

Algunas Arquitecturas de Redes Neuronales Convolucionales

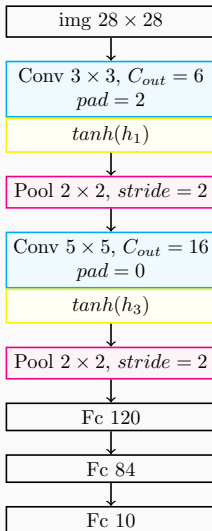
Rafael Villca Poggian

Club de Ciencia de Datos

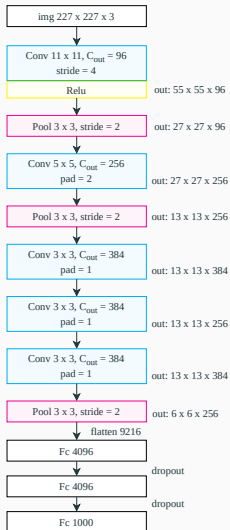
1. Dos ejemplos Iniciales
2. Hacia Arquitecturas más Profundas
3. Batch Normalization
4. Residual Layers
5. Inception Layers

Dos ejemplos Iniciales

Propuesta en 1998 por Yann LeCun [1]



Propuesta en 2012, se usa por primera vez Relu y Dropout [2]



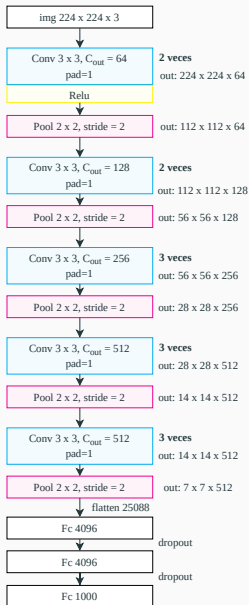
Recordando el diagrama detallado de una CNN



Hacia Arquitecturas más Profundas

- Se presenta a la competencia ImageNet del año 2014 [3]
- Obtiene una exactitud del 92.7% top 5
- Propone una mejora sobre la AlexNet al reducir el campo receptivo a filtros de 3×3

VGG16



Batch Normalization

- Se propuso como un método para reducir el desvanecimiento y explosión de los gradientes
- Empíricamente se observa que los gradientes tienen menor magnitud y el costo cambia menos
- Por esta razón se teoriza que el método suaviza la función de pérdida

Definición

- Consiste en normalizar cada canal del feature map para preservar la información compartida [4]
- Se llama *feature map* a $H_{(l,i)} = \mathcal{H}_{l-1} \star W_i + b_i$ (para el canal de salida i) al cual se aplica la normalización
- Calculando la media y varianza de un lote \mathcal{B}

$$\mu_{\mathcal{B},i} = \frac{1}{m} \sum_{j=1}^m H_{(l,i)_j} \qquad \sigma_{\mathcal{B},i}^2 = \frac{1}{m} \sum_{j=1}^n \left(H_{(l,i)_j} - \mu_{\mathcal{B}} \right)^2$$

- Normalizando con ε para mantener la estabilidad numérica

$$\widehat{H}_{(l,i)} = \frac{H_{(l,i)} - \mu_{\mathcal{B},i}}{\sqrt{\sigma_{\mathcal{B},i}^2 + \varepsilon}}$$

Definición

- Al normalizar se modifica lo que cada capa representa y puede inutilizar la no linealidad
- Se añaden parámetros γ y β luego de la normalización que en caso de ser innecesaria, permiten a la red escoger los valores correctos para deshacerla o atenuarla
- Así la salida de la capa de BatchNorm es

$$y_i = \gamma_i \widehat{H_{(l,i)}} + \beta_i$$

- El efecto del sesgo b se cancela al restar la media, por lo que se puede simplificar la operación

$$\mathcal{H}_l = g(\text{BatchNorm}(\mathcal{H}_{l-1} \star W_i))$$

Residual Layers

Skip Connections

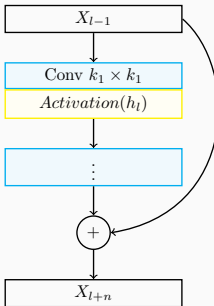
- Recordemos que BatchNorm se propuso para controlar el desvanecimiento y explosión de los gradientes
- Sin embargo con redes profundas con varias capas BatchNorm en realidad agrava estos efectos
- Para solucionarlo se proponen las conexiones con saltos
- Estas conexiones permiten al modelo decidir si tomar el camino complejo o un mapeo lineal
- Se descompone el mapeo esperado $\mathcal{H}(\mathbf{x})$ en un bloque de convoluciones no lineal $\mathcal{F}(\mathbf{x})$ y otro lineal \mathbf{x}

$$\mathcal{F}(\mathbf{x}) = \mathcal{H}(\mathbf{x}) - \mathbf{x}$$

- Así el objetivo será $\mathcal{H}(\mathbf{x}) = \mathcal{F}(\mathbf{x}) + \mathbf{x}$

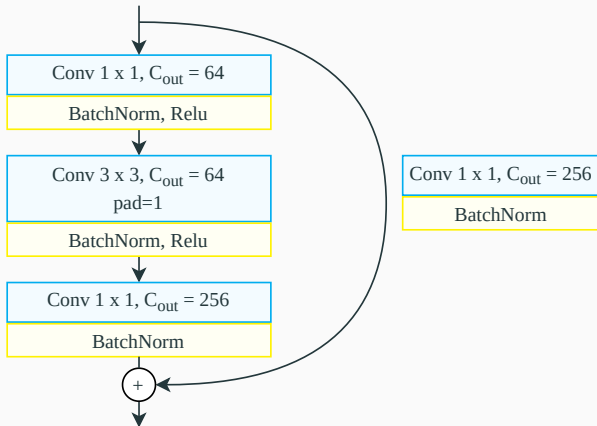
Residual Block

Denominados Bloques Residuales, propuestos en 2015 [5]

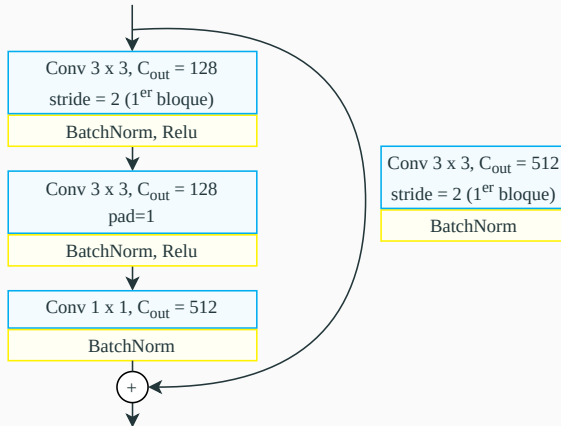


Estos componen la ResNet

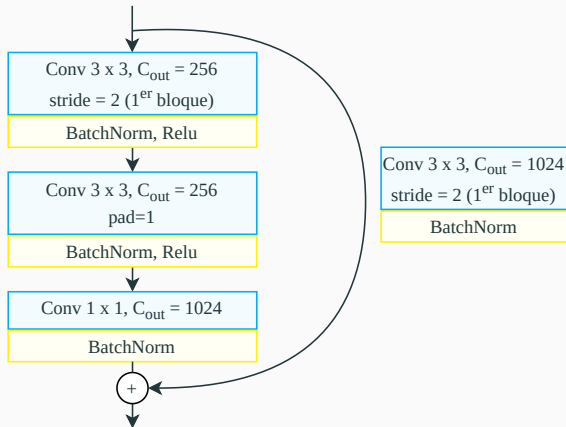
Bloque Residual 1



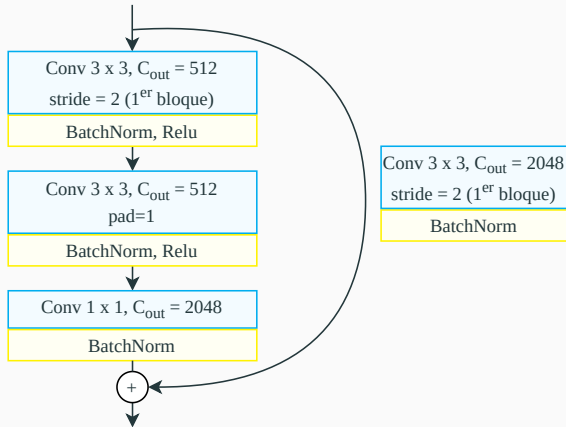
Bloque Residual 2



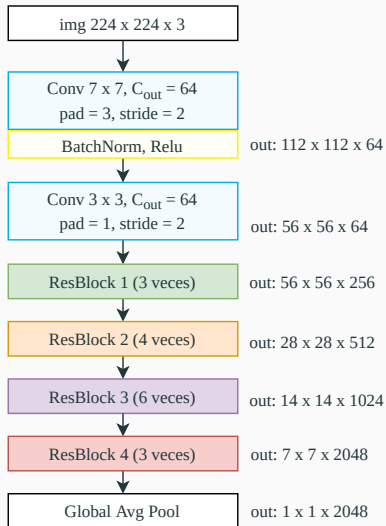
Bloque Residual 3



Bloque Residual 4



ResNet50






Inception Layers




Inception Blocks

- Se aplican reducciones mediante convoluciones 1×1 antes de aplicar nuevamente filtros de mayor dimensión
- Sigue la idea de que la información visual se procesa a distintas escalas en paralelo para luego combinarse en una siguiente etapa de abstracción
- Se busca reducir la dependencia de capas anteriores para poder extraer información que tal vez se pierde hasta llegar a una capa específica

- Se propuso durante la competencia ImageNet del 2014 [6]
- Cada capa es en realidad un inception block
- Se agregan capas de salida en medio de la red para retropropagar errores de subredes
- Esto con el fin de enfatizar el poder discriminativo de las capas iniciales
- Debido a la profundidad de la red, estos sub clasificadores permiten mantener una mejor señal de los gradientes



-  Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
-  Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012.
-  Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

-  Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: ICML’15. Lille, France: JMLR.org, 2015, pp. 448–456.
-  K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.
-  C. Szegedy et al. “Going deeper with convolutions”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9.