In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [4]:
```
import tensorflow as tf
from tensorflow import keras
```

In [5]:
```
digit_mnist = keras.datasets.mnist
(x_train_full,y_train_full),(x_test,y_test) = digit_mnist.load_data()
```

In [6]:
```
x_train_full.shape
```

Out[6]:
```
(60000, 28, 28)
```

In [7]:
```
y_train_full.shape
```

Out[7]:
```
(60000,)
```

In [8]:
```
x_test.shape
```

Out[8]:
```
(10000, 28, 28)
```

In [9]:
```
y_test.shape
```

Out[9]:
```
(10000,)
```

Reshaping Data

In [10]:
```
x_train_full = x_train_full.reshape(60000,28,28,1)
x_test = x_test.reshape(10000,28,28,1)
```

data normalization

In [11]:
```
x_train_n = x_train_full/255.0
x_test_n = x_test/255.0
```

Creating a validation set

In [12]:
```
x_valid, x_train = x_train_n[:6000],x_train_n[6000:]
y_valid, y_train = y_train_full[:6000],y_train_full[6000:]
x_test = x_test_n
```

Creating CNN model

In [13]:
```
np.random.seed(42)
tf.random.set_seed(42)
```

In [14]:
```
model = keras.models.Sequential()
```

In [15]:
```
model.add(keras.layers.Conv2D(filters = 32,kernel_size = (3,3), strides = 1, padding = 'valid',activation = 're
model.add(keras.layers.MaxPooling2D((2,2)))

model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(200, activation = 'relu'))
model.add(keras.layers.Dense(100, activation = 'relu'))
model.add(keras.layers.Dense(10, activation = 'softmax'))
```

In [16]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320

 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0
 )

 flatten (Flatten)           (None, 5408)              0

 dense (Dense)               (None, 200)               1081800

 dense_1 (Dense)             (None, 100)               20100

 dense_2 (Dense)             (None, 10)                1010

=================================================================
Total params: 1,103,230
Trainable params: 1,103,230
Non-trainable params: 0
_________________________________________________________________
```

In [17]:

```python
model.compile(loss = 'sparse_categorical_crossentropy',
             optimizer = 'sgd',
             metrics = ['accuracy'])
```

In [18]:

```python
model_history = model.fit(x_train,y_train,epochs = 60,batch_size = 64,validation_data = (x_valid,y_valid))
```

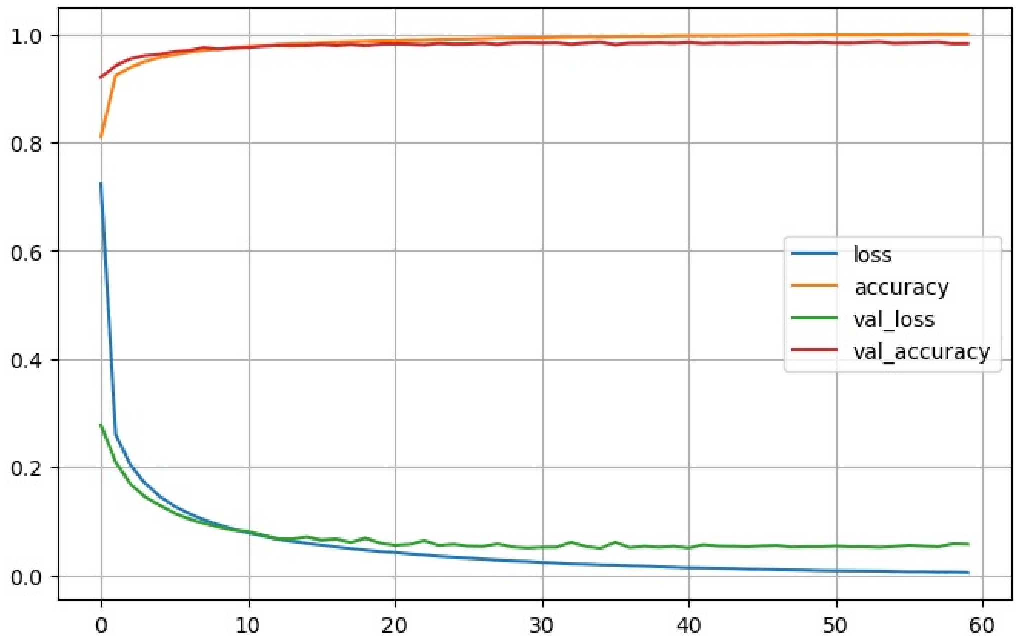
```
Epoch 1/60
844/844 [==============================] - 17s 19ms/step - loss: 0.7232 - accuracy: 0.8105 - val_loss: 0.2777 -
val_accuracy: 0.9202
Epoch 2/60
844/844 [==============================] - 16s 19ms/step - loss: 0.2605 - accuracy: 0.9234 - val_loss: 0.2095 -
val_accuracy: 0.9415
Epoch 3/60
844/844 [==============================] - 15s 18ms/step - loss: 0.2041 - accuracy: 0.9386 - val_loss: 0.1690 -
val_accuracy: 0.9542
Epoch 4/60
844/844 [==============================] - 15s 18ms/step - loss: 0.1703 - accuracy: 0.9490 - val_loss: 0.1452 -
val_accuracy: 0.9602
Epoch 5/60
844/844 [==============================] - 15s 18ms/step - loss: 0.1461 - accuracy: 0.9564 - val_loss: 0.1298 -
val_accuracy: 0.9625
Epoch 6/60
844/844 [==============================] - 15s 18ms/step - loss: 0.1280 - accuracy: 0.9615 - val_loss: 0.1147 -
val_accuracy: 0.9675
Epoch 7/60
844/844 [==============================] - 15s 18ms/step - loss: 0.1139 - accuracy: 0.9666 - val_loss: 0.1042 -
val_accuracy: 0.9698
Epoch 8/60
844/844 [==============================] - 15s 18ms/step - loss: 0.1027 - accuracy: 0.9696 - val_loss: 0.0965 -
val_accuracy: 0.9753
Epoch 9/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0940 - accuracy: 0.9719 - val_loss: 0.0896 -
val_accuracy: 0.9723
Epoch 10/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0854 - accuracy: 0.9746 - val_loss: 0.0837 -
val_accuracy: 0.9747
Epoch 11/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0788 - accuracy: 0.9768 - val_loss: 0.0815 -
val_accuracy: 0.9753
Epoch 12/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0730 - accuracy: 0.9782 - val_loss: 0.0744 -
val_accuracy: 0.9777
Epoch 13/60
844/844 [==============================] - 17s 20ms/step - loss: 0.0675 - accuracy: 0.9798 - val_loss: 0.0676 -
val_accuracy: 0.9792
Epoch 14/60
844/844 [==============================] - 16s 19ms/step - loss: 0.0632 - accuracy: 0.9818 - val_loss: 0.0671 -
val_accuracy: 0.9785
Epoch 15/60
844/844 [==============================] - 97s 116ms/step - loss: 0.0596 - accuracy: 0.9821 - val_loss: 0.0711
- val_accuracy: 0.9790
Epoch 16/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0561 - accuracy: 0.9835 - val_loss: 0.0652 -
val_accuracy: 0.9807
Epoch 17/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0529 - accuracy: 0.9844 - val_loss: 0.0673 -
val_accuracy: 0.9788
Epoch 18/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0495 - accuracy: 0.9854 - val_loss: 0.0611 -
val_accuracy: 0.9808
Epoch 19/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0468 - accuracy: 0.9863 - val_loss: 0.0689 -
val_accuracy: 0.9787
Epoch 20/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0439 - accuracy: 0.9872 - val_loss: 0.0601 -
```

```
val_accuracy: 0.9813
Epoch 21/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0425 - accuracy: 0.9873 - val_loss: 0.0558 -
val_accuracy: 0.9817
Epoch 22/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0396 - accuracy: 0.9882 - val_loss: 0.0576 -
val_accuracy: 0.9813
Epoch 23/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0377 - accuracy: 0.9889 - val_loss: 0.0645 -
val_accuracy: 0.9800
Epoch 24/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0354 - accuracy: 0.9899 - val_loss: 0.0555 -
val_accuracy: 0.9827
Epoch 25/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0334 - accuracy: 0.9903 - val_loss: 0.0579 -
val_accuracy: 0.9812
Epoch 26/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0321 - accuracy: 0.9904 - val_loss: 0.0544 -
val_accuracy: 0.9815
Epoch 27/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0302 - accuracy: 0.9912 - val_loss: 0.0539 -
val_accuracy: 0.9830
Epoch 28/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0283 - accuracy: 0.9921 - val_loss: 0.0586 -
val_accuracy: 0.9807
Epoch 29/60
844/844 [==============================] - 91s 108ms/step - loss: 0.0271 - accuracy: 0.9924 - val_loss: 0.0529
- val_accuracy: 0.9837
Epoch 30/60
844/844 [==============================] - 14s 17ms/step - loss: 0.0261 - accuracy: 0.9928 - val_loss: 0.0508 -
val_accuracy: 0.9843
Epoch 31/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0243 - accuracy: 0.9929 - val_loss: 0.0521 -
val_accuracy: 0.9835
Epoch 32/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0230 - accuracy: 0.9937 - val_loss: 0.0524 -
val_accuracy: 0.9842
Epoch 33/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0215 - accuracy: 0.9941 - val_loss: 0.0614 -
val_accuracy: 0.9808
Epoch 34/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0209 - accuracy: 0.9944 - val_loss: 0.0538 -
val_accuracy: 0.9835
Epoch 35/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0196 - accuracy: 0.9948 - val_loss: 0.0502 -
val_accuracy: 0.9852
Epoch 36/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0192 - accuracy: 0.9951 - val_loss: 0.0617 -
val_accuracy: 0.9802
Epoch 37/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0181 - accuracy: 0.9954 - val_loss: 0.0517 -
val_accuracy: 0.9832
Epoch 38/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0174 - accuracy: 0.9957 - val_loss: 0.0540 -
val_accuracy: 0.9833
Epoch 39/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0163 - accuracy: 0.9956 - val_loss: 0.0523 -
val_accuracy: 0.9842
Epoch 40/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0151 - accuracy: 0.9965 - val_loss: 0.0540 -
val_accuracy: 0.9833
Epoch 41/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0141 - accuracy: 0.9967 - val_loss: 0.0509 -
val_accuracy: 0.9848
Epoch 42/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0138 - accuracy: 0.9967 - val_loss: 0.0571 -
val_accuracy: 0.9827
Epoch 43/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0133 - accuracy: 0.9969 - val_loss: 0.0543 -
val_accuracy: 0.9838
Epoch 44/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0128 - accuracy: 0.9968 - val_loss: 0.0539 -
val_accuracy: 0.9833
Epoch 45/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0118 - accuracy: 0.9973 - val_loss: 0.0532 -
val_accuracy: 0.9840
Epoch 46/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0113 - accuracy: 0.9973 - val_loss: 0.0547 -
val_accuracy: 0.9837
Epoch 47/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0107 - accuracy: 0.9977 - val_loss: 0.0555 -
val_accuracy: 0.9840
Epoch 48/60
844/844 [==============================] - 17s 20ms/step - loss: 0.0102 - accuracy: 0.9980 - val_loss: 0.0527 -
val_accuracy: 0.9845
Epoch 49/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0097 - accuracy: 0.9980 - val_loss: 0.0534 -
val_accuracy: 0.9838
Epoch 50/60
```

```
844/844 [==============================] - 15s 18ms/step - loss: 0.0089 - accuracy: 0.9984 - val_loss: 0.0532 -
val_accuracy: 0.9848
Epoch 51/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0085 - accuracy: 0.9985 - val_loss: 0.0544 -
val_accuracy: 0.9837
Epoch 52/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0082 - accuracy: 0.9984 - val_loss: 0.0532 -
val_accuracy: 0.9835
Epoch 53/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0080 - accuracy: 0.9983 - val_loss: 0.0532 -
val_accuracy: 0.9847
Epoch 54/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0079 - accuracy: 0.9986 - val_loss: 0.0521 -
val_accuracy: 0.9855
Epoch 55/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0072 - accuracy: 0.9986 - val_loss: 0.0538 -
val_accuracy: 0.9830
Epoch 56/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0065 - accuracy: 0.9990 - val_loss: 0.0559 -
val_accuracy: 0.9837
Epoch 57/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0066 - accuracy: 0.9989 - val_loss: 0.0544 -
val_accuracy: 0.9843
Epoch 58/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0060 - accuracy: 0.9993 - val_loss: 0.0533 -
val_accuracy: 0.9852
Epoch 59/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0059 - accuracy: 0.9991 - val_loss: 0.0588 -
val_accuracy: 0.9820
Epoch 60/60
844/844 [==============================] - 15s 18ms/step - loss: 0.0055 - accuracy: 0.9991 - val_loss: 0.0581 -
val_accuracy: 0.9822
```

In [19]:

```python
pd.DataFrame(model_history.history).plot(figsize = (8,5))
plt.grid(True)
plt.show()
```



In [20]:

```python
ev = model.evaluate(x_test_n,y_test)
```

```
313/313 [==============================] - 2s 5ms/step - loss: 0.0515 - accuracy: 0.9848
```

In [21]:

```python
ev
```

Out[21]:

```
[0.05145538970828056, 0.9847999811172485]
```

In [22]:

```python
import os
os.environ['KMP_DUPLICATE_LIB_OK'] = 'True'
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js