Integration Method with Backpropagation

Nidhal H. AL-Assady Jamal S. Majeed Shahbaa I.Khaleel
shahbaaibrkh@uomosul.edu.iq
College of Computer Sciences and Mathematics
University of Mosul, Iraq

Received on: 16/12/2002 Accepted on: 25/01/2003

Abstract

In this research, a new method is discovered (combined method) to accelerate the backpropagation network by using the expected values of source units for updating weights, we mean the expected value of unit by the sum of the output of the unit and its error term multiplied by the factor Beta to accelerate the algorithm and also adjust the value of learning coefficient continuously if the value of energy function E decreases the learning rate is increased by a factor ρ, if the value of the energy function E increases , the value of the learning rate is decreased by a factor σ. To obtain the optimal weight with minimum iteration and minimum time, we applied a new method on many applications to prove the result of this method (pattern compression, encoding and recognition on Arabic, English digits and alphabetic.

Keywords: backpropagation , artificial neural networks.

طريقة الدمج بشبكة انتشار الخطأ خلفاً

نضال حسين الأسدي جمال صلاح الدين سيد مجيد شهباء إبراهيم خليل
كلية علوم الحاسبات والرياضيات
جامعة الموصل

تاريخ استلام البحث: 2002/12/16 تاريخ قبول البحث: 2003/01/25

الملخص

تم التوصل في هذا البحث إلى طريقة جديدة وهي أل Combined method لتسريع عمل شبكة انتشار الخطأ خلفاً وذلك عن طريق استخدام القيمة المتوقعة للخلية (expected value) في تغيير الوزن (weight) ونقصد بالقيمة المتوقعة حاصل جمع الإخراج الحقيقي للخلية (actual output) ونسبة الخطأ (error term) للخلية نفسها مضروباً بمعامل معين وهو (Beta) الذي يستخدم أيضا العمل لتسريع ، فضلاً عن تغيير قيمة نسبة التعلم (learning coefficient) باستمرار في أثناء التنفيذ وذلك عن طريق زيادة قيمة نسبة التعلم كلما قلت نسبة الخطأ وتقليل قيمة نسبة التعلم كلما زادت قيمة نسبة الخطأ المحسوبة عن طريق أل (energy function) فيتم بذلك توازن الشبكة ومن ثم الحصول على الوزن المثالي للشبكة بأقل وقت واقل عدد من الخطوات . وقد

تم تنفيذ هذه الطريقة على مجموعة من التطبيقات لإثبات نتائجها وهي كبس وتمييز وترميز النماذج و هي هنا الأرقام والحروف العربية والإنكليزية وبهيئات ومعماريات شبكات متعددة حسب النموذج.

الكلمات المفتاحية: شبكة انتشار الخطأ خلفا، الشبكات العصبية الاصطناعية.

المقدمة

إن شبكة انتشار الخطأ خلفاً (error backpropagation EBP) واسعة الاستخدام وهي من الشبكات العصبية المتعددة الطبقات وذات التغذية الأمامية (feedforward) .ويعرف قانون تعلم شبكة انتشار الخطأ خلفاً EBP ذات الطبقات المتعددة ، بخوارزمية انتشار الخطأ خلفاً (backpropagation BP) . وهي خوارزمية واسعة الانتشار وذلك لسهولتها ، وقابليتها على خزن المعلومات بصورة ضمنية في الارتباطات التي تمثل الأوزان التي تربط خلية بأخرى.وتستخدم في تطبيقات متنوعة (تمييز الأنماط ، تمييز الصوت ، معالجة الصور ، . . . الخ) ولكن بالرغم من ان هذه الشبكة ناجحة في الكثير من التطبيقات لكنها ليست الدواء لكل المسائل ، مثلاً تحتاج المسائل المعقدة حتى تصل الى الحل إلى وقت كبير جداً أي إن سلسلة التدريب عليها تتطلب مئات أو آلاف الخطوات نسبة الى المسائل البسيطة. هذا السبب قاد الى إجراء تحسين و تسريع لهذه الشبكة وذلك بإجراء تعديلات على خوارزمية عملها وفي هذا البحث تم التوصل إلى خوارزمية جديدة تسرع من عمل الشبكة وقبل التطرق إلى هذه الطريقة لا بد من إعطاء توضيح شامل لمعمارية وخوارزمية هذه الشبكة.[9]

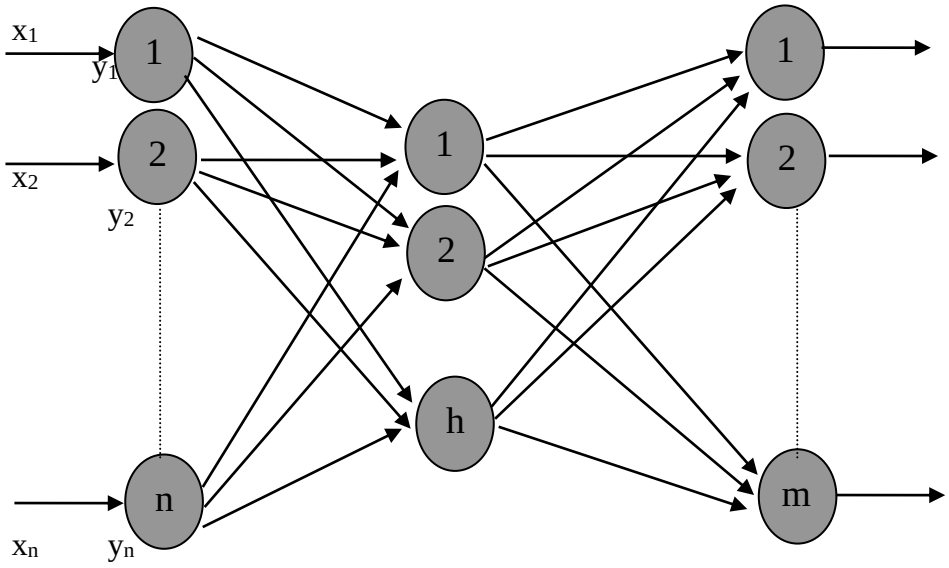
1 معمارية شبكة إنتشار الخطأ خلفاً

تتألف شبكة انتشار الخطأ خلفاً على الأقل من ثلاث طبقات من الخلايا : طبقة الإدخال، والطبقة الوسطى وتسمى hidden layer وطبقة الإخراج . كل طبقة بالشبكة ترتبط مع الطبقة التي تليها أي إن أية خلية node في طبقة الإدخال ترسل إخراجها إلى كل الخلايا في الطبقة الوسطى ، وخلايا الطبقة الوسطى ترسل إخراجها إلى كل خلية في طبقة الإخراج. [2] [3]

ويعتمد عدد الخلايا في الطبقة الوسطى على درجة تعقيد المسألة وحجم معلومات الإدخال . فإذا كان عدد الخلايا node في الطبقة الوسطى كبيراً جداً بالنسبة إلى عدد خلايا الإدخال فإنه لا يوصل الى حل ، من جهة أخرى إذا كان عدد خلايا الطبقة الوسطى قليلاً جداً فإنها سوف تأخذ عدداً كبيراً من الخطوات لتدريب الشبكة [4]

والشكل (1) يوضح الشبكة العصبية ذات الانتشار العكسي المتكونة من ثلاث طبقات طبقة الادخال ، الطبقة الوسطى ، وطبقة الإخراج على الترتيب .

إن كل خلية أو وحدة معالجة processing unit في أي طبقة ترتبط مع كل خلايا أو وحدات المعالجة للطبقة الأخرى عن طريق ارتباطات الأوزان . الإدخال للشبكة يمثل بمتجه $X = (x_1, x_2, x_3, \ldots, x_n)$ ، والإخراج بمتجه $Y = (y_1, y_2, y_3, \ldots, y_m)$ و n , m هي أبعاد متجهات الإدخال والإخراج. [7][8]

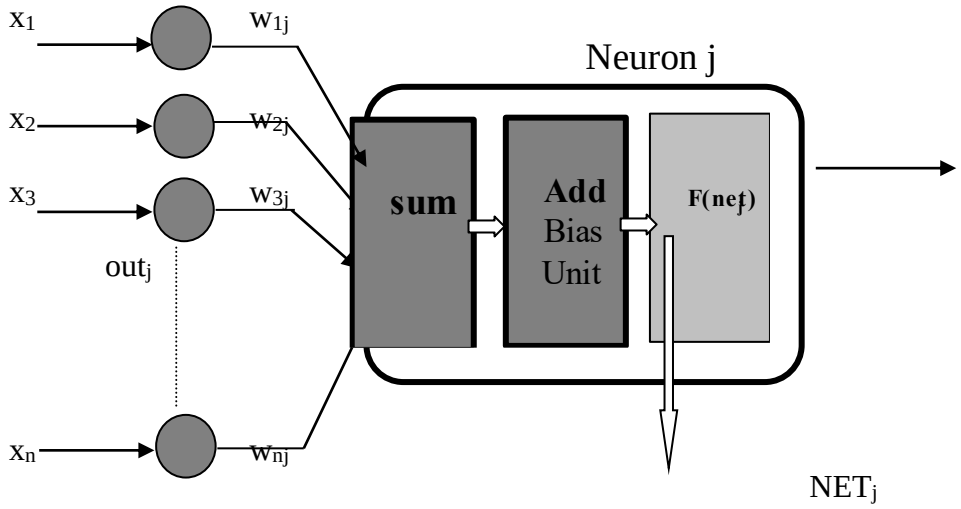


2 هيئات شبكة انتشار الخطأ خلفاً

2.1 الخلية

الشكل (2) يوضح أساس بناء الخلية لشبكة إنتشار الخطأ خلفاً. تطبق مجموعة الإدخالات الآتية من الخارج أو من الطبقة السابقة ، كل إدخال يضرب مع الوزن المقابل له ، ثم تجمع الإدخالات الموزونة وهذا الجمع يطلق عليه مصطلح NET . بعد حساب الـ NET تمرر على دالة الفعالية F لتحويلها الى إشارة الـ out [Wasserman , 1989]

$$NET_j = x_1 w_{1j} + x_2 w_{2j} + x_3 w_{3j} + \ldots + x_n w_{nj} \quad \ldots..(1)$$



j يمثل مؤشراً للخلية j في الطبقة ، x_I هو متجه الادخال ، w_{ij} يمثل الوزن ، و n طول متجه الإدخال أي عدد خلايا الإدخال . [5] [6]

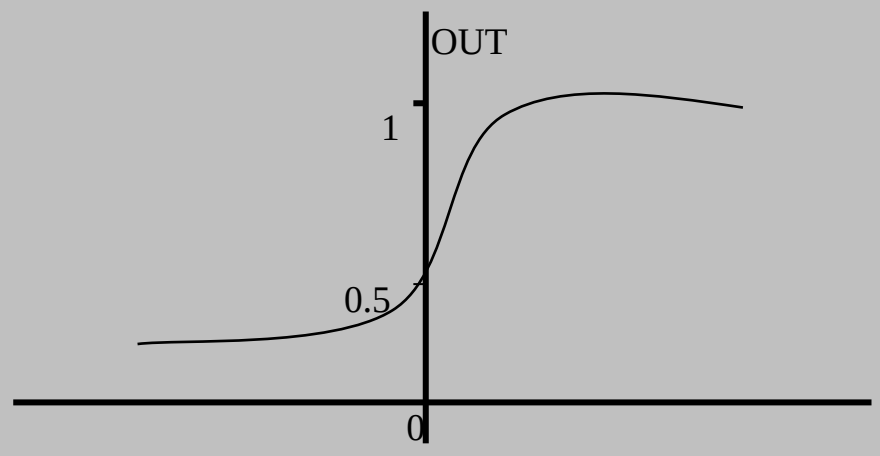
2.2 الدالة السجماوية

الشكل (3) يوضح دالة الفعالية المستخدمة في شبكة الـ BP ، الـ sigmoid function توصف بأن مشتقتها بسيطة ، وهي تستخدم في تنفيذ خوارزمية الـ BP

$$out = f(NET_j) = \frac{1}{1+e^{-NET_j}} \qquad (2)$$

$$f'(NET_j) = out\,(1 - out) \qquad (3)$$

وفي بعض الاوقات تسمى الـ sigmoid function بـ logistic أو دالة الاخماد squashing function وذلك لأنها تكبس مدى الـ NET_j وكذلك الـ out بين الصفر والواحد شبكة انتشار الخطأ خلفاً تستخدم هذه الدالة فقط. [1] [8]



$$Out = F(NET) = 1/(1+e^{-NETj})$$

2.3 إضافة خلية الانحياز

يفضل في شبكة الـ BP اضافة خلية الانحياز (Aneuron Bias) لتسريع عمل الشبكة والاقتراب من الحل وهي تشبه الـ threshold في شبكة الـ perceptron .

يتم تغير الوزن لهذه الخلية مثل بقية الخلايا للشبكة ما عدا أن الادخال لخلية الانحياز دائماً يكون 1+ بدل أن يكون الادخال كما في بقية خلايا الشبكة هو الاخراج للطبقة السابقة.[8]

2.4 استخدام المتغير β

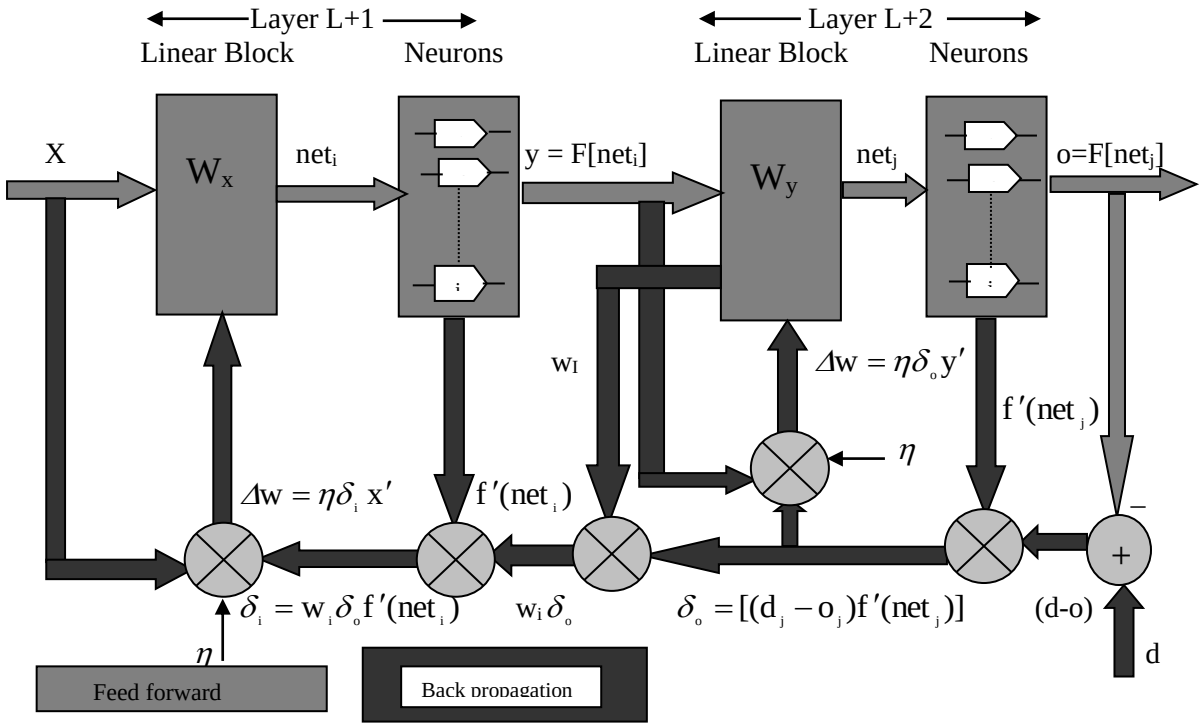
المتغير β يستخدم في دالة التنشيط في خوارزمية التدريب لشبكة انتشار الخطأ خلفاً BP وقيمة β المستخدمة في خوارزمية BP مداها محصور بالفترة [1.0 . . 0.1] .

كلما كانت قيمة المتغير β قليلة يقل أو تبطئ من الاقتراب بينما إذا كانت قيمة المتغير β كبيرة أي قريبة الى 1.0 تسرع من عملية التدريب للشبكة . يستخدم β في خوارزمية انتشار الخطأ خلفاً العامة Standard Backpropagation Algorithm(SBP) .

$$f'(net_{pj}^{L}) = \beta\ (1 - out_{pj}^{L})\, out_{pj}^{L} \qquad (4)$$

3 نظرة عامة إلى التدريب

الهدف من التدريب هو تغيير أوزان الشبكة ، وذلك للحصول على مجاميع الإخراجات المطلوبة للإدخالات المعطاة . مجاميع الإدخال والإخراج يشار إليها بمتجهات. زوج التدريب هو input vector الإدخال مع target vector الإخراج المطلوب . وتدرب الشبكة عادة على عدد من ازواج التدريب ، ويطلق على أزواج التدريب هذه مجموعة التدريب،و قبل بدء عملية التدريب ، يجب أن تبدأ كل الأوزان بأرقام عشوائية صغيرة . هذا يضمن للشبكة بأن لا تتشبع بأوزان ذات قيم كبيرة ،إن مجموعة التدريب هذه تستخدم للتدريب وتكون حاضرة للشبكة أوقات كثيرة خلال طور التدريب ، وتحضر نماذج الإدخال للشبكة.



الشكل (4) مخطط إشارات التغذية الأمامية والتغذية الخلفية

يدرب كل نموذج ويحسب له الإخراج output ، وهذا الإخراج يقارن مع الإخراج المطلوب (target output or desired output) وكذلك تحدد قيمة الخطأ ويتم نشر هذه الأخطاء خلفاً من طبقة الإخراج باتجاه طبقة الإدخال وهذا يحدث فقط في طور التعليم ، بعد توقف التدريب يفحص تنفيذ الشبكة والشكل (4) يوضح مخطط إشارات التغذية الأمامية الـ forward والتغذية الخلفية الـ backward . [10]

4 خوارزمية تدريب شبكة انتشار الخطأ خلفاً

إن معالجة التدريب لشبكة الـ BP هي معالجة متكررة (iterative process) وكل تكرار (iteration) يتكون من عدد من الخطوات (steps). [8]

الخطوة الأولى:اختيار قيم ابتدائية صغيرة بشكل عشوائي لأوزان كل الخلايا في الشبكة العصبية.
الخطوة الثانية : اختيار زوج التدريب من مجموعة التدريب [X_p , T_p] والذي يمثل الإدخال والإخراج المطلوبين ويتم حساب قيم الاخراج لكل وحدة حساب أي لكل خلية في كل طبقة L بالشبكة .

$$net_{pj}^{L+1} = \sum_{i=1}^{n^L} w_{ij}^{L}\, out_{i}^{L} + bias_{j}^{L+1} \qquad (5)$$

$$out_{pj}^{L+1} = f\left(net_{pj}^{L+1}\right) = \frac{1}{1+e^{-\beta\, net_{pj}^{L+1}}} \qquad (6)$$

net_{pj}^{L+1} هي مجموع ضرب كل الادخالات المرتبطة بالخلية j مع الاوزان المقابلة لها.
out_{pj}^{L+1} هو اخراج الخلية j بعد تطبيق الـ sigmoid function عليها في الطبقة L+1 .
$out_{pj}^{0} = x_j$ يمثل الادخال للنموذج و p رقم النموذج و j رقم الـ node.

الخطوة الثالثة: يتم حساب الخطأ بين الاخراج الحقيقي للشبكة out_{pj} والاخراج المطلوب من زوج التدريب ، بعد ذلك نستخدم out_{pj}^{o} الاخراج الحقيقي مع t_{pj} الاخراج المطلوب لحساب delta (δ)

$$\delta_{pj}^{o} = \left(t_{pj} - out_{pj}^{o}\right) f'(net_{pj}^{o}) \qquad (7)$$

t_{pj} هو الاخراج المطلوب للخلية j . out_{pj} هو الاخراج الحقيقي للخلية j.
$f'(net_{pj})$ هي مشتقة الدالة السجماوية f(net) .
الخطوة الرابعة : نقوم بحساب قيمة (δ) delta لكل الطبقات الوسطى باستخدام الخطأ خلفاً .

$$\delta_{pi}^{L+1} = \mathrm{f}'(\mathrm{net}_{\mathrm{pi}}^{\mathrm{L+1}}) \left[\sum_{j=1}^{m^{L+2}} \delta_{pj}^{L+2}\ \mathrm{w}_{\mathrm{ij}}^{\mathrm{L+1}} \right] \qquad (8)$$

m هي عدد الخلايا في الطبقة L .

الخطوة الخامسة : تغيير الاوزان باتباع المعادلات الآتية :

$$w_{ij}^{new} = \mathrm{w}_{\mathrm{ij}}^{\mathrm{old}} + \Delta\, \mathrm{w}_{\mathrm{ij}}^{\mathrm{L}} \qquad (9)$$

$$\Delta w_{ij}^{L} = \eta\, \delta_{pj}^{L+1}\, \mathrm{out}_{\mathrm{pi}}^{\mathrm{L}} \qquad (10)$$

الخطوة السادسة : الرجوع الى الخطوة الثانية ونكررها لكل نموذج p لمجموعة التدريب الى أن يصل الخطأ الى قيمة مقبولة .

خلال خطوات التدريب ، نلاحظ أن خوارزمية الـ BP تنفذ خطوة الانتشار الامامي ثم بعدها خطوة الانتشار الخلفي وتنفذ هاتين الخطوتين لكل نموذج خلال التدريب .

4.1 المسار الأمامي

الخطوة الثانية يمكن توضيحها كما يأتي : أخذ متجه الإدخال input vector والممثل بالمتجه X ونعمل عليه حسابات للحصول على متجه الإخراج output vector والممثل بالمتجه Y .ونقوم بعملية الحساب للشبكات ذات الطبقات (طبقة بعد طبقة) ونبدأ بالطبقة الأقرب إلى الإدخالات.

نحسب قيمة الـ net لكل خلية في أول طبقة عن طريق جمع حاصل ضرب الإدخالات لهذه الخلية مع الأوزان المقابلة لها . بعد ذلك يتم تطبيق دالة التنشيط على إشارة الـnet للحصول على الإخراج لكل خلية في تلك الطبقة، الإخراجات المحسوبة لكل خلية في هذه الطبقة تعتبر إدخالات للطبقة التي تليها وتتكرر هذه العملية ، طبقة بعد أخرى الى أن نحصل على آخر مجموعة للإخراج الحقيقي للشبكة.[4] [8]

4.2 المسار الخلفي

4.2.1 تغيير الأوزان لطبقة الإخراج

تتواصل خوارزمية انتشار الخطأ خلفاً للشبكات العصبية الاصطناعية ذات التغذية الأمامية feedforward artificial neural networks FFANNs بالعمل عن طريق إدخال النموذج إلى خلايا الإدخال أو الطبقة رقم صفر أي الى طبقة الإدخال ، بعد ذلك الشبكة

تنتج الإخراج الحقيقي، يقارن الإخراج الحقيقي بالإخراج المطلوب، والاختلاف بين الإخراج الحقيقي والإخراج المطلوب يسمى الخطأ error ويمكن حساب الخطأ لكل خلية E_{pj} لطبقة الإخراج لزوج التدريب (X_p, T_p) كما يأتي:

$$E_{pj} = t_{pj} - out^{o}_{pj} \qquad (11)$$

أن هدف خوارزمية التعلم هو استخدام الخطأ في تغيير الاوزان وذلك لتقليل الخطأ تدريجياً، ويتوقف التدريب عند الحصول على أقل نسبة خطأ لكل خلية من خلايا الإخراج، أو عندما لا نحصل على أي تحسين أو أي تقدم لنسبة الخطأ الناتجة أي أن الخطأ يستمر بالتزايد دون النقصان، ففي هذه الحالة نقوم بتغيير الاوزان الابتدائية (initial weight) للشبكة ونبدأ بتدريب الشبكة من جديد . الشكل (5) يوضح تغيير الوزن من خلايا i في الطبقة الوسطى hidden layer L+1 الى خلايا j في طبقة الإخراج output layer L+2 . الإخراج الحقيقي لخلايا طبقة الإخراج L+2 يطرح من الإخراج المطلوب للحصول على الخطأ error وهذا الخطأ يضرب بمشتقة دالة التنشيط (الفعالية). $\left[out^{o}_{pj}\left(1 - out^{o}_{pj}\right)\right]$ للحصول على قيمة δ.

$$\delta = out^{o}_{pj}\,(1 - out^{i}_{pj})\,(t_{pj} - out^{o}_{pj})$$

بعد ذلك تضرب δ بالناتج out للخلية i في الطبقة الوسطى ، ثم تضرب بنسبة التعلم η المحصورة بين [1.0 . . 0.01] والنتيجة تجمع مع الوزن القديم. وهذه هي معادلات تغيير الوزن :

$$w^{new}_{ij} = w^{old}_{ij} + \Delta w^{L+1}_{ij}$$
$$\Delta w^{L+1}_{ij} = \eta\, \delta^{L+2}_{pj}\, out^{L+1}_{pi}$$

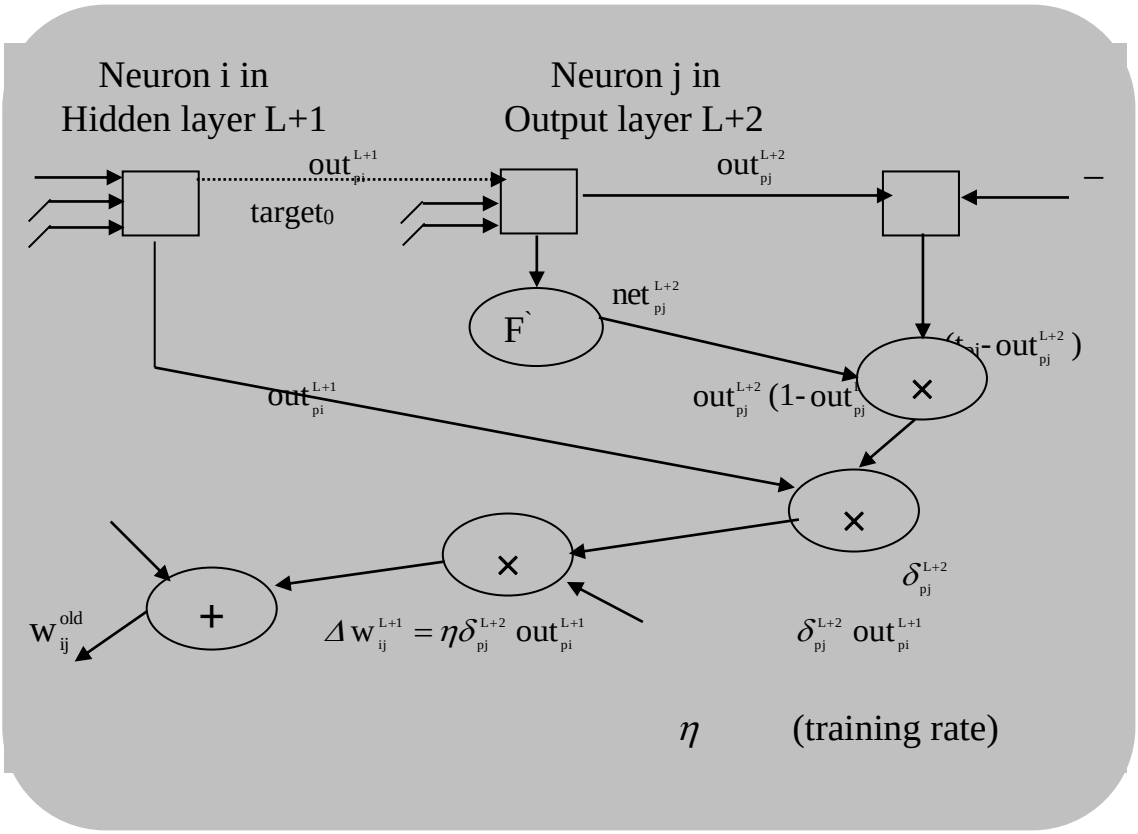
w^{old}_{ij} هو الوزن من الخلية i في الطبقة الوسطى hidden layer الى الخلية j في طبقة الإخراج قبل تغيير الوزن أي الوزن القديم.

w^{new}_{ij} الوزن الجديد أي الوزن بعد التغيير .

δ^{L+2}_{pj} قيمة الـ δ للخلية j في طبقة الاخراج L+2

out^{L+1}_{pi} الناتج للخلية I في الطبقة الوسطى L+1

و i, j مؤشران يشيران الى الخلايا في الطبقات L+1 و L+2

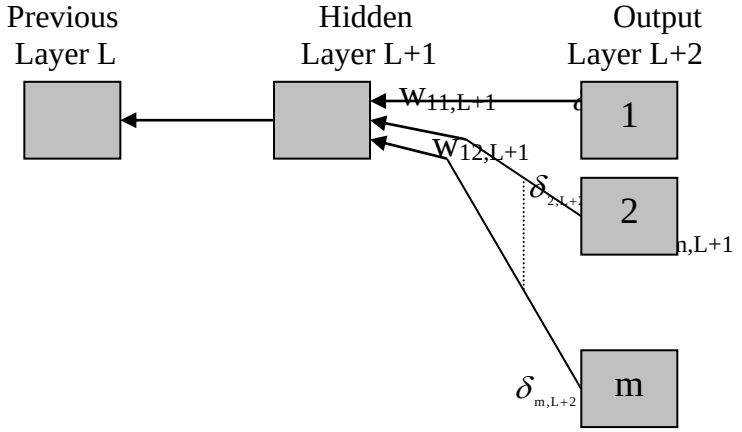


4.2.2 تغيير الأوزان للطبقات الوسطى

الطبقات الوسطى لا تحوي على الاخراج المطلوب ، شبكة الـ BP تدرب أي تغير الأوزان للطبقات الوسطى عن طريق انتشار الخطأ خلفاً من طبقة الإخراج باتجاه طبقة الإدخال وتغير الاوزان لكل طبقة . المعادلتان (7) و(8) تستخدمان لجميع الطبقات ، أي طبقة الإخراج والطبقات الوسطى . كما ذكرنا سابقاً تحسب δ لكل خلية في طبقة الإخراج ، وتستخدم δ الإخراج لتغيير الأوزان في الطبقة الوسطى إذ يتم ضرب δ لكل خلية في الإخراج بالوزن بين الخلية في طبقة الإخراج والخلية بالطبقة الوسطى وحاصل الجمع لضرب كل δ لكل خلية في طبقة الإخراج بالوزن المقابل لها يضرب بمشتقة الـ sequashing function أي $f'(net)$ لكل خلية في الطبقة الوسطى والمعادلة الآتية توضح ذلك :

$$\delta_{pi}^{L+1} = \mathrm{f}'(\mathrm{net}_{\mathrm{pi}}^{\mathrm{L+1}}) \left[\sum_{j=1}^{m^{L+2}} \delta_{pj}^{L+2} \ \mathrm{w}_{\mathrm{ij}}^{\mathrm{L+1}} \right]$$

والشكل (6) يوضح تغير الوزن للطبقة الوسطى لكل خلية من خلاياها تحسب δ وبعدها يتم تغير الوزن لخلايا الطبقة الوسطى.



الشكل (6) تغير الوزن في الطبقة الوسطى

5 طريقة الـ Combined

بالرغم من أن شبكة BP ناجحة في الكثير من التطبيقات لكنها معرضة للوقوع بمشكلة شلل الشبكة نتيجة الاختيار الخطأ للأوزان وخاصة إذا اختيرت الأوزان أرقاماً كبيرة فهذا يعني ان التغير في هذه الأوزان سوف ينتج أرقاماً كبيرة جداً مما يؤدي الى ان كل او معظم الخلايا في الشبكة تنتج قيماً كبيرة للإخراج الحقيقي out ، في حين ان مشتقة دالة الإخماد ذات قيم صغيرة جداً وهذا يؤدي الى شلل الشبكة، لذا يجب اختيار ارقام ابتدائية صغيرة للأوزان وتقليل نسبة التعلم (learning rate) ولكن سوف يزداد وقت التدريب . وبعمل عدد من الاختبارات المختلفة اتضح انه إذا كانت نسبة التعلم صغيرة جداً فإن هذا يؤدي إلى ان الاقتراب الى الحل الصحيح سوف يكون بطيئاً جداً، أما إذا كانت نسبة التعلم كبيرة جداً فهذا يؤدي الى شلل الشبكة.

ولحل جميع هذه المشاكل الموجودة بشبكة الـ BP و تسريع عملها تم استحداث او التوصل الى طريقة جديدة وهي الـ combined method التي تختلف عن الخوارزمية الأصلية لـ BP بإجراء تعديلات على معادلة حساب الأوزان. وذلك بدلاً من استخدام القيمة الحالية للخلية أي الـ (actual output) في تغيير الاوزان تم استخدام القيمة المتوقعة للخلية (expected value) في تغيير الوزن ونقصد بالقيمة المتوقعة هي حاصل جمع الإخراج الحقيقي للخلية

(actual output) ونسبة الخطأ (error term) للخلية نفسها مضروباً بمعامل Beta كما موضح بالمعادلة الأتية:

$$\Delta w_{ij}^{L} = \eta (out_{pi}^{L} + Beta\ \delta_{pi}^{L})\ \delta_{pj}^{L} \qquad (12)$$

هذه المعادلة تصبح مثل معادلة تغيير الوزن بالخوارزمية الأصلية اذا كانت قيمة المعامل Beta مساوية للصفر 0.0 وعادة نأخذ قيمته 1.0 وكلما تزداد يحصل تسريع بعمل الشبكة.في هذه الطريقة تم إجراء تغيير على معادلة حساب التغير بالوزن فضلاً عن ذلك يتم تغيير قيمة نسبة التعلم (learning coefficient) باستمرار في أثناء عمل الشبكة إذ يتم في بداية التدريب اختيار قيمة نسبة التعلم بصورة عشوائية وتكون ضمن المدى [0.9…0.1] ويحصل تدريب للشبكة وتحسب فيه E وهي الـ energy function وكلما قلت قيمة E نقوم بزيادة قيمة نسبة التعلم بمقدار معين وذلك بضرب قيمة نسبة التعلم بمعامل معين وهو ρ وقيمته أعلى من الواحد وهذا يساعدنا في الحصول على قيمة جيدة للخطوة اللاحقة أي ان قيمة نسبة التعلم تزداد بشكل exponentialy وازدياد قيمة E يعني أن آخر قيمة لنسبة التعلم أصبحت كبيرة جداً ويجب عمل ثلاث خطوات أولاً يلغى آخر تعديل للوزن والذي استخدم المعادلة (12) في حسابه وثانياً تقلل قيمة نسبة التعلم وذلك بضربها بمعامل معين وهو σ وقيمته اقل من الواحد وأخيرا تكرر الخطوتين السابقتين وإذا كانت المحاولة الجديدة تقلل قيمة الـ E فان مقدار التقليل لنسبة التعلم اصبح مقبولاً للخطوة اللاحقة وإلا فان تقليل نسبة التعلم يتكرر إلى أن نحصل على قيمة تقلل نسبة الخطأ E (Error) .

لنفرض ان معامل نسبة التعلم للوزن w_{ij}^{L} بالخطوة s هو η_{ij}^{L} [s] فان خوارزمية الـ combinedهي:

الخطوة الاولى: اختيار قيمة ابتدائية لنسبة التعلم η لتعليم الشبكة العصبية.

الخطوة الثانية: وضع قيمة نسبة التعلم $\eta_{ij}^{L}[0]=\eta$ ونفس قيمة نسبة التعلم لكل الاوزان w_{ij}^{L} بالشبكة العصبية.

الخطوة الثالثة: تنفيذ خوارزمية الـ BP بدون الـ momentum term أي حسب الخوارزمية الأصلية ما عدا ان تغير اوزان الشبكة العصبية يتم باستخدام المعادلة الجديدة

$$\Delta w_{ij}^{L} = \eta (out_{pi}^{L} + Beta\ \delta_{pi}^{L})\ \delta_{pj}^{L}$$

الخطـــوة الرابعـــة : اذا قلـــت قيمـــة E (energy function) يـــتم وضـــع $\eta_{ij}^{L}[s+1]=\rho\,\eta_{ij}^{L}[s]$ لجميـع اوزان $w_{ij}^{L}[s]$ الشـبكة العصـبية ، وقيمـة معامـل الزيـادة $\rho>1$ increment factor .

الخطوة الخامسة: اذا ازدادت قيمة E (energy function) نفذ الآتي :

. وضع قيمة $\eta_{ij}^{L}[s+1]=\sigma\,\eta_{ij}^{L}$ ومعامل التقليل decrement factor <1

. عودة للتغيير بالوزن السابق أي للخطوة السابقة مع إهمال آخر تغير بالوزن حصل بالخطوة الحالية

. وضع $\Delta w_{ij}^{L}[s+1]=0$

الخطوة السادسة: إعادة الخوارزمية من الخطوة الثالثة.

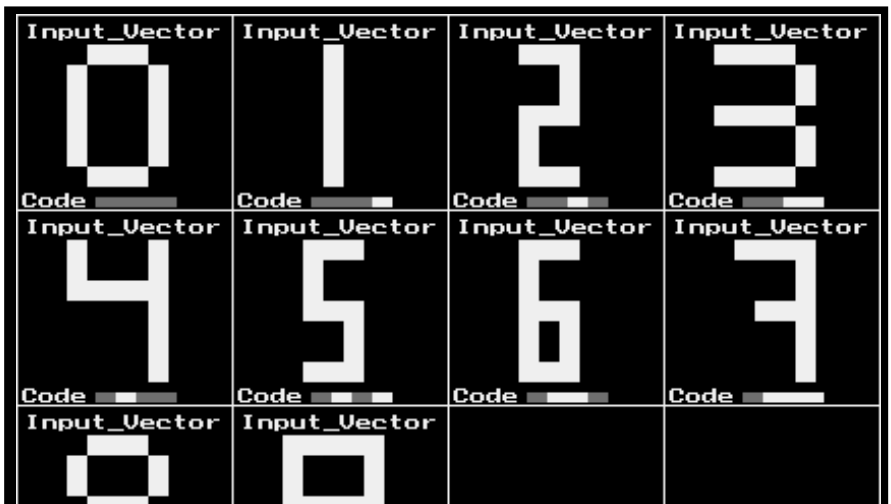
6 النتائج

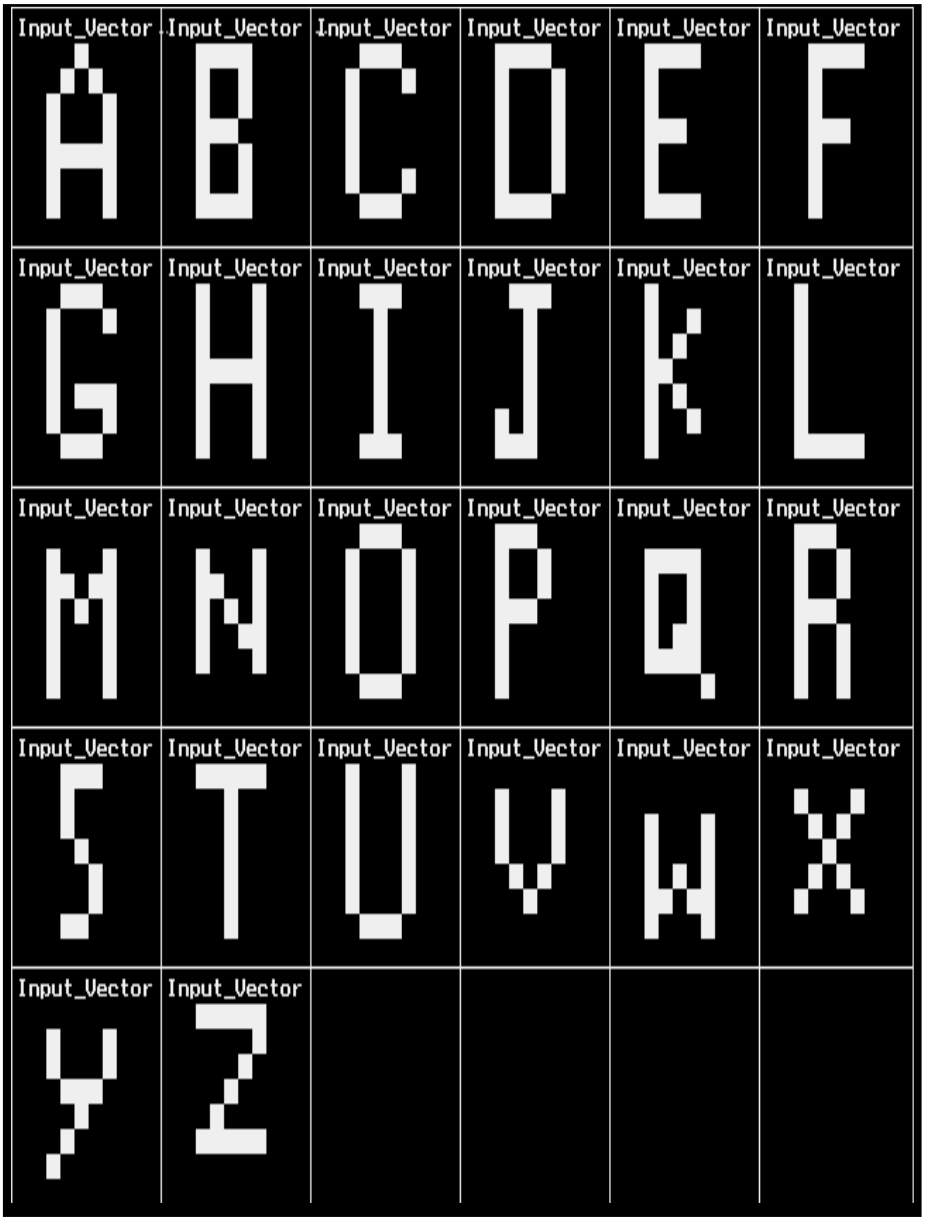
تم الاستنتاج بان طريقة الـ combined حققت سرعة اقتراب عالية، وذلك باستخدام expected value للخلية الأصلية بدل الـ actual output في تغيير الوزن مع تغيير قيمة نسبة التعلم تغييراً واضحاً، وذلك بالاعتماد على قيمة الـ total square error الذي يمثل ناتج الـ E .

ولمقارنة أداء شبكة انتشار الخطأ خلفاً باستخدام خوارزميتها الأصلية والخوارزمية الجديدة تم تنفيذ الخوارزميتين على مجموعة من التطبيقات بمعماريات شبكات مختلفة وحسب التطبيق وبوزن ابتدائي موحد وبشرط توقف ثابت للجميع واتضح أن خوارزمية الـ combined الجديدة هي أسرع بكثير من الخوارزمية الأصلية لشبكةBP . والجدول (1) فيه نتائج مقارنة الطريقتين الأصلية والجديدة لبعض المسائل . وتركيب الشبكة ممثل بالمتغيرات $n_I - n_h - n_o$ إذ أن n_I يمثل عدد خلايا طبقة الإدخال و n_h عدد خلايا الطبقة الوسطى و n_o عدد خلايا طبقة الإخراج للشبكة. تم استخدام نفس مجموعة الأوزان الابتدائية للشبكة ونفذت الخوارزميتان عليها لكل التطبيقات وهي التمييز والترميز والكبس للحروف والأرقام العربية والإنكليزية وبجميع هذه التطبيقات تم التوصل الى أن طريقة الـ combined هي أسرع بكثير من الطريقة الأصلية لشبكة BP وتصل إلى الحل بعدد خطوات اقل ووقت اقصر.

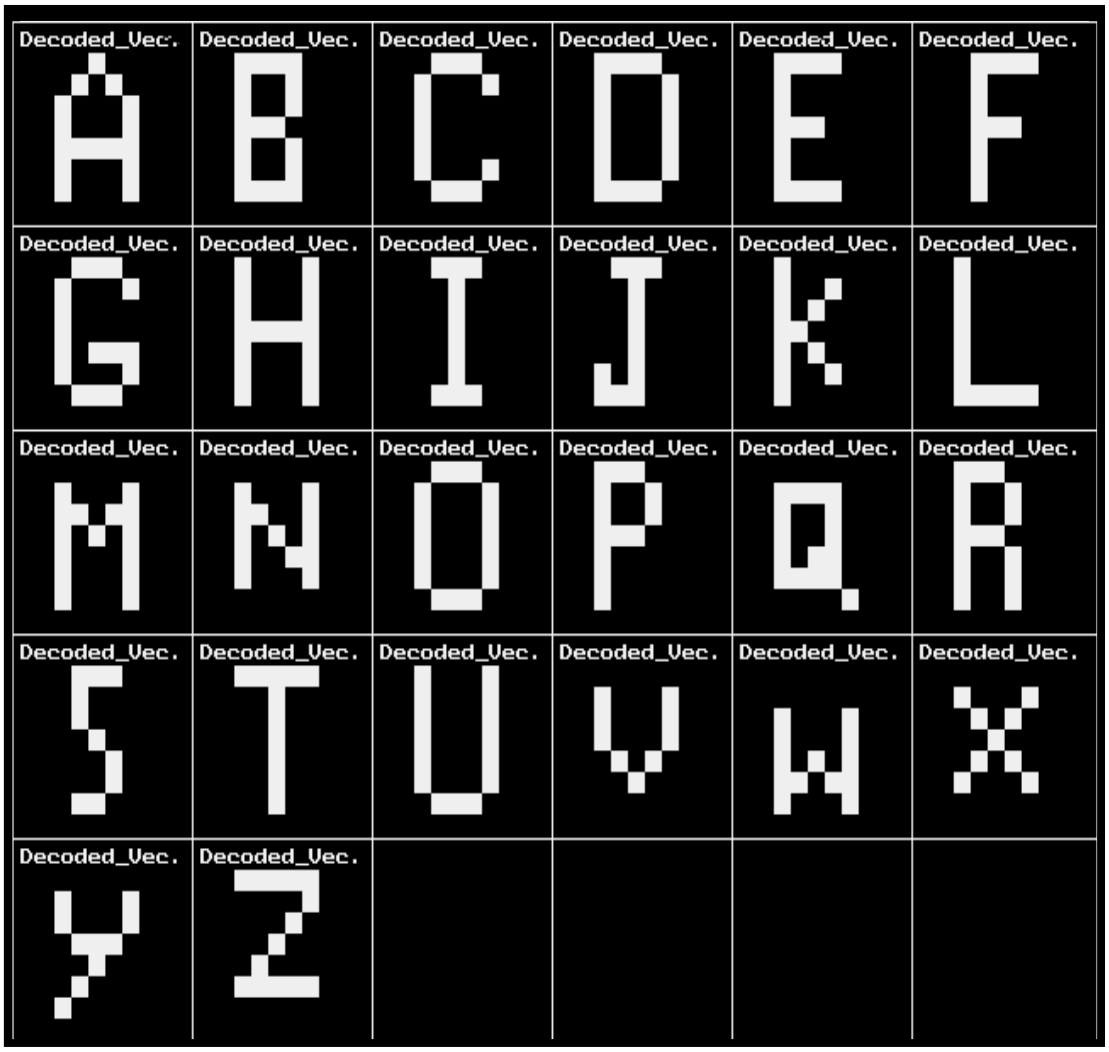
الجدول (1) يبين الفرق بين الخوارزمية الأصلية والخوارزمية الجديدة لشبكة الـ BP

	Net Structure and problem $n_i - n_h - n_o$	η	Beta	Iteration no.	Error
SBP	35- 10-35 compression of English digits	0.9	0.0	2271	0.099987
COM	35- 10-35 compression of English digits	0.9	16.0	31	0.097610
SBP	35- 5- 4 encoding of English digits	0.9	0.0	51	0.099640
COM	35- 5- 4 encoding of English digits	0.8	16.0	2	0.058707
SBP	35- 5- 10 recognition of English digits	0.9	0.0	625	0.099992
COM	35- 5- 10 recognition of English digits	0.7	12.0	18	0.095378
SBP	35- 10- 35 compression of Arabic digits	0.9	0.0	61311	0.01
COM	35- 10- 35 compression of Arabic digits	0.9	12.0	2	0.000007
SBP	35 –5- 4 encoding of Arabic digits	0.9	0.0	3796	0.009999
COM	35 –5- 4 encoding of Arabic digits	0.9	28.0	33	0.009804
SBP	35- 5- 10 recognition of Arabic digits	0.9	0.0	59545	0.01
COM	35- 5- 10 recognition of Arabic digits	0.9	30.0	2	0.007585
SBP	56- 6- 56 compression of Arabic alphabet	0.9	0.0	8228	0.1
COM	56- 6- 56 compression of Arabic alphabet	0.9	26.0	3	0.069963
SBP	56- 6- 5 encoding of Arabic alphabet	0.9	0.0	91	0.099886
COM	56- 6- 5 encoding of Arabic alphabet	0.9	12.0	2	0.005222
SBP	56-6-28 recognition of Arabic alphabet	0.9	0.0	357	0.399824
COM	56-6-28 recognition of Arabic alphabet	0.9	30.0	2	0.027258
	Net Structure and problems $n_i - n_h - n_o$	η	Beta	Iteration no.	Error
SBP	35- 10- 35 compression of English alphabet	0.9	0.0	53	0.099140
COM	35- 10- 35 compression of English alphabet	0.8	8.0	3	0.085929
SBP	35- 5- 5 encoding of English alphabet	0.9	0.0	104	0.099627
COM	35- 5- 5 encoding of English alphabet	0.9	10.0	10	0.092748
SBP	35- 5- 26 recognition of English alphabet	0.9	0.0	5423	0.099991
COM	35- 5- 26 recognition of English alphabet	0.9	28.0	19	0.098239




Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector
Input_Vector

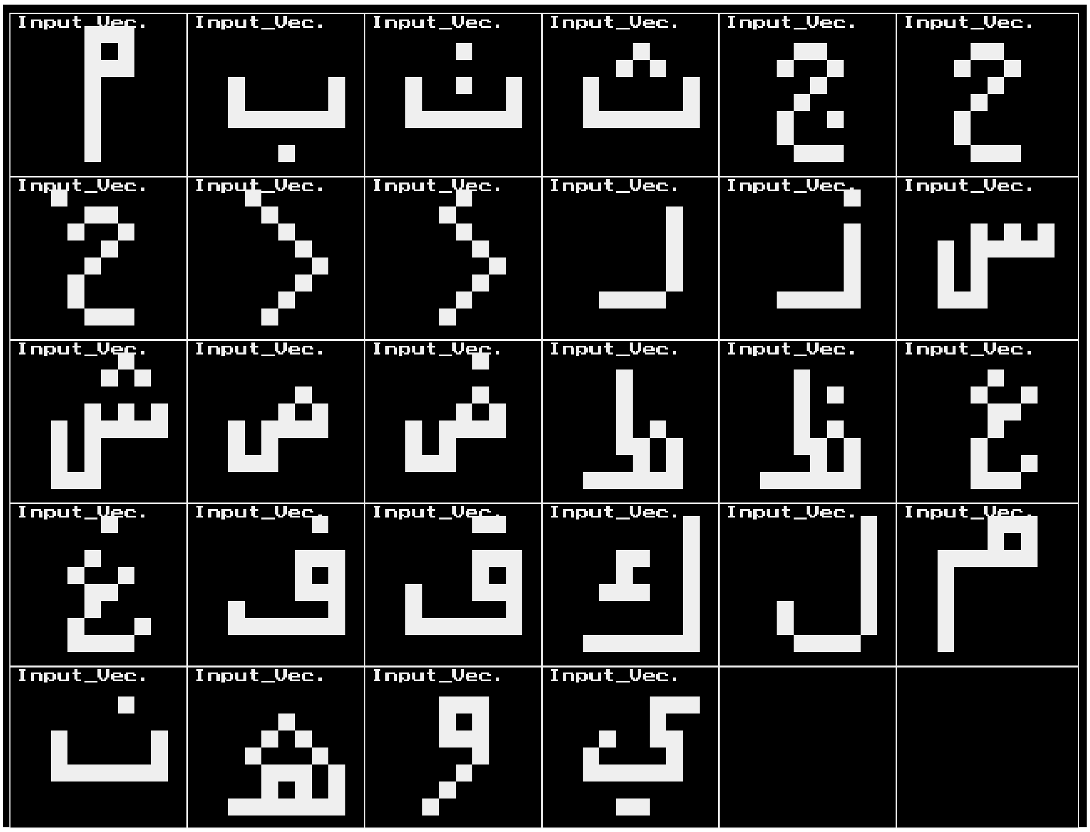
أ- قبل الكبس



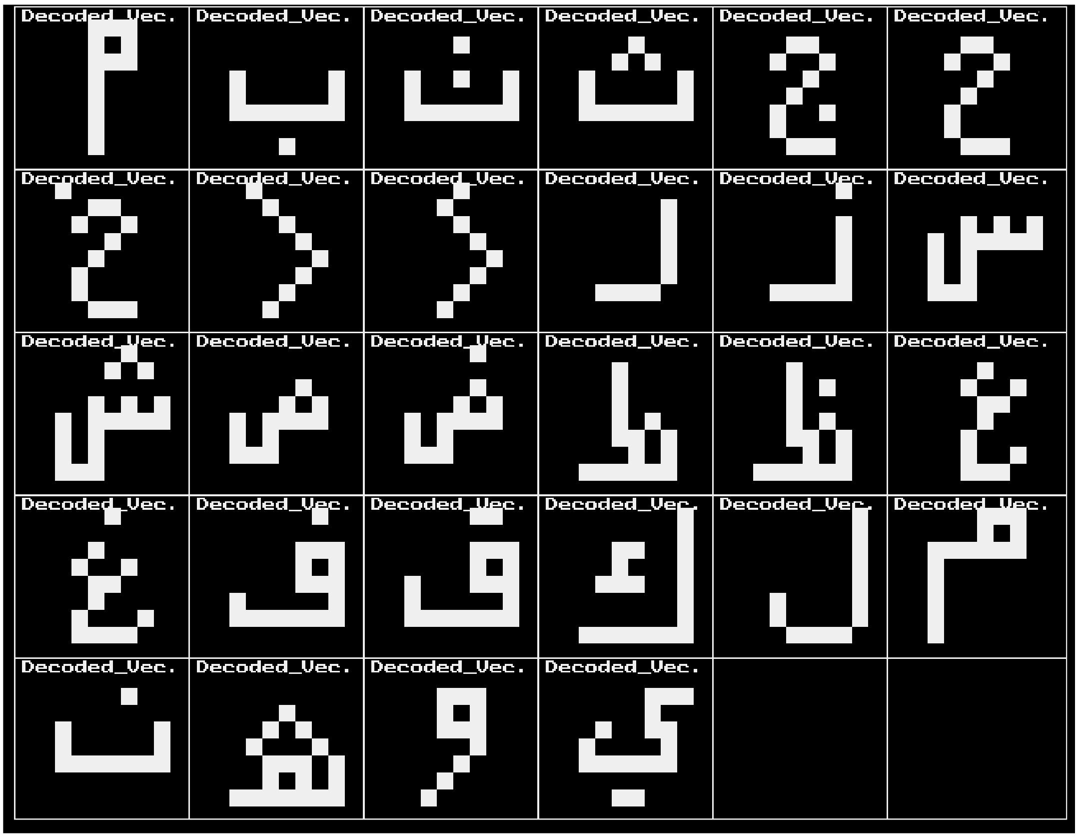
ب–النماذج بعد فك الكبس

الشكل (9) يوضح عملية الكبس للحروف الإنكليزية A..Z

أ– قبل الكبس ب – بعد فك الكبس



أ-النماذج الأصلية قبل الكبس



ب-النماذج بعد فك الكبس

الشكل (10) يوضح عملية الكبس للحروف العربية أ.ي

أ-النماذج الأصلية قبل الكبس ب – النماذج بعد فك الكبس

المصادر

[1] Abod L.K., (1998) *"***classification of satellite image using Neural Network***", Ph.D. Thesis, Department of physics, College of Science, University of Baghdad.*

[2] Baluja S., (june 1996) *"***Evolution of an Artificial Neural Network Based Autonomous Land Vehicle controller***", IEEE transaction or system, MAN, And Cybernetics-part Bi Cybernetics, Vol. 26, No. 3, PP.450-463.*

[3] Chen Fu-Chuang, (1988) *"***Backpropagation Neural network for Nonlinear Self-Tuning Adaptive Control***", IEEE*

[4] Chitra, S.P.(April 1993) *"***use Neural Networks for problem Solving**", *Chemical Engineering progress, PP. 44-52.*

[5] Lippmann R.P., (April 1987*)* "**Introduction to Computing With Neural Nets"**, *IEEE Assp Magazine, Vol. 4, No. 2, PP. 4-22*

[6] Rzempoluck E.J., (1998) *"***Neural Network Data Analysis Using Simulnet**", *Springer – Verlag NewYork.*

[7] Valluru, B. Rao and Hayagriva, V. Rao, (1993) *"***C++ Neural Networks and Fuzzy Logic***", Henry Holt and Company, Inc., NewYork*

[8] Wasserman, P. D., (1989) *"***Neural Computing Theory and Practice, Van Nostrand Reinhold***", NewYork*

[9] Werbos P.J. , (October 1990) *"***Backpropagation Through Time: What it Does and How to Do it***", IEEE, Vol. 78, No. 10, PP. 1550 – 1560.*

[10] Zurada J. M., (1994) *"****Introduction to Artificial Neural Systems****", Jaico Publishing House, Mumbai*