Funciones de pooling basadas en Penalty aplicadas a la reducción de características en Redes Neuronales Convolucionales

Iosu Rodriguez-Martinez[1] Pablo Aitor Lizarraga-Guerra[1]
Tiago da Cruz Asmus[1, 2] Graçaliz Pereira Dimuro[1, 2]
Angela Bernardini[1] Francisco Herrera[3] Humberto Bustince[1]

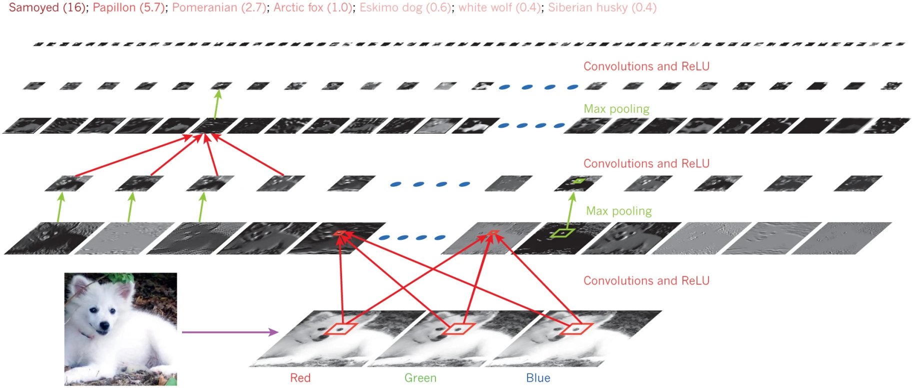
[1]Universidad Pública de Navarra, Pamplona, Spain

[2]Universidade Federal do Rio Grande, Rio Grande, Brasil

[3]Universidad de Granada, Granada, Spain

XXI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF22)
5 de septiembre, 2022. Toledo, España

Redes Neuronales Convolucionales: CNN



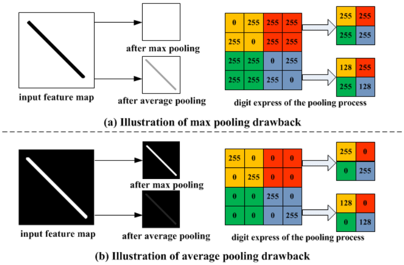
Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature, 521 (7553) (2015), 436-444.

Pooling

Las funciones de Pooling agregan las características extraídas por las capas convolucionales:

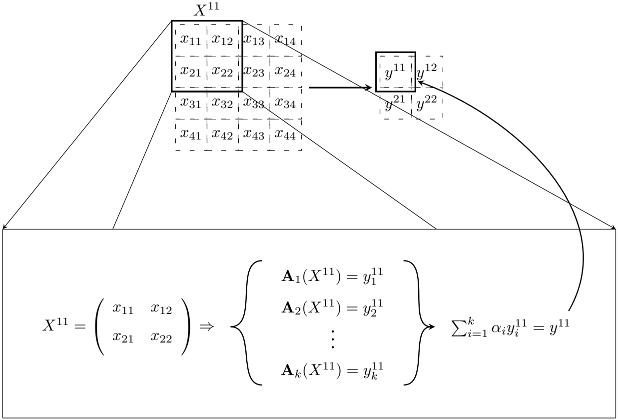
$\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$;
$\mathbf{A}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$

- La información de interés se resume localmente.
- Pooling max: $\mathbf{A}(\mathbf{x}) = \max_{i=1}^{n} \mathbf{x}_i$
- Pooling media: $\mathbf{A}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$



D. Yu, H. Wang, P. Chen, Z. Wei, *Mixed pooling for convolutional neural networks*. In International conference on rough sets and knowledge technology (2014) (pp. 364-375). Springer, Cham.

Capas CombPool



I. Rodriguez-Martinez, J. Lafuente, R. H. Santiago, G. P. Dimuro, F. Herrera, H. Bustince *Replacing pooling functions in Convolutional Neural Networks by linear combinations of increasing functions*. Neural Networks, 152 (2022), 380-393.

Motivación y objetivos

Objetivo: Extender las estrategias de combinación de funciones pooling.

- Podemos escoger la mejor reducción de características
  - Agregaciones basadas en funciones penalty.

Motivación y objetivos

Objetivo: Extender las estrategias de combinación de funciones pooling.

- Podemos escoger la mejor reducción de características
  - Agregaciones basadas en funciones penalty.
- O ponderar su contribución en base a su calidad
  - Agregaciones basadas en funciones penalty **suaves**.

Funciones de agregación

Definición

Una función $A : [0, 1]^n \to [0, 1]$ es una función de agregación si cumple lo siguiente

- A es creciente;
- $A(0, \ldots, 0) = 0$;
- $A(1, \ldots, 1) = 1$;

Definición

Una función de agregación se denomina promedio si está acotada por el mínimo y el máximo de sus argumentos

$$\min(\mathbf{x}) \leq A(\mathbf{x}) \leq \max(\mathbf{x})$$

Integral de Sugeno

A partir de este momento, asumimos que $2 \leq n \in \mathbb{N}$, $1 \leq r \in \mathbb{N}$.

Notación

Denotamos mediante $\mathbf{x}_\sigma = (x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ una permutación de $\mathbf{x}$.
Si $x_{\sigma(1)} \leq \cdots \leq x_{\sigma(n)}$, decimos que $\mathbf{x}_\sigma = \mathbf{x}_{(\nearrow)}$.

Integral de Sugeno

A partir de este momento, asumimos que $2 \leq n \in \mathbb{N}$, $1 \leq r \in \mathbb{N}$.

Notación

Denotamos mediante $\mathbf{x}_\sigma = (x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ una permutación de $\mathbf{x}$.
Si $x_{\sigma(1)} \leq \cdots \leq x_{\sigma(n)}$, decimos que $\mathbf{x}_\sigma = \mathbf{x}_{(\nearrow)}$.

Definición

La integral de Sugeno asociada a la medida difusa ν es la función $\mathbf{S}_\nu : \mathbb{R}^n \to \mathbb{R}$ dada por, para $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$
$\mathbf{S}_\nu(\mathbf{x}) = \max(\min(x_{\sigma(1)}, \nu(\sigma(1), \ldots, \sigma(n)), \ldots, \min(x_{\sigma(n)}, \nu(\sigma(n))),$
donde $\mathbf{x}_\sigma = \mathbf{x}_{(\nearrow)}$

Integral de Sugeno generalizada[1]

$$\mathbf{A}(\mathbf{x}) = \mathrm{G}(\mathrm{F}(x_{\sigma(1)}, \nu(\sigma(1), \dots, \sigma(n)), \dots, \mathrm{F}(x_{\sigma(n)}, \nu(\sigma(n)))$$

,

En nuestro caso, fijamos:

- $G(\mathbf{x}) = \sum_{i=1}^{n} x_i$
- $F(x, y) = xy$

Y nos referimos a esta versión de la integral de Sugeno mediante:

$$\mathbf{D}_\nu(\mathbf{x}) = x_{\sigma(1)}\nu(\sigma(1), \dots, \sigma(n)) + \cdots + x_{\sigma(n)}\nu(\sigma(n))$$

,

[1] F. Bardozzo, B. De La Osa, L. Horanská, J. Fumanal-Idocin, M. delli Priscoli, L. Troiano, R. Tagliaferri, J. Fernandez, H. Bustince, *Sugeno integral generalization applied to improve adaptive image binarization.* Information Fusion, 68 (2021), 37-45

Funciones penalty[2,3]

Definición

Una función $P : [a, b]^{n+1} \to \mathbb{R}^+$ es una función penalty si satisface lo siguiente:

1. $P(\mathbf{x}, y) \geq 0$ para todo $\mathbf{x}, y$;
2. $P(\mathbf{x}, y) = 0$ si todo $x_i = y$;
3. $P(\mathbf{x}, y)$ es quasi-convexa en y para cualquier $\mathbf{x}$, esto es $P(x, \lambda y_1 + (1-\lambda)y_2) \leq \max(P(x, y_1), P(x, y_2))$, para todo $\lambda \in [0, 1]$ y todo y_1, y_2 en su dominio.

[2] T. Calvo, G. Beliakov, *Aggregation functions based on penalties*. Fuzzy sets and Systems, 161(10) (2010), 1420-1436

[3] H. Bustince, G. Beliakov, G. P. Dimuro, B. Bedregal, R. Mesiar. *On the definition of penalty functions in data aggregation*. Fuzzy Sets and Systems (2017), 323, 1-18.

Funciones basadas en penalty

Definición

Dada una función penalty P, la función basada en penalty f es

$$f(\mathbf{x}) = \text{arg} \min_y P(\mathbf{x}, y)$$

si y es el único minimizador, y $y = \frac{a+b}{2}$ si el conjunto de minimizadores es el intervalo [a, b].

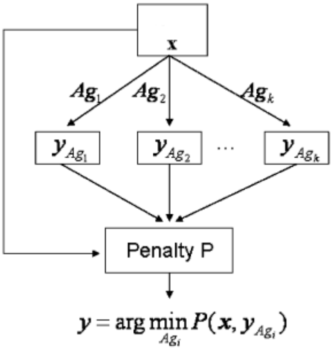
NOTA: Cualquier función de agregación promedio puede ser representada como una función basada en penalty:

- Mediana: $\text{arg} \min_y \sum_{i=1}^{n} |x_i - y|$
- Media aritmética: $\text{arg} \min_y \sum_{i=1}^{n} (x_i - y)^2$

Funciones de agregación basadas en una función penalty[4]

Dadas k funciones de agregación promedio $A_i : [0,1]^n \to [0,1]$, una función penalty $P : [a,b]^{n+1} \to \mathbb{R}^+$ y $\mathbf{x} \in [0,1]^n$:

- Agregamos $\mathbf{x}$ mediante cada A_i obteniendo k reducciones y_i.
- Escogemos como mejor reducción la que minimiza la función penalty P.

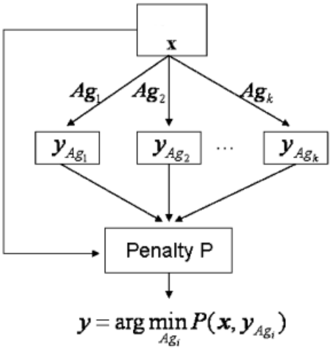


[4]D. Paternain, A. Jurio, G. Beliakov. *Color image reduction by minimizing penalty functions*. In 2012 IEEE International Conference on Fuzzy Systems (2012) (pp. 1-7). IEEE.

Funciones de agregación basadas en una función penalty[4]

Dadas k funciones de agregación promedio $A_i : [0,1]^n \to [0,1]$, una función penalty $P : [a,b]^{n+1} \to \mathbb{R}^+$ y $\mathbf{x} \in [0,1]^n$:

- Agregamos $\mathbf{x}$ mediante cada A_i obteniendo k reducciones y_i.
- Escogemos como mejor reducción la que minimiza la función penalty P.
- Podríamos considerar el resto de reducciones
    - Ponderándolas en base a P



[4] D. Paternain, A. Jurio, G. Beliakov. *Color image reduction by minimizing penalty functions*. In 2012 IEEE International Conference on Fuzzy Systems (2012) (pp. 1-7). IEEE.

Funciones de agregación basadas en penalty “suave” (I)

Dadas k funciones de agregación promedio, y una función penalty P. Queremos definir $F : [0,1]^n \to [0,1]$ como una combinación convexa

$$F(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i A_i(\mathbf{x})$$

donde:

- α_i sea inversamente proporcional a $P(\mathbf{x}, A_i(\mathbf{x}))$
- $0 \leq \alpha_i \leq 1$ y $\sum_{i=1}^{k} \alpha_i = 1$

Funciones de agregación basadas en penalty “suave” (I)

Dadas k funciones de agregación promedio, y una función penalty P. Queremos definir $F : [0,1]^n \to [0,1]$ como una combinación convexa

$$F(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i A_i(\mathbf{x})$$

donde:

- α_i sea inversamente proporcional a $P(\mathbf{x}, A_i(\mathbf{x}))$
- $0 \leq \alpha_i \leq 1$ y $\sum_{i=1}^{k} \alpha_i = 1$

NOTA: $F(\mathbf{x}) = \arg\min P(\mathbf{x}, A_i(\mathbf{x}))$ es una combinación convexa donde $\alpha_i = 1$ si $A_i(\mathbf{x}) \geq A_j(\mathbf{x})$ para todo $j \neq i$ y $\alpha_i = 0$ en caso contrario.

Funciones de agregación basadas en penalty "suave" (II)

- Queremos que α_i sea inversamente proporcional a $P(\mathbf{x}, A_i(\mathbf{x}))$
  - Necesitamos una función decreciente $f : \mathbb{R}^+ \to \mathbb{R}^+$
  - p. ej. $f(x) = \frac{\lambda}{x}$, con $\lambda \in \mathbb{R}^+$

Funciones de agregación basadas en penalty “suave” (II)

- Queremos que α_i sea inversamente proporcional a $P(\mathbf{x}, A_i(\mathbf{x}))$
  - Necesitamos una función decreciente $f : \mathbb{R}^+ \to \mathbb{R}^+$
  - p. ej. $f(x) = \frac{\lambda}{x}$, con $\lambda \in \mathbb{R}^+$
- Para que F sea una combinación convexa $0 \leq \alpha_i \leq 1$ y $\sum_{i=1}^k \alpha_i = 1$
  - Necesitamos normalizar los coeficientes resultantes
  - p. ej. $s : \mathbb{R}^k \to [0,1]^k$ la función softmax dada por $s(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}}$

Funciones de agregación basadas en penalty "suave" (III)

Dadas k funciones de agregación promedio $A_i : [0,1]^n \to [0,1]$, k parámetros $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k) \in (\mathbb{R}^+)^k$ y una función penalty $P : [a,b]^{n+1} \to \mathbb{R}^+$

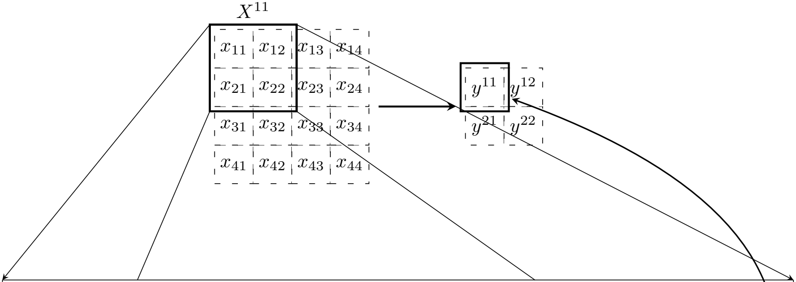
La función de agregación basada en la función penalty "suave" P es:

$$F(\mathbf{x}) = \sum_{i=1}^{k} s\left(\frac{\boldsymbol{\lambda}}{\mathbf{p}}\right)_i A_i(\mathbf{x}) = \sum_{i=1}^{k} \frac{e^{\frac{\lambda_i}{P(\mathbf{x}, A_i(\mathbf{x}))}}}{\sum_{j=1}^{k} e^{\frac{\lambda_j}{P(\mathbf{x}, A_j(\mathbf{x}))}}} A_i(\mathbf{x})$$

donde:

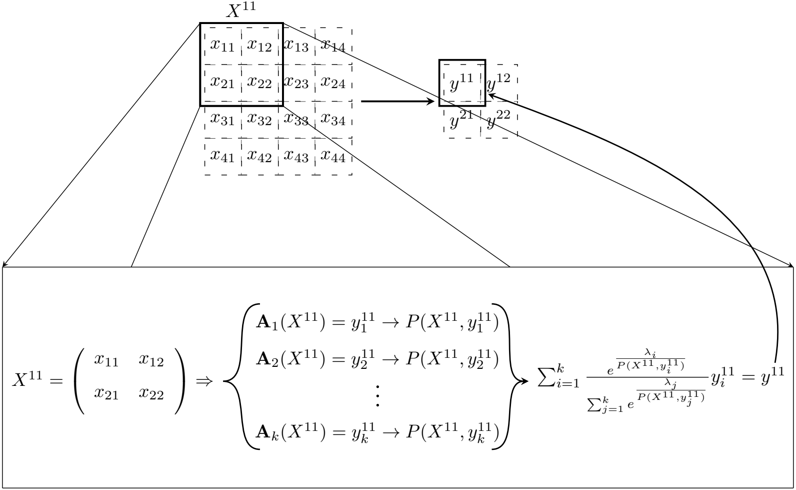
- $\mathbf{p} = (p_1, \ldots, p_k) = (P(\mathbf{x}, A_1(\mathbf{x})), \ldots, P(\mathbf{x}, A_k(\mathbf{x})))$

Pooling basado en penalty

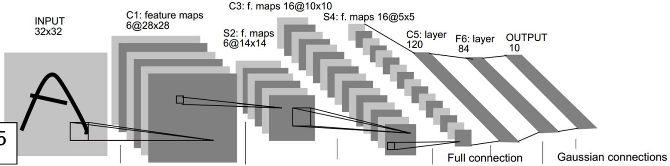


$$X^{11} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \Rightarrow \left\{ \begin{array}{c} \mathbf{A}_1(X^{11}) = y_1^{11} \to P(X^{11}, y_1^{11}) \\ \mathbf{A}_2(X^{11}) = y_2^{11} \to P(X^{11}, y_2^{11}) \\ \vdots \\ \mathbf{A}_k(X^{11}) = y_k^{11} \to P(X^{11}, y_k^{11}) \end{array} \right\} \arg \min_{y_i^{11}} P(X^{11}, y_i^{11}) = y^{11}$$

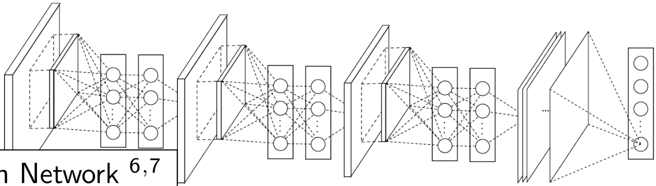
Pooling basado en penalty suave



Arquitecturas de Deep Learning



LeNet-5 [5]



Network in Network [6,7]

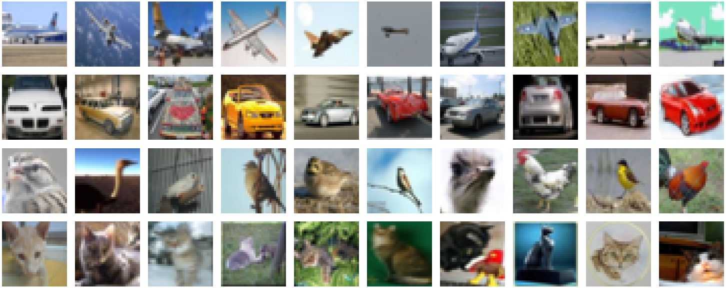
[5] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 86(11) (1998), 2278-2324.

[6] M. Lin, Q. Chen, S. Yan, *Network in network*. arXiv preprint arXiv:1312.4400 (2013)

[7] C. Y. Lee, P. W. Gallagher, Z. Tu, *Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree*. In Artificial intelligence and statistics (pp. 464-472) (2016). PMLR.

Dataset: CIFAR-10[8]

- Imágenes a color del mundo real $(32 \times 32$ pixels$)$
- 50000 imágenes de entrenamiento
- 10000 imágenes de test
- 10 clases



[8] A. Krizhevsky, G. Hinton, *Learning Multiple Layers of Features from Tiny Images* (2009)

Capas de pooling (I)

Capas Combpool:

$$F(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i A_i(\mathbf{x})$$

Funciones de agregación usadas:

- Máximo
- Media
- Mediana
- $\mathbf{S}_\nu$ con $\nu = |X|^2/n$
- $\mathbf{D}_\nu$ con $\nu = |X|^2/n$

Capas de pooling (II)

Capas PenaltyPool:

$$F(\mathbf{x}) = \text{arg} \min_{A_i(\mathbf{x})} P(\mathbf{x}, A_i(\mathbf{x}))$$

Probamos las siguientes funciones penalty P:

- $P_1(\mathbf{x}, y) = \sum_{i=1}^{n} |x_i - y|$
- $P_2(\mathbf{x}, y) = \sum_{i=1}^{n} (x_i - y)^2$
- $P_3(\mathbf{x}, y) = \sum_{i=1}^{n} |x_i - y|^3$

Capas de pooling (III)

Capas PenaltyCombPool:

$$F(\mathbf{x}) = \sum_{i=1}^{k} s\left(\frac{\boldsymbol{\lambda}}{\mathbf{p}}\right)_i A_i(\mathbf{x})$$

Con

- $\mathbf{p} = (p_1, \ldots, p_k) \in (\mathbb{R}^+)^k$, donde $p_i = P(\mathbf{x}, A_i(\mathbf{x}))$
- $s : \mathbb{R}^k \to [0,1]^k$ la función softmax dada por $s(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$
- $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k) \in (\mathbb{R}^+)^k$, parámetros adicionales del modelo.

Resultados

Nombre	Modelo 1 (acc)	Modelo 2 (acc)
Máximo	0.7439	0.8632
Media	0.7442	0.8602
CombPool	0.7453	0.8675
PenaltyPool (P_1)	**0.7558**	0.8492
PenaltyPool (P_2)	**0.7544**	0.8627
PenaltyPool (P_3)	0.7506	0.8633
PenaltyCombPool (P_1)	**0.7506**	**0.8732**
PenaltyCombPool (P_2)	**0.7559**	**0.8734**
PenaltyCombPool (P_3)	**0.7579**	**0.8749**

Conclusiones y trabajo futuro

Conclusiones

- Otras estrategias para combinar diferentes funciones de pooling pueden superar los resultados obtenidos por capas CombPool.
- Ponderar la contribución de cada función de agregación según una función Penalty es una estrategia eficaz.

Conclusiones y trabajo futuro

Conclusiones

- Otras estrategias para combinar diferentes funciones de pooling pueden superar los resultados obtenidos por capas CombPool.
- Ponderar la contribución de cada función de agregación según una función Penalty es una estrategia eficaz.

Trabajo futuro

- Buscar otros métodos de construcción de capas PenaltyCombPool y probar otras funciones
- Reemplazar los procesos de fusión de información de otras capas de las CNN (GAP, fusión de cardinalidad...)

Gracias por vuestra atención

Alguna pregunta?

iosu.rodriguez@unavarra.es
Departamento de Estadística, Informática y Matemáticas (UPNA)
Grupo de Inteligencia Artificial y Razonamiento Aproximado (GIARA)

Agradecimientos Grant PID2019-108392GB-I00/ funded by MCIN/AEI/10.13039/501100011033. T. Asmus and G. Dimuro are partially supported by the projects CNPq (301618/2019-4), FAPERGS (19/2551-0001279-9). F. Herrera is supported by the Andalusian Excellence project P18-FR4961. I. Rodriguez-Martinez is partially supported by the projects PC095-096 FUSIPROD, Tracasa Instrumental (iTRACASA) and Gobierno de Navarra - Departamento de Universidad, Innovación y Transformación Digital.