```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
#x= x1%w1 + x2%w2
x1=3
x2=5;
y=31
w1=3
w2=7 # Uncommented w2
#w3=7 # Removed unused w3
lr=0.01
#learning rate
```

```
for epoch in range(100):
  y_pred = x1 * w1 + x2 * w2
  error=(y-y_pred)**2
  dEw1=2*(y-y_pred)*(-x1)
  dEw2=2*(y-y_pred)*(-x2)
  w1=w1-lr*dEw1
  w2=w2-lr*dEw2
  print('value of w1',w1 ,'value of w2',w2,'prediction',y_pred,'Error',error,'change in w1',dEw1,'cha
```

```
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
value of w1 1.852941176470587 value of w2 5.088235294117648 prediction 31.0 Error 0.0 change in w1 -0
```

```
print(w1,w2)
```

```
1.852941176470587 5.088235294117648
```

```
print(x1*w1+x2*w2)
```

```
31.0
```

keras sequential model

```
from keras import models
from keras.layers import Dense,Dropout
from tensorflow.keras.utils import to_categorical
from keras.datasets import mnist
from tensorflow.keras.utils import model_to_dot
from IPython.display import SVG
import numpy as np
!pip install livelossplot
from livelossplot import PlotLossesKeras # Corrected import
plotloss = PlotLossesKeras() # Corrected instantiation
%matplotlib inline
```

```
Collecting livelossplot
  Downloading livelossplot-0.5.6-py3-none-any.whl.metadata (8.9 kB)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (from livelossplot
Requirement already satisfied: bokeh in /usr/local/lib/python3.12/dist-packages (from livelossplot) (3
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.12/dist-packages (from bokeh->live
Requirement already satisfied: contourpy>=1.2 in /usr/local/lib/python3.12/dist-packages (from bokeh->l
Requirement already satisfied: narwhals>=1.13 in /usr/local/lib/python3.12/dist-packages (from bokeh->l
Requirement already satisfied: numpy>=1.16 in /usr/local/lib/python3.12/dist-packages (from bokeh->live
Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.12/dist-packages (from bokeh->
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.12/dist-packages (from bokeh->live
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.12/dist-packages (from bokeh->li
```

```
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.12/dist-packages (from bokeh->liv
Requirement already satisfied: tornado>=6.2 in /usr/local/lib/python3.12/dist-packages (from bokeh->liv
Requirement already satisfied: xyzservices>=2021.09.1 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matpl
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matpl
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplo
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from ma
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from Jinja2>
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateuti
Downloading livelossplot-0.5.6-py3-none-any.whl (23 kB)
Installing collected packages: livelossplot
Successfully installed livelossplot-0.5.6
```

```
num_rows=28
num_clos=28
num_classs=10
batch_size=128
epoch=10
```

```
def data_summary(x_train,y_train,t_text,y_test):
  print('train image shape',x_train.shape)
  print('train label shape', y_train.shape)
  print('test image shape',x_train.shape)
  print('test label shape', y_test.shape)
  print('train label',y_train)
  print('test label',y_test)
```

```
mnist.load_data()
(x_train,y_train),(x_test,y_test)=mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```

```
data_summary(x_train,y_train,x_test,y_test)
```

```
train image shape (60000, 28, 28)
train label shape (60000,)
test image shape (60000, 28, 28)
test label shape (10000,)
train label [5 0 4 ... 5 6 8]
test label [7 2 1 ... 4 5 6]
```

```
sample=50
print('train label----->',y_train[sample])
```

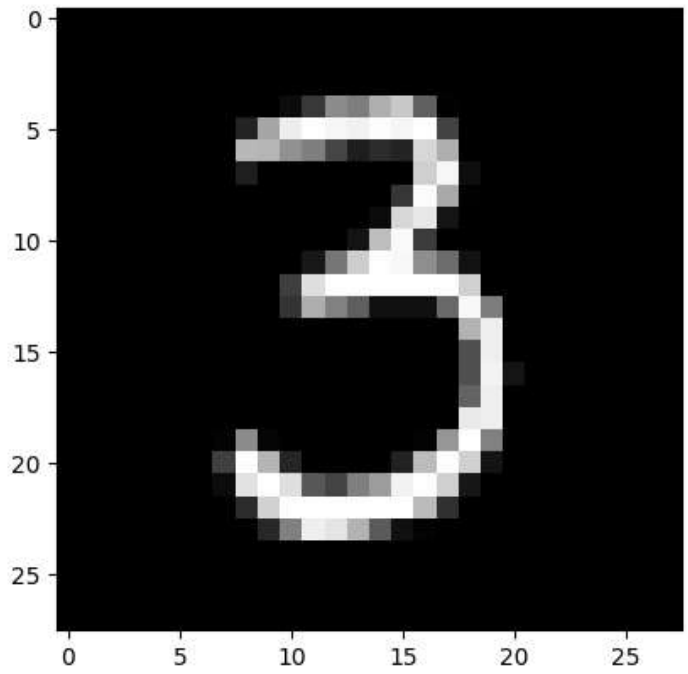
```
train label-----> 3
```

```
image=x_train[sample]
```

```
y_train[sample]
```

```
np.uint8(3)
```

```
fig=plt.figure
plt.imshow(image,cmap='gray')
plt.show()
```




```
x_train.shape
```

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

```
x_train=x_train.reshape(x_train.shape[0],num_rows*num_clos)
x_train=x_train.astype('float32')/255
y_train=to_categorical(y_train,num_classs)
x_test=x_test.reshape(x_test.shape[0],num_rows*num_clos)
y_test=to_categorical(y_test,num_classs)
```

```
data_summary(x_train,y_train,x_test,y_test)
```

```
train image shape (60000, 784)
train label shape (60000, 10)
test image shape (60000, 784)
test label shape (10000, 10)
train label [[0. 0. 0. ... 0. 0. 0.]
 [1. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 1. 0.]]
test label [[0. 0. 0. ... 1. 0. 0.]
 [0. 0. 1. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

```
# build neural network
model=models.Sequential()
```

```
model.add(Dense(512,activation='relu',input_shape=(num_rows*num_clos,)))
model.add(Dropout(0.5))
model.add(Dense(num_classs,activation='softmax'))
model.summary()
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/dense.py:93: UserWarning: Do not pass an
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

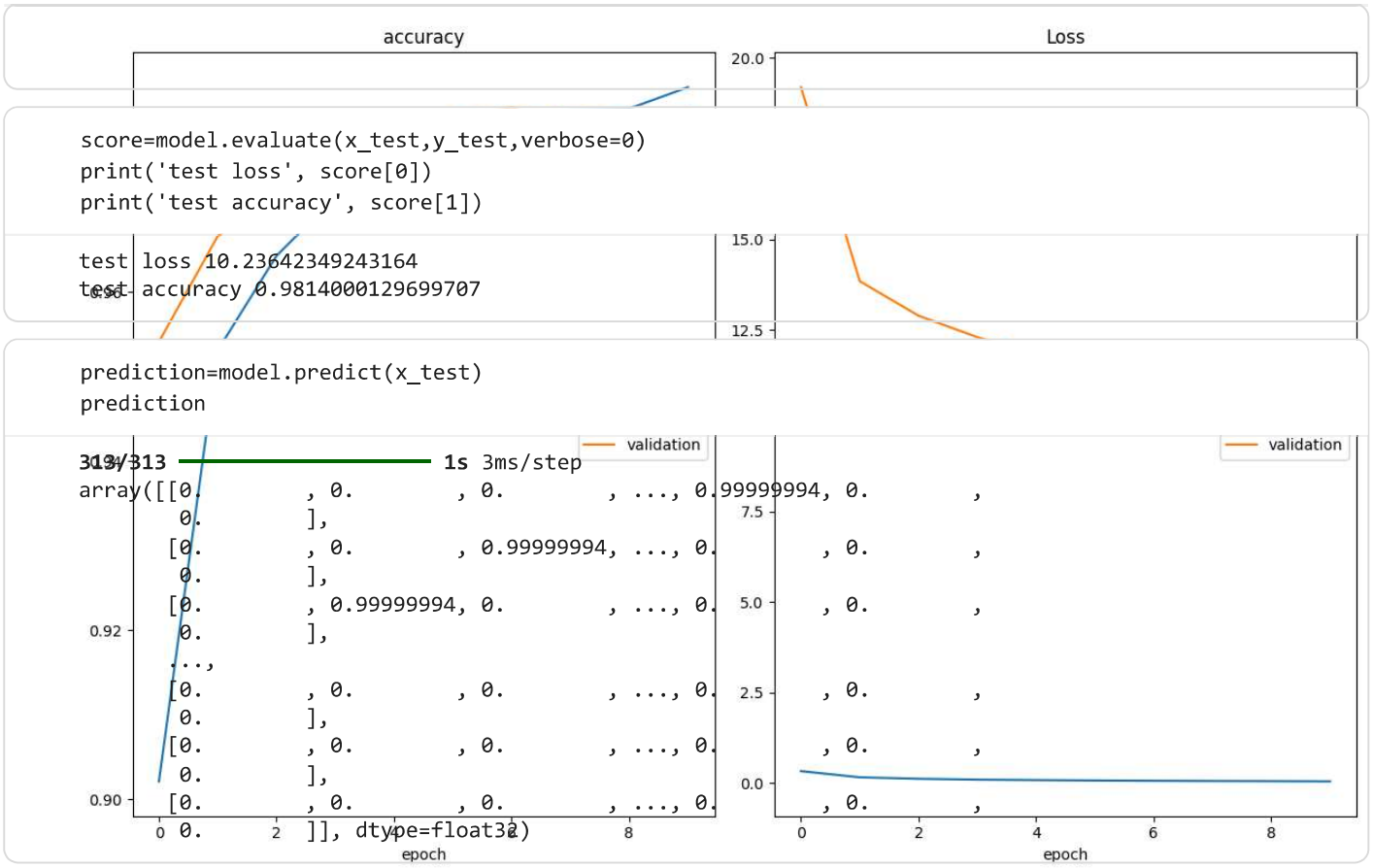
Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	401,920
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 10)	5,130

Total params: 407,050 (1.55 MB)
Trainable params: 407,050 (1.55 MB)
Non-trainable params: 0 (0.00 B)

```
#complie model
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
#train model
model.fit(x_train,y_train,batch_size=batch_size,epochs=10,callbacks=[plotloss],verbose=2,validation_d
```




```
score=model.evaluate(x_test,y_test,verbose=0)
print('test loss', score[0])
print('test accuracy', score[1])
```

```
test loss 10.23642349243164
test accuracy 0.9814000129699707
```

```
prediction=model.predict(x_test)
prediction
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step
array([[0.        , 0.        , 0.        , ..., 0.99999994, 0.        ,
        0.        ],
       [0.        , 0.        , 0.99999994, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.99999994, 0.        , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ]], dtype=float32)
```

Start coding or generate with AI.

```
    validation            (min:    0.934, max:    0.982, cur:    0.981)
Loss
    training              (min:    0.049, max:    0.331, cur:    0.049)
    validation            (min:   10.091, max:   19.196, cur:   10.236)
469/469 - 6s - 12ms/step - accuracy: 0.9842 - loss: 0.0486 - val_accuracy: 0.9814 - val_loss: 10.2364
<keras.src.callbacks.history.History at 0x7ac306ddb0b0>
```