

Taller Latino de Investigación de Operaciones

“ La IO aplicada en el desarrollo sustentable de las zonas indígenas de américa latina ”

Cálculo de Factores Franck-Condon utilizando Redes Neuronales Artificiales y Optimización mediante Cúmulos de Partículas

I. Chaman¹, C.A. Coello Coello¹, L. Sandoval²

¹Centro de Investigación y Estudios Avanzados del IPN

Departamento de Computación

²Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Resumen: En este artículo, se aborda el cálculo de los Factores Franck-Condon (FFC) para moléculas diatómicas para números cuánticos grandes, el cual es un problema importante en astrofísica. La determinación de estos factores requiere del cálculo de una integral definida, el cual suele realizarse con métodos numéricos. Sin embargo, debido a los avances tecnológicos actuales, se ha hecho necesario utilizar números cuánticos vibracionales grandes, lo cual ha hecho necesario recurrir a otros métodos. En este trabajo se plantea la aproximación de esta integral definida usando redes neuronales artificiales entrenadas con el algoritmo de optimización mediante cúmulos de partículas (*Particle Swarm Optimization*, o PSO). En nuestros experimentos, se probaron diversas variantes del algoritmo PSO, así como varias funciones de prueba de las que se conoce el valor real de su integral, para fines de validación de nuestro procedimiento. Nuestros resultados indican que nuestro enfoque es una alternativa viable para resolver este problema.

Abstract: In this paper, we deal with the computation of the Franck-Condon Factors (FCF) for large quantum number diatomic molecules, which is an important problem arising in astrophysics. Determining such factors requires the computation of a defined integral, which is normally done using numerical methods. However, given the recent technological developments, it has become necessary to use large vibrational quantum numbers, which has required the use of alternative approaches. In this work, we provide an approach to approximate this defined integral using artificial neural networks which are trained using a particle swarm algorithm. In our experiments, different PSO variants were adopted. Also, different test functions for which the real integral is known were adopted for validation purposes. Our results indicate

that our proposed approach is a viable alternative for solving this problem.

Keywords: numerical integration, artificial neural networks, highly oscillating functions, particle swarm optimization

Introducción

Una aplicación importante en la astrofísica es detectar las moléculas interestelares, para lo cual requerimos calcular los denominados “Factores Franck-Condon” (FFC) para moléculas diatómicas. El principio de Franck-Condon nos permite hacer predicciones bastantes exactas de los lugares donde son altamente probables las transiciones vibracionales entre un par de estados electrónicos. Esta idea fue propuesta inicialmente en el año de 1925 por Franck; y un poco más tarde, confirmada por Condon en la mecánica cuántica. En resumen, los FFC se pueden calcular mediante la siguiente función (más adelante se darán más detalles sobre esta ecuación):

$$FFC = \left| \int \Psi_v^* \Psi_{v''} dr \right|^2$$

La motivación de este trabajo fue, el de aproximar la integral definida que se plantea en el cálculo de los FCC utilizando redes neuronales artificiales entrenadas con el algoritmo de optimización mediante cúmulos de partículas (PSO, por sus siglas en inglés). Este es un problema muy interesante de métodos numéricos, ya que, conforme se ha aumentado la aplicación del software y mejorado la tecnología en espectrometría, se ha requerido aproximar los FFC para números cuánticos vibracionales grandes (v' y v''). Esto ha hecho necesario buscar nuevas formas de aproximar las integrales para funciones muy oscilantes.

En este trabajo se utilizan funciones de prueba tomadas de [11], para las cuales se conoce la integral real. La idea es usar estos problemas para ajustar la red neuronal, a fin de utilizar después este procedimiento en una molécula diatómica con números vibracionales grandes.

Descripción del Problema

Descripción de la Red Neuronal

El uso de las redes neuronales artificiales (RNAs) ha sido muy frecuente en diversas áreas del conocimiento. Las RNAs pueden utilizar diferentes funciones de activación para clasificar datos. En nuestro caso la Red Neuronal Artificial (RNA) proponemos que la función de activación esté dada de manera matricial, pues se obtienen matrices al expresar cualquier función como una combinación lineal de funciones coseno.

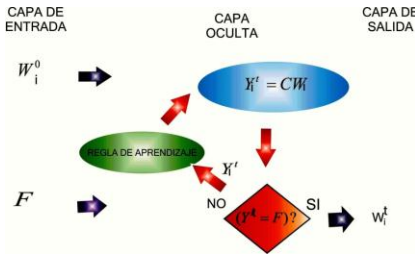


Fig. 1. Esquema general de la arquitectura de la RNA utilizada para el ajuste de una función.

La función de activación está basada en funciones de cosenos como base del espacio de funciones. La red consiste en una capa de entrada y una de salida. En la capa de entrada tenemos un solo vector X de entrada con $(n+1)$ elementos; la parte oculta consiste en un agente inteligente que tiene $(n+1)$ neuronas; en la capa de salida sólo se tiene un vector Y con $(n+1)$ elementos. La función de activación de la capa oculta es un vector CW con $(n+1)$ elementos, donde W es el vector de pesos de la Red Neuronal con $(n+1)$ elementos, C es una matriz de activación de la unidad oculta, y F es la salida deseada como vector salida (ver figura 1).

Definiremos la RNA como sigue: Suponemos que tenemos un patrón de entrada inicial $X = \{x_0, \dots, x_n\}^t \in \mathbb{R}^{n+1}$ como muestra y dada una matriz de activación $C \in M_{n+1, n+1}^{(\mathbb{R})}$ definida como:

$$C = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \cos(x_0) & \cos(x_1) & \dots & \cos(x_n) \\ \vdots & \vdots & \dots & \vdots \\ \cos(nx_0) & \cos(nx_1) & \dots & \cos(nx_n) \end{bmatrix},$$

dado un vector de pesos sinápticos aleatorios $W = \{w_0, \dots, w_n\}^t \in \mathbb{R}^{n+1}$ y una salida deseada $F = \{f(x_0), \dots, f(x_n)\}^t$, donde $f(x)$ es la función muestra.

Al construir un dispositivo sencillo que aproveche la función de la salida deseada F , a partir de un conjunto de datos conocidos de entrenamiento X , se utiliza la matriz de activación, que es una matriz construida con base en los patrones de entrada en un dominio $(n+1)$ -dimensional, obtenemos una salida deseada $Y = \{y(x_0), \dots, y(x_n)\}^t$ definida por:

$$Y = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \cos(x_0) & \cos(x_1) & \dots & \cos(x_n) \\ \vdots & \vdots & \dots & \vdots \\ \cos(nx_0) & \cos(nx_1) & \dots & \cos(nx_n) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

es decir:

$$Y = CW = F,$$

donde $W = \{w_0, \dots, w_n\}^t$ son llamados pesos sinápticos y son los pesos ponderados de la matriz de activación C , donde la suma ponderada es:

$$Y = \sum_{i=0}^n w_i \cos(ix),$$

que llamamos potencial sináptico y los parámetros $F = \{f(x_0), \dots, f(x_n)\}^t$ constituyen el umbral o sesgo de la red neuronal. De aquí sabemos que una función $f(x)$ se puede expresar de la siguiente forma:

$$y(x) = \sum_{i=0}^n w_i \cos(nx) \text{ ó } Y = CW$$

donde $x \in [0, \pi]$. Por tanto, podemos definir la función objetivo ϕ en términos del error cuadrático:

$$\phi = \frac{1}{2} \|E_i^t\|^2, \quad (1)$$

donde $E^k = F - Y^k$, es el vector error entre la red neuronal y la salida actual, y F es la salida deseada. Por ende, lo que deseamos minimizar es:

$$\min(\phi(W)), \text{ donde } \phi(W_i) = \frac{1}{2} \|E_i^t\|^2 \quad (2)$$

Ajuste de la Función

Una función cualquiera se puede aproximar como una combinación de funciones cosenos [2]. Por tanto, para aproximar la integral definida usando RNAs, recurriremos a dos fases en todos los algoritmos basados en PSO con que experimentaremos, excepto por el último, en el cual la parte del cálculo de la integral se incluirá como una restricción de un problema que definiremos más adelante.

En la primera de estas fases se utiliza una red neuronal con aprendizaje supervisado para ajustar los pesos w_i de la combinación lineal siguiente:

$$y(x) = \sum_{i=0}^n w_i \cos(ix) \quad (3)$$

Dado que el ajuste de la función se hace en un intervalo $[a, b]$, cuando este intervalo es diferente que el intervalo $[0, \pi]$, se hace un cambio de variable $f(y) = f(a + \frac{b-a}{\pi}x)$, para mejorar el ajuste de la función. Este cambio de variable no es necesario si el intervalo $[a, b]$ está contenido en el intervalo $[0, \pi]$ (ver Figura 2).

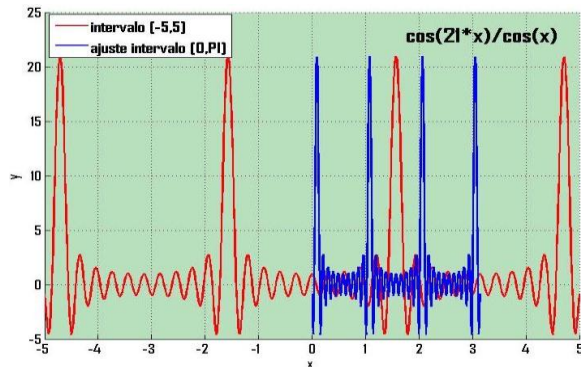


Fig. 2. Cambio de Variable

En este trabajo se realizaron diferentes versiones del PSO. La versión 1 es el algoritmo clásico del PSO [12]. La versión 2 y la versión 3 son para mejorar la velocidad de convergencia del algoritmo clásico del PSO [16]. La versión con gradiente descendiente [3,18] y la versión con gradientes conjugados [14] se utilizan para acelerar mucho más la convergencia del algoritmo del PSO. Finalmente una versión con restricción [16], la cual está sujeta a la diferencia de la integral real y la integral aproximada. Esto se realizó como parte de nuestro estudio a fin de poder determinar con cuál se obtenían mejores resultados.

A continuación se presenta el algoritmo del PSO para entrenar la RNA.

Optimización mediante cúmulos de partículas

El algoritmo de PSO fue desarrollado por Kennedy y Eberhart en 1995, basado en el comportamiento de los cúmulos de peces y las parvadas de aves. El algoritmo de PSO es más simple que otros algoritmos evolutivos, ya que, en su versión original, no utiliza operadores de cruce ni de mutación. Simplemente se realizan ajustes en las velocidades de cada partícula con base en el comportamiento de sus vecinos y de toda la población (o cúmulo) [12].

En nuestro caso utilizaremos vectores de valores reales para representar cada partícula. Este algoritmo busca en el espacio de la función objetivo para ajustar las trayectorias de las partículas, cuyas direcciones son formadas por tramos de vectores posicionales de una manera estocástica. Las partículas se mueven conforme a sus dos mejores componentes: uno estocástico y uno determinista. La partícula es atraída hacia la posición del mejor global actual G^* y sobre su mejor posición local W_i^* en la generación actual, mientras tiende al mismo tiempo a moverse aleatoriamente. Cuando una partícula encuentra una localización que es mucho mejor que cualquier otra posición encontrada, ésta se actualiza como el nuevo mejor actual de la partícula i . Entonces ésta se vuelve la mejor de las n partículas de su vecindario, y posteriormente se determina si es mejor que la mejor global.

Algoritmo clásico de PSO

Inicio

Función Objetivo $\phi(W)$, $\phi = \frac{1}{2} \|E_i^t\|^2$, donde

$$E^k = F - Y^k,$$

Posición Inicial W_i y velocidad inicial V_i de las n partículas

Mínimo inicial $\phi_{\min}^{t=0}(W)$

Mientras ($t < \text{numGeneraciones}$)

$t = t + 1$

Para todas la n partículas y las p dimensiones

Generar nueva velocidad de las partículas V_i^{t+1}

Calcular nuevas posiciones $W_i^{t+1} = W_i^t + V_i^{t+1}$

Evaluar la función objetivo de las nuevas posiciones W_i^{t+1}

Encontrar el nuevo mínimo actual $\phi_{\min}^{t+1}(W)$

Fin Para

Encontrar el mejor actual W_i^* y el mejor actual global G^*

Fin Mientras

Salida de los resultados W_i^* y G^*

Fin

A continuación se describen las diferentes versiones de PSO con las que experimentamos. La diferencia entre ellas radica únicamente en la forma de calcular el vector de la velocidad y el vector de posiciones de las partículas.

RNA PSO versión 1

Sean W y V_i el vector de posición y de velocidad, respectivamente. El nuevo vector de velocidad es determinado por la siguiente fórmula [12]:

$$V_i^{t+1} = V_i^t + \alpha \varepsilon_1 \odot [G^* - W_i^t] + \beta \varepsilon_2 \odot [W_i^* - W_i^t] \quad (4)$$

donde $\varepsilon_1, \varepsilon_2$ son dos vectores aleatorios cuyos valores están entre 0 y 1. α y β son los parámetros de entrenamiento o constantes de aceleración, modificadas conforme a la ecuación (9). La nueva posición se actualiza usando:

$$W_i^{t+1} = W_i^t + V_i^{t+1} \quad (5)$$

RNA PSO versión 2

El algoritmo clásico usa como raíz el mejor global actual G^* y el mejor individuo W_i^* . La razón por la que se usa al mejor individuo es principalmente para incrementar la diversidad de las soluciones. Una versión simplificada puede aumentar la convergencia usando sólo el mejor global. El vector de velocidad está

dado en este caso por la siguiente fórmula, donde $\frac{1}{2}$ se adopta por conveniencia [16].

$$V_i^{t+1} = V_i^t + \alpha \left[\varepsilon - \frac{1}{2} \right] + \beta [G^* - W_i^t] \quad (6)$$

La nueva posición se actualiza usando (5)

RNA PSO versión 3

Con el fin de acelerar aún más la convergencia, podemos escribir la actualización del vector de posición en un simple paso [16]:

$$W_i^{t+1} = (1 - \beta)W_i^t + \beta G^* + \alpha (\varepsilon - 0.5) \quad (7)$$

Esta versión da el mismo orden de convergencia con respecto al anterior. Las constantes de aceleración son modificadas de la misma forma que el caso anterior.

RNA PSO usando Gradiente Descendiente

Existen muchas formas en las que podemos extender la versión original del algoritmo de PSO, de tal manera que se aprovechen las direcciones que produce el gradiente del individuo con la mejor aptitud. La nueva posición se actualiza en este caso con [3,18]:

$$W_i^{t+1} = W_i^t + \eta C (F - (CG^*)) \quad (8)$$

Esto nos indica que la variación de la partícula W_i^{t+1} es proporcional a la suma del vector de pesos W_i^t anterior y el producto del error cuadrático con la matriz de activación en la generación t , con una constante de proporcionalidad η positiva, donde η está acotada como:

$$0 < \eta < \frac{2}{\|C\|^2} \quad (9)$$

RNA PSO con Gradientes Conjugados

Las aplicaciones de las Redes Neuronales Artificiales Supervisadas a menudo suponen ajustar varios cientos de pesos. El uso de Gradientes Conjugados es una buena opción para manejar de manera efectiva problemas a gran escala, por lo que se usa este método para entrenar nuestra red neuronal. Por tanto, para minimizar ϕ (1), actualizaremos la partícula W_i^t según una regla de Gradientes Conjugados [14]:

$$W_i^{t+1} = W_i^t + \alpha_i^t p_i^t \quad (10)$$

donde p_i^t es una dirección de descenso y α_i^t es el aprendizaje adaptable con respecto a t generaciones. Podemos describir el algoritmo de Gradientes Conjugados como sigue [3,18]:

$$p_i^t = C(F - Y_i^t), \quad g_i^t = -p_i^t,$$

$$\alpha_i^t = 0.5 \frac{C(F - (CG^*))}{\|g_i^t\|^2}$$

La nueva posición se actualiza usando:

$$W_i^{t+1} = W_i^t + \alpha_i^t p_i^t$$

RNA PSO con Restricción

Esta versión sólo puede aplicarse si se conoce la integral real de la función que se quiere aproximar numéricamente. Este método consiste en implementar directamente una transformación a un problema sin restricciones [16].

Esta implementación verifica que todas las nuevas partículas satisfagan las restricciones. La función objetivo que vamos a minimizar es:

$$\phi = \frac{1}{2} \|E_i^t\|^2$$

Y está sujeta a la siguiente restricción:

$$\Upsilon(W_i^t) = I_{Real} - I_{Aprox} = 0,$$

donde I_{Real} es la integral real y I_{Aprox} es la integral aproximada definida más adelante. Esto se transforma a un problema sin restricciones usando el método de penalización de multiplicadores de Lagrange, usando un parámetro de penalización $\lambda \gg 1$, por lo que la función objetivo queda de la siguiente manera

$$\phi = \frac{1}{2} \|E_i^t\|^2 + \lambda \Upsilon(W_i^t)^2 \quad (11)$$

Esta restricción se aplica en la versión anterior que utiliza gradientes conjugados.

Integración del Ajuste de la Función

En la segunda fase se obtiene la integral definida de la siguiente forma:

$$\int_a^b f(x) dx = \int_a^b \sum_{i=0}^N w_i \cos(ix) dx$$

De acuerdo con el cambio de variable, nos regresamos del intervalo $[0, \pi]$ al intervalo $[a, b]$, entonces la integral de este ajuste queda como [3,18]:

$$I = \int_a^b f(x) dx \approx (b-a) w_i + \sum_{i=1}^N \frac{b-a}{i} w_i [\sin(i\pi)] \quad (14)$$

Pruebas y Resultados

Se presentan los resultados obtenidos en dos partes. En la primera parte, se presentan las pruebas realizadas a funciones continuas y con singularidad para ajustar la RNA. En la segunda parte, se presentan pruebas de una aproximación de Factores Franck-Condon de Morse para una molécula diatómica con números cuánticos vibracionales grandes. Se comparan con resultados previamente reportados en la literatura especializada.

Las RNAs fueron implementadas en MatLab y en Java en una PC con un procesador Core2Duo, corriendo a 2GHz, y con 4GB en RAM. Las funciones oscilantes de esta prueba son las siguientes [11]:

- $\int_0^{\frac{\pi}{2}} \sin(100 \cot(x)) \sin(2x) dx = \frac{100\pi}{2} e^{-100}$
- $\int_0^{\pi} x \tan(x) dx = -\pi \ln(2)$
- $\int_0^{\pi} \frac{\cos(21x)}{\cos(x)} dx = (-1)^{10} \pi$
- $\int_0^{\pi} \frac{\cos(201x)}{\cos(x)} dx = (-1)^{100} \pi$
- $\int_0^{\pi} \frac{\sin(3x)}{\cos(x)} dx = 2 \sin^2(x) - 4 \ln(\cos(x))$

Las siguientes gráficas muestran la convergencia de los algoritmos de manera general, ya que para las funciones la convergencia es de manera similar.

Los parámetros utilizados usados son los siguientes:

- Tamaño de la Población = 15
- Longitud de la Partícula = 201

- Número Máximo de Generaciones = 1000

Gráfica de convergencia para la versión 1 (4).

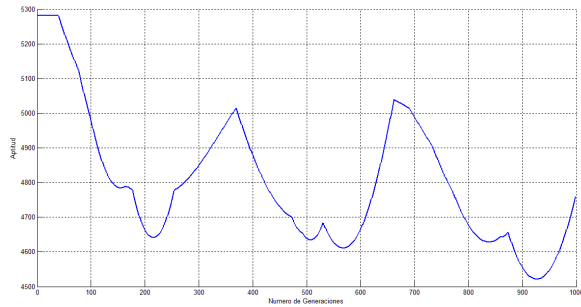


Fig. 3. Gráfica de Convergencia de RNA PSO versión 1

Gráfica de convergencia para la versión 2 (6).

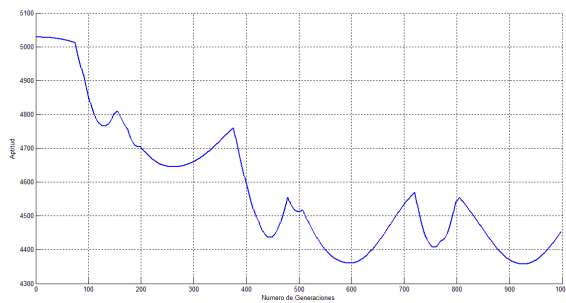


Fig. 4. Gráfica de Convergencia de RNA PSO versión 2

Gráfica de convergencia para la versión 3 (ver ecuación 7).

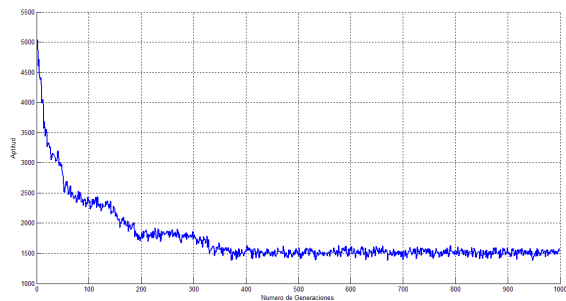


Fig. 5. Gráfica de Convergencia de PSO versión 3

Los parámetros utilizados usados son los siguientes:

- Tamaño de la Población = 15
- Longitud de la Partícula = 201
- Número Máximo de Generaciones = 2000

Gráfica de convergencia para la versión con Gradiente Descendiente (8).

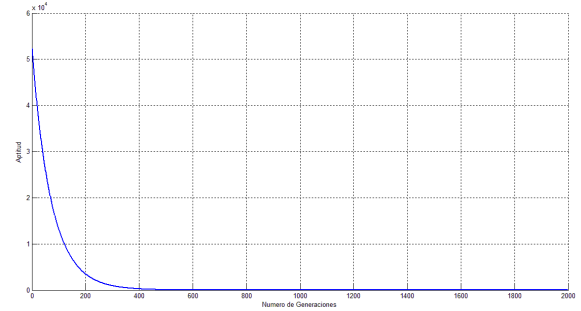


Fig. 6. Gráfica de Convergencia de RNA PSO usando Gradiente Descendiente

Los parámetros utilizados usados son los siguientes:

- Tamaño de la Población = 15
- Longitud de la Partícula = 201
- Número Máximo de Generaciones = 100

Gráfica de convergencia para la versión con Gradiente Conjugado (9).

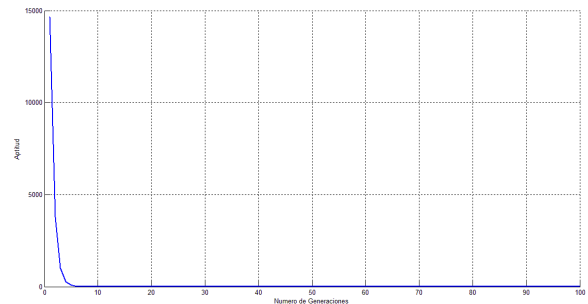


Fig. 7. Gráfica de Convergencia de RNA PSO con Gradientes Conjugados

Gráfica de convergencia para la versión con Restricción (10).

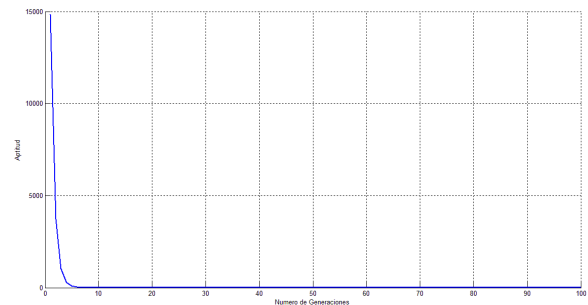


Fig. 8. Gráfica de Convergencia de RNAPSO con Restricción

Factores Franck-Condon

Una aplicación importante en la astrofísica es detectar las moléculas interestelares. Para ello, requerimos calcular los “Factores Franck-Condon” (FFC) para

moléculas diatómicas, los cuales están definidos de la siguiente forma:

El principio de Franck-Condon nos permite hacer predicciones bastantes exactas de dónde son altamente probables las transiciones vibracionales entre un par de estados electrónicos. La enunciación de este principio es el siguiente:

El salto de un electrón en una molécula se realiza tan rápidamente en comparación con el movimiento vibracional que inmediatamente después, el núcleo aún tiene la misma posición relativa y velocidad que tenía antes del salto.

En resumen, la probabilidad de transición entre dos estados, caracterizados por las eigenfunciones $\Psi_{v'}$ y $\Psi_{v''}$, es proporcional al cuadrado de los elementos correspondientes de la matriz del momento electrónico (o momento de transición):

$$R = \left| \int \Psi_{v'} M \Psi_{v''} dr \right|^2$$

A esta integral se le conoce como integral de traslape vibracional. Por tanto, el cuadrado de la integral sobre el producto de las eigenfunciones vibracionales de los dos estados involucrados (base y excitación), viene dado por:

$$\left| \int \Psi_{v'} \Psi_{v''} dr \right|^2$$

Los cuadrados de las integrales de traslape vibracional son conocidos como los Factores Franck-Condon.

Aplicando el potencial de Morse para las eigenfunciones vibracionales, tanto para el estado base como para el estado excitado, que intervienen en los Factores Franck-Condon se resume como:

$$FFC = \left| \int \Psi_{v'}^* \Psi_{v''} dr \right|^2$$

$$\text{donde } \Psi_v(r) = N_v e^{-\frac{z}{2}} z^{\frac{b}{2}} L_{k,v}(z),$$

$$k = \frac{4\pi(2\mu D)^{\frac{1}{2}}}{ah} = \frac{w_e}{w_e x_e}, \quad z = k e^{-a(r-r_e)},$$

$$b = k - 2v - 1.$$

N_v es la constante de normalización para el estado vibracional v , y $L_{k,v} = L_v^k$ son los polinomios generalizados de Laguerre.

Estas integrales para el potencial de Morse no se pueden obtener analíticamente, por lo que se aproximan numéricamente. En 1992, en la Escuela de Ciencias Físico-Matemáticas de la BUAP, se presentó una tesis desarrollada para aproximar los FFC numéricamente usando el Método de Simpson Adaptivo [1]. Con este software se han aproximado los FFC, lo cual ha permitido identificar las componentes de los cometas observados [4-10][13].

Probaremos un ejemplo con un Sistema de bandas $B^1\Pi_u \rightarrow X^1\Sigma_g^+$ del 7Li_2 , comparado con resultados experimentales usando un intervalo de integración de $[0.4, 4.5]$.

Los datos experimentales con los que se hace la comparación del sistema de bandas de esta molécula [15], son los siguientes:

	$B^1\Pi_u$	$X^1\Sigma_g^+$
w_e	269.69	351.43
$w_e x_e$	2.744	2.592
r_e	2.936	2.672
μ	3.50908	

Tabla 1. Datos Experimentales

Los valores siguientes corresponden a datos experimentales [13].

$v' v''$	11	12	13	14	15	16
6	0.1892	0.1052	0.0339	0.0070	0.0010	0.0001
7	0.0713	0.1871	0.1470	0.0602	0.0152	0.0026
8	0.0495	0.0150	0.1485	0.1779	0.0960	0.0303
9	0.0372	0.0905	0.0013	0.0845	0.1833	0.1368
10	0.0679	0.0017	0.0913	0.0328	0.0237	0.1523

Tabla 2. Datos Experimentales Sistema de bandas $B^1\Pi_u \rightarrow X^1\Sigma_g^+$

del 7Li_2 .

Los datos obtenidos con la Red Neuronal Newton son los siguientes [3]:

$v' v''$	11	12	13	14	15	16
6	8.2e-3	2.4e-3	5.9e-3	3.2e-6	2.8e-3	1.0e-4
7	1.8e-2	5.2e-3	4.0e-3	4.3e-3	8.9e-4	2.7e-3
8	1.2e-2	6.4e-3	7.4e-3	7.1e-4	3.4e-3	8.0e-5
9	2.1e-2	1.4e-2	4.8e-3	8.0e-3	1.2e-3	4.0e-3
10	4.4e-2	2.7e-3	1.8e-2	2.4e-4	6.8e-3	3.2e-5

Tabla 3. FFC-RNA Newton Sistema de $B^1\Pi_u \rightarrow X^1\Sigma_g^+$ del 7Li_2 .

Los datos obtenidos con la Red Neuronal entrenada con PSO usando Gradientes Conjugados son los siguientes:

$v' v''$	11	12	13	14	15	16
6	1.8e-1	8.7e-2	2.4e-2	4.1e-3	4.6e-4	3.5e-5
7	1.0e-1	1.9e-1	1.2e-1	4.1e-2	8.3e-3	1.0e-3
8	2.3e-2	4.7e-2	1.8e-1	1.5e-1	6.4e-2	1.5e-2
9	7.22e-2	6.4e-2	8.3e-3	1.4e-1	1.8e-1	9.3e-2
10	4.3e-2	2.9e-2	9.4e-2	1.2e-3	9.3e-2	1.8e-1

Tabla 4. RNA PSO con Gradientes conjugados en el Sistema de bandas $B^1\Pi_u \rightarrow X^1\Sigma_g^+$ del 1Li_2

Discusión de resultados

Se realizó un conjunto de funciones que tiene singularidad para comparar la convergencia de las diferentes versiones realizadas al algoritmo PSO.

Las primeras tres versiones del algoritmo presentan una convergencia muy lenta (ver Figura 3, 4 y 5), requiriéndose un gran número de generaciones (más de 1000). Adicionalmente, los valores obtenidos por estos algoritmos quedan muy lejos de los resultados óptimos. Esto motivó que se implementaran otros dos algoritmos para acelerar la convergencia del algoritmo.

En la primera versión para acelerar la convergencia nos aprovechamos de las direcciones que produce el gradiente como la mejor aptitud (ver Figura 6), produciendo mejores resultados que las versiones anteriores (se converge en 500 generaciones).

En la otra versión se aprovecharon las direcciones que se producen de manera conjugada utilizando el vector error, el cual se toma como la mejor aptitud. Esta técnica nos evita hacer el cálculo de la matriz Hessiana que es requerida en una técnica de Gradientes Conjugados clásica implementada anteriormente [3,18]. Este algoritmo acelera significativamente la convergencia (ver Figura 7) con respecto a los algoritmos anteriores (se converge en no más de 10 generaciones).

La última es una variante de la versión anterior, debido a su convergencia significativa, donde solo se puede aplicar si se conoce la integral real de la función que se quiere aproximar numéricamente (ver Figura 8).

Se probó, la metodología discutida aquí, con una molécula diatómica donde se conocen los valores experimentales (ver Tabla 2), donde las RNA's entrenadas con algoritmos clásicos fallaron (ver Tabla 3). Utilizando el algoritmo PSO con gradientes

conjugados y además se corrigen los intervalos de integración a $[0.4, 4.5]$ se mejora la aproximación de estas integrales. Como puede observarse en la Tabla 4, se obtuvieron mejores resultados que otros métodos para calcular los FFC al compararse con los datos experimentales.

Conclusiones

Se desarrollaron seis técnicas para aproximar numéricamente integrales definidas usando un enfoque de Redes Neuronales Artificiales Supervisadas utilizando el algoritmo de PSO para entrenar a la red neuronal. Para aproximar la integral de la función $f(x)$ en el intervalo $[a, b]$ se aplica el método que consiste en dos etapas. En la primera etapa, se ajusta a la función como una combinación lineal de las funciones coseno:

$$f(x) = \sum_{i=0}^N w_i \cos(ix)$$

Posteriormente, se integra la combinación lineal de las funciones coseno que se obtiene al entrenar a red neuronal, la cual es inmediata.

$$\int_a^b f(x) dx = \int_a^b \sum_{i=0}^N w_i \cos(ix) dx$$

Finalmente, se creó una aplicación en Java donde se implementaron las diferentes variantes para entrenar las Redes Neuronales que permite la aproximación de funciones de manera general y un apartado para el cálculo de los FFC para el potencial de Morse.

Agradecimientos

El primer autor agradece el apoyo de CONACyT para la realización de estudios de posgrado en el Departamento de Computación del CINVESTAV-IPN.

El segundo autor agradece el financiamiento obtenido a través del proyecto CONACyT 103570.

Finalmente la tercera autora agradece el financiamiento a la VIEP-BUAP, por el financiamiento a su proyecto de investigación.

Bibliografía

- [1] Abad, I. L. (1991). *Factores Franck-Condon para el Potencial de Morse*. Puebla, Puebla: Benemerita Universidad Autonoma de Puebla.
- [2] Burden, R. L., & Douglas Faires, J. (2002). *Numerical Analysis* (Vol. 7). Thomson Learning.

- [3] Chaman Garcia, I. (Mayo 2010). *Integración Numérica con Redes Neuronales*. Puebla: Benemérita Universidad Autónoma de Puebla.
- [4] Churyumov, K. I., Bereznoi, A., Chavushyan, V. H., andoval, L., & Palma, A. (2007). Franck-Condon Factors for Molecules Observed in Comets. 2650-2653.
- [5] Churyumov, K. I., Luk'yanyk, I. V., Bereznoi, A., Chavushyan, V. H., Sandoval, L., & Palma, A. (2002, Noviembre). Optical spectroscopy of comet C/2000 WM1 (LINEAR) at the Guillermo Haro Astrophysical Observatory in México. 90, 361-368.
- [6] Churyumov, K. I., Luk'yanyk, I. V., Bereznoi, A., Chavushyan, V. H., Sandoval, L., Palma, A., et al. (2003). Spectral Observations of comet C/2000 WM1 (LINEAR) in Mexico. 22, 1-5.
- [7] Churyumov, K. I., Luk'yanyk, I. V., Bereznoi, A., Chavushyan, V. H., Sandoval, L., Palma, A., et al. (2005). Exploration of Spectra of Periodic Comet 153P/KEYA-ZHANG. Kinematics and Physics of Celestial Bodies Suppl. 472-476.
- [8] Churyumov, K. I., Luk'yanyk, I. V., Bereznoi, A., Chavushyan, V. H., Sandoval, L., Palma, A., et al. (2005). Identification of Emission Lines in the spectrum of comet C/2002 V1 (NEAT). *Astronomical Schools's Report*, 111-114.
- [9] Churyumov, K. I., Luk'yanyk, I. V., Bereznoi, A., Chavushyan, V. H., Sandoval, L., Palma, A., et al. (2005). Peculiarities of Spectra of Comet C/2002 T7 (LINEAR) in January 2004. 2, 104-106.
- [10] Churyumov, K. I., Luk'yanyk, I. V., Bereznoi, A., Chavushyan, V. H., Sandoval, L., Palma, A., et al. (2005). Physical Parameters of the comet C/2002 C1 (Ikeya-Zhang) Atmosphere. 95-98.
- [11] Gradshteyn, I. S., & M., R. I. (2007). *Table of integrals, series and products* (Vol. 7). Elsevier.
- [12] Kennedy, J., & Eberhart, R. (2001). *Swarm Intelligence*. San Francisco: Morgan Kaufmann Publishers.
- [13] Kuzmenko, N. E., Kuznetsova, L. A., & Kuziakov, I. I. (1984). Franck-Condon factors of diatomic molecules. *Moscow, Izdatel'stvo Moskovskogo Universiteta, In Russian*, 344 p.
- [14] Luenberger, D. G. (August 1984). *Linear and Nonlinear Programming* (2 ed.). Massachusetts: Addison-Wesley Inc.
- [15] NIST Standard Reference Data is governed by the Standard Reference Data Act. . (2008). *Libro del Web de Química del NIST*. Retrieved Marzo 2010, from Base de Datos de Referencia Estándar del NIST Número 69: <http://webbook.nist.gov/chemistry/form-ser.html>
- [16] Yang, X.-S. (2008). *Nature-Inspired Algorithms*. Luniver Press.
- [17] Zeng, Z.-Z. W., & Yao-Nan, W. H. (Jul/Aug 2006). Numerical Integration Based on a Neural Network Algorithm. 8(pp. 42-48 7).
- [18] Chaman Garcia, I., Sandoval S. L. (Octubre 2011) Ajuste e Integración Numérica de Funciones muy oscilantes usando Redes Neuronales Artificiales Supervisadas. *Revista Investigación Operacional* Vol.32, No.3,pág.193-203

Datos de los Autores

Ing. Iván Christhofer Chaman García
CINVESTAV-IPN
Av. IPN No. 2508 Col. San Pedro Zacatenco
Depto. de Computación México, D.F. 07300
e-mail: cchaman@computacion.cs.cinvestav.mx

Dr. Carlos A. Coello Coello CINVESTAV-IPN
Av. IPN No. 2508 Col. San Pedro Zacatenco
Depto. de Computación México, D.F. 07300
e-mail: ccoello@cs.cinvestav.mx
Tel. 011 52 55 5747 3800 Fax: 011 52 55 5747 3757
ext. 6564 URL: <http://delta.cs.cinvestav.mx/~ccoello>

Dra. Lourdes Sandoval Solís
Facultad de Ciencias de la Computación



4to Taller Latino Iberoamericano de Investigación de Operaciones
16, 17 y 18 de Noviembre de 2011, Acapulco, Guerrero, México.

Benemérita Universidad Autónoma de Puebla
14 Sur esq. Av. San Claudio. Col. Jardines de San
Manuel, Puebla, Puebla. México.
Email: sandoval@cs.buap.mx