Effects of Initialization Biases on Deep Neural Network Training Dynamics

Nicholas Pellegrino[1]*, David Szczecina[1,2]*, & Paul Fieguth[1]
[1]Vision and Image Processing Group, Systems Design Engineering, University of Waterloo
[2]Mechanical & Mechatronics Engineering, University of Waterloo
{npellegr,dszczeci,pfieguth}@uwaterloo.ca

Abstract

Untrained large neural networks, just after random initialization, tend to favour a small subset of classes, assigning high predicted probabilities to these few classes and approximately zero probability to all others. This bias, termed Initial Guessing Bias*, affects the early training dynamics, when the model is fitting to the coarse structure of the data. The choice of loss function against which to train the model has a large impact on how these early dynamics play out. Two recent loss functions, Blurry and Piecewise-zero loss, were designed for robustness to label errors but can become unable to steer the direction of training when exposed to this initial bias. Results indicate that the choice of loss function has a dramatic effect on the early phase training of networks, and highlights the need for careful consideration of how Initial Guessing Bias may interact with various components of the training scheme.*

1. Introduction

In the supervised training of deep neural networks, an often-overlooked component of the training is the behaviour of the network before it has been exposed to any labelled data. Recent investigations [1–3] into randomly initialized networks have revealed a surprising and systematic phenomenon: many architectures exhibit a strong preference for a small subset of classes immediately after initialization, assigning high predicted probabilities to these favoured classes and near-zero probabilities to the rest, nearly regardless of the input provided to the model. This effect is demonstrated in Figure 1, whereby one class is favoured. This systematic tendency is termed *Initial Guessing Bias* (IGB) [1], which impacts not only the initial distributions but persists in more subtle ways even after training.

IGB most directly impacts models in the earliest stages of training, just after initialization. During this phase, the loss function plays a particularly critical role: it determines how strongly the model is encouraged to correct its initial biases and how quickly it moves toward a meaningful representation. While standard loss functions such as Cross-Entropy (CE) [4] provide strong gradients even when predicted probabilities are extremely small, recent work [5–15] has introduced alternative objectives designed to improve robustness to label errors.

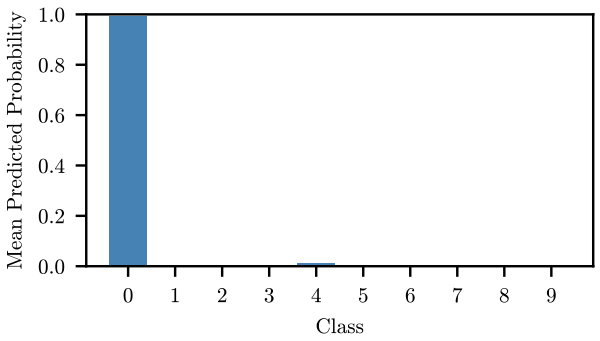

Figure 1. Predicted probability for each class, averaged over the validation set, directly after model initialization. Observe that one class (class 0, randomly) is *highly* favoured ($\bar{p}_0 \approx 1$) relative to other classes ($\bar{p}_y \approx 0$ for $y \neq 0$) as a result of severe Initial Guessing Bias.

Two such objectives — Blurry Loss (BL) and Piecewise-Zero Loss (PZ) [15] — attenuate or eliminate gradients when the model produces low probabilities for the target class. These loss functions were developed to reduce the harmful influence of incorrect labels, examples of which tend to exhibit low predicted probability from models that have generalized, but the behaviour of these losses in the presence of strong initialization-induced biases has not yet been carefully examined. If a loss function supplies strong gradients for low-probabilities (as CE does), the model can quickly correct these biases, but for losses such as BL and PZ, which produce zero or near-zero gradients on low-probability classes, the network may struggle to overcome its initial bias, leading to slow or stalled early learning. Thus, the choice of loss function may determine whether a model escapes the basin carved out by IGB or remains trapped during a critical early window of training.

*Indicates equal contribution, joint first-authorship.

This work studies how IGB interacts with CE, BL, and PZ during the earliest phase of training. To examine how different loss functions respond to these initial biases, the predicted probabilities, $p(y|x)$, and per-class accuracies are monitored batch-to-batch throughout training. CE results in a re-balancing effect whereby the initially dominant class is corrected downward while the suppressed classes rise, until all predicted probabilities tend to move as a group towards higher values. BL produces similar outcomes, but at a decreased rate; however, PZ does not allow classes with low predicted probability to contribute meaningfully to training, resulting in the initially favoured class dominating throughout.

These findings suggest that initialization interacts deeply with the choice of loss function. For robust training pipelines, designed to handle label errors, the combination of strong IGB and gradient-suppressing loss functions may lead to extremely slow convergence or complete failure to begin learning. These results underscore the importance of understanding how initialization, loss design, and early-stage optimization jointly shape training trajectories, especially in scenarios where robustness is a priority.

2. Background

2.1. Initial Guessing Bias (IGB)

Recent work [1–3] has shown that untrained neural networks exhibit a systematic and architecture-dependent Initial Guessing Bias (IGB) [1]. Immediately after random initialization, the predicted class distribution is far from uniform: instead, the model tends to assign disproportionately high probability to a small subset of classes while assigning nearly zero probability to the majority. This effect is consistent across a range of architectures and arises from subtle asymmetries introduced by weight initialization, activation patterns, and network depth. IGB therefore determines the "starting point" from which training trajectories begin, and can lead to more subtle biases that persist after training.

2.2. Loss Functions

In supervised classification, models are typically trained using the Cross-Entropy (CE) [4] loss, which encourages predicted probability distributions, $p_y = p(y|x)$, to match one-hot ground truth labels. For an input–label pair (x, y), the CE loss is defined as

$$\mathrm{CE}(p_y) = -\log(p_y). \tag{1}$$

CE provides large corrective gradients when the predicted probability of the true class is small, making it most sensitive to examples that are not being correctly classified.

Blurry Loss (BL) and Piecewise-Zero Loss (PZ) [15] were introduced to mitigate the influence of mislabelled data by down-weighting or eliminating gradients when the predicted probability of the target class is below a threshold.

Motivated by Focal Loss [16], which emphasizes difficult-to-classify examples, *Blurry Loss* was designed to de-emphasize such samples through the inclusion of a multiplicative factor with weighting parameter γ. The Blurry Loss is defined as

$$\mathrm{BL}(p_y) = -\,p_y^{\gamma} \cdot \log(p_y). \tag{2}$$

Note that BL has a region of *positive* gradient for $p_y < e^{-1/\gamma}$, steering training *against* away from the labelled class if p_y is sufficiently low. At $\gamma = 0$, Blurry Loss is equivalent to Cross Entropy Loss.

Piecewise-zero Loss was designed to ignore difficult-to-classify samples (those with low p_y) under the assumption that these are likely mislabelled. If predicted probability is beneath some cutoff, $p_y \leq c \in [0, 1]$, a loss of zero is assigned (with zero gradient, not impacting training). Piecewise-zero Loss is defined as

$$\mathrm{PZ}(p_y) = \begin{cases} 0 & p_y \leq c, \\ \mathrm{CE}(p_y) = -\log(p_y) & p_y > c. \end{cases} \tag{3}$$

Piecewise-zero Loss is equivalent to Cross-Entropy Loss for $c = 0$.

2.3. Addressing Label Errors in Training Data

In many real-world datasets, labels are corrupted by human error, automated annotation artifacts, or inherent class ambiguity [17]. Mislabelled data can mislead the model early in training, causing it to over-fit incorrect labels and degrade generalization [18–20]. Classical CE loss is known to be sensitive to mislabelled examples because it most heavily penalizes low predicted probabilities on the (possibly incorrect) targets. As a result, there has been recent interest [5–15] in designing robust loss functions, such as BL and PZ [15] described in Equations (2) and (3) in Section 2.2, that suppress the influence of potentially wrong labels.

3. Method

The interaction between Initial Guessing Bias and different loss functions are studied by examining the early training dynamics for a standard convolutional architecture on a simple, well-understood dataset. The experiments use a ResNet-50 [21], modified so that the first convolutional layer accepts single-channel input, with Kaiming He initialization [22], the current standard in machine learning and implemented as default in PyTorch. Training is performed on the CIFAR-10 [23] dataset. Three loss functions are considered: standard Cross-Entropy (CE) [4], Blurry Loss (BL), and Piecewise-Zero Loss (PZ) [15]. All runs

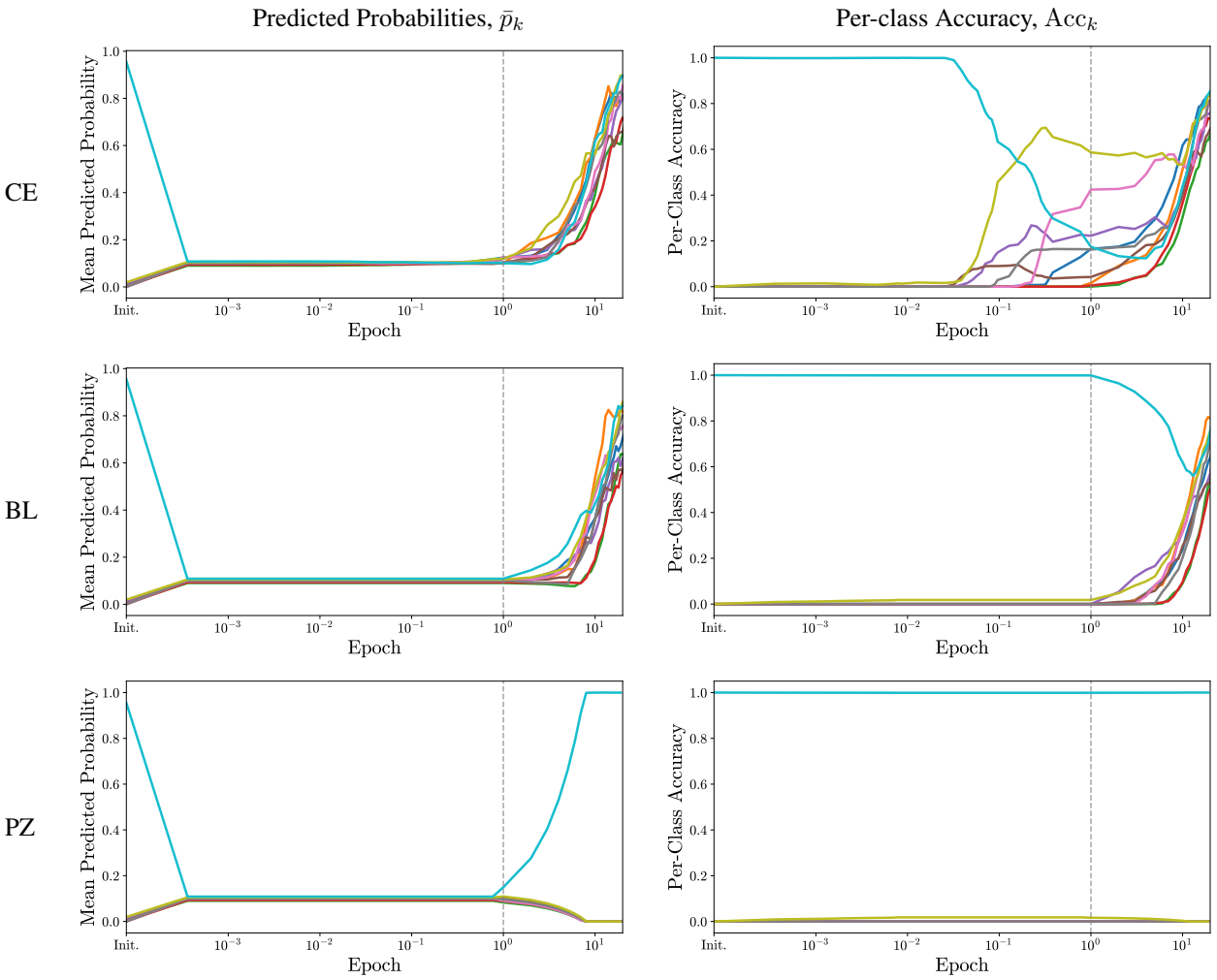

Figure 2. Training dynamics for three choices of loss function: Cross-Entropy (CE; top row), Blurry Loss (BL; second row), and Piecewise-zero Loss (PZ; bottom row). Averaged Softmax probabilities (left column) and per-class accuracy (right column) are shown throughout training. Training duration is plotted on a log-scale, starting at model initialization ("Init."; before training) and showing fractional parts of the first epoch (batches). At the outset, the IGB effect causes probabilities and accuracies to be highly distinct between favoured and unfavoured classes for all loss functions (see Figure 1); however, as training progresses differences emerge. In all cases, predicted probabilities rapidly move towards each other during the first batch and converge roughly to $p_y = 0.1$ (approximating the class distribution of the dataset). However, the subtle differences in the per-class predicted probabilities leads to larger changes in the per-class accuracies. By the end of the first epoch, differences resulting from the choice of loss function begin to reveal themselves. For Cross-Entropy, the predicted probabilities move as a group, gradually rising, with corresponding rises in per-class accuracies. Blurry loss behaves fairly similarly to Cross-Enropy, but with slightly slower dynamics and with accuracies lagging. For Piecewise-zero loss, predicted probability for the originally favoured class remains slightly above all others, and again rises, while all others fall. In this case, the resulting per-class accuracies remain largely unchanged.

use the same method of random initialization and identical training settings (batch size of 32, using Stochastic Gradient Descent optimization [24], and 20 training epochs) to isolate the effect of the loss function. PZ loss uses a cut-off parameter setting of $c = 0.1$. In [15], loss scheduling is normally used with PZ loss (initially using CE loss during the first d epochs before switching to PZ loss) to avoid having many *correctly* labelled samples be ignored before the model has sufficient time to generalize; however, here, PZ loss is immediately applied ($d = 0$), specifically to understand the adverse effects of arising from IGB. BL loss uses a weighting parameter setting of $\gamma = 0.7$.

Prior to studying training dynamics, the presence and severity of IGB is characterised for the ResNet-50 [21] architecture by visualizing the initial predicted probability distributions.

To characterize early training behaviour, two quantities are recorded:

1. **Average Predicted Probability,** $\bar{p}_k$:
   Softmax probabilities, $p(y = k|x)$, are averaged over samples of each class, k, to track comprehensive model behaviour throughout training. Samples of a given class, k, form a class subset $S_k = \{(x_i, y_i)|y_i = k\}$, such that the averaged Softmax probability is calculated as

$$\bar{p}_k = \frac{1}{|S_k|} \sum_{(x_i,y_i)\in S_k} p(y_i = k|x_i). \quad (4)$$

2. **Per-class Accuracy,** Acc_k:
   Validation accuracy, computed separately for each class, k, tracks model performance relative to each class and is calculated as

$$\mathrm{Acc}_k = \frac{1}{|S_k|} \sum_{(x_i,y_i)\in S_k} \mathbb{1}\{\arg\max_j p(j|x_i) = k\}. \quad (5)$$

These metrics are computed directly after model initialization and then during training at the end of every batch during the first epoch and at the end of each epoch thereafter.

4. Results and Discussion

Directly after initializing the ResNet-50 model, predicted probabilities were collected and averaged over the validation set of CIFAR-10, demonstrating the severity of the Initial Guessing Bias in Figure 1. Observe that one class (class 0, randomly) is *highly* favoured ($\bar{p}_0 \approx 1$) relative to other classes ($\bar{p}_y \approx 0$ for $y \neq 0$) indicating a high degree of bias.

Figure 2 shows the early-stage training dynamics for the three loss functions explored. Results for each loss function are plotted on separate rows, with predicted probabilities and per-class accuracy shown in the left and right columns, respectively. Note that training duration is plotted on a log-scale, starting at model initialization ("Init."; before training) and showing fractional parts of the first epoch (batches).

During the first epoch (before the 10^0 tick mark and dashed vertical line), predicted probabilities rapidly converge towards $p_y = 0.1$; however, per-class accuracy lags, remaining unchanged for longer, as correct classification occurs only for samples (x_i, y_i) in which $p(y_i|x_i)$ and is maximal (uncommon for $\bar{p}_k \approx 0.1$). Because Blurry and Piecewise-zero loss de-emphasize samples with low p_y, in this regime, very few samples have any sizable influence on model training. In contrast, Cross-Entropy is most sensitive to samples with low p_y, leading to earlier increases in predicted probabilities accuracy across all classes. While Blurry loss lags behind Cross-Entropy, for Piecewise-zero Loss there is insufficient training signal from samples of the non-favoured class, and training is almost entirely driven by the favoured class, leading to its rebound in predicted probability and constant perfect accuracy (while accuracy for all other classes remains poor).

Note that while the two robust loss functions de-emphasize (or entirely zero) the impacts of samples with low p_y, the training signal from samples (even just from one class) with high p_y may steer the model towards increased p_k for all classes.

5. Conclusion

Initial Guessing Bias strongly influences early-stage training, with impacts that can persist to the end of training. The choice of loss function plays a key role in how the model responds to this bias. Cross-Entropy quickly reduces the imbalance and improves accuracy across all classes, while Blurry Loss achieves similar effects more slowly. Piecewise-Zero Loss provides too little signal for classes with low predicted probability, allowing the initially favoured class to dominate and preventing meaningful learning from the other classes. These findings underscore the importance of accounting for initialization biases and exercising care when selecting loss functions and other components of the training scheme.

Acknowledgments

This research was enabled in part by support provided by Calcul Québec (calculquebec.ca) and the Digital Research Alliance of Canada (alliancecan.ca).

We acknowledge the support of the Government of Canada's New Frontiers in Research Fund (NFRF), [NFRFT-2020-00073], and the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) via NSERC-CGS D.

Nous remercions le Fonds Nouvelles Frontières en Recherche du gouvernement du Canada de son soutien (FNFR), [FNFRT-2020-00073], et le soutien du Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRSNG), CRSNG-BESC D.

Natural Sciences and Engineering Research Council of Canada | Conseil de recherches en sciences naturelles et en génie du Canada | Canada

References

[1] E. Francazi, A. Lucchi, and M. Baity-Jesi, "Initial guessing bias: How untrained networks favor some classes," in *International Conference on Machine Learning*, pp. 13783–13839, PMLR, 2024. 1, 2

[2] E. Francazi, F. Pinto, A. Lucchi, and M. Baity-Jesi, "Where you place the norm matters: From prejudiced to neutral initializations," *arXiv preprint arXiv:2505.11312*, 2025.

[3] A. Bassi, C. Albert, A. Lucchi, M. Baity-Jesi, and E. Francazi, "When the left foot leads to the right path: Bridging initial prejudice and trainability," *arXiv preprint arXiv:2505.12096*, 2025. 1, 2

[4] I. J. Good, "Rational decisions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 14, no. 1, pp. 107–114, 1952. 1, 2

[5] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018. 1, 2

[6] X. Ye, X. Li, T. Liu, Y. Sun, W. Tong, *et al.*, "Active negative loss functions for learning with noisy labels," *Advances in Neural Information Processing Systems*, vol. 36, pp. 6917–6940, 2023.

[7] A. Ghosh, N. Manwani, and P. Sastry, "Making risk minimization tolerant to label noise," *Neurocomputing*, vol. 160, pp. 93–107, 2015.

[8] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[9] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, "Normalized loss functions for deep learning with noisy labels," in *International conference on machine learning*, pp. 6543–6553, PMLR, 2020.

[10] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 322–330, 2019.

[11] Y. Lyu and I. W. Tsang, "Curriculum loss: Robust learning and generalization against label corruption," in *International Conference on Learning Representations*, 2019.

[12] X. Zhou, X. Liu, J. Jiang, X. Gao, and X. Ji, "Asymmetric loss functions for learning with noisy labels," in *International conference on machine learning*, pp. 12846–12856, PMLR, 2021.

[13] A. K. Menon, A. S. Rawat, S. J. Reddi, and S. Kumar, "Can gradient clipping mitigate label noise?," in *International conference on learning representations*, 2020.

[14] T. Pang, C. Du, Y. Dong, and J. Zhu, "Towards robust detection of adversarial examples," *Advances in neural information processing systems*, vol. 31, 2018.

[15] N. Pellegrino, D. Szczecina, and P. Fieguth, "Loss functions robust to the presence of label errors," *Presented at the 10th Annual Conference on Vision and Intelligent Systems*, 2024. arXiv preprint arXiv:2511.16512v1. 1, 2, 3

[16] T. Lin, "Focal loss for dense object detection," *arXiv preprint arXiv:1708.02002*, 2017. 2

[17] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," *Advances in Neural Information Processing Systems*, 2021. 2

[18] G. Algan and I. Ulusoy, "Image classification with deep learning in the presence of noisy labels: A survey," *Knowledge-Based Systems*, vol. 215, p. 106771, 2021. 2

[19] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 11, pp. 8135–8153, 2022.

[20] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1373–1411, 2021. 2

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 2, 3

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015. 2

[23] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep. 0, University of Toronto, Toronto, Ontario, 2009. 2

[24] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, 2010. 3