

UNIVERSIDAD MAYOR DE SAN ANDRÉS
Facultad de Ciencias Puras y Naturales
Carrera de Informática



TESIS DE GRADO

“RECONOCIMIENTO DE CROMOSOMAS HUMANOS EN IMÁGENES
BASADO EN UNA RED NEURONAL DE RETROPROPAGACION”

Para optar el título de licenciatura en Informática.
Mención Ingeniería de Sistemas Informáticos.

POSTULANTE : RODRIGO SIMON QUELCA SIRPA
TUTOR : MSc. FATIMA DOLZ SALVADOR
REVISOR : LIC. BRIGIDA CARVAJAL BLANCO

LA PAZ – BOLIVIA
2011

DEDICATORIA

A mí querido tío Francisco Sirpa, por todo el sacrificio y la colaboración que diste en mi formación. Por el apoyo moral y por todas las recomendaciones que me diste.

A mi querida tía Basilia Acarapi por brindarme tanto, apoyo y paciencia para la conclusión de mis estudios.

A mi primo Alex por haber compartido tantos años de buenos y malos momentos apoyándome.

A mi querida esposa Teresa y el tesoro de mi vida mi hijo Kevin que son el motivo de mi diario vivir.

AGRADECIMIENTOS

A la M.sc. Fatima Dolz Salvador por su constante apoyo en la estructura del y desarrollo del proyecto, por su guía en la aplicación de la metodología, por sus consejos en la aplicación de redes neuronales y por todo el tiempo que dedico a la revisión y corrección de la presente tesis.

A mi revisora Lic. Brígida Carvajal Blanco por el apoyo incondicional, por su comprensión y su tiempo que ha dedicado a la revisión, corrección y consejos oportunos para el diseño y desarrollo de la presente tesis.

También agradecer a todos los docentes de la carrera de la informática, por el esfuerzo y dedicación que ponen al enseñarnos, para que seamos buenos profesionales.

A mis compañeros de estudio, con los que compartí momentos maravillosos durante mi vida universitaria.

A Fernando Ontiveros, Gerente de Desarrollo de la empresa COLOSA Inc. donde actualmente trabajo y a Enrique Ponce, por toda la confianza, paciencia y por todas sus consideraciones.

RESUMEN

Citogenética es el estudio de las características cromáticas de los cromosomas humanos, todo con el fin de encontrar, detectar anomalías en la distribución de un cariotipo humano. Actualmente el tiempo estimado para el estudio y la clasificación de los cromosomas es aproximadamente 15 días, lo cual representa una gran inversión de tiempo y esfuerzo para los escasos profesionales de esta área en nuestro medio. Además que la individualización de cada cromosoma presente en un cariotipo induce a errores de clasificación, esto por diversos factores, ya sea cansancio de la vista, poca experiencia del profesional, y otros.

Este trabajo propone el análisis de imágenes de cromosomas que se encuentra en bandeado G, de forma automática. Luego se realiza la clasificación tomando en cuenta la etapa previa de la segmentación y la detección de bordes de las imágenes cromosómicas, todo esto para obtener una muestra completamente representativa. La etapa de reconocimiento requiere de una imagen de entrada, para procesarlo en la red neuronal previamente entrenada, y esta devuelve resultados de aproximación con respecto a los tipos de cromosomas.

ÍNDICE

CAPITULO I INTRODUCCIÓN

1.1. INTRODUCCIÓN	2
1.2. ANTECEDENTES	4
1.3. PROBLEMA	7
1.4. OBJETIVOS	8
1.4.1. OBJETIVO GENERAL	8
1.4.2. OBJETIVOS ESPECÍFICOS	8
1.5. HIPÓTESIS	9
1.6. JUSTIFICACIONES	9
1.6.1 CIENTÍFICA	9
1.6.2. ECONÓMICA	9
1.6.3. SOCIAL	9
1.7. ALCANCES Y LIMITES DEL TRABAJO	10
1.8. MÉTODOS Y TÉCNICAS	10

CAPITULO II RED NEURONAL DE RETROPROPAGACION

2.1. INTRODUCCIÓN	14
2.2. NEURONA BIOLÓGICA	15
2.3. REDES NEURONALES ARTIFICIALES	16
2.4. ARQUITECTURA DE UNA RED NEURONAL	17
2.5. NEURONA ARTIFICIAL	18
2.6. BACKPROPAGATION	19
2.6.1. ARQUITECTURA	17
2.6.2. APRENDIZAJE (REGLA DELTA GENERALIZADA)	20

CAPITULO III TRATAMIENTO DIGITAL DE IMÁGENES

3.1. IMÁGENES DIGITALES	29
3.2. RESOLUCIÓN	29
3.3. SEGMENTACIÓN POR UMBRAL	30

3.4 FUNDAMENTOS	31
3.2.4. UMBRALIZACIÓN BINARIA	32
3.5. UMBRALIZACIÓN ÓPTIMA	34
3.6. SELECCIÓN DE UMBRAL BASADA EN LAS CARACTERÍSTICAS DEL LÍMITE	37
3.8. UMBRALIZACIÓN MULTINIVEL	38
3.8.1. UMBRALIZACIÓN MULTINIVEL DE LUMINANCIA	39
2.2.8.2. UMBRALIZACIÓN MULTINIVEL DE COMPONENTES DE COLOR	42
CAPITULO IV CONSTRUCCIÓN DE LA RED NEURONAL DE RETROPROPAGACION	
<i>4.1. DISEÑO DE LA RED NEURONAL</i>	44
4.2. CODIFICACIÓN DE LA NEURONA ARTIFICIA	44
4.3. MODELACIÓN Y CODIFICACIÓN DE LA ARQUITECTURA	45
<i>4.4. CODIFICACIÓN DE LA PROPAGACIÓN HACIA ATRÁS</i>	50
4.5. CODIFICACIÓN PARA LA MODIFICACIÓN DE PESOS	52
4.6. CODIFICACIÓN DEL APRENDIZAJE	53
4.7. CODIFICACIÓN PARA CALCULAR LAS TASAS DE RECONOCIMIENTO Y ERRORES CUADRÁTICOS MEDIOS	55
4.8. TRATAMIENTO DE LA IMAGEN	54
4.8.1. DETECCIÓN DE BORDES DE LA IMAGEN	57
4.8.2. ALGORITMO DE PREWIT PARA LA DETECCIÓN DE BORDES.	58
4.8.3. ADELGAZAMIENTO DE LOS BORDES	59
4.8.3. OBTENCIÓN DE MUESTRAS	59
4.9. DESARROLLO DEL PROTOTIPO	61
4.9.1. ÁMBITO DEL SISTEMA DE CLASIFICACIÓN DE CROMOSOMAS	61
4.9.2. MODELO DE DESARROLLO	62
4.9.3. RECURSOS	61
4.9.3.1. RECURSOS DE SOFTWARE REUTILIZABLE	62
4.9.3.2 RECURSOS DE ENTORNO	62
4.9.3.2 RECURSOS DE ENTORNO	62
4.9.4 ANÁLISIS Y DISEÑO DEL PROTOTIPO	63
4.9.4.1 MODELO DEL PROTOTIPO	63

4.9.4.1.1. CASOS DE USO	64
4.9.4.1.2. EL MODELO OBJETO – RELACIÓN	67
4.9.4.1.3. MODELO OBJETO – COMPORTAMIENTO	63
a) DIAGRAMAS DE SECUENCIA	68
b) DIAGRAMA DE ESTADOS	70
4.9.4.2. MODELADO DEL DISEÑO	70
4.9.4.3. DISEÑO DEL SISTEMA	71
4.9.4.4. DISEÑO DE INTERFACES	72
A) VENTANA PRINCIPAL	73
B) VENTANA DE ALTAS BAJAS Y MODIFICACIONES	73
C) VENTANA DE OPERACIONES CON IMÁGENES	74
D) VENTANA DE CONFIGURACIÓN DE LA RED NEURONAL	74
E) VENTANA DE PARÁMETROS DE LA RED NEURONAL	75
F) VENTANA DE CLASIFICACIÓN	76
4.10. MÉTRICAS DE CALIDAD PARA SISTEMAS ORIENTADOS A OBJETOS	76
a) Tamaño de la clase	76
b) Factor de acoplamiento FA	77
4.11. JUSTIFICACIÓN DE MÉTODOS	78
4.11.1. Redes Neuronales artificiales de retropropagacion	78
4.11.2 Lenguaje de programación java	79

CAPITULO V RESULTADOS PRUEBA EXPERIMENTAL

5.1. INTRODUCCIÓN	81
5.2. DATOS PARA EL ENTRENAMIENTO	82
5.3. PRUEBAS	86
5.4. EVALUACIÓN DE RESULTADOS	91

CAPITULO VI CONCLUSIONES Y

RECOMENDACIONES

6.1. CONCLUSIONES GENERALES	93
6.2. CUMPLIMIENTO DE OBJETIVOS	94
6.3. ESTADO DE LA HIPÓTESIS	96
6.4. RECOMENDACIONES	96
6.5. TRABAJOS FUTUROS	97

BIBLIOGRAFÍA	98
ANEXOS	
A. ÁRBOL DE PROBLEMAS	103
B. ÁRBOL DE OBJETIVOS	104
C. MATRIZ DEL MARCO LÓGICO	105
D. RESULTADOS DE RECONOCIMIENTO	106

ÍNDICE DE FIGURAS Y CUADROS

Figura. 1.1. Procedimiento actual.	4
Figura. 1.2.Etapas del reconocimiento de imágenes.	12
Figura. 2.1.Neurona Biológica	15
Figura. 2.2.Modelo Computacional de una neurona	18
Figura. 2.3.Funciones de activación	19
Figura. 2.4.Arquitectura de la Red Neuronal	20
Figura. 3.1.Valores de píxel:	29
Figura. 3.2.Este ejemplo ilustra la aplicación del análisis de formas a la localización de aneurismas en una angiografía ocular	31
<i>Figura. 3.3. - Histogramas de nivel de gris que se pueden dividir en: (a) un umbral sencillo y (b) umbrales múltiples.</i>	32
Figura. 3.4.Gráfica Del Umbral.	33
Figura. 3.5. Neurona Biológica.	34
Figura. 3.6.Segmentación de imagenes	38
Figura. 3.7.Histograma de clases	39
Figura. 3.8.Histogramas de imágenes	40
Figura. 3.9.Segmentación de imágenes	41
Figura. 4.1.MODELACIÓN COMPUTACIONAL, MEDIANTE ESTRUCTURA DE DATOS, DE LAARQUITECTURA DE LA RED NEURONAL DE RETROPROPAGACION.	47
Figura. 4.2.SIMPLIFICACIÓN EN LA ESTRUCTURA DE DATOS DEL MODELO COMPUTACIONAL	49
Figura. 4.3.Filtro de Prewitt	58
Figura. 4.4.Imagen Procesada	58
Figura. 4.5.Resultado de la segmentación	60
Figura. 4.6.Triangulo de las muestras.	61
Figura. 4.7.Modelo de desarrollo	62
Figura. 4.8.Diagrama de Casos de uso de Alto Nivel	62
Figura. 4.9.Diagrama de Casos de uso de Bajo Nivel	65
Figura. 4.10.Diagrama de Clases del Sistema de Sistema de Clasificación de Cromosoma	67
Figura. 4.11.Diagrama de Secuencias del Sistema.	69
Figura. 4.12.Diagrama de Secuencias. Caso de uso: Identificación de Cromosoma.	69
Figura. 4.13 Diagrama de Estados del Sistema CCROM	70.
Figura. 4.14.Componente Red Neuronal	71
Figura. 4.15.Componente CCROM	71
Figura. 4.16.Diagrama de Subsistemas	72
Figura. 4.17.Ventana Principal	73
Figura. 4.18.Ventana de altas, bajas y modificaciones.	74
Figura. 4.19.Ventana de operaciones con imágenes	74

Figura. 4.19. Ventana de configuración de la Red Neuronal y de entrenamiento	75
Figura. 4.20.Ventana de configuración de la red neuronal	75
Figura. 4.21.Ventana de Resultados finales	76
Figura. 5.1.Esquema de las fases de reconocimiento de cromosomas	81
Figura. 5.2. Parámetros de entrada a la Red Neuronal y error.	85
Tabla 4.1. Tamaño de la Clase	77
Tabla 4.2.Relación entre clases	77
Tabla 5.1.Conjunto de muestras del tipo 1 para entrenamiento	83
Tabla 5.2.Conjunto de muestras del tipo 2 para entrenamiento	84
Tabla 5.3.Conjunto de muestras del tipo 3 para entrenamiento	84
Tabla 5.4.Conjunto de muestras del tipo 3 para entrenamiento	85
Tabla 5.5.Resultados de Reconocimiento de cromosomas de tipo 1	87
Tabla 5.6.Resultados de Reconocimiento de cromosomas de tipo 2	88
Tabla 5.7.Resultados de Reconocimiento de cromosomas de tipo 3	89
Tabla 5.8. Resultados de Reconocimiento de cromosomas para el caso 2	90
Tabla 5.9.Tabla de porcentajes de clasificación	91



CAPITULO I

INTRODUCCIÓN

RESUMEN

En el presente capítulo se describe los problemas identificados en el reconocimiento de cromosomas humanos, se establecen los objetivos, se plantea la hipótesis, y se especifica las justificaciones de la presente investigación, los alcances y limitaciones del objeto de estudio y finalmente se presenta la metodología que se emplea.

1.1.INTRODUCCIÓN

Como es de conocimiento general, la medicina es una de las áreas más beneficiadas con la acelerada evolución de la Informática que ofrece sus diversos productos para la prevención enfermedades, pero también existen múltiples campos en los que se pueden aplicar el procesamiento digital de imágenes, como también las redes neuronales que brindan las mejores alternativas de solución a diferentes problemas.

La citogenética es el estudio de los cromosomas, su estructura y su herencia. El estudio citogenético cromosómico se ha desarrollado rápidamente en los últimos años. Hoy en día es utilizado como elemento de diagnóstico que permiten detectar un gran número de patologías que implican: enfermedades congénitas, patológicas, reproductivas y aquellas relacionadas con disturbios metabólicos.

Los datos estadísticos son muy claros al respecto:
Entre uno de cada 150 y uno de cada 160 recién nacidos tiene una anomalía cromosómica.
Al menos el 50% de los abortos espontáneos durante el primer trimestre tiene una anomalía cromosómica.
Aproximadamente el 2% de los embarazos en mujeres de más de 35 años tiene una anomalía cromosómica

El Estudio, análisis y clasificación de los cromosomas en el Instituto de Genética de la Universidad Mayor de San Andrés (IG-UMSA) en la actualidad es muy complejo, requiere la participación de especialistas y técnicos, que con su capacidad visual y experiencia reconocen y clasifican los cromosomas. Pero este proceso demora entre 15 a 20 días. Siendo el costo para el paciente aproximadamente 300 Bs.

La elaboración de la presente tesis está dirigida a desarrollar herramientas adecuadas, para que a partir del procesamiento digital de imágenes reales y el diseño de una red neuronal, la clasificación de cromosomas sea automática. Para que colaboren con la obtención del diagnóstico médico y se facilite la labor de los especialistas.

Por razones ya mencionadas se propone reducir los tiempos, y garantizar un procedimiento de análisis y estudio más comprensible para los especialistas. Esto se logrará desarrollando algoritmos de análisis y comparación de imágenes digitalizadas, que permitirán la detección de pequeñas variaciones entre imágenes adquiridas en intervalos de tiempo menores que los que se utilizan habitualmente para poder realizar estudios similares basados en el sistema de visión humana.

Para este propósito Redes Neuronales Artificiales (RNA) son las más calificadas, puesto que presentan una analogía con el funcionamiento del cerebro humano, de hecho fueron desarrolladas en base al funcionamiento del cerebro y la interconectividad de sus neuronas, además son capaces de aprender y generalizar en función de ejemplos representados en patrones que se le presenta a la misma. Si bien no se está cerca del completo entendimiento el funcionamiento del cerebro humano en el reconocimiento de formas y figuras, las expectativas son grandes y apasionantes [Vivaracho, 2001].

Por lo anterior, el uso de las Redes Neuronales Artificiales en el reconocimiento de imágenes y reconocimiento de patrones en imágenes bidimensionales monocromáticas como herramienta fundamental es realmente importante, puesto que poseen características que la sitúan como método ventajoso [Hilera, 2000]. Que se detallan en el Capítulo 2 y a además en el Capítulo 4 se aplica el modelo de Red Neuronal de retropropagación.

Además los métodos para el tratamiento digital de imágenes que se aplica a las imágenes tomadas como muestras, como ser la segmentación de una imagen por medio de su Umbral, el algoritmo de Prewitt para la obtención de las muestras más representativas de una imagen se verán con mayor detalle en el Capítulo 4.

Las aplicaciones con estas características se orientan a sistemas para la aplicación en medicina, laboratorios, universidades, institutos y consultorios. Como podemos ver los sistemas de apoyo a la medicina hoy en día son bastante comerciales, y por ello también

requeridos para brindar un buen servicio a pacientes y también obtener los mejores resultados.

1.2.ANTECEDENTES

El procedimiento actual de clasificación e identificación de cromosomas en el Instituto de Genética se representa en la Figura 1.1.

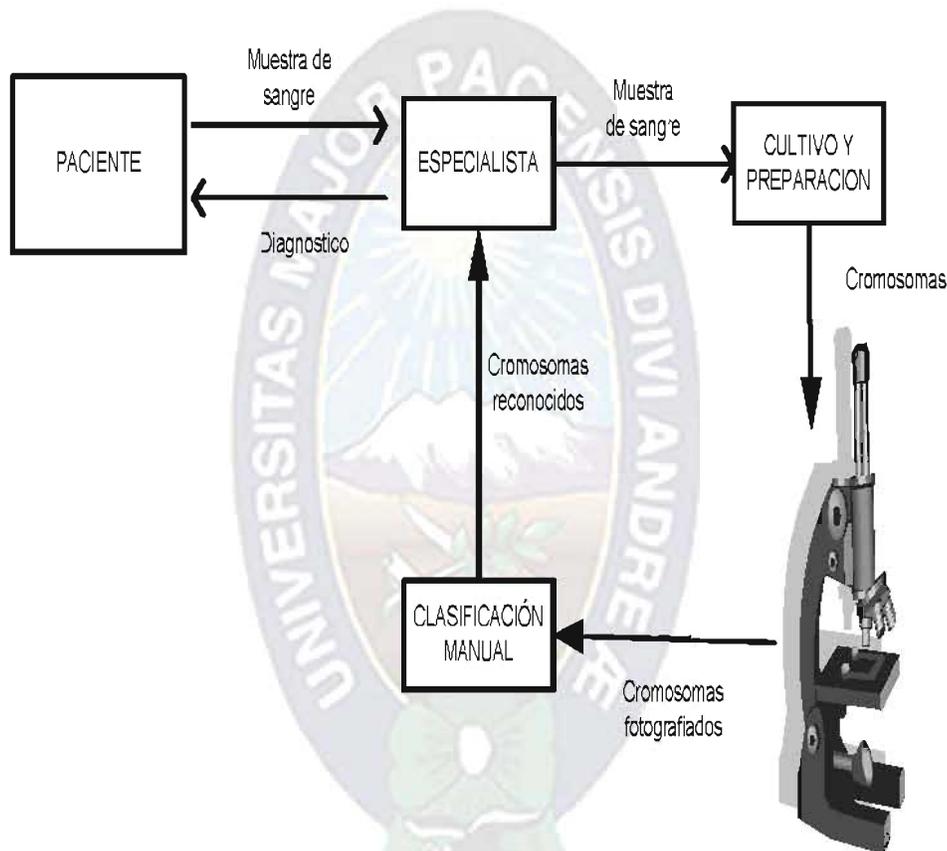


Figura. 1.1. Procedimiento actual.
Fuente: Elaboración propia

A la muestra se le aplica, el bandeo G cromosómico, técnica que permite la identificación morfológica de cada uno de los cromosomas de forma independiente y, por tanto, de sus alteraciones.

La muestra obtenida del paciente pasa por una etapa de cultivo y cuando se encuentra lista se procede al estudio microscópico, es ahí donde los especialistas reconocen de forma manual, dependiendo de las capacidades visuales para la individualización de cada uno de 46 los cromosomas se emite el diagnóstico médico.

La aplicación de redes neuronales artificiales en el reconocimiento de imágenes médicas, es la mejor alternativa de solución para la detección y prevención de enfermedades. Hoy en día existen sistemas que implementan las redes neuronales y son exitosos en el campo en que se desenvuelven, pero que además tienen un costo de venta alto.

- LEICAQ550CWdesarrollada porLEICA MICROSYSTEMS, España en 2007 Estación de trabajo citogenética y Soluciones de aplicaciones para citogenética Con los siguientes módulos:

Leica Chantal Una solución aplicativa de procesamiento de cariotipos programable y automatizada para el análisis de cromosomas humanos y no humanos, que se sirve de la tecnología de las redes neuronales, para incrementar la velocidad y la fiabilidad.

LeicaQSAT Una solución eficaz para la adquisición de imágenes y la introducción de datos en las configuraciones de red: constituye una alternativa económica a las técnicas fotográficas tradicionales

LeicaQFISH Una solución de formación de imágenes y documentación FISH, que se puede utilizar para la simple adquisición de imágenes de muestras fluorescentes o para la experimentación FISH en profundidad en el área de la investigación.

LeicaQCGH Una aplicación de procesado y medición de a imagen que realiza la comparación de medidas de perfil densitométrico para la hibridación comparativa de genomas.

LeicaMCKFISHmulticromático facilita la diagnosis rápida e inequívoca de aberraciones cromosómicas, tales como translocaciones complejas, mediante la identificación de cromosomas marcados diferencialmente.

LeicaCybase Una aplicación completa de base de datos y archivo de imágenes que se puede personalizar según las necesidades de cada laboratorio.

- CAIS–ANINTELLIGENTSYSTEMFORCHROMOSOMEANALYSIS AND CLASSIFICATION desarrollada por EkaterinaDetcheva - Institute of Mathematics and Informatics, Bulgaria en 2000, hace uso de las redes neuronales, para la identificación de patrones y aplica el mejoramiento de imágenes para presentar los informes.
- PAPNET: desarrollada por NEUROMEDICALSYSTEMSInc. en 1997 que utiliza procesamiento de imágenes y redes neuronales para la discriminación de imágenes.
- AUTOPAP: desarrollado por Neopath Inc. en 2001, hace uso de rutinas de clasificación y procesamiento de imágenes, hace uso de árboles de decisión, lógica difusa y redes neuronales.
- AutoCytePREPSsystem: desarrollado por Roche ImageAnálisisSystems, Inc. en 1999, hace uso de un cepillo especial para la recolección de muestras de frotis y la separación de células implica centrifugación y tratamientos químicos.
- A nivel internacional dentro en BIOMEDICALACM/IEEE, se encontró la publicación de los siguientes trabajos:
 - Nicolás Pérez de la Blanca, "Clasificación Automática de Cromosomas" Universidad de Granada, en 2004.
Proyecto de desarrollo implementado con redes neuronales, como también el tratamiento de imágenes solo para propósitos de investigación académica, este sistema no fue comercializado.
 - Nicolás Pérez de la Blanca " Desarrollo de una plataforma software para la detección automática de cromosomas a partir de imágenes digitales" Universidad de Granada, en 2003
Este proyecto también implementa redes neuronales y la red neuronal usada es de Hopfield para la identificación de patrones.

- Rafael Molina, "Imagen médica molecular y multimodalidad (IM3). Análisis y Tratamiento de Imagen médica"
Universidad de Granada, 2005
Implementado en la universidad de Granada, con propósitos académicos y médicos.

1.3. PROBLEMA

La situación actual del Instituto de Genética de la Universidad Mayor de San Andrés presenta la siguiente lista de problemas:

- Debido a la limitación visual humana para distinguir variaciones mínimas de niveles de gris los resultados obtenidos son en su mayoría subjetivas.
- Personal reducido de expertos en el área de citogenética, en Bolivia quienes de acuerdo a su capacidad y experiencia, demoran un tiempo superior a los 15 días para la obtención de resultados.
- Costo de estudio demasiado elevado para el paciente, este puede sobrepasar de los 300 Bs.
- Obtención de resultados equivocados, los cuales son obviados o no fueron apreciados por el especialista y fueron catalogados de manera equivocada.
- Dificultad de identificación individual de los 46 cromosomas existentes en un cariotipo humano.
- No existe documentación de la información de un paciente acerca de sus cromosomas y todos sus antecedentes.
- Tiempo de espera del paciente para la obtención de resultados de los exámenes, los pacientes esperan aproximadamente 15 días.
- Poca confianza en los resultados lo que es provocado por la obtención de resultados equivocados, debido a que el paciente no puede ver un informe con imágenes de sus cromosomas.
- Cansancio de la vista humana debido al esfuerzo realizado en el proceso de selección.
- Almacenamiento ineficiente de estas imágenes en una Base de Datos que permita realizar el seguimiento adecuado del paciente.

¿Usando el Modelo de Red Neuronal de Retropropagacion y aplicando métodos de tratamiento digital de imágenes será posible identificar los Cromosomas Humanos?

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Diseñar e implementar el modelo de red neuronal que analice, reconozca y clasifique las características cromáticas de los cromosomas humanos a partir de imágenes digitales.

1.4.2. OBJETIVOS ESPECÍFICOS

Los objetivos específicos que se esperan luego de implementas las herramientas son los siguientes:

- Mostrar la aplicabilidad de las redes neuronales al reconocimiento de cromosomas humanos.
- Mostrar los cromosomas clasificados.
- Mejorar la calidad de presentación de la información sometiendo las imágenes fotográficas a un tratamiento adecuado.
- Agilizar el estudio e identificación de los cromosomas reduciendo el tiempo que el especialista tarda en realizar la selección esto es aproximadamente 5 días.
- Plantear el modelo matemático para la optimización de la imagen.
- Brindar resultados más confiables a los pacientes.
- Desarrollar una herramienta que ayude al médico especialista, laboratoristas, bioquímicas, y estudiantes de citogenética a clasificar los cromosomas, de acuerdo a sus características cromáticas.

1.5. HIPÓTESIS

El modelo de red neuronal propuesto permite clasificar e identificar con mayor precisión los cromosomas humanos utilizando fotografías digitales, en comparación con el método tradicional de visión y selección humana

La red neuronal aprenderá de un conjunto de patrones e identificara los patrones de la imagen digital para determinar a qué conjunto o tipo cromosoma pertenece. Finalmente se individualizara cada uno de los 24 tipos de cromosomas.

1.6. JUSTIFICACIONES

1.6.1.CIENTÍFICA

El estudio de las redes neuronales artificiales y la necesidad de clasificar o reconocer las imágenes, resulta muy complejo ya que requiere de la experiencia y habilidad de un profesional médico con especialidad en citología, es así que la aplicación de estas técnicas no convencionales como son las redes neuronales de retropropagacion, sugiere una nueva forma de clasificar las imágenes digitales, abriendo nuevos caminos en el estudio de este campo, proporcionando herramientas para el desarrollo de futuros trabajos de investigación.

1.6.2.ECONÓMICA

Los beneficios que se obtendrán con el desarrollo de un modelo de reconocimiento de imágenes serán mayores a los costos y gastos del sistema tradicional de clasificación de las imágenes de cromosomas. El sistema reducirá los costos y tiempo de obtención de resultados en un examen de citogenética.

1.6.3. SOCIAL

El trascendido social que tendrá este trabajo de investigación básicamente será reflejado en el beneficio directo a hospitales o institutos de genética, con la obtención rápida y confiable de los resultados, en base a las imágenes que se utilizan. Desarrollar una red neuronal para clasificar cromosomas humanos mediante el tratamiento digital de imágenes, es un aporte importante a la sociedad, que requiere de resultados confiables y oportunas, para tomar decisiones que involucren nuestra salud y economía.

1.7. ALCANCES Y LIMITES DEL TRABAJO

Los Alcances del presente trabajo son:

- Construir un prototipo que permita operar la red neuronal.
- Diseñar un Modelo de automatización para el reconocimiento de patrones de los cromosomas en Bandeamiento G.
- Diseñar la red neuronal para el reconocimiento y clasificación de imágenes.
- Diseño y elaboración de herramientas de tratamiento de imágenes fotográficas digitales

Los límites del presente trabajo son:

- Para la obtención de las imágenes digitales se cuenta con una cámara que no está incorporada a un microscopio por tanto las herramientas trabajaran por separado del microscopio.
- No se contemplara el almacenamiento en gestores de bases de datos de la información fotográfica digital.
- Permite la identificación usando un solo método que es el bandemiento G.

1.8. MÉTODOS Y TÉCNICAS

En el presente trabajo se aplicara el modelo de la red neuronal de retro propagación o propagación hacia atrás como metodología, es conveniente revisar los modelos previos de redes neuronales, ya que permite conocer las limitaciones de dichos modelos. Se hace énfasis en el modelo de retro propagación debido a que es utilizado como clasificador para el reconocimiento de imágenes.

Un tipo de clasificación para las redes neuronales, son las denominadas redes de propagación hacia adelante, que se caracterizan por tener una arquitectura similar, donde sus conexiones son solamente hacia adelante entre las neuronas pertenecientes a diferentes capas, pudiendo ser mono capa o multicapas y con un tipo de aprendizaje supervisado [Caceres, 2002]. En este tipo de redes se pueden incluir los siguientes modelos:

- Perceptron
- Perceptron Multicapa
- Adaline
- Madaline
- Retropropagación

Para el estudio de la red neuronal de retropropagación, basta con comprender previamente a los modelos de Perceptron y Perceptron Multicapa. El estudio teórico de la red Adaline y Madaline pueden verse en detalle en el Proyecto de Licenciatura [Caceres, 2002].

Las etapas ya identificadas del reconocimiento de imágenes se muestran en la Figura 1.2. Se usarán técnicas de procesamiento de imágenes y técnicas de reconocimiento de patrones. Las etapas en que trabajaremos son:

Localización de los cromosomas. Se usará segmentación basada en color y otras técnicas que tengan en cuenta las características espaciales de los píxeles del código.

Segmentación de los cromosomas, para obtener cada uno por separado y analizarlo de forma individual parte fundamental y crítica.

Reconocimiento de cromosomas. Clasificación de patrones, mediante redes neuronales, para averiguar a qué grupo pertenece cada cromosoma. El desarrollo del trabajo se llevara en el orden ya mencionados: primero se atacará el problema de la localización, después el de la segmentación de los cromosomas, y por último el de la clasificación de patrones. Esto debe ser así porque cada etapa trabaja con los datos suministrados por la anterior, por lo que su diseño depende fundamentalmente de la forma en la que se realizaron las etapas previas. Este esquema de trabajo es flexible, por lo que en una fase del proyecto podrán modificarse aspectos de fases anteriores.

En cada etapa se examinarán las técnicas existentes para resolver el problema dado, y se escogerá aquella que mejor se ajuste a nuestras necesidades.

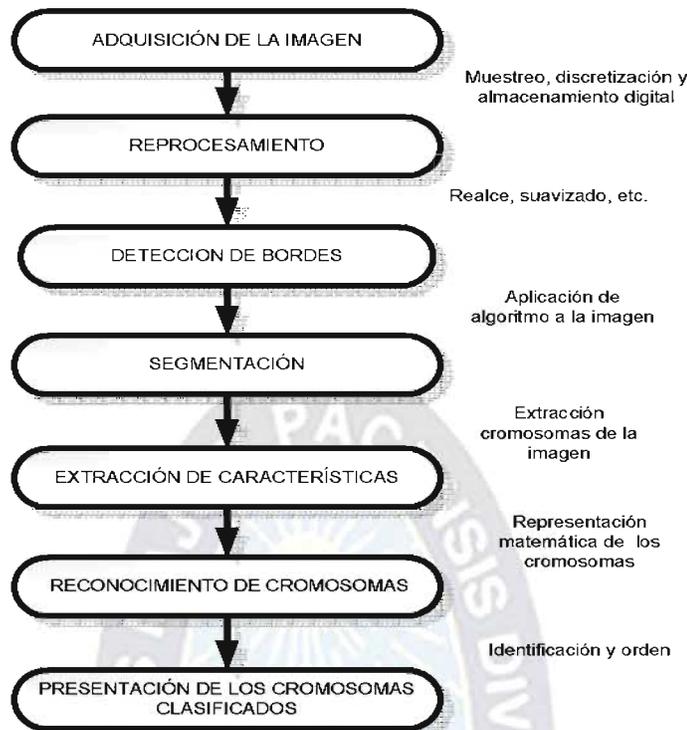


Figura 1.2. Etapas del reconocimiento de imágenes.
Fuente: Elaboración propia

Para la segmentación de cromosomas se aplicara la metodología de segmentación por Umbral que por tratarse de imágenes monocromáticas son la mejor opción, se separó las distintas metodologías tanto de segmentación como de clasificación, esto a que las metodologías existentes que realizan ambas tareas al mismo tiempo resultan insuficiente para la obtención de muestras.

Revisando los algoritmos para la obtención de muestras se aplicara la técnica de Prewit que obtendrá muestras o características más relevantes en una imagen, puesto que se adapta de forma natural a la segmentación de la imagen. Además es evidente la necesidad de estandarizar propiedades en la imagen como el brillo, intensidad, luminosidad y también el tamaño de imagen.



CAPITULO II

RED NEURONAL DE RETROPROPAGACION

RESUMEN

Este capítulo presenta a las redes neuronales artificiales como herramienta fundamental para el desarrollo del presente tema de investigación. Se muestra la analogía de las redes neuronales y las neuronas biológica, finalmente se muestra la Red Neuronal de retropropagacion.

2.1.INTRODUCCIÓN

Existen tareas para las cuales todavía no existen algoritmos, o para las cuales es virtualmente imposible escribir una serie de pasos lógicos o aritméticos que proporcionen la solución adecuada a una tarea determinada. Estas tareas tienen características importantes en común:

- Los humanos pueden saber cómo realizarlas.
- Se puede generar grandes cantidades de ejemplos de ellas.
- Cada tarea requiere realizar una asociación entre objetos de dos conjuntos.

Los ejemplos de este tipo de tareas cognitivas como reconocer un rostro familiar, hablar, comprender el lenguaje y recuperar contextualmente información apropiada desde la memoria, estas tareas están más allá del alcance de las computadoras programadas de manera convencional, así como de los sistemas expertos tradicionales.

Aplicaciones exitosas de Redes Neuronales Artificiales en ingeniería han demostrado que pueden ser utilizadas como un método alternativo de resolución de problemas por su eficiencia y exactitud. Sus características inherentes, tales como su capacidad de aprender, generalizar y procesar en paralelo, las han transformado en la herramienta ideal para muchas aplicaciones, aunque quizás su uso más difundido sea para el reconocimiento de patrones.

2.2. NEURONA BIOLÓGICA

La habilidad del cerebro para almacenar información sobre todas las neuronas y el proceso de recordar la información almacenada es un proceso colectivo entre todas las neuronas que conforman el sistema nervioso.[CHOQUE - 2005]

Una neurona consta de tres partes:

- **Dendritas:** un árbol de fibras de entrada que lleva los potenciales de acción de las neuronas transmisoras en la neurona.
- **Soma:** El cuerpo principal de la célula con un núcleo. Es en este lugar donde los potenciales de acción son construidos antes que la neurona se active.
- **Axón:** Es una fibra de salida simple que bifurca a otras neuronas y transmite los potenciales de acción generados por la neurona.[CHOQUE - 2005]

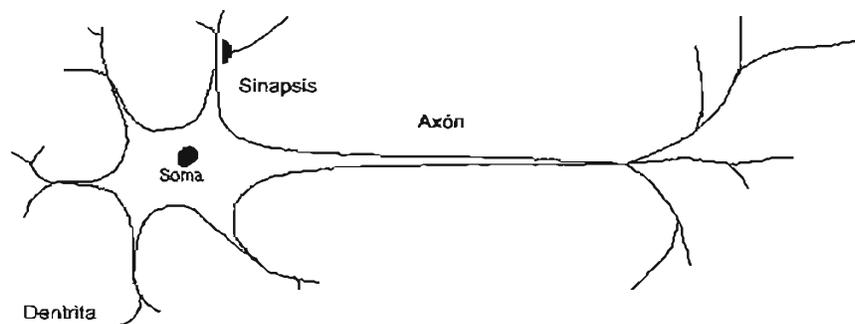


Figura Nº 2.1 - Neurona Biológica
Fuente:[CHOQUE - 2005]

Las dendritas y los axones pueden ser vistos como cadenas de comunicación entre las neuronas, y no se conocen los detalles exactos acerca de cómo funcionan actualmente. Las partes más interesantes son el soma y la sinapsis los cuales constituyen los puntos de conexión entre los axones y las dendritas.[CHOQUE - 2005]

Las neuronas reciben señales (entradas) de otras neuronas vía conexiones sinápticas que pueden ser activadoras o inhibitoras. En función de las señales recibidas, una neurona envía a sus vez una señal a otras neuronas por medio del axón.[CHOQUE - 2005]

Una neurona contiene un potencial interno continuo denominado potencial de membrana. Cuando este accede a un cierto valor umbral, la neurona puede transmitir todo

su potencial por medio del axón. Se estima que el cerebro humano contiene más de cien mil millones (10^{11}) de neuronas y que hay más de 1000 sinapsis a la entrada y a la salida de cada neurona.[CHOQUE - 2005]

2.3. REDES NEURONALES ARTIFICIALES

Una red neuronal artificial es un procesador distribuido en paralelo de forma masiva que tiene una tendencia natural para almacenar conocimiento de forma experimental y lo hace disponible para su uso.

Es una nueva forma de cómputo, que es capaz de manejar las imprecisiones e incertidumbres que aparecen cuando se trata de resolver problemas relacionados con el mundo real, ofreciendo soluciones precisas y de fácil implementación [CACERES- 2002]. Su principal objetivo de una red neuronal es emular el cerebro biológico con máquinas, explotan el procesamiento local en paralelo y la representación distribuida del cerebro [CÁCERES- 2002].

Las cualidades principales que hereda de las características propias de las neuronas biológicas son: [CÁCERES- 2002]

- Aprendizaje a partir de la experiencia.
- Generalización.
- Abstracción.

Las Fases de Operación de las Redes Neuronales Artificiales son:

Entrenamiento o Aprendizaje de la red, El usuario proporciona a la red un número preciso de datos de entrada y salida; la red entonces ajusta su pesos de interconexión o sinapsis hasta que la salida de la red se aproxime a la salida correcta.[PRATT, 2001]

- **Recuperación de lo Aprendido**, A la red se le presenta un conjunto de datos de entrada y salida similares, y esta simplemente re calcula la salida correcta. Por lo tanto el conocimiento o capacidad de dar una respuesta de una Red Neuronal Artificial, se encuentra en la función de activación utilizada [PRATT - 2001].

2.4. ARQUITECTURA DE UNA RED NEURONAL

Una Red Neuronal Artificial puede considerarse como un grafo dirigido ponderado en el que neuronas artificiales son nodos y las hojas dirigidas y ponderadas son conexiones entre las salidas y entradas de neuronas.

Basado en el tipo de conexiones existen dos principales tipos:

- FeedForward. En estas las señales viajan de entrada a salida y las capas internas son actualizadas antes que las de salida.
- Recurrentes. En estas existen conexiones entre capas y sus capas anteriores, y en algunos casos entre perceptores y ellos mismos.
- Redes de propagación hacia delante (feedforward): en las que los grafos no tienen bucles.
- Recurrentes o de Retroalimentación (feedback), en las cuales los bucles ocurren debido a conexiones de retroalimentación

Basado en el tipo de aprendizaje de las RNA, existen dos categorías:

De aprendizaje supervisado. En este tipo de aprendizaje, los pesos de la red son modificados con base al error encontrado entre el resultado de la red y el resultado esperado.

De aprendizaje no supervisado. Aquí la red actualiza sus pesos con base al patrón de entrada únicamente.

En el aprendizaje supervisado, la red actualiza los pesos con el fin de disminuir la diferencia entre la salida actual y su salida esperada, uno de los algoritmos más usados para este tipo de aprendizaje es el de retropropagación.

2.5. NEURONA ARTIFICIAL

En 1943, el neurofisiólogo Warren McCulloch y el matemático Walter Pitts en su artículo "A Logical Calculus of the Ideas Immanent in Nervous Activity" (McCulloch), proponen el primer modelo computacional de un sistema neuronal basado en circuitos eléctricos. El modelo estaba conformado por simples unidades binarias que podrían recibir dos tipos de estímulos de otras unidades o del exterior. Con este modelo, se inició la idea de una nueva forma de evaluar funciones computables. [CHOQUE - 2005]

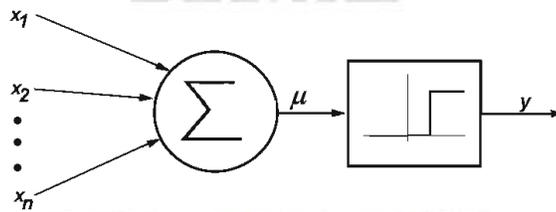


Figura Nº 2.2 – Modelo Computacional de una neurona
Fuente: [CHOQUE - 2005]

Esta neurona computa una suma ponderada de entrada $x_j, j = 1, 2, 3, \dots, n$, y genera un resultado de "1" si esta suma supera un cierto umbral μ y "0" en otro caso.

$$y = \theta \left(\sum_{j=1}^n w_j x_j - \mu \right)$$

Donde $\theta()$ es una función de paso de unidad en 0, y w_j es el peso de la sinapsis asociado con la j-esima entrada. Por simplicidad podemos considerar el umbral μ como otro peso $w_0 = -\mu$ asociado a la neurona con una entrada constante $x_0 = 1$. Pesos positivos corresponden a sinapsis excitadoras, mientras que los negativos a inhibitoras.

La Neurona de McCulloch y Pitts se ha generalizado de muchas maneras. En general, usando distintas funciones de activación. [CHOQUE - 2005]

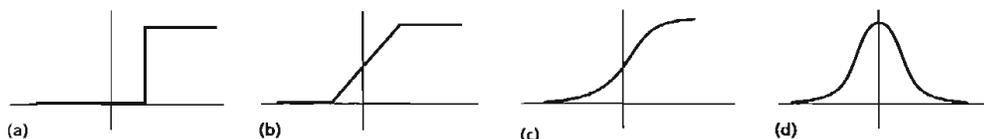


Figura Nº 2.3 – Funciones de activación
Fuente: [CHOQUE - 2005]

2.6. BACKPROPAGATION

Rumelhart, Hinton y Williams en 1986 (Rumelhart et al., 1986), basándose en los trabajos de investigadores como Werbos y Parker (Werbos, 1974; Parker, 1982), desarrollaron un tipo de red neuronal basada en la arquitectura de un Perceptron Multicapa pero con una nueva regla de aprendizaje denominada retropropagación o propagación del error hacia atrás, basado en una generalización de la “regla delta” (Cáceres, 2002; Freeman et al., 1993; Hiler et al., 1995) ampliando el rango de aplicaciones que las redes neuronales de este tipo poseían hasta entonces. [CHOQUE - 2005]

2.6.1. ARQUITECTURA

En general, en una red de retropropagación, existe una capa de entrada con “n” neuronas, la menos una capa oculta con “l” neuronas. No posee conexiones hacia atrás entre neuronas de diferentes capas, ni laterales en neuronas de la misma capa. Las neuronas i-ésimas de entrada se denominan I_i , las neuronas j-ésimas ocultas H_j y las neuronas k-ésimas de salida O_k . Las componentes de entrada a las neuronas I_i se denominan x_i y las componentes de salida a las neuronas O_k se denominan y_k . Los pesos asociados entre las neuronas I_i y H_j se denominan w_{ij}^{HI} y los asociados entre las neuronas H_j y las O_k se denominan w_{kj}^{OH} (Ver figura 3).

Una característica importante del algoritmo de retropropagación, es la representación interna del conocimiento que puede realizar gracias a las capas ocultas que lo permiten consiguiendo cualquier correspondencia entre los patrones de entrada y la salida. (Ver Figura 2.3) [CACERES-2002]

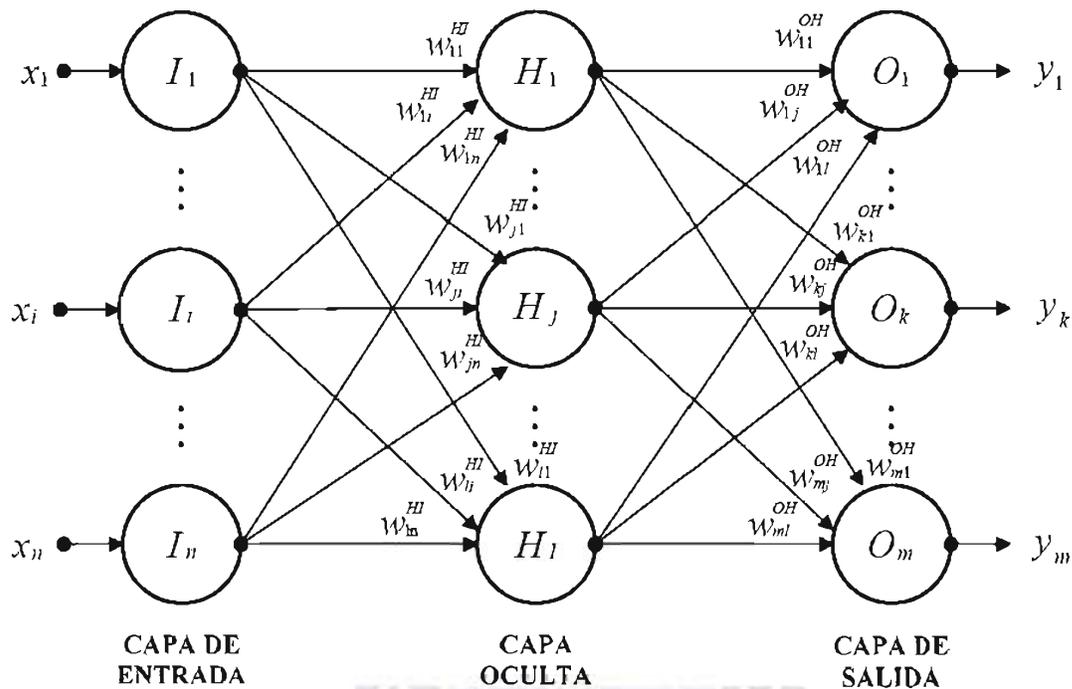


Figura N° 2.4 – Arquitectura de la Red Neuronal
FUENTE: [CÁCERES, 2002]

2.6.2. APRENDIZAJE (REGLA DELTA GENERALIZADA)

La regla delta propuesta por Widrow en 1960 (Freeman et al., 1993; Hiler et al., 1995; Widrow et al., 1960), fue generalizada a redes con conexiones hacia adelante, con múltiples capas, cuyas neuronas poseen funciones de transferencia continuas, no decrecientes y derivables, dando forma a lo que se conoce como “regla delta generalizada” o algoritmo de retropropagación del error. Esta red posee un mecanismo de aprendizaje offline, de tipo supervisado y por corrección de error. Relaciona los patrones de entrada con la de salida, en forma heteroasociativa y es considerada como una red análoga, ya que puede operar con patrones con valores continuos tanto en la entrada, como en la salida.

El método de aprendizaje está basado en dos fases de propagación:

Hacia adelante. En la fase de propagación hacia adelante los patrones de entrada son propagados por la red a través de sus múltiples capas hasta generar la salida. Obtenido el vector de salida, este es comparado con el deseado, mediante los errores cuadráticos cometidos.

Hacia atrás. En la fase de propagación del error hacia atrás, debido a que se desconoce la salida deseada para las neuronas pertenecientes a capas ocultas, el error es propagado desde la capa de salida hacia las capas ocultas modificando los pesos asociados a dichas conexiones hasta llegar a la capa de entrada. Así, se habrán modificado todos los pesos de la red, ajustándose al patrón que recientemente le fue presentado. Si se realiza nuevamente una propagación hacia adelante con el mismo patrón, la red habrá disminuido su error en la salida, en comparación con el error cometido en la propagación anterior.

Así sucesivamente, a la red se le podrán ir presentando nuevos patrones, para que vaya auto ajustando sus pesos con el fin de aprenderlos y de cometer un error que sea mínimo.

Los pesos asociados a las conexiones entre neuronas de diferentes capas, pueden representarse vectorialmente mediante matrices. En una red que posee (n)-capas, se requerirán (n-1)-matrices para representar sus pesos. En el caso particular que se analizará, por ser el más aplicado y utilizado, se requerirán tres capas: de entrada (I), oculta (H) y de salida (O). Por lo que se necesitarán dos matrices para representar todos los pesos de la red. La primera matriz, representará los pesos asociados a las conexiones entre la capa de entrada y la capa oculta, en un instante t (Ver Ecuación 2.11) y la segunda matriz, representa los pesos de las conexiones entre la capa oculta y la de salida, en un instante t (Ver Ecuación 2.12) [CACERES-2002].

$$W^{HI}(t) = (w_{ji}^{HI}(t)) = \begin{pmatrix} w_{11}^{HI}(t) \cdots w_{1l}^{HI}(t) \cdots w_{1n}^{HI}(t) \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ w_{j1}^{HI}(t) \cdots w_{ji}^{HI}(t) \cdots w_{jn}^{HI}(t) \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ w_{l1}^{HI}(t) \cdots w_{li}^{HI}(t) \cdots w_{ln}^{HI}(t) \end{pmatrix} \quad (2.11)$$

$$W^{OH}(t) = (w_{jn}^{OH}(t)) = \begin{pmatrix} w_{11}^{OH}(t) \cdots w_{1j}^{OH}(t) \cdots w_{1l}^{OH}(t) \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ w_{k1}^{OH}(t) \cdots w_{kj}^{OH}(t) \cdots w_{kl}^{OH}(t) \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ w_{m1}^{OH}(t) \cdots w_{mj}^{OH}(t) \cdots w_{ml}^{OH}(t) \end{pmatrix} \quad (2.12)$$

Donde:

$w_{yx}^{QR}(t)$: Peso asociado a la conexión entre una neurona N_y de la capa Q con otra neurona N_x de la capa R en un instante t y donde R es una capa inferior a Q .

Matemáticamente, el algoritmo de aprendizaje para una red neuronal de Retropropagación puede expresarse de la siguiente manera (Freeman et al., 1993; Hilera et al., 1995):

Iniciar todos los pesos de la red en el instante ($t=0$) con valores pequeños aleatorios comprendidos en el intervalo continuo $[-0.5; 0.5]$.

Presentar un nuevo p-ésimo patrón vectorial de entrada en el instante t (Ver Ecuación 2.13).

$$X_p(t) = \begin{pmatrix} x_{p1}(t) \\ \vdots \\ x_{pn}(t) \end{pmatrix} \quad (2.13)$$

Especificar la salida vectorial deseada para el nuevo p-ésimo patrón en el instante t (Ver Ecuación 2.14)

$$D_p^o(t) = \begin{pmatrix} d_{p1}^o(t) \\ \vdots \\ d_{pm}^o(t) \end{pmatrix} \quad (2.14)$$

Obtener la salida vectorial de la red para el p-ésimo patrón en el instante t (Ver ecuación 2.15).

$$Y_p^O(t) = \begin{pmatrix} y_{p1}^O(t) \\ \vdots \\ y_{pm}^O(t) \end{pmatrix} \quad (2.15)$$

Para obtener la salida $Y_p^O(t)$ es necesario conocer las salidas de cada una de las neuronas pertenecientes a cada capa de la red, lo que se logra realizando la propagación hacia adelante.

Para ello, se deberá hacer lo siguiente:

Calcular la función de activación para cada neurona N_j perteneciente a la capa oculta (H), con la información procedente de las neuronas N_i pertenecientes a la capa de entrada (I) y con los pesos asociados entre dichas capas. Además se debe utilizar el umbral de activación para todas las neuronas pertenecientes a todas las capas de la red, salvo la entrada. Recordar que el umbral de activación puede ser modelado como un peso adicional de conexión entre una neurona conocida con una neurona ficticia adicional de conexión entre una neurona conocida con una neurona ficticia adicional cuya salida siempre es unitaria.

El umbral de activación vectorial de la capa oculta para un p-ésimo patrón de entrenamiento en el instante t, se representa en la ecuación 2.16.

$$\theta_p^H(t) = \begin{pmatrix} \theta_{p1}^H(t) \\ \vdots \\ \theta_{pl}^H(t) \end{pmatrix} \quad (2.16)$$

La función de activación de la capa oculta, para un p-ésimo patrón de entrenamiento en el instante t, se puede calcular vectorialmente (Ver Ecuación 2.17) y en forma escalar (Ver Ecuación 2.17) y en forma escalar (Ver Ecuación 2.18).

$$NET_p^h(T) = W^{HI}(t)X_p(t) - \theta_p^H(t) = \begin{pmatrix} net_{p1}^H(t) \\ \vdots \\ net_{pi}^H(t) \end{pmatrix} \quad (2.17)$$

$$net_{pj}^H(t) = \sum_{i=1}^n w_{ji}^{HI}(t)x_{pi}(t) - \theta_{pj}^H(t) \quad (2.18)$$

Aplicando la idea de la neurona ficticia, la expresión escalar de la ecuación 1.18 puede ser reducida a la Ecuación 2.19.

$$net_{pj}^H(t) = \sum_{i=0}^n w_{ji}^{HI}(t)x_{pi}(t) \quad (2.19)$$

Donde:

$$w_{j0}^{HI} = -\theta_{pj}^H$$

$$x_{p0} = 1$$

Proceder a calcular vectorialmente las salidas de cada una de las neuronas de la capa oculta para el p-ésimopatrón en el instante t. Lo que se puede representar vectorialmente (Ver Ecuación 2.20) y en forma escalar (Ver Ecuación 2.21).

$$Y_p^H(t) = f_-^H(NEt_p^H(t)) \quad (2.20)$$

$$y_{pj}^H(t) = f_j^H(net_{pj}^H(t)); \forall j \in [1, L] \subset N \quad (2.21)$$

Donde:

f_j^H : Función de transferencia escalar de la neurona N_j perteneciente a la capa oculta.

f_-^H : Función de transferencia vectorial de todas las neuronas pertenecientes a la capa oculta.

Calcular la Función de activación para cada neurona N_k perteneciente a la capa de salida (O), con información procedente de las neuronas N_j pertenecientes a la capa oculta (H) y con los pesos asociados entre dichas capas.

El umbral de activación vectorial de la capa de salida para un p-ésimopatrón de entrenamiento en el instante t, se representa por la Ecuación 2.22.

$$\theta_p^O(t) = \begin{pmatrix} \theta_{p1}^O(t) \\ \vdots \\ \theta_{pm}^O(t) \end{pmatrix} \quad (2.22)$$

La función de activación de la capa de salida, para un p-ésimopatrón de entrenamiento en el instante t, se puede calcular vectorialmente (Ver ecuación 2.23) y en forma escalar (Ver Ecuación 2.24).

$$NET_p^O(t) = W^{OH}(t)Y_p^H(t) - \theta_p^O(t) = \begin{pmatrix} net_{p1}^O(t) \\ \vdots \\ net_{pm}^O(t) \end{pmatrix} \quad (2.23)$$

$$net_{pk}^O(t) = \sum_{j=1}^l w_{kj}^{OH}(t) y_{pj}^H(t) - \theta_{pk}^O(t) \quad (2.24)$$

Aplicando la idea de la neurona ficticia, la expresión escalar de la ecuación (2.24) puede ser reducida a la Ecuación 2.25.

$$net_{pk}^O(t) = \sum_{j=0}^l w_{kj}^{OH}(t) y_{pj}^H(t) \quad (2.25)$$

Donde:

$$w_{k0}^{OH} = -\theta_{pk}^O$$

$$y_{p0}^H = 1$$

Se procede a calcular vectorialmente las salidas de cada una de las neuronas de la capa de salida para el p-ésimopatrón en el instante t. Lo que se puede escribir vectorialmente (Ver Ecuación 2.26) y en forma escalar (Ver Ecuación 2.27).

$$Y_p^O(t) = f_{-}^O(NET_p^O(t)) \quad (2.26)$$

$$y_{pk}^O(t) = f_k^O(net_{pk}^O(t)); \quad \forall k \in [1, m] \subset N \quad (2.27)$$

Donde:

f_k^O : Función de transferencia de la neurona N_k perteneciente a la capa de salida.

f_{-}^O : Función de Transferencia vectorial de todas las neuronas pertenecientes a la salida.

Procedimiento para calcular los términos del error (δ) para todas las neuronas pertenecientes a todas las capas de la red, salvo la entrada, propagando el error hacia atrás.

Calculo del error en neuronas de salida (Ver Ecuación 2.28)

$$\delta_{pk}^O(t) = [d_{pk}^O(t) - y_{pk}^O(t)] \frac{\partial}{\partial net_{pk}^O(t)} [f_k^O(net_{pk}^O(t))] \quad \forall k \in [1, m] \subset N \quad (2.28)$$

Calculo escalar del error en neuronas ocultas (Ver Ecuación 2.29).

$$\delta_{pj}^H(t) = \left(\sum_{k=1}^m \delta_{pk}^O(t) w_{kj}^{OH}(t) \right) \frac{\partial}{\partial net_{pj}^H(t)} [f_j^H(net_{pj}^H(t))] \quad \forall j \in [1, l] \subset N \quad (2.29)$$

Actualización de los pesos de la red, corrigiendo hacia atrás desde la salida a.C. la entrada.

Corregir los pesos asociados a las conexiones entre la capa de salida y la oculta (Ver Ecuación 2.30).

$$w_{kj}^{OH}(t+1) = w_{kj}^{OH}(t) + \Delta w_{kj}^{OH}(t+1)$$

$$\Delta w_{kj}^{OH}(t+1) = \eta \delta_{pk}^O(t) y_{pj}^H(t) + \beta [w_{kj}^{OH}(t) - w_{kj}^{OH}(t-1)]$$

Donde:

η : Tasa de aprendizaje (LearningRate) ($0 < \eta \leq 1$).

β : Momento (Momentum) ($0 < \beta \leq 1$).

Por lo tanto:

$$w_{kj}^{OH}(t+1) = w_{kj}^{OH}(t) + \eta \delta_{pk}^O(t) y_{pj}^H(t) + \beta [w_{kj}^{OH}(t) - w_{kj}^{OH}(t-1)] \quad (2.30)$$

Corregir los pesos asociados a las conexiones entre la capa oculta y de la entrada (Ver Ecuación 2.31).

$$w_{ji}^{HI}(t+1) = w_{ji}^{HI}(t) + \Delta w_{ji}^{HI}(t+1)$$

Por lo tanto:

$$w_{jl}^{HI}(t+1) = w_{jl}^{HI}(t) + \eta \delta_{pj}^H(t) x_{pl}(t) + \beta [w_{jl}^{HI}(t) - w_{jl}^{HI}(t-1)] \quad (2.31)$$

El proceso de aprendizaje se repite para todos los patrones hasta que el término de error total (Ver Ecuación 2.32) sea considerado aceptablemente pequeño.

Sea:

$$\varepsilon_{pk}^O(t) = d_{pk}^O(t) - y_{pk}^O(t)$$

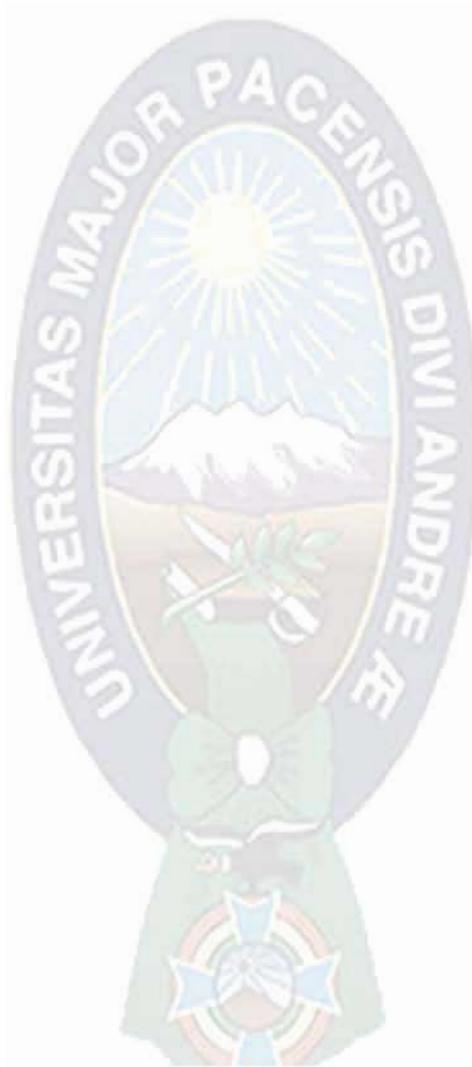
Entonces:

$$e_p(t) = \frac{1}{2} \sum_{k=1}^m \varepsilon_{pk}^O(t) \quad y \quad e_T(t) = \frac{1}{p} \sum_{s=1}^p e_s(t)$$

Por lo tanto

$$e_T(t) = \frac{1}{2p} \sum_{s=1}^p \sum_{k=1}^m [d_{sk}^O(t) - y_{sk}^O(t)]^2 \quad (2.32)$$





CAPITULO III

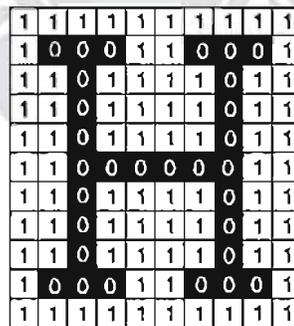
TRATAMIENTO DIGITAL DE IMÁGENES

RESUMEN

En este capítulo se describen las teorías básicas y las más fundamentales para realizar el procesamiento digital de imágenes, como ser el pre-procesamiento, la segmentación, la extracción de características y filtrado de las muestras las cuales están orientadas aplicarse en imágenes monocromáticas.

3.1. IMÁGENES DIGITALES

Son fotos electrónicas tomadas de una escena o escaneadas de documentos fotografías, manuscritos, textos impresos e ilustraciones. Se realiza una muestra de la imagen digital y se confecciona un mapa de ella en forma de cuadrícula de puntos o elementos de la figura (píxeles). A cada píxel se le asigna un valor tonal (negro, blanco, matices de gris o color), el cual está representado en un código binario (ceros y unos). Los dígitos binarios ("bits") para cada píxel son almacenados por una computadora en una secuencia, y con frecuencia se los reduce a una representación matemática (comprimida). Luego la computadora interpreta y lee los bits para producir una versión analógica para su visualización o impresión [KENNEY, RIEGER, 2003].



1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1		
1	1	0	1	1	1	1	0	1	1		
1	1	0	1	1	1	1	0	1	1		
1	1	0	1	1	1	1	0	1	1		
1	1	0	0	0	0	0	0	1	1		
1	1	0	1	1	1	1	0	1	1		
1	1	0	1	1	1	1	0	1	1		
1	1	0	1	1	1	1	0	1	1		
1	0	0	0	1	1	0	0	0	1		
1	1	1	1	1	1	1	1	1	1		

Figura 3.1 - Valores de píxel: Como se exhibe en esta imagen bitonal, a cada píxel se le asigna un valor tonal, en este ejemplo 0 para el negro y 1 para el blanco.

Fuente: [KENNEY, RIEGER, 2003].

3.2. RESOLUCIÓN

Es la capacidad de distinguir los detalles espaciales finos. Por lo general, la frecuencia espacial a la cual se realiza la muestra de una imagen digital (la frecuencia de

muestreo) es un buen indicador de la resolución. Este es el motivo por el cual dots-per-inch (puntos por pulgada) (dpi) o pixels-per-inch (píxeles por pulgada) (ppi) son términos comunes y sinónimos utilizados para expresar la resolución de imágenes digitales. Generalmente, pero dentro de ciertos límites, el aumento de la frecuencia de muestreo también ayuda a aumentar la resolución. [KENNEY, RIEGER, 2003]

3.3. SEGMENTACIÓN POR UMBRAL

La segmentación de una imagen consiste en la división o separación de una imagen en regiones de atributos similares. Es decir, la segmentación subdivide una imagen en sus partes constituyentes u objetos. El grado de subdivisiones depende del problema a resolver, por lo que la segmentación deberá detenerse cuando los objetos de interés de nuestro estudio hayan sido aislados.

Los algoritmos para segmentar imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores del nivel de gris: discontinuidad y similitud. En la primera categoría, la aproximación es dividir la imagen basándose en cambios abruptos en el nivel de gris. Las principales áreas de interés en esta categoría son de detección de puntos aislados, de líneas y de bordes en una imagen. La segunda categoría se basa en segmentar por niveles de umbral, en regiones de crecimiento y en corte y pegado de regiones.

En este documento nos centraremos en la segmentación por umbral, también denominada umbralización.

Las aplicaciones de estas técnicas son muchas y en diversos campos:

Medicina: detección de células en citología; localización de fracturas, análisis de ecografías, CAT's e imágenes de rayos X en traumatología.

Industria: supervisión automática de procesos industriales, como control de calidad, vigilancia o detección de fallos.

Análisis de fotografías aéreas o por satélite: clasificación de diferentes tipos de vegetación para el estudio de su degradación; clasificación y control de cultivos en agricultura; detección de objetos.

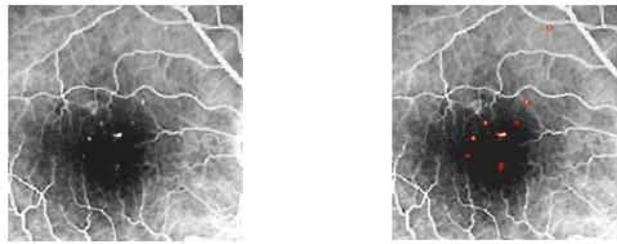


Figura N° 3.2. Este ejemplo ilustra la aplicación del análisis de formas a la localización de aneurismas en una angiografía ocular
Fuente:[Gonzales, Woods, 1996]

3.4. FUNDAMENTOS

Muchas imágenes se pueden caracterizar como un conjunto de objetos luminosos sobre un fondo de determinada oscuridad, de tal forma que los píxeles del objeto y el fondo tienen los niveles de gris agrupados en dos modos dominantes (ejemplos típicos pueden ser texto manuscritos y escritos a máquina, muestras biomédicas microscópicas, rutas de aeroplanos...) Un modo de extraer los objetos del fondo es elegir un determinado umbral T que aisle dichos modos.

Entonces cualquier punto (x,y) para el cual $f(x,y) > T$ es llamado un punto objeto; de otra manera, el punto es llamado un punto del fondo.

La misma aproximación básica clasifica un punto (x,y) como perteneciente a un tipo de objeto si $T_1 < f(x,y) \leq T_2$, a la otra clase de objeto si $f(x,y) > T_2$, y al fondo si $f(x,y) \leq T_1$, a este tipo de umbralización se le conoce como Umbralización multinivel, y suele ser menos seguro que la Umbralización binaria, a la cual también se le llama Binarización. El motivo es la dificultad de establecer umbrales múltiples que logren aislar realmente las regiones de interés, sobre todos cuando el número de modos de los histogramas es alto.

El histograma de una imagen es ampliamente utilizado como herramienta tanto cualitativa como cuantitativa. Este corresponde a un gráfico de la distribución de valores de intensidad de los píxeles de una imagen (niveles de gris) o de una porción de la misma.

Podemos denotar como $h(i)$, el número de píxeles que dentro de la región de interés tiene el valor de intensidad i , donde $i = 0, 1, 2, \dots, L-1$ es el número posible de niveles de gris para la imagen. Los valores $h(i)$, corresponderán entonces a los valores del histograma. El gráfico del histograma es bidimensional y en él se gráfica $h(i)$ en función de i . Tal gráfico, puede proporcionar importante información acerca del brillo y contraste de una imagen así como de su rango dinámico. En las siguientes figuras se muestran los dibujos de dos histogramas típicos.

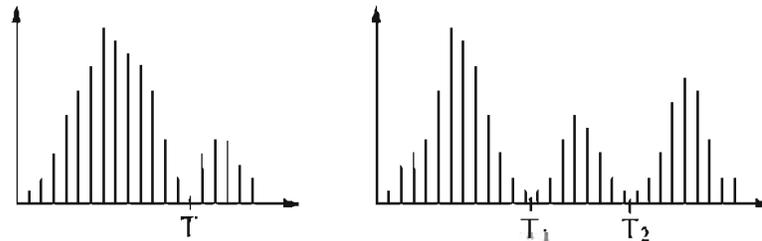


Figura N° 3.3. - Histogramas de nivel de gris que se pueden dividir en: (a) un umbral sencillo y (b) umbrales múltiples.

Fuente: [Gonzales, Woods, 1996]

En el eje de abscisas se representan los niveles de gris, y en el eje de ordenadas se representa $h(i)$, es decir, el número de ocurrencias del nivel de gris. Se puede decir que la iluminación de la imagen, desempeña un papel determinante en el establecimiento de la forma del histograma.

3.5. UMBRALIZACIÓN BINARIA

Para realizar la binarización, basta con explorar todos los puntos de la imagen y determinar qué puntos corresponderían al objeto y cuales al fondo.

Así que el éxito de éste método depende totalmente de cómo se divida el histograma.

La técnica más simple para realizar esto es dividir el histograma de la imagen usando un umbral simple T . La imagen particionada $g(x,y)$ está definida como:

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) > T \\ 0 & \text{si } f(x,y) \leq T \end{cases}$$

Entonces los píxeles etiquetados con 1 (o cualquier otro nivel de intensidad

conveniente) corresponden a los objetos, mientras que los píxeles etiquetados con 0 corresponden al fondo.

La selección del valor del umbral, se realiza generalmente a partir del histograma de la imagen. Así si una imagen está compuesta de un objeto que aparece en la escena sobre un fondo, entonces es de esperar que el histograma sea bimodal, es decir, si por ejemplo el objeto es más claro que el fondo, pues en el histograma aparecerán dos picos, el ubicado en los valores de gris más elevados correspondiente al objeto y otro pico para niveles de gris más bajos, correspondientes al fondo.

En la siguiente figura se muestra un histograma bimodal, en el cual el umbral se ubica entre los dos picos del histograma.

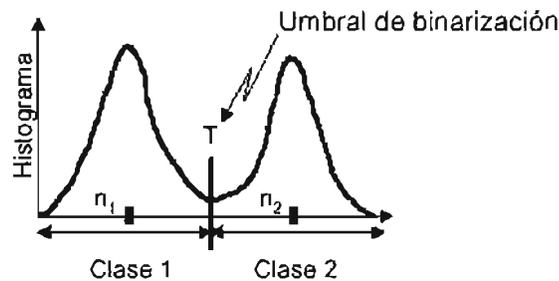


Figura Nº 3.4 –Gráfica Del Umbral.
Fuente: [Gonzales, Woods, 1996]

La selección automática del umbral, es un problema difícil, debido a que el histograma no siempre es bimodal, en cuyo caso resulta necesario combinar la información espacial presente en la imagen, con la información referente al nivel de gris.

A continuación, podemos observar un ejemplo:

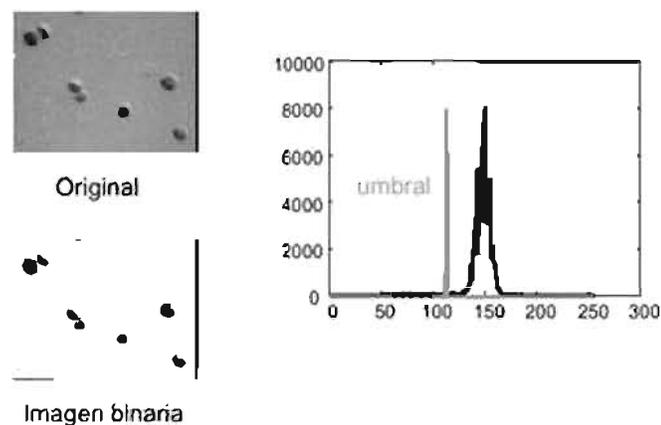


Figura N° 3.5 - Neuronal Biológica
Fuente: [KENNEY, RIEGER, 2003].

Para el caso del histograma aproximadamente bimodal, existen técnicas de detección automática del umbral, una de las cuales fueron ideadas por Otsu y ha sido reseñada en (Haralick y Shapiro, 1992) y se basa en la minimización de la varianza intra-grupo.

3.6. UMBRALIZACIÓN ÓPTIMA

La determinación del umbral es clave para conseguir resultados útiles, puesto que una mala elección del umbral nos podría llevar a errores.

Supongamos que una imagen contiene solo dos regiones de brillo principales. El histograma de una imagen de este tipo se puede considerar como una estimación de la función densidad de probabilidad de brillo $p(z)$.

Esta función de densidad total es la suma o mezcla de dos densidades unimodales, una para las regiones claras de la imagen y la otra para las oscuras. Además los parámetros de la mezcla son proporcionales a las superficies de cada región de brillo. Si se conoce o se supone la forma de las densidades, se puede determinar un umbral óptimo (en términos del error mínimo) por segmentación en las dos regiones de brillo de la imagen.

Supongamos que una imagen contiene dos valores combinados con ruido aditivo gaussiano. La función densidad de probabilidad de la mezcla es:

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$

Que para el caso gaussiano, es:

$$p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(z - \mu_1)^2}{2\sigma_1^2}\right] + \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(z - \mu_2)^2}{2\sigma_2^2}\right]$$

Donde μ_1 y μ_2 son los valores medios de los dos niveles de brillo, y σ_1 y σ_2 son las desviaciones estándar respecto a la media, y P_1 y P_2 son las probabilidades a priori de los dos niveles. Se debe satisfacer la restricción:

$$P_1 + P_2 = 1$$

Por lo que la densidad de la mezcla tiene cinco parámetros desconocidos. Si se conocen todos los parámetros, se puede determinar con facilidad el umbral óptimo.

Supongamos que las regiones oscuras corresponden al fondo y que las claras corresponden a los objetos. En este caso $\mu_1 < \mu_2$, y se puede definir un umbral T de forma que todos los píxeles con nivel de gris superior a T se consideren puntos objeto. La probabilidad de clasificar (erróneamente) un punto objeto como un punto del fondo es:

$$E_1(T) = \int_{-\infty}^T p_2(z) dz$$

De forma similar, la probabilidad de clasificar un punto fondo como un punto objeto es:

$$E_2(T) = \int_T^{\infty} p_1(z) dz$$

Así pues, la probabilidad total de error:

$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

Derivando $E(T)$ con respecto a T e igualando a 0, encontraremos el valor umbral para el que el error es mínimo. Procediendo así, tenemos:

$$P_1 p_1(T) = P_2 p_2(T)$$

Aplicando este resultado a la densidad gaussiana, tomando logaritmos, y simplificando, obtenemos una ecuación cuadrática:

$$AT^2 + BT + C = 0$$

Dónde:

$$A = \sigma_1^2 - \sigma_2^2$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)$$

$$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln(\sigma_2P_1/\sigma_1P_2)$$

La posibilidad de tener dos soluciones indica que se pueden necesitar dos valores de umbral para obtener la solución óptima.

Si las varianzas son iguales $s^2 = s_1^2 = s_2^2$, es suficiente con un umbral único :

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$

Si las probabilidades anteriores son iguales, $P_1 = P_2$, como tanto si $s = 0$, el umbral óptimo es el promedio de las medias.

La determinación del umbral óptimo se puede realizar de forma similar para otras densidades de un solo modo de la forma conocida, tales como la de Raleigh y la normal.

Se puede utilizar un método de error medio por mínimos cuadrados para estimar los parámetros de una imagen a partir de un histograma. Por ejemplo, el error cuadrático medio entre la densidad de la mezcla $p(z)$ y el histograma experimental $h(z_i)$ de n puntos es :

$$e_{ms} = \frac{1}{n} \sum_{i=1}^n [p(z_i) - h(z_i)]^2$$

En general, la determinación analítica de los parámetros que minimizan este error cuadrático medio no es un tema sencillo. No obstante, para el caso gaussiano, el cálculo directo de igualar las derivadas parciales a 0 conduce a un conjunto de ecuaciones simultáneas trascendentes que normalmente sólo se pueden resolver por procedimientos numéricos. Dado que el gradiente se puede calcular fácilmente, es posible usar para minimizar e_{ms} un gradiente conjugado o un método de Newton para ecuaciones no lineales simultáneas.

Con cualquiera de estos métodos iterativos, se pueden especificar los valores de salida. Puede ser suficiente con suponer que a priori todas las probabilidades iguales. Los valores de salida para las medias y las varianzas se pueden determinar detectando modos en el histograma o simplemente dividiendo el histograma en dos partes respecto a su valor medio, y calculando las medias y varianzas de las dos partes para utilizarlas como valores de salida [UMA-2006].

3.7. SELECCIÓN DE UMBRAL BASADA EN LAS CARACTERÍSTICAS DEL LÍMITE

La identificación fidedigna de los picos del modo de un cierto histograma, es uno de los aspectos más importantes de la selección de umbral.

Esta capacidad es particularmente importante para la selección automática de umbrales en situaciones donde las características de una imagen pueden variar en un amplio rango de distribuciones de intensidad.

De forma intuitiva se podría pensar que las posibilidades de seleccionar un “buen” umbral aumentan considerablemente si los picos del histograma son altos, estrechos, simétricos y separados por valles profundos.

Para mejorar la forma de los histogramas se puede solamente considerar aquellos píxeles que están situados en el límite entre los objetos y el fondo o cerca de él. De este modo los histogramas serán menos dependientes de los tamaños relativos de los objetos y del fondo.

Además, la utilización de píxeles que satisfacen alguna de las medidas sencillas basadas en los operadores gradiente y laplaciano tienen tendencia a hacer más profundo los valles entre los picos del histograma.

Este método tiene un problema, que es la suposición implícita de que se conoce el límite entre los objetos y el fondo. Esta información no está claramente disponible durante

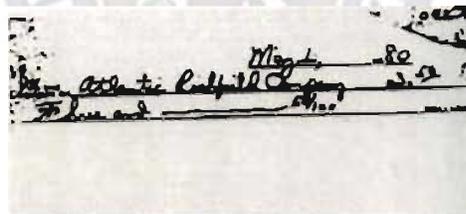
la segmentación, si se encuentra una división entre objetos y fondo, precisamente porque la segmentación está por todas partes.

Sin embargo, se podría obtener una indicación de si un píxel está en un borde calculando su gradiente. Además, el uso del laplaciano puede proporcionar información respecto a si un píxel dado está situado en la parte oscura (fondo) o clara (objeto) de un borde. El valor medio del laplaciano es cero en la transición de un borde, por lo que en la práctica los valles de los histogramas formados por los píxeles seleccionados por un criterio de gradiente/laplaciano, se puede esperar que estén escasamente poblados. Esta propiedad produce los deseados valles profundos de los que hablábamos. [UMA-2006]

Las siguientes imágenes muestran un ejemplo de esta técnica:



(a) Imagen original



(b) Imagen segmentada

Figura Nº 3.6 -Segmentación de imágenes
Fuente: [KENNEY, RIEGER, 2003].

3.8. UMBRALIZACIÓN MULTINIVEL

Podemos describir dos casos en la umbralización multinivel. Uno sería umbralización multinivel de luminancia y el otro umbralización multinivel de componentes de color.

3.8.1. UMBRALIZACIÓN MULTINIVEL DE LUMINANCIA

En este caso, la idea es similar a la binarización, solo que con varios umbrales que definen varias clases.

La idea de lo que se haría en el histograma la podemos ver gráficamente:

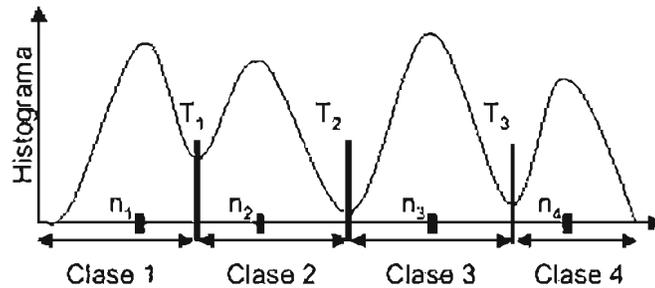


Figura N° 3.7.-Histograma de clases

Fuente: [UMA-2006]

Una efectiva segmentación multinivel se puede lograr aplicando un método iterativo (como propone Tomita). En la primera etapa del proceso la imagen es umbralizada para separar regiones con más brillo de aquellas más oscuras, para ello se localiza un mínimo entre los modos del histograma. [UMA-2006].

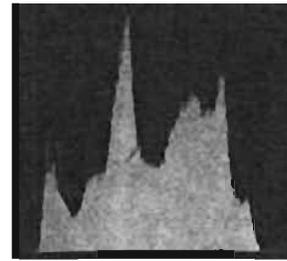
De cada parte segmentada, se forman otros histogramas. Si estos no son unimodales, las partes son umbralizadas otra vez.

El proceso se repite hasta que los histogramas de cada parte se concierten en unimodales. Veamos este método en un ejemplo:

Figura N° 3.8 -Histogramas de imágenes
Fuente: [UMA-2006]



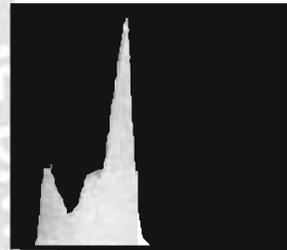
Imagen original



Histograma original



Segmento 0



Histograma del segmento 0



Segmento 1



Histograma del segmento 1

Estos resultados corresponderían al primer nivel de segmentación.

El segundo nivel de segmentación, daría los siguientes resultados:

Figura N° 3.9. - Segmentación de imágenes
Fuente: [UMA-2006]



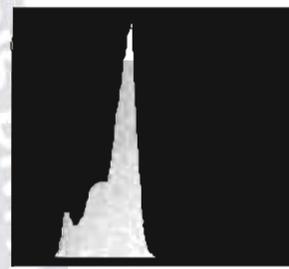
Segmento 00



Histograma del segmento 00



Segmento 01



Histograma del segmento 01



Segmento 10



Histograma del segmento 10



Segmento 11



Histograma del segmento 11

Como vemos la imagen inicial ha sido segmentada en cuatro segmentos de escala de grises.

3.8.2. UMBRALIZACIÓN MULTINIVEL DE COMPONENTES DE COLOR

En algunos casos, un sensor podría facilitar más de una variable para caracterizar a cada píxel de una imagen.

Un ejemplo típico es el de las imágenes en color, donde se utilizan los componentes rojo (R), verde (G) y azul (B) para formar una imagen de color compuesta.

Cada píxel estaría caracterizado por tres valores, lo cual permitiría construir un histograma tridimensional.

El concepto de umbralización se transforma ahora en encontrar agrupaciones de puntos en un espacio tridimensional.

Si suponemos que se han encontrado en el histograma K agrupaciones de puntos significativas, la imagen se puede segmentar asignando una intensidad a los píxeles cuyos componentes RGB están próximos a una agrupación y otra intensidad al resto de píxeles de la imagen.

La principal dificultad estriba en la búsqueda de agrupaciones, la cual se convierte en una tarea compleja que se incrementa conforme aumenta el número de variables[UMA-2006].



CAPITULO IV

CONSTRUCCION DE LA RED NEURONAL DE RETROPROPAGACION

RESUMEN

En el presente capítulo se establece el modelado de una neurona, la red neuronal de retropropagación, su arquitectura, además del análisis y desarrollo del sistema, en base a las herramientas y métodos que sugiere la ingeniería de software con el fin de obtener un producto de alta calidad que sirva para demostrar la hipótesis planteada. Se utiliza Java como lenguaje de programación, y se utiliza los diagramas UML para el modelado.

4.1. DISEÑO DE LA RED NEURONAL

La implementación computacional de la red neuronal de retropropagación, fue realizada en software, en un lenguaje de programación, orientado a objetos, de alto nivel, denominada "Java" y desarrollada por la empresa Oracle (Ex SunCorporation). La versión utilizada es "JAVA 1.6" concebida para plataformas: Windows NT/2000, Windows XP, Windows Vista, Windows 7, Linux y Mac. Este lenguaje es utilizado, debido a su buen rendimiento en dichas plataformas. Esto permite con un mismo código, sin necesidad de una gran modificación, el poder tener un solo programa para las tres plataformas de sistemas operativos: Windows, Linux y Mac.

Previamente a la implementación, es muy importante definir los principales elementos y características que conforman la red neuronal y homologar su modelación matemática a una posible modelación computacional a través de algoritmos y estructuras de datos.

La red neuronal de retropropagación, en este proyecto, se utiliza como un clasificador para reconocimiento de patrones y se espera que logre aprender a reconocer cromosomas humanos por medio de imágenes bidimensionales de sus cromosomas y de baja resolución.

4.2. CODIFICACIÓN DE LA NEURONA ARTIFICIAL

La neurona artificial se puede modelar como una estructura de dato denominada registro. Para cada neurona (registro Neurona) se deben definir tres valores reales que son tres características fundamentales que las definen. Una característica es su salida (y), otra

es su función de activación (net) y la tercera son los términos de error (δ) obtenidos de la propagación hacia atrás del error (Backpropagation) donde se propaga información desde la salida y hacia la entrada de la red para cada una de las neuronas que procesan datos. La neurona artificial puede definirse e implementarse computacionalmente a través de Java de la siguiente manera:

```
Class neurona {  
    private double y;  
    private double net ;  
    private double Dlta;  
}
```

Para cada neurona artificial existe una cuarta característica, que es la función de transferencia utilizada en red de retropropagación. La utilizada es la función sigmoideal (Cáceres, 2002) y que fue implementada y definida como tal, en JAVA, de la siguiente manera:

```
private double sigmoid (double x) {  
    return (1.0 / (1.0 + Math.exp(-x) ));  
}
```

4.3. MODELACIÓN Y CODIFICACIÓN DE LA ARQUITECTURA

La arquitectura de la red neuronal de retropropagación, utilizada en este proyecto, puede verse representada en la Figura 3.1, que representa al modelo matemático de la red. Así, también, la modelación computacional utilizada puede verse representada en la Figura 4.1.

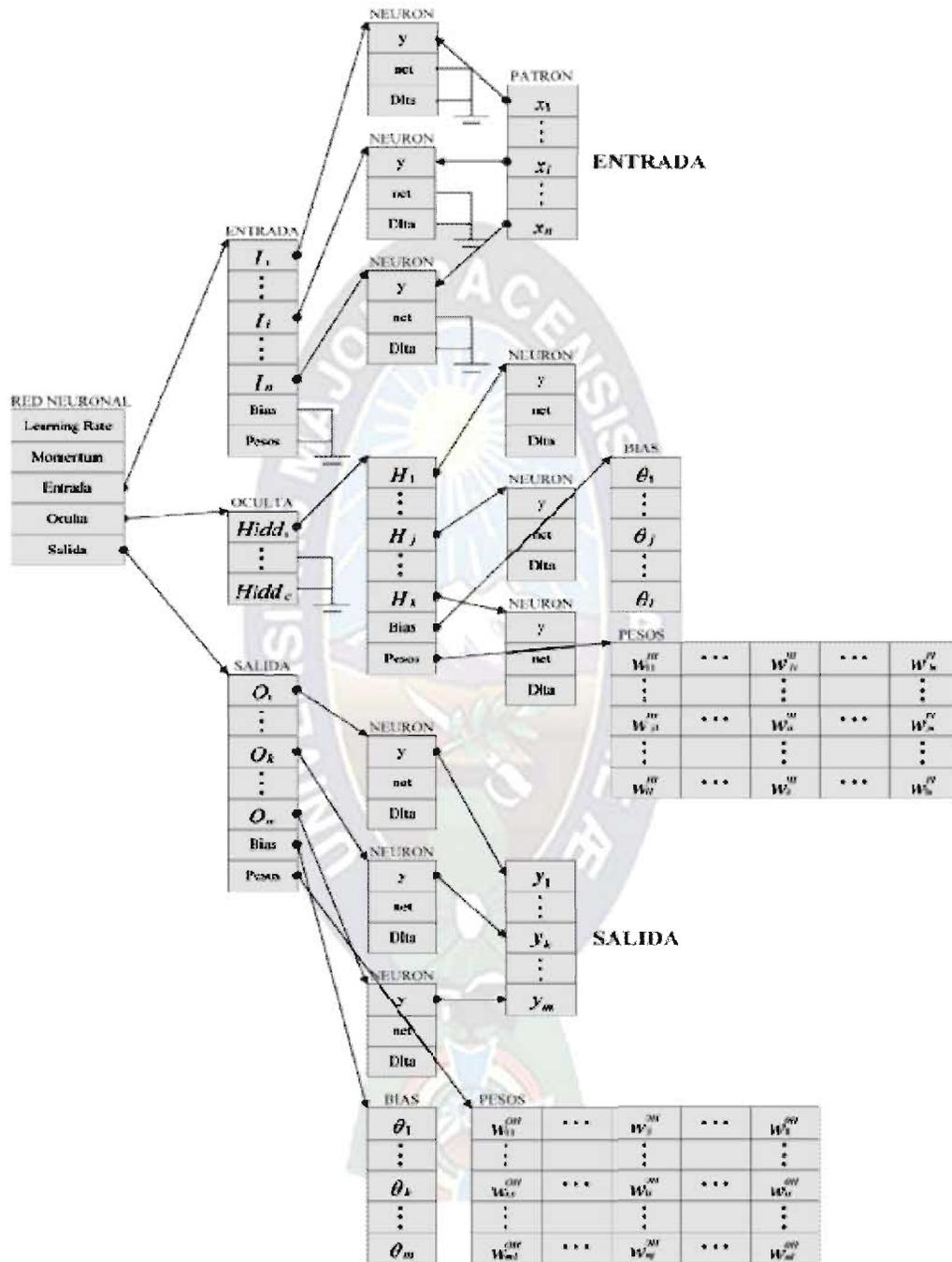
Muy importante son las conexiones interneuronales o pesos de la red, pues almacenan el conocimiento que la red adquiere en el proceso de aprendizaje. El modelo matemático, representa dicho conocimiento almacenado en matrices de conectividad (Ver Ecuaciones 2.11 y 2.12). El modelo computacional también representa los pesos como matrices, o mejor dicho, como vectores dinámicos de vectores dinámicos, que existen en Java.

Para implementar la memoria dinámica, en dicho lenguaje, se exige la utilización de listas dinámicas enlazadas, lo que es una muy buena herramienta de modelación dinámica, pero que almacena información en espacios de memoria disjuntos o en cualquier espacio de memoria disponible, accediendo a ella mediante direccionamientos de memoria conocida como punteros. La ventaja que posee la memoria estática, es que almacena en forma ordenada la información, en casilleros de memoria adjuntos, lo que permite acelerar considerablemente la búsqueda y el acceso a ella, su limitante era el hecho de ser estática, como su nombre bien lo dice, o sea, una vez definido el número de componentes del vector, no era posible variarlo en tiempo de ejecución. Java, permite la existencia de vectores dinámicos, lo que es una poderosa herramienta para modelar los vectores bidimensionales utilizados en la red neuronal. En la Figura 4.1, se muestran las estructuras de datos utilizadas para la implementación de la red neuronal en software. Principalmente, existe un vector de cinco componentes, denominado "Red Neuronal", que en las primeras dos componentes almacena los parámetros del entrenamiento, tales como: tasa de aprendizaje (learningrate) y momento (momentum), pero que además, almacena en sus tres componentes restantes los punteros a las estructuras vectoriales dinámicas que representan las capas de la red neuronal y que son: entrada, oculta y salida.

Las estructuras vectoriales dinámicas "entrada" y "salida", almacenan los punteros a los registros que representan las neuronas artificiales denominadas "Neuron", correspondiente, respectivamente, a las neuronas de entrada y de salida de la red neuronal. Pero, además, estas estructuras vectoriales dinámicas (entrada y salida) almacenan los punteros a los vectores dinámicos "bias" y "pesos", que representan los umbrales de activación de cada neurona y los pesos correspondiente al aprendizaje de la red neuronal, respectivamente.

Es claro que para la capa de entrada (vector dinámico entrada), las componentes bias y pesos, se muestran apuntadas a tierra (null), debido a que no existen umbrales, ni pesos en la capa de entrada de esta red neuronal. Lo que no ocurre para la capa de salida (vector dinámico salida), que sí posee umbrales de activación para las neuronas de salida y pesos que corregir en la etapa de entrenamiento.

FiguraN° 4.1 MODELACIÓN COMPUTACIONAL, MEDIANTE ESTRUCTURA DE DATOS, DE LAARQUITECTURA DE LA RED NEURONAL DE RETROPROPAGACION.



La estructura vectorial dinámica oculta, es levemente diferente de las otras dos ya mencionadas. Esta diferencia, radica en que las capas de entrada y de salida en una red neuronal son únicas, lo que no es el caso de las capas ocultas, donde pueden existir varias. Debido a que en este proyecto se utilizó una red neuronal de tres capas, y por tanto, de sólo una capa oculta, la estructura vectorial dinámica “oculta” posee en su primera componente el puntero hacia la estructura vectorial dinámica que representa la capa oculta, donde almacena los registros neurona, correspondiente a las neuronas ocultas y los punteros bias y pesos, correspondiente, respectivamente, a los umbrales de activación de las neuronas ocultas y a las conexiones neuronales entre la capa de entrada y la oculta.

La modelación computacional de los pesos de la red se simplifica considerablemente y se pueden modelar declarándolos como simples variables y que, en Java, se expresa de la siguiente forma:

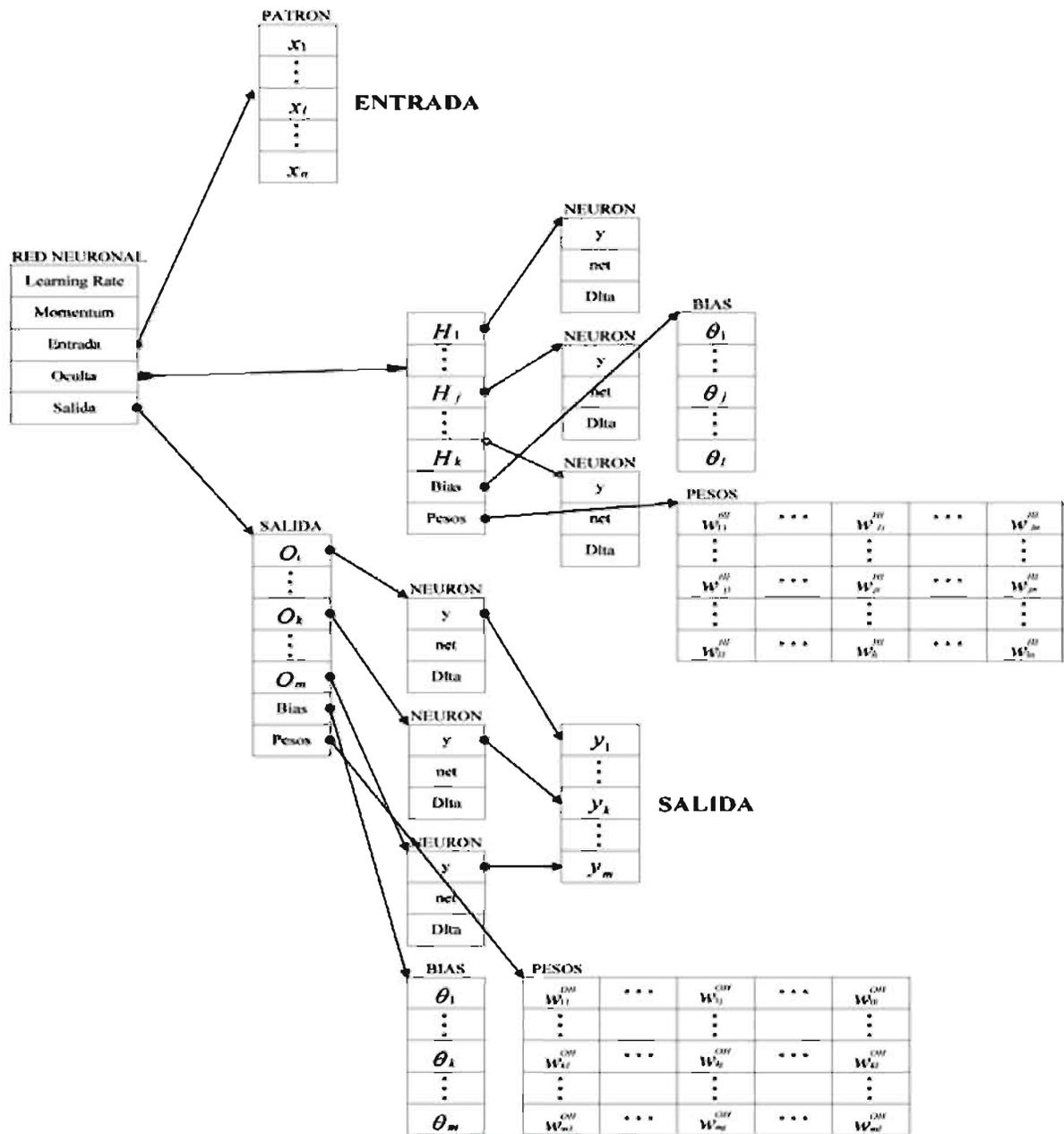
```
private double wHl[][] = new double[][];  
private double wHO[][] = new double[][];
```

Análogamente, otra característica importante que se define en la arquitectura de una red y en su aprendizaje son los umbrales de activación (Bias), que son necesarios en cada neurona para definir el corrimiento horizontal de la función de transferencia que varía en cada aprendizaje en la medida que mejor se adapta a los patrones aprendidos y que no siempre está centrada en el origen. Los bias se modelan también como variable, mediante la utilización de vectores dinámicos unidimensionales y que, en Java, se expresan de la siguiente forma:

```
private double [] wHl0 = new double [];  
private double [] wOH0 = new double [];
```

La red de retropropagación por ser una red multicapa, y que en este caso utiliza tres capas (entrada, oculta y salida), necesita que en cada capa el número de neuronas sea variable en tiempo de ejecución, por tanto, las capas de la red también se modelaron mediante la utilización de vectores dinámicos unidimensionales, donde cada capa, componente del vector, debe ser una estructura de dato del tipo Neurona definida anteriormente.

FiguraN° 4.2 SIMPLIFICACIÓN EN LA ESTRUCTURA DE DATOS DEL MODELO COMPUTACIONAL



La modelación, en Java, de las neuronas ocultas y de salida de la red neuronal se expresan de la siguiente manera:

```

neuronaNH = new neurona();
neurona NO = new neurona();
    
```

Donde:

NH: Representa a las neuronas ocultas (NeuronHidden).

NO: Representa a las neuronas de salida (Neuron Output).

Cabe hacer notar, que las neuronas de entrada sólo distribuyen los patrones de entrada a las neuronas ocultas, por lo tanto, el modelo computacional de la estructura de dato que representa la arquitectura de la red de retropropagación (Ver Figura 4.1) se modifica simplificándose considerablemente (Ver Figura 4.2).

Como se mencionó al inicio del capítulo, otras de las características importantes de la red, es considerar el tipo de patrones de entrada y de salida que ésta procesará. Debido a la modificación anterior, es necesario definirse el “vector de entrada” para la red (X), como el “vector de salida deseado” (NOD) que se espera que la red aprenda, y que, además, realiza la supervisión, permitiendo que la red calcule su error y corrija sus pesos. Dichos vectores se implementaron como vectores dinámicos unidimensionales, que en Java, se definieron de la siguiente manera:

```
double x[] = new double[];  
int NOD[] = new int[];
```

Cabe hacer notar que el vector de entrada representa una imagen en escala de grises con 2 niveles posibles, con valores enteros entre 0-255 por lo tanto, las componentes del vector de entrada se definieron como “int”. Por otra parte, las componentes del vector de salida por ser asintóticamente binarias [0, 1], se definieron como “double”.

4.4. CODIFICACIÓN DE LA PROPAGACIÓN HACIA ATRÁS

La modelación matemática del funcionamiento de la red neuronal de retropropagación, está basado en el método de propagación hacia adelante. Dicho modelo fue implementado computacionalmente, en Java como una función (function), debido a que tanto en la etapa de funcionamiento, como en la etapa de aprendizaje de la red, este es requerido y utilizado constantemente en cada iteración de dichos procesos.

Este procedimiento utiliza dos parámetros importantes en su definición. El parámetro "inputA" que representa al vector de entrada a la red neuronal y el parámetro "outputN", que representa al vector de salida de la red. Estos parámetros son importantes, ya que informan al procedimiento quien será el vector de entrada en un instante dado, o sea, que patrón está siendo propagado, y cuál será el vector de salida asociado por la red neuronal.

Es claro que el vector de salida obtenido por la red neuronal dependerá de los pesos almacenados, y por tanto. Existirá un error con respecto al vector de salida deseado o esperado (Ver ecuación 2.32).

Para asociar mejor la relación entre este procedimiento y dos formulación matemática se sugiere revisar las Ecuaciones 2.20 y 2.26 (Ver Capitulo II)

El procedimiento de retropropagación (feedForward) fue implementado en java de la siguiente manera:

```
private void feedForward()
{
    double sum2 = 0.0
    // calculo de los pesos de las neuronas ocultas
    for(int i = 0; i < numHidden; i++) {
        sum2 = biasH[i];
        for(int j = 0; j < numInput; j++) {
            sum2 += hiddenW[i][j] * inputA[j];
        }
        hiddenN[i] = sum2;
        hiddenA[i] = sigmoid(sum2);
    }

    // calculando los nuevos pesos de salida
    for(int i = 0; i < numOutput; i++) {
        sum2 = biasO[i];
        for(int j = 0; j < numHidden; j++) {
            sum2 += outputW[i][j] * hiddenA[j];
        }
        outputN[i] = sum2;
    }
}
```

Donde:

hiddenA: Representa la cantidad de neuronas de entradas.

numHidden: Representa la cantidad de neuronas ocultas.

outputN: Representa la cantidad de neuronas de salidas.

4.5. CODIFICACIÓN PARA LA MODIFICACIÓN DE PESOS

La implementación computacional de la modificación de pesos en una red neuronal de retropropagación es muy importante, debido a que en esta etapa la red logra adaptar sus conexiones neuronales para guardar su conocimiento, aprendiendo de su propio error. Este proceso está implementado, en Java, como un método. Se requiere la utilización de dos parámetros importantísimos para la modificación de pesos; uno es el parámetro “tasa de aprendizaje” o “learnigrate” (Lr), que define la forma como el vector gradiente avanza por la superficie de error. El otro parámetro es el “momento” o “momentum” (Mt), que otorga información al vector gradiente de las últimas modificaciones realizadas en el avance por la superficie de error, lo cual le permite tener una mayor tendencia a dicho curso, generando un aumento en la rapidez de la convergencia del algoritmo. Estos dos parámetros, bien seleccionados, permiten que el algoritmo encuentre el mínimo global de la superficie de error, sin caer en los posibles mínimos locales. Para asociar mejor la relación entre este procedimiento y su formulación matemática se sugiere revisar las Ecuaciones 3.30 y 3.31 (Ver Capítulo II).

El procedimiento de modificación de pesos de la red (WeightChange) fue implementado, en Java, de la siguiente manera:

```

public void WeightChange()
{
    inti,j,k;
    floatLR,Mon,aux;
    Lr=float(EditLearningRate.textfield);
    Mt=float(EditMomentum.textfield);
    for(k=0;k<=M-1;k++)
        {
            aux=wOH0[k];
            wHO0[k]= wHO0[k]+(Lr*NO[Dlta])+Mt*(wOH0[k]-wOH0[k]-wOH0_Aux[k]);
            wOH0_Aux[k]=aux;
            for(j=0;j<=L-1;j++)
                {
                    aux=wOH0[k,j];
                    wHO[k,j]=wOH[k,j]+(Lr*NO[k].Dlta*NH[j].y)+Mt*(wOH[k,j]-wOH_Aux[k,j]);
                    who_Aux[k,j]=aux;
                }
            }
        for (j=0;j<=L-1;j++)
            {
                aux=wHIO[j];
                wHIO[j]=wHIO[j]+(Lr*NH[j].Dlta)+Mt*(wHIO[j]-wHIO_Aux[j]);
                wHIO_Aux[j]=aux;
                for(i=0;i<=N-1;i++)
                    {
                        aux=wHI[j,i];
                        wHI[j,i]=wHI[j,i]+(Lr*NH[j].Dlta*x[i])+Mt*(wHI-wHI_Aux[j,i]);
                        wHI_Aux[j,i]=aux;
                    }
            }
    }
}

```

4.6. CODIFICACIÓN DEL APRENDIZAJE

La implementación del aprendizaje es fundamental para el entrenamiento de la red neuronal. La red utilizará todos los procedimientos explicados anteriormente y que le permitirán ajustar sus conexiones neuronales para aprender los patrones vectoriales que serán entrenados. De esta forma, la red para cada iteración del aprendizaje propagará hacia adelante un patrón. Calculará, entonces, los errores cometidos en las neuronas de las capas ocultas y de salida mediante la propagación hacia atrás. Con dicha información, procederá a realizar la modificación de los pesos para minimizar el error cuadrático medio total, obtenido de la comparación del vector de salida versus el esperado (supervisión).

La implementación computacional del aprendizaje de la red neuronal de retropropagación en este proyecto, requiere de los siguientes tres procedimientos:

Propagación hacia adelante

Propagación hacia atrás.

Modificación de pesos.

Existen dos procedimientos encargados de ingresar a la red los patrones de entrada y los de salida deseada, logrando así la heteroasociación. Dichos procedimientos son:

PatternsRead: que lee los patrones vectoriales de entrada.

OutputRead: que lee la clasificación de salida deseada o correcta.

Existen algunas variables locales definidas en el procedimiento de aprendizaje y que son utilizadas con finalidades específicas de almacenamiento en que:

p: p-ésimo patrón que está siendo entrenado.

Epoch: número de épocas que la red será entrenada (Cáceres, 2002).

Iteraciones: cantidad de procesos que la red ha completado.

MontoEpoch: épocas de entrenamientos deseadas por el usuario.

Análogamente, existen también tres variables globales utilizadas para almacenar el número de patrones de entrenamientos (NumPatternsTraining), el número de patrones de validación (NumPatternsValidation) y el número de patrones de prueba (NumPatternsTest).

La implementación del algoritmo de aprendizaje para la red neuronal de Retropropagación se puede expresar, en Java, de la siguiente manera:

```

public void WeightChange()
{
    inti,j,k;
    floatLR,Mon,aux;
    Lr=float(EditLearningRate.textfield);
    Mt=float(EditMomentum.textfield);
    for(k=0;k<=M-1;k++)
        {
            aux=wOH0[k];
            wHO0[k]= wHO0[k]+(Lr*NO[Dlta])+Mt*(wOH0[k]-wOH0[k]-wOH0_Aux[k]);
            wOH0_Aux[k]=aux;
            for(j=0;j<=L-1;j++)
                {
                    aux=wOH0[k,j];
                    wHO[k,j]=wOH[k,j]+(Lr*NO[k].Dlta*NH[j].y)+Mt*(wOH[k,j]-wOH_Aux[k,j]);
                    who_Aux[k,j]=aux;
                }
            }
        for (j=0;j<=L-1;j++)
            {
                aux=wHIO[j];
                wHIO[j]=wHIO[j]+(Lr*NH[j].Dlta)+Mt*(wHIO[j]-wHIO_Aux[j]);
                wHIO_Aux[j]=aux;
                for(i=0;i<=N-1;i++)
                    {
                        aux=wHI[j,i];
                        wHI[j,i]=wHI[j,i]+(Lr*NH[j].Dlta*x[i])+Mt*(wHI-wHI_Aux[j,i]);
                        wHI_Aux[j,i]=aux;
                    }
            }
    }
}

```

4.7. CODIFICACIÓN PARA CALCULAR LAS TASAS DE RECONOCIMIENTO Y ERRORES CUADRÁTICOS MEDIOS

El software para poder entregar las estadísticas con los porcentajes de las tasas dereconocimiento y del error cuadrático medio, debe utilizar el siguiente procedimiento que se proponen a continuación y que utiliza el criterio propuesto por R. Lippman(Lippman, 1987). Este procedimiento calcula el error que se comete por patrón entrenado y determina si puede ser, o no, considerado un acierto (1) o un rechazo (0). La implementación del procedimiento denominado “Estadísticas” se puede expresar, en Java, de la siguiente manera:

```

public void Estadisticas(double stre[],double stra[], intNumPat, intPt)
{
    intk,Clase,posmayor;
    double mayor;
    Clase=this.Trunc(Numpat/Pt);
    stre[Numpat]=0;
    stra[Numpat]=0;
    mayor=0;
    posmayor=0;
    for(k=0;k<=M-1;k++)
    {
        stre[Numpat]=stre[Numpat]+sqr(NOD[k]-NO[k].y);
        if(NO[k].y>mayor)
            {
                mayor=NO[k].y;
                posmayor=k;
            }
    }
    stre[Nompat]=0.5*stre[Numpat];
    if(Clase=posmayor)
        stra[Numpat]=0;
    else
        stra[Numpat]=1;
    }
}

```

El método “Estadísticas” requiere de la utilización de dos funciones definidas como “mayor” y “porcentaje”, que entregan, respectivamente, la posición de la componente de mayor en el vector de salida de la red neuronal y el porcentaje de los componentes del vector “str” a procesar, que puede ser tanto el vector de reconocimiento, como el de errores.

```

public int mayor(neuron strn)
{
    int k, pos;
    double aux;
    aux=0;
    pos=0;
    for(k=0; k<=M-1; k++)
    {
        if(strn[k].y>aux)
        {
            aux=strn[k].y;
            pos=k;
        }
    }
    return pos;
}

public double Porcentaje(soublestr, int Numpat)
{
    int j;
    Result=0;
    for(j=0; Numpat-1; j++)
    Result=Result+str[j];
    Return((100/Numpat)*Result);
}

```

4.8. TRATAMIENTO DE LA IMAGEN

Se vieron varias alternativas para de tratamiento digital de imágenes y se consideraron las ventajas y desventajas de los mismos, pero para propósitos de reconocimiento de cromosomas nos interesa más la detección de bordes, y la segmentación de imágenes.

4.8.1 DETECCIÓN DE BORDES DE LA IMAGEN

Una de las principales tareas de nuestra aplicación es la detección de especificas muestras en la imagen de un cromosoma, estas muestras son los elementos básicos para que la red neuronal pueda realizar los cálculos, asumiremos que la imagen está completamente en dos dimensiones (2D) por ejemplo la imagen que se muestra la figura, Se aplicara el conocido algoritmo de Prewit para la detección de bordes.

4.8.2. ALGORITMO DE PREWITT PARA LA DETECCIÓN DE BORDES.

El algoritmo de Prewitt produce una imagen donde los niveles altos en la escala de grises indican la presencia de bordes entre dos imágenes. El filtro de Prewitt de detección de bordes computa la raíz del cuadro de dos modelos de 3 x 3. Este es uno de los más populares filtros de detección de bordes de 3 x 3.

El filtro de detección de bordes Prewitt usa estos dos modelos 3 x 3 para calcular el valor de la gradiente.

X	-1	0	1
	-1	0	1
	-1	0	1

Y	1	1	1
	0	0	0
	-1	-1	-1

Figura N° 4.3. Filtro de Prewitt

Ahora consideremos la siguiente imagen de 3 x 3

a1	a2	a3
a4	a5	a6
a7	a8	a9

Figura N°4.4. Imagen Procesada

Dónde: a1, a2, a3..., a9. Son los niveles de gris de cada pixel en el filtro de la ventana.

$$X = -1 * a1 + 1 * a3 - 1 * a4 + 1 * a6 - 1 * a7 + 1 * a9$$

$$Y = 1 * a1 + 1 * a2 + 1 * a3 - 1a7 - 1a8 - 1a9$$

$$\text{Gradiente Prewitt} = \text{Math.sqrt}(x * x + y * y)$$

Todos los pixeles son filtrados, a fin de de filtrar los pixeles localizados al lado del borde de una imagen, los pixeles de los bordes son replicados para tener suficiente información.

4.8.3. ADELGAZAMIENTO DE LOS BORDES

Como se señaló anteriormente la idea de identificar una imagen del cromosoma de los tipos específicos es que el borde es suficiente para especificar el tipo de cromosomas al que pertenece. Para lograr esto es necesario identificar exactamente los bordes. Para la detección de borde aplicamos el algoritmo de Prewitt para identificar los bordes con un umbral pre configurado, y después de la detección de bordes tenemos que aplicar el algoritmo de adelgazamiento para reducir al mínimo el borde.

El algoritmo usado para el adelgazamiento es recursivo y minimiza las líneas encontrados con un ancho de un pixel compara el pixel actual con patrones específicos y luego se reduce al mínimo.

4.8.4. OBTENCIÓN DE MUESTRAS

La parte central de esta aplicación son las muestras de una imagen cromosómica que son encontradas después del procesamiento de la imagen.

La idea detrás de la transferencia de las muestras de la imagen a la red neuronal y su utilización es, que los ángulos seno y coseno de la figura representa los criterios de un patrón de reconocimiento.

La figura 4.5 muestra una parte de la imagen que fue recientemente procesada con los algoritmos de detección de bordes y adelgazamiento.

Para tener una idea que se muestra en la imagen:

- Línea verde: La figura del cromosoma luego de la detección de bordes y adelgazamiento
- Línea Roja: este cuadrado representa un punto en la figura del cromosoma que divide del próximo recuadro de la figura.

- Línea Azul: el compuesto del centro de los dos cuadrados de donde calcularemos los ángulos Seno y Coseno, la línea azul es la representación de las muestras del cromosoma.

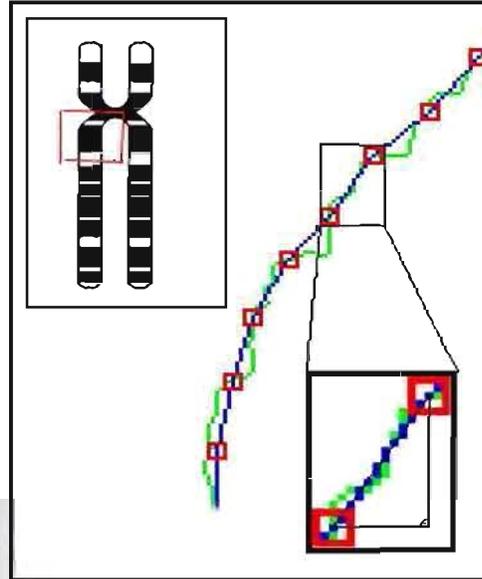


Figura N° 4.5. Resultado de la segmentación
Fuente: Elaboración Propia.

Una visión más profunda sobre el zoom del pequeño triángulo sobre esta imagen. Es de esta forma de representación que obtendremos las muestras de cromosoma y poder pasarlos a la red neuronal para su procesamiento.

En la figura 4.6 podemos observar el triángulo que representa el conjunto de muestras de un punto en la imagen del cromosoma, que serán evaluados por la red neuronal.

Los ángulos A y B son necesarios para ajustar las capas de la red neuronal, ya que con estos ángulos fácilmente se puede representar la dirección de la hipotenusa del punto P1 a P2 que es absolutamente necesario para la representación de una imagen de un cromosoma.

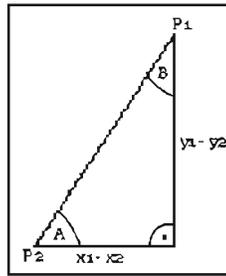


Figura 4.6 Triangulo de las muestras.
Fuente: Elaboración propia.

4.9. DESARROLLO DEL PROTOTIPO

Para el desarrollo del prototipo del sistema de Clasificación de Cromosomas (CCROM) se hace uso de la Ingeniería de software orientado a objetos que sigue los mismos lineamientos que el paradigma convencional que involucra análisis, diseño, implementación y pruebas, con la principal ventaja de promover la reutilización de código, debido a que este hace uso del modelo evolutivo de desarrollo, la suma de estos conceptos hacen del paradigma orientado a Objetos ideal en la construcción del prototipo.

Dentro de las fases de análisis y diseño Orientado a Objetos se hace uso de la notación UML. La implementación del prototipo se realiza en el lenguaje de programación Java. También se diseñan los casos de prueba y se evalúa la misma mediante métricas de calidad orientada a objetos. Para verificar la validez del prototipo.

4.9.1. ÁMBITO DEL SISTEMA DE CLASIFICACIÓN DE CROMOSOMAS

El prototipo identifica una imagen de un cromosoma humano presentado al sistema en una imagen bidimensional, que es leída por el sistema la imagen con el cromosoma es procesada para una posterior detección del cromosoma en la imagen y posterior clasificación.

4.9.2. MODELO DE DESARROLLO

Para la construcción del prototipo, se eligió el Modelo de desarrollo en Espiral debido a que el sistema presenta una naturaleza evolutiva, además que permite el desarrollo de una versión incremental de CCROM.

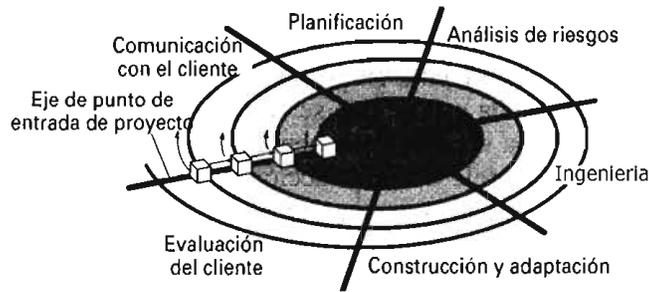


Figura N° 4.7. Modelo de desarrollo
Fuente: Elaboración según [Pressman, 2002]

4.9.3. RECURSOS

Para el desarrollo del producto se requerirán de los siguientes recursos de software como también hardware.

4.9.3.1. RECURSOS DE SOFTWARE REUTILIZABLE

Los recursos de software reutilizable con el que cuenta son, la interface de usuario misma que se establece mediante una biblioteca de clases GUI. Esta interfaz es simple, pero dada su construcción es posible su evolución para posteriores versiones. También el modelo de red neuronal se desarrolla con esta modalidad.

4.9.3.2. RECURSOS DE ENTORNO

El modelado para el desarrollo del CCROM (Sistema de Clasificación de Cromosomas) como se menciona anteriormente se lo construye mediante el Lenguaje Unificado de Modelado UML puesto que se hace uso del modelo de desarrollo espiral y un lenguaje de desarrollo orientado a objetos.

La implementación del CCROM se realiza bajo la Plataforma Java como el Kit de desarrollo J2SDK 1.5.0 y además de la librería para el manejo de imágenes JAI, ya que ambos proporcionan una gran variedad de funciones para el manejo de imágenes, facilidad de uso además JAVA es un lenguaje orientado a objetos.

El editor utilizado para la construcción del producto es el NetBeans 6.9.1 por ser uno de los mejores IDE's de desarrollo en este lenguaje, además de contar con una interface simple, intuitiva y personalizable y muchas características útiles.

4.9.4. ANÁLISIS Y DISEÑO DEL PROTOTIPO

Con el fin de comprobar la hipótesis planteada en el trabajo se diseña un prototipo basándose en el modelo de proceso evolutivo y análisis, diseño orientado a objeto, utilizando UML como metodología.

4.9.4.1. MODELO DEL PROTOTIPO

Para esta finalidad se hace uso de los diagramas de casos de uso, también identificaremos los actores, su participación con el sistema y el significado de sus roles.

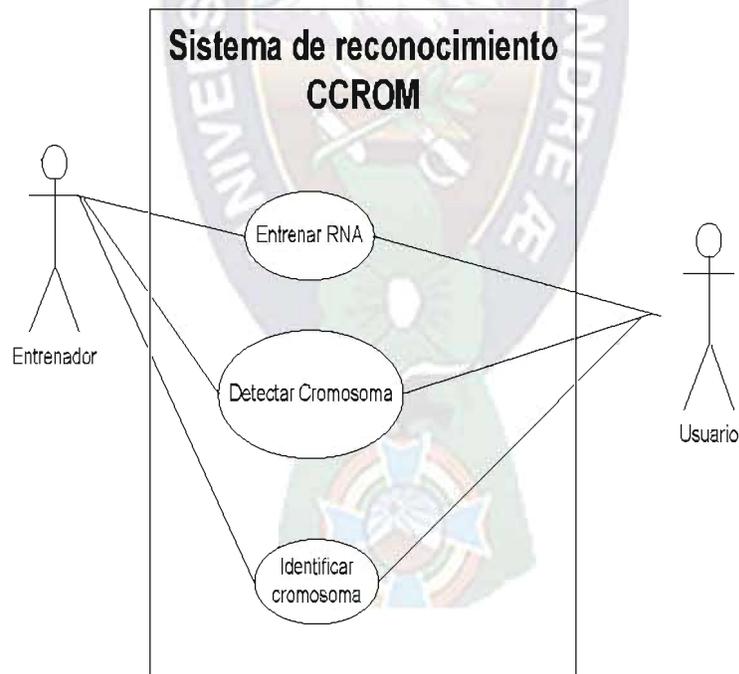


Figura N° 4.8. Diagrama de Casos de uso de Alto Nivel
Fuente: Elaboración según (Booch, Rumbaugh & Jacobson. 1999)

4.9.4.1.1. CASOS DE USO

El diagrama de Casos de Uso del proceso de detección de cromosomas se muestra en la Figura 4.9

Caso de Uso: Entrenar la RNA (Red Neuronal Artificial).

Actor: Entrenador, Imagen Cromosoma.

Propósito: Dar a la RNA la capacidad de aprender a detectar o identificar un cromosoma.

Activación: Cuando el Entrenador hace uso de esta opción ya sea para detección o identificación.

Curso normal

1. el entrenador configura los parámetros de aprendizaje de la red, tanto tasa de entrenamiento, factor momento, error de entrenamiento y el número de neuronal en la capa oculta.
2. El CCROM genera o carga los patrones de entrenamiento según indique el Entrenador
3. El CCROM Calcula el error entre la salida deseada y la obtenida en cada patrón, en función de este error modifica los pesos hasta que el error sea el establecido por el Entrenador.
4. El CCROM guarda los pesos una vez terminado el entrenamiento.

Caso de Uso: Detectar Cromosoma

Actor: Entrenador, Usuario, Imagen Cromosoma.

Propósito: Rastrear al cromosoma en la imagen para establecer los parámetros y tamaño aproximado.

Activación: Cuando el Entrenador hace uso de la opción de detección del cromosoma.

Curso normal

1. El Entrenador o Usuario abre una imagen para ser procesada.
2. El CCROM obtiene una posición Entrenador configura los parámetros de rastreo de la imagen.
3. El CCROM empieza a generar patrones en función de los parámetros establecidos el mismo que es presentado a la RNA que evalúa al patrón y genera una salida.
4. El CCROM despliega a la imagen resaltando la imagen segmentada.

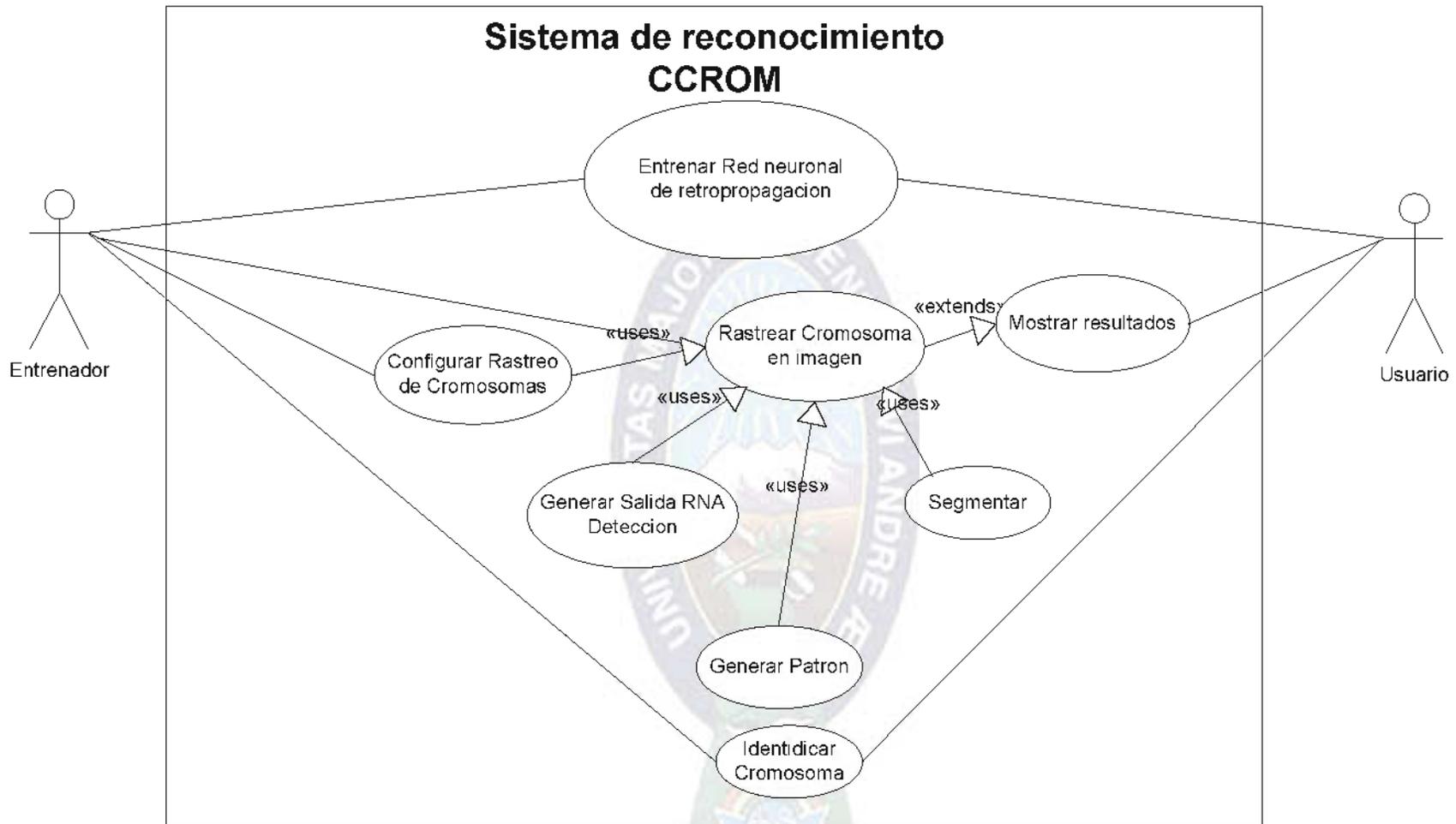


Figura N°4.9. Diagrama de Casos de uso de Bajo Nivel

Fuente: Elaboración según [Booch, Rumbaugh & Jacobson. 1999]

Caso de Uso: Identificar el Cromosoma

Actor: Entrenador, Usuario, Imagen Cromosoma.

Propósito: Identificar al cromosoma en la imagen.

Activación: Cuando el Entrenador hace uso de esta opción de identificación del cromosoma.

Curso normal

1. El entrenador o Usuario abre una imagen para ser identificada.
2. El Entrenador establece las características de la imagen.
3. El CCROM genera un patrón en función de la posición especificada.
4. El CCROM despliega las imágenes de cromosomas y porcentaje de aproximación cuya evaluación por la RNA sea similar.

Caso de Uso: Generar Patrón

Actor: Imagen Cromosoma.

Propósito: Generar un vector característico de una región, que representa la imagen de la forma que esta sea comprensible por la computadora.

Activación: Cuando el entrenador o usuario hace uso de la opción de detección e identificación del cromosoma el mismo hace uso de las opciones en entrenamiento.

Curso Normal

1. El entrenador o usuario abre la imagen para ser procesada.
2. El CCROM obtiene una imagen segmentada de la imagen.
3. El CCROM Construye el histograma de la imagen obtenida en base a tendencias de color especificadas.
4. El CCROM construye el vector patrón buscado en base a los histogramas obtenidos
5. El CCROM guarda y/o retoma el patrón.

4.9.4.1.2. EL MODELO OBJETO – RELACIÓN

Se describen los elementos que son parte del sistema experimental de reconocimiento. Este modelo se muestra en la Figura 4.10.

4.9.4.1.3. MODELO OBJETO – COMPORTAMIENTO

Este modelo nos detalla el comportamiento del sistema y cómo responderá a eventos o estímulos externos.

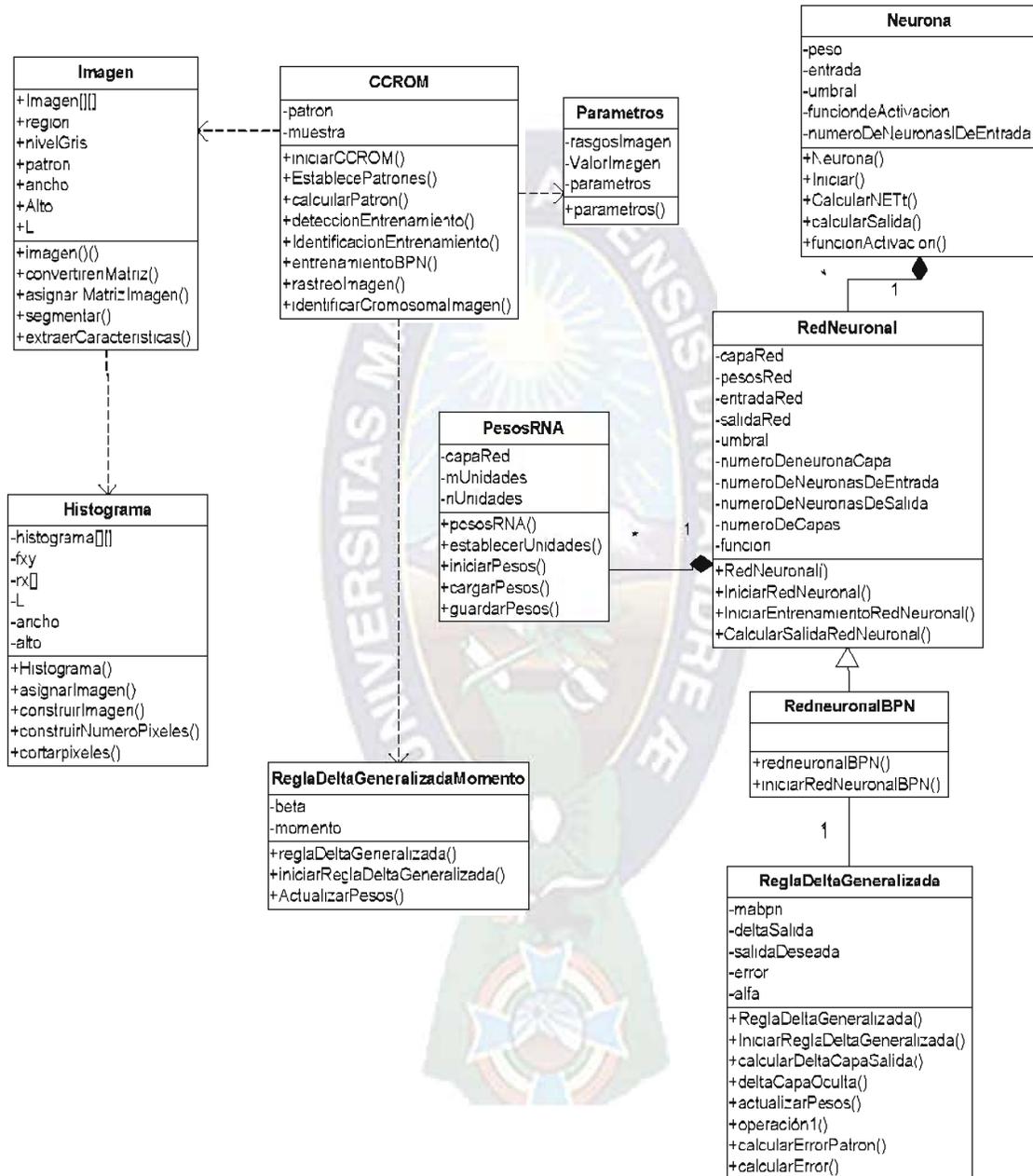


Figura N° 4.10 Diagrama de Clases del Sistema de Sistema de Clasificación de Cromosomas
Fuente: Elaboración según [Booch, Rumbaugh & Jacobson. 1999]

a) DIAGRAMAS DE SECUENCIA

Para modelar el flujo de control por ordenación temporal de cada uno de los casos de uso se utilizar los diagramas de secuencias. Los diagramas de secuencia destacan el paso de mensajes tal y como se desarrolla a lo largo del tiempo lo cual permite modelar el comportamiento dinámico de los casos de uso [Booch, Rumbaugh & Jacobson, 1999].

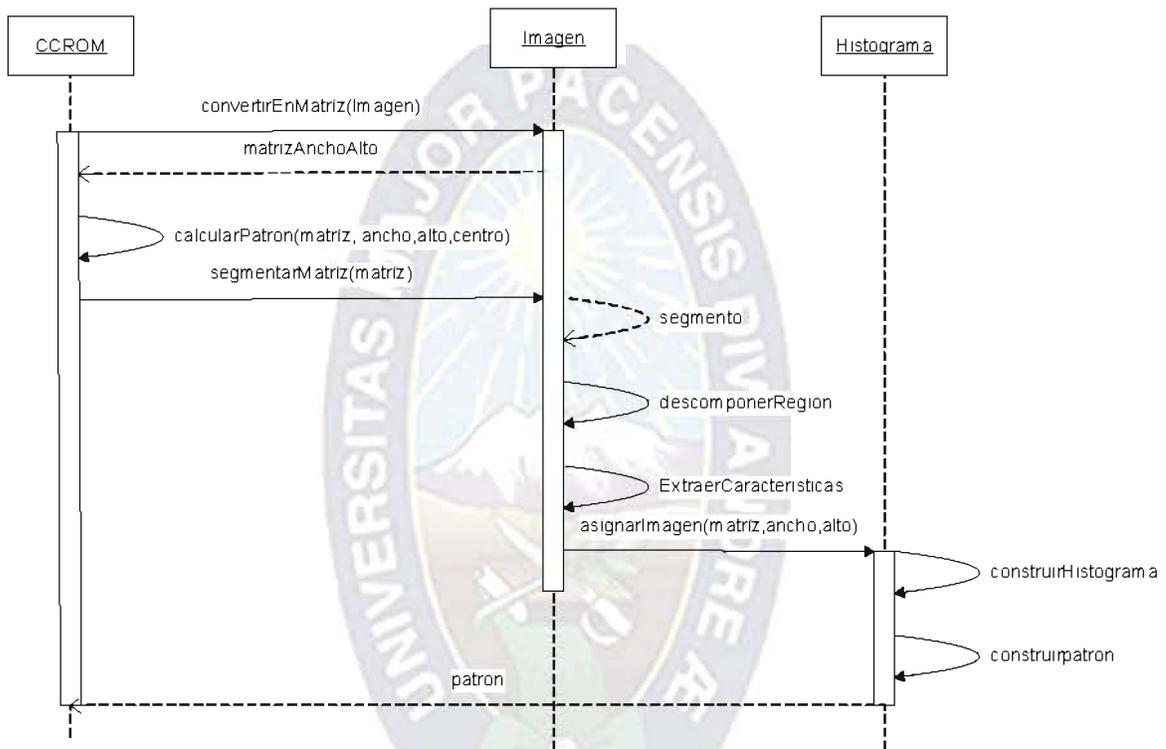


Figura N°4.11. Diagrama de Secuencias del Sistema.
Fuente: Elaboración según [Booch, Rumbaugh & Jacobson, 1999]

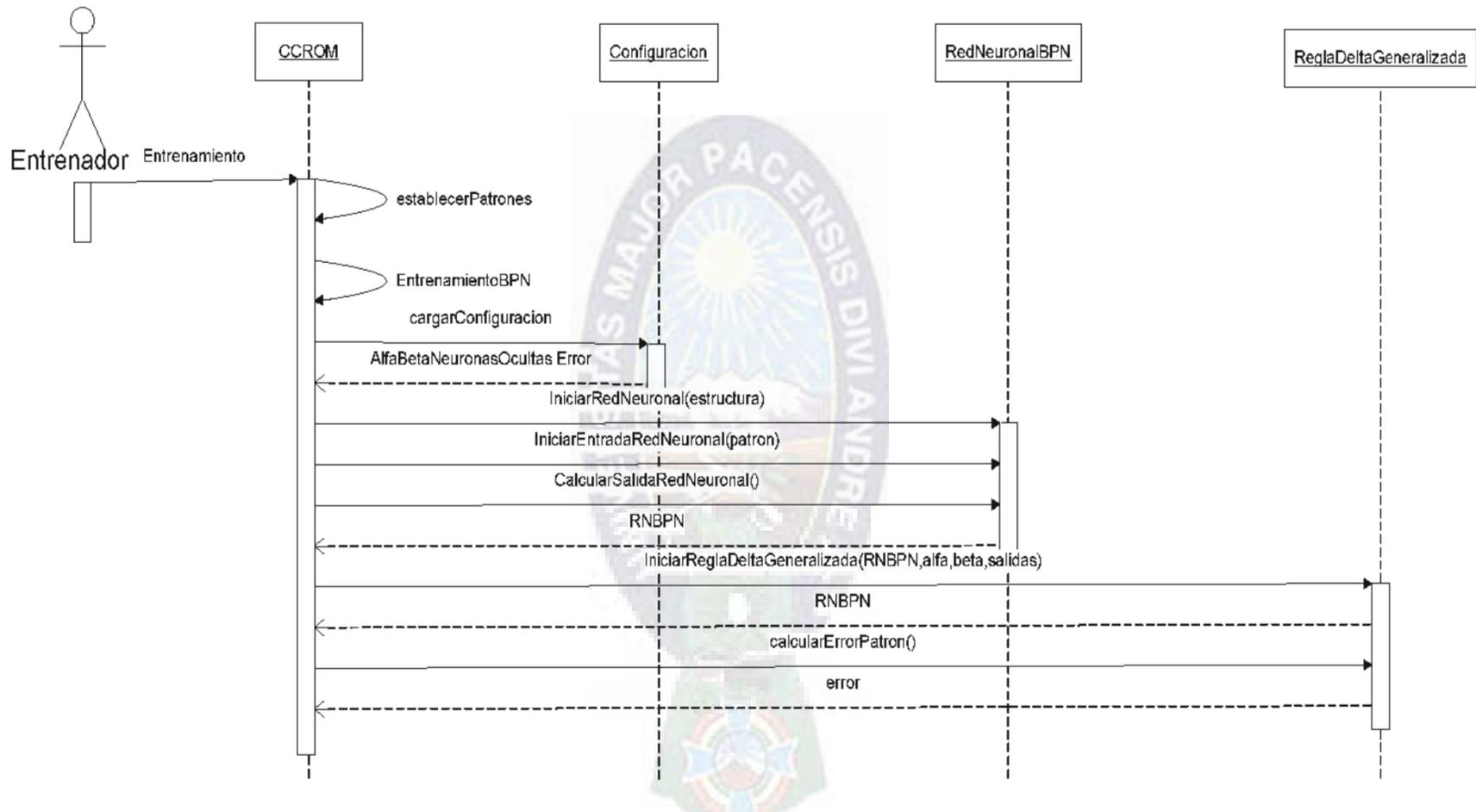


Figura N° 4.12. Diagrama de Secuencias. Caso de uso: Identificación de Cromosoma.
Fuente: Elaboración según [Booch, Rumbaugh & Jacobson. 1999]

b) DIAGRAMA DE ESTADOS

En el modelo de implementación de sistemas orientado a objetos se consideran dos categorizaciones de estado: el estado de cada objeto cuando el sistema ejecuta su función, y el estado del sistema observado desde el exterior cuando se ejecuta su función. Dicho diagrama se describe en la Figura 4.13.

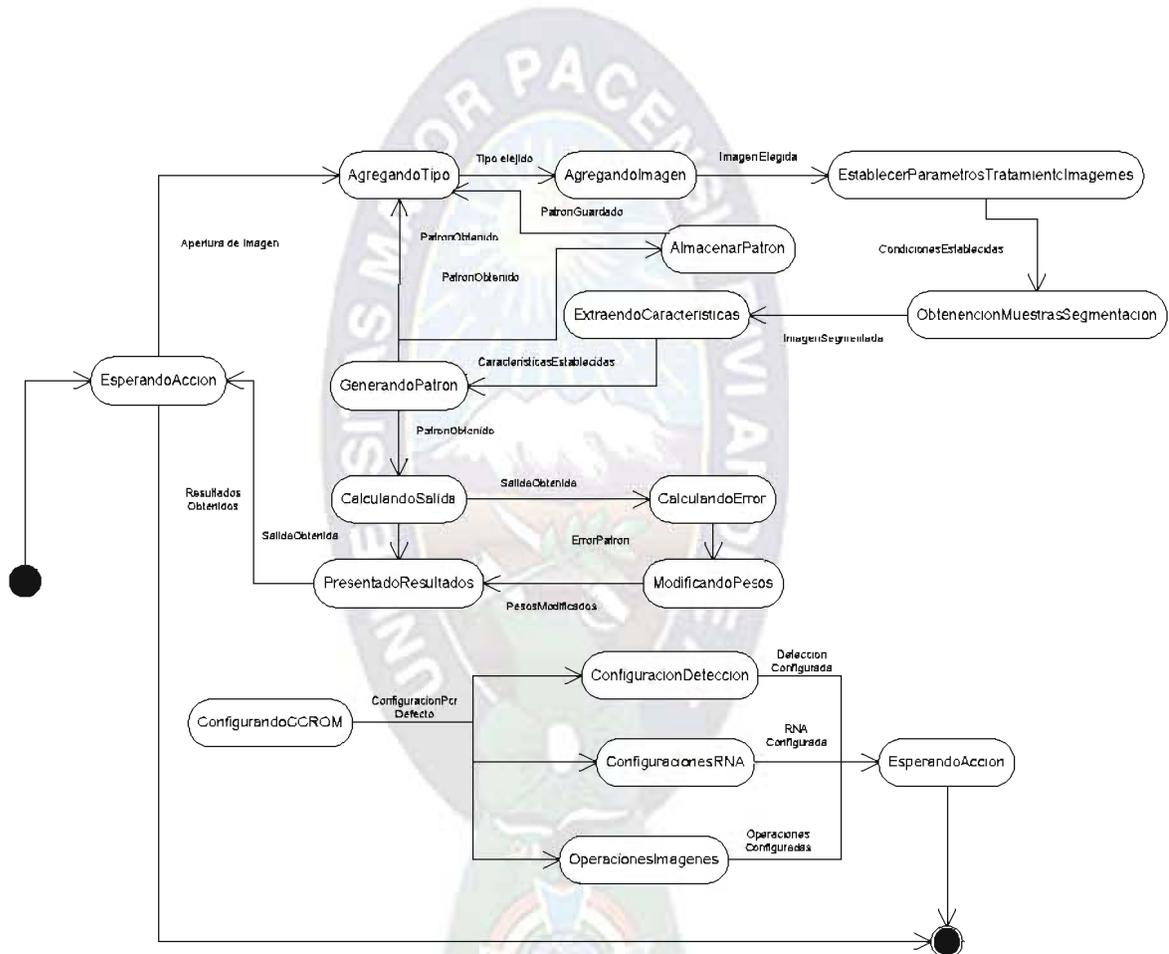


Figura 4.13 Diagrama de Estados del Sistema CCROM
Fuente: Elaboración según [Booch, Rumbaugh & Jacobson, 1999]

4.9.4.2. MODELADO DEL DISEÑO

La metodología orientada a objetos para el diseño detalla diferentes niveles de modularidad. La mayoría de los componentes de un sistema, están organizados en subsistemas un módulo a nivel sistema.

También el diseño orientado a objetos, proporciona de cuatro características importantes para el diseño del software: abstracción, ocultación de información, independencia funcional y modularidad.[Pressman, 2002]

4.9.4.3. DISEÑO DEL SISTEMA

a) Componentes

Los componentes pertenecen al mundo material de los bits y por tanto son un bloque de construcción importante cuando se modelan los aspectos físicos de un sistema [Booch, rumbaugh& Jacobson, 1999]

Las clases definidas en el modelado del análisis, se agrupan en los siguientes componentes:BackProp, CCROM observado en las Figuras 4.14, 4.15.

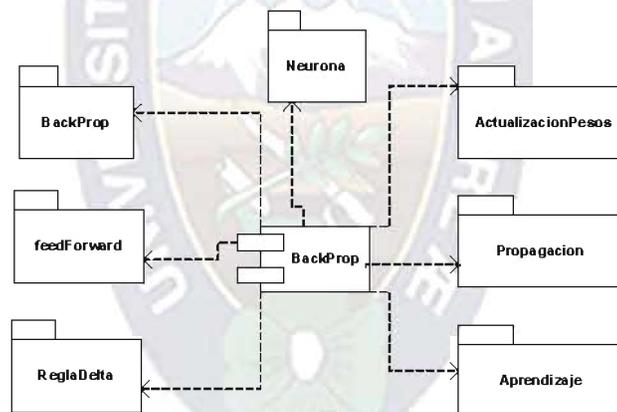


Figura 4.14. Componente Red Neuronal
Fuente: Elaboración según [Booch, Rumbaugh& Jacobson. 1999]

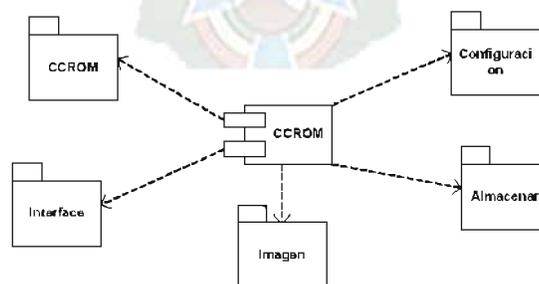


Figura 4.15. Componente CCROM
Fuente: Elaboración según [Booch, Rumbaugh& Jacobson. 1999]

b) Diseño de los objetos

Para el desarrollo del sistema CCROM se tomó en cuenta lo siguientes subsistemas: la detección del cromosoma, y la identificación del mismo, cuyos subsistemas serían los primarios, que además se complementan mediante dos subsistemas adicionales: sistema de entrenamiento de la red neuronal, y sistema de entrenamiento para la detección, y otro sistema para la identificación del cromosoma. En la figura 4.16 se observa los detalles mencionados.

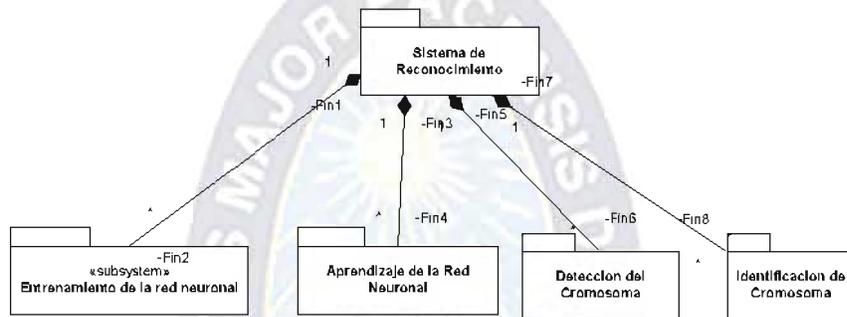


Figura 4.16.: Diagrama de Subsistemas

Fuente: Elaboración según [Booch, Rumbaugh & Jacobson. 1999]

4.9.4.4. DISEÑO DE INTERFACES

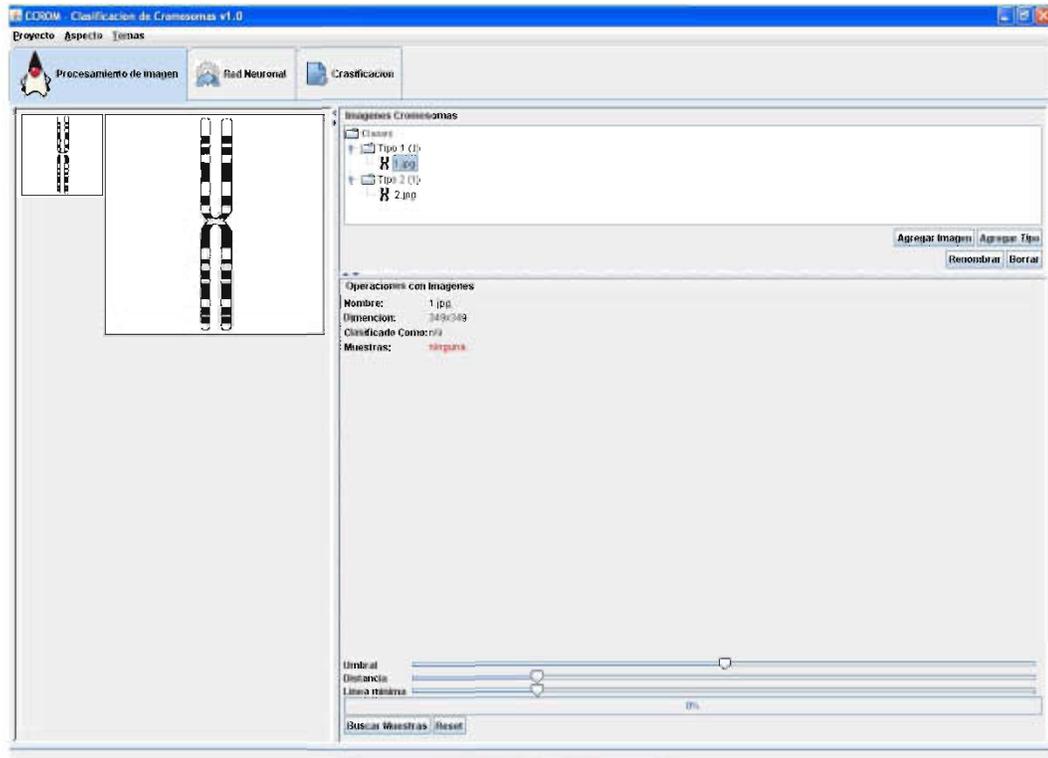
El Sistema de clasificación de cromosomas presenta una interfaz gráfica diseñada en función al análisis y diseño ya presentados para una información detallada sobre el funcionamiento del sistema.

El sistema se compone básicamente de los siguientes elementos en su interfaz de usuario:

- a) Ventana principal.
- b) ventana de altas bajas y modificaciones.
- c) ventana de operaciones con imágenes.
- d) ventana de configuración de red neuronal.
- e) ventana de configuración de la red neuronal.
- f) Ventana de clasificación.

A) VENTANA PRINCIPAL

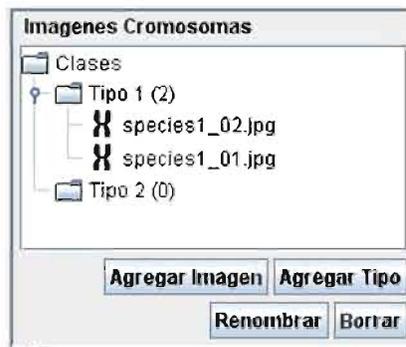
Esta ventana permite al usuario u entrenador la ejecución de todas las funcionalidades con las que cuenta el sistema y por las que fue concebido, desde aquí se ofrece modos de acceso a todos los procesos fundamentales del sistema. Esta se muestra en la Figura 4.17



FiguraNº4.17 Ventana Principal

B) VENTANA DE ALTAS BAJAS Y MODIFICACIONES

En esta sección del sistema se puede agregar, borrar, renombrar las imágenes además también se puede ver las imágenes cargadas en el sistema ordenados en un árbol de clases.



FiguraN°4.18. Ventana de altas, bajas y modificaciones.

C) VENTANA DE OPERACIONES CON IMÁGENES

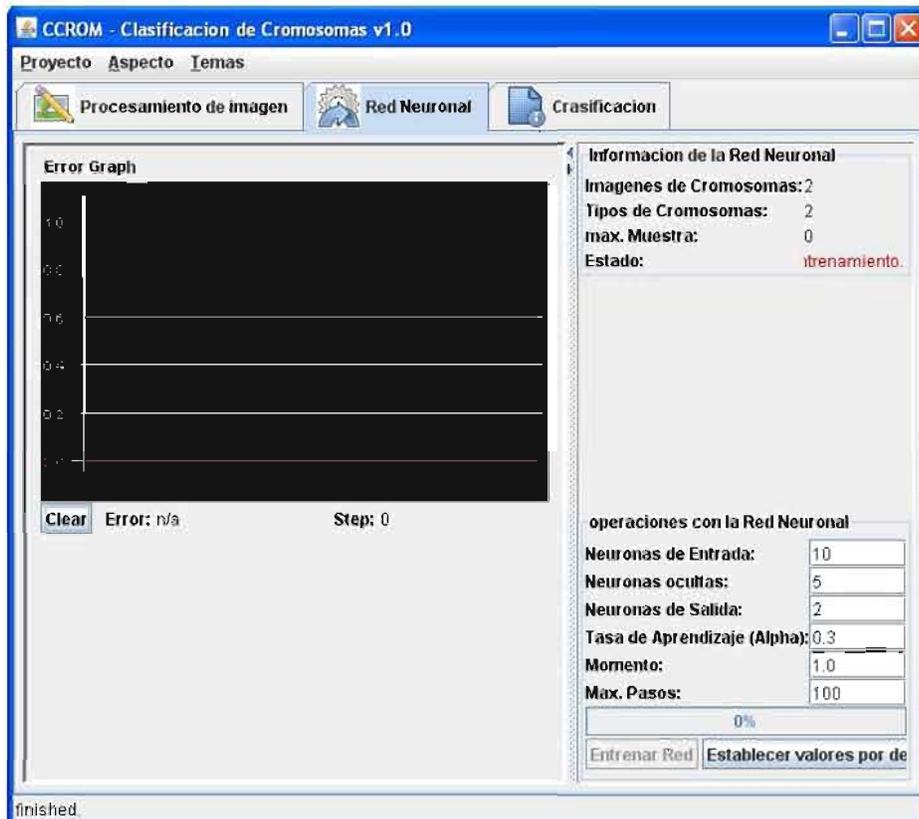
En esta ventana se puede establecer los parámetros para el procesamiento digital de la imagen, antes de ser evaluado por la red neuronal.



FiguraN°4.19. Ventana de operaciones con imágenes

D) VENTANA DE CONFIGURACIÓN DE LA RED NEURONAL

En esta pantalla es donde se establece la configuración del sistema respecto a la red neuronal, los parámetros de detección, identificación y configuración del rastreo del cromosoma. Estos dos paneles se muestran en la figura 4.20.



FiguraN° 4. 20. Ventana de configuración de la Red Neuronal y de entrenamiento

Fuente: Elaboración propia

E) VENTANA DE PARÁMETROS DE LA RED NEURONAL

Permite establecer los parámetros de entrada para el entrenamiento adecuado de la red neuronal y su correcto funcionamiento en el proceso de identificación, e individualización de los cromosomas.

operaciones con la Red Neuronal	
Neuronas de Entrada:	10
Neuronas ocultas:	5
Neuronas de Salida:	2
Tasa de Aprendizaje (Alpha):	0.3
Momento:	1.0
Max. Pasos:	100
0%	
Entrenar Red	Establecer valores por de

FiguraN° 4.20. Ventana de configuración de la red neuronal

Fuente: Elaboración propia

F) VENTANA DE CLASIFICACIÓN

En esta ventana se obtienen los resultados de la clasificación de un cromosoma es la etapa final donde se encuentra el resultado final. Teniendo en cuenta que la red ya se encuentra entrenada para realizar tareas de clasificación.

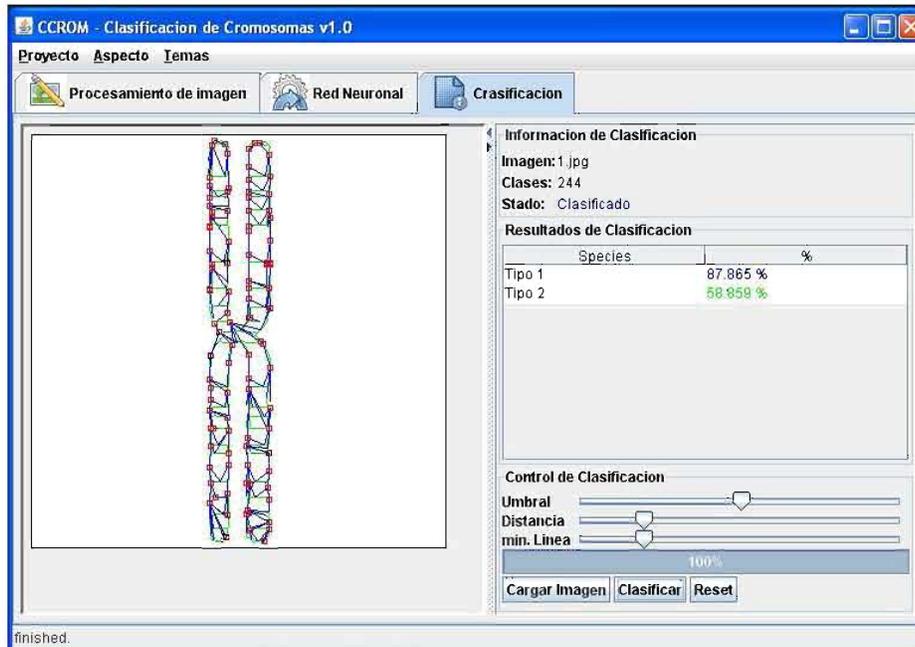


Figura N° 4.21. Ventana de Resultados finales

4.10. MÉTRICAS DE CALIDAD PARA SISTEMAS ORIENTADOS A OBJETOS

Para evaluar la calidad del prototipo se utilizó las métricas orientadas a clases, debido a que la clase es la unidad fundamental de un sistema orientado a objetos, donde se consideran las medidas y métricas de calidad para una clase individual. La jerarquía de clases y las colaboraciones de clases para evaluar la calidad del diseño [Pressman, 2002].

c) Tamaño de la clase

El tamaño general de una clase se puede determinar conociendo el número total de operaciones que se encapsulan dentro de una clase y el número de atributos en la clase. La tabla 4.1.

N	Nombre de clase	Operaciones	Atributos
1	RedNeuronal	4	10
2	Backpropagation	3	0
3	Neurona	5	5
4	CapaRNA	5	4
5	PesosRNA	7	4
6	ReglaDeltaGeneralizada	6	5
7	ReglaDeltaGeneralizadaMomento	6	3
8	Imagen	6	10
9	Histograma	5	8
10	CCROM	11	2
11	Entradas	1	3
	Total	59	54
	Promedio	5,4	4,8

Tabla 4.1 Tamaño de la Clase
Fuente: Elaborado según [Presman, 2002]

d) Factor de acoplamiento FA

El factor de acoplamiento se define de la siguiente manera:

C_j C_i	RedNeuronal	Backpropagation	Neurona	CapaRNA	PesosRNA	ReglaDeltaGeneralizada	ReglaDeltaGeneralizadaMomento	Imagen	Histograma	CCROM	Parámetros
RedNeuronal	0	1	1	1	1	0	0	0	0	1	0
Backpropagation	1	0	1	1	1	1	1	0	0	1	0
Neurona	1	1	0	1	0	0	0	0	0	0	0
CapaRNA	1	1	1	0	1	0	0	0	0	0	0
PesosRNA	1	1	0	1	0	1	1	0	0	0	0
ReglaDeltaGeneralizada	0	1	0	0	1	0	0	0	0	0	1
ReglaDeltaGeneralizadaMomento	0	1	0	0	1	0	0	0	0	0	1
Imagen	0	0	0	0	0	0	0	0	1	1	0
Histograma	0	0	0	0	0	0	0	1	0	0	0
CCROM	1	1	0	0	0	0	0	1	0	0	0
Parámetros	0	0	0	0	0	1	1	0	0	0	0

Tabla 4.2. Relación entre clases
Fuente: Elaborado según [Presman, 2002]

$$FA = \left[\sum_i \sum_j escliente(C_i, C_j) \right] / (TC^2 - TC)$$

Dónde:

$\sum_i \sum_j$: son los límites que van desde $i=1$ y desde $j=1$ hasta TC

Escliente: es igual a 1 si existe una relación entre la clase cliente C_c y la clase servidora C_s y C_c es distinto de C_s . Igual a 0 en cualquier otro caso.

TC : número total de clases del sistema.

Y calculamos con los valores obtenidos de la Tabla 3.11

$$FA = \frac{38}{(11^2 - 11)} = 0.34$$

Este resultado indica que la compresión, mantenimiento y reutilización del sistema tiene una complejidad del 34%

4.11. JUSTIFICACIÓN DE MÉTODOS

4.11.1. Redes Neuronales Artificiales

Para el desarrollo del sistema CCROM se trabajó con redes neuronales artificiales de retropropagación debido a las ventajas ya mencionadas en el Capítulo 2:

1. Aprendizaje Adaptativo. Capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.
2. Autoorganización: Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
3. Tolerancia a Fallos. La destrucción parcial de una red conduce a una degradación de su estructura sin embargo, algunas capacidades de la red neuronal pueden aprender a reconocer patrones con ruido, distorsionado incompletos.

4.11.2 Lenguaje de programación Java

La construcción del Sistema se realizó en el lenguaje de programación JAVA por las siguientes razones que se destacan a continuación:

1. **Orientado a Objetos.** Soporta las tres características propias del paradigma orientado a objetos: encapsulación, herencia y polimorfismo.
2. **Robusto.** El lenguaje Java realiza verificaciones en busca de errores tanto en tiempo de compilación como en tiempo de ejecución.
3. **Portable.** El lenguaje Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, PC o MAC.
4. **Dinámico.** El Lenguaje Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no implementa conectar todos los módulos que comprenden una aplicación, hasta el mismo tiempo de ejecución.
5. **Seguro.** El Código Java pasa muchos test de ejecutarse en la máquina. El código pasa a través de un verificador de ByteCode que comprueba el formato de los fragmentos de código que aplica un probador de teoremas para detectar fragmentos de código ilegal.



CAPITULO V

RESULTADOS PRUEBA EXPERIMENTAL

RESUMEN

En el presente capítulo se detalla los resultados obtenidos en la etapa de aprendizaje y de reconocimiento de la red neuronal, además de establecer los parámetros de entrada adecuados para el correcto funcionamiento de la red neuronal

5.1. INTRODUCCION

Con el propósito de probar la hipótesis planteada en el presente trabajo se comienza con la recopilación de la información de la imagen y la obtención, posteriormente se verifica si la imagen presenta información suficiente para el reconocimiento del cromosoma, en caso contrario se aplica el mejoramiento de la imagen, finalmente si la imagen es adecuada se pasa a la red neuronal. Este esquema podemos visualizarlo en la Figura 5.1.

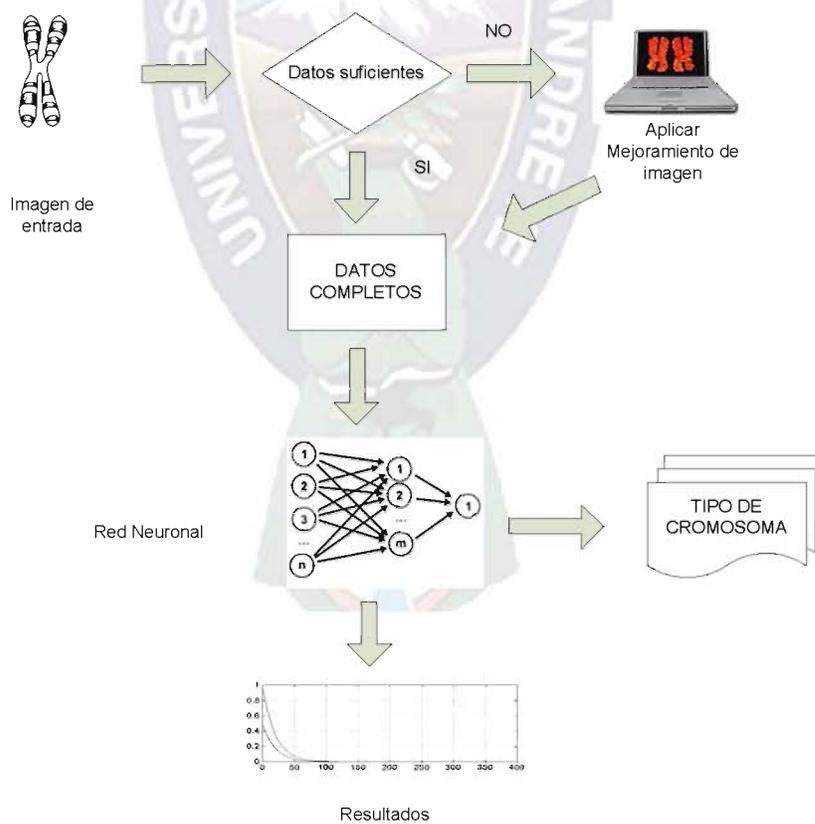


Figura 5.1. Esquema de las fases de reconocimiento de cromosomas
Fuente: Elaboración propia

Previamente debemos evaluar el comportamiento del aprendizaje de la red neuronal, que posteriormente será el factor determinante para la obtención de resultados óptimos.

5.2. DATOS PARA EL ENTRENAMIENTO

Se muestrean cinco repeticiones con inicializaciones de pesos aleatorias, de donde se obtienen los resultados de las tasas medias de reconocimiento de cromosomas humanos y de sus respectivos promedios de los errores cuadráticos medios, tomados del conjunto de prueba. Se presentan los resultados, que permitieron obtener las mejores arquitecturas para la red neuronal, de cada uno de los cinco casos estudiados, con respecto a la resolución de la imagen de entrada, y la cantidad de neuronas ocultas. Con esto se concluye las cinco mejores arquitecturas para la red neuronal, para los cinco diferentes tamaños de imágenes de entrada. Con esta información se obtiene la mejor arquitectura de red neuronal que permite en forma eficaz y eficiente el reconocimiento de cromosomas humanos, para la primera partición, con máxima tasa media de reconocimiento y mínimo promedio de los errores cuadráticos medios.

Se presentan los resultados obtenidos en la etapa de entrenamiento, aprendizaje y reconocimiento, también se muestran los resultados de las cinco repeticiones consecutivas de aprendizajes, con inicialización de pesos aleatorios, para las tres mejores arquitecturas de la red neuronal determinadas para este propósito.

Para este fin se tomaron 110 imágenes que serán para la etapa previa de entrenamiento, estas muestras se obtuvieron en el Instituto de Genética de la UMSA y algunas otras de la web. Debido a que la información y el acceso a esta información es restringida en nuestro medio se tuvo que optar por obtener información de la web para propósitos de prueba del sistema. Las siguientes figuras muestran los patrones de aprendizaje del sistema.

Tabla 5.1. Conjunto de muestras del tipo 1 para entrenamiento
Fuente: Instituto de genética UMSA

Tabla 5.2. Conjunto de muestras del tipo 2 para entrenamiento
Fuente: Instituto de genética UMSA

Tabla 5.3. Conjunto de muestras del tipo 3 para entrenamiento
Fuente: Instituto de genética UMSA

La figura 5.2 muestra los resultados del aprendizaje de la red neuronal de retropropagación y sus factores de entrada, correspondientes a la entrada de la red neuronal.

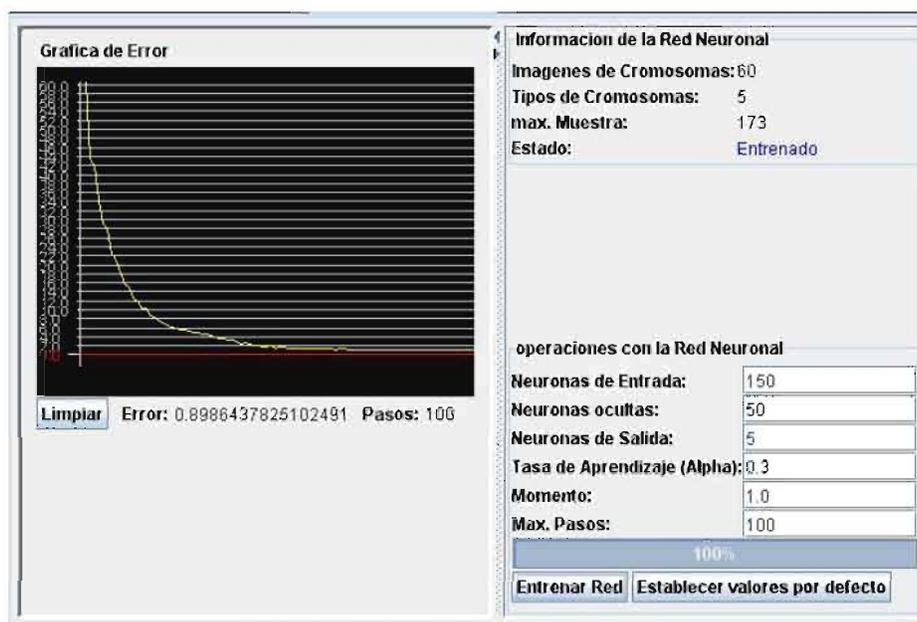


Figura Nro. 5.2. Parámetros de entrada a la Red Neuronal y error.

Fuente: Elaboración propia

La Tabla 5.4 muestra los resultados obtenidos en proceso de aprendizaje, que además se obtuvo los valores medios como valores deseables en la entrada de la red neuronal para un óptimo proceso de aprendizaje.

Nro. Imágenes	Neuronas de Salida	Neuronas de Entrada	Neuronas Ocultas	Tasa de Aprendizaje	Pasos	Error
60	5	150	50	0.3	100	0.898
60	5	150	50	0.3	100	1.93
60	5	173	50	0.3	100	0.99
60	5	174	50	0.3	100	0.93
60	5	180	50	0.3	100	1.01
60	5	173	60	0.3	100	0.98
60	5	173	70	0.3	100	0.87
60	5	173	80	0.3	100	0.91
60	5	173	90	0.3	100	1.21
60	5	150	50	0.3	100	0.87
60	5	150	50	0.4	100	0.97
60	5	150	50	0.5	100	0.9
60	5	150	50	0.6	100	1.06
60	5	150	50	0.3	100	0.93
60	5	150	50	0.3	110	0.87
60	5	150	50	0.3	120	1.93
60	5	150	50	0.3	130	0.83
60	5	150	50	0.3	130	0.91
Valores Medios		155	50	0.35	110	0.98

Tabla 5.4. Resultados de entrenamiento de la RNA.
Obtención de los valores medios de entrada a la red neuronal

Con las pruebas descritas en esta sección, se puede concluir los cinco mejores parámetros para el funcionamiento adecuado de la red neuronal de retropropagación con cinco tipos de cromosomas permite reconocer cromosomas, obteniéndose el desempeño del sistema con los cinco diferentes parámetros de entrada en las imágenes utilizadas.

5.3. PRUEBAS

Cuando la etapa de entrenamiento ha concluido se realiza el análisis del comportamiento de la red con valores de prueba y después con valores que sean ajenos a los valores de prueba, para analizar el comportamiento de la red neuronal tomaremos dos casos posibles:

Caso 1. Cuando la imagen de entrada fue tratada en la etapa de aprendizaje.

Caso 2. Cuando la imagen de entrada no fue tratada en la etapa de aprendizaje.

Estas posibles situaciones nos permitirán demostrar la correcta funcionalidad de la red neuronal, y por ello también se demostrara la hipótesis planteada en el Capítulo I.

Para las pruebas tenemos 110 muestras de imágenes de cromosomas 50 obtenidas del instituto de genética de la UMSA y las restantes obtenidas de la WEB. Ya que la información es bastante limitada para estos propósitos.

Caso 1.

Cuando la imagen de entrada fue tratada en la etapa de aprendizaje.

En las tablas 5.5, 5.6 y 5.7 en la segunda columna se presenta la imagen de entrada puesta a prueba, la tercera columna muestra la segmentación además de la extracción de características de la imagen no cromática del cromosoma puesto a prueba y la cuarta columna presenta el resultado en porcentaje de aproximación al tipo de cromosoma al que pertenece la muestra en evaluación.

#	Imagen de entrada	Imagen segmentada	Resultados de la RNA	Resultados del especialista
1			Tipo 1 98.484 % Tipo 2 80.159 % Tipo 3 61.339 % Tipo 4 41.480 % Tipo 5 21.494 %	 Cromosoma tipo 1
2			Tipo 1 98.022 % Tipo 2 80.416 % Tipo 3 60.949 % Tipo 4 41.717 % Tipo 5 21.758 %	 Cromosoma tipo 1
3			Tipo 1 99.211 % Tipo 2 80.454 % Tipo 3 60.508 % Tipo 4 40.679 % Tipo 5 20.780 %	 Cromosoma tipo 1
4			Tipo 1 99.497 % Tipo 2 80.150 % Tipo 3 60.431 % Tipo 4 40.460 % Tipo 5 20.498 %	 Cromosoma tipo 1
5			Tipo 1 98.838 % Tipo 2 79.492 % Tipo 3 59.584 % Tipo 4 40.234 % Tipo 5 21.150 %	 Cromosoma tipo 1
6			Tipo 1 98.532 % Tipo 2 80.326 % Tipo 3 61.065 % Tipo 4 41.405 % Tipo 5 21.454 %	 Cromosoma tipo 1
7			Tipo 1 98.755 % Tipo 2 79.732 % Tipo 3 60.274 % Tipo 4 41.186 % Tipo 5 21.237 %	 Cromosoma tipo 1

Tabla 5.5. Resultados de Reconocimiento de cromosomas de tipo 1

Fuente: Elaboración propia

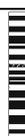
#	Imagen de entrada	Imagen segmentada	Resultados de la RNA	Resultados del especialista
1			Tipo 1 79.308 % Tipo 2 98.382 % Tipo 3 79.738 % Tipo 4 60.262 % Tipo 5 40.565 %	 Cromosoma tipo 2
2			Tipo 1 79.673 % Tipo 2 98.659 % Tipo 3 79.379 % Tipo 4 60.051 % Tipo 5 40.254 %	 Cromosoma tipo 2
3			Tipo 1 80.770 % Tipo 2 96.998 % Tipo 3 78.466 % Tipo 4 58.945 % Tipo 5 39.147 %	 Cromosoma tipo 2
4			Tipo 1 79.340 % Tipo 2 97.622 % Tipo 3 78.598 % Tipo 4 59.254 % Tipo 5 39.965 %	 Cromosoma tipo 2
5			Tipo 1 79.093 % Tipo 2 99.032 % Tipo 3 79.822 % Tipo 4 60.364 % Tipo 5 40.883 %	 Cromosoma tipo 2
6			Tipo 1 79.791 % Tipo 2 98.552 % Tipo 3 79.401 % Tipo 4 60.020 % Tipo 5 40.164 %	 Cromosoma tipo 2
7			Tipo 1 80.000 % Tipo 2 98.382 % Tipo 3 79.575 % Tipo 4 59.646 % Tipo 5 39.991 %	 Cromosoma tipo 2

Tabla 5.6. Resultados de Reconocimiento de cromosomas de tipo 2

Fuente: Elaboración propia

#	Imagen de entrada	Imagen segmentada	Resultados de la RNA	Resultados del especialista
1			Tipo 1 60.328 % Tipo 2 80.236 % Tipo 3 97.574 % Tipo 4 78.476 % Tipo 5 59.654 %	 Cromosoma tipo 3
2			Tipo 1 60.672 % Tipo 2 78.857 % Tipo 3 97.755 % Tipo 4 79.136 % Tipo 5 59.251 %	 Cromosoma tipo 3
3			Tipo 1 59.243 % Tipo 2 79.123 % Tipo 3 97.837 % Tipo 4 78.749 % Tipo 5 60.089 %	 Cromosoma tipo 3
4			Tipo 1 59.090 % Tipo 2 79.075 % Tipo 3 98.707 % Tipo 4 79.828 % Tipo 5 60.867 %	 Cromosoma tipo 3
5			Tipo 1 59.829 % Tipo 2 79.092 % Tipo 3 98.113 % Tipo 4 79.579 % Tipo 5 60.128 %	 Cromosoma tipo 3
6			Tipo 1 59.727 % Tipo 2 79.618 % Tipo 3 98.415 % Tipo 4 79.681 % Tipo 5 60.207 %	 Cromosoma tipo 3
7			Tipo 1 70.782 % Tipo 2 88.695 % Tipo 3 88.686 % Tipo 4 69.053 % Tipo 5 49.172 %	 Cromosoma tipo 3

Tabla 5.7. **Resultados de Reconocimiento de cromosomas de tipo 3**

Fuente: Elaboración propia

Caso 2. Se evalúa el sistema con imágenes nuevas para el sistema

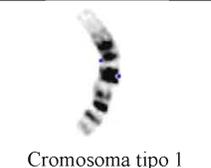
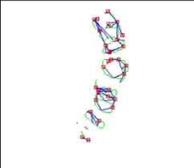
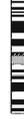
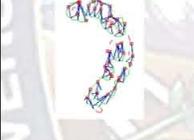
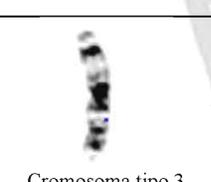
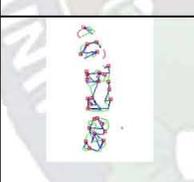
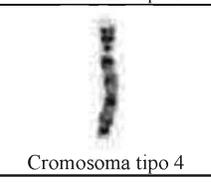
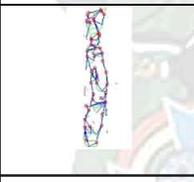
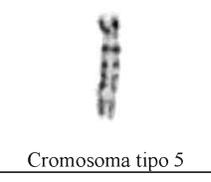
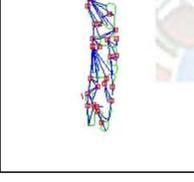
Nro de Imagen	Imagen de entrada	Imagen segmentada	Resultados de la RNA	Resultados del especialista
1	 Cromosoma tipo 1		Tipo 1 95.446 % Tipo 2 76.323 % Tipo 3 58.297 % Tipo 4 42.494 % Tipo 5 24.505 %	 Cromosoma tipo 1
2	 Cromosoma tipo 1		Tipo 1 97.715 % Tipo 2 80.030 % Tipo 3 61.026 % Tipo 4 41.467 % Tipo 5 22.273 %	 Cromosoma tipo 1
3	 Cromosoma tipo 1		Tipo 1 99.338 % Tipo 2 80.144 % Tipo 3 60.296 % Tipo 4 40.352 % Tipo 5 20.394 %	 Cromosoma tipo 1
4	 Cromosoma tipo 1		Tipo 1 91.999 % Tipo 2 87.476 % Tipo 3 67.816 % Tipo 4 47.865 % Tipo 5 27.880 %	 Cromosoma tipo 1
5	 Cromosoma tipo 2		Tipo 1 77.529 % Tipo 2 97.071 % Tipo 3 81.662 % Tipo 4 61.837 % Tipo 5 42.464 %	 Cromosoma tipo 2
6	 Cromosoma tipo 3		Tipo 1 61.181 % Tipo 2 73.369 % Tipo 3 81.923 % Tipo 4 78.017 % Tipo 5 58.776 %	 Cromosoma tipo 3
7	 Cromosoma tipo 4		Tipo 1 66.656 % Tipo 2 86.126 % Tipo 3 87.325 % Tipo 4 70.359 % Tipo 5 53.335 %	No pudo se pudo identificar
8	 Cromosoma tipo 5		Tipo 1 31.601 % Tipo 2 51.579 % Tipo 3 71.563 % Tipo 4 83.824 % Tipo 5 88.382 %	 Cromosoma tipo 5

Tabla 5.8. Resultados de Reconocimiento de cromosomas Para el caso 2

Fuente: elaboración propia

En la primera columna se muestra la imagen de entrada, en la tercera columna la imagen segmentada y en la cuarta los resultados que se obtiene del sistema.

5.4. EVALUACIÓN DE RESULTADOS

En la tabla 5.8 se observa que los resultados del sistema, se acerca a los resultados obtenidos por el especialista en citogenética obteniendo una tasa de error del 6% y una precisión del 94 %. Observemos que en el caso 7 el especialista no pudo identificar el cromosoma tampoco así el sistema, por la baja información que ofrece esta muestra.

Además se calcula el error que producen las salidas generadas por el modelo:

$$E_p = \frac{1}{2} \sum_k (d - y)^2$$

Donde d representa la salida deseada y y es la salida obtenida, entonces se calcula $(d - y)^2$ para cada una de las salidas obtenidas, como se observa en el cuadro 5.9.

Luego se obtiene la sumatoria de $(d - y)^2$ del total de los datos, entonces obtenemos:

Tipo de Cromosoma	Numero de imágenes de entrenamiento	Numero de imágenes de prueba	Porcentaje de clasificación	$(d - y)$	$(d - y)^2$
Tipo 1	12	25	80 %	0,2	0,04
Tipo 2	12	25	88 %	0,12	0,014
Tipo 3	12	25	84 %	0,16	0,026
Tipo 4	12	25	87 %	0,13	0,017
Tipo 5	12	25	90 %	0,1	0,01
TOTAL					0,107

Tabla 5.9. Tabla de porcentajes de clasificación

Fuente: Elaboración propia

$$\sum_k (d - y)^2 = 0.107$$

Finalmente obtenemos

$$E_p = \frac{1}{2} \sum_k (d - y)^2 = 0,054$$

$$Precisión = (1 - E_p) * 100\% = 94.6\%$$

Para esta estimación también tomamos en cuenta las pruebas que se muestran en la tabla del Anexo D.



CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

RESUMEN

En el presente capítulo se establecen las conclusiones finales que se obtuvieron respecto a la investigación realizada, además se detalla los objetivos que fueron satisfechos. También se evaluara la hipótesis planteada, y finalmente se concluye con las recomendaciones para esta investigación.

6.1. CONCLUSIONES GENERALES

Las redes neuronales han demostrado ser versátiles en el reconocimiento de patrones, específicamente en el reconocimiento de patrones debido a su capacidad de aprendizaje originada en la analogía de la estructura humana y su fundamento matemático. Además que la construcción de patrones es fundamental e este trabajo para el proceso de reconocimiento de cromosomas.

Debido a que las investigaciones de reconocimiento de cromosomas humanos y de cualquier otro tipo de formas y figuras no se basan en imágenes cromáticas, no se puede hacer una caracterización de las imágenes donde la pérdida de información es notoria.

Estas razones que fueron descritas en el párrafo anterior, nos llevó a desarrollar en esta tesis un sistema experimental de reconocimiento, donde el color de la imagen, las redes neuronales artificiales, especialmente la red de retropropagación y el tratamiento de digital imágenes fueron preponderantes en el desarrollo del trabajo. Para probar el funcionamiento de lo planteado se desarrolló un prototipo computarizado, empleando para ello ingeniería del software orientado a objetos con notación UML y para la implantación se utilizó el lenguaje de programación Java. Así también se realizó la evaluación de la misma a través de métricas de calidad, concluyendo que cuando se trabaja solamente con imágenes mono cromáticas las características importantes en los cromosomas se mantienen.

6.2. CUMPLIMIENTO DE OBJETIVOS

Como podemos apreciar en el capítulo 3, estudiamos los métodos empleado para cumplir con los objetivos planteados.

- Mostrar la aplicabilidad de las redes neuronales al reconocimiento de cromosomas humanos.
- Detectar un cromosoma en una imagen y la obtención de muestras según el algoritmo de Prewit.

Tomando como base los procedimientos propios del tratamiento digital de imágenes y redes neuronales explicadas en el capítulo 2 se derivan los métodos mencionados y la forma de aprendizaje.

Del mismo modo en el capítulo 3 con la construcción de la red neuronal específica para este problema. Reconocimiento de cromosomas, se logra la construcción del reconocedor de cromosomas para el cumplimiento de los objetivos:

- Mostrar los cromosomas clasificados, siendo estos 46 cromosomas humanos.
- Proporcionar la información de los bordes de la imagen y las coordenadas de ubicación.
- Determinar el tamaño aproximado de la imagen

La identificación además se realiza en base a la obtención de muestras que logramos obtener aplicando el algoritmo de detección de bordes Prewit. Que se establecen como la salida en la respectiva red neuronal.

El proceso de segmentación de la imagen y la obtención de muestras la podemos encontrar en el capítulo 3 todo este previo procedimiento al entrenamiento de la red neuronal, es para el cumplimiento al objetivo:

- Mejorar la calidad de presentación de la información sometiendo las imágenes fotográficas a un tratamiento adecuado.

Se desarrolló un prototipo para probar el funcionamiento del método de reconocimiento planteado, empleando para ello la ingeniería de software orientado a objetos con la notación UML y para la implementación se utilizó el lenguaje Java.

El modelado del análisis detallado en el capítulo 4 presenta los siguientes modelos

- ✓ Casos de uso
- ✓ Modelo objeto-relación
- ✓ Modelo objeto-comportamiento

Los métodos desarrollados para la fase de diseño fueron:

- ✓ Diseño de sistema
- ✓ Diseño de objetos
- ✓ Diseño de Interfaces
- ✓ Diseño Procedimental

Con lo cual se construye el prototipo experimental para cumplir los siguientes objetivos:

- Mostrar los cromosomas clasificados.
- Agilizar el estudio de los cromosomas reduciendo el tiempo que el especialista tarda en realizar la clasificación.

La extracción de muestras presentada en el capítulo 2, 3 y aplicada en el capítulo 4 sobre imágenes en base a histogramas, permiten el cumplimiento del objetivo:

- Mejorar la calidad de presentación de la información sometiendo las imágenes fotográficas a un tratamiento adecuado.

El capítulo 2 y 3 presenta las bases para la construcción y entrenamiento de las redes neuronales los mismos que son aplicados en el capítulo 4 para la construcción de los métodos y prototipo, de tal forma se dan el cumplimiento de los siguientes objetivos:

- Mostrar la aplicabilidad de las redes neuronales al reconocimiento de cromosomas humanos.

Finalmente tras la implementación, prueba del prototipo y la evaluación estadística de los resultados se dio cumplimiento a los restantes objetivos.

- Reducir el tiempo de espera del paciente en la obtención de los resultados, aparte de los 10 días que tarda el cultivado de las muestras, se obtendrá los resultados en máximo un día.
- Brindar resultados más confiables.

6.3 ESTADO DE LA HIPÓTESIS

De acuerdo a las conclusiones establecidas anteriormente, las cuales nos permiten evaluar la hipótesis planteada en el capítulo 1 se concluye que:

Efectivamente es posible el reconocimiento de cromosomas humanos en base a redes neuronales artificiales utilizando para ello fotografías digitales, clasificando los cromosomas humanos en base a las características que obtenemos con la metodología de bandeamiento G.

6.4 RECOMENDACIONES

- a) Para propósitos de prueba del prototipo se tomó 12 imágenes por cromosoma para el entrenamiento en formato GIF se recomienda hacer uso de imágenes de otro tipo de formato y de dimensionen mayores dado que el tiempo de proceso en imágenes de 200 x 400 pixeles no resulta costoso en tiempo de proceso se recomienda imágenes de dimensiones mayores al 50 % de las empleadas en este estudio.
- b) Se recomienda establecer un mejor conjunto de pesos para el inicio del entrenamiento en cada red, dado que el generar pesos aleatorios conlleva a que la red neuronal pueda ser lejana a un máximo global.
- c) Ver la posibilidad de aplicar filtros y máscaras a las imágenes, previo al proceso de extracción de características, los mismos que puedan realzar los rasgos cromáticos en el cromosoma.
- d) Reestructurar la salida de la red neuronal en base a imágenes y información de los rasgos de coincidencia de patrones.
- e) El conocimiento previo adquirido por las Redes Neuronales Artificiales en base al proceso de aprendizaje, proporciona un conjunto inicial de pesos que para una nueva muestra de entrenamiento acelerara el proceso, se recomienda la implementación de tal situación.

6.5. TRABAJOS FUTUROS

- a) Aplicar la metodología de reconocimiento a imágenes en tercera dimensión. Para así poder realizar el reconocimiento en tiempo real y en video.
- b) Mejorar el tiempo de entrenamiento de las Redes neuronales, es posible aplicar algoritmos genéticos a fin de reducir el tiempo de entrenamiento y obtener el conjunto de pesos cercano a la generación de salidas alrededor de un máximo global , no así un máximo local.
- c) Contemplar el uso de redes neuronales mediante la aplicación de lógica difusa, es decir hacer uso de Redes NeuroDifusas en la solución del problema de reconocimiento de cromosomas.





BIBLIOGRAFÍA

REDES NEURONALES

[CHOQUE, 2002]

G. Choque Aspiazu "INTELIGENCIA ARTIFICIAL", 2002

Publicación de la carrera de Informática, UMSA

[IEEE, 2005]

IEEE TRANSACTIONS ON IMAGE PROCESSING www.ieee.org/sp/tip

A publication of the IEEE signal processing, Octubre 2005, volume 14. (ISSN 1057-7149)

[CACERES, 2002]

Cáceres, Francisco (2002) "Bases Biológicas y Teóricas de las Redes Neuronales Artificiales". Proyecto de Licenciatura de Ingeniería Civil Electrónica. Facultad de Ingeniería.

Universidad Mayor. Santiago. Chile.

TRATAMIENTO DIGITAL DE IMAGENES

[GONZALES&WOODS, 2002]

Rafael C. Gonzales, Richar E. Woods "Digital image processing", 2002

Editorial Prentice Hall Inc. Second edition.

[JENSEN, 1996]

Jensen, J. R., "Introductory Digital ImageProcessing"

Remote Sensing Perspective, Prentice Hall, 1996.

[Gonzales, Woods, 1996]

González, Rafael C. y Woods, Richard E. (1996) "Tratamiento Digital de Imágenes".

Addison-

Wesley Iberoamericana. S.A.

[AJENJO, 1993]

Ajenjo, Alberto D. (1993) "Tratamiento Digital de Imágenes". Ediciones Anaya Multimedia.

S.A. Madrid. España.

[KENNEY, RIEGER, 2003].

Anne R. Kenney, Bibliotecaria Auxiliar de la Universidad, Oya Y. Rieger, "Tutorial de digitalización de imágenes"

Biblioteca de la Universidad de Cornell / Departamento de Investigación

Richard Entlich, Bibliotecario de Proyectos Digitales

[PRATT, 2001]

William K.Pratt "Digital image processing", 2001

Editorial John Wiley & Sons, Inc. Thirdedition.

CITOGENETICA Y OTROS

[GUIMARÁES&OTTO,1998]

Priscila Guimarães Otto, Paulo Alberto Otto, Osvaldo Frota-Pessoa

"Genética humana e clínica",1998

Editorial Roca, 1ra edicionpag 3-30.

[LAHOZ, 2000]

Rafael Lahoz – Beltrán "BIOINFORMÁTICA", 2000

Editorial PEARSON

[MENDOZA&ROMO, 2004]

Víctor M. Mendoza, Ma. Rebeca Romo Pinales, Martha A. SánchezRodríguez, Ma. Silvia Hernández Zabala "INVESTIGACIÓN – Introducción a la metodología",2004

Universidad nacional autónoma de México, Facultad de Estudios Superiores Zaragoza.

[SABINO, 1993]

Carlos A. Sabino "Como hacer una tesis", 1993

Editado por Panamericana, Bogotá, y Lumen, Buenos Aires

[SALAMANCA,1998]

DR. Fabio Salamanca Gómez "CITOGENÉTICA HUMANA - Fundamentos y aplicaciones clínicas", 1988

Editorial medica panamericana.

[Booch, Rumbaugh& Jacobson]

Booch Grady, Rumbaugh James and Jacobson Ivar, 1999: "El lenguaje unificado de modelado".

1ra ed. Ed. Addison Wesley.

[Presmman, 2002]

Pressman Roger S., 2002: "Ingenieria del Software – Un enfoque Practico" 5taed.

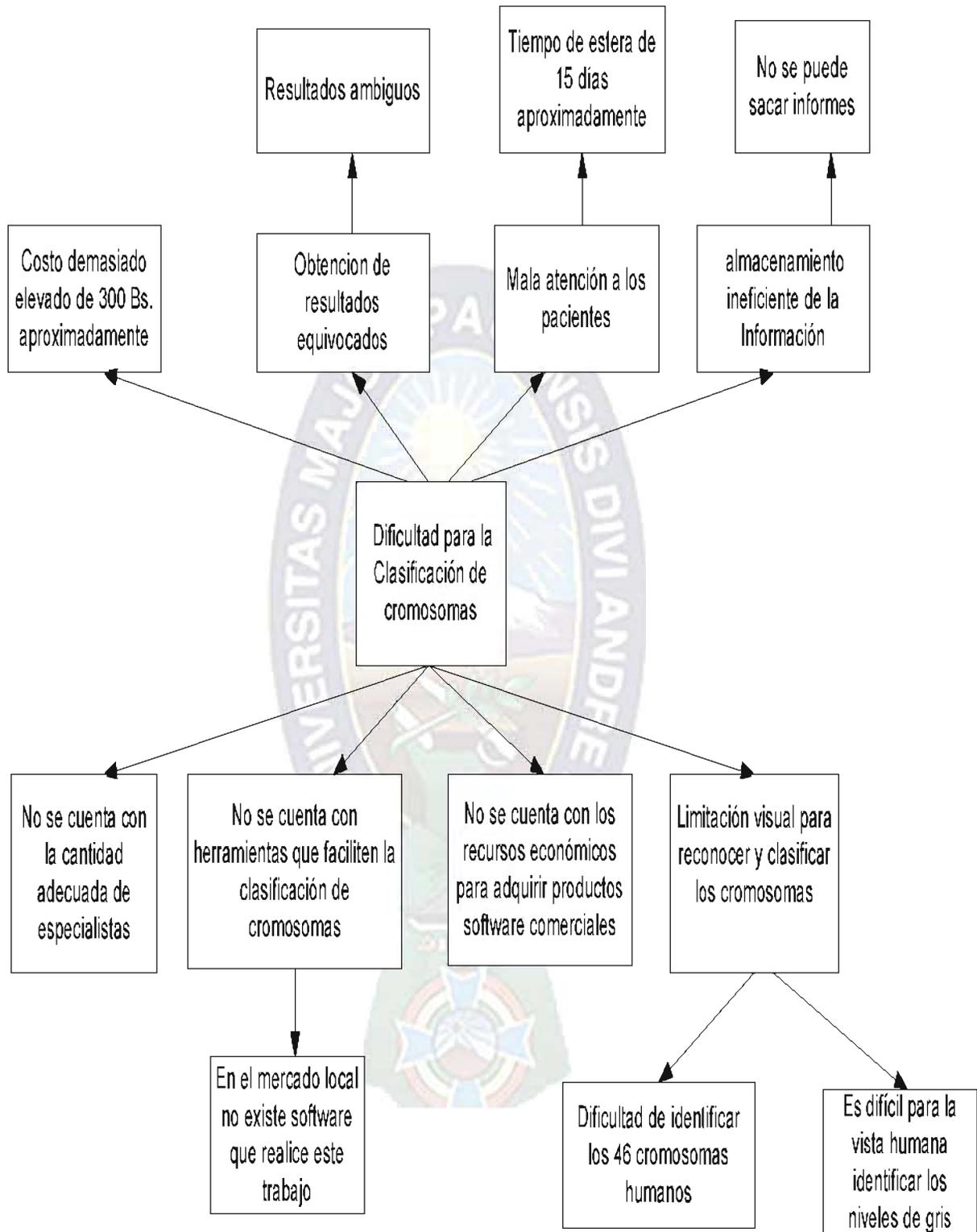
Ed. MacGraw Hill



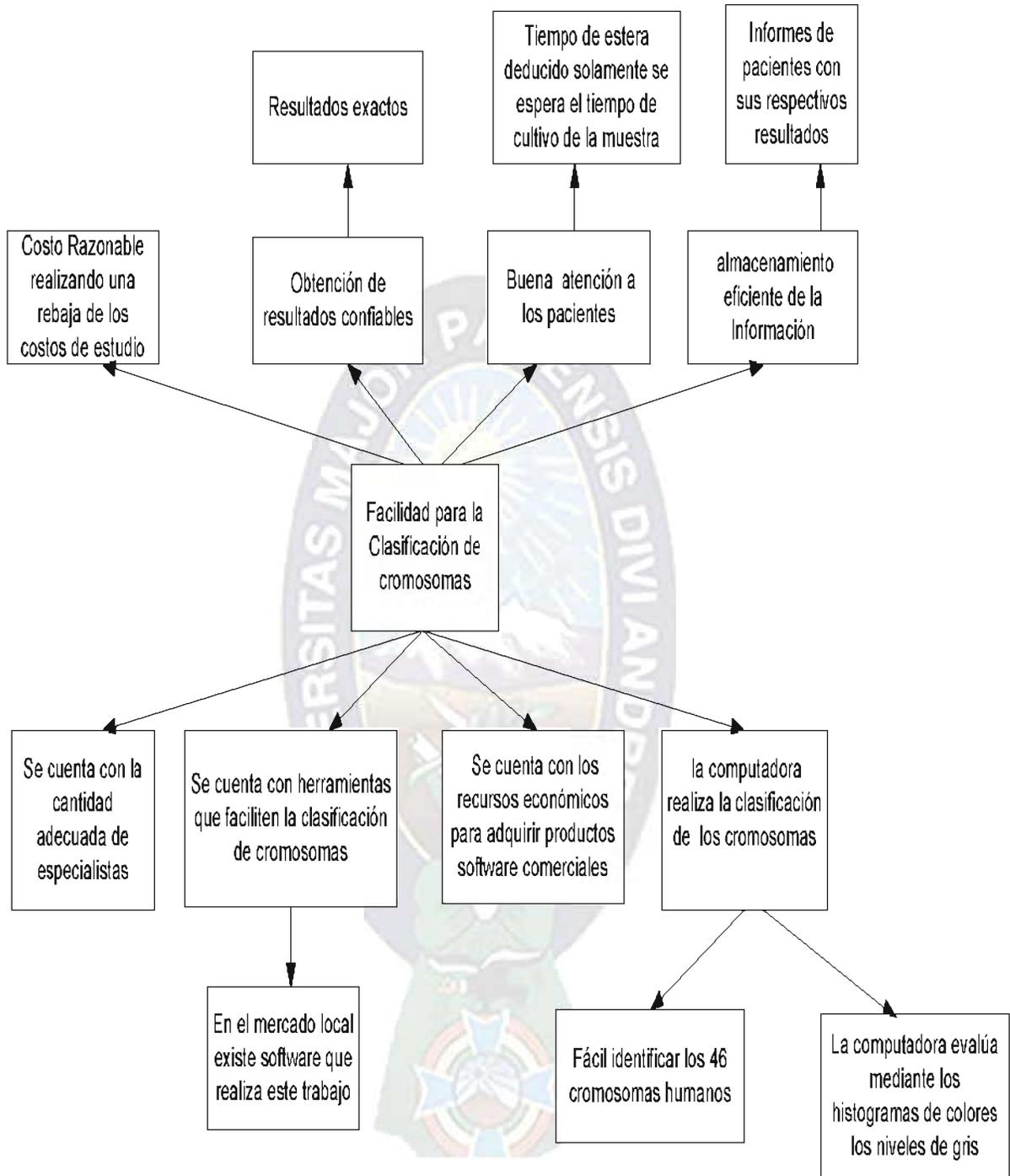


ANEXOS

ANEXO A. ÁRBOL DE PROBLEMAS



ANEXO B. ÁRBOL DE OBJETIVOS



ANEXO C. MATRIZ DEL MARCO LÓGICO

OBJETIVOS	INDICADORES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<p><u>FIN</u></p> <p>Reducir el Costo y el tiempo de atención a los pacientes</p>	<p>Tasa de atención a pacientes elevada Hasta diciembre 2011</p>	<p>Encuestas Informes de evaluación anual del Instituto de Genética</p>	<p>Verificación y aceptación de las herramientas de investigación en el ámbito Local</p>
<p><u>PROPOSITO</u></p> <p>Agilizar el proceso de selección, estudio y analisis de cromosomas</p>	<p>Alto porcentaje de aceptación del producto software por parte de los investigadores. Alta velocidad del proceso de obtención de resultados hasta diciembre del 2011</p>	<p>Informes de investigadores, incluyendo fechas y resultados. Verificación de encuestas Entrevistas Aval de culminación de proyecto Informes de optimización de velocidad del proceso de investigación</p>	<p>Propósitos cumplidos Acuerdos para el mantenimiento de sistema Obtener la aprobación investigadores.</p>
<p><u>PRODUCTOS</u></p> <p>Herramientas de Clasificación de Cromosomas mediante Redes Neuronales</p>	<p>Proyecto presentado al instituto de Genética Hasta diciembre del 2011.</p>	<p>Nota de entrega de productos otorgado por el instituto de Genética</p>	<p>Sistema implantado y funcionando en la institución. probación de producto software control de calidad</p>

<p><u>ACTIVIDADES</u></p> <p>Estudio de la situación actual</p> <p>Identificación de los problemas</p> <p>Requerimientos de desarrollo.</p> <p>Requerimientos de Implantación.</p> <p>Identificación de la metodología para el desarrollo del Sistema.</p> <p>Recopilación de información relacionada</p> <p>Análisis y diseño</p> <p>Implantación</p> <p>Pruebas</p> <p>Instalación</p>	<p>Equipamiento computacional</p> <p>Bases de datos de información digital.</p> <p>Reglamento interno del instituto</p> <p>Normas de presentación de informes</p> <p>Plan de implantación y pruebas</p> <p>Informe de costes y beneficios</p>	<p>Nota de compra de equipamiento.</p> <p>Nota de entrega de sistema de bases de datos.</p> <p>Documentos de la institución</p> <p>Nota de recepción de plan de implantación pruebas.</p> <p>Informe de rendición de cuentas.</p>	<p>Decisión de la dirección</p> <p>Aprobación del proyecto de desarrollo</p> <p>Aprobación presupuestaria</p>
---	---	---	---

ANEXO D.RESULTADOS DE RECONOCIMIENTO DE CROMOSOMAS

Tipo	Imagen	Tipo 1	Tipo 2	Tipo 3	Tipo 4	Tipo 5
1	1	98,48	80,15	61,31	41,48	21,49
1	2	98,02	80,41	60,94	41,71	21,75
1	3	98,41	79,92	61,13	41,48	21,49
1	4	99,39	79,49	60,35	40,52	20,54
1	5	99,21	80,45	60,5	40,67	20,74
1	6	99,49	80,15	60,43	40,46	20,49
1	7	98,83	79,49	59,58	40,23	21,15
1	8	99,74	79,95	60,03	40,11	20,19
1	9	98,53	80,32	61,06	41,4	21,45
1	10	98,75	79,73	60,27	41,17	21,23
1	11	98,99	80,37	60,57	40,75	20,99
1	12	98,67	79,69	60,17	40,84	21,32
2	13	79,3	98,38	79,73	60,26	40,56
2	14	79,67	98,65	79,37	60,05	40,25
2	15	80,77	96,99	78,46	58,94	39,14
2	16	79,94	97,62	78,59	59,25	39,96
2	17	79,67	98,65	79,37	60,05	40,25
2	18	79,79	98,4	79,4	60,02	40,16
2	19	80	98,38	79,57	59,64	39,99
2	20	78,97	98,85	79,84	60,65	40,99
2	21	79,44	98,87	79,71	60,16	40,52
2	22	79,8	98,71	79,73	60	40,1
2	23	79,8	99,27	79,97	60,03	40,07
2	24	78,33	97,78	78,85	59,8	41,61
3	25	79,67	98,65	79,37	60,05	40,25
3	26	60,67	78,85	97,75	79,13	59,25
3	27	59,24	79,12	97,83	78,74	60,08
3	28	59,09	79,07	98,7	79,82	60,86
3	29	59,82	79,57	98,11	79,57	60,12
3	30	59,72	72,61	98,41	79,68	60,2
3	31	70,78	88,69	88,69	69,05	49,17
3	32	59,7	79,61	98,15	79,89	60,27
3	33	59,54	79,42	96,67	78,68	60,483
3	34	59,3	79,26	98,05	79,43	79,43
3	35	59,44	79,05	97,65	79,43	60,53
3	36	59,8	78,12	96,99	78,46	60,18

4	37	41,59	60,69	79,66	97,82	78,26
4	38	40,57	59,97	78,67	97,44	79,44
4	39	49,02	60,08	78,56	97,56	77,65
4	40	39,83	58,97	78,47	98,27	80,03
4	41	40,99	60,15	79,45	98,23	78,49
4	42	39,81	59,74	79,73	99,35	80,17
4	43	40,75	59,91	79,73	98,29	79,51
4	44	40,69	60,18	79,48	98,45	79,21
4	45	39,84	59,84	79,81	98,5	80,15
4	56	40,76	60,14	79,98	97,26	79,19
4	47	41,26	61,23	79,59	97,07	78,69
4	48	40,17	60,14	79,48	98,48	79,81
5	49	21,22	41,21	61,09	79,47	98,72
5	50	21,59	41,46	60,66	79,22	98,33
5	51	20,57	40,57	60,57	80,39	98,93
5	52	21,9	41,88	61,3	80,54	98,07
5	53	21,26	41,25	61,25	80,21	98,73
5	54	21,48	41,45	61,48	80,43	98,51
5	55	21,23	41,21	61,18	80,27	98,68
5	56	21,25	41,24	61,2	80,4	98,62
5	57	21,15	41,05	60,53	80,12	98,67
5	58	22,56	41,99	60,61	80,25	97,25
5	59	20,83	40,83	60,82	80,61	99,15
5	60	19,56	39,6	57,7	76,65	98

