

Matlab matrix II



Newsfeed

Welcome Jiann-Ming Wu

This is your personal newsfeed. As we learn what you like, we'll update you on cool Kaggle stuff that matches your interests. You can also choose to follow topics, kernels, and people you want to keep up with.



Srikar Kashyap • Follow
created this kernel a year ago

SitrakaMatthew upvoted this notebook



12

Jiann-Ming Wu

Joined 2 years ago

Notebook

Data

Output

Comments

Data Sources

- ▼ Fruits 360 dataset
 - ▼ fruits-360
 - LICENSE
 - ▼ papers
 - fruit_recognition_deep_learning.pdf
 - readme.md
 - ▼ Test
 - ▼ Apple Braeburn
 - 321_100.jpg
 - 322_100.jpg
 - 323_100.jpg
 - 324_100.jpg



Fruits 360 dataset

A dataset with 75 fruits and 50590 images

Last Updated: a year ago (Version 1 of 62)

About this Dataset

Fruits 360 dataset: A dataset of images containing fruits

Version: 2018.07.01.0

Content

The following fruits are included: Apples (different varieties: Golden, Golden-Red, Granny Smith, Red, Red Delicious), Apricot, Avocado, Avocado ripe,

Apple Golden 1

Nectarine Flat



搜尋



搜尋



3_100.jpg

4_100.jpg

5_100.jpg

6_100.jpg

7_100.jpg

46_100.jpg

47_100.jpg

50_100.jpg

52_100.jpg

53_100.jpg



8_100.jpg

9_100.jpg

32_100.jpg

33_100.jpg

34_100.jpg

58_100.jpg

59_100.jpg

60_100.jpg

61_100.jpg

63_100.jpg



35_100.jpg

36_100.jpg

37_100.jpg

38_100.jpg

39_100.jpg

66_100.jpg

67_100.jpg

68_100.jpg

69_100.jpg

70_100.jpg



```
load('matlab_train.mat')
import netTransfer using Deep Network Designer
```

The screenshot displays the Deep Network Designer (DND) software interface. The main workspace shows a neural network diagram with the following layers:

- imageInputLayer** (Input Layer)
- conv_1 convolution2dL...** (Convolutional Layer)
- relu_1 reluLayer** (ReLU Layer)

The interface includes a menu bar with options: FILE, BUILD, NAVIGATE, LAYOUT, ANALYSIS, EXPORT. The Layer Library on the left lists various layer types under categories: INPUT, CONVOLUTION AND FULLY CONN... The Workspace on the right shows the following variables and their values:

Name	Value
accuracy	0.9141
ans	18159x1 c
augimdsTrain	1x1 augme
augimdsValida...	1x1 augme
D	9x1 struct
imds	1x1 Image
IMG_5673	4-D uint8
netTransfer	1x1 Series
options	1x1 Traini
probs	18159x11.
Training	1x1 Image
Validation	1x1 Image
x14_100	100x100x.
x30_100	100x100x.
YPred	18159x1 c

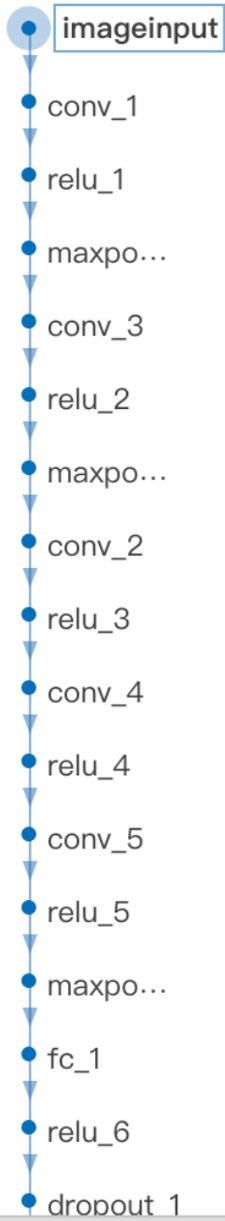
Network from Deep Network Designer

Analysis date: 04-Oct-2019 00:08:46

23  layers

0  warnings

0  errors



ANALYSIS RESULT

	Name	Type	Activations	Learnables
1	imageinput 100x100x3 images	Image Input	100×100×3	-
2	conv_1 96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]	Convolution	23×23×96	Weights 11×11×3×96 Bias 1×1×96
3	relu_1 ReLU	ReLU	23×23×96	-
4	maxpool_1 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	11×11×96	-
5	conv_3 256 5x5x96 convolutions with stride [1 1] and padding 'same'	Convolution	11×11×256	Weights 5×5×96×256 Bias 1×1×256
6	relu_2 ReLU	ReLU	11×11×256	-
7	maxpool_2 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	5×5×256	-
8	conv_2 384 3x3x256 convolutions with stride [1 1] and padding 'same'	Convolution	5×5×384	Weigh... 3×3×256×3... Bias 1×1×384
9	relu_3 ReLU	ReLU	5×5×384	-
10	conv_4 384 3x3x384 convolutions with stride [1 1] and padding 'same'	Convolution	5×5×384	Weigh... 3×3×384×3... Bias 1×1×384
11	relu_4 ReLU	ReLU	5×5×384	-
12	conv_5 256 3x3x384 convolutions with stride [1 1] and padding 'same'	Convolution	5×5×256	Weigh... 3×3×384×2... Bias 1×1×256
13	relu_5 ReLU	ReLU	5×5×256	-
14	maxpool_3 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	2×2×256	-

HOME PLOTS APPS EDITOR PUBLISH VIEW

Insert Comment Indent Breakpoints Run Run and Advance Run Section Advance Run and Time

FILE NAVIGATE EDIT BREAKPOINTS RUN

Users > apple > Desktop > Jiann-Ming Wu > graduateStudent > to boss

- Current Folder
- class_num.mat
 - csv2h5.py
 - fruit_testing.m
 - fruit_testing.m~
 - fruit_training.m
 - keras_test.py
 - keras_train.h5
 - keras_train_keras_tes...
 - keras & TensorFlow.key
 - matlab_train.h5
 - matlab_train.mat
 - net2csv_flatten.m
 - test_h5.m

```
15 'InitialLearnRate', 1e-4, ...
16 'ValidationData', Validation, ...
17 'ValidationFrequency', 6, ...
18 'Verbose', false, ...
19 'ExecutionEnvironment', 'parallel', ...
20 'Plots', 'training-progress');
21 % start training our model
22 % netTransfer = trainNetwork(Training, net, options);
23 load('matlab_train.mat');
24
25 % validation
26 [YPred, probs] = classify(netTransfer, augimdsValidation
27 accuracy = mean(YPred == Validation.Labels);
28 fprintf('testing accuracy %f5.4\n',accuracy);
```

Workspace

Name	Value
accuracy	0.9141
ans	18159x1 c
augimdsTrain	1x1 augme
augimdsValida...	1x1 augme
D	9x1 struct
imds	1x1 Image
IMG_5673	4-D uint8
netTransfer	1x1 Series
options	1x1 Traini
probs	18159x11.
Training	1x1 Image
Validation	1x1 Image
x14_100	100x100x.
x30_100	100x100x.
YPred	18159x1 c

Details

Select a file to view details

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> fruit_testing
testing accuracy 0.9141475.4
fx >>
```

Turing Award 2018

A.M. TURING CENTENARY CELEBRATION WEBCAST



acm

MORE ACM AWARDS



A.M. TURING AWARD



A.M. TURING AWARD WINNERS BY...

ALPHABETICAL LISTING YEAR OF THE AWARD RESEARCH SUBJECT

Geoffrey E Hinton



GEOFFREY HINTON AND YANN LECUN TO DELIVER TURING LECTURE AT FCRC 2019

June 23, 5:15 - 6:30 P.M., Symphony Hall

We are pleased to announce that Geoffrey Hinton and Yann LeCun will deliver the Turing Lecture at FCRC 2019. Hinton's talk, "The Deep Learning Revolution," and LeCun's talk, "The Deep Learning Revolution: The Sequel," will be presented June 23rd from 5:15-6:30pm in Symphony Hall, Phoenix, Arizona.

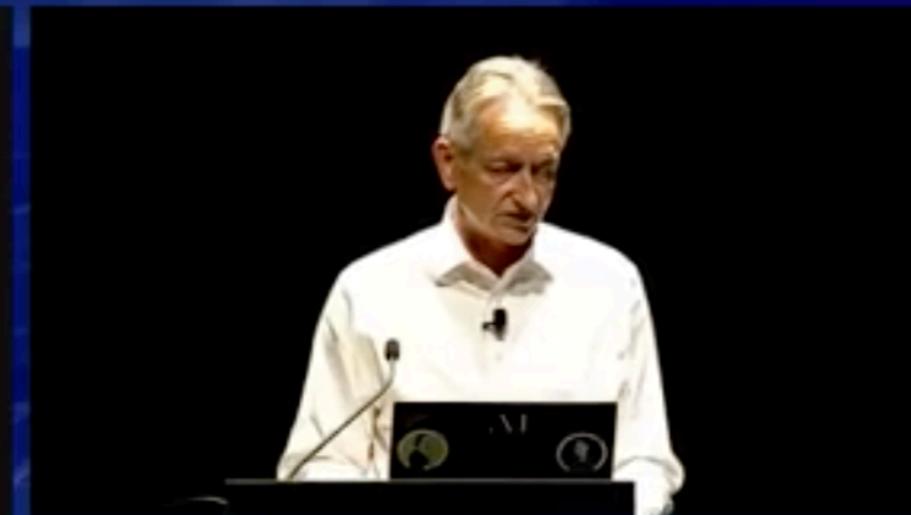
No registration or tickets necessary to attend.

[View the Livestream](#)



Two views of internal representations

- Internal representations are symbolic expressions.
 - A programmer can give them to a computer using an unambiguous language.
 - New representations can be derived by applying rules to existing representations.
- Internal representations are nothing like language.
 - They are large vectors of neural activity.
 - They have direct causal effects on other vectors of neural activity.
 - These vectors are learned from data.



為具備*i*個橫列，*j*個直列的矩陣

1: *i* * *j* can generate a vector. Please use the reshape command to convert the content of vector 1: *i* * *j* into a matrix with *i* horizontal columns and *j* straight columns.

Command Window

```
>> i = 3;
```

```
>> j = 4;
```

```
>>
```

```
ans =
```

```
1    4    7   10  
2    5    8   11  
3    6    9   12
```

```
reshape(1 : i * j, i, j)
```

```
reshape(1:i*j,i,j)
```

變數A為具備i個橫列，j個直列的矩陣，請將A矩陣的第2個橫列取出，儲存在B向量

Variable A is a matrix with i columns and j columns. Please take out the second column of matrix A and store it in vector B.

```
>> i = 3;  
>> j = 4;  
>> A = reshape(1 : i * j, i, j);  
>> B = A
```

```
B =
```

```
2    5    8   11
```

```
(2, :)
```

```
(2,:)
```

變數A為具備i個橫列，j個直列的矩陣，請將A矩陣的第2個直列取出，儲存在C向量

The variable A is a matrix with i horizontal columns and j columns. Please take out the second column of the A matrix and store it in the C vector.

```
>> i = 3;  
>> j = 4;  
>> A = reshape(1 : i * j, i, j);  
>> C = A
```

```
C =
```

```
4
```

```
5
```

```
6
```

```
(:, 2)
```

```
(:,2)
```

變數A為具備i個橫列，j個直列的矩陣，從A矩陣取出兩個直列，並更新矩陣A的內容

Variable A is a matrix with i horizontal columns and j straight columns. Take two straight columns from matrix A and update the contents of matrix A.

```
>> i = 3;  
>> j = 4;  
>> A = reshape(1 : i * j, i, j);  
>> A = A
```

A =

```
7 4  
8 5  
9 6
```

(:, [3 2])

(:,[3 2])

變數A為具備i個橫列，j個直列的矩陣，從矩陣A取出一個子矩陣，並覆蓋矩陣A的內容

Variable A is a matrix with i horizontal columns and j straight columns. Take a submatrix from matrix A and overwrite the contents of matrix A.

```
>> i = 4;  
>> j = 9;  
>> A = reshape(1 : i * j, i, j);  
>> A = A
```

A =

```
10  14  18  22  
11  15  19  23  
12  16  20  24
```

(2 : 4, 3 : 6)

(2:4,3:6)

變數A為具備2個橫列，2個直列的矩陣，以重複矩陣指令repmat，複製A矩陣，並將複製結果儲存在B矩陣

Variable A is a matrix with 2 horizontal columns and 2 straight columns. Use the repeat matrix command repmat to copy the A matrix and store the copy result in the B matrix.

Command Window

```
>> A = [1 2;3 4];
```

```
>> a = 3; b = 4;
```

```
>> B =
```

```
B =
```

```
1 2 1 2 1 2 1 2  
3 4 3 4 3 4 3 4  
1 2 1 2 1 2 1 2  
3 4 3 4 3 4 3 4  
1 2 1 2 1 2 1 2  
3 4 3 4 3 4 3 4
```

```
repmat(A, 3, 4)
```

```
repmat(A,3,4)
```

變數A為具備4個橫列，4個直列的矩陣，使用變數 a、b、c，自矩陣A取出3個橫列，形成以下矩陣

Variable A is a matrix with 4 horizontal columns and 4 straight columns. Using variables a, b, and c, 3 horizontal columns are taken out from matrix A to form the following matrix

```
>> A = reshape(1 : 16, 4, 4);
```

```
>> a = 3; b = 1; c = 2;
```

```
>> A
```

```
ans =
```

```
3    7   11   15
```

```
1    5    9   13
```

```
2    6   10   14
```

```
([a b c], :)
```

```
([a b c],:)
```

變數A為具備4個橫列，4個直列的矩陣，使用變數 a、b、c，自矩陣A取出3個直列，形成以下矩陣

Variable A is a matrix with 4 horizontal columns and 4 straight columns. Using variables a, b, and c, 3 straight columns are taken out from matrix A to form the following matrix

```
>> A = reshape(1 : 16, 4, 4);
```

```
>> a = 1; b = 2; c = 3;
```

```
>> A
```

```
ans =
```

```
    9    1    5  
   10    2    6  
   11    3    7  
   12    4    8
```

(:, [c a b])

(:,[c a b])

變數V為具備3個橫列，4個直列的矩陣，請完成以下指令，將矩陣V的第2橫列與第3橫列對調

The variable V is a matrix with 3 horizontal columns and 4 straight columns. Please complete the following instructions to swap the 2nd column and the 3rd column of the matrix V.

```
>> V=reshape(1:12,3,4);
```

```
>> V([3 2],:)=V
```

```
V =
```

```
     1     4     7    10  
     3     6     9    12  
     2     5     8    11
```

```
([2 3], :)
```

```
([2 3],:)
```

變數V為具備3個橫列，4個直列的矩陣，請將矩陣V的第2直列與第3直列對調

The variable V is a matrix with 3 horizontal columns and 4 columns. Please swap the 2nd column and the 3rd column of the matrix V.

```
>> V=reshape(1:12,3,4);
```

```
>> V
```

```
V =
```

```
1    7    4   10
```

```
2    8    5   11
```

```
3    9    6   12
```

```
V(:, [2 3]) = V(:, [3 2])
```

```
V(:, [3 2]) = V(:, [2 3])
```

變數V為具備3個橫列，4個直列的矩陣，請計算矩陣V的各直列總和

The variable V is a matrix with 3 horizontal columns and 4 straight columns. Please calculate the sum of each straight column of the matrix V.

```
>> V=reshape(1:12,3,4);
```

```
>>
```

```
ans =
```

```
6 15 24 33
```

```
sum(V)
```

變數V為具備3個橫列，4個直列的矩陣，請計算矩陣V的各橫列總和

The variable V is a matrix with 3 horizontal columns and 4 straight columns. Please calculate the sum of each column of the matrix V.

```
>> V=reshape(1:12,3,4);
```

```
>> sum
```

```
ans =
```

```
22
```

```
26
```

```
30
```

(V, 2)

(V,2)

變數V為具備3個橫列，4個直列的矩陣，請計算矩陣V的各直列平均值

The variable V is a matrix with 3 horizontal columns and 4 straight columns. Please calculate the average value of each straight column of the matrix V.

```
>> V=reshape(1:12,3,4);
```

```
>>
```

```
ans =
```

```
2 5 8 11
```

mean(V)

變數V為具備3個橫列，4個直列的矩陣，請計算矩陣V的各橫列平均值

The variable V is a matrix with 3 horizontal columns and 4 straight columns. Please calculate the average value of each horizontal column of the matrix V.

```
>> V=reshape(1:16,4,4);
```

```
>>
```

```
ans =
```

```
7
```

```
8
```

```
9
```

```
10
```

```
mean(V, 2)
```

```
mean(V,2)
```