Proceedings of CONF-CIAP 2026 Symposium: Applied Mathematics and Statistics
DOI: 10.54254/2753-8818/2026.CH31855

Comparing Regularization Techniques for Overfitting in CNN-Based Music Genre Recognition

Qiaomu Liu

Bayview Secondary School, Richmond Hill, Canada
jacky.liu080@gmail.com

Abstract. A key component of music information retrieval (MIR) is music genre classification, which has several uses, including automatic labelling and recommendation systems found on numerous music applications. Considering previous studies that exhibited overfitting, this study compares three widely used regularization techniques—Dropout, L2-Regularization, and Batch Normalization—to evaluate their effectiveness in minimizing overfitting. The GTZAN dataset was used, where the raw audio waveforms were converted into log-scaled Mel-spectrograms, which were used as the input to the convolutional neural network (CNN) model. A baseline CNN model and three regularized variants were trained and evaluated based on classification test accuracy, the generalization gap, and the generalization ratio. The results indicate that Batch Normalization was the most effective strategy, achieving the highest test accuracy of 75.00%. In contrast, L2-Regularization provided moderate improvement compared to the baseline model. At the same time, using Dropout by itself made the model worse at handling new songs, because randomly removing neurons interfered with learning stable patterns in how the frequency changes over time in spectrograms made from the original audio data. These results indicate that the effectiveness of regularization methods in audio-based CNNs depends strongly on the specific technique employed, with Batch Normalization offering the most promising approach for enhancing generalization in CNN-based music classification systems.

Keywords: Music genre classification, Mel-spectrograms, Convolutional neural networks, Overfitting.

1. Introduction

Music genre classification is a fundamental task in music information retrieval (MIR) with a variety of applications, such as automated tagging and recommendation systems seen in many musical platforms. Convolutional Neural Networks (CNNs) have emerged as a powerful tool for this task, as they are good at finding and understanding patterns at different levels of detail once a picture is fed to them. Previous studies have shown that they outperform other machine learning models such as support vector machines (SVMs), XGB (eXtreme Gradient Boosting), and random forests [1]. Since audio spectrograms contain visual representations of sound and show how frequencies (pitch) and their intensity (loudness) change over time, they match closely in terms of their characteristics to those of RGB images [2]. As such, many deep learning models used in computer vision, specifically

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2026.CH31855

CNNs, can also be used for music genre classification [2]. However, previous interactions with CNNs on Mel-spectrograms have shown the deep learning model to be highly susceptible to the issue of overfitting. This issue occurs when a model learns noise and specific patterns from the training data that do not generalize to unseen data, leading to a significant performance gap.

To combat overfitting, various regularization techniques have been developed. As shown in Figure 1, Dropout randomly deactivates a subset of neurons during training, preventing neurons from becoming overly dependent on each other to make correct predictions. Past experimentations by Srivastava et al demonstrated that for all data sets, Dropout helped improve generalization performance [3].

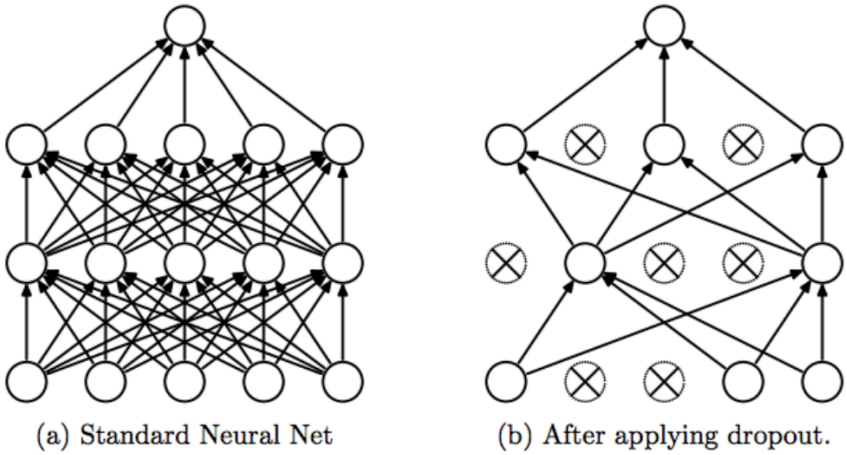


Figure 1. A simple model demonstrating the Dropout neural network. Reproduced from [3]

Another way of reducing overfitting is through the use of L2-Regularization (weight decay), which adds a penalty to the loss function that is proportional to the square of each weight's value. As explained in [4], this penalty prevents the neural network from assigning too much significance to a single feature, thereby encouraging a simpler and more generalizable model. Moreover, Batch Normalization is another regularization technique that standardizes intermediate activations within each mini-batch so that they have zero mean and unit variance. This stabilizes and accelerates training while also having a regularizing effect. Studies from Sergey Ioffe and Christian Szegedy have shown that using a collection of batch-normalized neural networks, they were able to reach 4.82% top-5 test error, which exceeds the accuracy of any human raters, thus improving upon the best published results on ImageNet classification [5]. While these techniques are well-established in computer vision, their comparative effectiveness in the domain of audio-based CNNs has not been thoroughly explored.

This paper aims to conduct a comparative study of Dropout, L2-Regularization, and Batch Normalization. This paper investigates their individual impact on overfitting within an identical CNN architecture trained on the GTZAN dataset. The primary research question is: Which of these three techniques is most effective at reducing overfitting in a CNN-based music genre classification system? The code supporting the findings of this study is available in the following GitHub repository: https://github.com/JLboosh/Music_genre_classification_model_code.

DOI: 10.54254/2753-8818/2026.CH31855

2. Music genre classification method

2.1. Dataset and preprocessing

This study utilized the GTZAN dataset [6], a widely recognized benchmark in music information retrieval. The dataset contains 1,000 audio tracks evenly distributed across 10 genres. Each song lasts 30 seconds and has a sampling rate of 22,050 Hz. These songs include blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. Because each genre has an equal number of samples, no genre is overrepresented or underrepresented. This prevents the model from being biased toward genres with more data.

To prepare the audio data for CNN processing, raw waveforms must first be converted into log-scaled Mel-spectrograms. It begins as a standard Mel-spectrogram, which is essentially a two-dimensional representation of time on the horizontal axis and frequency on the vertical axis, resembling a colored image. The color of a Mel-spectrogram represents the intensity of the sound, which is the amplitude or energy of the sound at that specific time and frequency. However, raw amplitude values in a spectrogram typically span an enormous range. This is because, in a piece of music, the difference in physical energy between the quietest meaningful sound (like the decay of a cymbal or a faint harmonic) and the loudest sound (like a kick drum hit) can be enormous, sometimes a factor of 1000 or 10,000 times or more. Directly using these linear amplitude values can be problematic for neural networks, as they may become overly sensitive to very high-energy components like drum hits while ignoring quieter but important details like faint harmonics. Therefore, a logarithmic transformation converting the power spectrogram to decibels (dB) must be done to compress the wide range into a more manageable scale, thereby achieving a log-Mel-spectrogram. This aligns the representation closer to human auditory perception, as studies from Sable et al prove that changes in sound intensity are perceived proportionally to the absolute value of the logarithm of the standard interval ratio instead of linearly [7]. This study uses the log-Mel-spectrograms obtained from the GTZAN dataset as the input feature, while allocating 80% of the dataset for training and 10% for validation and 10% for testing. Figure 2 below are examples of the log-Mel-spectrogram obtained from the GTZAN dataset.

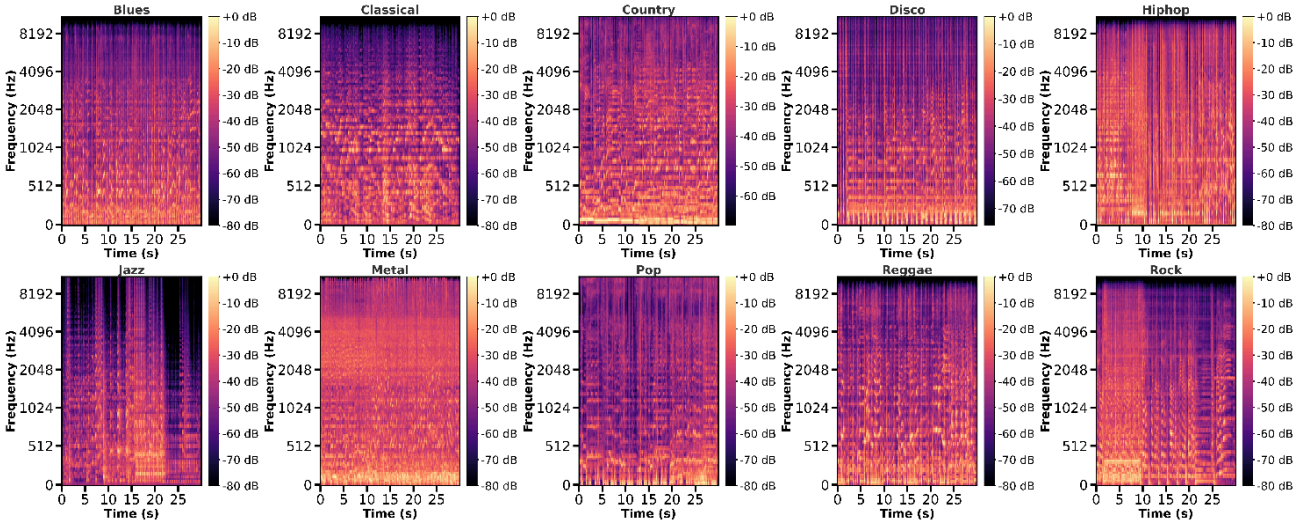


Figure 2. Examples of log-Mel-spectrogram obtained from the GTZAN dataset

DOI: 10.54254/2753-8818/2026.CH31855

The data will be shuffled before splitting into training and validation sets to randomize the order of samples, thereby eliminating any unintended sequential patterns that could be learned by the model. More importantly, a stratified split would be employed, meaning that the class proportions of the original 10-genre dataset were preserved in each subset. This prevents a scenario where, by random chance, a particular genre is over- or underrepresented in a split, which would skew the model's learning and lead to misleading performance metrics. Additionally, to facilitate stable gradient descent and faster convergence, the log-Mel-spectrogram features were normalized using Z-score standardization. The mean μ and standard deviation σ were computed exclusively from the training partition to prevent data leakage. Each feature point x_i was transformed into $x^{'}$ such that the resulting distribution possessed a mean of 0 and a unit variance of 1. This is calculated using the formula:

$$x^{'} = \frac{x_i - \mu}{\sigma} \tag{1}$$

2.2. Model architecture

Four convolutional blocks make up the neural network. Because of this, the model can process 2D spectrograms and progressively pick up more intricate audio characteristics at each level. To identify basic local patterns across the spectrogram, the first block employs 32 filters with tiny 3x3 kernels. The model progressively captures more intricate audio characteristics as the input passes through the subsequent blocks. The basic patterns found in the first block, like basic rhythms, are combined in the second block using 64 filters. After that, the third block expands to 128 filters and begins to record more detailed information, such as instrument timbre. The last block combines these elements using 256 filters, thereby creating representations that can span a large region of the spectrogram. Between the convolutional and activation layers, the network can optionally apply Batch Normalization when enabled, which stabilizes training by normalizing activations within each mini-batch. The ReLU activation function is then used to introduce non-linearity by setting negative values to zero. Dimensionality reduction is then performed through the implementation of a 2x2 MaxPooling layer, which reduces the spatial size of the feature maps by half, lowering computational cost and making the model more robust to small timing shifts in the audio data. After pooling, Dropout may be applied when enabled, randomly deactivating 30% of neurons during training to prevent over-reliance on specific connections. Additionally, when enabled, L2 regularization is applied to the convolutional weights, penalizing large values to discourage overly complex models. After the fourth convolutional block, the 3D feature maps of height, width, and channels are flattened into a 1D vector. This vector is passed to a fully connected layer with 256 neurons (units), which combines the extracted features in a non-linear way to allow the neural network to learn complex patterns and support classification. The decision of whether to apply a single regularization technique—Batch Normalization, Dropout, and L2 weight decay—is also made at this stage. The final output layer contains 10 units, one for each music genre, and uses a Softmax activation function to produce a probability distribution over the genres. The network architecture is controlled by a single function that can be called at any time. This ensures that all models share the same structure and that they only differ in terms of the regularization techniques chosen. With each model containing tens of millions of trainable parameters, regularization would be crucial in order to prevent the issue of overfitting.

DOI: 10.54254/2753-8818/2026.CH31855

2.3. Experimental setup

A controlled ablation study was performed to isolate and evaluate the efficacy of three canonical regularization techniques against a non-regularized baseline. Four distinct model configurations were instantiated from the same architectural blueprint. The first being the Baseline Model, which served as control, containing no explicit regularization mechanisms. The second model is the Dropout-Only Model, which incorporates Dropout layers with a rate of 0.3. The third and fourth model integrates Batch Normalization layers and L2 weight decay (with a strength of $1e^{-4}$) respectively.

2.4. Training and evaluation metrics

All models were trained under identical conditions as to ensure their comparability. The Adam optimizer was used throughout the experimentation with a fixed learning rate of 0.001. The Categorical Cross-Entropy (CCE) was chosen as the loss function. The batch size was set constant at 16 in consideration of the limitations of the Nvidia 3060 Ti GPU memory (VRAM). Research from Keskar et al. has also shown that smaller batch sizes tend to converge to flat minima in the loss landscape [8], which is desired since the model is more robust to small variations in the audio, which would help the model generalize better to the test set. Training proceeded for a maximum of 50 epochs, and a callback was implemented to monitor validation loss with a patience of 8 epochs, restoring the model weights from the epoch with the best performance to prevent overfitting from extended training. A dedicated validation set (10% of the data) was used for hyperparameter-free monitoring during training, whereas a test set (10% of the data), unseen during both training and validation, provided the final accuracy metric. The key performance indicator was the overfitting/generalization gap, calculated as follows:

$$\Delta G = Accuracy_{train} - Accuracy_{validation} \tag{2}$$

A large positive gap indicates high overfitting. Effective regularization should minimize this value toward zero. While the gap is an absolute difference, the generalization ratio provides a relative measure of how much knowledge is actually transferable to unseen data. This is calculated as follows:

$$R_g = \frac{Accuracy_{test}}{Accuracy_{train}} \tag{3}$$

A ratio of 1.0 represents perfect generalization. Additionally, this study employs learning curves (Loss and Accuracy vs. Epochs). These curves are analyzed to identify the precise epoch at which the model begins to diverge from generalized patterns.

3. Results and discussion

3.1. Model performance and generalization metrics

Table 1 below summarizes the training, validation, and test accuracies for all four models, alongside the generalization gap and generalization ratio.

DOI: 10.54254/2753-8818/2026.CH31855

Table 1. Regularization strategy summary

Strategy Type	Train Acc	Val Acc	Test Acc	Gen. Gap	Gen. Ratio
Baseline	0.9837	0.5000	0.4800	0.4837	0.4879
Dropout Only	0.9499	0.3500	0.3900	0.5999	0.4106
BatchNorm Only	0.9975	0.6400	0.7500	0.3575	0.7519
L2 Only	0.9825	0.6500	0.6200	0.3325	0.6311

As seen above, the baseline CNN achieved a high training accuracy of 0.9837, but its performance deteriorated substantially on unseen data, with a validation accuracy of 0.5000 and a test accuracy of 0.4800. The resulting generalization gap of 0.4837 and low generalization ratio (0.4879) indicate severe overfitting. This suggests that, in the absence of explicit regularization, the model rapidly memorized training-specific spectro-temporal patterns in the Mel-spectrograms rather than learning features that generalize across songs within the same genre. Furthermore, the Dropout model showed even higher overfitting than the baseline model. It achieved a low test accuracy of 0.3900 and the largest generalization gap (0.5999) of all model variants. Even though the training accuracy became the lowest at 0.9499, this reduction did not translate into either improved validation or test accuracy. These metrics show that randomly turning off neurons during training has interrupted the learning of patterns in the spectrograms. Audio spectrograms rely heavily on patterns in time and frequency, like rhythms, melodies, and harmonic progressions. When Dropout randomly turns off certain neurons, it can break important connections between time and frequency. Because the model can no longer reliably learn important time-frequency patterns during training, it consequently struggles to form robust representations. As a result, the network may memorize the training examples where certain features somehow happen to survive the Dropout, leading to increased overfitting. On the other hand, the L2-regularized model reduced overfitting better than both the baseline and Dropout models. It achieved a test accuracy of 0.6200 and the smallest generalization gap of 0.3325, demonstrating that penalizing large weights indeed helped prevent the network from relying too heavily on specific features of the audio spectrograms during training. By encouraging the model to spread importance more evenly across all features, L2 regularization was able to reduce sensitivity to noise in the training data. However, despite this improvement, its generalization ratio of 0.6311— showing that around 63% of training knowledge is able to transfer to unseen data—is lower than that of the Batch Normalization (BN) model. This indicates that even though L2 regularization made the model simpler and less prone to overfitting than the BN model, it did not stabilize the distribution of activations in intermediate layers, so the features learned during training were less reliably applicable to new data, limiting the transfer of knowledge. As seen in Table 1, the BN model achieved the strongest overall performance, with a test accuracy of 0.7500 and the highest generalization ratio (0.7519). Although its training accuracy was high (0.9975), the generalization gap (0.3575) remained substantially lower than that of the baseline and Dropout models. This indicates that Batch Normalization allowed the model to learn highly accurate representations of the audio patterns. The improved performance can be attributed to BN's ability to stabilize the activations of intermediate convolutional and dense layers through training via normalization. This allows each layer to receive consistent inputs without the need to constantly readjust. This also results in training loss that decreases more stably (as seen in Figure 3). As a result, the model was better able to extract robust patterns that are relevant to a specific genre from the Mel-spectrograms.

DOI: 10.54254/2753-8818/2026.CH31855

3.2. Training and validation curve

A careful analysis of the training and validation accuracy and loss curves suggests a few differences in overfitting across models. This is represented by Figure 3.

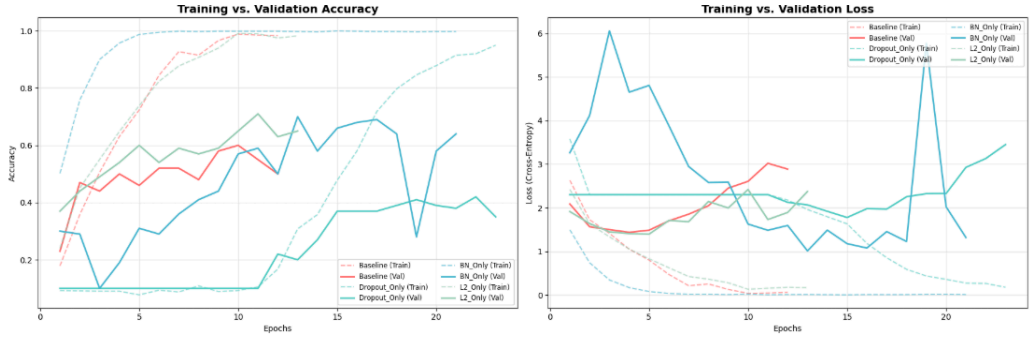


Figure 3. Learning curves. The left graph depicts training vs. validation accuracy. The right graph depicts training vs. validation loss

For the baseline model, there seems to be a consistent decrease in training loss starting from epoch 1, and validation loss begins increasing after approximately epoch 3. Although training accuracy continues to improve, validation accuracy seems to be much lower. Nevertheless, the divergence of validation loss from training loss suggests early overfitting. An interesting observation is that the Dropout-only model exhibited the latest divergence, with validation loss only beginning to rise at Epoch 15. This is significantly later than the baseline and L2 models. This indicates that the random deactivation of neurons is able to prevent early overfitting. However, a notable phenomenon was observed in the model, where the validation loss remained in a plateau state until epoch 11. This suggests that the random deactivations of neurons initially prevented the network from identifying a stable gradient direction. This effectively prevented early overfitting, but the immediate divergence after merely 4 epochs indicates that the model still transitioned into memorizing training noise without achieving a lasting phase of meaningful audio pattern extraction. Furthermore, the L2-regularized model behaved in a very similar way to the baseline model, with validation loss diverging at around the same epoch as the baseline. However, the validation loss constantly fluctuates up and down instead of continuously increasing like the baseline model. Its validation accuracy is also higher than that of the baseline despite having almost identical training accuracy curves. This shows that the L2-regularized model can learn somewhat accurate representations of the audio features but is unable to consistently apply them to new data due to it not being able to stabilize the distributions of intermediate activation layers, as discussed earlier. Another interesting phenomenon was observed with the Batch Normalization model, where there seems to be a consistent gap between the validation and training loss, with the validation loss seemingly decreasing along with the training loss and fluctuating up and down. Looking at the training loss curve, it can be said that BN effectively smoothed the optimization landscape, allowing for consistent feature extraction and stable gradient updates. The observed fluctuations are caused by the fact that BN normalizes the activations using mean and variance computed per mini-batch, but not the whole dataset. Since each mini-batch is slightly different from one another, normalization introduces small randomness into the training process. This randomness consequently causes fluctuations in both validation accuracy and loss.

DOI: 10.54254/2753-8818/2026.CH31855

4. Conclusion

This study initially sought to conduct an analysis of the regularization techniques of Dropout, L2-Regularization, and Batch Normalization for minimizing the issue of overfitting in a CNN-based music genre classification system using the GTZAN dataset. The results of this research indicate a unique effect of every one of these techniques on minimizing overfitting or the absence of it. The starting model without any regularization demonstrated significant overfitting, indicated by a substantial generalization gap. This proves the necessity to use regularization methods in spectrogram-based CNNs. The use of Dropout by itself resulted in the highest level of overfitting, indicating that its random neuron deactivation might interfere with the model's ability to acquire learning of time-frequency patterns that are important for audio representation. L2-Regularization showed a better reduction in overfitting as compared to the baseline and Dropout models. This is because it applies a penalty to large weights, thereby encouraging a simpler model and reducing the model's sensitivity to training noise. However, Batch Normalization proved to be the most effective technique, reaching the highest test accuracy of 75%. This is because it normalizes activation distributions during the training process. It allowed the model to acquire strong audio representations while minimizing the memorization of noise that is present in the audio itself. These results show that regularization techniques do not perform equally well across different types of tasks. In audio classification, Dropout is more detrimental than not using any regularization initially, in contrast to traditional image recognition models, in which Dropout often performs extremely well. In the field of audio classification, the patterns in spectrograms depend heavily on relationships across time and frequency. This is why methods like Batch Normalization, which keep layer outputs (activations) stable, help the model learn these patterns more reliably. Future work could explore hybrid approaches, such as combining Batch Normalization with the penalty from L2 regularization.

Acknowledgements

This work was conducted under the mentorship of Professor Mark Volgesberger at MIT. The author gratefully acknowledges the resources he provided and his guidance.

References

[1] Cheng, Y.-H. and Kuo, C.-N. (2022) Machine Learning for Music Genre Classification Using Visual Mel Spectrum. Mathematics, 10, 4427.
[2] Wu, P., Gao, W., Chen, Y., Xu, F., Ji, Y., Tu, J. and Lin, H. (2025) An Improved ViT Model for Music Genre Classification Based on Mel Spectrogram. PLOS ONE, 20, e0319027.
[3] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15, 1929–1958.
[4] EITCA Academy (2024) How Do Regularization Techniques Like Dropout, L2 Regularization, and Early Stopping Help Mitigate Overfitting in Neural Networks? EITCA Academy – Artificial Intelligence (EITC/AI/ADL Advanced Deep Learning), May 21.
[5] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proceedings of the 32nd International Conference on Machine Learning (ICML), Vol. 37, 448–456.
[6] Olteanu, A. (2020) GTZAN Dataset – Music Genre Classification. Kaggle Dataset. Available from: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification
[7] Sable, J.J., Gratton, G. and Fabiani, M. (2003) Sound Presentation Rate Is Represented Logarithmically in Human Cortex. European Journal of Neuroscience, 17, 2492–2496.

DOI: 10.54254/2753-8818/2026.CH31855

[8] Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M. and Tang, P.T.P. (2017) On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. Proceedings of the Journal of Machine Learning Research, 37, 1–12. https: //doi.org/10.48550/arXiv.1609.04836