Understand the Effect of Importance Weighting in Deep Learning on Dataset Shift

Junrong Zha, Shuang Gao, Yue Ma, Yanqi Xu
Center for Data Science
New York University
jz3741@nyu.edu, sg5963@nyu.edu, ym1970@nyu.edu, yx2105@nyu.edu

1 Introduction

Importance weighting is an essential method in statistics and machine learning to estimate a quantity on some target distribution, but can only sample from a different source distribution. Though deep learning becomes to dominate a broad set of prediction tasks, the effect of importance weighting over deep neural networks is little known. In our project, we try to leverage the paper [1] to investigate how importance weighting interact with deep neural networks.

We start by following the paper [1] to conduct a series of experiments across a variety of architectures. We train Logistic Regression and MLP on synthetic 2-dimensional toy datasets and visualize the change of their decision boundaries in training process, in order to compare the impact of importance weighting on different models in different scenarios. We also train the self-defined convolutional neural networks (CNNs) for binary classification task on CIFAR-10 images. In this process, we investigate the effects of importance weighting on deep neural networks on label shift over the course of training. We train the models with and without L2 regularization and dropout. Aside from this, we also test the effects of importance weighting for class imbalance by sub-sampling CIFAR-10 training examples with different class ratios.

We go beyond the paper by examining the effects of importance weighting in the covariate shift problem, rather than the label shift problem. We create the train and test date sets with covariate shift by sub-sampling and combining CIFAR-10 examples from cat, dog, automobile and truck classes. We train the CNNs with and without incorporating importance weighting to test the effects of importance weighting in the covariate shift problem.

We compare our results with the paper [1]. The paper concludes that for standard neural networks, weighting has a significant effect early in training. However, as training progresses, the effect dissipates. We find the claims from the paper is consistent with the results of our experiments. Our experiment results also agree with the paper that L2-regularization but not dropout restores certain effects of importance weight, but it only holds true in the case of balanced training data. In general, consistent with the conclusions in the paper, our results call into question the application of importance weighting in the context of deep networks.

2 Related Work

Importance weighted risk minimization is a standard tool in many machine learning tasks [2, 3]. To formalize it rigorously, given n samples from a source distribution $q(x)$, the task is to estimate some function of data $f(x)$ with samples from the target distribution $p(x)$. An unbiased estimator can be achieved using importance weighting

$$E_q[\frac{p(x)}{q(x)}f(x)] = \int_x f(x)\frac{p(x)}{q(x)}q(x)dx = E_p[f(x)] \tag{1}$$

In the situation of domain adaptation, there are two types of data shift that often happen in real world dataset and can be adjusted by importance weighting. One is covariate shift where training samples

$P_{train}(x)$ and testing samples $P_{test}(x)$ are from two distributions [4, 5]. The other is label shift where the distribution of training and testing labels $P_{train}(y)$, $P_{test}(y)$ are different [6].

Several works also try to incorporate importance weighting into deep neural networks. The work [7] uses weighted loss function to estimate individual treatment effects with different treated and control distributions. The importance-weighted risk minimization has been adopted in deep networks to combat label shifts [6, 8]. Importance weighting has also been widely applied in deep reinforcement learning. It is used to learn from logged contextual bandit feedback [9] and applied as weighted sampling for performing TD updates [10].

3 Experimental Design

3.1 Label Shift

Synthetic Data To validate the effect of importance weights during the training progress, we generate the 2-D toy data and visualize the results as described in the paper [1]. We first sample 512 points from truncated multinomial distribution with mean [0, 0] and covariance matrix $\mathbf{I}_2$, and denote these points with positive labels. Then we rotate and translate the points to create the 512 negative samples, and make sure this data set is linearly separable. We generate test data in the same way.

We train a Logistic Regression and a Multi-Layer Perceptron on the synthetic dataset. The MLP has a single hidden layer with 64 units and use ReLU as activation function. Both models are trained by the SGD optimizer with a batch size of 8 for 10000 epochs, and the learning rate is $\frac{0.01}{\sigma_{max}(X)}$, where $\sigma_{max}(X)$ is the maximum singular value of the data matrix. The learning rate is arond $1e-04$ in our scenario. We train both models with and without L2 regularization in different experiments, and try an extra experiment with MLP using dropout without regularization.

We also experiment on the 2-D moon dataset, which is not linearly separable. We generate 1024 samples evenly distributed on positive and negative classes, and split it into training set and testing set. By applying the Logistic Regression and the Multi-Layer Perceptron, we explore how the decision boundary changes with inappropriate or appropriate models. Additionally, we generate imbalanced training set for the moon data by sub-sampling with positive to negative ratios $r \in \{10:1, 1:10\}$. For each imbalanced training set, we train LR and MLP with loss function weighted by $\frac{1}{r}$ and with unweighted loss function to show the impact of importance weights on imbalanced training examples.

CIFAR-10 Binary Classification We apply class-conditioned weights of various strengths on the binary classification of CIFAR-10 images to evaluate the impact of the weights on the learned decision boundaries. The CIFAR-10 are labeled subsets of the 80 million tiny images dataset. The dataset consists of 60000 color images of size 32×32 in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. We train a binary classifier on training images labeled as cats or dogs (5000 per class). We use a simple convolutional neural network following the paper [1]: two convolution layers with 64 33 filters each and stride 1, followed by a 22 max pooling layer, followed by three convolution layers with 128 33 filters each and stride 1, followed by a second 22 max pooling layer, followed by two dense layers with 512 and 128 hidden units respectively and finally a dense layer to give binary outputs. All hidden layers employ ReLU activation functions. Two dropout layers are applied before each dense layers for models using dropout.

The models are trained for 500 epochs using minibatch SGD with a batch size of 16 and momentum as 0.9. The learning rate for SGD is 0.01. We evaluate on all 10000 test images from all classes, as well as 1000 random noise images. Experiments were run with importance weights $w \in \{1:128, 1:32, 1:8, 1:1, 8:1, 32:1, 128:1\}$. We also run experiments using the CNN classifier with the L2 regularization and the dropout. For L2 regularization, we set the penalty coefficient as 0.001. For the dropout models, we set the dropout rate as 0.5. In order to compare the agreement between models with different importance weights in different test distribution, we compute the fraction of images classified as dogs in the cat and dog classes, other 8 classes, and random noise images separately. Each model runs with 3 different random seeds to consider the standard deviation.

CIFAR-10 Binary Imbalanced We also conduct experiments to investigate the effect of importance weighting on class imbalance. We train a binary classifier for two classes of CIFAR-10: cat and dog. To simulate the class imbalance situation, we sub-sample the dog and cats training examples with ratios of $r \in \{16:1, 8:1, 4:1, 1:4, 1:8, 1:16\}$. For each ratio, we train models

without importance weighting and with loss function weighted by $1/r$. We also train models with weighted loss function and $L2$ regularization. We use the same CNN architecture as defined above and hyperparameters also hold the same. For each experiment, we train 3 models with a different random seed to compute mean and standard deviation.

3.2 Covariate Shift

Having noticed the effects of the importance weight on the label shift problem, we became interested in testing its impacts on the covariate shift problem. Unlike the label shift, which we could easily create by making the training and testing datasets imbalanced, the covariate shift is hard to quantify in the image classification scenario. Theoretically, each pixel of each channel serves as a feature of an image, and we would need to specify the joint distribution of all the pixels in both training and testing datasets to measure the covariate shift and come up with sensible importance weights. However, given the infinite number of values each pixel could take, it is extremely challenging for us to even approximate such distribution. Therefore, we would like to produce the covariate shift on a higher level by combining the original 4 classes cat, dog, car and trunk into 2 new classes: animal and vehicle. Our new task became classifying the images of cat, dog, car or truck as one of the two new classes. Two images originally belonging to cat and dog respectively contain the separate information that resembles the information conveyed by two different pixels in the original image classification task. And the covariate shift originally bonded to the change of distribution of pixels was then realized as the different ratios of cat/dog (or car/trunk) in training and testing datasets. By introducing this abstraction, we were able to precisely define the importance weights to be the inverse of the cat/dog ratio (or similarly car/trunk ratio), and therefore study the impacts of the importance weight in the image classifcation scenario.

We anticipated the difficulties of our new image classification task. The crucial patterns for identifying a cat could be different from those of identifying a dog, making the neural networks struggled to discover common features. Therefore, besides the weighted model, we would also fit an unweighted model, and a model trained on a dataset with no covariate shift. We would expect our weighted model to outperform the unweighted model on the dataset with covariate shift but to underperform the model on the dataset with no covariate shift. As long as the importance weights make the performances closer to those resulted from the optimal setting (with no covariate shift), we would conclude the importance weights indeed played an role.

4 Results

4.1 Synthetic Data

For the linear-separable dataset, Figure 1, 2, 3 show that no matter what weights applied to the loss function, the decision boundary will eventually converge to the max-margin separator for both Logistic Regression and MLP. For the moon dataset, Figure 4 shows the experiment results. Although the decision boundary of MLP converges and separates the two classes completely with different importance weights, we observe that the decision boundary of the Logistics Regression model depends on the importance weights in epoch 10000. With positive:negative = 10:1, the decision boundary tends to place all positive samples in one side by misclassifying more negative samples, and vice versa. With the equal importance weights, the decision boundary makes errors on the two classes equally. This is because the LR model cannot separate the moon data perfectly. When the model inevitably makes misclassifications, it shows partiality for the side with higher importance in loss function to minimize the weighted risk.

We question whether the difference between the visualizations associated with epoch 1 is caused by down-weighting or up-weighting at the initialized state, and whether the convergence depends on the initialized state. To validate this, we repeat the experiments on moon data for 5 trails and record the tracks of the fraction of positive predictions and the test accuracy during the training process. In Figure 5, we observe that the initialized states are highly random, but the lines associated with the same weights gather rapidly. The final states at the end of epoch 10000 are related to importance weights for Logistic Regression while independent on importance weights for MLP.

Figure 6 and 7 show the results for imbalanced moon data. We find that weighting the loss function according to the ratio of positive to negative labels in the training set can improve both the Logistic

Regression and the MLP. With weighted loss, the models can make predictions as accurately as they are on the balanced training set, while with unweighted loss function, the decision boundaries show partiality to the side with more training examples.

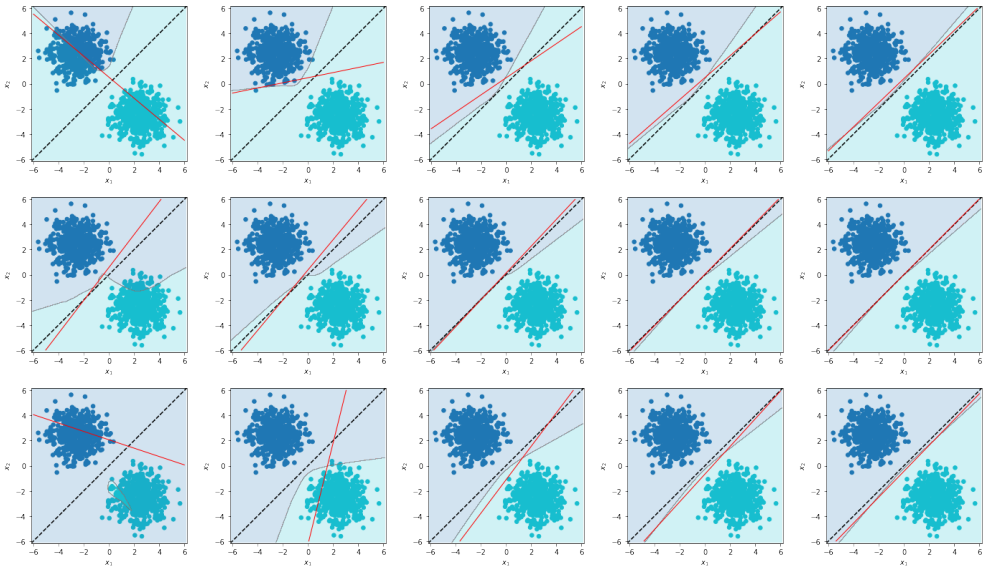


Figure 1: Results on linearly-separable dataset. The model is LR and MLP without L2 regularization. From top to bottom, the rows show plots for $positive : negative = \{10 : 1, 1 : 1, 1 : 10\}$ respectively. From left to right, the columns represent epoch 1, 10, 100, 1000 and 10000. The positive samples are colored with light blue, while the negative samples are colored with dark blue. The background shading depicting the decision surface of an MLP. The red lines are decision boundaries of Logistic Regression, and the dashed black lines are max-margin separators.

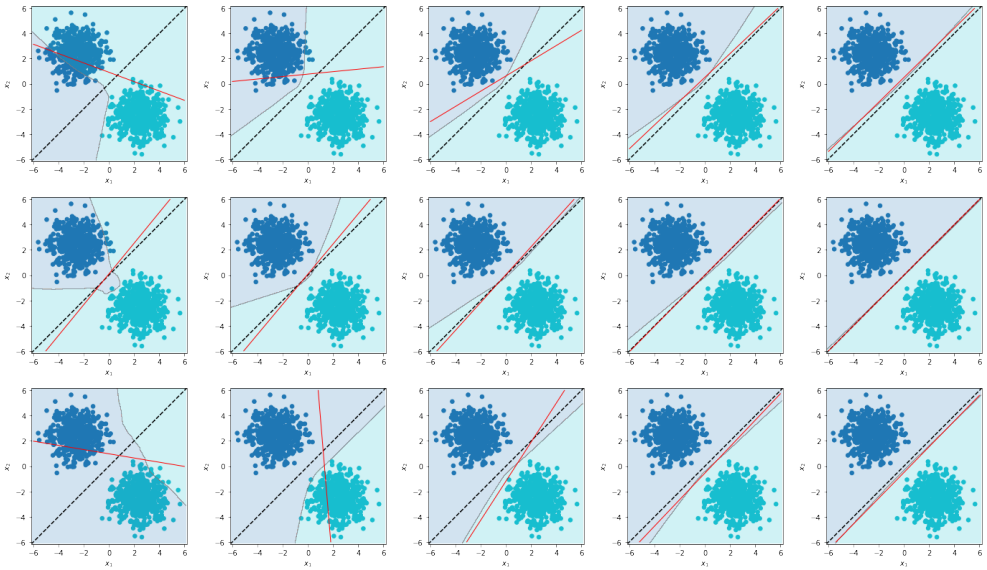


Figure 2: Results on linearly-separable dataset. The model is LR and MLP with L2 regularization.

4.2 CIFAR-10 Binary Classification

Figure 8 shows the results for CIFAR-10 binary classification. From (a)-(c), the fractions of images classified as dogs in the cat and dog test set (a), other eight classes (b), and random images (c) vs

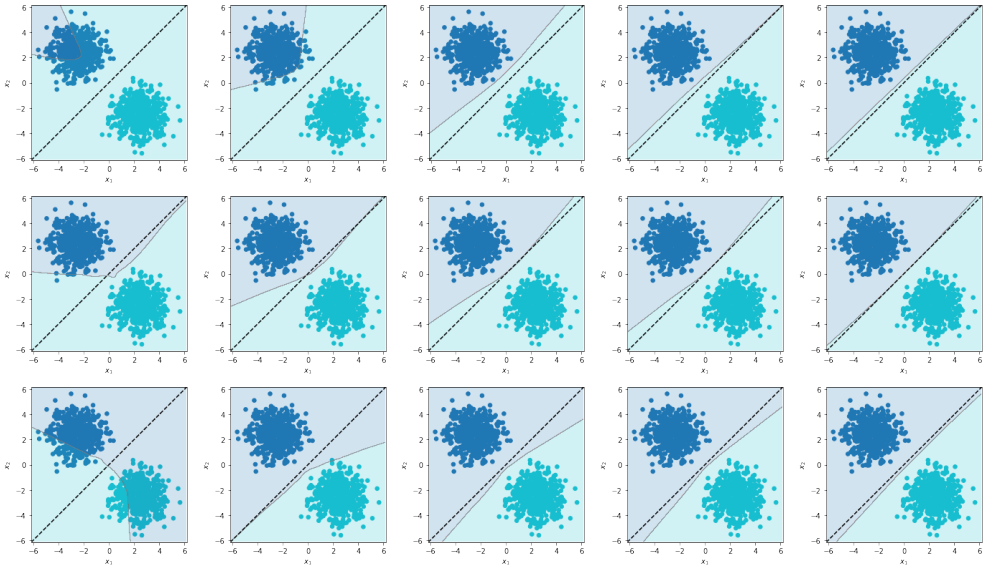


Figure 3: Results on linearly-separable dataset. The model is MLP with dropout rate = 0.5.

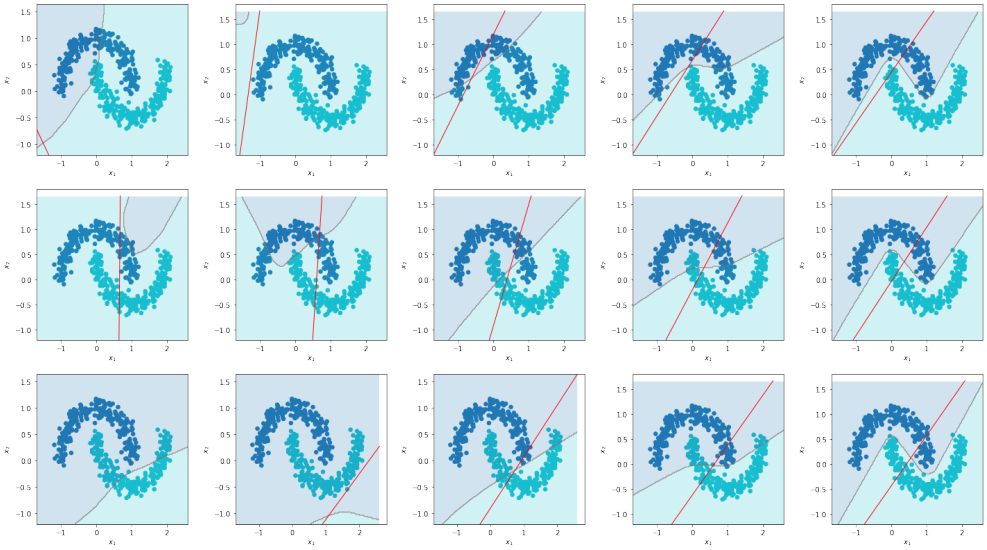


Figure 4: Results on not linearly-separable moons dataset. The models are LR and MLP without L2 regularization.

training epochs on different important weights are plotted. In Figure 8a, we observe that even though during early epochs, the fractions of images classified as dogs corresponds to the importance weights, the discrepancy keeps diminishing. When the epoch reaches to around 230, the fractions of dogs on all important weighted models converge to 0.5, which is equal to the fraction of dogs in the cat and dog test set as expected. Similar patterns are discovered among the test images of other eight classes in Figure 8b and random noise images in Figure 8c. When epoch reaches to around 230, the fractions of images predicted as dogs for all weighted models converge to around 0.3 among other 8 classes and converge to near 0 among random images. The model with different weighting ratios agrees on the out-of-sample images. Besides, it seems that all models have near-perfect agreement on random noise images which are nearly-always classified as cats. These three figures show that as training progresses, the effects due to importance weighting vanish. The weighting impacts the CIFAR-10 binary classification early in training, however, after many epochs of training, there is no

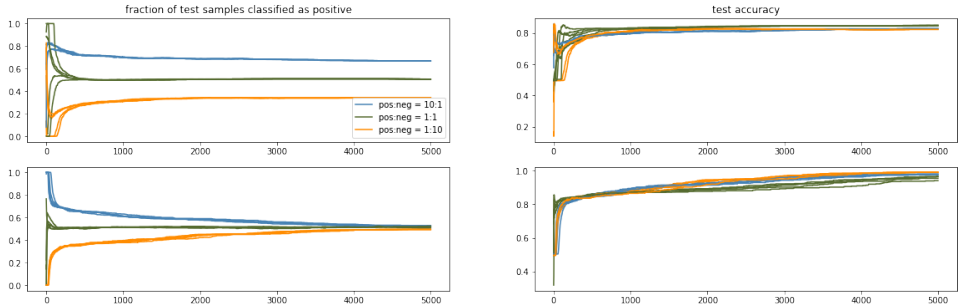


Figure 5: Results on not linearly-separable moons dataset. The top shows faction of positive prediction and test accuracy against number of training epochs corresponding to LR, and the bottom shows fraction and accuracy plots corresponding to MLP.

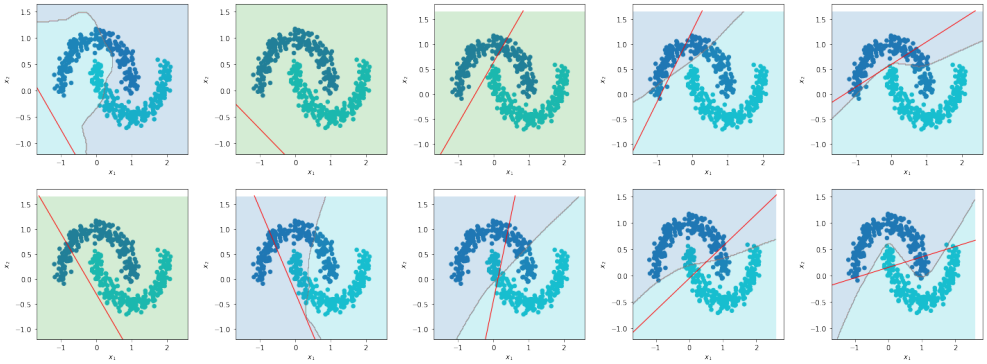


Figure 6: Results on imbalanced moon dataset. In training set, $positive : negative = 10 : 1$. The top shows the change of decision boundaries with unweighted loss function. The bottom shows the change of decision boundaries with loss function up-weighting the negative samples by 10. The green background shadows means that the MLP classify the whole surface as the same label.

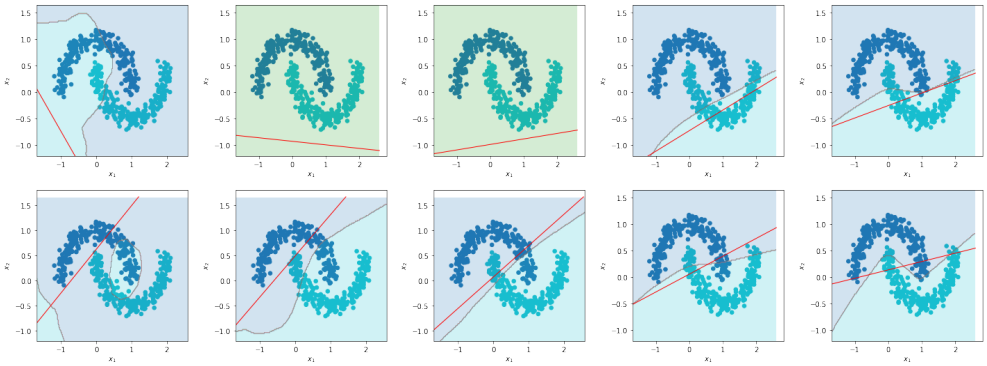
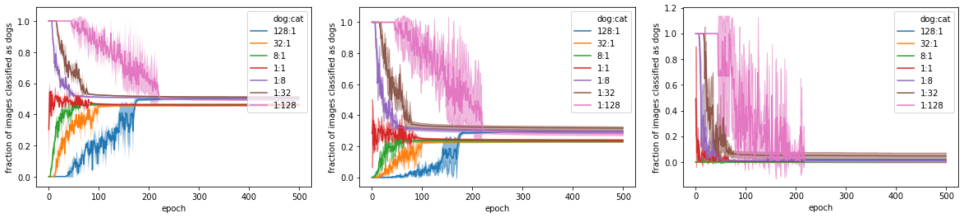


Figure 7: Results on imbalanced moon dataset. In training set, $positive : negative = 1 : 10$. The top shows the change of decision boundaries with unweighted loss function. The bottom shows the change of decision boundaries with loss function up-weighting the positive samples by 10. The green background shadows means that the MLP classify the whole surface as the same label.

clear relation between the class-based importance weights and the classification ratios on either test set images, out-of-domain images, or random images.

Figures 8d-8f demonstrate the fractions of examples classified as dog with different importance weights at epoch 500 using vanilla model, model trained with L2 regularization and model trained with dropout respectively. For the vanilla model and model with dropout, the fractions of images classified as dogs at 500 epochs just have slight fluctuation among all the weights for each part of the

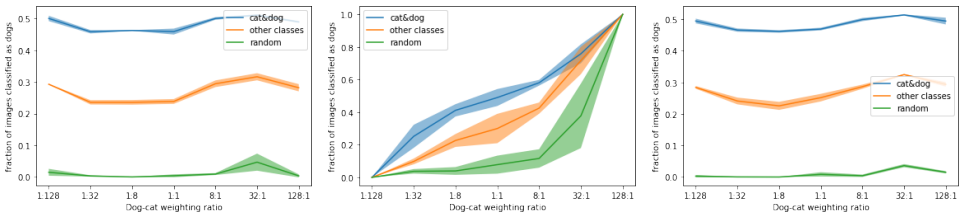
test set. This visualization further confirms the results of Figure 8a-8c and shows that adding dropout would not restore the effect of importance weighting. However, when model is trained with L2 regularization, the fractions increases from around 0 to 1 as the weighing ratio changes from $1:128$ to $128:1$ for all three parts of the test set as shown in Figure 8e. In summary, we note that, not only do differently weighted CIFAR models converge to similar classification ratios, but they also tend to agree on example labels, i.e., they learn similar separators. Adding L2 regularization to models, but not dropout, would restore some effect of importance weights when performing CIFAR-10 binary classification task.



(a) CIFAR-10 cat and dog test images.

(b) CIFAR-10 other 8 classes test images.

(c) Random images.



(d) Classification ratios at epoch 500.

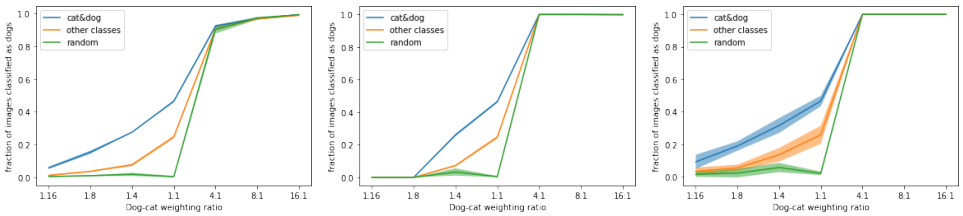
(e) Classification ratios with L2 regularization on weights.

(f) Classification ratios with dropout.

Figure 8: (a-c) Relationship between early stopping and importance weighting. We plot the fraction of images classified as dogs (y-axis) in the cat and dog test set (a), other eight classes (b), and random images (c) vs training epochs (x-axis). (d-f) Fraction of examples classified as dogs (y-axis) vs importance weights (x-axis) after 500 epochs of training. We also show results from models trained with L2 regularization (e) and dropout (f). In all plots error bands show standard deviation across three random initializations, and lines represent means.

4.3 CIFAR-10 Binary Imbalanced

Importance weighting is commonly used to correct for class imbalance. Figure 9 shows the results of training imbalanced data without weighting, with weighting and with weighting along with L2 regularization. Figure 9a indicates that the imbalanced training data makes the models more incline to predict images as the class with more training data. However, as demonstrated in Figure 9b, importance weighting does not help with class imbalance in this case. Moreover, adding L2 regularization does not make any difference neither.



(a) Without importance weighting

(b) With importance weighting

(c) With weighting and add L2

Figure 9: Imbalanced binary classification with and without importance weighting

4.4 Covariate Shift

By combining two original classes into a new class twice, we were able to explicitly form the desired covariate shift and calculate the optimal importance weights, but at the same time, under each new class, we created a heterogeneous dataset, making it challenging for our models to extract common patterns and produce well-generalized predictions. Therefore, as illustrated by the right plot in Figure 10, the validation accuracies for three methods were around 0.503. Due to the heterogeneity under each new class, our models actually optimized by relying on features of only one particular group, e.g. either dog or cat, of each class to make classifications. Therefore, whether or not we chose to downsample one particular group did not bring a huge impact on the performances of the models. The importance weights, however, forced the model to pay similar attentions to both groups in a class, leading to the decreased training accuracy as indicated by the gap between the green and the blue/orange lines (Figure 10). Had there been less heterogeneity within the new classes and more between, we could expect the weighted model successfully capturing features of both groups of each new class to outperform the unweighted model and gradually approach the optimal accuracy achieved when there is no covariate shift between training and testing datasets.

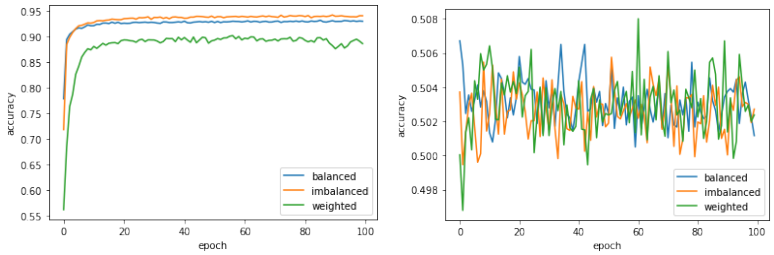


Figure 10: Results on dataset with covariate shift with train on the left and test on the right.

5 Conclusions and Discussion

In this project, we discuss the effect of importance weighting in different scenarios thoroughly. With the synthetic data, we show the importance of weighting affects the decision boundary in early state, but along with the increasing number of training epochs, the impact vanishes and depends on datasets and models at the final states. When data is linearly separable, simple machine learning models can easily make predictions with 100% accuracy, so the effect of importance weighting in loss function diminishes and the decision boundaries converge to the max-margin separators (Figure 1). When data is not linearly separable, the impact is different for misspecified model and correctly specified model (Figure 4). A linear model (LR) trained on the inseparable data classifies more test examples as the label assigned higher weights at the final states. A non-linear model (MLP), as is correctly specified, can separate the two classes perfectly no matter how the loss function is weighted.

In the case of training deep neural networks for binary image classification on CIFAR-10, we discover that, similar to the paper [1], the importance weighting effects the predictions early in training. However, the effect dissipates after training for enough epochs and all the weighted models converges to similar prediction ratios. We suspect that since the deep learning models are over-parameterized and very expressive, they can usually separate the classes after training for appropriate number of epochs regardless of the weightings. Therefore, deep nets approach similar solutions across different weighted models on either test set images, out-of-domain images, or random vectors in our experiments, after several hundreds of training epochs.

To test the impact of importance weights when combining with regularization techniques, we use L2 regularization and dropout. With the linearly separable training set, there is little difference in epoch 10000 over the decision boundaries when applying L2 regularization or dropout or not either Figure 1, 2, 3. This is reasonable as our linearly-separable dataset is so simple that the regularization techniques are not effective on the results. However, we notice there are obvious impacts of importance weights when combining L2 regularization, but not dropout, to deep nets on the CIFAR-10 dataset. This is consistent with the conclusion as described in [1] that L2 regularization restores certain effects of importance weighting. Intuitively, it makes sense that importance weighting works better with more

biased models and l2 regularization produces a more biased model, while dropout regularizes the model through reducing the number of parameters.

To test the impact of weight correction on imbalanced data, we sub-sample one class in the training set while keeping the testing set distributed evenly on the two classes. With synthetic imbalanced moon data (Figure 6, 7), both the unweighted linear model (LR) and unweighted non-linear model (MLP) classify all test samples in the class with more training samples correctly but make mistakes on another class. But when using importance weights to correct the imbalance, the models improve a lot and classify the test set almost evenly in epoch 10000, and the non-linear model can almost perfectly separate the two classes. This indicates the effectiveness of importance weighting on imbalance correction when using traditional machine learning models. When classifying CIFAR-10 imbalanced data, deep neural networks, on the other hand, do not manifest any differences before and after using weighted loss function to combat the class imbalance, which is consistent with the appendix 6 of the paper [1]. However, surprisingly, contrary to the conclusion we made on CIFAR-10 binary classification, L2 regularization does not restore any effects of importance weighting in the case of class imbalance. More experiments are needed to make rigorous conclusions from this conflicting results. We suspect one possible reason might be that we have different total training samples for different sub-sampled training sets.

We don't see much difference among the test accuracies in the three experiments we conducted for the covariate shift problem. It seems that whether we train the model with covariate shift exists between train and test sets or not, the self-defined CNNs can't generalize well on the test set. The importance weighting might play a role to some extent early in training, but the increase is too subtle to make solid conclusions from it. The possible reason could be the train and test dataset we create is not suitable for examining the covariate shift problem. In our course, the sample data point is 1-D dimension, but the image data is 2-D dimension. If we have more time, we may leverage other more reasonable dataset to test the effect of importance weight in the covariate shift problem. We may also apply more sophisticated models with importance weighting to enhance the model performance.

References

[1] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881. PMLR, 2019.

[2] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[3] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method*. Wiley Publishing, 3rd edition, 2016.

[4] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[5] A. Gretton, A.J. Smola, J. Huang, Marcel Schmittfull, K.M. Borgwardt, Bernhard Schölkopf, J. Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning, 131-160 (2009)*, 01 2009.

[6] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018.

[7] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR, 2017.

[8] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

[9] Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018.

[10] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.