كلية الهندسة-جامعة المنصورة

Lab

C2113

معمل الحاسبات الشخصية (ب) C2113

Computer and Control Systems Engineering Departments

قسم هندسة الحاسبات ونظم التحكم

Laboratory Book

Computer and Control Systems Engineering Departments

2021

معمل الحاسبات الشخصية (ب) C2113

Laboratory Book

Table of Contents

1. Laboratory Basic Information……1

١ .....البيانات الأساسية للمعمل

2. Laboratory Instruments……2

٢ .....ثانيا قائمة بالأجهزة والمعدات الموجودة بالمعمل

3. Laboratory Beneficiaries……3

٣.....ثالثاً الخدمات المجتمعية التى يؤديها المعمل

4. Laboratory Student Beneficiaries……4

٤.....رابعاً الخدمات الطلابية التى يؤديها المعمل

5. Laboratory experiments ……٥

٥.....خامسا التجارب المعمليه

Part

١

1. Laboratory Basic Information

أولاً: البيانات الأساسية للمعمل

اسم المعمل	الحاسبات الشخصية (ب) (الرقم الكودى : C2113)
القسم العلمى	هندسة الحاسبات ونظم التحكم
مشرفى المعمل	أ.د/ أميرة يسن هيكل د./ محمد صبرى سراية
مهندس المعمل	م./ عبير الكومى
أمين المعمل	أ./ ثناء محمد زينة
التليفون	داخلى ١٢٧٤
الموقع بالنسبة للكلية	الناحية القبلية
مساحة المعمل	١٥٠ متر مربع

Part

٢

2. **Laboratory Instruction**

ثانياً : قائمة بالأجهزة والمعدات الموجودة بالمعمل

م	اسم الجهاز (المعالج)	العدد	Mother Board
١	I3.7100.3.9 GHZ	٢١ جهاز	Giga Byte H110M-H

Part

3

3:-Laboratory Beneficiaries

ثالثاً الخدمات المجتمعية التى يؤديها المعمل:-

- عقد ورش عمل للفرق العلمية والأسر الطلابية بشكل دورى.
- ورش عمل لطلاب مشاريع التخرج بالقسم العلمى والبرامج المميزة.
- اجتماعات طلاب الدراسات العليا والخريجين للتطوير ودراسة احتياجات سوق العمل.

Part 4

4: Laboratory Student Beneficiaries

رابعاً الخدمات الطلابية التى يؤديها المعمل:

عدد الطلاب المستفيدين من المعمل	فى خلال مدة أسبوع ٢٥٠ طالب
الأقسام العلمية المستفيدة من المعمل	قسم هندسة الحاسبات ونظم التحكم
الفرق الدراسية المستفيدة من المعمل	إعدادى + ٤ فرق القسم
المقررات الدراسية التى تستفيد من المعمل	برمجة للفرقة الإعدادى
الأنشطة الطلابية داخل المعمل	ندوات طلابية - اجتماعات اللجنة العلمية للقسم - عقد جلسات مناقشة حلقات البحث لطلاب تمهيدى الماجستير - مناقشة مشاريع التخرج لطلاب قسم هندسة الحاسبات ونظم التحكم - عقد جلسات الامتحانات الشفوية لمقررات التحكم الآلي - عقد الامتحانات العملية لمقررات التحكم - التدريب الصيفى لطلاب الصف الاول بقسم هندسة الحاسبات ونظم التحكم - عمل مشاريع التخرج واللقاءات بين الطلبة للتجهيز للمسابقات.
عدد طلاب الدراسات العليا المستفيدين من المعمل	طلاب القسم كل عام
عدد الدورات التدريبية التى تمت فى المعمل	٢ دورتين في الأنظمة المدمجة
المسابقات العملية التى شارك فيها طلاب من المستفيدين من المعمل	Ten in Black

Part

5

5: Laboratory experiments

Experiment1: Batch Gradient Descent with Momentum

بيانات عامه:

اسم التجربه: Batch Gradient Descent with Momentum

الفرقه المقرر عليها التجربة : الرابعه

الفصل الدراسى : الثانى

الأدوات المطلوبة: جهازحاسب متقدم

الأساس النظرى للتجربة:

- **Acting like a low-pass filter, momentum allows the network to ignore small features in the error surface.**
- **The magnitude of the effect that the last weight change is allowed to have is mediated by a momentum constant, mc, which can be any number between 0 and 1.**
- **When the momentum constant is 0, a weight change is based solely on the gradient.**
- **When the momentum constant is 1, the new weight change is set to equal the last weight change and the gradient is simply ignored.**
- **The gradient is computed by summing the gradients calculated at each training example, and the weights and biases are only updated after all training examples have been presented.**
- **The traingdm function is invoked using the same steps shown above for the traingd function, except that the mc, lr and max_perf_inc learning parameters can all be set.**

خطوات تنقيذ التجربة :

```
p = [      -1 -1 2 2;0 5 0 5];
t = [      -1 -1 1 1];
net=newff(minmax      (p),[3,1],{'tansig','purelin'},'traingdm');
net.trainParam.show = 50;
net.trainParam.lr = 0.05;
net.trainParam.mc = 0.9;
net.trainParam.epochs = 300;
net.trainParam.goal = 1e-5;
[net,tr]=train(net,p,t);
```

TRAINGDM, Epoch 0/300, MSE 3.6913/1e-05, Gradient 4.54729/1e-10 - **44** -
TRAINGDM, Epoch 50/300, MSE 0.00532188/1e-05, Gradient 0.213222/1e-10
TRAINGDM, Epoch 100/300, MSE
6.34868e-05/1e-05, Gradient 0.0409749/1e-10 TRAINGDM, Epoch 114/300, MSE 9.06235e-06/1e-05,
Gradient 0.00908756/1e-10 TRAINGDM, Performance goal met.

```
a = sim(net,p)
```

This method is often too slow for practical problems.

In this section we discuss several high performance algorithms that can converge from ten to one hundred times faster than the algorithms discussed previously.

All of the algorithms operate in the batch mode and are invoked using train.

Experiment 2: Variable Learning Rate (traingda, traingdx)

بيانات عامه:

اسم التجربه: Variable Learning Rate (traingda, traingdx)

الفرقه المقرر عليها التجربة : الرابعه

الفصل الدراسى : الثانى

الأدوات المطلوبة: جهازحاسب متقدم

الأساس النظرى للتجربة:

This algorithm uses heuristic techniques, which were developed from an analysis of the performance of the standard steepest descent algorithm.

- The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge. An adaptive learning rate will attempt to keep the learning step size as large as possible while keeping learning stable.

 An adaptive learning rate requires some changes in the training procedure used by traingd.
 First, the initial network output and error are calculated.
 At each epoc h new weights and biases are calculated using the current learning rate.
New outputs and errors are then calculated.
 When a larger learning rate could result in stable learning, the learning rate is increased.
 When the learning rate is too high to guar antee a decrease in error, it gets decreased until stable learning resumes.

خطوات تنقيذ التجربة :

Here is how it is called to train our previous two-layer network:

```
 p = [      -1 -1 2 2;0 5 0 5];
 t = [      -1 -1 1 1];
 net=newff(minmax(p),[3,1],{'tansig','purelin'},'traingda');
 net.trainParam.show = 50;
 net.trainParam.lr = 0.05;
 net.trainParam.lr_inc = 1.05;
 net.trainParam.epochs = 300;
 net.trainParam.goal = 1e
 [net,tr]=train(net,p,t);
```

النتائج :

TRAINGDA, Epoch 0/300, MSE 1.71149/1e-05, Gradient 2.6397/1e-06 TRAINGDA, Epoch 44/300, MSE 7.47952e-06/1e-05, Gradient 0.00251265/1e-06
TRAINGDA, P**erformance goal met.**

a = sim(net,p)
a = -1.0036 -0.9960 1.0008 0.9991

مناقشة النتائج :

The function **traingdx** combines adaptive learning rate with momentum training. It is invoked in the same way as traingda, except that it has the momentum coefficient mc as an additional training parameter.

Experiment 3: Creating Discrete-Time Models

بيانات عامه:

اسم التجربه: Creating Discrete-Time Models

الفرقه المقرر عليها التجربة : الرابعه

الفصل الدراسى : الثانى

الأدوات المطلوبة: جهازحاسب متقدم

الأساس النظرى للتجربة:

Discrete time systems (sampled) are dynamic systems in which one or more variables can change only at discrete instants of time.

. These instants (denoted by tk , k=0,1,2,..) may specify the time at which some physical measurement is performed or the time at which the memory of a digital computer is read out.

. The time interval between two discrete instants is taken to be short so that the data for the time between these discrete instants can be approximated by simple interpolation.

. Discrete time sys. Differs from continuous ones in that the signals for a discrete systems are in **sampled data form**.

. Discrete time sys.s arise in practice whenever the measurements necessary for control are obtained in an intermittent fashion.. Or when computer is time shared by several plants so that a control signal is sent out to each plant periodically or whenever a digital computer is used to perform computations necessary for control.

. Many modern industrial control systems are discrete systems since they include some elements whose inputs and/or outputs are discrete in time.

خطاخ الإنشاء: The syntax for creating discrete-time models is similar to that for continuous-time models, except that you must also provide a sampling time (sampling interval in seconds).

For example, to specify the discrete-time transfer function: with sampling period Ts = 0.1 s, type:

- **num = [1 -1];**
- **den = [1 -1.85 0.9];**
- **H = tf(num, den,0.1) or equival ently:**
- **z = tf('z',0.1);**

- **H = (z - 1) / (z^2 - 1.85*z + 0.9);**

Similarly, to specify the discrete-time state-space model: with sampling period Ts = 0.1 s, type:
sys = ss(.5,1,.2,0,0.1);

Recognizing Discrete-Time Systems

There are several ways to determine if your LTI model is discrete:

The display shows a nonzero sampling time value
sys.Ts or get(sys,'Ts') return a nonzero sampling time value. isdt(sys) returns true.
For example, for the transfer function H specified above, H.Ts ans = 0.1000 isdt(H)
ans = 1

النتائج :

The following plots show these characteristic traits: step(H)