

Panorama general de los Sistemas evolutivos

Entrevista con el licenciado Fernando Galindo Soria

ESCOM día 8 de Octubre de 1999 tuvimos una entrevista con el Lic. Fernando Galindo Soria quien ha contribuido directamente en la investigación sobre Los sistemas evolutivos, desde su nacimiento hasta el día de hoy. He aquí un resumen del mismo.

Ultima revisión Junio del 2009

Normalmente cuando programamos usamos una técnica que se engloba dentro de lo que llamamos sistemas formales, un sistema formal es un conjunto de reglas que nos permiten resolver un problema. Cuando hacemos un programas en Basic o C, lo que se esta haciendo es un sistema formal, se le esta diciendo a la computadora, si aplicas esta regla, o esta otra, vas a resolver el problema , los sistemas formales son los que mas se usan para programar, son muy usados y muy buenos, el problema que tienen los sistemas formales es cuando existe algún problema que no puedan resolver, supóngase que hacemos un programa que reconoce la letra A, en el programa deberían de incluirse las características de la letra A, de modo que identificara A's similares, pero si se quiere reconocer una B no la reconocería , o tampoco reconocería la letra a minúscula, a cuadrada y otras A's incluso, a menos que explícitamente se le diga como hacerlo, lo cual a la larga se vuelve muy pesado, porque siempre existe algo no cubierto por las reglas que tiene el sistema formal.

En 1930 Kurt Gödel estableció un teorema que lleva su nombre, el teorema de Gödel puede considerarse como un teorema fundamental en informática. El teorema dice en general que: un sistema formal, es incompleto, o inconsistente, un sistema formal es incompleto cuando tarde o temprano, no importando el número de reglas que tenga, va a existir un problema que se supone debería poder resolver y que no puede resolver, o sea le faltan reglas para poderlo resolver, e inconsistente significa que tarde o temprano, aplicando un mismo grupo de reglas encuentra una solución verdadera y una falsa para el mismo problema.

Si se logra que un sistema formal sea completo será inconsistente y si es consistente entonces será incompleto. Lo anterior no significa que un sistema formal no sea útil , por el contrario nos han resuelto un gran numero de problemas, por que la forma común de programar es mediante sistemas formales, hagamos de cuenta que los sistemas formales es como la teoría de Newton con respecto a la teoría de Einstein.

La física de Newton la usamos para muchas cosas, por ejemplo, cuando calculamos la velocidad de un proyectil no usamos la teoría de Einstein, con Newton de hace cientos de años nos funciona muy bien, si quiero ver como rebota una pelota, con Newton nos basta, si queremos saber el comportamiento de el proyectil cuando se acerca a la velocidad de la luz entonces ya entramos con Einstein; los sistemas formales son así , para espacios normales restringidos convencionales funcionan muy bien, el problema es que nosotros ya no estamos en esos espacios, los problemas a los que ahora nos enfrentamos ya rebasan un poco a esa cotidianidad, los informáticos rompieron ya esa cotidianidad hace aproximadamente unos cincuenta años, cuando empezaron a enfrentar problemas que ya no eran tan fácilmente resolubles utilizando sistemas formales, ya requerían de otro tipo de tratamiento, desde los años 50 con el surgimiento de la Inteligencia Artificial, se empezaron a atacar una serie de problemas que ya no eran tan obvios de resolver , por ejemplo un problema de esa época era el reconocimiento de letras.

En los años 60 surge el área de Learning Machines (Maquinas que aprenden) la idea de una Learning Machine es que se invierte la situación, en lugar de que yo me preocupe de darle las reglas para que reconozca un objeto, le doy la capacidad de que reconozca o aprenda como esta hecho el objeto, los sistemas Learning Machine tienen un programa al cual por ejemplo le damos una A y el programa se las arregla para encontrar la estructura de esa A. Estos programas construyen un archivo o un sistema de ecuaciones dónde están representadas el conjunto de reglas que representan a esa A. Lo bonito de un Learning Machine es que uno le puede dar por ejemplo una A cuadrada, rectangular, o chiquita y va construyendo la representación de todas esas A's .

Las Learning Machines tiene dos grandes etapas: “de Aprendizaje” y Formal, en la etapa de aprendizaje se le da a la computadora un montón de ejemplos de aquello que queremos que reconozca y el sistema encuentra las reglas que describen ese tipo de cosas o sistemas, y ya que tienen ese conjunto de reglas, se las pasa a un programa formal, al que por ejemplo se le da una A cuadrada y entonces el sistema la reconoce como A cuadrada, lo que nos ahorramos es el trabajo de estar encontrando las reglas, El libro básico sobre el área “Learning Machines” de Nilsson se escribió en 1965 .

El problema de los Learning Machines es que si por ejemplo se le da un montón de ejemplos de A's, reconoce A's, pero si se le pide reconocer una A Itálica y no se le enseñó a reconocer ese tipo de A's, no la reconoce.

Este tipo de problema nos mantuvo ocupados durante los años setenta y principios de los ochenta, observamos en aquellos años que, cuando se requiere hacer una modificación para reconocer algo que no se le ha enseñado es necesario volver a enseñar al sistema y se tiene que estar reestructurando continuamente, de donde salió la idea de hacer un sistema que se encargara de hacer las modificaciones automáticamente y que éste se modificara asimismo.

De donde la idea sería utilizar un sistema que contuviera las reglas por ejemplo para reconocer una A, y que cuando esté no reconociera un tipo especial de A, mandara llamar a otro tipo de programa que se encargara de aprender, lo que se convirtió en el principio o la primera idea de lo que es un sistema evolutivo o sea un programa que permanentemente esta aprendiendo y reestructurando sus reglas de donde el programa no mandaría Fatal error o warnings, simple y sencillamente si el programa no sabe, aprende.

En 1983 se pensó en la modificación de las reglas en tiempo real. La idea de que un programa modificara su propio código resultaba una tarea muy compleja, de donde surgió la idea de hacer que las reglas no se encontraran dentro del código original del programa pero si en un archivo legible y modificable por el mismo programa y que las reglas fueran tomadas de ese archivo. Para lo cual se tomo en cuenta que un sistema de información contienen tres bloques, el bloque de Datos, el de Procesos y el de Estructura. Por lo que se vigiló que estos tres bloques quedaran separados relativamente, eso se conoce como independencia relativa, un modelo de un sistema evolutivo tiene por un lado los datos, por otro la estructura y por el otro los procesos.

Las primeras estructuras de sistemas evolutivos realmente eran compiladores modificados para que cuando encontrarán una regla que no esta dentro de la gramática modificara la gramática, la modificación es que, cuando el programa no reconoce una estructura no mande fatal error, si no que modifique la gramática, de donde obtendríamos un sistema evolutivo relativamente sencillo, concepto que ha crecido desde 1983.

Los sistemas evolutivos mas simples se conocen como sistemas de reescritura, son sistemas muy elementales, a los cuales se les dan entradas que son buscadas por la computadora en archivos que tienen dos columnas, buscan la entrada en la primera columna y si encuentra esa entrada le asocia su significado, pero si no encuentra la entrada entonces pregunta y uno le da el significado, un ejemplo es un sistema elemental traductor de Idiomas, si le damos la entrada "Hola Perro", si la computadora no encuentra en su lista esta frase, entonces pregunta, ¿ que es "hola perro "? , y entonces le decimos "Hello Dog", entonces la computadora almacena en su primera columna "Hola perro" y en la otra "Hello Dog", la siguiente vez que introduzcamos "Hola Perro" la computadora va a decir "Hello dog".

la arquitectura de programación de los sistemas evolutivos constituyen una ampliación a las posibilidades a los programas convencionales, no los substituyen , si no que los mejora. tendríamos que mencionar que no solo estamos hablando de manejo de texto o sonido si no también de imágenes, con lo cual los sistemas evolutivos tienen que tener mecanismos de percepción, no solamente del teclado, con sistemas de percepción la velocidad de adquisición de conocimiento será mucho mayor. En 1988 hablábamos ya de sistemas evolutivos interactivos, por sus mismos mecanismos de percepción y de interacción por ejemplo un brazo, una cámara de video, etc.

Una de las técnicas que se desarrollo en el enfoque de los sistemas evolutivos fue el de Matrices evolutivas, es una herramienta orientada a espacios masivos de información, una matriz evolutiva es un mecanismo que me permite representar grandes cantidades de información utilizando un mecanismo que se transforma muy fácilmente, un ejemplo es si necesitamos reconocer imágenes u objetos utilizamos un mecanismo de redes neuronales representadas en forma de matriz, a esta matriz se le enseñan distintos modelos, objetos o cosas, y la red neuronal es capaz de reconocerla, pero si le dábamos a reconocer un objeto no grabado previamente este no la reconocería, a mediados de los ochenta se empieza a trabajar en problemas de reconocimiento de imágenes, si se le da la foto de un perro almacenaba la foto de un perro en la matriz, si le enseñábamos otra del mismo perro iba encimando (Sumando) fotos del perro.

Lo que se conoce como cluster analisis (análisis de cúmulos) es una técnica de los años sesentas para reconocimiento de formas, se basa en una idea muy simple, supóngase que se quisiera discriminar entre personas, pondríamos unas variables para distinguir su tamaño y altura, entonces se puede graficar para cada personas un punto en dos dimensiones en donde una coordenada sea el tamaño y la otra coordenada sea la altura, obviamente después de cierta cantidad de personas empiezan a haber colisiones, podríamos entonces poner una variable de edad y por si no fuese suficiente otra para el color de piel, y a lo mejor no habría tanta colisión, de donde con un grupo de variables podríamos graficar a cada persona como un punto en el espacio.

Si en un momento dado se ve a la misma persona en dos momentos diferentes, es posible que el color de piel no sea exactamente igual, o la estatura, a lo mejor esta persona anda descalza, cada vez que se miden los parámetros de una persona habrán ciertas variaciones, pero no son muchas, esta persona no crecerá 2 metros mas, o de blancos a morenos, entonces se forma una nube de puntos que representan a la misma persona, para otra persona con otra nube y para otra mas con una nube más, estas nubes se pueden caracterizar por su media, o sea la media de la nube y la varianza de la nube, esto se conoce como cluster o cúmulo, por ejemplo se puede discriminar entre perros y gatos y sacar el cúmulo, estos son perros y estos otros son gatos, los clusters a veces colisionan, pero en general permiten discriminar bien. El problema surgía cuando por ejemplo tomaba la foto de mi perro en otro momento y se salía de su cluster, entonces la máquina decía : "No reconozco a este objeto", o sea había que programar todo de nuevo para que reconociera a ese perro.

La solución ... Para empezar las matrices evolutivas están vacías, esa es la primera característica y para continuar si llega una foto del perro, en lugar de decir, no reconozco al sujeto si no lo encuentra pregunta ¿oye que es esto? Y contestaríamos mi perro, entonces lo integra, como está vacía la matriz la primera vez no tiene nada, yo doy la foto de mi perro, y va a buscar en la matriz y va a preguntar: ¿ que es esto pues no lo encuentro? (no va a decir "no hay nada") nos va a decir : "No lo encuentro", y si le decimos es mi perro, entonces crea un renglón llamado mi perro y aquí pondría todos los parámetros de mi perro, entonces eso sería un punto en el espacio, la siguiente vez que le de una foto de mi perro ira a buscarla y por mala suerte supongamos que tampoco la encontró, y preguntaría: ¿ oye que es eso ? y le volveremos a decir: " mi perro", entonces haría lo que un Learning Machine, lo suma a este, saca el promedio, saca la varianza pero en términos prácticos es que saca otro punto y saca la media, y si le doy otra foto que también es mi perro, y se sale de la nube de puntos en que se encontraba normalmente, entonces saca la media, saca la varianza,

Si observamos, originalmente la media se situaba en un punto en el espacio, después cambio el punto a otras coordenadas, todos los puntos anteriores se van quedando en la grafica formando un cúmulo de puntos y cada vez que se saca otra media por la introducción de otra imagen por ejemplo, otro punto aparecerá en el espacio formando parte de la nube que representaría a mi perro, y conforme obtengamos mas puntos, el proceso se afinara por consecuencia, y hará más sencillo el reconocimiento, la característica fundamental es que esta representación siempre se va estar transformando, nunca me pregunta si se va a transformar o no, simplemente se transforma, entonces, aquí ocurren fenómenos preciosos, para empezar, este tipo de sistemas evolutivos se están transformando permanentemente en tiempo real, únicamente por el flujo masivo de información que esta pasando.

El área que estudia los procesos de cambio se llama evolución.

INSTITUTO POLITÉCNICO NACIONAL

"Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas"

"Ingeniería del Conocimiento"

Este curso fué realizado por la Lic. Claudia Marina Vicario Solórzano