DataAugmentationExample

May 22, 2018

1 Data Preprocessing and Augmentation

```
* we Augment our data via random transformation so that our model does not see any image twice
* This helps in preventing Overfitting
```

```
In [1]: import tensorflow as tf
```

```
C:\Users\jsidd\Anaconda3\lib\site-packages\h5py\__init__.py:34: FutureWarning: Conversion of t
  from ._conv import register_converters as _register_converters
```

The following class allows us to configure transformation and normalization operation on our image data

```
In [2]: from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array,
```

```
Using TensorFlow backend.
```

```
In [3]: # Keep on playing with the Values to understand the augmentation better
        datagen = ImageDataGenerator(
                rotation_range=40,      # it is used to randomly rotate pictures
                width_shift_range=0.2,  # it is used to translate pictures horizontally
                height_shift_range=0.2, # it is used to translate pictures vertically
                rescale=1./255,         # value by which we multiply our data
                shear_range=0.8,        # used to apply shearing transformation
                zoom_range=0.2,         # randomly zoom inside the pictures
                horizontal_flip=True,   # randomly flips half of the images horizontally
                vertical_flip=True,     # randomly flips half of the images vertically
                fill_mode='nearest')    # used to fill newly created pixels which can appear a
```

```
In [4]: img = load_img('dataset/single_prediction/testDog.jpg')
        x = img_to_array(img)  # this is a Numpy array with shape (3, 150, 150)
        x = x.reshape((1,) + x.shape)  # new shape (1, 3, 150, 150)

        # the .flow() generates batches of randomly transformed images
        i = 0
        for batch in datagen.flow(x, batch_size=1,
```

```
                          save_to_dir='preview', save_prefix='cat', save_format='jpeg'):
    i += 1
    if i > 20:
        break  # otherwise the generator would loop indefinitely

# the results are stored in the "preview/" directory
```