تمرین‌های سری یک Machine Learning

فرهاد دلیرانی

۹۶۱۳۱۱۲۵

dalirani@aut.ac.ir

dalirani.1373@gmail.com

-۱

یادگیری با نظارت: هرگاه داده های آموزش شامل نمونه‌هایی باشد که به وضوح مشخص شده باشد که خروجی چه باید باشد آنگاه می‌توان گفت در محدوده‌ی یادگیری با نظارت هستیم. مانند تشخیص رقم‌ها, اسپم بودن/نبودن ایمیل. در این دسته از روش‌های یادگیری, مجموعه‌ی آموزش به صورت زیر است:

(X1, Y1) (X2, Y2) ... (Xn, Yn)

که Xi داده‌ی آموزش است و Yi برچسب آن است.

یادگیری نیمه نظارتی: یادگیری نیمه نظارتی نوعی یادگیری بین دسته‌ی یادگیری بانظارت و بدون نظارت است. در این نوع یادگیری برای آموزش هم از داده‌های با برچسب و هم بدون برچسب استفاده می‌شود.

یادگیری بدون نظارت: در یادگیری بدون نظارت مجموعه‌ی یادگیری هیچگونه برچسبی ندارد. یادگیری بدون نظارت را می‌توان به عنوان کاری دید که به دنبال ساختار و روابط میان داده‌های ورودی است. مانند دسته‌بندی تعدادی کتاب در موضوعات مختلف به طوری که کتاب‌های هر دسته از نظر موضوعی شباهت داشته باشند. همین طور از یادگیری بدون نظارت می توان به عنوان یک روش برای نمایش و ارایه‌ی داده‌ها به صورت قابل درک و معنا دار استفاده کرد.

یادگیری تقویتی: در یادگیری تقویتی برخلاف یادگیری بانظارت که هر نمونه در مجموعه‌ی آموزش برچسبی دارد, تابع target مشخصی که یک برچسب را به یک نمونه نسبت بدهد را نداریم. بلکه در این نوع یادگیری برای هر اکشن یک مقیاس داریم که مشخص می‌کند که اکشن چه‌میزان خوب یا بد بوده است. در یادگیری با نظارت مجموعه‌ی یادگیری به صورت (input, correct output) است در حالی که در یادگیری با تقویتی مجموعه‌ی یادگیری به صورت زیر است:

(Input, Some Output, Grade for this output)

این نوع یادگیری کاربردهای زیادی در یادگیری انجام بازی دارد.

یادگیری برخط: در این نوع یادگیری مجموعه‌ای که برای یادگیری استفاده می‌شود به صورت یکجا در اختیار نیست بلکه هر زمان یک نمونه در اختیار آن قرار می گیرد. به عبارت دیگر دیتاهای آموزش یک دنباله هستند هر عضو دنباله در زمان متفاوتی در اختیار الگوریتم قرار می‌گیرد.

-۲

در صورتی که مقدار تابع هزینه (خطا) بر روی ۸۰ درصد داده آموزش کم باشد ولی مقدار تابع هزینه برای ۲۰ درصد باقی‌مانده زیاد باشد متوجه می‌شویم مدل مورد استفاده دچار بیش برازش شده است زیرا خطای آن بر روی داده‌هایی که با آن‌ها آموزش دیده است کم است ولی توانایی پیشبینی درست برای داده‌هایی را که ندیده است را ندارد.

-۳

خطای MSE : خطای Mean Square Error خطایی است که میزان میانگین مربع خطاها را محاسبه می-کند. منظور از خطا اختلاف بین تخمین مدل و مقدار واقعی است. که از رابطه‌ی زیر محاسبه می‌شود:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

خطای RMSE : خطای Root-Mean Square Error یک مقیاس پرکاربرد برای سنجش خطای تخمین-های مدل و مقدار واقعی است. که میزان جذر میانگین مربع خطاها است که از رابطه‌ی زیر قابل محاسبه است:

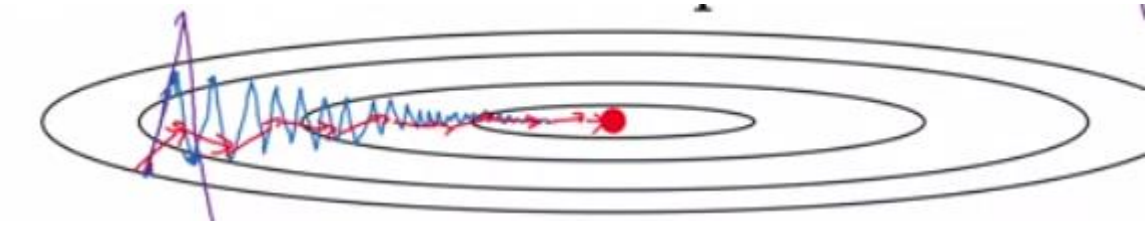
$$\mathrm{RMSD}(\theta_1, \theta_2) = \sqrt{\mathrm{MSE}(\theta_1, \theta_2)} = \sqrt{\mathrm{E}((\theta_1 - \theta_2)^2)} = \sqrt{\frac{\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2}{n}}.$$

برای دیتاست با داده‌های پرت و ناهنجار استفاده از RMSE بهتر است زیر در مقابل داده‌های پرت با توجه به جذری که در آن در مقایسه با MSE هست Robust تر است و مقدارش به صورت در مقابله با آن داده‌ها کم تر تغییر می‌کند.

-۴

ایده‌ی اصلی Gradient Descent With Momentum استفاده از Exponentially Weighted Average گرادیان ها است. این روش همیشه سریع تر از روش گرادیان نزولی عادی عمل می کند.

در تصویر کانتورهای یک تابع هزینه را مشاهده می‌کنید که در راستای افق کشیده تر از راستای عمودی است, بردارهای آبی گرادیان نزولی عادی را نشان می دهد. همان‌طور که مشاهده می کنید تعداد گام‌هایی که گرادیان طی می کند بسیار زیاد است و همین طور بردارهای آبی زاویه‌ی زیادی با نقطه‌ی آپتیمال دارند و همین باعث می‌شود که نتوانیم ضریب یادگیری بزرگی انتخاب کنیم. بردارهای قرمز به عنوان مثال گرادیان نزولی به همراه تکانه را نشان می دهد, این روش همان طور که در تصویر زیر مشاهده می‌کنید باعث شده‌است که مولفه‌ی عمودی هرگام کوچک باشد و مولفه ی افقی آن بلند و این در مثال زیر باعث می‌شود بسیار سریع به نقطه‌ی آپتیمال برسیم زیر هر گام نسبت به گرادیان نزولی بیشتر در جهت نقطه‌ی آپتیمال است و گرادیان نزولی همراه تکانه مسیر مستقیم تری را طی می‌کند.



-۵

تتای صفر ضریب X0 است. X0ویژگی است که ما به sample ها اضافه می‌کنیم و مقدارش همیشه برابر یک است. از آنجایی که مقدار X0 همواره برابر یک است ضریب معادل آن را وارد Regularization نمی‌کنیم زیرا تاثیر چندانی بر نتیجه‌ی نهایی ندارد. تتای صفر نقش بایاس (عرض از مبدا) را دارد.

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m} \theta_j \quad \text{for } j \geq 1$$

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$

$$\min_\theta J(\theta)$$

۶-

افزایش داده‌های آموزش, **احتمال** رخ دادن over fitting را کم می‌کند.

زیرا اضافه کردن تعداد بیشتری دیتا diversity را بیشتر می‌کند و هر چه diversity بیشتر باشد مدل که پیدا می‌شود جنرال‌تر است و در مدل جنرال‌تر احتمال over fitting کم‌تر است.

- افزایش داده‌های مجموعه‌ی آموزش: هرچه تعداد داده‌های مجموعه‌ی آموزش بیشتر باشد diversity داده‌ها بیشتر می‌شود و diversity بالا منجر به می‌شود مدلی که براثر آموزش ایجاد می‌شود جنرال‌تر باشد و در مدل جنرال احتمال رخ داد بیش برازش کم می‌شود.
- کاهش تعداد features : هر چه تعداد features بیشتر باشد امکان پیدا کردن مدل‌های پیچیده‌تری فراهم می‌شود و بنابراین ممکن است تعداد فیچیرهای زیاد منجر به این شده باشند که مدلی که پیدا کرده ایم برای داده ها پیچیده باشد به طوریکه خطای مجموعه‌ی آموزش بسیار کم شده باشد در حالی که خطا برای داده‌هایی که تاکنون مشاهده نکرده‌ایم زیاد باشد. با کم کردن فیچرها می توان از پیچیدگی مدل کاست و از بیش برازش دوری کرد.
- رگولاریزیشن: ریگولاریزیشن تکنیکی در یادگیری ماشین است که با استفاده از آن می‌توان از بیش-برازش دوری کرد. ایده‌ی اصلی پشت آن افزودن پنالتی به تابع هزینه است که باعث می شود مقدار تابع هزینه در صورت پیچیده‌شدن و over fitting زیاد شود و از آنجایی که به دنبال مینیموم کردن هزینه‌ها هستیم باعث ایجاد تعادل بین پیچیدگی زیاد و رسیدن به مینیموم تابع هزینه‌ها می شود یکی از راه‌های رگولاریزیشن در MSE را در تصویر زیر مشاهده می‌کنید:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$

- تغییر الگوریتم یادگیری: الگوریتم‌های مختلف یادگیری ویژگی‌های مختلفی دارند با تغییر الگوریتم‌های یادگیری و انتخاب مدلی مناسب می‌توان از بیش برازش جلوگیری کرد.

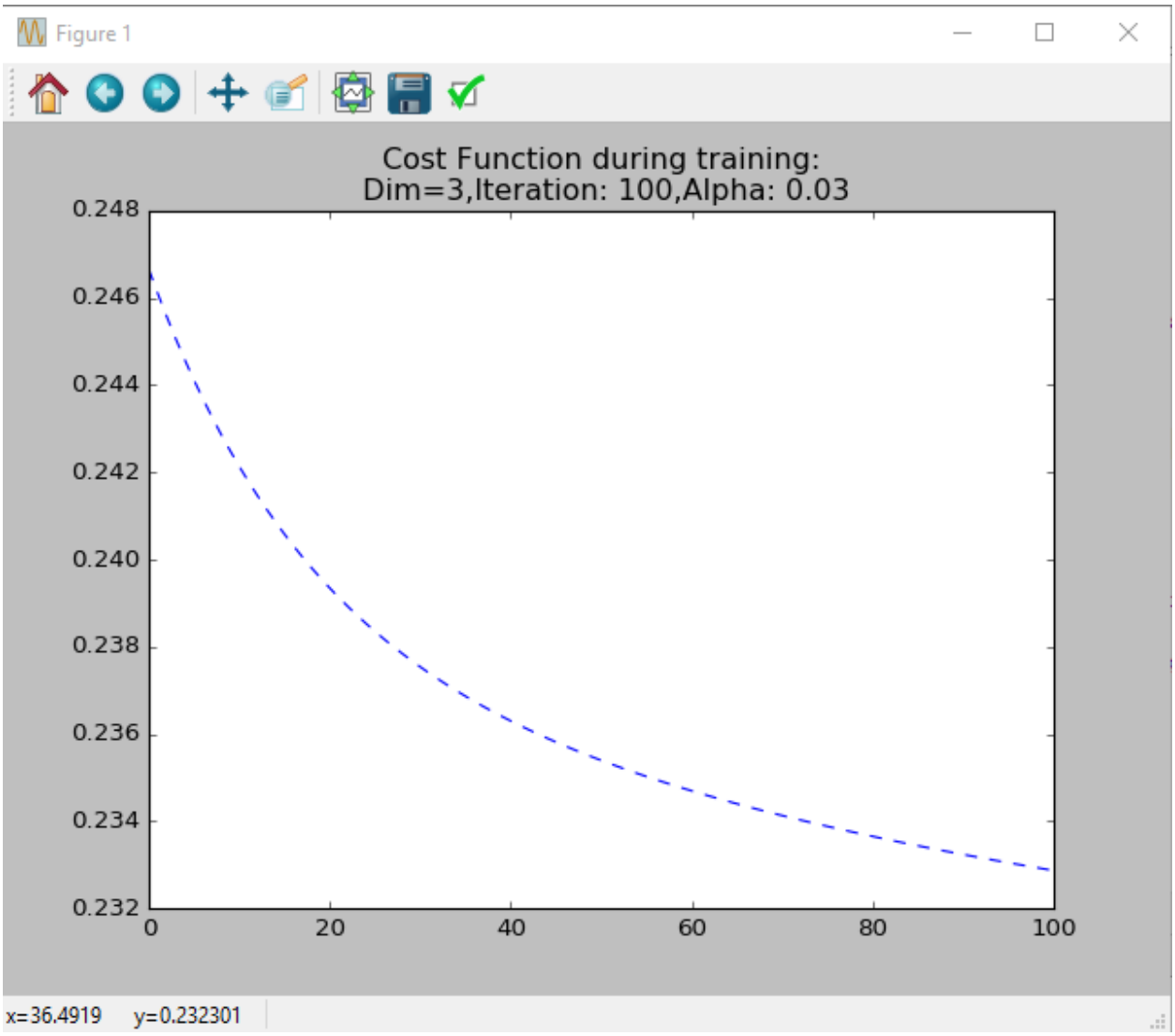
۸ –

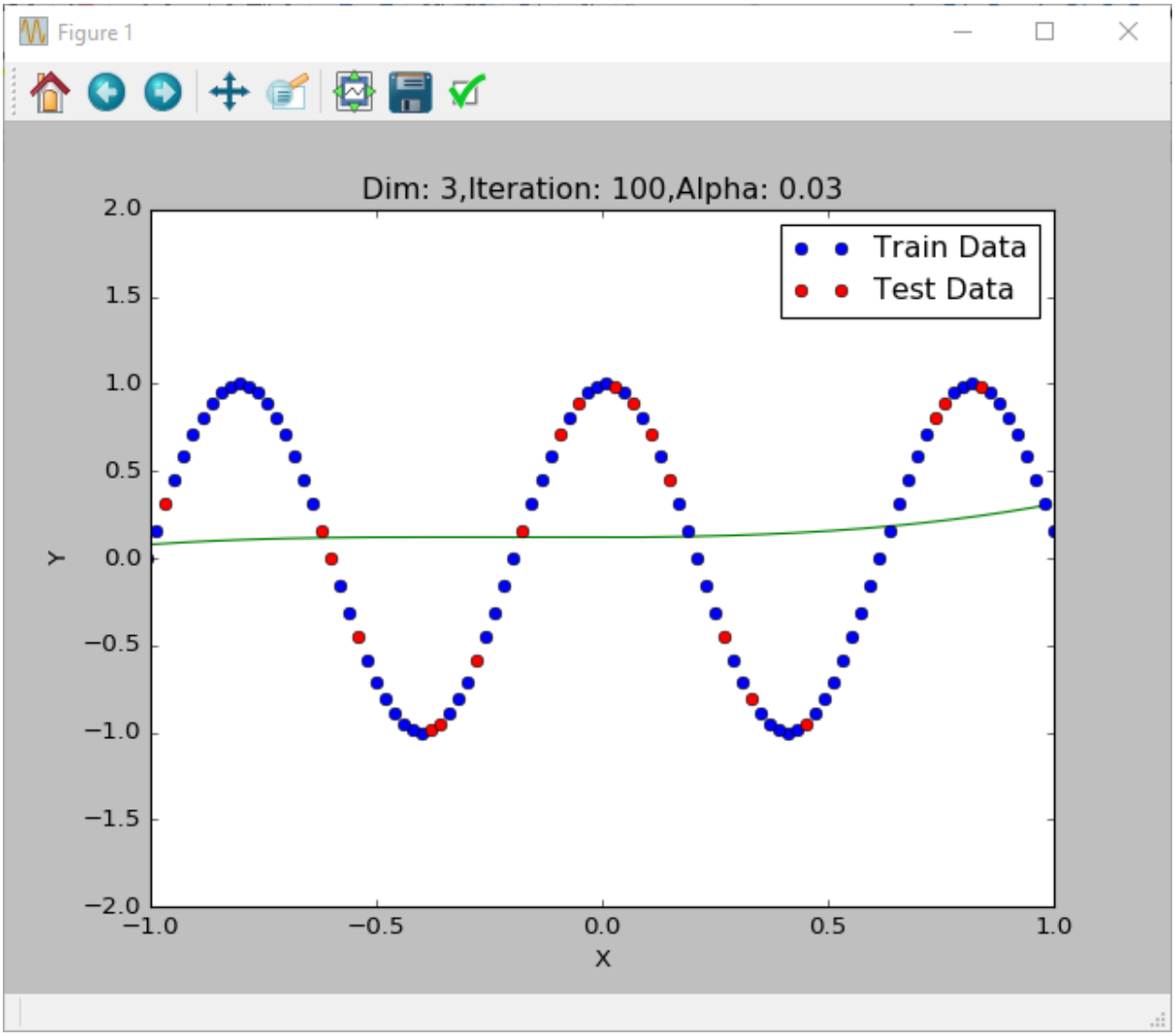
الف)

کد این بخش از سوال هشت در فایل gradientDescent.py موجود است که به خوبی کامنت گذاری شده است.

نتایج حاصل را در تصویرهای زیر می‌بینید:

درجه‌ی ۳ – گرادیان نرولی در ۱۰۰ گام – میزان تابع خطا در طول ۱۰۰ گام





درجه‌ی ۳ – گرادیان نرولی در ۱۰۰۰ گام – میزان تابع خطا در طول ۱۰۰۰ گام آموزش:

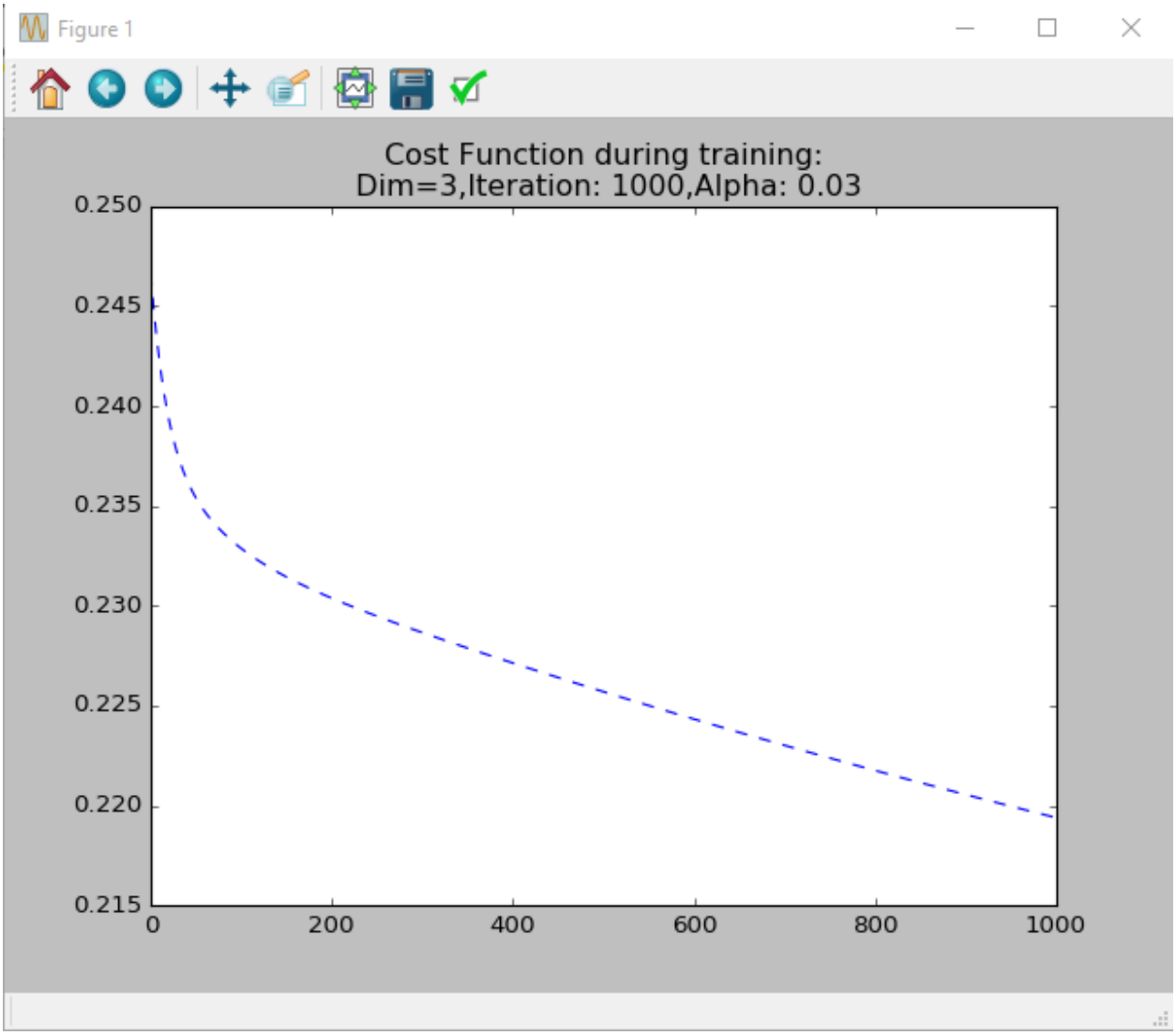
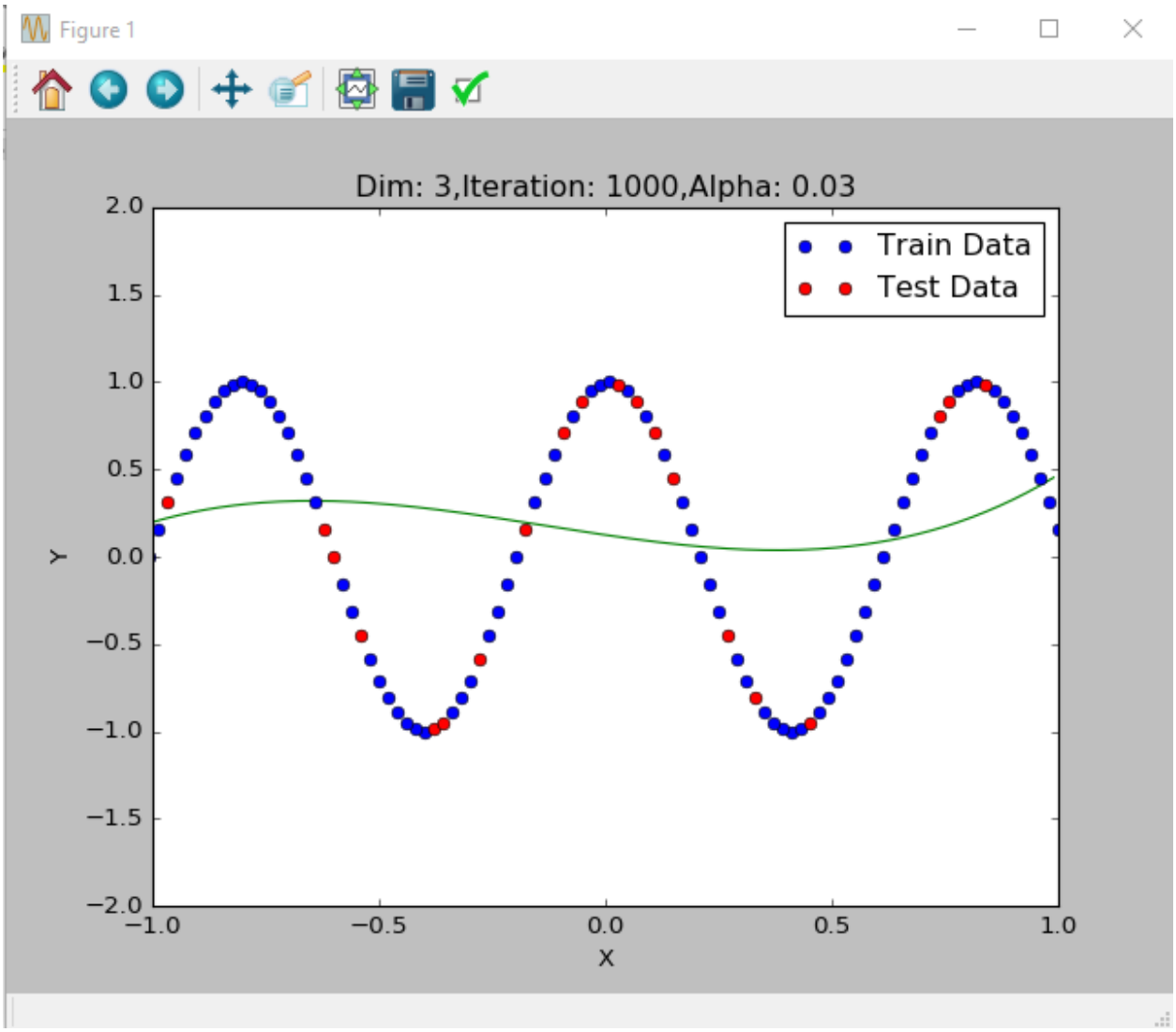


Figure 1
Cost Function during training:
Dim=3,Iteration: 1000,Alpha: 0.03
0.250
0.245
0.240
0.235
0.230
0.225
0.220
0.215
0
200
400
600
800
1000



درجه‌ی ۳ – گرادیان نرولی در ۱۰۰۰۰ گام – میزان تابع خطا در طول ۱۰۰۰۰ گام آموزش:

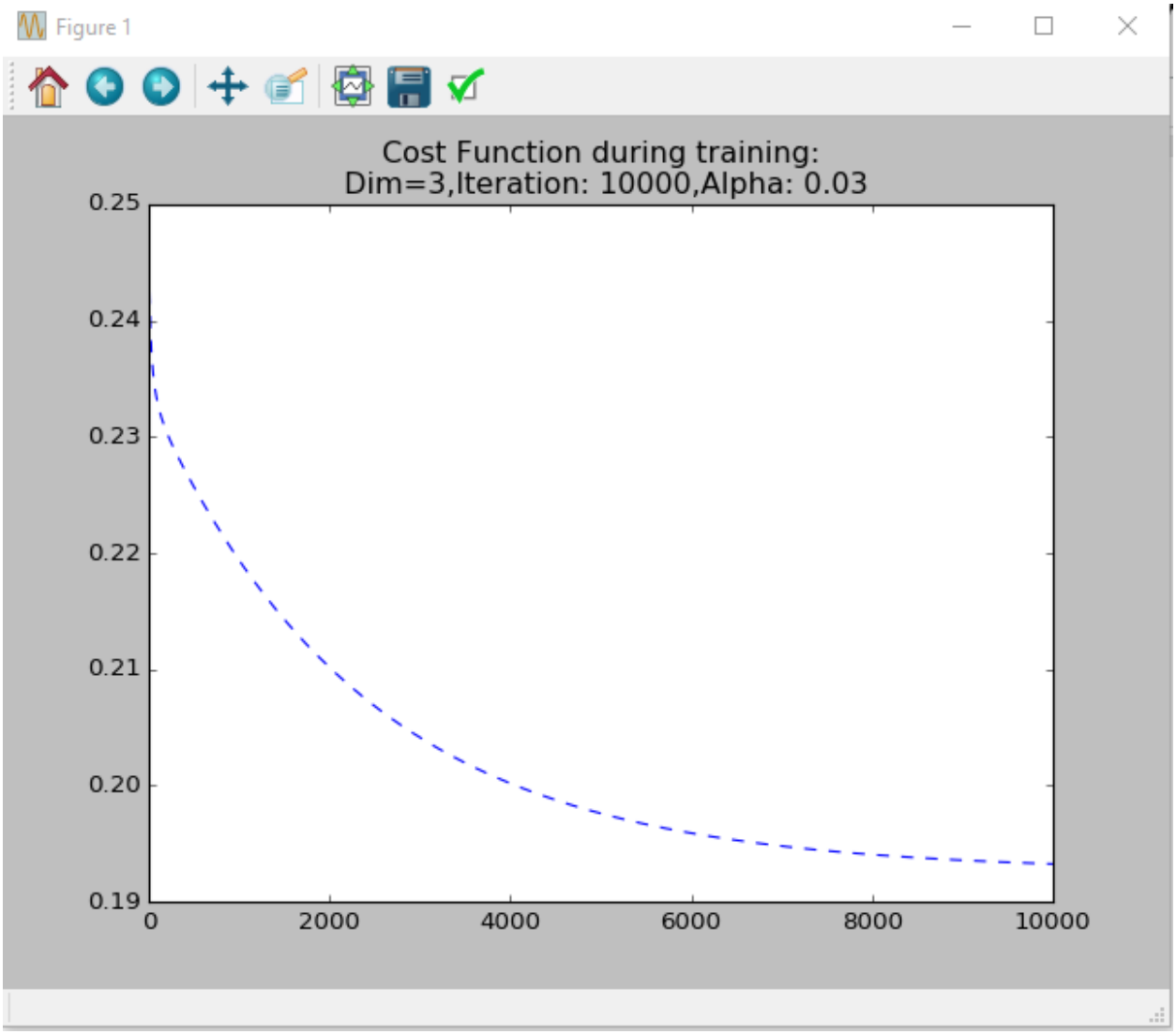
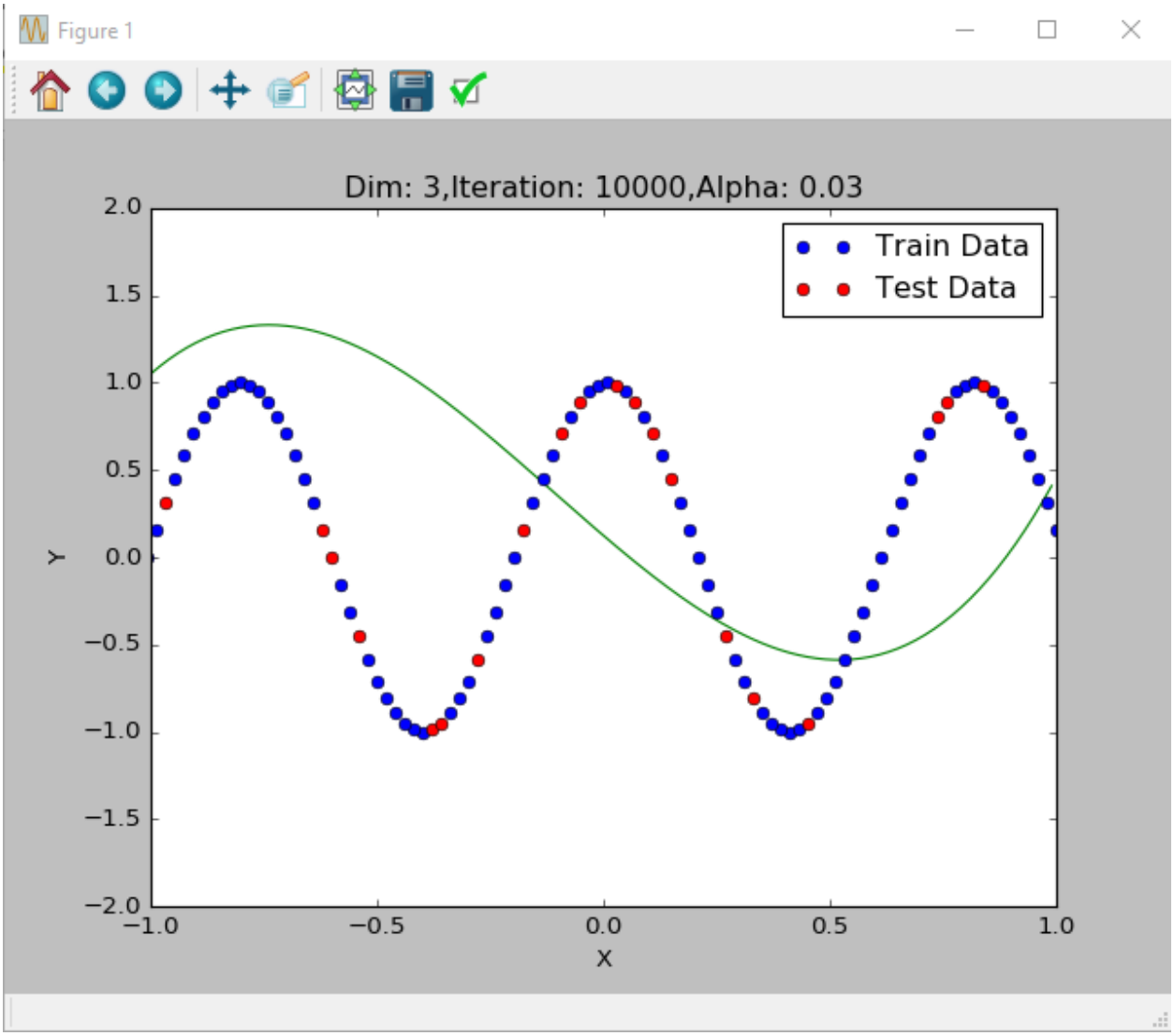


Figure 1
Cost Function during training:
Dim=3,Iteration: 10000,Alpha: 0.03
0.25
0.24
0.23
0.22
0.21
0.20
0.19
0
2000
4000
6000
8000
10000



درجه‌ی ۵ – گرادیان نرولی در ۱۰۰ گام – میزان تابع خطا در طول ۱۰۰ گام آموزش:

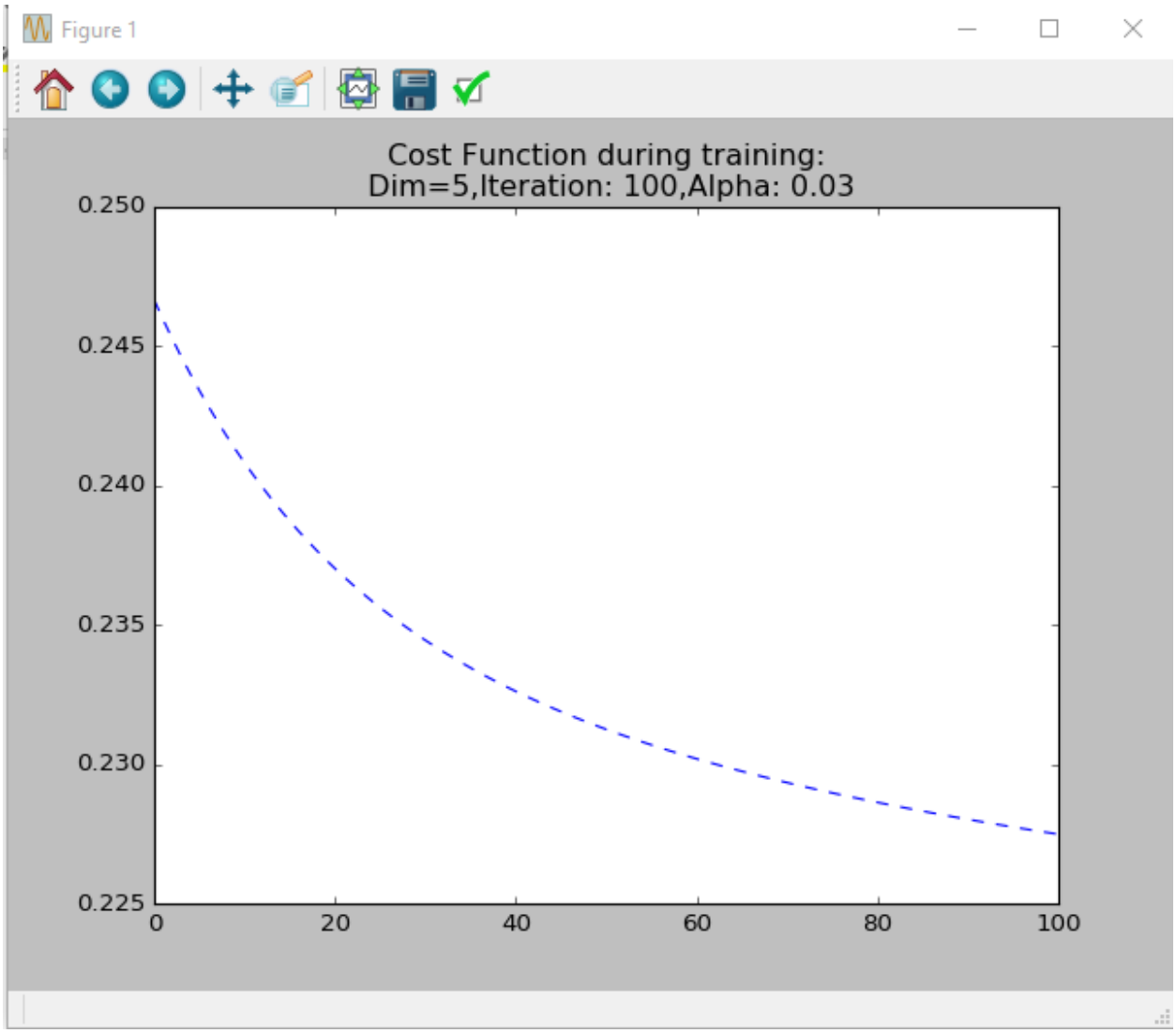
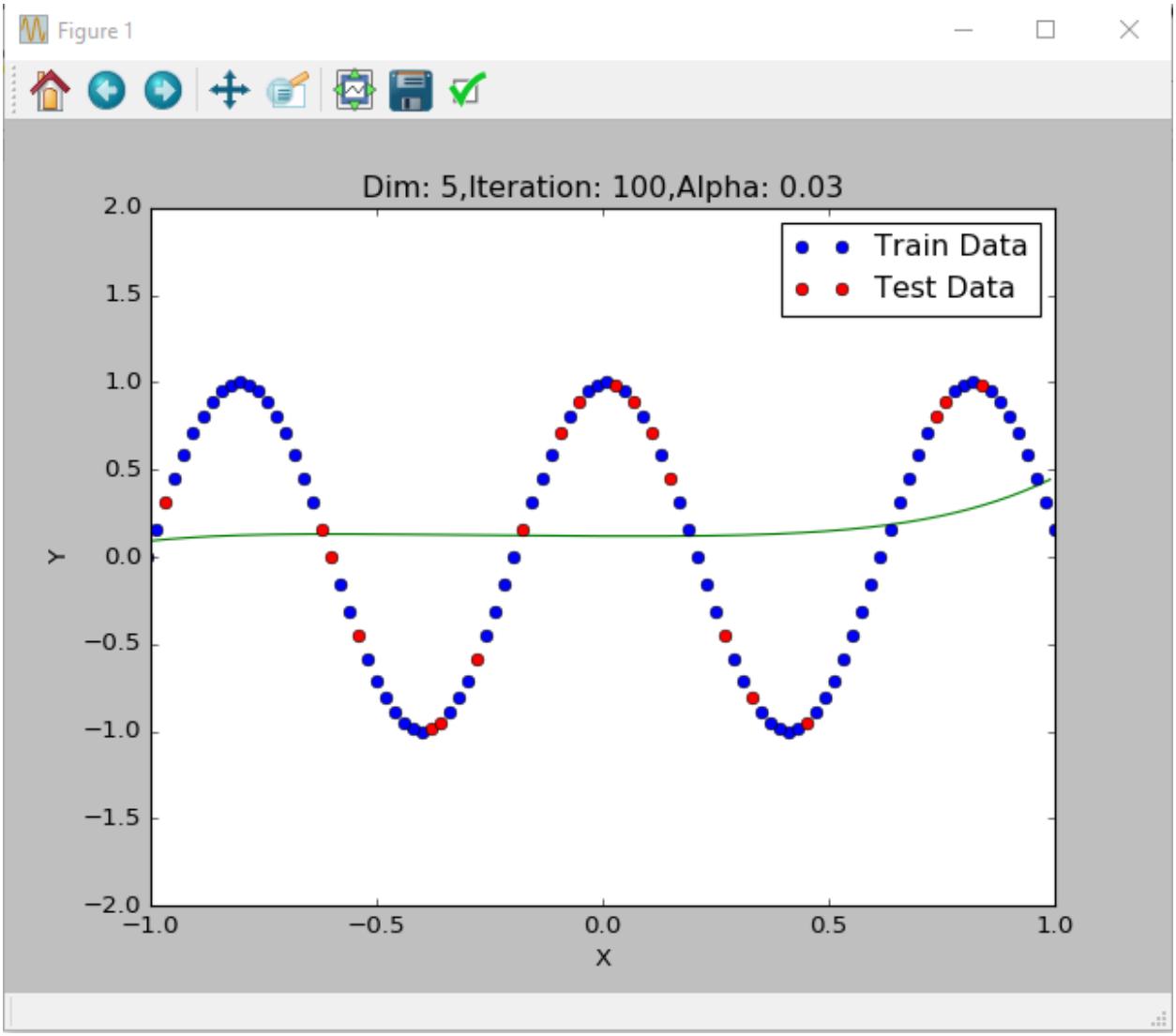


Figure 1
Cost Function during training:
Dim=5,Iteration: 100,Alpha: 0.03
0.250
0.245
0.240
0.235
0.230
0.225
0
20
40
60
80
100



درجه‌ی ۵ – گرادیان نرولی در ۱۰۰۰ گام – میزان تابع خطا در طول ۱۰۰۰ گام آموزش:

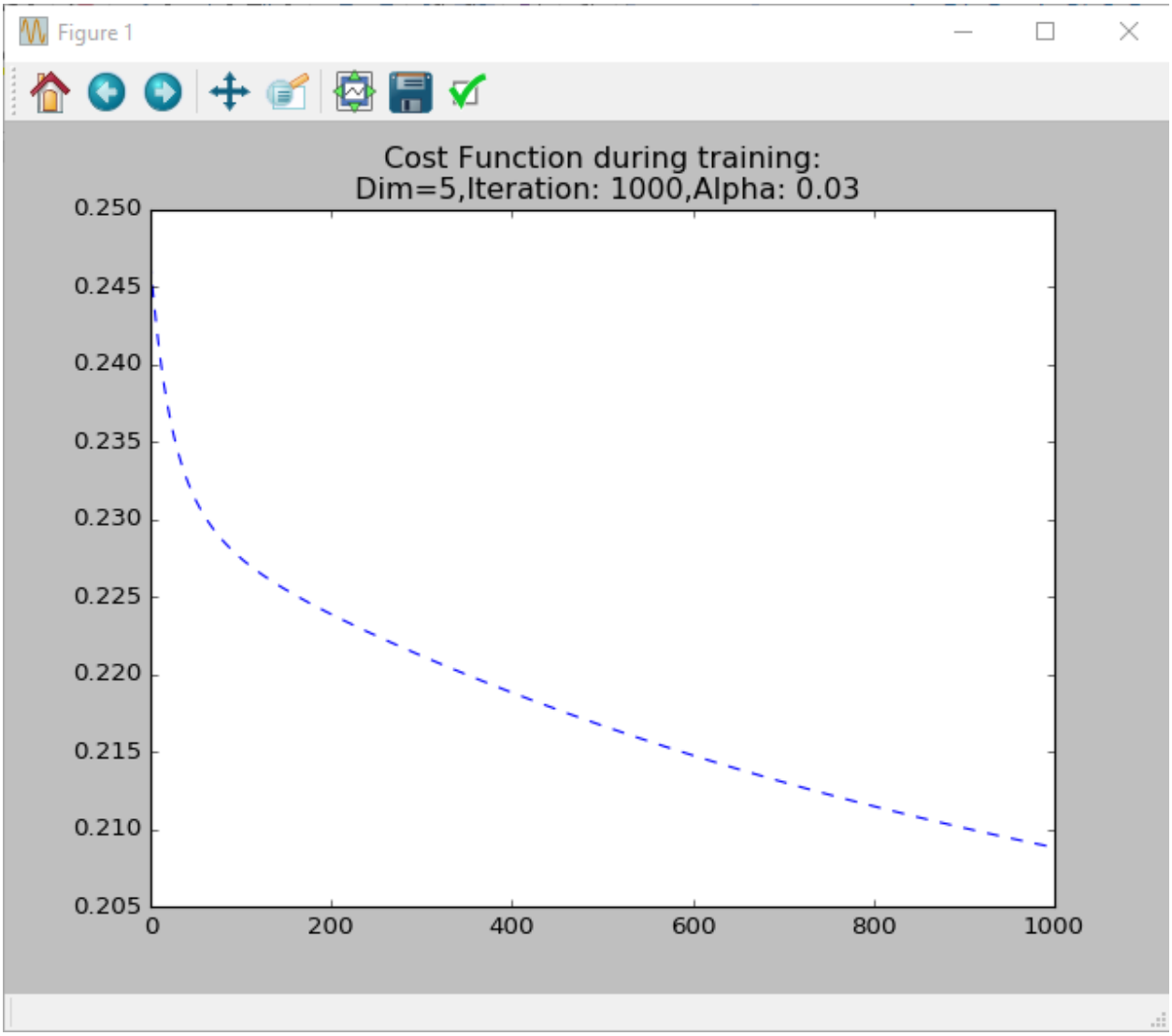
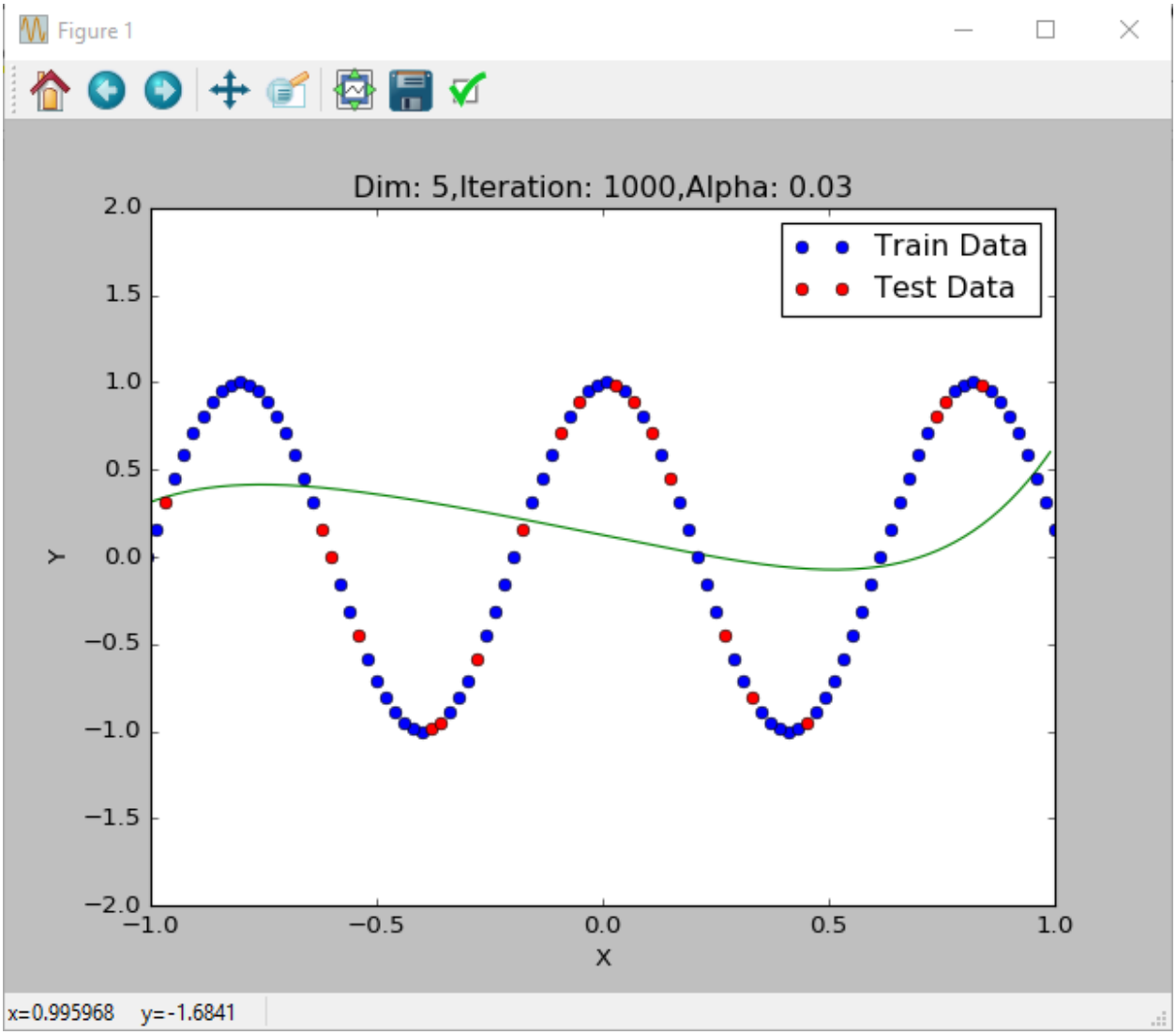


Figure 1
Cost Function during training:
Dim=5,Iteration: 1000,Alpha: 0.03
0.250
0.245
0.240
0.235
0.230
0.225
0.220
0.215
0.210
0.205
0
200
400
600
800
1000



درجه‌ی ۵ – گرادیان نرولی در ۱۰۰۰۰ گام – میزان تابع خطا در طول ۱۰۰۰۰ گام آموزش:

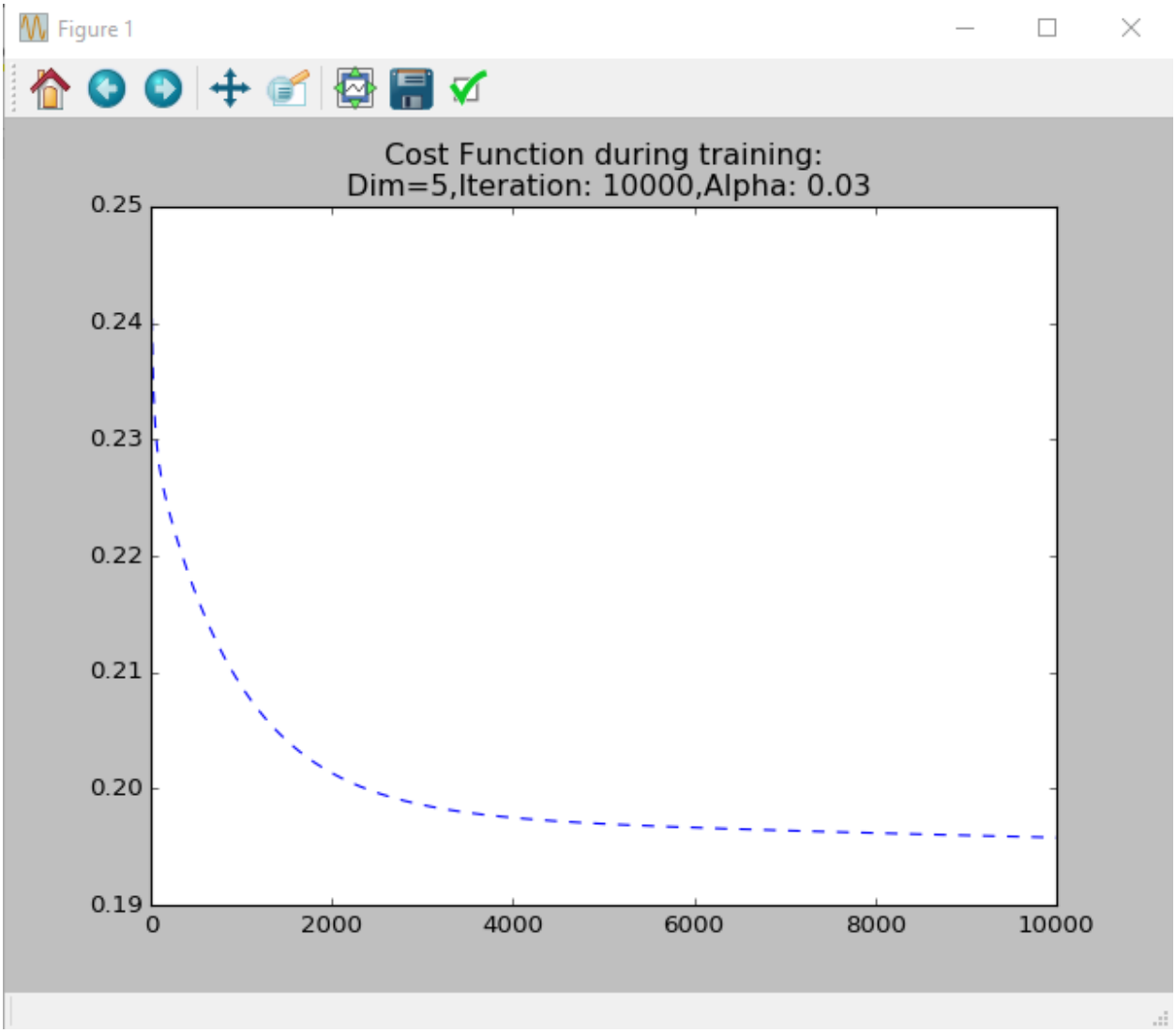
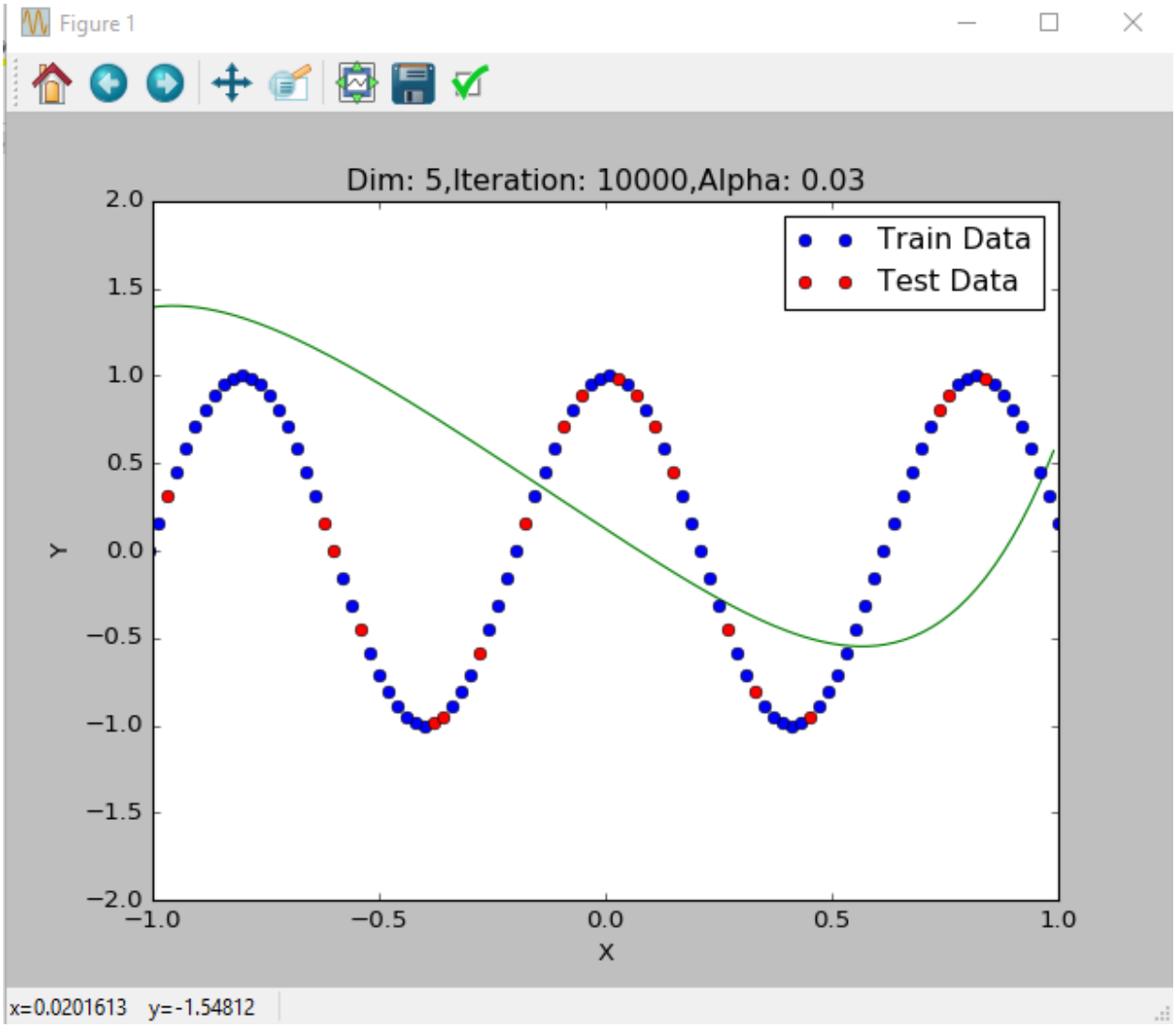


Figure 1
Cost Function during training:
Dim=5,Iteration: 10000,Alpha: 0.03
0.25
0.24
0.23
0.22
0.21
0.20
0.19
0
2000
4000
6000
8000
10000



درجه‌ی ۷ – گرادیان نرولی در ۱۰۰ گام – میزان تابع خطا در طول ۱۰۰ گام آموزش:

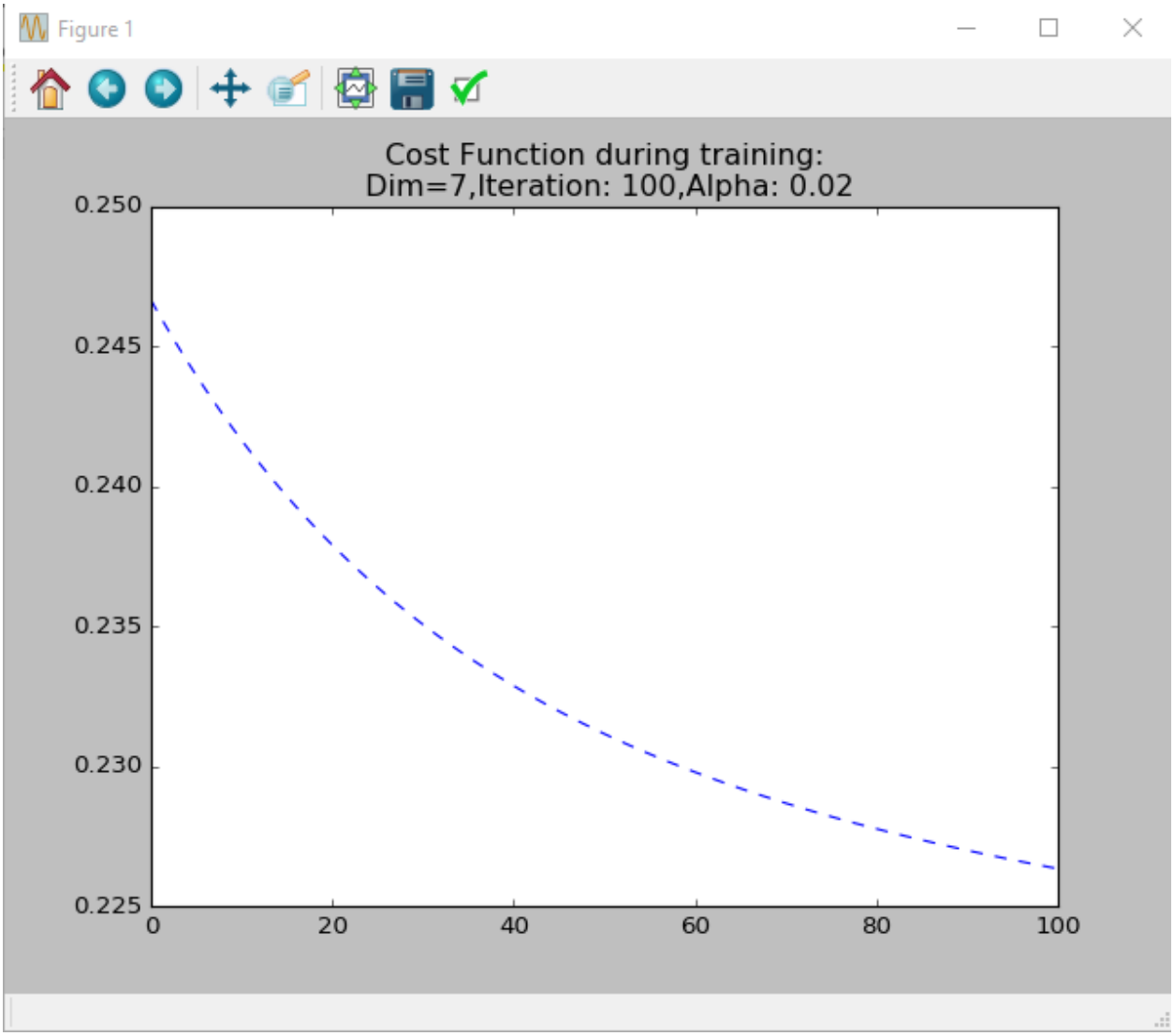
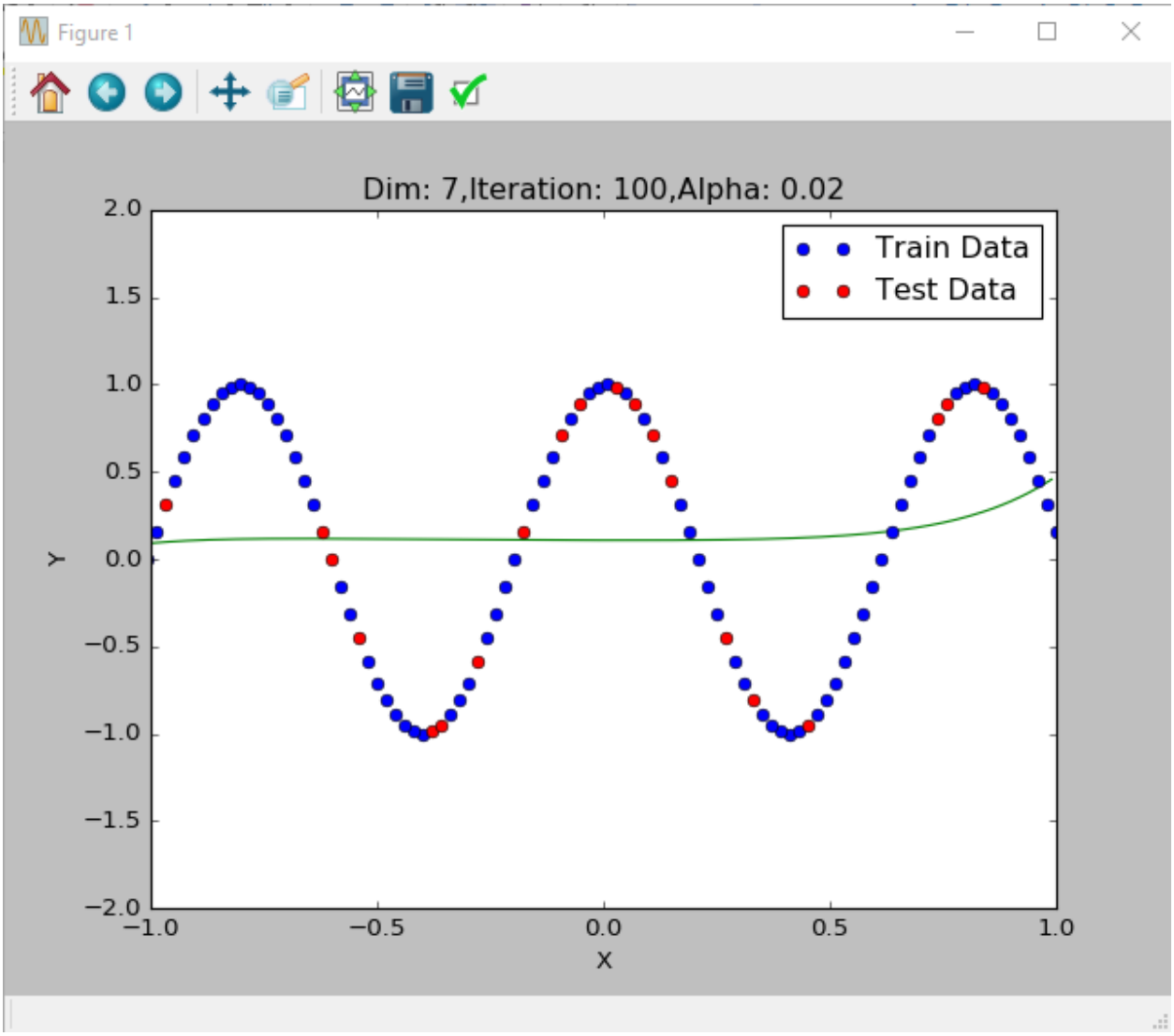


Figure 1
Cost Function during training:
Dim=7,Iteration: 100,Alpha: 0.02
0.250
0.245
0.240
0.235
0.230
0.225
0
20
40
60
80
100



درجه‌ی ۷ – گرادیان نرولی در ۱۰۰۰ گام – میزان تابع خطا در طول ۱۰۰۰ گام آموزش:

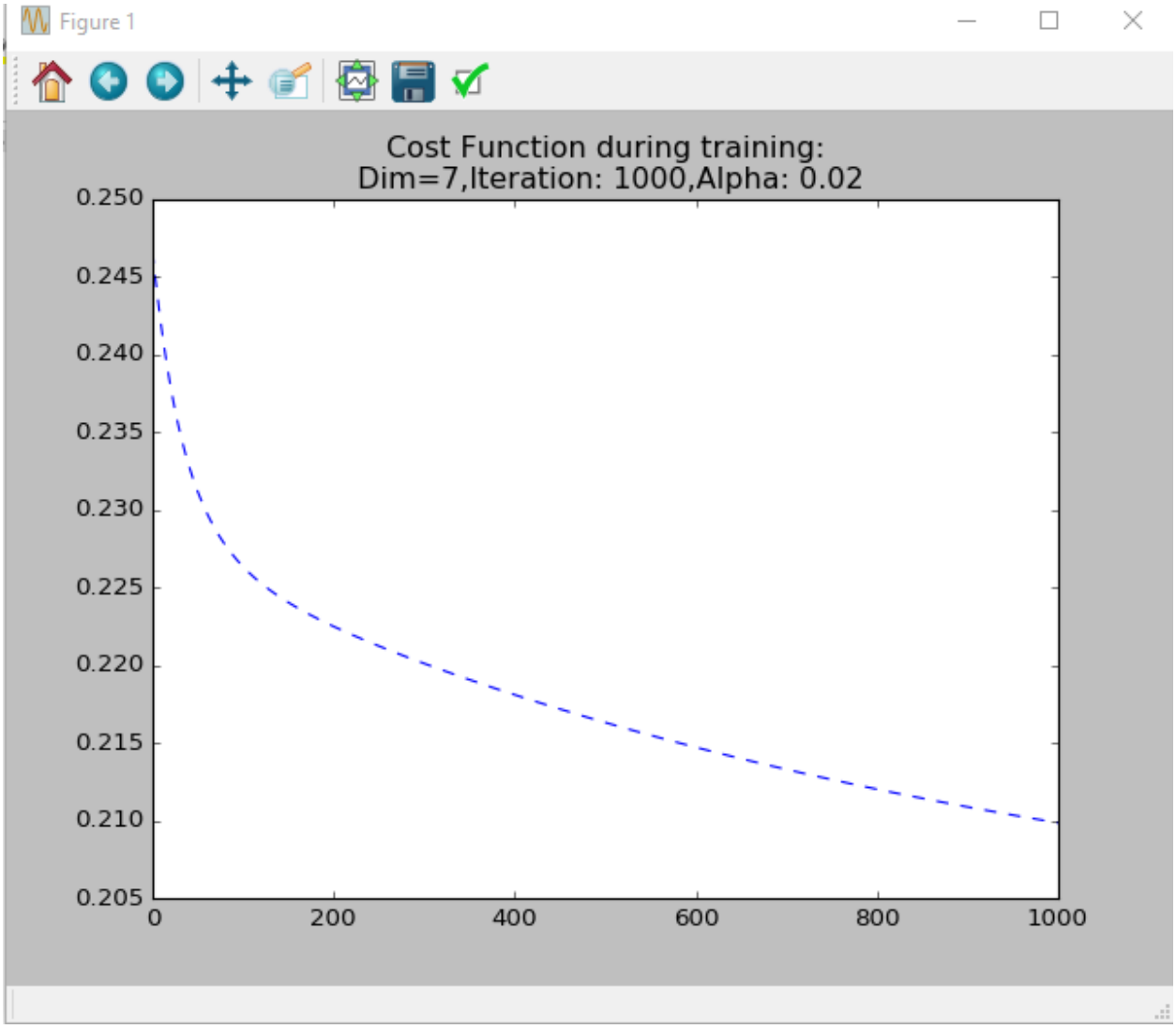
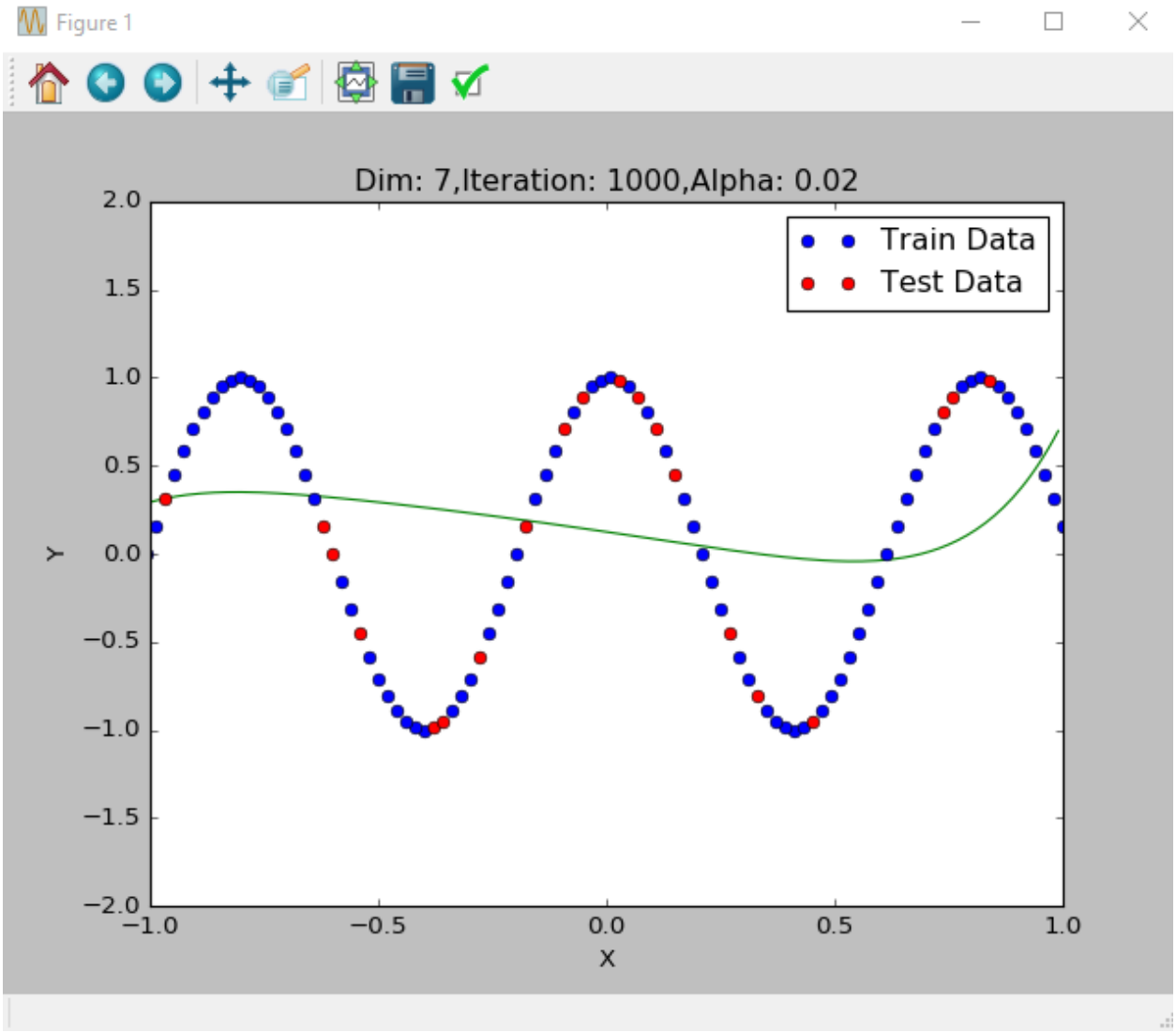


Figure 1
Cost Function during training:
Dim=7,Iteration: 1000,Alpha: 0.02
0.250
0.245
0.240
0.235
0.230
0.225
0.220
0.215
0.210
0.205
0
200
400
600
800
1000



درجه‌ی ۷ – گرادیان نرولی در ۱۰۰۰۰ گام – میزان تابع خطا در طول ۱۰۰۰۰ گام آموزش:

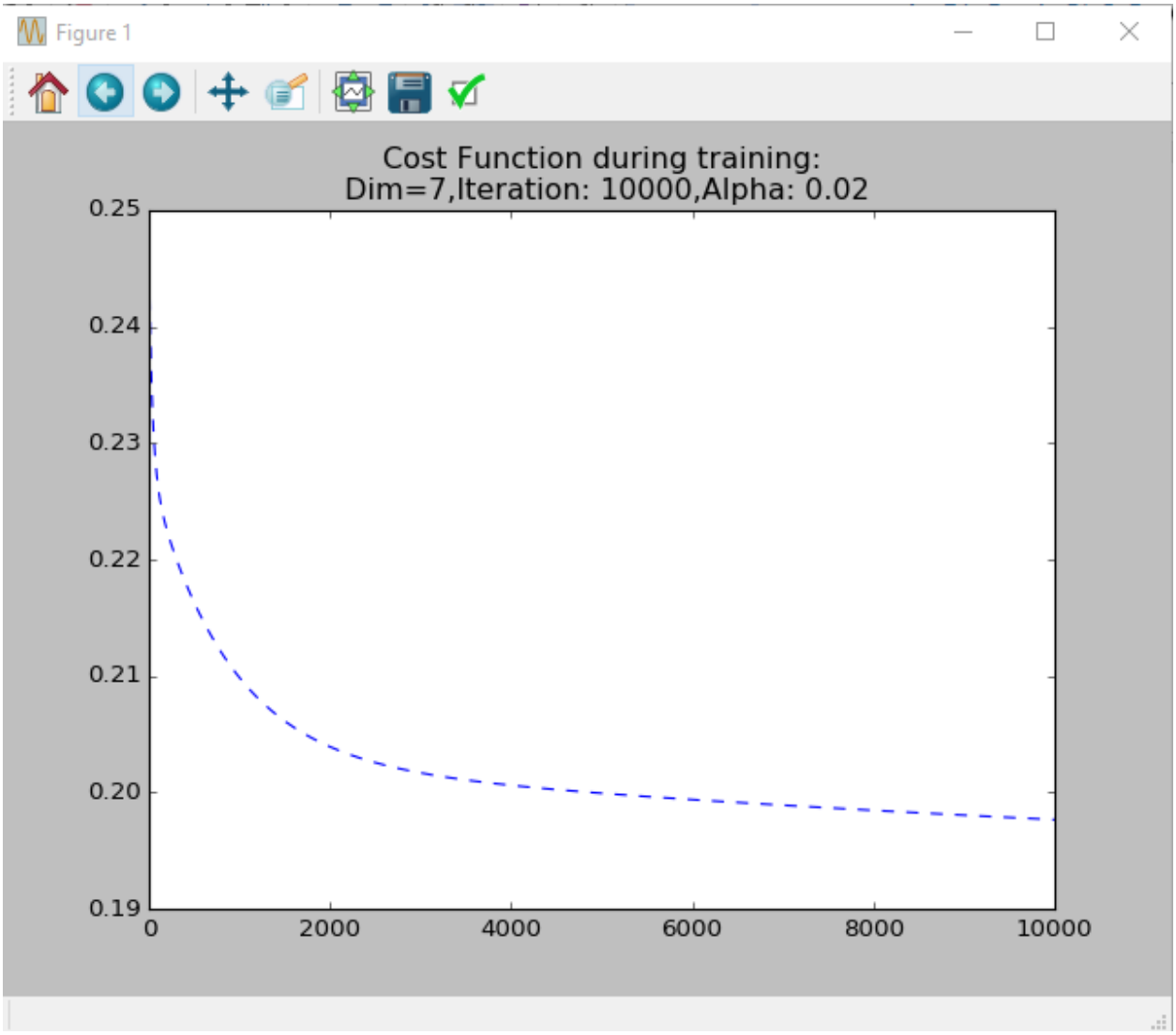
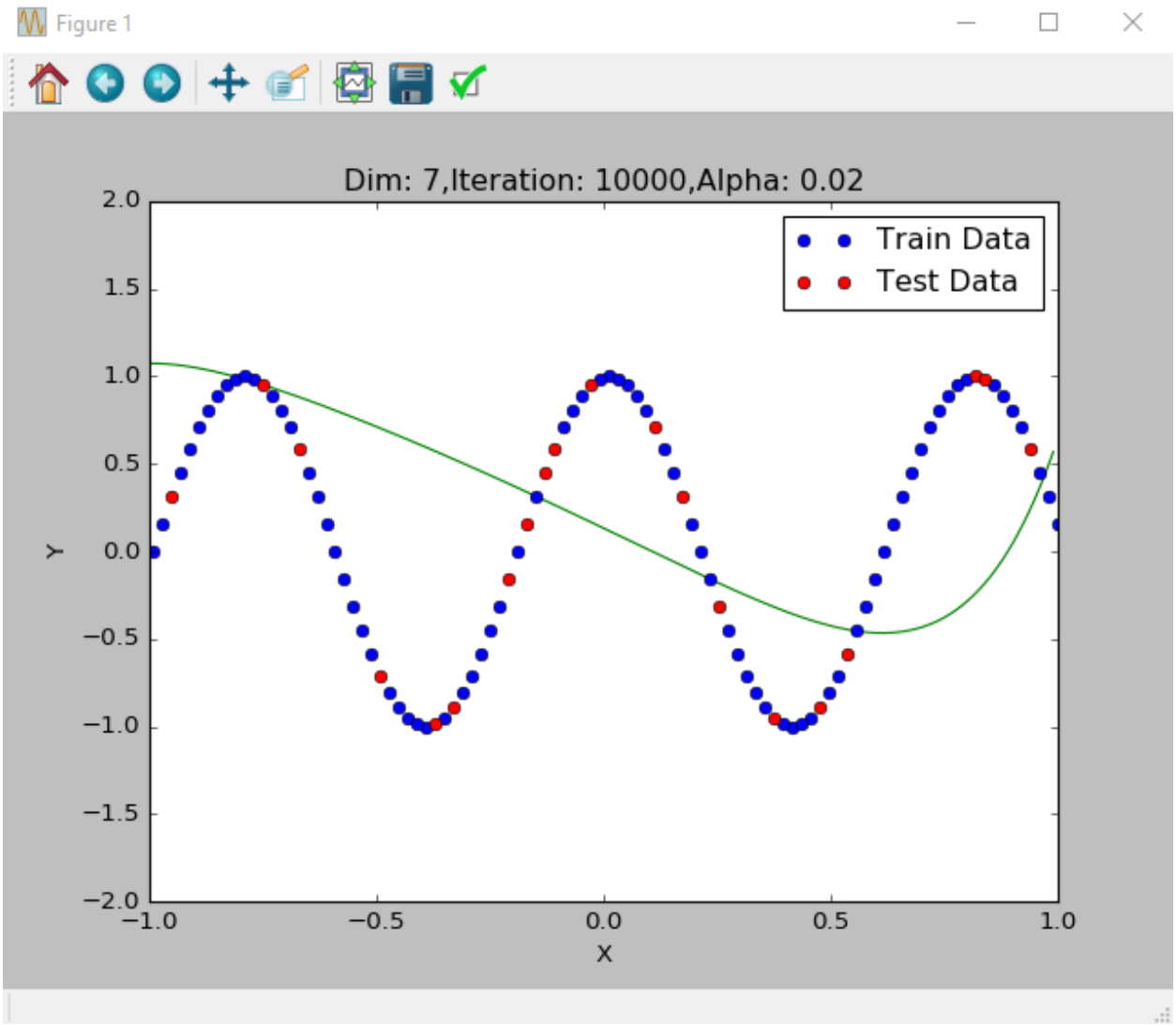
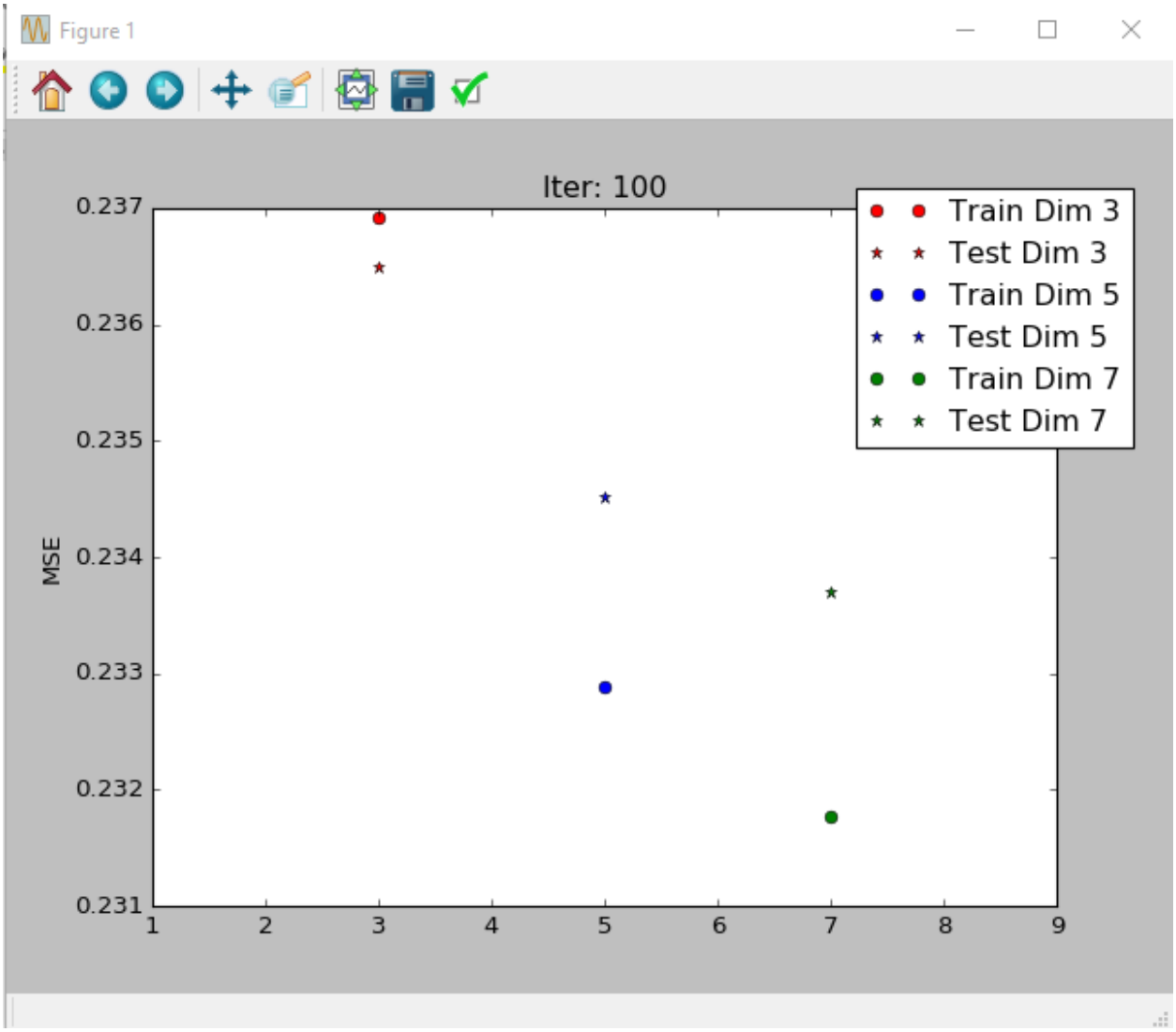


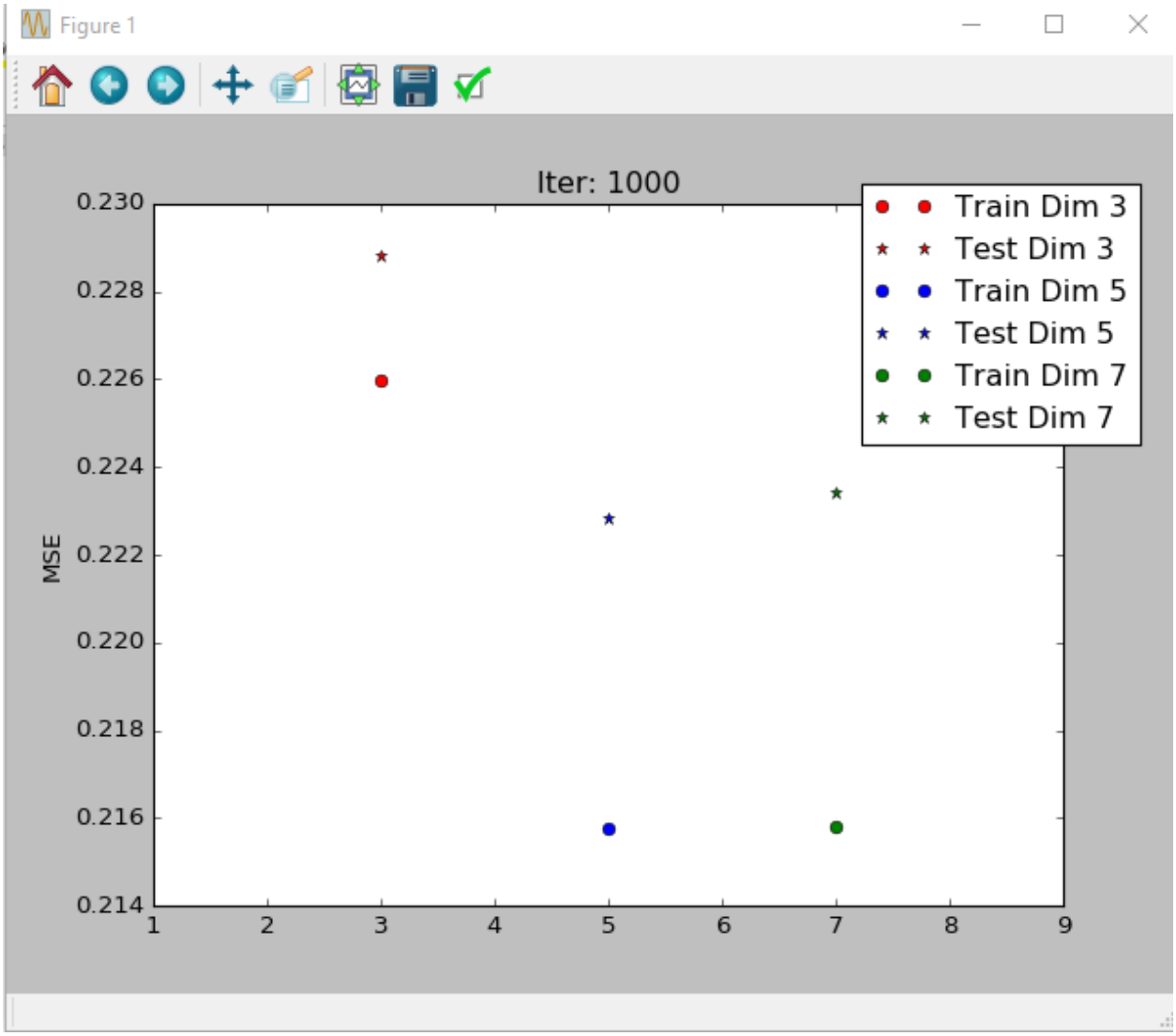
Figure 1
Cost Function during training:
Dim=7,Iteration: 10000,Alpha: 0.02
0.25
0.24
0.23
0.22
0.21
0.20
0.19
0
2000
4000
6000
8000
10000



میزان خطای مجموعه‌ی آموزش و تست برای درجه‌های مختلف ۳ ۵ و ۷ با استفاده از گرادیان نزولی با ۱۰۰ گام:



میزان خطای مجموعه‌ی آموزش و تست برای درجه‌های مختلف ۳ ۵ و ۷ با استفاده از گرادیان نزولی با ۱۰۰۰ گام:



میزان خطای مجموعه‌ی آموزش و تست برای درجه‌های مختلف ۳ ۵ و ۷ با استفاده از گرادیان نزولی با ۱۰۰۰۰ گام:

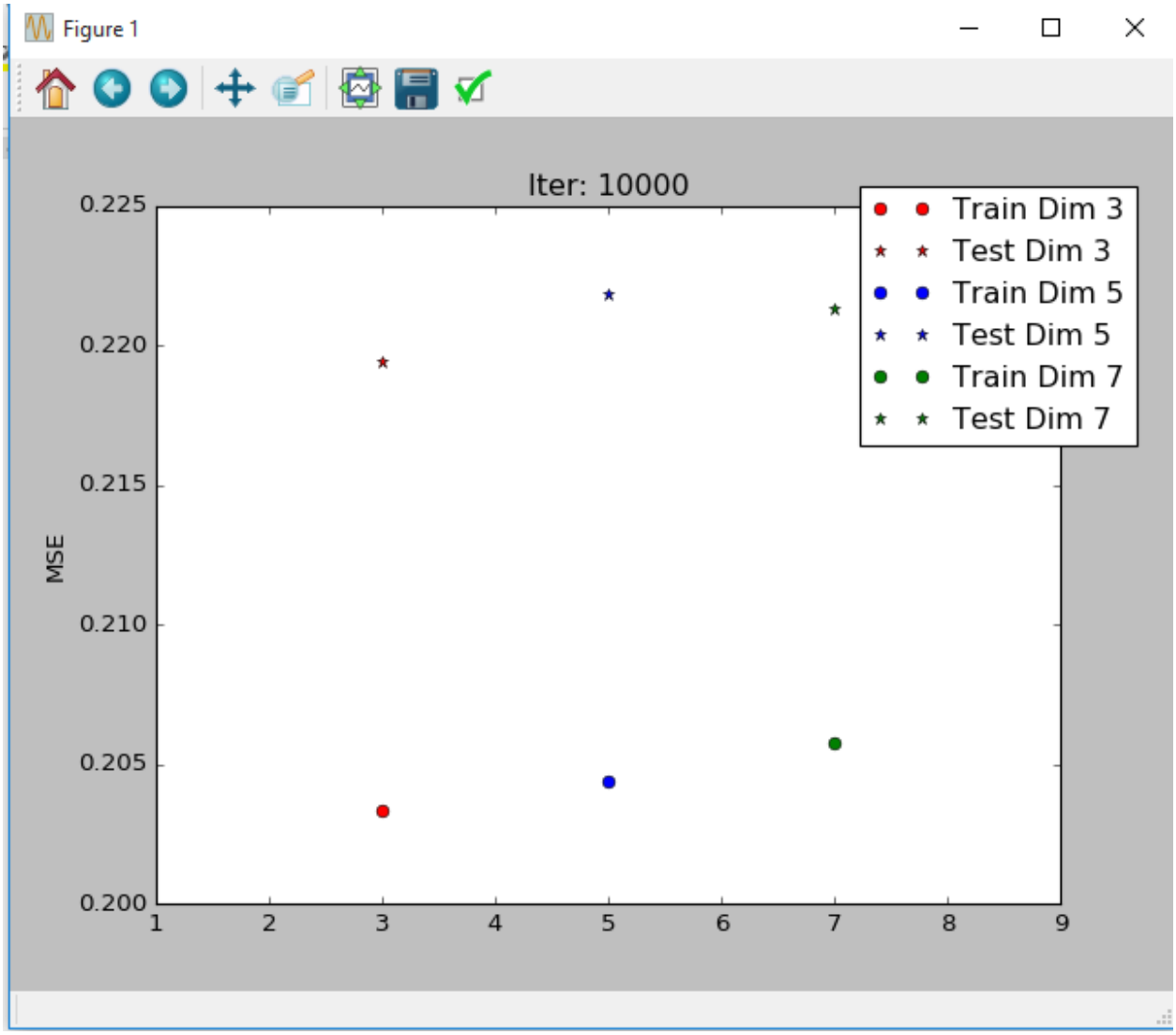
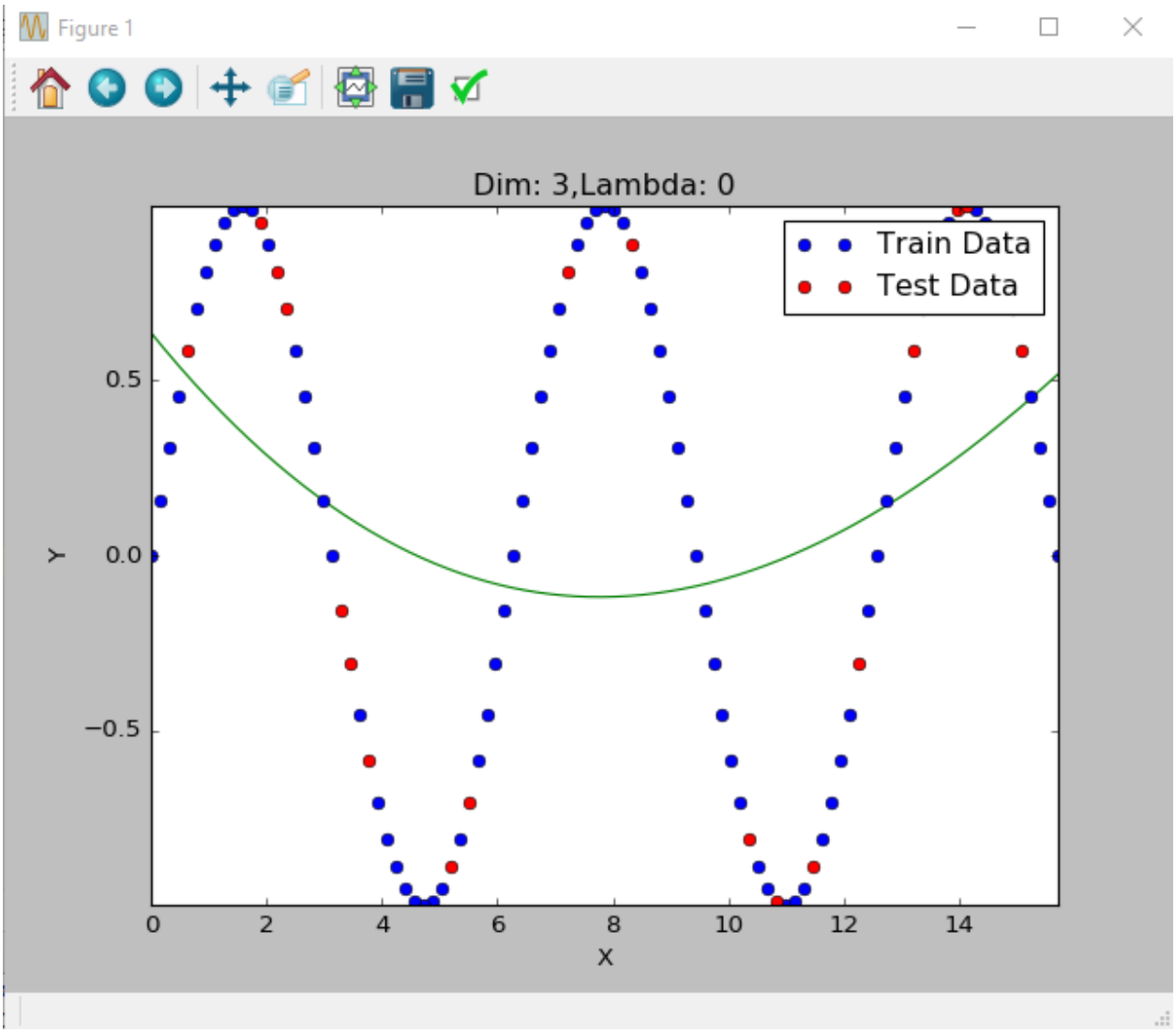


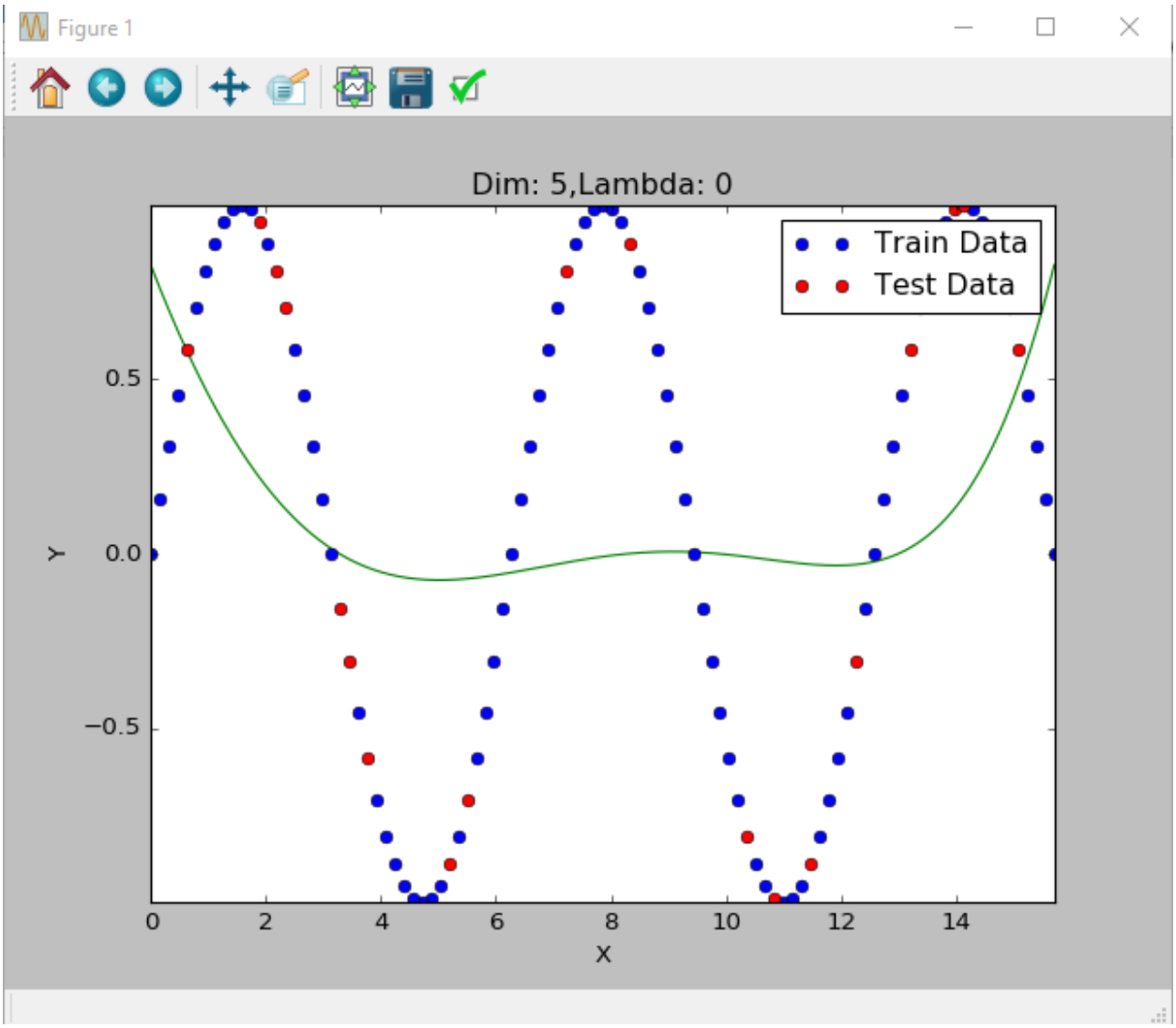
Figure 1
Iter: 10000
MSE
0.225
0.220
0.215
0.210
0.205
0.200
1
2
3
4
5
6
7
8
9
Train Dim 3
Test Dim 3
Train Dim 5
Test Dim 5
Train Dim 7
Test Dim 7

ب) کدهای قسمت ب در فایل b-normal equation.py موجود است که به خوبی کامنت گذاری شده است. نتایج حاصل را در تصویرهای زیر مشاهده می‌کنید:

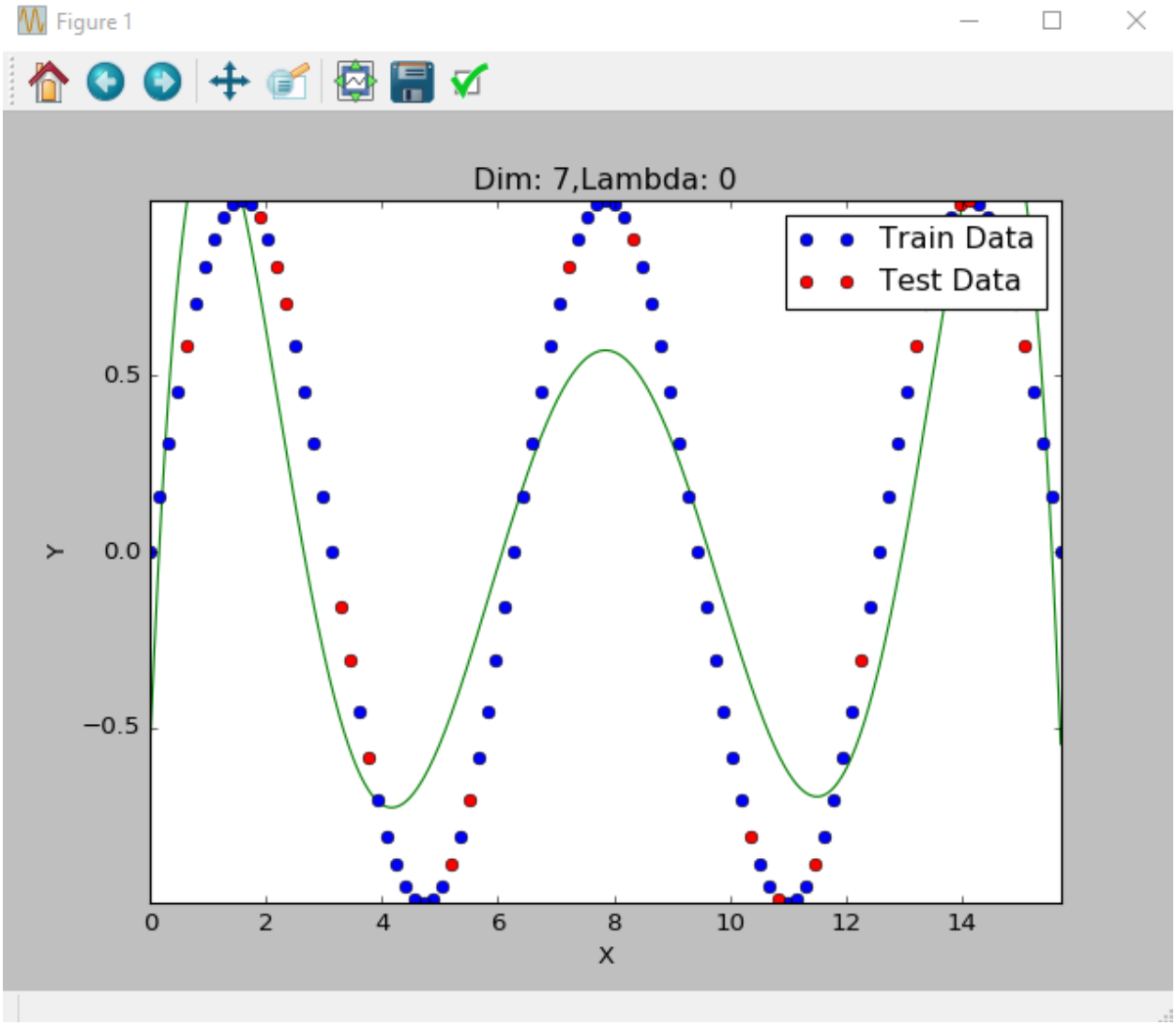
تصویر زیر منحنی پیدا شده با استفاده از معادله نرمال با لامبدا صفر و nonlinear transform به درجه‌ی ۳ را نشان می‌دهد.



تصویر زیر منحنی پیدا شده با استفاده از معادله نرمال با لامبدا صفر و nonlinear transform به درجه‌ی ۵ را نشان می‌دهد.



تصویر زیر منحنی پیدا شده با استفاده از معادله نرمال با لامبدا صفر و nonlinear transform به درجه‌ی ۷ را نشان می‌دهد.



در تصویر زیر خطای مجموعه ی تست و آموزش را برای درجه‌های ۳ ۵ و ۷ را مشاهده می‌کنید:

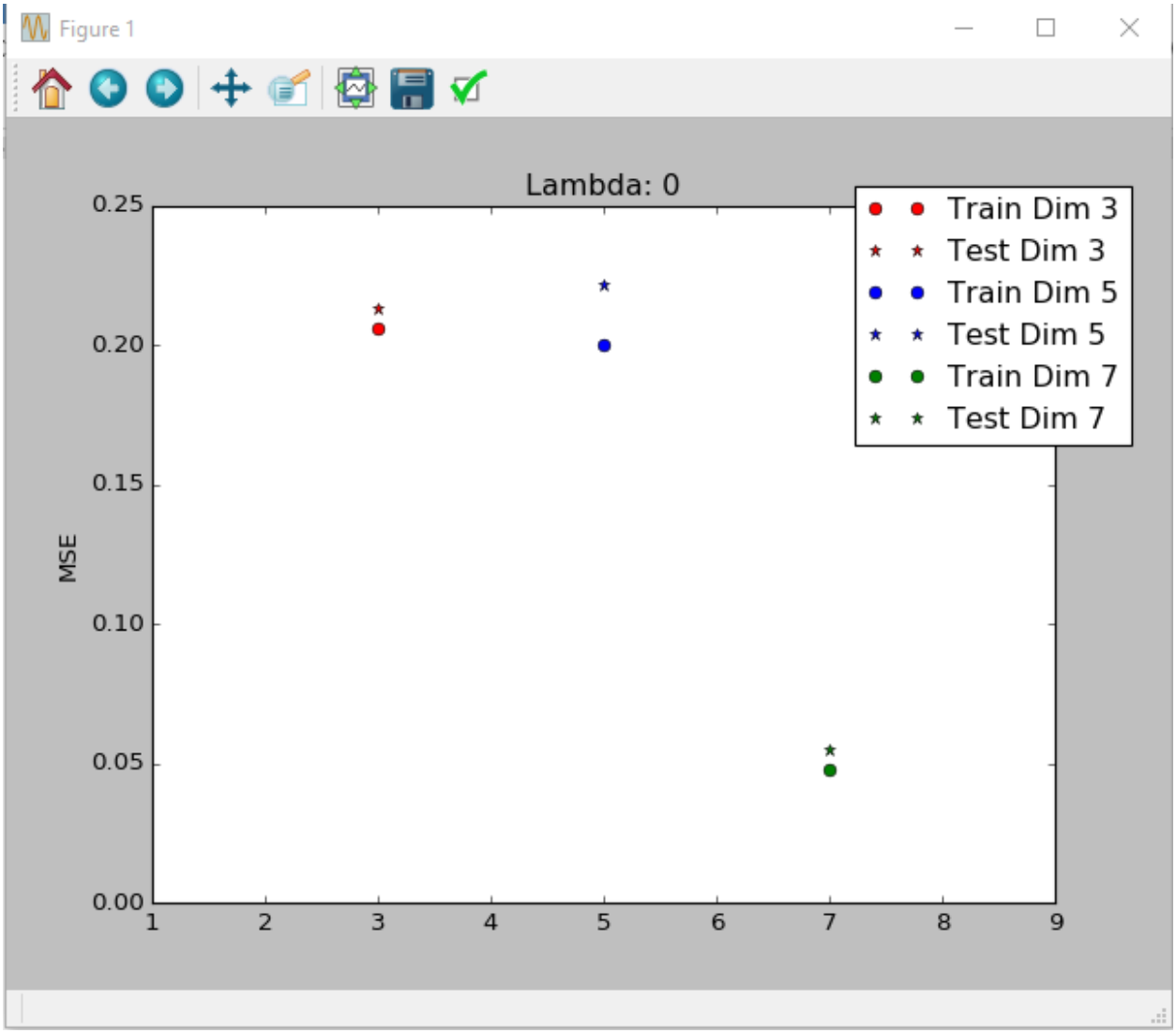
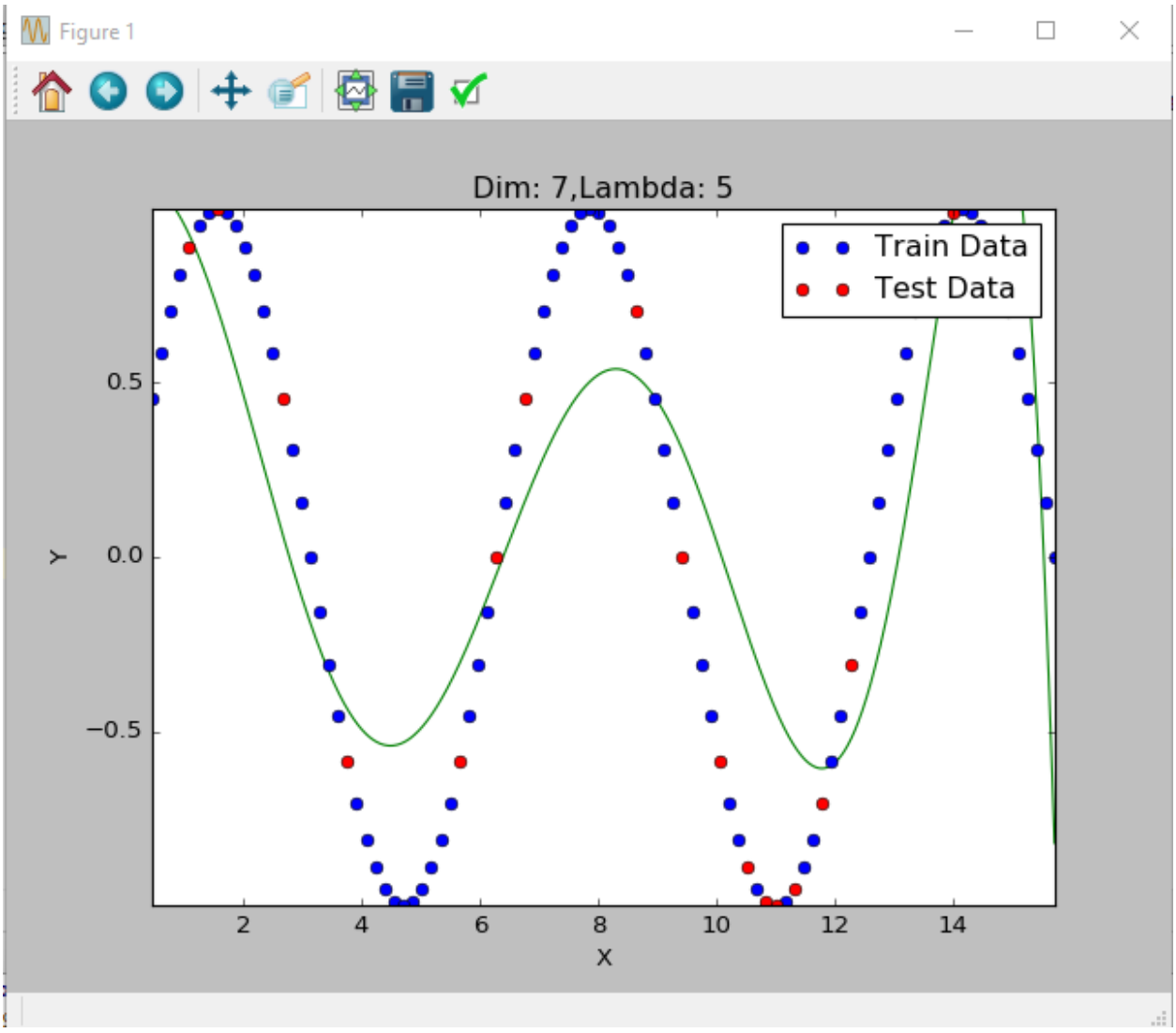


Figure 1
Lambda: 0
MSE
0.25
0.20
0.15
0.10
0.05
0.00
1
2
3
4
5
6
7
8
9
Train Dim 3
Test Dim 3
Train Dim 5
Test Dim 5
Train Dim 7
Test Dim 7

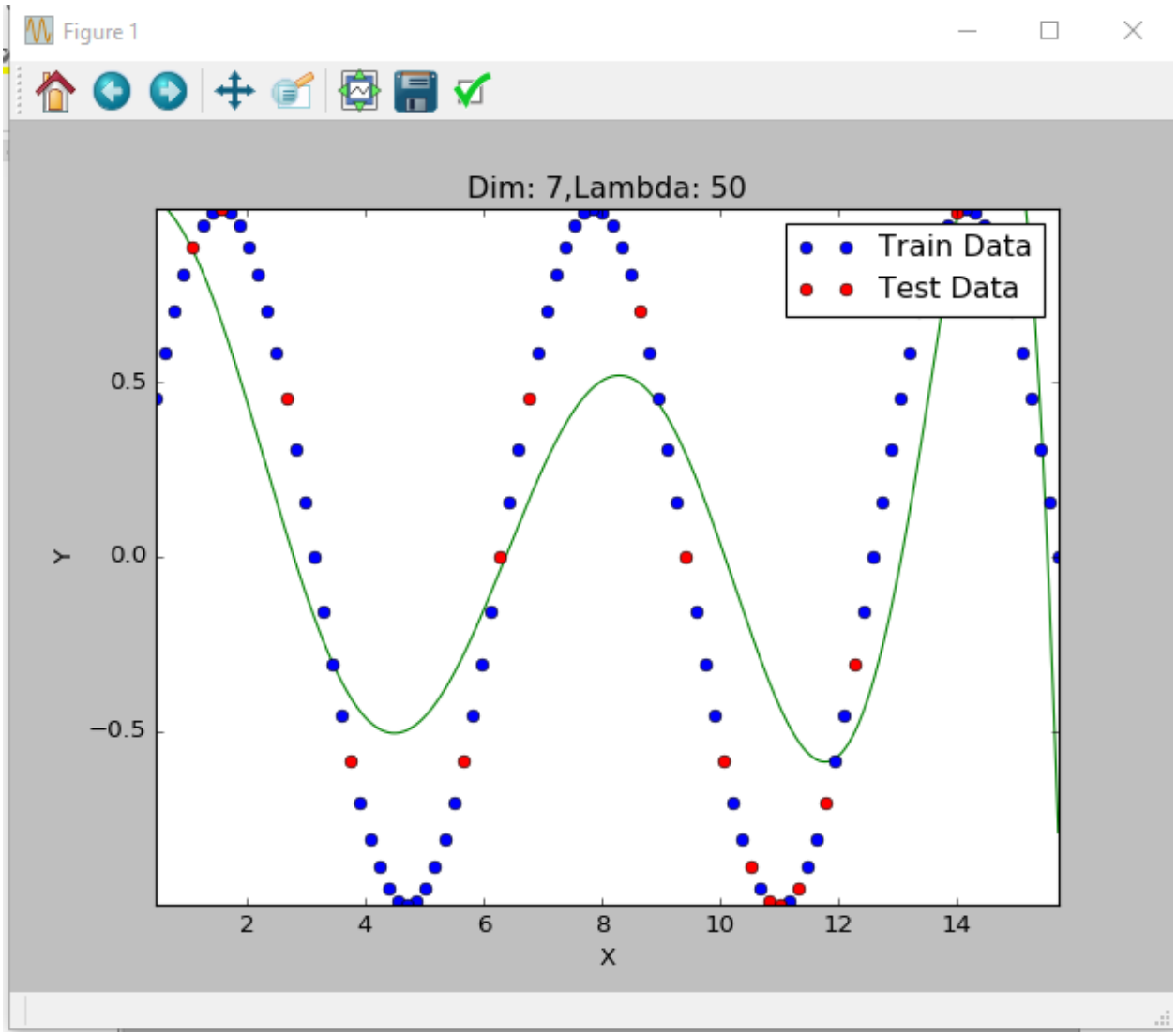
ج)

کدهای کامنت گذاری شده‌ی این سوال در فایل c-normal equation with lambda.py موجود است در زیر نتایج حاصل را مشاهده می کنید:

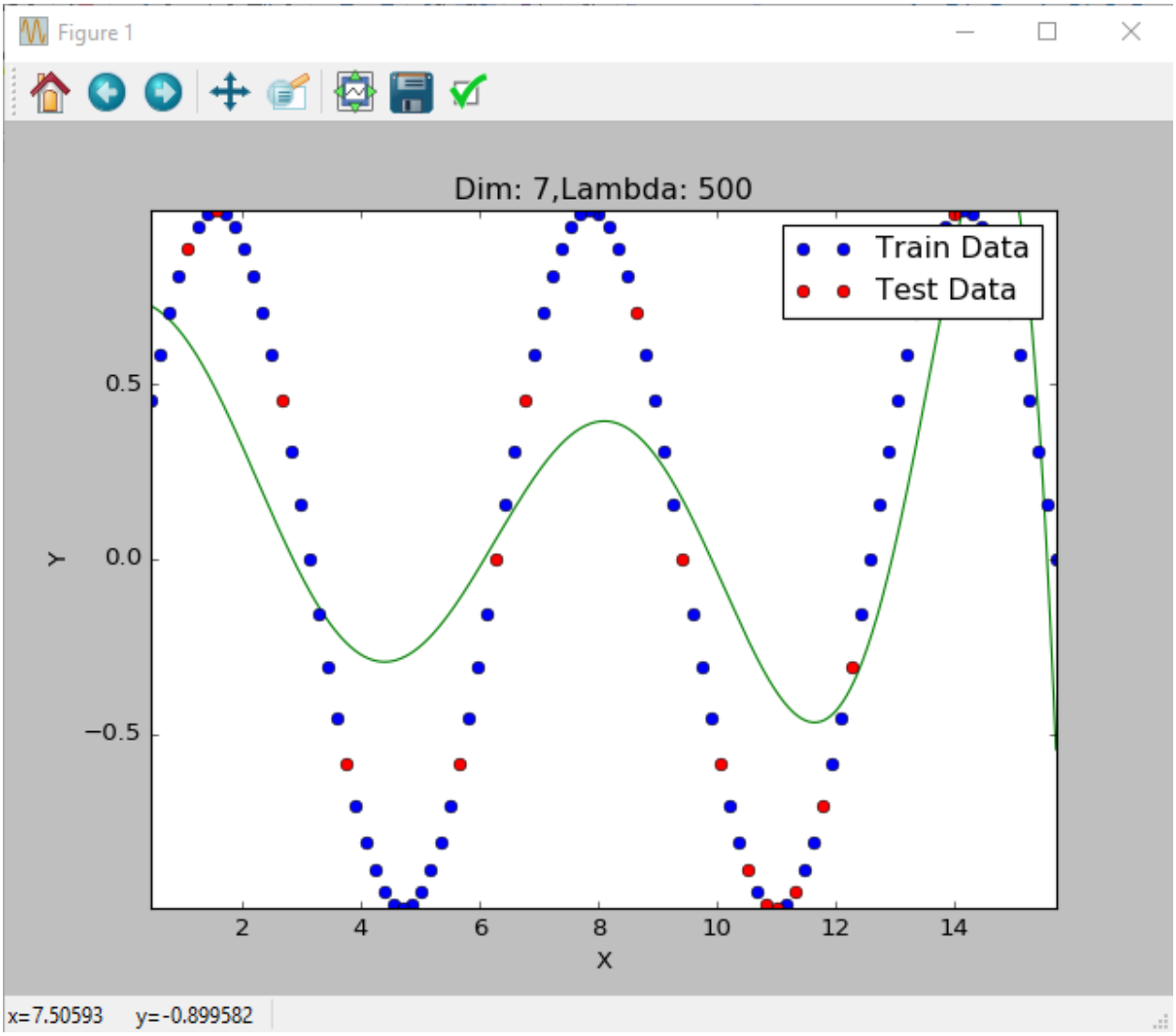
تصویر زیر منحنی پیدا شده با استفاده از معادله نرمال با لامبدا ۵ و nonlinear transform به درجه‌ی ۷ را نشان می‌دهد.



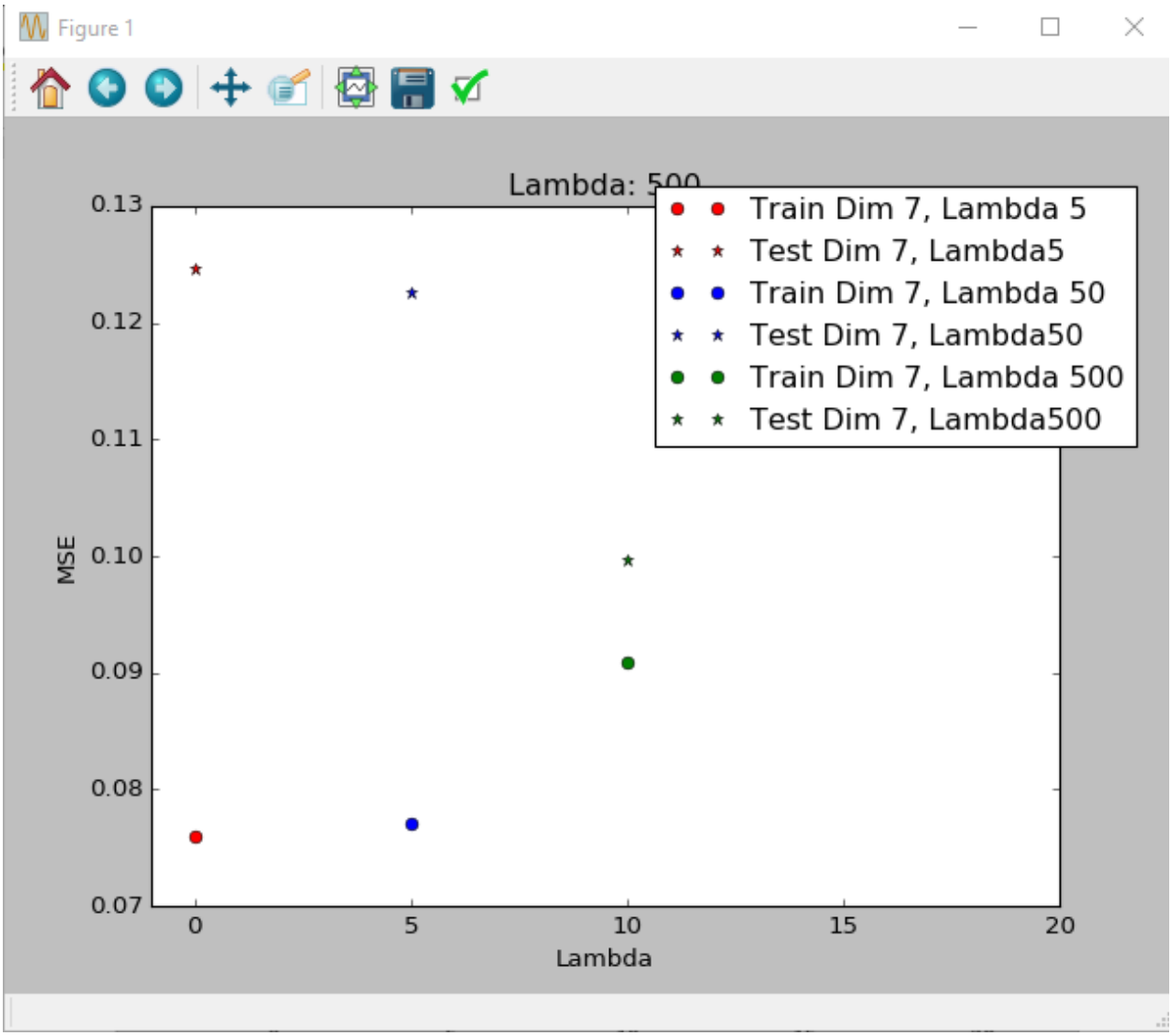
تصویر زیر منحنی پیدا شده با استفاده از معادله نرمال با لامبدا ۵۰ و nonlinear transform به درجه‌ی ۷ را نشان می‌دهد.



تصویر زیر منحنی پیدا شده با استفاده از معادله نرمال با لامبدا ۵۰۰ و nonlinear transform به درجه‌ی ۷ را نشان می‌دهد.



در تصویر زیر خطای مجموعه ی تست و آموزش را برای درجه‌های ۷ و لامبدا ۵ ۵۰ و ۵۰۰ را مشاهده می‌کنید:



بردار ضرایب را در تصویر زیر مشاهده می‌کنید:

[0.899918345184096, 0.1236303863411357, -0.17686276738759243, -0.07085961625524564, 0.040111897769712 23, -0.006116879169635015, 0.0003829147471432252 6, -8.603266864153988e-06]

[0.8987631698431089, -0.003081871078771103 3, -0.07158218883752263, -0.10084796378836713, 0.044083630696047676, -0.006379930207877106, 0.000391141872402130 6, -8.694624590195496-06]

[0.6265504234736387, -0.010743062492449038, -0.04327715885458728 6, -0.07454483475453565, 0.03263561180331699, -0.004783144393980185, 0.000296800532991666 2, -6.665154442997611e-06]