What is the Effect of Importance Weighting in Deep Learning?

Jonathon Byrd [1] **Zachary C. Lipton** [1]

Abstract

Importance-weighted risk minimization is a key ingredient in many machine learning algorithms for causal inference, domain adaptation, class imbalance, and off-policy reinforcement learning. While the effect of importance weighting is well-characterized for low-capacity misspecified models, little is known about how it impacts over-parameterized, deep neural networks. Inspired by recent theoretical results showing that on (linearly) separable data, deep linear networks optimized by SGD learn weight-agnostic solutions, we ask, *for realistic deep networks, for which many practical datasets are separable, what is the effect of importance weighting?* We present the surprising finding that while importance weighting impacts deep nets early in training, so long as the nets are able to separate the training data, its effect diminishes over successive epochs. Moreover, while L2 regularization and batch normalization (but not dropout), restore some of the impact of importance weighting, they express the effect via (seemingly) the wrong abstraction: why should practitioners tweak the L2 regularization, and by how much, to produce the correct weighting effect? We experimentally confirm these findings across a range of architectures and datasets.

1. Introduction

Importance sampling is a fundamental tool in statistics and machine learning often used when we want to estimate a quantity on some *target* distribution, but can only sample from a different *source* distribution (Horvitz & Thompson, 1952; Kahn & Marshall, 1953; Rubinstein & Kroese, 2016; Koller et al., 2009). Concretely, given n samples $x_1, ..., x_n \sim p(x)$, and the task of estimating some function of the data, say $f(x)$, under the target distribution $E_q[f(x)]$, importance sampling produces an unbiased estimate by weighting each sample x according to the likelihood ratio $q(x)/p(x)$:

$$\mathbb{E}_p\left[\frac{q(x)}{p(x)}f(x)\right] = \int_x f(x)\frac{q(x)}{p(x)}p(x)dx$$
$$= \int_x f(x)q(x)dx = \mathbb{E}_q\left[f(x)\right]$$

Machine learning practitioners commonly exploit this idea in two ways: (i) by re-sampling to correct for the discrepancy in likelihood or (ii) by weighting examples according to the likelihood ratio (Rubinstein & Kroese, 2016; Shimodaira, 2000; Koller et al., 2009). For this reason, among others, weighted risk minimization is a standard tool that emerges in a wide variety of machine learning tasks.

In domain adaptation, when source and target data share support, practitioners commonly adjust for distribution shift by estimating likelihood ratios. This is done either as a function of the input x: $q(x)/p(x)$ (in the case of covariate shift) or of the label y: $q(y)/p(y)$ (in the case of label shift), training a *corrected* classifier with *importance-weighted empirical risk minimization (IW-ERM)* (Shimodaira, 2000; Gretton et al., 2009; Lipton et al., 2018). One related use of IW-ERM is to correct for sampling bias in active learning (Beygelzimer et al., 2009; Settles, 2010). The technique is also frequently employed in off-policy reinforcement learning (Precup, 2000; Mahmood et al., 2014; Swaminathan & Joachims, 2015), where we desire to learn a new policy given offline samples collected from a preexisting policy. Weighted loss functions also arise in a number of other contexts, including label noise and crowdsourcing.

1.1. Deep learning and weighted risk minimization

When our hypothesis class consists of low-capacity models that are misspecified, importance weighting has well-known benefits. Consider the simple case of fitting a linear model to data generated by a higher-order polynomial. For a reasonably large training set, our model must make errors somewhere. By altering the relative contribution of mistakes on various training points to our loss function, importance weights typically lead us to fit a different model.

As deep learning has come to dominate a broad set of prediction tasks, importance-weighted risk minimization has

[1] Carnegie Mellon University. Correspondence to: Jonathon Byrd <jabyrd@cmu.edu>, Zachary C. Lipton <zlipton@cmu.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

remained a standard technique. (Shalit et al., 2017) employ neural networks to estimate individual treatment effects by weighting their loss function to compensate for differences in treatment group size. In a work on deep learning and crowdsourcing, Khetan et al. (2018) proposed a weighted loss as part of an iterative scheme for jointly estimating worker quality and learning a classifier from noisy data. In recent work on label shift, Lipton et al. (2018); Azizzadenesheli et al. (2019), propose adapting deep networks with importance-weighted risk minimization.

Applications of importance weighting also abound in deep reinforcement learning. For example, Joachims et al. (2018) use the technique to learn from logged contextual bandit feedback. Other applications include deep imitation learning (Murali et al., 2016). In one paper, Schaul et al. (2016), employ a weighted sampling to choose experiences from the replay buffer for performing TD updates. These weights are not based on likelihood ratios, but are chosen heuristically to be proportional to the Bellman errors.

Despite the popularity of importance sampling in combination with deep neural networks, how and when it works remain open questions. Unlike linear models, deep neural networks are generally over-parameterized, capable of fitting training datasets to perfect accuracy (Zhang et al., 2017). Moreover, it is now recognized that for many tasks deep neural networks continue to improve generalization error past the point of achieving zero training error (Soudry et al., 2017). Thus they are not only *capable* of separating the training set (given enough epochs) but actually are trained to do so in common practice. Since neural networks are capable of shattering the training set (and often do), it is not clear that any trade-offs must be made among classifying each of the training points. Thus any effects of importance weighting depend crucially on how they impact the dynamics of optimization, an actively-studied but still poorly-understood topic.

1.2. Salient findings

In this paper, we investigate the effects of importance weighting in deep learning across a variety of architectures, tasks, and data sets. We present the surprising result that importance weighting may or may not have any effect in the context of deep learning, depending on particular choices regarding early stopping, regularization, and batch normalization. Our experiments focus on classification problems: we apply class-conditioned weights of various strengths, evaluating the impact of the weights on the learned decision boundaries. We consider the effect of weighting on both how training and test points are classified, and also the effect on off-manifold data points and random noise. Our experiments address both the classification of CIFAR-10 images and paraphrase detection using data from the Microsoft Research Paraphrase Corpus (MRPC). We also build intuition by considering 2D synthetic datasets for which we can visualize decision boundaries.

Across tasks, architectures and datasets, our results confirm that for standard neural networks, weighting has a significant effect early in training. However, as training progresses the effect dissipates and for most weight ratios considered (between 256:1 and 1:256) the effect of importance weighting is indistinguishable from unweighted risk minimization after sufficient training epochs. While L2 regularization restores some of the impact of importance weighting, this has the perplexing consequence of expressing the *amount* by which importance weights affect the learned model in terms of a seemingly unrelated quantity—the degree of regularization—prompting the question: *how does one appropriately choose the L2 regularization given importance weights?* Interestingly, dropout regularization, which is often used interchangeably with L2 regularization, does not exhibit any such interaction with importance weighting. Batch normalization also appears to interact with importance weights, although as we will discuss later, the precise mechanism remains unclear.

1.3. Contributions

In summary, our contributions are the following:

1. We demonstrate the surprising finding that for unregularized neural networks optimized by *stochastic gradient descent (SGD)*, the impact of importance weighting diminishes over epochs of training.

2. We show that L2 regularization and batch normalization, (but not dropout) interact with importance weights, restoring (some) impact on learned models.

3. We replicate our results across a variety of networks, tasks, and datasets.

4. Our results call into question the standard application of importance weighting when applied to deep networks, a finding with practical consequences on the fields of causal inference, domain adaptation, and off-policy reinforcement learning.

2. Theoretical Motivation

The empirical questions addressed in this paper draw inspiration from recent developments in the theory of deep learning. In particular we are motivated by finds of Soudry et al. (2017) and Gunasekar et al. (2018), who investigate the decision boundaries learned by neural networks. Note that although these theoretical analyses cover only *shallow linear*, *deep linear*, and *deep convolutional linear* neural networks, our experiments draw intuition from the results and confirm empirically that the hypotheses hold on more practical nonlinear networks.

Soudry et al. (2017) note that it is common practice to train neural network classifiers to overfit badly, and that even past the point (in terms of training epochs) of achieving zero training error, although the negative log-likelihood on holdout data begins to increase, the generalization error often continues to decrease. To analyze this phenomenon, they restrict their attention to linear networks which are presently more amenable to the available tools of analysis.

They consider the simple case where the model consists of a linear separator, the data is linearly separable, the optimization objective is cross-entropy loss, and the optimization algorithm is SGD. Notably, there is no finite minimizer w^* of the objective, since for any w that separates the data, an even lower loss could be achieved by scaling up the weights w. Thus the weights themselves do not converge. However, noting that the learned decision boundary depends only on the direction of the weights (but not their magnitude), Soudry et al. (2017) examine what, if anything, the *direction* $w_t/||w_t||$ converges to (over training iterations of SGD). Surprisingly, they conclude that the weights converge in direction to the solution of the hard-margin *support vector machine*. In short, the proof follows because over epochs of training, the norm of the weight vector increases, causing the support vectors to dominate the loss function (under a set of conditions satisfied by the cross-entropy loss). Subsequent results confirm that this finding holds for deep fully-connected networks of linear units (Gunasekar et al., 2018), and that for deep convolutional networks of linear units, a related result holds (Gunasekar et al., 2018), showing implicit bias towards minimizing the $\ell_{2/L}$ bridge penalty in the frequency domain of the corresponding single-layer linear predictor.

One interesting ramification of this theoretical result is that the hard-margin solution depends only on the *location* of data points, and thus is unaffected by *oversampling/reweighting*. While Soudry et al. (2017) and Gunasekar et al. (2018)'s analyses only address linear networks and linearly-separable data, their findings motivate our hypothesis that a similar weight-invariance property might hold for typical modern deep (nonlinear) neural networks, for which many datasets of practical interest are separable (Zhang et al., 2017).

These results also motivate our follow-up questions concerning the effect of regularization. Common regularization methods like L2 regularization penalize the large-norm solutions that minimize cross-entropy on separable data. Since L2 regularization prevents such large-norm solutions, what if anything is the impact of importance weights in this case? Moreover, while *dropout* (Srivastava et al., 2014) is often thought of as a regularization method for deep networks, it does not penalize large-norm solutions. Thus we hypothesize that these regularization methods would have differential impacts on the solutions found by SGD on deep networks in conjunction with importance weighting.

3. Experiments

We investigate the effects of importance weighting on neural networks on two-dimensional toy datasets, the CIFAR-10 image dataset, and the Microsoft Research Paraphrase Corpus (MRPC) text dataset. Our experiments address the label shift scenario, weighting examples based on their class. Specifically, we down-weight the loss contributions of examples from a particular class. We also test the combination of regularization and IW-ERM on both CIFAR-10 and a toy dataset. For L2 regularization, we set the penalty coefficient as 0.001, and when using dropout on deep networks, we set the values of hidden units to 0 during training with probability $\frac{1}{2}$.

Synthetic Data In order to visualize decision surfaces, we conduct an experiment with a synthetic two-dimensional linearly-separable dataset. To form the *positive examples*, we sample 512 points from a 2D truncated normal distribution. To generate *negative examples*, we rotate and translate the positive examples (see Figure 1). We train both a logistic regression model (without regularization) and a multi-layer perceptron (MLP) using minibatch SGD for 10,000 epochs with a batch size of 8. The MLP has a single hidden layer of 64 hidden units with ReLU activations. Both models use a fixed learning rate of $\frac{1}{\sigma_{max}(\mathbf{X})}$, where $\sigma_{max}(\mathbf{X})$ is the maximum singular value of the data matrix. This learning rate was chosen to match the experiments of Soudry et al. (2017), and took a value of ≈ 0.045 on our dataset. Results are shown in Figures 1, 2, and 3. We also present results for experiments on a two-dimensional moons dataset and a two-dimensional overlapping Gaussian distribution dataset that are not linearly-separable (Figures A.1 and A.2).

CIFAR-10 Binary Classification We also conduct experiments on the CIFAR-10 dataset (see results in Figure 4). Here, we train a binary classifier on training images labeled as cats or dogs (5000 per class), evaluating on all 10000 test images from all 10 classes as well as 1000 random noise images. The classifier is a convolutional network with the following structure: two convolution layers with 64 3×3 filters each and stride 1, followed by a 2×2 max pooling layer, followed by three convolution layers with 128 3×3 filters each and stride 1, followed by a second 2×2 max pooling layer, followed by two dense layers with 512 and 128 hidden units respectively, and finally the binary output layer. All hidden layers employ ReLU activation functions. The models are trained for 1000 epochs using minibatch SGD with a batch size of 16 and no momentum. All models

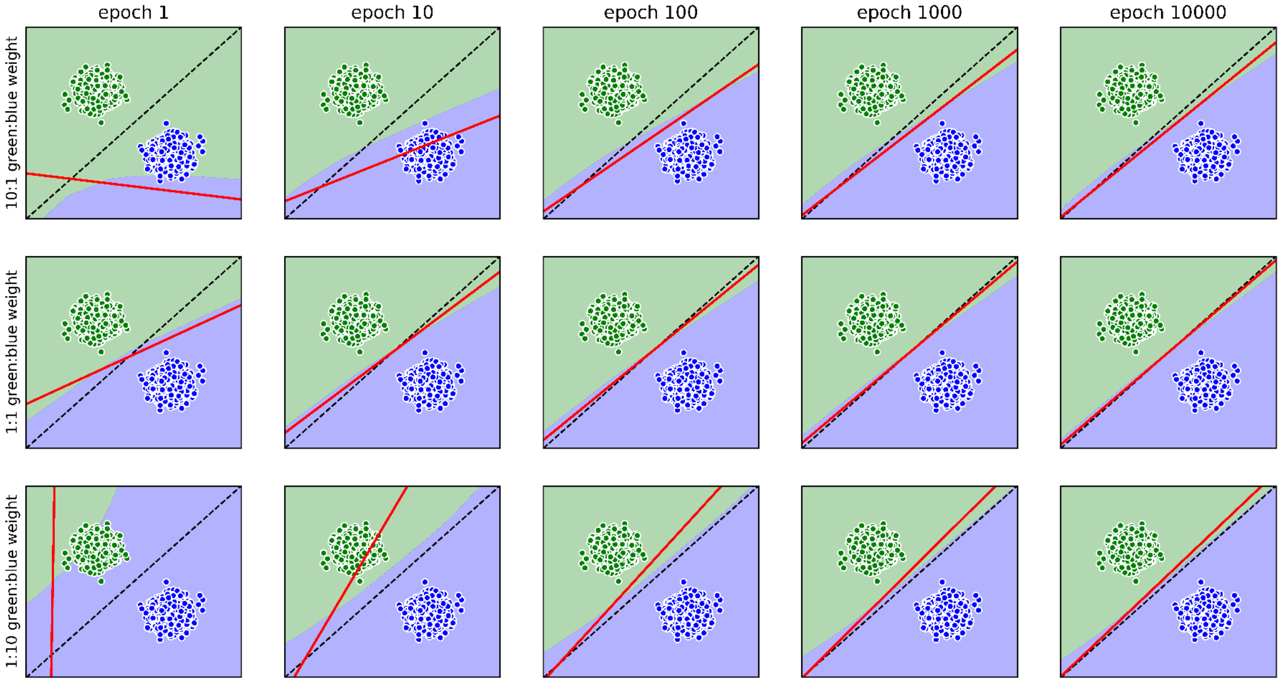


Figure 1. Convergence of decision boundaries over epochs of training with different importance weights (top to bottom). Points are colored according to their true labels, with background shading depicting the decision surface of an MLP with a single hidden layer of size 64. The red line shows the logistic regression decision boundary. The dotted black line shows the max-margin separator.

trained with SGD use a constant learning rate of 0.1, except for the dropout models with no importance weighting which used a learning rate of 0.05 due to weight divergence issues. We also ran experiments with the Adam optimizer (Kingma & Ba, 2015) with learning rate $1e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e{-}8$ (Figure A.9). Experiments were run with importance weights of inverse powers of 2 up to 2^{-8} for each class, as well as with no importance weighting for unregularized models. Results are given for importance weights of inverse powers of 4 for regularized models. Figure A.7 shows results on CIFAR-10 cats and dogs with label noise in the training data where the underlying noisy distributions are not separable, but the finite sample is separable. We create samples with label noise by flipping the labels of 5% of training examples from each class. Figure A.8 shows results from training the convolutional network model on CIFAR-10 images labeled automobile or truck.

All unregularized classifiers without data sub-sampling or label noise achieve (unweighted) test accuracy between 80% and 85%. In addition to noting the similarity of models across important weightings both in terms of accuracy and the fraction of examples predicted to belong to each category, we investigated the extent to which the models agreed with each other on precisely which examples belonged to each class. We compare (i) the agreement between models with different importance weights with (ii) the agreement between models run with different random seeds. To compute (i), we first compute test set predictions for each importance weighting by taking a majority vote over 9 different random seeds. We find that, on average, 82% of the 17 differently-weighted models agree on the label of a given test set example (74% agreement on out-of-sample CIFAR images from the other 8 classes). For (ii), we calculate the fraction of random initializations that agree on each example for each importance weighting. Then we average over both examples and importance weightings to find that, on average, 78% of random initializations agree on the label of a given test set example (71% agreement on out-of-sample images). We note that all models have near-perfect agreement on random noise images which are nearly-always classified as cats.

We repeat the same CIFAR experiment using the popular deep residual network (ResNet) architecture (He et al., 2016) consisting of a 5×5 convolution with 64 filters followed by two residual blocks with 64 filters, then two residual blocks with 128 filters, then two residual blocks with 256 filters, followed by average pooling, a dense layer with 512 nodes, and finally, the output layer. Each residual block consists of two 3×3 convolution layers. The first layer with 128 filters and the first layer with 512 filters have stride of 2. All other hyperparameters were left unchanged. Figure A.5 shows results both with and without batch normalization (Ioffe & Szegedy, 2015) applied between all convolution layers.

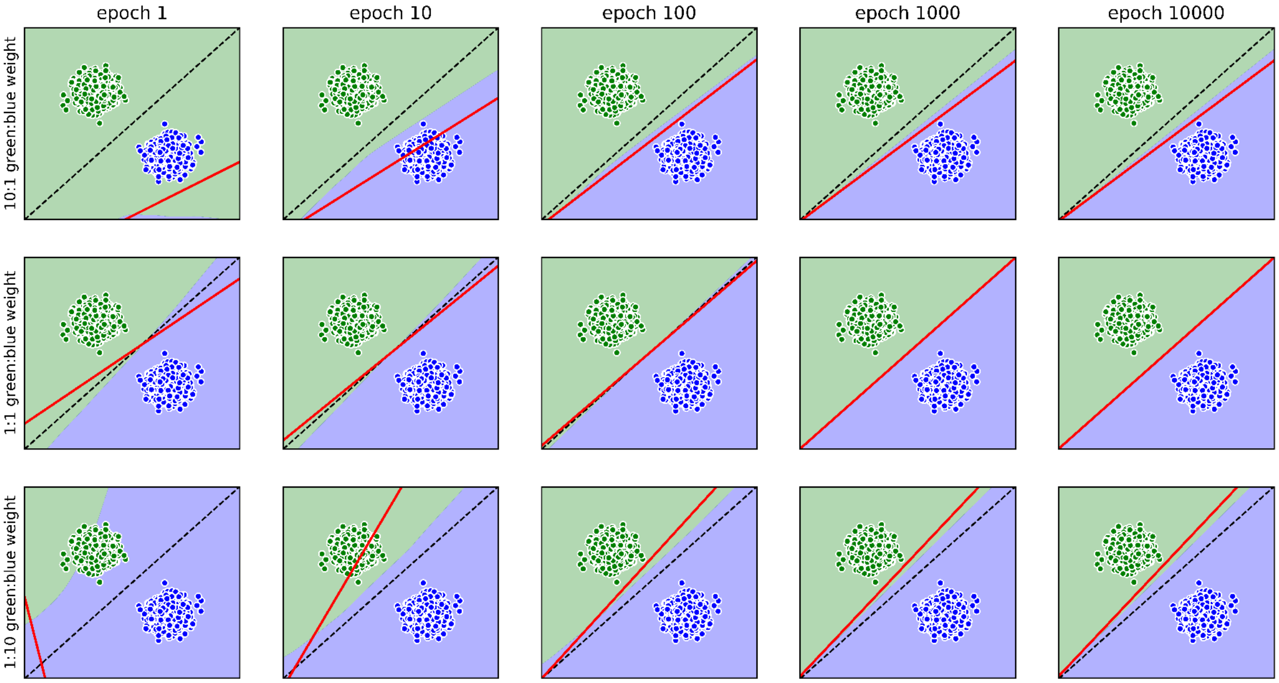


Figure 2. Same scenario as Figure 1, except both logistic regression and MLP are trained with L2 regularization.

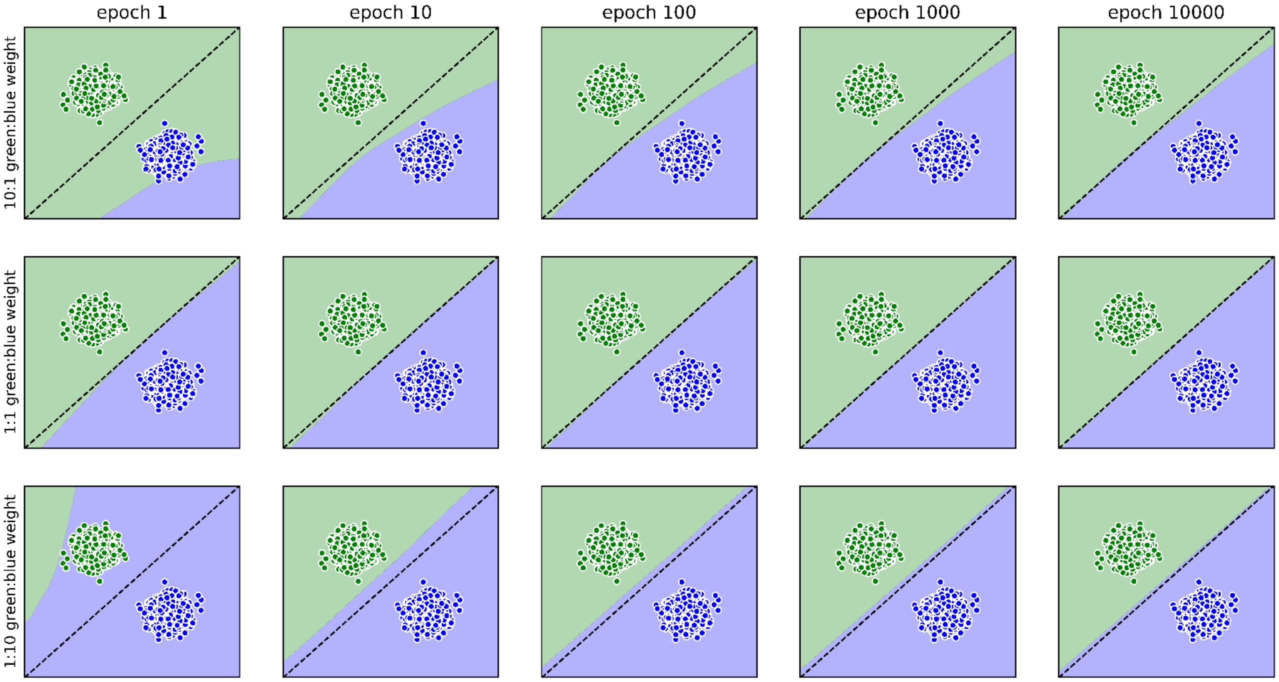
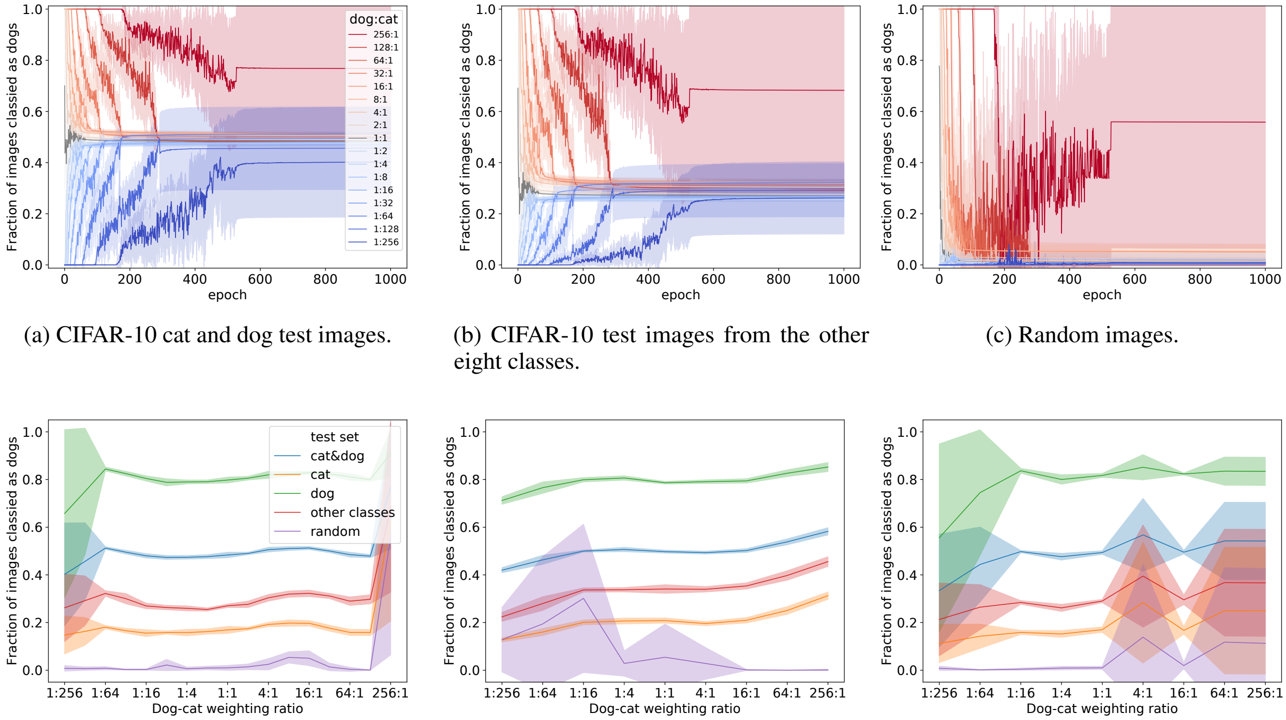


Figure 3. Same scenario as Figures 1 and 2, except the MLP is trained with dropout (no logistic regression model shown).



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

(b) CIFAR-10 test images from the other eight classes.

(c) Random images.

(d) Classification ratios at epoch 1000.

(e) Classification ratios with L2 regularization on weights.

(f) Classification ratios with dropout.

Figure 4. (a-c) Relationship between early stopping and importance weighting. We plot the fraction of images in the test set (a), other 8 classes (b), and random vectors (c), classified as dogs (y-axis) vs training epochs (x-axis). (d-f) Fraction of examples classified as dogs (y-axis) vs importance weights (x-axis) after 1000 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 nine random initializations, and lines represent means.

CIFAR-10 Imbalanced Importance weighting is commonly used to correct for class imbalance. To simulate this situation, we train on an imbalanced training set created by sub-sampling CIFAR-10 examples from either the cat or dog class (Figure A.6). We downweight the loss function for the class that wasn't sub-sampled by the same factor used to sub-sample the other class.

CIFAR-10 Multiclass In addition to binary classification, we also explore class weighting in the multiclass setting using all CIFAR-10 classes (Figure 5). We set up experiments similar to Lipton et al. (2018), where we weight the loss function contributions of one class by $\rho \in [0.2, 0.5, 0.9]$, and applying a weight of $\frac{1-\rho}{9}$ to the other nine classes. At test time, we apply the same weights to each class's contribution to accuracy in order to simulate correcting for label shift in the test set. Here, the classifier is a two-layer MLP with 256 hidden units trained using weighted cross-entropy loss with a learning rate of 0.01.

MRPC To verify that our findings hold in other domains, we conduct similar experiments on (sequential) natural language data using the Microsoft Research Paraphrase Corpus

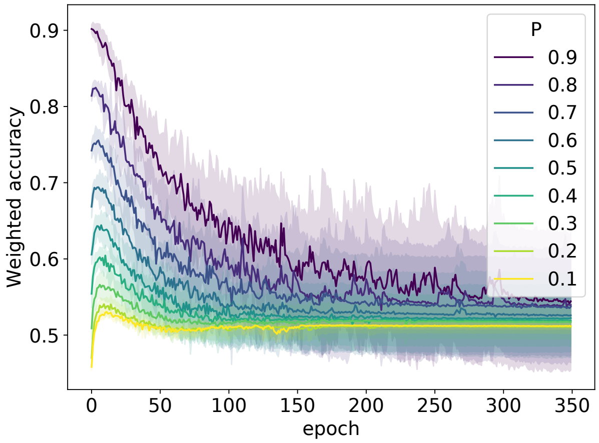


Figure 5. Effect of early stopping on test label shift correction for multiclass CIFAR-10. For each value of $\rho \in [0.1 \ldots 0.9]$, we weight one class's contributions to the loss function and test accuracy by ρ, and the other nine classes by $\frac{1-\rho}{9}$. We plot the relationship between the weighted test accuracy (y-axis) vs training epochs (x-axis). Lines and errors show the means and standard deviations over the ten CIFAR-10 classes respectively.

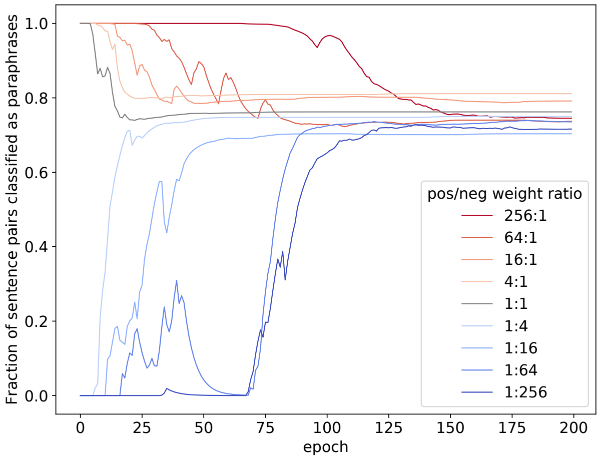


Figure 6. Relationship between early stopping and importance weighting on MRPC dataset. We plot the fraction of sentence pairs in the test set, classified as paraphrases (y-axis) vs training epochs (x-axis) with different importance weights.

(MRPC) (Dolan & Brockett, 2005), where the task is to identify whether or not a pair of sentences have the same meaning (Figure 6). We fine-tune the BERT$_{\text{BASE}}$ model as described in Devlin et al. (2018), except without weight decay and using SGD with a learning rate of 0.01 instead of the Adam optimizer. Our implementation is adapted from Wolf & Sanh (2018). Experiments were run with importance weights of inverse powers of 4 for each class, including with no importance weighting. All classifiers achieved test accuracies between 78% and 85%.

4. Discussion

Our results show that as training progresses, the effects due to importance weighting vanish (Figures 1, 4, 5, 6). While weighting impacts a model's decision boundary early in training, in the limit, models with widely-varying weights appear to approach similar solutions. After many epochs of training, there is no clear correlation between the class-based importance weights and the classification ratios on either test set images, out-of-domain images, or random vectors. When correcting for label shift, IW-ERM confers a benefit early on that dissipates over epochs of training as the impact of the weighting wears off (Figure 5). In the previous section, we noted 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.

We show that these findings hold for both simple convolutional networks trained on images, as well state-of-the-art attention/transformer-based models fined-tuned in a transfer learning scheme to text data (Figures 4 and 6). These effects are present when training on other pairs of CIFAR-10 classes such as cars/trucks (Figure A.8), and continue to hold when models are optimized by the Adam optimizer, although the motivating theory applies to SGD but not ADAM (Soudry et al., 2017) (Figure A.9).

In contrast, sub-sampling the training set instead of down-weighting the loss function, *does* have a noticeable effect on classification ratios. Models assign more CIFAR-10 test images from all classes (both in-domain and out-of-domain) as well as more random noise images to the majority class. (Figure A.6). Notably, weighting the loss function to counteract this imbalance during training does not balance the classification ratios.

One might note that because the true labeling function is deterministic, and because our models are sufficiently expressive to avoid phenomena due to model misspecification, perhaps we should not be surprised that import weighting has no effect. Indeed, for our examples of separable Gaussians and CIFAR images, we have not altered the Bayes optimal predictions. To address the matter, we conducted experiments where CIFAR images are corrupted by label noise. Thus the true classes are no longer separable but owing to the finite sample and the expressive power of deep nets, our training data is still nevertheless separable by our model. Indeed, our experiments showed that under label noise, IW-ERM still does not meaningfully impact classification ratios (Figure A.7).

In all experiments, models with more extreme weighting converge more slowly in decision boundary, and convergence in classification ratio begins to occur long after perfect training accuracy is achieved. For example, the BERT model is typically fine-tuned for 3 or 4 epochs (Devlin et al., 2018), however it took over 100 epochs for the test classification ratios of models with more extreme importance weights to stabilise (Figure 6). Lipton et al. (2018) trained neural networks with importance weights on CIFAR-10 for 10 epochs to correct for test label shift, but we find that it can take up to 200 epochs for some networks to converge in test accuracy.

An effect of importance weighting on classification ratios is present after training ResNet models for 1000 epochs. However, when batch normalization is removed from the model, classification ratios during training resemble those of the ordinary convolutional network (Figure A.5).

We also show that the presence of L2 regularization impacts importance-weighted classifiers (Figure 4). For the synthetic data, both logistic regression and the neural network partition less of the sample space to the down-weighted class (Figure 2). In the CIFAR experiments, L2 regularization slows the convergence in classification ratios of all models (Figure A.3). However, these effects diminish when L2 regularization is replaced with dropout (Figures 3 4, A.4). In this case, the classifiers behave similar to the unregularized models.

5. Related Work

To our knowledge, no previous paper explicitly studies the effects of importance weighting on the decision boundaries learned by modern deep neural networks. In Section 1, we referenced numerous papers applying importance weighting in a variety of contexts to both to classical and deep models and in Section 2 we referenced those papers whose theoretical contributions motivated our study. Here, we briefly recap the most related works.

Theoretical Inspiration Our experiments draw inspiration primarily from the works of Soudry et al. (2017); Gunasekar et al. (2018), who proved that deep linear nets are (importance) weight-agnostic when optimized by SGD to minimize cross entropy loss on (linearly) separable data, and the work of (Shimodaira, 2000) which clearly motivates the efficacy of importance weighting to model misspecification.

Importance weighting and deep learning A number of papers have employed IW-ERM with varying results. Joachims et al. (2018) uses deep networks to learn from logged contextual bandit feedback, and Murali et al. (2016) weight certain training demonstrations in the context of deep imitation learning. Interestingly, Kostrikov et al. (2019) propose an imitation learning algorithm that ought to require importance sampling, but omit it, noting that empirically the algorithm works regardless. Schaul et al. (2016) propose a heuristic algorithm that up-samples experiences from the replay buffer. In the case of domain adaptation, Azizzadenesheli et al. (2019); Lipton et al. (2018) use IW-ERM to correct classifiers to account for label shift. Investigating deep nets for causal inference, (Shalit et al., 2017) weights the loss function to account for the sample size of the treatment group. In curriculum learning (Bengio et al., 2009);(Matiisen et al., 2017);(Jiang et al., 2015), training examples are re-weighted by a teacher during the training process with the objective of improving or accelerating training.

6. Conclusions

Our experiments suggest that effects from importance weighting on deep networks may only occur in conjunction with early stopping, disappearing asymptotically. For these over-parameterized models, capable of fitting any training set, the learned solution may be determined solely by the location of training examples, independent of their density. For example, when correcting for label shift in test data, test accuracy may deteriorate over training epochs even as the classifier improves owing to the diminishing effect of importance weighting. Not only do we fail to find any clear correlation between importance weighting and the fraction of test examples partitioned to each class, but models with different importance weightings also have high agreement even on out-of-domain images, providing further evidence that the learned decision boundaries are similar. Our findings should raise concerns amongst practitioners who might re-evaluate its use on the various problems for which importance weighting is a standard tool.

We find similar patterns across various models (MLPs, convolutional networks, and attention-based transformer networks) and domains (synthetic 2D data, images, and natural language). While importance weighting does appear to have some effect when applied with residual networks we observe that these effects vanish when batch normalization is removed. Batch normalization counteracts the effect of exploding weight norms by normalizing the magnitude of weights for all but the final classification layer. However, in our experiments, we observe that models with batch normalization, still have large final-layer weights, resulting in large logit values after training. Thus we speculate that it may be possible for batch normalization to interact with importance weighting by some other mechanism.

Some effect of importance weighting can be realized when applied in combination with L2 regularization. We believe that in this case, the L2 penalty prevents SGD from reaching the large norm solutions whose loss is dominated by the support vectors, thus preventing convergence to max-margin-like solutions. This aligns with our related finding that dropout, which does not penalize such large-norm solutions, does not affect the fractions of examples partitioned to each class (for importance-weighted classifiers) in the limit.

We find that weighting the loss function of deep networks fails to correct for training set class imbalance. However, sub-sampling a class in the training set clearly affects the network's predictions. This finding indicates that perhaps sub-sampling can be an alternative to importance weighting for deep networks on sufficiently large training sets.

While as previously noted, importance weighting has been shown (empirically) to be useful for deep networks by several others (Lipton et al., 2018; Schaul et al., 2016; Burda et al., 2015), our findings nevertheless support rethinking the standard application of importance weighting in combination with deep learning, suggesting that practitioners should exercise caution when making use of them and raising new questions such as: if importance weighting is only useful for deep networks in conjunction with early stopping or weight decay, then is there a principled way to choose stopping times or weight decay coefficients when importance weighting is desired?

Acknowledgments

We would like to thank Daniel Soudry, Ferenc Huszár, Ben Eysenbach, and the reviewers for providing valuable feedback that has helped to improve the draft. We would also like to thank Amazon, Salesforce, Adobe, the AI Ethics and Governance Fund, and the Center for Machine Learning in Health, whose generous support has helped to advance this line of research on robust machine learning.

References

Azizzadenesheli, K., Liu, A., Yang, F., and Anandkumar, A. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations (ICLR)*, 2019.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *International Conference on Machine Learning (ICML)*, 2009.

Beygelzimer, A., Dasgupta, S., and Langford, J. Importance weighted active learning. In *International Conference on Machine Learning (ICML)*, 2009.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations (ICLR)*, 2015.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

Dolan, B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*, 2005.

Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., and Schölkopf, B. Covariate shift by kernel mean matching. *Journal of Machine Learning Research*, 2009.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. *arXiv preprint arXiv:1806.00468*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Horvitz, D. G. and Thompson, D. J. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association (JASA)*, 1952.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

Jiang, L., Meng, D., Zhao, Q., Shan, S., and Hauptmann, A. Self-paced curriculum learning. In *AAAI Conference on Artificial Intelligence*, 2015.

Joachims, T., Swaminathan, A., and Rijke, M. d. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018.

Kahn, H. and Marshall, A. W. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1953.

Khetan, A., Lipton, Z. C., and Anandkumar, A. Learning from noisy singly-labeled data. *International Conference on Learning Representations (ICLR)*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Koller, D., Friedman, N., and Bach, F. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Kostrikov, I., Agrawal, K. K., Levine, S., and Tompson, J. Addressing sample inefficiency and reward bias in inverse reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2019.

Lipton, Z. C., Wang, Y.-X., and Smola, A. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*, 2018.

Mahmood, A. R., van Hasselt, H. P., and Sutton, R. S. Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Matiisen, T., Oliver, A., Cohen, T., and Schulman, J. Teacher-student curriculum learning. *CoRR*, abs/1707.00183, 2017.

Murali, A., Garg, A., Krishnan, S., Pokorny, F. T., Abbeel, P., Darrell, T., and Goldberg, K. Tsc-dl: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning. In *Robotics and Automation (ICRA)*, 2016.

Precup, D. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, 2000.

Rubinstein, R. Y. and Kroese, D. P. *Simulation and the Monte Carlo method*. John Wiley & Sons, 2016.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2016.

Settles, B. Active learning literature survey. Technical report, University of Wisconsin-Madison, 2010.

Shalit, U., Johansson, F. D., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning (ICML)*, 2017.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 2000.

Soudry, D., Hoffer, E., and Srebro, N. The implicit bias of gradient descent on separable data. In *Inernational Conference on Learning Representations (ICLR)*, 2017.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

Swaminathan, A. and Joachims, T. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning (ICML)*, 2015.

Wolf, T. and Sanh, V. Pytorch pretrained bert. `https://github.com/huggingface/pytorch-pretrained-BERT`, 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.