

طبقه‌بندی و تشخیص الگو با استفاده از شبکه عصبی فازی

حمید عزیززاده
دانشجوی کارشناسی ارشد

محمدباقر منهای
استادیار

دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

چکیده

در این مقاله، ضمن استفاده از سه نوع نرون فازی، یک نرون فازی جدید تعریف می‌گردد و بر اساس آنها یک شبکه عصبی فازی و الگوریتم ویژه جهت آموزش آن مطرح خواهد شد. برای طبقه‌بندی و تشخیص الگو، از ساختار متوالی شبکه عصبی فازی استفاده می‌شود، به این معنی که توسط FNN طبقه اول، الگوها طبقه‌بندی شده و برای هر طبقه، FNN جدیدی تشکیل می‌گردد. این ساختار در تشخیص ورودیهای شیفت داده شده و نویزی به خوبی عمل می‌کند. هنگامی که یک آرایه جدید وارد می‌شود، در هر دو طبقه از شبکه عصبی فازی، شبکه ابتدا آن را به صورت فازی درمی‌آورد و سپس میزان تشابه ورودی را به تمامی الگوها یا طبقه‌های آموزش یافته محاسبه می‌نماید. نهایتاً شبکه با انتخاب شبیه‌ترین الگو یا کلاسه آموزش یافته به ورودی، یک خروجی غیرفازی ارائه می‌دهد. جهت نشان دادن تواناییهای الگوریتم پیشنهادی، تعداد ۷۲ کاراکتر فارسی که هر یک در آرایه‌های 32×16 پیکسل جای گرفته بودند به عنوان الگوهای اصلی آموزشی استفاده شدند. در شبیه‌سازی، ابتدا ساختار متوالی را با الگوهای اصلی و شیفت داده شده به میزان ۱ بیت، آموزش دادیم، در این حال پاسخ صحیح سیستم به الگوهای اصلی برابر 100% و به ورودیهای شیفت داده شده به میزان ۱ بیت برابر 96.1% و به ورودیهای شیفت داده شده به میزان ۲ بیت، برابر 92.8% محاسبه گردید. ضمناً پاسخ سیستم به ورودیهای نویزی 93.9% بوده است.

Pattern Classification and Recognition Using a Cascaded Fuzzy Feed-forward Neural Net

M.B. Menhaj
Assist. Prof.

H. Azizzadeh
Graduate student

Electrical. Eng. Dept. Amirkabir University of Tech

Abstract

This paper presents a cascaded fuzzy Feed-forward Neural Net (CFNN) for pattern classification and recognition. By introducing a new type of fuzzy neurons, our proposed network topology consists of a multi-layer fuzzy neural net and its associated learning scheme. When an input pattern is applied to the network, the first fuzzy neural net classifies the patterns and then for each class an associated fuzzy neural net accomplishes the task of pattern recognition. Generally speaking, in each fuzzy neural net a process of fuzzification, competition and defuzzification is accomplished.

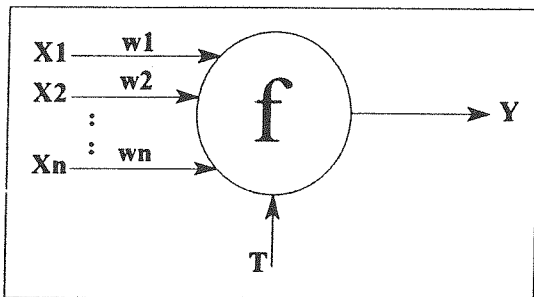
The proposed CFNN performs well even for the cases in which the shifted and noisy patterns are used as training data. To show this, the 72 Farsi characters, each represented by 32×16 pixels, were used as original training patterns. The CFNN has been trained by the original and 1-pixel shifted patterns. After the training, the CFNN recalls all of the learned patterns with 100% of recognition rate; it also recognizes, the 1-bit pixel shifted patterns with an average of 96.1% recognition rate, patterns shifted by 2 pixels with an average recognition rate of 92.8%, and noisy (distorted) pattern with 93.9% recognition rate.

چهار لایه استفاده خواهیم کرد که قادر است در طبقه‌بندی و بازشناسی الگو مورد استفاده قرار گیرد.

در بخش بعدی، نرون‌های فازی بررسی می‌شوند. در بخش سوم با استفاده از نرون‌های فازی متفاوت در لایه‌های چهارگانه، ساختار شبکه عصبی فازی (CFNN) معرفی می‌گردد. در بخش چهارم با توجه به مسأله تشخیص کاراکترهای فارسی، ساختار متوالی از (CFNN) جهت طبقه‌بندی و تشخیص الگوها ارائه می‌گردد. الگوریتم آموزش CFNN در بخش پنجم ارائه خواهد شد، این الگوریتم لایه سوم و چهارم را به طور خودکار تشکیل داده و کامل می‌کند. بخش ششم به تحلیل شبکه عصبی فازی می‌پردازد. نتایج شبیه‌سازی موضوع بخش هفتم بوده و در بخش هشتم نیز نتیجه مقاله ارائه می‌گردد.

۲- نرون‌های فازی

یک نرون غیرفازی عادی، شامل N ورودی وزن‌یافته و یک خروجی می‌باشد.



شکل (۱) نرون عادی

نرون، ورودی‌های x_i را در وزن‌های مربوط به آن (w_i) ضرب کرده و همه را با هم جمع می‌کند ($i=1$ to N) و نتیجه را به یک تابع غیرخطی f (به نام تابع فعال Activating)

۱- مقدمه

هر شبکه عصبی (Neural Network, NN)، ساختار موازی حجیمی دارد که از ترکیب اجزاء پردازشگر بسیار زیادی تشکیل شده است و تمامی این اجزاء به واسطه وزن‌هایی به مابقی اجزاء متصل هستند [۱]. شبکه‌های عصبی با توجه به سیستم اعصاب مغز ساخته شده‌اند [۲]. یک NN در واقع یک سیستم دینامیکی غیرخطی و با قابلیت آموزش بوده و پاسخ‌های سریعتر و عملکرد بهتری در برابر روشهای معمول جهت شبیه‌سازی مغز انسان از خود نشان می‌دهد. استفاده از NNها در مسأله بازشناسی الگو بارها مورد استفاده قرار گرفته‌است. Neocognitron برای بازشناسی الگوهای جابجاشده و تغییر شکل یافته استفاده شد [۳]. روش آموزش Back-Propagation در NN پاسخ‌های مناسبی در تشخیص حروف و کاراکترهای دست‌نویس از خود نشان داد [۴ و ۵]. به هر حال در اکثر روشهای فوق که بر پایه شبکه‌های عصبی کار می‌کنند، با افزایش درجه شبکه، تعداد وزن‌های آن به صورت تصاعدی افزایش می‌یابد.

از طرف دیگر، منطق فازی [۶ و ۸] ابزار قدرتمندی را برای مدلسازی تفکر و ادراک انسانی ارائه داده‌است. سیستمهای فازی از روی مجموعه‌های چند مقداره و قواعد فازی به استنتاج می‌پردازند. ثابت شده‌است که تئوری مجموعه‌های فازی نقش بارزتری را در مسائل بازشناسی الگو ایفا می‌کند.

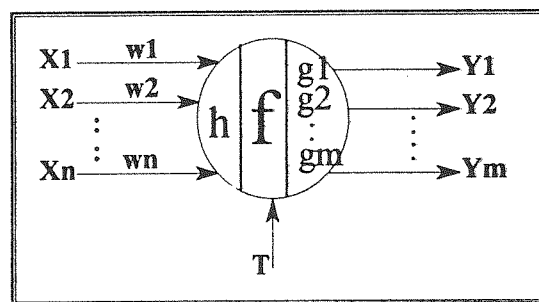
برخی‌ها از سیستمهای عصبی فازی برای بازشناسی الگو استفاده کرده‌اند [۸ و ۹]. هدف این تحقیقات، تلفیق ویژگی سیستم‌های فازی (قابلیت پردازش اطلاعات فازی با استفاده از الگوریتم‌های فازی) و خصوصیات شبکه‌های عصبی (قابلیت آموزش و ساختار موازی با سرعت بالا) و ایجاد شبکه‌های عصبی فازی می‌باشد. در این مقاله، ما ضمن بررسی تعاریف رایج در نرون‌های فازی (FN) یک نوع جدید از این نرون را تعریف کرده و از آنها برای ساختن یک شبکه عصبی فازی

(Function) اعمال می‌کند. خروجی چنین نرونی را می‌توان توسط رابطه زیر بیان نمود.

$$y = f \left[\sum_{i=1}^N w_i x_i - T \right]$$

T نقش مقدار آستانه‌ای نرون را بازی می‌کند.

یک نرون فازی شامل N ورودی وزن یافته (x_i با وزن‌های w_i $i=1$ to N و M خروجی y_j ($j=1$ to M)) می‌باشد.



شکل (۲) نرون فازی

تمامی ورودی‌ها مقادیر حقیقی بوده و خروجی‌ها اعداد حقیقی متعلق به فاصله [0,1] خواهند بود. هر خروجی، یک مقدار عضویت در مفهوم فازی می‌باشد، یعنی هر خروجی درجه تعلق آرایه ورودی $\{x_1, x_2, \dots, x_N\}$ را به یک مجموعه فازی نشان می‌دهد. روابط حاکم بر نرون فازی به صورت زیر بیان می‌شوند:

$$z = h [w_1 x_1, w_2 x_2, \dots, w_N x_N]$$

$$s = f [z - T]$$

$$y_j = g_j [s] \quad \text{for } j = 1 \text{ to } M$$

در روابط فوق z ورودی خالص (Net Input) نرون فازی می‌باشد. h تابع اجتماع (Aggregation Function)، حالت نرون فازی، f تابع فعال، T آستانه فعالیت (Activating Threshold)، $\{g_j | j=1 \text{ to } M\}$ تعداد M تابع خروجی (Output - Functions) این نرون فازی می‌باشد که ارائه‌دهنده توابع عضویت ورودی به شکل آرایه $\{x_1, x_2, \dots, x_N\}$ در تمامی M مجموعه فازی خواهند بود. در نتیجه، نرون‌های فازی می‌توانند اطلاعات فازی را توضیح داده و پردازش نمایند.

در حالت عمومی، وزن‌ها، آستانه فعالیت و توابع خروجی که روابط داخلی را در درون نرون فازی تشریح می‌کنند، می‌توانند طی عمل یادگیری تنظیم گردند. بنابراین نرون‌های فازی تطبیقی بوده و یک شبکه عصبی فازی (FNN) که شامل تعدادی نرون فازی است می‌تواند آموزش داده شود.

تابع اجتماع و تابع فعالیت از مشخصه‌های ذاتی یک نرون فازی هستند. اگر توابع متفاوتی از h و f در نرون‌ها استفاده گردد، این نرون‌ها خصوصیات و ویژگیهای متفاوتی خواهند داشت. انواع زیادی نرون‌های فازی را می‌توان با انتخاب h و f تعریف نمود.

در یک نرون فازی، وزن‌ها و مقدار آستانه می‌توانند به صورت یک تابع نیز تعریف شوند. در ادامه چهار نوع از نرون‌های فازی (FNها) را تعریف می‌کنیم:

الف - FN ورودی FN Input-FN [9]

اگر از یک FN در لایه ورودی یک FNN استفاده شود و این نرون فقط دارای یک ورودی x باشد:

$$z = x$$

آن‌گاه این FN را یک FN ورودی می‌نامیم.

ب - FN ماکزیمم FN MAX-FN [9]

اگر در تابع اجتماع از یک تابع ماکزیمم‌یاب استفاده شود و داشته باشیم:

$$z = \max_{i=1}^N (w_i x_i)$$

آن‌گاه این FN را یک FN ماکزیمم می‌نامیم.

ج - FN میانگین FN MEAN-FN

اگر در تابع اجتماع از یک تابع میانگین استفاده شود و داشته باشیم:

$$z = \frac{1}{N} \sum_{i=1}^N (w_i x_i)$$

آن‌گاه این FN را یک FN میانگین می‌نامیم.

د - FN رقابتی FN COMP-FN [9]

اگر یک FN دارای متغیر آستانه‌ای T بوده و فقط یک

خروجی داشته باشد.

$$y = g[s-T] = \begin{cases} 1 & \text{if } s \leq T \\ 0 & \text{if } s > T \end{cases}$$

$$T = t [C_1, C_2, \dots, C_k]$$

که در روابط فوق s حالت FN، t تابع آستانه و C_k متغیرهای رقابت در FN هستند، آنگاه FN فوق را رقابتی می‌نامیم.

انواع نرون‌های فازی به اینجا ختم نمی‌شوند و نرون‌های فازی به اشکال دیگر نیز قابل تعریف می‌باشند.

همانطور که بیان شد، هر نرون فازی با توابع مربوط به خود، ویژگی خاصی خواهد داشت. تعاریف فوق به دلیل نیاز ساختار مورد نظر به آنها مورد توجه بوده‌اند.

۳- ساختار شبکه عصبی فازی

شبکه پیشنهادی یک FNN دارای چهار لایه Feedforward به صورت شکل (۳) می‌باشد [۹].

الف: لایه اول لایه ورودی بوده و آرایه‌های ورودی را پذیرفته و به درون شبکه داخل می‌نماید. در این لایه، از نرون‌های ورودی (Input-FN) استفاده کرده‌ایم. هر نرون ورودی در این لایه متناظر با یک Pixel از آرایه ورودی

خواهد بود. FN‌های ورودی به صورت دوبعدی نمایش داده شده و شماره گذاری شده‌اند و تعداد FN‌ها در این لایه برابر تعداد کل Pixel‌های آرایه ورودی خواهد بود. فرض کنیم آرایه ورودی یک ماتریس $N_1 \times N_2$ باشد، آنگاه تعداد نرون‌های فازی در این لایه برابر $N_1 \times N_2$ نرون خواهد بود. الگوریتم حاکم بر نرون فازی (i,j) در لایه اول به صورت زیر است:

$$s_{ij}^{[1]} = z_{ij} = x_{ij} \quad i = 1 \text{ to } N_1 \text{ و } j = 1 \text{ to } N_2$$

$$y_{ij}^{[1]} = s_{ij}^{[1]} / X_{max} \quad i = 1 \text{ to } N_1 \text{ و } j = 1 \text{ to } N_2$$

در روابط فوق x_{ij} مقدار ورودی $m(i,j)$ از آرایه ورودی بوده و مقدار آن $x_{max} \geq x_{ij} \geq 0$ می‌تواند باشد. X_{max} حداکثر مقدار قابل تصور برای تمامی ورودیهاست. در حقیقت با عمل تقسیم به X_{max} تمامی خروجی‌های $y_{ij}^{[1]}$ بین 0 و 1 نرمالیزه می‌شوند. واضح است که اگر ورودیها بین 0 و 1 باشند، آنگاه $X_{max}=1$ خواهد بود.

ب: لایه دوم نیز به صورت دوبعدی نمایش داده شده و شامل تعداد $N_1 \times N_2$ نرون فازی ماکزیمم MAX-FN است. وظیفه این لایه، فازی کردن آرایه ورودی با استفاده از تابع وزن $w(m,n)$ می‌باشد. حالت نرون $m(p,q)$ در این لایه به صورت:

$$s_{pq}^{[2]} = \max_{i=1}^{N_1} \left[\max_{j=1}^{N_2} \left[w |p-i, q-j| y_{ij}^{[1]} \right] \right]$$

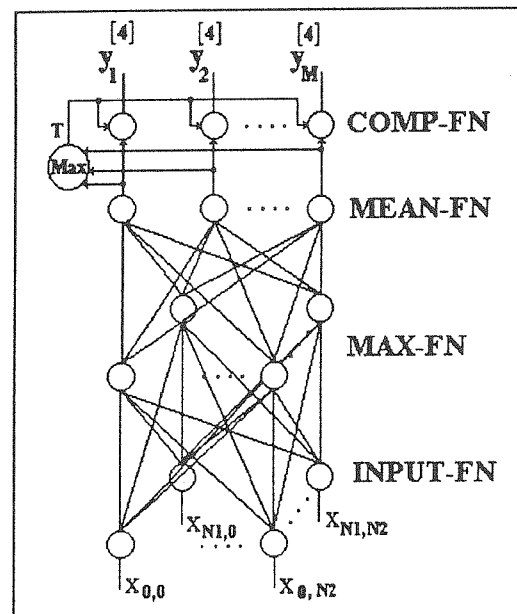
تعریف می‌شود. مقدار $w[p-i, q-j]$ وزن ارتباط دهنده FN ورودی $m(i,j)$ از لایه اول با FN ماکزیمم $m(p,q)$ در لایه دوم می‌باشد. تابع وزن به صورت زیر تعریف می‌گردد:

$$W[m,n] = \exp \left[-\beta^2 \left(m^2 + n^2 \right) \right]$$

$$m = -(N_1-1) \text{ to } (N_1-1), n = -(N_2-1) \text{ to } (N_2-1)$$

تابع وزن $w[m,n]$ ، تابع فازی کننده یا Fuzzification Function نیز نامیده می‌شود (بعبارتی وسیعتر، یک عمل فیلترینگ پایین‌گذر برای هموار نمودن نقاط مرزی انجام می‌گیرد). هر MAX-FN در این لایه تعداد M خروجی متفاوت خواهد داشت (م در M تعداد FN‌ها در لایه سوم می‌باشند). خروجی‌های FN ماکزیمم $m(p,q)$ این لایه به صورت زیر می‌باشند:

$$y_{pqm}^{[2]} = g \left[\frac{s_{pqm}^{[2]}}{p^q} \right] \quad \text{و } p=1 \text{ to } N_1, q=1 \text{ to } N_2, m=1 \text{ to } M$$



شکل (۳) ساختار شبکه عصبی فازی

فازی مقایسه‌ای را به خود اختصاص خواهد داد، بنابراین در این لایه تعداد M نرون خواهیم داشت. خروجی نرون‌ها، غیر فازی خواهند بود. اگر یک آرایه ورودی، خیلی به الگوی آموزش یافته می‌گردند. m ام شبیه باشد، آنگاه خروجی m از روابط حاکم COMP-FN برابر 1 بوده و مابقی خروجی‌ها همه بر لایه چهارم به صورت زیر می‌باشد:

$$s_m^{[4]} = z_m^{[4]} = y_m^{[3]} \quad m = 1 \text{ to } M$$

$$y_m^{[4]} = g \left[s_m^{[4]} - T \right] = \begin{cases} 0 & \text{if } s_m^{[4]} > T \\ 1 & \text{if } s_m^{[4]} \leq T \end{cases} \quad m = 1 \text{ to } M$$

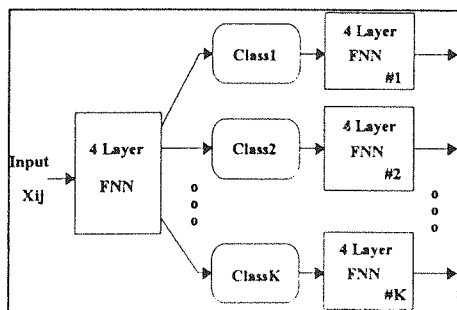
$$T = \max_{m=1}^M \left[y_m^{[3]} \right] \quad m = 1 \text{ to } M$$

که T ، آستانه فعالیت برای تمام نرون‌های رقابتی در لایه چهارم می‌باشد.

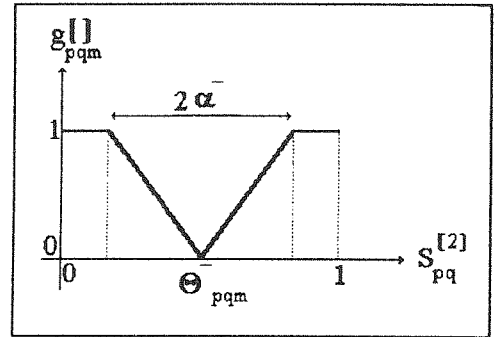
۴- ساختار متوالی جهت طبقه‌بندی و تشخیص الگو

در بخش قبلی تشریح شد که در لایه سوم از شبکه عصبی فازی خروجی‌های هر نرون فازی از این لایه (که یک عدد فازی و یا در واقع عدد عضویت مربوط به سری فازی می‌باشد) میزان عدم شباهت آرایه ورودی را به الگوهای آموزش یافته نشان می‌دهد.

پس با بیان فوق می‌توانیم از FNN جهت طبقه‌بندی الگوها استفاده کنیم. آرایه ورودی پس از عبور از سه لایه اول به یکی از الگوها (یا در واقع طبقه‌ها) شبیه‌تر بوده و جزو آن طبقه قرار داده خواهد شد و سپس توسط FNN دیگری تشخیص داده می‌شود. هر طبقه خود حاوی یک FNN چهار لایه با ضرایب و پارامترهای ویژه خواهد بود. به این ترتیب ساختار زیر که آنرا "ساختار متوالی" می‌نامیم پیشنهاد می‌گردد:



شکل (۵) ساختار متوالی



شکل (۴) تابع g

که $y_{pqm}^{[2]}$ خروجی m ام از FN ماکزیمم (p,q) می‌باشد که به FN مینیمم m ام در لایه سوم متصل می‌گردد. تابع خروجی $g_{pqm}^{[2]}$ توسط الگوریتم آموزش دهنده تعیین می‌گردد. در اینجا ما از یک تابع مثلثی معکوس استفاده کرده‌ایم. شکل تابع در زیر ترسیم شده است (شکل ۴).

تابع خروجی لایه دوم را می‌توان به صورت زیر بیان نمود:

$$y_{pqm}^{[2]} = g_{pqm}^{[2]} \left[s_{pq}^{[2]} \right] = \begin{cases} \frac{|s_{pq}^{[2]} - \Theta_{pqm}|}{\alpha} & \text{if } |s_{pq}^{[2]} - \Theta_{pqm}| \leq \alpha \\ 1 & \text{otherwise} \end{cases}$$

$$1 \geq Q_{pqm} \geq 0, 1 \geq \alpha > 0$$

$$P=1 \text{ to } N_1, q=1 \text{ to } N_2, m=1 \text{ to } M$$

در الگوریتم آموزش دهنده، مقادیر α و Θ_{pqm} برای هر سری از m, q, p تعیین خواهند شد.

ج: در لایه سوم از نرون‌های فازی میانگین و یا در لایه نشان دهنده MEAN-FN استفاده کرده‌ایم. هر MEAN-FN سوم یک الگوی آموزش یافته خواهد بود. بنابراین، تعداد فقط عمل FN های موجود در لایه سوم (M) می‌تواند پس از اتمام یادگیری معلوم گردد.

خروجی FN میانگین m ام در لایه سوم به صورت:

$$y_m^{[3]} = s_m^{[3]} = \text{mean} \left[y_{pqm}^{[2]} \right] \frac{1}{N_1 \times N_2} \left[\sum_{p=1}^{N_1} \left[\sum_{q=1}^{N_2} y_{pqm}^{[2]} \right] \right]$$

$$i = 1 \text{ to } M$$

بوده و $s_m^{[3]}$ حالت FN میانگین m ام در لایه سوم خواهد بود.

د: لایه چهارم لایه خروجی می‌باشد. ما از COMP-FN در لایه چهارم استفاده کرده‌ایم. هر الگوی آموزش یافته یک نرون

ساختار فوق دارای این مزیت است که اجزاء درونی کل سیستم یکدست بوده و پیاده کردن و اجرای عملی چنین سیستمی ساده تر خواهد بود. اگر قرار است ساختار فوق از طریق نرم افزار اجرا گردد، زیر برنامه ها و Procedure های یکسانی مورد استفاده قرار خواهند گرفت و اگر سخت افزاری را برای پیاده کردن سیستم فوق در نظر بگیریم، اجزای مشابهی برای آن طراحی خواهد شد.

در بخش گذشته بیان گردید که FNN مزایای روش های شبکه عصبی و تئوری فازی را بطور همزمان دارا می باشد. بدلیل استفاده از FNN در قسمت طبقه بندی، این قسمت نیز مزایای فوق را خواهد داشت.

در بخش گذشته بیان گردید که FNN مزایای روش های شبکه عصبی و تئوری فازی را بطور همزمان دارا می باشد. بدلیل استفاده از FNN در قسمت طبقه بندی، این قسمت نیز مزایای فوق را خواهد داشت.

در عمل مشاهده شد که اگر آموزش شبکه بطور مناسبی انجام پذیرد، "ساختار متوالی" قدرت، سرعت و دقت قابل ملاحظه ای را از خود نشان می دهد.

۵- تحلیل شبکه عصبی فازی (CFNN)

در بخش پیشین، ساختار شبکه چهار لایه عصبی فازی جهت شناسایی الگوها تشریح گردید. همانند شبکه های عصبی، این شبکه عصبی فازی یا CFNN یک سیستم موازی می باشد که تمامی اجزاء یک آرایه وزودی را بطور همزمان پردازش می کند. FNN پیشنهادی، شامل چهار نوع نرون فازی می باشد که می تواند اطلاعات فازی را تجزیه و تحلیل نموده و با استفاده از الگوریتم فازی، در مسأله شناسایی الگو استفاده گردد.

اولین لایه از شبکه، اطلاعات آرایه وزودی را پذیرفته و به درون شبکه اعمال می کند. FNN های وزودی در این اعداد Pixel های وزودی را به مقادیر بین فاصله [0,1] نرمالیزه می نمایند. لایه دوم شبکه، آرایه وزودی را به صورت اعداد فازی در درمی آورد. هر MAX-FN لایه دوم، توسط تابع وزن $w[m,n]$ به تمامی نرون های لایه وزودی متصل هستند و بزرگترین مقدار ورودیهای وزن یافته را به عنوان حالت خود در نظر می گیرند. نتیجه حاصل از استفاده MAX-FN و استفاده از وزن های فازی کننده این است که هر نقطه سیاه از آرایه وزودی توسط لایه دوم از FNN به صورت فازی درآمده است. درجه

فازی شدن و یا عدد فازی برای هر FN لایه دوم به مقدار β بستگی دارد. هر چه β کوچکتر باشد تعداد FN هایی که در لایه دوم فقط توسط یک نقطه سیاه تأثیر می پذیرند بیشتر خواهد بود. این پدیده به این معناست که مقدار β ، گستردگی فازی کردن را کنترل می کند. اگر β خیلی کوچک باشد، FNN نخواهد توانست بین برخی الگوهای کاملاً مجزا تفاوت قائل شود و اگر β خیلی بزرگ باشد، FNN قابلیت و قدرت خود را در تشخیص برخی ورودی های تغییر یافته (نسبت به الگوی آموزشی) از دست خواهد داد. β باید طوری انتخاب گردد که تمامی الگوهای آموزشی مجزا توسط FNN، متفاوت تشخیص داده شده و همچنین FNN، قدرت تشخیص قابل قبولی را در ورودیهای تغییر یافته داشته باشد. خروجی m ام از FN ماکزیمم (p,q) در لایه دوم $(y_{pqm}^{[2]})$ ، میزان عدم شباهت جزء (p,q) ام و اجزاء اطراف آن را به جزء (p,q) ام و اجزاء اطراف آن در الگوی آموزشی m ام در مفهوم فازی بیان می دارد. تابع خروجی $[S_{pq}^{[2]}]$ در واقع یک تابع عضویت برای این مجموعه فازی (الگوی m ام) بوده و باید شامل اطلاعات اجزاء اطراف جزء (p,q) ام از الگوی آموزشی m ام باشد. Θ_{pqm} دقیقاً به معنای فوق اشاره دارد.

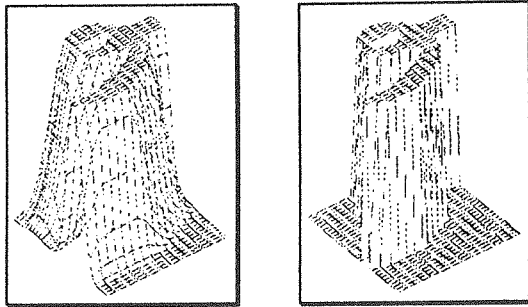
FN های موجود در لایه سوم میزان عدم شباهت آرایه وزودی به تمامی الگوهای آموزشی را به دست می دهند. ما از MEAN-FN ها در لایه سوم استفاده کرده ایم و میزان عدم شباهت آرایه وزودی $X = \{x_{ij}\}$ به الگوی آموزشی m ام توسط FN میانگین m ام در لایه سوم از FNN محاسبه خواهد شد.

$$y_m^{[3]} = \begin{cases} \text{mean}_{pq} \left[\frac{|y_m^{[2]} - \Theta_{pqm}|}{\alpha} \right] & \text{if } \max_{pq} \left[|y_m^{[2]} - \Theta_{pqm}| \right] \geq \alpha \\ 0 & \text{if otherwise} \end{cases} \quad m = 1 \text{ to } M$$

که در رابطه فوق $[y_{pq}^{[2]}]$ خروجی FN ماکزیمم (p,q) ام در لایه دوم می باشد. می توان ملاحظه نمود که α نوعی متغیر "وسعت دید" برای شبکه بوده و مقدار α روی محاسبه عدم تشابهات تأثیر می گذارد. وقتی که آرایه وزودی برابر یکی از الگوهای آموزشی باشد، تنها یک $y_m^{[3]}$ متشابه خواهیم داشت که برابر 0 می شود، و اگر آرایه وزودی هیچیک از الگوهای آموزشی نباشد، تمامی $y_m^{[3]}$ ها بزرگتر از 0 بوده و $y_m^{[3]}$ که از نظر FNN بیشترین شباهت را به آرایه وزودی داشته باشد، کمترین عدد را

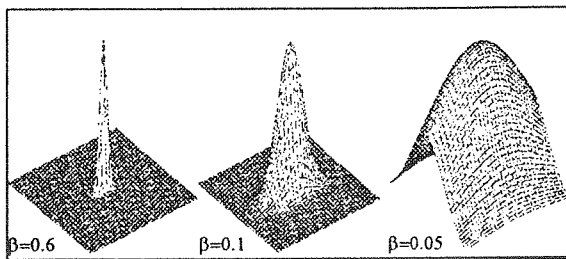
خواهد داشت (کمترین عدم تشابه).

الگوریتم فازی کردن، مرکز این عدسی را روی تک تک نقاط ماتریس ورودی قرار می‌دهد و از درون این عدسی به آن نگاه می‌کند، مسلماً غیر از خود نقطه، اطراف آن نیز قابل مشاهده خواهد بود. بطور مثال به شکل‌های (۷) که الگوی خاص را به صورت عادی و بعد از فازی شدن نشان می‌دهد توجه نمایید:



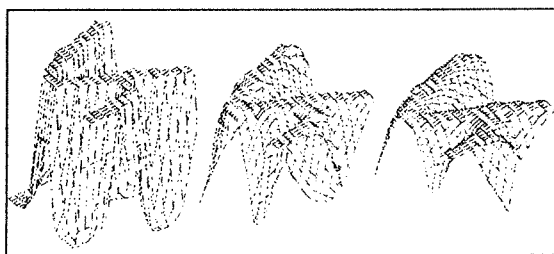
شکل (۷) شکل معمولی و فازی شده

تغییر مقدار β بر روی ضخامت و باریکی لوز مورد بحث تأثیر می‌گذارد. شکل (۸) تصویر w را به ازاء مقادیر مختلف β نشان می‌دهد:



شکل (۸) تابع فازی‌کننده با مقادیر مختلف β

فازی کردن الگوی (۸) توسط تابع وزن‌های فوق به صورت شکل (۹) خواهد بود.



شکل (۹) تأثیر β

لایه خروجی از FNN برای انجام عمل غیرفازی کردن و رسیدن به خروجی مطلق استفاده می‌شود. این لایه کمترین عدم تشابه (بیشترین تشابه) را به عنوان آستانه فعالیت برای تمامی COMP-FN ها در لایه چهارم انتخاب می‌کند. اگر m لایه دارای کمترین مقدار عددی بین FN های لایه سوم باشد، آنگاه خروجی FN رقابتی m در لایه چهارم برابر 1 بوده و خروجی مابقی COMP-FN برابر 0 خواهند بود.

پس بطور خلاصه، پروسه تشخیص در FNN پیشنهادی در چهار مرحله انجام می‌گردد: ورود اطلاعات (لایه اول)، فازی کردن اطلاعات (لایه دوم)، استنتاج فازی (لایه سوم)، غیرفازی کردن (لایه چهارم).

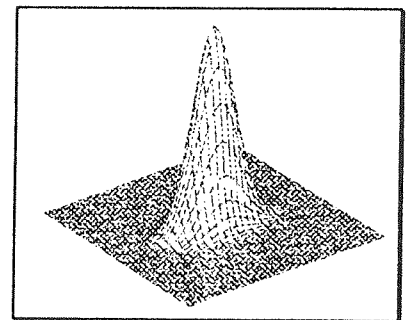
برای داشتن حالت عمومی و نیز ایجاد انعطاف پذیری در این سیستم، پارامترهایی در آن پیش‌بینی شده‌است که با تغییر آنها، رفتار سیستم و پاسخ آن در قبال یک ورودی ثابت تغییر خواهد نمود. به عبارت دیگر، متغیرهایی در دسترس هستند که با تنظیم آنها می‌توان به بهترین پاسخ برای یک نوع خط خاص دست یافت. پارامترهای فوق را یکبار دیگر مرور می‌کنیم:

الف - β (beta): این متغیر میزان تأثیر Pixel ها را بر روی نرون‌های فازی اطراف خود تعیین می‌کند. ما از یک تابع وزن به نام w جهت Fuzzification الگوی ورودی استفاده می‌کنیم که فرمول آن به صورت زیر می‌باشد:

$$w(m,n) = \exp \left[-\beta^2 \left[m^2 + n^2 \right] \right]$$

$$m = -(N_1-1) \text{ to } (N_1-1), n = -(N_2-1) \text{ to } (N_2-1)$$

اگر شکل این تابع را به صورت سه بعدی مشاهده کنیم، خواهیم دید که شبیه یک عدسی دورین می‌باشد (شکل ۶).



شکل (۶) تابع β

آمده است. پس از آموزش و هنگام استفاده از شبکه آموزش یافته، آرایه ورودی به یک FNN اعمال شده و در نهایت به یکی از طبقه‌های 1 تا K تعلق می‌گیرد. هر طبقه توسط N الگوی مجزا آموزش یافته است و بنابراین، آرایه ورودی پس از اعمال به FNN چهار لایه مربوط به طبقه خود، به یکی از N الگوی آموزشی شبیه‌تر تشخیص داده شده و خروجی مربوطه فعال می‌گردد و عمل تشخیص پایان می‌یابد.

با روش فوق و با توجه به محدودیت حافظه در کامپیوتر مورد استفاده در می‌یابیم که تعداد طبقه‌ها یعنی K و نیز تعداد الگوهای هر کلاس یعنی N هر دو باید محدود باشند و در واقع حاصلضرب N×K از یک مقدار حداکثر نمی‌تواند بیشتر باشد. آموزش شبکه عصبی فازی در دو مرحله انجام می‌پذیرد. در مرحله اول طبقه‌ها آموزش می‌یابند و در مرحله دوم FNN هر طبقه آموزش می‌یابند. هنگام آموزش، الگوهای آموزش‌دهنده در هر دو مرحله مشترک خواهند بود.

مرحله اول: آموزش طبقه‌ها

۱- عدد K "حداکثر تعداد طبقه‌ها" و مقدار β_1 و α_1 را تعیین کنید. توضیح اینکه تابع وزن $w(m,n)$ با استفاده از β_1 بدست می‌آید.

۲- Q "تعداد طبقات آموزش یافته" را برابر صفر قرار دهید.

۳- پارامتر Θ را برای الگوی ورودی بعدی به صورت زیر محاسبه کنید:

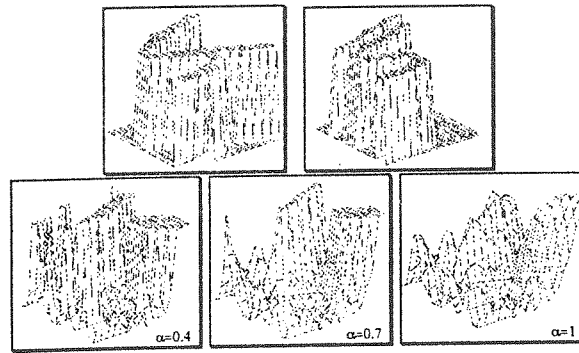
$$\Theta = \Theta_{pq} = \max_{i=1}^{N_1} \left[\max_{j=1}^{N_2} \left[w [p-i, q-j] x_{ij} \right] \right]$$

$p=1 \text{ to } N_1, q=1 \text{ to } N_2$

۴- $O = O+1$ ، اگر $O \geq K$ ، آنگاه $\Theta = \Theta_0$ قرارداد و به مرحله ۳ بروید و اگر نه به مرحله ۵ بروید.

۵- چون به این مرحله آمده‌اید بنابراین تمامی طبقه‌ها پر هستند. پس شبیه‌ترین دو طبقه و میزان شباهت این دو را به یکدیگر به روش زیر بیابید:

$$y_{mn}^{[3]} = \begin{cases} \max_{pq} \left[\frac{|\Theta_{pqm} - \Theta_{pqn}|}{\alpha_1} \right] & \text{if } \max_{pqm, pqn} \left[|\Theta_{pqm} - \Theta_{pqn}| \right] \geq \alpha_1 \\ 0 & \text{otherwise} \end{cases}$$



شکل (۱۰) تأثیر α

ب - α (Alpha): در شبکه عصبی فازی، تابع خروجی لایه دوم یک تابع مثلثی معکوس می‌باشد که ضمن محاسبه میزان اختلاف بین تک تک نقاط الگوهای فازی شده با آرایه فازی شده ورودی، اختلاف‌هایی را که از یک حد بیشتر باشد به ماکزیمم مقدار ممکن تبدیل می‌کند. شکل‌های (۱۰)، خروجی لایه دوم را برای یک آرایه ورودی و یکی از الگوها به ازاء مقادیر مختلف α نشان می‌دهند.

ج - تعداد طبقه‌ها K و تعداد الگوها N: در ساختار متوالی، دو شبکه عصبی فازی جهت طبقه‌بندی و تشخیص الگوها به کار رفته است. همانطور که در بخش آموزش شبکه توضیح داده شد، حداکثر تعداد طبقه‌بندی و حداکثر تعداد الگوهای تشخیص برای هر طبقه جزو پارامترهایی است که باید به الگوریتم آموزش داده شوند. این دو عدد با رابطه $K \times N = \text{Const}$ به هم وابسته هستند و مقدار ثابت را حجم حافظه کامپیوتر تعیین می‌کند. با تغییر این پارامترها به نتایج مختلفی خواهیم رسید که در بخش بعدی به آنها خواهیم پرداخت. در ضمن در بخش بعدی در بخش آموزش FNN، ملاحظه خواهید نمود که آموزش FNN می‌تواند یک پروسه پیوسته باشد. به معنای دیگر قبل از آموزش دوباره FNN می‌توان از آن برای تشخیص آموخته‌های قبلی خود استفاده نمود و در فرصت‌های دیگر، با کمترین لطمه به آموخته‌های قبلی، الگوهای جدیدی را به FNN آموزش داد. این امر شباهت زیادی به قدرت یادگیری مغز انسان دارد!

۶- آموزش شبکه عصبی فازی CFNN

ساختار پیشنهادی برای شبکه عصبی فازی در بخش پیشین ارائه گردید و تصویر بلوک دیاگرام آن نیز در شکل (۵)

۷- شبیه‌سازی

تعداد ۷۲ کاراکتر فارسی انتخاب شدند:

آ ا ب پ ت ث ج ج چ ح خ د ذ ر ز س ش ص ض
ض ط ظ ع ص ع غ غ ف ف ق ک ک گ گ ل م ن و ه ه ه ه ی
ی ۹ ۸ ۷ ۶ ۵ ۴ ۳ ۲ ۱ ۰

هر یک از کاراکترهای فوق را در یک آرایه 32×16 بیکسل جای دادیم.

الگوهای به شکل فوق به عنوان الگوهای اصلی مورد استفاده قرار می‌گیرند. ضمناً هر الگو با حرکت در جهات هشت‌گانه به میزان ۱ پیکسل و یا ۲ پیکسل، الگوی اغتشاش‌یافته‌ای را نسبت به الگوی اصلی ایجاد می‌کند.

شبیه‌سازی شبکه عصبی فازی CFNN در یک کامپیوتر 486-33 به زبان اسمبلی و UniComal انجام پذیرفت. ۷۲ الگوی اصلی و شکل‌های شیفت‌داده‌شده به میزان ۱ بیت برای آموزش شبکه عصبی فازی CFNN استفاده شدند و برای آزمایش FNN از الگوی اصلی، شیفت‌داده‌شده به میزان ۱ بیت، شیفت‌داده‌شده به میزان ۲ بیت و نویزی به میزان ۱۵٪ مورد استفاده قرار گرفتند.

جداول (۱) تا (۳) میزان صحت پاسخ‌دهی CFNN را به ازای مقادیر مختلف α ، β و K نشان می‌دهند:

گروه الف	الگوهای اصلی
گروه ب	الگوهای شیفت‌داده‌شده به میزان ۱ بیت
گروه ج	الگوهای شیفت‌داده‌شده به میزان ۲ بیت
گروه د	الگوهای نویزی

جدول (۱) $\alpha = 0.9$ ، $N = 20$ ، $K = 60$

β	۰/۲	۰/۳	۰/۴	۰/۵	۰/۶
گروه الف	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰
گروه ب	۹۴/۲	۹۴/۳	۹۴/۱	۹۲/۸	۹۰/۹
گروه ج	۸۵/۵	۸۵/۴	۸۶/۳	۸۵/۴	۸۲/۷
گروه د	۸۰/۳	۸۱/۳	۸۰/۹	۷۹/۷	۷۸/۹

otherwise

$p=1$ to N_1 , $q=1$ to N_2 , $m=1$ to K , $n=1$ to K

$$y_{\min} = \min_{m,n} (y_{m,n})$$

۶- شبیه‌ترین طبقه به الگوی ورودی و میزان شباهت را بیاید:

$$t_m = \begin{cases} \text{mean}_{pq} \left[\frac{|\Theta - \Theta_{pqm}|}{\alpha l} \right] & \text{if } \max \left[\left| \Theta - \Theta_{pqm} \right| \right] \geq \alpha l \\ 0 & \text{otherwise} \end{cases}$$

otherwise

$p=1$ to N_1 , $q=1$ to N_2 , $m=1$ to K

$$t_{\min} = \min_m (t_m)$$

۷- اگر $t_{\min} \geq y_{\min}$ بنابراین به مرحله ۳ بروید و اگر $t_{\min} < y_{\min}$ باشد به این معناست که میزان شباهت الگوی ورودی به طبقه‌ها، کمتر از میزان شباهت دو الگوی طبقه می‌باشد و در واقع دو الگو در طبقه‌ها وجود دارند که بیش از حد به یکدیگر شبیه هستند، بنابراین به مرحله ۷ بروید.

۸- Θ مربوط به الگوی جدید را به جای یکی از دو الگوی شبیه به هم (که مکان آن را در مرحله ۵ یافته‌اید) قرار دهید. اگر الگوهای ورودی به اتمام نرسیده باشند، به مرحله ۳ بروید وگرنه آموزش طبقه‌ها به اتمام رسیده است.

مرحله دوم: آموزش الگوهای طبقه‌ها

آموزش الگوهای هر طبقه شبیه آموزش خود طبقه‌هاست به جز اینکه:

الف - در ابتدا با استفاده از $\{\beta_1, \alpha_1\}$ ، طبقه مربوط به الگوی ورودی را بیاید.

ب - حداکثر تعداد الگوهای هر طبقه را برابر N قراردادده و دو پارامتر دیگر را برابر β_2 و α_2 قرار دهید و مراحل شبیه به آنچه برای آموزش خود طبقه‌ها بیان شد را طی کنید.

الگوریتم پیشنهادی باعث خواهد شد تا غیر شبیه‌ترین الگوها به عنوان طبقه‌ها انتخاب گردند. اگر هنگام آموزش طبقه‌ها و الگوهای طبقه‌ها، از اکثر ورودی‌های ممکن استفاده کنیم و در صورت انتخاب صحیح K ، $\{\beta_1, \alpha_1\}$ ، N ، β_2 و α_2 بهترین پاسخ را از شبکه خواهیم گرفت.

ازای مقدار $0/3$ برای β ، صحت تشخیص برابر $81/3\%$ بوده است.

۸- نتیجه‌گیری

در این مقاله، ضمن تعریف چهار نوع نرون فازی، شبکه عصبی فازی چهار لایه معرفی گردید و ساختار متوالی جهت طبقه‌بندی و تشخیص کاراکترهای فارسی ارائه شد. CFNN ضمن داشتن قابلیت ساختاری و قدرت یادگیری شبکه‌های عصبی، الگوریتم‌های فازی را برای پردازش اطلاعات استفاده می‌کند. CFNN قادر است در هر زمان الگوهای جدید را آموزش ببیند، بدون آنکه الگوهای شبیه به الگوهای آموزش یافته را دوباره آموزش یابد. سرعت تشخیص آن نیز قابل توجه است. قدرت CFNN در تشخیص الگوهای شیف‌ت داده شده و نویزی می‌باشد که به آن ارزش می‌دهد. CFNN می‌تواند در دیگر مسائل تشخیص الگو نیز مورد استفاده قرار گیرد.

جدول (۲) $K=60, N=20, \beta=0/3$

α	0/6	0/7	0/8	0/9	0/1
گروه الف	100	100	100	100	100
گروه ب	96/0	96/0	96/2	94/3	88/9
گروه ج	90/1	91/2	87/4	85/4	80/8
گروه د	87/6	89/9	87/8	81/3	78/9

جدول (۳) $\alpha=0/7, \beta=0/3$

N,K	10,110	15,75	20,60	25,45
گروه الف	100	100	100	100
گروه ب	95/1	96/1	96/0	90/8
گروه ج	90/5	92/8	91/2	85/1
گروه د	91/2	93/9	89/9	83/7

بطور مثال، در جدول (۱) که به ازای مقادیر $0/9 = \alpha$ ،

مراجع

- [1] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA: MIT Press, 1986.
- [2] C. Lau, Ed., *Neural Networks. Theoretical Foundations and Analysis*. Piscataway, NJ: IEEE Press, 1992.
- [3] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition," *IEEE Trans. Syst., Man, and Cyber.*, Vol. SMC-13, no. 5, pp. 826-834. Sept./Oct. 1983.
- [4] G. L. Martin and J. A. Pittman, "Recognizing hand-printed Letters and Digits Using Backpropagation Learning," *Neural Computation*, Vol. 3, No. 2, pp. 258-267, summer 1991.
- [5] I. Guyon, P. Albrecht, Y. IL. Cun, J. Denker, and W. Hubbard, "Design of a Neural Net Work Character Recognizer for a Touch Terminal," *Pattern Recognition*, Vol. 24, No. 2, PP. 105-117, Feb. 1991.
- [6] L. A. Zadeh, "The Role of Fuzzy Logic in the Managment of Uncertainty in Expert Systms," in *Approximate Reasoning in Expert System*, Gupta, Kandel, Bandler, and Kiszka, Eds. New York: Elsevier, 1985.
- [7] A. Kandel, *Fuzzy Techniques in Pattern Recognition*. New York: Wiley, 1982.
- [8] T. Yamakawa and S. Tomoda, "A Fuzzy Neuron and Its Application to Pattern Recognition." in *Proc. Third Int. Fuzzy System Associat. Congress, Japan, 1989*, pp. 30-38.
- [9] H. K. Kwan, Y. Cai, "A Fuzzy Neural Network and Its Application to pattern recognition," *IEEE Trans. Fuzzy Systems*, Vol. 2, No.3, Aug. 1994, pp. 185-193.

References

- [1] R. H. DuHamel and D. E. Isbell, "Broadband Logarithmically Periodic Antenna Structures", 1957 IRE Nat. Conv. Rec., vol. 5, pt. 1, pp. 119-128.
- [2] D. E. Isbell, "Log Periodic Dipole Arrays", IRE Tran. Ant. and Prop., vol. Ap-8, pp. 260 - 267, May 1960.
- [3] R. L. Carrel, "Analysis and Design of the Log-periodic Dipole Antenna", Antenna Lab., University of Illinois, Urbana, Tech. Rep. 52, October 1961.
- [4] K. E. Jones and R. Mittra, "Continuously Scaled Transmission Lines", presented at the 1962 URSI Fall Meeting, Ottawa, Ont., Canada.
- [5] K. E. Jones and R. Mittra, "Non-uniform Transmission Lines with Application to the Log-periodic Antennas", Proc. 1964 Nat. Electron. Conf., vol. 20, pp. 23-28.
- [6] W. M. Cheong and R. W. P. King, "Log-periodic Dipole Antenna", Radio Sci., vol. 2 (new series), No. 11, pp. 1303-1314, Nov. 1967.
- [7] J. Wolter, "Solution of Maxwell's Equations for Log-Periodic Dipole Antennas", IEEE, Trans. on Ant. and Prop., vol. 18, No. 6, pp. 734 - 740, Nov. 1970.
- [8] J. Herrault, R. Moini, A. Reineix and B. Jecko, "A New Approach of Microstrip Antenna Using a Mixed Analysis Transient - Frequency", IEEE, Trans. on Ant. and Prop., vol.38, No. 8, Aug. 1990.
- [9] E. K. Miller, "An Integrodifferential Equation Technique for Time Domain Analysis of Thin Wire Structures Part I", Computational Physics, No. 12, pp.24-28, 1973.
- [10] R. Moini, R. Aghajafari, and A. Tavakoli, "Susceptibility of Log Periodic Dipole Arrays", Proc. Euroelectromagnetics-EUROEM 94, Bordeaux, France, pp. 1470 - 1477, 1994.