GANBoost for Imbalanced Image datasets

Koh Liang Ping

1. Problem Statement

- In important classification problems, like cancer detection, datasets can be imbalanced
- Imbalanced can lead to poor classification on minority classes
- Current methods to address class imbalance are more suited for non-image problems
- Contribution: algorithm based on GANs + Boosting to address the class imbalance

2. Dataset and Pre-processing

- CIFAR10 consists of 60,000 images, in 10 categories
- Train-test split of 50,000 - 10,000
- Truck images are intentionally undersampled, only 500 images of trucks for training

3. Description of Alternative Methods

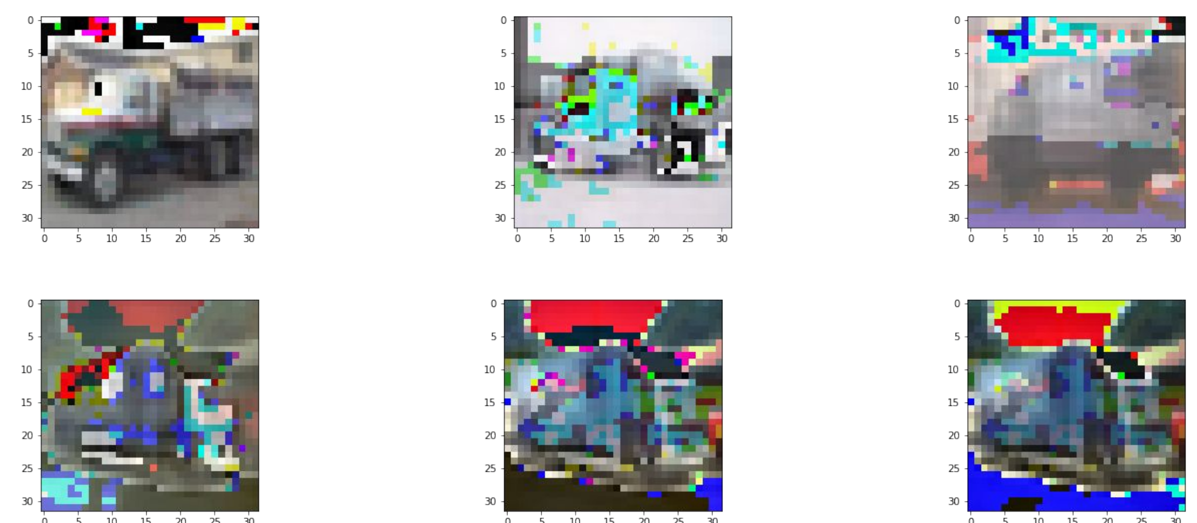
- The following prior art baseline methods are evaluated:
  - Oversampling: Sampling with replacement from minority class to rebalance the dataset
  - Synthetic Minority Oversampling Technique (SMOTE): Creates new data points as random linear combinations of minority training examples
  - Adaptive synthesis (ADASYN): Certain minority class examples are difficult to separate from the majority class. We apply SMOTE to these examples, synthesizing new examples from 'difficult regions of the data'

4. Baseline Method Evaluation

- Train a classifier for 100 epochs on the following training sets, and evaluate the test accuracy and confusion matrix:
  - With all 50,000 training samples
  - Imbalanced with 45,500 training examples: 4,500 removed for trucks
  - Oversampling: imbalanced with 45,500 training examples, plus 4,500 generated by sampled from the minority class
  - SMOTE: Imbalanced with 45,500 training examples, plus 4,500 synthesized examples
  - ADASYN: Imbalanced with 45,500 training examples, plus 4,500 synthesized by SMOTE

Model	Test Accuracy	Truck Class Recall
Full Data	77.5%	85%
Undersampled Data	76.1%	56%
Oversampled Data	66.6%	62%
SMOTE Data	75.7%	51%
ADASYN Data	76.6%	56%

- Performance worsens for minority class
- Oversampling improves recall, but worsens accuracy. Likely overfitting to minority
- SMOTE and ADASYN do not worsen accuracy overall, but do not improve minority class recall. Image examples:



- Linear combinations of image examples produce poor output with a lot of noise. It is an unnatural operation for image data and does not improve the error rate.

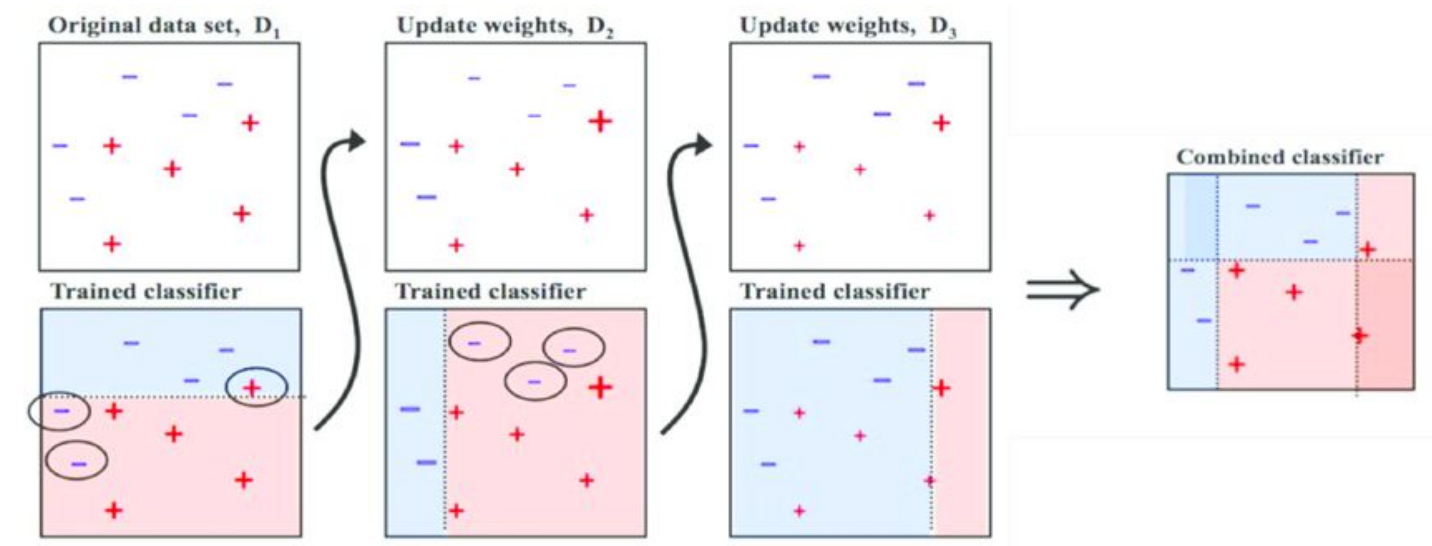
6. GAN

- We train a generator to generate images from the minority class. These are the images that our GAN produces:

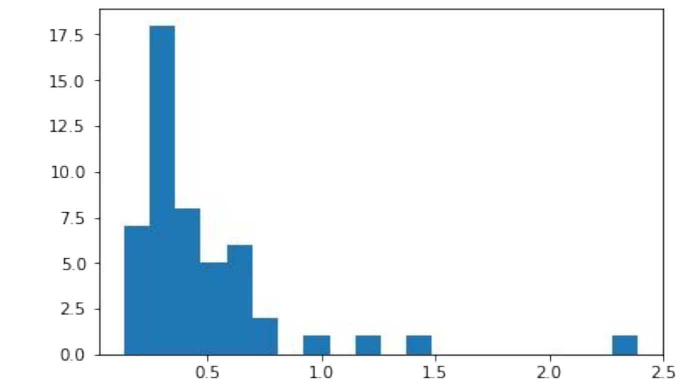
- We fill in the minority class with 4500 generated images
- GAN did not perform better than the other synthetic methods, with **75.6%** test accuracy and **52%** minority class recall
- Our images look somewhat like trucks, but it seems like we are merely generating similar images, but noisier
- Intuition: For generative methods, need to find a way to simulate the real distribution; existing methods using current dataset cannot break out of information trap

7. Adaptive Boosting

- Ensemble method:



- Algorithm : Fit new model, reweigh to prioritize misclassified examples, combine models to predict (higher say for better models)
- Trained 100 models with 30 epochs each.
- Results:
  - Test accuracy: **64%**, truck recall: **23%**; trucks misclassified as cars
- Classifier Weights:



- Explanation:
  - Re Weighing to prioritize misclassified examples creates a series of models overfitted to specific classes
  - Most models perform equally badly overall; combining models dilutes stronger classifiers (See histogram)
  - Strong classifiers on minority class have low weights overall accuracy and hence low weight in final classification, explaining poor minority class recall

Conclusion

- Two key takeaways:
  - Current synthetic methods cannot simulate true distribution for image data
  - Current prediction agglomeration technique (by overall test accuracy) for boosting dilutes classification strength for majority class and biases against weak classifiers
- Next steps (For final report):
  - Experiment with use of data augmentation/transfer learning for better GAN generation
  - Improve boosting algorithm by re-writing ensembling technique. We weigh each classifier for each specific class, and hence each class will be predicted with a combination of classifiers strongest for itself