

## Assignment 4 Report

Name: Manish Tanwar

Entry Number: 2016CS10363

**Data Generation (For every part)**

The datapoints(image sequences) are generated with following procedure:

- Chose every positive example with probability  $4/5$ .
- Negative examples are chosen with probability  $1/20$ .
- Each of the  $\binom{6}{2}$  are chosen with probability  $2/3$ .
- Using these probabilities the sequences generated are in following ratio: Positive vs. Neagative = 2 : 5

**Part 1: Fixed Algorithms:****(A) PCA + SVM:**

## 1. Libraries Used:

- Scikit-learn's PCA and Incremental PCA.
- Scikit-learn's SVM (SVC)

## 2. Preprocessing:

- Converted images to greyscale.
- Pixels values are normalised to  $[0, 1]$  by dividing by 255.
- Used Incremental PCA to transform down dimension to 50.
- For Incremental PCA : Batch Size = 100
- Variance Achieved = 82%

## 3. Implementation Details:

- Stacked five frames of size (50) to get (250) as each data-point input.

4. **Linear Kernel:**

- Trained Linear Kernel on 30000 examples for 25000 iterations.
- On Validation data : F-score = 0.29

5. **Gaussian Kernel:**

- Trained Linear Kernel on 30000 examples for 25000 iterations.
- On Validation data : F-score = 0.08

## (B) Convolutional Neural Network (CNN):

1. Libraries Used: **keras**
2. Preprocessing:
  - Pixels values are normalised to  $[0, 1]$  by dividing by 255.
  - Cropped the image from (210,160,3) to (183,154,3) in the following manner:

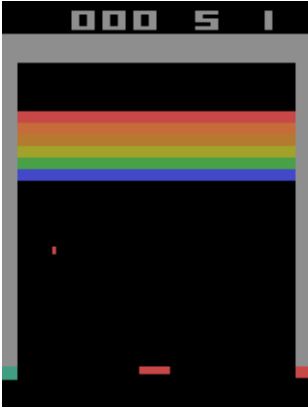


Figure 1: Original Image

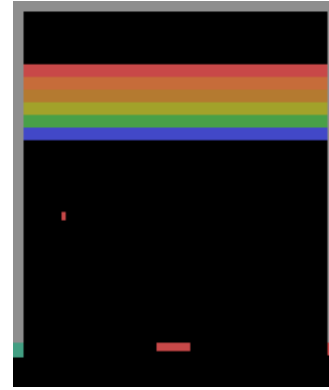


Figure 2: Cropped Image

3. Architecture Details:
  - Stacked five frames of size (183,154,3) to get (183,154,15) as each data-point input.
  - Architecture specified in the problem statement is used.
  - Optimizer = 'Adam'
  - Learning Rate = 0.001
  - Batch Size = 512
  - Class Weights =  $\{1, 7/4\}$  to get equal positive and negative examples.
4. Results on Best Model:
  - Half of the provided Validation data used as test data and other half as validation data.
  - Validation Data:
    - F1-score = 0.49
    - Accuracy = 95.67%
  - Testing Data:
    - F1-score = 0.56
    - Accuracy = 96.34%

## Part 2: Competition Part:

1. Libraries Used: **keras**
2. Preprocessing:
  - Converted images to greyscale.
  - Pixels values are normalised to  $[0, 1]$  by dividing by 255.

- Cropped the image from size (210,160,3) to (162,154,1) in the following manner:

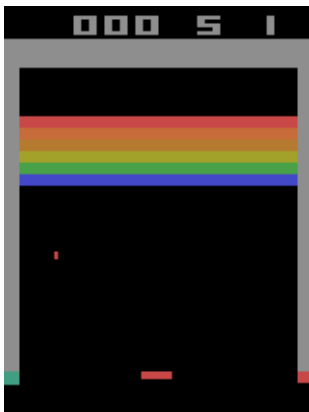


Figure 3: Original Image

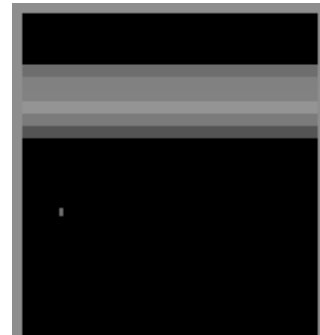


Figure 4: Cropped Image

### 3. Architecture Details:

- Stacked five frames of (162,154,1) to get (162,154,5) as each data-point input.
- Architecture specified in the problem statement is used with Dropout(0.35) after the Dense layer.
- Optimizer = 'Adam'
- Learning Rate = 0.001
- Batch Size = 512
- Class Weights =  $\{1, 7/4\}$  to get equal positive and negative examples.
- I trained three different models and predicted the value predicted by the majority of the models.

### 4. Results on Best Model:

- Training Data:
  - F-score = 0.9985
  - Accuracy = 99.92%
- Validation Data:
  - F-score = 0.5726
  - Accuracy = 97.09%
- Testing Data(on kaggle):
  - F-score = **0.5945**