CH2. Mathematical Building Blocks of Deep-learning

Il-Youp Kwak
Chung-Ang University

Mathematical Building Blocs of Deep-learning

A first look on neural network

Data representations for neural network

The gear of neural networks: tensor operations

The engine of neural networks: gradient-based optimization

Looking back at our first example

A first look on neural network

Figure 2.1 MNIST sample digits

Mnist data analysis example

Hand written digit (28,28 pixcels) for 10 categories (0-9)

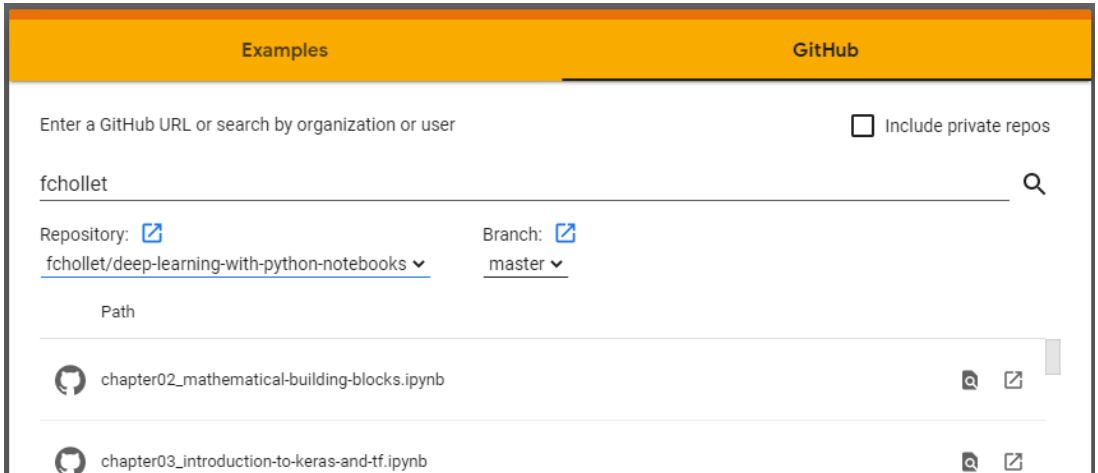
60,000 training set and 10,000 test set

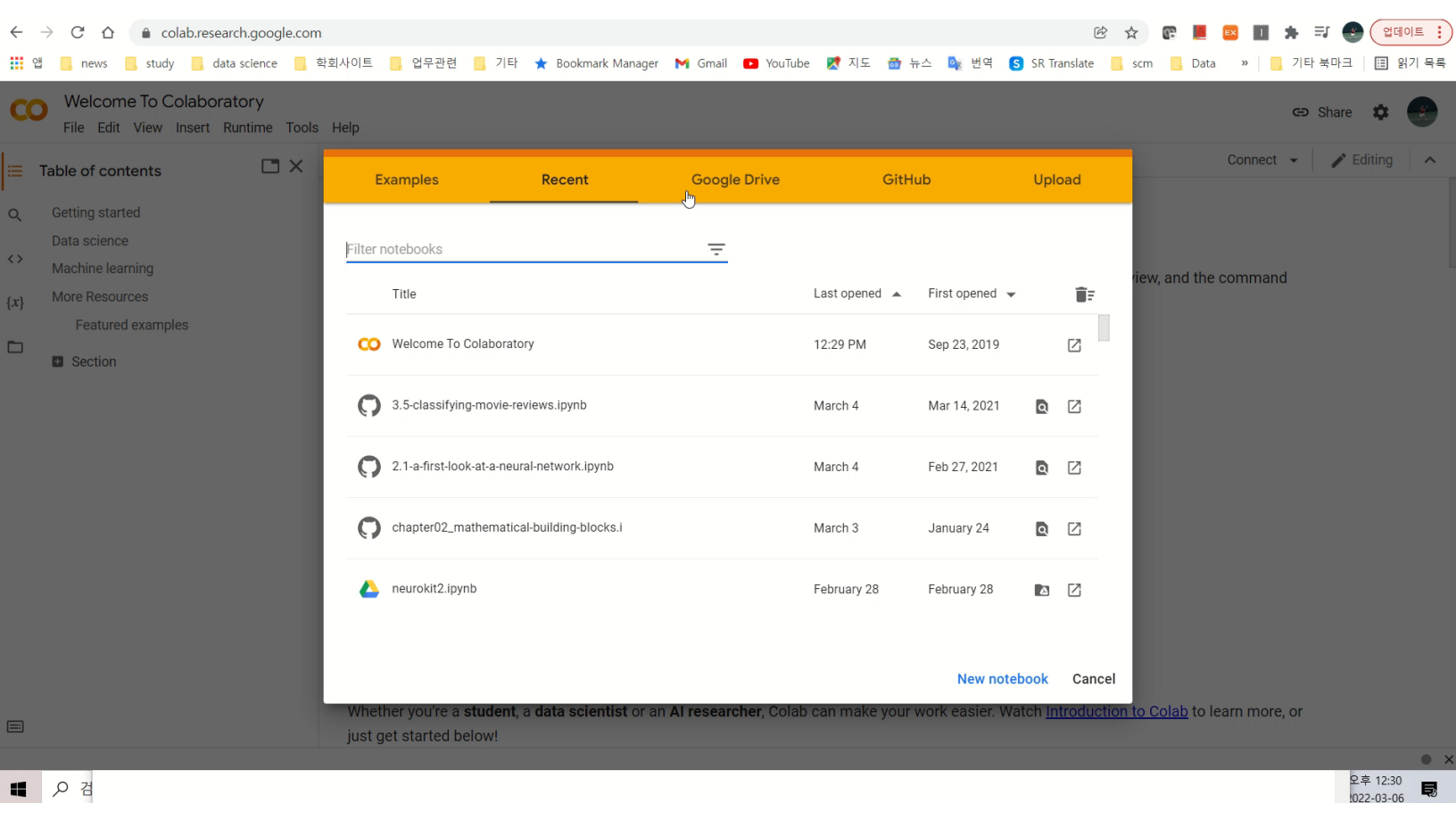
Mnist data

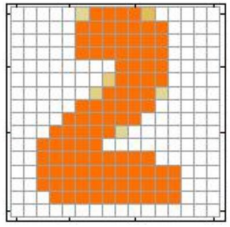
Listing 2.1 Loading the MNIST dataset in Keras

```
from tensorflow.keras.datasets import mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

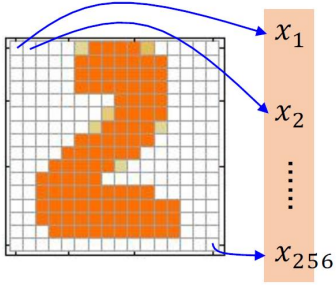
Try these codes from https://colab.research.google.com/github/



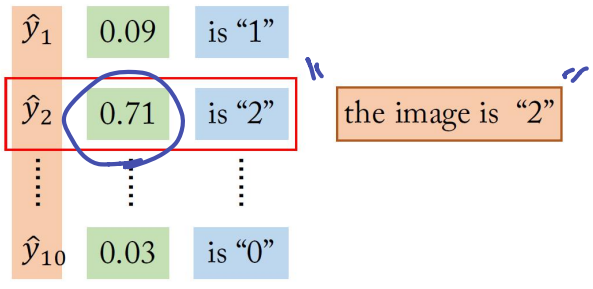

colab.research.google.com
업데이트
news
study
data science
학회사이트
업무관련
기타
Bookmark Manager
Gmail
YouTube
지도
뉴스
번역
SR Translate
scm
Data
기타 북마크
읽기 목록
Welcome To Colaboratory
File Edit View Insert Runtime Tools Help
Share
Connect
Editing
Table of contents
Getting started
Data science
Machine learning
More Resources
Featured examples
Section
Examples
Recent
Google Drive
GitHub
Upload
Filter notebooks
Title
Last opened
First opened
Welcome To Colaboratory
12:29 PM
Sep 23, 2019
3.5-classifying-movie-reviews.ipynb
March 4
Mar 14, 2021
2.1-a-first-look-at-a-neural-network.ipynb
March 4
Feb 27, 2021
chapter02_mathematical-building-blocks.i
March 3
January 24
neurokit2.ipynb
February 28
February 28
New notebook
Cancel
Whether you're a student, a data scientist or an AI researcher, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!



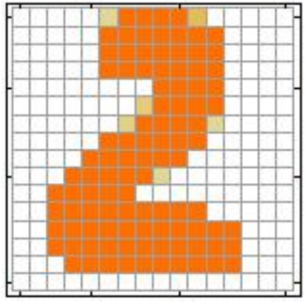
“2”



$16 \times 16 = 256$
ink → 1
no ink → 0



Each dimension represents the confidence of a digit.



input

x_1

x_2

$\vdots$

x_{256}

Machine

output

$\hat{y}_1$

$\hat{y}_2$

$\vdots$

$\hat{y}_{10}$

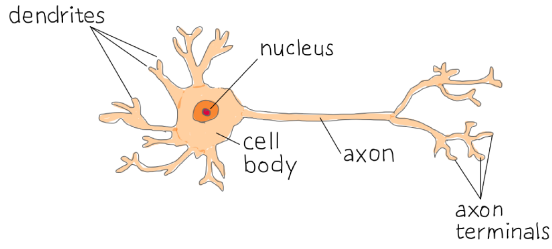
$f : R^{256} \rightarrow R^{10}$

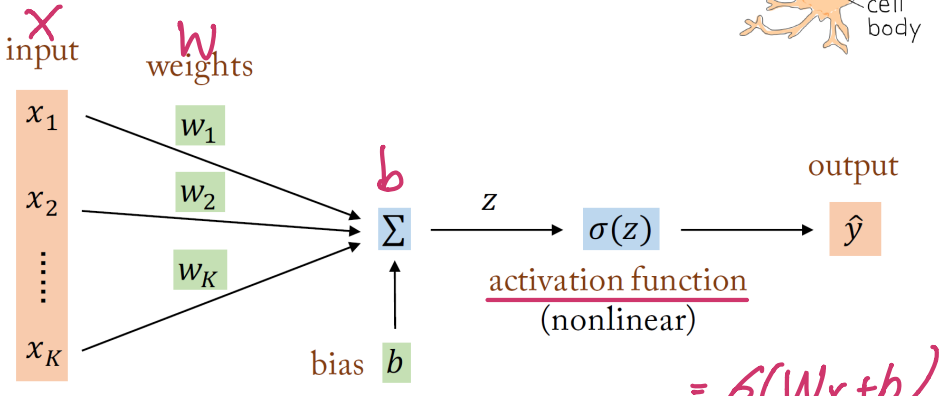
NN

In Deep Learning, the function f is represented by a Neural Network.

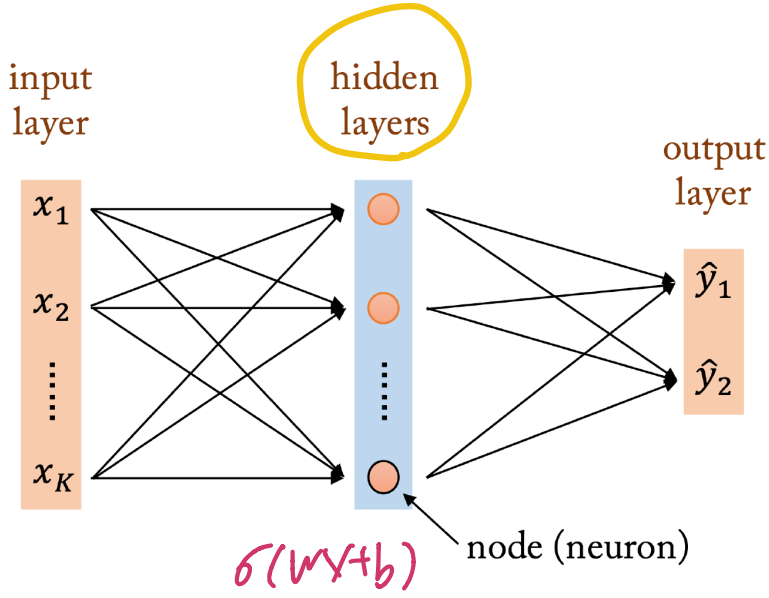
Perceptron (neuron) [Frank Rosenblatt, 1957]

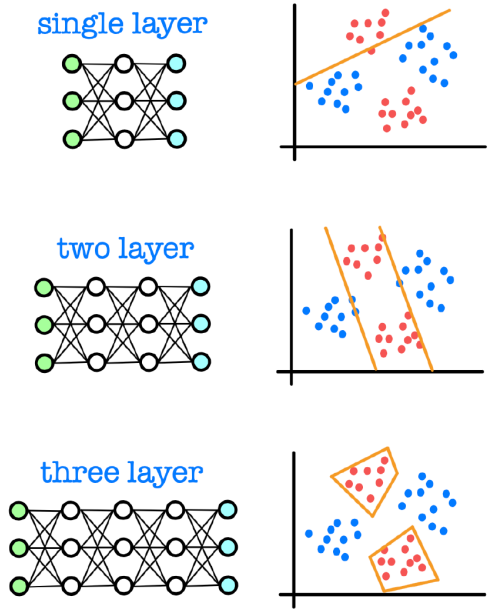
↳ One layer model





Multilayer Perceptron (MLP)



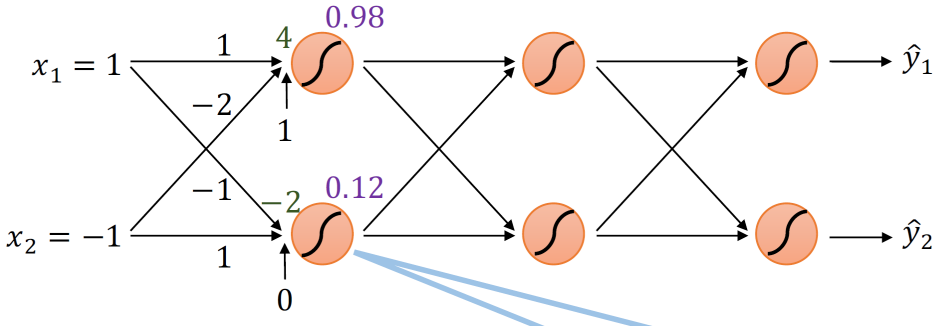


Forward pass computation

$\sigma(Wx+b)$

$$\sigma(w_1x_1 + w_2x_2 + \cdots + w_Kx_K + b)$$

$$0.98 = \sigma(1 \cdot 1 + (-1) \cdot (-2) + 1)$$

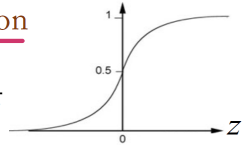


$$\sigma\left(\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \sigma\left(\begin{bmatrix} 4 \\ -2 \end{bmatrix}\right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

W x b

sigmoid function

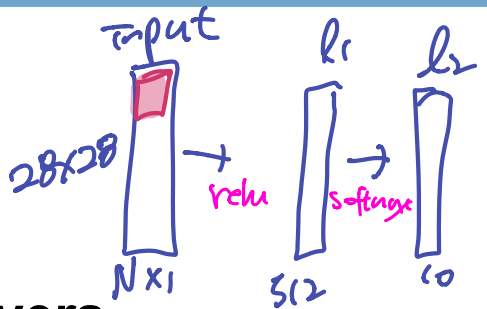
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



activation function

Listing 2.2 The network architecture

```
from tensorflow import keras
from tensorflow.keras import layers
model = keras.Sequential([
    layers.Dense(512, activation="relu"),
    layers.Dense(10, activation="softmax")
])
```



Core building block of NN is layers

Layers extract *representations*

This step define our model f(x; weights)

$$Y = \text{softmax}(W_2 \text{relu}(W_1 x + b_1) + b_2)$$

10×512 $\quad 512 \times N \quad N \times 1 \quad 512 \times 1 \quad 10 \times 1$

$\Rightarrow 512 \times 1$

$\Rightarrow 10 \times 1$

parameter: W_1, W_2, b_1, b_2

chapter02_mathematical-building-blocks.i

File Edit View Insert Runtime Tools Help Cannot save changes

Share

Table of contents

+ Code + Text Copy to Drive

RAM Disk Editing

```
[8] train_images.shape
```

```
(60000, 28, 28)
```

```
[9] len(train_labels)
```

```
60000
```

```
train_labels
```

```
array([5, 0, 4, ..., 5, 6, 8], dtype=uint8)
```

```
[11] test_images.shape
```

```
(10000, 28, 28)
```

```
[12] len(test_labels)
```

```
10000
```

```
[13] test_labels
```

```
array([7, 2, 1, ..., 4, 5, 6], dtype=uint8)
```

The network architecture

Warning: you are connected to a GPU runtime, but not utilizing the GPU. Change to a standard runtime

Completed at 12:36 PM

What's next?

$f_\theta(x) = y$

Loss function: How the network measure its performance

Optimizer: Mechanism through which the network will update itself based on the data it sees and its loss function

Metrics to monitor during training and testing: **Accuracy**

Listing 2.3 The compilation step

```
model.compile(optimizer="rmsprop",
              loss="sparse_categorical_crossentropy",
              metrics=["accuracy"])
```

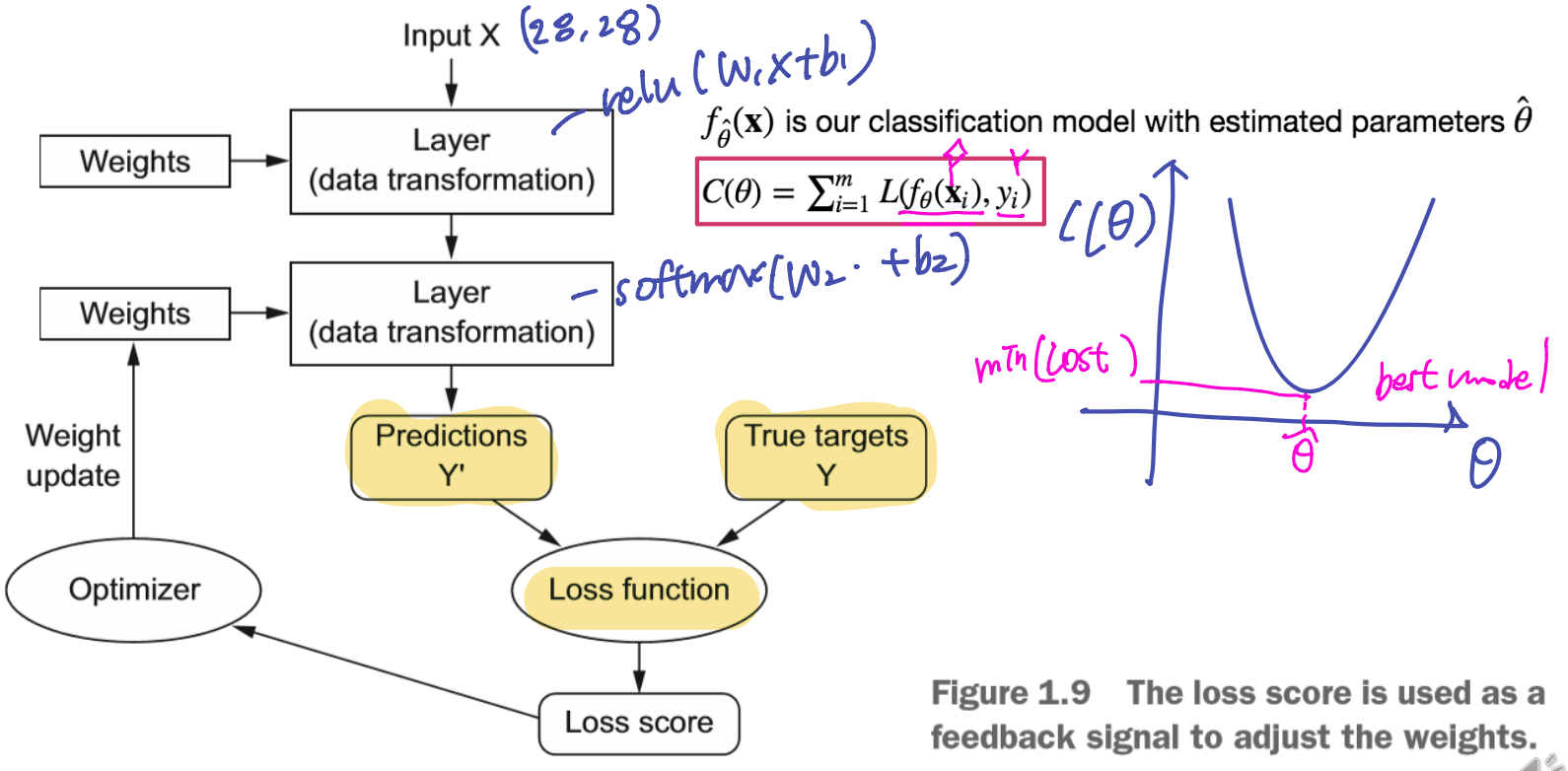


Figure 1.9 The loss score is used as a feedback signal to adjust the weights.

chapter02_mathematical-building-blocks.i

File Edit View Insert Runtime Tools Help Cannot save changes

Share

Table of contents

+ Code + Text Copy to Drive

RAM Disk Editing

[14] 3s
```
from tensorflow import keras
from tensorflow.keras import layers
model = keras.Sequential([
    layers.Dense(512, activation="relu"),
    layers.Dense(10, activation="softmax")
])
```

The compilation step

+ Code + Text

0s
```
model.compile(optimizer="rmsprop",
              loss="sparse_categorical_crossentropy",
              metrics=["accuracy"])
```

[ ]

Preparing the image data

[ ]
```
train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype("float32") / 255
test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype("float32") / 255
```

"Fitting" the model

[ ]
```
model.fit(train_images, train_labels, epochs=5, batch_size=128)
```

0s completed at 1:03 PM

Rescale values from [0,255] to [0,1]

Listing 2.4 Preparing the image data

```
train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype("float32") / 255
test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype("float32") / 255
```

Fitting the model

Listing 2.5 "Fitting" the model

```
>>> model.fit(train_images, train_labels, epochs=5, batch_size=128)
Epoch 1/5
60000/60000 [==============================] - 5s - loss: 0.2524 - acc: 0.9273
Epoch 2/5
51328/60000 [=====================>.....] - ETA: 1s - loss: 0.1035 - acc: 0.9692
```

Make Predictions

Listing 2.6 Using the model to make predictions

test data 중 10개만 예측해봐

```
>>> test_digits = test_images[0:10]
>>> predictions = model.predict(test_digits)
>>> predictions[0]
array([1.0726176e-10, 1.6918376e-10, 6.1314843e-08, 8.4106023e-06,
       2.9967067e-11, 3.0331331e-09, 8.3651971e-14, 9.9999106e-01,
       2.6657624e-08, 3.8127661e-07], dtype=float32)
```

→ 1번 image에 대하여 예측값.

↳ label = 7 이다!!

Evaluating the model

Listing 2.7 Evaluating the model on new data

x y

```
>>> test_loss, test_acc = model.evaluate(test_images, test_labels)
>>> print(f"test_acc: {test_acc}")
test_acc: 0.9785
```

chapter02_mathematical-building-blocks.i

File Edit View Insert Runtime Tools Help Cannot save changes

Share

Table of contents

+ Code + Text Copy to Drive

RAM Disk Editing

```
from tensorflow import keras
from tensorflow.keras import layers
model = keras.Sequential([
    layers.Dense(512, activation="relu"),
    layers.Dense(10, activation="softmax")
])
```

[param]

401920 = 512 × 784 + 512

5130 = 10 × 512 + 10

The compilation step

```
model.compile(optimizer="rmsprop",
              loss="sparse_categorical_crossentropy",
              metrics=["accuracy"])
```

Preparing the image data

```
train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype("float32") / 255
test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype("float32") / 255
```

"Fitting" the model

```
model.fit(train_images, train_labels, epochs=5, batch_size=128)
```

0s completed at 1:04 PM