Overview of different loss functions to improve robustness in gesture recognition

SHEETAL BORAR, Aalto University, Finland

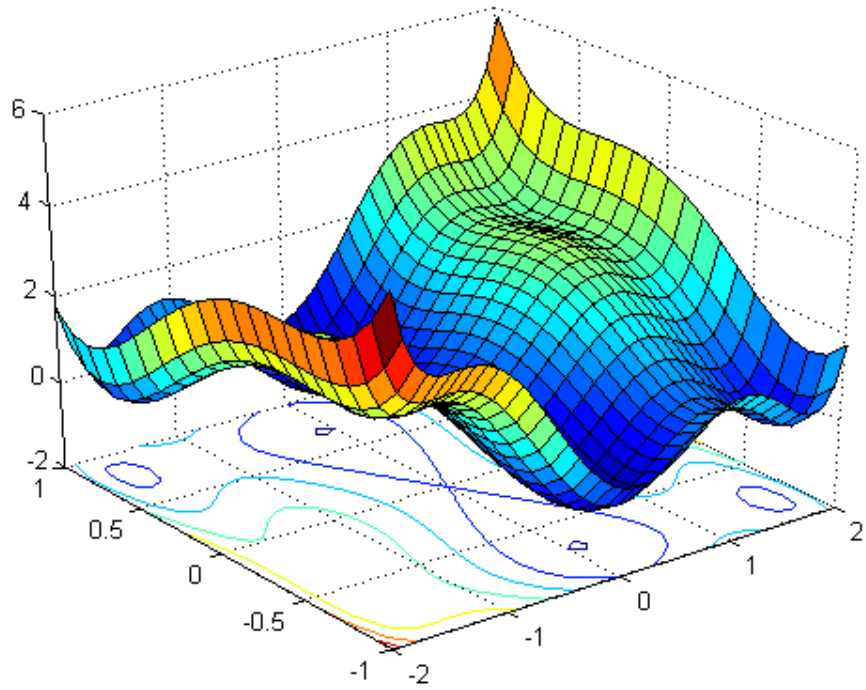

Fig. 1. Loss Functions

Abstract: Typically, loss functions are fixed or chosen heuristically from a limited set of options in machine learning tasks. Multiple research studies have evaluated the role of different loss functions on model robustness, training time along with other metrics. This research aims to extend these studies and provide an empirical evaluation of how nine different loss functions can impact training accuracy for a complex image classification task. The results show that higher order hinge losses and expectation losses provide the highest accuracy gains and the paper describes the possible reasons behind these observations.

Author's address: Sheetal Borar, sheetal.borar@aalto.fi, Aalto University, Espoo, Finland, 02150.

XXXX-XXXX/2021/6-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS Concepts: • **Human-centered computing → HCI theory, concepts and models**.

Additional Key Words and Phrases: machine learning, deep learning, loss functions, robustness, gesture recognition, multi-class classification, image classification

ACM Reference Format:
Sheetal Borar. 2021. Overview of different loss functions to improve robustness in gesture recognition. 1, 1 (June 2021), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With machine learning applications becoming commonplace in the industry, the robustness of ML systems is becoming more and more imperative. In today's world, machine learning is used ubiquitously in facial recognition, handwriting recognition, diction, and many other applications. In the last decade, many studies have shown that these systems are vulnerable to many issues like adversarial training samples, noisy or biased input data that can lead to unexpected results in production. In this research, I have focused on evaluating various loss functions to improve the robustness of smartwatch gesture recognition. The research provides an empirical study of how different loss functions impact the accuracy of a model trained to differentiate between gestures by using spectrogram images generated from IMU sensor data. It builds on top of the work published by Janocha et al [5] which evaluated different loss functions for simple image classification tasks. The overall aim of the project was to replicate the performance and improve the robustness of a smartwatch gesture recognition experiment conducted by Laput et al [7] and adapt it to Huawei smartwatches. This project was done in collaboration with Aalto University and Huawei HCI Lab in Munich.

In supervised learning, loss functions are commonly used to compute the difference between the expected output and the predicted output. The loss value is used to update the model iteratively

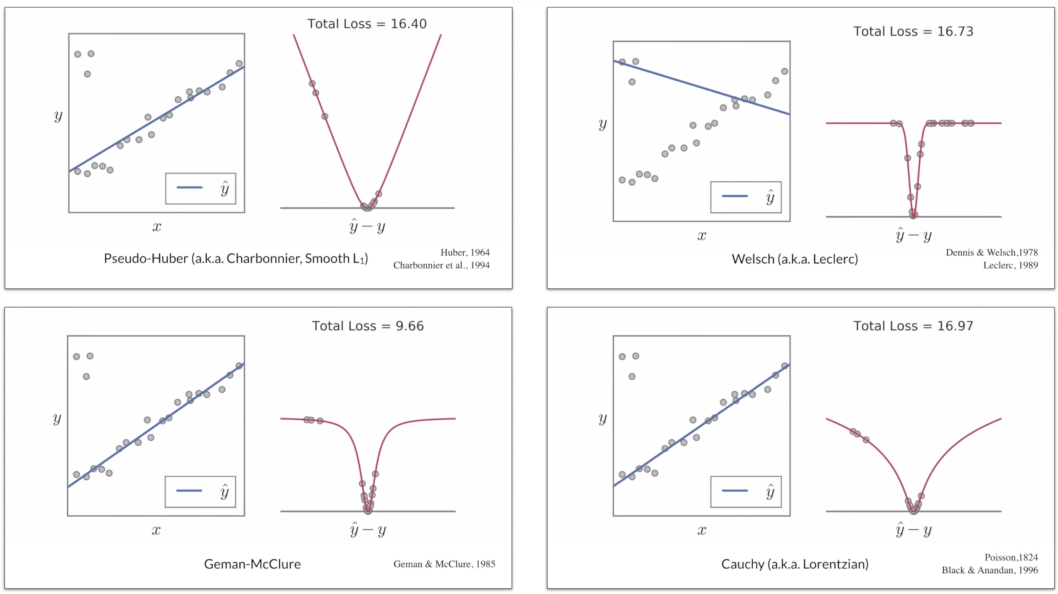

Fig. 2. How different loss functions respond to noisy samples [1]

and help it predict better by reducing the loss. Cross-Entropy Loss has been established as the most popular loss function used in multi-class classification tasks due to its speed of convergence in comparison to other loss functions. Different loss functions provide a different optimization landscape due to which some can be more influenced by noisy input samples in comparison with others. This is shown in Figure 1 from [1]. Hence, using loss functions well suited to the data can potentially help improve robustness through either ignoring noisy samples, label noise, improving regularization, or maximizing an objective function that is closer to the training task goal.

In this research paper, we will describe various loss function that can be used with image classification tasks and their properties, share the experimental results obtained by using these loss functions while training a gesture recognizer using spectrogram images, and conclude with a discussion of the results and remarks for future work.

2 RELATED WORK

Various studies have proposed novel loss functions, new approaches for learning the loss function and provided empirical analysis of the affect of different loss functions.

Janocha et al. [5] experimented with eight different loss functions on image classification tasks, evaluated their impact on model robustness and explained the intuition behind the empirical findings. They introduced some loss functions like expectation losses (L1, L2 norm) which can be used to optimize the difference in the classification probability between the target data and the model outputs. They also tested which loss functions were robust to noise by adding Gaussian noise to input samples. Expectation Losses and higher-order hinge losses reported more robustness in comparison with the standard log loss. The reason behind the increased robustness was that a single misclassified example has an enormous impact on the loss surface of log loss whereas, it does not impact these loss functions so significantly. Kumar et al [6] proposed a loss function based on MAE (mean absolute error) rather than mean squared error which can help with risk minimization and would be more robust to label noise.

Barron et al. [1] presented a robust adaptive loss function that can adapt the loss function surface and formula according to the data during training. They presented a general formula, which is a superset of several loss functions like Cauchy Schwarz Divergence loss, and Welsh Loss, with a parameter which can be learned while training. Furthermore, there have been more developments in the area of loss meta-learning. Several other algorithms have been proposed to learn loss functions. Bechtle et al [2] presented a framework to learn a parametric loss function during training. The framework involves two loop wherein the loss gets trained in the outer loop and is used by the inner loop to train the model parameters. Gonzalez et al. [4] developed a genetic loss optimization algorithm that takes units like 1,-1, x, y, log(o), o^2 along with operators like +,-,/,*, generates a large number of loss functions and runs a genetic algorithm to generate optimal loss functions on the MNIST image classification task. They presented a loss function called the Baikal loss function which achieved higher accuracy and faster training than log loss on MNIST and could be transferred to the CIFAR classification task as well.

Most of these previous works have evaluated loss functions on simple image classification problems like CIFAR and MNIST. This research aims to compare and empirically evaluate the performance of these different loss functions on a more complex image classification task.

3 LOSS FUNCTION PROPERTIES

These are the symbols used in the following loss function formula -

- y is true label as one-hot encoding.
- $\hat{y}$ is true label as +1/-1 encoding.
- o is the output of the last layer of the network.
- j denotes jth dimension of a given vector.
- $\sigma(.)$ denotes probability estimate.

3.1 Log Loss

Log loss measures how close the prediction probability is to the true label. If, the prediction probability diverges from true label then log loss is higher.

$$-\sum_{j} y^j log(\sigma(o^j))$$

3.2 Expectation Loss

Expectation losses are calculated between the one-hot encoding of the label and the output of the last layer of the network. L1 and L2 losses help minimize the misclassification probability and are better suited to improve robustness in comparison with log loss, as log loss only tries to maximize the probability of completely correct labeling. They are expected to converge at a slower rate and hence, are generally avoided. This is generally due to the vanishing gradient issue in heavily misclassified examples.

$$Expectation\ Loss(L1) : \sum_{j} \|y - \sigma(o)\|_1$$

$$Regularized\ Expectation\ Loss(L2) : \sum_{j} \|y - \sigma(o)\|_2^2$$

3.3 Baikal Loss

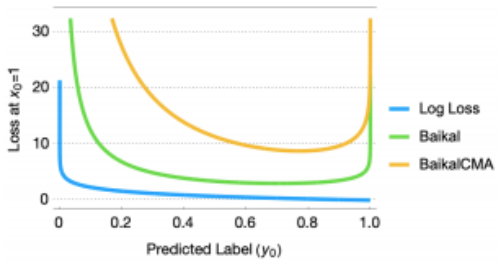

Fig. 3. Log Loss vs Baikal Loss from [4]

Baikal loss was proposed by Gonzalez et al. [4]. It was generated by using a genetic optimization algorithm along with units like log(o), o^2, $\sqrt{o}$, +, , , ÷, x, y, 1, 1, where x represents a true label, and y represents a predicted label.

The authors tested the loss functions against each other on simple image classification tasks like MNIST and CIFAR. They further generated children of the best performers iteratively to get the best loss functions. Baikal loss surface mostly matches the log loss surface shown in Figure

3. When the model predicts anything with high confidence, then the baikal loss function reports a larger error. This property creates an implicit regularization, making training more robust by preventing overconfidence in the models. Baikal loss function produced faster training and higher accuracy than log loss according to the results published by Gonsalez et al [4]

$$\sum_j log(\sigma(o)^j) - \frac{y^j}{\sigma(o)^j}$$

3.4 Hinge Loss and its higher orders

Hinge loss is less sensitive to exact probabilities than log loss and offers higher robustness to outliers. [5] Squared and Cube hinge loss reported faster convergence and better performance. This may be due to implicit regularization in Margin Losses. Higher-order hinge losses would penalize misclassified examples more severely. Margin-based losses have also previously outperformed other loss families in terms of generalization ability.

$$Hinge\ Loss : \sum_j max(0, \frac{1}{2} - \hat{y^j}.\sigma(o^j))$$

$$Squared\ Hinge\ Loss : \sum_j max(0, \frac{1}{2} - \hat{y^j}.\sigma(o^j))^2$$

$$Cube\ Hinge\ Loss : \sum_j max(0, \frac{1}{2} - \hat{y^j}.\sigma(o^j))^3$$

3.5 Cauchy Schwarz Divergence Loss

Cauchy Schwarz Divergence Loss is a measure of how divergent two probability distributions are [3]. Janocha et al [5] theoretically proved that this measure is equivalent to cross-entropy loss, which is regularised with half of expected Renyi's quadratic entropy of the prediction. Furthermore, they have shown empirically that on simple image classification tasks like CIFAR and MNIST, Cauchy Schwarz Divergence Loss consistently gave a better performance with respect to log loss.

$$-log\frac{\sum_j \sigma(o)^j y^j}{\|\sigma(o)\|_2 \|y\|_2}$$

3.6 Robust Adaptive loss

Robust adaptive loss function is a superset of loss functions like : L2 loss (= 2), Charbonnier loss (= 1), Cauchy loss (= 0), Geman-McClure loss (= 2), and Welsch loss (= ∞). As the value of alpha changes, the loss surface changes as well. This method allows us to generate a loss function that accommodates the noise in the input data and generates a loss function that best fits the training objective. Figure 4 shows how the change in the value of α changes the shape of the loss function.

$$\sum_j \frac{|\alpha - 2|}{\alpha}((\frac{(\frac{\|y^j - \sigma(o^j)\|_1}{c})^2}{|\alpha - 2|} + 1)^{\frac{\alpha}{2}} - 1)$$

4 ORIGINAL EXPERIMENT

4.1 Data

We are using the data collected by Laput et al. [7]. They collected IMU data from the accelerometers in ViBand for 25 hand actions and activities. We used the spectrograms generated from the IMU

data for classification. The data was collected for 12 participants over 4 rounds. The model was trained using the following architecture which was presented in the paper and further adapted by the researchers at Aalto User Interfaces lab -

> VGG16 architecture with the following unit - a convolution layer followed by batch-normalization, ReLU, and a max-pooling layer. This was combined with a dropout layer at the end of the second fully connected layer to avoid over-fitting.

4.2 Validation Methods

Laput et al. [7] performed the following cross-validation tests to evaluate the model -

2. Per-user - Every users' data from round 1,2 was used to predict that user's gestures from round 3.
1. Post-removal - The user removes the watch and wears it again in the middle of the training. This validation is useful to test robustness towards changes in orientation and tightness of the watch.
3. Leave-one-user-out - One user's data from all four rounds is kept as a hold-out set and the model is trained on the other users' data.
4. All users - All the users' data from rounds 1,2,3 is used to predict the gestures in round 4.

These validation tests were performed to better evaluate the robustness of the models to different users and environments.

One of the observations made by Laput et al. [7] was that post-removal accuracy is significantly lower (6% lower) than the best results obtained on the per-user accuracy. This decrease in accuracy shows that the collected data or the training methods were not robust to changes in hand or wrist orientations. Furthermore, leave-one-user (79.2%) is much lower than per-user accuracy (95.2%), which might be due to the lower generalization power of the models over multiple users.

For this research, we have only used Post removal, Per user, Leave-user-out as these methods are sufficient to test the robustness of the loss functions.

5 RESULTS

Different loss functions have helped improve the performance in different validation tasks. For example, L1 expectation loss helped improve the accuracy for the per-user validation from 93.88% to 94.54% which was a 0.66% increase over the models trained with log loss. Cube Hinge Loss helped improve the accuracy in the Post-removal validation from 90.14 to 90.70% which was a 0.56% improvement over the baseline. Furthermore, Cube Hinge Loss also helped reduce the standard deviation of the accuracy from 10.85% to 8.81% in the post-removal validation. Surprisingly, the two loss functions which had reported accuracy gains in [5] - squared hinge loss and L2 were not the best performers. Baikal loss helped improve the leave-user-out accuracy to 82% which is a 0.66% improvement over the rest. The rest of the results have been reported in table 1.

Table 1. Results

Loss Function	Per-user	Post-Removal	Leave-one-user-out
Baseline using Log loss	93.88% ± 3.67%	90.14% ± 10.85%	81.83% ± 7.25%
Baikal Loss	93.36% ± 3.54%	89.12% ± 12.99%	**82.49**% ± **6.97**%
Hinge Loss	93.24% ± 3.96%	90.37% ± 10.91%	81.18% ± 7.02%
Squared Hinge Loss	93.75% ± 3.75%	90.27% ± 10.60%	81.21% ± 7.28%
Cube Hinge Loss	93.80% ± 3.26%	**90.95**% ± **8.81**%	81.35% ± 7.17%
L1 Expectation Loss	**94.54**% ± **3.52**%	90.70% ± 10.50%	80.95% ± 7.61%
L2 Expectation Loss	93.71% ± 3.60%	89.95% ± 10.90%	81.23% ± 7.96%
Cauchy Schwarz Divergence Loss	93.10% ± 3.48%	89.26% ± 12.16%	81.08% ± 7.23%
Robust Adaptive Loss	94.29% ± 3.79%	90.40% ± 9.85%	80.92% ± 7.68%

Table 2 shows the range of accuracy mean and standard deviation for different validation tasks. We can observe that experimenting with various loss functions resulted in a difference of up to 1.83% in accuracy for some validation tasks along with a 3.14% difference in standard deviation.

Table 2. Range of accuracy and standard deviation caused by using different loss functions

Loss Function	Per-user	Post-Removal	Leave-user-out
Accuracy	1.44%	1.83%	1.57%
Standard deviation	0.7%	3.14%	0.99%

6 CONCLUSION

In this work, we evaluated nine different loss functions on a gesture recognition image classification task. Empirical results showed that Cube hinge loss, L1 expectation loss give the best performance for image classification tasks based on spectrograms. Further analysis reveals that adding log transformation to input data will make log loss perform better on the data than the other losses, as the log transformation changes the input data distribution significantly. It might be useful to further analyze and study how certain transformations may impact the accuracy of the models trained by different loss functions. Furthermore, the overall difference in accuracy mean and standard deviation caused by changing the loss functions is significant only for some validation tasks. It might be useful to further experiment with other modules like the activation function and optimizer to evaluate whether changing those can result in any changes in the accuracy.

ACKNOWLEDGMENTS

I would like to extend my gratitude to Professor Antti Oulasvirta, head of the User Interfaces lab at Aalto University, for giving me an opportunity to work on this project and providing me his guidance and supervision through the project. I would also like to thank Ilhan Aslan, Huawei HCI Lab, Munich, for funding this research and for his constant guidance through the project.

REFERENCES

[1] Jonathan T. Barron. 2019. A General and Adaptive Robust Loss Function. arXiv:1701.03077 [cs.CV]
[2] Sarah Bechtle, Artem Molchanov, Yevgen Chebotar, Edward Grefenstette, Ludovic Righetti, Gaurav Sukhatme, and Franziska Meier. 2021. Meta-Learning via Learned Loss. arXiv:1906.05374 [cs.LG]
[3] Wojciech Marian Czarnecki, Rafał Józefowicz, and Jacek Tabor. 2015. Maximum Entropy Linear Manifold for Learning Discriminative Low-dimensional Representation. arXiv:1504.02622 [cs.LG]

[4] Santiago Gonzalez and Risto Miikkulainen. 2020. Improved Training Speed, Accuracy, and Data Utilization Through Loss Function Optimization. arXiv:1905.11528 [cs.LG]

[5] Katarzyna Janocha and Wojciech Marian Czarnecki. 2017. On Loss Functions for Deep Neural Networks in Classification. *CoRR* abs/1702.05659 (2017). arXiv:1702.05659 http://arxiv.org/abs/1702.05659

[6] Himanshu Kumar and P. S. Sastry. 2018. Robust Loss Functions for Learning Multi-class Classifiers. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 687–692. https://doi.org/10.1109/SMC.2018.00125

[7] Gierad Laput and Chris Harrison. 2019. Sensing Fine-Grained Hand Activity with Smartwatches. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300568