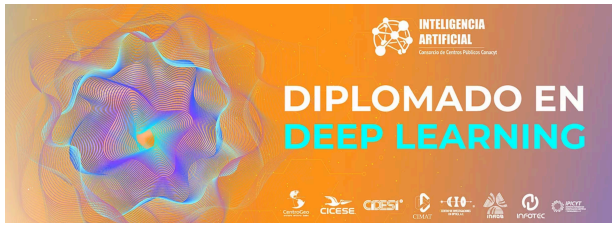


Variantes y Retos del Aprendizaje Profundo

Hugo Jair Escalante

hugojair@inaoep.mx

https://ccc.inaoep.mx/~hugojair/

@hugojair

Contenido

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

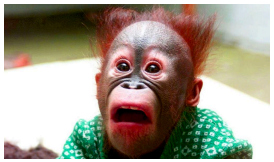
- Retos, oportunidades de investigación

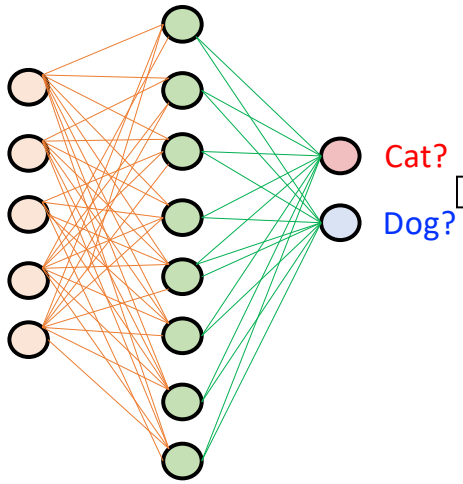
Aprendizaje profundo
Principales variantes

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*





Outputs

Variables - U

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									

Variables - W

U W

8x2 double

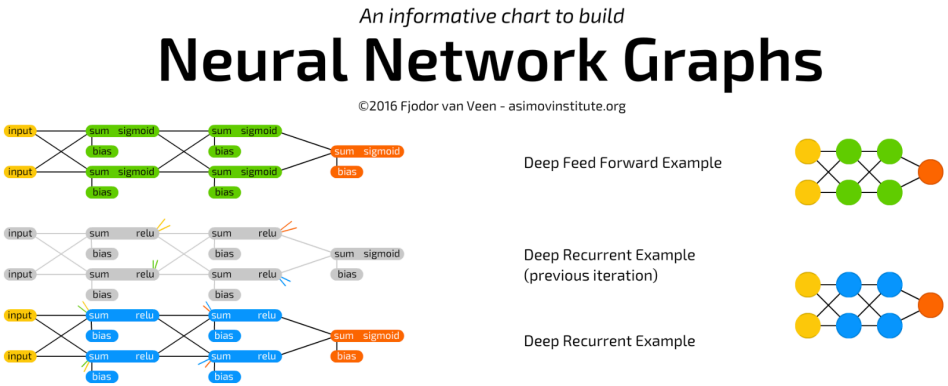
	1	2	3
1	0.5884	0.1427	
2	0.9775	0.6085	
3	0.2137	0.6286	
4	0.6291	0.0238	
5	0.7810	0.7787	
6	0.3571	0.4912	
7	0.9407	0.4116	
8	0.9616	0.7775	
9			
10			
11			
12			
13			

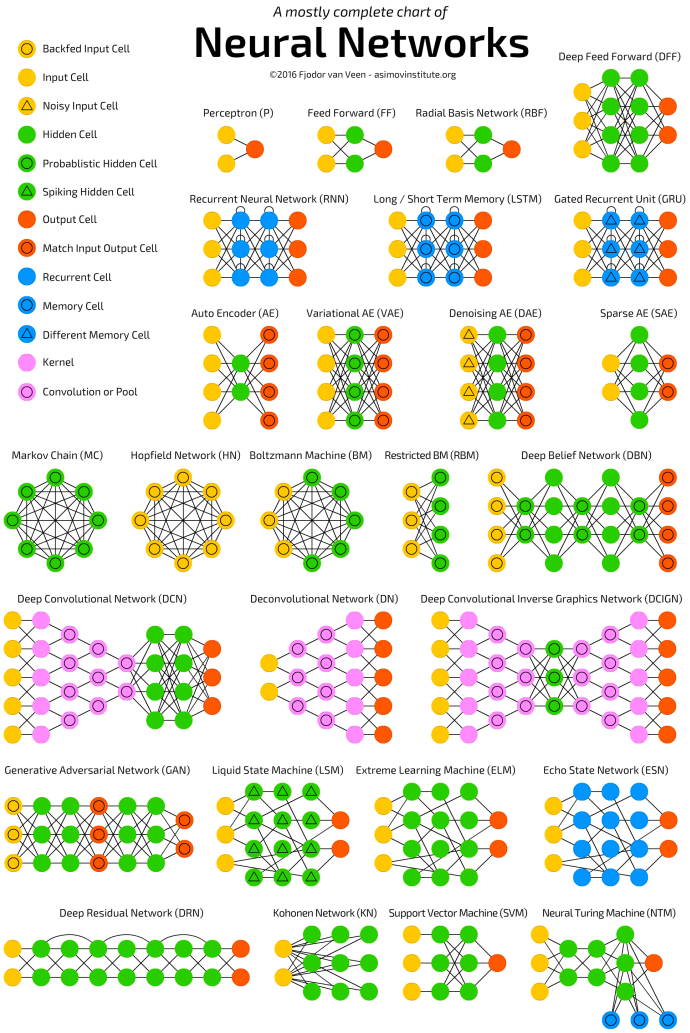
$$f(\mathbf{x}) = s(\mathbf{w}\phi(\mathbf{x}) + b)$$

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/

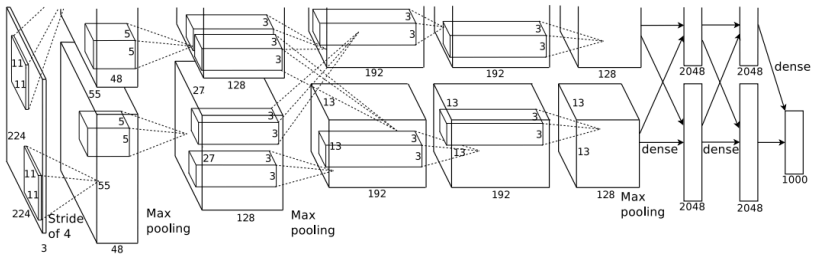
Variantes de aprendizaje profundo

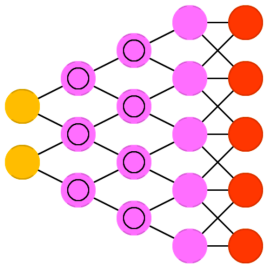
- Principales modelos de DL
  - Redes neuronales profundas (DNNs, MLPs)
  - Redes neuronales convolucionales (CNNs)
  - Restricted Boltzman machines
  - Deep belief networks
  - Autoencoders
  - Long short term memory networks

- Otros paradigmas
  - Redes residuals
  - Inception networks
  - Gated recurrent NNs
  - Generative adversarial networks

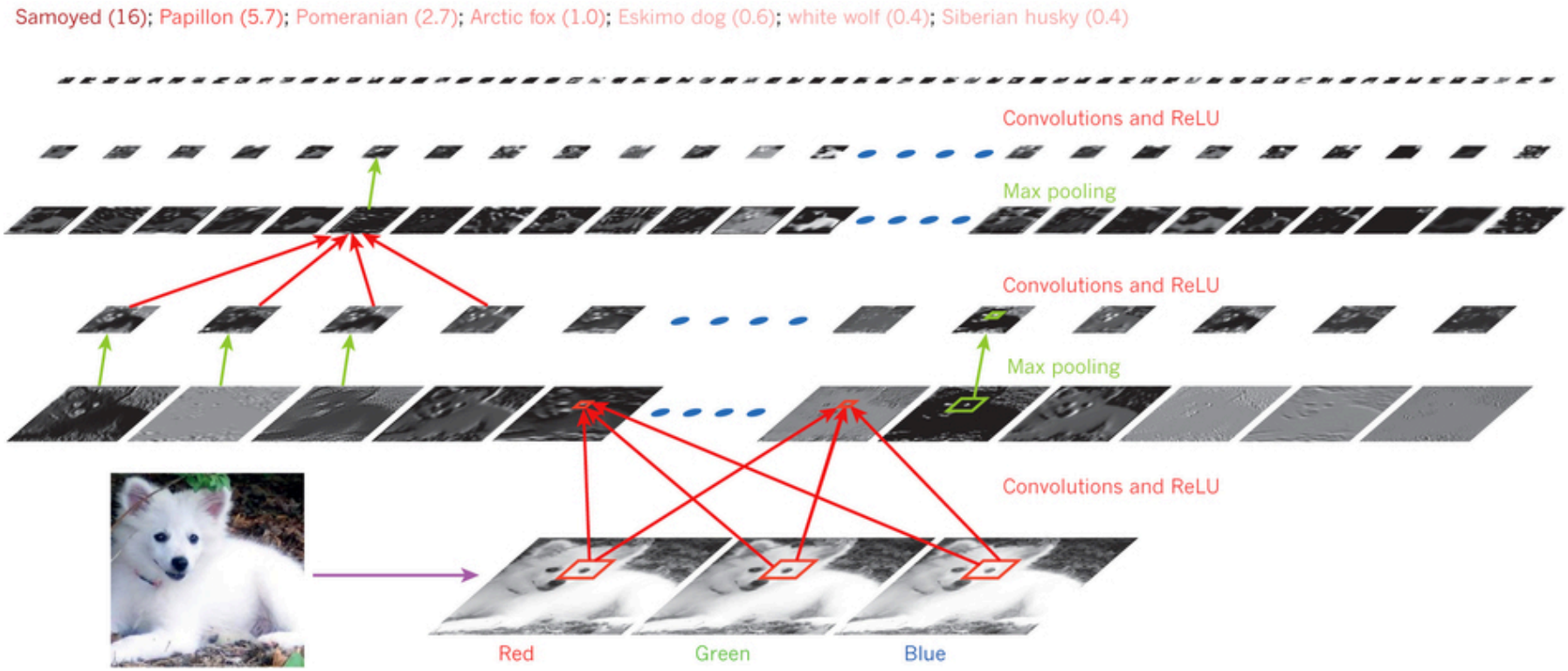
CNNs – redes neuronales convolucionales

- 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



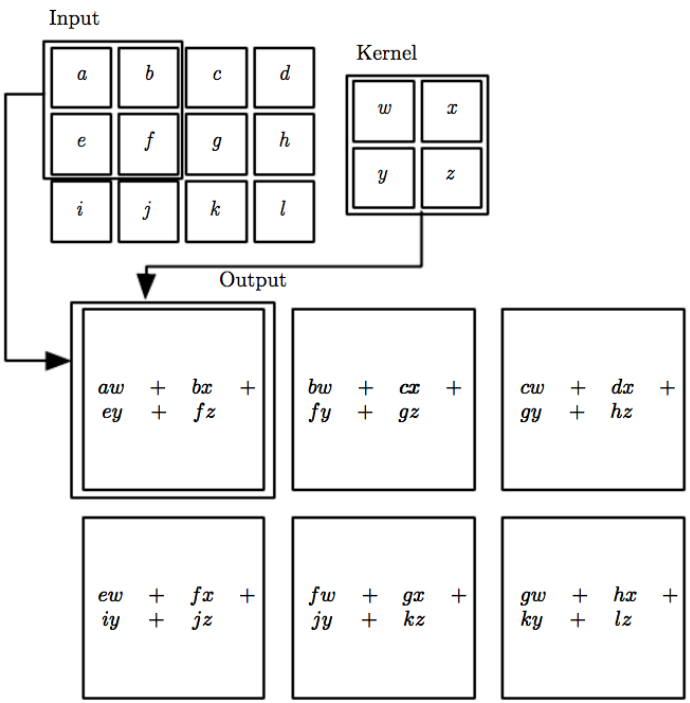


CNNs


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

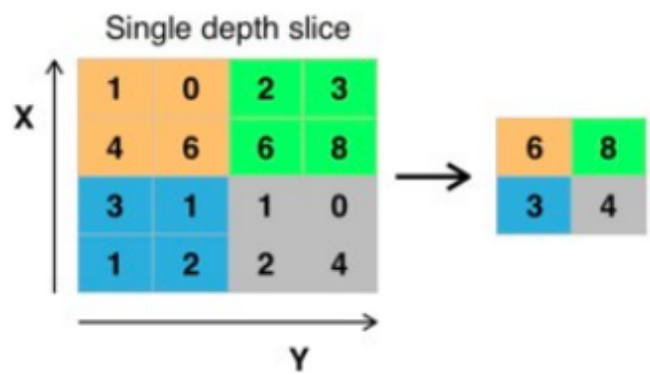
CNNs - componentes

- Convolución 2D

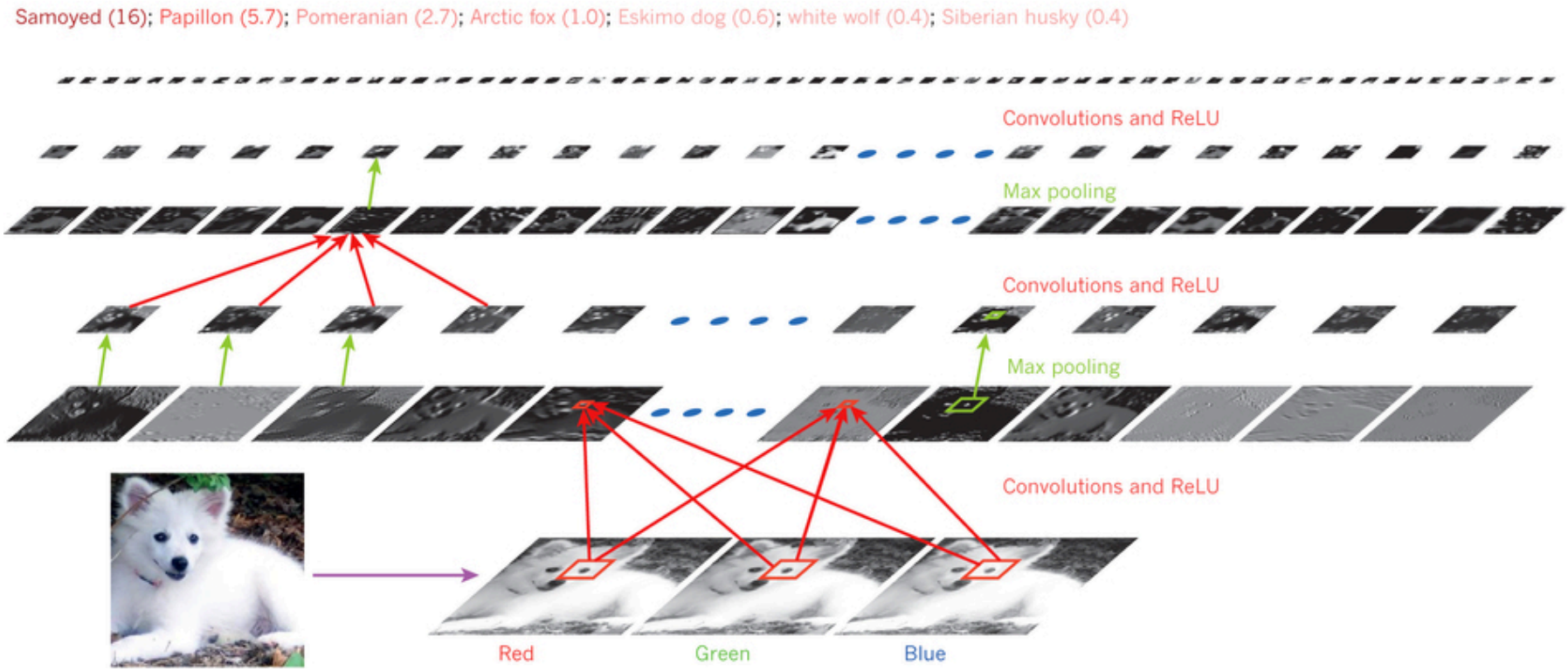


CNNs - componentes

- **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

Convolutional neural networks

- These networks employ the mathematical operation called convolution
  - CNNs are simply NNs that use convolution in place of matrix multiplication in at least one of their layers

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

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

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

Convolutional neural networks

- Convolution in NN terminology:
  - x – the input
  - w – the kernel
  - s – the feature map

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

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

- Discrete convolution:

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

Convolutional neural networks

- 2D convolution

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

- Convolution is commutative

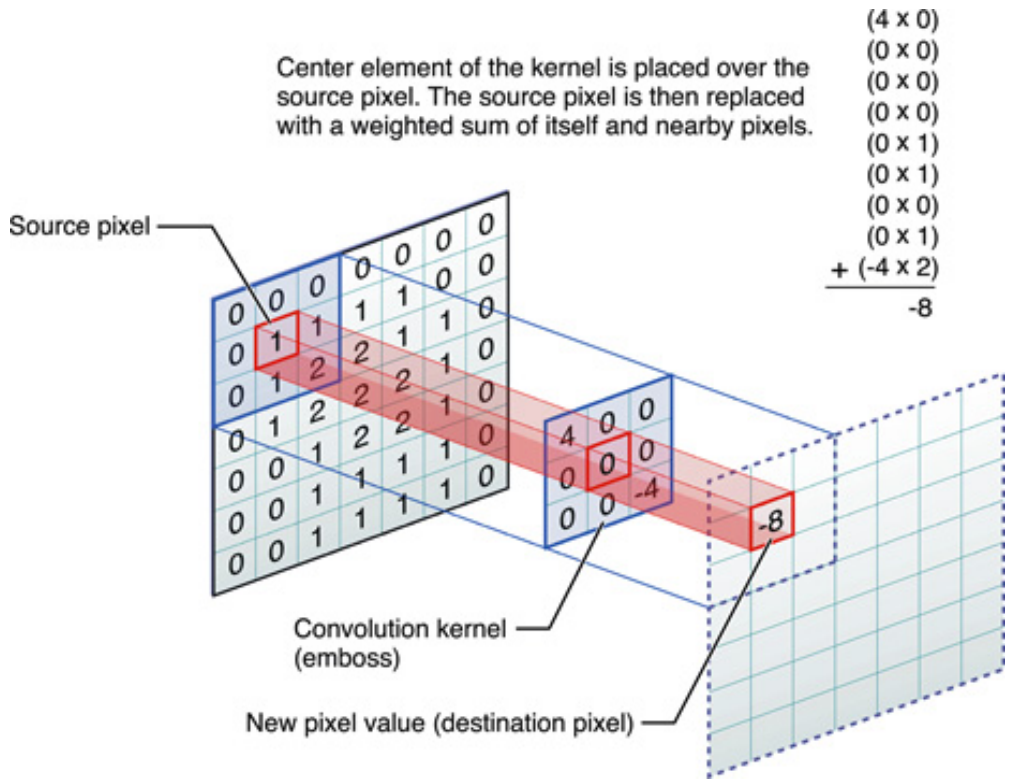
$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n).$$

Convolutional neural networks

- 2D convolution

Original Emboss

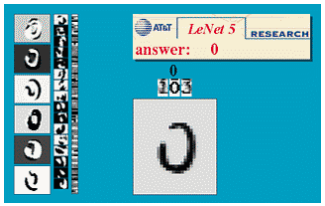
-2	-2	0
-2	6	0
0	0	0



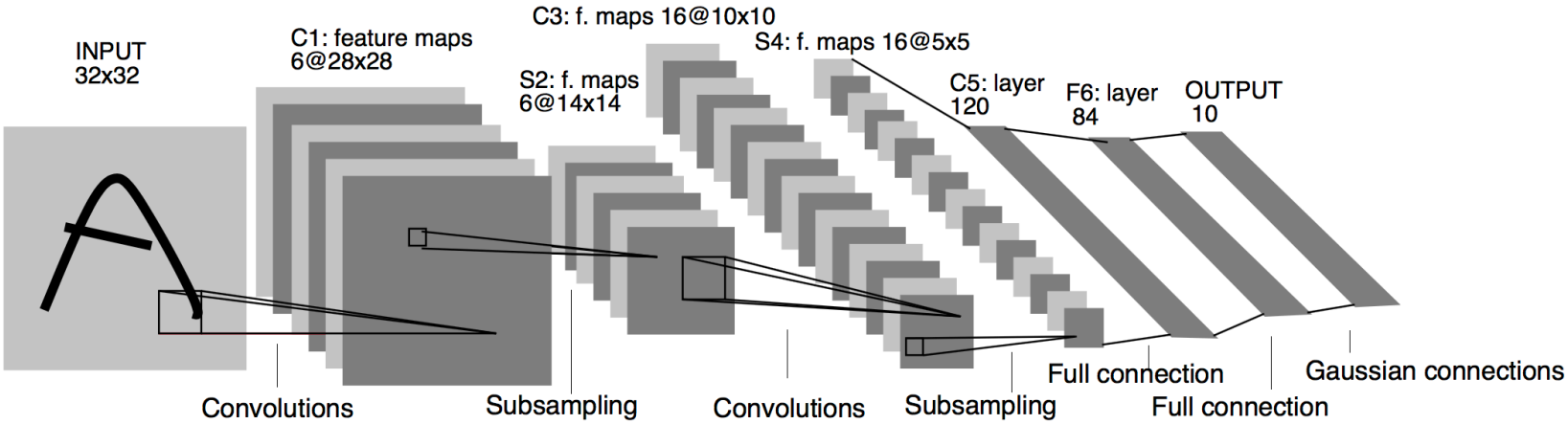
https://developer.apple.com/library/content/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html

CNNs

- Arquitectura típica I



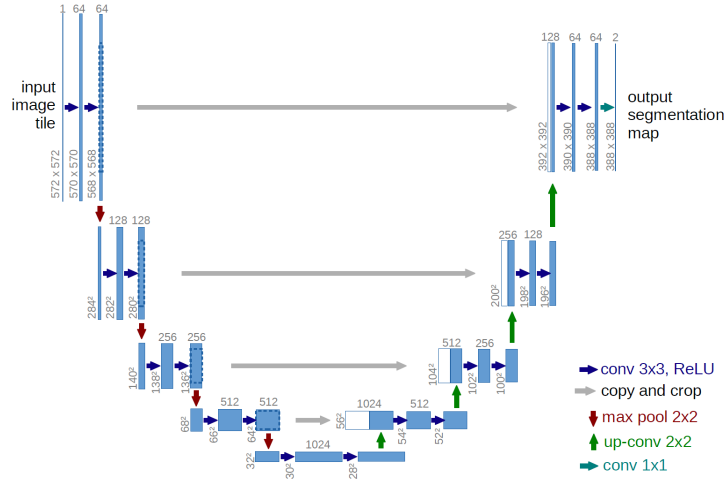
http://yann.lecun.com/exdb/lenet/

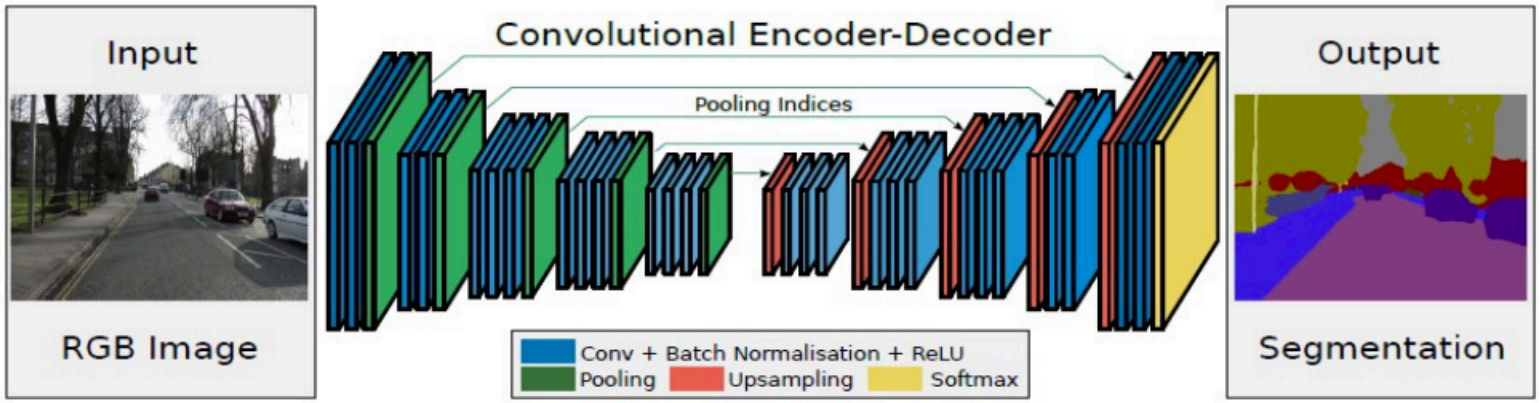


Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, november 1998.

CNNs

- Arquitectura típica II



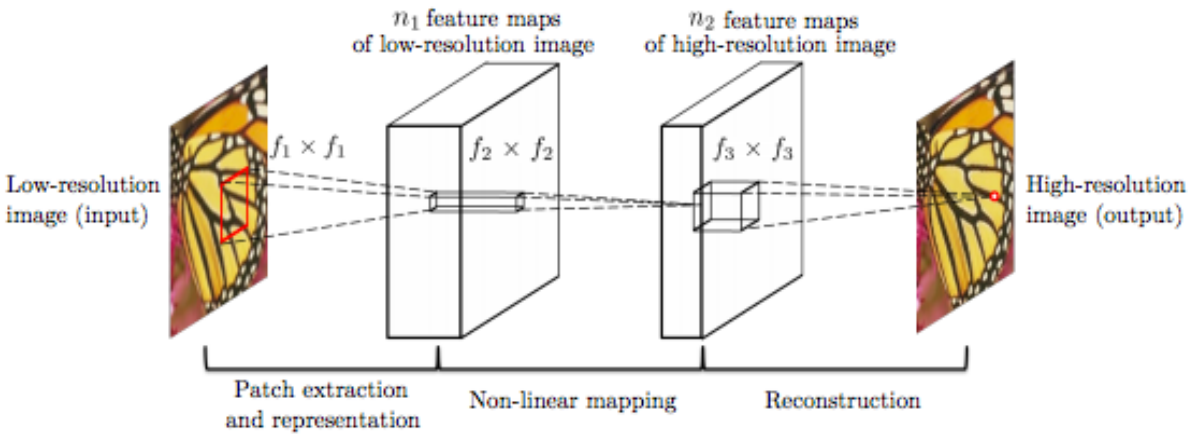


Vijay Badrinarayanan, et al. . **SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,** *ArXiv, 1511.00561, 2015*

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

CNNs

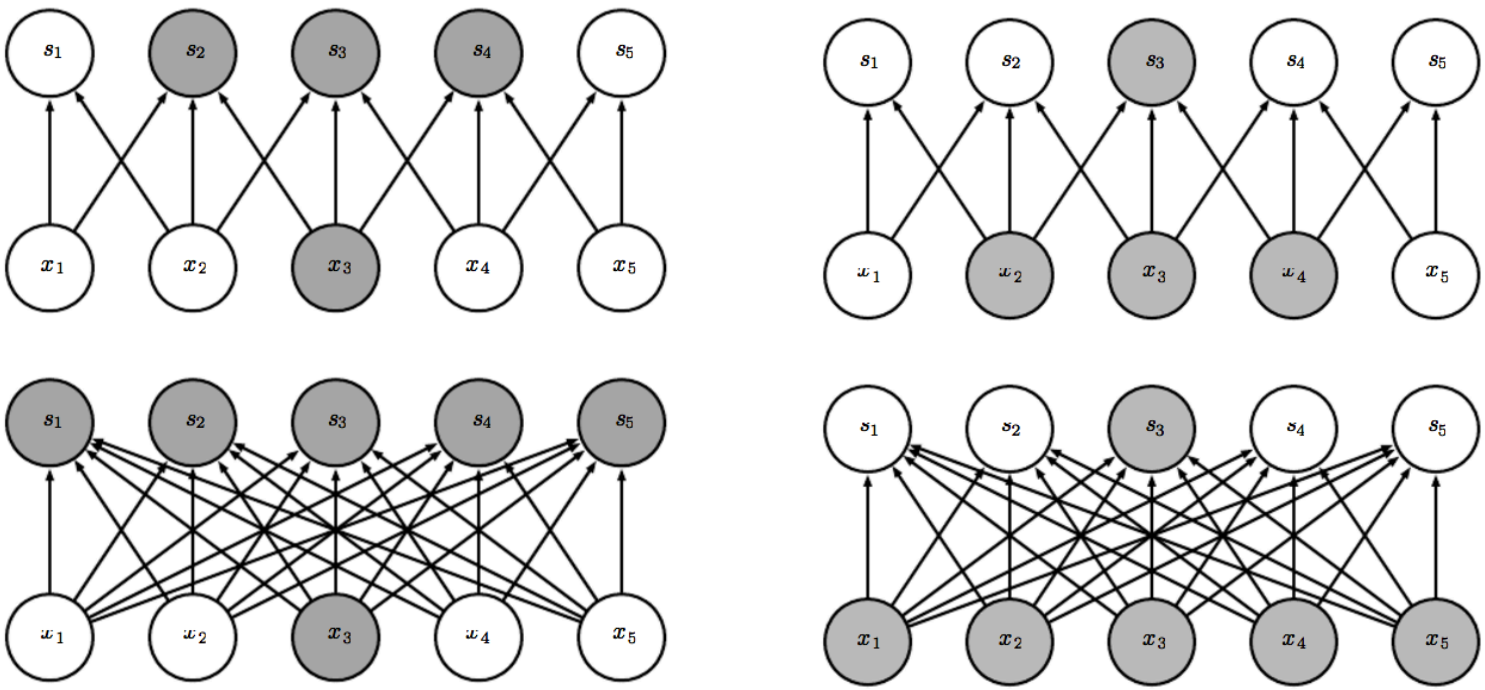
- Arquitectura típica II



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

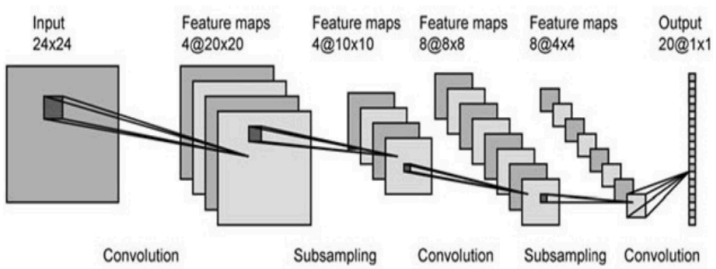
Convolutional neural networks

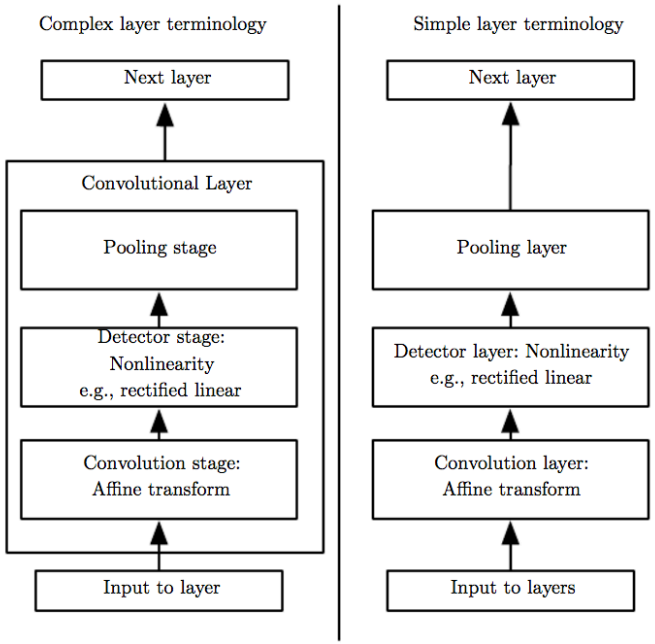
- Why convolution layers?
  - Sparse connectivity: less weights/parameters



CNNs

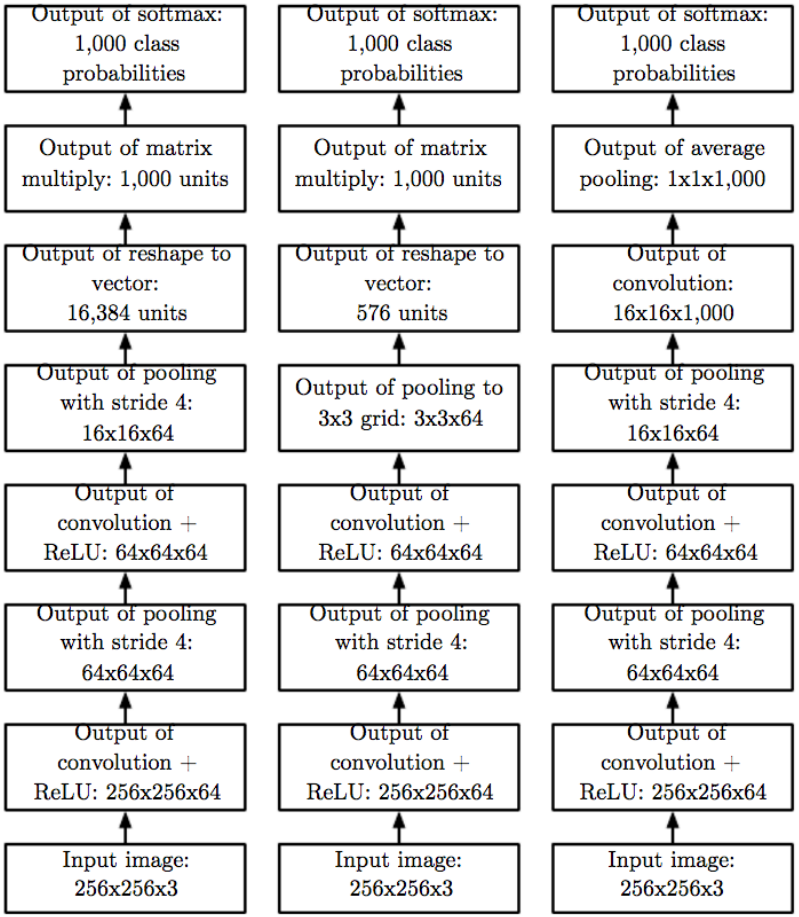
- Arquitectura típica de una capa





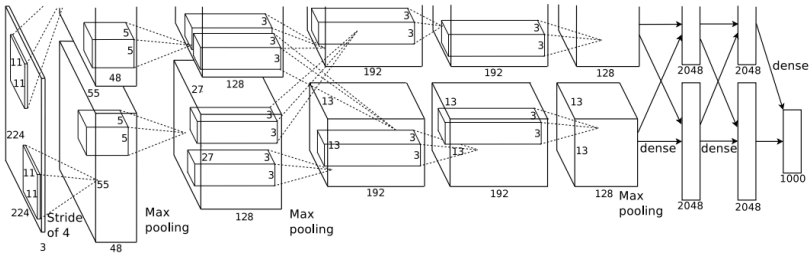
CNNs

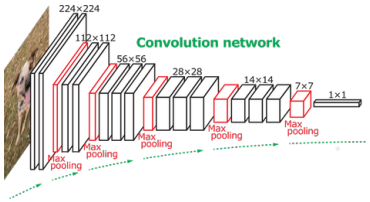
- Arquitecturas comunes

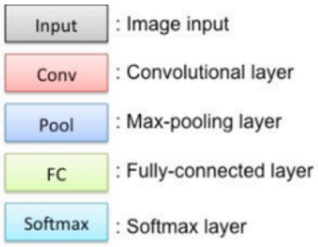


CNNs

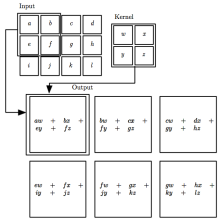
- Qué tan profundo?

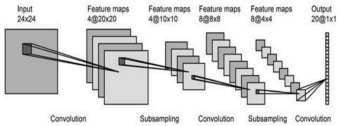




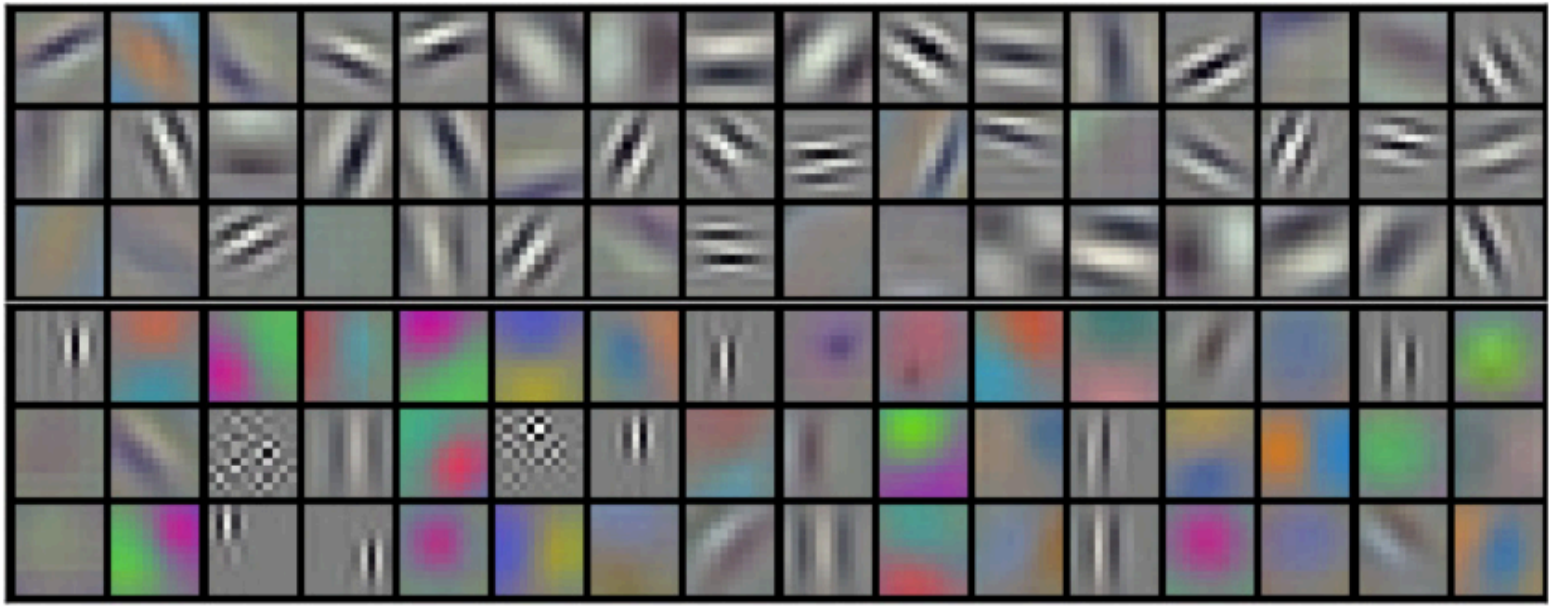


CNNs



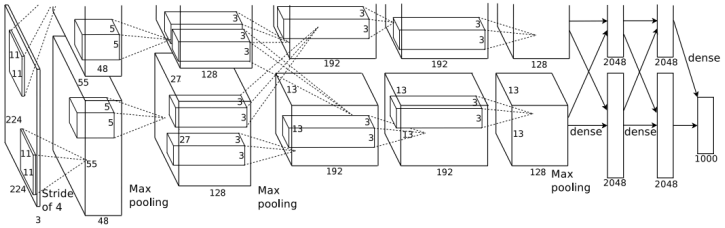


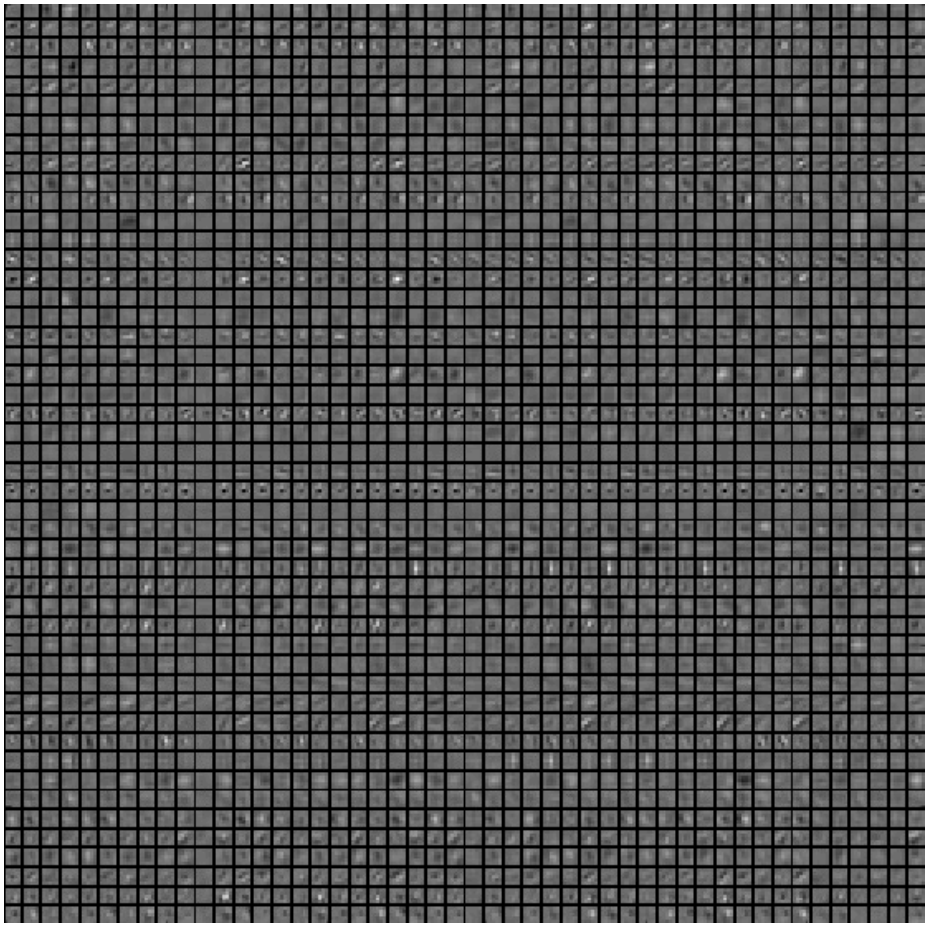
- Qué son los pesos aprendidos?



CNNs

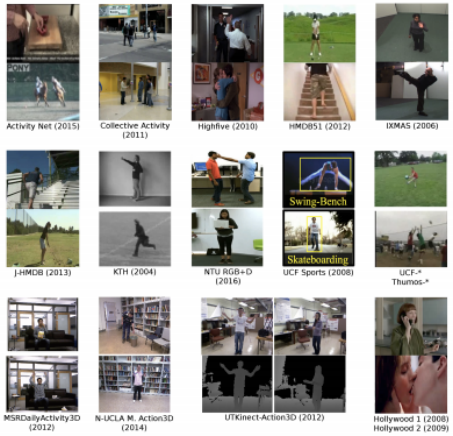
- Qué son los pesos aprendidos?

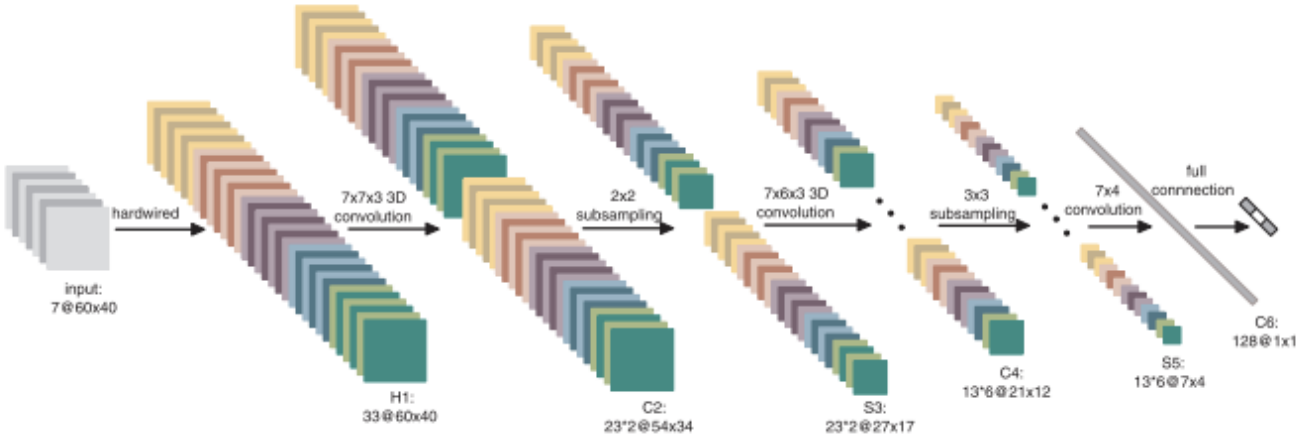




CNNs

- Qué pasa con 3D?

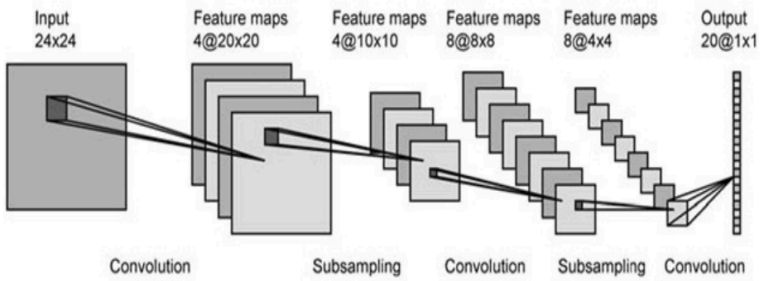


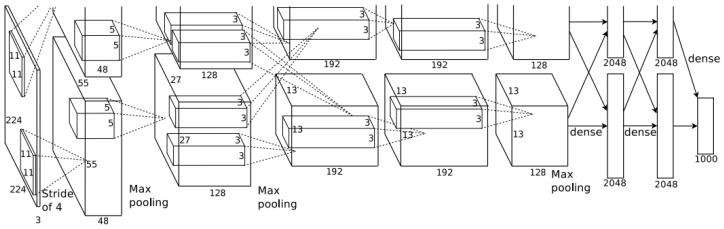


Shuiwang Ji, Wei Xu, Ming Yang, Kai Yu. **3D Convolutional Neural Networks for Human Action Recognition.** TPAMI Vol. 35(1):221-231, 2013

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

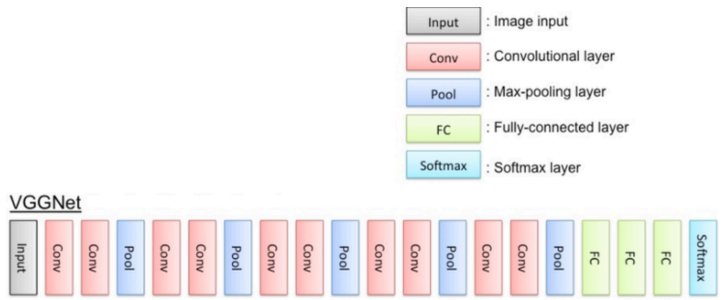


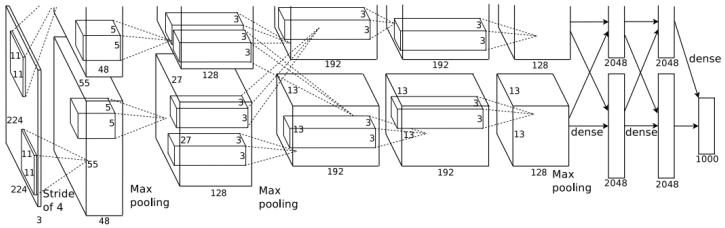


Yosinski J, Clune J, Bengio Y, and Lipson H. **How transferable are features in deep neural networks?** In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation, 2014

CNNs

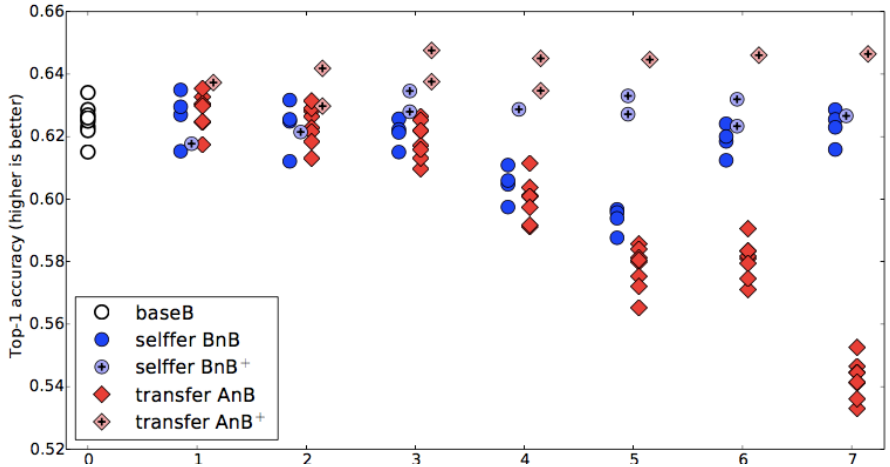
- CNNs pre-entrenadas populares:
  - AlexNet
  - VGG
  - GoogleNet
  - PlacesNet
  - FaceNet





Convolutional neural networks

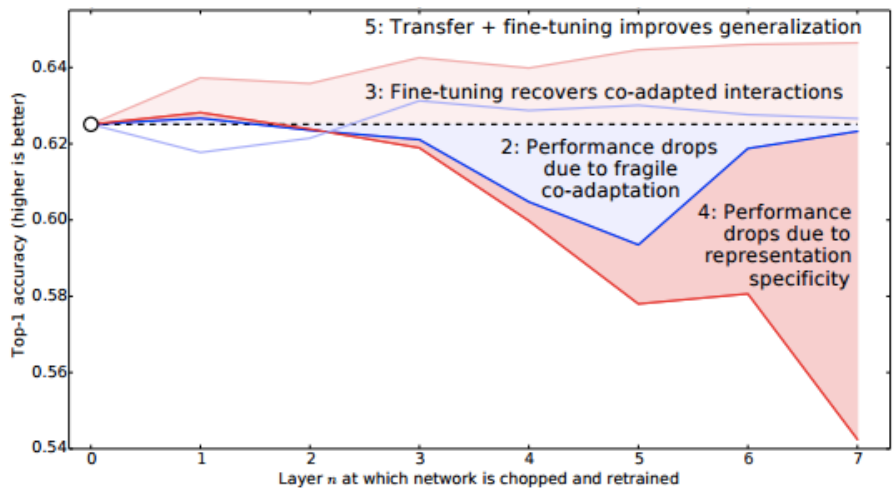
- **Transfer learning:** it is common to re-use pretrained architectures (the weights) trained with millions of images



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

- **Transfer learning:** it is common to re-use pretrained architectures (the weights) trained with millions of images



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)
  - Restricted Boltzman machines
  - Deep belief networks
  - Autoencoders
  - Long short term memory networks

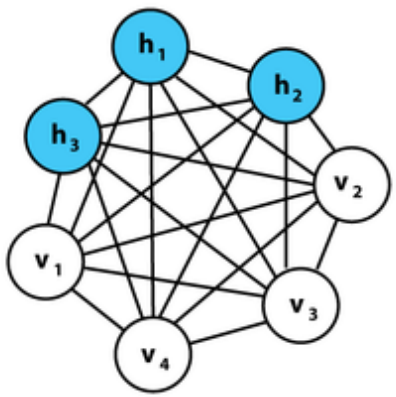
Modelos generativos profundos!

- Otros paradigmas
  - Redes residuals
  - Inception networks
  - Gated recurrent NNs
  - Generative adversarial networks

Modelos generativos profundos

- Máquinas de Boltzmann. Un modelo basado en energía que se define sobre un vector d-dimensional de variables binarias $\mathbf{x} \in \{0,1\}^d$
- La distribución conjunta del modelo está dada por:

$$P(\boldsymbol{x}) = \frac{\exp\left(-E(\boldsymbol{x})\right)}{Z},$$



- Donde:

Solo unidades visibles

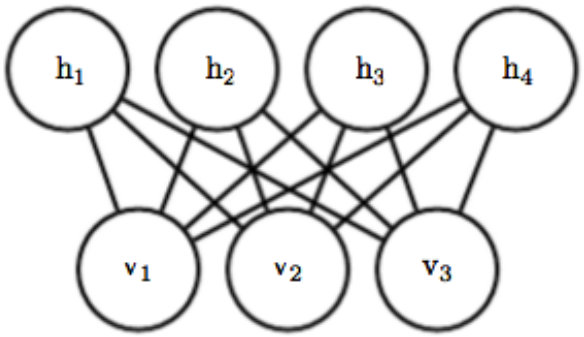
$$E(\boldsymbol{x}) = -\boldsymbol{x}^\top \boldsymbol{U}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x},$$

Unidades visibles y ocultas

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{v}^\top \boldsymbol{R}\boldsymbol{v} - \boldsymbol{v}^\top \boldsymbol{W}\boldsymbol{h} - \boldsymbol{h}^\top \boldsymbol{S}\boldsymbol{h} - \boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h}.$$

Modelos generativos profundos

- Máquinas restringidas de Boltzmann (RBMs). Una máquina de Boltzmann restringida a:
  - Tener dos capas: con unidades visibles y ocultas
  - No hay conexiones entre unidades en la misma capa
  - Las unidades tienen típicamente funciones de activación de Bernoulli
  - La primitiva para construir modelos generativos profundos



Modelos generativos profundos

- Una RBM es un modelo basado en energía, con la probabilidad conjunta especificada por una función de energía:

$$P(\mathbf{v} = \boldsymbol{v}, \mathbf{h} = \boldsymbol{h}) = \frac{1}{Z} \exp\left(-E(\boldsymbol{v}, \boldsymbol{h})\right).$$

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h},$$

$$Z = \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \exp\left\{-E(\boldsymbol{v}, \boldsymbol{h})\right\}.$$

The Energy of a joint configuration

(ignoring terms to do with biases)

binary state of visible unit i

binary state of hidden unit j

$$E(v,h) = -\sum_{i,j} v_i h_j w_{ij}$$

Energy with configuration v on the visible units and h on the hidden units

weight between units i and j

$$-\frac{\partial E(v,h)}{\partial w_{ij}} = v_i h_j$$

Slide from G. Hinton's tutorial at NIPS 2007

Modelos generativos profundos

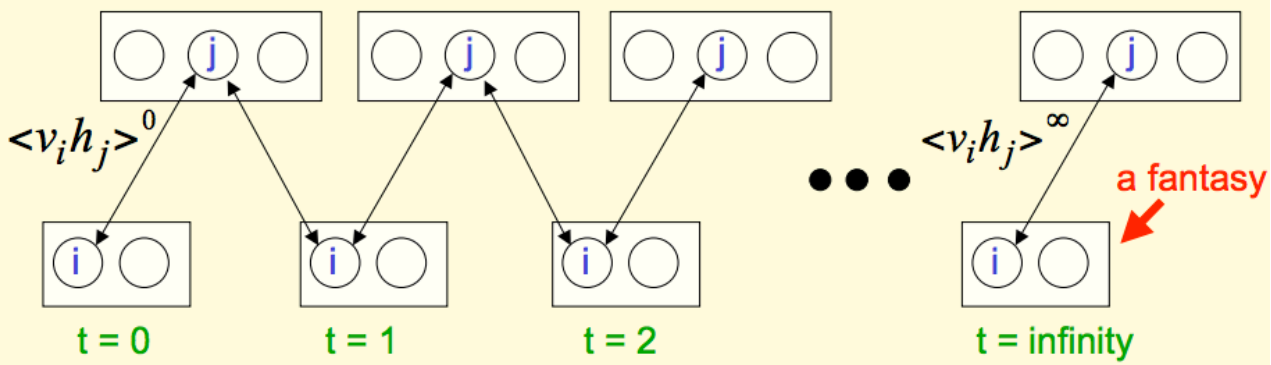
- The bipartite graph structure of the RBM has the special property that its conditional distributions $P(\mathbf{h}|\mathbf{v})$ and $P(\mathbf{v}|\mathbf{h})$ are factorial and simple to compute / sample from:

$$p(\mathbf{h} \mid \mathbf{v}) = \Pi_i p(\mathrm{h}_i \mid \mathbf{v}) \qquad p(\mathbf{v} \mid \mathbf{h}) = \Pi_i p(\mathrm{v}_i \mid \mathbf{h}).$$

$$p(h_j = 1|\mathbf{v};\theta) = \sigma\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right)$$

$$p(v_i = 1|\mathbf{h};\theta) = \sigma\left(\sum_{j=1}^{J} w_{ij} h_i + b_i\right)$$

A picture of the maximum likelihood learning algorithm for an RBM

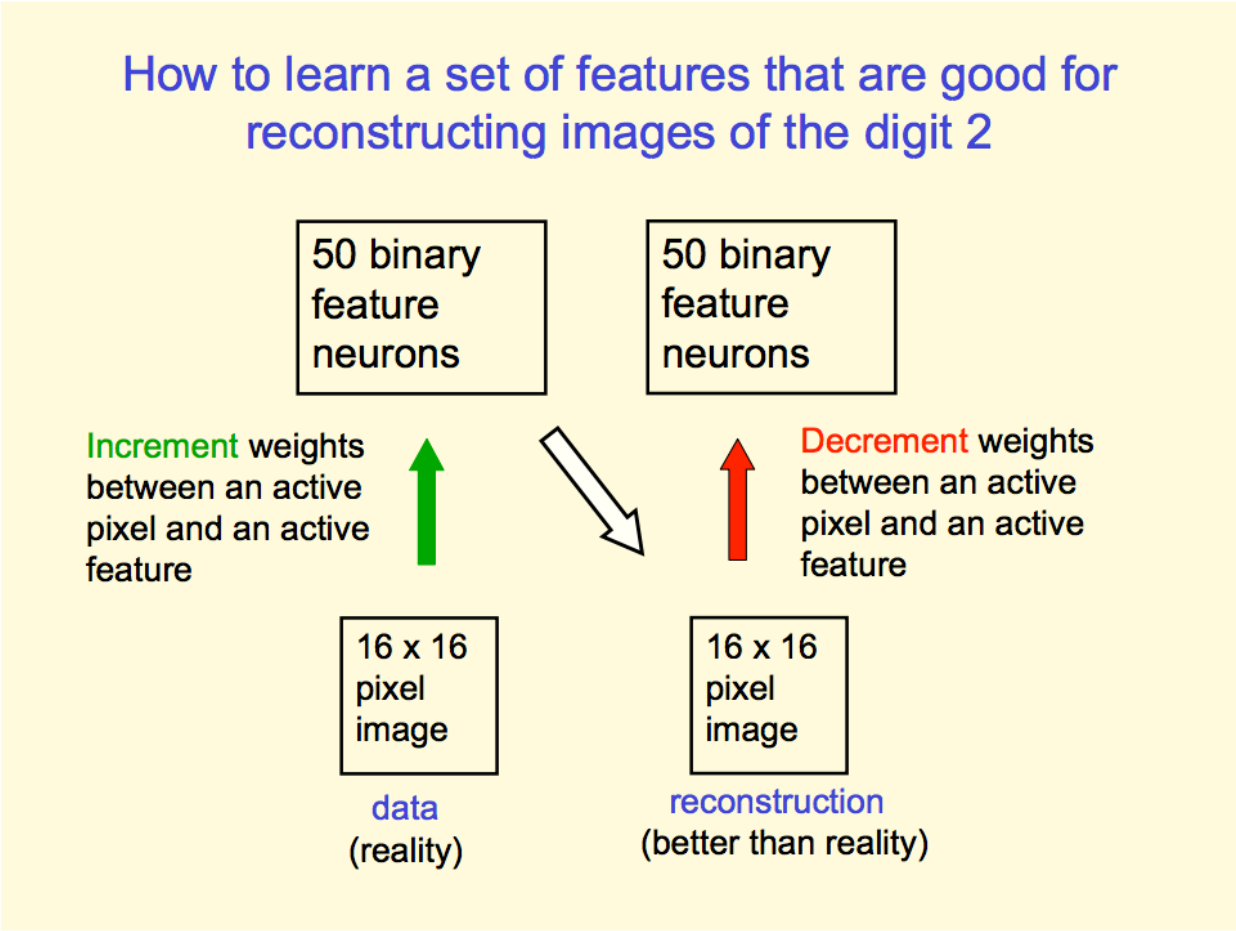


Start with a training vector on the visible units.

Then alternate between updating all the hidden units in parallel and updating all the visible units in parallel.

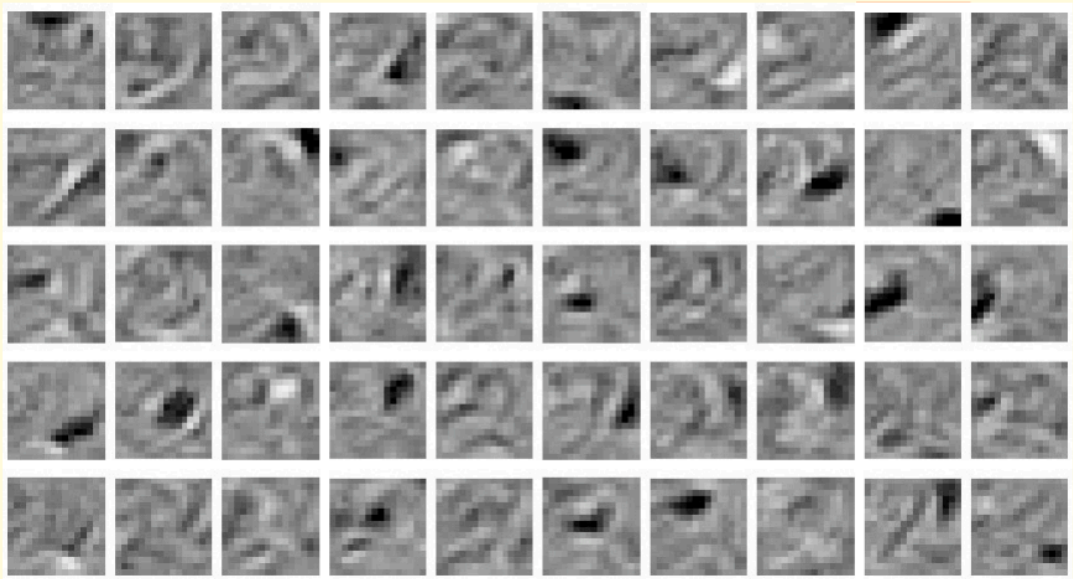
$$\frac{\partial \log p(v)}{\partial w_{ij}} = <v_i h_j>^0 - <v_i h_j>^\infty$$

Slide from G. Hinton's tutorial at NIPS 2007



Slide from G. Hinton's tutorial at NIPS 2007

The final 50 x 256 weights

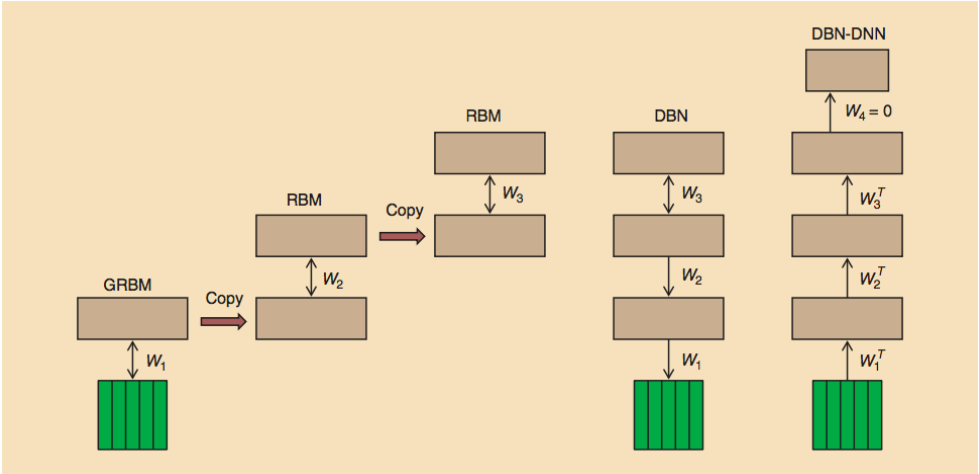


Each neuron grabs a different feature.

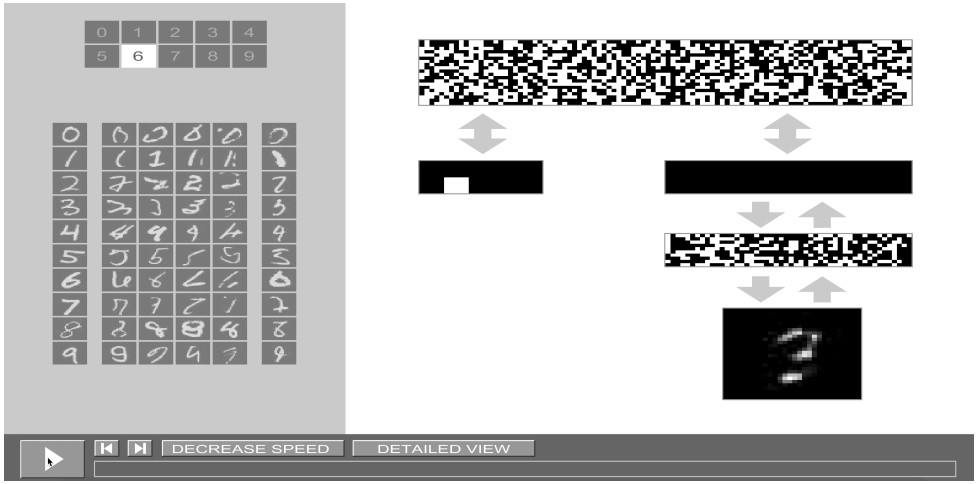
Slide from G. Hinton's tutorial at NIPS 2007

Modelos generativos profundos

- Este algoritmo fue un parteaguas en el renacimiento de las DNNs en 2006:
  - Idea: entrenar modelos profundos usando capas anidadas de RBMs
  - Cada capa se pre-entrena independientemente
  - En una etapa final, la red complete es sujeta a un ajuste fino de parámetros



Modelos generativos profundos



http://www.cs.toronto.edu/~hinton/digits.html

Resultados sobresalientes en ese entonces!

Version of MNIST Task	Learning Algorithm	Test Error %
Permutation invariant	Our generative model: 784 → 500 → 500 ↔ 2000 ↔ 10	1.25
Permutation invariant	Support vector machine: degree 9 polynomial kernel	1.4
Permutation invariant	Backprop: 784 → 500 → 300 → 10 cross-entropy and weight-decay	1.51
Permutation invariant	Backprop: 784 → 800 → 10 cross-entropy and early stopping	1.53
Permutation invariant	Backprop: 784 → 500 → 150 → 10 squared error and on-line updates	2.95
Permutation invariant	Nearest neighbor: all 60,000 examples and L3 norm	2.8
Permutation invariant	Nearest neighbor: all 60,000 examples and L2 norm	3.1
Permutation invariant	Nearest neighbor: 20,000 examples and L3 norm	4.0
Permutation invariant	Nearest neighbor: 20,000 examples and L2 norm	4.4
Unpermuted images; extra data from elastic deformations	Backprop: cross-entropy and early-stopping convolutional neural net	0.4
Unpermuted de-skewed images; extra data from 2 pixel translations	Virtual SVM: degree 9 polynomial kernel	0.56
Unpermuted images	Shape-context features: hand-coded matching	0.63
Unpermuted images; extra data from affine transformations	Backprop in LeNet5: convolutional neural net	0.8
Unpermuted images	Backprop in LeNet5: convolutional neural net	0.95

Hinton, G. E., Osindero, S. and Teh, Y. (2006) A fast learning algorithm for deep belief nets. Neural Computation, 18, pp 1527-1554.

Modelos generativos profundos

- Modelan la probabilidad conjunta de variables observables y ocultas
- Aprendizaje eficiente de parámetros
- No se restringen a unidades de Bernoulli
- No han tenido tanto auge de la comunidad (con respecto, e.g., a CNNs)

Hinton, G. E., Osindero, S. and Teh, Y. (2006) A fast learning algorithm for deep belief nets. Neural Computation, 18, pp 1527-1554.

Variantes de aprendizaje profundo

- Principales modelos de DL
  - Redes neuronales profundas (DNNs, MLPs)
  - Redes neuronales convolucionales (CNNs)
  - Restricted Boltzman machines
  - Deep belief networks
  - Autoencoders
  - Long short term memory networks

Aprendizaje no supervisado

- Otros paradigmas
  - Redes residuals
  - Inception networks
  - Gated recurrent NNs
  - Generative adversarial networks

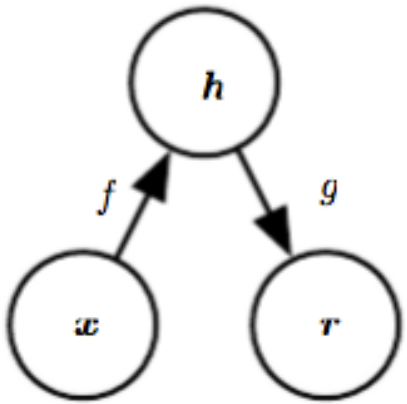
Autoencoders

- 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: $\mathbf{r} = g(\mathbf{h})$

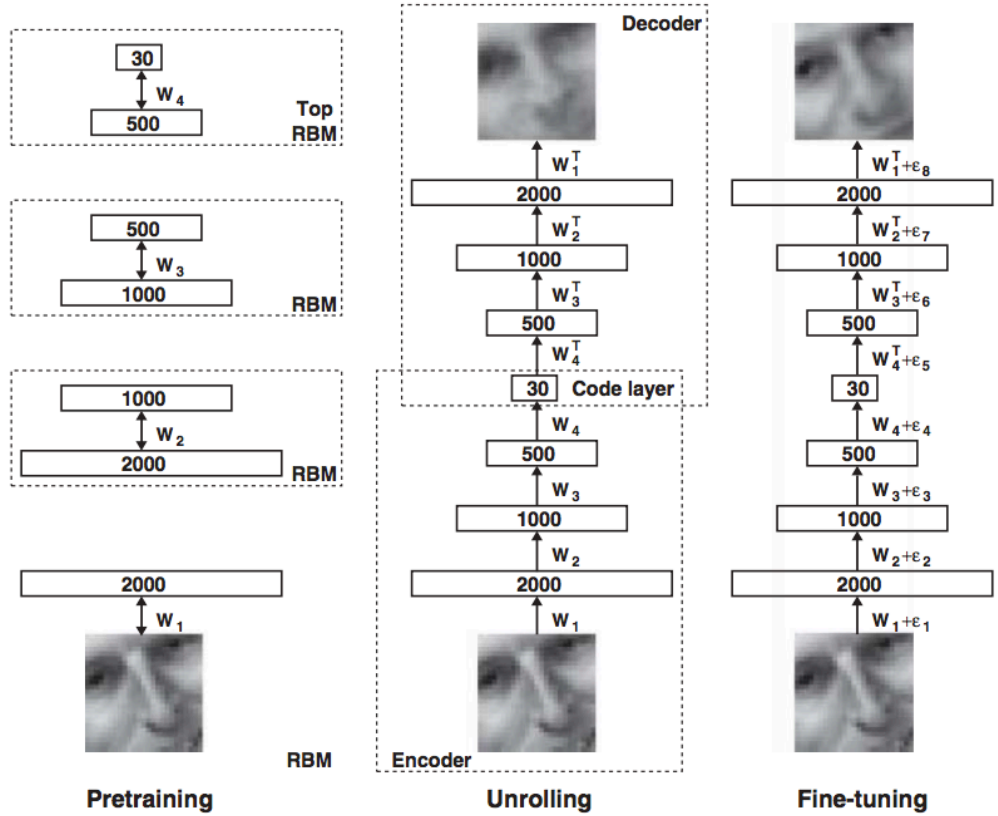
- Usualmente la dimensionalidad de h es menor que la de las entradas

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



Autoencoders

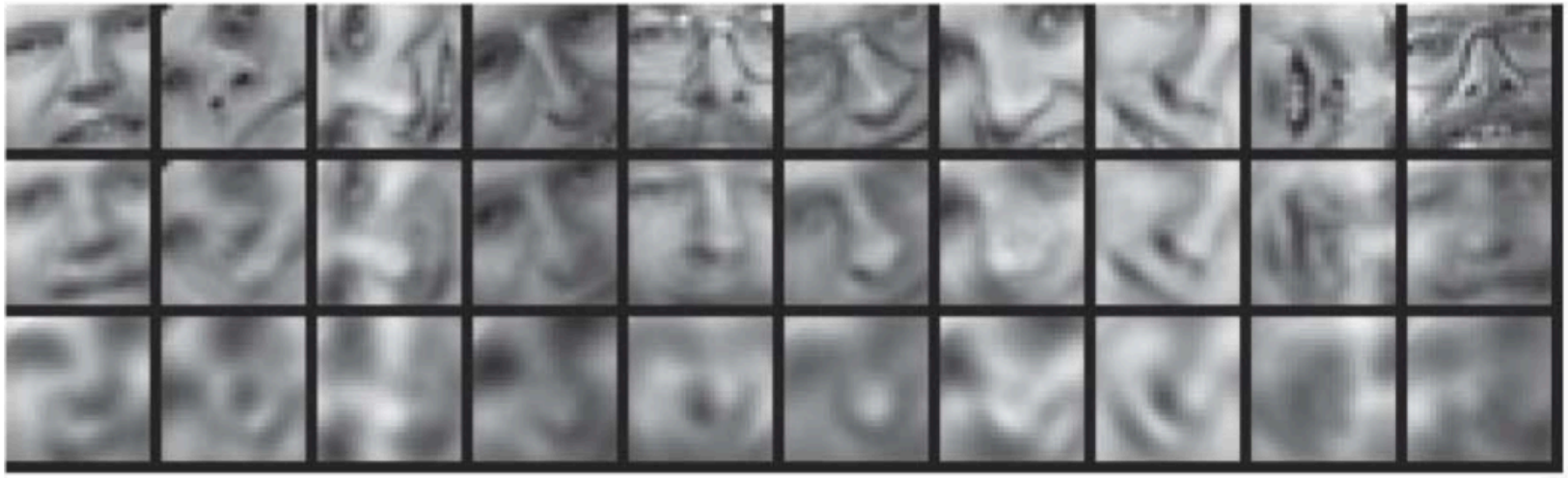
- Autoencoders profundos!
  - Pre entrenamiento de capas usando RBMs
  - Unfolding!
  - Fine tunning con backprop



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

Autoencoders

- Autoencoders profundos!



Variantes de aprendizaje profundo

- Principales modelos de DL
  - Redes neuronales profundas (DNNs, MLPs)
  - Redes neuronales convolucionales (CNNs)
  - Restricted Boltzman machines
  - Deep belief networks
  - Autoencoders
  - Long short term memory networks

Aprendizaje secuencial

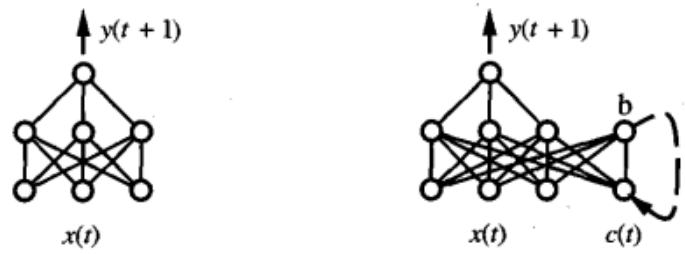
- Otros paradigmas
  - Redes residuals
  - Inception networks
  - Gated recurrent NNs
  - Generative adversarial networks

Modelando datos secuenciales

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

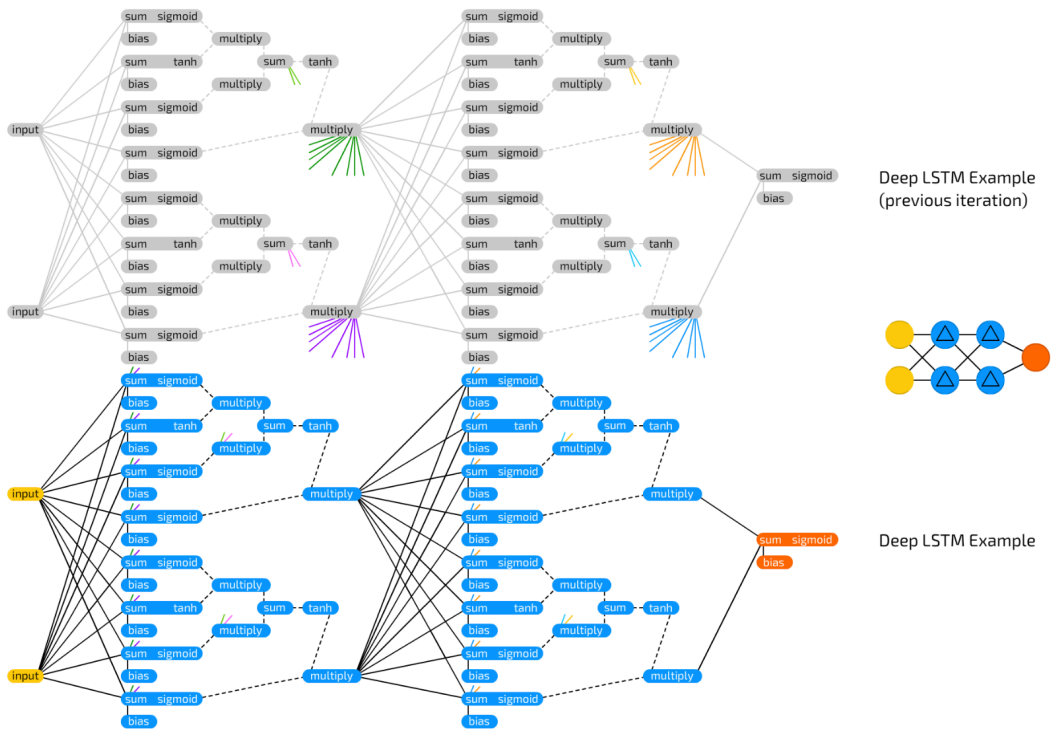
- Las unidades ocultas pueden especificarse como:

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



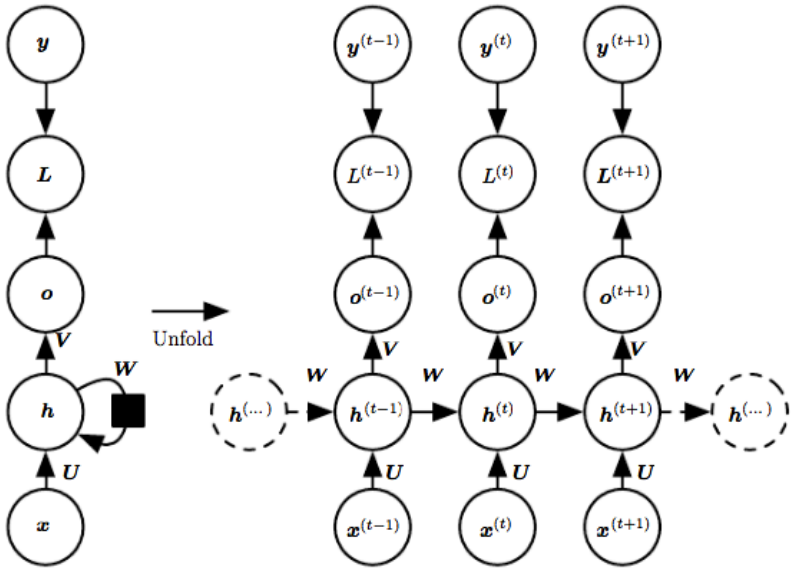
(a) Feedforward network (b) Recurrent network

Modelando datos secuenciales



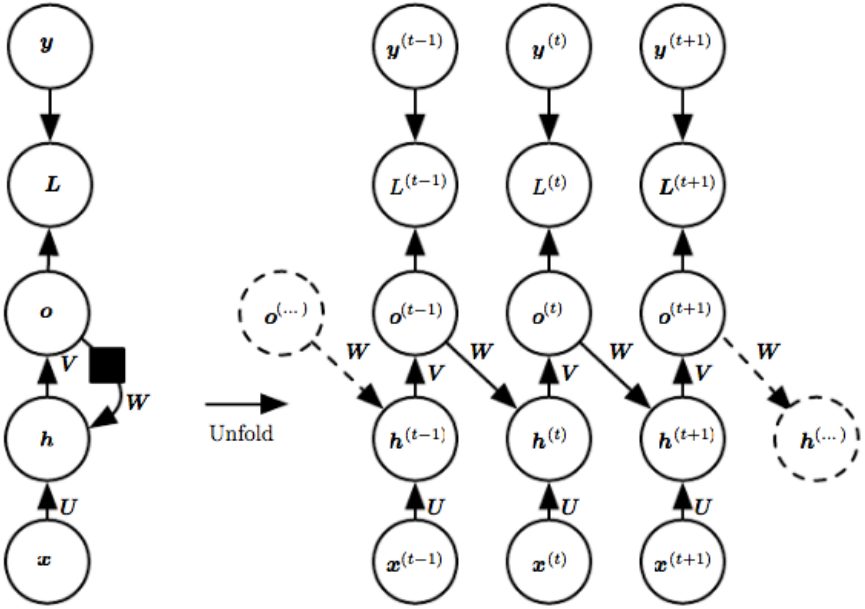
Modelando datos secuenciales

- Arquitectura típica I



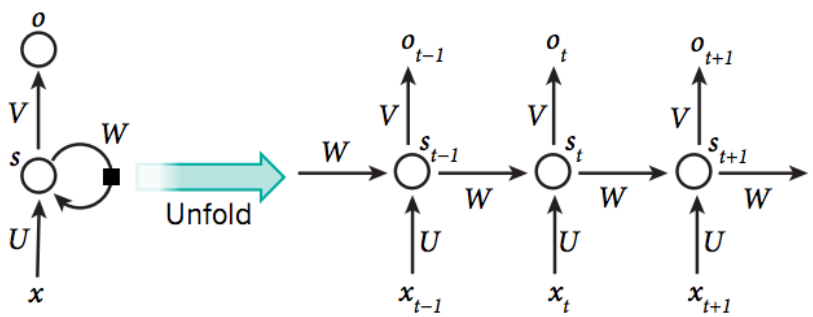
Modelando datos secuenciales

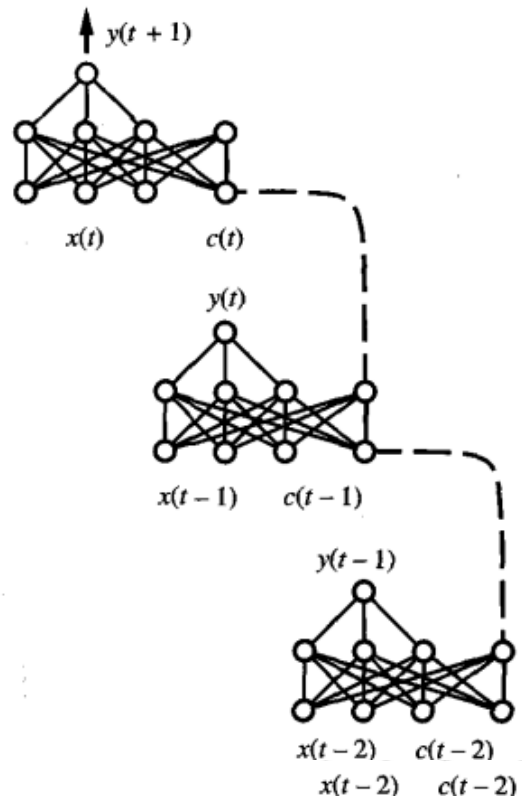
- Arquitectura típica II



Modelando datos secuenciales

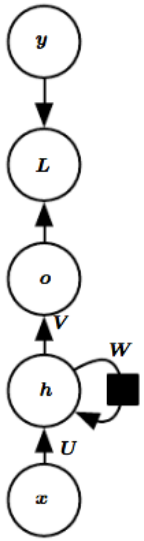
- Entrenamiento de RNNs
  - BPTT: Unfolding + backprop

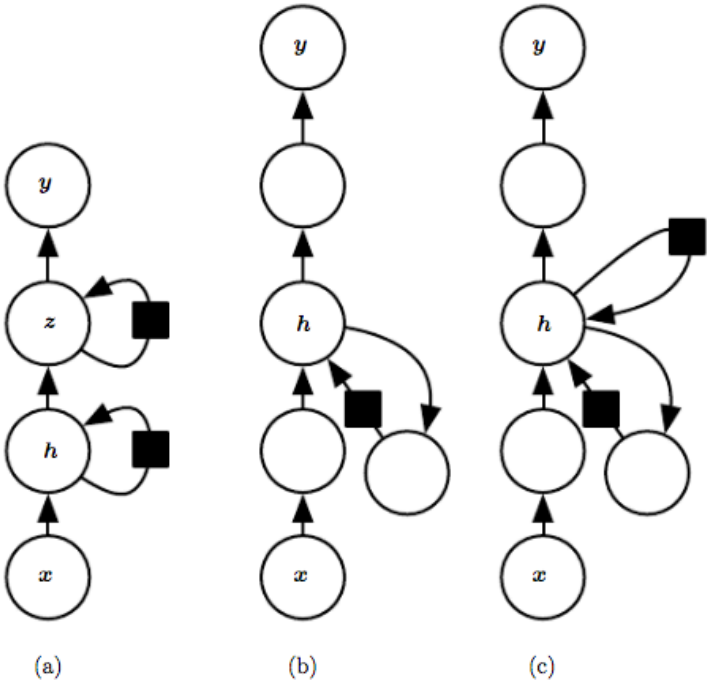




Modelando datos secuenciales

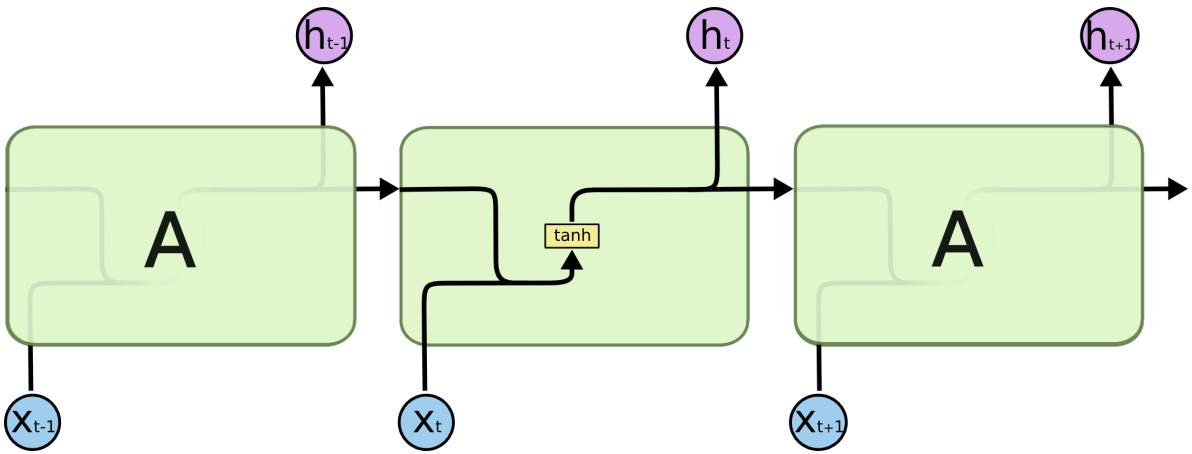
- Going deep with RNNs





Modelando datos secuenciales

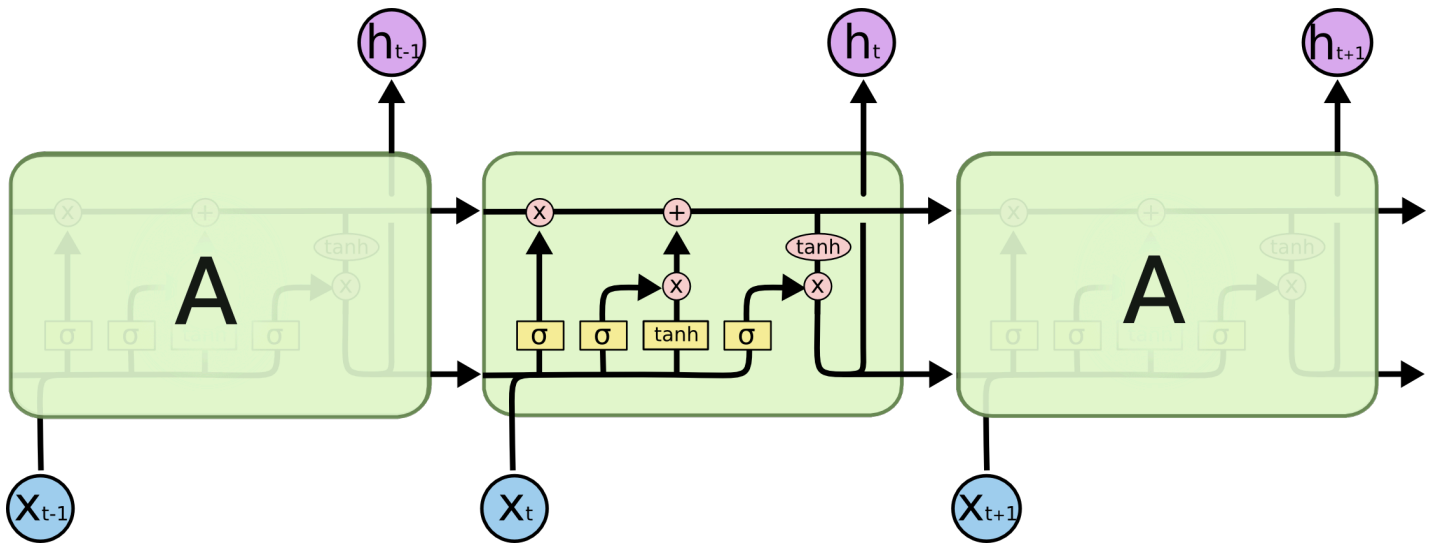
- LSTM: Long short-term memory



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

Modelando datos secuenciales

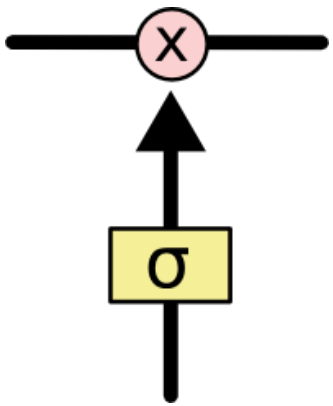
- LSTM: Long short-term memory



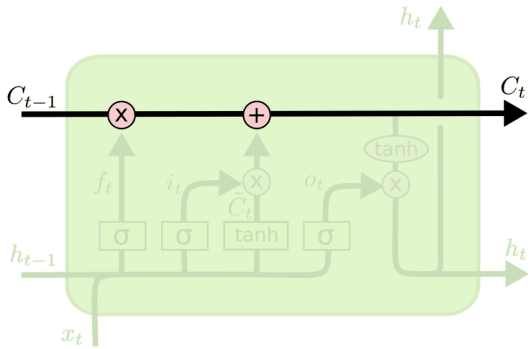
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Modelando datos secuenciales

- LSTM: Long short-term memory



Gated units

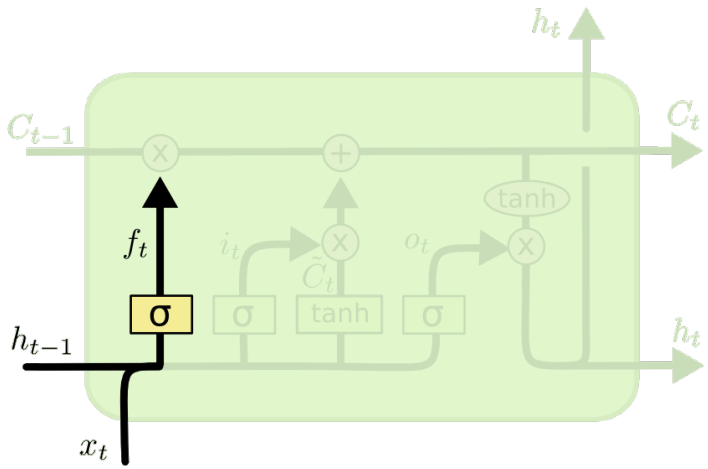


Celllstate

Modelando datos secuenciales

- LSTM: Long short-term memory

Cuanto información considerar del estado previo?

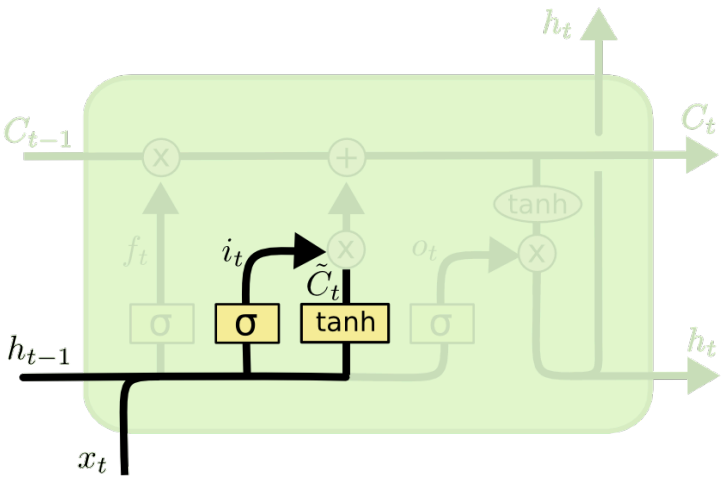


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Modelando datos secuenciales

- LSTM: Long short-term memory

Qué puede añadirse al nuevo estado, y cuánto?



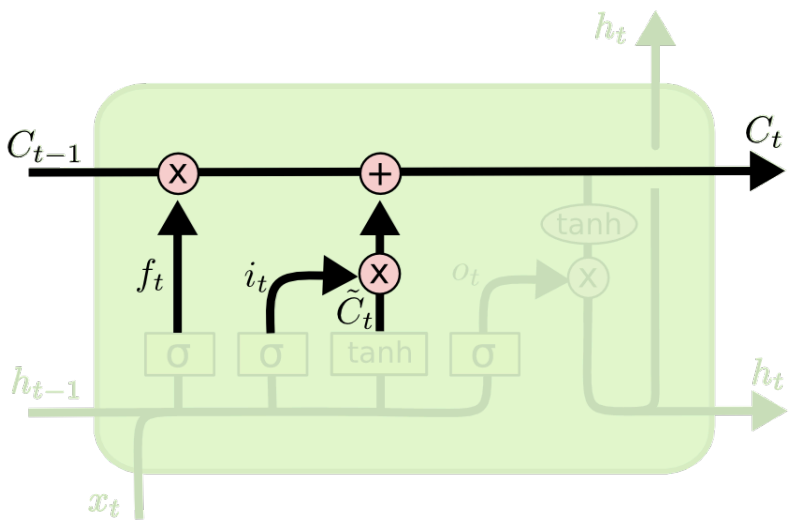
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

Modelando datos secuenciales

- LSTM: Long short-term memory

Actualizar estado

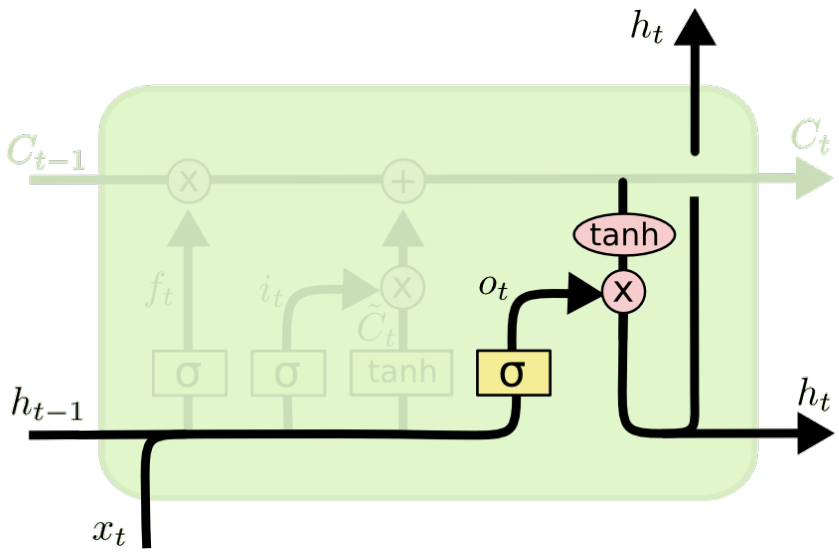


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Modelando datos secuenciales

- LSTM: Long short-term memory

Qué dar de salida?

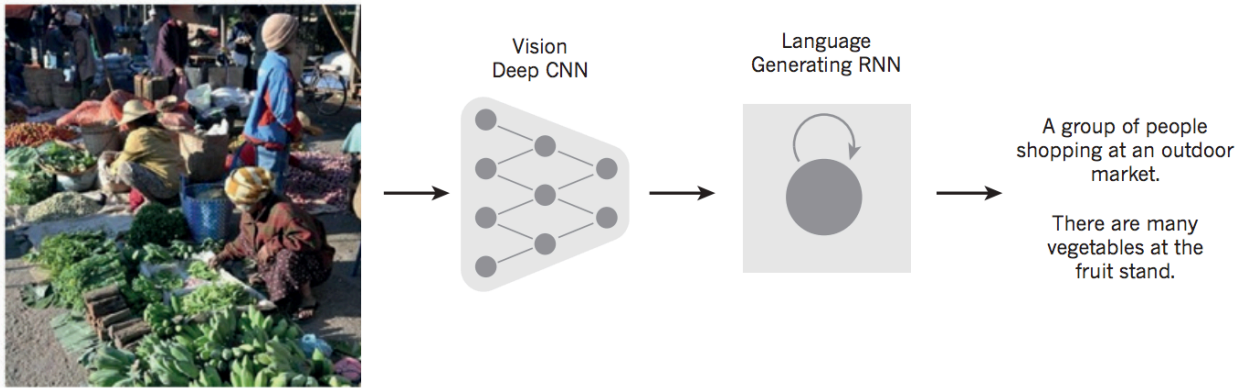


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

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

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



Variantes de aprendizaje profundo

- Principales modelos de DL
  - Redes neuronales profundas (DNNs, MLPs)
  - Redes neuronales convolucionales (CNNs)
  - Restricted Boltzman machines
  - Deep belief networks
  - Autoencoders
  - Long short term memory networks

- Otros paradigmas
  - Redes residuales
  - Inception networks
  - Gated recurrent NNs
  - Generative adversarial networks

Retos del aprendizaje profundo

Extensiones, mejoras…

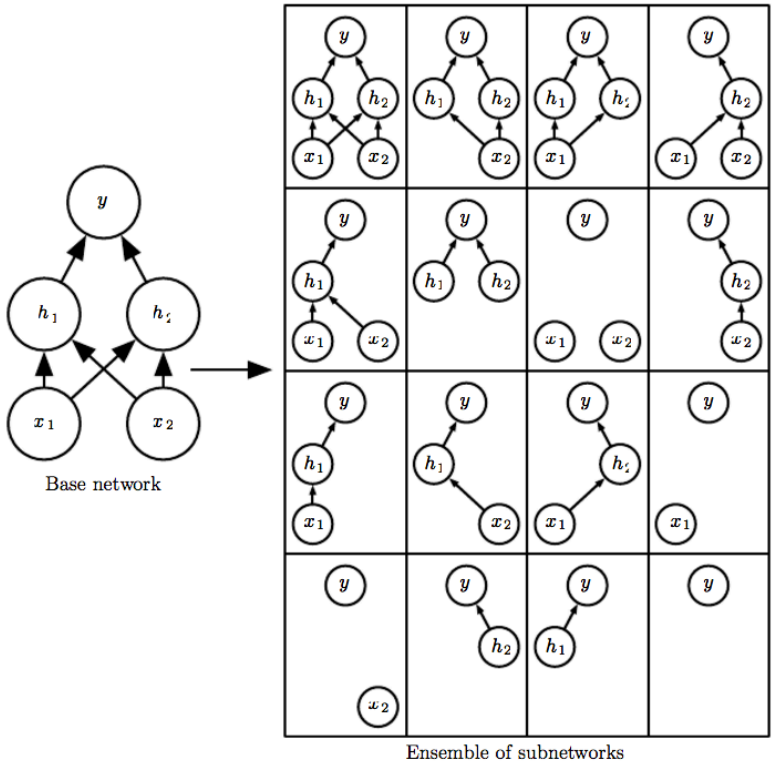
- Dropout
- Adversarial training
- Multi task learning
- Multi stream architectures
- Inception
- Residual learning
- …

Dropout

- Idea: desconectar (*drop out*) unidades internas con cierta probabilidad; usando mini-batches para actualizar los parámetros de la red bajo diferentes configuraciones

- Similar a bagging:

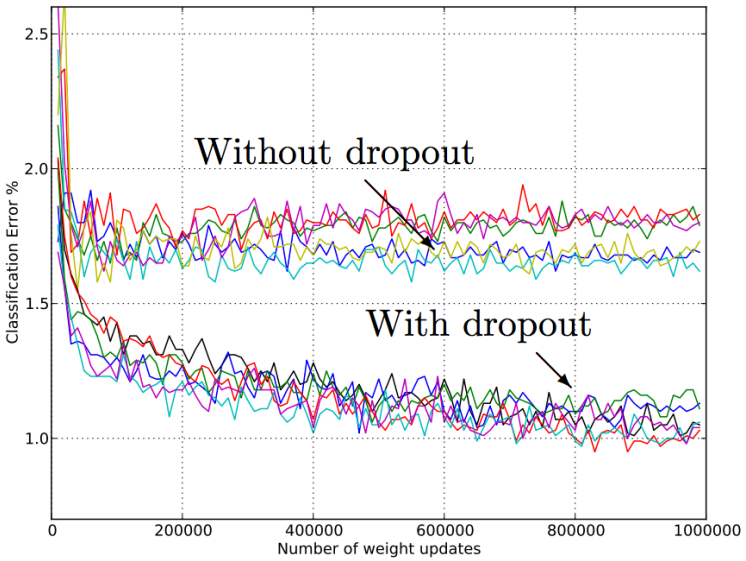
$$\frac{1}{k}\sum_{i=1}^{k} p^{(i)}(y \mid \boldsymbol{x}). \qquad \sum_{\boldsymbol{\mu}} p(\boldsymbol{\mu}) p(y \mid \boldsymbol{x}, \boldsymbol{\mu})$$



Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. **Dropout: A Simple Way to Prevent Neural Networks from Overfitting.** 15(Jun):1929–1958, 2014.

Dropout

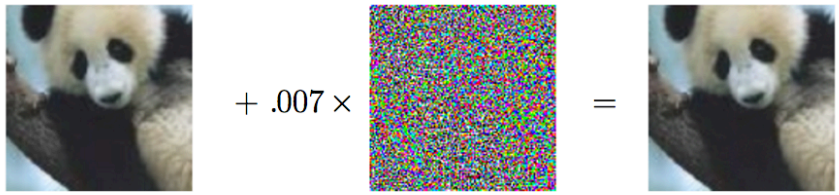
- Idea: desconectar (*drop out*) unidades internas con cierta probabilidad; usando mini-batches para actualizar los parámetros de la red bajo diferentes configuraciones



Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. **Dropout: A Simple Way to Prevent Neural Networks from Overfitting.** 15(Jun):1929–1958, 2014.

Entrenamiento con adversario

- Idea: Introducir ejemplos adversos durante la fase de entrenamiento
  - Ejemplo adverso: una instancia que después de una modificación menor (imperceptible) hace a la red cometer errores



$\boldsymbol{x}$ | $+ .007 \times$ | $\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ | $=$ | $\boldsymbol{x} + \epsilon\,\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

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

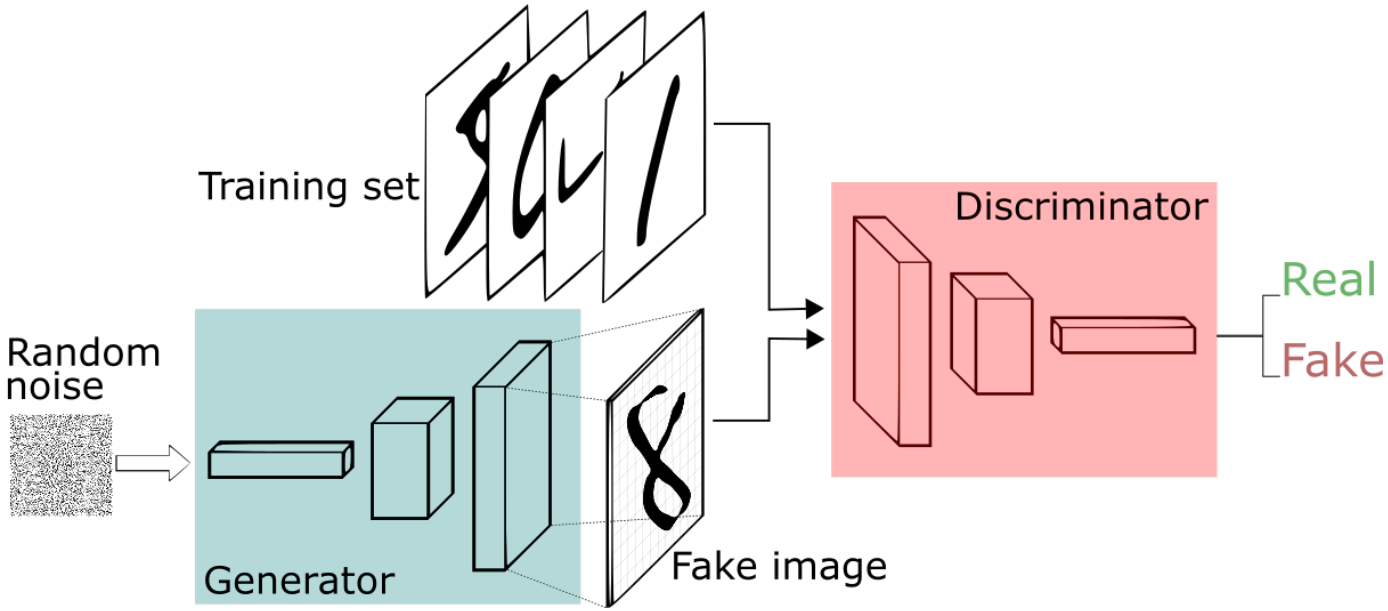
"nematode" w/ 8.2% confidence

"gibbon" w/ 99.3 % confidence

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

Entrenamiento con adversario

- GANs

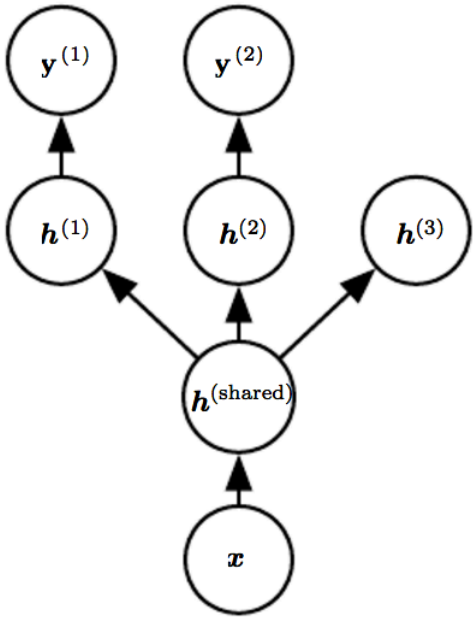


https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/

Aprendizaje multi tarea

- Idea: Aprender modelos que comparten capas genericas de parámetros y al mismo tiempo aprenden tareas especificas a través de capas especializadas

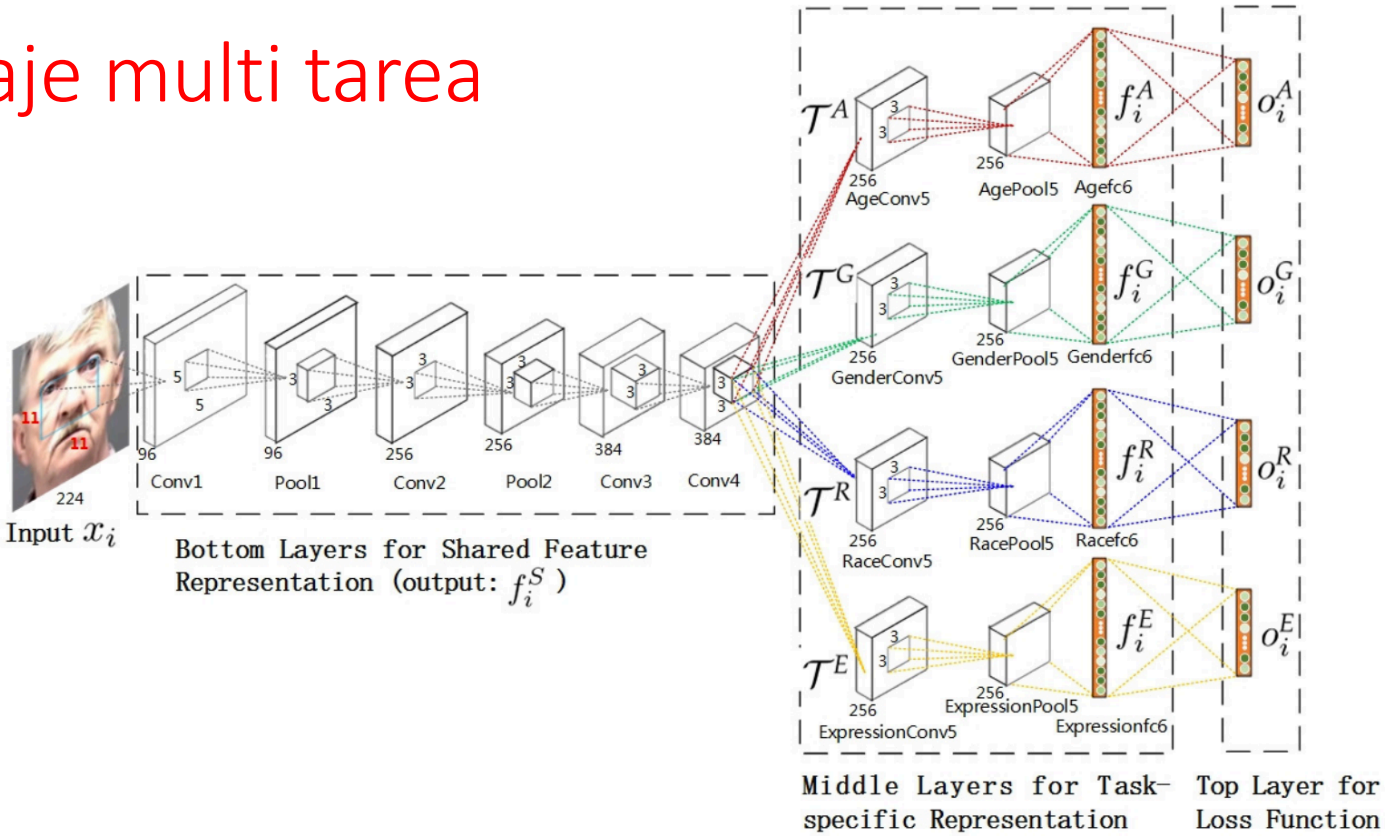
- Creencia: De entre los factores que explican variaciones en los adtos asociados a diferentes tareas, algunas se comparten a través de dos o más tareas



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

Aprendizaje multi tarea

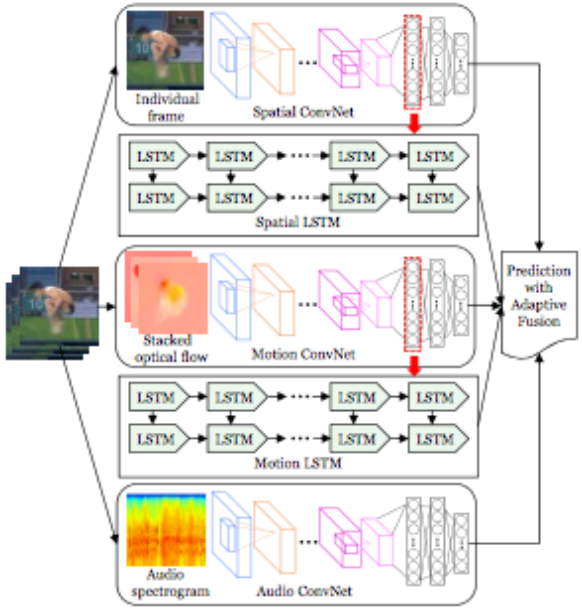
- E.g., simultaneamente predecir: genero, edad, etnicidad y expression facial a partir de una image



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

Modelos multi - stream

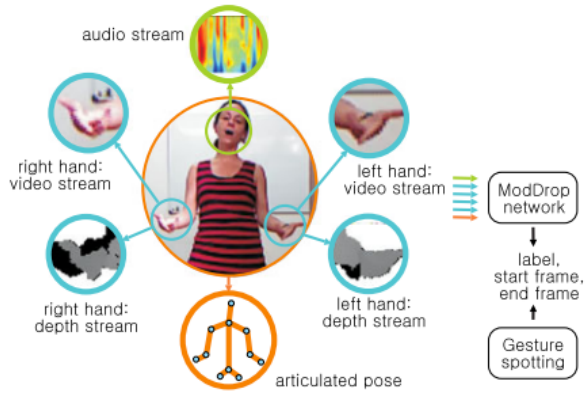
- Idea: tener diferentes rutas internas dentro del modelo asociadas a fuentes de datos diferentes que eventualmente convergen en la misma capa (misma tarea)

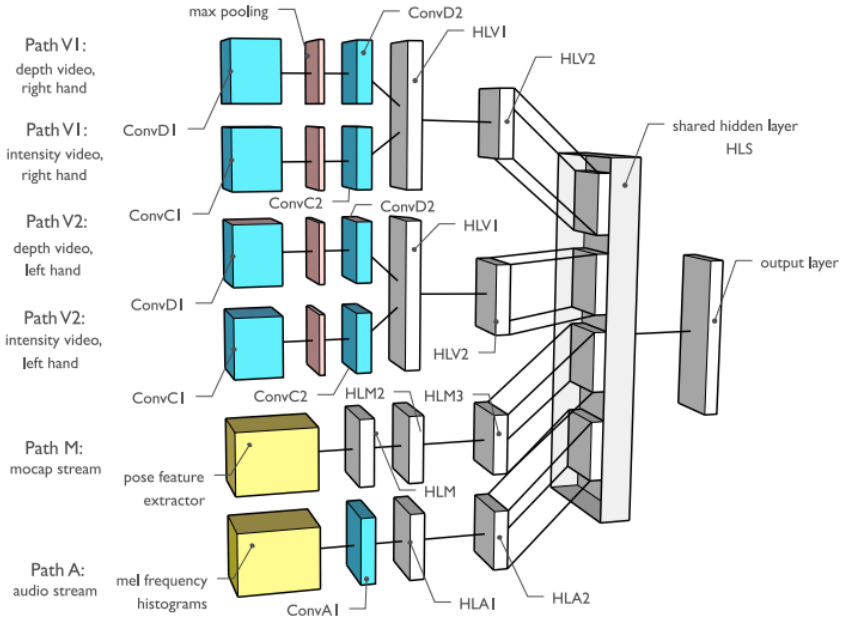


Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, Xiangyang Xue, Jun Wang. **Fusing Multi-Stream Deep Networks for Video Classification.** arXiv:1509.06086, 2015

Modelos multi - stream

- E.g., Reconocimiento multimodal de gestos

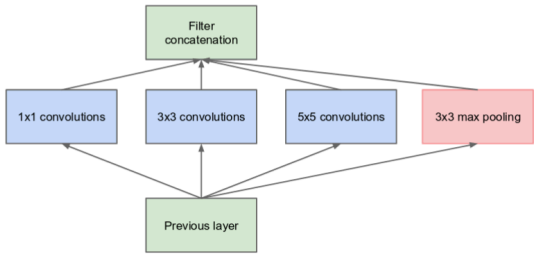




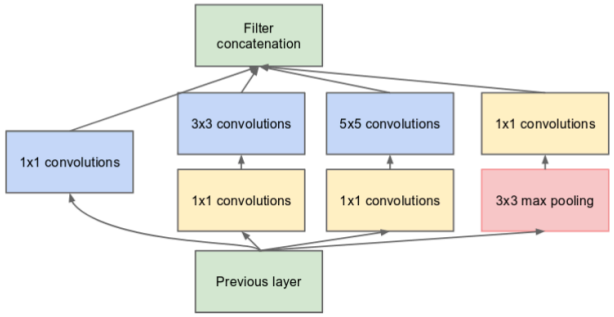
Natalia Neverova, Christian Wolf, Graham Taylor, Florian Nebout. ModDrop: adaptive multi-modal gesture recognition. TPAMI, Vol, 38(8):1692--1706, 2016

Inception

- Idea: redes dentro de redes!



(a) Inception module, naïve version

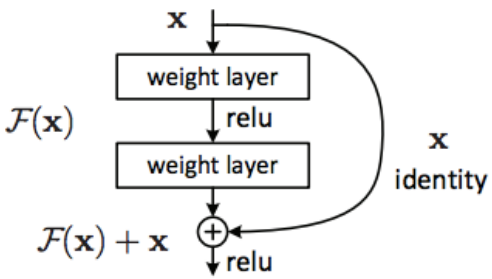


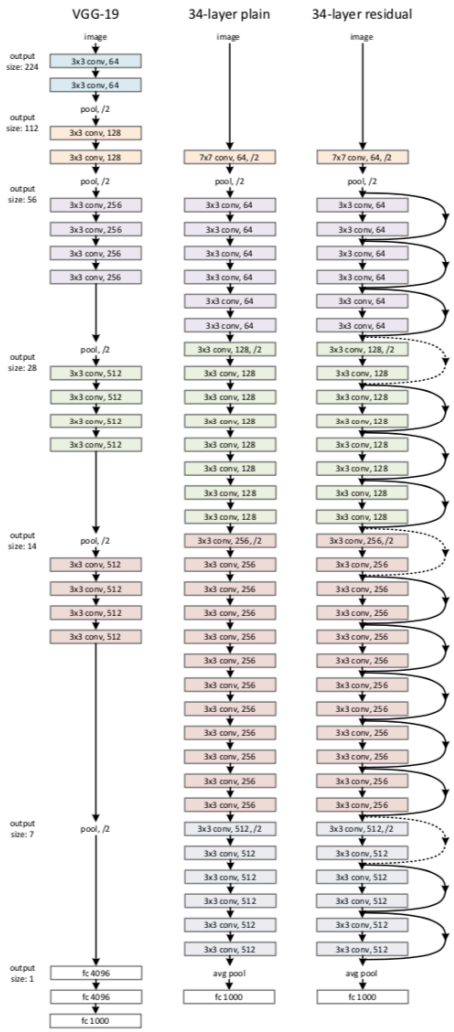
(b) Inception module with dimension reductions

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

Aprendizaje residual

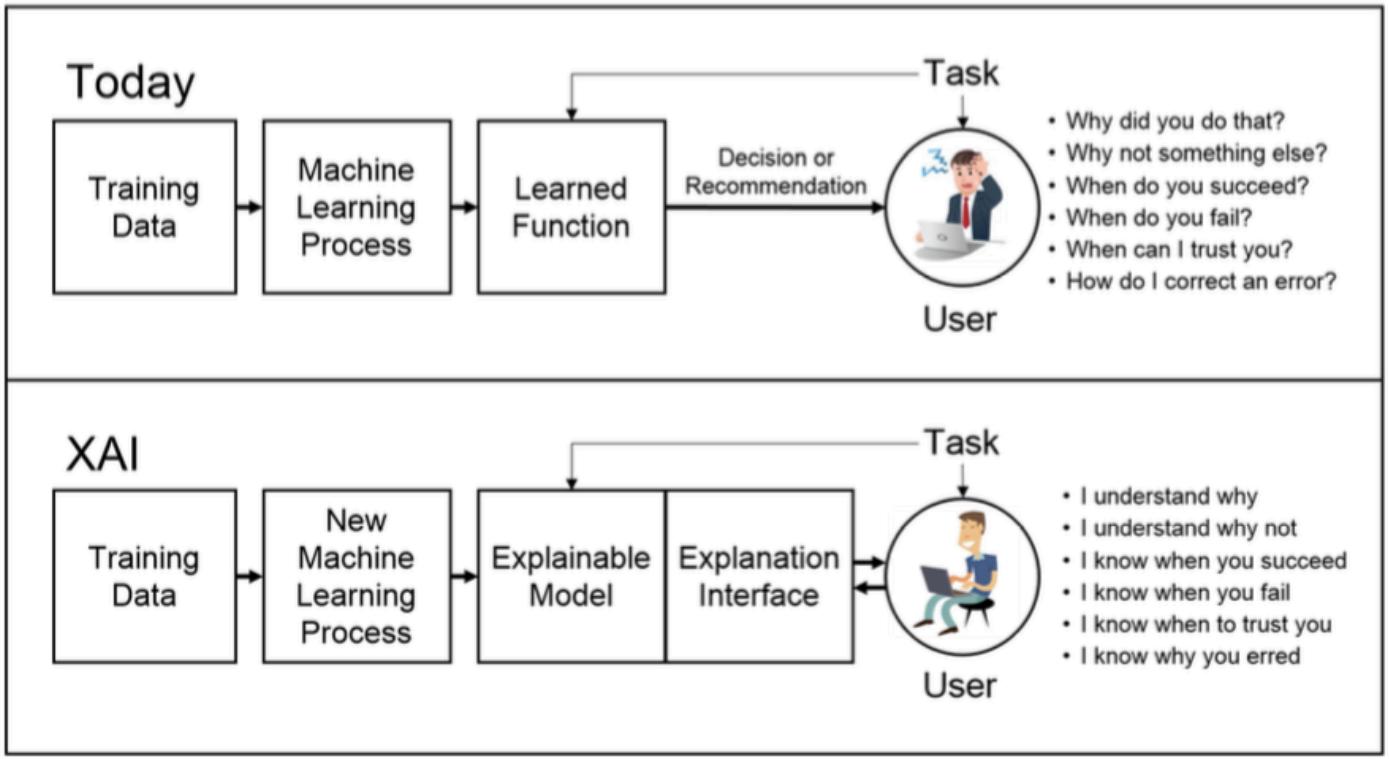
- Idea: Introducir capas que pueden usarse o no





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

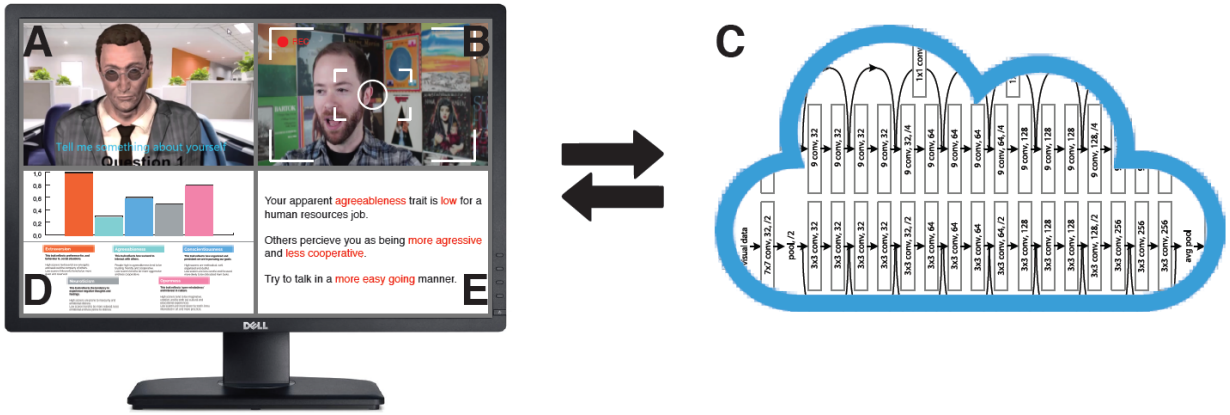
Explicabilidad



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

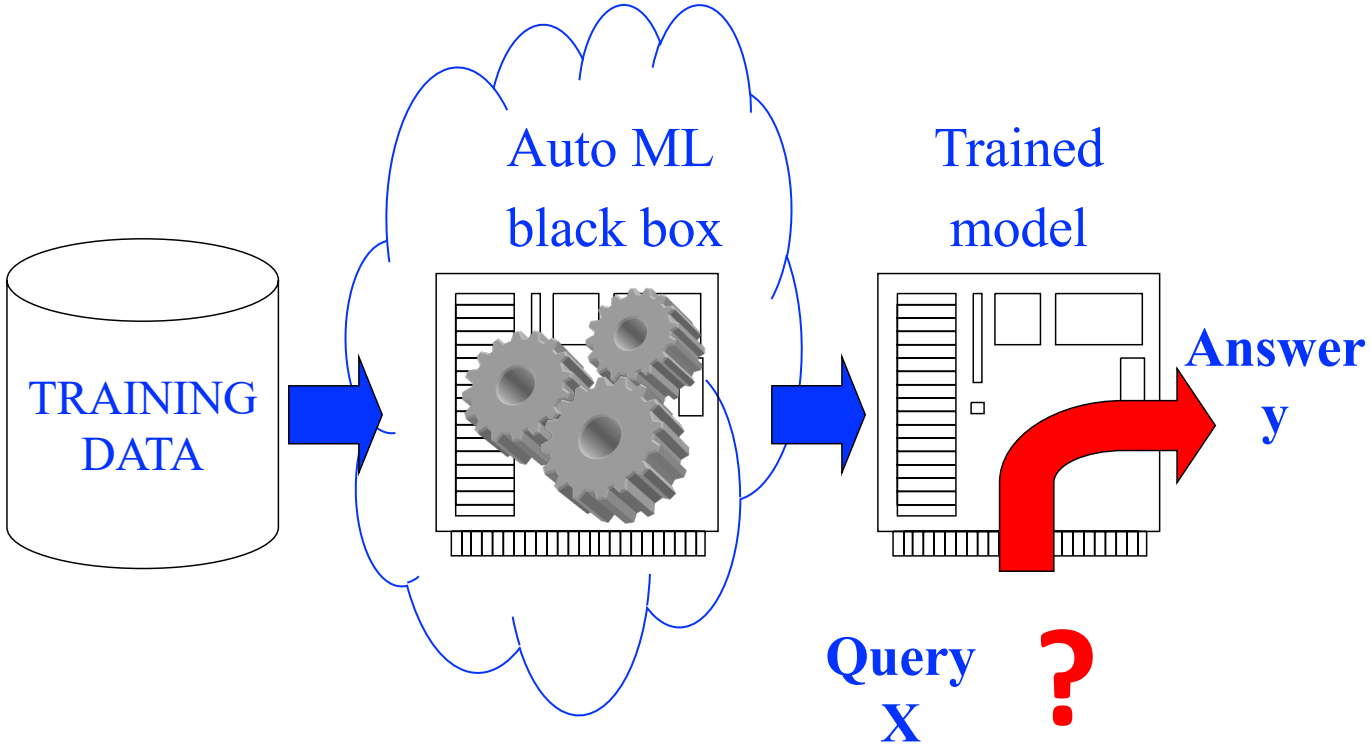
Explicabilidad

- Desarrollar modelos interpretables y cuyas salidas puedan ser explicadas



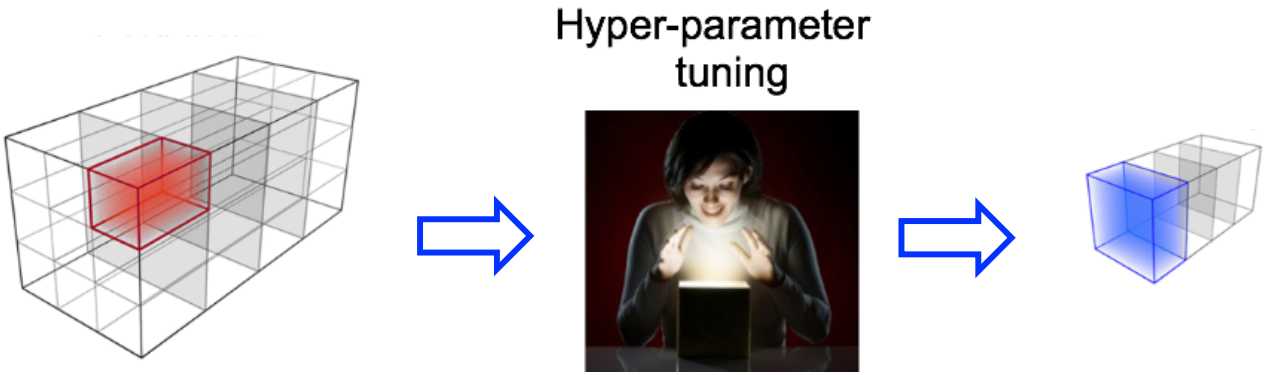
Hugo Jair Escalante et al. **Explaining First Impressions: Modeling, Recognizing, and Explaining Apparent Personality from Videos.** CoRR abs/1802.00745 (2018)

Automatic Machine Learning - AutoML



Automatic Machine Learning - AutoML

Automatic Deep Learning

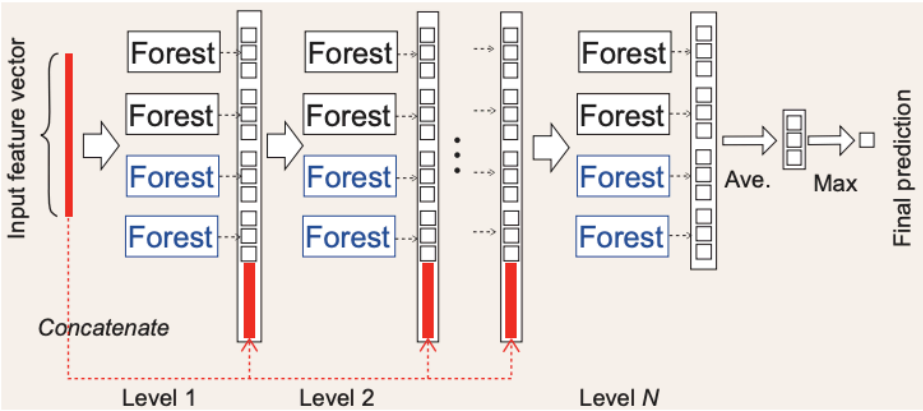


https://autodl.chalearn.org/

Aprendizaje profundo sin redes neuronales?

- Son las redes neuronales el único medio para el aprendizaje profundo?

- Esfuerzos recientes:
  - *Deep forest*
  - *Deep genetic programming*
  - *…?*

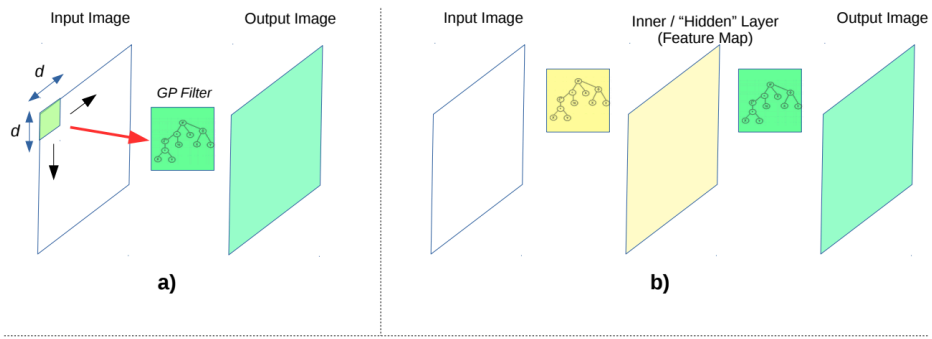


Zhi-Hua Zhou, Ji Feng. **Deep Forest.** National Science Review 00: 1–13, 2018

Aprendizaje profundo sin redes neuronales?

- Son las redes neuronales el único medio para el aprendizaje profundo?

- Esfuerzos recientes:
  - *Deep forest*
  - *Deep genetic programming*
  - *…?*



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

Otros temas…

- Aprendizaje profundo por transferencia
- Búsqueda de arquitecturas neuronales (NAS, AutoML)
- Compresión de parámetros
- Teoría del aprendizaje profundo
- Aprendizaje con datos escasos
- …

Comentarios finales

- Beneficios
  - Extremadamente útiles para aprender representaciones y modelos a partif 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 principals sub áreas de IA: CV, PR, NLP, SP, and in shortly will be ruling over other domains
- Es complicado seguir el progreso de aprendizaje profundo

Lecturas sugeridas

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

Tarea 2

- Entrenar una red convolucional para la clasificación de imágenes
  - Conseguir base de datos MNIST + clasificación de imágenes
  - Implementar una CNN “estándar”
  - Variar número de capas convolucionales, gráfica desempeño vs. Complejidad
  - Realizar un análisis de performance vs parámetros (drop out rate, activation units, strides, epochs, MB size, optimizer, etc.)
  - Reportar tabla/gráfica con resultados
  - Enviar: reporte + código, deadline: Junio 13, 11:59 pm

Tarea 2

- Seleccionar y presentar un artículo de: ICML2019, CVPR2019, NeurIPS2018, ICLR2019, ACL2019, NAACL2019,

- Seleccionar artículo con base a intereses personales (de preferencia asociado a proyecto) + debe ser sobre/usando DL

- Reporte de una cuartilla + Presentación de 10 minutos