What is the effect of Importance Weighting in Deep Learning?

Jonathon Byrd, Zachary C. Lipton
Carnegie Mellon University
jabyrd@cmu.edu, zlipton@cmu.edu

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. This work is inspired by recent theoretical results showing that on (linearly) separable data, deep linear networks optimized by SGD learn weight-agnostic solutions, prompting us to 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 models early in training, 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? Our experiments 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 and Thompson, 1952, Kahn and Marshall, 1953, Rubinstein and 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 \(q(x)/p(x)\):

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

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 and 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.

Some applications of weighted risk minimization include propensity score matching in causal inference...
Rosenbaum and Rubin 1983 and correcting for sampling bias in active learning Beygelzimer et al. 2009 Settles 2010. 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. The technique is also frequently employed in off-policy reinforcement learning Precup 2000 Mahmood et al. 2014 Swaminathan and 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 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. Since neural networks are capable of (and indeed do) shatter the training set, 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 binary 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. 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, across epochs, 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 of by which importance weights effect the learned model in terms of a seemingly unrelated quantity: the degree of regularization, prompting the question: how does one choose the appropriate L2 regularization to appropriately incorporate the 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.
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 in practice it is common 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, the 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, yet 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/re-weighting. While Soudry et al. [2017] and Gunasekar et al. [2018]’s analyses only address linear networks and separable data, their findings motivate our hypothesis that a similar weight invariance 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. If L2-regularization prevents such large-norm behavior, what, if anything, do the weights converge to 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 hypothesized that these regularization methods would have different 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 importance weighting on 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 $\frac{1}{2}$. Table 1 shows t-tests comparing importance-weighted classifiers to unweighted classifiers for several experiments.

Synthetic Data In order to visualize model decision boundaries, 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 $\frac{1}{\sigma_{max}(X)}$, where $\sigma_{max}(X)$ is the maximum singular value of the
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.)
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 classified as dogs (y-axis) vs importance weights (x-axis) on a log₂ scale 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 different runs, and lines represent means.

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** 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, then the binary output layer. All hidden layer activation functions are ReLU functions. The models are trained for 1000 epochs using minibatch SGD with a batch size of 16, and no momentum. All models 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 and Ba, 2015] with learning rate $1 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$ (Figure A.6). 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.5 shows results from training the convolutional network model on CIFAR-10 images labeled automobile or truck.

All unregularized models 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. For each importance weighting, we took a majority vote over nine different random seeds. Across 17 different importance weights, all majority votes agreed on the label of 35.5% of cat/dog test images, and at least 16 (88.2%) of the weightings agreed on 75.3% of cat/dog test images. In comparison, when looking at all 153 individual runs, only 49.9% of examples had greater than 16/17 (88.2%) of models agree on the label. Exectly 16/17 differently-weighted models agree on the labels of 98.7% of the randomly-generated images.

Many modern image applications use deep residual networks (ResNets) [He et al., 2016]. We repeat the same CIFAR experiment using a ResNet architecture consisting of a $5 \times 5$ 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, before a dense layer with 512 nodes, then the output layer. Each residual block consists of two $3 \times 3$ convolution layers, and the first layer with 128 filters plus the first layer with 512 filters have stride of 2. All other hyperparameters were left unchanged. Figure 5 shows results both with and without batch normalization [Ioffe and Szegedy, 2015] applied between all convolution layers.

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 (MRPC) [Dolan and Brockett, 2005], where the task is to identify whether or not a pair of sentences have the same meaning. We fine-tune the BERT 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 and Sanh, 2018]. Experiments were run with importance weights of inverse powers of 4 for each class, including with no importance weighting. All models achieved test accuracies between 78% and 85%. Results are shown in Figure 6.

4 Discussion

Our results show that as training progresses, the effects due to importance weighting vanish (Figures 1, 4, 6, Table 1). 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. This is exemplified by the absence of any monotonic relationship between the t-statistics and importance
Figure 5: Relationship between early stopping and importance weighting (a,b), and final classification ratios vs. importance weighting (c,d) for ResNet models on CIFAR with and without batch normalization. Plots are structured as in Figure 4.

Table 1: Welch’s t-tests comparing the fraction of test examples classified as positive in importance-weighted models vs. unweighted models. The left column specifies the whether L2-regularization, dropout, or no regularization is used. The top row specifies the ratio of importance weighting. Each cell reports the corresponding t-statistic and the p-value in parentheses.
weights in Table 1. In the previous section, we noted that not only do differently-weighted CIFAR models converge to similar classification ratios, but they also have high agreement on examples, even random noise images (which, incidentally, are almost exclusively classified as cats), 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 fine-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.5), 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.6).

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).

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 over training resemble those of the first convolutional network (Figure 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. Moreover, in Section 1, we referenced numerous papers applying importance weighting in a variety of contexts to both classical and deep models and in Section 2, we referenced those papers whose theoretical contributions shaped this section, 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 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. In deep reinforcement learning, Joachims et al. [2018] uses deep networks to learn from logged contextual bandit feedback, and deep imitation learning [Murali et al., 2016]. 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 upsamples 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. 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 random noise images, providing further evidence that the learned decision boundaries are similar. Our findings should raise concerns amongst practitioners, addressing 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 neutralizes 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 classification ratios in the limit.

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?

References

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In International Conference on Learning Representations (ICLR), 2019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In International Conference on Machine Learning (ICML), 2009.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In International Conference on Machine Learning (ICML), 2009.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In International Conference on Learning Representations (ICLR), 2015.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805, 2018.

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

Arthur Gretton, Alexander J Smola, Jiayuan Huang, Marcel Schmittfull, Karsten M Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. Journal of Machine Learning Research, 2009.
Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. *arXiv preprint arXiv:1806.00468*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

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

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander Hauptmann. Self-paced curriculum learning. In *AAAI Conference on Artificial Intelligence*, 2015.

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

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

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

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

Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Ilya Kostrikov, Kumar Krishna Agrawal, Sergey Levine, and Jonathan Tompson. Addressing sample inefficiency and reward bias in inverse reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2019.

Zachary C Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*, 2018.

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

Tambet Mattiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *CoRR*, abs/1707.00183, 2017.

Adithyavairavan Murali, Animesh Garg, Sanjay Krishnan, Florian T Pokorny, Pieter Abbeel, Trevor Darrell, and Ken Goldberg. Tsc-dl: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning. In *Robotics and Automation (ICRA)*, 2016.

Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, 2000.
Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 1983.

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

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2016.

Burr Settles. Active learning literature survey. Technical report, 2010.

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

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

Daniel Soudry, Elad Hoffer, and Nathan Srebro. The implicit bias of gradient descent on separable data. In *International Conference on Learning Representations (ICLR)*, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

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

Thomas Wolf and Victor Sanh. Pytorch pretrained bert. [https://github.com/huggingface/pytorch-pretrained-BERT](https://github.com/huggingface/pytorch-pretrained-BERT), 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.
Figure A.1: Results on non-linearly separable moons dataset. The setup is the same as figure

Figure A.2: Results on overlapping Gaussians dataset. The setup is the same as figure
(a) CIFAR10 cat and dog test images.

(b) CIFAR10 test images from the other eight classes.

(c) Random images.

Figure A.3: Effect of early stopping on L2-regularized models. The same setup is used as in Figure 4.

(a) CIFAR10 cat and dog test images.

(b) CIFAR10 test images from the other eight classes.

(c) Random images.

Figure A.4: Effect of early stopping on models with dropout. The same setup is used as in Figure 4.
Figure A.5: Results for training on automobile and truck classes from CIFAR10. Same setup as 4, but without standard deviations over multiple runs.
(a) CIFAR10 cat and dog test images.

(b) CIFAR10 test images from non-cat/dog classes.

(c) Random images.

(d) Classification ratios after training on cat/dog images.

Figure A.6: Results from training a convolutional network with the Adam optimizer with learning rate $1e^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-8}$. The setup and all other model hyperparameters are the same as in \[\text{Fig} \] but without standard deviations over multiple runs.