In [1]:

```
from keras.datasets import imdb
import numpy as np

(train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words=10000)

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

x_train = vectorize_sequences(train_data)
x_test = vectorize_sequences(test_data)
y_train = np.asarray(train_labels).astype('float32')
y_test = np.asarray(test_labels).astype('float32')
```

```
Using TensorFlow backend.
```

In [2]:

```
from keras import models, layers

original_model = models.Sequential()
original_model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
original_model.add(layers.Dense(16, activation='relu'))
original_model.add(layers.Dense(1, activation='sigmoid'))

original_model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
```

```
WARNING:tensorflow:From C:₩Users₩J₩Anaconda3₩lib₩site-packages₩tensorflow₩python₩f
ramework₩op_def_library.py:263: colocate_with (from tensorflow.python.framework.op
s) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

In [3]:

```
smaller_model = models.Sequential()
smaller_model.add(layers.Dense(6, activation='relu', input_shape=(10000,)))
smaller_model.add(layers.Dense(6, activation='relu'))
smaller_model.add(layers.Dense(1, activation='sigmoid'))

smaller_model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
```

In [4]:

```
original_hist = original_model.fit(x_train, y_train, epochs=20, batch_size=512,
                                   validation_data = (x_test, y_test))
```

```
WARNING:tensorflow:From C:₩Users₩J₩Anaconda3₩lib₩site-packages₩tensorflow₩python₩o
ps₩math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated
and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 25000 samples, validate on 25000 samples
Epoch 1/20
25000/25000 [==============================] - 8s 310us/step - loss: 0.4440 - acc:
0.8251 - val_loss: 0.3286 - val_acc: 0.8835
Epoch 2/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.2573 - acc:
0.9078 - val_loss: 0.2864 - val_acc: 0.8882
Epoch 3/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1991 - acc:
0.9292 - val_loss: 0.2821 - val_acc: 0.8891
Epoch 4/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.1666 - acc:
0.9412 - val_loss: 0.2939 - val_acc: 0.8844
Epoch 5/20
25000/25000 [==============================] - 4s 144us/step - loss: 0.1435 - acc:
0.9501 - val_loss: 0.3116 - val_acc: 0.8804
Epoch 6/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1257 - acc:
0.9558 - val_loss: 0.3483 - val_acc: 0.8721
Epoch 7/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.1104 - acc:
0.9615 - val_loss: 0.3598 - val_acc: 0.8722
Epoch 8/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0977 - acc:
0.9669 - val_loss: 0.3975 - val_acc: 0.8662
Epoch 9/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.0841 - acc:
0.9721 - val_loss: 0.4339 - val_acc: 0.8611
Epoch 10/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.0755 - acc:
0.9757 - val_loss: 0.4997 - val_acc: 0.8524
Epoch 11/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.0684 - acc:
0.9778 - val_loss: 0.4871 - val_acc: 0.8587
Epoch 12/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0563 - acc:
0.9831 - val_loss: 0.5220 - val_acc: 0.8557
Epoch 13/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0513 - acc:
0.9843 - val_loss: 0.5351 - val_acc: 0.8579
Epoch 14/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0423 - acc:
0.9881 - val_loss: 0.6093 - val_acc: 0.8501
Epoch 15/20
25000/25000 [==============================] - 4s 149us/step - loss: 0.0363 - acc:
0.9894 - val_loss: 0.6115 - val_acc: 0.8558
Epoch 16/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.0310 - acc:
0.9909 - val_loss: 0.6426 - val_acc: 0.8536
Epoch 17/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0265 - acc:
0.9926 - val_loss: 0.7693 - val_acc: 0.8417
Epoch 18/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.0208 - acc:
0.9946 - val_loss: 0.7221 - val_acc: 0.8516
Epoch 19/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0198 - acc:
0.9949 - val_loss: 0.7739 - val_acc: 0.8476
Epoch 20/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.0151 - acc:
0.9965 - val_loss: 0.9723 - val_acc: 0.8298
```

In [5]:

```
smaller_hist = smaller_model.fit(x_train, y_train, epochs=20, batch_size=512,
                                 validation_data = (x_test, y_test))
```

```
Train on 25000 samples, validate on 25000 samples
Epoch 1/20
25000/25000 [==============================] - 4s 154us/step - loss: 0.5670 - acc:
0.7410 - val_loss: 0.4849 - val_acc: 0.8256
Epoch 2/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.3924 - acc:
0.8895 - val_loss: 0.3609 - val_acc: 0.8780
Epoch 3/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2791 - acc:
0.9174 - val_loss: 0.3012 - val_acc: 0.8879
Epoch 4/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2213 - acc:
0.9278 - val_loss: 0.2823 - val_acc: 0.8900
Epoch 5/20
25000/25000 [==============================] - 4s 152us/step - loss: 0.1877 - acc:
0.9375 - val_loss: 0.2816 - val_acc: 0.8863
Epoch 6/20
25000/25000 [==============================] - 4s 148us/step - loss: 0.1649 - acc:
0.9448 - val_loss: 0.2858 - val_acc: 0.8856
Epoch 7/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1467 - acc:
0.9518 - val_loss: 0.2954 - val_acc: 0.8817
Epoch 8/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.1330 - acc:
0.9558 - val_loss: 0.3067 - val_acc: 0.8796
Epoch 9/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.1196 - acc:
0.9620 - val_loss: 0.3223 - val_acc: 0.8773
Epoch 10/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1095 - acc:
0.9648 - val_loss: 0.3358 - val_acc: 0.8752
Epoch 11/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.0988 - acc:
0.9690 - val_loss: 0.3549 - val_acc: 0.8712
Epoch 12/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0898 - acc:
0.9728 - val_loss: 0.3790 - val_acc: 0.8692
Epoch 13/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0816 - acc:
0.9759 - val_loss: 0.3900 - val_acc: 0.8676
Epoch 14/20
25000/25000 [==============================] - 4s 148us/step - loss: 0.0743 - acc:
0.9772 - val_loss: 0.4130 - val_acc: 0.8654
Epoch 15/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0675 - acc:
0.9806 - val_loss: 0.4394 - val_acc: 0.8631
Epoch 16/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0605 - acc:
0.9828 - val_loss: 0.4665 - val_acc: 0.8602
Epoch 17/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0554 - acc:
0.9840 - val_loss: 0.4742 - val_acc: 0.8618
Epoch 18/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0500 - acc:
0.9867 - val_loss: 0.5050 - val_acc: 0.8592
Epoch 19/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0456 - acc:
0.9883 - val_loss: 0.5210 - val_acc: 0.8591
Epoch 20/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.0409 - acc:
0.9896 - val_loss: 0.5714 - val_acc: 0.8536
```

In [7]:

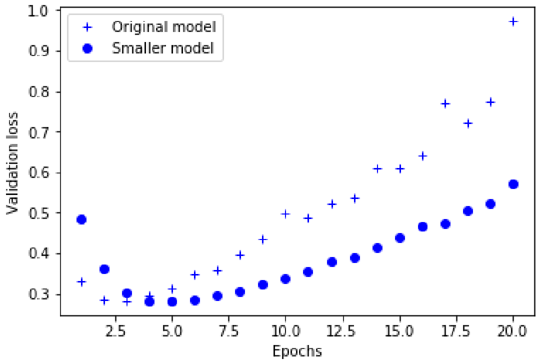
```
import matplotlib.pyplot as plt

epochs = range(1,21)
original_val_loss = original_hist.history['val_loss']
smaller_model_val_loss = smaller_hist.history['val_loss']
```

In [8]:

```
plt.plot(epochs, original_val_loss, 'b+', label='Original model')
plt.plot(epochs, smaller_model_val_loss, 'bo', label='Smaller model')
plt.xlabel('Epochs')
plt.ylabel('Validation loss')
plt.legend()

plt.show()
```



In [9]:

```
bigger_model = models.Sequential()
bigger_model.add(layers.Dense(1024, activation='relu', input_shape=(10000,)))
bigger_model.add(layers.Dense(1024, activation='relu'))
bigger_model.add(layers.Dense(1, activation='sigmoid'))

bigger_model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
```

In [16]:

```
bigger_model_hist = bigger_model.fit(x_train, y_train, epochs=20, batch_size=512,
                                     validation_data = (x_test, y_test))
```

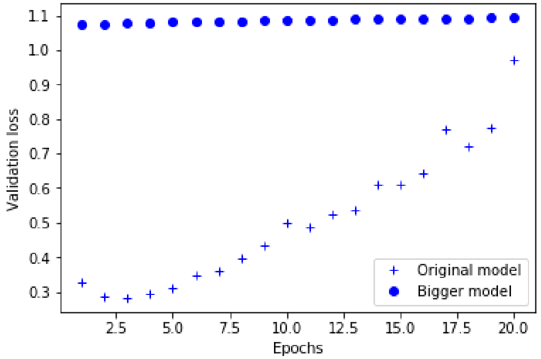
```
Train on 25000 samples, validate on 25000 samples
Epoch 1/20
25000/25000 [==============================] - 4s 162us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0740 - val_acc: 0.8813
Epoch 2/20
25000/25000 [==============================] - 4s 158us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0756 - val_acc: 0.8812
Epoch 3/20
25000/25000 [==============================] - 4s 159us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0777 - val_acc: 0.8814
Epoch 4/20
25000/25000 [==============================] - 4s 159us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0794 - val_acc: 0.8814
Epoch 5/20
25000/25000 [==============================] - 4s 159us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0808 - val_acc: 0.8813
Epoch 6/20
25000/25000 [==============================] - 4s 157us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0819 - val_acc: 0.8812
Epoch 7/20
25000/25000 [==============================] - 4s 158us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0829 - val_acc: 0.8813
Epoch 8/20
25000/25000 [==============================] - 4s 157us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0840 - val_acc: 0.8813
Epoch 9/20
25000/25000 [==============================] - 4s 160us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0849 - val_acc: 0.8812
Epoch 10/20
25000/25000 [==============================] - 4s 160us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0859 - val_acc: 0.8811
Epoch 11/20
25000/25000 [==============================] - 4s 160us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0864 - val_acc: 0.8813
Epoch 12/20
25000/25000 [==============================] - 4s 159us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0875 - val_acc: 0.8812
Epoch 13/20
25000/25000 [==============================] - 4s 159us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0882 - val_acc: 0.8812
Epoch 14/20
25000/25000 [==============================] - 4s 159us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0891 - val_acc: 0.8812
Epoch 15/20
25000/25000 [==============================] - 4s 158us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0899 - val_acc: 0.8812
Epoch 16/20
25000/25000 [==============================] - 4s 158us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0908 - val_acc: 0.8812
Epoch 17/20
25000/25000 [==============================] - 4s 158us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0911 - val_acc: 0.8812
Epoch 18/20
25000/25000 [==============================] - 4s 159us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0921 - val_acc: 0.8812
Epoch 19/20
25000/25000 [==============================] - 4s 158us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0926 - val_acc: 0.8812
Epoch 20/20
25000/25000 [==============================] - 4s 158us/step - loss: 6.4483e-04 -
acc: 1.0000 - val_loss: 1.0933 - val_acc: 0.8812
```

In [17]:

```python
bigger_model_val_loss = bigger_model_hist.history['val_loss']

plt.plot(epochs, original_val_loss, 'b+', label='Original model')
plt.plot(epochs, bigger_model_val_loss, 'bo', label='Bigger model')
plt.xlabel('Epochs')
plt.ylabel('Validation loss')
plt.legend()

plt.show()
```

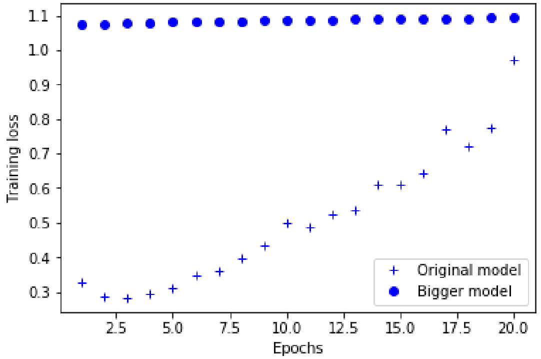


In [18]:

```python
original_train_loss = original_hist.history['loss']
bigger_model_train_loss = bigger_model_hist.history['loss']

plt.plot(epochs, original_val_loss, 'b+', label='Original model')
plt.plot(epochs, bigger_model_val_loss, 'bo', label='Bigger model')
plt.xlabel('Epochs')
plt.ylabel('Training loss')
plt.legend()

plt.show()
```



In [20]:

```
from keras import regularizers

l2_model = models.Sequential()
l2_model.add(layers.Dense(16, kernel_regularizer=regularizers.l2(0.001),
                          activation='relu', input_shape=(10000,)))
l2_model.add(layers.Dense(16, kernel_regularizer=regularizers.l2(0.001),
                          activation='relu'))
l2_model.add(layers.Dense(1, activation='sigmoid'))

l2_model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
```

In [21]:

```
l2_model_hist = l2_model.fit(x_train, y_train, epochs=20, batch_size=512,
                                   validation_data = (x_test, y_test))
```

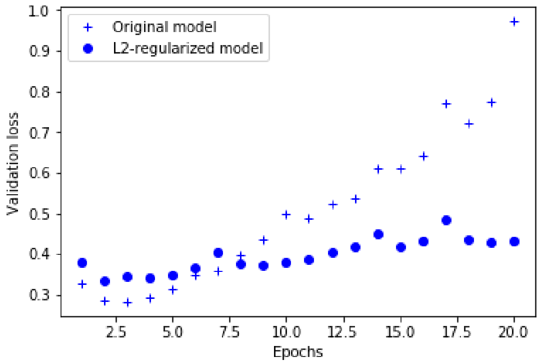
```
Train on 25000 samples, validate on 25000 samples
Epoch 1/20
25000/25000 [==============================] - 4s 165us/step - loss: 0.4844 - acc:
0.8288 - val_loss: 0.3787 - val_acc: 0.8763
Epoch 2/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.3069 - acc:
0.9058 - val_loss: 0.3335 - val_acc: 0.8861
Epoch 3/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2653 - acc:
0.9203 - val_loss: 0.3456 - val_acc: 0.8781
Epoch 4/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2418 - acc:
0.9296 - val_loss: 0.3399 - val_acc: 0.8814
Epoch 5/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2289 - acc:
0.9349 - val_loss: 0.3490 - val_acc: 0.8800
Epoch 6/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2176 - acc:
0.9398 - val_loss: 0.3647 - val_acc: 0.8760
Epoch 7/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2116 - acc:
0.9424 - val_loss: 0.4034 - val_acc: 0.8650
Epoch 8/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.2076 - acc:
0.9425 - val_loss: 0.3751 - val_acc: 0.8760
Epoch 9/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2033 - acc:
0.9472 - val_loss: 0.3720 - val_acc: 0.8770
Epoch 10/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.1955 - acc:
0.9491 - val_loss: 0.3796 - val_acc: 0.8742
Epoch 11/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1947 - acc:
0.9489 - val_loss: 0.3852 - val_acc: 0.8747
Epoch 12/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1915 - acc:
0.9506 - val_loss: 0.4024 - val_acc: 0.8698
Epoch 13/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1855 - acc:
0.9520 - val_loss: 0.4194 - val_acc: 0.8678
Epoch 14/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1799 - acc:
0.9552 - val_loss: 0.4504 - val_acc: 0.8620
Epoch 15/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.1765 - acc:
0.9573 - val_loss: 0.4195 - val_acc: 0.8668
Epoch 16/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.1751 - acc:
0.9579 - val_loss: 0.4330 - val_acc: 0.8637
Epoch 17/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1732 - acc:
0.9583 - val_loss: 0.4834 - val_acc: 0.8548
Epoch 18/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1682 - acc:
0.9605 - val_loss: 0.4346 - val_acc: 0.8624
Epoch 19/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1620 - acc:
0.9642 - val_loss: 0.4273 - val_acc: 0.8703
Epoch 20/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.1640 - acc:
0.9620 - val_loss: 0.4314 - val_acc: 0.8677
```

In [23]:

```python
l2_model_val_loss = l2_model_hist.history['val_loss']

plt.plot(epochs, original_val_loss, 'b+', label='Original model')
plt.plot(epochs, l2_model_val_loss, 'bo', label='L2-regularized model')
plt.xlabel('Epochs')
plt.ylabel('Validation loss')
plt.legend()

plt.show()
```



In [ ]:

```python
'''
L1 규제
regularizers.l1(0.001)

L1 L2 병행
regularizers.l1_l2(l1=0.001, l2=0.001)
'''
```

In [24]:

```python
l1_model = models.Sequential()
l1_model.add(layers.Dense(16, kernel_regularizer=regularizers.l1(0.0001),
                          activation='relu', input_shape=(10000,)))
l1_model.add(layers.Dense(16, kernel_regularizer=regularizers.l2(0.0001),
                          activation='relu'))
l1_model.add(layers.Dense(1, activation='sigmoid'))

l1_model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
```

In [25]:

```
l1_model_hist = l1_model.fit(x_train, y_train, epochs=20, batch_size=512,
                             validation_data = (x_test, y_test))
```

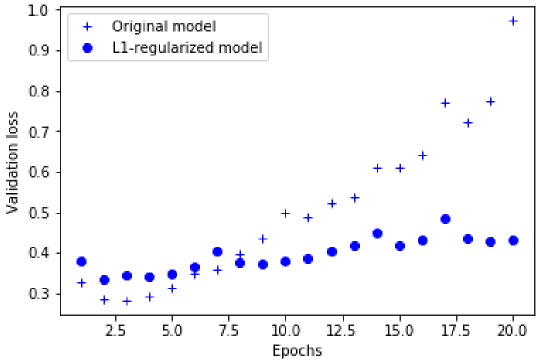
```
Train on 25000 samples, validate on 25000 samples
Epoch 1/20
25000/25000 [==============================] - 4s 163us/step - loss: 0.5466 - acc:
0.8162 - val_loss: 0.4430 - val_acc: 0.8526
Epoch 2/20
25000/25000 [==============================] - 4s 144us/step - loss: 0.3646 - acc:
0.8952 - val_loss: 0.3756 - val_acc: 0.8836
Epoch 3/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.3273 - acc:
0.9064 - val_loss: 0.3729 - val_acc: 0.8821
Epoch 4/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.3149 - acc:
0.9092 - val_loss: 0.3584 - val_acc: 0.8875
Epoch 5/20
25000/25000 [==============================] - 4s 144us/step - loss: 0.3019 - acc:
0.9150 - val_loss: 0.3782 - val_acc: 0.8782
Epoch 6/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2960 - acc:
0.9183 - val_loss: 0.3675 - val_acc: 0.8852
Epoch 7/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2880 - acc:
0.9206 - val_loss: 0.3672 - val_acc: 0.8848
Epoch 8/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2874 - acc:
0.9200 - val_loss: 0.3726 - val_acc: 0.8834
Epoch 9/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2792 - acc:
0.9257 - val_loss: 0.3853 - val_acc: 0.8799
Epoch 10/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2730 - acc:
0.9276 - val_loss: 0.3778 - val_acc: 0.8830
Epoch 11/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2696 - acc:
0.9279 - val_loss: 0.3779 - val_acc: 0.8814
Epoch 12/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2643 - acc:
0.9323 - val_loss: 0.3862 - val_acc: 0.8789
Epoch 13/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2581 - acc:
0.9334 - val_loss: 0.4063 - val_acc: 0.8688
Epoch 14/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2526 - acc:
0.9358 - val_loss: 0.3857 - val_acc: 0.8808
Epoch 15/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2467 - acc:
0.9390 - val_loss: 0.4058 - val_acc: 0.8736
Epoch 16/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2393 - acc:
0.9411 - val_loss: 0.4041 - val_acc: 0.8751
Epoch 17/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2379 - acc:
0.9417 - val_loss: 0.4034 - val_acc: 0.8760
Epoch 18/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2310 - acc:
0.9453 - val_loss: 0.4666 - val_acc: 0.8551
Epoch 19/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2287 - acc:
0.9437 - val_loss: 0.4176 - val_acc: 0.8745
Epoch 20/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2250 - acc:
0.9476 - val_loss: 0.4191 - val_acc: 0.8736
```

In [26]:

```
l1_model_val_loss = l1_model_hist.history['val_loss']

plt.plot(epochs, original_val_loss, 'b+', label='Original model')
plt.plot(epochs, l2_model_val_loss, 'bo', label='L1-regularized model')
plt.xlabel('Epochs')
plt.ylabel('Validation loss')
plt.legend()

plt.show()
```



In [27]:

```
l1l2_model = models.Sequential()
l1l2_model.add(layers.Dense(16, kernel_regularizer=regularizers.l1_l2(l1=0.0001,
                                                                      l2=0.0001),
                            activation='relu', input_shape=(10000,)))
l1l2_model.add(layers.Dense(16, kernel_regularizer=regularizers.l1_l2(l1=0.0001,
                                                                      l2=0.0001),
                            activation='relu'))
l1l2_model.add(layers.Dense(1, activation='sigmoid'))

l1l2_model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
```

In [28]:

```
l1l2_model_hist = l1l2_model.fit(x_train, y_train, epochs=20, batch_size=512,
                                   validation_data = (x_test, y_test))
```

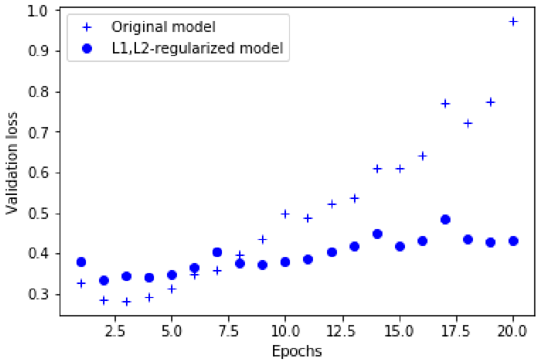
```
Train on 25000 samples, validate on 25000 samples
Epoch 1/20
25000/25000 [==============================] - 4s 168us/step - loss: 0.5664 - acc:
0.8176 - val_loss: 0.4531 - val_acc: 0.8545
Epoch 2/20
25000/25000 [==============================] - 4s 148us/step - loss: 0.3821 - acc:
0.8925 - val_loss: 0.3792 - val_acc: 0.8850
Epoch 3/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.3408 - acc:
0.9028 - val_loss: 0.3641 - val_acc: 0.8891
Epoch 4/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.3218 - acc:
0.9090 - val_loss: 0.3616 - val_acc: 0.8881
Epoch 5/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.3134 - acc:
0.9110 - val_loss: 0.3694 - val_acc: 0.8851
Epoch 6/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.3064 - acc:
0.9141 - val_loss: 0.3666 - val_acc: 0.8850
Epoch 7/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.3021 - acc:
0.9168 - val_loss: 0.3728 - val_acc: 0.8836
Epoch 8/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2952 - acc:
0.9210 - val_loss: 0.4185 - val_acc: 0.8630
Epoch 9/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.2929 - acc:
0.9199 - val_loss: 0.3837 - val_acc: 0.8808
Epoch 10/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2906 - acc:
0.9229 - val_loss: 0.4410 - val_acc: 0.8616
Epoch 11/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2887 - acc:
0.9216 - val_loss: 0.3815 - val_acc: 0.8828
Epoch 12/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2826 - acc:
0.9266 - val_loss: 0.3887 - val_acc: 0.8806
Epoch 13/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2821 - acc:
0.9275 - val_loss: 0.3956 - val_acc: 0.8794
Epoch 14/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2785 - acc:
0.9282 - val_loss: 0.3890 - val_acc: 0.8799
Epoch 15/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2794 - acc:
0.9275 - val_loss: 0.3916 - val_acc: 0.8805
Epoch 16/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2740 - acc:
0.9298 - val_loss: 0.4177 - val_acc: 0.8712
Epoch 17/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2717 - acc:
0.9313 - val_loss: 0.3956 - val_acc: 0.8776
Epoch 18/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2643 - acc:
0.9343 - val_loss: 0.4064 - val_acc: 0.8765
Epoch 19/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2663 - acc:
0.9321 - val_loss: 0.4010 - val_acc: 0.8767
Epoch 20/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2577 - acc:
0.9373 - val_loss: 0.4867 - val_acc: 0.8456
```

In [29]:

```
l1l2_model_val_loss = l1l2_model_hist.history['val_loss']

plt.plot(epochs, original_val_loss, 'b+', label='Original model')
plt.plot(epochs, l2_model_val_loss, 'bo', label='L1,L2-regularized model')
plt.xlabel('Epochs')
plt.ylabel('Validation loss')
plt.legend()

plt.show()
```



In [ ]:

In [ ]:

In [ ]: