Teórico

1. **Que es un histograma. Explicar ecualización de histogramas.**

Un histograma es un grafico que representa las frecuencias relativas de los niveles de gris de una imagen. Este es utilizado para el cálculo de coeficientes estadísticos tales como: el valor medio, la desviación estándar y la entropía. También es utilizado para aplicar un procedimiento de mejoramiento de imágenes denominado ecualización o igualación de histograma.

Este proceso se basa en una transformación g -> g’ de manera de lograr una reacomodación de las líneas del histograma en nuevos intervalos Δg. Basándonos en: .

Así, cuanto más alto sea h(g) mayor será la separación entre las barras.

Para el cálculo de el nuevo valor de g usaremos la siguiente fórmula: *gnuevo = gprevio + k h(g)*

1. **Explicar filtros de imagen (Filtro mediana, Filtro valor medio)**

Estos operadores calculan un pixel para la imagen transformad, en función de la vecindad local en la imagen original.

El filtro de la mediana consiste en, por ejemplo tomar una ventana de 1x5, reemplazar el valor central por el valor de la mediana de los números q la componen. (La mediana se calcula ordenando los elementos y tomando el valor q queda al centro)

El filtro de la media es similar pero se reemplaza el valor del medio por el promedio del conjunto

1. **Convolución de imágenes digitales definición, ejemplo, Mascaras, Aplicaciones.**La convolución de imágenes digitales es un método mediante el cual se puede hacer resaltar  o disminuir la presencia de ciertos valores en una región  deseada mediante el uso de mascaras.

La convolución discreta consiste en realizar la suma de los productos entre los componentes, de la imagen y de la máscara, que ocupan la misma posición en la imagen, y reemplazar el pixel de referencia de la vecindad por el nuevo valor.

Las aplicaciones más importantes de la convolución es el suavizado de imágenes q consiste en la perdida de nitidez de la imagen y en la detección de bordes.

Para esto se usan distintos tipos de mascaras.

Prewit Sobel Prewit Sobel

 Bordes VerticalesBordes Horizontales

1. **Clasificador polinomial: Fundamentos, graficar un clasificador polinomial con vector de entradas de 2do orden y explicar el funcionamiento**

En casos en que deben resolverse problemas de clasificación tales como clasificación no lineal o clasificaciones donde la información está contenida en vectores de dimensiones de dos dígitos debe introducirse el concepto de funciones de decisión generalizada, lo cual significa aplicar funciones a cada una de las componentes del vector de patrones, tomando a estas como las componentes de un nuevo vector

Es decir q dado ***x****= (x1; x2; x3;…;xn)* se transformara como ***x\*****=(f1(x1);f2( x2);f3( x3);…;fn(xn)) = (x1\*; x2\*; x3\*;…;xn\*).*

En el caso en que la función a aplicar es la función cuadrática transformamos los patrones según la expresión:

X\* = (x1;x2)2 = (x12;x22;x1x2;x1;x2;1) = *(x1\*; x2\*; x3\*; x4\*; x5\*;1)*

La cual surge del vector de patrones expandido ***x****= (x1; x2; 1)* y el cálculo de las combinaciones posibles de sus elementos tomados de 2 en 2.

Con esta transformación la función polinomial toma la forma:

D(x\*) = w1.x12 + w2.x22 + w3.x1x2 + w4x1 + w5x2 + w61

O bien d(**x**\*) = **w.x\***  con **w** = (w1,w2,w3,w4,w5,w6).



En el aumento de las características de los problemas reales se hace inviable el cálculo de la matriz de coeficiente, por lo que se aplica otro procedimiento. Se introduce un nuevo vector z = (x1,x2,x3,…,xn, y1,y2,y3,…,ym) y una nueva matriz Mz.



Para resolver el clasificador polinomial simplemente debemos construir la matriz M y aplicar transformaciones por filas.

1. **Perceptrón: Dibujar el Perceptrón con 2 entradas, explicar proceso y fase de prueba que utiliza, dibujar el resultado en el plano cartesiano, si se modifica la función escalón, a qué tipo de red se convierte**

Perceptrón es un esquema que resuelve el problema de clasificación lineal de patrones y que se corresponde con el modelo biológico de una neurona humana.

En este modelo en vez de expandir el vector **x** se adiciona una entrada a la neurona que siempre recibe valor 1

X1

X1

Xb = 1

w1

w2

*NET*

*F(****wx*** *+ b) = F(NET)*

*y*

Al recibir el vector **x**, la red calcula la función de decisión lineal y le aplica 1 función umbral que la fuerza a obtener solo 2 valores de salida: 0 y 1, o 1 y -1 según como se defina la función F.

El vector **w** se calcula por un proceso iterativo de aproximación en vez de usar cálculos matriciales.

La ley de aprendizaje del Perceptrón considera que el ajuste de los coeficientes no interviene directamente en la salida **y**, sino la diferencia entre la salida obtenida al procesar la información y la salida deseada. O sea:

*Δwij = xi (yi – yj)*

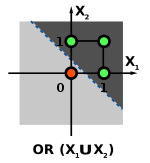
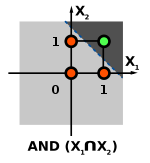
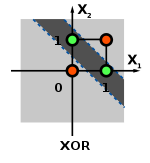
/ yi = salida obtenida

yj = salida deseada

de esta manera el nuevo valor de **W** se obtiene

***W****i =* ***W****i-1 + xi (y’i –* ***W’****i-1 xi)*

Si w1 y w2 son dos clase entre las q se quiere clasificar a **x**, si xЄw1 y w(k).x(j) ≤ 0 o xЄw2 y w(k).x(j) ≥ 0 se debe modificar W, sino no.

Al reemplazar la función escalón por una función rampa se obtiene el modelo ADALINE.

1. **Perceptrón lineal (adaline) fundamentos, esquema, entrenamiento, se puede utilizar adaline en redes multicapa, como?**

Se usa para mejorarla resolución de problemas no lineales usando redes multicapa.

Si reemplazamos la función de función de salida abrupta de un Perceptrón por una función rampa, podremos generalizar el Perceptrón para E/S continuas y además aplicar la técnica del gradiente o descenso iterativo.

Usamos una ley de aprendizaje denominada Ley Delta o Ley de Widrow/Hoff, que combina las de Hebb y la de Perceptrón:

*Δwij = lr (yi – yj).xi* = lr.δ xi

En vez de Perceptrón los llamaremos ADALINAS

Ahora consideremos el error cuadrático medio

LMSE =

Siendo: Yk = salida deseada

Ok = salida e la red

En cada paso modificaremos el valor de w hasta alcanzar el optimo haciendo mínimo el error.

Si trabajamos en Rn de orden n-1, la superficie del grafico del error sería muy irregular por lo que deberemos conformarnos con encontrar una solución lo suficientemente buena.

Existe una versión multicapa de la ADALINE denominada MADALINE (Múltiple ADALINE, múltiples Adalides) que consiste en una red neuronal con neuronas similares a las de la ADALINE pero que contiene capas de neuronas ocultas.

1. **Diferencia entre red Perceptrón y Adaline. (Acá mencionar la ley Delta y el cambio de F de escalón a lineal)**

La principal diferencia entre Perceptrón y Adaline es la función de Salida.

En Perceptrón la salida es binaria, en Adaline es real

El Perceptrón utiliza la salida de la función umbral (binaria) para el aprendizaje. Sólo se tiene en cuenta si se ha equivocado o no.

En Adaline se utiliza directamente la salida de la red (real) teniendo en cuenta cuánto se ha equivocado.

Otra diferencia importante es la ley de aprendizaje q ambos utilizan.

Perceptrón usa la Ley de Aprendizaje q lleva su nombre.

*Δwij = xi (yi – yj)*

Adaline en cambio utiliza la Ley delta que es una combinación de la ley de Hebb y la de Perceptrón

*Δwij = lr (yi – yj).xi* = lr.δ xi

En Adaline hay una razón de aprendizaje γ para regular lo que va a afectar cada equivocación a la modificación de los pesos. Es siempre un valor entre 0 y 1 para ponderar el aprendizaje

1. **Redes Retropropagacion: Explicación; Grafico; Propagación del error**

Es un algoritmo que surge del problema de aplicar una ley de aprendizaje para entrenar un Perceptrón multicapa.

La solución es la posibilidad de calcular el error a la salida de cada capa en función del error de la capa.

Es un método de aprendizaje supervisado de gradiente descendente, en el que se distinguen claramente dos fases: primero se aplica un patrón de entrada, el cual se propaga por las distintas capas que componen la red hasta producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de las capas intermedias. Cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido, se ajustan los errores de los pesos sinápticos de cada neurona.

*La grafica y la formula la saqué tal cual del libro por eso no la agregué*

1. **Que es un sistema Experto. Como es su estructura.**

Es un programa destinado a generar inferencias en un área especifica del conocimiento de manera similar a la q se espera de un experto humano

Su estructura es la siguiente:

Usuario

Interface

Explicaciones

Knowledge Data Base

Experto

Motor de Inferencias

Editor

1. **Función de membrecía de lógica difusa: explique qué significa, enumerar y definir al menos dos operadores difusos**

La función de membrecía o pertenencia es una función que representa en qué nivel o medida una variable de entrada pertenece al conjunto difuso. Esta es la que le dará forma a los conjuntos difusos

Las formas de funciones de membrecía más usadas son:

|  |  |  |  |
| --- | --- | --- | --- |
| Trapezoidal | triangular | gaussiana | singleton |
|  |  |  |  |

Operadores difusos:

Cuando se utiliza más de una variable lingüística es necesario usar una variable de salida q sea inferida por alguna operación que tenga en cuenta el grado de membrecía de cada una de las variables.

Los operadores más conocidos son:

(Max): equivalente al operador OR. Asigna la correspondiente al valor máximo

(Min): equivalente al operador AND. Asigna el correspondiente al valor mínimo.

(Complemento): se calcula la negación de las variables.

Etapas para la aplicación de las reglas de lógica difusa

**Paso 1 – Fuzzyficacion de las entradas:** se toman las entradas y se determina el grado de pertenencia en los conjuntos difusos a través de la función de membrecía. Esta variará en el intervalo entre 0 y 1.

**Paso 2 – aplicación de operadores difusos:** si existen múltiples partes en el antecedente se aplican operadores difusos para obtener un numero que representa el resultado del antecedente para esa regla.

**Paso 3 – Aplicación de la implicación:** es la conformación del consecuente. Se emplea el grado de soporte de la regla para conformar el conjunto fuzzy completo a la salida. Si el antecedente es parcialmente cierto luego el conjunto difuso se trunca según el método de implicación.

**Paso 4 – Agregación:** el proceso de unificación de las salidas de cada regla uniendo los procesos paralelos. Se combinan las salidas en un único conjunto difuso.

**Paso 5 – defuzzyficacion:** la salida de este proceso es un numero. Consiste en el cálculo del centroide del grafico obtenido en el paso 4.

1. **Transformada de Hough. Como se calcula y para qué sirve?**

La transformada de Hough es un método que resulta útil para la detección de ciertas morfologías simples de los bordes tales como bordes rectilíneos y círculos. Modo de operación es principalmente estadístico y consiste en que para cada punto que se desea averiguar si es parte de una línea se aplica una operación dentro de cierto rango, con lo que se averiguan las posibles líneas de las que puede ser parte el punto. Esto se continúa para todos los puntos en la imagen, al final se determina qué líneas fueron las que más puntos posibles tuvieron y esas son las líneas en la imagen.

1. **Enunciar las características principales de LISP.**

LISP es un lenguaje de programación en el que todos los componentes (variables, constantes, sentencias de cualquier tipo, funciones, parámetros, datos) son expresados a través de información almacenada en listas.

Los dos componentes básicos son **Listas** y **Símbolos.**

Un código LISP está compuesto por formas; el intérprete lee una forma, la evalúa, e imprime el resultado. Este procedimiento se denomina ciclo de lectura, evaluación, impresión.

Si…

Las listas se representan encerrándolas entre paréntesis, y siempre se evalúan a partir de la evaluación del primer símbolo de la lista, y en función de este se decidirá como emplear los elementos que le siguen. Es decir que se utiliza una notación prefija, así para codificar una suma primero escribiremos el operador y luego los parámetros.

1. **Complejidad temporal: Concepto. Análisis; Funciones O(n); Ejemplos de problemas reales, discuta sucintamente la función O(n) de complejidad de cada uno de los algoritmos ejemplificados.**

La complejidad temporal se refiere a la cantidad de intervalos que demanda completar la ejecución de n proceso

A fin de conocer la razón de crecimiento entre el indicador de tiempo y la dimensión de los datos, que representa el parámetro medible, tendremos complejidad temporal lineal, polinomica, logarítmica exponencial, etc. Según la naturaleza de la expresión que las vincula.

El límite de crecimiento de esta medida se denomina complejidad temporal asintótica y es lo que determina el tamaño del problema que puede ser resuelto con cierto algoritmo.

En general se dice que una función g(n) tiene un orden de complejidad temporal O(f(n)) si existe una constante C tal que

g(n) ≤ C f(n)

para todos los valores posibles de n

*Para el punto de los ejemplos agregar los que están al final de la fotocopia de complejidad (P Ej.: algoritmos de Búsqueda y de clasificación de patrones)*

1. **Método de resolución (de Prolog-por refutación)**

El procedimiento de resolución es en proceso iterativo cuya finalidad es evaluar la verdad o falsedad de una premisa. En cada paso dos cláusulas son resueltas para obtener una tercera llamada resolvente.

El algoritmo de resolución por refutación consiste en probar una proposición P con respecto a un conjunto de axiomas C.

1. Convertir todas las afirmaciones C a la forma clausal.
2. Negar P y convertir a la forma clausal. Añadir al conjunto C.
3. Repetir lo siguiente hasta que no haya progreso o se detecte una contradicción.
   1. Seleccionar dos clausulas padres
   2. Resolverlas, obteniendo el resolvente.
   3. Si el resolvente es la clausula vacía, se detecto una contradicción, sino añadirlo al conjunto de clausulas.
4. **Que es una búsqueda heurística. Cuál es la diferencia entre primero el mejor y A\*.**

Una búsqueda heurística es una técnica que aumenta la eficiencia de un proceso de búsqueda agregando información que permite reducir la cantidad de combinaciones y que permite seleccionar una ruta con menores costos.

El conocimiento Heurístico se incorpora en las reglas de producción o con una función heurística, determinando estados deseables y asignando pesos a ciertos aspectos del problema, puedo estimar que tan deseable es un nodo o un camino dado.

La búsqueda por Primero el mejor utiliza una función Heurística que se basa en el costo del nodo a seleccionar, a diferencia de la búsqueda por A\* que utiliza dos funciones, una que usa el costo de un camino desde el nodo actual hasta la solución y otra que utiliza el costo desde la raíz hasta el nodo actual.