مشروع تدريب شبكة عصبونية للحصول على تابع التربيع:

عمل الطالبتين:

آية بج 205012

رنا نوح 205011

# تدريب الشبكة العصبونية:

تدريب الشبكات العصبية الصنعية يقصد به إيجاد القيم الأنسب لكافة أوزان الدخل لعصبونات الشبكة بحيث تكون قادرة على إعطاء الإجابات الصحيحة بأقل هامش خطأ في مخرجاتها وذلك للمدخلات المعروضة عليها، في حين يقع على عاتق المطور فن تحديد المعمارية الملائمة من حيث عدد الطبقات ونوع تابع تفعيل كل منها وعدد عصبوناته وطريقة ربطها مع بعضها البعض.

قد تكون مهمة تدريب الشبكات العصبية الصنعية تحديا حسابيا هائلا تتطلب كما كبيرا من المعالجات الرياضية والتي قد تستغرق زمنا طويلا حتى لو أجريت على حواسيب فائقة، لكن بمجرد الوصول إلى الحل وإيجاد قيم الأوزان الملائمة التي تعطي نتائج مرضية عند معالجتها للبيانات، تصبح مسألة الاستخدام والتطبيق في منتهى البساطة، فكل ما عليك القيام به هو تمرير وحيد عبر طبقات الشبكة يتم خلاله إجراء بضع عمليات ضرب وجمع ومن ثم تطبيق تابع رياضي ما (تابع التفعيل)، وهكذا تمرر القيم من طبقة إلى أخرى وصولا إلى طبقة الخرج النهائية لتظهر من خلالها الإجابة.

# كيفية حفظ وتصدير الشبكات العصبية الصنعية:

بعد إتمام عملية تدريب الشبكة العصبية الصنعية والرضى عن نتائجها، كل ما نحتاج إليه هو حفظ بنيتها من حيث عدد الطبقات، وعدد العصبونات في كل طبقة، وطريقة ارتباط عصبونات الطبقات المختلفة بعضها ببعض، وكذلك تابع التفعيل المستخدم في كل طبقة، إضافة إلى قيم الأوزان لكل عصبون على حدة. هذا كل ما في الأمر! فإن حصلت على هذه الأرقام مجددا فأنت قادر على إعادة بناء واستخدام تلك الشبكة العصبية الصنعية بشكلها النهائي بعد أن اكتملت عملية تدريبها.

# كيف تعمل الشبكات العصبونية؟

الدماغ البشري هو مصدر الإلهام لهندسة الشبكات العصبونية. تشكل خلايا الدماغ البشري، التي يطلق عليها العصبونات، شبكة معقدة ومترابطة للغاية وترسل إشارات كهربائية إلى بعضها لمساعدة البشر على معالجة المعلومات. على نحو مماثل، تتكون الشبكات العصبونية الاصطناعية من عصبونات اصطناعية تعمل معًا لحل مشكلة ما. العصبونات الاصطناعية هي وحدات برمجية تطلق عليها العقد، بينما تعرف الشبكات العصبونية الاصطناعية بوصفها برامج برمجية أو لوغاريتمات تستخدم بشكل أساسي أنظمة حوسبة لحل العمليات الحسابية.

# هندسة الشبكات العصبونية البسيطة:

تتكون الشبكات العصبونية الأساسية من عصبونات اصطناعية مترابطة في ثلاث طبقات:

1. **طبقة الإدخال:**

تدخل المعلومات من العالم الخارجي إلى الشبكات العصبونية الاصطناعية عبر طبقة الإدخال. تعالج عقد الإدخال البيانات أو تحللها أو تصنفها وتمررها إلى الطبقة التالية.

1. **الطبقة الخفية:**

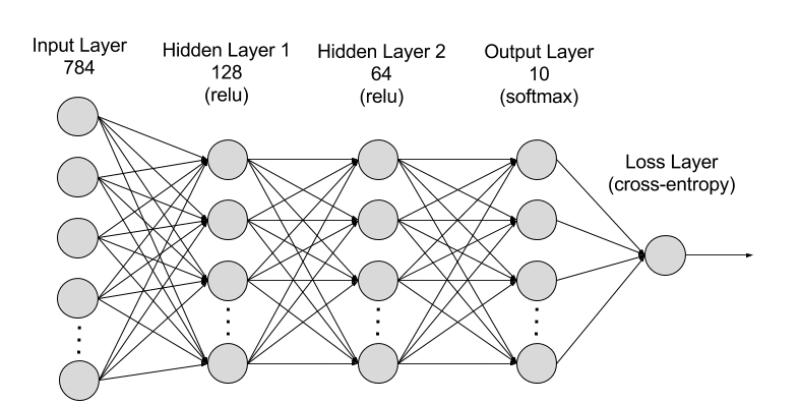
تأخذ الطبقات الخفية مدخلاتها من طبقة الإدخال أو الطبقات الخفية الأخرى. يمكن أن تتضمن الشبكات العصبونية الاصطناعية عددًا كبيرًا من الطبقات الخفية. تحلل كل طبقة خفية المخرجات من الطبقة السابقة وتعالجها بشكل أكبر وتمررها إلى الطبقة التالية.

1. **طبقة الإخراج:**

تعطي طبقة الإخراج النتيجة النهائية لجميع عمليات معالجة البيانات التي أجرتها الشبكات العصبونية الاصطناعية. وقد تتضمن عقدًا فردية أو متعددة. على سبيل المثال، إذا كانت لدينا مشكلة تصنيف ثنائي بـ (نعم/ لا)، فستتضمن طبقة الإخراج عقدة إخراج واحدة تعطي النتيجة 1 أو 0. على الرغم من ذلك، إذا كانت لدينا مشكلة تصنيف متعدد الفئات، فقد تتألف طبقة الإخراج من أكثر من عقدة إخراج واحدة.

# هندسة الشبكات العصبونية العميقة:

تتكون الشبكات العصبونية العميقة أو شبكات التعلم العميق من طبقات خفية عديدة مع ملايين العصبونات الاصطناعية المترابطة معًا. يمثل العدد، الذي يعرف بالوزن، الروابط بين العقدة الواحدة وغيرها من العقد. يكون الوزن عددًا موجبًا إذا كانت عقدة تحفز من عقدة أخرى أو عددًا سالبًا إذا كانت عقدة تلغي عقدة أخرى. تتميز العقد ذات قيم الوزن الأعلى بتأثير أكبر في غيرها من العقد.  
نظريًا، يمكن للشبكات العصبونية العميقة تعيين أي نوع من المدخلات إلى أي نوع من المخرجات. ومع ذلك، فإنها بحاجة كذلك إلى تدريب أكثر بكثير مقارنةً بأساليب التعلم الآلي الأخرى. إنها بحاجة إلى ملايين الأمثلة على بيانات التدريب بدلاً من المئات أو الآلاف التي قد تحتاج إليها شبكات أبسط.

عنوان المشروع:

# تدريب شبكة عصبونية من الصفر لتربيع القيم

محاور المشروع:

* التدريب على جزء من المجال
* زيادة عدد الطبقات
* تعديل توابع التفعيل
* تغيير عدد محاولات التدريب
* تغيير عدد العصبونات
* تغيير عدد عينات التدريب

الكود الذي نستطيع من خلاله تدريب هذه الشبكة لتعمل على تربيع القيم:

p= [1:10];

t=[p.^2];

net=newff (p, t, [3], {'logsig' 'purelin'});

net. divideparam. trainRatio=1;

net. divideparam. testRatio=0;

net. divideparam. valRatio=0;

net.trainparam.min\_grad=1e-20;

net. trainparam. goal=1e-30;

net. trainparam. epochs=300;

net=train (net, p, t);

plot([1:100].^2,'x')

hold on

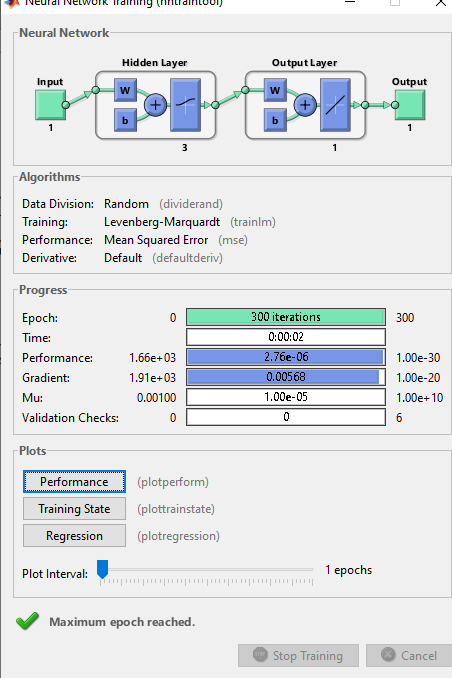
plot (round (net (1:100)),'o')

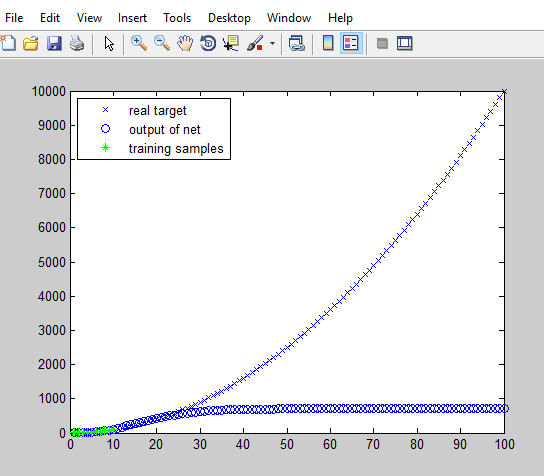
hold on

plot (p, t,'\*g')

legend ('real target', 'output of net', 'training samples','Location','northwest')

بعد كتابة الكود السابق باستخدام بيئة الماتلاب نحصل على النتائج التالية:





ولرسم الشبكة ضمن بيئة ال SIMULINKنستخدم التعليمة:

Genism (net, -1)

بعد تنفيذ هذه التعليمة يظهر لدينا الشكل التالي:

