In [1]:
```
from keras.datasets import reuters

(train_data, train_labels), (test_data, test_labels) = reuters.load_data(num_w
ords= 10000)
```

In [2]:
```
import numpy as np


def vectorize_sequences(sequences, dimensions=10000):
    results = np.zeros((len(sequences), dimensions))
    for i, sequence in enumerate(sequences):
        results[i, sequence] = 1.
    return results

x_train = vectorize_sequences(train_data)
x_test = vectorize_sequences(test_data)

from keras.utils.np_utils import to_categorical

one_hot_train_labels = to_categorical(train_labels)
one_hot_test_labels = to_categorical(test_labels)
```

In [3]:
```
from keras import models, layers

model = models.Sequential()
model.add(layers.Dense(64, activation= 'relu', input_shape= (10000,)))
model.add(layers.Dense(64, activation= 'relu'))
model.add(layers.Dense(46, activation= 'softmax'))

model.compile(optimizer= 'rmsprop',
              loss= 'categorical_crossentropy',
              metrics= ['accuracy'])

x_val = x_train[:1000]
partial_x_train = x_train[1000:]
y_val = one_hot_train_labels[:1000]
partial_y_train = one_hot_train_labels[1000:]

history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs= 20,
                    batch_size= 512,
                    validation_data= (x_val, y_val))
```

```
Epoch 1/20
16/16 [==============================] - 0s 31ms/step - loss: 2.6199 - accura
cy: 0.5549 - val_loss: 1.7187 - val_accuracy: 0.6440
Epoch 2/20
16/16 [==============================] - 0s 21ms/step - loss: 1.3858 - accura
cy: 0.7117 - val_loss: 1.2898 - val_accuracy: 0.7250
Epoch 3/20
16/16 [==============================] - 0s 18ms/step - loss: 1.0230 - accura
cy: 0.7883 - val_loss: 1.1191 - val_accuracy: 0.7650
Epoch 4/20
16/16 [==============================] - 0s 17ms/step - loss: 0.8018 - accura
cy: 0.8349 - val_loss: 1.0300 - val_accuracy: 0.7860
Epoch 5/20
16/16 [==============================] - 0s 16ms/step - loss: 0.6381 - accura
cy: 0.8696 - val_loss: 0.9732 - val_accuracy: 0.8020
Epoch 6/20
16/16 [==============================] - 0s 17ms/step - loss: 0.5074 - accura
cy: 0.8945 - val_loss: 0.9392 - val_accuracy: 0.8180
Epoch 7/20
16/16 [==============================] - 0s 15ms/step - loss: 0.4108 - accura
cy: 0.9188 - val_loss: 0.9264 - val_accuracy: 0.8100
Epoch 8/20
16/16 [==============================] - 0s 16ms/step - loss: 0.3349 - accura
cy: 0.9300 - val_loss: 0.8985 - val_accuracy: 0.8320
Epoch 9/20
16/16 [==============================] - 0s 17ms/step - loss: 0.2762 - accura
cy: 0.9416 - val_loss: 0.9502 - val_accuracy: 0.8080
Epoch 10/20
16/16 [==============================] - 0s 16ms/step - loss: 0.2333 - accura
cy: 0.9485 - val_loss: 0.9636 - val_accuracy: 0.8060
Epoch 11/20
16/16 [==============================] - 0s 14ms/step - loss: 0.2065 - accura
cy: 0.9490 - val_loss: 0.9200 - val_accuracy: 0.8130
Epoch 12/20
16/16 [==============================] - 0s 16ms/step - loss: 0.1782 - accura
cy: 0.9544 - val_loss: 0.9360 - val_accuracy: 0.8240
Epoch 13/20
16/16 [==============================] - 0s 18ms/step - loss: 0.1617 - accura
cy: 0.9544 - val_loss: 0.9871 - val_accuracy: 0.8120
Epoch 14/20
16/16 [==============================] - 0s 15ms/step - loss: 0.1488 - accura
cy: 0.9543 - val_loss: 0.9844 - val_accuracy: 0.8080
Epoch 15/20
16/16 [==============================] - 0s 15ms/step - loss: 0.1393 - accura
cy: 0.9558 - val_loss: 1.0481 - val_accuracy: 0.7990
Epoch 16/20
16/16 [==============================] - 0s 18ms/step - loss: 0.1302 - accura
cy: 0.9558 - val_loss: 1.0339 - val_accuracy: 0.8110
Epoch 17/20
16/16 [==============================] - 0s 23ms/step - loss: 0.1241 - accura
cy: 0.9549 - val_loss: 1.0677 - val_accuracy: 0.8070
Epoch 18/20
16/16 [==============================] - 0s 19ms/step - loss: 0.1194 - accura
cy: 0.9562 - val_loss: 1.0503 - val_accuracy: 0.8100
Epoch 19/20
16/16 [==============================] - 0s 14ms/step - loss: 0.1216 - accura
cy: 0.9559 - val_loss: 1.1385 - val_accuracy: 0.7910
```

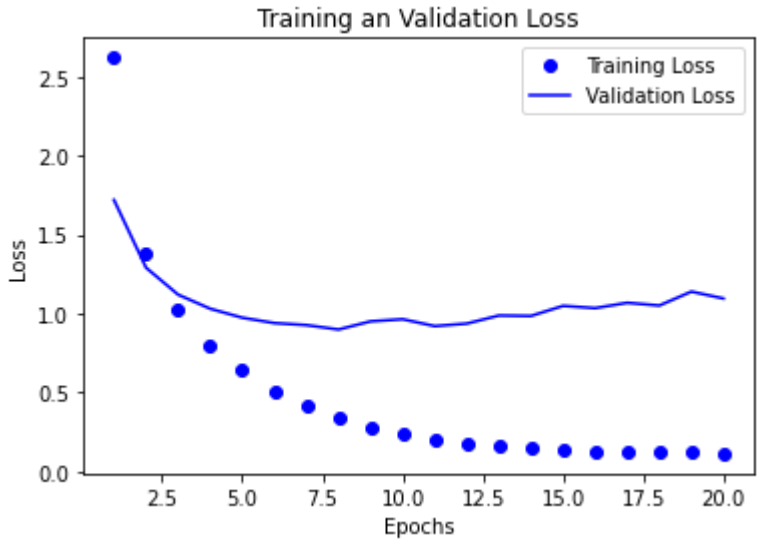
```
Epoch 20/20
16/16 [==============================] - 0s 14ms/step - loss: 0.1125 - accura
cy: 0.9572 - val_loss: 1.0948 - val_accuracy: 0.8070
```

In [4]:
```
import matplotlib.pyplot as plt

history_dict = history.history
loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
epochs = range(1, len(history_dict['accuracy']) + 1)

plt.plot(epochs, loss_values, 'bo', label= 'Training Loss')
plt.plot(epochs, val_loss_values, 'b', label= "Validation Loss")
plt.title('Training an Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
```

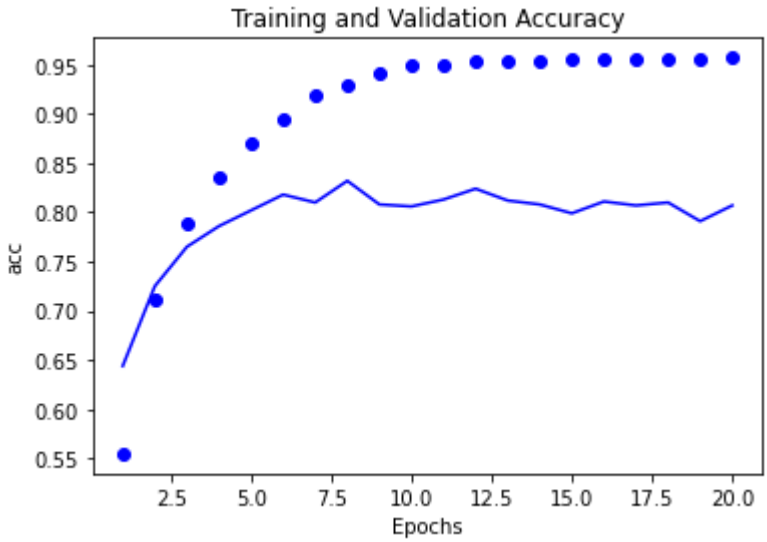
Out[4]:
```
<matplotlib.legend.Legend at 0x7f59a11e59a0>
```



In [5]:
```
acc_values = history_dict['accuracy']
val_acc_values = history_dict['val_accuracy']

plt.plot(epochs, acc_values, 'bo', label= 'Training Acc')
plt.plot(epochs, val_acc_values, 'b', label= 'Validation Acc')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('acc')
```

Out[5]:
```
Text(0, 0.5, 'acc')
```



In [10]:
```
model = models.Sequential()
model.add(layers.Dense(184, activation= 'relu', input_shape= (10000,)))
model.add(layers.Dense(184, activation= 'relu'))
model.add(layers.Dense(92, activation= 'relu'))
model.add(layers.Dense(46, activation= 'softmax'))

model.compile(optimizer= 'rmsprop',
              loss= 'categorical_crossentropy',
              metrics= ['accuracy'])

x_val = x_train[:1000]
partial_x_train = x_train[1000:]
y_val = one_hot_train_labels[:1000]
partial_y_train = one_hot_train_labels[1000:]

history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs= 9,
                    batch_size= 512,
                    validation_data= (x_val, y_val))

results = model.evaluate(x_test, one_hot_test_labels)
results
```

```
Epoch 1/9
16/16 [==============================] - 1s 35ms/step - loss: 2.1165 - accura
cy: 0.5309 - val_loss: 1.3741 - val_accuracy: 0.6720
Epoch 2/9
16/16 [==============================] - 0s 28ms/step - loss: 1.1059 - accura
cy: 0.7526 - val_loss: 1.3997 - val_accuracy: 0.6450
Epoch 3/9
16/16 [==============================] - 0s 31ms/step - loss: 0.7804 - accura
cy: 0.8218 - val_loss: 0.9643 - val_accuracy: 0.7950
Epoch 4/9
16/16 [==============================] - 0s 29ms/step - loss: 0.5568 - accura
cy: 0.8736 - val_loss: 0.9376 - val_accuracy: 0.8030
Epoch 5/9
16/16 [==============================] - 0s 27ms/step - loss: 0.4041 - accura
cy: 0.9110 - val_loss: 0.9219 - val_accuracy: 0.8090
Epoch 6/9
16/16 [==============================] - 0s 27ms/step - loss: 0.2988 - accura
cy: 0.9335 - val_loss: 0.9269 - val_accuracy: 0.8090
Epoch 7/9
16/16 [==============================] - 0s 27ms/step - loss: 0.2224 - accura
cy: 0.9480 - val_loss: 0.9450 - val_accuracy: 0.8090
Epoch 8/9
16/16 [==============================] - 0s 26ms/step - loss: 0.2097 - accura
cy: 0.9484 - val_loss: 0.9453 - val_accuracy: 0.8140
Epoch 9/9
16/16 [==============================] - 0s 28ms/step - loss: 0.1669 - accura
cy: 0.9535 - val_loss: 1.0507 - val_accuracy: 0.7940
71/71 [==============================] - 0s 3ms/step - loss: 1.2310 - accurac
y: 0.7814
```

Out[10]:
```
[1.231042742729187, 0.7813891172409058]
```

In [ ]:

In [ ]: