تشفير النصوص عن طريق استخدام شبكة(Hebbian)
الملائمة مع حساب الوقت

نضال حسين الاسدي **عامرة استقلال بدران**
كلية علوم الحاسبات والرياضيات
جامعة الموصل

Abstract

This research contains two parts, in the first part, a ciphering system is built using the classical Hebbian network to protect data against many expected threats during the transfer of the data. In the second part, deciphering has been built by using the Hebbian neural network.

The time has been calculate for both cipher and decipher. In the ciphering process, a Hebbian network has been developed through a qualitative primary weight which has large value. Then, an equation has been applied to minimize the weight matrix. Here, The idea of Stream Ciphering has been used so as to feed the network entries at the ciphering stage. The work has been applied by using (Visual Basic) language, issue (6.0) with the Object Oriented Programming (OOP) on a computer of the (P III, 600MHz) type.

الملـخص

يحوي هذا البحث على جزئين:

في الجزء الأول تم بناء خوارزمية للتشفير عن طريق استخدام شبكة (Hebbian) التقليدية لحماية البيانات ضد الكثير من التهديدات المتوقعة التي تتعرض لها أثناء نقل البيانات.أما الجزء الثاني فقد تم بناء خوارزمية لفك الشفرة عن طريق استخدام الشبكة العصبية(Hebbian).

وفي كلا الجزئين تم حساب الوقت المستغرق أي التشفير وفك الشفرة لمعرفة كم من الوقت تستغرق. وفي عملية التشفير تم تطوير شبكة (Hebbian) وذلك من خلال وزن أولي نوعي الذي يكون حجمه كبير، إذ تم تطبيق معادلة عليه مما أدى إلى تصغير مصفوفة الوزن . وقد تم استخدام فكرة التشفير الانسيابي (Stream Cipher) لغرض تغذية مداخل الشبكة في مرحلة التشفير. وضع هذا العمل قيد التطبيق باستخدام لغة (Visual Basic) الإصدار (6.0) مع أسلوب البرمجة الشيئية (OOP) وطبق العمل على حاسبة من نوع (PIII,600MHz).

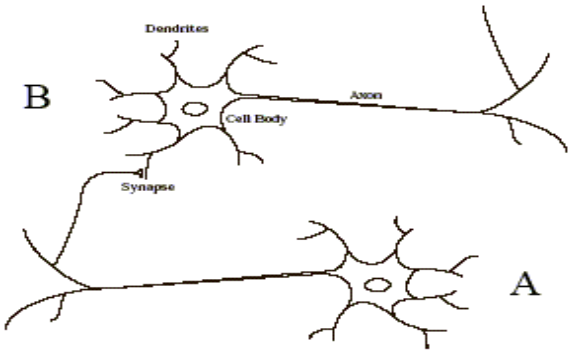
١. المقدمة

من الضروري التوجه إلى الشبكات العصبية الاصطناعية (Artificial Neural Networks) التي تعد من التطبيقات الحديثة في مجال الذكاء الاصطناعي إذ اعتمدت على أسس بيولوجية في محاولة محاكاة السلوك البشري[4] .

تم في هذا البحث اخذ فكرة التشفير الانسيابي (Stream Cipher) بدون استخدام القواعد المحددة لطريقة التشفير، ونتيجة لهذا العمل تم تغذية شبكة (Hebbian) بإدخالات الملف وتم إجراء عملية التشفير باستخدام هذه الشبكة ومن ثم تمت عملية فك الشفرة الناتجة من عملية التشفير أيضاً باستخدام شبكة (Hebbian).

٢. شبكة Hebbian:

لقد تم اكتشاف شبكة (Hebbian) من قبل العالم دونالد هيب (Donald Hebb)عام ١٩٤٩. حيث قدم العالم (Hebb) أول قاعدة لتعليم الشبكة العصبية اطلق عليها (Hebbian learning Rule) اعتمدت كقاعدة أساسية لتطوير خوارزميات التعليم[1][7]. وان الهدف من هذه الشبكة هو إعادة تعديل مصفوفة الوزن التي تمثل مصفوفة الارتباط بين العقد. أي انه في حالة تدريب شبكة (Hebbian) فان الأوزان بين عقد الشبكة سيتم تعديلها وفقاً لعلاقات التمثيل بين العقد. وقد تم اعتماد شبكة(Hebbian) بصورة أساسية في إعادة تعديل مصفوفة الوزن فاذا كانت احتمالية العقدة (A) تثير العقدة (B) بصورة عالية نسبياً، فان قوة الترابط بين العقدتين (A) و(B) سوف تزداد، وتسمى هذه الحالة بـ(Learning from Memory) وذلك لان الشبكة سوف تستخدم المعلومات المستنبطة من الأحداث السابقة لغرض تعديل الوزن بين العقد المترابطة كما في الشكل(١)[10] .



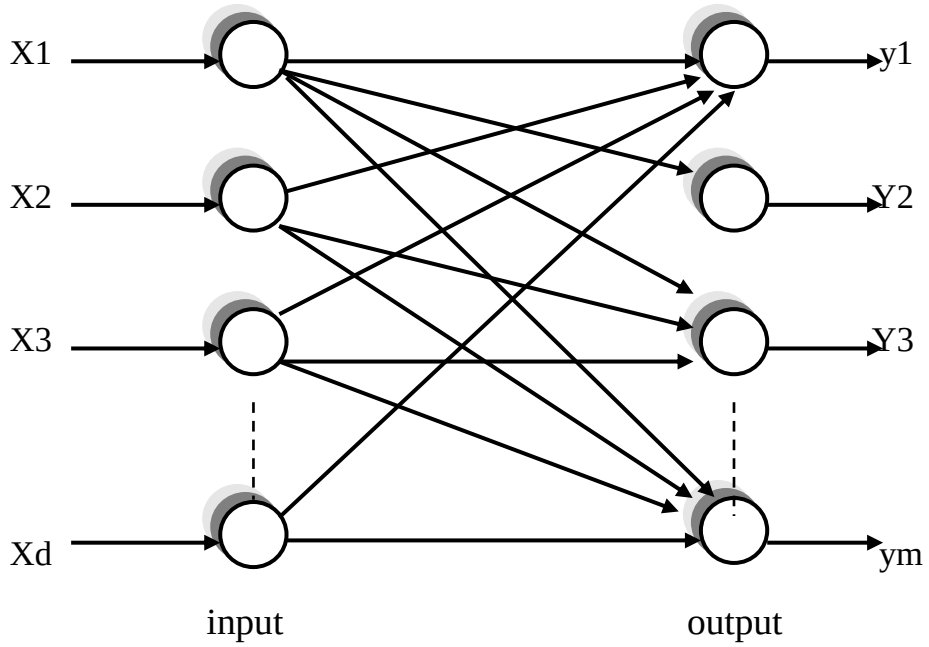
الشكل (١)

يوضح تأثير عقدة على عقدة أخرى

إن نموذج التعلم لشبكة (Hebbian) يمكن أن يقع تحت أسلوبين وهما: التدريب بمعلم، والتدريب بدون معلم[9] . وقد تم استخدام شبكة (Hebbian) بدون معلم في هذا البحث، وتمتاز شبكة (Hebbian)من نوع بدون معلم بأنها تتكون من طبقة واحدة (Single layer)، أما اتجاه سير العمل فيها يكون من نوع السير إلى الأمام (Feed forward). وتمتاز هذه الشبكة بعدم احتواء جسم الخلية على دالة التحفيز (Activation function)، بينما يحدث التحديث على الوزن (Weight)[6] .

معمارية شبكة (Hebbian):

يبين الشكل(٢) الآتي بصورة عامة معمارية شبكة (Hebbian) التي تتكون من طبقة واحدة (Single layer) ويتم فيه توضيح الإدخالات والاخراجات.



الشكل(٢)

مخطط يوضح معمارية شبكة (Hebbian)

لأجل الحصول على قيم الاخراج يمكن متابعة المعادلة الآتية:-

$$y_j(n) = \sum_{i=1}^{d} W_{ji}(n) * X_i(n) \rightarrow j = 1,2,....., m \qquad .....(١)$$

- لأجل تعديل مصفوفة الوزن فيمكن متابعة المعادلة الآتية:-

$$\Delta W_{ij}(n) = \eta(n) y_j(n) \left[X_i(n) - \sum_{k=1}^{j} W_{ki}(n) y_k(n) \right] \qquad ..... \qquad (٢)$$

$$for \qquad i = 1,2,....., d \qquad and \qquad j = 1,2,....., m$$

حيث ان:-

X_i = تمثل قيمة الادخال i .

y_j = تمثل قيمة الاخراج j .

W_{ij} =تمثل قيمة مصفوفة الوزن ij .

ΔW= التعديل في مصفوفة الوزن.

η = معامل التعلم $= \frac{1}{n^{\alpha}}$ $\frac{1}{2} < \alpha \leq 1$

m = عدد قيم الاخراج.

d = عدد قيم الادخال.

n = عدد الدورات.

خوارزمية شبكة Hebbian:-

يمكن التوصل الى خوارزمية شبكة (Hebbian) من خلال الخطوات الاتية:[6]

١. تهيئة مصفوفة الوزن بقيم عشوائية عند (n=1). وتخصيص قيمة موجبة صغيرة لـ(η).

٢. حساب المعادلتان (١)ثم (٢).

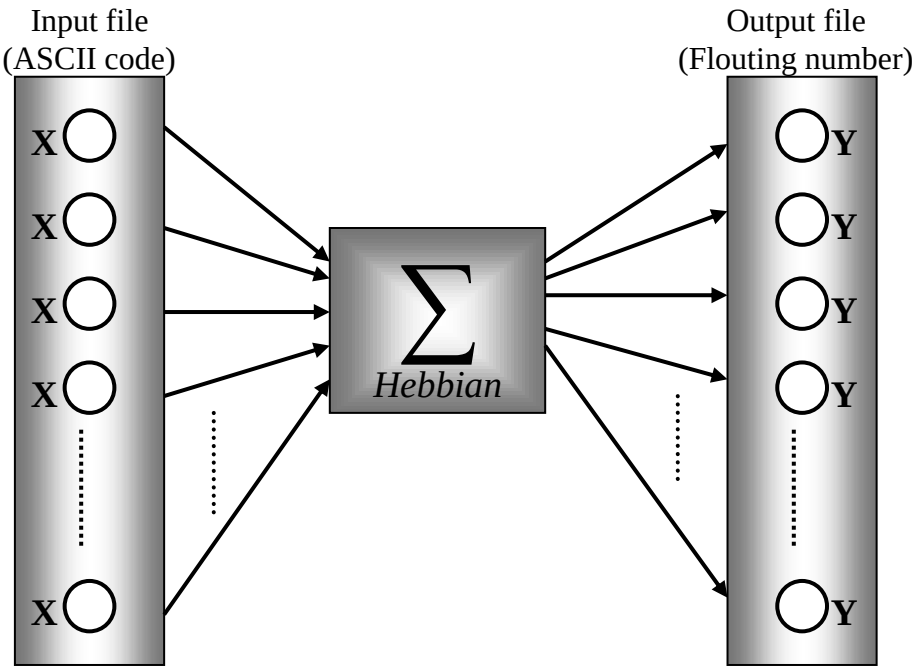
٣. زيادة (n) بقيمة واحدة والذهاب إلى الخطوة الثانية، الاستمرار بالدوران إلى حد الوصول إلى القيمة الثابتة.

لقد تم توضيح خطوات العمل في خوارزميتين الاتيتين:[3]

الخوارزميـة الأولـى تتضـمن خطـوات عمليـة تشـفير الـنص المـدخل إلـى شـبكة (Hebbian). الخوارزمية الثانية تتضمن خطوات فك تشفير النص باستخدام شبكة (Hebbian).وبعد تطبيق خوارزمية التشفير وفك الشفرة يتم حساب الوقت المستغرق لكل حالةً.ولقد تم استخدام شبكة (Hebbian) في التشفير وذلـك لان هـذه الشـبكة تمتـاز بأنهـا خطيـة Linear أي ان لهـا أسـلوب خطـي فـي عمليـة حسـاب الاخراجات[10]، فضلاً عن انها لا تحتوي على طبقة خفية (Hidden layer) مما يؤدي إلى زيادة سرعة التدريب نسبياً. إضافة إلى سهولة استخدام معادلات حساب الاخراجات ومعادلات تعديل الوزن، وكذلك عدم احتوائها على دالة التحفيز(Activation function)[6] .

٣ . خوارزمية التشفير:

- تبدأ عملية التشفير باستخدام شبكة (Hebbian) باعتماد الخوارزمية الآتية (شكل(٣))[3]:-



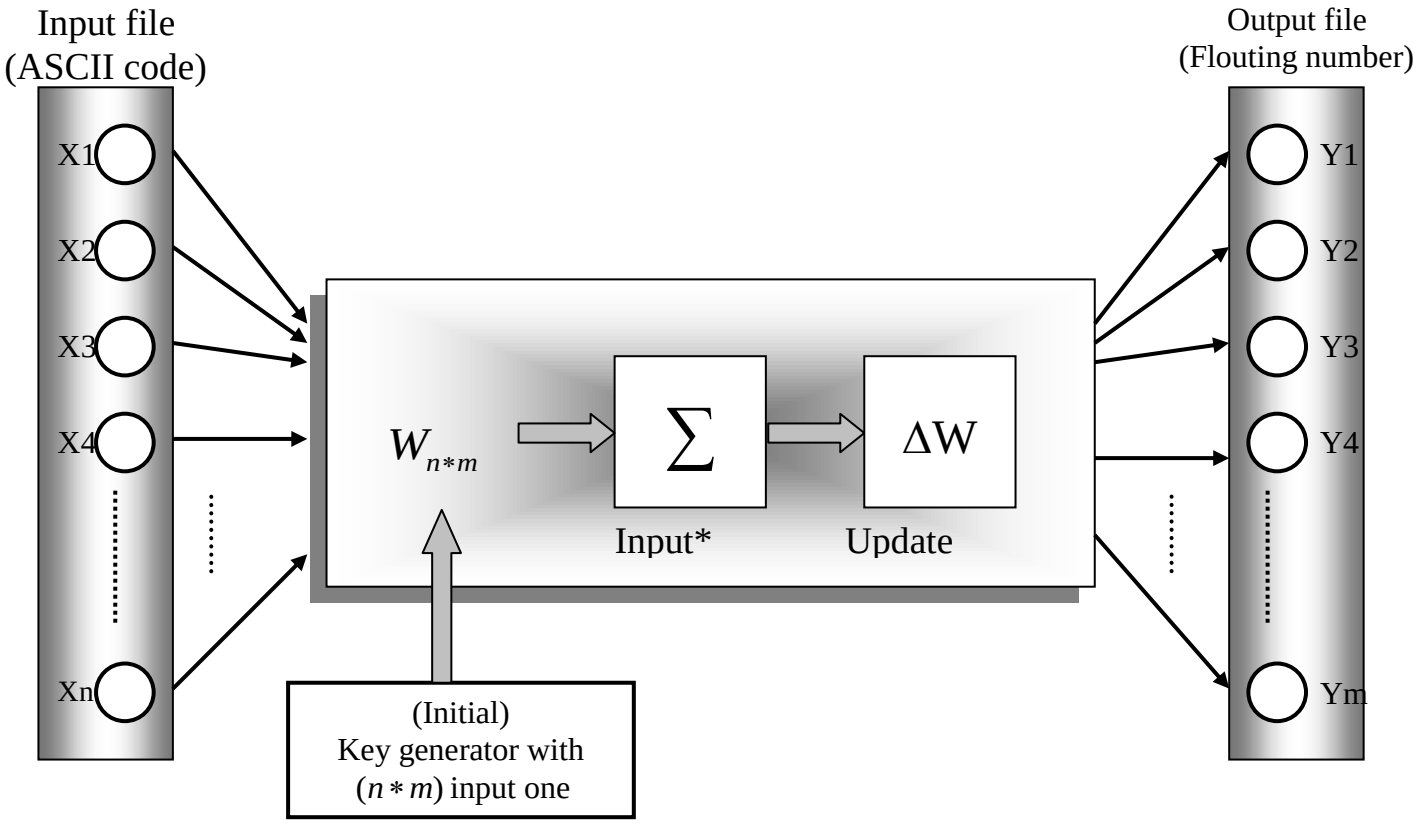
الشكل(٣)

مخطط عام لعملية التشفير باستخدام شبكة(Hebbian)

١- فتح الملف المراد تشفيره .

٢- تهيئة حجم مصفوفة الوزن (Weight) بحجم الملف.

٣- توليد مفاتيح عشوائية وإحلالها إلى مصفوفة الوزن (شكل (٤)).

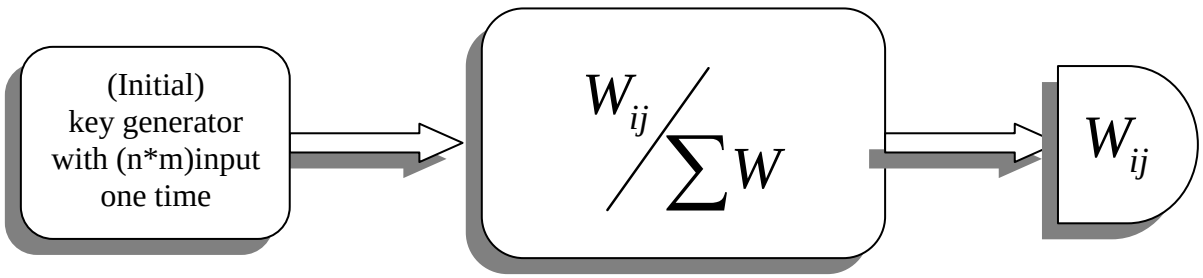


الشكل (٤)

مخطط تفصيلي لعملية التشفير باستخدام شبكة**(Hebbian)**

٤- أجراء عملية (Normalization) لمصفوفة الوزن لتجنب حصول حالة (Over flow) للقيم وذلك بالاعتماد على معادلة التبسيط الآتية:

$$W_{ij} = W_{ij} \Big/ \sum W \qquad \ldots\ldots \qquad (٣)$$



الشكل(٥)

مخطط تفصيلي يوضح عملية (Normalization) لمصفوفة الوزن

إن إجراء عملية التعديل على مصفوفة الوزن وذلك من اجل تصغير قيم مصفوفة الوزن التي تم تكوينها بطريقة عشوائية (Random) كما في الشكل أعلاه (الشكل(٥)).

٥- تحديد إدخالات الشبكة من خلال القيم المقابلة لكل حرف (ASCII code) في الملف.

٦- البدء عند (n=1).

٧- تهيئة قيمة صغيرة لـ(η).

٨- حساب اخراجات الشبكة من خلال اعتماد معادلة رقم (١).

٩- تعديل مصفوفة الوزن من خلال اعتماد معادلة رقم(٢).

١٠- زيادة قيمة (n) بقيمة واحدة.

١١- الرجوع إلى الخطوة (٧).

بعد الانتهاء من عملية التدريب لـ(n) لخطوات عدة، يتم الحصول على قيم الاخراجات التي تمثل القيم المشفرة المقابلة لكل حرف مدخل. وتمثل عدد عقد الاخراجات بعدد عقد الإدخالات أي ان (d=m).

تعتبر مصفوفة الوزن الناتجة من الدورة الأخيرة لـ(n) هي المفتاح الذي سوف يتم استخدامها في عملية فك الشفرة[8] .ومن الجدير بالذكر انه قد تم الاستفادة من فكرة التشفير الانسيابي(Cipher Stream) في عملية تغذية شبكة (Hebbian) بالإدخالات[3].

٤. خوارزمية فك الشفرة:

تبدأ عملية فك الشفرة باستخدام شبكة (Hebbian) من خلال اعتماد الخوارزمية الآتية (شكل(٦))[3]:-

١. فتح الملف المشفر وتهيئته كإدخالات .

٢. تهيئة مصفوفة الوزن (Weight) التي تمثل مصفوفة المفاتيح بقيم مصفوفة الوزن الناتجة من عملية التشفير.

٣. حساب معكوس المصفوفة (Inversion) لمصفوفة الوزن، (أي اخذ آخر مصفوفة تم الحصول عليها من عملية التحديث على الـ(Weight) التي تم إجراءها بتطبيق معادلة رقم(2) ومن ثم حساب معكوس المصفوفة(Inversion).

٤. حساب اخراجات الشبكة من خلال اعتماد معادلة رقم (1).

٥. تمثلت اخراجات الشبكة القيم الصريحة للنص الأصلي (ASCII code).

ولقد تم محاولة تهجين شبكة (Hebbian) مع الخوارزمية الجينية لعملية التشفير وفك الشفرة، إذ كانت الخوارزمية الناتجة تسمى بالخوارزمية الهجينة ولكن عند إجراء العمل تبين أن العملية تكاد أن تكون من الناحية العملية مستحيلة وذلك لان الخوارزمية الجينية تعتمد على تشفير قياسي أما الشبكة العصبية فهي تعتمد على التشفير غير القياسي وبهذا أصبح وجود تناقض ما بين الطريقتين بسبب هذا الاختلاف. وذلك لصعوبة تحديد معادلة مدى اللياقة (Fitness value)[3].

5. خوارزمية حساب الوقت:

إن عملية حساب الوقت المستغرق في عملية التشفير أو فك الشفرة يتم حسابه باعتماد الخوارزمية الآتية:[3]

١. قراءة قيمة الوقت قبل عملية التشفير أو فك الشفرة بـ(دقيقة/ثانية).

٢. إجراء عملية التشفير أو فك الشفرة.

٣. قراءة قيمة الوقت بعد الانتهاء من عملية التشفير أو فك الشفرة بـ(دقيقة/ثانية).

٤. حساب الوقت لإجراء عملية التشفير أو فك الشفرة(الوقت المستغرق) باعتماد المعادلة الآتية:

الوقت المستغرق=الوقت بعد العملية(تشفير أو فك الشفرة)– الوقت قبل العملية(تشفير أو فك الشفرة).

٦. التطبيق العملي للخوارزمية

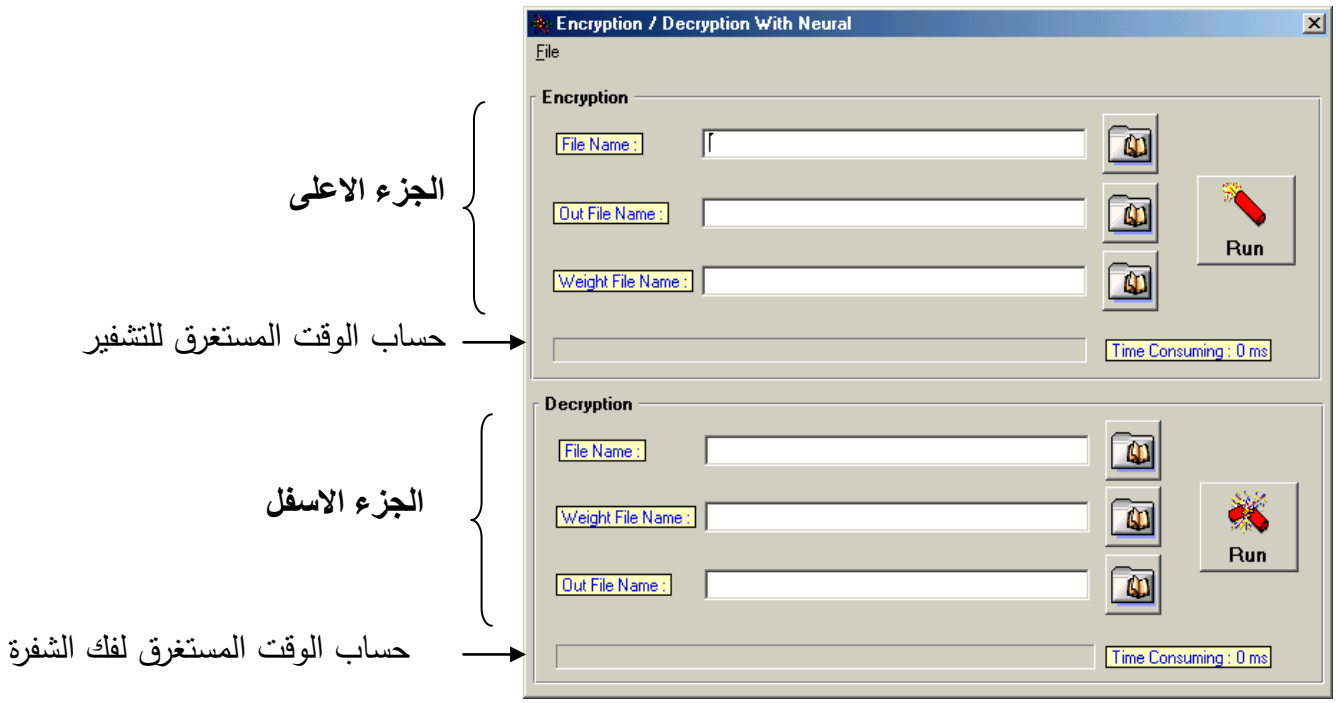
في هذا الجزء من الفصل سوف نتطرق إلى عملية تنفيذ البرنامج الذي تم تنفيذه باستخدام لغةVisual Basic 6.0[2][5] ويمكن تتبع خطوات التنفيذ عن طريق المراحل الآتية:

7. الجزء التنفيذي للبرنامج

عند تنفيذ البرنامج سوف تظهر الشاشة الآتية كما في الشكل (٧) والتي تحوي علىجزئين:

الاول: الجزء الاعلى خاص بالتشفير مع حساب الوقت المستغرق للتشفير .

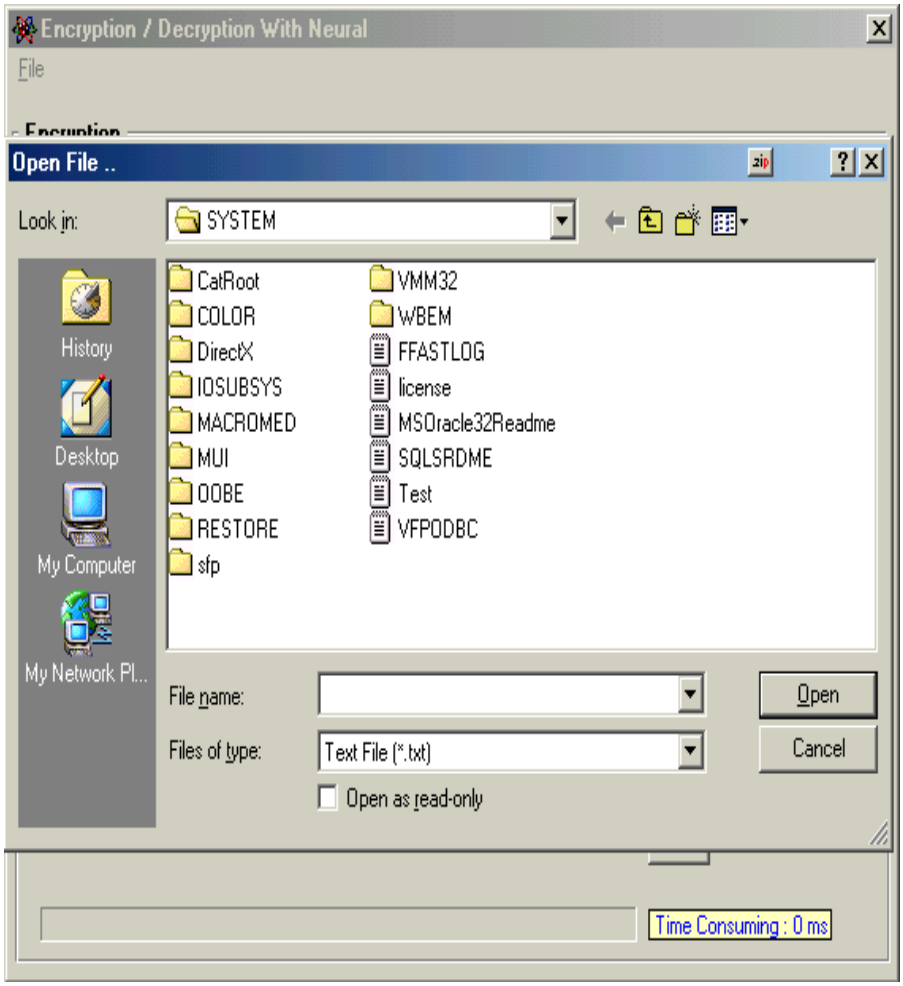
الثاني: الجزء الاسفل خاص بفك الشفرة مع حساب الوقت المستغرق لفك الشفرة.



الشكل (٧)

نافذة بداية تنفيذ البرنامج

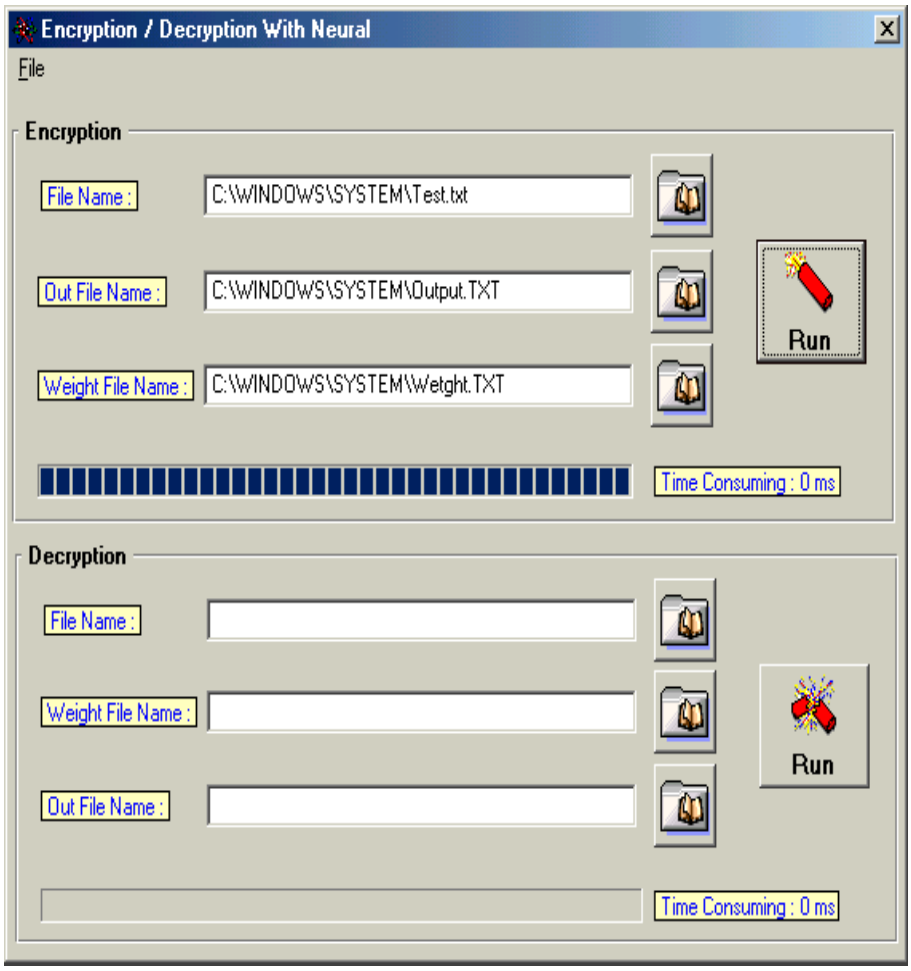
عند النقر على هذا الايقون تظهر الشاشة كما في الشكل (٨) التي تطالب المستخدم بتحديد الملف المطلوب تشفيره



شكل (٨)

نافذة تحديد الملف

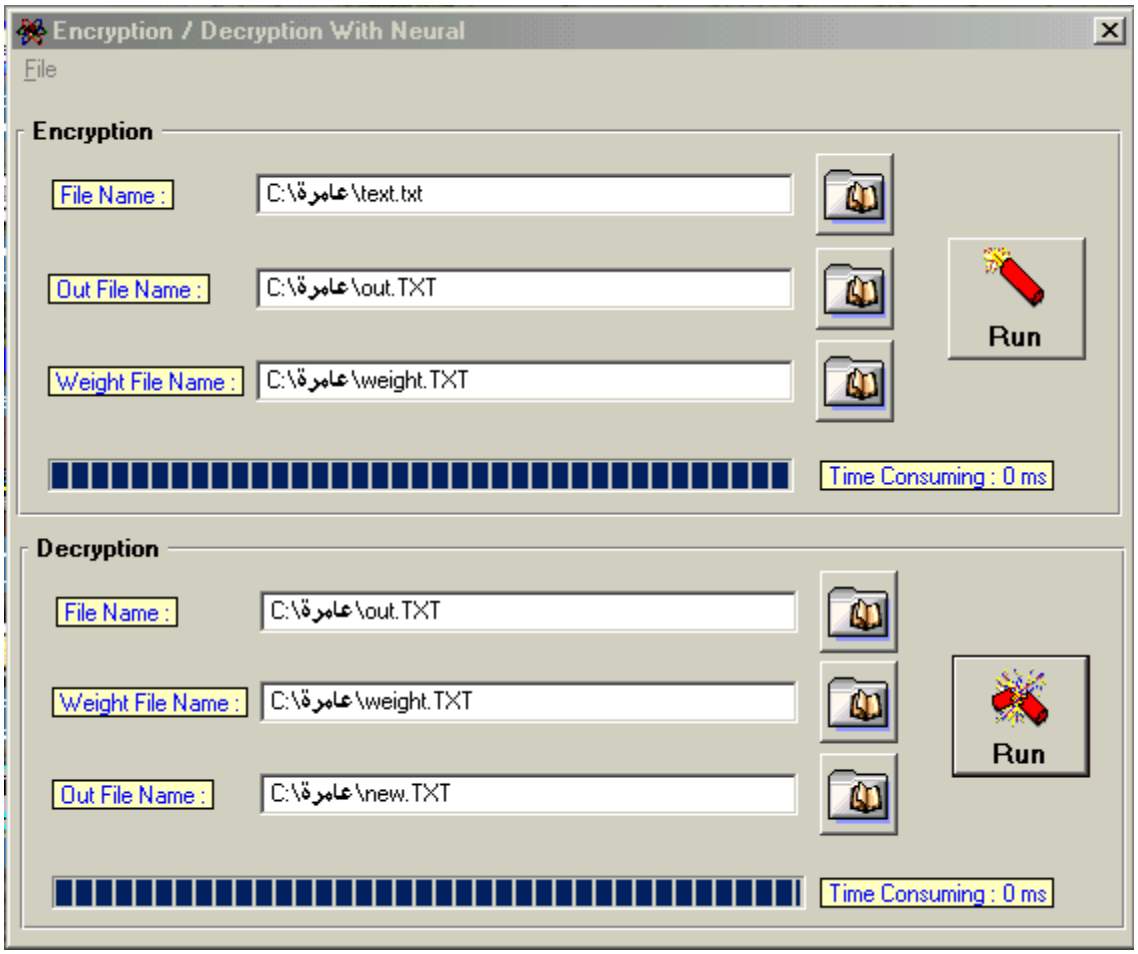
عند النقر على هذا الايقون تبدأ عملية التشفير كما في الشكل (٩) والخاص بالجزء العلوي.



الشكل (٩)

نافذة عملية التشفير

عند النقر على هذا الايقون تبدأ عملية فك الشفرة كما في الشكل (١٠) والخاص بالجزء السفلي.

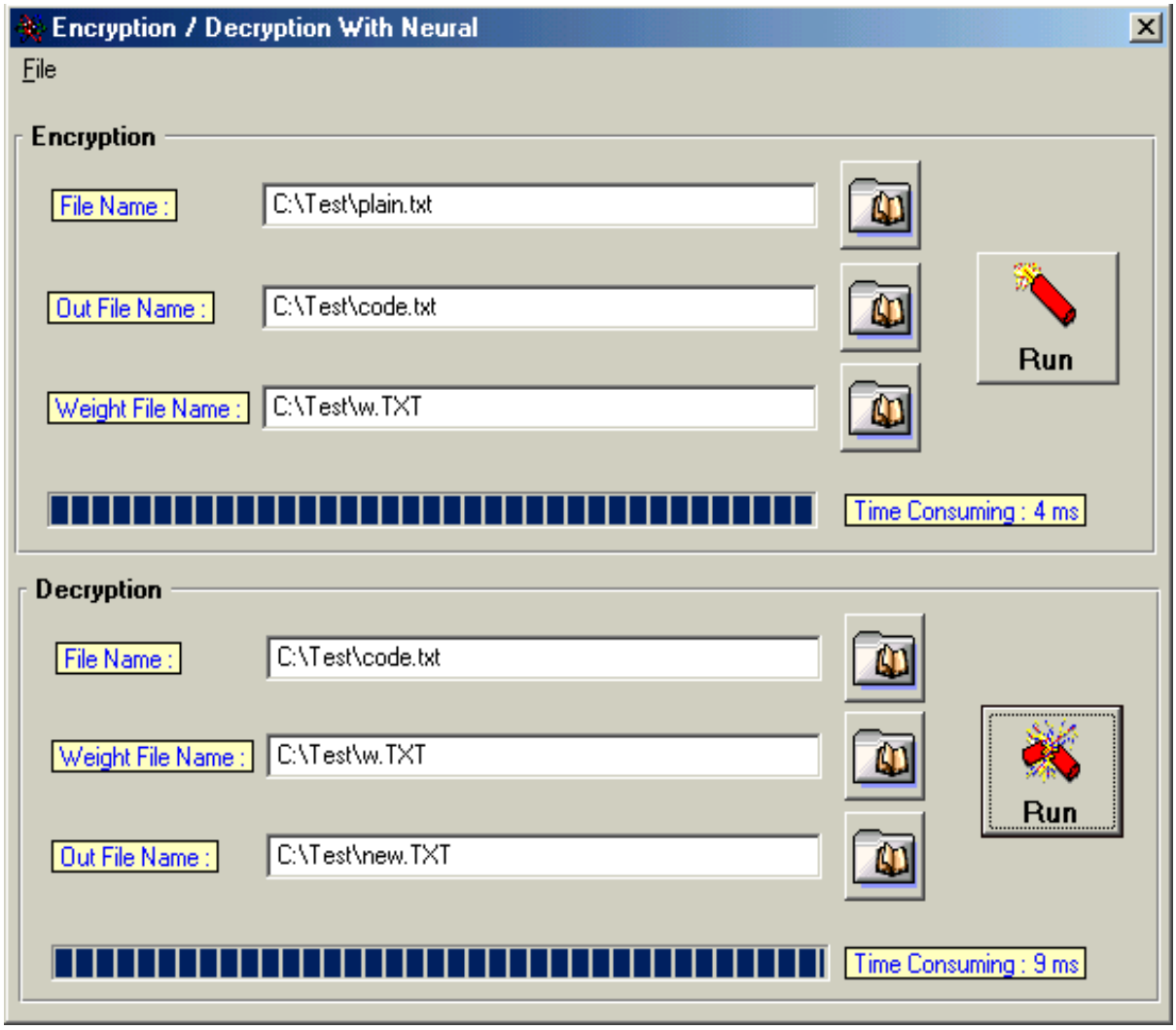


الشكل(١٠)

نافذة عملية فك الشفرة

يجب ملاحظة الوقت المستغرق في عملية التشفير وفك الشفرة بأجزاء الثانية وكما واضح على شاشة التنفيذ (Time Consuming) (الشكل(١٠)) مع ملاحظة إن المثال أعلاه نفذ على نص ذات حجم صغير جداً لذلك فان الوقت المستغرق في التشفير وفك الشفرة يساوي صفر.

مثال توضيحي آخر للوقت المستغرق في التشفير وفك الشفرة، حيث أن الوقت المستغرق في التشفير 4ms والوقت المستغرق في فك الشفرة 9ms ، نلاحظ بان الوقت أصبح أطول في التشفير وفك الشفرة وذلك لكبر حجم النص المشفر كما في الشكل(١١) .



الشكل (١١)

نافذة حساب الوقت

٨. نتائج تنفيذ البرنامج:

❖ مثال الأول:

- محتويات الملف (Test.txt) قبل عملية التشفير:

الصلاة والسلام على سيدنا محمد

- محتويات الملف (Output.txt) بعد عملية التشفير والتي تمثل القيم المقابلة لكل حرف:

706526	**708949**	**700087**	**734204**	**751943**	**760865**	**661419**
816768	**724433**	**751138**	**718914**	**742737**	**651712**	**805492**
716994	**778714**	**729870**	**660690**	**675075**	**676841**	**920238**
774103	**649365**	**790856**	**759235**	**654780**	**664285**	**670931**
757549						

- محتويات الملف (New.txt) بعد عملية فك الشفرة:

الصلاة والسلام على سيدنا محمد

❖ مثال الثاني:

- محتويات الملف (Test.txt) قبل عملية التشفير:

Microsoft Corporation

- محتويات الملف (Output.txt) بعد عملية التشفير والتي تمثل القيم المقابلة لكل حرف:

229472	**270831**	**350196**	**224732**	**316244**	**366492**	**232854**
277980	**229456**	**295496**	**171597**	**280570**	**277579**	**276161**
304861	**294848**	**317306**	**298337**	**347712**	**332726**	**282559**

محتويات الملف (New.txt) بعد عملية فك الشفرة:

Microsoft Corporation

الاستنتاجات

إن عملية التشفير باستخدام الشبكة العصبية(Hebbian) تتضمن نوعاً من السرية العالية بسبب كبر المفتاح نسبياً وذلك لكونه يمثل مصفوفة ثنائية ذات أبعاد بحجم النص. وان عملية فك الشفرة تتطلب وجود (مصفوفة الوزن) التي تمثل مفتاح (فك الشفرة). بالإضافة إلى أن حجم الملف الناتج من عملية التشفير يكون اكبر من حجم الملف الأصلي لان طبيعة البيانات الناتجة تكون أرقاماً حقيقية (Floating Number). ومن المهم أن نعرف إن الوقت المستغرق في عملية التشفير يكون (بالاعتماد على عدد الدوراتNumber of iteration)، ومتناسب بشكل طردي مع حجم النص بالإضافة إلى أن الوقت المستغرق في عملية فك الشفرة يكون كبيرا وذلك بسبب عملية الحساب لمعكوس المصفوفة (Inversion). لقد تم التأكد من صحة النتائج في عملية فك الشفرة بنسبة (١٠٠%).

التوصيات:

١. استخدام شبكة تتكون أكثر من طبقة للمقارنة بين الوقت المستغرق في عملية التشفير وشبكة (Hebbian) أحادية الطبقة.
٢. استخدام فكرة التشفير الكتلي (Block Cipher) في عملية تغذية مداخل الشبكة في حالة كون الملف المشفر كبيراً.
٣. القيام على توفير سرية عالية على (مصفوفة الوزن) الناتجة من عملية التشفير (مصفوفة المفاتيح).
٤. تطبيق فكرة التهجين في (Neural) وخاصة مع شبكة (Hebbian) للحصول على (Typical initial weight) للحصول على (Weight) أمثل. وممكن أن نستغني عن معادلة (Weight) في شبكة(Hebbian) باستخدام الخوارزمية الجينية.
٥. إدخال عملية كبس لمصفوفة الوزن باستخدام أسلوب الكبس بالشبكات العصبية (شبكة Backprobagation) لغرض تقليص حجم المصفوفة.
٦. استخدام شبكة تحتوي على دالة التحفيز (Activation functionِ) لان شبكة (Hebbian) لا تحتوي على دالة التحفيز (Activation functionِ).

المصادر

١. العبيدي، محمود خليل ابراهيم (٢٠٠٠): "الشبكات العصبية الاصطناعية"، مجلة أبحاث الحاسوب، مدرس علم الحاسوب، الجامعة التكنلوجية، بغداد.

٢. الناظر، سائد محمود (١٩٩٧): "كتاب المبرمج Visual basic 5.0"، دار شعاع للنشر والعلوم، سورية حلب، الطبعة الأولى.

٣. بدران، عامرة استقلال (2003):"استخدام شبكة (Hebbian) في التشفير"بحث ماجستير، كلية علوم الحاسبات والرياضيات، جامعة الموصل.

٤. سليمان، أنعام محمد (٢٠٠٢): "التداخل الشبكي الجيني (GA-Hf) لحل المسائل من نوع Np-problem (TSMP)"، بحث ماجستير، كلية علوم الحاسبات والرياضيات، جامعة الموصل.

٥. هالفرسون، مايكل (١٩٩٩): " Visual basic 6.0 خطوة خطوة"، الدار العربية للعلوم، لبنان.

6. Haykin, S. [1999]: "Neural Network: A comprehensive Foundation", Second edition, prentice Hall, London. ,CH2.
7. Patterson, Dan W. (1996): ‘‘Artificial neural networks, theory and application’’, Prentice Hall.

Web reference:

8. WWW.comp.glam.ac.uk/digimag.
9. WWW.csse.monash.edu.au/~app/L01.pdf.
10. WWW.cs.hmc.edu/courses/ch07-pres.pdf.