



UNIVERSIDAD AUSTRAL

TESIS DE MAESTRÍA

Taming dataset bias via Domain Adaptation

Autor:

Franco LIANZA

Supervisor:

Dr. Leandro BUGNON

19 de agosto de 2022

UNIVERSIDAD AUSTRAL

Resumen

Facultad de Ingeniería

Magister en Explotación de Datos y Gestión del Conocimiento

Taming dataset bias via Domain Adaptation

by Franco LIANZA

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor. . .

Índice general

Resumen	III
Acknowledgements	V
1. Introducción	1
1.1. Elecciones en Argentina	1
1.2. Digitalización de telegramas electorales	2
1.3. Transferencia del aprendizaje	4
2. Estado del Arte	7
2.1. Reconocimiento de dígitos	7
2.2. Redes Neuronales	7
2.3. Domain Adaptation	7
A. Anexo: Telegramas	9
A.1. Ejemplo de telegrama	10
Bibliografía	11

Índice de figuras

1.1. Proceso electoral. TODO: CAMBIAR?	1
1.2. Ejemplos del dataset MNIST	2
1.3. Arquitectura de la red LeNet-5	3
1.4. Estructura de pre-entrenar y fine tuning.	5

Índice de cuadros

1.1. Precisión obtenida al entrenar una LeNet-5 con distintos data-sets de dígitos.	3
---	---

For/Dedicated to/To my...

Capítulo 1

Introducción

1.1. Elecciones en Argentina

En Argentina se celebran elecciones cada 2 años a excepción de las presidenciales que se realizan cada 4 años. En Argentina se realizan tres tipos de elecciones:

- Elecciones nacionales, para elegir a las autoridades federales del país: el Poder Ejecutivo, constituido por el Presidente y el vicepresidente y el Congreso Nacional, formado por Senadores y Diputados.
- Elecciones provinciales y de la Ciudad de Buenos Aires o locales, para elegir a las autoridades de cada provincia: los poderes ejecutivos de las provincias y sus legislaturas.
- Elecciones municipales, regidas por las leyes y procedimientos de cada provincia.

Si bien emitir el sufragio es diferente en cada una de ellas, generalmente consta de ingresar a un cuarto oscuro, elegir el candidato que se desea y depositar el voto en una urna. Al finalizar la jornada, las autoridades de mesas recuentan los votos y llenan una planilla a mano alzada donde se resume la cantidad de votos obtenidos por cada candidato o partido político. Dicha planilla es escaneada y enviada a través de un telegrama correo argentino al centro de cómputo para su procesamiento. Una vez allí, se contabilizan en un sistema informático una por una. Por la metodología de contabilización, idealmente lo escrito a mano en el telegrama y lo computado en el sistema es lo mismo. Sin embargo, como esta tarea es realizada por personas, es plausible pensar que pueden haber errores y demoras en dicho proceso.



FIGURA 1.1: Proceso electoral. TODO: CAMBIAR?

A su vez, durante la jornada electoral existe una ansiedad generalizada de la población por saber los resultados parciales y finales de la misma, por lo que se debe contratar a una gran cantidad de personas destinadas al centro de cómputo. En las elecciones legislativas del 2021 se gastaron unos \$17.000 millones de pesos de los cuales \$4.000 millones de pesos fueron destinados a sueldos para el personal¹. Mejorar el proceso manual de contabilización de los telegramas supondrá un ahorro considerable en el presupuesto de las elecciones, agilizará la obtención de los resultados y aportará transparencia al proceso en general.

La presente tesis enfocará el estudio en las elecciones legislativas de la provincia de Santa Fe del año 2021. Los telegramas son públicos y se encuentran subidos en la [página oficial del estado argentino](#). En el anexo A se adjunta un ejemplo de uno de ellos.

1.2. Digitalización de telegramas electorales

Una solución para bajar los costos de las elecciones podría ser la digitalización automática de los telegramas al sistema de cómputo general. Se puede entrenar un modelo de clasificación de dígitos y utilizarlo al momento de la contabilización de los votos.

La clasificación de dígitos es un problema resuelto desde 1998. En uno de los primeros exponentes de lo que luego se denominó *deep learning*, LeCun et al., 1998 se crea un nuevo dataset de dígitos modificando el existente NIST y se propone la red neuronal LeNet-5 para la clasificación de los mismos. El dataset MNIST consiste de 60.000 imágenes de entrenamiento y 10.000 de testing. Cada una de las imágenes es de un tamaño de 28x28 píxeles.



FIGURA 1.2: Ejemplos del dataset MNIST

¹Fuente: [El cronista](#)

La red posee capas convolucionales las cuales se encargan de extraer características o patrones de las imágenes para luego ser procesadas por capas densas que se encargan de la clasificación.

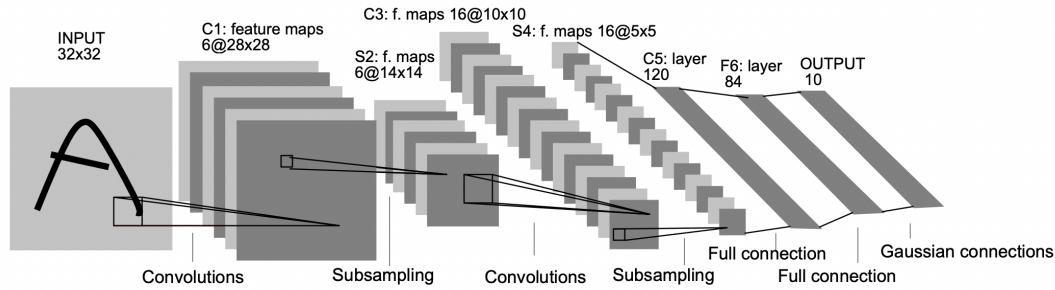

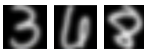



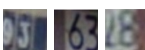


FIGURA 1.3: Arquitectura de la red LeNet-5

Sin embargo, aunque LeNet-5 (o cualquier otro modelo) presente buenas métricas de performance, no significa que pueda ser aplicado a otro dataset de dígitos. El cuadro 1.1 muestra la precisión que se obtiene al entrenar una red LeNet-5 en distintos datasets de dígitos: MNIST (LeCun et al., 1998), USPS (Hull, 1994) y SVHN (Netzer et al., 2011). La arquitectura de la red posee la capacidad de aprender los patrones de cada uno de los ellos pero no logra generalizar a otros. Si bien cada uno de ellos muestran los mismos números, lo hacen de forma diferente. Esto se debe al *sesgo* que existe en los datos. Cuando se entrena un modelo se aprende a reconocer características propias del dataset que le permite resolver el problema, incluyendo el sesgo de los mismos. Esto hace que un problema "sencillo" de clasificación de imágenes de 28x28 píxeles no sea trivial. Todos los datasets se encuentran sesgados de alguna forma y es imposible armarlos de tal forma que no presenten algún nivel de sesgo (Khosla et al., 2012). Cuando los datos con los que se quiere evaluar un modelo provienen de un dominio o distribución diferente al de entrenamiento, se está ante un *dataset shift* (Quinonero-Candela et al., 2008). Es decir, se le muestran datos al modelo con características que nunca vió, provocando predicciones incorrectas.

Entrenamiento	Testing		
	MNIST 	USPS 	SVHN 
MNIST 	99.17 %	78.08 %	31.50 %
USPS 	57.10 %	95.42 %	26.94 %
SVHN 	61.92 %	64.28 %	89.52 %

CUADRO 1.1: Precisión obtenida al entrenar una LeNet-5 con distintos datasets de dígitos.

El problema que abordará la presente tesis es cómo se puede entrenar un modelo que aprenda a clasificar dígitos para ser utilizados en los telegramas de elecciones legislativas de Santa Fe del 2021 (distinto al dominio de entrenamiento) pese al sesgo existente en los datasets mencionados previamente. Para poder lograrlo, se necesita entrenar de cierta manera un modelo en alguno de los datasets públicos etiquetados de forma tal que pueda generalizar a otro similar.

1.3. Transferencia del aprendizaje

Cuando se habla de *Deep learning*, se hace referencia a una serie de algoritmos de *machine learning* que son capaces de utilizar múltiples capas de procesamiento de forma que puedan aprender representaciones de los datos con diferentes niveles de abstracción (LeCun, Bengio e Hinton, 2015). Estos algoritmos, denominados redes neuronales profundas (o DNNs por sus siglas en inglés), poseen la capacidad de encontrar variables que expliquen la naturaleza del comportamiento de los datos.

(TODO: agregar algun grafico de una red donde se muestre que las capas van extrayendo features)

Los modelos obtenidos a partir del *deep learning* han demostrado tener gran capacidad de aprendizaje para todo tipo de problemas, como ser *computer vision* (Szeliski, 2010; Redmon et al., 2016), procesamiento del lenguaje natural (Devlin et al., 2018), reconocimiento del habla (Hannun et al., 2014), juegos (Silver et al., 2016), generación de imágenes a partir de descripciones (Ramesh et al., 2022), entre otros.

Aunque la utilidad de estos modelos se encuentra demostrada y día a día son utilizados en diferentes ámbitos de la vida, presentan un gran problema: la enorme cantidad de datos etiquetados que requieren para su entrenamiento. La mayoría de los modelos que mejores métricas de performance presentan, necesitan millones de datos en sus datasets de entrenamiento. Esto implica que, para que los mismos sean de utilidad, resultan de suma importancia los procesos de recolección y etiquetado de los datos. La eficacia de los modelos queda altamente relacionada con la calidad de los datos que se posean o se logren conseguir. El etiquetado de los datos es una tarea costosa, ineficiente y hasta a veces resulta inviable de realizar (Reis, 2022). Año a año los telegramas de las elecciones son completados a mano por distintas personas, por lo que resulta imposible contar con un dataset lo suficientemente general como para ser utilizado en la clasificación de los dígitos.

Una posible solución a este problema consiste en emular la capacidad que tienen los humanos de adquirir conocimiento relevante en un área y aplicarlo en otra similar (Thrun y Pratt, 1998). Es decir, poder *transferir* lo aprendido. En el caso del *deep learning*, lo que se busca es que la red aprenda representaciones lo suficientemente generales para que puedan ser utilizados en el entrenamiento de una tarea similar. Esto implica que se puede contar con una red pre-entrenada y luego, continuar entrenándola con los datos propios del problema particular que se quiere resolver. A esto último se lo denomina

fine tuning. Lo que se busca es acortar tiempos de entrenamiento, reducir la cantidad de datos necesarios y construir modelos más robustos.

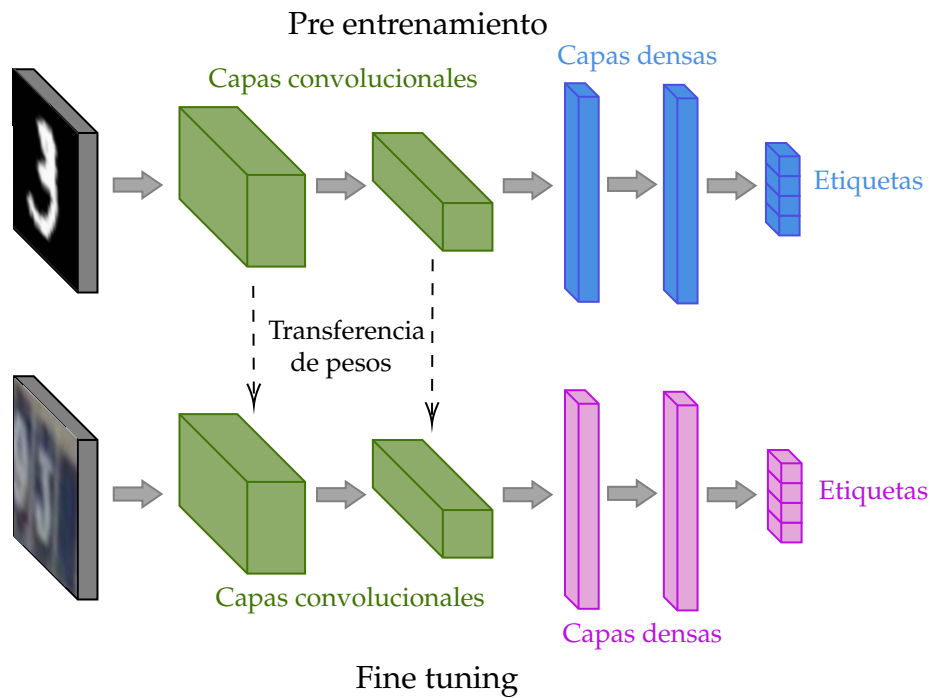


FIGURA 1.4: Estructura de pre-entrenar y fine tuning.

El proceso de pre-entrenar y *fine tuning* ha mejorado considerablemente los resultados obtenidos en diversos problemas del estado del arte, incluso las redes pre-entrenadas pueden ser fácilmente adaptadas a otras tareas con pocos datos etiquetados. No obstante, en muchos escenarios no se cuenta con datos etiquetados, imposibilitando el *fine tuning*. De aquí es que surge la necesidad de transferir el aprendizaje obtenido en un dominio de origen con datos etiquetados a otro de destino donde no se poseen etiquetas (Ben-David et al., 2006). Los modelos de aprendizaje profundo se ven afectados negativamente cuando existe un *dataset shift* como se mostró previamente en el cuadro 1.1. Por lo tanto, la *adaptación de dominio* aparece como una solución cuando la distribución de los datos de entrenamiento difiere de los datos de testing. La idea que persigue es reducir la diferencia que existe entre las distribuciones de origen y destino.

La tesis abordará el uso de diferentes técnicas de *adaptación de dominio* para entrenar un modelo capaz de transferir el conocimiento del dominio del MNIST a los dígitos escritos en los telegramas de las elecciones.

Capítulo 2

Estado del Arte

2.1. Reconocimiento de dígitos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

2.2. Redes Neuronales

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.



2.3. Domain Adaptation

While pre-training on large-scale datasets can gain transferable knowledge in deep models, performing task adaptation with the target data is still necessary for most applications, as the target task is usually different from the pre-training task. When the labeled data for the target task is not enough, domain adaptation from a related source domain with labeled data to boost the performance on the target domain is also necessary in many applications.

Apéndice A

Anexo: Telegramas

A.1. Ejemplo de telegrama

DESTINATARIO:
CON COPIA:

JUNTA ELECTORAL NACIONAL - Distrito - SANTA FE
DIRECCION NACIONAL ELECTORAL

HOJA 1/1

DISTRITO
SANTA FE

SECCION / CIRCUITO
1 - BELGRANO 402 - ARMSTRONG

MESA N°:
1

ELECCIONES GENERALES - 14 DE NOVIEMBRE DE 2021
TELEGRAMA DEL PRESIDENTE DE MESA

CANTIDAD DE SOBRES UTILIZADOS EXTRAIDOS DE LA URNA	2	8	3
TOTAL DE ELECTORES QUE HAN VOTADO SEGÚN EL PADRON	2	8	3
DIFERENCIA ENTRE LAS ANTERIORES	0	0	0

N°	Partido Político /Alianza	CANTIDAD DE VOTOS OBTENIDOS	
		SENADORES NACIONALES	DIPUTADOS NACIONALES
87	UNITE POR LA LIBERTAD Y LA DIGNIDAD	4	5
501	FRENTE AMPLIO PROGRESISTA	15	16
502	FRENTE DE IZQ. Y DE TRABAJADORES-UNIDAD	8	8
503	JUNTOS POR EL CAMBIO	135	135
504	PRIMERO SANTA FE	2	2
505	SOMOS FUTURO	7	7
506	PODEMOS	1	1
507	SOBERANIA POPULAR	6	6
508	FRENTE DE TODOS	94	92
VOTOS EN BLANCO (discriminados por categoría)		5	5
VOTOS NULOS (discriminados por categoría)		6	6
VOTOS RECURRIDOS (discriminados por categoría)		-	-
VOTOS DE IDENTIDAD IMPUGNADA (por todas las columnas debe aparecer la misma cantidad)		-	-
TOTAL DE VOTOS POR CATEGORIA		283	283

SR. PRESIDENTE DE MESA: LA SUMA DE LOS TOTALES POR COLUMNA (CATEGORIA) DEBE COINCIDIR CON LA CANTIDAD DE SOBRES UTILIZADOS QUE SE EXTRAJERON DE LA URNA

PRESIDENTE DE MESA

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
Apellido y Nombres
Firma
N° Documento

1° - VOCAL

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
Apellido y Nombres
Firma
N° Documento

2° VOCAL

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
Apellido y Nombres
Firma
N° Documento

FISCALES PARTIDARIOS PRESENTES

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
Apellido y Nombres
Firma
N° Documento

NO INTRODUCIR EN LA URNA - ENTREGAR AL AGENTE DE CORREOS

Bibliografía

- Ben-David, Shai et al. (2006). «Analysis of Representations for Domain Adaptation». En: *Advances in Neural Information Processing Systems*. Ed. por B. Schölkopf, J. Platt y T. Hoffman. Vol. 19. MIT Press. URL: <https://proceedings.neurips.cc/paper/2006/file/b1b0432ceafb0ce714426e9114852ac7-Paper.pdf>.
- Devlin, Jacob et al. (2018). «Bert: Pre-training of deep bidirectional transformers for language understanding». En: *arXiv preprint arXiv:1810.04805*.
- Hannun, Awni et al. (2014). «Deep speech: Scaling up end-to-end speech recognition». En: *arXiv preprint arXiv:1412.5567*.
- Hull, Jonathan J. (1994). «A database for handwritten text recognition research». En: *IEEE Transactions on pattern analysis and machine intelligence* 16.5, págs. 550-554.
- Khosla, Aditya et al. (2012). «Undoing the damage of dataset bias». En: *European Conference on Computer Vision*. Springer, págs. 158-171.
- LeCun, Yann, Yoshua Bengio y Geoffrey Hinton (2015). «Deep learning». En: *nature* 521.7553, págs. 436-444.
- LeCun, Yann et al. (1998). «Gradient-based learning applied to document recognition». En: *Proceedings of the IEEE* 86.11, págs. 2278-2324.
- Netzer, Yuval et al. (2011). «Reading digits in natural images with unsupervised feature learning». En.
- Quinonero-Candela, Joaquin et al. (2008). *Dataset shift in machine learning*. Mit Press.
- Ramesh, Aditya et al. (2022). «Hierarchical text-conditional image generation with clip latents». En: *arXiv preprint arXiv:2204.06125*.
- Redmon, Joseph et al. (2016). «You only look once: Unified, real-time object detection». En: *Proceedings of the IEEE conference on computer vision and pattern recognition*, págs. 779-788.
- Reis, Pedro Miguel Lima de Sousa (2022). «Data Labeling tools for Computer Vision: a Review». En.
- Silver, David et al. (2016). «Mastering the game of Go with deep neural networks and tree search». En: *nature* 529.7587, págs. 484-489.
- Szeliski, Richard (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.
- Thrun, Sebastian y Lorian Pratt (1998). *Learning to learn*.