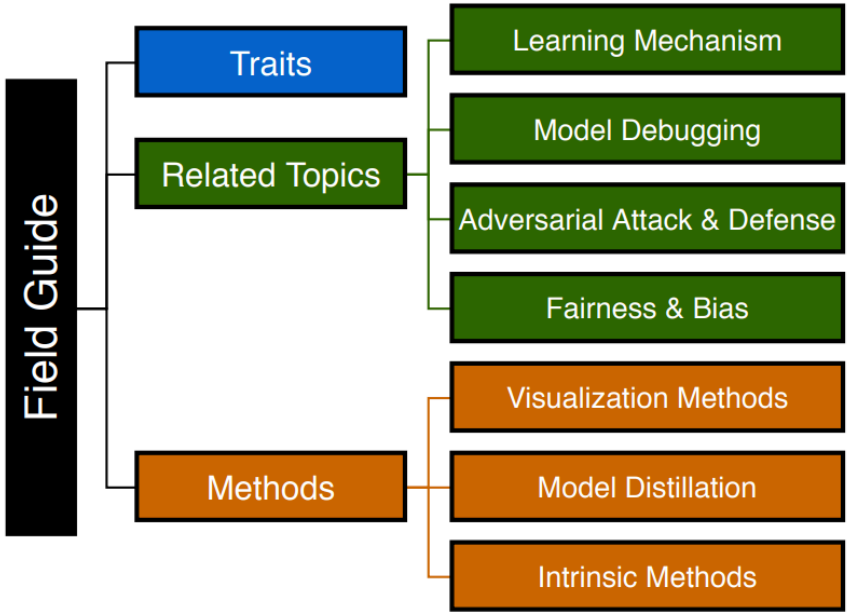


Figure 6.2: Our joint classification and explanation model, aka GVE. We extract visual features using a fine-grained classifier before sentence generation and, unlike other sentence generation models, condition sentence generation on the predicted class label. A novel discriminative loss encourages generated sentences to include class specific attributes.

Akata, Zeynep (et al.) Generating Post-Hoc Rationales of Deep Visual Classification Decisions. In Explainable and Interpretable Models in Computer Vision and Machine Learning, 135-154, 2019

State of the art on explainability



Traits
Intent: What are the objectives of deep learning explanations? How is explainability evaluated?

Related Topics
Context: How is deep learning explainability linked with other research topics? How does deep learning explainability contrast with other work?

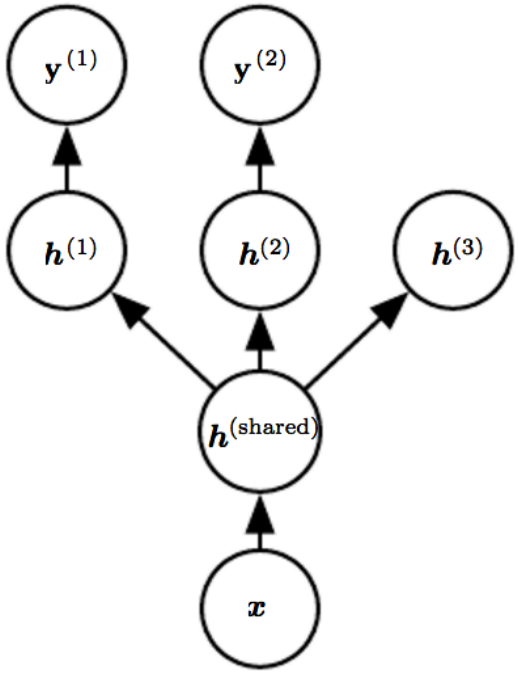
Methods
Foundations: What concepts and methods does much of the recent literature build from? What algorithms are "foundational" for deep learning explainability?

Ning Xie, Gabrielle Ras, Marcel van Gerven, Derek Doran Explainable Deep Learning: A Field Guide for the Uninitiated. https://arxiv.org/abs/2004.14545

Multi task learning

- Idea: learning models sharing generic layers and at the same time they learn specific tasks with specialized layers

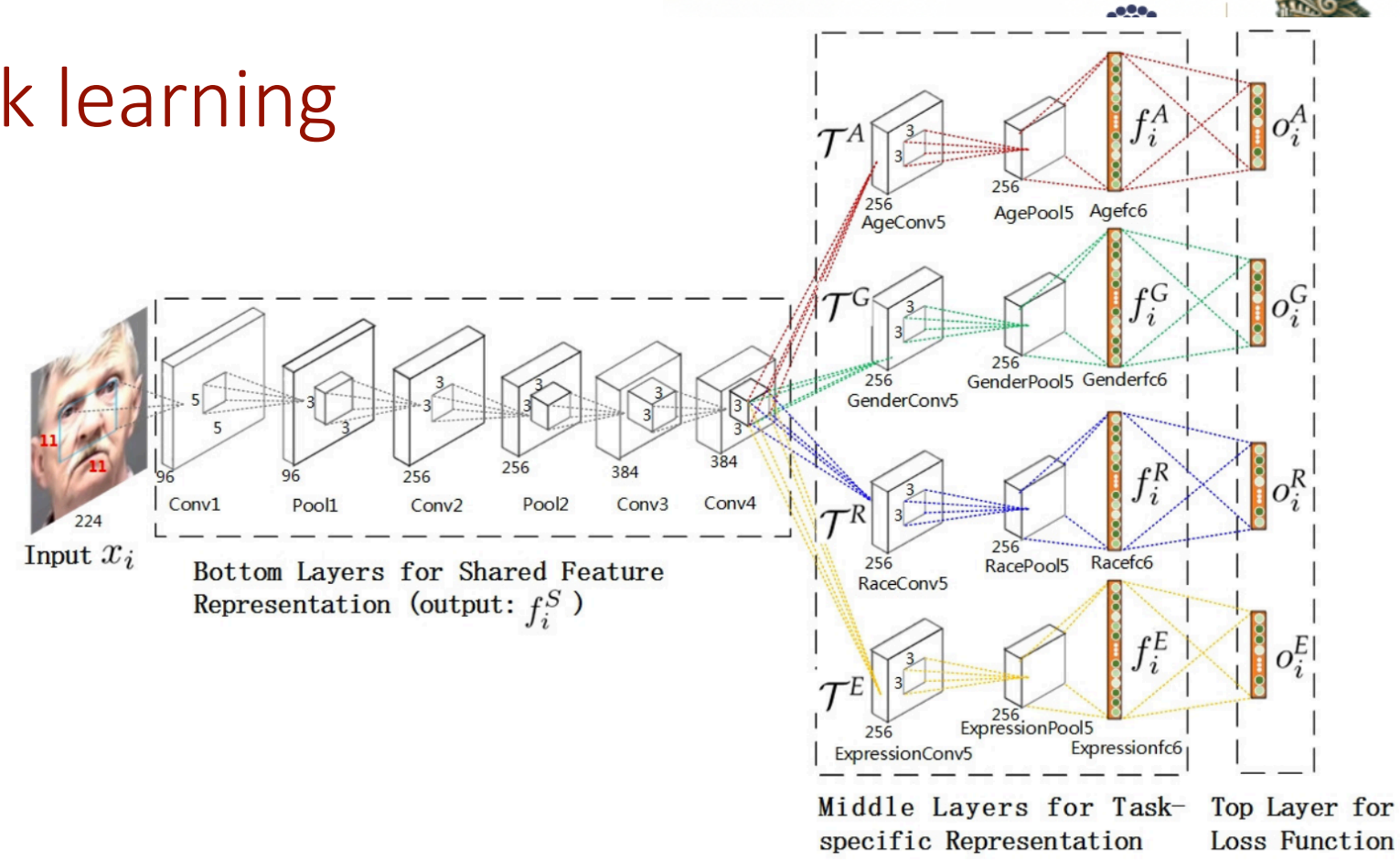
- Hypothesis: Among the factors that explain variations in data for different tasks, some of them are shared among two or more tasks.



R. Caruana. **Multitask Connectionist Learning**. Proc. Connectionist models summer school, 372--379, 1993

Multi task learning

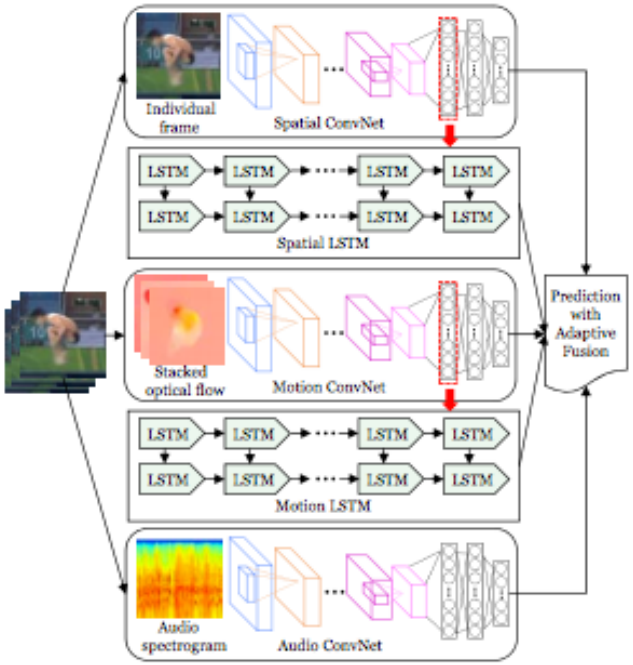
- E.g., Simultaneously predict: gender, aga, ethnicity and facial expression from images



J. Wan, S. Zhou, Z. Tan, H.J. Escalante, Y. Liang, Z. Lei, G. Guo, S.Z. Li. **Deep Nonholonomic Label Information Learning for Globally Fine-grained Face Attribute Analysis**, Submitted to TPAMI, 2017

Multi stream models

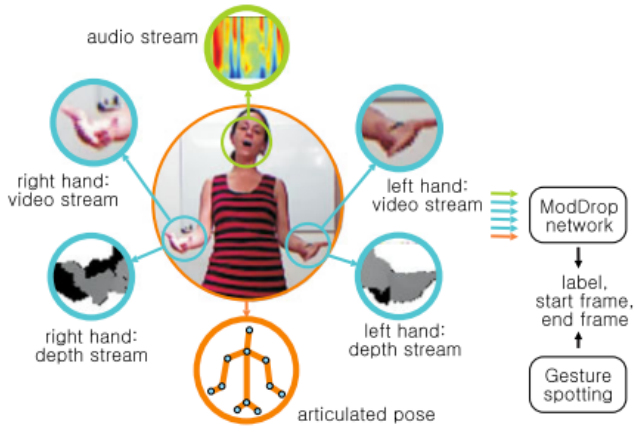
- Idea: Having different internal paths within the model associated to different information sources, eventually converging to the same layer (single-task)

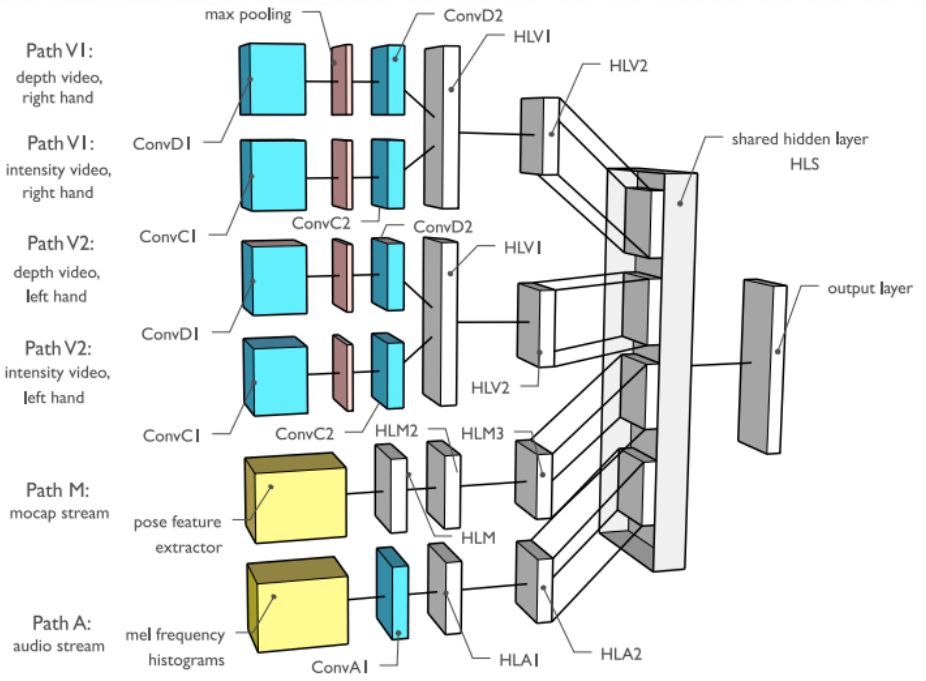


Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, Xiangyang Xue, Jun Wang. **Fusing Multi-Stream Deep Networks for Video Classification.** arXiv:1509.06086, 2015

Multi stream models

- E.g., Multi modal gesture recognition



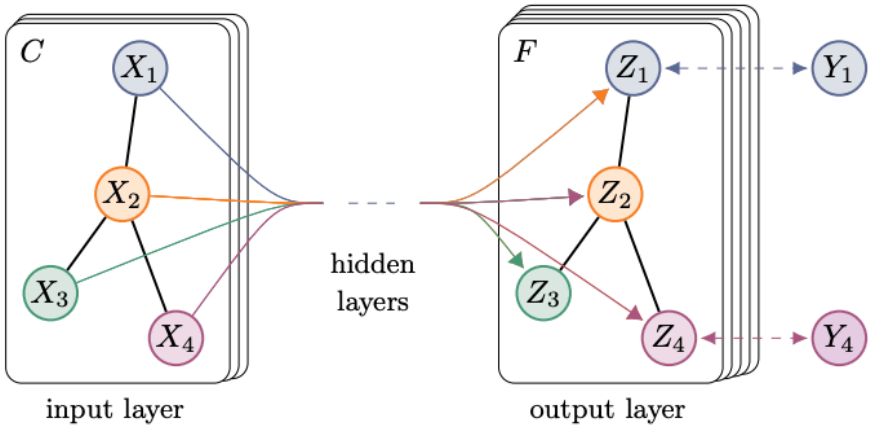


Natalia Neverova, Christian Wolf, Graham Taylor, Florian Nebout. ModDrop: adaptive multi-modal gesture recognition. TPAMI, Vol, 38(8):1692--1706, 2016

Graph networks

- Neural networks that process graph data, inputs are graphs and outputs could be graph, labels for whole graphs, labels for nodes or edges

- They address a structured input-output problem



Thomas N. Kipf, Max Welling. **Semi-Supervised Classification with Graph Convolutional Networks.** ICLR 2017
Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, George E. Dahl. **Neural Message Passing for Quantum Chemistry.** ICML 2017
Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, Jure Leskovec. **Strategies for Pre-training Graph Neural Networks.** ICLR 2020

Model compression

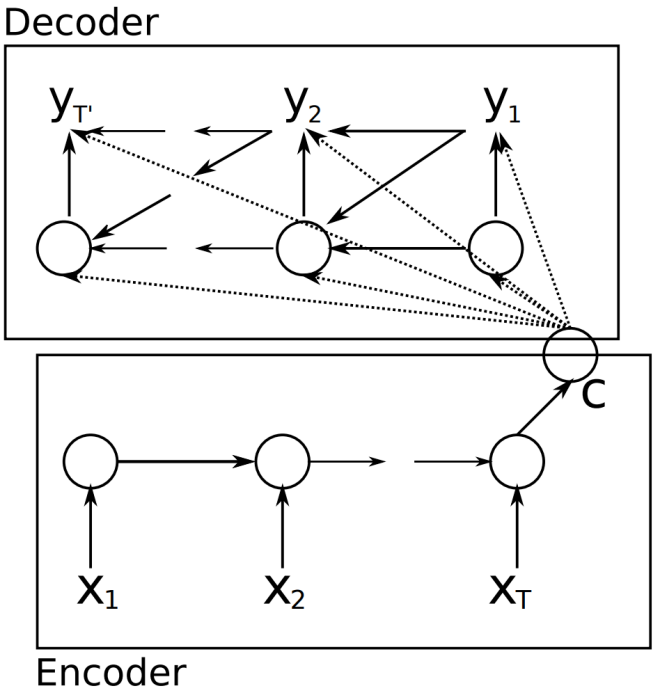
- DL models are inherently complex, which limits their application in resource constrained envorinments
- Recently, efforts from the ML community have been devoted to compress and accelerate DL models,
- Main strategies:

Theme Name	Description	Applications	More details
Parameter pruning and sharing	Reducing redundant parameters which are not sensitive to the performance	Convolutional layer and fully connected layer	Robust to various settings, can achieve good performance, can support both train from scratch and pre-trained model
Low-rank factorization	Using matrix/tensor decomposition to estimate the informative parameters	Convolutional layer and fully connected layer	Standardized pipeline, easily to be implemented, can support both train from scratch and pre-trained model
Transferred/compact convolutional filters	Designing special structural convolutional filters to save parameters	Convolutional layer only	Algorithms are dependent on applications, usually achieve good performance, only support train from scratch
Knowledge distillation	Training a compact neural network with distilled knowledge of a large model	Convolutional layer and fully connected layer	Model performances are sensitive to applications and network structure only support train from scratch

Y Chen et al. **A Survey of Model Compression and Acceleration for Deep Neural Networks.** IEEE SIGNAL PROCESSING MAGAZINE, 2020
Tejalal Choudhary, et al. **A comprehensive survey on model compression and acceleration.** Artificial Intelligence Review (2020)

The encoder – decoder architecture for modeling sequential data

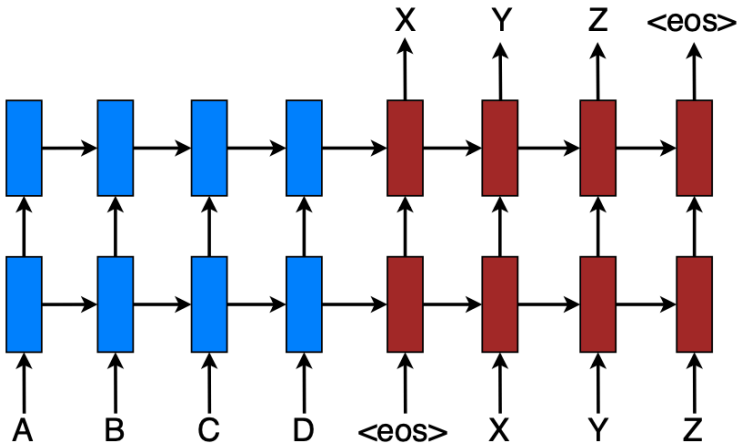
- Idea: mapping inputs of variable length to a (fixed length) embedding, then generating variable length outputs from the embedding



Kyunghyun Cho et al. **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation**, EMNLP 2014
https://arxiv.org/abs/1406.1078

Attention mechanisms

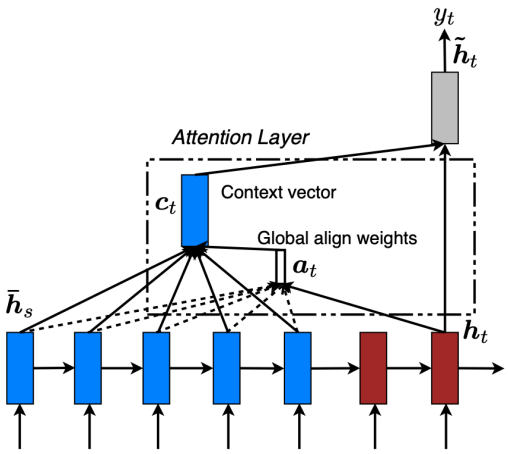
- Context: Neural Machine Translation
- Idea: introducing a weight for the decoder that takes into account, inputs, hidden states and a (learnable) context vector derived from hidden stantes
- Intuition: the context alings input and output sequences, it indicates what input words are more likely to be related to each output word

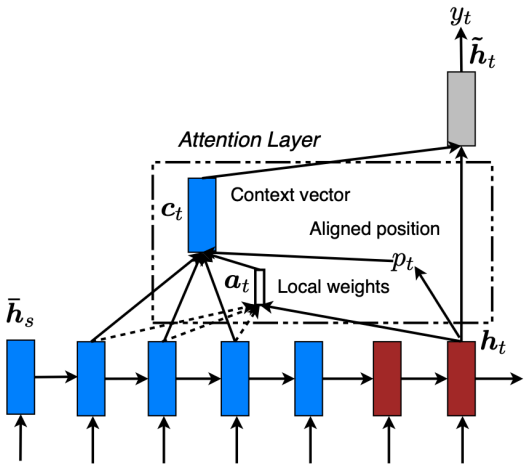


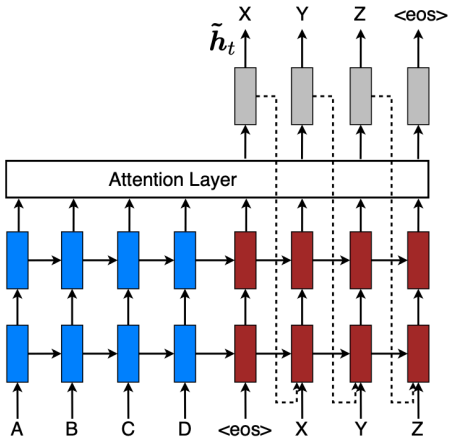
Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICLR
Minh-Thang Luong et al. Effective Approaches to Attention-based Neural Machine Translation. EMNLP 2015

Attention mechanisms

- How to define/learn the context vector?



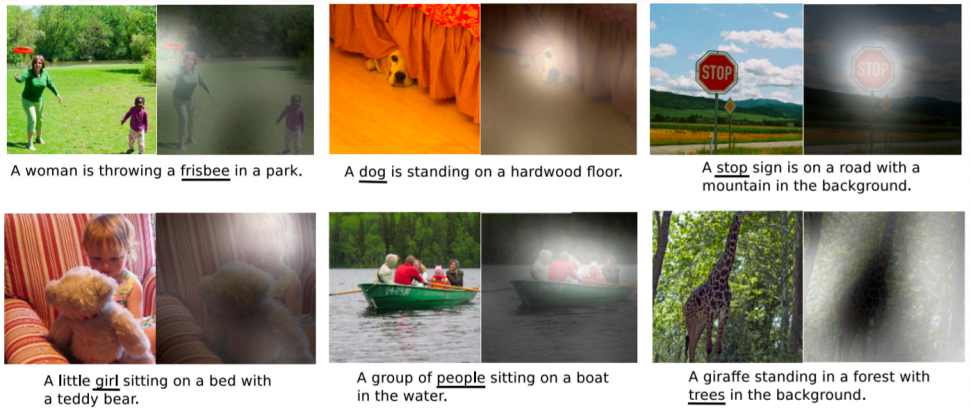




Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICLR
Minh-Thang Luong et al. Effective Approaches to Attention-based Neural Machine Translation. EMNLP 2015

Attention mechanisms

- What / where to look in an image?
- What image region (input) is more related to an output word?



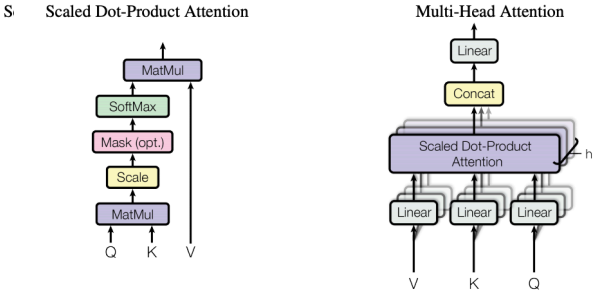
Kelvin Xu et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. ICML 2016

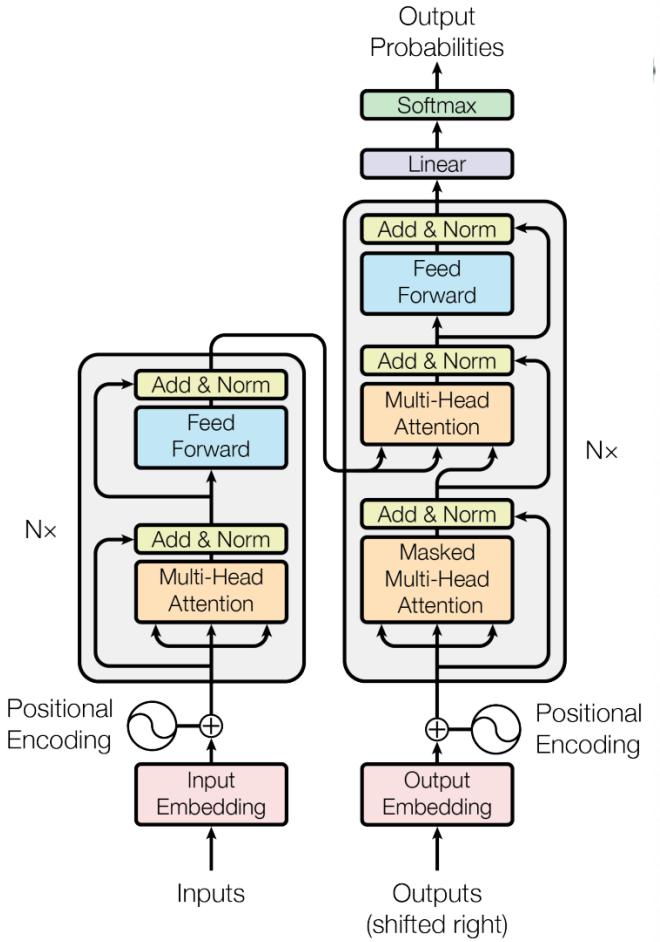
https://medium.com/@shairozsohail/a-survey-of-visual-attention-mechanisms-in-deep-learning-1043eb25f343

https://icml.cc/media/Slides/icml/2019/halla(10-09-15)-10-15-45-4343-a_tutorial_on.pdf

Transformers

- Forget about RNNs
- Self attention (what are the most important words in the same sentence)
- Multi- headed attention



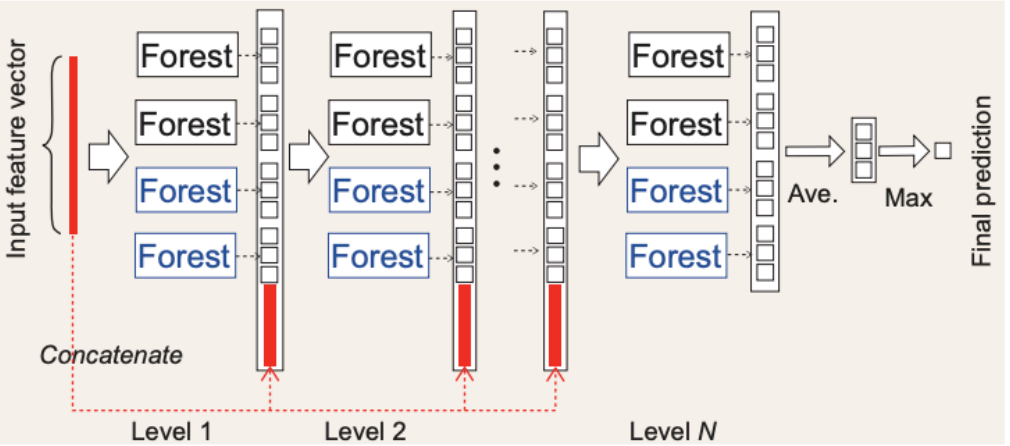


Ashish Vaswani et al. Attention is all you need. NeurIPS 2017,

http://nlp.seas.harvard.edu/2018/04/03/attention.html

Representation learning without NNs?

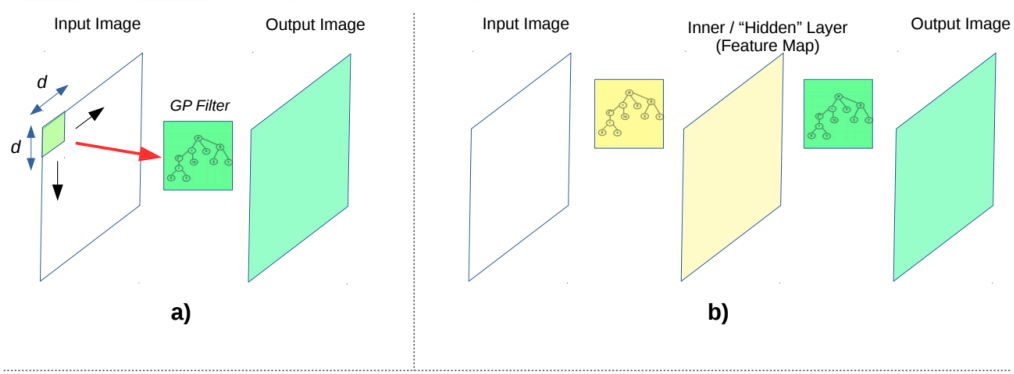
- Are NNs the only means to deep learning?
- Recent efforts:
  - *Deep forest*
  - *Deep genetic programming*
  - *…?*



Zhi-Hua Zhou, Ji Feng. **Deep Forest.** National Science Review 00: 1–13, 2018

Representation learning without NNs?

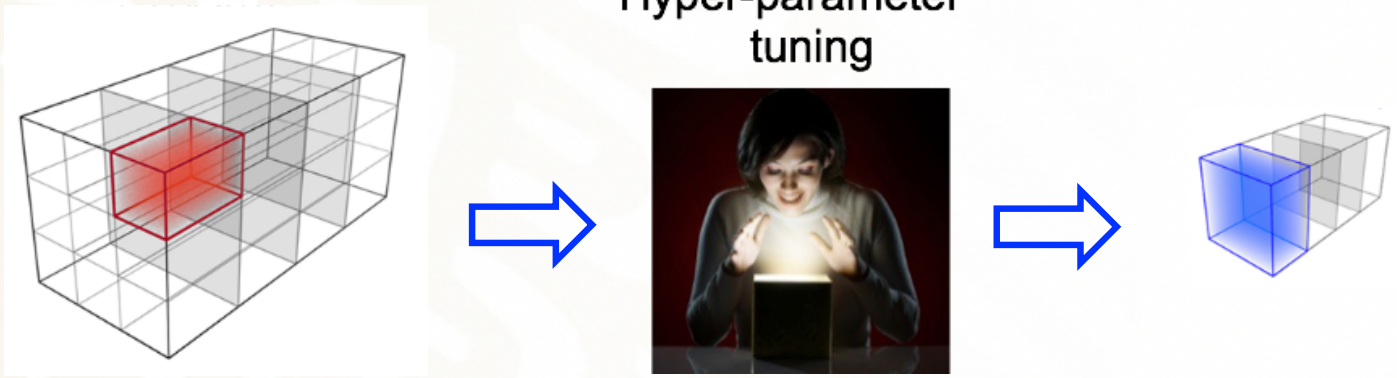
- Are NNs the only means to deep learning?
- Recent efforts:
  - *Deep forest*
  - *Deep genetic programming*
  - *…?*



Lino Rodriguez, Alicia Morales, Hugo Jair Escalante. Evolving Autoencoding Structures through Genetic Programming. Genetic Programming and Evolvable Machines, 2019

Automatic Machine Learning - AutoML

Automatic Deep Learning



https://autodl.chalearn.org/

Few shot learning

*Definition 2.1 (**Machine Learning** [92, 94]).* A computer program is said to learn from experience E with respect to some classes of task T and performance measure P if its performance can improve with E on T measured by P.

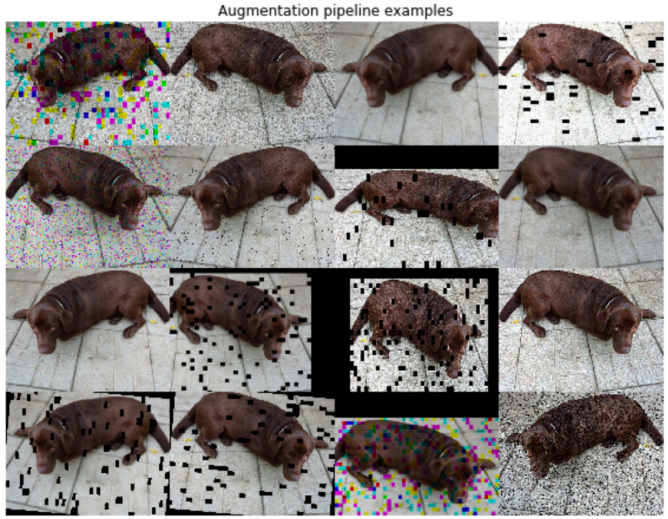
Definition 2.2. ***Few-Shot Learning*** (FSL) is a type of machine learning problems (specified by E, T and P), where E contains only a limited number of examples with supervised information for the target T.

task T	experience E		performance P
	supervised information	prior knowledge	
character generation [76]	a few examples of new character	pre-learned knowledge of parts and relations	pass rate of visual Turing test
drug toxicity discovery [4]	new molecule's limited assay	similar molecules' assays	classification accuracy
image classification [70]	a few labeled images for each class of the target T	raw images of other classes, or pre-trained models	classification accuracy

Yaqing Wang, Quanming Yao, James Kwok, Lionel M. Ni Generalizing from a Few Examples: A Survey on Few-Shot Learning, 2020. https://arxiv.org/abs/1904.05046

Data augmentation

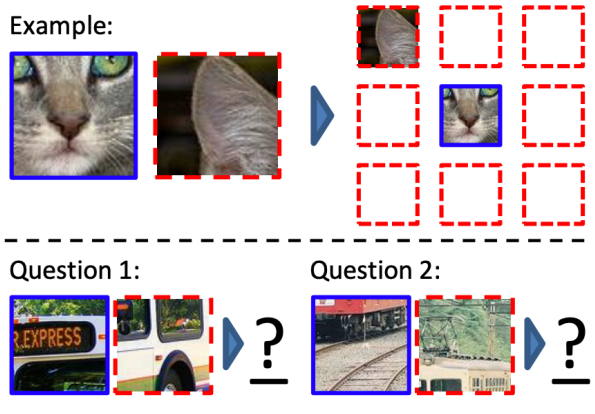
- Goal: generating synthetic samples that can help the DL model to better generalize and learn from scarse data

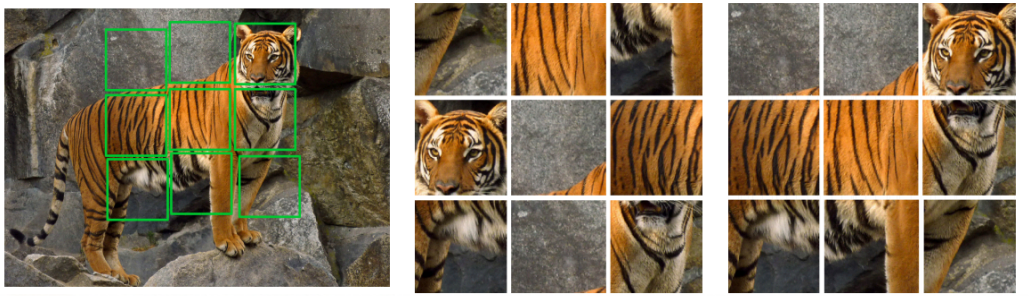


AutoAugment: Learning Augmentation Policies from Data Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, Quoc V. Le https://arxiv.org/abs/1805.09501
https://towardsdatascience.com/data-augmentation-for-deep-learning-4fe21d1a4eb9

Self supervised learning

- Idea: pretrain a model for “pretext” tasks that are associated to the task (downstream) of ultimate interest





Carl Doersch, Abhinav Gupta, Alexei A. Efros. Unsupervised Visual Representation Learning by Context Prediction, 2015, arxiv.org/abs/1505.05192
Mehdi Noroozi and Paolo Favaro Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles, 2016. https://arxiv.org/pdf/1603.09246.pdf

https://www.fast.ai/2020/01/13/self_supervised/

Comentarios finales

- Beneficios
  - Extremadamente útiles para aprender representaciones y modelos a partir de grandes cantidades de datos “crudos”
  - Entrenamiento eficiente, capacidad de procesamiento paralelo masivo
  - Capacidad de generalización sobre saliente
- Limitantes
  - Requieren de grandes cantidades de datos (aunque hay versiones para celular)
  - Demandan recursos computacionales
  - Modelos de caja negra, no explicables / interpretables

Comentarios finales

- Introducción al aprendizaje profundo, muy superficial
- El aprendizaje profundo domina en las principales sub áreas de IA: CV, PR, NLP, SP,
- Es complicado seguir el progreso de aprendizaje profundo

Lecturas sugeridas

- https://github.com/floodsung/Deep-Learning-Papers-Reading-Roadmap

INTELIGENCIA ARTIFICIAL

Consorcio de Centros Públicos Conacyt

¡GRACIAS!

CONACYT | CIMAT

México 2021 Año de la Independencia