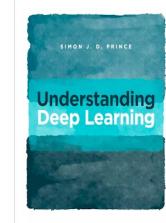


Bibliografía

- Understanding Deep Learning. Capítulos 14,15.



Tema 7 – Aspectos Avanzados

Aprendizaje Automático II - Grado en Inteligencia Artificial
Universidad Rey Juan Carlos

Iván Ramírez Díaz
ivan.ramirez@urjc.es

José Miguel Buenaposada Biencinto
josemiguel.buenaposada@urjc.es

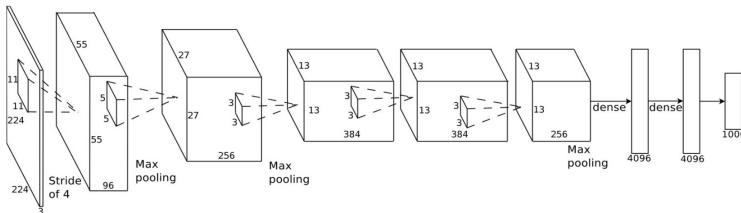
7.1 Aprendizaje de Representaciones

- ¿Qué es una representación? Representaciones jerárquicas
- Estrategias de aprendizaje
 - Aprendizaje por Transferencia (Transfer Learning)
 - Aprendizaje Multi-tarea (Multi-task learning)
 - Aprendizaje Auto-supervisado (self-supervised learning)
 - Aprendizaje por Adaptación de Dominio

7.1 Aprendizaje de Representaciones

- ¿Qué es una representación? Representaciones jerárquicas
- Estrategias de aprendizaje
 - Aprendizaje por Transferencia (Transfer Learning)
 - Aprendizaje Multi-tarea (Multi-task learning)
 - Aprendizaje Auto-supervisado (self-supervised learning)
 - Aprendizaje por Adaptación de Dominio

AlexNet (2012)

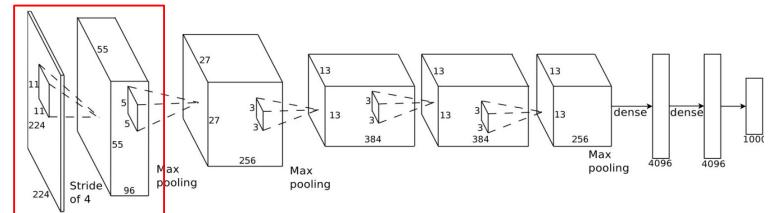


Ideas principales:

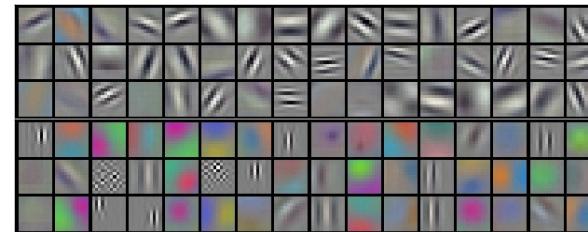
- 650K neuronas, 60M parámetros, 630M conexiones.
- Entrenamiento más rápido por la ReLU (¡1er modelo profundo!)
- **Data augmentation** en train y test (5 imágenes generadas, resultados promediados)
- SGD, lr₀=0.01, reducido manualmente, momentum=0.9, batch sz=128
- Regularización: **L2 weight decay**, **Dropout** (p=0.5) en las dos últimas capas FC
- Escrito en Python/C++/CUDA
- **Ensemble** de 7 CNNs: 16.4% top-5 error rate, 38.1% top-1 error rate (en validación)

A. Krizhevsky, I. Sutskever, G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems* 25, pp. 1097–1105, 2012.

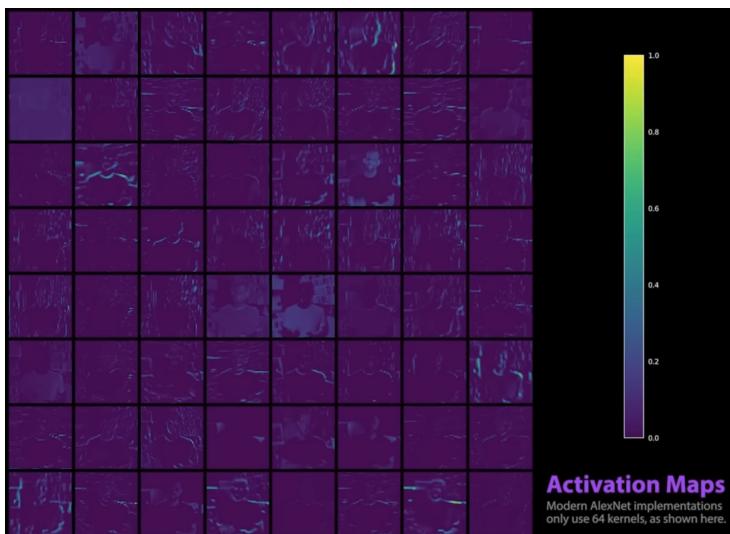
¿Qué aprende AlexNet en ImageNet?



En la primera capa convolucional 96 kernels (11x11 en RGB):



AlexNet activaciones en la primera capa

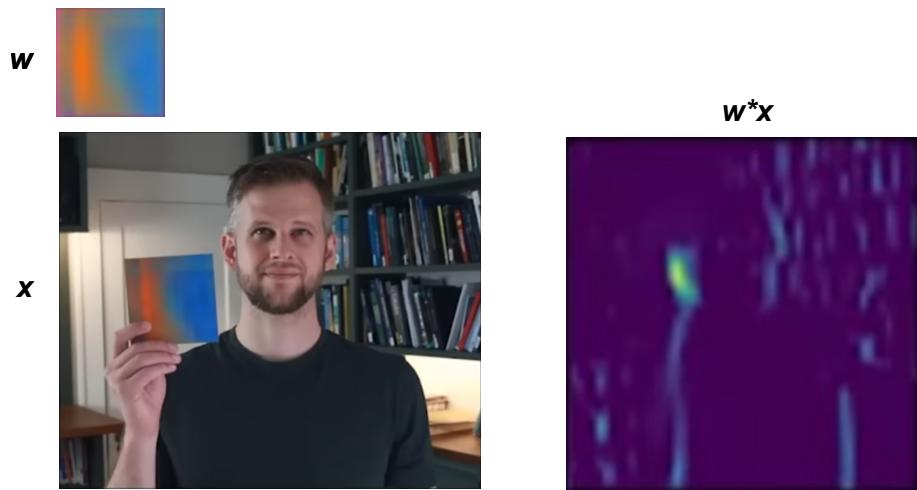


<https://youtu.be/UZDiGooFs54?feature=shared>

La correlación cruzada compara patrones

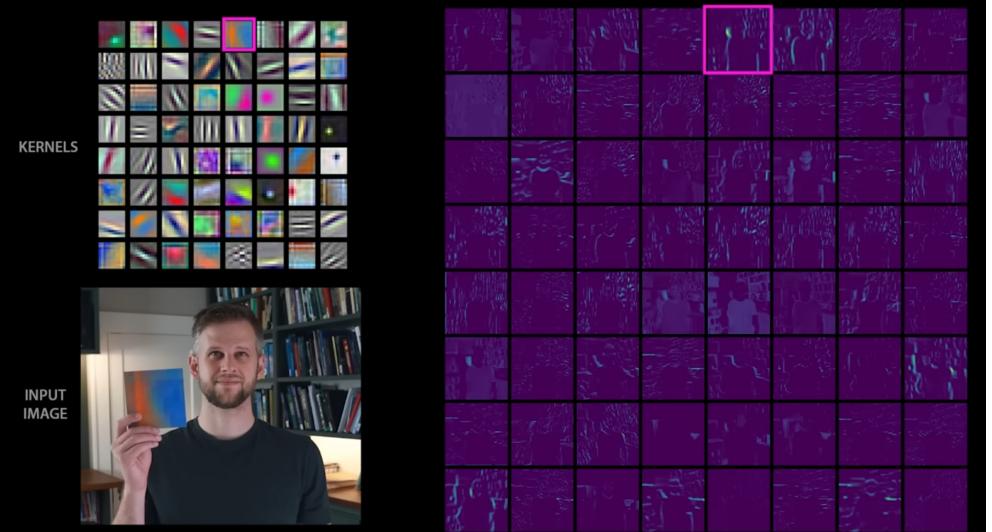
- En cada paso de la correlación cruzada se calcula el producto escalar de w con una parte de la entrada x , $p=(x_i, \dots x_{i+K-1})^T$.
- Este producto escalar es:
$$w \cdot p = \|w\| \|p\| \cos \alpha$$
 donde α es el ángulo entre w y p
- Suponiendo $\|w\| = \|p\|$ y de valor fijo,
 - La máxima respuesta del filtro w (el valor de $w \cdot p$) se dará cuando se encuentre en la misma dirección que p ($\alpha = 0$) ¡cuando w y p sean aproximadamente iguales!

La correlación cruzada compara patrones



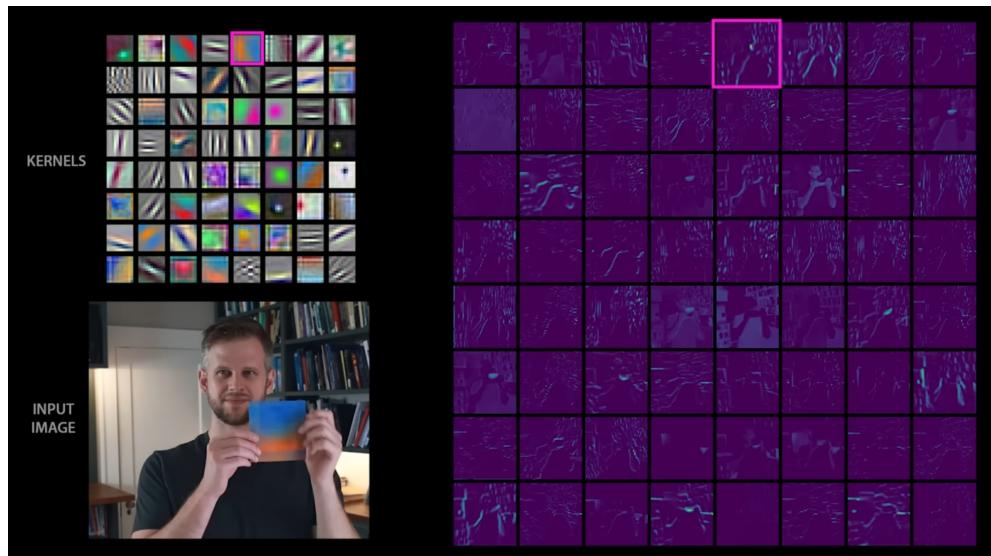
<https://youtu.be/UZDiGooFs54?feature=shared>

La correlación cruzada compara patrones



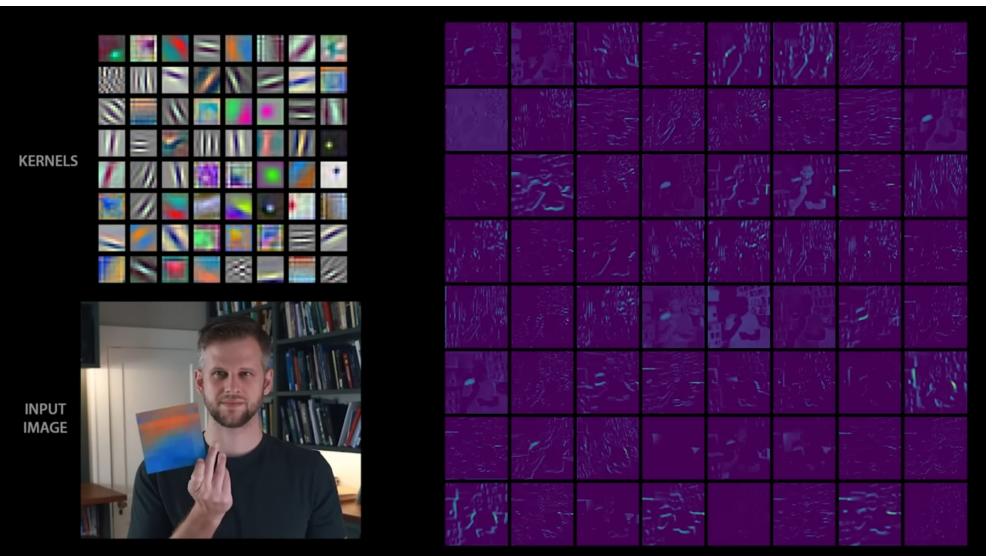
<https://youtu.be/UZDiGooFs54?feature=shared>

La correlación cruzada compara patrones



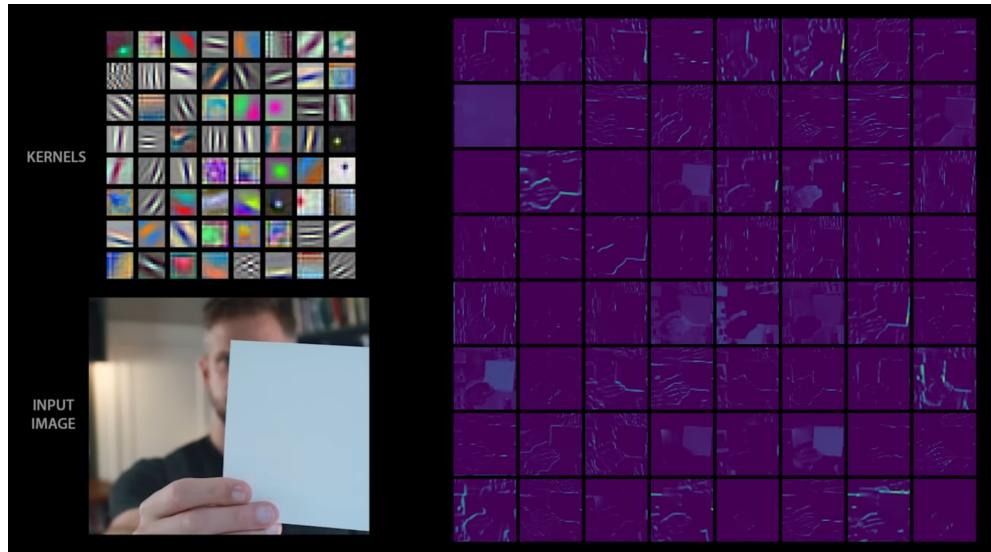
<https://youtu.be/UZDiGooFs54?feature=shared>

La correlación cruzada compara patrones



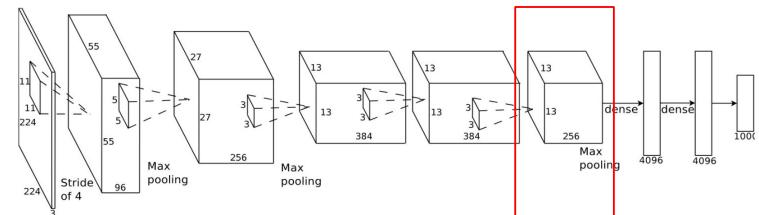
<https://youtu.be/UZDiGooFs54?feature=shared>

La correlación cruzada compara patrones



<https://youtu.be/UZDiGooFs54?feature=shared>

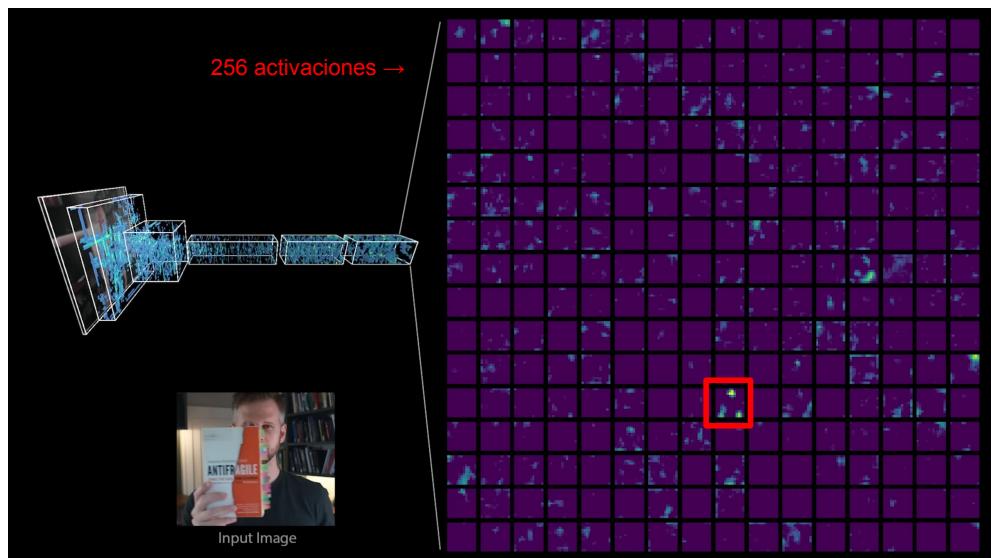
¿Qué aprende AlexNet en ImageNet?



En las **capas más profundas** es difícil de visualizar los kernels, sin embargo:

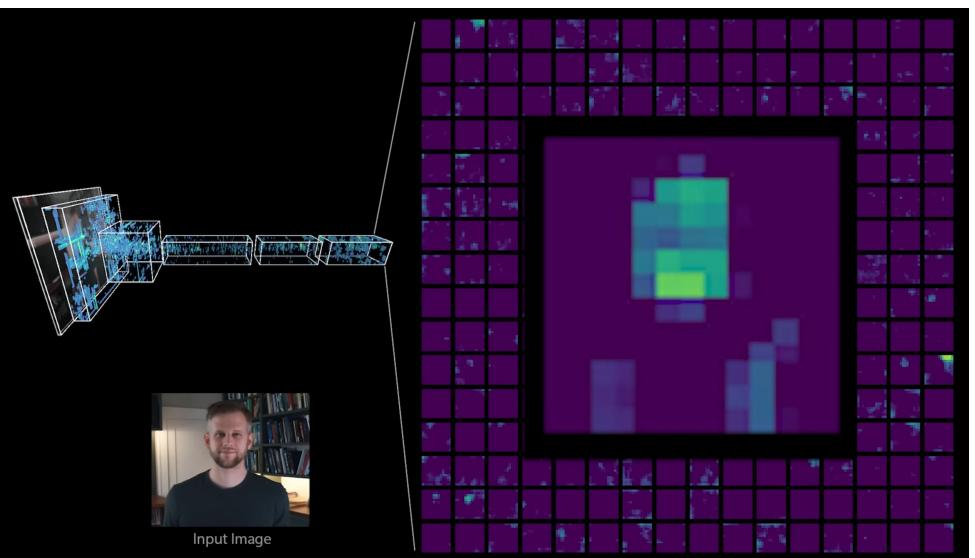
- En las activaciones “emergen” detectores de objetos completos o partes de ellos.
- No se han proporcionado esos objetos etiquetados.

Activaciones en capas profundas ...



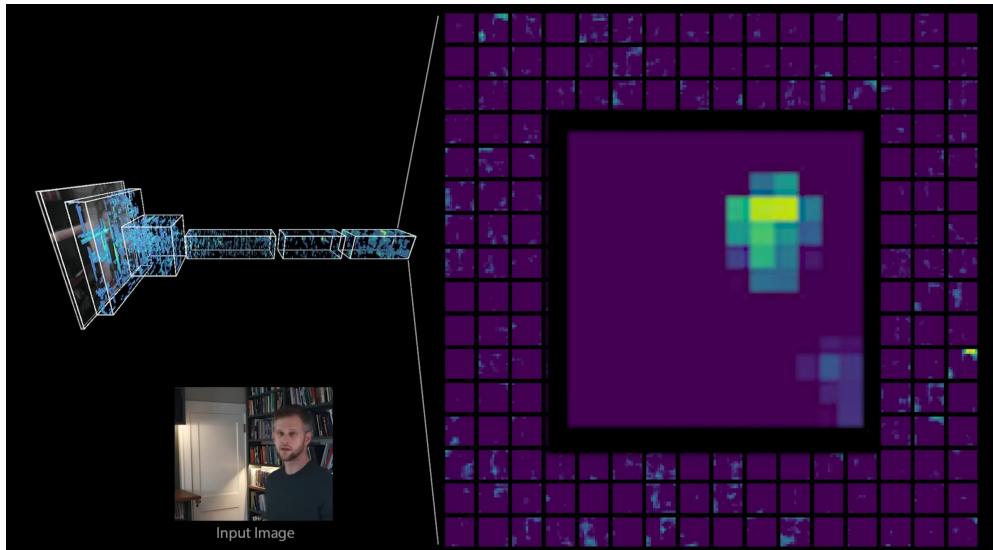
<https://youtu.be/UZDiGooFs54?feature=shared>

Activaciones en capas profundas ...



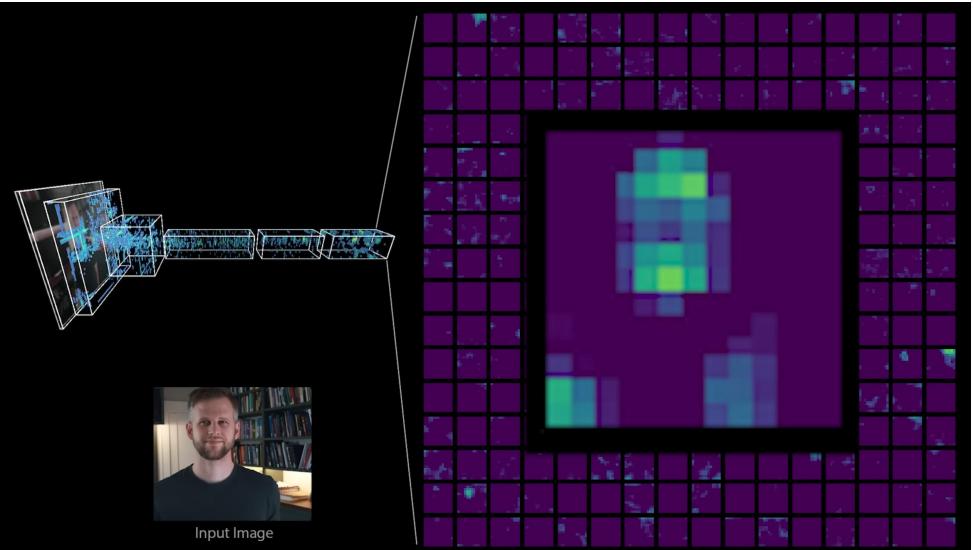
<https://youtu.be/UZDiGooFs54?feature=shared>

Activaciones en capas profundas ...



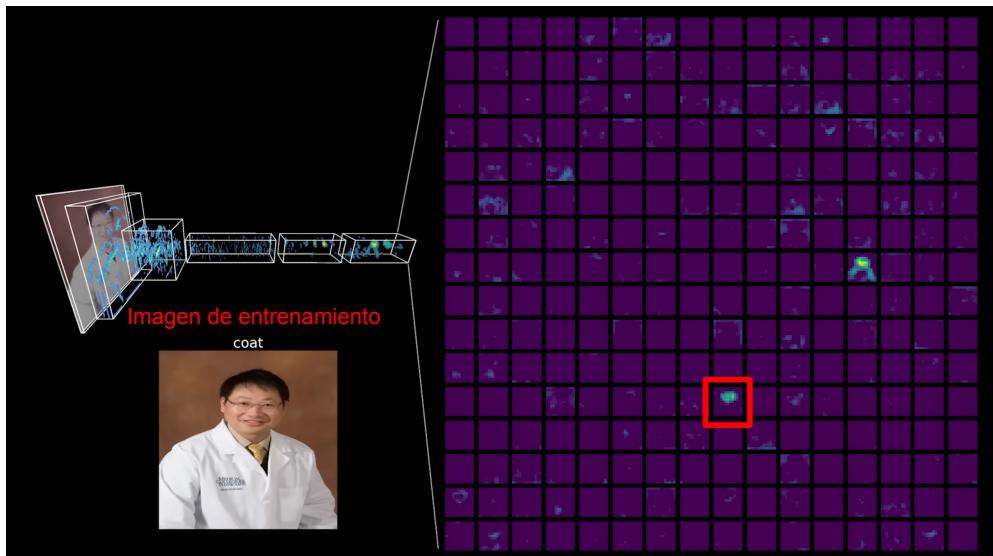
<https://youtu.be/UZDiGooFs54?feature=shared>

Activaciones en capas profundas ...



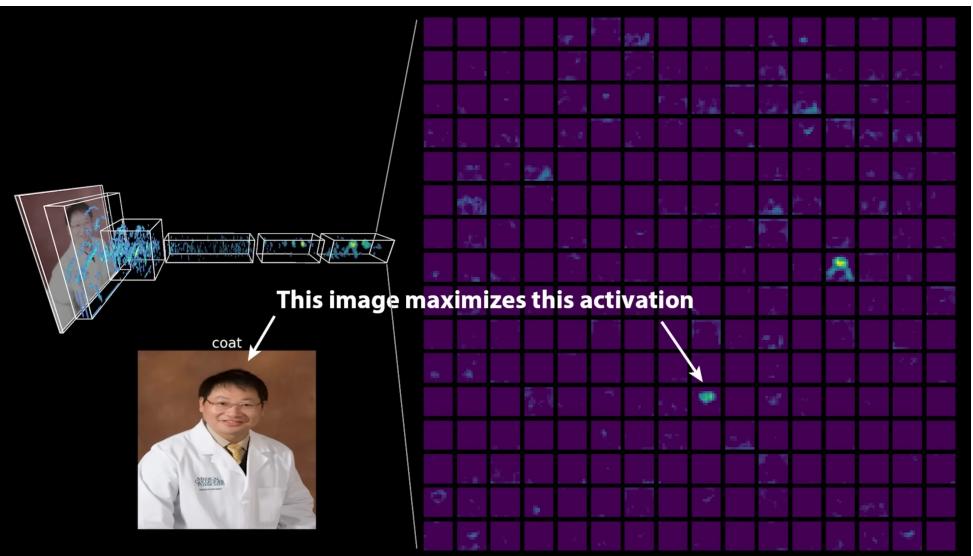
<https://youtu.be/UZDiGooFs54?feature=shared>

Activaciones en capas profundas ...



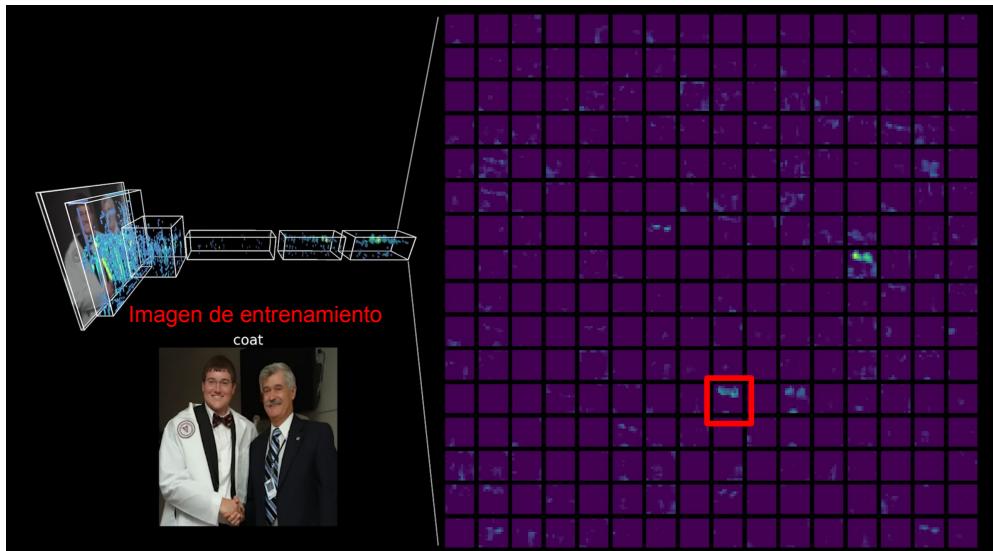
<https://youtu.be/UZDiGooFs54?feature=shared>

Activaciones en capas profundas ...

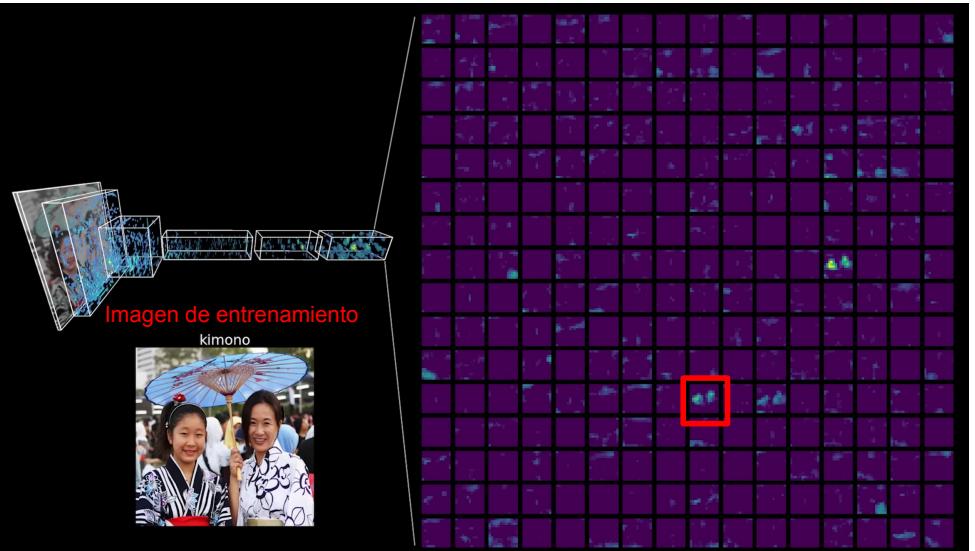


<https://youtu.be/UZDiGooFs54?feature=shared>

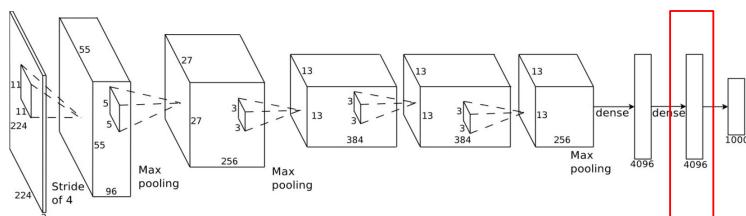
Activaciones en capas profundas ...



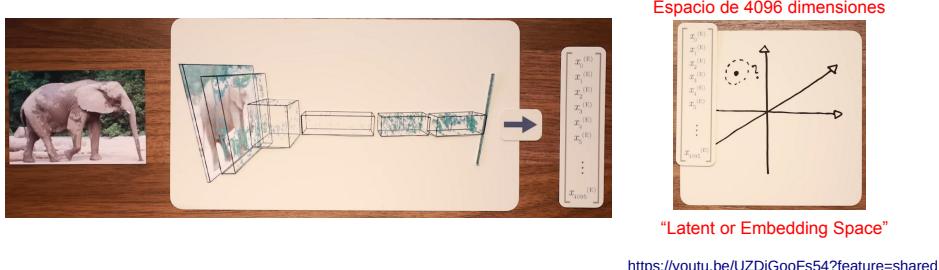
Activaciones en capas profundas ...



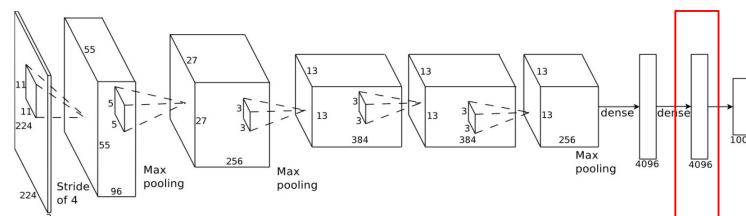
Representación aprendida en AlexNet



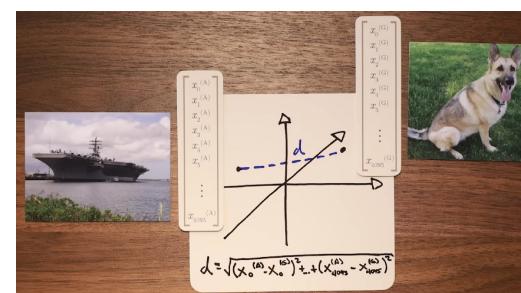
Última capa fully connected → descriptor (una representación) de cada imagen



Representación aprendida en AlexNet



Última capa fully connected → descriptor (una representación) de cada imagen



Representación aprendida en AlexNet

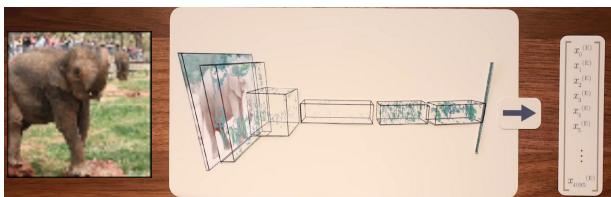
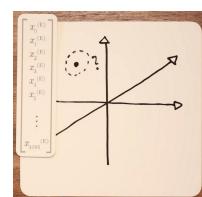


Imagen test ImageNet

<https://youtu.be/UZDiGooFs54?feature=shared>



6 imágenes de train más cercanas a la anterior en el espacio de 4096 dimensiones



A. Krizhevsky, I. Sutskever, G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems* 25, pp. 1097–1105, 2012.

Representación aprendida en AlexNet

Imagen test ImageNet 6 imágenes de train más cercanas a la anterior en el espacio de 4096 dimensiones



A. Krizhevsky, I. Sutskever, G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems* 25, pp. 1097–1105, 2012.

Representación aprendida en AlexNet

Imagen test ImageNet

6 imágenes de train más cercanas a la anterior en el espacio de 4096 dimensiones



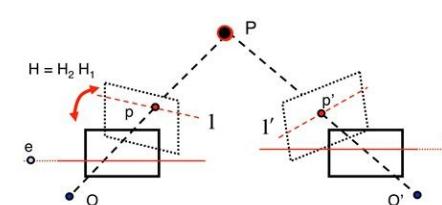
La distancia L2 la representación (AlexNet) de dos imágenes funciona aunque el contenido de la imagen sea muy diferente (poses y escalas diferentes) → ¡Comparar las imágenes directamente con L2 no funcionaría!



A. Krizhevsky, I. Sutskever, G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems* 25, pp. 1097–1105, 2012.

Representaciones jerárquicas profundas

- ¿Qué es una representación?
 - Un sistema formal que
 - Hace explícitas ciertas entidades y tipos de información
 - Puede ser manejada por un algoritmo para lograr alguna meta de procesamiento de información
- Las representaciones difieren en términos de la información que hacen explícita y en términos de qué algoritmos permiten



Representación de la geometría entre dos cámaras en Visión por Computador (geometría epipolar)

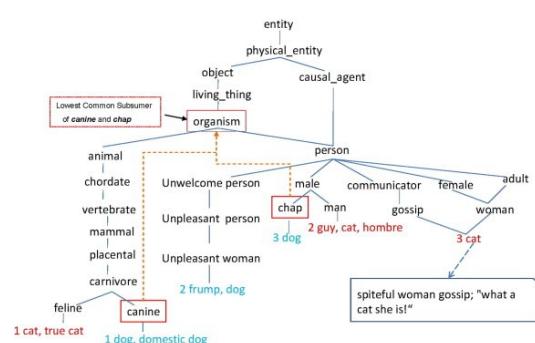
Representaciones jerárquicas profundas

- ¿Qué es una representación?
 - Un sistema formal que
 - Hace explícitas ciertas entidades y tipos de información
 - Puede ser manejada por un algoritmo para lograr alguna meta de procesamiento de información
- Las representaciones difieren en términos de la información que hacen explícita y en términos de qué algoritmos permiten

Una red de neuronas profunda es una representación de un problema que emerge del proceso de entrenamiento

Representaciones jerárquicas profundas

- Representaciones jerárquicas:
 - El Deep Learning está relacionado con aprender **representaciones jerárquicas**, organizadas en múltiples niveles
 - Estas representaciones jerárquicas están relacionadas con una jerarquía de abstracciones, como la que usamos en el lenguaje



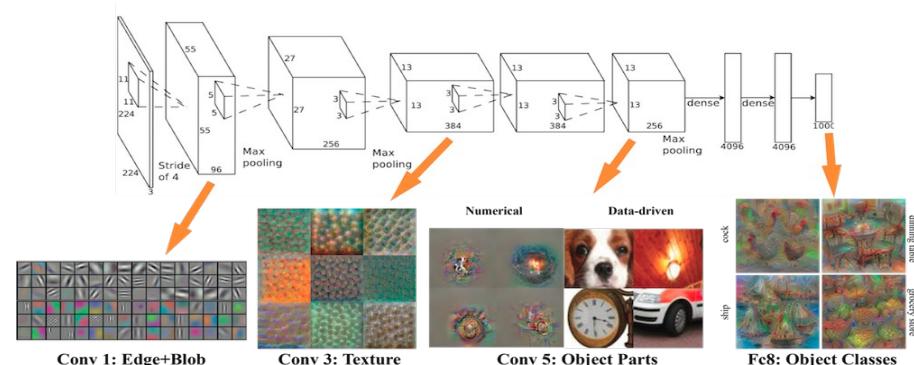
<https://wordnet.princeton.edu/>

Representaciones jerárquicas profundas

- ¿Por qué las representaciones profundas funcionan bien?
 - Emergen al construir modelos del mundo que nos rodea
 - Que son suficientemente complejos para representar situaciones del mundo real
 - Con gran (aunque siempre limitada) cantidad de datos.
 - Los modelos profundos ofrecen 2 ingredientes clave
 - Representaciones jerárquicas
 - Mecanismos potentes de regularización

Representaciones jerárquicas profundas

- Ventaja de una representación jerárquica



Una red neuronal profunda puede descubrir características que generalizan mejor a muestras no vistas, por lo que requieren menos datos de entrenamiento.

Recordatorio: Eficiencia de profundidad

Mayor eficiencia con el aumento de las capas (profundidad)

- Hay algunas funciones que requieren una red no profunda con un **número exponencialmente mayor** de unidades ocultas que una red profunda para lograr una aproximación equivalente.
- Esto se conoce como **eficiencia de profundidad** de las redes profundas.

Representaciones jerárquicas profundas

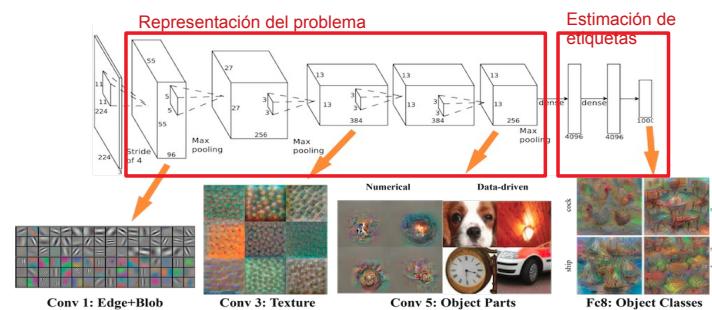
- Las representaciones aprendidas por una red profunda
 - Están guiadas por la tarea de aprendizaje utilizada:
 - **Supervisada:** dependerá más del clasificador/funciónde pérdida
 - **No supervisada:** será más general y se podrá usar en otras tareas.

7.1 Aprendizaje de Representaciones

- ¿Qué es una representación? Representaciones jerárquicas
- **Estrategias de aprendizaje**
 - Aprendizaje por Transferencia (Transfer Learning)
 - Aprendizaje Multi-tarea (Multi-task learning)
 - Aprendizaje Auto-supervisado (self-supervised learning)
 - Aprendizaje por Adaptación de Dominio

Estrategias de aprendizaje

- Eficiencia de datos. Atacando la necesidad de más datos.



Un modelo profundo ofrece:

- Buen Rendimiento
- Representación jerárquica

La representación del problema:

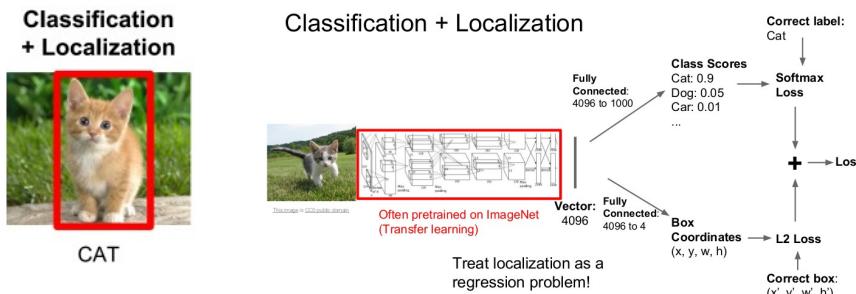
- Es la clave para el rendimiento
- Se puede reutilizar en otras tareas

Estrategias de aprendizaje

- ¿Qué es una tarea (*task*) de aprendizaje automático?

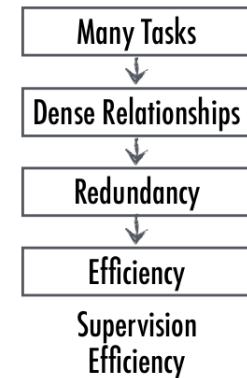
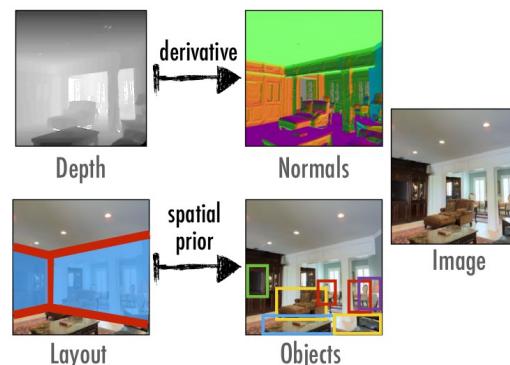
Un problema a resolver mediante un modelo computacional, con :

- Un conjunto de ejemplos asociados; y
- Una función representando la solución deseada



Estrategias de aprendizaje

- ¿Están relacionadas ciertas tareas?



Zamir, Taskonomy: Disentangling Task Transfer Learning CVPR 2018

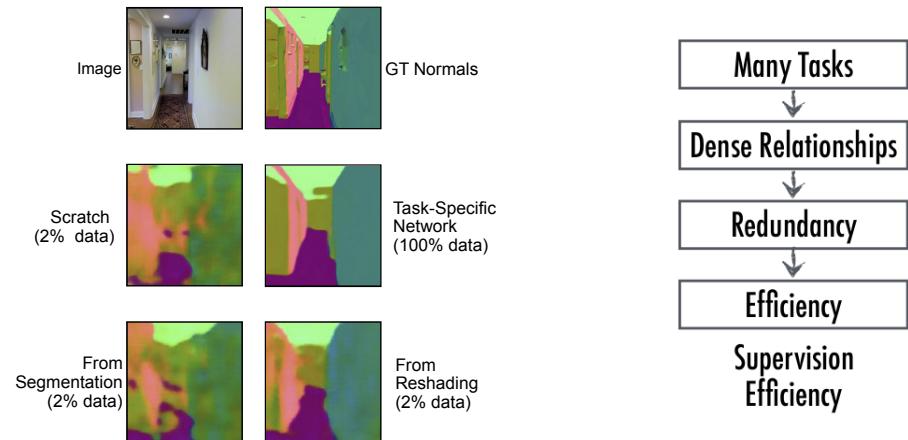
7.1 Aprendizaje de Representaciones

- ¿Qué es una representación? Representaciones jerárquicas
- Estrategias de aprendizaje
 - Aprendizaje por Transferencia (*Transfer Learning*)
 - Aprendizaje Multi-tarea (*Multi-task learning*)
 - Aprendizaje Auto-supervisado (*self-supervised learning*)
 - Aprendizaje por Adaptación de Dominio

Estrategias de aprendizaje por transferencia

- Aprendizaje por Transferencia (*Transfer learning*):
 - El modelo tiene que resolver 2 o más tareas diferentes, pero muchos factores que explican las variaciones en T1 también lo hacen en T2.
 - En aprendizaje supervisado en el que la entrada es parecida pero la salida es de diferente naturaleza.
 - Ejemplo: T1: gatos/perros , T2: hormigas/avispas se puede entrenar una representación con los muchos datos de T1 y reutilizarla en la T2 con menos datos.

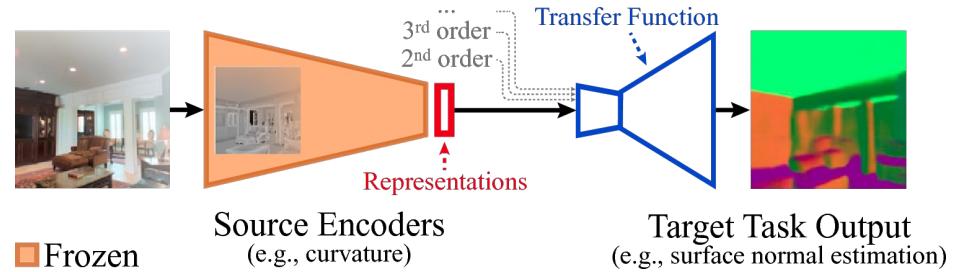
Estrategias de aprendizaje: Transfer Learning



Zamir, Taskonomy: Disentangling Task Transfer Learning CVPR 2018

Estrategias de aprendizaje por transferencia

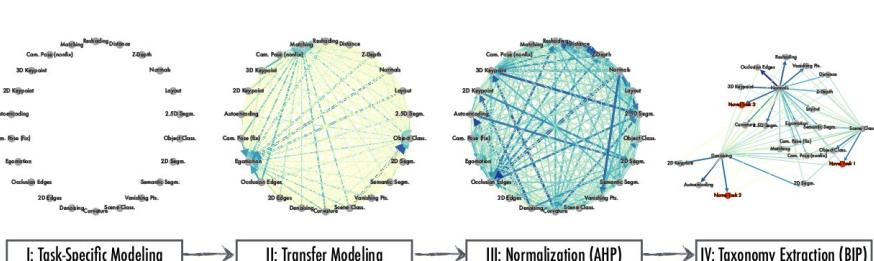
- Usar modelo entrenado para tarea A en la solución de tarea B



Zamir, Taskonomy: Disentangling Task Transfer Learning CVPR 2018

Estrategias de aprendizaje por transferencia

- Taskonomy: Modelo computacional para cuantificar la relación entre tareas. Un modelo unificado para “transfer learning”.



Zamir, Taskonomy: Disentangling Task Transfer Learning CVPR 2018

7.1 Aprendizaje de Representaciones

- ¿Qué es una representación? Representaciones jerárquicas
 - Estrategias de aprendizaje
 - Aprendizaje por Transferencia (Transfer Learning)
 - Aprendizaje Multi-tarea (Multi-task learning)
 - Aprendizaje Auto-supervisado (self-supervised learning)
 - Aprendizaje por Adaptación de Dominio

Estrategia de aprendizaje multi-tarea

- Aprendizaje Multitarea (*Multi-task learning*)
 - Estrategia de aprendizaje en la que múltiples tareas se resuelven al mismo tiempo
- Se construye una representación compartida, de tal manera que cada tarea ayuda a las otras a aprender mejor.
- Potencialmente puede:
 - Disminuir el tiempo de aprendizaje
 - Disminuir el tiempo de inferencia
 - Obtener modelos más compactos
 - Incrementar la tasa de acierto
 - Incrementar la eficiencia de cada muestra
 - Aprender mejores representaciones

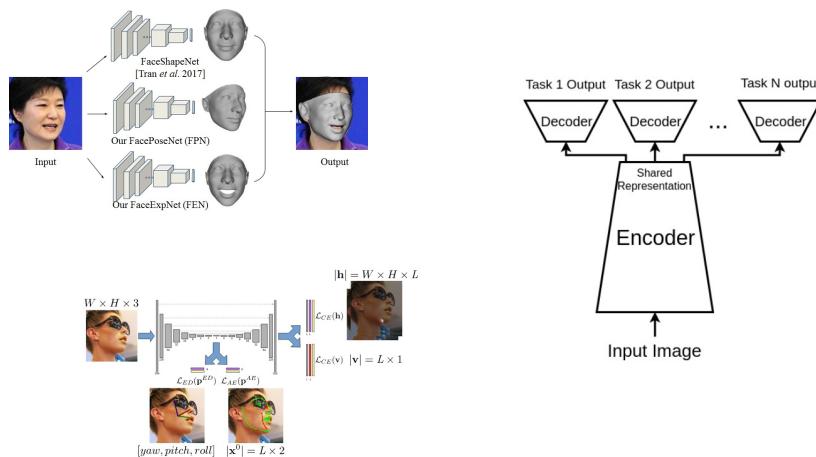
Estrategia de aprendizaje multi-tarea

- No siempre mejor que aprender con una tarea (**Transferencia Negativa**)
- Depende de las relaciones entre las tareas:
 - Las tareas pueden aprender a ritmos diferentes
 - Una tarea puede dominar el aprendizaje
 - Los gradientes pueden interferir
 - La superficie de la función de coste puede ser más complicado
- También depende de:
 - Tamaño de la red (¿suficientes parámetros para todas las tareas?)
 - Localización de los estimadores de cada tarea en la arquitectura

Standley, Which Tasks Should Be Learned Together in Multi-task Learning? ICML 2020

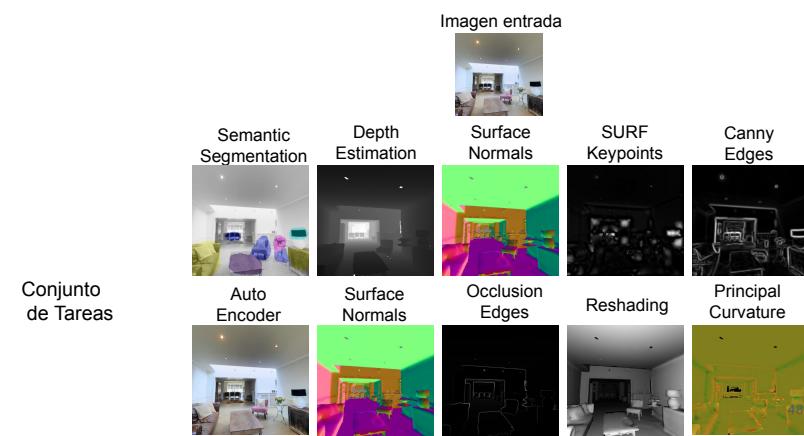
Estrategia de aprendizaje multi-tarea

- El estimador de cada tarea se puede colocar de diferentes formas



Estrategia de aprendizaje multi-tarea

- **Ejemplo:** Análisis de imágenes en interiores

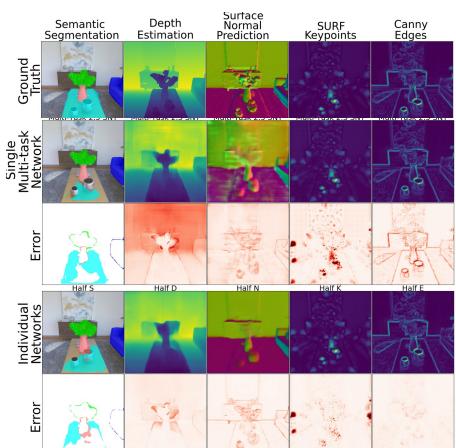


Standley, Which Tasks Should Be Learned Together in Multi-task Learning? ICML 2020

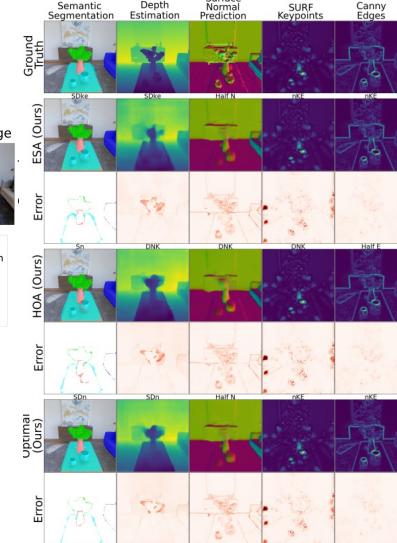
Estrategia de aprendizaje multi-tarea

- Ejemplo: Análisis de imágenes en interiores.

Resultados cualitativos:



Standley, Which Tasks Should Be Learned Together in Multi-task Learning? ICML 2020



Ground Truth

Single-task Network

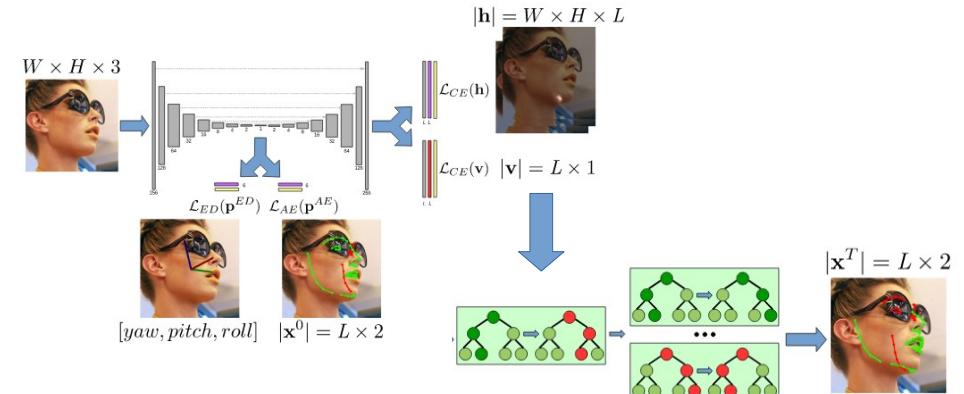
Error

Individual Networks

Error

Estrategia de aprendizaje multi-tarea

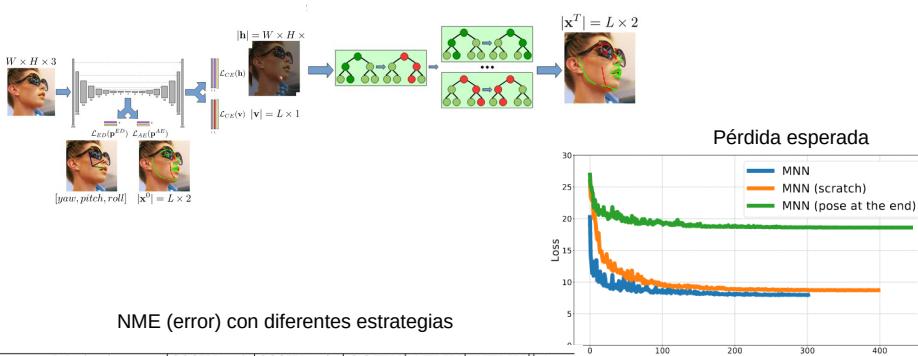
- Ejemplo: Orientación de la cabeza, localización de landmarks y su visibilidad



Valle, Multi-task head pose estimation in the wild, PAMI 2021

Estrategia de aprendizaje multi-tarea

- Ejemplo: Orientación de la cabeza, localización de landmarks y su visibilidad



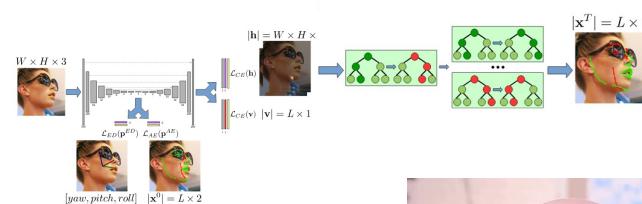
NME (error) con diferentes estrategias

Method	300W pub	300W priv	COFW	AFLW	WFLW	Avg
Single task	1.91	2.22	2.67	3.43	2.46	2.54
Sym	1.76	1.97	2.57	3.35	2.10	2.35
Multi-task	1.59	1.96	2.36	3.22	2.08	2.24
Pre+Sym	1.56	1.96	2.34	3.23	2.11	2.24

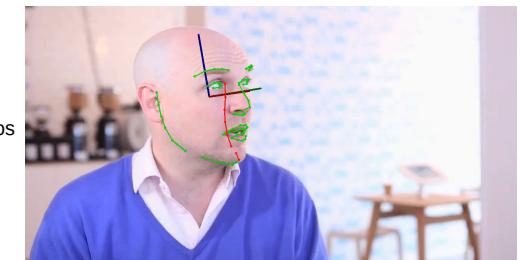
Valle, Multi-task head pose estimation in the wild, PAMI 2021

Estrategia de aprendizaje multi-tarea

- Ejemplo: Orientación de la cabeza, localización de landmarks y su visibilidad



Resultados cualitativos



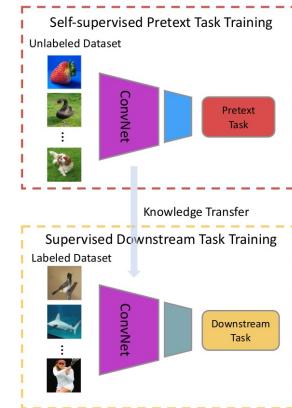
Valle, Multi-task head pose estimation in the wild, PAMI 2021

7.1 Aprendizaje de Representaciones

- ¿Qué es una representación? Representaciones jerárquicas
- Estrategias de aprendizaje
 - Aprendizaje por Transferencia (Transfer Learning)
 - Aprendizaje Multi-tarea (Multi-task learning)
 - **Aprendizaje Auto-supervisado (self-supervised learning)**
 - Aprendizaje por Adaptación de Dominio

Estrategia de aprendizaje auto-supervisado

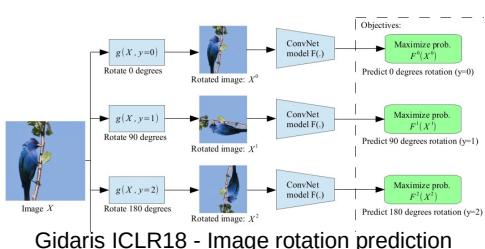
- Aprendizaje auto-supervisado (*Self-supervised learning*)
 - El modelo se entrena con etiquetas generadas automáticamente
- Tareas difíciles de resolver y que producen una buena representación.
- Las etiquetas permiten formular la función de pérdida automáticamente
- Tipos de tareas
 - Imágenes
 - Vídeo
 - Multi-modal
 - Generadas sintéticamente



Jing, Self-supervised Visual Feature Learning with Deep Neural Networks, PAMI 2021

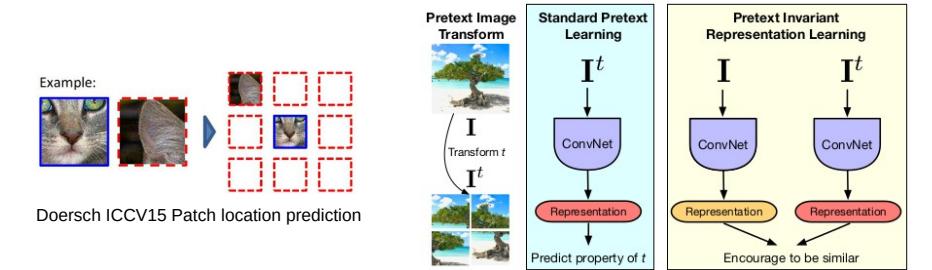
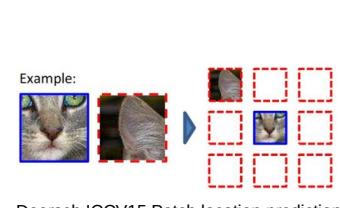
Estrategia de aprendizaje auto-supervisado

- Tareas basadas en imágenes:



Estrategia de aprendizaje auto-supervisado

- Tareas basadas en el contexto en imágenes



Misra CVPR20 – Pretext-invariant Representations

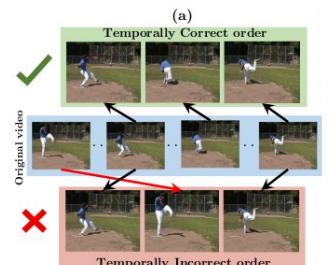


Estrategia de aprendizaje auto-supervisado

- Tareas basadas en el contexto en vídeos



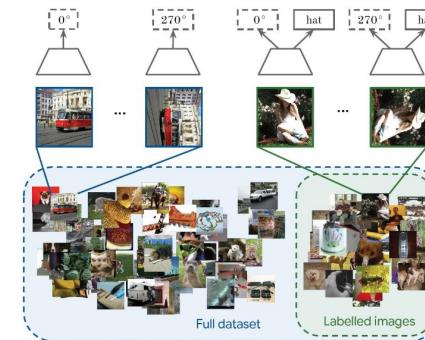
Agarwal ICCV15 Learning to See by Moving



Misra ECCV16-Temporal Order Verification

Aprendizaje semi-supervisado y auto-supervisado

- Utilizar una pequeña cantidad de datos etiquetados junto con una gran cantidad de datos no etiquetados

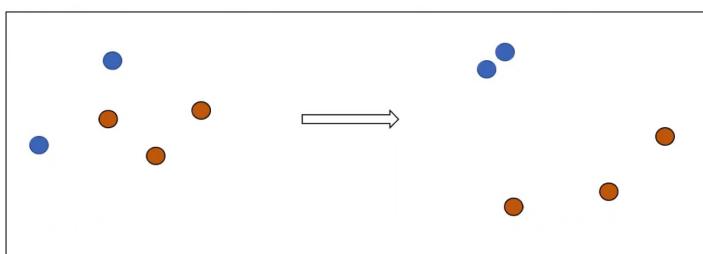


Zhai, S4L: Self-Supervised Semi-Supervised Learning ICCV 2019

Aprendizaje auto-supervisado por contraste

- Aprendizaje por Contraste (*Contrastive Learning*).

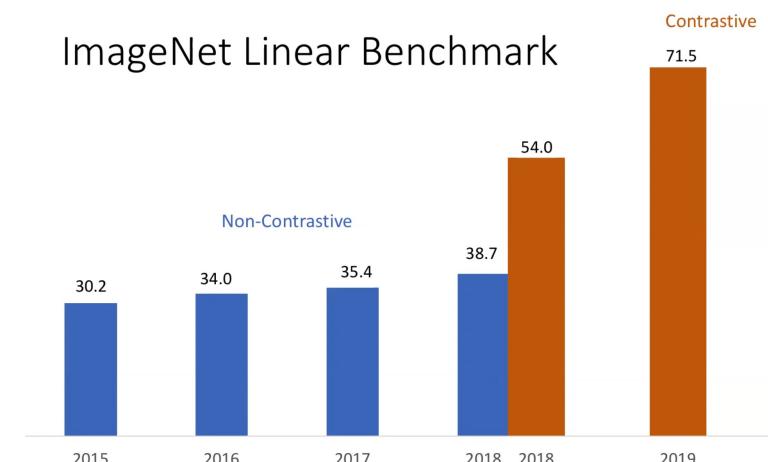
Acerca la representación de parejas de datos similares y aleja los pares no similares



Tian - Contrastive Learning: A General Self-supervised Learning Approach, 2020

Aprendizaje auto-supervisado por contraste

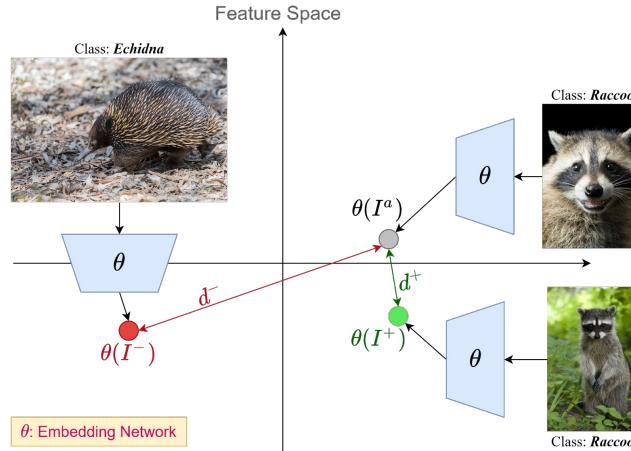
ImageNet Linear Benchmark



Tian - Contrastive Learning: A General Self-supervised Learning Approach, 2020

Aprendizaje auto-supervisado por contraste

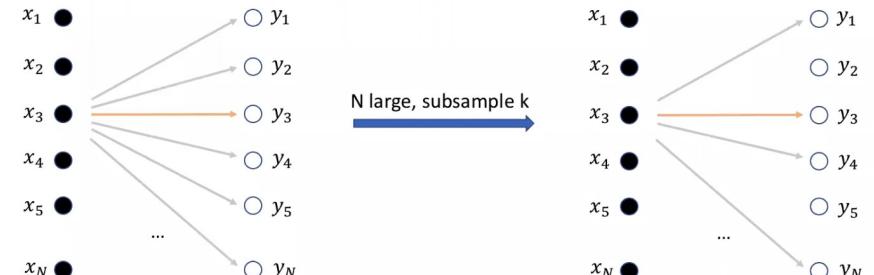
- La tarea de aprendizaje por contraste



Tian - Contrastive Learning: A General Self-supervised Learning Approach, 2020

Aprendizaje auto-supervisado por contraste

- A set pf paired samples $\{x_i, y_i\}_{i=1}^N$

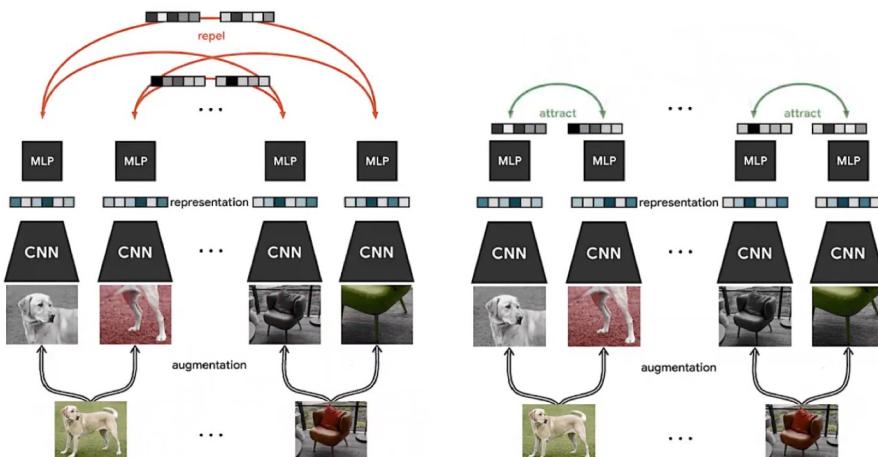


$$\ell = -\log \frac{\exp(\text{sim}(x_+, y_+)/\tau)}{\exp(\text{sim}(x_+, y_+)/\tau) + \sum_{i=1}^k \exp(\text{sim}(x_+, y_-^i)/\tau)}$$

N-pair contrastive Loss

Tian - Contrastive Learning: A General Self-supervised Learning Approach, 2020

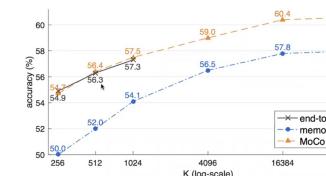
Aprendizaje auto-supervisado por contraste



Tian - Contrastive Learning: A General Self-supervised Learning Approach, 2020

Aprendizaje auto-supervisado por contraste

- El rendimiento mejora significativamente con K



- Sin embargo entrenar con valores de K grande es difícil. Solución:
 - Memory Bank.
 - Momentum (MoCo).
 - Large Batch (SimCLR).

Chen, A Simple Framework for Contrastive Learning of Visual Representations, ICML 20

Tian - Contrastive Learning: A General Self-supervised Learning Approach, 2020

Aprendizaje auto-supervisado por contraste

Method	Arch.	Param.	im/s	Linear	k -NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [11]	RN50	23	1237	69.1	60.7
MoCov2 [13]	RN50	23	1237	71.1	61.9
InfoMin [54]	RN50	23	1237	73.0	65.3
BarlowT [66]	RN50	23	1237	73.2	66.0
OBoW [21]	RN50	23	1237	73.8	61.9
BYOL [23]	RN50	23	1237	74.4	64.8
DCv2 [9]	RN50	23	1237	75.2	67.1
SwAV [9]	RN50	23	1237	75.3	65.7
DINO	RN50	23	1237	75.3	67.5

Table 2: **Linear and k -NN classification on ImageNet.** We report top-1 accuracy for linear and k -NN evaluations on the validation set of ImageNet for different self-supervised methods.

Caron – DINO: Emerging Properties in Self-Supervised Vision Transformers 2023.

Aprendizaje con Adaptación de Dominio

- **Doman Adaptation.** Aprovecha los datos etiquetados en el dominio fuente para entrenar un modelo con datos etiquetados o no en un dominio objetivo diferente.



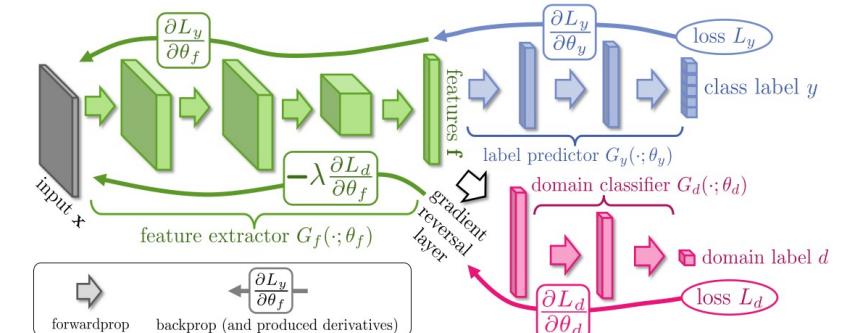
G. Csurka, Domain Adaptation for Visual Applications: A Comprehensive Survey 2017

7.1 Aprendizaje de Representaciones

- ¿Qué es una representación? Representaciones jerárquicas
- Estrategias de aprendizaje
 - Aprendizaje por Transferencia (Transfer Learning)
 - Aprendizaje Multi-tarea (Multi-task learning)
 - Aprendizaje Auto-supervisado (self-supervised learning)
 - Aprendizaje por Adaptación de Dominio

Aprendizaje con Adaptación de Dominio

- **Doman Adaptation.** Aprovecha los datos etiquetados en el dominio fuente para entrenar un modelo con datos etiquetados o no en un dominio objetivo diferente.
- **Adversarial Domain Adaptation**



Ganin, Domain-Adversarial Training of Neural Networks, JMLR 2017

Aprendizaje de Representaciones

- El aprendizaje de representaciones es un concepto central en el desarrollo de métodos de aprendizaje automático para IA.
- Conceptos como:
 - Transfer learning
 - Multi-task learning
 - Unsupervised learning
 - Domain Adaptation

son diferentes manifestaciones del aprendizaje de representaciones.

7.2 Modelos Generativos

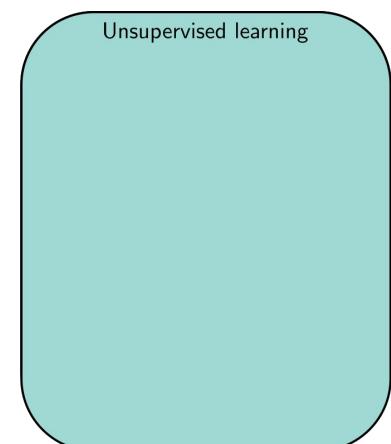
- Modelos Generativos
- Generative Adversarial Models (GANs)

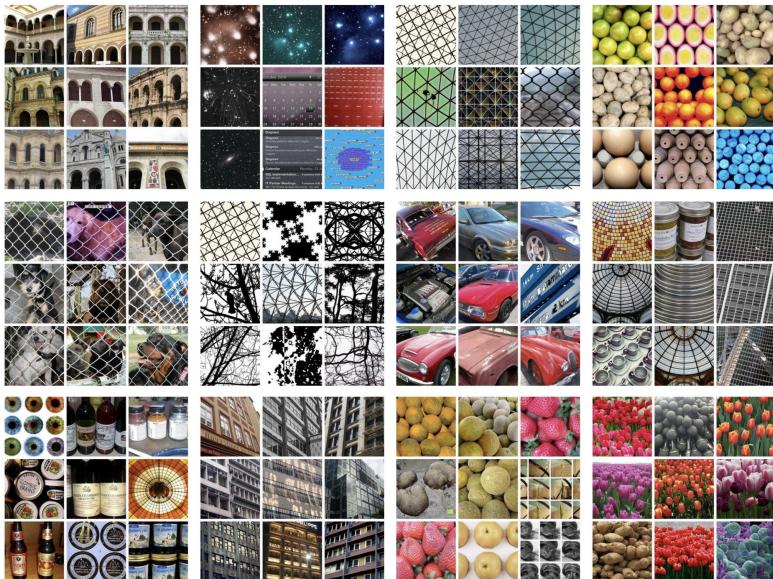
7.2 Modelos Generativos

- Modelos Generativos
- Generative Adversarial Models (GANs)

Aprendizaje no supervisado

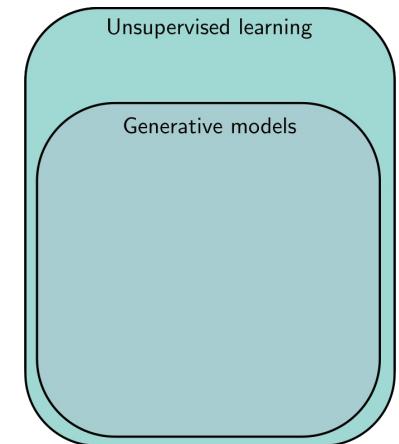
- Aprender de los datos sin etiquetas:
 - Clustering





Aprendizaje no supervisado

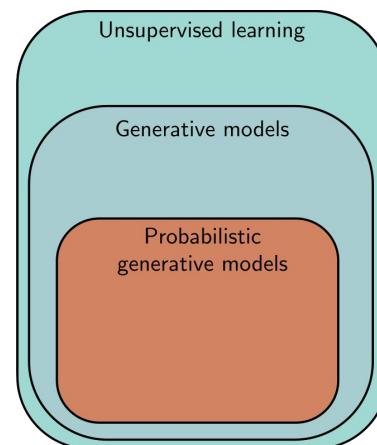
- Aprender de los datos sin etiquetas:
 - Clustering
- Modelos generativos para crear ejemplos:
 - Generative Adversarial Networks.



• DeepCluster: Deep clustering for unsupervised learning of visual features. Caron et al. ECCV 2018.

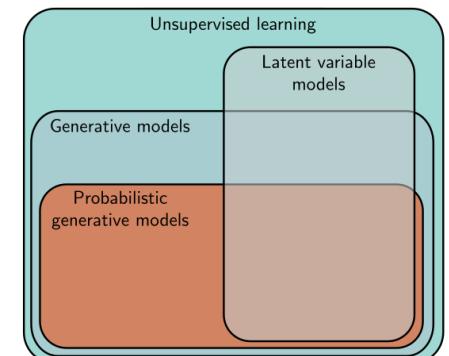
Aprendizaje no supervisado

- Aprender de los datos sin etiquetas:
 - Clustering
- Modelos generativos para crear ejemplos:
 - Generative Adversarial Networks.
- Modelos probabilísticos generativos para aprender la distribución de los datos:
 - Variational Autoencoders
 - Normalizing flows
 - Diffusion models



Variables latentes

- **Estrategia:** definir una función entre los ejemplos, \mathbf{x} , y un conjunto de **variables latentes**, \mathbf{z} (no observadas):
 - Capturan la estructura subyacente de los datos
 - Menor dimensión que \mathbf{x}
- Si encontramos $\text{Pr}(\mathbf{z})$ podemos generar aleatoriamente nuevas variables latentes y llevarlos al espacio original: $\mathbf{z} \rightarrow \mathbf{x}$



Deseable en un modelo generativo ...

- **Muestreo eficiente.** Generar datos de con poco coste computacional.
- **Muestras de gran calidad.** Indistinguibles de las reales.
- **Cobertura.** Debe representar la distribución completa de los datos de entrenamiento.
- **Espacio latente coherente.** Cada $\mathbf{z} \rightarrow \mathbf{x}$ genera un ejemplo plausible. Cambios suaves en \mathbf{z} generan cambios suaves en \mathbf{x} .
- **Espacio latente desentrelazado.** Manipular cada dimensión en \mathbf{z} debería corresponder a un cambio en una propiedad interpretable de los datos.
- **Cálculo eficiente de la verosimilitud.** Si el modelo es probabilístico la probabilidad de un ejemplo se debe calcular rápido.

Un buen modelo generativo ...

Model	Efficient	Sample quality	Coverage	Well-behaved latent space	Disentangled latent space	Efficient likelihood
GANs	✓	✓	✗	✓	?	n/a
VAEs	✓	✗	?	✓	?	✗
Flows	✓	✗	?	✓	?	✓
Diffusion	✗	✓	?	✗	✗	✗

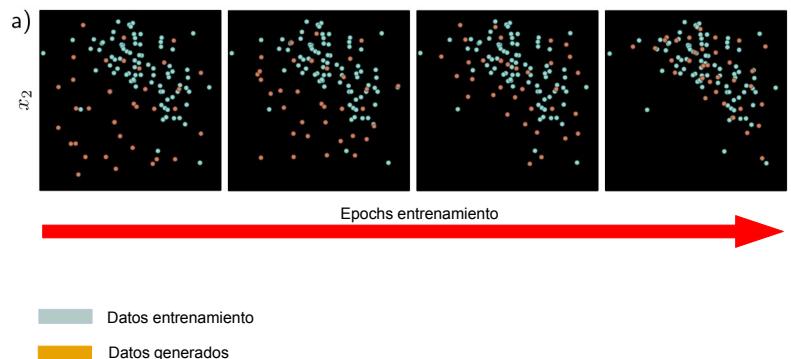
Figure 14.3 Properties of four generative models. Neither generative adversarial networks (GANs), variational autoencoders (VAEs), normalizing flows (Flows), nor diffusion models (diffusion) have the full complement of desirable properties.

7.2 Modelos Generativos

- Modelos Generativos
- Generative Adversarial Models (GANs)

Generative Adversarial Networks (GANs)

- Generan ejemplos, \mathbf{x} , a partir de variables latentes \mathbf{z} , utilizando una función de pérdida que favorece que los ejemplos generados sean indistinguibles de los reales.



Generative Adversarial Networks (GANs)

- **Generador.** Red de neuronas que crea ejemplos a partir de una entrada \mathbf{z}_j generada aleatoriamente:

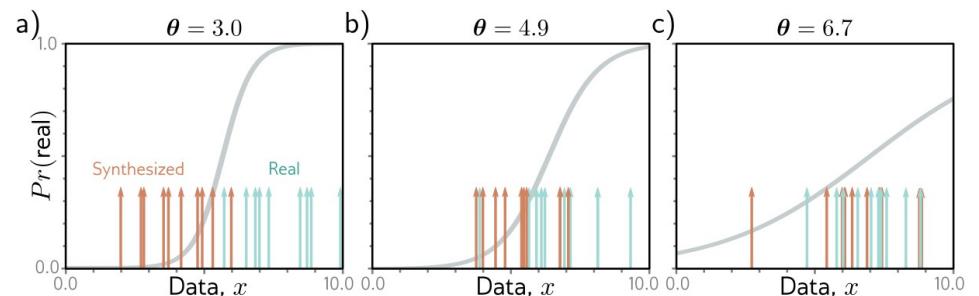
$$\mathbf{x}_j^* = g[\mathbf{z}_j, \theta]$$

- **Discriminador.** Red de neuronas entrenada para distinguir los ejemplos reales \mathbf{x} , de los generados:

$$\begin{aligned} f[\mathbf{x}, \Omega] &\rightarrow \text{"Real"} \\ f[g[\mathbf{z}, \theta], \Omega] &\rightarrow \text{"Generated"} \end{aligned}$$

Ejemplo sencillo de GAN

- **Generador:** $x_j^* = g[z_j, \theta] = z_j + \theta$, $z_j \sim N(\cdot; 0, 1)$
- **Discriminador.** Regresión logística
- Se alterna entre actualizar θ (generador) y Ω (discriminador)



Función de pérdida de la GAN

- **Discriminador.** Etiqueta $y = 1$ cuando la entrada es real, $y = 0$ con entrada generada:

$$\hat{\phi} = \operatorname{argmin}_{\phi} \left[\sum_i -(1 - y_i) \log [1 - \operatorname{sig}[f[\mathbf{x}_i, \phi]]] - y_i \log [\operatorname{sig}[f[\mathbf{x}_i, \phi]]] \right]$$

Función de pérdida de la GAN

- **Discriminador.** Etiqueta $y = 1$ cuando la entrada es real, $y = 0$ con entrada generada:

$$\hat{\phi} = \operatorname{argmin}_{\phi} \left[\sum_j -\log [1 - \operatorname{sig}[f[\mathbf{x}_j^*, \phi]]] - \sum_i \log [\operatorname{sig}[f[\mathbf{x}_i, \phi]]] \right]$$

Hemos separado reales (i) y generados (j) en dos sumatorios

Función de pérdida de la GAN

- **Discriminador.** Etiqueta $y = 1$ cuando la entrada es real, $y = 0$ con entrada generada:

$$\hat{\phi} = \operatorname{argmin}_{\phi} \left[\sum_j -\log [1 - \text{sig}[f[g[\mathbf{z}_j, \theta], \phi]]] - \sum_i \log [\text{sig}[f[\mathbf{x}_i, \phi]]] \right]$$

Introduciendo el generador: $\mathbf{x}_j^* = g[\mathbf{z}_j, \theta]$

Función de pérdida de la GAN

- **Discriminador.** Etiqueta $y = 1$ cuando la entrada es real, $y = 0$ con entrada generada:

$$\hat{\theta} = \operatorname{argmax}_{\theta} \left[\min_{\phi} \left[\sum_j -\log [1 - \text{sig}[f[g[\mathbf{z}_j, \theta], \phi]]] - \sum_i \log [\text{sig}[f[\mathbf{x}_i, \phi]]] \right] \right]$$

queremos generar datos con los que el discriminador se equivoque → encontrar θ que **maximiza la cross entropy binaria del discriminador**

Entrenamiento de la GAN

- El **optimizador** busca los parámetros que son un mínimo en el discriminador y un máximo en el generador (**minmax problem**)

- El **discriminador** intenta encontrar mejores maneras de distinguir ejemplos reales de los generados

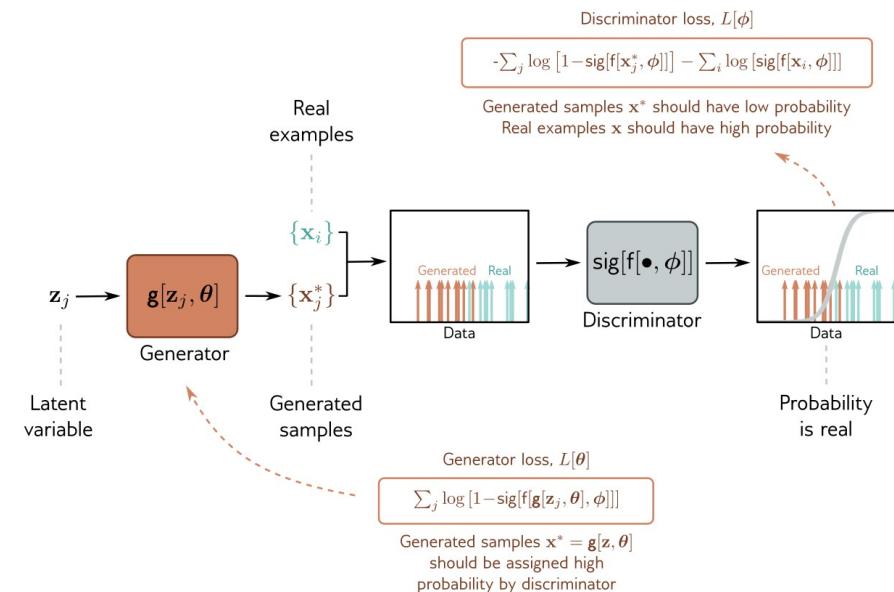
$$L[\phi] = \sum_j -\log [1 - \text{sig}[f[g[\mathbf{z}_j, \theta], \phi]]] - \sum_i \log [\text{sig}[f[\mathbf{x}_i, \phi]]]$$

- El **generador** trata de encontrar la manera de engañar al discriminador

$$L[\theta] = \sum_j \log [1 - \text{sig}[f[g[\mathbf{z}_j, \theta], \phi]]]$$

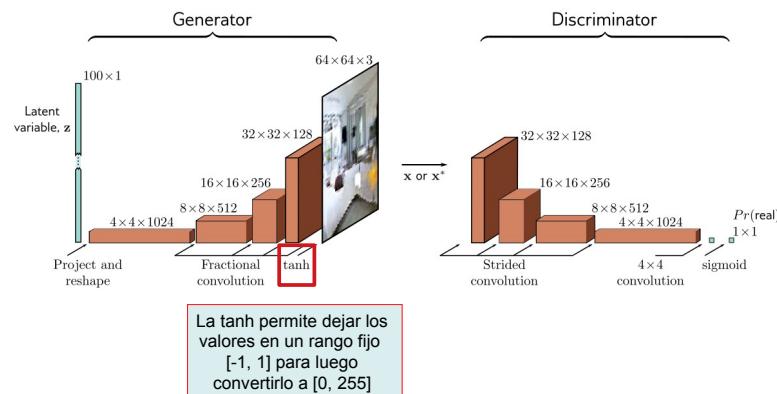
Minimizamos la cross-entropy cambiada de signo para los datos generados → la maximizamos

Entrenamiento de la GAN



Deep Convolutional GAN (DCGAN)

- Ejemplo de GAN que genera imágenes
 - z es de 100 dimensiones con distribución uniforme



Deep Convolutional GAN (DCGAN)



Figure 15.4 Synthesized images from the DCGAN model. a) Random samples drawn from DCGAN trained on a faces dataset. b) Random samples using the ImageNet database (see figure 10.15). c) Random samples drawn from the LSUN scene understanding dataset. Adapted from Radford et al. (2015).

Deep Convolutional GAN (DCGAN)

- Para que “funcione”:
 - Convoluciones con stride y convoluciones traspuestas
 - Batch normalization (excepto en primeras capas del generador y últimas del discriminador)
 - Leaky ReLU en el discriminador
 - Adam pero con parámetro de momento con valor bajo

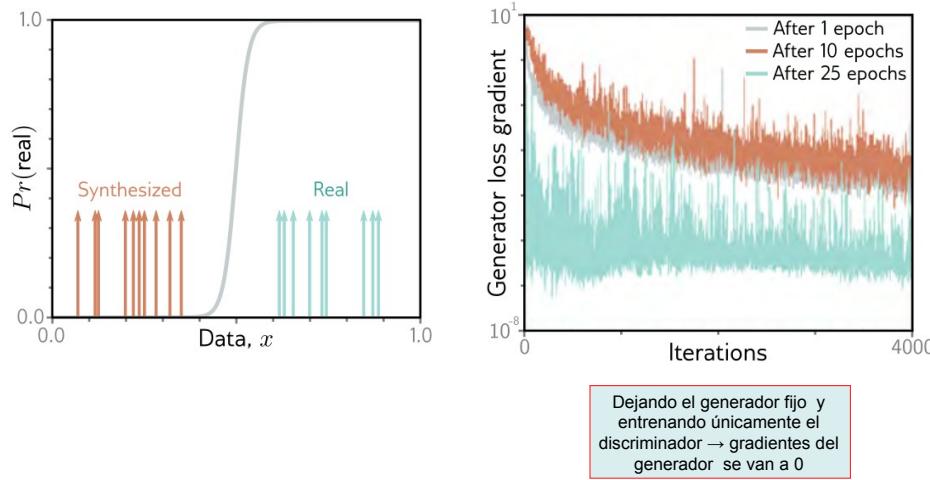
Problemas con GANs

- **Mode dropping.** Algunos tipos de imágenes no se generan (ej. caras con barba).
- **Mode collapse.** Siempre se genera un conjunto limitado de imágenes (ignorando z).



Figure 15.5 Mode collapse. Synthesized images from a GAN trained on the LSUN scene understanding dataset using an MLP generator with a similar number of parameters and layers to the DCGAN. The samples are low quality, and many are similar. Adapted from Arjovsky et al. (2017).

Con discriminador “demasiado bueno” ...



Wasserstein GAN

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α , the learning rate. c , the clipping parameter. m , the batch size. n_{critic} , the number of iterations of the critic per generator iteration.

Require: : w_0 , initial critic parameters. θ_0 , initial generator's parameters.

```

1: while  $\theta$  has not converged do
2:   for  $t = 0, \dots, n_{\text{critic}}$  do
3:     Sample  $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$  a batch from the real data.
4:     Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
5:      $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))]$ 
6:      $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$ 
7:      $w \leftarrow \text{clip}(w, -c, c)$ 
8:   end for
9:   Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
10:   $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))$ 
11:   $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$ 
12: end while

```

Wasserstein GAN

- Función de pérdida de la GAN original:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \left[\sum_j -\log [1 - \operatorname{sig}[f[g[z_j, \theta], \phi]]] - \sum_i \log [\operatorname{sig}[f[x_i, \phi]]] \right]$$

- Wasserstein GAN minimiza:

$$\begin{aligned} L[\phi] &= \sum_j f[x_j^*, \phi] - \sum_i f[x_i, \phi] \\ &= \sum_j f[g[z_j, \theta], \phi] - \sum_i f[x_i, \phi], \end{aligned}$$

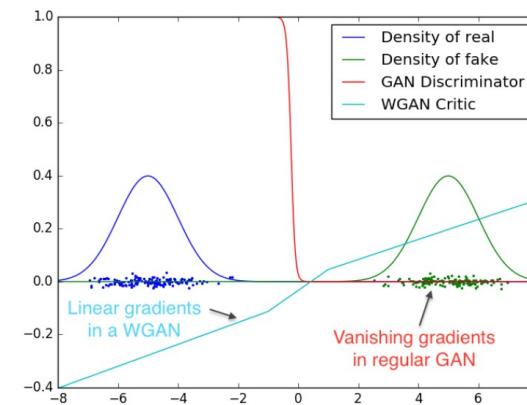
Sujeto a:

$$\left| \frac{\partial f[\mathbf{x}, \phi]}{\partial \mathbf{x}} \right| < 1$$

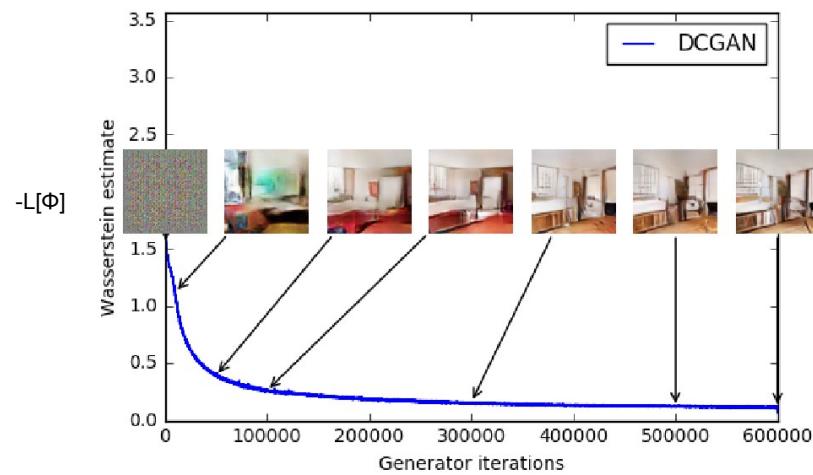
Truncando los valores de Φ al intervalo [-0.01, 0.01]

Wasserstein GAN

- La función f que se aprende con WGAN vs GAN :



Wasserstein + DCGAN

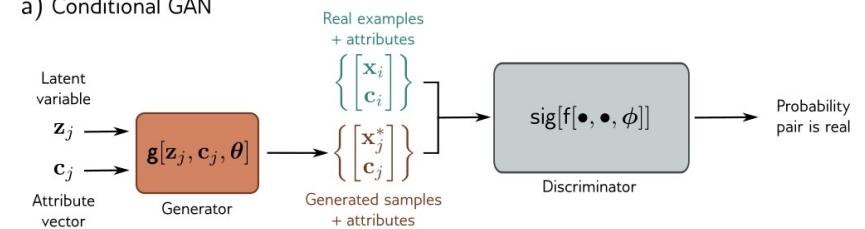


Wasserstein GAN. Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017

GANs condicionadas

- Entrada que condiciona la imagen generada:

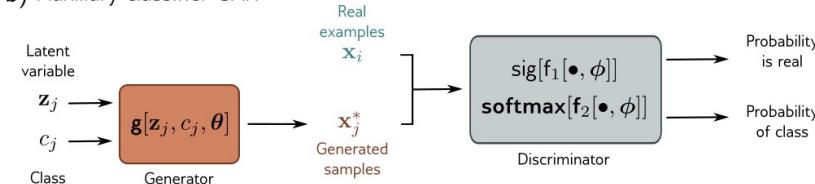
a) Conditional GAN



GANs condicionadas

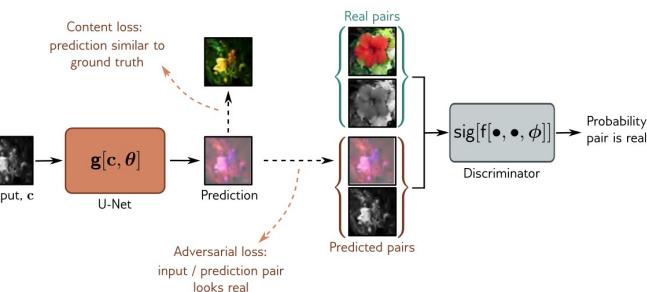
- Entrada que condiciona la imagen generada:

b) Auxiliary classifier GAN

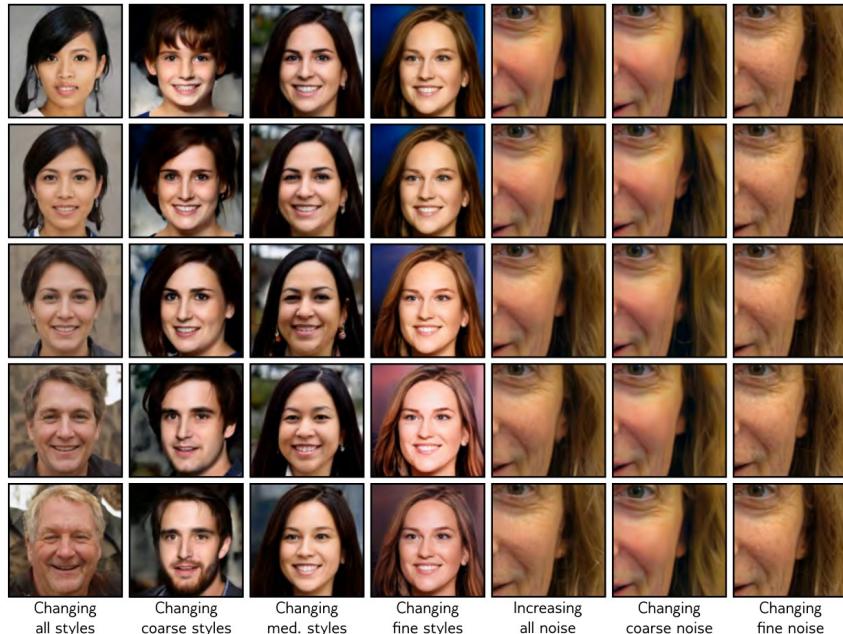
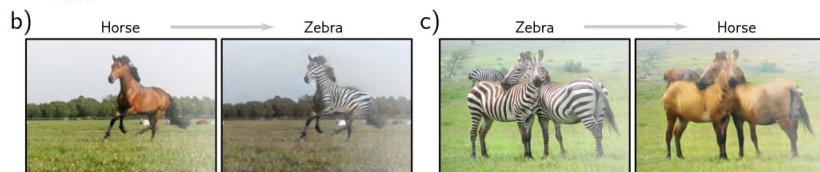
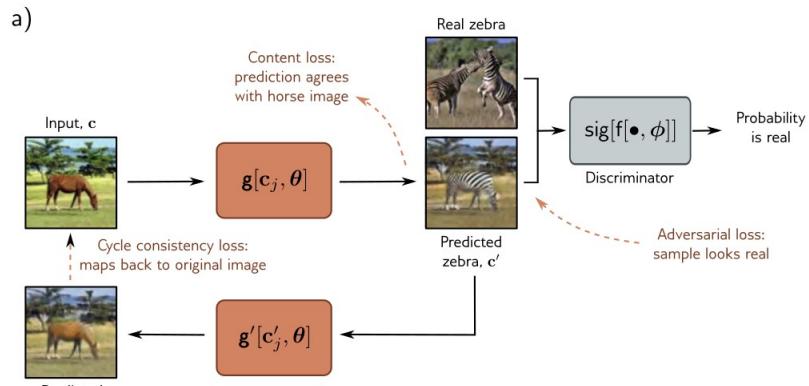


GANs condicionadas

- Entrada que condiciona la imagen generada (Pix2Pix):



CycleGAN



StyleGAN

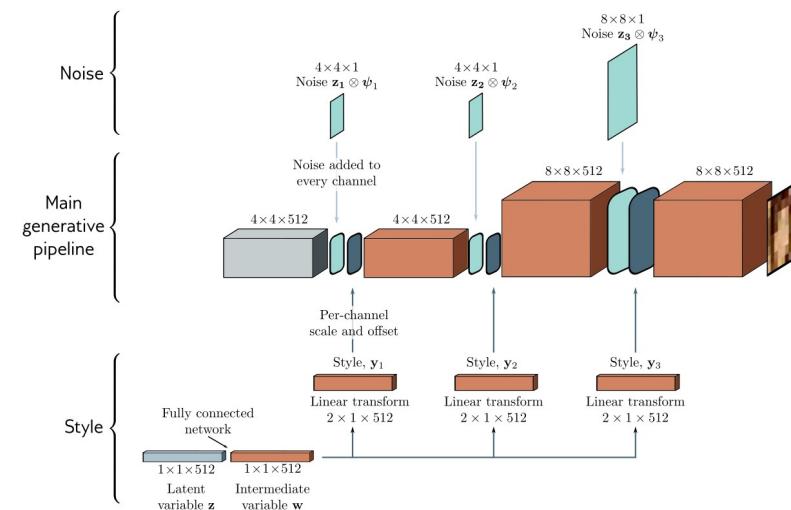


Figure 15.20 StyleGAN results. First four columns show systematic changes in style at various scales. Fifth column shows the effect of increasing noise magnitude. Last two columns show different noise vectors at two different scales.