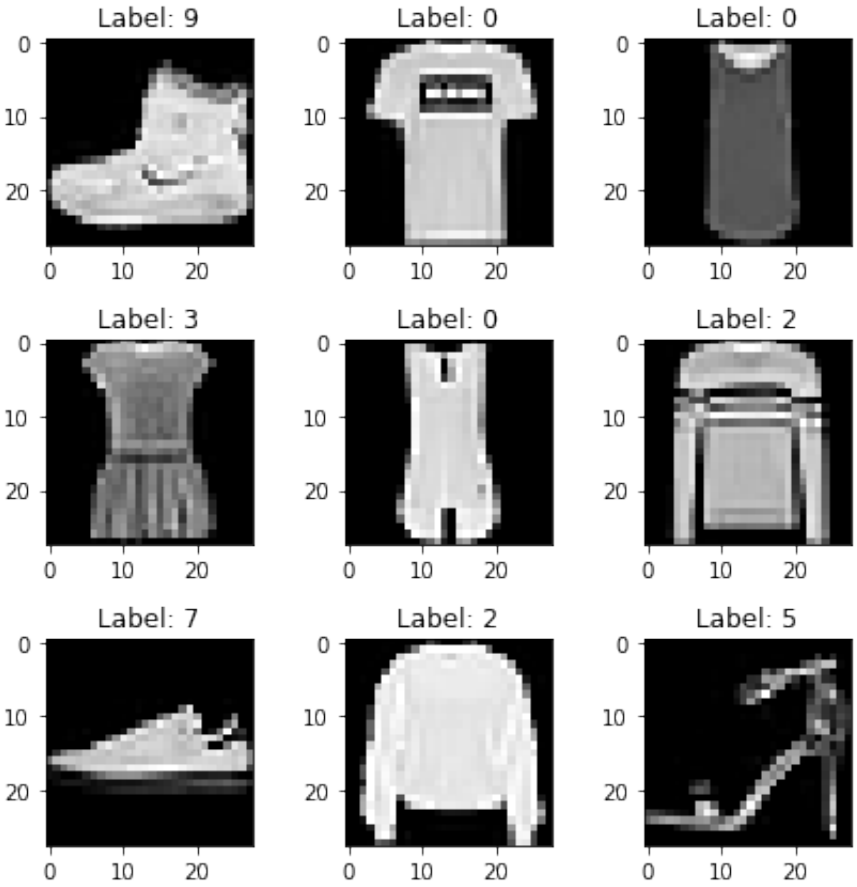
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

from utils import mnist_reader
from sklearn.linear_model import LogisticRegression

import warnings
warnings.filterwarnings('ignore')

seed = 1234
np.random.seed(seed)
```

In [2]:
```
X_train, y_train = mnist_reader.load_mnist('data/fashion', kind='train
X_test, y_test = mnist_reader.load_mnist('data/fashion', kind='t10k')
```

In [3]:
```
plt.rcParams['image.interpolation'] = 'nearest'
plt.rcParams['image.cmap'] = 'gray'

fig, axes = plt.subplots(3, 3, figsize=(6, 6))

k = 0
while k < 9:
    for i in range(3):
        for j in range(3):
            axes[i,j].imshow(X_train[k].reshape([28,28]))
            axes[i,j].set_title(f'Label: {y_train[k]}')
            k = k+1

fig.tight_layout()
```



In [4]:
```
classifier = LogisticRegression(penalty='none')
classifier.fit(X_train,y_train)

test_predictions = classifier.predict(X_test)
test_accuracy = classifier.score(X_test,y_test)

print('Number of training examples: ', X_train.shape[0])
print('Number of testing examples: ', X_test.shape[0])
print('\nTesting accuracy:', format( 100*test_accuracy , '.2f') )
```

```
Number of training examples:  60000
Number of testing examples:  10000

Testing accuracy: 84.05
```

In [5]:
```
classifier = LogisticRegression(penalty='none',fit_intercept=False)
classifier.fit(X_train,y_train)

test_predictions = classifier.predict(X_test)
test_accuracy = classifier.score(X_test,y_test)

print('Number of training examples: ', X_train.shape[0])
print('Number of testing examples: ', X_test.shape[0])
print('\nTesting accuracy:', format( 100*test_accuracy , '.2f') )
```

```
Number of training examples:  60000
Number of testing examples:  10000

Testing accuracy: 84.09
```

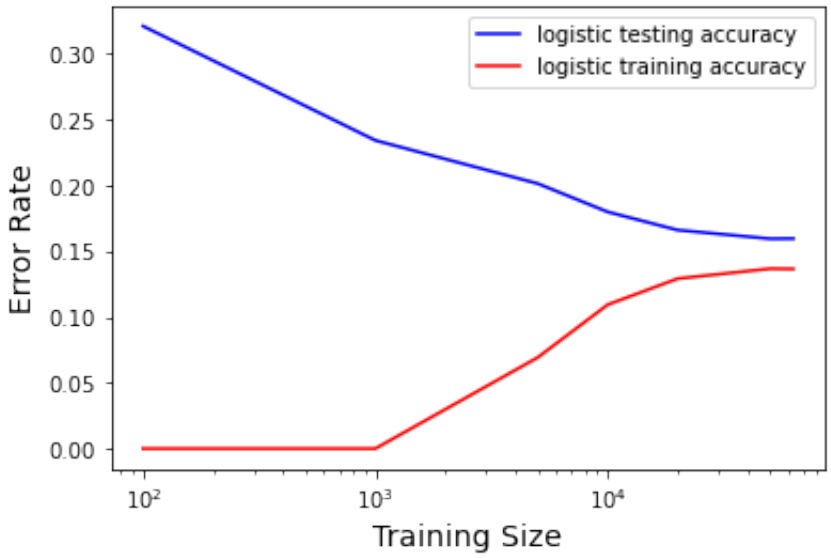
In [13]:
```
classifier = LogisticRegression(penalty='l1',solver='liblinear',fit_ir
classifier.fit(X_train,y_train)

test_predictions = classifier.predict(X_test)
test_accuracy = classifier.score(X_test,y_test)

print('Number of training examples: ', X_train.shape[0])
print('Number of testing examples: ', X_test.shape[0])
print('\nTesting accuracy:', format( 100*test_accuracy , '.2f') )
```

```
Number of training examples:  60000
Number of testing examples:  10000

Testing accuracy: 83.78
```

Different training data

In [6]:
```
n_tr = [100, 1000, 5000, 10000, 20000, 50000, 63000]
logistic_train_accuracies = []
logistic_test_accuracies = []

for i in n_tr:
    classifier = LogisticRegression(random_state=seed, penalty='none',
    classifier.fit(X_train[:i],y_train[:i])

    logistic_test_accuracy = classifier.score(X_test,y_test)
    logistic_train_accuracy = classifier.score(X_train[:i],y_train[:i]

    logistic_test_accuracies.append(1-logistic_test_accuracy)
    logistic_train_accuracies.append(1-logistic_train_accuracy)

fig, axes = plt.subplots()

axes.semilogx(n_tr, logistic_test_accuracies, color='blue', label='log
axes.semilogx(n_tr, logistic_train_accuracies, color='red', label='log

axes.set_xlabel('Training Size', fontsize=14)
axes.set_ylabel('Error Rate', fontsize=14)

axes.legend()
```

Out[6]:
```
<matplotlib.legend.Legend at 0x7f9dab6d5ee0>
```



In [7]:
```
classifier = LogisticRegression(penalty='none',fit_intercept=False)
classifier.fit(X_train[:50000],y_train[:50000])

train_accuracy = classifier.score(X_train[:50000],y_train[:50000])
test_accuracy = classifier.score(X_test,y_test)

print('Number of training examples: ', X_train.shape[0])
print('Number of testing examples: ', X_test.shape[0])
print('\nTesting accuracy:', format( 100*test_accuracy , '.2f') )
print('\nTraining accuracy:', format( 100*train_accuracy , '.2f') )
```

```
Number of training examples:  60000
Number of testing examples:  10000

Testing accuracy: 84.06

Training accuracy: 86.42
```

In [14]:
```
classifier = LogisticRegression(penalty='none',fit_intercept=False)
classifier.fit(X_train[:63000],y_train[:63000])

train_accuracy = classifier.score(X_train[:63000],y_train[:63000])
test_accuracy = classifier.score(X_test,y_test)

print('Number of training examples: ', X_train.shape[0])
print('Number of testing examples: ', X_test.shape[0])
print('\nTesting accuracy:', format( 100*test_accuracy , '.2f') )
print('\nTraining accuracy:', format( 100*train_accuracy , '.2f') )
```

```
Number of training examples:  60000
Number of testing examples:  10000

Testing accuracy: 84.09

Training accuracy: 86.34
```

In [9]:
```
classifier = LogisticRegression(penalty='none',fit_intercept=False)
classifier.fit(X_train[:20000],y_train[:20000])

train_accuracy = classifier.score(X_train[:20000],y_train[:20000])
test_accuracy = classifier.score(X_test,y_test)

print('Number of training examples: ', X_train.shape[0])
print('Number of testing examples: ', X_test.shape[0])
print('\nTesting accuracy:', format( 100*test_accuracy , '.2f') )
print('\nTraining accuracy:', format( 100*train_accuracy , '.2f') )
```

```
Number of training examples:  60000
Number of testing examples:  10000

Testing accuracy: 83.27

Training accuracy: 87.09
```