Data Augmentation (M1-L3)

Augmentation → By creating new samples

train_X => Every alternate 1200
train_Y => (∵ dataset is very huge).

we will use nearest neighbours

Augmentation 1: Rotation.
original image: captures notion of 8
rotating it ±10 also remain 8
but rotating it 180 makes it totally different.

∴ Reasonable constraints

Prev tutorial: text case: Frequencies
Image case: # of holes, pixels.

Taking entire image as vector / feature.

Angle constraints: multiple samples
with various angles.

Let max angle: 60, no. of augmentations = 5

iterating through various angles.

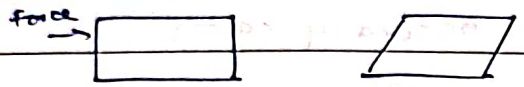
If NO ROTATION;
64.72% (∵ we are not adding new samples)

TREND: Increases upto 50° and then decreases again

Augmentation 2: Shear

assume we have a rectangle, uneven layers put force on the top most layer.

∴ Transformation of object (quantified by shear constraint)



Shear fn → SKlearn

or shear must be quantified numerically

takes the sample, how much we want to shear, then returns sheared transformation.

H.W try 0.2, 0.4, 0.6 top layer moves a little towards left

0 Shear ⇒ Original image, should give the same accuracy rate as given prev (helps with sanity check)

rotate then shear
or
Shear then rotate
} commutative.

2-D image: Angle constraint → 0-60
Shear const → 0 - 1.6

Hyperparameter search: grid search

1 value of angle constr $\xrightarrow{\text{maps to}}$ 1 value of Shear constraint

loop through all hyper-parameters

pass it NN fn
to check accuracy of the image.
(n x m)

More the yellow → Better Accuracy rates.

Combination of both - Shear & rotation