Accuracy Comparison of Different Batch Size for a Supervised Machine Learning Task with Image Classification

Noor Baha Aldin
Electrical and Electronics Engineering
Hasan Kalyoncu University
Gaziantep, Turkey
noor.aldin@hku.edu.tr

Shaima Safa Aldin Baha Aldin
Continuing Education Center
Al-Nahrain University
Baghdad, Iraq
shaima.safaaldin@nahrainuniv.edu.iq

2022 9th International Conference on Electrical and Electronics Engineering (ICEEE) | 978-1-6654-6754-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICEEE55327.2022.9772551

***Abstract*—Machine learning is a type of artificial intelligence where computers solve issues by considering examples of real-world data. Within machine learning, there are various types of techniques or tasks such as supervised, unsupervised, reinforcement, and many hyperparameters have to be tuned to have high accuracy especially in image classification. The batch size refers to the total number of images required to train a single reverse and forward pass. It is one of the most essential hyperparameters. In our paper, we have studied the supervised task with image classification by changing batch size with epoch. The characterization effect of increasing the batch size on training time and how this relationship varies with the training model have been studied, which leads to extremely large variation between them. According to our results, a larger batch size does not always result in high accuracy.**

Keywords—machine learning, supervised task, image classification, batch size

I. Introduction

Machine learning is creating rapid and fascinating developments across all levels of society. It is the engine behind the recent technological breakthroughs in industries such as self-driving cars. It enables more accurate and rapid text translation into hundreds of languages. It powers the artificial intelligence assistants we might find in our home. Also, it has the potential to improve worker safety and accelerate the discovery of new medications.

Although machine learning can help in many aspects, we may face a challenging task in machine learning, which is the detection of an object. Solving this in the usual manner would include paying close attention to features such as lighting conditions, different types of objects, and numerous poses the object may be in. The problem solver in machine learning abstracts away a part of their solution as a component known as a model, and then applies a model training technique in order to modify model to real-world data. The end result is a trained model that can predict outcomes that are not included in the data set used to train it. The main objective is to utilize a model generated by a training algorithm to make predictions or find patterns in data that can be utilized to address a problem [1].

Many hyperparameters need to be adjusted to appropriately train our model to classify images; the model's performance will be affected by these hyperparameters. The batch size is one of the most important hyperparameters to be tuned, in each epoch, this is the total number of images utilized in the network's training. The model may take too long to achieve convergence if the hyperparameter is set too high. However, if it's set too low, the model will bounce without achieving the desired performance. The effect of batch size on image classification performance has been studied. We showed that how batch size affects model quality when we have used different batch sizes [2].

II. Machine Learning Tasks

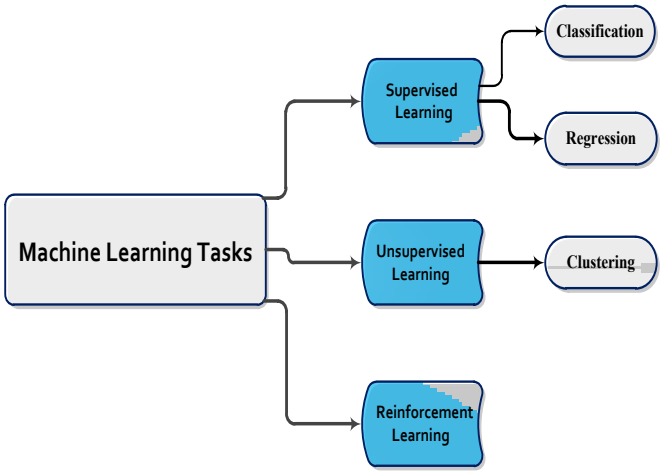

Fig. 1. Machine learning tasks.

All model training techniques, as well as the model itself, take data as input. Their outputs might differ considerably and depending on the task they must achieve, they are divided into several groups. Often, when designing a machine learning task we consider the type of data required to train a model. A machine learning task is frequently determined by the presence or absence of labeling in our data. The main three tasks in machine learning are supervised, unsupervised, and

reinforcement. In our study, we have used a supervised task since we use labeled data for classification images. The supervised task completes its task using labeled data that it has obtained in advance in order to create a prediction model. Fig. 1 shows the tasks of machine learning [3].

A. *Supervised Learning*

A label or output value is assigned to each training sample in the dataset. The algorithm learns to anticipate output values or labels [4].

B. *Unsupervised Learning*

The training data doesn't have any labels. A machine learning algorithm tries to figure out what the data's underlying patterns or distributions are. This implies that we don't have to provide any labels or solutions to the model while it's being trained [1].

C. *Reinfrocement Learning*

The algorithm determines the best actions to do in a given situation in order to maximize a reward while achieving a specified goal [5].

III. Difference between Batch and Epoch

A. *Batch Size*

The batch size is a gradient descent hyperparameter that determines how many data points must be analyzed before the internal model parameters are updated. It is an important hyperparameter that determines the learning algorithm's dynamics. It controls the accuracy with which the error gradient is estimated when we train our model. The predictions are compared to the predicted output variables at the end of the batch, and an error is computed. Based on this error, the algorithm is used to enhance the model. The training dataset can be divided into many batches. The batch size has an influence on how fast a model learns and how stable the learning process is [2, 6].

B. *Epoch*

In gradient descent, the number of epochs is a hyperparameter that specifies how many complete passes of the training dataset are performed. Since one epoch is too large to provide to the computer all at once, we divide it into several smaller batches. We have used several epochs because of the restricted dataset, and we optimized the learning using gradient descent, which is an iterative process. As a result, a single pass or epoch is insufficient to update the weights.

The internal model parameters can be improved for each sample in the training dataset once every epoch. An epoch has one or more batches [7].

C. *The Relation between Batch Size and Epoch*

We have trained a dataset with 3000 images and we have chosen different batch size and epochs. Equation 1 shows the relationship between batch size and epoch [2, 7].

$$Epoch = \frac{Number\ of\ dataset}{Batch\ Size} \tag{1}$$

IV. Image Classification Model

Image classification tasks in machine learning have three major components: the machine learning model, the model training algorithm, and the model inference algorithm.

A model is a highly general program that has been made particular by data in order to be trained. It utilized to solve different problems. Model training algorithms analyze the current iteration to determine what improvements might be done to bring the model closer to the desired goal. Those modifications in our model are performed by changing the hyperparameters, and the iteration process is continued until the model meets our criteria. Then, the trained model is utilized to make predictions. Our goal was to predict the image and feed it into the model. We accomplished this by providing the model with labeled data and running a supervised machine learning task with different batch sizes and epochs [8, 9, 10].

In our study, we have fed 3000 training data into the model, and updated the model parameters to reduce loss and improve image classification accuracy. We continued to cycle through these steps until we reached our destination, which is determined by the batch size and the number of training cycles. Then, using different batch sizes, we estimated the loss and accuracy function, which controls the accuracy of the error gradient estimate while training neural networks [11, 12].

A. *Model Training*

The dataset is randomly divided in the first step of model training. This enables us to hide specific data during training in order to test our model before we put it into production. We get two sets of data: test dataset and training dataset when we divide our dataset. The model has been trained using the training dataset, while the test dataset contain data that was kept hidden from the model during training, and we utilized this to see how effectively our model generalizes to new data. The percentage of labeled dataset is shown in Fig.2.

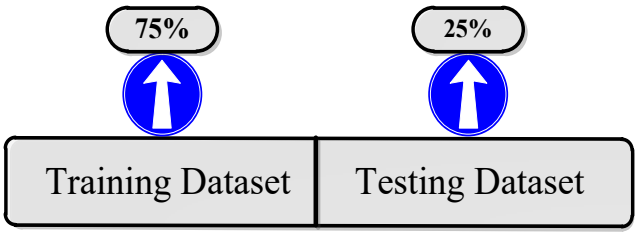

Fig. 2. Percentage of labeled dataset.

Also, we are splitting our data into training and test data prior to beginning model training because if we utilize all of the data acquired during training, we would not have any data with which to test the model during the model evaluation phase [13].

B. *Loss Crossentropy*

The crossentropy loss function is utilized in image classification tasks. Loss functions measure how far an estimated value is from its true value, Fig. 3. shows the loss crossentropy function [14, 1].

The loss of an image classification is calculated using the crossentropy loss function, which computes the following sum,

if the divergence of the predicted probability from the actual label increases then the cross-entropy increases. The equation of loss function is:

$$J = -\sum_{i=1}^{N} y_i \log\big(h_\theta(x_i)\big) + (1 - y_i)\log(1 - h_\theta(x_i)) \quad (2)$$

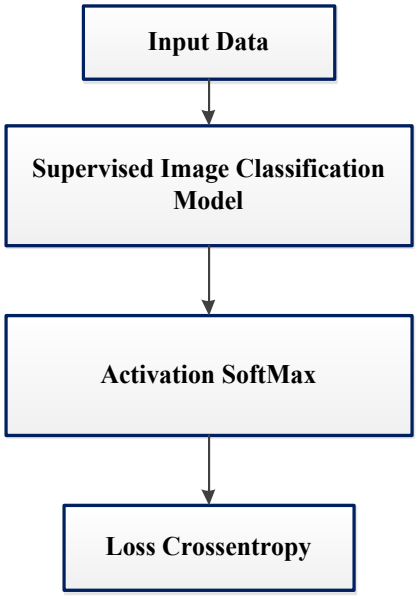

Fig. 3. Loss crossentropy function.

Where y_i is the value of true label and $h_\theta(x_i)$ is the predicted value. Using crossentropy, the model learns to give the right classification a high probability and the other classification a low probability.

The SoftMax in Fig3 represents the activation function that is utilized with the crossentropy loss function. The model's output just has to be positive in order for the logarithm of every output value to exist. However, the fundamental advantage of this loss function is that it can be used to compare the probabilities. The SoftMax activation rescales the model output to get the desired properties [15].

V. Results and Conclusions

The model parameters that set or configure the training algorithm have been modified to analyze how the model performs. We have changed the batch size and epoch, which are important hyperparameters that determine the learning algorithm's dynamics. The model's distance from the goal has been represented using the loss function. Also, we have utilized model accuracy as an evaluation metric, which is frequently used, where the percentage of correct predictions made by a model is referred to as its accuracy. For both the train and test sets, all plots are created that indicate the accuracy and loss over the training epochs [16, 17].

When the batch size is 100, the execution time and model loss are 30 minutes and 1.1336, respectively, and model accuracy is 0.6932 in Fig. 4. As seen in Fig.4, the training loss decreases after each training, whereas the test loss decreases to a point and then begins to increase. This result is in overfitting. Overfiting causes the model to become more specialized to training data, reducing its ability to generalize to new data, leading to an increase in generalization error. Our model error is measured by the model loss over time; the lower our loss, the better our model performance will be. In Fig. 5 we can observe the learning process's behavior, despite its ups and downs, the model's loss decreases over time after the second training, showing that it is learning. When the batch size is 200, the execution time is 14 minutes, the model loss is 0.9541, and the model accuracy is 0.6855 in Fig.5. We see also that the model's accuracy is increasing over time, indicating that the model is improving as it learns. The more accurate our model, the better it is.

In Fig. 6 the execution time is 10 minutes, the model loss is 0.9440, and the model accuracy is 0.6702 when the batch size is 300. While in Fig. 7 the execution time is 6 minutes, the model loss is 0.9817, and the model accuracy is 0.6512 when the batch size is 428. We can say that the curves have a good match in loss, because the training loss declines to a point of stability and the test loss decreases to a point of stability with a small gap. We can observe that the model properly converged in the loss.

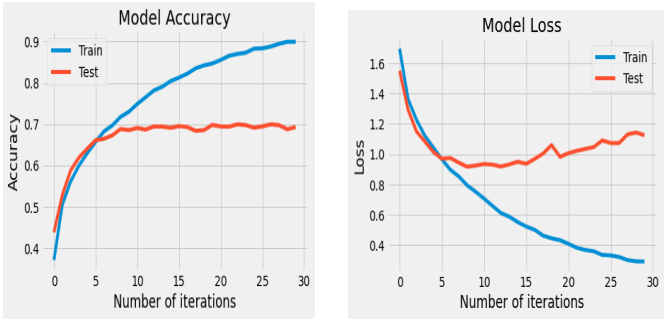

Fig. 4. Model accuracy and loss with 100 batch-sizes.

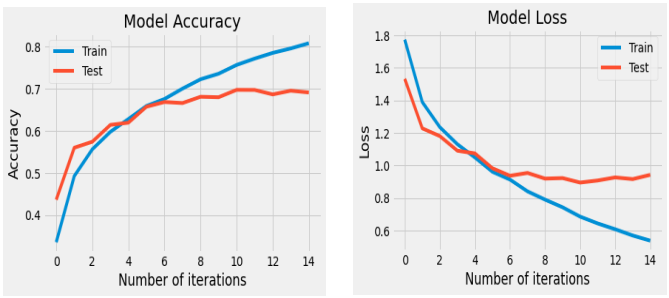

Fig. 5. Model accuracy and loss with 200 batch-sizes.

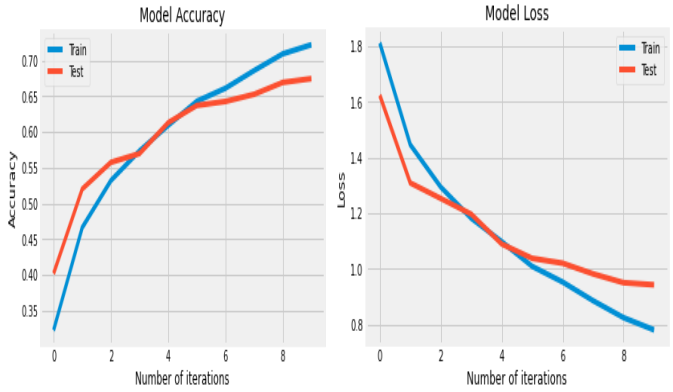

Fig. 6. Model accuracy and loss with 300 batch-sizes.

The Figures show that the small batch size resulted in improved image classification accuracy, but it takes more time to run and the loss function is higher but not in all cases. As seen in Fig 6, the loss function decreased and then increased again in Fig 7 and Fig 8. Where in Fig.8 the execution time is 5 minutes, the model loss after the second training is 1.05, and the model accuracy is 0.6269 when the batch size is 500.

We consider accuracy, recall, and F1 score as an indicator of classifier performance in Table 1.

TABLE I. PRECISION, RECALL, F1 SCORE PARAMETERS WITH DIFFERENT SIZES OF BATCH

Batch Sizes	Precision	Recall	F1 score
100	0.6973	0.6732	0.6850
200	0.6782	0.6893	0.6837
300	0.6428	0.6613	0.6519
428	0.6265	0.6092	0.6177
500	0.5974	0.5730	0.5849

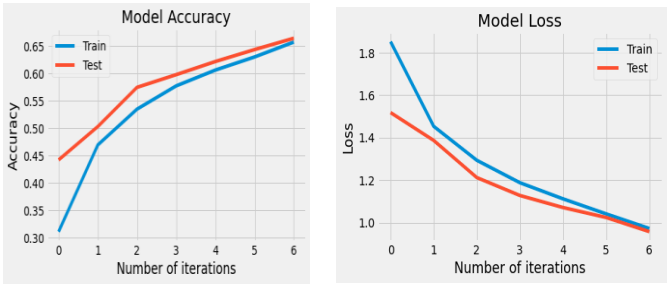

Fig. 7. Model accuracy and loss with 428 batch-sizes

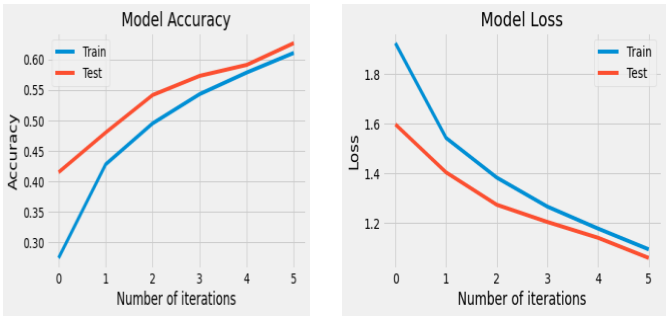

Fig. 8. Model accuracy and loss with 500 batch-sizes.

The Figures show that a large batch size results in lower accuracy in image classification, but it takes less time to execute. Whereas the loss function is low, but it sometimes becomes high and then returns to a low value after a few trainings, this indicating that our model is learning.

Furthermore, the Figures reveal that small batches result in slower learning and increased variance in classification accuracy. Larger batch sizes speed up the learning process, but the latter stages provide a more stable model, as seen by decreased variance in classification accuracy. Our goal was to find model parameters that would minimize the model's error on the training dataset while increasing accuracy. We concluded that when the number of epochs increases, the weight in the neural network is modified more, resulting in improved learning especially after a few trainings.

We can modify more hyperparameters for different models in the future and analyze their effects on these models, not just for image classification.

REFERENCES

[1] J. Glaser, A. Benjamin, R. Farhoodi, K. Kording, "The roles of supervised machine learning in systems neuroscience," Progress in Neurobiology, 2019, pp. 126–137.

[2] Z. Ma, Y. Xu, H. Xu, Z. Meng, L. Huang and Y. Xue, "Adaptive batch size for federated learning in resource-constrained edge computing," IEEE Transactions on Mobile Computing, 2021, pp. 1–1

[3] J. Dongsheng, Z. Zhang, Y. Zaho, Q. Zaho, "Research on classification of COVID-19 chest x-ray image model feature fusion based on deep learning," Journal of Healthcare Engineering, 2021, 12 pages.

[4] P. Wang, En. Fan, P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," Pattern Recognition Letters, 2021, pp. 61–67.

[5] J. Liue, F. Ping, "Image classification algorithm based on deep learning-kerbel function," Scientific Programming, 2020.

[6] S. Tripathi, R. Kumar, "Image classification using small convolutional neural network," 2019 9th International Conference on Cloud Computing, Data Science & Engineering, 2019, pp. 483–487.

[7] B. Liu, W. Shen, P. Li, X. Zhu, "Accelerate mini-batch machine learning training with dynamic batch size fitting," International Joint Conference on Neural Networks (IJCNN), 2019, PP.1-8.

[8] S. Panigrahi, A. Nanda, T. Swarnkar, "Deep learning approach for image classification," 2nd International Conderence on Data Science and Business Analytics (ICDSBA), 2018, pp. 511–516.

[9] K. Bressem, L. Adams, C. Erxleben, etc., " Comparing different deep learning architectures for classification of chest radiographs,"Scientific Reports, 2020.

[10] C. Shorten, T. Khoshgoftaar, " A survey on image data augmentation for deep learning," Journal of Big Data, 2019.

[11] S. Tripathi, R. Kumar, "Image classification using small convolutional neural network," International Conference on Cloud Computing, Data Science and Engineering, 2019, pp.483-487.

[12] S. Chaganti, I. Nanda,K. Pandi,etc.,"Image classification using SVM and CCN," International Conference on Computer Science, Engineering and Applications (ICCSEA), 2020, PP.1-5.

[13] G. Hinton, L. Deng, D. Yu, et al, " Deep neural networks for acoustic modeling in speech recognition," IEEE Signal Processing Magazine, 2012, 29. H. Zhao, O. Gallo, I. Frosio, J. Kautz, "Loss functions for image restoration with neural networks," in IEEE Transactions on Computational Imaging, 3, pp. 47-57, 2017.

[14] G. Goh, N. Hodas, A. Vishnu, "Deep laerning for computational chemstry," Journal of Computational Chemistry, 2017, pp. 12921-1307.

[15] N. Chen, J. Zhu, F. Sun, E. Xing, "Large-margin predictive latent subspace learning for multiview data analysis," IEEE Transactions on Pattern Analysis, 2012, pp. 2365-2378.

[16] H. Zhao, O. Gallo, I. Frosio, J. Kautz, "Loss functions for image restoration with neural networks," in IEEE Transactions on Computational Imaging, 3, pp. 47-57, 2017.

[17] F. Idrees, M. Rajarajan, M. Conti, T. Chen, et al, " A novel Android malware detection system using enemble learning methods," Comut. Secur, 2017, pp. 36-46.