



Reglamento y Apuntes de Sistemas de Inteligencia Artificial

La búsqueda de la Inteligencia Artificial

Paula Oseroff

Santiago Reyes

Marco Scilipoti

Ing. Eugenia Sol Piñeiro

Ing. Marina Fuster

Ing. Alan Pierri

Dr. Ing. Rodrigo Ramele

Ing. Luciano Bianchi

Copyright © 2023 Catedra de SIA

PUBLISHED BY ITBA

[HTTPS://GITHUB.COM/EUGEPINEIRO/SIA-TIPS](https://github.com/EUGEPINEIRO/SIA-TIPS)

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, March 2023

Contents

1	Reglamento	5
1.1	Sistemas de Inteligencia Artificial	5
1.2	Clases	5
1.3	Evaluación de la Materia	5
1.4	Trabajos Prácticos	6
1.4.1	Desarrollo	6
1.4.2	Presentación	6
1.4.3	Defensa	6
1.5	Aprobación de la Materia	6
1.6	Comunicación	6
2	Introducción	9
2.1	Nota sobre Bibliografía	9
2.2	Nomeclatura	10
2.2.1	Transformación de los datos	10

I

Algoritmos de búsqueda

II

Algoritmos genéticos

III

Perceptrón simple

IV	Perceptrón Multicapa	
2.2.2	Ecuaciones	23
2.2.3	Backpropagation	24
V	Deep Learning	
3	Autoencoders	29
3.1	Proba	29
3.2	Estimador por Máxima Verosimilitud	29
3.3	Regularización	29
3.4	Derivación del Métodos de Cuadrados Mínimos en base a modelos gráficos 30	
3.5	Tied Weights	31
VI	Guía para el apurado	
4	Balanceo de clases	35
	Index	39

1. Reglamento

1.1 Sistemas de Inteligencia Artificial

La materia tiene el proposito de introducir a los alumnos en las técnicas y la problemática del desarrollo artificial y computacional de lo que denominamos inteligencia. Arranca desde el inicio, incluyendo aspectos filosóficos sobre el concepto. Luego se adentra en las herramientas denominadas clásicas. El núcleo se focaliza en el abordaje en profundidad de las redes neuronales artificiales vistas como un problema de optimización. Finalmente, se abordan técnicas de Aprendizaje Profundo. El enfoque es ofrecer un marco teórico que facilite un desarrollo práctico para entender los conceptos en profundidad.

1.2 Clases

La materia es híbrida. Eso implica que las clases son mayoritariamente remotas, pero algunas clases relevantes son presenciales. Las clases consisten de módulos relacionados pero bastante independientes. Todas las presentaciones estarán disponibles en campus, así como material extra de libros, papers y artículos de internet.

1.3 Evaluación de la Materia

La materia está orientada a "aprender haciendo". Esto implica que la evaluación se realiza mediante 5 Trabajos Prácticos que los alumnos tienen que completar y defender. Adicionalmente a cada trabajo práctico, y durante la defensa, los alumnos tienen que contestar 5 exámenes orales de dos preguntas, que se realiza durante la defensa de cada TP, para evaluar el entendimiento de los temas vistos. El contenido de las preguntas es tanto sobre las clases como sobre los conceptos utilizados en el desarrollo de los TPs. La Nota del TP es **grupal**, y la nota de cada evaluación oral es **individual**.

Adicionalmente hay un conjunto de ejercicios obligatorios durante la cursada que son sin nota pero que hay que entregarlos de manera obligatoria. Por ejemplo, los ejercicios sobre TP0, PCA, el ensayo final (Essay), Kahoot, Transformers y GAN.

Aquellos alumnos que aprueben **TODOS** los ejercicios, derivan la nota de la cursada como nota de aprobación del final.

1.4 Trabajos Prácticos

Los TPs son centrales en esta materia. Son 5, cada uno orientado a una de las temáticas importantes que estudiamos: Inteligencia Artificial Clásica, Algoritmos Genéticos, Redes Neuronales, Aprendizaje No Supervisado, Deep Learning.

1.4.1 Desarrollo

El desarrollo de cada TP conlleva varias partes. La primera es estudiar y repasar los conceptos, luego investigar cómo implementarlos. El paso siguiente es la implementación en sí, que es seguido por la ejecución de los experimentos que sirven para obtener datos que puedan ser analizados. Finalmente la elaboración del informe con los resultados.

Código, Referencias, Librerías y Herramientas

En relación al código, previo al día de defensa de cada TP, los alumnos tienen que entregar un repo referenciando un commit o tag, con los fuentes de la implementación.

Invitamos a los alumnos a que investiguen e incorporen fuentes y referencias variadas para resolver el problema. Nosotros vamos a estar disponibilizando la literatura oficial de la cátedra, pero ustedes pueden complementarlo con cualquier recurso que crean conveniente y que les resulte útil.

Debido a la naturaleza de ser trabajos prácticos que tienen un objetivo didáctico de aprendizaje, para cada uno de los trabajos prácticos hay un conjunto de librerías que NO pueden utilizar, justamente porque son las que tienen que implementar ustedes mismos.

Invitamos también a que utilicen todas las herramientas que necesiten para hacer el trabajo más eficiente. Pero es mandatorio que:

- Se citen con precisión TODAS las fuentes utilizadas de las cuales se obtuvo información.
- Se referencien y aclaren las herramientas utilizadas.
- Se apoyen en el uso de herramientas de construcción, pero entendiendo lo que están implementando y haciendo.

1.4.2 Presentación

Las primeras clases se ofrecen instrucciones sobre cómo es el desarrollo de estas presentaciones, qué tema tienen que enfatizar, como hacer los gráficos y cómo mostrar resultados.

1.4.3 Defensa

La defensa tiene que ser concreta, y no debe superar más de 30 minutos.

1.5 Aprobación de la Materia

Tanto los Trabajos Prácticos como las Evaluaciones Orales se aprueban con nota mayor o igual a 4(cuatro). La aprobación de la materia depende de la calificación obtenida de las 10 notas, y se obtiene como la media aritmética de todas las notas.

1.6 Comunicación

Disponemos tres canales de comunicación

-
- **Campus:** Este es el canal **oficial** de comunicación mediante los mensajes de Campus. Lo concerniente a modalidad de clase (presencial/virtual) fechas de entrega, información de las clases será informada por este canal.
 - **Email:** Disponen de todos los emails de los Profesores que conforman la catedra para comunicarse por cualquier tema relacionado a la materia.
 - **Slack:** Tienen a disposición un canal de Slack de la Materia con un canal general y canales por grupo para que puedan preguntar, sobre todo lo relacionado a la elaboración de los TPs.

2. Introducción

Disclaimer: Este apunte está en preparación y aún no está terminado. La idea es que sirva como una ayuda soporte para los alumnos.

Estos apuntes tratan de la materia Sistemas de Inteligencia Artificial.

2.1 Nota sobre Bibliografía

El area está actualmente en la cresta de la ola (circa 2024) de la innovación, y pocas veces hubo un auge tan grande de trabajos y avances nuevos en el área, de forma tan vertiginosa, hasta el punto que es humanamente imposible seguirle el paso [1].

El libro clásico de Inteligencia Artificial es [2]. Este libro es útil para entender sobre todo la visión más asociada a la inferencia lógica de esta área, la visión más clásica. En una época estudiar IA era estudiar este libro. Dos libros, también clásicos, pero que eran obligatorios como introducción a las redes neuronales son el de Hertz [3] y el de Haykin [4]. Estos libros evidencias las raíces compartidas entre el estudio de las redes neuronales, la física computacional, los sistemas dinámicos y los sistemas complejos (son libros muy *físicos*!).

Llegando a Deep Learning, un excelente libro es el Goodfellow [5]. Por otro lado, el libro de Aggarwal [6] es fácil de leer, y muy completo. Hay que ver si no tiene alguna actualización sobre todo por los últimos desarrollos que le han dado mucha forma a la versión actual de Deep Learning.

Para matemáticas, un excelente libro con una didáctica increíble es *Linear Algebra Done Right* [7]. Les puede servir a aquellos que tengan poco tiempo y quieran hacer un repaso pragmático de algebra lineal. Por el lado de Cálculo, el compendio de 2000 hojas de Jean Gallier es abarcativo hasta el hartazgo, pero contiene todas los temas posibles que pueden necesitar para las clases; usenlo para buscar y repasar un tema puntual[8].

En Optimización, un excelente libro es el de Boyd *Convex Optimization*[9]. Boyd es un investigador de Standford, uno de los primeros que comenzó a subir los videos de sus clases a internet.

Durante la materia vamos a estar compartiendo también muchos papers que les recomendamos que lean, o que al menos conozcan que existen y de qué tratan.

2.2 Nomenclatura

- Aprender bien en el training set: converger
- Aprender bien en el test set: generalizar
- Aprender bien: dada una entrada obtener la salida esperada
- Training set : conjunto de datos usados para calcular los pesos
- Test set: conjunto de datos que NO fueron usados para calcular los pesos
- Parámetros: son las clavijas libres, los pesos, las variables independientes de la función de costo, que la optimización busca encontrar. Los x^* .
- Hyperparámetros: son los parámetros del algoritmo que se ajustan **a mano** generales del algoritmo. No son parte de la función de costo.

2.2.1 Transformación de los datos

- Normalizar: $X/||X||$
- Min-Max Scaling $X_{new} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}} (b - a) + a$
- Feature Scaling $X_{new} = \frac{X_{old}}{X_{max}}, X_{min} = 0, b = 1, a = 0$
- Z-Score, Standarization = "Statistical Standardization" $X_{new} = (X_{old} - X_{mean})/X_{std}$
- Box-Cox Normalization, convierte los datos mediante una optimización para que luzcan "normales" (que provienen de una distribución $\mathcal{N}()$).

Este capítulo está en desarrollo.

Este capítulo está en desarrollo.

2.2.2 Ecuaciones

Ecuaciones útiles a tener en cuenta para el desarrollo del perceptrón multicapa.

$$h = \sum_{i=1}^n w_i \xi_i \quad (2.1)$$

$$\sigma = g(h) \quad (2.2)$$

$$g(h) = tgh(h) \quad (2.3)$$

Función de error

$$\begin{aligned} E &= \frac{1}{2} \sum_{\mu} (\sigma^{\mu} - \zeta^{\mu})^2 \\ &= \frac{1}{2} \sum_{\mu} (\sigma^{\mu} - g(\sum_{i=1}^n w_i \xi_i))^2 \\ \Delta w_i &= -\eta \frac{\partial E}{\partial w_i} \end{aligned} \quad (2.4)$$

$$\begin{aligned} \frac{\partial E}{\partial w_i} &= \frac{\partial \frac{1}{2} \sum_{\mu} (\sigma^{\mu} - \zeta^{\mu})^2}{\partial w_i} \\ &= \frac{\partial \frac{1}{2} \sum_{\mu} (\sigma^{\mu} - g(\sum_{i=1}^n w_i \xi_i))^2}{\partial w_i} \\ &= \frac{\frac{1}{2} \sum_{\mu} 2(\sigma^{\mu} - g(\sum_{i=1}^n w_i \xi_i))}{\partial w_i} \\ &= \frac{\sum_{\mu} (\sigma^{\mu} - g(\sum_{i=1}^n w_i \xi_i))}{\partial w_i} \\ &= \frac{\sum_{\mu} (\sigma^{\mu} - \zeta^{\mu}) g'(h) (\xi_i)}{\partial w_i} \end{aligned}$$

$$\Delta w_i = -\eta \sum_{\mu} (\sigma^{\mu} - \zeta^{\mu}) g'(h) (\xi_i)$$

Para cada uno de los patrones en particular,

$$\Delta w_i = -\eta (\sigma^{\mu} - \zeta^{\mu}) g'(h) (\xi_i)$$

2.2.3 Backpropagation

$$V_j = g(h_j) = g\left(\sum_k w_{jk} \xi_k\right)$$

$$\sigma_1 = g(h_1) = g\left(\sum_j w_{1j} \xi_j\right)$$

El error producido por toda la red multicapa es

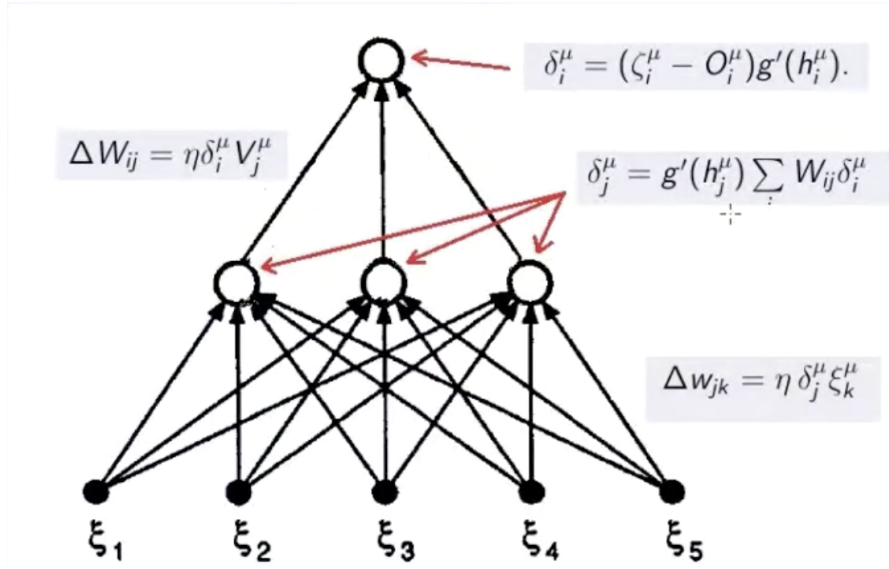
$$\begin{aligned} E &= \frac{1}{2} \sum_{\mu, i} (\sigma_i^\mu - O_i^\mu)^2 \\ &= \frac{1}{2} \sum_{\mu, i} (\sigma_i^\mu - g(\sum_j w_{ij} V_j^\mu))^2 \\ &= \frac{1}{2} \sum_{\mu, i} (\sigma_i^\mu - g(\sum_j w_{ij} g(\sum_k w_{jk} \xi_k^\mu)))^2 \end{aligned}$$

Como se computa la actualización de los pesos sobre muchas capas? Se parte de la actualización para la última capa

$$\begin{aligned} \Delta w_{ij} &= -\eta (\zeta_i^\mu - O_i^\mu) g'(h) V_j^\mu \\ \Delta w_{ij} &= -\eta \delta_i^\mu V_j^\mu \end{aligned}$$

y ahora para las capas anteriores:

$$\begin{aligned} \Delta w_{jk} &= -\eta \frac{\partial E}{\partial V_j^\mu} \frac{\partial V_j^\mu}{\partial w_{jk}} \\ \frac{\partial E}{\partial w_{jk}} &= \frac{\partial \frac{1}{2} \sum_i (\zeta_i - O_i)^2}{\partial w_{jk}} \\ &= \frac{\partial \sum_i (\zeta_i - O_i)}{\partial w_{jk}} \\ &= \sum_i (\zeta_i - O_i) \left(-\frac{\partial O_i}{\partial w_{jk}}\right) \\ &= -\sum_i (\zeta_i - O_i) g'(h_i) \frac{\partial h_i}{\partial w_{jk}} \\ &= -\sum_i (\zeta_i - O_i) g'(h_i) \frac{\partial \sum_j w_{ij} V_j}{\partial w_{jk}} \text{ Cero excepto el término donde } w_{ij} = w_{jk} \\ &= -\sum_i (\zeta_i - O_i) g'(h_i) w_{ij} \frac{\partial V_j}{\partial w_{jk}} \\ &= -\sum_i (\zeta_i - O_i) g'(h_i) w_{ij} \frac{\partial g(\sum_k w_{jk} \xi_k)}{\partial w_{jk}} \\ &= -\sum_i (\zeta_i - O_i) g'(h_i) w_{ij} g'(h_j) \xi_k \end{aligned}$$



Es decir, si esa expresión la tomamos considerando cada patrón en particular,

$$\Delta w_{jk} = \eta \sum_{\mu i} \delta_i^\mu W_{ij} g'(h_j^\mu) \xi_k^\mu$$

Teniendo en cuenta la definición de δ para la última capa

$$\delta_i^\mu = g'(h_i) \cdot 1 \cdot (\zeta - O_i) \quad (2.5)$$

y extendemos esta definición a

$$\delta_j^\mu = g'(h_j) \cdot \sum_i W_{ij} \delta_i^\mu$$

$$\Delta w_{jk} = \eta \sum_{\mu i} \delta_i^\mu W_{ij} g'(h_j^\mu) \xi_k^\mu$$

$$\Delta w_{jk} = \eta \sum_{\mu} \sum_i \delta_i^\mu W_{ij} g'(h_j^\mu) \xi_k^\mu$$

$$\Delta w_{jk} = \eta \sum_{\mu} g'(h_j^\mu) \sum_i W_{ij} \delta_i^\mu \xi_k^\mu$$

$$\Delta w_{jk} = \eta \sum_{\mu} \delta_j^\mu \xi_k^\mu$$



Deep Learning

3	Autoencoders	29
3.1	Proba	
3.2	Estimador por Máxima Verosimilitud	
3.3	Regularización	
3.4	Derivación del Métodos de Cuadrados Mínimos en base a modelos gráficos	
3.5	Tied Weights	

3. Autoencoders

3.1 Proba

$$N(x/\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (3.1)$$

$$N(\mathbf{x}/\vec{\mu}, \Sigma) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\vec{\mu})^T \Sigma^{-1}(\mathbf{x}-\vec{\mu})\right) \quad (3.2)$$

3.2 Estimador por Máxima Verosimilitud

MLE, Maximum Likelihood Estimator. Muestras (x_1, x_2, \dots, x_n) y una p.d.f. $p(x)$. La pregunta que se intenta abordar es, dadas las muestras x_i , ¿Para qué valor de ϕ es más probable que la muestra observada x haya ocurrido ?.

Entonces $L(x_i; \phi) = p(X = x_1)p(X = x_2)\dots p(X = x_n)$. Se busca entonces maximizar $L(\phi)$ mediante:

$$\frac{dL}{d\phi} = 0 \text{ o } \frac{d \ln L}{d\phi}.$$

Theorem 3.2.1 — Máximo de función monótonica.

$$\hat{\phi}_{esMLE} de \phi \Rightarrow f(\hat{\phi})_{esMLE} de f(\phi) \quad (3.3)$$

3.3 Regularización

La Regularización reduce los errores de generalización. Es una manera de agregar restricciones a un "funcional" de optimización para forzar al aprendizaje de algo útil. Se traduce como un término de penalización que se busca que se mantenga pequeño al buscar minimizar una función de costo. Ayuda a no considerar las soluciones complejas o extremas en el propio problema de optimización.

Este método, proviene de la teoría de optimización y se lo conoce como *Tikhonov Regularización*. En ML, puede ser L^2 -regularized o L^1 -regularized. Por ejemplo, en cuadrados mínimos,

$$\min_{\phi} \frac{1}{N} \|y - x\phi\|^2 + (\lambda \|\phi\|^2) \quad (3.4)$$

El método de MLE se transforma en MAP cuando se considera un adicional de regularización. Early Stopping Dropout o Weight decay ($w_{t+1} = w_t - \alpha \Delta_w - \lambda w_t$) son métodos que intrínsecamente involucran una regularización.

3.4 Derivación del Métodos de Cuadrados Mínimos en base a modelos gráficos

Y depende de X y de W.

El objetivo es minimizar $L(w) = \sum_{i=1}^N (W^T x_i - y_i)^2$ para encontrar $\hat{w} = \arg \min_w L(w)$

$$P(y, w, x) = P(y/w, x)P(w)P(x) \quad (3.5)$$

Para eso, la idea sería encontrar los valores de w que maximizan la conjunta de tener los y que tenemos pero dados los x que observamos, por lo que a la condicional tenemos que condicionarlo a esos x que podemos observar (los valores x para los cuales queremos encontrar la resolución por cuadrados mínimos).

$$\begin{aligned} P(y, w, x) &= P(y/w, x)P(w)P(x) \\ P(y, w, x/x) &= P(y/w, x, x)P(w/x)P(x/x) \\ P(y, w/x) &= P(y/w, x)P(w)1 \end{aligned} \quad (3.6)$$

Teniendo en cuenta que $P(x/x)$ es 1. Para especificar este modelo, podemos asumir dos cosas, primero que los valores de y dados x y w están todos alrededor de una normal con media, más o menos, $w^T x$, y desvío $\sigma^2 I$ (es decir solo la diagonal de la matriz de covarianza).

Además podemos asumir que los valores de w tienen media cero, y también un desvío estandar "diagonal" $\omega^2 I$.

Esto es,

$$\begin{aligned} P(y/w, x) &= N(y/w^T x, \sigma^2 I) \\ P(w) &= N(w/0, \gamma^2 I) \end{aligned} \quad (3.7)$$

Asumiendo esto, entonces, $P(y, w/x)$ es

$$\begin{aligned} P(y, w/x) &= P(y/w, x)P(w) \\ &= \log(P(y/w, x)P(w)) \\ &= \log(P(y/w, x)) + \log(P(w)) \\ &= \log\left(\exp\left(-\frac{1}{2}(y - w^T x)^T (\sigma^2)^{-1} (y - w^T x)\right)\right) + \log\left(\exp\left(-\frac{1}{2}w^T (\gamma^2)^{-1} w\right)\right) \quad (3.8) \\ &= -\frac{1}{2\sigma^2}(y - w^T x)^T (y - w^T x) - \frac{1}{2\gamma^2}w^T w \\ &= (-1)[\alpha \|y - w^T x\|^2 + \delta \|w\|^2] \end{aligned}$$

Maximizar eso es lo mismo que minimizar directamente $\|y - w^T x\|^2 + \delta \|w\|^2$. El primer término es entonces el resultado esperado que corresponde a la solución del método de cuadrados mínimos para encontrar una aproximación lineal de la relación entre x e y . Pero en este caso, adicionalmente, nos está dando un término regularizador, lo cual determina que de todas las soluciones posibles, aquella donde w es lo más pequeña mejor, parecería ser la que genera la solución más probable (dadas las distribuciones que asumimos).

3.5 Tied Weights

a) Para un valor fijo de V , la matrix W óptima satisface $D^T D(W^T V^T V - V) = 0$.

$$\begin{aligned} DW^T V^T - D &= 0 \\ D(W^T V^T - I) &= 0 \\ D^T D(W^T V^T - I) &= D^T 0 \\ D^T D(W^T V^T - I)V &= D^T 0V \\ D^T D(W^T V^T V - V) &= 0 \end{aligned} \tag{3.9}$$

b) Si D tiene rango d , entonces, $D^T D$ es invertible y no puede ser cero. Por eso en 3.15, no queda otra que $W^T V^T V - V = 0$ lo cual implica que $W^T V^T V = V$.

c) Demostrar que si $W^T V^T V = V$ entonces, $W = (V^T V)^{-1} V^T$ siempre y cuando $V^T V$ sea invertible.

$$\begin{aligned} W^T V^T V &= V \\ W^T (V^T V) &= V \\ W^T (V^T V) (V^T V)^{-1} &= V (V^T V)^{-1} \text{ invertibilidad de } V^T V \\ W^T &= V (V^T V)^{-1} \\ (W^T)^T &= (V (V^T V)^{-1})^T \\ W &= (V (V^T V)^{-1})^T \\ W &= ((V^T V)^{-1})^T V^T \\ W &= ((V^T V)^T)^{-1} V^T \text{ La T de la inversa, es la inversa de la T} \\ W &= (V^T (V^T)^T)^{-1} V^T \\ W &= (V^T V)^{-1} V^T \text{ Listo} \end{aligned} \tag{3.10}$$

d) Ahora que pasa si los pesos del autoencoder son los mismos (a, b y c). Esto se da cuenta $W = V^T$ donde las columnas de V son ortonormales.

a)

$$\|DW^T V^T - D\| \|D(W^T W) - D\| \tag{3.11}$$

de ahí surge que

$$\begin{aligned} D^T D(W^T W - I) &= 0 \\ W^T W &= I \end{aligned} \tag{3.12}$$

b)

$$\begin{aligned} W^T V^T V &= V \\ W^T W V &= V \\ V &= V \end{aligned} \tag{3.13}$$

y c)

$$\begin{aligned} W^T V^T V &= V \\ W^T (W) V &= V \\ W^T W V &= V \\ V &= V \end{aligned} \tag{3.14}$$

Si los w_i son ortonormales,

$$w_i^T w_j = \begin{cases} 0 & j \neq i \\ 1 & j = i \end{cases} \tag{3.15}$$



Guía para el apurado

4	Balanceo de clases	35
	Index	39

4. Balanceo de clases

Como se sugiere en [10], el balanceo de clases es muy importante.

Bibliography

- [1] M. Academics, *Neurips Conference Analytics*, 2024 (accessed Feb 27, 2024), <https://www.microsoft.com/en-us/research/project/academic/articles/neurips-conference-analytics/>.
- [2] S. J. Russell and P. Norvig, *Artificial intelligence a modern approach*. London, 2010.
- [3] J. A. Hertz, *Introduction to the theory of neural computation*. Crc Press, 2018.
- [4] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [6] C. C. Aggarwal *et al.*, “Neural networks and deep learning,” *Springer*, vol. 10, no. 978, p. 3, 2018.
- [7] S. Axler, *Linear algebra done right*. Springer Nature, 2023.
- [8] J. Gallier and J. Quaintance, “Algebra, topology, differential calculus, and optimization theory for computer science and engineering,” 2019.
- [9] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [10] A. Zhou, F. Tajwar, A. Robey, T. Knowles, G. J. Pappas, H. Hassani, and C. Finn, “Do deep networks transfer invariances across classes?” in *International Conference on Learning Representations*, 2021.



Index

P

Paragraphs of Text 25–27

R

Resumen de Probabilidad 25