

Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica



**Modelado de forma de larvas utilizando máquinas restringidas
de Boltzmann**

Documento de tesis sometido a consideración para optar por el grado académico de
Maestría en Electrónica con Énfasis en Procesamiento Digital de Señales

David Soto Vásquez

Cartago, 23 de mayo, 2015

Declaro que el presente documento de tesis ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos y resultados experimentales propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de tesis realizado y por el contenido del presente documento.

David Soto Vásquez

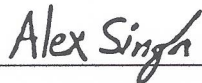
Cartago, 23 de mayo de 2015

Céd: 1-1334-0923

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Tesis de Maestría
Tribunal Evaluador

Tesis de maestría defendida ante el presente Tribunal Evaluador como requisito para optar por el grado académico de maestría, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Dr. Alexander Singh Alvarado
Profesor Lector



Dr. Juan Luis Crespo Mariño
Profesor Lector



Dr. Pablo Alvarado Moya
Profesor Asesor

Los miembros de este Tribunal dan fe de que la presente tesis de maestría ha sido aprobada y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 23 de mayo de 2015

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Tesis de Maestría
Tribunal Evaluador
Acta de Evaluación

Tesis de maestría defendida ante el presente Tribunal Evaluador como requisito para optar por el grado académico de maestría, del Instituto Tecnológico de Costa Rica.


Estudiante: David Soto Vásquez

Nombre del Proyecto: *Modelado de forma de larvas utilizando máquinas restringidas de Boltzmann*

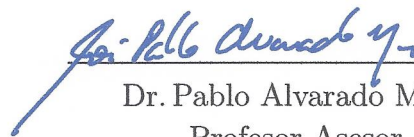
Miembros del Tribunal



Dr. Alexander Singh Alvarado
Profesor Lector



Dr. Juan Luis Crespo Mariño
Profesor Lector



Dr. Pablo Alvarado Moya
Profesor Asesor

Los miembros de este Tribunal dan fe de que la presente tesis de maestría ha sido aprobada y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Nota final de la Tesis de Maestría: 100

Cartago, 23 de mayo de 2015

Resumen

La oclusión y presencia de formas incompletas en imágenes digitales de estructuras vermiciformes como las larvas dificultan los procesos automatizados de segmentación y conteo de las mismas. El presente trabajo aporta un método que considera la información parcial de forma tomada de una imagen para regenerar el borde de una larva de acuerdo a la información global y local previamente aprendida a partir de datos disponibles. La oclusión se puede observar como un caso especial de forma incompleta y por tanto es considerado por el método.

El presente trabajo tiene como objetivo principal el desarrollo de un modelo de forma que incluya la información global y local de la estructura que representa a una larva y que a su vez sea invariable a las modificaciones causadas por la rotación, traslación y cambio de escala. La solución propuesta se basa en el uso del contorno de la larva, propiamente de segmentos de tamaño fijo traslapados a lo largo del contorno, que posteriormente son sometidos a un *análisis de componentes principales* (PCA) y a un cambio de escala y sistema numérico, para finalmente ser procesados por una *máquina restringida de Boltzman* (RBM) o un conjunto de ellas mediante *aprendizaje profundo* con el objetivo de aprovechar las capacidades generativas de la misma. El método sugerido involucra dos etapas; el entrenamiento y la evaluación de la RBM. De esta forma, al haber entrenado la RBM con segmentos de larvas completas las salidas generadas corresponden a la larva que mejor representa la probabilidad conjunta de sus entradas, que pueden ser larvas incompletas, o un traslape de larvas.

Palabras clave: Modelo de forma, segmentación, componentes principales, código gray, máquinas restringidas de Boltzman, aprendizaje profundo

Abstract

Occlusion and presence of incomplete shapes on digital images of vermiform structures as larvae make the automated process for larvae counting and segmentation hard to complete. The suggested solution provides a method that considers the partial shape taken from an image to generate the larva border according to the global and local information previously learned from the available data. Occlusion is a particular case of incomplete shapes so it is considered indirectly as well.

The main goal of this work is to develop a shape model that includes global and local information of larvae's structure, being at the same time rotation, translation and scale invariant. The suggested solution uses the contour or boundaries of a larva to create fixed size overlapped slices that are later passed through a *principal component analysis* (PCA) block. Subsequently a scale change and Gray code numeric system is applied over each slice to be finally processed by a *Restricted Boltzmann Machine* (RBM) or a set of them using *Deep Learning*. This method involves two stages: training and evaluation of the RBM. After training the RBM with slices corresponding to complete larvae the RBM's output will correspond to the best complete larvae that best describes the joint probability distribution of the inputs according to the distribution learned during the training stage, even if the inputs correspond to incomplete or overlapped larvae.

Keywords: Shape model, segmentation, principal components, gray code, restricted Boltzmann Machine, deep learning

A mi amada madre...

Agradecimientos

Primeramente me gustaría agradecerle a Dios por darme las fuerzas y la salud necesaria para cumplir mi meta y superar exitosamente los obstáculos encontrados.

A mi madre Flor Vásquez Jiménez quien siempre me ha motivado a seguir adelante y ha confiado en mis capacidades para alcanzar objetivos. Sin ella jamás lo hubiese logrado.

A mis hermanas y a mi novia Mary Cruz por toda la comprensión y la motivación que me ha dado durante el desarrollo de este trabajo.

Al Dr. Pablo Alvarado, quién ha sido mi mentor y guía más que un simple profesor a lo largo de todos estos años. Muchas gracias por sus consejos y por compartir su conocimiento. A Alex Singh y Juan Luis Crespo por la comprensión, retroalimentación y soporte. Muchas gracias.

Finalmente a la empresa RidgeRun Limitada por el financiamiento otorgado durante cada uno de los periodos del programa de maestría y el tiempo para trabajar en el mismo.

David Soto Vásquez

Cartago, 23 de mayo de 2015

Índice general

Índice de figuras	III
Índice de tablas	VII
Lista de símbolos y abreviaciones	IX
1. Introducción	1
1.1. Estudio de organismos biológicos vermiformes	1
1.2. Segmentación de larvas en imágenes digitales	2
1.3. Propuesta para la reconstrucción de larvas	3
1.4. Objetivos y estructura del documento	4
2. Marco teórico	7
2.1. Segmentación y modelado de forma	7
2.1.1. Segmentación	7
2.1.2. Modelado estadístico de forma	8
2.2. Análisis de Procrustes	9
2.3. Análisis de componentes principales	9
2.3.1. Blanqueo de datos	11
2.4. Función Gaussiana	12
2.5. Cambio de escala	13
2.5.1. Mapeo lineal	13
2.5.2. Función de error	13
2.6. Código Gray	14
2.7. Máquinas restringidas de Boltzmann y aprendizaje profundo	15
2.7.1. Modelos basados en energía	16
2.7.2. Máquinas restringidas de Boltzmann	17
2.7.3. Probabilidad condicional y entrenamiento por gradiente descendente estocástico	18
2.7.4. Divergencia contrastante	20
2.7.5. Alias de verosimilitud	21
2.7.6. Aprendizaje profundo	23
2.8. Trabajos anteriores de modelado con RBM	24
3. Modelado de forma de larvas	27

3.1.	Extracción del contorno de larva	28
3.2.	Adaptación de segmentos a representación de forma	32
3.2.1.	Alineamiento de segmentos	32
3.2.2.	Análisis de componentes principales de segmentos	33
3.2.3.	Cambio de escala	34
3.2.4.	Cambio de sistema numérico	36
3.3.	Creación del modelo de forma de larva	37
3.4.	Reconstrucción de larvas	37
3.4.1.	Generación de larvas por RBM	38
3.4.2.	Ajuste de larva a la imagen	38
4.	Análisis de Resultados	41
4.1.	Extracción del borde de larva	42
4.2.	Extracción de segmentos	42
4.3.	Alineamiento de segmentos a la forma media	44
4.4.	Modelo de distribución de puntos	44
4.5.	Escalado	47
4.5.1.	Mapeo Lineal	47
4.5.2.	Blanqueo de datos y función de error	48
4.6.	Representación numérica del conjunto de entrenamiento	50
4.7.	Creación del modelo de forma: entrenamiento de RBM	50
4.8.	Reconstrucción de larvas: generación de hipótesis	54
4.8.1.	Sistema numérico	55
4.8.2.	Cambio de escala y aprendizaje profundo	58
4.8.3.	Cantidad de unidades ocultas	61
4.8.4.	Cantidad de épocas	64
4.8.5.	Divergencia Contrastante	70
4.8.6.	Reconstrucción a partir del segmento medio	71
4.9.	Ajuste de larvas generadas a la imagen	73
4.10.	Oclusión	74
5.	Conclusiones	79
	Bibliografía	81
A.	Vectores propios del conjunto de segmentos de larva	87

Índice de figuras

1.1. Imagen de ejemplo de larvas proporcionada por la Universidad de Heidelberg	1
1.2. Diagrama de bloques del sistema de segmentación de larvas	2
1.3. Oclusión y larvas incompletas	3
1.4. Diagrama de bloques de la solución propuesta	4
2.1. Imágenes segmentadas y etiquetadas	8
2.2. Conjunto de datos y sus dos componentes principales	11
2.3. Función Gaussiana	12
2.4. Función de error	14
2.5. Ecuación de histograma utilizando la función de error	15
2.6. Máquina Restringida de Boltzmann	16
2.7. Divergencia contrastante	20
2.8. Arquitectura de aprendizaje profundo	23
3.1. Diagrama de bloques para el modelado y reconstrucción de larvas	27
3.2. Diagrama de flujo para la creación del modelo de forma	29
3.3. Diagrama de flujo para la reconstrucción de larvas	30
3.4. Imagen de larvas	30
3.5. Fragmentación de contorno de larva	31
4.1. Extracción de bordes	42
4.2. Probabilidad para el largo de contornos de larvas en imágenes de HCI	43
4.3. Segmentación del contorno	43
4.4. Probabilidad de tamaño de larva por cantidad de segmentos	44
4.5. Segmentos alineados utilizando Procrustes	45
4.6. Variabilidad capturada según la cantidad de valores propios utilizados	45
4.7. Variación causada por el primer vector propio	46
4.8. Variación causada por el segundo vector propio	46
4.9. Variación causada por el tercer vector propio	47
4.10. Histograma de pesos resultantes de PCA	47
4.11. Histograma de pesos resultantes del cambio de escala lineal	48
4.12. Histograma de pesos resultantes del blanqueo de datos	49
4.13. Histograma de pesos resultantes del cambio de escala con blanqueo y ERF	49
4.14. Variación en el costo de reconstrucción según el mapeo	52
4.15. Entropía cruzada para RBM con diferentes cantidades de unidades ocultas	53

4.16. Entropía cruzada de entrenamiento para representación binaria y Gray . . .	53
4.17. Variación del costo de reconstrucción para diferente cantidad de épocas o tiempo de entrenamiento en DBN	54
4.18. Variación del costo de reconstrucción en <i>RBM10</i>	54
4.19. Números de bits cambiados por iteración para binario y código Gray	56
4.20. Promedio de deterioro de la distancia euclideana entre vectores con representación binaria y código Gray en las RBM	56
4.21. Secuencia de pesos de una larva real y 10 secuencias generadas por la RBM binaria y de código Gray	57
4.22. Curvas de tolerancia al cambio de bits para el sistema numérico Gray y Binario	57
4.23. Vector de unidades visibles generado por las RBM con representación binaria y Gray	58
4.24. Números de bits cambiados por iteración para mapeo lineal y mapeo por ERF	58
4.25. Secuencia de pesos de una larva real y 10 secuencias generadas por la RBM con mapeo lineal y con mapeo por ERF	59
4.26. Secuencia de pesos de una larva real y 10 secuencias generadas por aprendizaje profundo con diferentes mapeos de cambio de escala	59
4.27. Curvas de tolerancia al cambio de bits para cambios de escala lineal y por ERF en RBM	60
4.28. Curvas de tolerancia al cambio de bits para cambios de escala lineal y por ERF en aprendizaje profundo	60
4.29. Vector de unidades visibles generado por las RBM para cambios de escala lineal y por ERF	61
4.30. Vector de unidades visibles generado por las DBN para cambios de escala lineal y por ERF	61
4.31. Números de bits cambiados por iteración para diferentes cantidades de unidades ocultas	62
4.32. Promedio de deterioro de la distancia euclideana entre vectores para diferentes cantidades de unidades ocultas en las RBM	63
4.33. Secuencia de pesos generados por RBM para diferentes cantidades de unidades ocultas	63
4.34. Curvas de tolerancia al cambio de bits en RBM para diferentes cantidades de unidades ocultas	64
4.35. Vector de unidades visibles generado por las RBM con diferentes cantidades de unidades ocultas	64
4.36. Números de bits cambiados por iteración para RBM entrenadas con diferente cantidad de épocas	65
4.37. Cantidad promedio máxima de bits cambiantes en función de la cantidad de épocas de entrenamiento	65
4.38. Promedio de deterioro de la distancia euclideana para RBM con diferentes cantidades de épocas	66

4.39. Curvas de tolerancia al cambio de bits para RBM con diferente cantidad de épocas de entrenamiento	66
4.40. Vector de unidades visibles generado por las RBM con diferente cantidad de épocas de entrenamiento	67
4.41. Promedio de números de bits cambiados por iteración para DBN entrenadas con diferente cantidad de épocas	68
4.42. Promedio de deterioro de la distancia euclídeana para DBN con diferente cantidad de épocas de entrenamiento	68
4.43. Curvas de tolerancia al cambio de bits para DBN con diferente cantidad de épocas de entrenamiento	69
4.44. Vector de unidades visibles generado por las DBN con diferente cantidad de épocas de entrenamiento	69
4.45. Promedio de números de bits cambiados por iteración para RBM con diferentes divergencias contrastantes	70
4.46. Promedio de deterioro de la RBM en distancia euclídeana para diferentes divergencias contrastantes	70
4.47. Secuencia de pesos generados por RBM para diferentes divergencias contrastantes	71
4.48. Curvas de tolerancia de RBM al cambio de bits para diferentes divergencias contrastantes	71
4.49. Secuencia de pesos generados por RBM para diferentes divergencias contrastantes	72
4.50. Reconstrucción de larva a partir del segmento medio con RBM y aprendizaje profundo	73
4.51. Hipótesis de corrección ajustadas a la imagen	74
4.52. Larvas sintéticas para la evaluación de oclusión	75
4.53. Hipótesis de corrección de la RBM para larvas sintéticas y segmento traslapado	76
4.54. Hipótesis de corrección de la DBN para larvas sintéticas y segmento traslapado	77

Índice de tablas

2.1. Código Gray y binario de 3 bits	15
3.1. Porcentaje de larvas presente en los conjuntos de prueba, entrenamiento y validación	37
4.1. Cantidad de larvas consideradas antes y después del corrimiento cíclico . .	44
4.2. Valores mínimos y máximos en el espacio de pesos	48
4.3. Parámetros de mapeo lineal de pesos	48
4.4. Descripción del conjunto de representación de larvas	50
4.5. Cantidad de larvas por conjunto	50
4.6. Resultados de entrenamiento de RBM/DBN	51
4.7. Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original para diferentes sistemas numéricos	57
4.8. Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original para diferentes mapeos en RBM y DBN	62
4.9. Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al variar la cantidad de unidades ocultas en la RBM	65
4.10. Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al variar la cantidad épocas de entrenamiento en la RBM	67
4.11. Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al variar la cantidad épocas de entrenamiento en la DBN	69
4.12. Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al variar la divergencia contrastante	72
4.13. Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al utilizar el bits del segmento medio	72
A.1. Vectores propios del conjunto de segmentos	87

Lista de símbolos y abreviaciones

Abreviaciones

ASM	Modelos activos de forma
CD-k	Divergencia contrastante
CE	Entropía cruzada
DBN	Redes de aprendizaje profundo
DCC	Códigos diferenciales
DE	Desviación estándar
ERF	Función de error
HCI	Universidad de Heidelberg
MB	Máquina de Boltzmann
MCMC	Cadena Markov Monte Carlo
MRF	Campos aleatorios de Markov
PA	Análisis de Procrustes
PCA	Análisis de componentes principales
PL	Pseudo-verosimilitud
RBM	Máquinas restringidas de Boltzmann
SVD	Valores singulares

Notación general

A	Matriz. $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$
c	Umbrales de unidades ocultas
$\mathcal{B}(n, p)$	Distribución binomial n intentos con probabilidad p
b	Umbrales de unidades visibles
δ	Cantidad de segmentos completados con el segmento medio
\mathcal{D}_{oe}	Distancia de la larva original a la estimada
\mathcal{D}_{og}	Distancia de la larva original a la generada
\mathcal{D}_{om}	Distancia de la larva original a la modificada
E	Energía
E_p	Fase positiva del gradiente
$E_{\bar{p}}$	Fase negativa del gradiente
\mathcal{C}^P	Conjunto de vectores de larva \mathbf{v}^ℓ de prueba

\mathcal{F}	Energía libre
h_i	Unidad oculta i
\mathcal{L}	Función de verosimilitud
$\mathbf{v}^{\tilde{\cup}}$	Unidades visibles de hipótesis de larva en binario
\mathbf{v}^{Ω}	Unidades visibles de hipótesis de corrección de la RBM
ϖ	Cantidad de segmentos de larva real en una larva incompleta
Γ^s	Matriz de vectores de larva con segmentos S_i
Γ^ℓ	Matriz de vectores de larva con segmentos \hat{S}_i
Θ^ℓ	Parámetros del modelo de forma de larvas
η	Tamaño de segmento S_i en puntos (x, y)
$p^{(x,y)}$	Punto i en (x, y) sobre la imagen
p	Probabilidad
ψ	Matriz de segmentos S_i de entrada para Procrustes
ψ^a	Matriz de segmentos S_i alineados por Procrustes
$\hat{\Psi}$	Matriz de segmentos \hat{S}_i con cambio de escala
Ψ^ρ	Matriz de segmentos \tilde{S}_i proyectados por PCA
Ψ'	Matriz de segmentos blanqueados
S_i	Segmento i de puntos (x, y)
\bar{S}	Segmento medio del conjunto de larvas
\tilde{S}_i	Segmento i de pesos PCA $w_a^\rho, w_b^\rho, w_c^\rho$
\hat{S}_i	Segmento i de pesos con cambio de escala $\hat{w}_a, \hat{w}_b, \hat{w}_c$
τ	Cantidad de larvas individuales utilizadas
\mathcal{C}^E	Conjunto de vectores de larva \mathbf{v}^ℓ de entrenamiento
\mathcal{T}	Matriz de transformación isométrica
$\mathcal{T}^b [\cdot]$	Transformación a binario
$\mathcal{T}^g [\cdot]$	Transformación a código Gray
v_i	Unidad visible i
φ	Número de bits utilizados para representar cada dimensión en la RBM
\mathbf{v}^b	Vector de larva en sistema binario
\mathbf{v}^c	Vector de puntos (x, y) de contorno
\mathcal{C}^V	Conjunto de vectores de larva \mathbf{v}^ℓ de validación
\mathbf{v}^g	Vector de larva en código gray
\mathbf{v}^ℓ	Vector de larva en la entrada a la RBM
\mathbf{v}^ρ	Vector de segmentos con pesos PCA
\mathbf{v}^s	Vector de segmentos de puntos S_i
\mathbf{v}^λ	Vector de varianzas o valores propios
$\mathbf{v}^{\hat{s}}$	Vector de segmentos \hat{S}_i con cambio de escala
\mathbf{W}	Pesos de RBM
w_i^ρ	Peso i según proyección de PCA
\tilde{w}_i	Peso i resultante del cambio de escala
w_i'	Peso i obtenido del blanqueo de datos
χ	Cantidad de segmentos S_i en una larva
Z	Función de partición

ζ Sigmoide

Capítulo 1

Introducción

1.1. Estudio de organismos biológicos vermiformes

Organismos como las larvas de *Drosophila* describen estructuras vermiformes cilíndricas con diferencias en grosor y longitud pero con similitud en apariencia visual en cuanto a forma tal y como se ilustra en la figura 1.1.

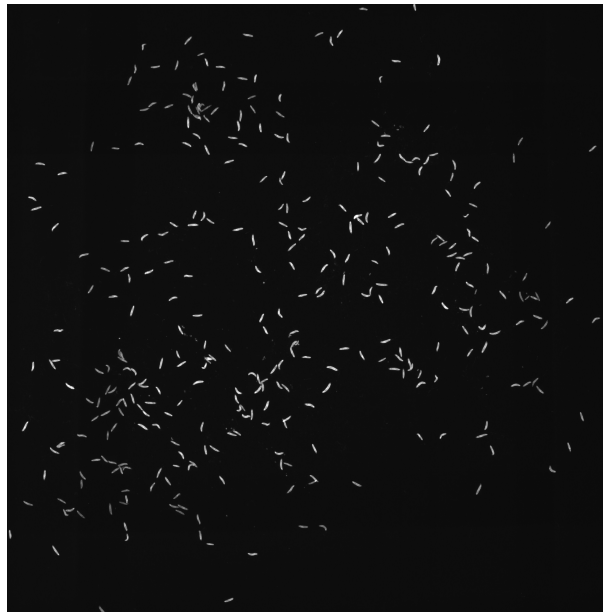


Figura 1.1: Imagen de ejemplo de larvas proporcionada por la Universidad de Heidelberg [35]

Para el caso de larvas de *Drosophila* su estudio se ha basado en varios factores o parámetros que su actividad locomotora describe, como la velocidad de desplazamiento y su incremento con la edad de la larva [21]. Algunos otros parámetros considerados son el número de episodios de actividad e inactividad, la duración de la actividad [41] y su tendencia a la dispersión cuando varias larvas son situadas con una proximidad muy cercana

en un área pequeña, donde estas tienden a ubicarse a los lados más que en las zonas centrales [54, 21].

La ubicación, locomoción y comportamiento de larvas de *Drosophila* muestran que estas exhiben un aprendizaje asociativo y no asociativo y a su vez poseen habilidades en la toma de decisiones. Esta es tan solo una de las razones que ha motivado a los investigadores a estudiar aún más las bases genéticas de las larvas. Dichas investigaciones han sido utilizadas exitosamente para identificar los mecanismos que subyacen a las enfermedades neurodegenerativas humanas como el Parkinson y el Alzheimer, y han contribuido a un mejor entendimiento del comportamiento social y de estados emocionales anormales como la depresión y la ansiedad [30].

El presente trabajo contribuye al desarrollo de sistemas de ubicación, seguimiento y conteo individual de larvas mediante el uso de imágenes digitales, como apoyo al trabajo de biólogos y agrónomos.

La figura 1.2 ilustra los bloques involucrados en el proceso de segmentación automática de larvas en imágenes digitales.

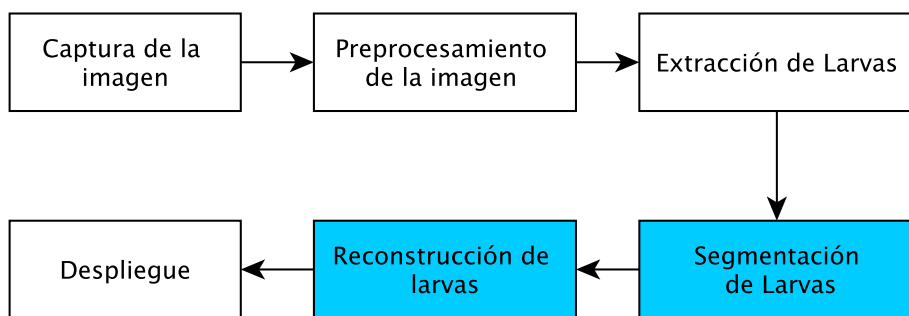


Figura 1.2: Diagrama de bloques del sistema de segmentación de larvas

El presente trabajo evalúa un modelo para segmentación y reconstrucción de larvas incompletas o traslapadas y por tanto se ubica en los bloques de segmentación y reconstrucción. Se hace uso además del trabajo realizado en [35] para la mejora y procesamiento de la imagen y la extracción de posibles larvas o agrupamientos de larvas. Para evaluación se utiliza el conjunto de imágenes de larvas proporcionadas por la Universidad de Heidelberg (HCI) en Alemania utilizada en [35].

1.2. Segmentación de larvas en imágenes digitales

La segmentación de larvas tiene como objetivo la ubicación y seguimiento de individuos en imágenes digitales. Esto se dificulta por la oclusión o traslape de larvas, tal y como se ilustra en la figura 1.3.

Cuando la tarea es realizada por expertos, el análisis de la muestra toma en cuenta una sola aglomeración a la vez con el objetivo de no perder la ubicación de las larvas individuales.



Figura 1.3: Oclusión y larvas incompletas. Tomado de HCI

Se ha buscado apoyo en la tecnología mediante el uso de videos, donde el experto debe reproducir constantemente el video hacia delante y hacia atrás esperando una separación de las larvas para poder definir cuantas hay presentes [35].

Utilizando la metodología basada en videos, Fiaschi [35] ha sugerido el uso de un modelo gráfico determinista que permite analizar dicha separación durante la duración del video y de esta forma estimar la cantidad de larvas en los hacinamientos. No obstante, esto no permite ubicarlas individualmente en el área de oclusión. Por otra parte, en algunas ocasiones los videos no exponen la separación de las larvas o solo se cuenta con fotografías. Dicho planteamiento ignora las larvas incompletas y carece de un modelo de forma y de segmentación de larvas que permita generar una hipótesis del tamaño completo que tendría dicha larva ya que estas pueden variar en tamaño así como en posición u orientación en la imagen. Sumado a esto no es posible tener un criterio único entre expertos acerca de qué tamaño tendría la larva considerando el segmento de larva disponible.

De esta forma es posible plantear la problemática del presente trabajo mediante la siguiente pregunta: *¿Es posible realizar un proceso de segmentación en imágenes de larvas que permita la regeneración de información faltante en casos de oclusión y larvas incompletas?*

1.3. Propuesta para la reconstrucción de larvas

El presente trabajo propone un método de segmentación y reconstrucción que parte de la creación de un modelo de forma de larva individual. Este utiliza una representación de forma compuesta por una secuencia de segmentos de contorno traslapados representados mediante componentes principales (PCA). De esta forma una larva individual es repre-

sentada por el conjunto de coeficientes que describen a los segmentos que la componen.

Posteriormente se adapta la representación de esos coeficientes para hacerla apta como entrada a una máquina restringida de Boltzmann binaria o un conjunto de estas, con el objetivo de explorar sus capacidades regenerativas sobre el modelo de forma diseñado, propiamente mediante la generación una hipótesis de larva que luego debe ser ajustada a la imagen. La figura 1.4 muestra un diagrama de bloques funcional del sistema propuesto.

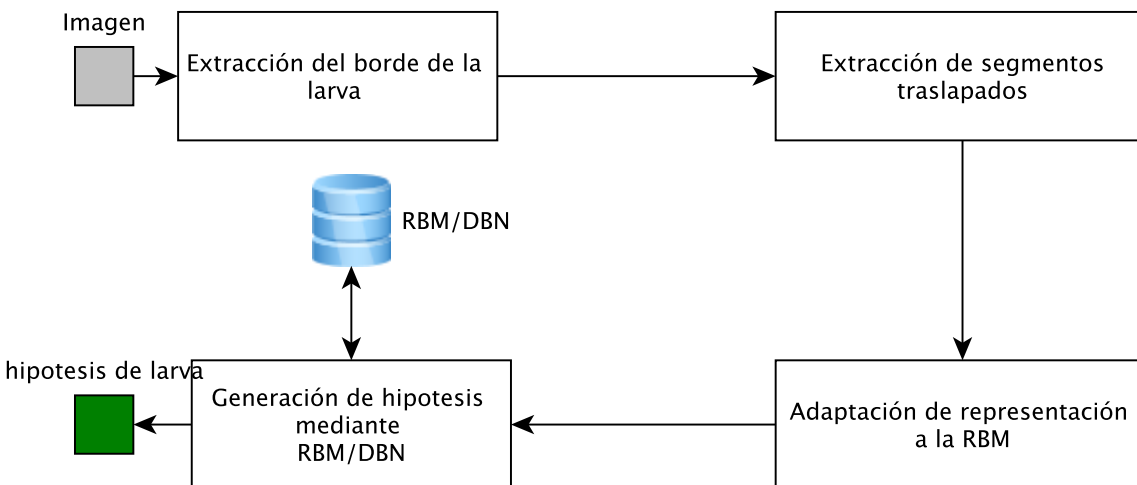


Figura 1.4: Diagrama de bloques de la solución propuesta

Para la solución se proponen algoritmos de procesamiento y análisis de imágenes que han sido implementados haciendo uso del lenguaje C++, así como de módulos existentes en la biblioteca de software para visión por computador y procesamiento digital de imágenes LTI-Lib. Por su parte, los módulos de cambio de escala y sistema numérico han sido implementados en python con el objetivo de utilizar la implementación de máquinas restringidas de Boltzmann (RBM por sus siglas en inglés) y aprendizaje profundo (DBN por sus siglas en inglés) disponible en la biblioteca Theano [7, 5], así como su aceleración mediante el uso de CUDA [46].

1.4. Objetivos y estructura del documento

El principal objetivo del presente trabajo es explorar el uso de modelos probabilísticos generativos como las máquinas restringidas de Boltzmann (RBM) o su generación multicapa de aprendizaje profundo (DBN) mediante el desarrollo de un modelo de forma invariante a la posición, escala y rotación que permita la reconstrucción de larvas incompletas y el manejo de oclusión. Para esto, dicho modelo debe incluir información local y global de la larva de acuerdo a los resultados expuestos en [2] y deberá ser utilizado en las entradas o capas visibles de una red neuronal RBM/DBN. Por ende un modelo de entrenamiento y evaluación necesita ser concebido.

En el capítulo 2 se detallan aquellos conceptos en los que se basa la solución presentada en este trabajo, conformando así el marco teórico de la solución. En el capítulo 3 se expone el diseño conceptual de la solución propuesta y en el capítulo 4 se realiza un análisis de los resultados obtenidos. Por último en el capítulo 5 se incluyen las conclusiones obtenidas mediante el análisis de resultados y se exponen algunas de las recomendaciones para trabajos posteriores .

Capítulo 2

Marco teórico

El presente capítulo detalla los fundamentos teóricos de la solución propuesta. Inicialmente se define el concepto de segmentación y modelo de forma, seguidamente se realiza una introducción al alineamiento mediante el análisis de Procrustes y se brindan los fundamentos teóricos en los cuales el análisis de componentes principales se basa. Por otra parte se incluyen las características y parámetros que describen a una distribución gaussiana y en qué consiste el blanqueo de datos. A su vez dos tipos de cambio de escala son mencionados: el lineal y el mapeo mediante función de error. Posteriormente se incluye una descripción de la representación numérica basada en códigos Gray. Finalmente se proporcionan el marco teórico que define el funcionamiento de las máquinas restringidas de Boltzmann y su variante de aprendizaje profundo, completando el presente capítulo con una reseña de los trabajos previos realizados para solucionar el problema de regeneración durante la segmentación de larvas.

2.1. Segmentación y modelado de forma

2.1.1. Segmentación

La *segmentación* y el *etiquetado* de regiones son técnicas básicas necesarias en el desarrollo de tareas que involucran el agrupamiento y organización de regiones de imágenes en partes coherentes de interés para la aplicación específica[33]. Para algunos autores la segmentación se refiere al agrupamiento de píxeles de la imagen en partes de interés sin aplicar una etiqueta determinada a dichas partes y por su parte, el etiquetado de regiones involucra la asignación de un nombre o categoría específica a las regiones segmentadas [24], por ejemplo: cabello, piel y fondo en aplicaciones de reconocimiento de caras. Algunos ejemplos de segmentación y etiquetado son mostrados en la figura 2.1.

No existe una definición única y generalmente aceptada de segmentación, para otros autores existen dos tipos de segmentación: la parcial y la completa [43]. La parcial es aquella que encuentra regiones en la imagen que corresponden a superficies en el mundo real sin

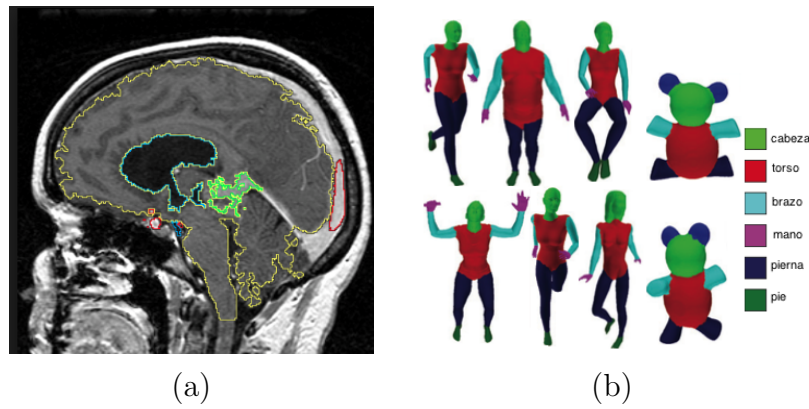


Figura 2.1: (a) Imagen segmentada, (b) Imagen etiquetada. Tomado de [24]

agruparlas en objetos y la completa que detecta objetos reales, es decir, los etiqueta.

En el presente trabajo se utiliza el módulo desarrollado en [35] para realizar una segmentación parcial inicial sobre la imagen de larvas y la solución propuesta pretende generar una mejor hipótesis de segmentación que permita a los siguientes módulos realizar una segmentación completa en condiciones en las que el objeto se presenta incompleto.

2.1.2. Modelado estadístico de forma

El *modelo de forma* corresponde a la distribución estadística que describe la representación de forma seleccionada a lo largo de un conjunto utilizado como muestra (usualmente llamado conjunto de entrenamiento) [12], donde dicha representación pertenece al objeto o superficie de interés sobre la imagen. La forma s de un objeto es típicamente representada mediante un conjunto de puntos N -dimensionales x_i concatenados en un solo vector [57]

$$s = (x_1, x_2, \dots, x_N) \quad (2.1)$$

Para dos o tres dimensiones usualmente los modelos deben considerar las transformaciones de similitud (traslación, rotación y escala) a las que la forma está normalmente expuesta. En general una buena representación de forma permite obtener un modelo que mejora el rendimiento de la aplicación en cuestión.

En [63] se establece que un modelo de forma probabilístico generativo es robusto si cumple con dos requerimientos:

- Realismo: Las muestras del modelo lucen realistas
- Generalización: El modelo puede generar muestras no presentes en el conjunto de entrenamiento.

2.2. Análisis de Procrustes

El análisis de Procrustes (PA) realiza transformaciones de escalamiento, rotación y traslación de la forma bajo análisis tal que se encuentre el mejor ajuste entre los puntos correspondientes a la forma de referencia y la forma a alinear [50]. Estas transformaciones permiten que las diferencias con respecto a la referencia se puedan minimizar preservando los ángulos y líneas paralelas y por lo tanto preservando la forma misma. Estas transformaciones son llamadas transformaciones isométricas.

El análisis ortogonal generalizado de Procrustes es descrito mediante el algoritmo 1 [57]

Algoritmo 1 Análisis de Procrustes

Entrada: Conjunto de formas a alinear representadas por puntos de referencia

Salida: Conjunto de formas alineadas

- 1: Seleccionar una forma del conjunto como primera aproximación de forma media o forma objetivo (Puede ser la primera forma del conjunto)
 - 2: Asignar forma objetivo a nueva forma media
 - 3: **hacer**
 - 4: Asignar nueva forma media como forma media actual
 - 5: **para** cada forma en el conjunto **hacer**
 - 6: Calcular centroide
 - 7: Mover centroide al origen
 - 8: Normalizar cada forma a tamaño unitario
 - 9: Rotar forma según forma media actual
 - 10: **fin para**
 - 11: Calcular nueva forma media del conjunto parcialmente alineado
 - 12: **mientras** nueva forma media \neq forma media actual
-

El cálculo del centroide y las matrices de transformación se describe en detalle en [50, 56].

2.3. Análisis de componentes principales

Cuando se cuenta con un conjunto de datos que posee una cantidad de dimensiones que dificulta el análisis de los mismos es deseable el uso de técnicas que permitan disminuir la dimensionalidad del problema.

El análisis de componentes principales (PCA) permite identificar patrones dentro del conjunto de datos y expresarlos con una dimensionalidad menor como una función de sus características más descriptivas [52], extrayendo así la información relevante del conjunto mediante un filtrado de ruido y de dimensiones redundantes en los datos.

PCA convierte un conjunto de variables correlacionadas en un conjunto de valores descritos por la combinación lineal de sus componentes principales no correlacionados utilizados como nueva base ortonormal para representar (una aproximación de) dichos datos [51].

Sea \mathbf{X} una matriz de dimensiones $m \times n$ correspondientes a n muestras con media cero de dimensión m donde cada dimensión corresponde a una variable del experimento en cuestión y \mathbf{Y} otra matriz $m \times n$, PCA busca encontrar la matriz de transformación lineal \mathbf{P} que relaciona al conjunto \mathbf{X} con el conjunto \mathbf{Y}

$$\mathbf{Y} = \mathbf{P}\mathbf{X} \quad (2.2)$$

donde los vectores fila \mathbf{p}_i con $i \in \{1, \dots, m\}$ de \mathbf{P} corresponden a los componentes principales de \mathbf{X} .

Con el objetivo de encontrar los vectores base que mejor describan al conjunto de datos, PCA plantea la búsqueda de la matriz ortornomal \mathbf{P} tal que la matriz de covarianza

$$\mathbf{C}_Y \equiv \frac{1}{n-1} \mathbf{Y}\mathbf{Y}^T \quad (2.3)$$

sea una matriz diagonal y por ende

- a) Que la covarianza entre las dimensiones j e i , $C_{ij} = 0 \forall i \neq j$ con $i, j \in \{1, \dots, m\}$, es decir, que no exista redundancia entre dimensiones.
- b) Que los términos en la diagonal $C_{ij} \neq 0 \forall i = j$ correspondan a la varianza de la dimensión en cuestión. Donde una mayor varianza es indicativo de mayor dinámica en los datos de interés.

Es posible demostrar[51] que los componentes principales \mathbf{p}_i de \mathbf{X} corresponden a los vectores propios de $\mathbf{X}\mathbf{X}^T$ y que el valor en la diagonal de \mathbf{C}_Y corresponde al i -ésimo valor propio y varianza de \mathbf{X} a lo largo de \mathbf{p}_i .

De esta forma, si los datos siguen una distribución normal, se puede realizar una disminución de la dimensionalidad del conjunto de datos utilizando solamente κ ($\kappa < m$) vectores o componentes principales con las mayores varianzas o valores propios, obteniendo así una proyección \mathbf{Y} de menor dimensión que aproxima cada punto de m dimensiones mediante κ pesos de PCA. Sonka [55] sugiere la selección de κ mediante

$$\frac{\sum_{i=1}^{\kappa} \lambda_i}{\sum_{j=1}^m \lambda_j} \geq \alpha \quad (2.4)$$

con λ_i el i -ésimo valor propio y donde α gobierna qué tanta variación del conjunto de datos puede ser representada por el modelo, $\alpha = 1$ incluye todas las m dimensiones para la representación.

La figura 2.2 muestra un conjunto de datos descrito por dos de sus componentes principales.

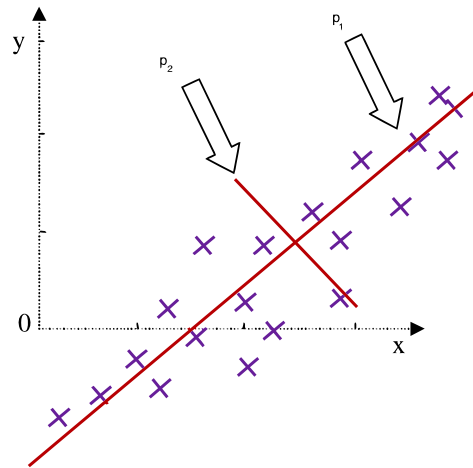


Figura 2.2: Conjunto de datos y sus dos componentes principales \mathbf{p}_1 y \mathbf{p}_2 . Tomado de [23]

2.3.1. Blanqueo de datos

Una vez eliminada la correlación entre los datos mediante la diagonalización de la matriz de correlación, los valores propios en la diagonal describen la dispersión de los datos en la dirección del vector propio correspondiente [47].

El blanqueo de datos convierte a la matriz de correlación en la matriz identidad

$$\mathbf{Y}\mathbf{Y}^T = \mathbf{I} \quad (2.5)$$

transformando las superficies equiprobables de elipsoides a esferoides al normalizar la dispersión en todas las dimensiones [47].

Es posible demostrar mediante *descomposición de valores singulares* [3] (SVD) que los datos con blanqueo \mathbf{Y}_w^T se obtienen al escalar los valores proyectados \mathbf{Y}^T por la matriz diagonal que contiene los valores propios [32]

$$\mathbf{Y}_w^T = \mathbf{Y}^T \mathbf{\Lambda}^{-\frac{1}{2}} \quad (2.6)$$

donde

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \lambda_k \end{bmatrix}$$

con λ_i el i -ésimo valor propio.

2.4. Función Gaussiana

Se define una variable aleatoria como aquella función que asigna o relaciona un número perteneciente a un espacio muestral a cada resultado de un experimento [61]; puede ser discreta o continua. Ésta se encuentra distribuida normalmente si tiene una función de densidad de probabilidad descrita por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (2.8)$$

Esta función es denominada función o distribución gaussiana, con valor medio μ y varianza σ^2 . La función gaussiana es simétrica respecto a su parámetro μ en el cual alcanza su valor máximo y describe una curva en forma de campana con una dispersión según su varianza [49]. Un 99,74% de esta dispersión se encuentra comprendida en el intervalo $[\mu - 3\sigma, \mu + 3\sigma]$ tal y como se ilustra en la figura 2.3.

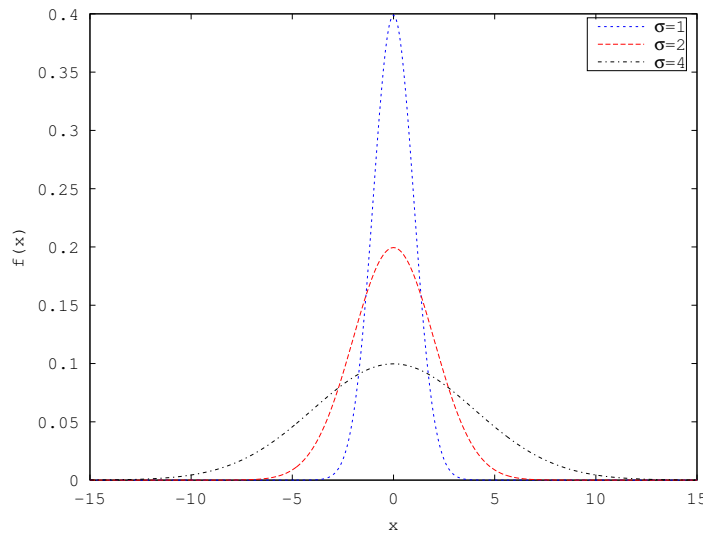


Figura 2.3: Función Gaussiana

La función gaussiana está completamente descrita a través de sus dos parámetros μ y σ^2 y los cuales son sus momentos de primer y segundo orden respectivamente. Para el caso de variables aleatorias discretas, es posible encontrar su valor utilizando las siguientes ecuaciones:

$$\mu = \frac{\sum_{i=0}^n x_i}{n} \quad (2.9)$$

$$\sigma^2 = \frac{\sum_{i=1}^n x_i^2 - n\mu^2}{n} \quad (2.10)$$

2.5. Cambio de escala

El cambio de escala permite variar el rango de valores descrito por un conjunto de datos a un rango que sea adecuado para el sistema que los procesa mediante un *mapeo*. Los mapeos estudian como transformar una región específica de un plano z en otra región del plano w al aplicar $w = f(z)$ [1].

La idea general del mapeo es realizar una transformación entre dos conjuntos Z y W permitiendo otro modo de visualización y análisis de los datos para simplificar modelos geométricos o adecuar el rango de datos al deseado en el sistema. En ingeniería son aquellas funciones f invertibles las que encuentran mayor aplicación tal que

$$w = f(z) \Rightarrow z = f^{-1}(w) = f^{-1}(f(z)) \quad (2.11)$$

En esta sección se describen dos tipos de mapeo de variable real ($Z, W \subseteq \mathbb{R}$), el lineal y el realizado por la función de error, estos son utilizados en el capítulo 3.

2.5.1. Mapeo lineal

Un mapeo lineal [1] es realizado por una función f de variable real de la forma

$$f(z) = w = \alpha z + \beta, \quad \alpha, \beta \in \mathbb{R} \quad (2.12)$$

De esta forma es posible realizar un cambio de escala considerando la escala original como el dominio Z de la función f y su rango W la nueva escala.

Si se conocen dos pares ordenados (z_{min}, w_{min}) y (z_{max}, w_{max}) que definen los rangos de entrada y salida, α y β se obtienen con [58]

$$\alpha = \frac{w_{max} - w_{min}}{z_{max} - z_{min}} \quad (2.13)$$

$$\beta = w_{max} - \alpha z_{max} \quad (2.14)$$

2.5.2. Función de error

La función de distribución acumulativa $\Phi(x)$ correspondiente a una distribución de probabilidad normal o gaussiana se describe mediante la función de error (ERF) $\text{erf}(x)$ definida como [37]

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (2.15)$$

la cual es ilustrada en la figura 2.4a. Entre sus características están [37]

- $\text{erf}(-\infty) = -1$
- $\text{erf}(-x) = -\text{erf}(x)$
- $\text{erf}(+\infty) = 1$
- $\text{erf}(x^*) = [\text{erf}(x)]^*$

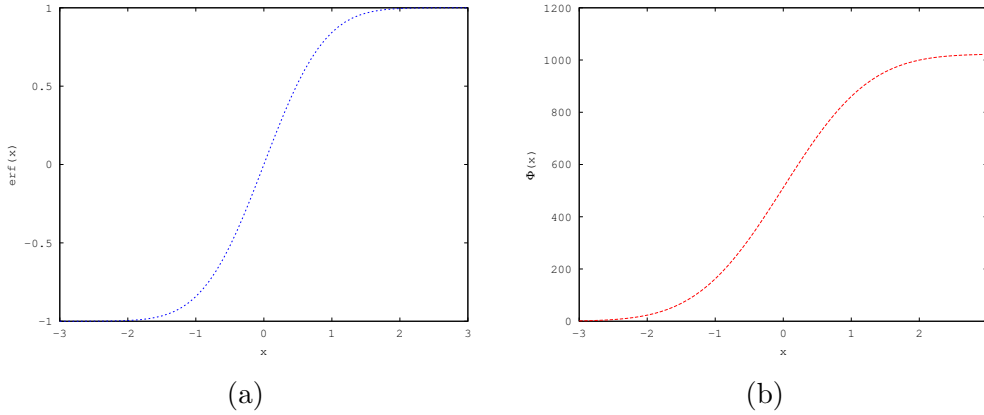


Figura 2.4: (a) Función de error, (b) Función de cambio de escala $\psi\Phi(x)$ con $\psi = 1023$, $\mu = 0$ y $\sigma = 1$

De esta forma se define $\Phi(x)$ como [62]

$$\Phi(x) = \frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{x - \mu}{\sigma\sqrt{2}}\right) \quad (2.16)$$

Mediante el uso de $\Phi(x)$ y una amplitud ψ a definir es posible un cambio de escala como se ilustra en la figura 2.4b.

$$w = \psi\Phi(x) = \psi\left(\frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{x - \mu}{\sigma\sqrt{2}}\right)\right) \quad (2.17)$$

Al aplicar este cambio sobre un conjunto de datos Z se realiza a su vez un cambio en la distribución o histograma como se muestra en la figura 2.5a y 2.5b para el caso de una distribución normal ($\mu = 0$ y $\sigma = 1$) que posteriormente son mapeados mediante $\psi\Phi(x)$ dando como resultando un histograma ecualizado [22, 15].

2.6. Código Gray

La naturaleza de los sistemas electrónicos requiere un sistema numérico capaz de ser utilizado en operaciones aritméticas y en forma general para la representación digital de los datos. La selección de dicho sistema numérico depende de la naturaleza intrínseca del problema a resolver.

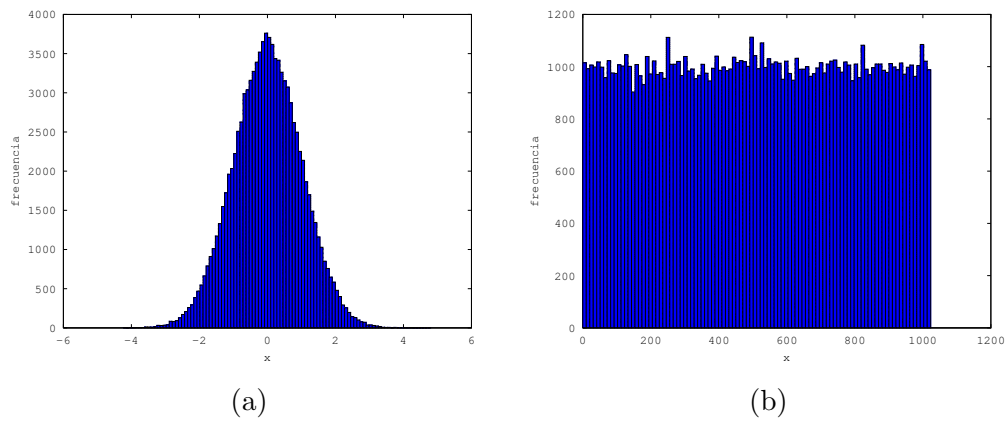


Figura 2.5: (a) Distribución normal de Z con $\mu = 0$ y $\sigma = 1$, (b) Histograma resultante después de aplicar ERF

Tabla 2.1: Código Gray y binario de 3 bits

Decimal	Binario			Gray		
	b2	b1	b0	g2	g1	g0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	0	1	0	0

El código Gray o código binario reflejado [31], fue propuesto en 1947 por Frank Gray con el objetivo de evitar espurias en las salida de un sistema electromecánico durante el estado de transición de interruptores. Se basa en el ordenamiento binario de 2^n dígitos tal que todo par de elementos numéricos consecutivos difiera únicamente por una posición, por tanto es considerado un código de distancia unitaria en su representación geométrica [45]. La Tabla 2.1 muestra el código Gray para 3 bits.

Es posible observar que a diferencia de la representación binaria usual un incremento unitario en la representación decimal involucra el cambio de solo un bit.

2.7. Máquinas restringidas de Boltzmann y aprendizaje profundo

Una *Máquina restringida de Boltzmann* (RBM), ilustrada en la figura 2.6, es un modelo gráfico probabilístico capaz de representar una distribución de probabilidad p sobre un

vector de unidades visibles \mathbf{v} y un vector \mathbf{h} de unidades ocultas o latentes [14, 34]. Una RBM es un tipo de red Hopfield [4] diseñado como una versión restringida de las Máquinas de Boltzmann (MB) en donde a diferencia de estas, en las RBM las unidades estocásticas (visibles y ocultas) no se encuentran completamente conectadas, haciendo manejable su requerimientos computacionales en el entrenamiento [14].

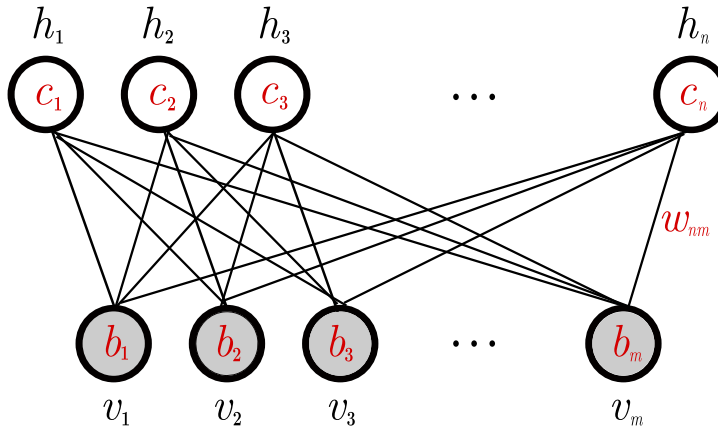


Figura 2.6: Máquina Restringida de Boltzmann

La RBM es considerada un modelo de aprendizaje de máquina basado en una representación de energía con capacidades de inferencia y generación, ya que después de aprender los pesos sinápticos de un conjunto de observaciones, la red es capaz de generar una secuencia de estados o unidades visibles cuyas probabilidades coinciden con las observaciones presentadas durante el entrenamiento [4].

El uso de RBMs fue inicialmente propuesto por Smolensky en 1986 [53]; sin embargo su entrenamiento no era computacionalmente manejable hasta que Hinton et al. [27] desarrollaron la *Divergencia Contrastante*; un algoritmo de entrenamiento basado en el *muestro de Gibbs* para obtener una secuencia de muestras de una distribución posterior [8] y así poder estimar uno de los parámetros del modelo. El muestreo de Gibbs es una cadena Markov Monte Carlo [48] que mediante un proceso iterativo genera una cadena de estados en donde cada estado corresponde a una muestra más cercana a la distribución posterior deseada [60]. Como se describe matemáticamente en las siguientes secciones el algoritmo de divergencia contrastante realiza k -pasos en dicha cadena para actualizar los pesos de la RBM durante el entrenamiento y existe una variante *persistente* a dicho proceso [59].

2.7.1. Modelos basados en energía

Los modelos basados en energía asocian un escalar de energía a la configuración de variables de interés mediante un conjunto de ecuaciones inspiradas en los modelos de la mecánica estadística, en donde un sistema termodinámico es caracterizado mediante los posibles arreglos de átomos y moléculas de acuerdo a la teoría desarrollada por Boltzmann [60]. La energía asociada a la probabilidad de que un conjunto de partículas ocupe

un cierto nivel o macro estado [39, 44] es definida por la ecuación de la *distribución de Boltzmann*

$$p(i) = \frac{1}{Z} e^{-\beta E_i} \quad (2.18)$$

con $\beta = \frac{1}{k_b T}$, T la temperatura, k_b la constante de Boltzmann y Z la *función de partición* utilizada en la teoría de probabilidad como una constante de normalización considerada en los modelos de energía como la suma de todos los micro estados del sistema

$$Z(\beta) = \sum_{i=1}^n e^{-\beta E_i} \quad (2.19)$$

Los modelos de aprendizaje de máquina aprenden por medio de la optimización de la probabilidad en la distribución de Boltzmann, modificando la energía del sistema. Para esto se utiliza el negativo del logaritmo de la probabilidad como función objetivo sobre la cual se aplica el *gradiente descendente estocástico* [11].

2.7.2. Máquinas restringidas de Boltzmann

Las máquinas restringidas de Boltzmann (RBM) son caracterizadas por su topología o estructura bipartida [13]. Poseen enlaces únicamente entre las unidades ocultas \mathbf{h} y visibles \mathbf{v} pero no entre unidades de una misma capa, y por tanto no existen dependencias entre nodos visibles o entre nodos ocultos [60] (figura 2.6).

Las RBM son un caso particular de los modelos basados en energía utilizados para el aprendizaje de máquina en donde usualmente $\beta = 1$ [60]. Su modelo representa una distribución de energía de Boltzmann y considera que existen variables visibles y ocultas en el sistema. La función de energía es dada para un conjunto de pesos \mathbf{W} y umbrales \mathbf{c} , \mathbf{b} por [6]

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{b}^T \mathbf{v} - \mathbf{c}^T \mathbf{h} \quad (2.20)$$

Es posible reescribir la ecuación de energía en función de los valores individuales para la configuración conjunta (\mathbf{v}, \mathbf{h}) [6, 26]

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{ocultas}} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (2.21)$$

con v_i, h_j los estados binarios de la unidad visible i y la unidad oculta j respectivamente, a_i, b_j son los umbrales asociados a estas y w_{ij} el valor del peso que las asocia.

La distribución de probabilidad está dada por

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v}, \mathbf{h})}}{Z} \quad (2.22)$$

donde la *funcion de partición* Z es definida sobre todos los posibles vectores \mathbf{v} y \mathbf{h}

$$Z = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} \quad (2.23)$$

De esta forma la probabilidad asignada por la RBM a un vector visible \mathbf{v} se calcula sumando todos los posibles vectores ocultos [6](distribucion marginal)

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} \quad (2.24)$$

Finalmente, utilizando (2.20) en (2.24) se obtiene [60]

$$p(\mathbf{v}) = \frac{e^{-\mathcal{F}(\mathbf{v})}}{Z} \quad (2.25)$$

$$Z = \sum_{\mathbf{v}} e^{-\mathcal{F}(\mathbf{v})} \quad (2.26)$$

donde \mathcal{F} es llamada *energía libre* inspirado en el concepto de la mecánica estadística y es

$$\mathcal{F}(\mathbf{v}) = -\mathbf{b}^T \mathbf{v} - \sum_{j=1}^n \log \sum_{h_j} e^{h_j(c_j + W_j v)} \quad (2.27)$$

Para el caso binario donde $\mathbf{h}_j \in \{0, 1\}$ $\mathcal{F}(\mathbf{v})$ se simplifica tal que

$$\mathcal{F}(\mathbf{v}) = -\mathbf{b}^T \mathbf{v} - \sum_{j=1}^n \log (1 + e^{(c_j + W_j v)}) \quad (2.28)$$

De esta forma utilizando (2.25), (2.26), (2.28) y gradiente descendente estocástico se ha desarrollado un método de inferencia como se muestra en la sección 2.7.3.

2.7.3. Probabilidad condicional y entrenamiento por gradiente descendente estocástico

Dado un conjunto de entrenamiento \mathbf{V} conformado por subconjuntos de observaciones o unidades visibles \mathbf{v} ; $\mathbf{v} \subset \mathbf{V}$, el proceso de entrenamiento de la RBM consiste en un proceso iterativo de optimización de los parámetros \mathbf{W} , \mathbf{b} y \mathbf{c} [13, 14] tal que estos logren describir la distribución de probabilidad de los vectores \mathbf{v} .

Con el objetivo de completar el análisis es necesario encontrar inicialmente las probabilidades condicionales $P(\mathbf{h}|\mathbf{v})$ y $P(\mathbf{v}|\mathbf{h})$ [6, 19]

$$\begin{aligned}
P(\mathbf{h}|\mathbf{v}) &= \frac{p(\mathbf{v}, \mathbf{h})}{p(\mathbf{v})} \\
&= \frac{e^{(\mathbf{b}^T \mathbf{x} + \mathbf{c}^T \mathbf{h} + \mathbf{h}^T \mathbf{W} \mathbf{v})}}{\sum_{\tilde{\mathbf{h}}} e^{(\mathbf{b}^T \mathbf{x} + \mathbf{c}^T \tilde{\mathbf{h}} + \tilde{\mathbf{h}}^T \mathbf{W} \mathbf{v})}} \\
&= \frac{\prod_i e^{(c_i h_i + h_i W_i v)}}{\prod_i \sum_{\tilde{h}_i} e^{(c_i \tilde{h}_i + \tilde{h}_i W_i v)}} \\
&= \prod_i \frac{e^{(h_i (c_i + W_i v))}}{\sum_{\tilde{h}_i} e^{(\tilde{h}_i (c_i + W_i v))}} \\
&= P(h_i|\mathbf{v})
\end{aligned} \tag{2.29}$$

donde para el caso binario $h_i \in \{0, 1\}$, sustituyendo se obtiene [60]

$$P(h_i = 1|\mathbf{v}) = \frac{e^{c_i + W_i v}}{1 + e^{c_i + W_i v}} = \zeta(c_i + W_i \mathbf{v}) \tag{2.30}$$

con ζ la *función sigmoide*. De forma análoga [14] es posible obtener $P(\mathbf{v}|\mathbf{h})$, tal que

$$P(\mathbf{h}|\mathbf{v}) = \zeta(\mathbf{c} + \mathbf{W} \mathbf{v}) \tag{2.31}$$

$$P(\mathbf{v}|\mathbf{h}) = \zeta(\mathbf{b} + \mathbf{W}^T \mathbf{h}) \tag{2.32}$$

Si se cuenta con las unidades visibles es posible obtener la probabilidad de las unidades ocultas y viceversa. Por otra parte, se define la función de probabilidad $p(\mathbf{v})$ como la función de verosimilitud [13]

$$\mathcal{L} = p(\mathbf{v}) = \frac{e^{-\mathcal{F}(\mathbf{v})}}{Z} \tag{2.33}$$

y se calcula el gradiente de \mathcal{L} con respecto al conjunto de parámetros θ del modelo [6]

$$\begin{aligned}
-\frac{\partial \log p(v)}{\partial \theta} &= \frac{\partial \mathcal{F}(v)}{\partial \theta} - \frac{1}{Z} \sum_{\tilde{v}} e^{-\mathcal{F}(\tilde{v})} \frac{\partial \mathcal{F}(\tilde{v})}{\partial \theta} \\
&= \frac{\partial \mathcal{F}(v)}{\partial \theta} - \sum_{\tilde{v}} p(\tilde{v}) \frac{\partial \mathcal{F}(\tilde{v})}{\partial \theta}
\end{aligned} \tag{2.34}$$

Se obtiene un estimado del gradiente mediante el uso del valor esperado [26, 25]

$$-\frac{\partial \log p(v)}{\partial \theta} \approx E_p \left[\frac{\partial \mathcal{F}(v)}{\partial \theta} \right]_{\text{datos}} - E_{\tilde{p}} \left[\frac{\partial \mathcal{F}(\tilde{v})}{\partial \theta} \right]_{\text{modelo}} \tag{2.35}$$

donde E_p se obtiene manteniendo las unidades visibles \mathbf{v} constantes y muestreando \mathbf{h} , lo cual es posible mediante (2.23), (2.31) y (2.32) y es llamada *fase positiva*.

Por su parte $E_{\tilde{p}}$ es llamada *fase negativa* y se debe calcular sobre la distribución $p(\mathbf{v}, \mathbf{h})$ generada por el modelo, por tanto es necesario obtener ambas muestras; \mathbf{v} y \mathbf{h} , de p por medio de un método computacionalmente manejable para así contar con un estimador estocástico del gradiente [6]. Para lograrlo se hace uso de la divergencia contrastante sugerida por Hinton et al. [27].

2.7.4. Divergencia contrastante

La fase negativa del gradiente $E_{\tilde{p}}$ en (2.35) se calcula mediante el muestreo de Gibbs [48], es decir, fijando las unidades visibles \mathbf{v}_0 seguido de un muestreo de las unidades ocultas $\tilde{\mathbf{h}}_0$ mediante (2.31), seguido de otro muestreo de las unidades visibles $\tilde{\mathbf{v}}_1$ y repitiendo el proceso k veces con $k \rightarrow \infty$, causando que el modelo olvide el punto de inicio \mathbf{v}_0 y obtenga muestras de la distribución de equilibrio, formando una cadena Markov Monte Carlo (MCMC) [34, 48]. No obstante, este tipo de aproximación puede resultar no manejable y computacionalmente intensa, traduciendo en un consumo de tiempo no apto para aplicaciones prácticas [14, 6].

La Divergencia Contrastante (CD- k) [27] consiste en un método de aproximación del gradiente (2.35) por medio de un muestreo de Gibbs de una cantidad limitada de pasos k , donde k pasos corresponden a $k + 1$ muestreos de las unidades ocultas. Esto es ilustrado en la figura 2.7.

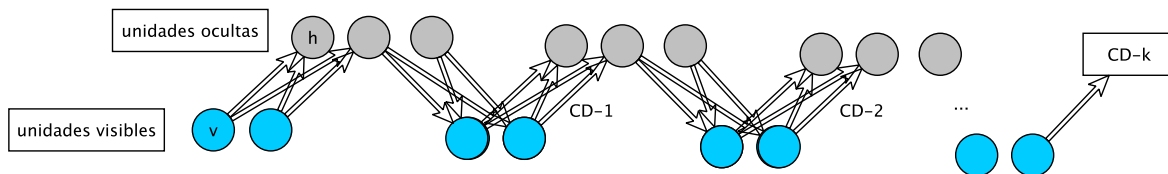


Figura 2.7: Divergencia contrastante

Hinton et al. [27, 26] demostraron que la utilización de $k = 1$ pasos (CD-1) produce resultados adecuados a menudo ya que permite disminuir la energía que el modelo asigna a los vectores de entrenamiento e incrementa la energía de las reconstrucciones de vectores de entrenamiento generadas por el modelo. No obstante Krizhevsky [34] menciona que tiene como debilidad que los lugares de la superficie de energía que no están cerca de los datos no son explícitamente modificados por el algoritmo aunque estos sean afectados por los cambios que CD-1 induce.

Una vez realizado CD-1 se actualizan los parámetros utilizando la tasa de aprendizaje ϵ y los valores de las derivadas $\frac{\partial \mathcal{L}(v)}{\partial w}$, $\frac{\partial \mathcal{L}(v)}{\partial b}$ y $\frac{\partial \mathcal{L}(v)}{\partial c}$ [6]

$$\mathbf{W} \leftarrow \mathbf{W} + \epsilon \left(\tilde{\mathbf{h}}_0 \mathbf{v}_0^T - p(\tilde{\mathbf{h}}_1 = 1 | \tilde{\mathbf{v}}_1) \tilde{\mathbf{v}}_1^T \right) \quad (2.36)$$

$$\mathbf{b} \leftarrow \mathbf{b} + \epsilon (\mathbf{v}_0 - \tilde{\mathbf{v}}_1) \quad (2.37)$$

$$\mathbf{c} \leftarrow \mathbf{c} + \epsilon \left(\mathbf{h}_0 - p(\tilde{\mathbf{h}}_1 = 1 | \tilde{\mathbf{v}}_1) \right) \quad (2.38)$$

De esta forma es posible resumir el entrenamiento o actualización de parámetros de la RBM mediante el Algoritmo 2.

Algoritmo 2 Actualización de parámetros RBM binaria

Entrada: unidades visibles \mathbf{v}_0 , tasa de aprendizaje ϵ , pesos iniciales \mathbf{W} , umbral unidades visibles \mathbf{b} y umbral unidades ocultas \mathbf{c}

Salida: parámetros \mathbf{W} , \mathbf{b} y \mathbf{c} actualizados

- 1: **para toda** unidad oculta j **hacer**
 - 2: Calcular $p(\tilde{\mathbf{h}}_{0j} = 1 | \mathbf{v}_0)$
 - 3: Obtener muestra $\tilde{\mathbf{h}}_{0j}$ de $p(\tilde{\mathbf{h}}_{0j} | \mathbf{v}_0)$
 - 4: **fin para**
 - 5: **para toda** unidad visible i **hacer**
 - 6: Calcular $p(\tilde{\mathbf{v}}_{1i} = 1 | \tilde{\mathbf{h}}_0)$
 - 7: Obtener muestra $\tilde{\mathbf{v}}_{1i}$ de $p(\tilde{\mathbf{v}}_{1i} | \tilde{\mathbf{h}}_0)$
 - 8: **fin para**
 - 9: **para toda** unidad oculta j **hacer**
 - 10: Calcular $p(\tilde{\mathbf{h}}_{1j} = 1 | \tilde{\mathbf{v}}_1)$
 - 11: **fin para**
 - 12: Actualizar $\mathbf{W} \leftarrow \mathbf{W} + \epsilon \left(\tilde{\mathbf{h}}_0 \mathbf{v}_0^T - p(\tilde{\mathbf{h}}_1 = 1 | \tilde{\mathbf{v}}_1) \tilde{\mathbf{v}}_1^T \right)$
 - 13: Actualizar $\mathbf{b} \leftarrow \mathbf{b} + \epsilon (\mathbf{v}_0 - \tilde{\mathbf{v}}_1)$
 - 14: Actualizar $\mathbf{c} \leftarrow \mathbf{c} + \epsilon \left(\mathbf{h}_0 - p(\tilde{\mathbf{h}}_1 = 1 | \tilde{\mathbf{v}}_1) \right)$
-

Finalmente Tieleman en [59] sugiere una variante al algoritmo de Divergencia Contrastante en donde cada iteración de CD-k se ejecuta utilizando como punto de inicio $\tilde{\mathbf{v}}_1$ de la iteración anterior, dando un efecto *persistente* en el muestreo de Gibbs en lugar de iniciar de un conjunto de unidades visibles dado por el entrenamiento. Esta variante es llamada *Divergencia Contrastante Persistente*.

2.7.5. Alias de verosimilitud

Es posible utilizar un indicador o alias con el objetivo de evaluar la similitud entre la distribución que describe el modelo de probabilidad \tilde{p} de la RBM y la distribución de probabilidad p que describen las unidades visibles utilizadas en el entrenamiento. Dos de estos indicadores son la *entropía cruzada* y la *pseudo-verosimilitud*.

Entropía cruzada

La entropía cruzada (CE) [9, 8] provee una medición del error entre el conjunto de unidades visibles $\tilde{\mathbf{v}}$ generadas por el modelo y las unidades visibles \mathbf{v} de entrenamiento. Es utilizada como el *costo de reconstrucción* [29] en RBMs que utilizan generalmente divergencia contrastante

$$CE = - \sum_{l=1}^L v_l \log(\tilde{p}(v_l)) + (1 - v_l) \log(1 - \tilde{p}(v_l)) \quad (2.39)$$

con L la cantidad de unidades visibles en la muestra, $\tilde{p}(v_l)$ la probabilidad de la unidad visible generada mediante CD-k para la unidad visible v_l y donde se ha considerado que para el caso binario la probabilidad del conjunto de entrada durante el entrenamiento $p(\mathbf{v}) = \mathbf{v}$.

Para un conjunto de entrenamiento con T muestras es posible obtener la CE del conjunto de entrenamiento mediante

$$CE_{total} = \sum_{i=1}^T CE_i^{\mathbf{v}_i} \quad (2.40)$$

Pseudo-verosimilitud

La pseudo-verosimilitud (PL) permite evaluar si el conjunto de parámetros estimados por la RBM maximiza el producto de todas distribuciones condicionales $P(v_i | v_{-i})$ dado un conjunto de datos de entrenamiento [40].

$$PL(v) = \prod_i^N P(v_i | v_{-i}) \quad (2.41)$$

Es posible utilizar la expresión logarítmica con el objetivo de expresarlo como la suma de las probabilidades de cada bit v_i condicionado por el estado de los restantes bits [20, 28]

$$\log(PL(v)) = \sum_i^N \log(P(v_i | v_{-i})) \quad (2.42)$$

donde v_{-i} es el conjunto de bits correspondientes a las N unidades visibles excluyendo el valor del bit i , el cual para efectos del análisis se invierte [40]. No obstante al incrementar la cantidad de bits o unidades visibles este cálculo podría ser computacionalmente inadecuado ya que se deben calcular N probabilidades. Es posible utilizar una aproximación estocástica variando únicamente un bit i por muestra de unidades visibles tal que [17]

$$g = N \log P(v_i|v_{-i}) \quad (2.43)$$

$$\log PL(v) \approx E[g] \quad (2.44)$$

Sea v_φ el conjunto de unidades visibles con el bit i invertido, se estima $\log PL(v)$ con

$$\log PL(v) \approx (N \cdot \log \zeta(\mathcal{F}(v_\varphi) - \mathcal{F}(v))) \quad (2.45)$$

y encontrar el promedio de pseudo-verosimilitud del conjunto de entrenamiento con T elementos

$$costo \approx \frac{1}{T} \sum_{t=1}^T (N \cdot \log \zeta(\mathcal{F}(v_\varphi) - \mathcal{F}(v))) \quad (2.46)$$

2.7.6. Aprendizaje profundo

Las redes de aprendizaje profundo (DBN por sus siglas en inglés: Deep Belief Networks) extienden la arquitectura de las RBM a múltiples capas ocultas. Para esto durante la etapa de entrenamiento de la capa l los pesos de la capa $l - 1$ se mantienen fijos y se utiliza de entrada para la capa l las salidas de las unidades ocultas de la capa $l - 1$ [34]. Es decir, las unidades ocultas de $l - 1$ corresponden a las unidades visibles de la capa l tal y como se ilustra en la figura 2.8.

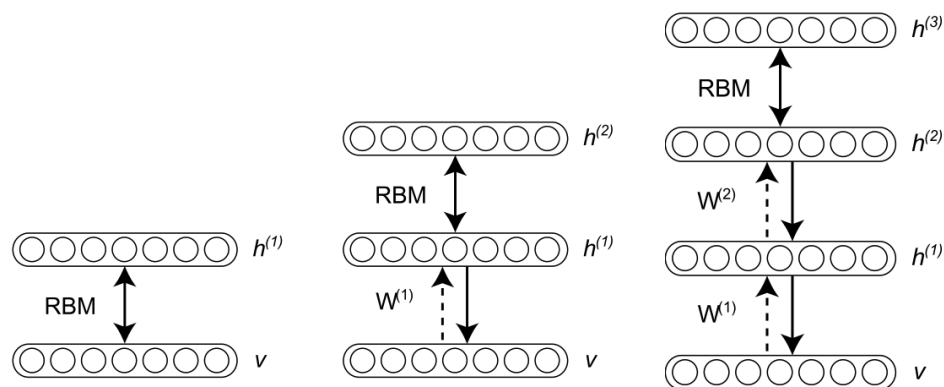


Figura 2.8: Arquitectura de aprendizaje profundo. Tomado de [36].

El entrenamiento de las DBN se realiza de forma segmentada, por partes y en una secuencia definida. La capa l es entrenada una vez establecidos y entrenados los pesos de la capa $l - 1$. Hinton [25] demostró la validez de este tipo de entrenamiento por capas. Krizhevsky [34] hace referencia a como si la segunda capa de pesos tiene la misma cantidad que la

primera y a su vez la inicialización de la segunda son los pesos de la primera la arquitectura es capaz de mejorar la verosimilitud o probabilidad de los datos $p(\mathbf{v})$ descrita por el modelo de una sola capa.

Las DBN son capaces de aprender jerarquías de *factores de variación* o *características* del conjunto de datos para los cuales no necesariamente se cuenta con un entendimiento analítico [6]. Estas logran aprender características o factores de alto nivel formados por la composición de factores de más bajo nivel mediante abstracción automática que le permite al sistema aprender mapeos complejos sin depender de factores indicados manualmente.

2.8. Trabajos anteriores de modelado con RBM

El uso de RBMs y DBNs para el modelado de formas es una propuesta recientemente explorada por la comunidad científica con el objetivo de aprovechar principalmente sus propiedades generativas. Eslami et al. [16] hacen énfasis en la importancia de un modelo de forma robusto con el objetivo de obtener un mayor rendimiento en tareas de segmentación. Para esto sugieren un tipo de DBN denominado *ShapeBM* el cual presenta una arquitectura, un algoritmo y una técnica de muestreo capaz de hacer uso de la información global y local disponible en máscaras binarias previamente alineadas y de tamaño fijo de caballos principalmente (base de datos de Weizmann [10]). Definen el realismo en las imágenes segmentadas resultantes y la capacidad de generalización como los criterios utilizados para clasificar a un modelo de forma como robusto.

Además destaca cómo las representaciones de forma basadas en campos aleatorios de Markov (MRFs) son capaces solamente de considerar restricciones locales de forma, como la conectividad entre píxeles vecinos, el ángulo de las patas del frente de un caballo, etc. y cómo técnicas basadas únicamente en el algoritmo de forma media (mean shape) en combinación con PCA para representar el modelo de forma solo son capaces de considerar restricciones globales y no representan formas con distribución multimodal [12].

ShapeBM impone restricciones adicionales de conectividad y capacidad a las DBNs, dando como resultado una simplificación en la cantidad de parámetros necesarios para modelar la forma expuesta por el conjunto de datos de entrenamiento así como una disminución en el tamaño de dicho conjunto. Para esto utiliza dos capas ocultas \mathbf{h}^1 y \mathbf{h}^2 y $N \times N$ unidades visibles \mathbf{v} que corresponden a los píxeles de la imagen, sin embargo las unidades visibles son organizadas en subconjuntos que se traslapan b píxeles y que son conectadas a un subconjunto de unidades ocultas en \mathbf{h}^1 . Esto permite que las unidades presentes en \mathbf{h}^1 modelen las restricciones locales y las unidades presentes en \mathbf{h}^2 modelen las globales.

Esta consideración de diseño y el entrenamiento por capas realizado por Eslami le permiten a ShapeBM presentar capacidades de generalización, realismo y categorización incluso cuando la imagen presente en las unidades visibles está incompleta [16]. No obstante, ShapeBM no posee invariabilidad a la escala ni a la rotación o traslación ya que requiere un alineamiento de la imagen a ingresar en las unidades visibles en sus etapas iniciales.

Por otra parte, Kae *et al.* [33] propone una solución al problema de etiquetado de regiones en imágenes de caras, específicamente piel, cabello y fondo, mediante la combinación de campos aleatorios condicionales (CRFs) y RBMs. Los primeros proporcionan modelos con cualidades para la etiquetación de segmentos de imágenes, capaces de modelar la interacción local de regiones adyacentes, en este caso superpíxeles, y se hace uso de RBMs para el modelado de la forma global producida por la segmentación, donde el modelo aprendido por las unidades ocultas puede ser interpretado como atributos de la cara aprendidos por la RBM. Esta combinación es llamada modelo *GLOC* y utiliza superpíxeles en la capa visible con el objetivo de disminuir la cantidad de unidades visibles así como el procesamiento necesario, ya que se realiza una menor cantidad de operaciones.

El modelo *GLOC* logra un 25,41 % en la reducción del error comparado con el método CRF individual y además permite un etiquetado de diversos tipos de cabello, posiciones de cabeza, ropa e incluso etiquetado en regiones con obstrucciones, sin embargo al igual que ShapeBM este requiere una etapa inicial de alineamiento de la forma *completa* y por tanto comparten muchas de las desventajas ya que en una aplicación como la de las larvas esto requeriría haber segmentado previamente las larvas lo cual es el problema a solucionar en primera instancia.

Finalmente, Alvarado [2] explora el modelado de estructuras vermiformes basado en la utilización de máquinas restringidas de Boltzmann, con el objetivo de soportar modelos generativos de formas. Para esto utiliza cadenas de códigos diferenciales (DCC) [42] [38] con un tamaño de paso fijo para representar la forma, dando como resultado invariabilidad a la rotación y al corrimiento. A su vez, este menciona la problemática existente con representaciones de formas mediante un vector de puntos o marcas de referencia localizadas en el perímetro o borde de la forma (ASM) ya que esta representación varía ante un cambio de escala, traslación o rotación y además requiere una correspondencia entre la posición de cada uno de los puntos de referencia y sus índices en los vectores de formas por lo cual necesita un alineamiento previo y esto no siempre es posible. No obstante, este tipo de representaciones son robustas al ruido, lo cual es deseado.

Alvarado propone la representación de forma mediante el uso de DCC, donde cada cambio de dirección de los puntos de referencia sobre el contorno de la forma es codificado tal que a lo largo contorno, solo una secuencia de códigos es necesaria y donde cada elemento de la cadena alinea su código C_o con la dirección previa. Dicha codificación se conforma por 32 bits, donde cada bit representa una dirección y por lo tanto solo un bit puede estar activo a la vez. La RBM es modificada para generar una salida de este tipo, dando como resultado una representación binaria para las unidades visibles de la RBM que facilita su modelado y ajuste [19]. Como consideración adicional utiliza una cadena de códigos lo suficientemente larga (32 elementos) para rodear al menos dos veces el tamaño más largo esperado en la forma con lo cual se busca establecer un patrón periódico que pueda ser aprendido por la RBM.

La solución propuesta por Alvarado se basa en la generación de una hipótesis de segmentación que posteriormente es ajustada a la información de la imagen mediante un proceso

iterativo que involucra el modelo estocástico que fue entrenado en la RBM previamente.

Los resultados obtenidos por Alvarado demuestran cómo el uso de un conjunto de entrenamiento con corrimiento cíclico mejora la estabilidad en las cadenas generadas por la RBM a través de las iteraciones y como la degradación de puntos a través de las iteraciones siempre está presente; sin embargo la misma disminuye al aumentar las iteraciones y es inversamente proporcional a la cantidad de unidades ocultas utilizadas en la RBM.

Alvarado concluye que el uso de DCC para la representación de forma no es adecuado para procesos generativos con RBMs y DBNs y como las mismas no son capaces de ajustar el factor periódico presente en la representación de forma, por lo tanto recomienda una investigación de nuevas representaciones capaces de incorporar mayor información de los patrones de entrada con el objetivo de hacer mayor uso tanto de la información global como local de la imagen que permitan aprovechar las capacidades generativas de las RBMs.

Capítulo 3

Modelado de forma de larvas

La presente propuesta para modelado de forma se basa en el diseño de una representación que involucre información local y global de la larva, tal que dicha representación pueda ser modelada estadísticamente mediante una máquina restringida de Boltzmann entrenada con un conjunto de imágenes de larvas reales. La figura 3.1 ilustra el diagrama de bloques del sistema propuesto.

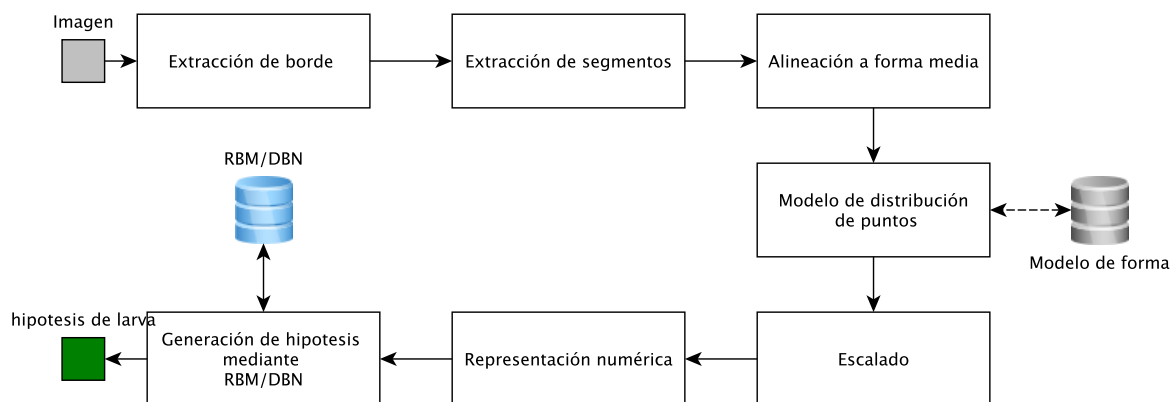


Figura 3.1: Diagrama de bloques para el modelado y reconstrucción de larvas

La creación del modelo de forma utiliza la información contenida en el contorno de la larva, en oposición a los métodos que utilizan la rejilla completa de píxeles [16, 33]. La etapa inicial del método localiza τ posibles larvas individuales y las describe mediante su contorno. Dicho contorno es posteriormente fraccionado en segmentos de tamaño fijo η tal que existe un traslape $\eta/2$ entre dos segmentos consecutivos, donde la cantidad de segmentos χ a utilizar para representar cada una de las larvas es un parámetro dado por el usuario y debe mantenerse constante a lo largo del algoritmo. De este parámetro depende la cantidad de unidades visibles a utilizar en la RBM. Para los casos en los cuales la larva es más pequeña que χ segmentos se utiliza redundancia cíclica, es decir, se repasa el contorno hasta alcanzar el número deseado de segmentos.

Encontrados todos los segmentos del conjunto de imágenes disponibles estos se alinean

mediante el análisis de Procrustes al segmento de forma media y se realiza una transformación de cada uno de ellos al espacio de componentes principales que describe al conjunto, permitiendo de esta forma una reducción en la dimensionalidad del segmento y del vector $\mathbf{v}_i^p; i \in \{0, \dots, \tau\}$, que representa a una larva completa. Para utilizar cada uno de estos vectores \mathbf{v}_i^p que representan larvas completas como entrada para la RBM se realiza un mapeo de cada uno de los k_i pesos que representan un segmento a una escala representada por 10 bits. Dos tipos de mapeos son evaluados, el lineal y el mapeo por blanqueo de datos en conjunto con la función de error. El vector resultante finalmente debe ser transformado a su representación binaria para ser presentado en las unidades visibles de la RBM, la cual es binaria también. Dos tipos de transformaciones son evaluadas, la binaria y la transformación a código Gray. La figura 3.2 presenta el diagrama de flujo de la estrategia anteriormente mencionada, dando como resultado un modelo de forma constituido por el conjunto de pesos \mathbf{W} y los umbrales \mathbf{c} y \mathbf{d} que describen las formas de las larvas.

Una vez creado el modelo este es utilizado para evaluar un contorno hipótesis de larva el cual puede corresponder a una larva del conjunto de evaluación generado en etapas previas al entrenamiento, a una larva incompleta o a una sección con traslape de larvas. La larva es presentada a la RBM en sus unidades visibles, las cuales son utilizadas para obtener una muestra de los pesos \mathbf{h} de la RBM y finalmente a partir de estos se obtiene una muestra \mathbf{v}^g que corresponde a la larva individual generada por la RBM que posteriormente debe ser ajustada a la imagen original. La figura 3.3 muestra un diagrama de flujo de los pasos involucrados en la evaluación o reconstrucción de forma.

En el presente trabajo se utilizan las larvas del subconjunto de evaluación creado en etapas previas al entrenamiento y por ende se cuenta con las matrices de transformación que permiten ajustar la larva a la imagen real, esto solo con el objetivo de mostrar las larvas generadas sobre la imagen original.

En lo restante de esta sección los detalles involucrados en el desarrollo de la solución propuesta son explicados, siendo posible dividir el diseño en las siguientes etapas:

- Extracción del contorno de larva
- Fragmentación del contorno
- Adaptación de segmentos a representación de forma
- Creación del modelo de forma de larva
- Reconstrucción de larvas

3.1. Extracción del contorno de larva

Las larvas contenidas en las imágenes pueden variar su tamaño, ángulo, forma y posición dentro de la imagen. Sumado a esto en algunas situaciones existe oclusión de larvas y en caso de imágenes que conforman un video las regiones con oclusión pueden variar de ubicación dentro de la imagen a lo largo del tiempo. La figura 3.4 muestra una de las imágenes de larvas del conjunto de datos utilizado, donde todo aquello considerado no

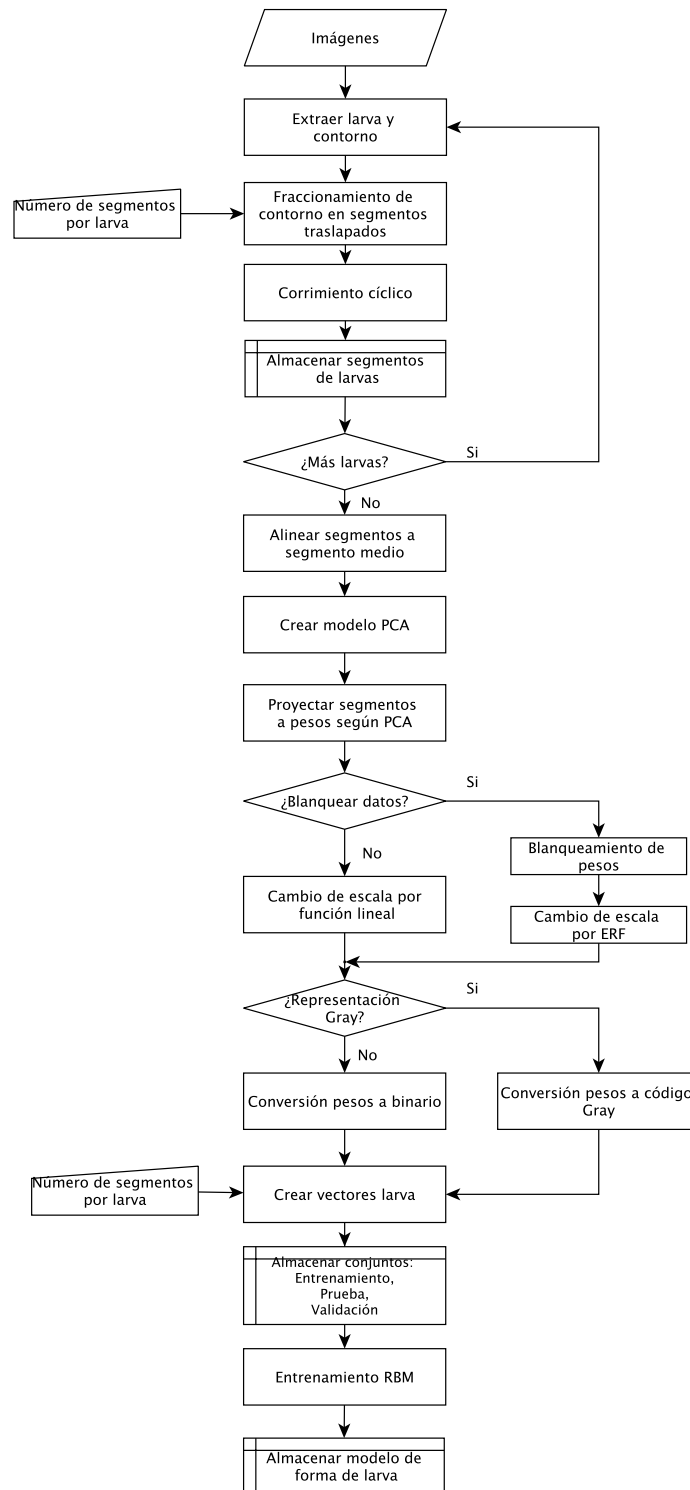


Figura 3.2: Diagrama de flujo para la creación del modelo de forma

larva es asignado a una baja intensidad (negro).

En el presente trabajo se utiliza la solución propuesta por Fiaschi en [35] para la ubicación inicial de larvas. Fiaschi realiza un etiquetado de las regiones que considera larvas en las imágenes brindadas por HCI y asigna un número de etiqueta a cada región, el cual

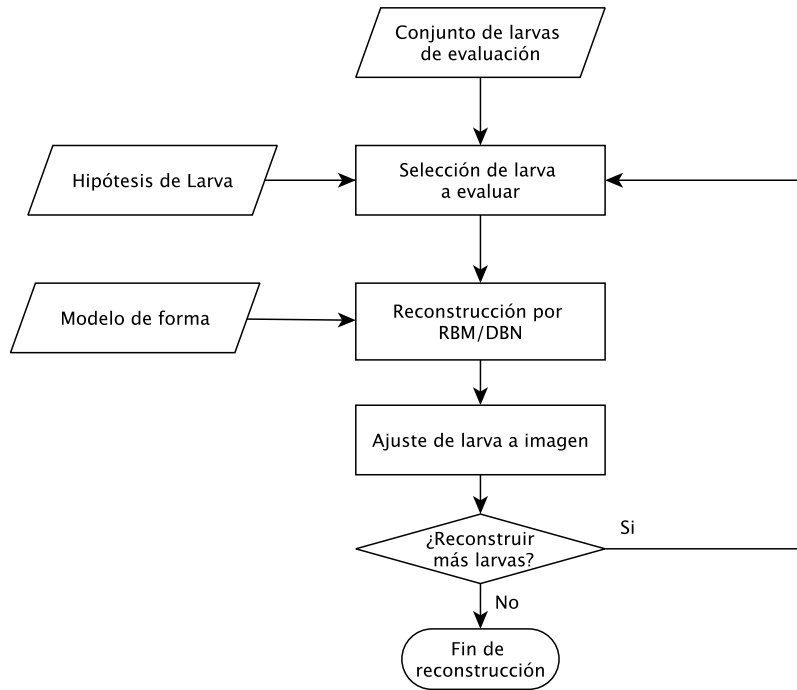


Figura 3.3: Diagrama de flujo para la reconstrucción de larvas

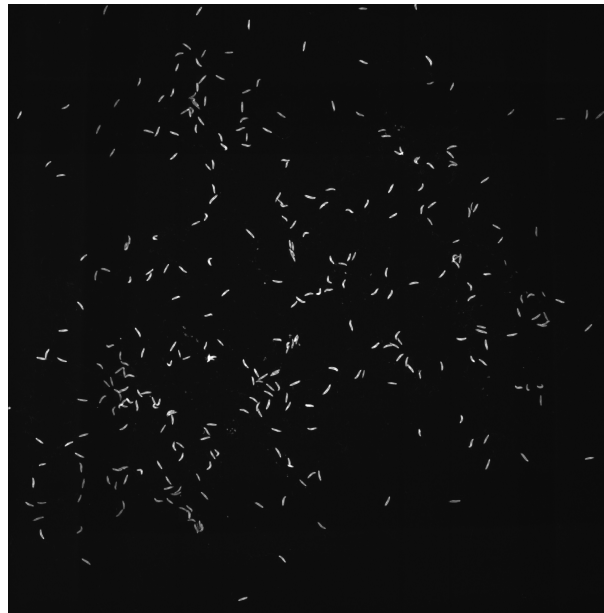


Figura 3.4: Imagen de larvas [35]

corresponde a un estimado de la cantidad de larvas presentes en la región etiquetada.

Este método no permite ubicar las larvas individuales en regiones con más de dos larvas y por tanto para el desarrollo del entrenamiento de la RBM se consideran únicamente aquellas regiones o áreas con etiqueta *uno*. Estas regiones son extraídas de la imagen original y posteriormente sus bordes, los cuales son descritos como la secuencia de pares ordenados $p^{(x,y)}$ alrededor de la región de la larva. Es posible representar dicho contorno

como el vector \mathbf{v}^c de $2N$ dimensiones

$$\mathbf{v}^c = [p_1, p_2, \dots, p_N]^T \quad (3.1)$$

y al conjunto de contornos de τ larvas encontrados como la matriz Γ^c

$$\Gamma^c = \begin{bmatrix} \mathbf{v}_0^{cT} \\ \mathbf{v}_1^{cT} \\ \vdots \\ \mathbf{v}_\tau^{cT} \end{bmatrix} \quad (3.2)$$

Cada contorno \mathbf{v}^c se fragmenta en χ segmentos de tamaño η tal y como se ilustra en la figura 3.5. Se realiza una fragmentación de segmentos traslapados $\lfloor \eta/2 \rfloor$ con $\eta = 9$ puntos $p^{(x,y)}$, es decir, cada segmento S_i es representado por 18 dimensiones y posee un traslape con el segmento anterior de 4 puntos $p^{(x,y)}$.

De esta forma cada segmento contiene información que los relaciona con el segmento S_{i-1} anterior con el objetivo de incrementar las capacidades de predicción o generación de la RBM. Para cada larva presentada en el entrenamiento se espera que la RBM aprenda dicha relación generando un patrón o factor global de variación que describe la conectividad de segmentos en la forma de la larva.

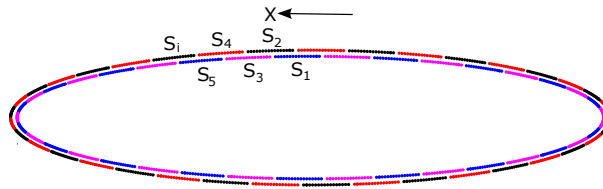


Figura 3.5: Fragmentación de contorno de larva

Debido a que la RBM cuenta con una cantidad constante de entradas es necesario representar cada una de las larvas con una misma cantidad χ de segmentos. En el presente trabajo se utiliza $\chi = 20$ ya que describe el mayor tamaño de larvas encontradas en el conjunto de imágenes de HCI. En los casos en los que la larva no posee 20 segmentos se continua recorriendo el contorno hasta completar la cantidad necesaria, causando una periodicidad de segmentos en larvas de tamaño menor a χ .

Es posible representar al conjunto de segmentos que describen a una larva mediante el vector \mathbf{v}^s

$$\mathbf{v}^s = [S_1, S_2, \dots, S_\chi]^T \quad (3.3)$$

y de forma similar al conjunto de larvas mediante

$$\mathbf{\Gamma}^s = \begin{bmatrix} \mathbf{v}_0^{sT} \\ \mathbf{v}_1^{sT} \\ \vdots \\ \mathbf{v}_\tau^{sT} \end{bmatrix} \quad (3.4)$$

Por otra parte, en [2] Alvarado reporta que el uso de un corrimiento cíclico sobre el contorno de la larva para la generación de muestras de entrenamiento permite extender el conjunto de entrenamiento tal que la RBM es entrenada con larvas sintéticas que inician en puntos arbitrarios, mejorando así la estabilidad del modelo aprendido por la RBM. En el presente solución se adopta el mismo criterio para realizar el entrenamiento, realizando un corrimiento de 15 puntos $p^{(x,y)}$ en cada larva hasta finalizar el contorno.

3.2. Adaptación de segmentos a representación de forma

El uso de 20 segmentos de 18 dimensiones por larva implica que la RBM requiere de $20 \times 18 \times \varphi$ unidades visibles para analizar la forma de la larva, donde φ corresponde al número de bits utilizados para representar cada dimensión. Para un valor de $\varphi = 10$ bits se necesitan 3600 unidades visibles y un número similar de unidades ocultas para poder modelar la distribución conjunta de las entradas. Considerando que mientras más unidades de entrada, más recursos computacionales son necesarios para el entrenamiento y más riesgos de sufrir las consecuencias de “la maldición de la dimensionalidad”. [8], es necesaria una etapa de adaptación de la representación de forma a la RBM cuyo objetivo principal es la reducción de la dimensionalidad del problema manteniendo invariable la información contenida en la representación de larva, esta adaptación es descrita en las siguientes subsecciones.

3.2.1. Alineamiento de segmentos

La reducción de la dimensionalidad del problema es realizada mediante la creación un *modelo de distribución de puntos* [55] basado en el análisis de componentes principales. Para esto es necesario realizar un alineamiento inicial de los segmentos obtenidos durante la extracción de larvas de las imágenes consideradas para la creación del conjunto de datos a utilizar. El alineamiento es realizado mediante el análisis de Procrustes (algoritmo 1). Sea la matrix Ψ el conjunto de formas a alinear

$$\Psi = \begin{bmatrix} \mathbf{S}_1^{(0)T} \\ \vdots \\ \mathbf{S}_\chi^{(0)T} \\ \vdots \\ \mathbf{S}_1^{(\tau)T} \\ \vdots \\ \mathbf{S}_\chi^{(\tau)T} \end{bmatrix} \quad (3.5)$$

donde cada fila de Ψ corresponde a un segmento de 9 puntos (x, y) representado ahora como un vector de 18 dimensiones

El conjunto resultante Ψ^a alineado al segmento medio \bar{S} es posteriormente analizado con PCA. Como resultado del análisis de Procrustes además se obtienen indirectamente las matrices de transformaciones isométricas \mathcal{T}_i^j para cada segmento, con i el segmento a transformar y j la larva correspondiente.

3.2.2. Análisis de componentes principales de segmentos

El análisis de componentes principales es aplicado a la matriz Ψ^a con el objetivo de encontrar la matriz \mathbf{P} tal que

$$\Psi^a \mathbf{P} = \Psi^\rho \quad (3.6)$$

Para esto se encuentran los vectores propios ρ_i ; $i \in \{1, \dots, 2\eta\}$, de $\Psi^a \Psi^{aT}$ mediante SVD [3, 51] y los valores propios λ_i correspondientes a cada vector.

$$\mathbf{P} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{2\eta} \end{bmatrix} \quad (3.7)$$

$$\mathbf{v}^\lambda = [\lambda_1, \lambda_2, \dots, \lambda_{2\eta}]^T \quad (3.8)$$

Para la disminución de la dimensionalidad se utiliza un subconjunto con κ vectores de los 2η vectores propios disponibles, donde los vectores propios se ordenan de mayor a menor según su valor propio. Se utiliza para la selección de κ el criterio (2.4) con $\alpha = 0,85$, lo que resulta $\kappa = 3$, permitiendo obtener la proyección de los segmentos S_i ; $i \in \{0, \dots, \tau \times \chi\}$ del espacio de forma al espacio de pesos de PCA \tilde{S}_i y viceversa por medio de

$$\mathbf{P}^\kappa = \begin{bmatrix} \rho_1^T \\ \rho_2^T \\ \vdots \\ \rho_\kappa^T \end{bmatrix} \quad (3.9)$$

$$\tilde{\mathbf{S}}_i = (\mathbf{S}_i - \bar{\mathbf{S}})\mathbf{P}^{\kappa T} \quad (3.10)$$

$$\mathbf{S}_i \approx \tilde{\mathbf{S}}_i\mathbf{P}^\kappa + \bar{\mathbf{S}} \quad (3.11)$$

con $\tilde{\mathbf{S}}_i = [w_a^\rho, w_b^\rho, w_c^\rho]^T$ siendo w_a el peso de PCA correspondiente al primer vector propio, dando como resultado la nueva matriz de segmentos Ψ^ρ representados por 3 pesos de PCA.

$$\Psi^\rho = \begin{bmatrix} \tilde{\mathbf{S}}_1^{(0)T} \\ \vdots \\ \tilde{\mathbf{S}}_\chi^{(0)T} \\ \vdots \\ \tilde{\mathbf{S}}_1^{(\tau)T} \\ \vdots \\ \tilde{\mathbf{S}}_\chi^{(\tau)T} \end{bmatrix} \quad (3.12)$$

De esta forma con $\chi = 20$, $\varphi = 10$ y $\kappa = 3$ (3 pesos por segmento) se requieren 600 unidades visibles en la RBM, es decir, un 16,7% del tamaño de la representación completa.

3.2.3. Cambio de escala

La cantidad de unidades visibles necesarias en la RBM es definida por los parámetros χ , φ y κ que describen el tamaño de la representación de forma de larva. Es necesario convertir cada peso w_a^ρ , w_b^ρ y w_c^ρ a su representación binaria para finalmente ser presentado a la RBM. Esto implica que cada peso debe ser cuantificado a 2^φ valores. Para $\varphi = 10$ bits es posible utilizar 1024 valores y es necesario realizar un cambio de escala. La solución propuesta evalúa dos de ellos: el mapeo lineal y el cambio de escala utilizando la función de error y el blanqueo de datos.

Escala lineal

Para realizar un cambio de escala lineal se obtienen los valores mínimos y máximos entre todas las dimensiones a , b y c de $\tilde{\mathbf{S}}_i$ y se asignan a z_{min} y z_{max} respectivamente, y junto

con los valores $w_{min} = 0$ y $w_{max} = 1023$ se introducen en (2.13) y (2.14) para obtener los valores de α y β que permiten el mapeo sobre cada uno de los pesos de $\tilde{\mathbf{S}}_i$ por medio de

$$\tilde{w}_a = \alpha w_a^p + \beta \quad (3.13)$$

$$\tilde{w}_b = \alpha w_b^p + \beta \quad (3.14)$$

$$\tilde{w}_c = \alpha w_c^p + \beta \quad (3.15)$$

dando como resultado la matriz $\hat{\Psi}$ de segmentos $\hat{\mathbf{S}} = [\tilde{w}_a, \tilde{w}_b, \tilde{w}_c]^T$

$$\hat{\Psi} = \begin{bmatrix} \hat{\mathbf{S}}_1^{(0)T} \\ \vdots \\ \hat{\mathbf{S}}_\chi^{(0)T} \\ \vdots \\ \hat{\mathbf{S}}_1^{(\tau)T} \\ \vdots \\ \hat{\mathbf{S}}_\chi^{(\tau)T} \end{bmatrix} \quad (3.16)$$

Función de error y blanqueo de datos

El uso de un mapeo lineal permite realizar el cambio de escala necesario para la cantidad de valores a utilizar en la representación binaria (2^φ). Sin embargo, el uso del rango de valores disponibles en el mapeo depende de la varianza λ_i descrita por el peso w_i en una dimensión específica i . Para un conjunto de pesos con distribución gaussiana como la mostrada en la figura 2.3, si los datos poseen media cero (como los dados por PCA) y la varianza es pequeña con respecto a la escala disponible, el histograma resultante no se encuentra ecualizado y no se hace uso de toda la escala disponible.

Para esto, asumiendo una distribución gaussiana en cada dimensión de los pesos w_a , w_b y w_c se normaliza primeramente la varianza mediante el blanqueo de datos utilizando

$$\Psi'^T = \Psi^{\rho T} \Lambda^{-\frac{1}{2}} \quad (3.17)$$

donde

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_{\kappa=3} \end{bmatrix}$$

con λ_i el i -ésimo valor propio dado por PCA. Por último se utiliza el mapeo (2.17) para $\mu = 0$ y $\sigma = 1$

$$\tilde{w}_a = \psi \left(\frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{w'_a}{\sqrt{2}} \right) \right) \quad (3.19)$$

$$\tilde{w}_b = \psi \left(\frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{w'_b}{\sqrt{2}} \right) \right) \quad (3.20)$$

$$\tilde{w}_c = \psi \left(\frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{w'_c}{\sqrt{2}} \right) \right) \quad (3.21)$$

con $\psi = 1023$, logrando un mapeo al rango de valores deseado y una ecualización del histograma como se muestra en la figura 2.5b. Este mapeo permite evaluar el rendimiento de la RBM ante un uso optimizado del rango de valores para los pesos w_i , es decir, cuando el completo de unidades visibles se utiliza en la representación. El conjunto de segmentos $\hat{\mathbf{S}} = [\tilde{w}_a, \tilde{w}_b, \tilde{w}_c]^T$ se expresa nuevamente como matriz $\hat{\Psi}$.

3.2.4. Cambio de sistema numérico

Finalmente, es necesario realizar un cambio de sistema numérico para cada uno de los pesos de la matriz $\hat{\Psi}$. Como criterio de diseño cada peso es aproximado al entero más cercano y es representado por $\varphi = 10$ bits. Se estudia el rendimiento de la RBM para una transformación decimal binaria \mathcal{T}^b y una transformación a código gray \mathcal{T}^g de 10 bits. Para esto los segmentos con cambio de escala $\hat{\mathbf{S}}_i$; $i \in \{0, \dots, \chi\}$ que representan a una larva son organizados en un vector $\mathbf{v}^{\hat{\mathbf{S}}}$ conformado por $\chi = 20$ segmentos

$$\mathbf{v}^{\hat{\mathbf{S}}} = [\hat{S}_1, \hat{S}_2, \dots, \hat{S}_\chi]^T \quad (3.22)$$

y se realiza una transformación a binario o a código gray

$$\mathbf{v}^b = \mathcal{T}^b [\mathbf{v}^{\hat{\mathbf{S}}}] \quad (3.23)$$

$$\mathbf{v}^g = \mathcal{T}^g [\mathbf{v}^{\hat{\mathbf{S}}}] \quad (3.24)$$

Se evalúa la capacidad de la RBM para describir el modelo cuando las unidades visibles \mathbf{v}^l tienen como entrada \mathbf{v}^b ó \mathbf{v}^g .

3.3. Creación del modelo de forma de larva

Para la creación del modelo de forma se utiliza una RBM binaria y el conjunto de τ larvas \mathbf{v}^ℓ extraídas de la base de imágenes proporcionada por HCI.

$$\mathbf{\Gamma}^\ell = \begin{bmatrix} \mathbf{v}_0^{\ell T} \\ \mathbf{v}_1^{\ell T} \\ \vdots \\ \mathbf{v}_\tau^{\ell T} \end{bmatrix} \quad (3.25)$$

El conjunto $\mathbf{\Gamma}^\ell$ se divide en tres subconjuntos: entrenamiento \mathcal{C}^E , prueba \mathcal{C}^P y validación \mathcal{C}^V con tamaños indicados en la Tabla 3.1.

Tabla 3.1: Porcentaje de larvas presente en los conjuntos de prueba, entrenamiento y validación

Conjunto	Absoluto	Total de larvas [%]
\mathcal{C}^E	$\frac{\tau}{2}$	50
\mathcal{C}^P	$\frac{\tau}{4}$	25
\mathcal{C}^V	$\frac{\tau}{4}$	25

Cada larva \mathbf{v}_i^ℓ presente en \mathcal{C}^E se utiliza como entrada para entrenar la RBM utilizando el algoritmo 2. Cada unidad oculta h_j o visible v_j se obtiene mediante un muestreo de una distribución binomial [18, 64] $\mathcal{B}(1, p)$ con p dado por (2.31) y (2.32) respectivamente. Para esto se evalúa la aproximación del gradiente mediante divergencia contrastante y divergencia contrastante persistente.

Una vez realizado el entrenamiento de la RBM sobre el conjunto \mathcal{C}^E los parámetros de la RBM resultantes \mathbf{W} , \mathbf{b} y \mathbf{c} son almacenados como los parámetros del modelo de forma de larvas Θ^ℓ .

3.4. Reconstrucción de larvas

La reconstrucción de la larva parte de un vector de representación de larva \mathbf{v}^δ correspondiente a una hipótesis de contorno que es presentado ante la RBM (previamente entrenada) como vector de unidades visibles $\mathbf{v}^\ell = \mathbf{v}^\delta$. Por medio de la probabilidad conjunta p de unidades aprendidas por la RBM durante el entrenamiento se genera una hipótesis de corrección \mathbf{v}^Ω , dicha hipótesis se basa en la probabilidad p descrita por el conjunto de parámetros del modelo de forma Θ^ℓ , es decir, mediante los valores de \mathbf{W} , \mathbf{b} y \mathbf{c} aprendidos, los cuales en esta etapa no cambian.

Finalmente, a partir de la hipótesis de corrección \mathbf{v}^Ω es posible aplicar el proceso inverso descrito en las secciones anteriores del presente capítulo con el fin de obtener el conjunto de puntos $p^{(x,y)}$ que ajustan la larva sobre la imagen.

3.4.1. Generación de larvas por RBM

Los casos de larvas incompletas se representan como vectores $\mathbf{v}^{\hat{s}}$ compuestos por una cantidad ϖ de segmentos $\hat{\mathbf{S}} = [\tilde{w}_a, \tilde{w}_b, \tilde{w}_c]^T$, donde $\varpi < \chi$.

Debido a que la RBM cuenta con una cantidad fija de unidades visibles es necesario completar el vector de segmentos con $\delta = \chi - \varpi$ nuevos segmentos. Para esto se utiliza la representación \hat{S}_μ correspondiente al PCA con cambio de escala ($\varphi = 10$) y blanqueo del segmento medio dado por el análisis de Procrustes

$$\hat{\mathbf{S}}_\mu = [511, 511, 511]^T \quad (3.26)$$

es posible representar el vector de larva incompleta como

$$\mathbf{v}^{\hat{v}} = \mathcal{T}^{b \vee g} \left[[\hat{S}_1, \hat{S}_2, \dots, \hat{S}_\varpi, \hat{S}_{\mu_1}, \dots, \hat{S}_{\mu_\delta}]^T \right] \quad (3.27)$$

y obtener la hipótesis de corrección \mathbf{v}^Ω utilizando (2.31) y (2.32), esta hipótesis contiene la reconstrucción de los δ segmentos faltantes de acuerdo a la probabilidad conjunta principalmente descrito por los ϖ segmentos de larva original.

Por otra parte, si el contorno de larva describe dos o más larvas este es presentado de igual forma como hipótesis de larva $\mathbf{v}^{\hat{v}}$ a la RBM. Debido a que la distribución con la que la RBM es entrenada describe a larvas individuales la hipótesis de corrección \mathbf{v}^Ω corresponde a una larva individual.

3.4.2. Ajuste de larva a la imagen

Para el análisis de hipótesis de larvas $\mathbf{v}^{\hat{v}}$ de imágenes reales en los cuales no se conoce si la hipótesis corresponde a una larva real, un algoritmo debe ser concebido que permita ajustar la hipótesis de corrección \mathbf{v}^Ω a la imagen.

No obstante, para los objetivos del presente trabajo se utilizan larvas del conjunto de prueba \mathcal{C}^P , para los cuales se cuenta con las matrices de transformación isométrica \mathcal{T} de cada segmento y por tanto pueden ser utilizadas para ajustar a la imagen cada uno de los segmentos que componen \mathbf{v}^Ω mediante el algoritmo 3.

Algoritmo 3 Ajuste de hipótesis de corrección a imagen

Entrada: Hipótesis de corrección \mathbf{v}^Ω , transformación \mathcal{T}^ϱ utilizada (\mathcal{T}^g ó \mathcal{T}^b), segmento medio del conjunto de larvas \bar{S} , matriz de componentes principales \mathbf{P}^κ , vector de varianzas o valores propios \mathbf{v}^λ , cantidad de segmentos por larva χ , matrices de transformación isométrica \mathcal{T} e Imagen.

Salida: Vector de contorno \mathbf{v}^c correspondiente a \mathbf{v}^Ω e indicación sobre la imagen.

- 1: convertir hipótesis \mathbf{v}^Ω a un vector decimal mediante $\mathcal{T}_{g\sqrt{b}}^{-1}[\mathbf{v}^\Omega]$
 - 2: **para todo** segmento en $\mathbf{v}^{\Omega'}$ **hacer**
 - 3: Desblanqueo datos
 - 4: Mapeo inverso de cambio de escala
 - 5: Mapeo del espacio de componentes principales al espacio de forma utilizando (3.11)
 - 6: Aplicar matriz de transformación isométrica \mathcal{T} correspondiente.
 - 7: Adjuntar segmento a vector de contorno \mathbf{v}^c .
 - 8: **fin para**
 - 9: Dibujar vector de contorno \mathbf{v}^c sobre la imagen
-

Dando como resultado el vector de contorno \mathbf{v}^c producido por la RBM indicado sobre la imagen correspondiente.

Capítulo 4

Análisis de Resultados

En este capítulo se presentan los resultados obtenidos para cada uno de los módulos desarrollados e implementados en el presente trabajo para el modelado de forma de larvas utilizando máquinas restringidas de Boltzmann. Se realiza un análisis en etapas de los resultados obtenidos, divididas según el módulo bajo evaluación y finalmente se exponen los resultados obtenidos del funcionamiento completo del sistema propuesto. Para realizar las pruebas se utilizó una computadora con seis procesadores AMD Phenom II X6 1100T con una frecuencia de 3300MHz y 8GB de RAM en conjunto con una tarjeta de video NVIDIA 650 Ti de 768 procesadores CUDA con una frecuencia de 928MHz y 2GB de RAM. Se hace uso del procesamiento paralelo disponible mediante el uso de CUDA y la biblioteca Theano.

Lo restante del presente capítulo incluye las siguientes secciones:

- Extracción del borde de larva
- Extracción de segmentos
- Alineamiento de segmentos
- Modelo de distribución de puntos
- Escalado
- Representación numérica del conjunto de entrenamiento
- Creación del modelo de forma: entrenamiento de RBM
- Reconstrucción de larvas: generación de hipótesis
- Ajuste de larvas generadas a la imagen
- Oclusión

El análisis presentado se basa en el conjunto de datos proporcionado por el HCI [35]. El conjunto de datos incluye 1000 imágenes en alta resolución ($1590 \times 1600px$) como la ilustrada en la figura 1.1, segmentando con el método expuesto por Fiaschi et al. en [35], con el cual se localizan un total de 249633 larvas individuales, es decir un promedio de 249 larvas individuales por imagen.

4.1. Extracción del borde de larva

Para ilustrar los resultados de extracción de contorno se utiliza una imagen del conjunto de datos procesada por el algoritmo de Fiaschi [35] y se extrae el borde de las larvas identificadas como individuales. Los contornos obtenidos se demarcan sobre la imagen original para facilitar su interpretación. La imagen original y los bordes resultantes se muestran en la figura 4.1.

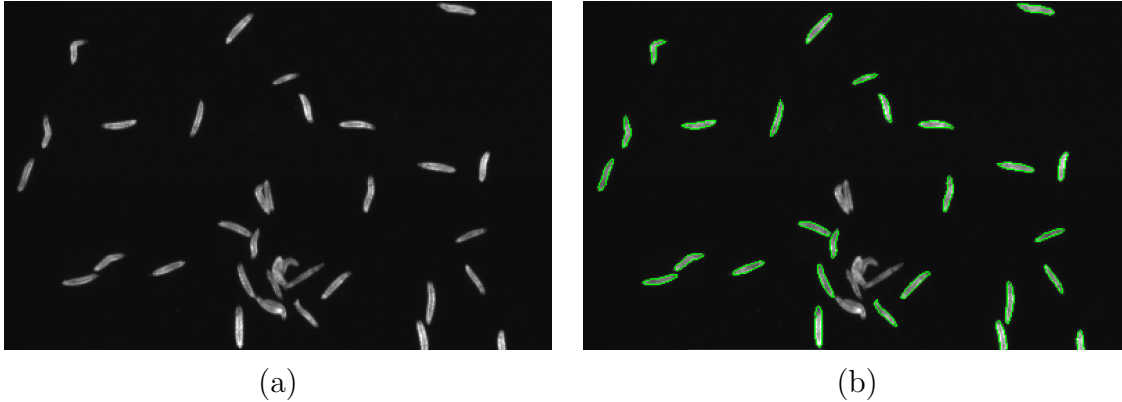


Figura 4.1: (a) Larvas, (b) Larvas y bordes extraídos.

Con el objetivo de disminuir el tiempo de entrenamiento se utiliza un 5% de las larvas obtenidas en el resto del documento, es decir 12481 larvas.

Con la totalidad de las larvas se calcula la probabilidad de cada longitud de contorno de larva. La distribución descrita se muestra en la figura 4.2. Se utiliza una longitud fija de 90 píxeles para la representación de larva, lo cual cubre a todas las larvas individuales presentes en el conjunto. Esto implica que para larvas con 45 píxeles (tamaño medio) de contorno existe una periodicidad de dos en la representación ya que se hace un recorrido cíclico hasta completar los 90 píxeles o la cantidad de segmentos deseados.

4.2. Extracción de segmentos

Para la fragmentación del contorno se utiliza $\chi = 20$ segmentos con $\eta = 9$ píxeles por segmento. Dichos segmentos se traslapan $\lfloor \eta/2 \rfloor$ permitiendo así cubrir los 90 píxeles de contorno máximo establecido. La figura 4.3 muestra un ejemplo con un contorno de longitud 63px, donde los puntos de color verde indican los inicios de los segmentos impares y los rojos los segmentos pares. El recorrido completo de la larva se realiza con 7 segmentos.

Para el total de larvas encontradas se obtuvo la probabilidad del tamaño de larva en segmentos traslapados sin considerar un recorrido cíclico sobre el borde, dando como resultado la distribución de la figura 4.4.

Se puede observar que el tamaño medio en segmentos es de 10, y que por tanto la media en la cantidad de periodos descritos por cada larva en un mismo vector de segmentos \mathbf{v}^c

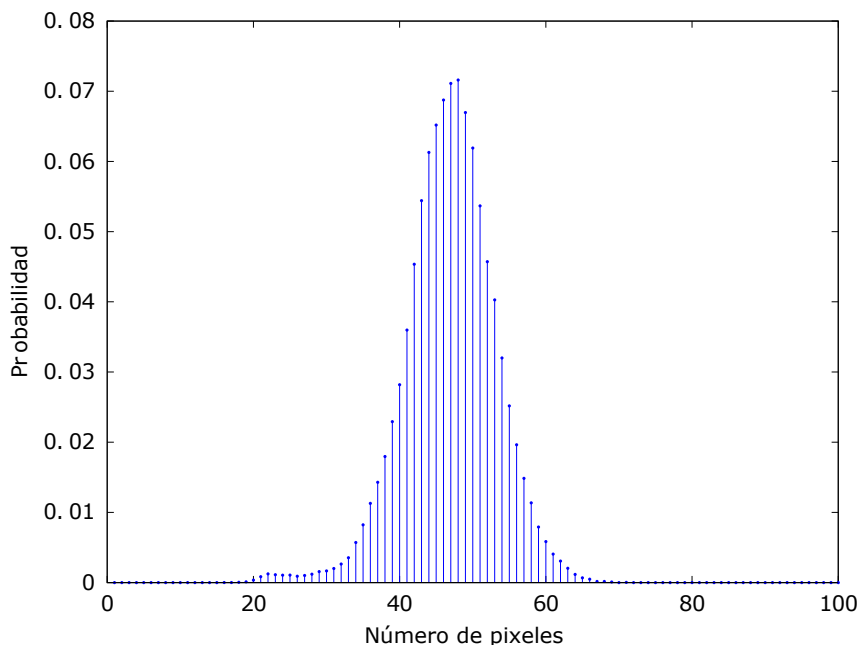


Figura 4.2: Probabilidad para el largo de contornos de larvas en imágenes de HCI

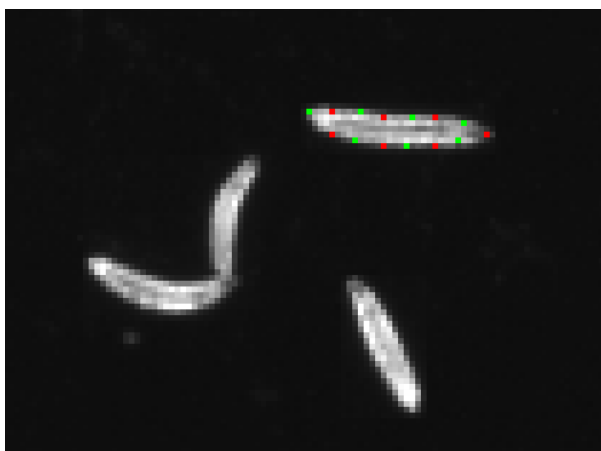


Figura 4.3: Segmentación del contorno. Los puntos verdes corresponden a inicios de S_i pares y los rojos a inicios de S_i impares

con $\chi = 20$ es de 2 periodos.

Por otra parte se realiza un corrimiento cíclico del punto de inicio de la larva con el objetivo de incluir en el conjunto de entrenamiento casos que se pueden presentar durante la evaluación de larvas reales, donde no es posible iniciar todas las larvas siempre desde el mismo punto. Se realizan un corrimiento de 15 píxeles del punto de inicio hasta alcanzar el final de la larva permitiendo incrementar el conjunto de entrenamiento en un factor de 3,38 tal y como se muestra en la Tabla 4.1.

Para $\chi = 20$ el conjunto de segmentos considerados resultante es de 947740 segmentos.

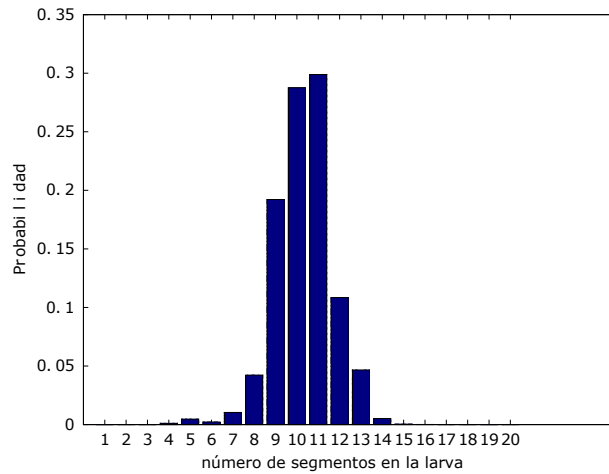


Figura 4.4: Probabilidad de tamaño de larva por cantidad de segmentos

Tabla 4.1: Cantidad de larvas consideradas antes y después del corrimiento cíclico

Larvas sin corrimiento	Larvas con corrimiento
12848	47387

4.3. Alineamiento de segmentos a la forma media

El análisis de Procrustes se realiza sobre los 947740 segmentos de 18 dimensiones ($2 \times \eta$), normalizando y alineando cada una de las formas o segmentos inicialmente al primer segmento de larva. Después de 4 iteraciones se obtiene como resultado el segmento medio \bar{S} mostrado en la figura 4.5 junto a los 20 segmentos alineados correspondientes a la primera larva del conjunto de formas.

El análisis de Procrustes permite alinear y normalizar los segmentos tal que el centroide de los elementos se encuentra aproximadamente en $(0,0)$ y todas las coordenadas de los segmentos se encuentran en el mismo rango de valores $[-2,2]$ aproximadamente. De esta forma cada larva es representada por 20 segmentos de 18 dimensiones alineados, dando como resultado una representación de larva de 360 dimensiones en decimal.

4.4. Modelo de distribución de puntos

Para la reducción de la dimensionalidad mediante el análisis de componentes principales se organizan los 947740 segmentos alineados en una matriz Ψ^a con dimensiones 947740×18 , para la cual se obtiene mediante *SVD* los componentes principales o vectores propios que permiten diagonalizar la matriz de covarianza, estos son ordenados tal que el primero de ellos sea el que posee un mayor valor propio y son mostrados en el apéndice A.

El conjunto de valores propios (varianzas) se ilustra en la figura 4.6.

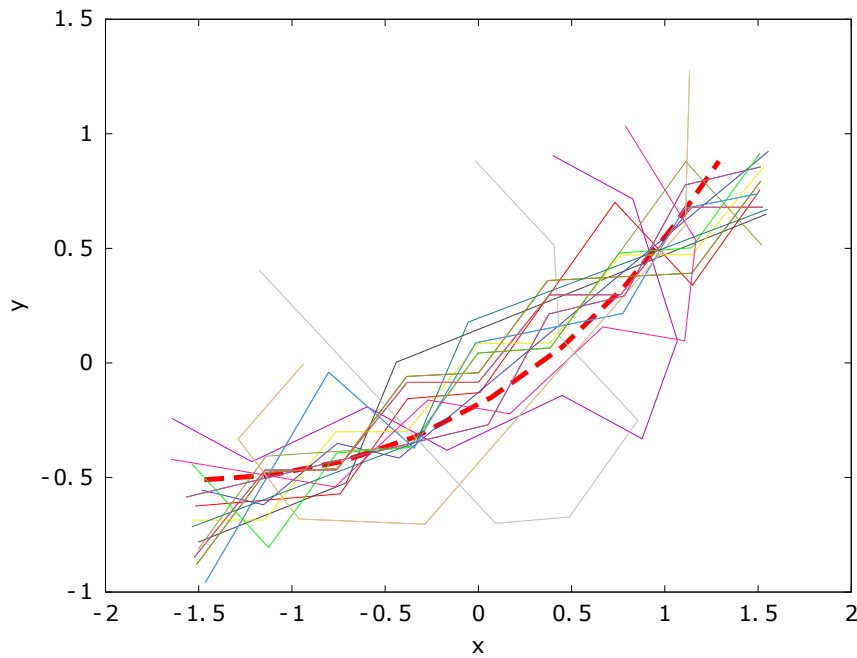


Figura 4.5: Segmentos alineados utilizando Procrustes. Segmento medio \bar{S} representado por la línea punteada de color rojo y $\chi = 20$ segmentos correspondientes a la primera larva

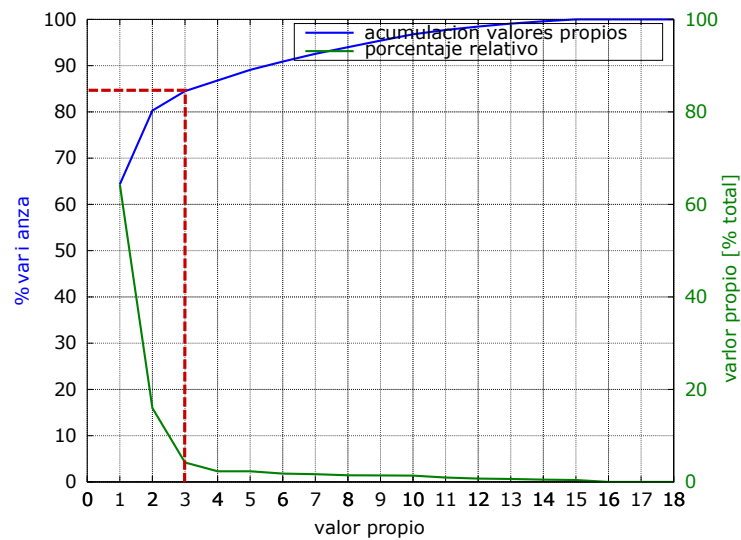


Figura 4.6: Variabilidad capturada según la cantidad de valores propios utilizados

donde además del porcentaje acumulativo de varianza se muestra el porcentaje relativo de cada peso sobre el total mediante

$$\text{porcentaje relativo}_i = \frac{\lambda_i}{\sum_{j=1}^m \lambda_j} \times 100 \quad (4.1)$$

De acuerdo a los valores de λ_i y utilizando (2.4) se obtiene que para representar un 85% de la variación descrita por las formas o segmentos es necesario utilizar al menos 3 de los

componentes principales y por tanto en el presente trabajo se utilizan ρ_1 , ρ_2 y ρ_3 en \mathbf{P} para realizar la reducción de dimensiones.

Para observar la variación descrita por cada uno de los tres vectores propios seleccionados estos se utilizan en conjunto con el segmento medio \bar{S} en

$$S'_j = \bar{S} + j\sqrt{\lambda_i}\rho_i, \quad \forall i \in \{1, \dots, 3\}, \quad \forall j \in \{-3, \dots, 3\} \quad (4.2)$$

donde para 6 valores de j distanciados una unidad se obtienen las variaciones mostradas en la figuras 4.7, 4.8 y 4.9 para ρ_1 , ρ_2 y ρ_3 respectivamente. Se ilustra que ρ_1 y ρ_2 realizan las mayores modificaciones en curvatura sobre el segmento.

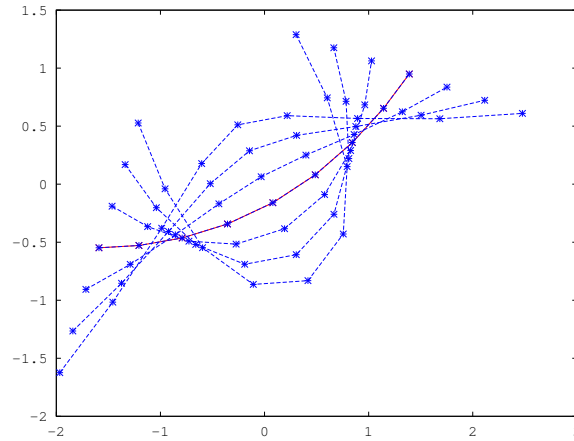


Figura 4.7: Variación causada por el primer vector propio ρ_1 sobre el segmento medio

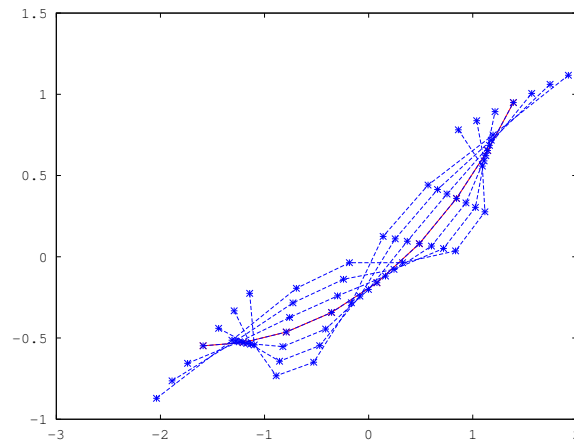


Figura 4.8: Variación causada por el segundo vector propio ρ_2 sobre el segmento medio

Se realiza la proyección de cada uno de los segmentos en Ψ^a del espacio de formas al espacio de vectores propios, dando como resultado la matriz Ψ^ρ de dimensiones 947740×3 constituida por segmentos $\tilde{S}_i = [w_a^\rho, w_b^\rho, w_c^\rho]$, permitiendo así reducir la cantidad de dimensiones en un 83 %.

La figura 4.10 ilustra la distribución que describen los pesos w_i^ρ , $i \in \{a, b, c\}$.

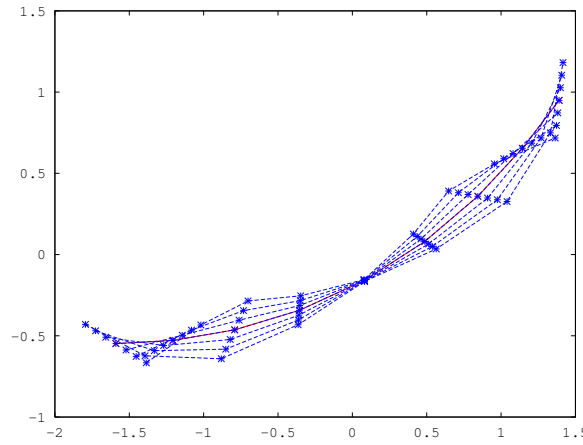


Figura 4.9: Variación causada por el tercer vector propio ρ_3 sobre el segmento medio

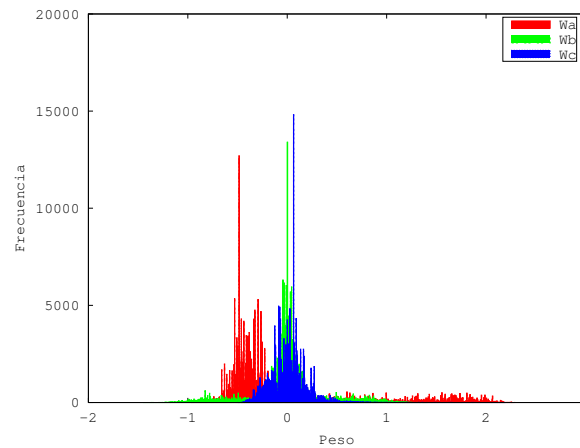


Figura 4.10: Histograma de pesos w_a , w_b y w_c resultantes de PCA para los segmentos para una celda de tamaño $\frac{5}{1024}$

Se observa que a pesar de tener una media cero en cada una de las dimensiones de PCA la mayor cantidad de los pesos para el caso de w_a no se encuentra en la media, por lo tanto la distribución no es gaussiana.

4.5. Escalado

4.5.1. Mapeo Lineal

Para $\varphi = 10$ bits para representar un peso es necesario utilizar una escala de 0 a 1023. Para determinar los valores de α y β en (2.13) y (2.14) se obtienen los valores mínimos y máximos para cada una de las dimensiones resultantes en PCA (Tabla 4.2).

Se utiliza una ecuación única de mapeo lineal para los tres pesos w_a , w_b y w_c utilizando el mínimo global de todas las dimensiones así como el máximo global. Seleccionando

Tabla 4.2: Valores mínimos y máximos en el espacio de pesos

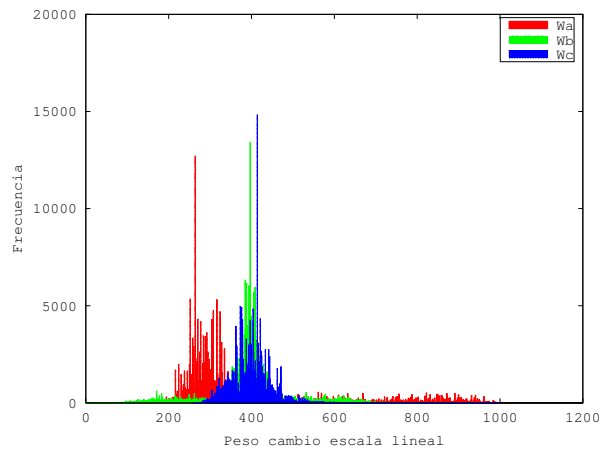
Dimensión	Min	Max
a	-0.969	2.300
b	-1.450	1.412
c	-0.500	2.193

$z_{min} = -1.450$, $z_{max} = 2.3$, $w_{min} = 0$ y $w_{max} = 1023$ se obtienen los valores de α y β de la Tabla 4.3.

Tabla 4.3: Parámetros de mapeo lineal de pesos

α	β
273.54	396.66

Una vez realizado el mapeo de cada uno de los segmentos se obtienen las distribuciones mostradas en la figura 4.11 para cada peso \tilde{w}_i , donde, debido a la naturaleza de la transformación lineal, la distribución de los pesos no cambia y se concentra en el centro del rango de valores posibles.

Figura 4.11: Histograma de pesos \tilde{w}_a , \tilde{w}_b y \tilde{w}_c para el cambio de escala lineal

4.5.2. Blanqueo de datos y función de error

Para realizar el cambio de escala y ecualizar el histograma se crea la matriz diagonal Λ con los 3 primeros valores propios

$$\Lambda = \begin{bmatrix} \sqrt{0,502} & 0 & 0 \\ 0 & \sqrt{0,125} & 0 \\ 0 & 0 & \sqrt{0,033} \end{bmatrix}$$

y se multiplica por la matriz de segmentos Ψ^{ρ} , dando como resultado la distribución de pesos ilustrada en la figura 4.12.

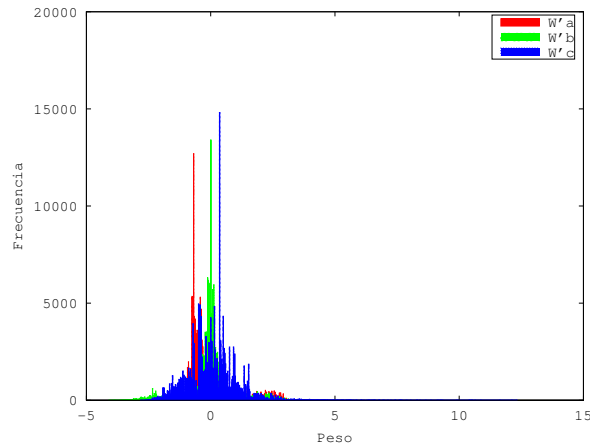


Figura 4.12: Histograma de pesos w'_i resultantes del blanqueo de datos

Al realizar el blanqueo para una distribución originalmente gaussiana un 99,74% de los pesos se debería encontrar en el rango de $[-3, 3]$ ya que la nueva desviación estandar es $\sigma = 1$ y la mayor cantidad de datos debería encontrarse en la media. Para el caso de las distribuciones obtenidas se puede observar que existen datos inclusive cerca de 5 y que la mayor cantidad de datos no está en la media confirmando así algunas de las diferencias existentes con respecto a una distribución normal.

Finalmente se realiza un cambio de escala mediante los mapeos basados en la función de error descritos en (3.19), (3.20) y (3.21). La figura 4.13 ilustra las distribuciones resultantes para los pesos \tilde{w}_i que componen los segmentos \hat{S}_i .

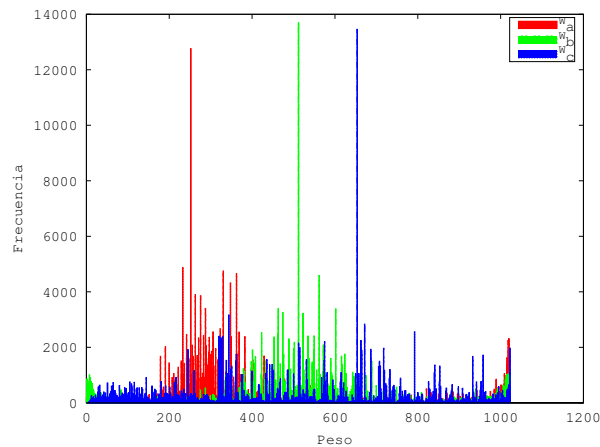


Figura 4.13: Histograma de pesos resultantes del cambio de escala con blanqueo y ERF

Es posible observar que el mapeo realizado con la función de error permite un uso homogéneo del rango de escala disponible. Un ejemplo de esto es la distribución observada en

la dimensión c , ya que en el caso lineal sus valores se encuentran en el intervalo $[300, 600]$ en contraste con el mapeo mediante blanqueo de datos y función de error donde se encuentran distribuidos en toda la escala, utilizando una mayor cantidad de bits para representar la dimensión. También se observa que al no estar los datos uniformemente distribuidos en la figura 4.12 la distribución resultante no es igual de uniforme como en la figura 2.5b.

4.6. Representación numérica del conjunto de entrenamiento

Para la creación del conjunto de entrenamiento de los RBM y DBN se realizan dos tipos de conversión de decimal a binario sobre cada uno de los pesos que componen los segmentos $\hat{\mathbf{S}} = [\tilde{w}_a, \tilde{w}_b, \tilde{w}_c]^T$. La conversión binaria, dando lugar a una matriz $\mathbf{\Gamma}^{\ell b}$ y la conversión a código Gray de 10 bits, produciendo la matriz $\mathbf{\Gamma}^{\ell g}$. Posteriormente los segmentos resultantes se reagrupan en vectores que describen una misma larva, el conjunto obtenido es descrito en la Tabla 4.4.

Tabla 4.4: Descripción del conjunto de representación de larvas

Número de larvas	Segmentos por larva	Pesos por segmento	Bits por peso	Bits por larva
47387	20	3	10	600

Seguidamente, el conjunto se divide en entrenamiento \mathcal{C}^E , prueba \mathcal{C}^P y validación \mathcal{C}^V según se muestra en la Tabla 4.5. Finalmente, el conjunto \mathcal{C}^E es seccionado en lotes o subconjuntos de 10 larvas, que son presentados a la RBM.

Tabla 4.5: Cantidad de larvas por conjunto

Conjunto	Número de larvas
\mathcal{C}^E	23693
\mathcal{C}^P	11847
\mathcal{C}^V	11847

4.7. Creación del modelo de forma: entrenamiento de RBM

Para la creación del modelo de forma se entrenan RBM y DBN con las configuraciones mostradas en la Tabla 4.6 utilizando un vector \mathbf{v}^ℓ de 600 unidades visibles.

Se evalúa el entrenamiento mediante la capacidad de describir la distribución conjunta de los 2369 lotes que conforman el conjunto de entrenamiento \mathcal{C}^E . Dicha evaluación se hace en la etapa de entrenamiento basado en la entropía cruzada para los casos en los que se utiliza divergencia contrastante no persistente y la pseudo-verosimilitud si se utiliza CD persistente (PCD). En ambos casos se considera el conjunto de entrenamiento y los vectores de unidades visibles generados por la RBM.

Tabla 4.6: Resultados de entrenamiento de RBM/DBN

Nombre	Unidades ocultas	Tipo de Mapeo	Representación	épocas	CD	Costo	Tiempo (min)
RBM1	128	lineal	Gray	600	CD-1	262	46.34
RBM2	8192	lineal	Gray	600	CD-1	2	419.33
RBM3	11400	lineal	Gray	600	CD-1	2	575.36
RBM4	128	ERF	Gray	600	CD-1	312	47.63
RBM5	8192	ERF	Gray	600	CD-1	5	414.22
RBM6	11400	ERF	Gray	600	CD-1	2	575.73
RBM7	8192	lineal	Binario	600	CD-1	2	417.31
DBN8	1200 14400	lineal	Gray	600	CD-1	120 181	1067.26
DBN9	1200 14400	ERF	Gray	600	CD-1	149 131	1071.74
RBM10	11400	lineal	Gray	1200	CD-1	0	1152.35
RBM11	11400	lineal	Gray	2400	CD-1	0	2305.41
RBM12	11400	ERF	Gray	2400	CD-1	0	2399
RBM13	11400	lineal	Gray	4800	CD-1	0	4643.71
DBN14	1200 14400	lineal	Gray	4800	CD-1	104 152	8693.7
RBM15	11400	lineal	Gray	600	PCD	1987	874.10

La figura 4.14a muestra la curva de costo obtenida durante el entrenamiento de dos RBM con 8192 unidades ocultas pero con diferente tipo de mapeo para el cambio de escala

Se observa como la *RBM2* con un mapeo lineal alcanza una entropía cruzada (CE) 3 unidades menor que la lograda para la misma cantidad de épocas por *RBM5*, lo que indica que durante el entrenamiento un mapeo lineal conduce a una mejor descripción de la probabilidad conjunta de las larvas, no obstante requiere 5.11 minutos más para entrenarse.

En el caso de las DBN, una comparación entre *DBN8* y *DBN9* ilustrada en la figura 4.14b permite observar que para la misma configuración de unidades ocultas es posible obtener una CE 50 unidades menor si se cuenta con un histograma ecualizado como el descrito por el mapeo o cambio de escala mediante blanqueo de datos y función de error. Esto quiere decir que la DBN aprende de mejor forma los factores de variación que describen las larvas si se utilizan mayor cantidad de bits o más rango de la escala disponible para representar los pesos, actuando de forma opuesta a la RBM. Esto se comprueba de igual forma en la figura 4.14b donde al finalizar el entrenamiento de la primera capa, *DBN9* presenta mayor CE que *DBN8*, no obstante al añadirse una capa e iniciar el modelo de aprendizaje profundo se logra una menor CE con *DBN9* y por ende un mejor modelado de la distribución de larvas en el entrenamiento.

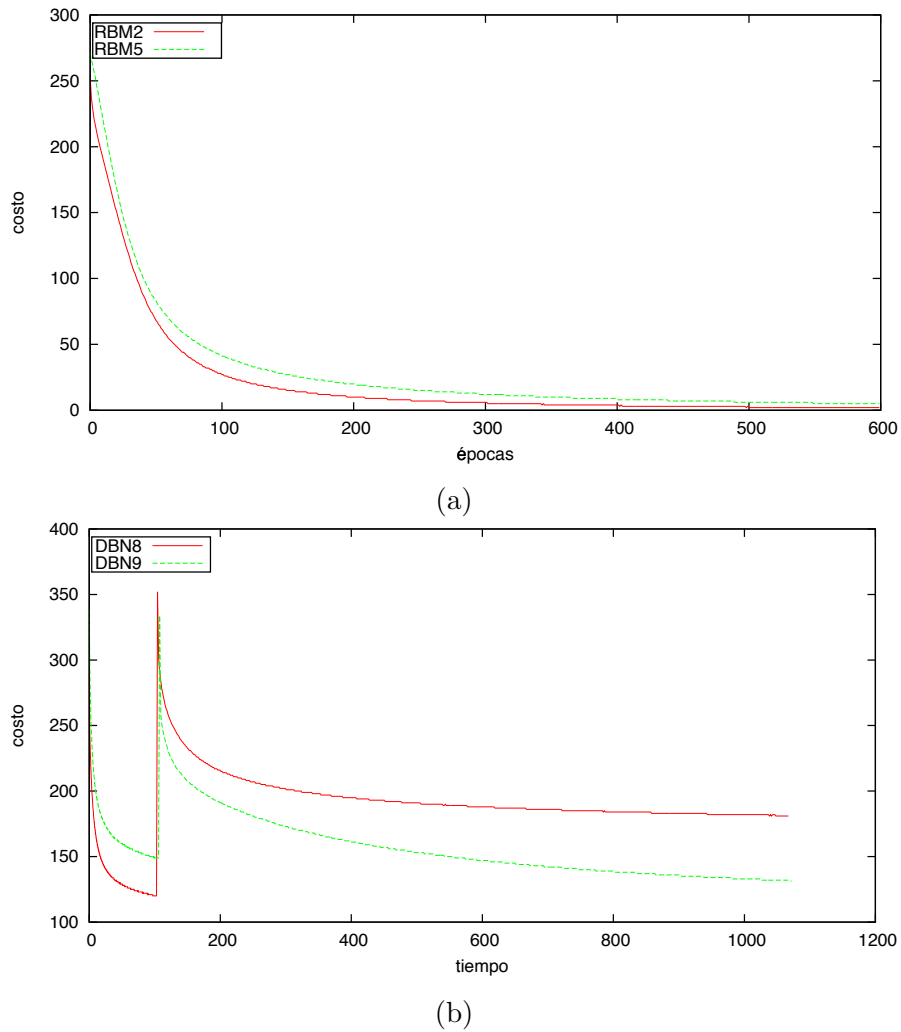


Figura 4.14: Variación en el costo de reconstrucción según el mapeo (a) Comparación de costo de reconstrucción en la RBM, (b) Comparación de costo de reconstrucción en la DBN. El pico en la gráfica indica el inicio del entrenamiento de la segunda capa

Por otra parte, como se muestra en la Tabla 4.6 para el conjunto $RBM1$, $RBM2$ y $RBM3$ así como para el conjunto $RBM4$, $RBM5$ y $RBM6$ la entropía cruzada es inversamente proporcional a la cantidad de unidades ocultas utilizadas, una mayor cantidad de unidades ocultas permite que la RBM utilice más bits para modelar el comportamiento de una unidad visible y que aprenda de mejor manera la distribución. Por ejemplo, en el caso de $RBM5$ existen 13,65 unidades ocultas por cada unidad visible en la larva. No obstante el incremento de unidades ocultas requiere un mayor tiempo de entrenamiento tal y como se ilustra en la figura 4.15 para el costo de reconstrucción del segundo grupo de RBM mencionado.

Para el caso de la representación binaria y la representación por código Gray, el costo de reconstrucción durante el entrenamiento no muestra diferencias, tal y como se ilustra en la figura 4.16 para $RBM2$ y $RBM7$.

Los entrenamientos de DBN y RBM descritos en la Tabla 4.6 indican que en ninguno de

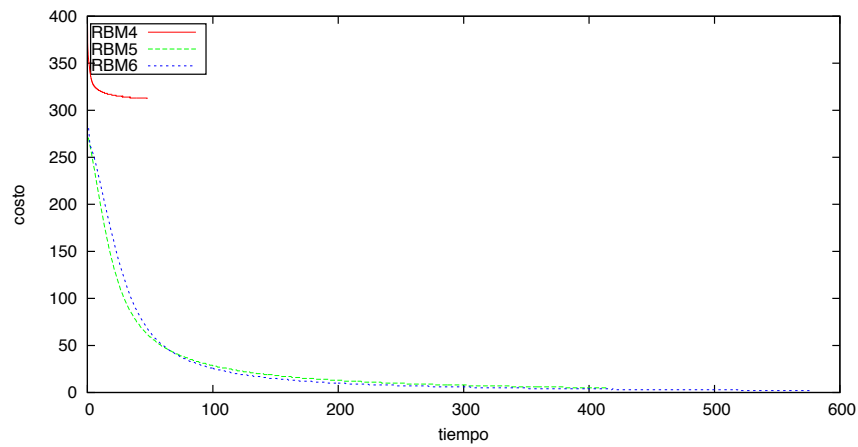


Figura 4.15: Entropía cruzada para RBM con diferentes cantidades de unidades ocultas

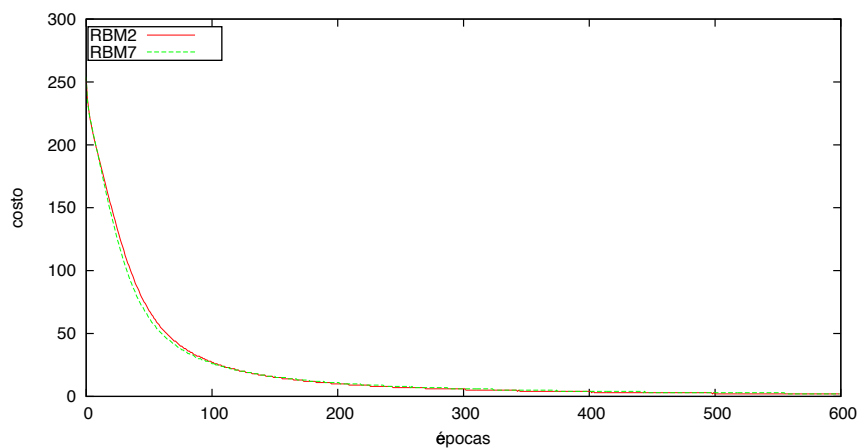


Figura 4.16: Entropía cruzada de entrenamiento para representación binaria y Gray

los casos de DBN fue posible obtener una entropía cruzada de cero como se obtuvieron con las RBM. No obstante, esto se debe a la cantidad de tiempo o épocas de entrenamiento requeridas por las DBN para disminuir la entropía cruzada, dicho comportamiento se ilustra en la figura 4.17, donde se puede observar que *DBN8* y *DBN14* al pasar de 600 épocas a 4800 logran disminuir la entropía cruzada en 29 unidades, pero requieren un incremento en el consumo de tiempo por un factor de 8.15.

En el caso de las RBM, estas logran una CE igual a cero antes de las 1200 épocas tal y como se ilustra en la figura 4.18, se observa como una vez alcanzada una entropía cruzada de cero esta se mantiene hasta el final del entrenamiento.

Para el caso de la *RBM15* se utiliza divergencia contrastante persistente, la cual utiliza como costo de reconstrucción la pseudo-verosimilitud y por ende no se da una comparación con las restantes RBM de la Tabla 4.6, no obstante el tiempo requerido para entrenar 600 épocas es 51,9% más que el tiempo requerido para entrenar *RBM3* con divergencia no persistente.

Del entrenamiento se obtiene que las RBM que mejor describen el conjunto \mathcal{C}^E son *RBM13* y *RBM9*. *RBM13* ya que es entrenada una mayor cantidad de épocas con

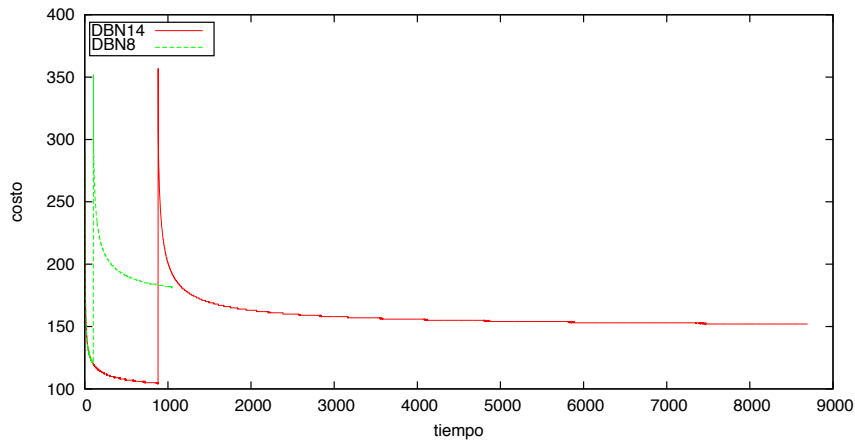


Figura 4.17: Variación del costo de reconstrucción para diferente cantidad de épocas o tiempo de entrenamiento en DBN

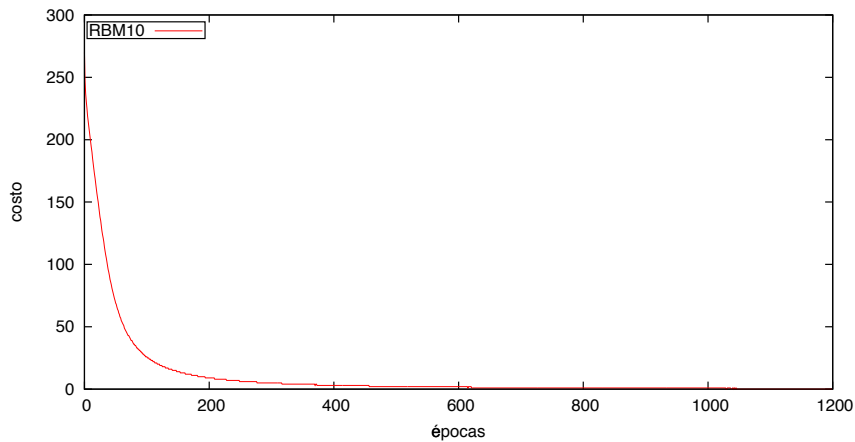


Figura 4.18: Variación del costo de reconstrucción en *RBM10*

CE nula permitiendo así un refinamiento de los pesos en dicho lapso, y *DBN9* al utilizar *ERF* obtiene *CE* igual a 131 siendo entrenada inclusive con 87.5% épocas menos que *DBN14*.

4.8. Reconstrucción de larvas: generación de hipótesis

Para evaluar el desempeño de los RBM y DBN se mide la degradación que causan en los modelos de forma, así como su capacidad de regenerar modelos incompletos. Para ello se utilizan vectores del conjunto de pruebas \mathcal{C}^P .

La prueba de degradación consiste en presentar una larva individual \mathbf{v}^ℓ real del conjunto \mathcal{C}^P como hipótesis \mathbf{v}_i^U de larva, con el objetivo de obtener una hipótesis de corrección \mathbf{v}_i^Ω de la RBM, a partir de la cual se determinan la cantidad de bits modificados por la RBM a una larva real así como la distancia euclidiana de \mathbf{v}_i^Ω a la larva original \mathbf{v}^ℓ . Posteriormente

\mathbf{v}_i^Ω es utilizado como hipótesis de entrada $\mathbf{v}_{i+1}^{\tilde{\Omega}} = \mathbf{v}_i^\Omega$ y se repiten las mediciones durante 10 iteraciones para 300 larvas y se obtiene el cambio promedio así como la desviación estándar (DE). El objetivo es determinar cuantos de los bits que estaban bien en la iteración n cambian en la iteración $n + 1$ y su impacto en la distancia total a la larva original.

En el caso de la evaluación de la regeneración, para una larva real \mathbf{v}^ℓ del conjunto \mathcal{C}^P se elimina uno a uno la información de los 600 bits que la conforman, asignando 0 al bit en cuestión, generando así 600 hipótesis de entrada $\mathbf{v}_i^{\tilde{\Omega}}$, $i \in \{0, \dots, 599\}$ y para cada una de estas se obtiene la distancia euclídeana entre la hipótesis de corrección \mathbf{v}_i^Ω , $i \in \{0, \dots, 599\}$ y la larva real \mathbf{v}^ℓ , consideradas como un vector de 60 dimensiones, donde cada dimensión corresponde a un peso \tilde{w}_i . De esta forma se obtiene una curva de tolerancia al cambio de bits que simula larvas incompletas e indica la capacidad de regeneración de la RBM.

Adicionalmente se evalúa la capacidad de la RBM/DBN para regenerar 50 bits de \mathbf{v}^ℓ cuando estos se sustituyen por ceros. En este caso la evaluación se realiza sobre 300 larvas para las cuales se miden tres distancias euclídeanas promedio: la distancia \mathcal{D}_{oe} entre la larva original completa y la estimada por la RBM/DBN cuando la larva original es presentada, la distancia \mathcal{D}_{om} entre la larva original completa y la modificada al reemplazar los bits y la distancia \mathcal{D}_{og} entre la larva original y la larva regenerada por la RBM/DBN.

En lo restante de la presente sección se presentan los resultados de degradación y regeneración al variar:

- Sistema numérico
- Cambio de escala y aprendizaje profundo
- Cantidad de unidades ocultas
- Cantidad de épocas
- Divergencia contrastante

Finalmente se muestran los resultados de regeneración obtenidos al utilizar el segmento medio en las secciones de larva no presentes.

4.8.1. Sistema numérico

La figura 4.19 ilustra el deterioro o degradación de *RBM2* (Gray) y *RBM7* (binario). Se observa que en ambos casos las RBM no son capaces de recuperarse de los cambios de bits realizados en iteraciones previas y que el código Gray presenta una menor cantidad de bits modificados por iteración.

De igual forma se obtiene el comportamiento de *RBM2* y *RBM7* en cuanto a distancia euclídeana entre las representaciones vectoriales $\mathbf{v}^\ell \in \mathbb{R}^{60}$ de la larva real y la generada por la RBM durante 10 iteraciones para 300 larvas, esto con el objetivo de determinar si los bits que están cambiando en la figura 4.19 corresponden a los más significativos de cada peso \tilde{w}_i . El comportamiento se ilustra en la figura 4.20.

A su vez se describe la magnitud de los 60 pesos \tilde{w}_i que representan a \mathbf{v}^ℓ en decimal y se

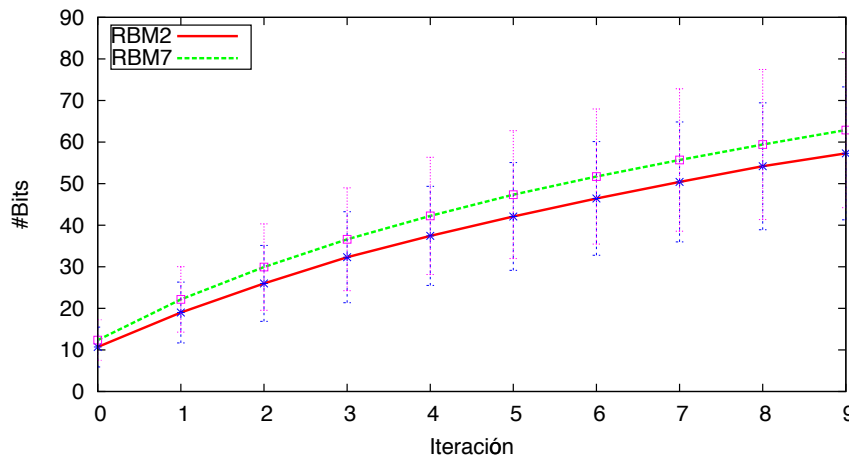


Figura 4.19: Promedio de números de bits cambiados por iteración para representación binaria y código Gray. Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

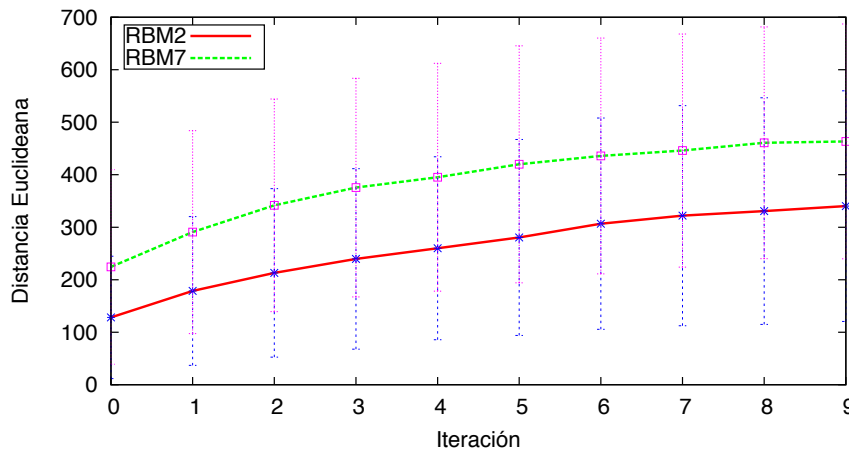


Figura 4.20: Promedio de deterioro de la distancia euclídeana entre vectores con representación binaria y código Gray en las RBM. Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

ilustran en la figura 4.21, así como las 10 hipótesis de corrección generadas por las RBM. Para el caso de la *RBM2* con código Gray 11 de los pesos se encuentran distanciados de los originales por más de 30 unidades y, para el caso binario 12 de ellos. Por lo tanto se observa que el código Gray aproxima mejor el valor decimal de los pesos, lo cual comprueba lo obtenido en las pruebas de cambio de bits y distancia euclídeana de las figuras 4.19 y 4.20.

Para la evaluación de la capacidad de regeneración se obtienen las curvas de tolerancia al cambio de bits mostradas en la figura 4.22, en las cuales la *RBM2* presenta una mayor tolerancia al cambio de bits hasta alcanzar 100 bits modificados, ya que en este punto se da un salto y describe la misma tolerancia que la representación binaria *RBM7*. Por este motivo en las siguientes evaluaciones se adopta el uso de representación de pesos mediante código Gray.

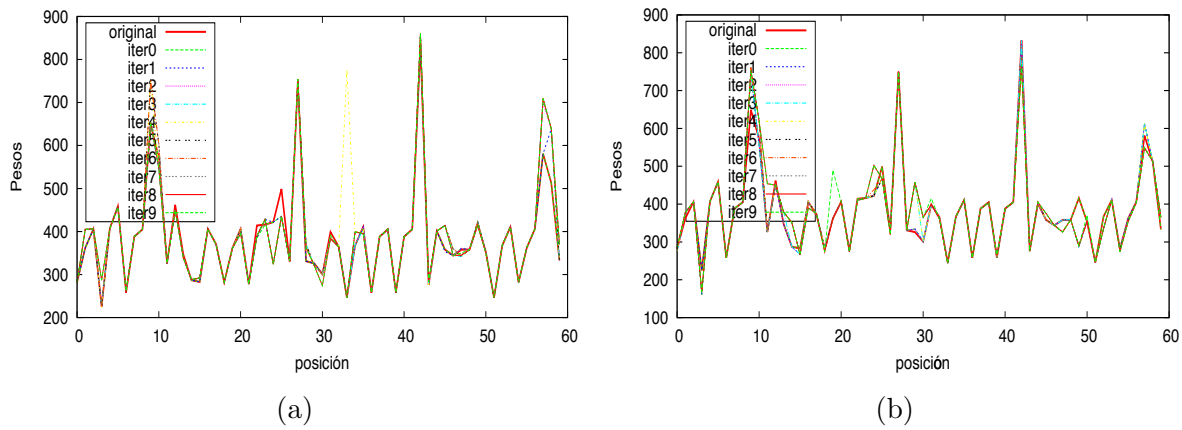


Figura 4.21: Secuencia de pesos de una larva real y 10 secuencias generadas por la RBM (a) *RBM2* con código Gray, (b) *RBM7* en binario

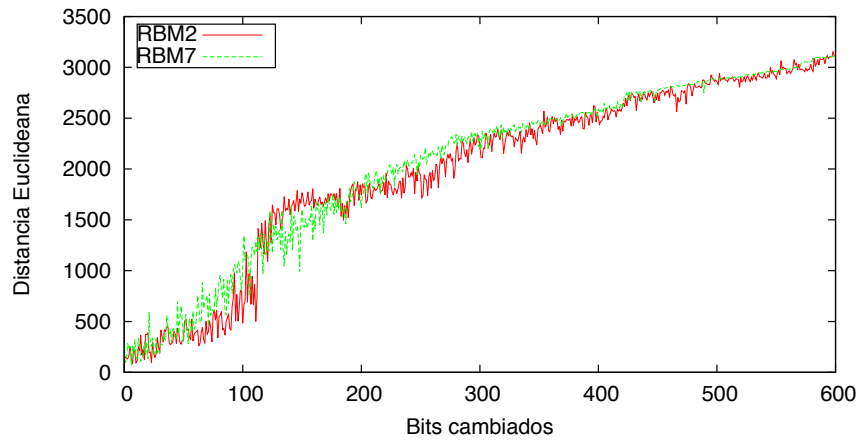


Figura 4.22: Curvas de tolerancia al cambio de bits para el sistema numérico Gray y Binario.

Finalmente, se alteran 50 bits (5 pesos con $\varphi = 10$ bits) en la región inicial de la larva real \mathbf{v}^ℓ ajustando su valor a cero y se genera una hipótesis de corrección \mathbf{v}^Ω . Utilizando 300 larvas se calculan las distancias promedio mostradas en la Tabla 4.7 donde se muestra que ambas RBM reducen la distancia causada por la modificación en la larva pero la RBM que utiliza código Gray realiza una mejor regeneración.

Tabla 4.7: Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original para diferentes sistemas numéricos

Modelo	\mathcal{D}_{oe}	DE_{oe}	\mathcal{D}_{om}	DE_{om}	\mathcal{D}_{og}	DE_{og}
RBM2	118.976	116.52	901.382	161.475	514.765	230.068
RBM7	206.22	185.308	901.382	161.475	622.298	202.265

Es posible observar en la figura 4.23 que para este caso tanto *RBM2* como *RBM7* no son capaces de generar todos pesos alterados, lo cual justifica la diferencia entre las distancias

\mathcal{D}_{oe} y \mathcal{D}_{og} de la Tabla 4.7 y se observa que la modificación de una región de bits no altera la interpretación de la RBM en la región de complemento.

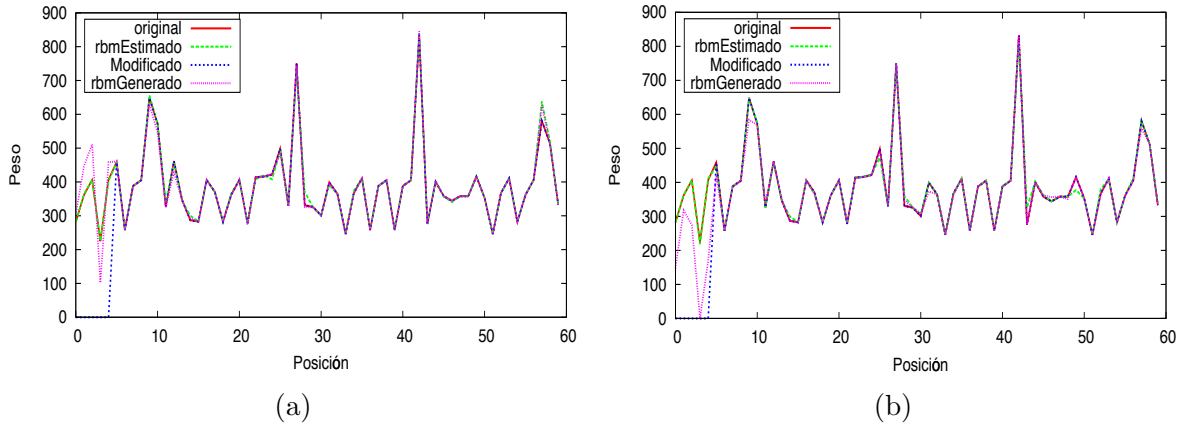


Figura 4.23: Vector de unidades visibles generado por las RBM con representación binaria y Gray ante una alteración de 50 bits ajustados a cero (a) *RBM2* con código Gray, (b) *RBM7* en binario

4.8.2. Cambio de escala y aprendizaje profundo

Para la evaluación del efecto causado por el método utilizado para el cambio de escala se evalúa inicialmente la diferencia en el deterioro mostrado por *RBM2* (mapeo lineal) y *RBM5* (mapeo por ERF), ambas con 8192 unidades ocultas y con código Gray. La figura 4.24 muestra el cambio de bits por iteración.

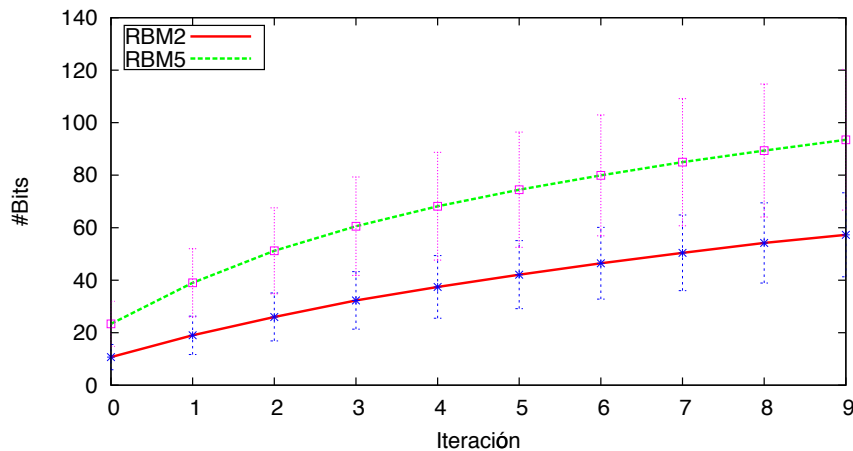


Figura 4.24: Promedio de números de bits cambiados por iteración para una representación con mapeo lineal (*RBM2*) y con mapeo por ERF (*RBM5*). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

Se observa que el cambio de bits es menor si se utiliza una representación por mapeo lineal. Para comprobar dicho efecto en el deterioro se obtiene nuevamente la distribución de los pesos en valor decimal para ambas RBM y se muestran en las figuras 4.25a y 4.25b. Donde

para el caso lineal, 11 de los 60 pesos se encuentran distanciados más de 30 unidades en contraste con el mapeo por ERF que muestra 16 pesos fuera del límite de las 30 unidades.

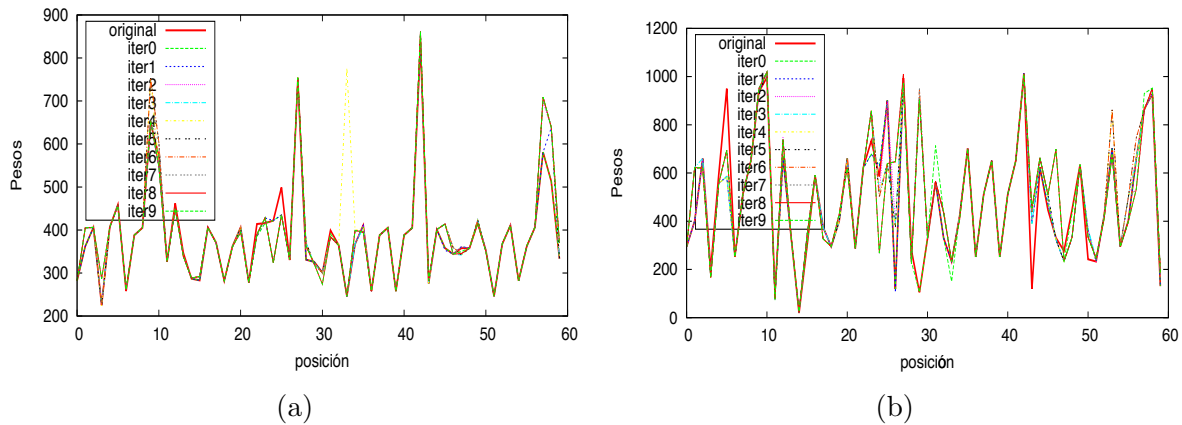


Figura 4.25: Secuencia de pesos de una larva real y 10 secuencias generadas por la RBM (a) $RBM2$ con mapeo lineal, (b) $RBM5$ con mapeo por ERF

Similarmente, las figuras 4.26a y 4.26b ilustran la distribución de pesos al utilizar DBN8 (lineal) y DBN9 (ERF), en donde se han utilizado únicamente 4 iteraciones con el objetivo de facilitar la interpretación del gráfico, ya que como es posible observar, estas DBN no poseen un deterioro deseable y no logran describir la probabilidad conjunta de las unidades visibles del entrenamiento al iterar sobre una hipótesis modificada varias veces. No obstante, la DBN de mapeo lineal ($DBN8$) describe mejor los pesos, en especial las regiones de pesos del 10 al 30 y del 40 al 60.

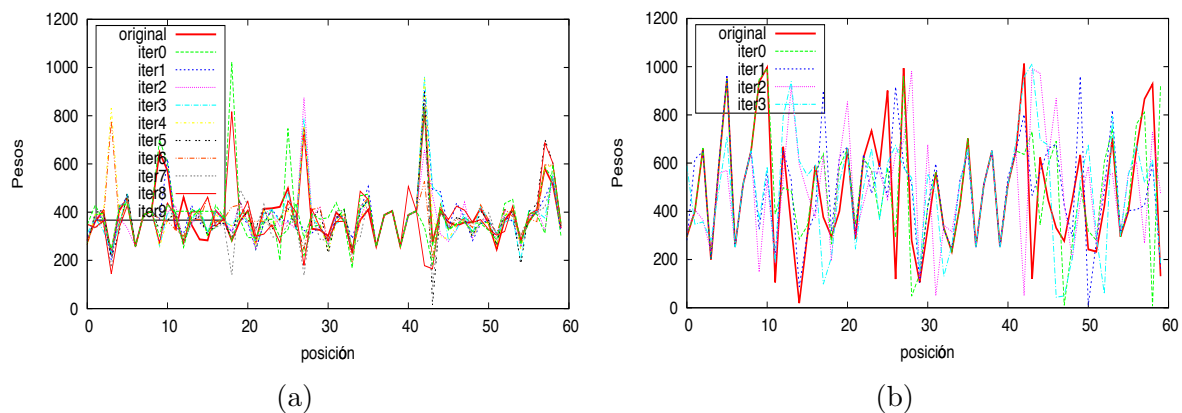


Figura 4.26: Secuencia de pesos de una larva real y 10 secuencias generadas por aprendizaje profundo con diferentes mapeos de cambio de escala (a) $DBN8$ con mapeo lineal, (b) $DBN9$ con mapeo por ERF

Por otra parte, la capacidad de generación de las RBM con mapeo lineal y por ERF es mostrada en las curvas de tolerancia al cambio de bits de la figura 4.27, en donde se observa que el mapeo lineal $RBM2$ posee una mayor tolerancia al cambio en las unidades visibles, manteniendo la distancia por debajo de 1000 unidades para cambios menores a

100 bits, es decir, para larvas que al menos contengan la información de 83.33% de los pesos.

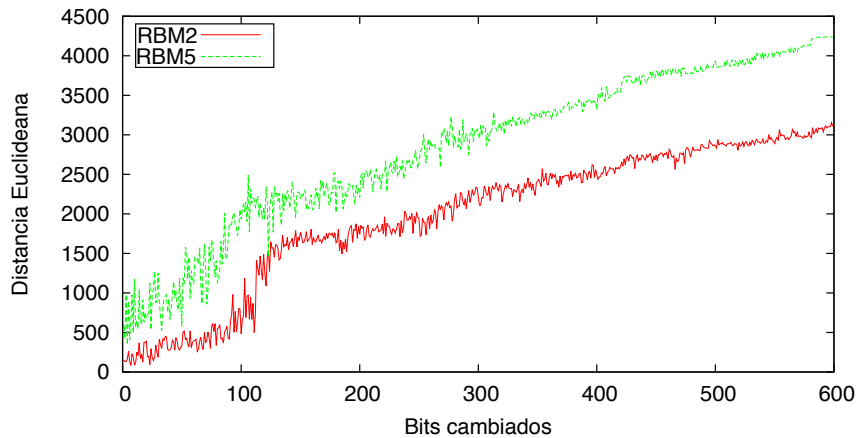


Figura 4.27: Curvas de tolerancia al cambio de bits para cambios de escala lineal y por ERF en RBM

Además, se obtienen las curvas de tolerancia para el modelo de aprendizaje profundo para el caso lineal (*DBN8*) y para el caso de ERF con blanqueo de datos (*DBN9*), dichas curvas son ilustradas en la figura 4.28 donde se observa que las DBN exponen una mayor tolerancia al cambio de bits y se corrobora que el mapeo lineal permite una menor distancia entre la larva hipótesis \mathbf{v}^{Ω} del conjunto de evaluación y la hipótesis de corrección \mathbf{v}^{Ω} inclusive ante un cambio de 30 pesos (300 bits).

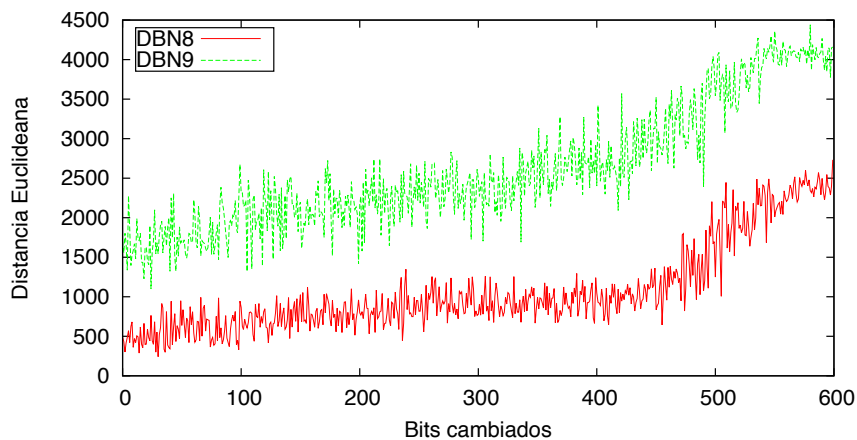


Figura 4.28: Curvas de tolerancia al cambio de bits para cambios de escala lineal y por ERF en aprendizaje profundo

Finalmente, en las figuras 4.29 y 4.30 se ilustra la hipótesis de larva original completa así como la hipótesis de corrección generada por las RBM y las DBN para ambos mapeos. En ambos casos se modifican 50 bits en la región inicial de la larva real \mathbf{v}^{ℓ} ajustando su valor a cero y se genera una hipótesis de corrección \mathbf{v}^{Ω} . A su vez se calculan las distancias promedio de 300 larvas mostradas en la Tabla 4.8.

Para el caso de las RBM sin importar el tipo de mapeo estas no logran reconstruir con

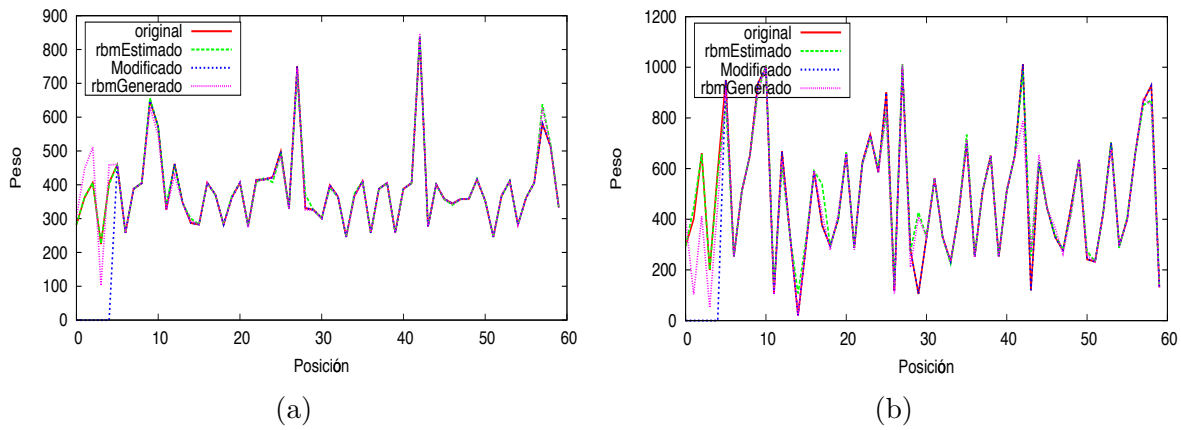


Figura 4.29: Vector de unidades visibles generado por las RBM para cambios de escala lineal y por ERF ante una alteración de 50 bits ajustados a cero (a) *RBM2* con mapeo lineal, (b) *RBM5* con mapeo por ERF

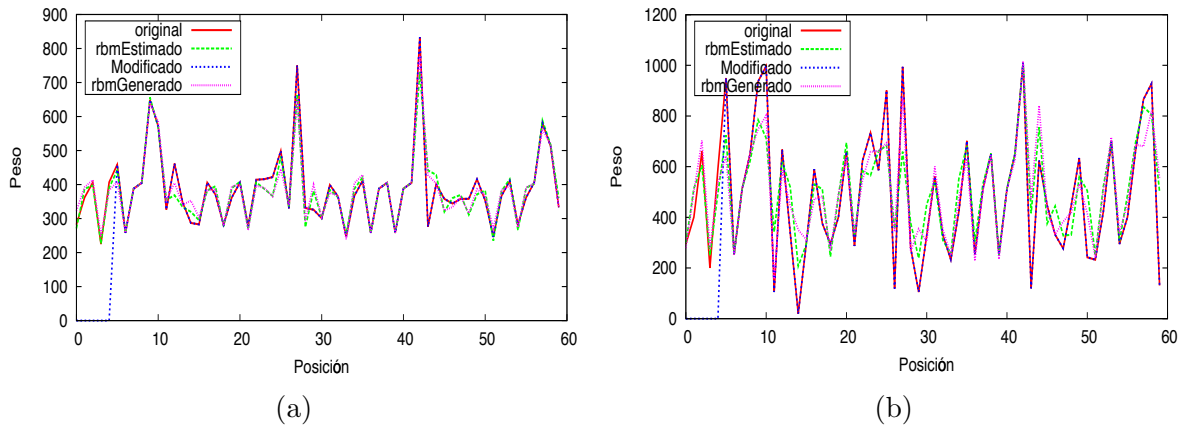


Figura 4.30: Vector de unidades visibles generado por las DBN para cambios de escala lineal y por ERF ante una alteración de 50 bits ajustados a cero (a) *DBN8* con mapeo lineal, (b) *DBN9* con mapeo por ERF

exactitud los pesos originales. No obstante, las DBN logran reconstruir los pesos modificados y por tanto la larva. Sin embargo, en las DBN la región de complemento no se describe con la misma precisión y exactitud con que lo hace la RBM pero se observa que las distancias obtenidas en promedio con un mapeo lineal (*RBM2* y *DBN8*) son menores y producen mejores resultados de regeneración.

4.8.3. Cantidad de unidades ocultas

Para evaluar el efecto causado por la cantidad de unidades ocultas en el modelo de forma se realiza una comparación entre *RBM1* con 128 unidades ocultas y *RBM3* con 11400 unidades ocultas, las cuales poseen el deterioro mostrado en la figura 4.31.

Considerando 11400 unidades ocultas el modelo de forma permite un deterioro promedio máximo del 8.94 % de los bits comparado con el 33.32 % dado por una RBM con 128 uni-

Tabla 4.8: Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original para diferentes mapeos en RBM y DBN

Modelo	\mathcal{D}_{oe}	DE_{oe}	\mathcal{D}_{om}	DE_{om}	\mathcal{D}_{og}	DE_{og}
RBM2	118.976	116.52	901.382	161.475	514.765	230.068
RBM5	462.205	296.143	1196.110	283.925	1130.64	362.329
DBN8	474.5	208.255	901.382	161.475	620.600	246.317
DBN9	1441.210	403.382	1196.110	283.925	1580.373	424.403

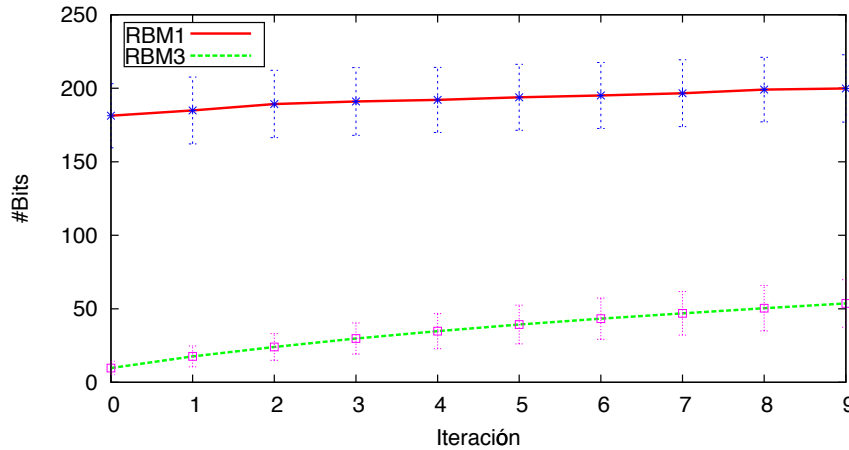


Figura 4.31: Promedio de números de bits cambiados por iteración para una RBM con 128 unidades ocultas (*RBM1*) y con 11400 (*RBM3*). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

dades ocultas, no obstante el tiempo de entrenamiento requerido por *RBM3* es 1141.61 % mayor que el requerido por *RBM1*. Para determinar si los bits que cambian corresponden a los menos significativos de cada peso se obtiene la distancia euclideana del vector de unidades visibles generado \mathbf{v}^Ω al vector de entrada \mathbf{v}^Ω , la evaluación se repite para 300 larvas y se ilustra en la figura 4.32 las distancias promedio así como la desviación estándar observada en cada iteración.

Para el caso de 128 unidades ocultas la distancia promedio máxima es de 662 unidades en la novena iteración no obstante desde la primera iteración la distancia promedio es de 455 unidades en comparación con *RBM3* que presenta una distancia promedio máxima de 332 en la iteración nueve. Por tanto los bits que cambian no corresponden únicamente a los menos significativos y un aumento en la cantidad de unidades ocultas permite una mejora en el deterioro descrito por el modelo de forma de la RBM pero no su tendencia. La figura 4.33 muestra los pesos generados por ambas RBM durante 10 iteraciones, donde *RBM3* presenta 8 pesos con distancias mayores a 30 unidades principalmente después de la iteración 5.

Para evaluar la capacidad de regeneración se obtienen las curvas de tolerancia al cambio

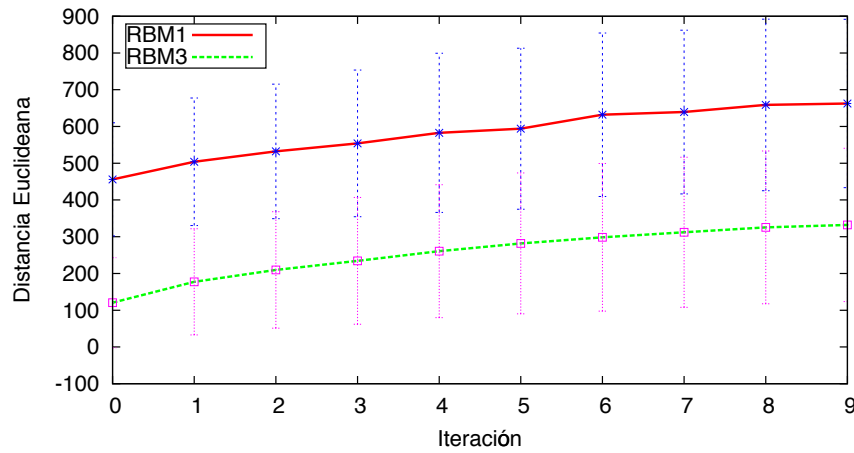


Figura 4.32: Promedio de deterioro de la distancia euclídeana entre vectores para RBM con 128 unidades ocultas (*RBM1*) y 11400 unidades ocultas (*RBM3*). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

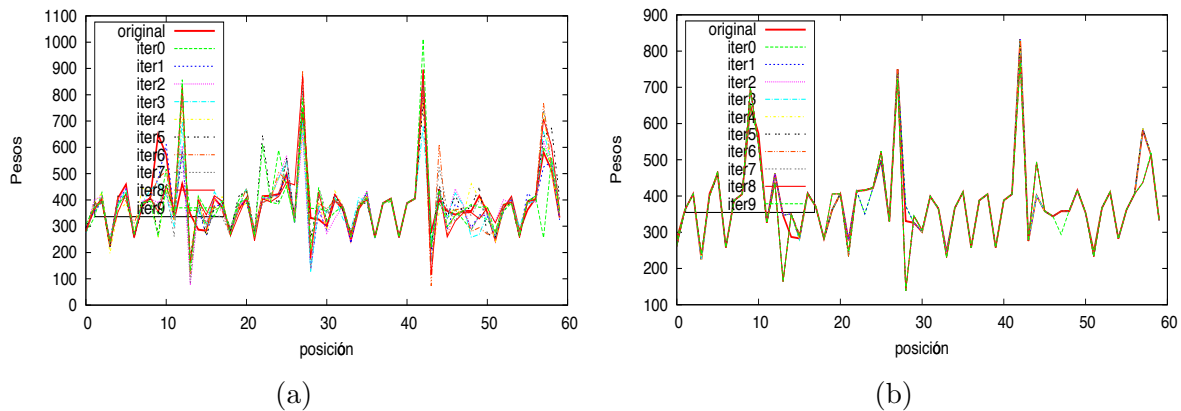


Figura 4.33: Secuencia de pesos generados por RBM para diferentes valores de unidades ocultas (a) *RBM1* con 128 unidades ocultas, b) *RBM3* con 11400 unidades ocultas

de bits de *RBM1* y *RBM3* ilustradas en la figura 4.34. Se observa que la RBM con mayor cantidad de unidades oculta permite obtener una distancia por debajo de 150 unidades antes del cambio de 50 bits y menor a 500 antes de los 90 bits. No obstante, para una modificación mayor a 100 bits la RBM no tiene capacidad de reconstrucción, mostrando que la RBM con mayor cantidad de unidades ocultas en la Tabla 4.6 no puede describir la capacidad de regeneración obtenidas por las DBN.

Finalmente es posible observar en la figura 4.35 como el incremento en las unidades ocultas permite mejorar la exactitud de los pesos dados por la RBM en la región de complemento (pesos del cinco en adelante) no obstante no es posible generar con exactitud los pesos modificados.

La Tabla 4.9 muestra las distancias de regeneración promedio obtenidas para 300 larvas, donde se observa que se obtiene una menor distancia al incrementar la cantidad de unidades ocultas (*RBM3*) y por ende una mejor regeneración.

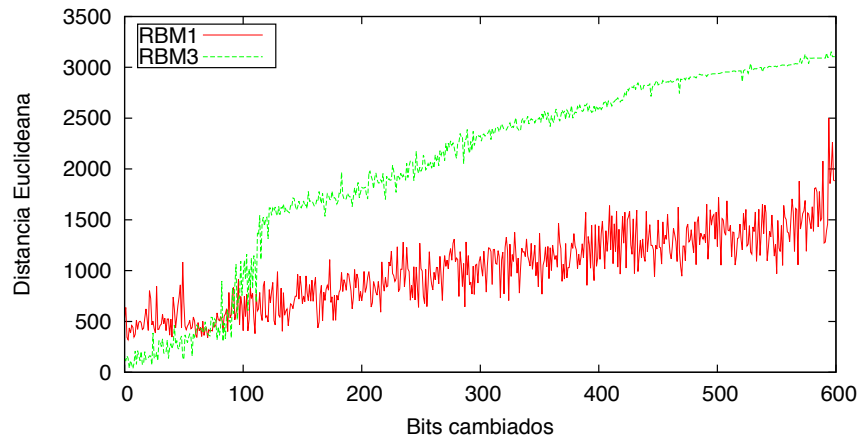


Figura 4.34: Curvas de tolerancia al cambio de bits en RBM para diferentes cantidades de unidades ocultas

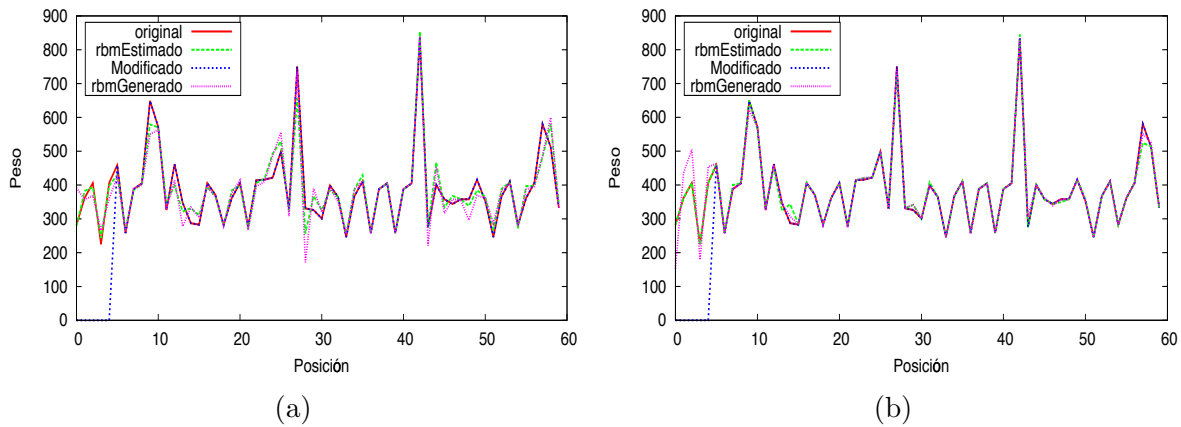


Figura 4.35: Vector de unidades visibles generado por las RBM con diferentes cantidades de unidades ocultas ante una alteración de 50 bits ajustados a cero (a) *RBM1* con 128 unidades ocultas, b) *RBM3* con 11400 unidades ocultas

4.8.4. Cantidad de épocas

La evaluación de la variación en el deterioro y en la capacidad de generación debido a las épocas de entrenamiento se realiza considerando modelos basados en RBM y en aprendizaje profundo. Inicialmente se consideran los modelos definidos por *RBM3*, *RBM11* y *RBM13* entrenados con 600, 2400 y 4800 épocas respectivamente, para los cuales se obtiene el deterioro promedio en bits para 300 larvas mostrado en la figura 4.36.

Se muestra que la cantidad de bits que cambian por iteración en la representación de forma de larvas es inversamente proporcional a la cantidad de épocas. Entre *RBM3* y *RBM11* existe una diferencia de un 37% en la cantidad máxima de cambio de bits al incrementar las épocas en un 300%. La figura 4.37 ilustra como varía el cambio de bits máximo en función de la cantidad de épocas para las RBM de 11400 unidades ocultas de la Tabla 4.6.

De igual forma para el caso de la distancia euclídeana de la figura 4.38 se obtiene que

Tabla 4.9: Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al variar la cantidad de unidades ocultas en la RBM

Modelo	\mathcal{D}_{oe}	DE_{oe}	\mathcal{D}_{om}	DE_{om}	\mathcal{D}_{og}	DE_{og}
RBM1	463.051	159.798	901.382	161.475	637.309	192.756
RBM3	116.735	134.088	901.382	161.475	518.494	231.817

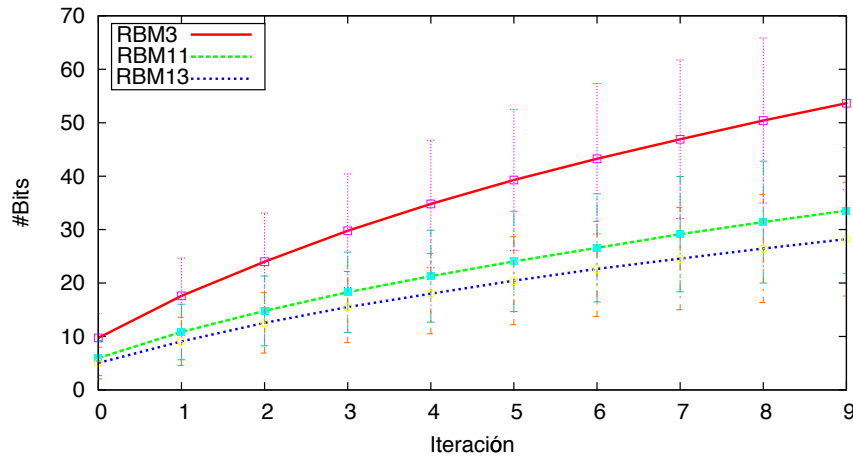


Figura 4.36: Promedio de números de bits cambiados por iteración para RBM entrenadas con diferente cantidad de épocas: *RBM3* (600 épocas), *RBM11* (2400 épocas) y *RBM13* (4800 épocas). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

al incrementar la cantidad de épocas del entrenamiento se logra disminuir la distancia entre el vector de entrada y la hipótesis generada por las RBM, sin importar si la RBM

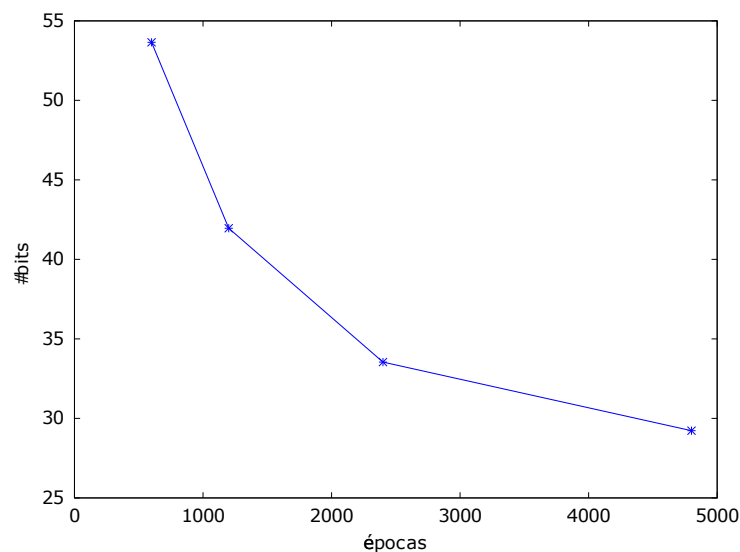


Figura 4.37: Cantidad promedio máxima de bits cambiantes en función de la cantidad de épocas de entrenamiento para 11400 unidades ocultas

a partir de las 1200 épocas ya había alcanzado una entropía cruzada de cero (Tabla 4.6). Esto se debe a que durante el entrenamiento el muestreo de Gibbs continua iterando y mejorando el valor de los parámetros internos de la RBM y por ende el modelo de forma. Se disminuye la cantidad de bits cambiantes pero no la tendencia al aumento en cada iteración.

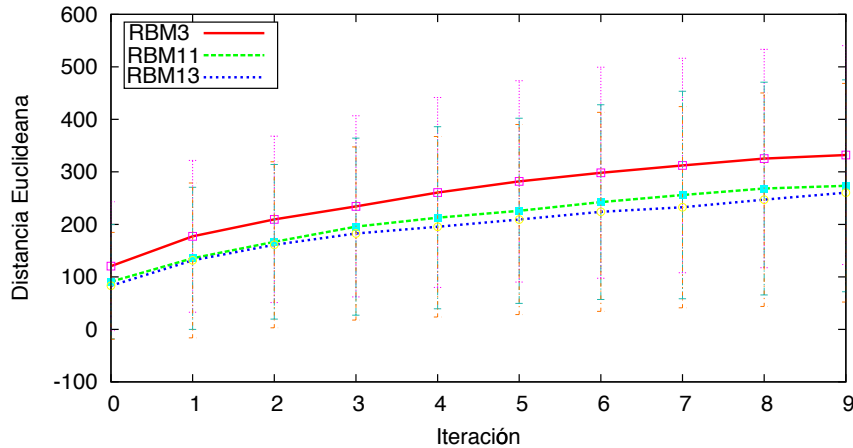


Figura 4.38: Promedio de deterioro de la distancia euclídeana para RBM con diferentes cantidades de épocas: *RBM3* (600 épocas), *RBM11* (2400 épocas) y *RBM13* (4800 épocas). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

Por su parte, la capacidad de generación no se ve mejorada de forma significativa con el incremento de épocas de entrenamiento tal y como se muestra en las curvas de tolerancia al cambio de bits de la figura 4.39 y en las secuencias de pesos generadas en la figura 4.40, en donde en ninguno de los casos la RBM fue capaz de reconstruir con exactitud la larva modificada.

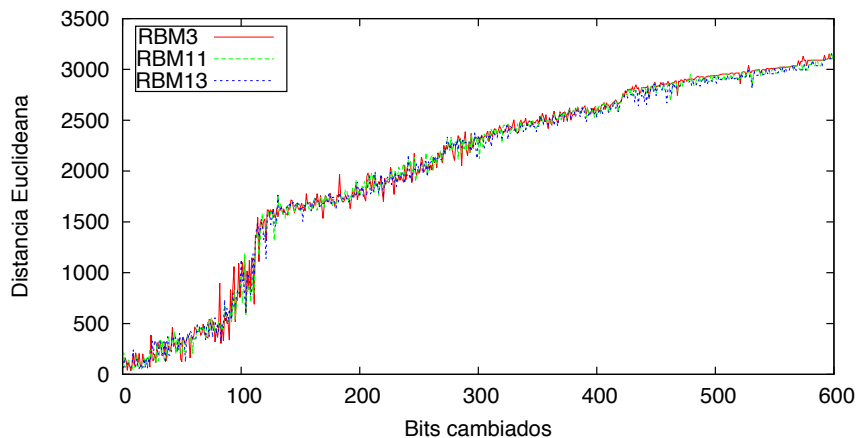


Figura 4.39: Curvas de tolerancia al cambio de bits para RBM con diferente cantidad de épocas de entrenamiento: *RBM3* (600 épocas), *RBM11* (2400 épocas) y *RBM13* (4800 épocas)

La Tabla 4.10 muestra las distancias de regeneración promedio obtenidas para 300 larvas utilizando *RBM3*, *RBM11* y *RBM13*, donde se observa que al aumentar la cantidad

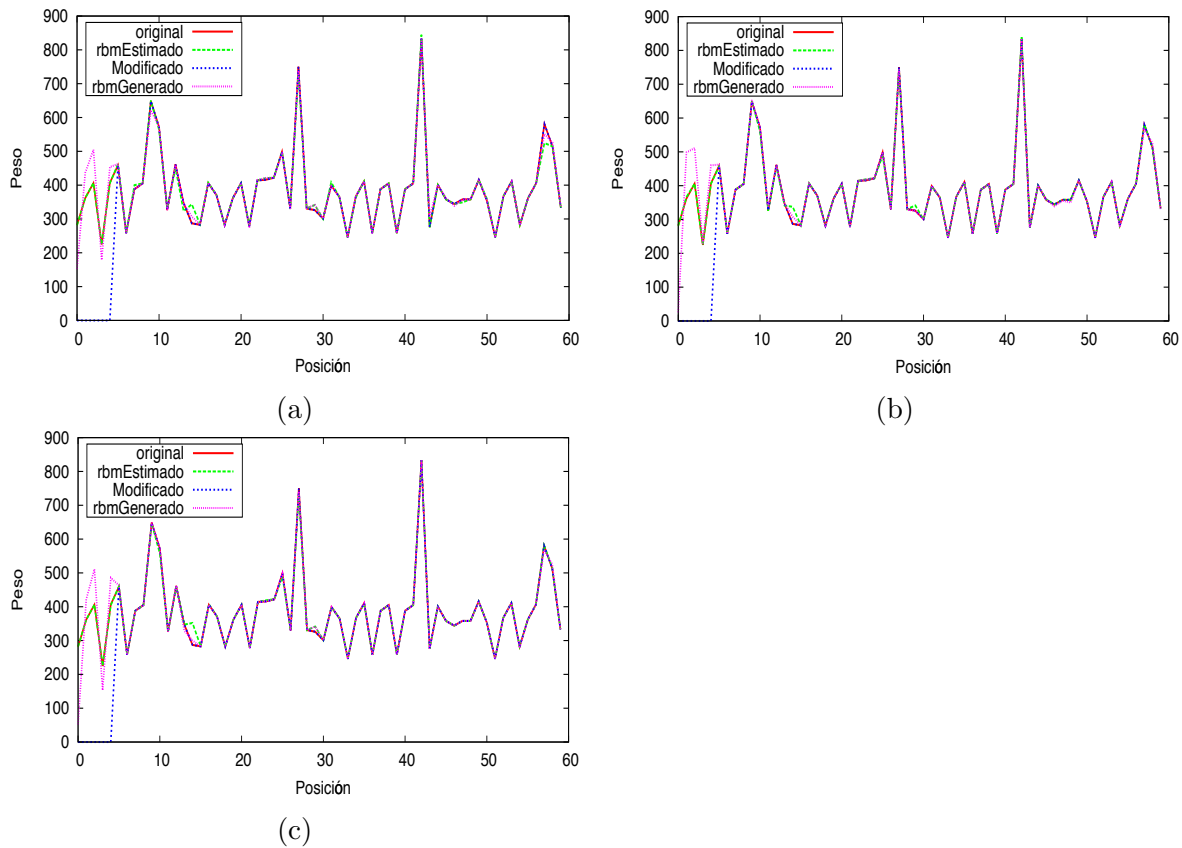


Figura 4.40: Vector de unidades visibles generado por las RBM con diferente cantidad de épocas de entrenamiento ante una alteración de 50 bits ajustados a cero (a) *RBM3* con 600 épocas, (b) *RBM11* con 2400 épocas, (c) *RBM13* con 4800 épocas

de épocas se obtiene una menor distancia promedio para la larva estimada al recibir una larva completa como entrada, no obstante para la larva regenerada no se observa una tendencia de optimización de la distancia al aumentar las épocas en la RBM.

Tabla 4.10: Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al variar la cantidad épocas de entrenamiento en la RBM

Modelo	\mathcal{D}_{oe}	DE_{oe}	\mathcal{D}_{om}	DE_{om}	\mathcal{D}_{og}	DE_{og}
RBM3	116.735	134.088	901.382	161.475	518.494	231.817
RBM11	99.413	116.045	901.382	161.475	515.985	240.918
RBM13	92.263	123.639	901.382	161.475	527.083	236.595

Para el caso de aprendizaje profundo se realiza la comparación utilizando los modelos descritos por *DBN8* y *DBN14*, ambos con 1200 unidades ocultas en la primera de las capas y 14400 unidades en la segunda capa oculta. El primero de ellos es entrenado 600 épocas y el segundo 4800 épocas. Se obtiene el deterioro en bits de la figura 4.41 para la cual se observa que al igual que en la RBM un mayor número de épocas permite disminuir

la cantidad de bits que cambian en cada iteración, sin embargo la tendencia al crecimiento se mantiene para los bits así como para las distancias euclidianas tal y como se ilustra en la figura 4.42.

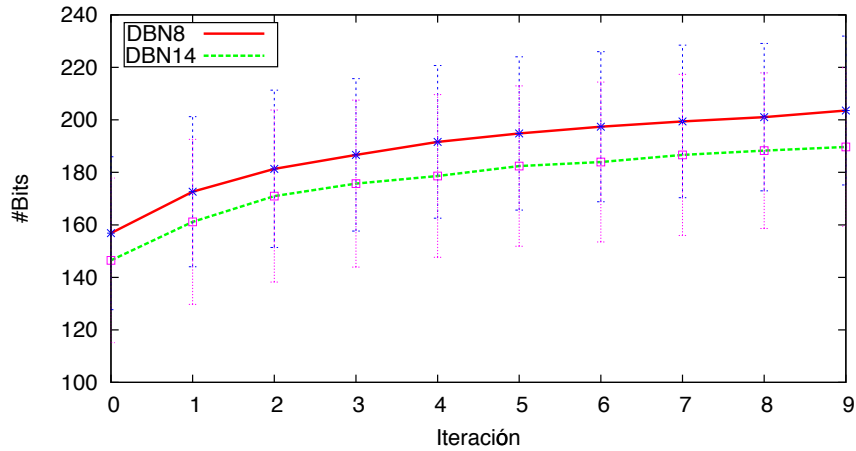


Figura 4.41: Promedio de números de bits cambiados por iteración para DBN entrenadas con diferente cantidad de épocas: *DBN8* (600 épocas) y *DBN14* (4800 épocas). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

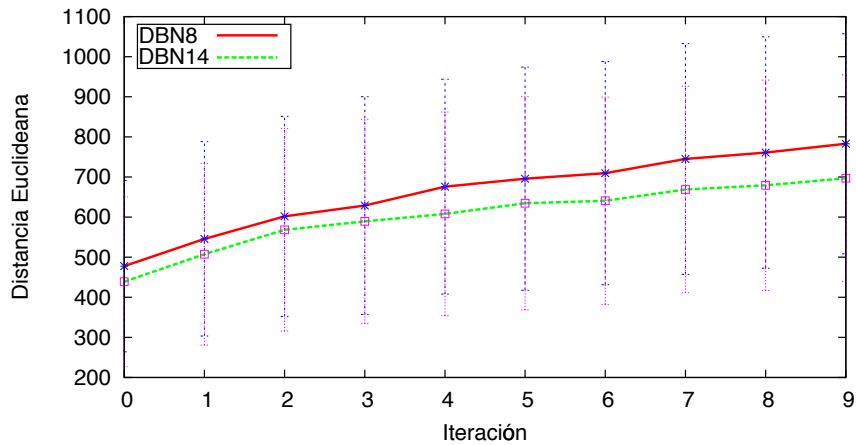


Figura 4.42: Promedio de deterioro de la distancia euclídeana para DBN con diferente cantidad de épocas de entrenamiento: *DBN8* (600 épocas) y *DBN14* (4800 épocas). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

En cuanto al aspecto regenerativo, se observa en la curva de tolerancia de la figura 4.43 que *DBN14* con una mayor cantidad de épocas de entrenamiento permite disminuir la distancia máxima de 2500 a 2000 unidades, es decir, un 20% y que en ambos casos la DBN es capaz de regenerar el segmento alterado pero no se observa en la figura 4.44 una diferencia significativa en dicha región ni en la región de complemento entre *DBN8* y *DBN14*.

Con el objetivo de determinar si existen una mejora en la regeneración se calculan las

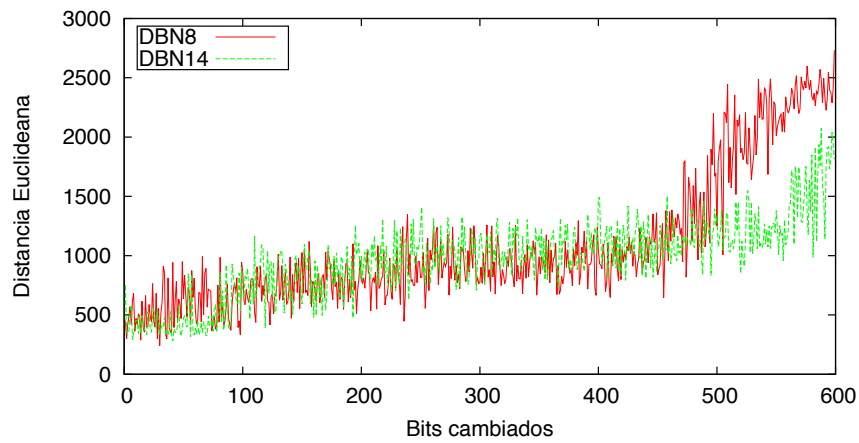


Figura 4.43: Curvas de tolerancia al cambio de bits para DBN con diferente cantidad de épocas de entrenamiento: *DBN8* (600 épocas) y *DBN14* (4800 épocas)

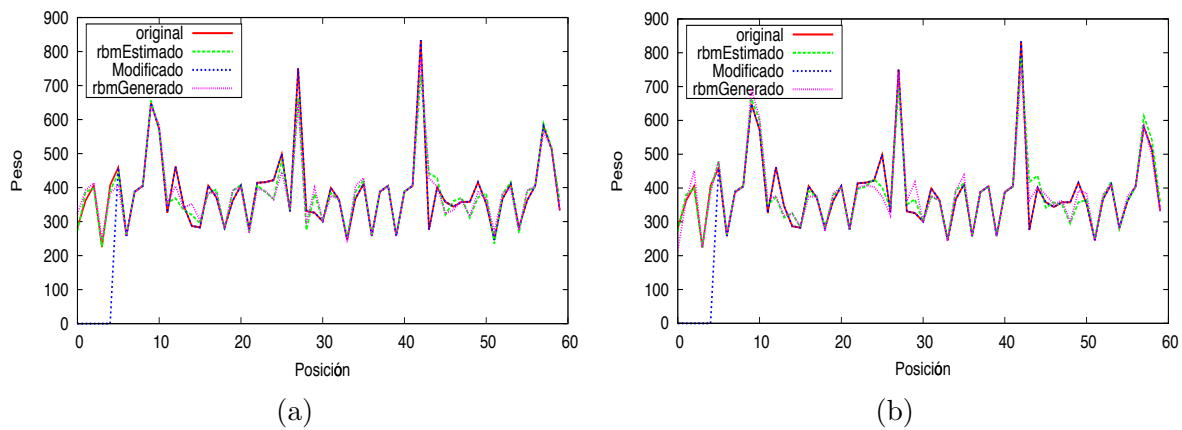


Figura 4.44: Vector de unidades visibles generado por las DBN con diferente cantidad de épocas de entrenamiento ante una alteración de 50 bits ajustados a cero (a) *DBN8* con 600 épocas, (b) *DBN14* con 4800 épocas

distancias de regeneración promedio para 300 larvas mostradas en la Tabla 4.11, donde se puede observar que un aumento en la cantidad de épocas de entrenamiento en la RBM disminuye la distancia de la larva regenerada a la original y por ende mejora la regeneración.

Tabla 4.11: Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al variar la cantidad épocas de entrenamiento en la DBN

Modelo	\mathcal{D}_{oe}	DE_{oe}	\mathcal{D}_{om}	DE_{om}	\mathcal{D}_{og}	DE_{og}
DBN8	474.500	208.255	901.382	161.475	620.600	246.317
DBN14	419.057	188.020	901.382	161.475	593.838	229.261

4.8.5. Divergencia Contrastante

Para el caso de la divergencia contrastante se realiza una comparación entre *RBM3* y *RBM15* para los cuales durante el entrenamiento se utiliza divergencia contrastante (CD-1) y divergencia contrastante persistente (PCD) respectivamente. El deterioro de bits es ilustrado en la figura 4.45. Para el caso de CD-1 se obtiene un menor cambio de bits por iteración que para el caso persistente. No obstante, al obtener la variación de la distancia euclidea se observa que PCD presenta una menor distancia a partir de la iteración 3 y por tanto los bits que están cambiando son los menos significativos provocando mejores distancias al vector original tal y como se ilustra en la figura 4.46.

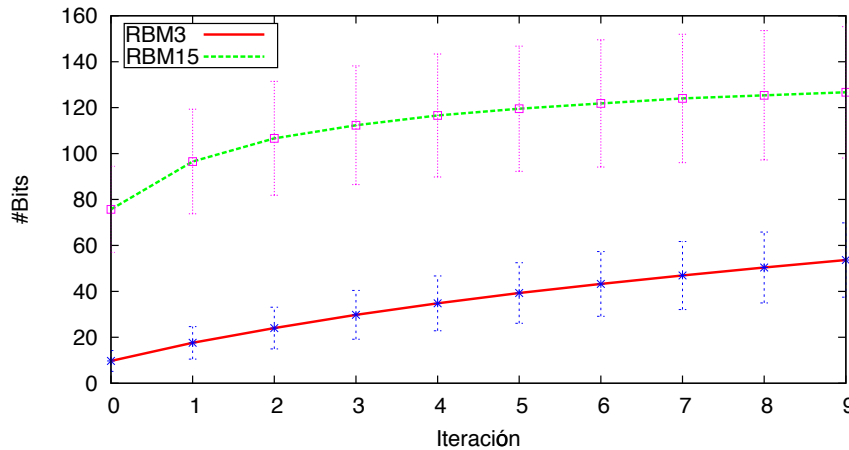


Figura 4.45: Promedio de número de bits cambiados por iteración para RBM con diferentes divergencias contrastantes: *RBM3* (CD-1) y *RBM15* (PCD). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

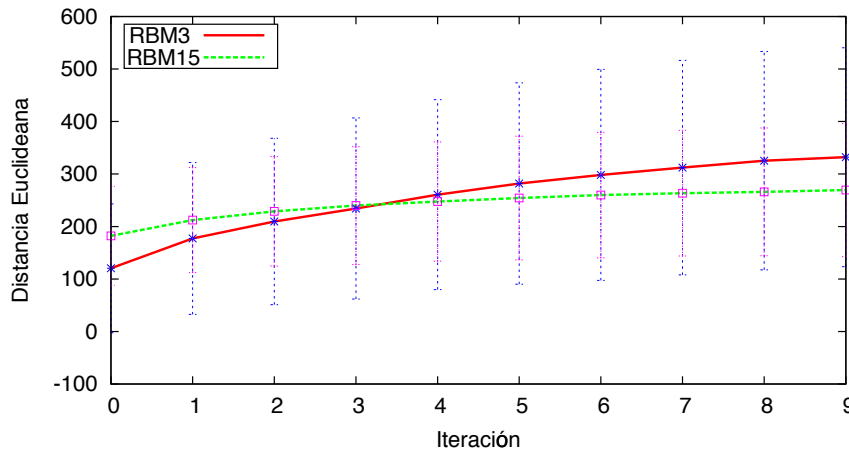


Figura 4.46: Promedio de deterioro de la RBM en distancia euclidea para diferentes divergencias contrastantes: *RBM3* (CD-1) y *RBM15* (PCD). Las líneas verticales representan la desviación estándar observada durante la evaluación de 300 larvas

Seguidamente se obtiene la distribución de pesos descrita por *RBM3* y *RBM15*, esta es ilustrada en la figura 4.47, donde se puede observar que la RBM entrenada con PCD

aproxima de forma adecuada dos de los pesos mal representados por *RBM3*, propiamente los pesos 19 y 57.

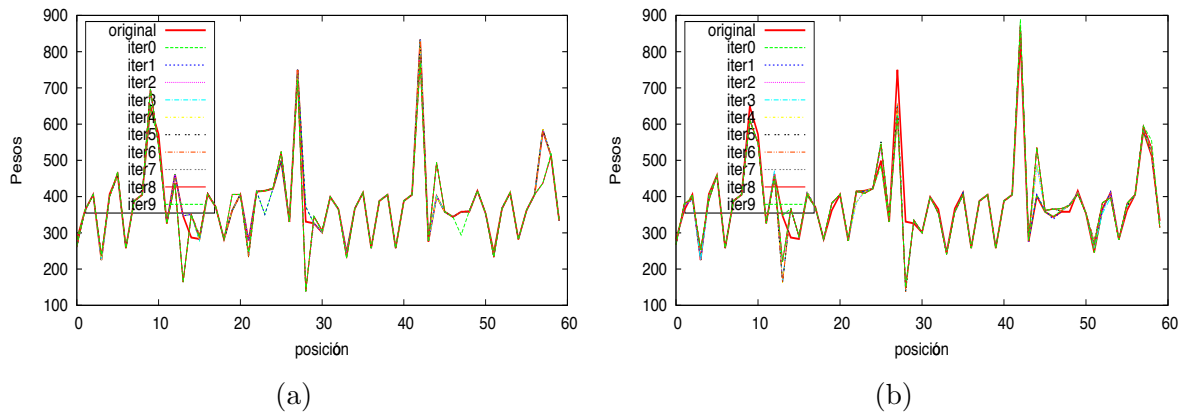


Figura 4.47: Secuencia de pesos generados por RBM para diferentes divergencias contrastantes (a) *RBM3* con CD-1, (b) *RBM15* con PCD

Finalmente, las capacidades de regeneración de ambas RBM son descritas mediante la figuras 4.48 y 4.49 donde se observa que la RBM con CD-1 posee una mejor tolerancia al cambio de bits hasta llegar a los 100 bits; no obstante, de dicho punto en adelante PCD presenta una mejor tolerancia.

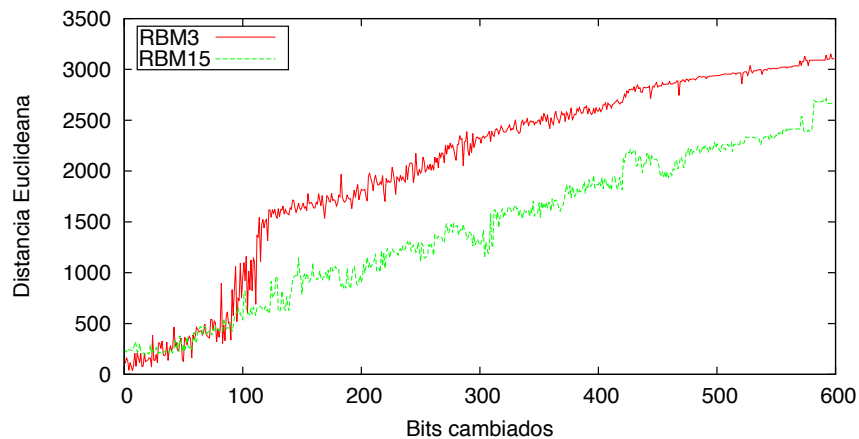


Figura 4.48: Curvas de tolerancia de RBM al cambio de bits para diferentes divergencias contrastantes: *RBM3* (CD-1) y *RBM15* (PCD)

En ambos casos la RBM no es capaz de generar la larva incompleta. La Tabla 4.12 con las distancias de regeneración promedio para 300 larvas mostradas indica que PCD permite una mejor regeneración de los 50 bits alterados al inicio de la larva.

4.8.6. Reconstrucción a partir del segmento medio

Las pruebas de generación de las secciones anteriores asignan un valor de cero a la región de bits que se desea reconstruir, un conjunto de $\varphi = 10$ unidades visibles en cero equivale

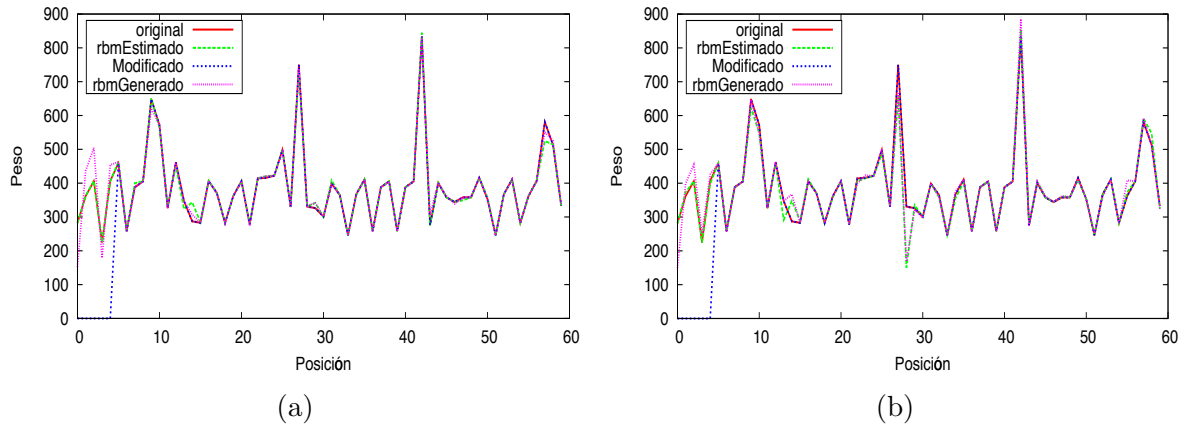


Figura 4.49: Secuencia de pesos generados por RBM para diferentes divergencias contrastantes ante una alteración de 50 bits ajustados a cero (a) *RBM3* con CD-1, (b) *RBM15* con PCD

Tabla 4.12: Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al variar la divergencia contrastante

Modelo	\mathcal{D}_{oe}	DE_{oe}	\mathcal{D}_{om}	DE_{om}	\mathcal{D}_{og}	DE_{og}
RBM3	116.735	134.088	901.382	161.475	518.494	231.817
RBM15	180.383	90.036	901.382	161.475	369.223	196.304

a utilizar el valor de $-3\lambda_i$ para la dimensión i ya que los pesos son previamente mapeados a una escala de 0 a 1023, lo cual podría dificultar la regeneración de la forma tanto para las RBM como para las DBN ya que el peso w_i se encontraría en la forma más extrema tal y como se ilustra en las figuras 4.7, 4.8 y 4.9. Para evaluar el efecto causado por dicha asignación se reemplazan 2 segmentos \hat{S}_i de \mathbf{v}^ℓ con \hat{S}_μ para crear una hipótesis de larva incompleta \mathbf{v}^ψ y esta es evaluada utilizando *RBM13* y *DBN14* en conjunto con 300 larvas para las cuales se obtienen las distancias euclidianas promedio de la Tabla 4.13.

Tabla 4.13: Distancias promedio de larvas estimadas, modificadas y regeneradas con respecto a la larva original al utilizar el bits del segmento medio

Modelo	\mathcal{D}_{oe}	DE_{oe}	\mathcal{D}_{om}	DE_{om}	\mathcal{D}_{og}	DE_{og}
RBM13 $_{\hat{S}_\mu}$	92.264	123.639	426.199	45.082	407.523	124.371
RBM13	92.264	123.639	984.486	147.849	619.816	250.540
DBN14 $_{\hat{S}_\mu}$	419.057	188.021	426.199	45.082	545.654	211.117
DBN14	419.057	188.020	984.486	147.849	621.344	232.927

Se observa que para el caso de la RBM el uso del segmento medio permite reducir la distancia euclideana promedio de la larva generada a la original en un 34.25% y para la DBN en un 12.18%. La figuras 4.50a y 4.50b muestran una de las hipótesis de corrección

obtenidas, donde no es posible identificar de forma visual que el remplazo de las secciones faltantes con \hat{S}_μ mejora la capacidad de regeneración sin embargo los datos en la Tabla 4.13 lo comprueban.

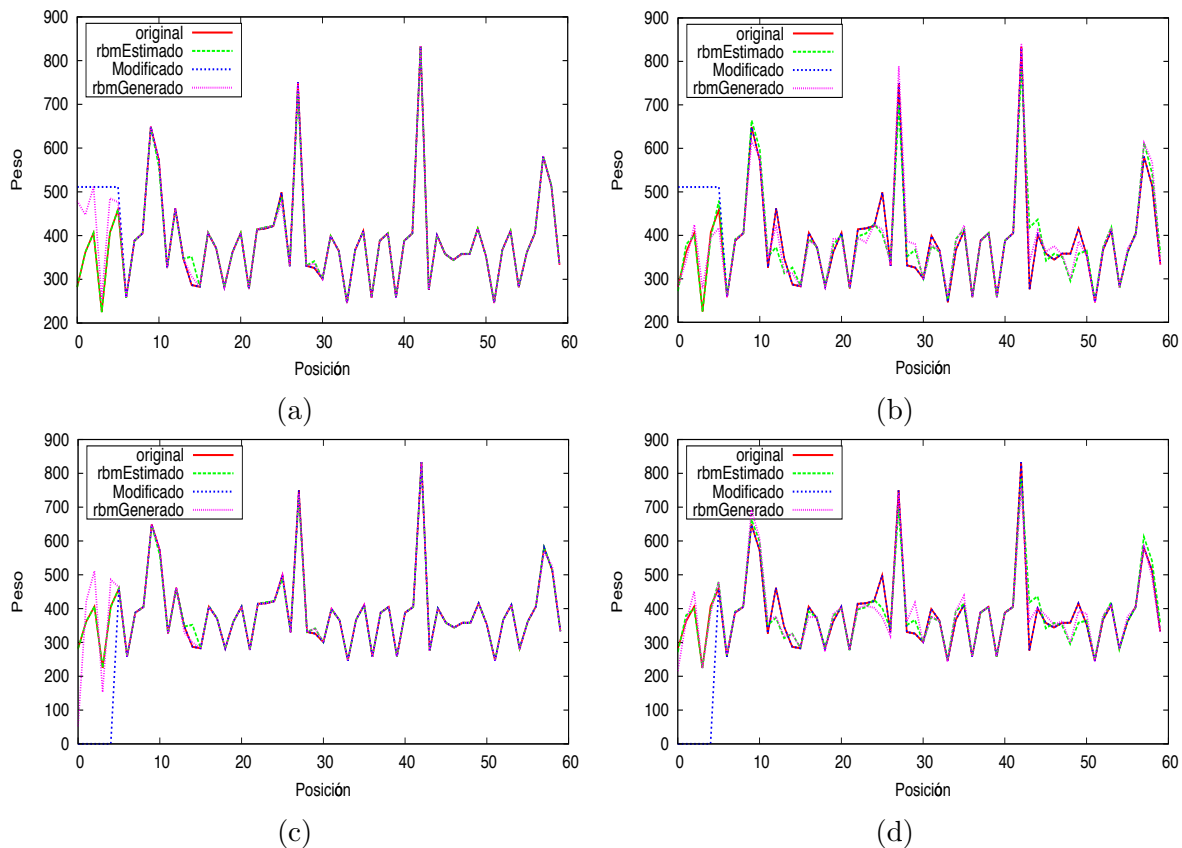


Figura 4.50: Reconstrucción de larva a partir del segmento medio con RBM y aprendizaje profundo (a) Reconstrucción con $RBM13$ utilizando \hat{S}_μ , (b) Reconstrucción con $DBN14$ utilizando \hat{S}_μ , (c) Reconstrucción con $RBM13$ utilizando ceros, (d) Reconstrucción con $DBN14$ utilizando ceros

4.9. Ajuste de larvas generadas a la imagen

Utilizando seis hipótesis de larva $\mathbf{v}_i^U \forall i \in \{1, \dots, 6\}$ sin ser alteradas del conjunto de prueba \mathcal{C}^P en combinación con los modelos de forma $RBM13$ y $DBN14$ se obtienen las hipótesis de corrección $\mathbf{v}_i^O \forall i \in \{1, \dots, 6\}$ para cada modelo, las cuales son ajustadas posteriormente a la imagen utilizando el algoritmo 3, dando como resultado los contornos mostrados en la figura 4.51, donde inicialmente se pinta de rojo el contorno \mathbf{v}_i^U y posteriormente en verde el contorno de \mathbf{v}_i^O con el objetivo de observar como cambia el segmento sugerido por la RBM comparado con el segmento original.

Es posible observar que los contornos generados por la RBM así como por la DBN cubren los contornos originales. Sin embargo se observa que algunos segmentos no se alinean al borde de la larva dando como resultado picos salidos del contorno, en especial para la

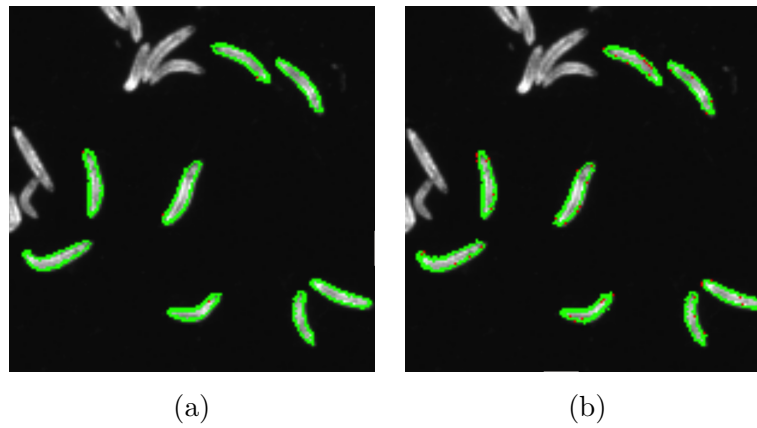


Figura 4.51: Hipótesis de corrección ajustadas a la imagen. Rojo para \mathbf{v}_i^{Ω} , verde para \mathbf{v}_i^{Ω} (a) Hipótesis generada por *RBM13*, (b) Hipótesis generada por *DBN14*

DBN. Esto es debido al error introducido por la RBM/DBN binaria en la estimación de los pesos tal y como se muestra en las curvas de deterioro y generación de la subsección 4.8.4, donde se observa como el cambio de bits ocasiona un error en algunos casos mayor a 30 unidades en el valor decimal de los pesos y por ende los segmentos generados pierden la alineación con los segmentos previos.

4.10. Oclusión

Debido a que el presente trabajo utiliza un método de ajuste de larvas a la imagen mediante las matrices de transformación isométrica no es posible realizar una evaluación directa del rendimiento del modelo de forma en casos específicos de oclusión. No obstante, la oclusión puede ser considerada un caso de larva incompleta, en donde la región “faltante” únicamente contiene información de la larva traslapada que es una variante a los casos en los cuales se utilizan ceros o el segmento medio como se estudia en secciones anteriores.

Sin embargo, con el objetivo de ilustrar la tendencia en la hipótesis de corrección generada en estos casos por el modelo de forma, se crean dos larvas sintéticas \mathbf{v}^{azul} y \mathbf{v}^{negro} traslapadas modeladas como elipses, estas son ilustradas en la figura 4.52a, donde para ambos casos se conoce la secuencia de 64 puntos que describen el contorno de las larvas individuales. Utilizando la información de dichos contornos se genera un contorno de traslape como el ilustrado en la figura 4.52b y su representación en pesos \mathbf{v}^{tras} .

El contorno de traslape de 64 puntos $p^{(x,y)}$ de longitud está compuesto en un 42% por puntos del contorno correspondiente a \mathbf{v}^{azul} y el restante 58% por puntos del contorno correspondiente a \mathbf{v}^{negro} . Finalmente, \mathbf{v}^{azul} , \mathbf{v}^{negro} y \mathbf{v}^{tras} son utilizados como hipótesis de larva en *RBM13* y *RBM14*, dando como resultado las hipótesis de corrección de la figura 4.53 y 4.54.

Para las hipótesis de corrección de las larvas sintéticas \mathbf{v}^{azul} y \mathbf{v}^{negro} la RBM logra modelar

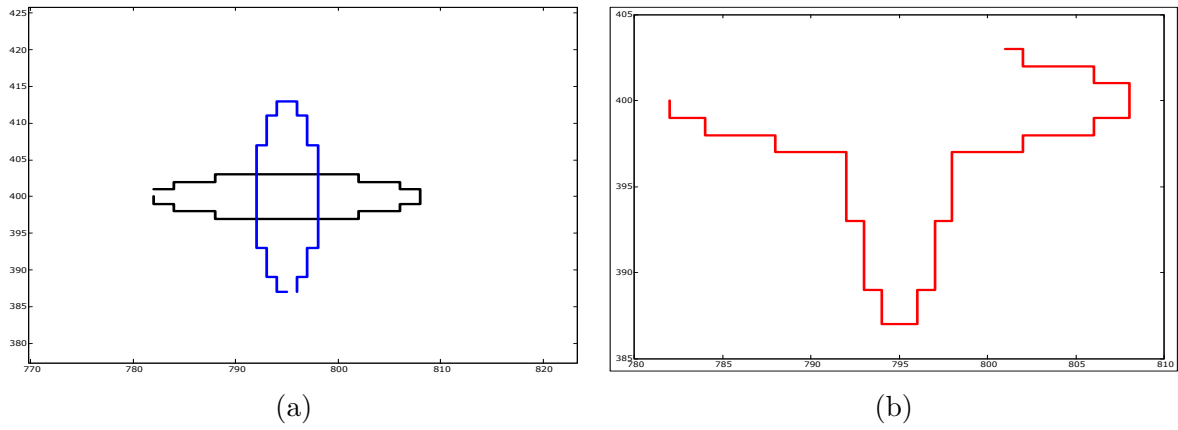
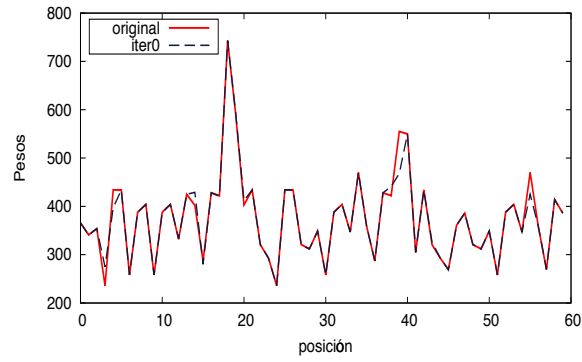
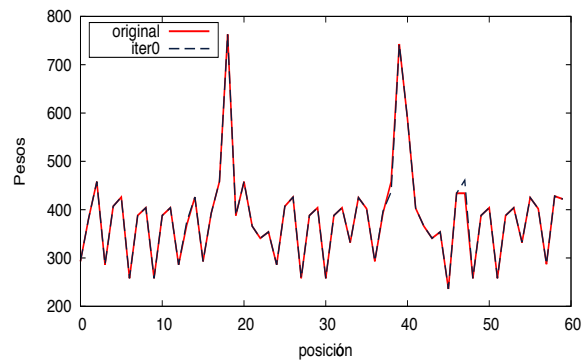


Figura 4.52: Larvas sintéticas para la evaluación de oclusión (a) Larvas \mathbf{v}^{azul} y \mathbf{v}^{negro} , (b) Contorno de traslape \mathbf{v}^{tras} , compuesto en sentido antihorario por 14 puntos de \mathbf{v}^{negro} , 27 puntos de \mathbf{v}^{azul} y 23 puntos de \mathbf{v}^{negro}

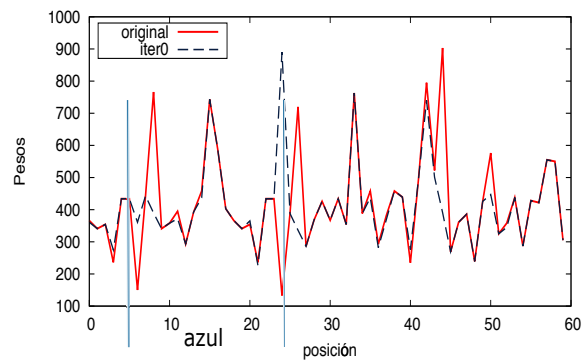
la probabilidad conjunta descrita por las unidades visibles de entrada y la DBN no, sin embargo esto se puede deber a que las larvas son sintéticas fueron aproximadas o modeladas como elipses. Por otra parte, para $\eta = 9$ puntos $p^{(x,y)}$ por segmento, el traslape está aproximadamente compuesto inicialmente por dos segmentos (6 pesos) de \mathbf{v}^{negro} , seis segmentos (18 pesos) de \mathbf{v}^{azul} y seis segmentos (18 pesos) de \mathbf{v}^{negro} con un recorrido cíclico en el complemento. Se observa en la figura 4.53c y 4.54c que tanto la RBM como para la DBN modifican los pesos que se encuentran de la posición 6 a la posición 24 correspondientes al traslape o región de \mathbf{v}^{azul} lo cual podría significar una corrección al traslape.



(a)

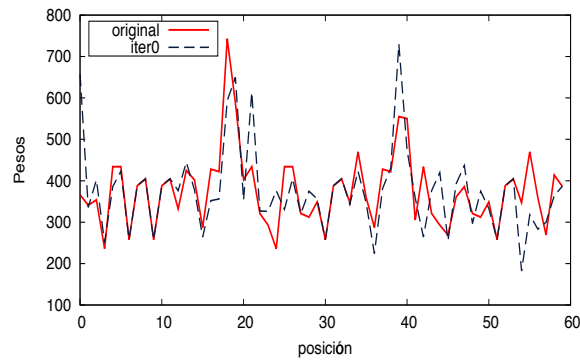


(b)

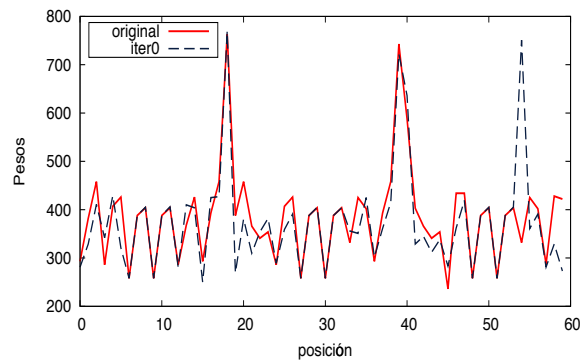


(c)

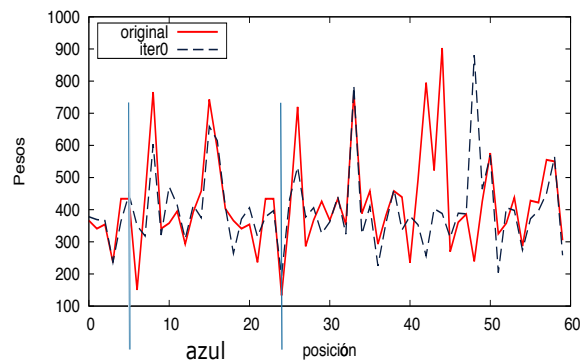
Figura 4.53: Hipótesis de corrección de la RBM para larvas sintéticas y segmento traslapado (a) Corrección de RBM_{13} para \mathbf{v}^{negro} , (b) Corrección de RBM_{13} para \mathbf{v}^{azul} , (c) Corrección de RBM_{13} para \mathbf{v}^{tras}



(a)



(b)



(c)

Figura 4.54: Hipótesis de corrección de la DBN para larvas sintéticas y segmento traslapado (a) Corrección de DBN_{14} para \mathbf{v}^{negro} , (b) Corrección de DBN_{14} para \mathbf{v}^{azul} , (c) Corrección de DBN_{14} para \mathbf{v}^{tras}

Capítulo 5

Conclusiones

En el presente trabajo se proporciona una estrategia para el modelado de larvas basado en modelos probabilísticos que persiguen la reconstrucción parcial de las mismas. Se comprobó que el uso de una representación de forma basada en los segmentos traslapados que componen el contorno en conjunto con el análisis de Procrustes dan como resultado una representación invariante a la escala, rotación y traslación.

Por otra parte, se realizó una reducción de un 83 % de la dimensionalidad de la representación de larva mediante el uso de 3 componentes principales que representan un 85 % de la variación descrita por los segmentos de larva, los cuales no describen una distribución gaussiana pero son descritos correctamente mediante PCA.

Se exploró el uso de las máquinas restringidas de Boltzmann y su variante de aprendizaje profundo (DBN) para la generación del modelo de forma. Para el entrenamiento de ambas se obtuvo que un incremento en la cantidad de unidades ocultas da como resultado una menor entropía cruzada entre larvas reales y generadas por el modelo, no obstante, dicho incremento requiere un mayor tiempo de entrenamiento y por ende un mayor costo computacional. De igual forma se comprobó que durante el entrenamiento del modelo las representaciones de larva en código Gray y en binario no alteran los valores de entropía cruzada obtenidos ya que son los mismos. Sin embargo, el uso de un cambio de escala lineal en las RBM permite disminuir el valor de entropía cruzada al final del entrenamiento, contrario a la DBN, en donde un mayor uso de bits causado por un cambio de escala mediante blanqueo de datos y función de error disminuye la entropía cruzada.

Por otra parte, para la reconstrucción de larvas se comprobó que al utilizar la representación mediante código Gray el cambio de bits y la distancia Euclideana resultante de la larva generada a la larva original es menor que al utilizar una representación únicamente binaria y que a su vez mejora las capacidades de generación descritas por la RBM y la DBN. Se obtuvo también que dicha distancia euclideana es menor durante la utilización de un cambio de escala por mapeo lineal que cuando se utiliza un mapeo por blanqueo de datos y función de error, indicando que tanto la RBM como la DBN describen mejor la probabilidad conjunta del contorno de larva cuando menos bits o unidades visibles están

cambiando. Similarmente, para el caso de divergencia contrastante se evidencia que el uso de una cadena de Gibbs no persistente de un solo paso produce un mejor deterioro en bits, no obstante en cuanto a distancia euclideana se refiere PCD produce una menor distancia después de 3 iteraciones.

Se demostró que el incremento de unidades ocultas tanto en la RBM como en la DBN reduce el deterioro pero no cambia su tendencia creciente, no obstante sí disminuye la distancia entre las larvas reales y las larvas generadas cuando larvas completas son presentadas. De igual forma un aumento en la cantidad de unidades oculta mejora las capacidades de regeneración de larvas incompletas tanto de la RBM como de la DBN. Un resultado similar se obtuvo al aumentar la cantidad de épocas de entrenamiento sin embargo en el caso de las épocas de entrenamiento dicha mejora no es significativa.

Finalmente, se comprobó que la estrategia de modelado de larvas propuesta en el presente trabajo mediante RBM es capaz de modelar en su mayoría la distribución conjunta descrita por los contornos de la larvas individuales pero algunos de los pesos no son aproximados con la exactitud deseada y que si se desean aprovechar las capacidades generativas del modelo es necesario utilizar el aprendizaje profundo (DBN).

El cambio de bits presente en las larvas generadas por las RBM/DBN binarias afecta las posiciones más y menos significativas de los pesos que representan a las larvas y por tanto afecta el ajuste posterior del segmento reconstruido a la imagen y por ende la forma descrita por el contorno total. Se recomienda explorar el uso de RBM/DBN no binarias con el objetivo de eliminar dicha sensibilidad del ajuste a la imagen ocasionada por el cambio de bits así como el desarrollo de un método de ajuste a la imagen más robusto que considere la información contenida en la imagen durante dicho proceso y que permita indicar sobre la imagen las hipótesis generadas por el modelo en los casos de oclusión.

Bibliografía

- [1] P Alvarado. *Señales y Sistemas. Fundamentos Matemáticos*. 2008.
- [2] Pablo Alvarado-Moya. Shape analysis of vermiform structures. Technical Report Research Stay at HCI. 2013.
- [3] Kirk Baker. Singular Value Decomposition Tutorial. *Linguistics Department, The Ohio State University*, 2005, 2005.
- [4] Adriano Barra, Alberto Bernacchia, Enrica Santucci, and Pierluigi Contucci. On the equivalence of Hopfield Networks and Restricted Boltzmann Machines. 2011. URL <http://arxiv.org/abs/1105.2790>.
- [5] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.
- [6] Yoshua Bengio. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.
- [7] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.
- [8] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1 edition, January 1996. URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0198538642>.
- [9] CM Bishop and NM Nasrabadi. *Pattern recognition and machine learning*, volume 4. 2006. URL <http://www.library.wisc.edu/selectedtocs/bg0137.pdf>.
- [10] Eran Borenstein. Combining top-down and bottom-up segmentation. In *In Proceedings IEEE workshop on Perceptual Organization in Computer Vision, CVPR*, 2004.

- [11] Leon Bottou. Stochastic Gradient Descent Tricks. *Neural Networks: Tricks of the Trade*, 1(1):421–436, 2012. URL http://link.springer.com/chapter/10.1007/978-3-642-35289-8_25.
- [12] Tim Cootes, Er Baldock, and J Graham. An introduction to active shape models. *Image Processing and Analysis*, pages 223–248, 2000. URL http://person.hst.aau.dk/lasse/teaching/IACV/doc/asm_overview.pdf.
- [13] Guillaume Desjardins, Aaron Courville, Yoshua Bengio, Pascal Vincent, and Olivier Delalleau. Tempered Markov Chain Monte Carlo for training of Restricted Boltzmann Machines. *International Conference on Artificial Intelligence and Statistics*, 2010.
- [14] V Dumoulin and Ij Goodfellow. On the Challenges of Physical Implementations of RBMs. *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2013. URL <http://arxiv.org/abs/1312.5258>.
- [15] S Anna Durai and E Anna Saro. Image Mapping with Cumulative Distribution Function for Quick Convergence of Counter Propagation Neural Networks in Image Compression. 2006.
- [16] S. M. Ali Eslami, Nicolas Heess, Christopher K. I. Williams, and John Winn. The Shape Boltzmann Machine: A Strong Model of Object Shape. *International Journal of Computer Vision*, November 2013. URL <http://link.springer.com/10.1007/s11263-013-0669-1>.
- [17] Johannes Fauske. An empirical study of the maximum pseudo-likelihood for discrete Markov random fields. *Norwegian University of Science and Technology*, 2009.
- [18] Joel Feldman. Mean and Variance of Binomial Random Variables, 2000.
- [19] Asja Fischer and Christian Igel. Training restricted Boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39, 2014.
- [20] Amir Globerson and David Sontag. PGM Lecture Notes : Pseudo-likelihood. 1:1–2, 2012.
- [21] Raul Godoyherrera, Barrie Burnet, Kevin Connolly, and John Gogarty. The development of locomotor activity in *Drosophila melanogaster* larvae. 52:63–75, 1984.
- [22] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
- [23] Xin He. An outlier detection based approach for PCB testing. *Test Conference, 2009. ITC 2009. International*, 2011. URL http://digitool.library.colostate.edu//exlibris/dtl/d3_1/apache_media/L2V4bGlicmlzL2R0bC9kM18xL2FwYWNoZV9tZWRpYS8xMTk4NTE=.pdf.

- [24] Aaron Hertzmann. Learning 3D Mesh Segmentation and Labeling. (July):1–12, 2010.
- [25] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006. URL <http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed&uid=16873662&cmd=showdetailview&indexed=google>.
- [26] Geoffrey Hinton. A Practical Guide to Training Restricted Boltzmann Machines A Practical Guide to Training Restricted Boltzmann Machines. *Computer*, 9(3):1, 2010. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.170.9573&rep=rep1&type=pdf>.
- [27] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- [28] Fuchun Huang and Yosihiko Ogata. Generalized pseudo-likelihood estimates for Markov random fields on lattice, 2002.
- [29] Zhen Huang, Jinyu Li, Chao Weng, and Chin-Hui Lee. Beyond Cross-entropy: Towards Better Frame-level Objective Functions For Deep Neural Network Training In Automatic Speech Recognition. (September):1214–1218, 2014.
- [30] Konstantin G Iliadi. The genetic basis of emotional behavior: has the time come for a Drosophila model? *Journal of neurogenetics*, 23(1-2):136–146, 2009.
- [31] David Joyner and Jim McShea. Some applications of graph theory. 2010.
- [32] R. Loganathan K. P. Soman. *Elementary Real and Complex Analysis*. Prentice-Hall of India Pvt.Ltd, 2009.
- [33] Andrew Kae, Kihyuk Sohn, Honglak Lee, and Erik G. Learned-Miller. Augmenting crfs with boltzmann machine shape priors for image labeling. In *CVPR*. IEEE, 2013. URL <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#KaeSLL13>.
- [34] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. pages 1–60, 2009.
- [35] B. Afonso M. Zlatic F. A. Hamprecht L. Fiaschi, G. Konstantin. *International Symposium Biomedical Imaging, accepted oral*.
- [36] Nicolas Le Roux and Yoshua Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural computation*, 20(6):1631–1649, 2008.
- [37] Nikolai G Lehtinen. “Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Error functions”. 2010.
- [38] Kun Liu and Ramjee Prasad. Performance analysis of differential chain coding. *European Transactions on Telecommunications*, 3(4):323–330, 1992. URL <http://dx.doi.org/10.1002/ett.4460030404>.

- [39] Sanchez Mandl, Bowley. *The Statistical Theory of Thermodynamics*. chapter 3.
- [40] Benjamin M. Marlin, Kevin Swersky, Bo Chen, and O De Freitas. *Inductive principles for restricted boltzmann machine learning*, 2010.
- [41] J.R. Martin. *A portrait of locomotor behaviour in Drosophila determined by a video-tracking paradigm*. 67:207–219, 2004.
- [42] Bryan S Morse. *Lecture 7 : Shape Description (Contours)*. 2000.
- [43] José Pablo Alvarado Moya. *Segmentation of color images for interactive 3D object retrieval*. PhD thesis, RWTH Aachen University, 2004. URL <http://sylvester.bth.rwth-aachen.de/dissertationen/2004/121/index.htm>. <http://d-nb.info/971943745>.
- [44] R J Needs. *Thermal and Statistical Physics*. 2007.
- [45] Asger Munk Nielsen and Computer Science. *Number systems and codes*. (August), 1997.
- [46] NVIDIA Corporation. *NVIDIA CUDA Compute Unified Device Architecture Programming Guide*. NVIDIA Corporation, 2007.
- [47] Rosalind W Picard. *From Lecture: Pattern Recognition and Analysis, Topic: Decorrelating and then Whitening data. Lecture: PATTERN RECOGNITION AND ANALYSIS*, (1):2–3, 2010. URL <http://courses.media.mit.edu/2010fall/mas622j/whiten.pdf>.
- [48] Sm Rajtmajer. *Introduction to Markov Random Fields*. pages 1–15, 2012. URL http://www.mne.psu.edu/ray/me577/me577RandomFields_Sarah.pdf.
- [49] Mi Ribeiro. *Gaussian probability density functions: Properties and error characterization. Institute for Systems and Robotics, Technical Report*, (February):1–30, 2004. URL http://hans.fugal.net/comps/papers/ribeiro_2004.pdf.
- [50] Amy Ross. *Procrustes Analysis. Food Quality and Preference*, 5(1-2):115–120, 1994. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.119.2686>.
- [51] Jonathon Shlens. *A Tutorial on Principal Component Analysis*. 2005.
- [52] Lindsay I Smith. *A tutorial on Principal Components Analysis Introduction. Statistics*, 51:52, 2002. URL <http://www.mendeley.com/research/computational-genome-analysis-an-introduction-statistics-for-biology-and-health/>.
- [53] P. Smolensky. *Information Processing in Dynamical Systems: Foundations of Harmony Theory*. In D. E. Rumelhart and J. L. McClelland and others, editor, *Parallel Distributed Processing: Volume 1: Foundations*, pages 194–281. MIT Press, 1986.

- [54] B Soibam, L Chen, G W Roman, and G H Gunaratne. Exploratory activity and habituation of *Drosophila* in confined domains. 1803:1787–1803, 2014.
- [55] Roger Boyle Sonka, Milan, Vaclav Hlavac. *Image Processing, Analysis, and Machine Vision: A Matlab Companion*. 2008.
- [56] Mikkel B. Stegmann and David Delgado Gomez. A brief introduction to statistical shape analysis, 2002. Informatics and Mathematical Modelling, Technical University of Denmark, DTU.
- [57] Z. Su. Statistical shape modelling: automatic shape model building. 2011. URL <http://discovery.ucl.ac.uk/1213097/>.
- [58] Soo Tang Tan. *Finite mathematics for the managerial, life, and social sciences/*. Brooks/cole, California, 6th edition, 2000.
- [59] Tijmen Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. *ICML; Vol. 307*, page 7, 2008.
- [60] Hasari Tosun. Atomic Energy Models for Machine Learning: Atomic Restricted Boltzmann Machines. 2014.
- [61] Marta Vilchez. Variables aleatorias discretas. Notas de curso. *Instituto Tecnológico de Costa Rica*, 2009.
- [62] Christian Walck and Particle Physics Group. Hand-book on statical distributions for experimentalists. *Hand-book on statical distributions for experimentalists*, (September), 2007. URL <http://www.stat.rice.edu/~dobelman/textfiles/DistributionsHandbook.pdf>.
- [63] Christopher K I Williams and John Winn. A Strong Model of Object Shape. *International Journal of Computer Vision manuscript*.
- [64] Jacqueline Wroughton and Tarah Cole. Distinguishing Between Binomial , Hypergeometric and Negative Binomial Distributions. *Journal of Statistics Education*, 21(1):1–16, 2013.

Apéndice A

Vectores propios del conjunto de segmentos de larva

Tabla A.1: Vectores propios del conjunto de segmentos

vector	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	x_5	y_5	x_6	y_6	x_7	y_7	x_8	y_8	x_9	y_9
ρ_1	0.18	-0.42	0.38	-0.02	-0.51	-0.09	-0.07	-0.02	0.00	0.31	-0.32	-0.21	-0.06	-0.02	-0.00	-0.01	0.33	0.16
ρ_2	0.51	-0.31	-0.22	-0.35	0.08	-0.13	-0.14	-0.18	0.20	-0.17	0.03	-0.04	-0.03	-0.00	0.00	-0.34	-0.01	-0.47
ρ_3	0.12	-0.10	0.35	0.03	-0.09	-0.01	-0.03	-0.05	0.10	-0.32	0.53	0.47	0.29	-0.06	0.01	-0.01	0.33	0.16
ρ_4	0.23	0.01	0.17	0.22	-0.01	0.24	0.24	0.50	-0.48	0.14	0.05	0.09	0.07	-0.02	0.01	-0.33	-0.01	-0.36
ρ_5	0.09	0.09	0.17	-0.06	0.21	0.05	0.05	0.15	-0.09	-0.40	0.16	-0.39	-0.57	0.29	-0.06	-0.01	0.33	0.14
ρ_6	-0.04	0.25	0.33	0.46	0.01	0.20	0.09	-0.45	0.40	0.10	-0.02	-0.07	-0.12	0.07	-0.02	-0.33	-0.01	-0.23
ρ_7	0.12	0.16	0.02	-0.10	0.29	0.20	0.13	-0.03	0.05	-0.09	-0.23	-0.21	0.19	-0.60	0.43	-0.01	0.33	0.10
ρ_8	-0.25	0.29	0.16	0.06	-0.12	-0.57	-0.48	0.13	-0.16	-0.16	-0.02	-0.10	0.01	-0.19	0.15	-0.33	-0.01	-0.10
ρ_9	0.16	0.23	-0.00	0.01	0.31	-0.03	-0.25	-0.00	-0.05	0.13	-0.24	0.03	0.27	0.02	-0.70	-0.02	0.33	0.05
ρ_{10}	-0.32	0.12	0.01	-0.39	-0.32	-0.01	0.52	0.05	0.12	-0.27	-0.11	-0.07	0.14	-0.09	-0.34	-0.33	-0.02	0.03
ρ_{11}	0.13	0.33	-0.14	-0.07	-0.00	-0.34	0.29	0.04	0.09	0.18	-0.16	0.18	0.16	0.53	0.36	-0.02	0.33	-0.02
ρ_{12}	-0.24	-0.04	0.08	-0.33	0.06	0.51	-0.39	-0.10	-0.16	-0.04	-0.18	0.10	0.16	0.37	0.22	-0.33	-0.02	0.15
ρ_{13}	-0.02	0.26	-0.36	-0.07	-0.36	0.18	-0.08	-0.27	-0.26	0.10	0.05	0.33	-0.44	-0.23	-0.04	-0.02	0.33	-0.11
ρ_{14}	-0.10	-0.08	0.06	-0.21	0.30	-0.09	0.02	0.29	0.33	0.46	0.13	0.28	-0.36	-0.17	-0.01	-0.33	-0.02	0.25
ρ_{15}	-0.25	-0.05	-0.35	0.10	-0.18	0.20	-0.19	0.31	0.33	0.18	0.36	-0.37	0.22	0.05	0.01	-0.02	0.33	-0.19
ρ_{16}	0.04	-0.09	-0.18	0.04	0.10	-0.16	0.17	-0.41	-0.39	0.26	0.40	-0.31	0.16	0.01	-0.01	-0.33	-0.02	0.34
ρ_{17}	-0.51	-0.50	-0.05	0.18	0.33	-0.16	0.15	-0.13	-0.16	-0.09	-0.17	0.16	-0.05	0.01	-0.00	-0.02	0.33	-0.28
ρ_{18}	0.16	-0.16	-0.43	0.49	-0.10	0.02	-0.03	0.16	0.13	-0.32	-0.27	0.13	-0.02	0.02	0.00	-0.33	-0.02	0.41

