In [1]:

```
! wget --header="Host: archive.ics.uci.edu" --header="User-Agent: Mozil
la/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
cko) Chrome/72.0.3626.81 Safari/537.36" --header="Accept: text/html,app
lication/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.
8" "https://archive.ics.uci.edu/ml/machine-learning-databases/00463/XOR
_Arbiter_PUFs.zip" -O "XOR_Arbiter_PUFs.zip" -c
```

```
--2019-02-15 00:52:00--  https://archive.ics.uci.edu/ml/machine-learnin
g-databases/00463/XOR_Arbiter_PUFs.zip
Resolving archive.ics.uci.edu (archive.ics.uci.edu)... 128.195.10.249
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.249
|:443... connected.
HTTP request sent, awaiting response... 416 Requested Range Not Satisfi
able

    The file is already fully retrieved; nothing to do.
```

In [2]:

```
!unzip 'XOR_Arbiter_PUFs.zip'
```

```
Archive:  XOR_Arbiter_PUFs.zip
replace XOR_Arbiter_PUFs/.DS_Store? [y]es, [n]o, [A]ll, [N]one, [r]enam
e: All
  inflating: XOR_Arbiter_PUFs/.DS_Store
  inflating: XOR_Arbiter_PUFs/5xor_128bit/.DS_Store
  inflating: XOR_Arbiter_PUFs/5xor_128bit/README.txt
  inflating: XOR_Arbiter_PUFs/5xor_128bit/test_5xor_128dim.csv
  inflating: XOR_Arbiter_PUFs/5xor_128bit/train_5xor_128dim.csv
  inflating: XOR_Arbiter_PUFs/6xor_64bit/.DS_Store
  inflating: XOR_Arbiter_PUFs/6xor_64bit/README.txt
  inflating: XOR_Arbiter_PUFs/6xor_64bit/test_6xor_64dim.csv
  inflating: XOR_Arbiter_PUFs/6xor_64bit/train_6xor_64dim.csv
  inflating: __MACOSX/XOR_Arbiter_PUFs/._.DS_Store
  inflating: __MACOSX/XOR_Arbiter_PUFs/5xor_128bit/._.DS_Store
  inflating: __MACOSX/XOR_Arbiter_PUFs/5xor_128bit/._README.txt
  inflating: __MACOSX/XOR_Arbiter_PUFs/5xor_128bit/._test_5xor_128dim.c
```

```
sv
  inflating: __MACOSX/XOR_Arbiter_PUFs/6xor_64bit/._.DS_Store
  inflating: __MACOSX/XOR_Arbiter_PUFs/6xor_64bit/._README.txt
```

In [3]:
```
import pandas as pd
X_train = pd.read_csv(r'./XOR_Arbiter_PUFs/5xor_128bit/train_5xor_128di
m.csv',header = None).sample(1000000)
X_train.head()
```

Out[3]:

	0	1	2	3	4	5	6	7	8	9	...	119	120	121	122	123	124	125	126	127	1
3247711	-1	-1	-1	-1	1	1	-1	1	1	-1	...	-1	-1	-1	1	1	1	-1	-1	1	1
1849253	-1	1	1	1	1	1	-1	-1	1	1	...	1	1	-1	1	1	1	1	-1	1	1
1905889	-1	-1	1	-1	-1	-1	-1	1	1	-1	...	-1	-1	-1	1	-1	1	-1	1	1	1
2313636	-1	1	-1	-1	1	-1	-1	1	1	1	...	-1	1	-1	-1	-1	1	1	-1	1	-1
1793696	-1	1	1	-1	1	-1	-1	1	1	-1	...	1	1	-1	1	-1	-1	1	1	-1	-1

5 rows × 129 columns

In [4]:
```
from keras.utils import np_utils
from keras.initializers import he_normal
import seaborn as sns
from keras.models import Sequential
from keras.layers import Dense , Activation
from keras import optimizers
```

```
Using TensorFlow backend.
```

In [5]:
```
X_train.shape
```

Out[5]:
```
(1000000, 129)
```

In [6]:
```
X_train.describe()
```

Out[6]:

	0	1	2	3	4
count	1000000.00000	1000000.000000	1000000.000000	1000000.00000	1000000.000000
mean	-0.00071	0.001124	-0.000848	0.00092	-0.001446
std	1.00000	1.000000	1.000000	1.00000	0.999999
min	-1.00000	-1.000000	-1.000000	-1.00000	-1.000000
25%	-1.00000	-1.000000	-1.000000	-1.00000	-1.000000
50%	-1.00000	1.000000	-1.000000	1.00000	-1.000000
75%	1.00000	1.000000	1.000000	1.00000	1.000000
max	1.00000	1.000000	1.000000	1.00000	1.000000

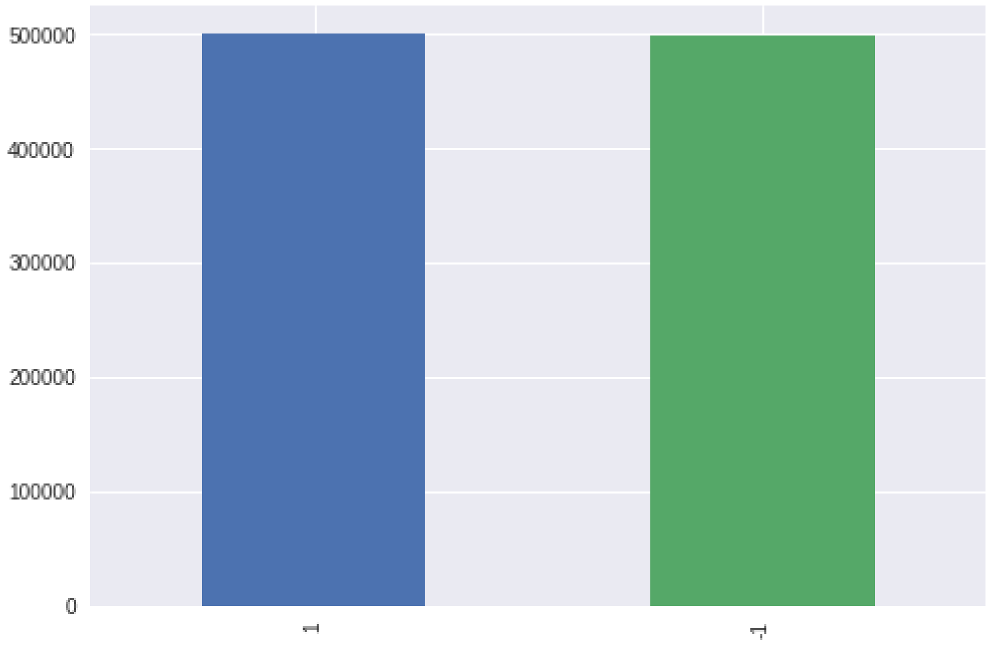
8 rows × 129 columns

In [7]:

```
distrb = X_train.iloc[:,128].value_counts()
import matplotlib.pyplot as plt
distrb.plot(kind = 'bar')
```

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa3f3c266a0>
```



In [8]:

```
import numpy as np
X_train.isnull().values.any()
```

Out[8]:

```
False
```

In [9]:

```
Y_train = X_train[[128]]
X_train.drop([128],axis = 1,inplace = True)
X_train.shape
```

Out[9]:

```
(1000000, 128)
```

In [0]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
X_train , X_test ,Y_train ,Y_test = train_test_split(X_train,Y_train,te
st_size=0.3)
```

In [11]:

```
import numpy as np
X_test.isnull().values.any()
```

Out[11]: False

In [0]:
```
y_train = np_utils.to_categorical(Y_train, 2)
y_test = np_utils.to_categorical(Y_test, 2)
```

In [0]:
```
from keras.layers.normalization import BatchNormalization
from keras.layers import Dropout
from keras.layers.merge import concatenate
from keras.utils import plot_model
from keras.layers import Input
from keras.models import Model
```

In [0]:
```
nepoch = 30
outlayer = 2
batch_size = 1000
```

In [32]:
```
input_layer = Input(shape = (128,))

out1 = Dense(32,activation = 'relu')(input_layer)
out1 = Dropout(0.5)(out1)
out1 = BatchNormalization()(out1)

out2 = Dense(32,activation = 'relu')(input_layer)
out2 = Dropout(0.5)(out2)
out2 = BatchNormalization()(out2)

out3 = Dense(32,activation = 'relu')(input_layer)
out3 = Dropout(0.5)(out3)
out3 = BatchNormalization()(out3)

merge = concatenate([out1,out2,out3])


output = Dense(2,activation = 'sigmoid')(merge)

model = Model(inputs=input_layer, outputs=output)
# summarize layers
```

```
print(model.summary())

# plot graph
plot_model(model, to_file='MODEL.png')

adam = optimizers.Adam(lr = 0.001)
model.compile(loss='binary_crossentropy', optimizer = adam, metrics=['accuracy'])
```

```
_______________________________________________________________________
___________________________
Layer (type)                    Output Shape         Param #     Connec
ted to
=======================================================================
===========================
input_8 (InputLayer)            (None, 128)          0

_______________________________________________________________________
___________________________
dense_29 (Dense)                (None, 32)           4128        input_
8[0][0]
_______________________________________________________________________
___________________________
dense_30 (Dense)                (None, 32)           4128        input_
8[0][0]
_______________________________________________________________________
___________________________
dense_31 (Dense)                (None, 32)           4128        input_
8[0][0]
_______________________________________________________________________
___________________________
dropout_18 (Dropout)            (None, 32)           0           dense_
29[0][0]
_______________________________________________________________________
___________________________
dropout_19 (Dropout)            (None, 32)           0           dense_
30[0][0]
_______________________________________________________________________
___________________________
dropout_20 (Dropout)            (None, 32)           0           dense_
```

```
31[0][0]
_______________________________________________________________________
___________________________
batch_normalization_10 (BatchNo (None, 32)           128         dropou
t_18[0][0]
_______________________________________________________________________
___________________________
batch_normalization_11 (BatchNo (None, 32)           128         dropou
t_19[0][0]
_______________________________________________________________________
___________________________
batch_normalization_12 (BatchNo (None, 32)           128         dropou
t_20[0][0]
_______________________________________________________________________
___________________________
concatenate_6 (Concatenate)     (None, 96)           0           batch_
normalization_10[0][0]
                                                                 batch_
normalization_11[0][0]
                                                                 batch_
normalization_12[0][0]
_______________________________________________________________________
___________________________
dense_32 (Dense)                (None, 2)            194         concat
enate_6[0][0]
=======================================================================
===========================
Total params: 12,962
Trainable params: 12,770
Non-trainable params: 192
_______________________________________________________________________
___________________________
None
```

In [33]:

```
hist = model.fit(X_train, y_train, epochs=nepoch, batch_size=batch_size
,validation_data = (X_test,y_test))
# Final evaluation of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```
Train on 700000 samples, validate on 300000 samples
Epoch 1/30
700000/700000 [==============================] - 14s 20us/step - loss:
0.1213 - acc: 0.9481 - val_loss: 9.7079e-04 - val_acc: 1.0000
Epoch 2/30
700000/700000 [==============================] - 12s 17us/step - loss:
0.0012 - acc: 1.0000 - val_loss: 1.9538e-04 - val_acc: 1.0000
Epoch 3/30
700000/700000 [==============================] - 12s 17us/step - loss:
3.6302e-04 - acc: 1.0000 - val_loss: 7.6580e-05 - val_acc: 1.0000
Epoch 4/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.6382e-04 - acc: 1.0000 - val_loss: 3.7571e-05 - val_acc: 1.0000
Epoch 5/30
700000/700000 [==============================] - 12s 17us/step - loss:
8.6901e-05 - acc: 1.0000 - val_loss: 2.0992e-05 - val_acc: 1.0000
Epoch 6/30
700000/700000 [==============================] - 12s 17us/step - loss:
5.0414e-05 - acc: 1.0000 - val_loss: 1.2463e-05 - val_acc: 1.0000
Epoch 7/30
700000/700000 [==============================] - 12s 17us/step - loss:
3.0994e-05 - acc: 1.0000 - val_loss: 7.7399e-06 - val_acc: 1.0000
Epoch 8/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.9611e-05 - acc: 1.0000 - val_loss: 4.9111e-06 - val_acc: 1.0000
Epoch 9/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.2673e-05 - acc: 1.0000 - val_loss: 3.2129e-06 - val_acc: 1.0000
Epoch 10/30
700000/700000 [==============================] - 12s 17us/step - loss:
8.2530e-06 - acc: 1.0000 - val_loss: 2.1084e-06 - val_acc: 1.0000
Epoch 11/30
700000/700000 [==============================] - 12s 17us/step - loss:
5.4745e-06 - acc: 1.0000 - val_loss: 1.4155e-06 - val_acc: 1.0000
Epoch 12/30
700000/700000 [==============================] - 12s 17us/step - loss:
3.6970e-06 - acc: 1.0000 - val_loss: 9.4335e-07 - val_acc: 1.0000

Epoch 13/30
700000/700000 [==============================] - 12s 17us/step - loss:
```

```
2.4792e-06 - acc: 1.0000 - val_loss: 6.4181e-07 - val_acc: 1.0000
Epoch 14/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.6760e-06 - acc: 1.0000 - val_loss: 4.3449e-07 - val_acc: 1.0000
Epoch 15/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.1570e-06 - acc: 1.0000 - val_loss: 3.0162e-07 - val_acc: 1.0000
Epoch 16/30
700000/700000 [==============================] - 12s 17us/step - loss:
8.0217e-07 - acc: 1.0000 - val_loss: 2.1292e-07 - val_acc: 1.0000
Epoch 17/30
700000/700000 [==============================] - 12s 17us/step - loss:
5.5896e-07 - acc: 1.0000 - val_loss: 1.5990e-07 - val_acc: 1.0000
Epoch 18/30
700000/700000 [==============================] - 12s 17us/step - loss:
3.9986e-07 - acc: 1.0000 - val_loss: 1.3138e-07 - val_acc: 1.0000
Epoch 19/30
700000/700000 [==============================] - 12s 17us/step - loss:
2.9693e-07 - acc: 1.0000 - val_loss: 1.1753e-07 - val_acc: 1.0000
Epoch 20/30
700000/700000 [==============================] - 12s 17us/step - loss:
2.2805e-07 - acc: 1.0000 - val_loss: 1.1225e-07 - val_acc: 1.0000
Epoch 21/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.8341e-07 - acc: 1.0000 - val_loss: 1.1041e-07 - val_acc: 1.0000
Epoch 22/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.5607e-07 - acc: 1.0000 - val_loss: 1.0985e-07 - val_acc: 1.0000
Epoch 23/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.3829e-07 - acc: 1.0000 - val_loss: 1.0969e-07 - val_acc: 1.0000
Epoch 24/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.2742e-07 - acc: 1.0000 - val_loss: 1.0964e-07 - val_acc: 1.0000
Epoch 25/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.2048e-07 - acc: 1.0000 - val_loss: 1.0962e-07 - val_acc: 1.0000

Epoch 26/30
700000/700000 [==============================] - 12s 17us/step - loss:
```

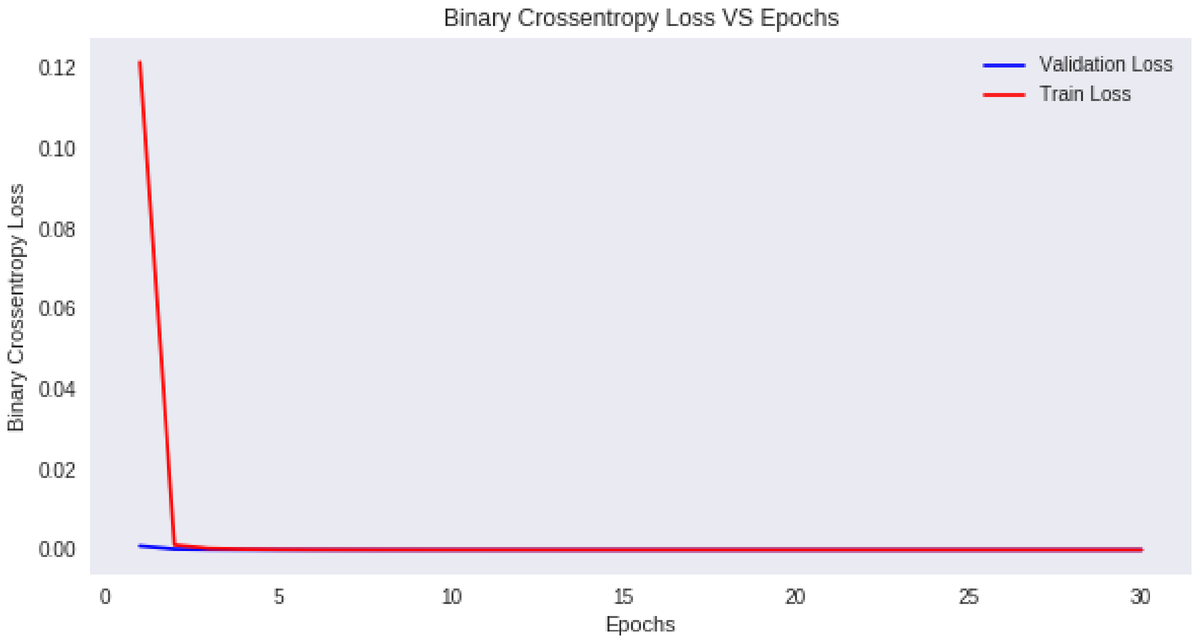
```
1.1620e-07 - acc: 1.0000 - val_loss: 1.0961e-07 - val_acc: 1.0000
Epoch 27/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.1378e-07 - acc: 1.0000 - val_loss: 1.0961e-07 - val_acc: 1.0000
Epoch 28/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.1233e-07 - acc: 1.0000 - val_loss: 1.0961e-07 - val_acc: 1.0000
Epoch 29/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.1141e-07 - acc: 1.0000 - val_loss: 1.0960e-07 - val_acc: 1.0000
Epoch 30/30
700000/700000 [==============================] - 12s 17us/step - loss:
1.1076e-07 - acc: 1.0000 - val_loss: 1.0960e-07 - val_acc: 1.0000
Accuracy: 100.00%
```

In [0]:

```
def plt_dynamic(x, vy, ty):
  plt.figure(figsize=(10,5))
  plt.plot(x, vy, 'b', label="Validation Loss")
  plt.plot(x, ty, 'r', label="Train Loss")
  plt.xlabel('Epochs')
  plt.ylabel('Binary Crossentropy Loss')
  plt.title('\nBinary Crossentropy Loss VS Epochs')
  plt.legend()
  plt.grid()
  plt.show()
```

In [35]:

```
import matplotlib.pyplot as plt
x = list(range(1,31))
vy = hist.history['val_loss']
ty = hist.history['loss']
plt_dynamic(x, vy, ty)
```


Binary Crossentropy Loss VS Epochs
Validation Loss
Train Loss
Binary Crossentropy Loss
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
25
30
Epochs