Modelos de clasificación de imágenes

Visión por Computador II

Juan Carlos Arbeláez

Contenido

- Capas Convolucionales
- Pooling
- Capas FC
- Dropout
- Arquitectura de modelo de clasificación

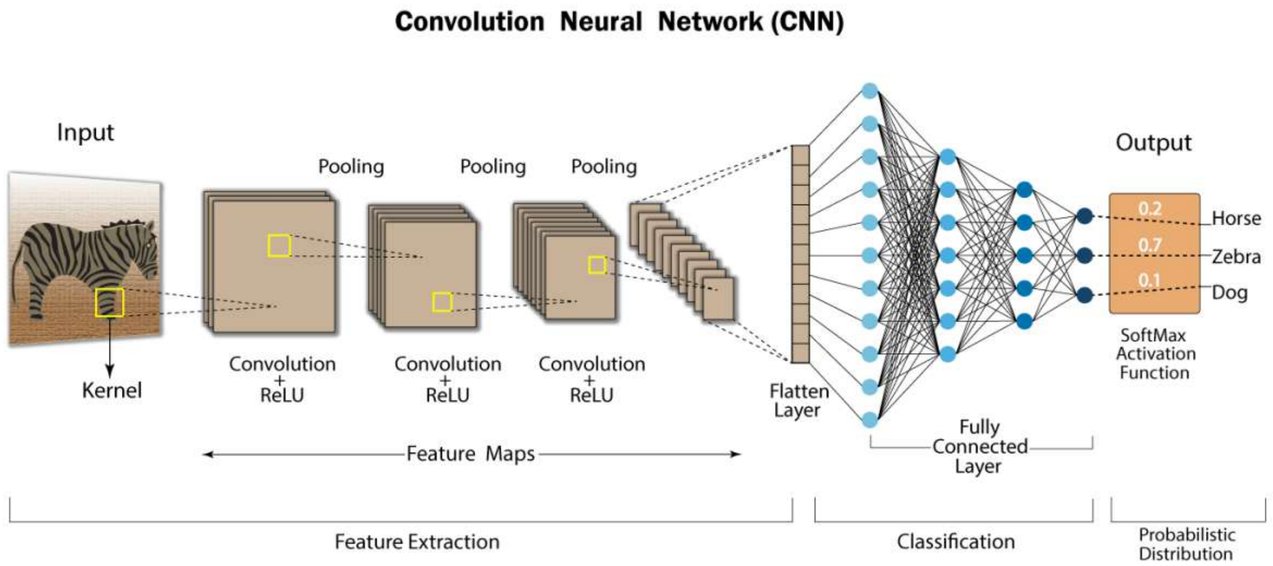
Convolutional Neural Network (CNN)

Capas densas son ineficientes en imágenes

- Píxel por neurona da un gran número de parámetros
- Tiende al *overfitting*

Convoluciones

- Trabaja con volúmenes 3D
- Entrenan multiples filtros para extraer características
- Aprendizaje profundo (Deep Learning)
- Normalmente la última capa es FC



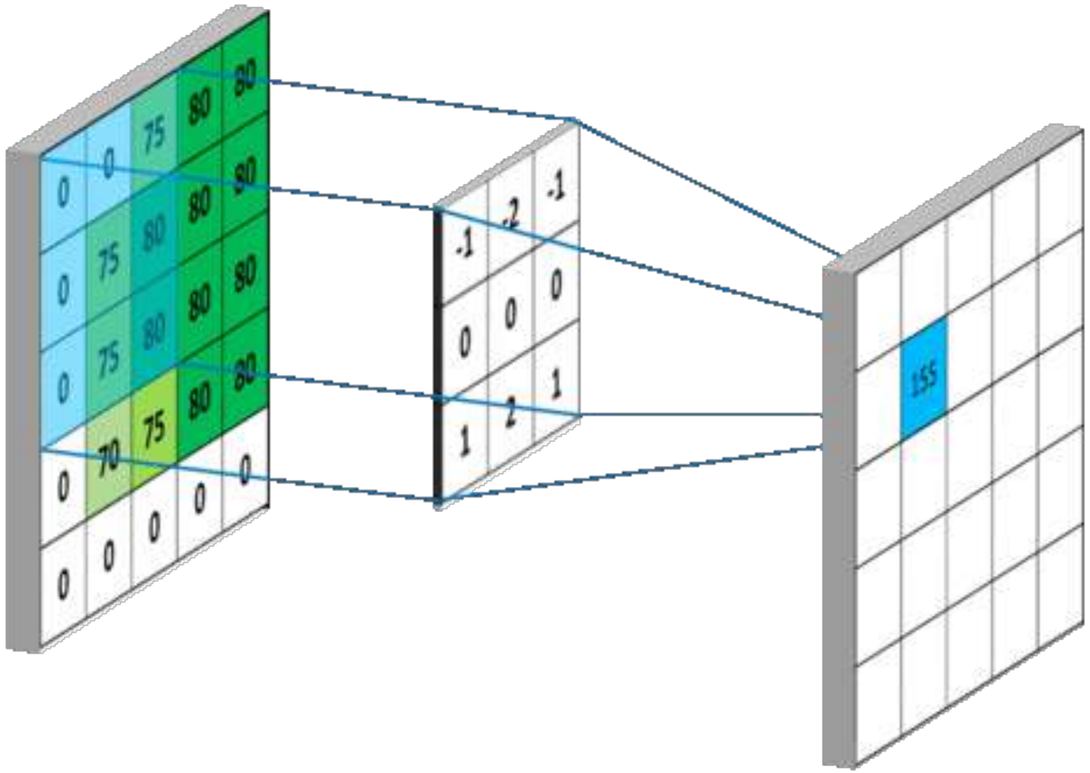
Arquitectura básica de red neuronal convolucional

https://developersbreach.com/convolution-neural-network-deep-learning/

Convolución

- El valor de un píxel de salida se determina como una suma ponderada de los valores de los píxeles de entrada

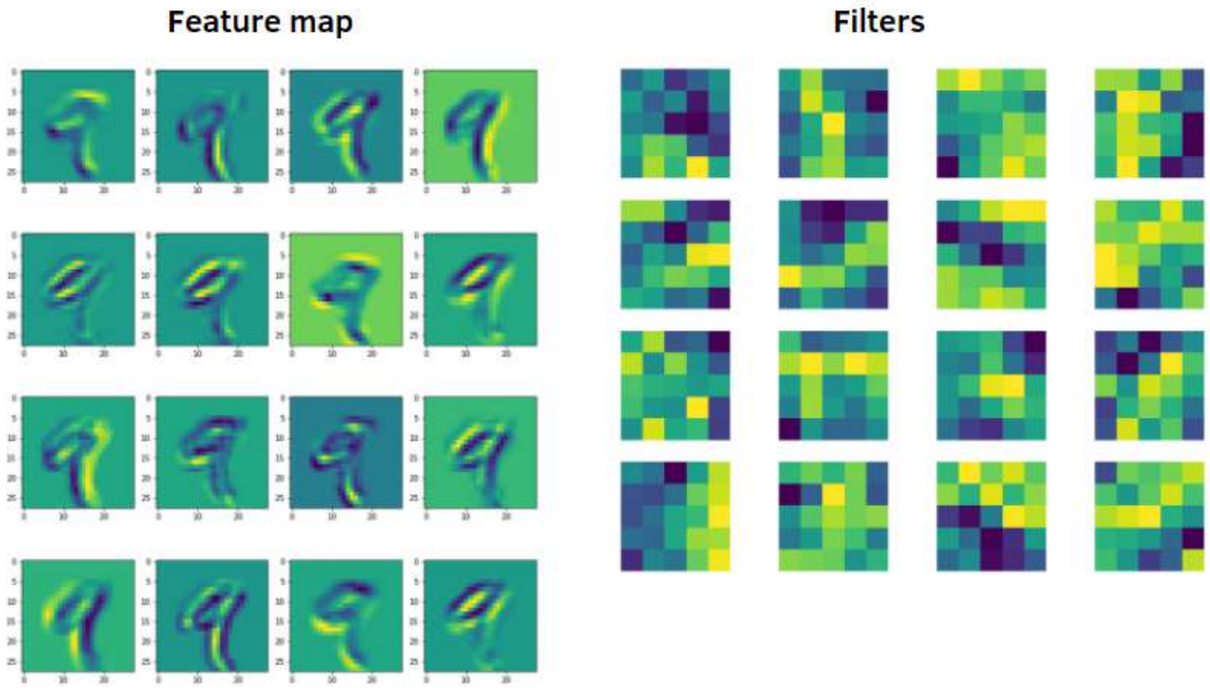
- Los valores del kernel o máscara $h(k, l)$ son llamados coeficientes del filtro



https://mlnotebook.github.io/post/CNN1/

Convolutional Neural Network (CNN)

- **La convolución** es una operación utilizada para extraer características de una imagen
- La convolución se define mediante un *kernel*
- El kernel es una matriz con menor dimensión que la imagen
- Dos tamaños típicos son de 3x3 y 5x5



https://medium.com/dataseries/visualizing-the-feature-maps-and-filters-by-convolutional-neural-networks-e1462340518e

Convolutional Neuronal Networks

- Aprender *Kernels* → Extraer características útiles
- Las convoluciones hacen el trabajo pesado
- Existen diferentes ***tipos de capas:***
  - Convolutional
  - Pooling
  - Normalization
  - Fully Connected

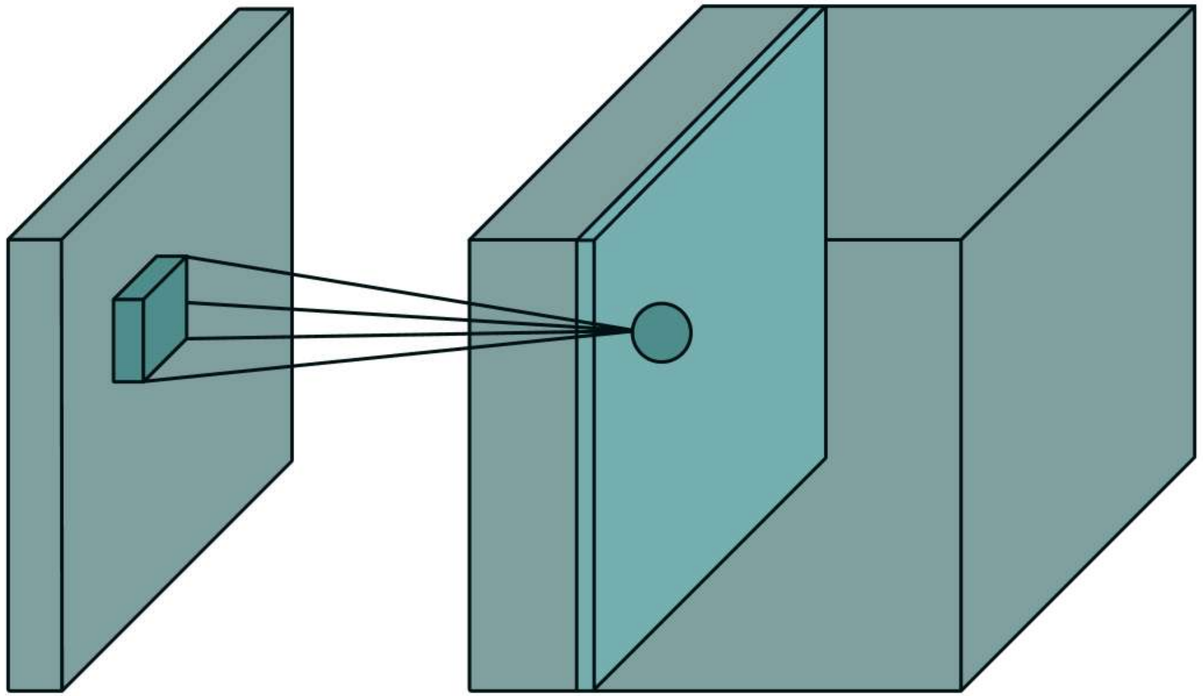
Arquitectura de VGG16

https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918

Convolutional Neuronal Networks

Capa Convolucional

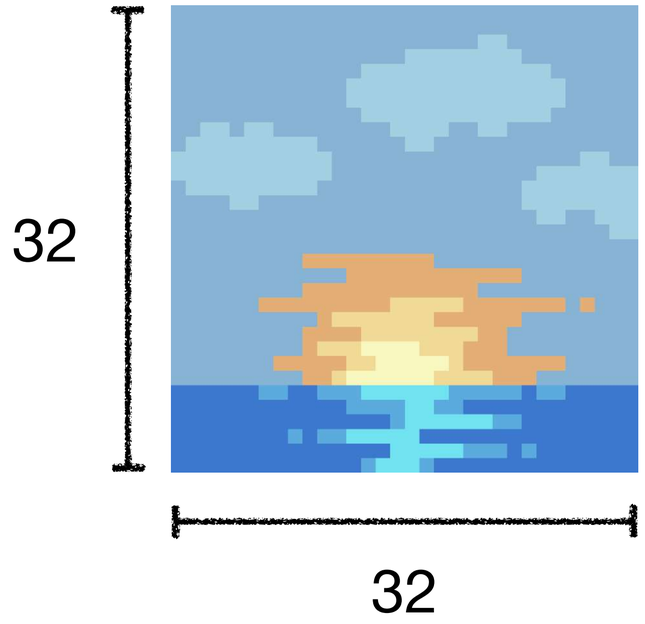
- Compuesta de varios *kernels* cuyos valores son aprendidos:
    i. Realiza las convocaciones
    ii. Se apilan las activaciones
- Tienen la propiedad de que el filtro es compartido para toda la imágen
- Tamaño del filtro: [W, H, Depth (full)]. Ejemplo: [5,5,3] (RGB)



https://serokell.io/blog/introduction-to-convolutional-neural-networks

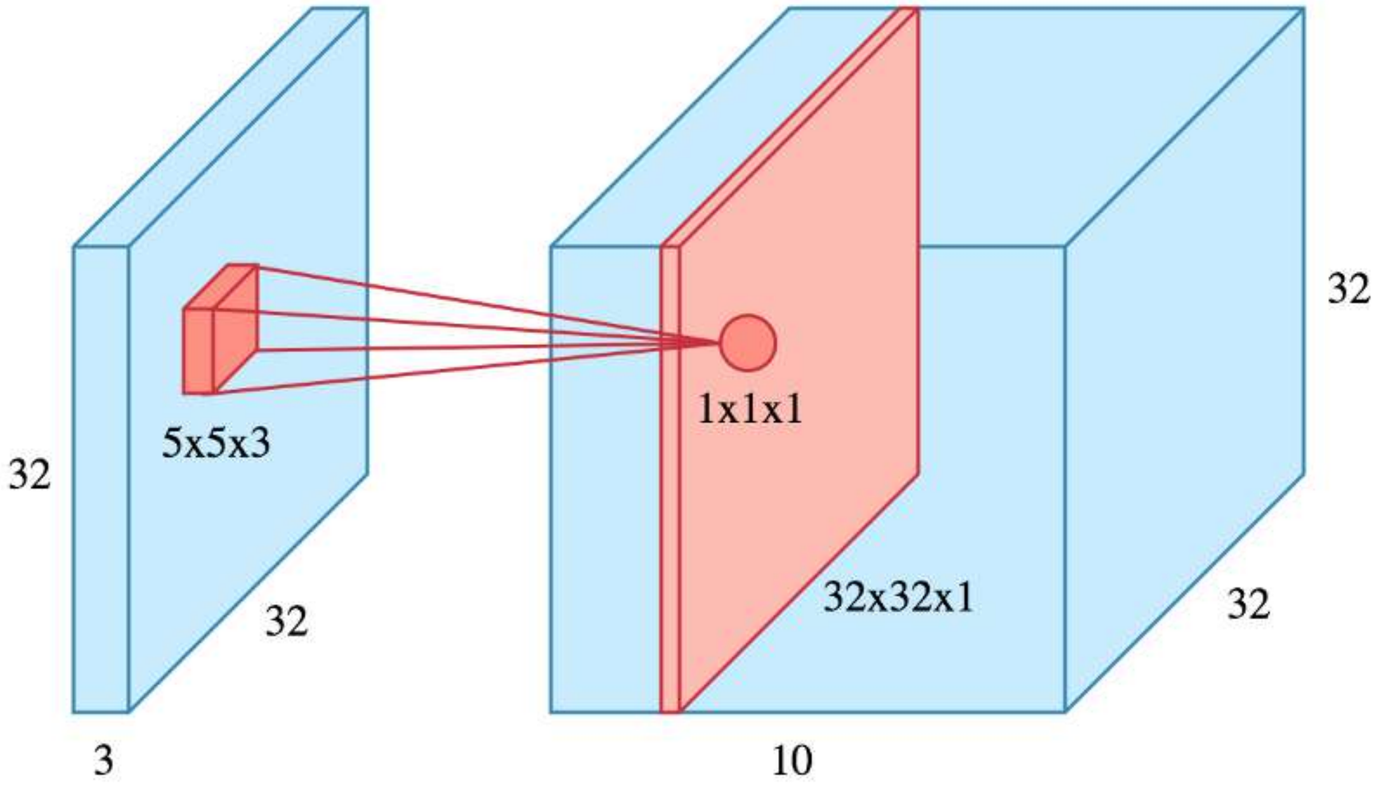
Convolutional Neuronal Networks

De qué tamaño es el volumen resultante de una capa convolucional de 10 filtros de 5x5 aplicada sobre una imagen de 32x32px



Convolutional Neuronal Networks

De qué tamaño es el volumen resultante de una capa convolucional de 10 filtros aplicada sobre una imagen de 32x32px



https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Pooling

Problemas de las capas Convolucionales

Si el mapa de características es conectado directamente con una NN:

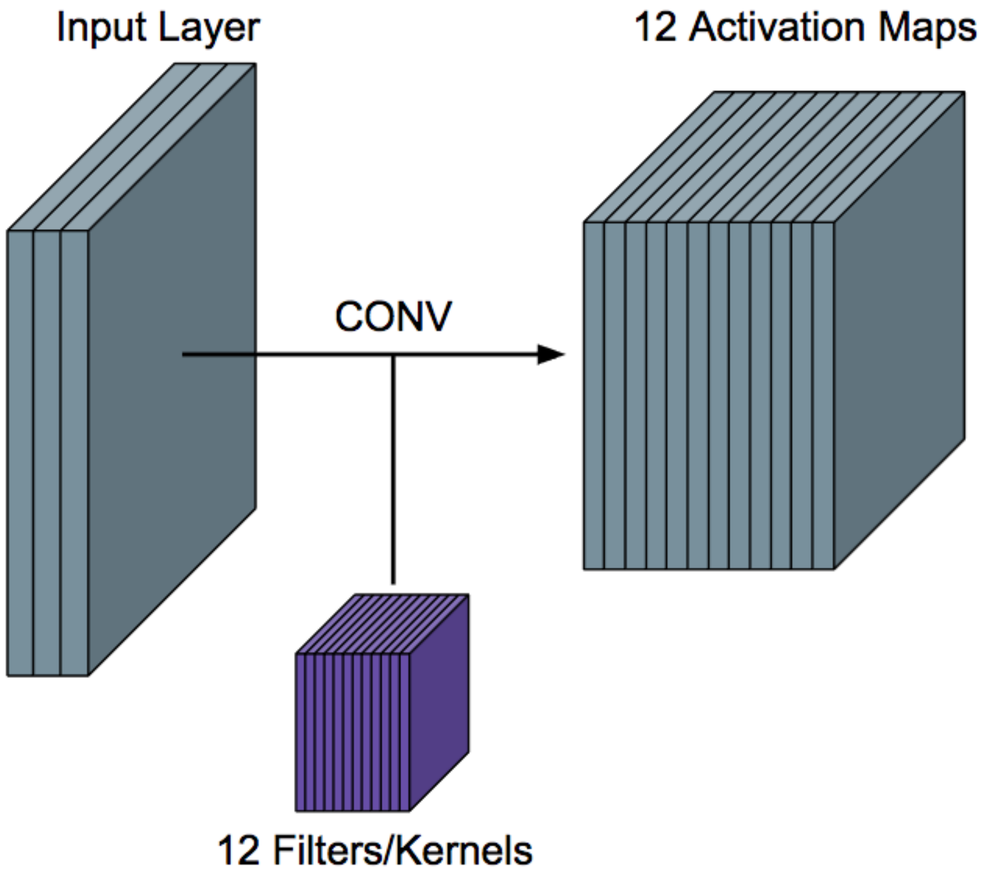
1. Costo computacional
2. *Overfitting*
3. No es invariable a la translación

https://www.azcentral.com/story/entertainment/kids/2021/05/11/public-pools-open-phoenix-mesa-gilbert-chandler-scottsdale/4894434001/

Pooling

1. Costo computacional

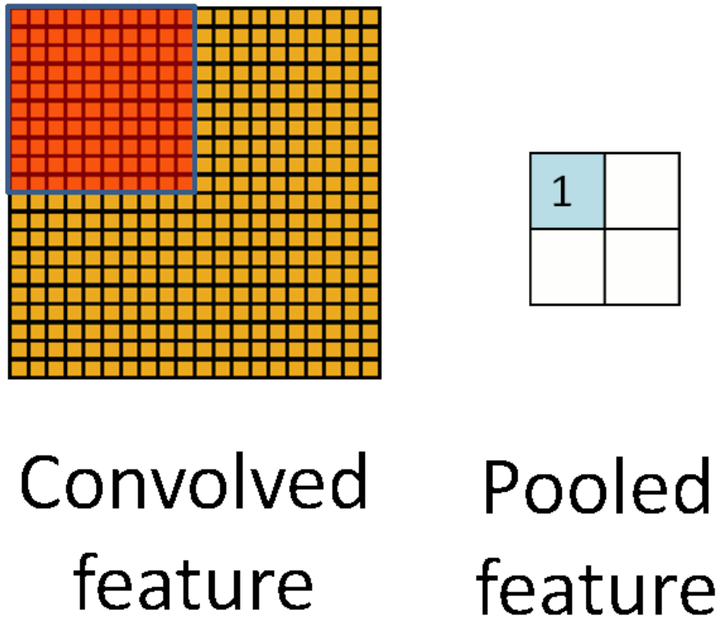
- Imaginemos el mapa de características resultante de aplicar 100 filtros de 3x3 en una imagen de 96x96px con padding de 1.
- ¿De qué tamaño es la el mapa?
- ¿De cuántas unidades es la capa Densa?



https://gamedevacademy.org/introduction-to-convolutional-neural-networks-for-vision-tasks/

Pooling

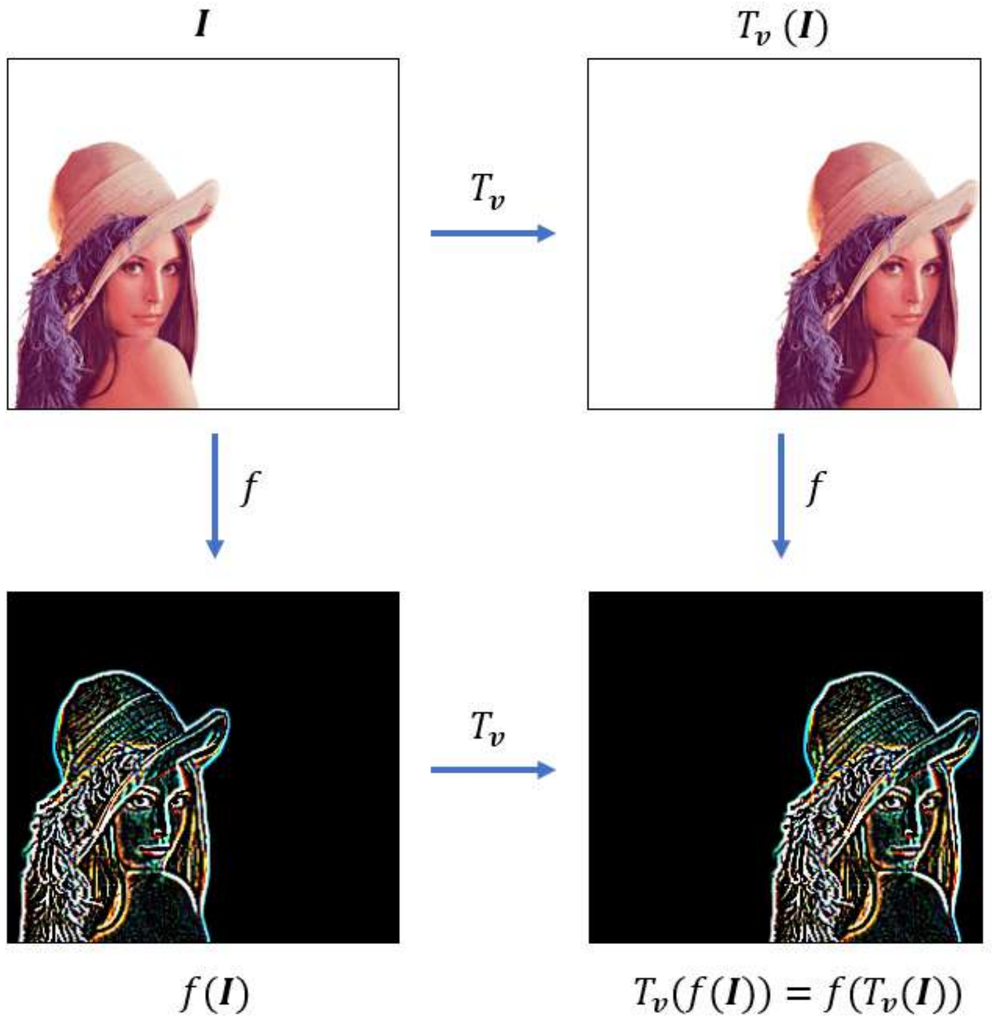
- Para reducir el número de características usamos *Pooling*
- Usamos Convoluciones porque las imágenes tienen la propiedad de ser estacionarias
- Las características que son útiles en una región probablemente también lo sean en otra.
- **Podríamos resumir las características de varias partes de la imagen**



http://deeplearning.stanford.edu/tutorial/supervised/Pooling/

Invariabilidad translacional

- Aunque las convoluciones son equivariantes de traslación y no invariantes
- Una limitación del mapa de características es que registran la posición exacta de las características de la entrada
- Se puede conseguir invariancia de traslación aproximada con el operador de *Pooling*



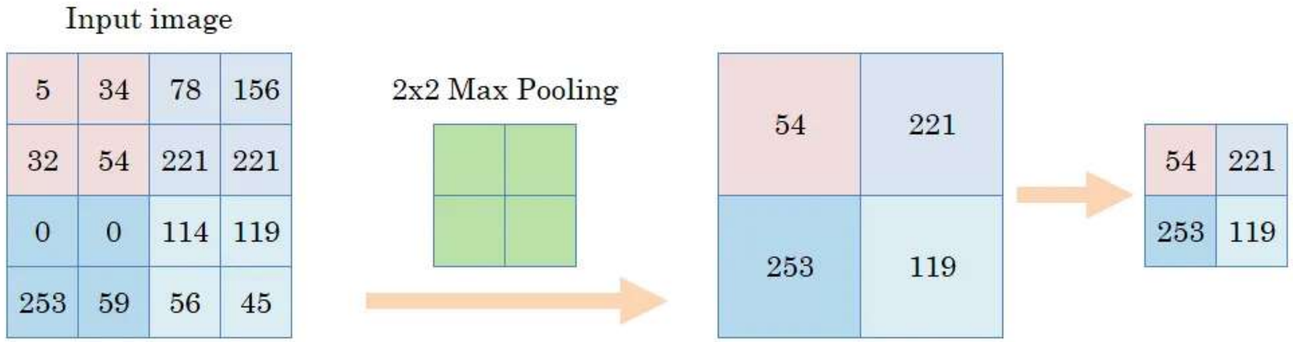
https://www.baeldung.com/cs/translation-invariance-equivariance

Pooling

- Tomamos grupos de píxeles (por ejemplo, grupos de 2x2 píxeles) y realizar una *"agregación"* sobre ellos.

- Una de las posibles *agregaciones* es tomar el valor máximo de los píxeles del grupo (Max Pooling)

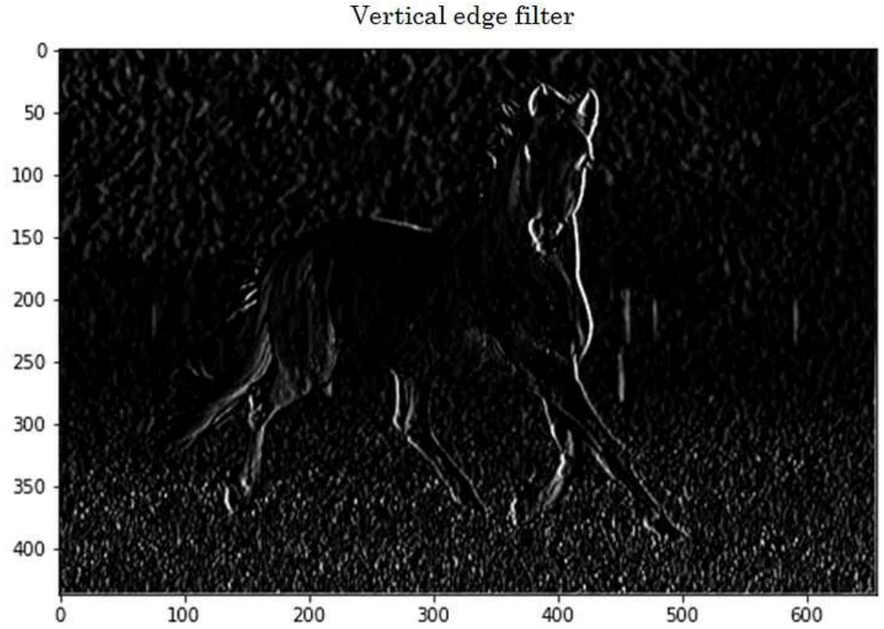
- Otra agregación habitual es tomar la media (Average Pooling)…

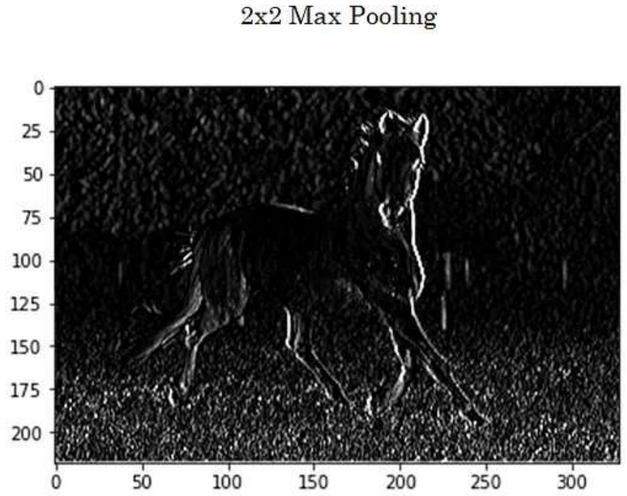


https://towardsdatascience.com/understanding-convolutions-and-pooling-in-neural-networks-a-simple-explanation-885a2d78f211

Pooling

- Si aplicamos un Max Pooling de 2x2
- Reducir a la mitad la información que contiene la imagen
- **Los bordes no sólo se mantienen, sino que se intensifican.**
- Se conservan las características útiles de los filtros





https://towardsdatascience.com/understanding-convolutions-and-pooling-in-neural-networks-a-simple-explanation-885a2d78f211

Pooling in CNN

Una CNN básica puede verse como una secuencia de capas de convolución y capas de Pooling.

Capas convolucionales

Cuando la imagen las atraviesa, se extraen las característica

Capas Pooling

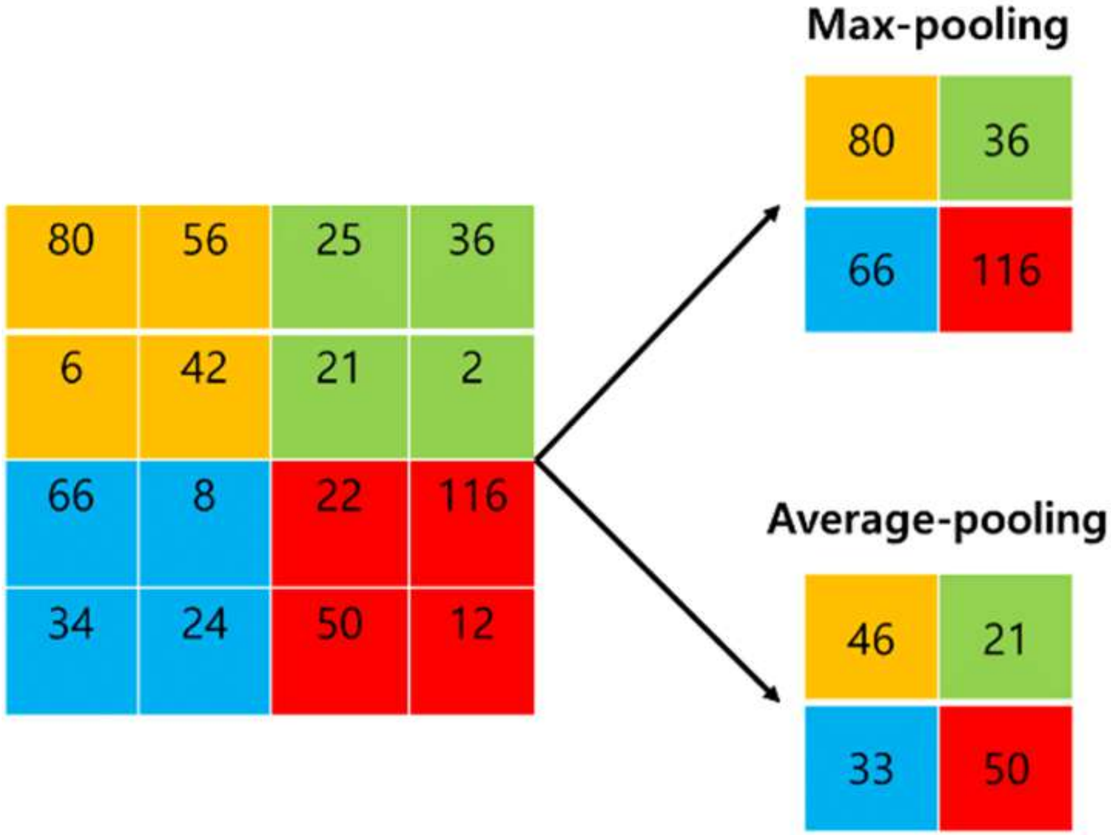
- Las características se intensifican y se mantienen
- Se descarta toda la información que no aporta nada a la tarea.

Types of Pooling

Max pooling: valor máximo del conjunto de pixeles

Min pooling: valor mínimo del conjunto de pixeles

Average pooling: valor promedio del conjunto de pixeles



A Deep Convolutional Neural Network-Based Multi-Class Image Classification for Automatic Wafer Map Failure Recognition in Semiconductor Manufacturing

Tipos de pooling

- No hay un método de agrupación mejor que otro
- La selección depende del tipo de imagen o dato que estemos tratando
- **Average Pooling** suaviza la imagen y, por lo tanto, es posible que no se identifiquen las características
- **Max Pooling** selecciona los píxeles más brillantes de la imagen.

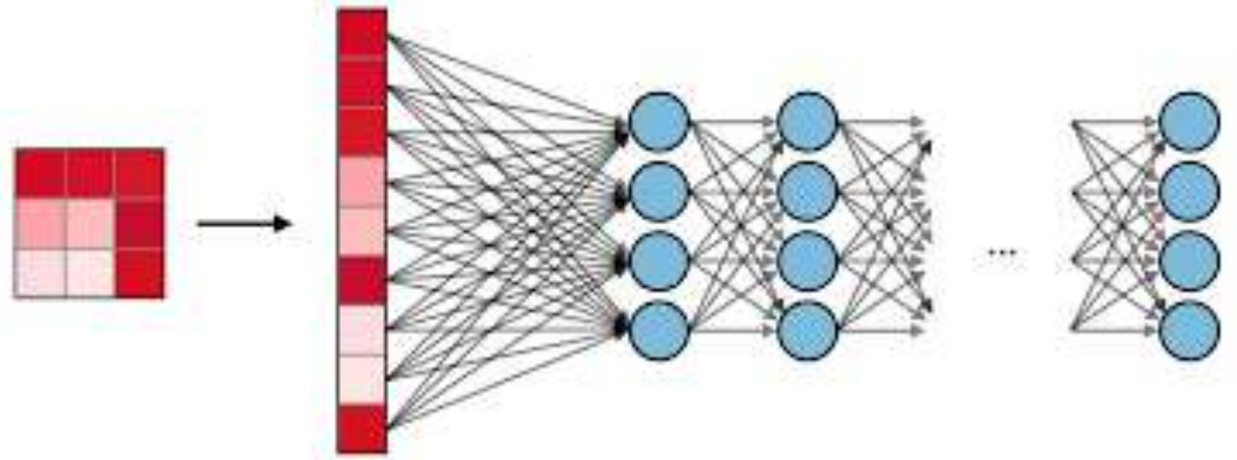
https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9

Capas *Fully Connected*

- **Capas tipo *Densas*** que toman como entrada los valores resultantes de las capas convolucionales

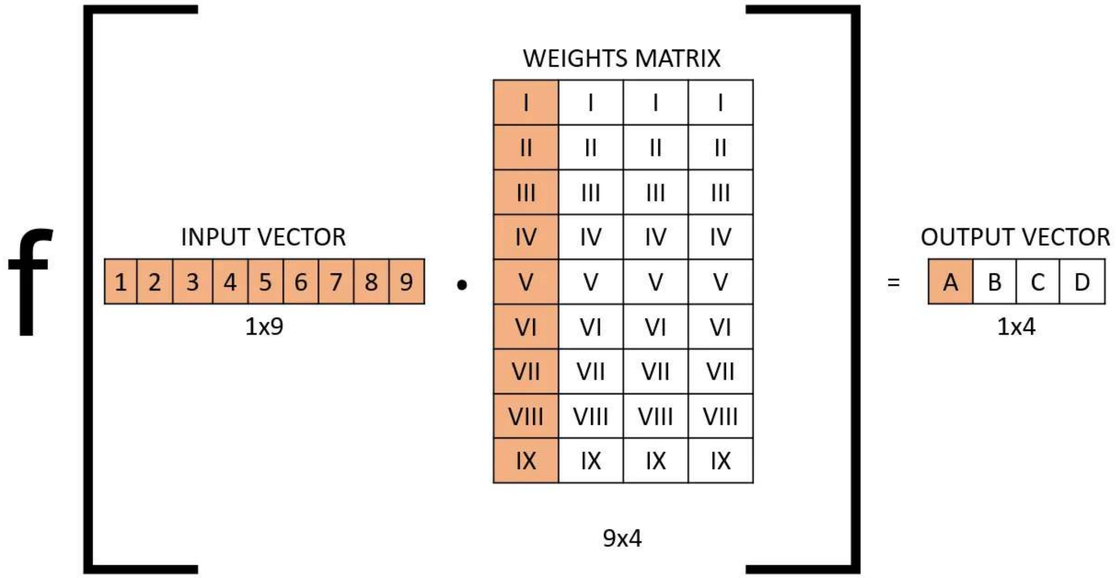
- Recordemos: Las redes neuronales son un conjunto de funciones no lineales dependientes.

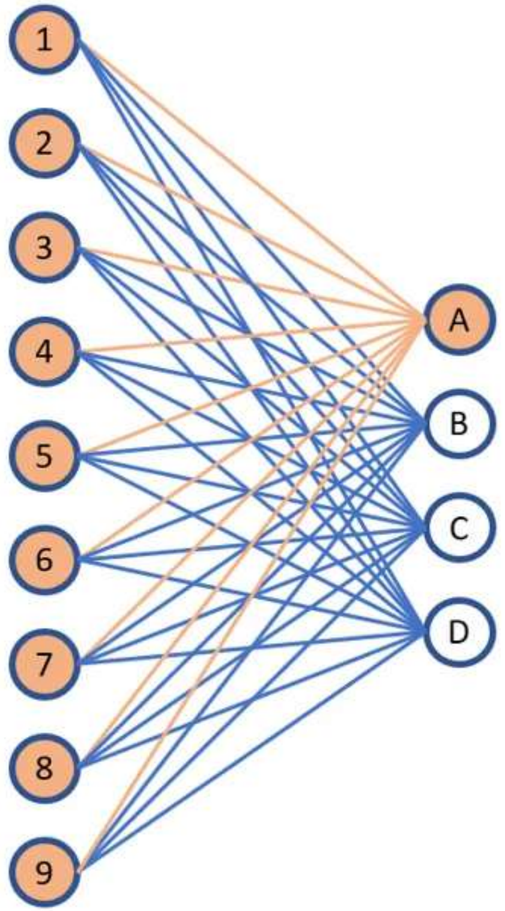
- Cada función individual la neurona aplica una transformación lineal al vector de entrada a través de una matriz de pesos.



https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

Capas FC





Cada activación (o neurona) se calcula

$$a = \phi\left(\sum_{i}^{n} x^T\theta\right)$$

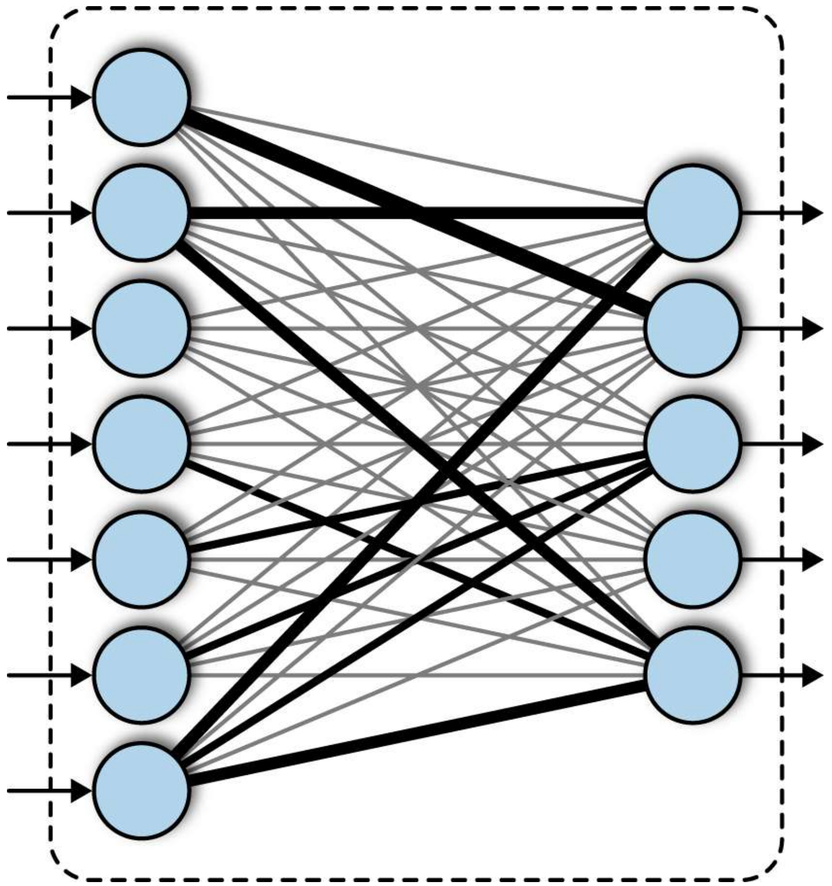
https://towardsdatascience.com/convolutional-layers-vs-fully-connected-layers-364f05ab460b

Capa FC

- Estas capas son llamadas *“Fully Connected”* o *“densely connected”*

- Cada entrada del vector de entrada influye en cada salida del vector de salida. Sin embargo, no todos los pesos afectan a todas las salidas

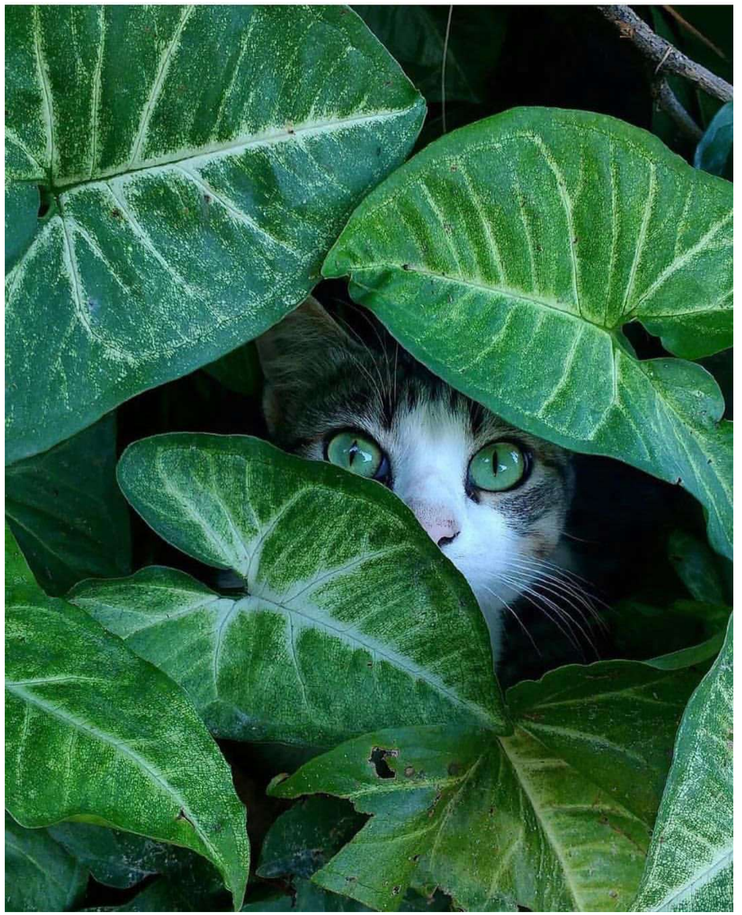
- Los pesos de cada unidad solo afectan el output de esa unidad más no de las demás



2019- Applying Neural Networks for Tire Pressure Monitoring Systems

Dropout

- Las neuronas desarrollan co-dependencias entre sí durante el entrenamiento,
- Frena la potencia individual de cada neurona y conduce a un ajuste excesivo de los datos de entrenamiento.
- Disponemos de varias técnicas de regularización que penalizan los pesos de la red, pero no son suficientes.
- **Dropout es una técnica para prevenir el *Overfitting***



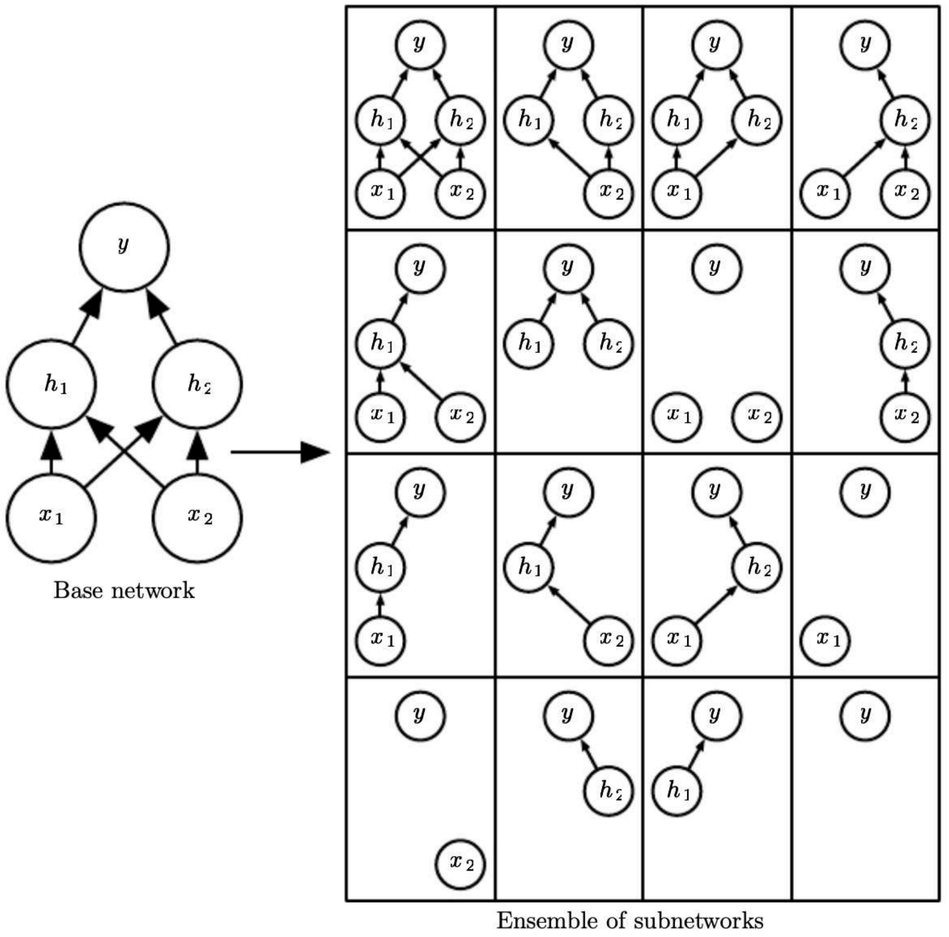
https://pin.it/17zKteW

Dropout

- **Es un método de regularización que ayuda a reducir la dependencia entre las neuronas.**

- El término *dropout* se refiere a la eliminación de los nodos (capa de entrada y oculta) de una red

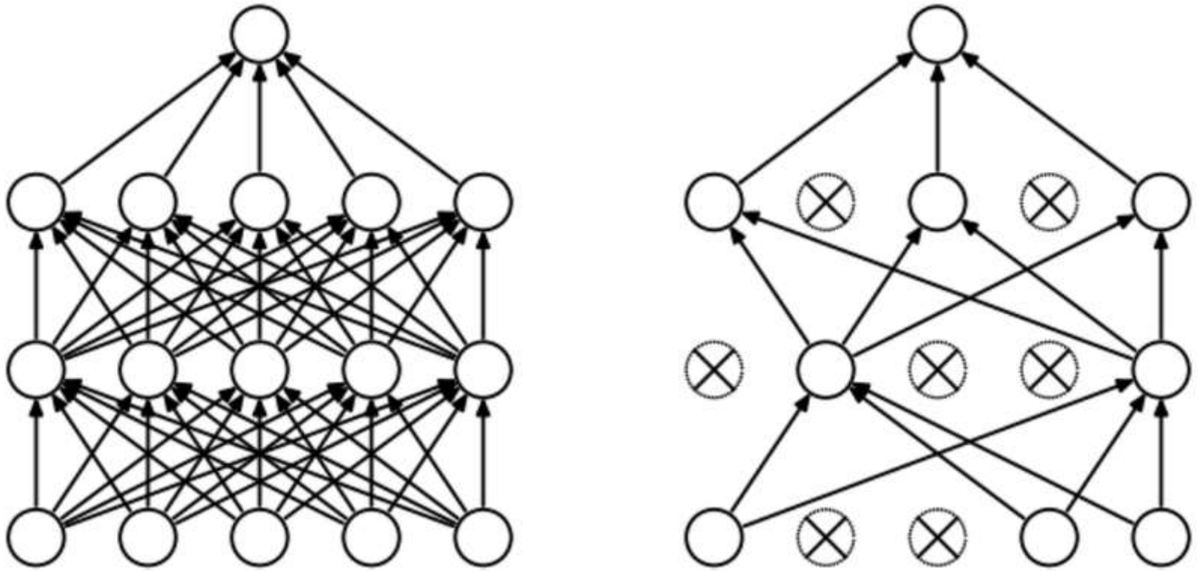
- Entrena el conjunto formado por todas las subredes que pueden formarse eliminando las unidades que no son de salida de una red base subyacente



Deep Learning - Ian Goodfellow
CHAPTER 7. REGULARIZATION FOR DEEP LEARNING

Dropout

- Se refiere a ignorar ciertas *unidades* durante la fase de entrenamiento que se eligen al azar.
- Estas unidades no se tienen en cuenta durante una determinada pasada hacia delante y hacia atrás.
- Los nodos se descartan con una probabilidad de abandono de *p*



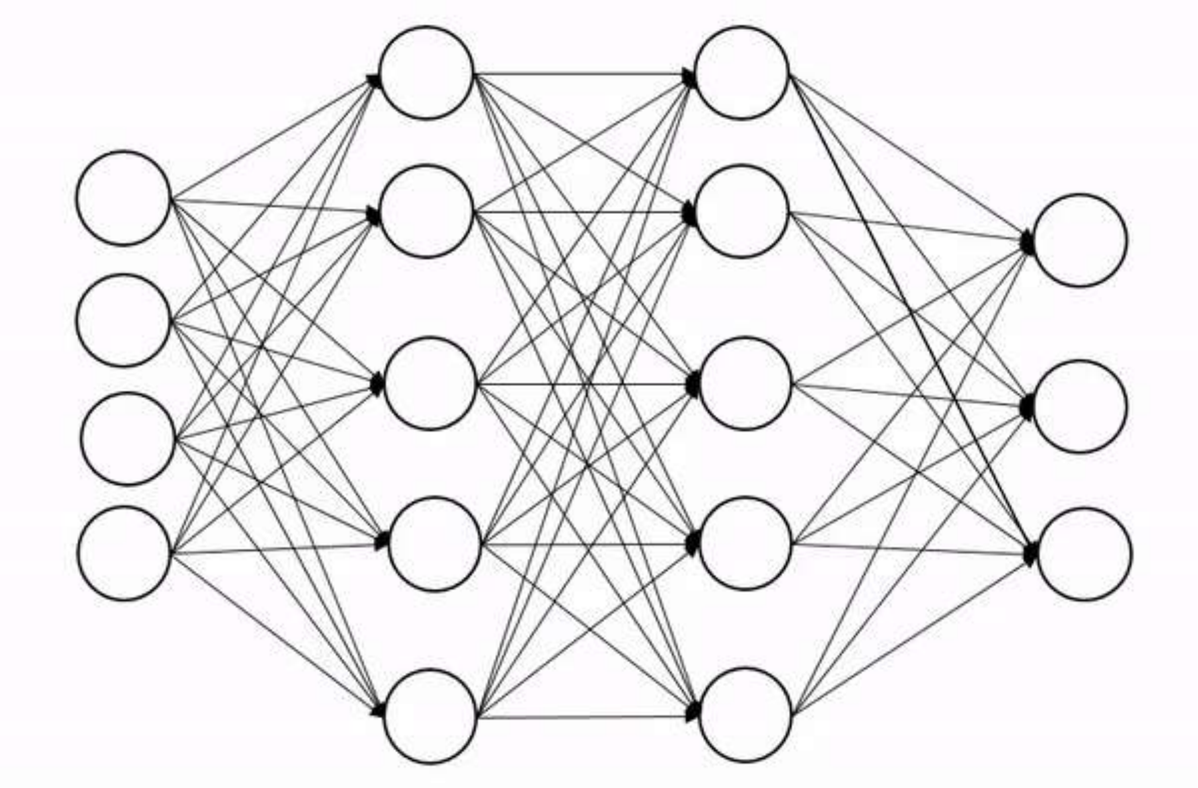
(a) Standard Neural Net

(b) After applying dropout.

2019 - Mutual Information Input Selector and Probabilistic Machine Learning Utilisation for Air Pollution Proxies

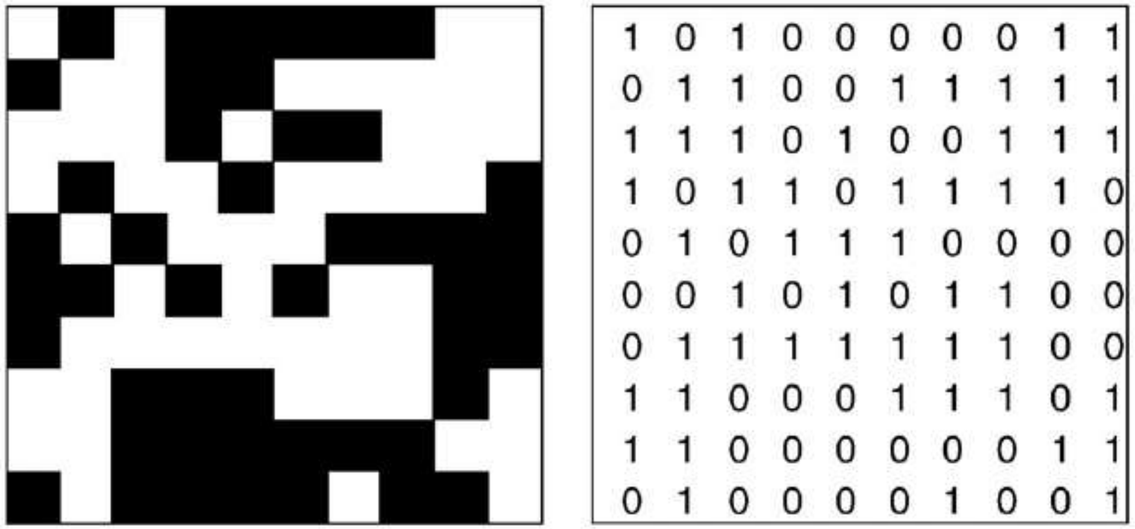
Dropout

- Podemos eliminar una *unidad* de una red multiplicando su valor de salida por cero.
- Cada vez que tomamos un batch: definimos una máscara binaria aleatoria
- **La probabilidad del valor de la máscara es un hiper-parámetro**
- Realizamos *feedforward* y *back propagation* como lo hemos venido haciendo



https://www.analyticsvidhya.com/blog/2022/08/dropout-regularization-in-deep-learning/

Dropout



1	0	1	0	0	0	0	0	1	1
0	1	1	0	0	1	1	1	1	1
1	1	1	0	1	0	0	1	1	1
1	0	1	1	0	1	1	1	1	0
0	1	0	1	1	1	0	0	0	0
0	0	1	0	1	0	1	1	0	0
0	1	1	1	1	1	1	1	0	0
1	1	0	0	0	1	1	1	0	1
1	1	0	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	1

2008 - Analysis of lacunarity and scales of spatial homogeneity in IKONOS images of Amazonian tropical forest canopies.

Notas de implementación

- Dropout pone aleatoriamente 0 las unidades de entrada en cada paso de entrenamiento
- Las entradas que no se ponen a 0 se escalan en 1/(1 - tasa) de forma que la suma de todas las entradas no varíe.

Dropout

Calculo de cada activación en una NN:

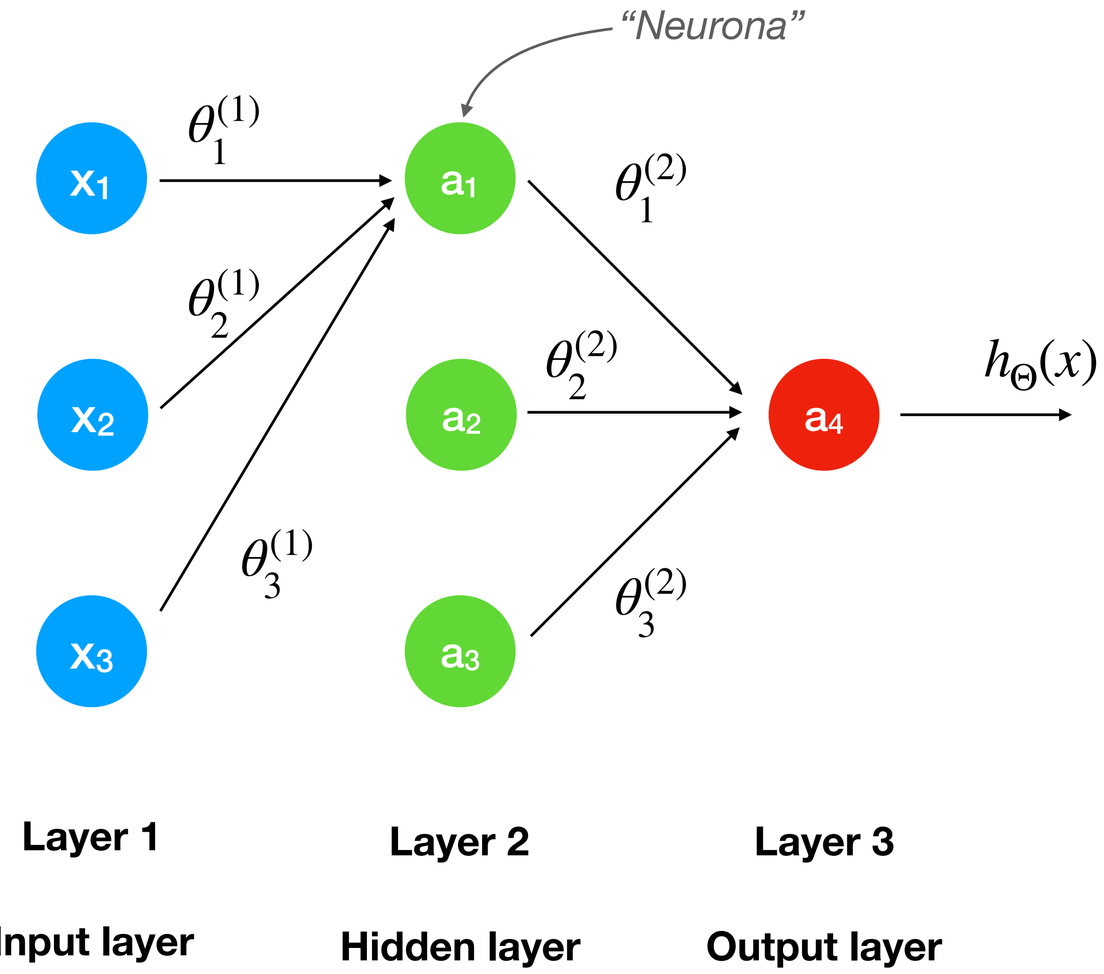
$$z = \theta_1 a1 + \theta_2 a_2 + \theta_3 a_3 \ldots + \theta_n a_n$$

$$a_4 = \phi(z)$$

Con dropout

$$a_4 = \frac{p(c)z}{1 - c}$$

Donde $p(c)$ es la probabilidad con ratio c



Dropout

Calculo de siguiente capa en una NN

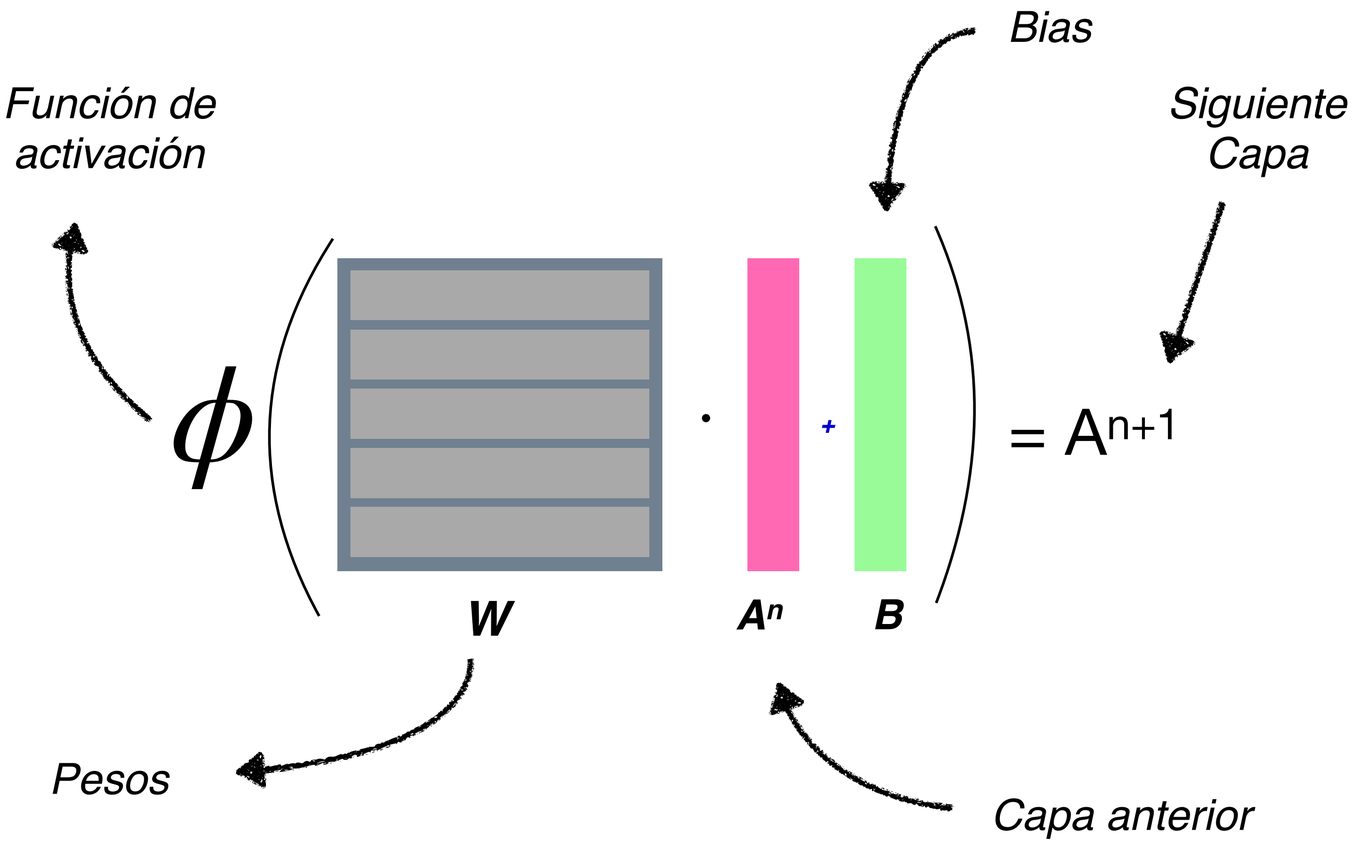
$$Z = W \cdot A_n + B$$

$$A_{n+1} = \phi(Z)$$

Agregando Dropout a la capa

$$A_{n+1} = \frac{\phi(Z) \odot p(c)}{1 - c}$$

$p(c)$ máscara de 0 y 1 generada con un ratio c



Dropout

Consideraciones

- Obliga a una red a aprender **características más robustas** que son útiles en subconjuntos aleatorios diferentes de las otras neuronas.
- Duplica aproximadamente el número de iteraciones necesarias para converger. Sin embargo, el tiempo de entrenamiento para cada época es menor.
- En la fase de consumo del modelo, se considera toda la red y cada activación se reduce en un factor p
- **Valores típicos**
  - Capas ocultas 0.5 drop
  - Capas de entrada 0.2 drop

Diseño de clasificador

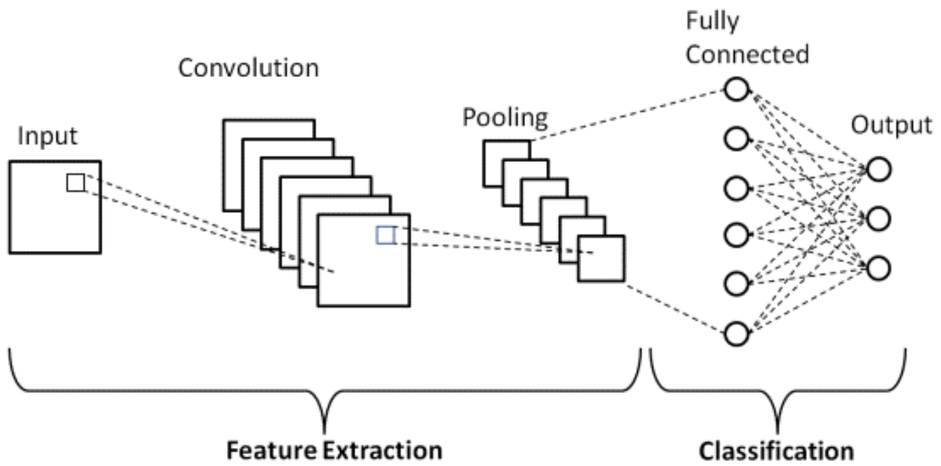
- Diseñar una red neuronal implica elegir muchas características de diseño:
  - Tamaños y número de capas
  - Batch Normalization
  - Dropout
  - Función de activación
- **La arquitectura** de un modelo se refiere a las operaciones internas que realiza para obtener una predicción deseada
- Cada operación está definida como una capa dentro del modelo

https://news.harvard.edu/gazette/story/2020/10/the-role-of-activism-in-architecture/

Diseño de clasificador

Una arquitectura básica de clásica de un modelo para clasificación de imágenes esta compuesta de:

1. Capas convolucionales
2. Normalización en baches
3. Pooling
4. Capas FC

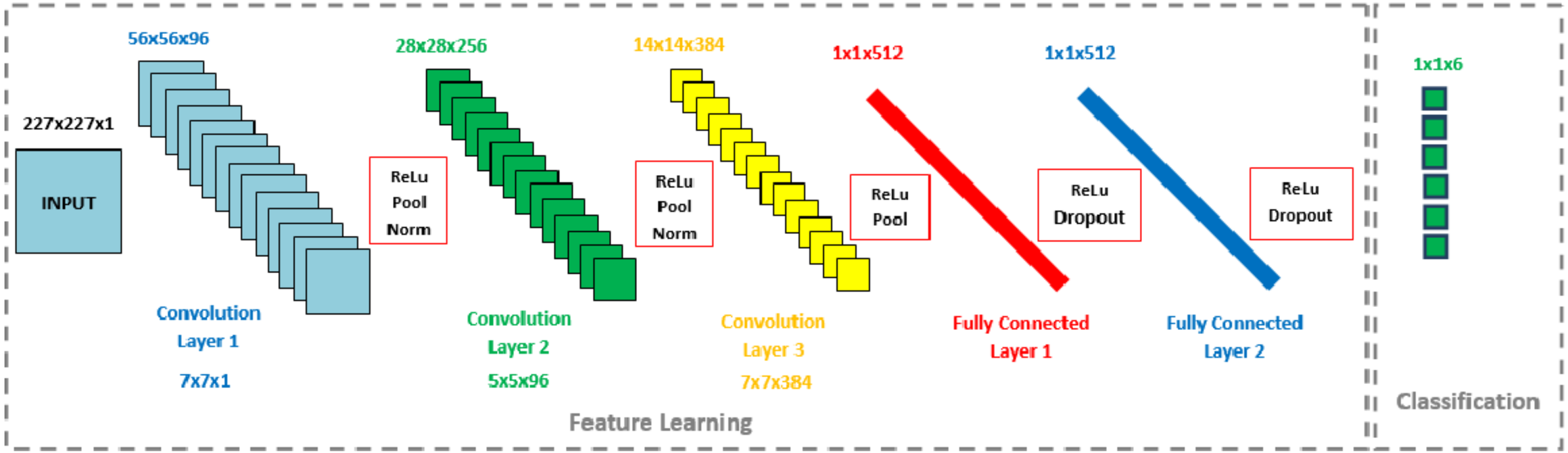


https://www.upgrad.com/blog/basic-cnn-architecture/

Diseño de clasificador

- **Bloques convolucionales**: Extraen características de las imágenes
- **Batch Normalization**: Normalizan las salidas de las capas ocultas por bache
- **Pooling**: reducción dimensional e invariabilidad a la translación
- **Capas FC**: calcular la salida deseada del modelo

Diseño de clasificador



2017 - Comparison of Three Different CNN Architectures for Age Classification

Log de entrenamiento

									Source Dataset				Target Dataset			
	BATCH SIZE	FILE	DATA SETS	TEST SPLIT	EPOCH	LR	IMAGE SIZE	ARCHITECTURE CHANGES	ACC MAT	ACC TYPE	ACC BRAND	ACC AVG	ACC MAT	ACC TYPE	ACC BRAND	ACC. AVG
T0	50	multi_label0	D0	T0	5	0,0003	224x224		0,698	0,677	0,010	0,462				0,000
T1	100	multi_label1	D1	T0	10	0,0003	224x224		0,683	0,779	0,101	0,521				0,000
T2	200	multi_label1	D1	T0	10	0,0003	224x224		0,596	0,630	0,082	0,436				0,000
T3	100	multi_label1	D1	T1	10	0,001	224x224		0,846	0,822	0,236	0,635	0,644	0,678	0,055	0,459
T4	200	multi_label1	D1	T1	15	0,001	224x224		0,837	0,841	0,197	0,625	0,610	0,527	0,027	0,388
T5	100	multi_label1	D1	T1	10	0,001	224x224	~~Removed intermedium dense layer~~	0,793	0,817	0,173	0,595	0,507	0,486	0,034	0,342
T6	100	multi_label1	D1	T0	10	0,001	224x224	~~Increase intermedium dense units to 500~~	0,846	0,817	0,327	0,663				0,000
T7	100	multi_label1	D1	T0	10	0,001	224x224	~~Increase intermedium dense units to 500~~ ~~Added dropout~~	0,841	0,817	0,313	0,657				0,000

Log de entrenamiento para exploración inicial de modelos de ML

Ejercicio

- En google Colab realizar un modelo CNN para la clasificación de imágenes del dataset CIFAR-10 (https://www.tensorflow.org/tutorials/images/cnn)
- Realiza cambio a los hiper-parámetros o a la arquitectura para lograr una mayor precisión en el dataset de pruebas
- Llevar un “log” en Excel de los cambio realizados vs el resultado del modelo