**Dogs and Cats**

*Architecture Design*

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Input | Output | Filter |
| Conv\_1 | (?, 64, 64, 3) | (?, 64, 64, 32) | 3\*3 |
| Pool\_1 | (?, 64, 64, 32) | (?, 32, 32, 32) | 2\*2 |
| Conv\_2 | (?, 32, 32, 32) | (?, 32, 32, 64) | 3\*3 |
| Conv\_3 | (?, 32, 32, 64) | (?, 32, 32, 64) | 3\*3 |
| Pool\_2 | (?, 32, 32, 64) | (?, 16, 16, 64) | 2\*2 |
| FA\_1 | (?, 16, 16, 64) | (?, 512) | N/A |
| Dropout = 0.5 | | | |
| FA\_2 | (?, 512) | (?, 2) | N/A |

test\_size = 30 n\_epoch = 25s

train\_size = 120 batch\_size = 10

total\_size = 150 learning\_rate = 0.0005

Iteration Count = 10 | Loss = 0.69 | Accuracy = 100 %

*Data/Image augmentation*techniques are used to artificially expand the dataset. Few of the popular augmentations techniques people use are grayscales, horizontal flips, vertical flips, random crops, color jitters, translations, rotations, and much more. By applying a couple of these transformations to training data, we can easily double or triple the number of training examples.

*Image Preprocessing*

* Used as an argument of input\_data.
* Defined pre-processing methods is applied both at training and testing time.
* Functions
  + add\_featurewise\_zero\_center()
    - Zero center every sample with specified mean
  + add\_featurewise\_stdnorm()
    - Scale each sample by the specified standard deviation.
    - If no std specified, std is evaluated over all samples data.

*Image Augmentation*

* Applied only at training time
* Similar to Image Preprocessing
* Functions
  + add\_random\_flip\_leftright
    - Randomly flip an image
  + add\_random\_rotation
    - Randomly rotate an image by a random angle within a range

*Results*

