Aprendizaje multiclase de videoimágenes deportivas con arquitecturas profundas

Resumen Las arquitecturas profundas permiten representar de manera compacta funciones altamente no lineales. Entre ellas, las redes convolucionales han adquirido gran protagonismo en la clasificación de imágenes debido a la invarianza traslacional de sus features. Este trabajo propone investigar un abordaje naïve para la clasificación de videoimágenes con redes profundas, comparar la performance de redes pre-entrenadas con la de redes ad-hoc y finalmente crear un mecanismo de visualización de la representación interna de la arquitectura. Como ejemplo de aplicación se utilizarán segmentos de videos deportivos con diferentes acciones grupales.

Keywords: aprendizaje profundo, redes convolucionales, video, deportes

1. Introducción

La mayoría de los métodos de análisis de datos se basan en lo que puede definirse como arquitecturas poco profundas (Redes Neuronales con una capa oculta, SVM, Árboles de Decisión, etc.), aunque desde hace bastante tiempo se sabe que las arquitecturas profundas pueden ser mucho más eficientes a la hora de representar ciertas funciones. Esto se debe, probablemente, a que solo recientemente se pudo desarrollar un método de aprendizaje efectivo para arquitecturas profundas [2]. Estas redes profundas han mostrado gran efectividad resolviendo problemas sobre todo del área de *Machine Vision*: reconocimiento de imágenes [2, 1], secuencias de video [3] y datos de captura de movimiento [5]. En particular, resulta de especial interés para este trabajo el desarrollo de redes convolucionales: redes profundas con conexiones locales y pesos compartidos que pueden ser utilizadas para la extracción de features robustas a variaciones traslacionales [6, 7]. Con estos modelos se abrió toda una nueva área de estudio dentro de Aprendizaje Automatizado, conocida como *Deep Learning*.

Muchos de los trabajos actuales que usan estas técnicas para reconocimiento de imágenes parten de un modelo pre-entrenado con un conjunto de datos grande (generalmente *ImageNet*) para luego adaptarlo al problema particular que se deba atacar [11, 8]. Esto se debe a dos razones: por un lado, para capturar conceptos que puedan ser importantes para la tarea puede ser necesario un modelo con gran profundidad, y por otro lado, para entrenar un modelo de estas características se necesita una gran cantidad de datos, que generalmente no se tienen, y mucho tiempo de cómputo.

El objetivo de este trabajo es investigar distintos modelos de *Deep Learning*, aplicados particularmente al problema de clasificación de acciones deportivas

en videos de rugby. Este problema resulta de vital importancia para el cuerpo técnico, dado que actualmente deben clasificar estas acciones manualmente para luego utilizarlas durante los entrenamientos tácticos en pos de corregir defectos de una formación en particular. De esta manera, el entrenador puede, por ejemplo, mostrar todos los lines, uno tras otro, sin tener que revisar todo el video del partido.

2. Redes convolucionales

Una red convolucional es una red neuronal con pesos compartidos y conexiones locales. Como se ilustra en la Figura 1, un *feature map* o *mapa* es un conjunto de neuronas ocultas cuyas conexiones se limitan a una porción de la entrada y cuyos vectores de pesos $\boldsymbol{w} = (w_1, w_2, ..., w_n)$ son idénticos. Cada capa convolucional cuenta con uno o varios mapas. Es importante observar que las conexiones locales limitan el rango de visión de una neurona dada. Esto implica que las primeras capas tendrán una visión localizada, mientras que las capas superiores desarrollarán una visión más global del espacio de entrada.

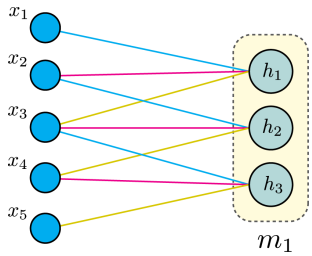


Fig. 1: Capa convolucional con m_1 como único *feature map*. Cada neurona en m_1 se conecta con tan solo tres neuronas de entrada. Las conexiones del mismo color comparten el peso w_i.

Las redes convolucionales son especialmente útiles en la tarea de reconocer patrones en imágenes por varias razones, entre ellas:

- Invarianza traslacional: Cada neurona dentro de un mapa comparte el vector de pesos pero se enfoca en un lugar distinto de la entrada. Esto permite que las neuronas aprendan un feature sin importar la posición que este ocupe en el campo visual de entrada. Se dice que las conexiones locales y los pesos compartidos explotan la propiedad *estacionaria* de las imágenes naturales.
- Eficiencia temporal: Las redes convolucionales tienen una eficiencia superior a las redes neuronales tradicionales durante el entrenamiento, dado que reducen ampliamente la cantidad de parámetros a aprender. Los vectores $\boldsymbol{w}$ tienen muchísimas menos componentes debido a las conexiones locales

y además son compartidos por todas las neuronas de un mismo mapa. Este factor es clave a la hora de trabajar con imágenes dado que ciertos conceptos que resultan de interés para los humanos son solo visibles a partir de cierta resolución.

3. Rugby

El grupo de investigación de *Aprendizaje Automatizado y Aplicaciones* del *CIFASIS-CONICET* cuenta con un conjunto importante de videos de rugby, provenientes de cámaras personales y grabaciones televisivas. Se etiquetaron manualmente 9 partidos (30 videoclips por partido) en 3 clases (10 clips por clase) que representan acciones grupales fácilmente reconocibles por el ojo humano. Estas acciones son (ver Figura 2): Scrum, Line y Juego. Cada videoclip dura exactamente 3 segundos y se compone de 45 fotogramas (15 fotogramas por segundo). Estos datos se separan en 8 partidos para entrenar (Train: 10800 imágenes) y 1 para generalizar (Test: 1350 imágenes).

Fig. 2: Imágenes de ejemplo, extraídas de videos del dataset de rugby. Las etiquetas, por filas de arriba hacia abajo, son: *scrum, scrum, line, line, juego, juego.*

Intrínsecamente, este problema requiere de un modelo robusto a cambios irrelevantes (p. ej., colores de las camisetas, tipo de campo de juego, presencia de tribunas, etc.). Sin embargo, en estos videos aparece una alta gama de variaciones escénicas: diferentes ángulos, intensidad de luz, sombras, cortes de cámara y zoom. Algunas filmaciones personales incluso introducen obstáculos entre la cámara y la acción grupal (ver Figura 2). Todas estas variaciones en el video representan dificultades adicionales para aprender el problema.

4. Overfeat

Overfeat es un framework integral que permite usar redes convolucionales profundas para las tareas de clasificación, localización y detección [7]. Nuestro interés está puesto en la versión *rápida* de la red para clasificación (que llamaremos simplemente *Overfeat*), cuya arquitectura es similar a la ganadora del *ILSVRC12* [9] presentada por *Krizhevsky et al* en [6]. La idea es aprovecharse de los features aprendidas por Overfeat en *ImageNet* [9] (1,2 millones de imágenes etiquetadas en 1000 clases distintas), que hasta cierta capa serán características relevantes en todo tipo de imágenes naturales, y de allí en adelante entrenar una red propia para distinguir las clases de Rugby. Concretamente, propagaremos nuestro dataset a través de Overfeat hasta cierta profundidad, y luego almacenaremos esas propagaciones como un conjunto de datos intermedio que servirá para entrenar y validar otros modelos. Este tipo de aprendizaje es conocido como *Transfer Learning*.

Tabla 1: Arquitectura de Overfeat. Las líneas dobles verticales separan la región convolucional de la región full.

Capa	1	2	3	4	5	6	7	8
Dimensión de entrada	231x231	24x24	12x12	12x12	12x12	6x6	1x1	1x1
Procesos	conv + max	conv + max	conv	conv	conv + max	full	full	full
# mapas	96	256	512	1024	1024	3072	4096	1000
Tamaño de filtro	11x11	5x5	3x3	3x3	3x3	-	-	-
Paso de convolución	4x4	1x1	1x1	1x1	1x1	-	-	-
Tamaño de pooling	2x2	2x2	-	-	2x2	-	-	-
Paso de pooling	2x2	2x2	-	-	2x2	-	-	-

En la Tabla 1 se puede ver la arquitectura de Overfeat. Cada columna representa una capa del modelo. Las capas de la 1 a la 5 son capas convolucionales con unidades con activaciones lineales rectificadas (*rectified linear unit* o *relu*). Algunas de ellas cuentan con un max-pooling luego de la convolución. Las capas 6, 7 y 8 están completamente conectadas, siendo la última capa una Softmax, útil para la clasificación.

Este modelo fue entrenado con ImageNet. Cada imagen del dataset fue escalada hasta que su dimensión más pequeña fuera 256, y luego se extrajeron cinco recortes aleatorios de 231×231 y se presentaron a la red junto a sus espejados horizontalmente en mini-batches de 128. Los pesos se inicializaron aleatoriamente tomados de una distribución normal con $(\mu, \sigma) = (0, 1 \times 10^{-2})$. Luego se actualizaron utilizando el algoritmo de descenso por el gradiente estocástico, acompañado por un término de momentum de 0,6 y un término de weight decay $\mathcal{L}_2$ de 1×10^{-5}. El learning rate fue inicialmente 5×10^{-2} y posteriormente fue decrementando a un factor de 0,5 luego de $(30, 50, 60, 70, 80)$ épocas de entrenamiento. Dropout fue usado en las capas completamente conectadas (capas 6 y 7 del clasificador).

5. Entrenando modelos basados en Overfeat

El enfoque que tendrá el entrenamiento de las redes será *naïve*, en tanto la clasificación de cada clip de video dependerá de la clasificación de cada frame individual que lo compone. Esto significa que, en principio, no será posible extraer información temporal de los videos y que contaremos únicamente con un canal de features espaciales (a diferencia del trabajo propuesto por *Wu et al* en [10] que además extrae características temporales a través de un flujo óptico generado entre cada par de frames). Entendemos que esto será posible dado que las acciones grupales que estamos interesados en identificar son reconocibles a partir de imágenes estáticas.

Usaremos las siguientes abreviaciones para referirnos a los modelos (o a partes de ellos):

- OF7: Overfeat hasta la capa 7.
- OF6: Overfeat hasta la capa 6.
- DNN: Red neuronal con capas full e inicializaciones aleatorias.
- SDAE-DNN: Red neuronal con capas full e inicializaciones mediante Stacked Denoising Autoencoders [4] preentrenados usando datos no etiquetados.
- SM: Capa Softmax

Tanto DNN como SDAE-DNN contarán con una cantidad de capas ocultas y una cantidad de neuronas en las capas ocultas determinadas durante la optimización de hiperparámetros.

Recuerde que Overfeat es una red ya entrenada, por lo que el entrenamiento de los modelos compuestos que involucren a OF6 u OF7 constarán de una etapa preliminar que simplemente propagará los datos de entrada a través de Overfeat, produciendo una salida que será la entrada para el submodelo restante.

6. Resultados

Haciendo uso de las técnicas descriptas en las secciones anteriores obtuvimos los resultados que aparecen en la Tabla 2 para el conjunto de test del dataset de rugby.

Por un lado, es importante notar que la utilización de autoencoders para inicializar los pesos de las DNN encima de Overfeat, no resultó ser mejor que la inicialización aleatoria. Por otro lado, las propagaciones hasta la capa 6 arrojaron mejores resultados que aquellas hasta la capa 7. Entendemos que esto se debe a que los features de la capa 7 son más específicos de las clases de ImageNet y por lo tanto menos útiles para resolver el problema de rugby. Finalmente, las capas *full* encima de las convoluciones no incrementaron sustancialmente la performance.

7. Visualizaciones

El aprendizaje profundo es a menudo criticado por funcionar como un sistema de caja negra, ya que no es trivial entender los conceptos aprendidos. Dependiendo de la distribución de los datos, los modelos profundos podrían ajustar los

Tabla 2: Resultados para el dataset de rugby. Clasificación de los videos mediante votación simple. La relación $A + B$ representa una conexión completa de la salida de A con la entrada de B.

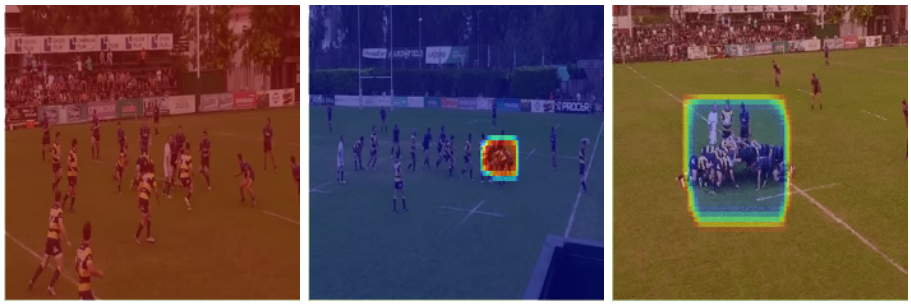
Arquitectura	**Precisión Frames** (%)	**Precisión Videos** (%)
OF7 + SDAE-DNN + SM	$81{,}4 \pm 2{,}1$	$84{,}2 \pm 2{,}1$
OF7 + DNN + SM	$81{,}5 \pm 1{,}6$	$86{,}7 \pm 2{,}1$
OF7 + SM	$81{,}4 \pm 1{,}9$	$84{,}2 \pm 2{,}2$
OF6 + SDAE-DNN + SM	$82{,}3 \pm 1{,}4$	$84{,}2 \pm 2{,}1$
OF6 + DNN + SM	$\mathbf{84{,}8} \pm 2{,}2$	$\mathbf{88{,}7} \pm 2{,}0$
OF6 + SM	$\mathbf{84{,}6} \pm 1{,}8$	$\mathbf{88{,}8} \pm 2{,}7$

parámetros basándose en features irrelevantes para la clasificación. Imagine, por ejemplo, que todos los videos de lines fueron tomados en días nublados y todos los de scrum en días soleados. Este tipo de modelo podría dar una increíble precisión durante el entrenamiento pero no así intentando generalizar con un conjunto más variado de videos. Para mitigar este problema, proponemos algunos mecanismos de visualización. Uno de ellos, consiste en ocultar disitintos sectores de la imagen e iterativamente predecir utilizando algún modelo entrenado con el fin de entender qué partes de la imagen son realmente importantes para la correcta clasificación de la misma, según ese modelo. El proceso toma un parche gris de tamaño p y lo mueve a través de toda la imagen con un salto de $s_x \times s_y$ (ver Figura 3).

Fig. 3: Iteración de Occlude sobre la 27ava fila con un tamaño de oclusión de 69px.

Cada imagen generada de esta manera, se propaga a través de la red y se obtiene la probabilidad de la clase real, generando así un mapa de $\frac{w}{s_x} \times \frac{h}{s_y}$ predicciones, siendo w y h, el ancho y la altura de la imagen en píxeles. Luego para cada pixel de la imagen real se promedian las predicciones de todas las imágenes que lo ocultaron obteniendo así un mapa de probabilidades con las mismas dimensiones que la imagen original. Finalmente, escalamos los valores del mapa para poder apreciar el resultado, lo ploteamos como un mapa de calor y lo ubicamos sobre la imagen original. Las visualizaciones *Occlude* del modelo OF6 + SM (en la Figura 4) exhiben claramente los conceptos aprendidos. Notar, por ejemplo, que la probabilidad de la correcta clasificación del *scrum* fue baja únicamente cuando se ocultó el scrum *per sé*, y ocultar cualquier otra región no alteró la probabildad de la clase correcta. Por el contrario, el line de la Figura 4,

fue siempre mal clasificado salvo cuando de la imagen se quitaron los primeros hombres del line, que están en una posición inclinada, típica del scrum, y en cuyo caso la predicción arrojó una probabilidad muy alta de la clase real. Por último, en la clase *juego* es difícil evaluar las porciones relevantes, siendo la clase de descarte, y tal como se ve en la Figura 4 es bastante robusta a oclusiones.



Juego	Line	Scrum
Oclusión: 69 × 69 px	Oclusión: 19 × 19 px	Oclusión: 69 × 69 px

Fig. 4: Visualizaciones *Occlude* del modelo OF6 + SM para imágenes cuyas clases reales son, de izquierda a derecha, *juego*, *line* y *scrum*.

8. Conclusiones

El presente trabajo propuso varias arquitecturas profundas que mostraron ser robustas a variaciones escénicas de todo tipo (algunas más que otras) y que establecieron la vara inicial del estado del arte en este dataset. Concluímos que con pocos datos, no es posible sacar ventaja de capas completamente conectadas sobre las capas convolucionales y la pre-inicialización con autoencoders no modificó este hecho.

Se investigó además cómo etiquetar un video a partir de la probabilidad de cada uno de sus fotogramas. Los distintos métodos estudiados se comportaron igual o peor que la votación simple, que es el método menos costoso computacionalmente, por lo que se conservó ese método como estándar.

Se propusieron métodos de visualización que revelan sectores críticos de una imagen para un modelo en particular. Esto es especialmente útil para las arquitecturas profundas que componen varias capas de funciones no lineales con millones de parámetros.

Finalmente, es importante destacar que los fotogramas de un video son imágenes estáticas que generan en los modelos características espaciales, únicamente. No obstante, las características presentes en las clases del dataset de rugby son mayormente reconocibles en frames aislados, es por esto que podemos alcanzar una performance satisfactoria. Aún así, entendemos que el problema se podría resolver con mayor eficiencia si considerásemos la componente temporal.

Referencias

1. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy Layer-Wise Training of Deep Networks. In: Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006, págs. 153-160 (2006)
2. Hinton, G.E., Osindero, S., Teh, Y.W.: A Fast Learning Algorithm for Deep Belief Nets. Neural Computation 18(7), 1527-1554 (2006)
3. Sutskever, I., Hinton, G.E., Taylor, G.W.: The Recurrent Temporal Restricted Boltzmann Machine. In: Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008, págs. 1601-1608 (2008)
4. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.: Extracting and composing robust features with denoising autoencoders. In: Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008, págs. 1096-1103 (2008)
5. Taylor, G.W., Hinton, G.E.: Factored conditional restricted Boltzmann Machines for modeling motion style. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009, págs. 1025-1032 (2009)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States. Págs. 1106-1114 (2012)
7. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. CoRR abs/1312.6229 (2013)
8. Wang, N., Yeung, D.: Learning a Deep Compact Image Representation for Visual Tracking. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. Págs. 809-817 (2013)
9. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3), 211-252 (2015)
10. Wu, Z., Wang, X., Jiang, Y., Ye, H., Xue, X.: Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. CoRR abs/1504.01561 (2015)
11. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. CoRR abs/1502.03044 (2015)