An Investigation of the Learning Rate-Batch Size Correlation For Image Classification.

Rohin Harikumar, Maria Alappat and Swapna Kumar

November 2023

Introduction

- Let Ω_n denote the weights and biases of a neural network after the n^{th} update, and $\mathcal{J}$ be the loss function. The stochastic gradient descent step is given by:

$$\Omega_{n+1} = \Omega_n - \rho * \nabla \mathcal{J}(\Omega_n)$$

  Here, ρ is the learning rate, and $\nabla \mathcal{J}(\Omega_n)$ is the gradient expectation computed over a subset of the training set.
- The cardinality of this subset is the batch size.
- A larger batch size allows more confidence in the stochastic gradient, permitting a larger step controlled by the learning rate.
- Conversely, a smaller batch size suggests a smaller learning rate is preferable, as the stochastic gradient is computed over a smaller subset of the training set.
- This correlation was verified for a binary image classification problem using a CNN (namely the VGG-16 model) by Kandel and Castelli [1].
- In this project, we investigate if such a correlation can be found for categorical classification problems using CNNs. We also investigate how the optimization algorithm used can impact the strength of such a correlation.
- Finally, we conduct similar experiments using a simpler neural network that only utilizes dense layers to demonstrate that such a correlation is a universal property of NNs and see the impact of the optimization algorithms used in such NNs.
- All experiments in this project are done using Tensorflow [2].

MLP : Workflow

1. A multi-layered perceptron (MLP), neural networks comprised entirely of fully interconnected layers known as dense layers, is initialized.
2. For each pair of batch size $\in \{2^2, \ldots, 2^8\}$ and learning rate $\in \{10^{-4}, 10^{-3}, 10^{-2}\}$, the MLP is trained.
3. Training is terminated at 200 epochs or if a stopping condition (loss stabilisation : $|J(\Omega_n) - J(\Omega_{n+1})| < 10^{-2}$ maintained for atleast 10 epochs) is met.
4. Accuracy of the MLP is measured over the test partition and recorded.
5. Steps (I - III) performed sequentially is referred to as an experiment.
6. To observe the impact of the optimization algorithm on the correlation, experiments are conducted using SGD and Adam.

MLP : Dataset 1

- Classification Task:
  - To classify 32x32 greyscale images of hand written digits 0-9.

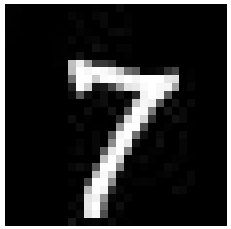


Figure: Sample image

- Dataset Used: [Handwritten Digits MNIST]
  - Approximately 43,000 images of handwritten digits, with 80% used for training and 20% used for testing.
- Network Architecture:
  - As dense layers cannot process 2D arrays, images are flattened to vectors of length 32x32. Thus, the input layer is a flatten layer accepting 2D arrays of dimensions 32x32.
  - The hidden layers comprise of a dense layer with 100 neurons followed by another dense layer of 50 neurons. The ReLu activation function is used in these layers.
  - The output layer is another dense layer with a neuron assigned for each digit and the SoftMax activation function is used.

MLP : Dataset 1

- Accuracy recordings from experiment 1 using SGD optimizer.

batch size	lr = 0.0001	lr = 0.001	lr = 0.01
4	0.931548	0.965833	0.968929
8	0.917143	0.960714	0.969286
16	0.905476	0.947024	0.969167
32	0.897901	0.944179	0.968273
64	0.872615	0.920802	0.964814
128	0.832332	0.903606	0.958413
256	0.744910	0.900450	0.942472

Table: The maximum recorded accuracy in each column is highlighted.

- We observe the expected correlation for learning rates 0.001 and 0.01. Potentially using finer batch size increments will emphasize the correlation among learning rates 0.0001 and 0.001.

MLP : Dataset 1

- Accuracy recordings from experiment 1 using Adam optimizer.

batch size	lr = 0.0001	lr = 0.001	lr = 0.01
4	0.965119	0.965238	0.924048
8	0.964048	0.957500	0.932976
16	0.969167	0.963690	0.955000
32	0.965052	0.965887	0.947042
64	0.965768	0.963979	0.953602
128	0.964303	0.964183	0.953125
256	0.964844	0.962595	0.962713

Table: The maximum recorded accuracy in each column is highlighted.

- We see that higher batch size combined with higher learning rates yield models with higher accuracy. The hypothesised correlation is observed for both optimizers with Adam exhibiting a stronger correlation than SGD.

MLP : Dataset 2

- Classification Task:
  - To classify 28x28 greyscale images of 10 articles of clothing.

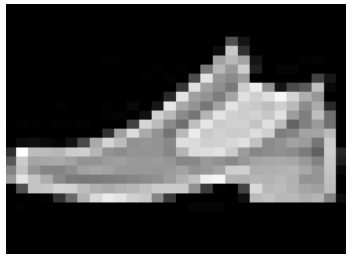


Figure: Sample Image

- Dataset Used: Fashion MNIST
  - Approximately 70,000 images of various articles of clothing such as trousers, ankle boots etc., with 80% used for training and 20% used for testing.
- Network Architecture:
  - As dense layers cannot process 2D arrays, images are flattened to vectors of length 28x28. Thus, the input layer is a flatten layer accepting 2D arrays of dimensions 28x28.
  - The hidden layers comprise of a dense layer with 100 neurons followed by another dense layer of 50 neurons. The ReLu activation function is used in these layers.
  - The output layer is another dense layer with a neuron assigned for each class and the SoftMax activation function is used.

MLP : Dataset 2

- Accuracy recordings from experiment 2 using SGD optimizer.

batch size	lr = 0.0001	lr = 0.001	lr = 0.01
4	0.797535	0.839789	0.838028
8	0.806338	0.841549	0.836268
16	0.741197	0.815141	0.830986
32	0.697183	0.816901	0.838028
64	0.658451	0.806338	0.846831
128	0.577465	0.776408	0.815141
256	0.336268	0.723592	0.809859

Table: The maximum recorded accuracy in each column is highlighted.

- We observe the expected correlation for learning rates 0.001 and 0.01. Potentially using finer batch size increments will emphasize the correlation among learning rates 0.0001 and 0.001.

MLP : Dataset 2

- Accuracy recordings from experiment 2 using Adam optimizer.

batch size	lr = 0.0001	lr = 0.001	lr = 0.01
4	0.889500	0.884833	0.778667
8	0.881333	0.887333	0.822333
16	0.889167	0.890000	0.860000
32	0.890000	0.892667	0.862333
64	0.890167	0.895000	0.875333
128	0.895167	0.893667	0.883000
256	0.892500	0.890000	0.876333

Table: The maximum recorded accuracy in each column is highlighted.

- For learning rates 0.001 and 0.01, we observe the hypothesised correlation. For a very low learning rate 0.0001, we see an anomalous increase in accuracy at high batch sizes. This could be due to a bias in the test partition.

MLP : Dataset 3

- Classification Task:
  - To classify 8x1 vectors of spectral characteristics of astronomical images into three classes : Galaxy, Star and Quasar.

	alpha	delta	u	g	r	i	z	redshift
0	135.689107	32.494632	23.87882	22.27530	20.39501	19.16573	18.79371	0.634794
1	144.826101	31.274185	24.77759	22.83188	22.58444	21.16812	21.61427	0.779136
2	142.188790	35.582444	25.26307	22.66389	20.60976	19.34857	18.94827	0.644195
3	338.741038	-0.402828	22.13682	23.77656	21.61162	20.50454	19.25010	0.932346
4	345.282593	21.183866	19.43718	17.58028	16.49747	15.97711	15.54461	0.116123

Figure: Figure shows spectral characterisitcs used for classification of the source into 3 classes. Each component of the 8x1 vector corresponds to one spectral characterisitc such as 'g' intensity of green spectrum, 'i' intensity of infrared spectrum etc.

- Dataset Used [Stellar Classification Dataset - SDSS17]:
  - Approximately 100,000 labeled samples, with 80% used for training and 20% used for testing.
- Network Architecture:
  - The input layer accepts 8x1 vectors with all inputs normalized to $[0, 1]$ range.
  - The hidden layers comprise of 3 dense layers with 200, 90 and 50 neurons respectively. The Sigmoid activation function is used in these layers.
  - The output layer is another dense layer with a neuron assigned for each class and the SoftMax activation function is used.

MLP : Dataset 3

- Accuracy recordings from experiment 3 using SGD optimizer.

batch size	lr = 0.0001	lr = 0.001	lr = 0.01	lr = 0.1
4	0.594450	0.594450	0.961500	0.963550
8	0.594450	0.594450	0.957850	0.961800
16	0.594450	0.594450	0.594450	0.965650
32	0.594450	0.594450	0.594450	0.963350
64	0.594450	0.594450	0.594450	0.964350
128	0.594450	0.594450	0.594450	0.951800
256	0.594450	0.594450	0.594450	0.594450

Table: The maximum recorded accuracy in each column is highlighted.

- For low learning rates 0.0001 and 0.001, the network training failed completely for all batch sizes. SGD fails to minimize the loss in these regimes. For higher learning rates 0.01 and 0.1, we see the expected correlation. Note that for higher batch sizes, the training failure repeats.

MLP : Dataset 3

- Accuracy recordings from experiment 3 using Adam optimizer.

batch size	lr = 0.0001	lr = 0.001	lr =0.01
4	0.964950	0.970600	0.967700
8	0.963000	0.969650	0.969150
16	0.960450	0.970450	0.973500
32	0.964450	0.959250	0.969700
64	0.961050	0.969000	0.971000
128	0.962650	0.966500	0.971650
256	0.956950	0.969000	0.965350

Table: The maximum recorded accuracy in each column is highlighted.

- We observe the expected correlation for learning rates 0.001 and 0.01. Potentially using finer batch size increments will emphasize the correlation among learning rates 0.0001 and 0.001.

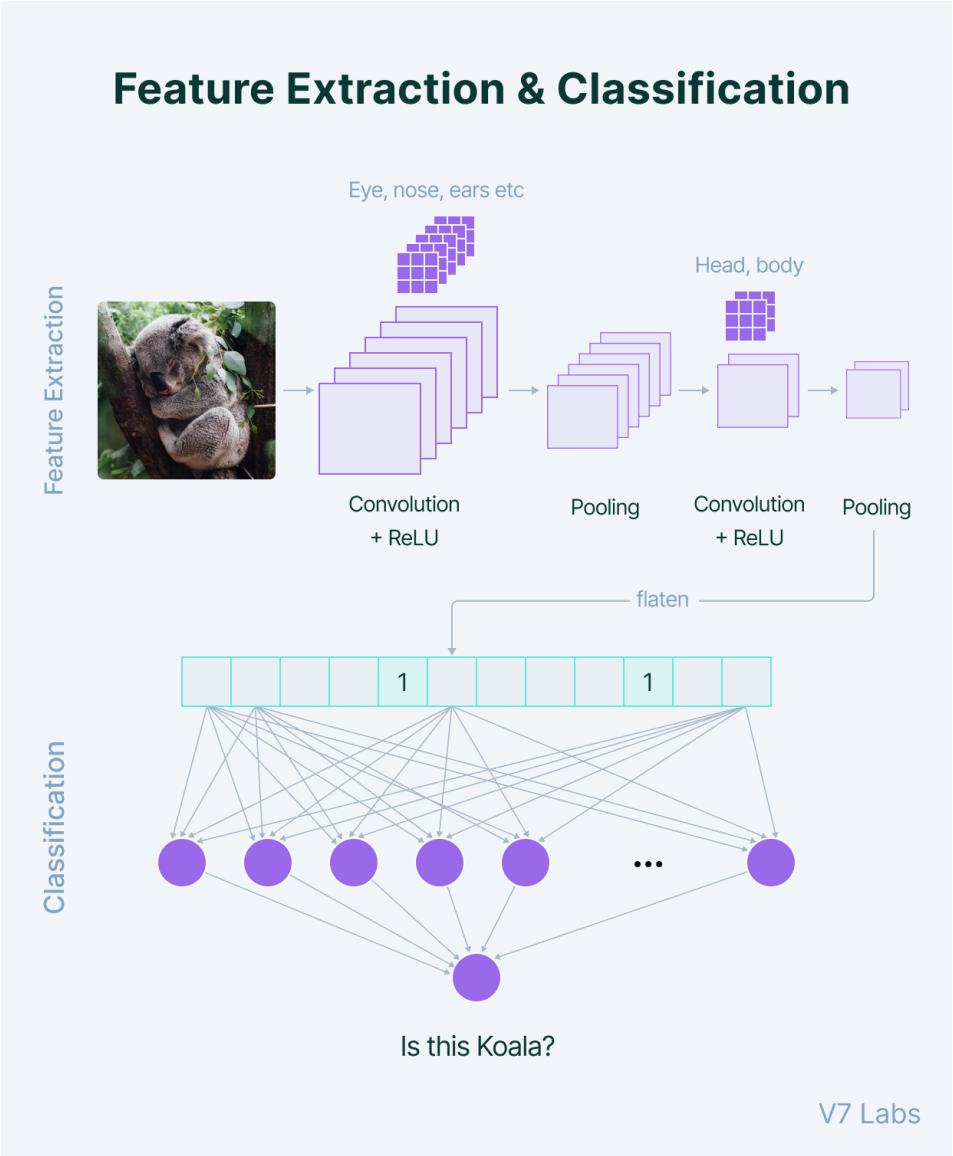
MLP : Conclusion

- Conclusion: We observe that for categorical classification tasks using MLPs, the model quality improves when higher batch sizes are combined with higher learning rates and lower batch sizes are combined with lower learning rates. The optimizer and dataset used have a significant impact on the strength of the correlation. Both the Adam and SGD optimizers show a degree of correlation across all 3 datasets and the strength depends on the dataset used to some extent as well.

How does a CNN work

- Convolutional Neural Networks is a type of Feed-Forward Neural Networks used in tasks like image analysis, natural language processing, and other complex image classification problems. A CNN has hidden layers of convolutional layers that form the base of ConvNets.
- Features refer to minute details in the image data like edges, borders, shapes, textures, objects, circles, etc.
- At a higher level, convolutional layers detect these patterns in the image data with the help of filters. The higher-level details are taken care of by the first few convolutional layers.
- The deeper the network goes, the more sophisticated the pattern searching becomes.
- For example, in later layers rather than edges and simple shapes, filters may detect specific objects like eyes or ears, and eventually a cat, a dog, and what not.
- In the next slide, we can see how a cnn model does feature selection and classification.

How does a CNN work



Convolutional Neural Network

- Convolution Neural Networks employ convolutional layers with small matrices of numbers called kernels or filters
- Each filter is iteratively applied over the input image and transformed into convolved image
- The resulting output is a representation of the presence of the feature represented by the filter, a.k.a "feature map"
- The formula to represented by:

$$G[m,n] = (f * h)[m * n] = \sum_{j}^{n} \sum_{k}^{n} (h[j,k] f[m-j, n-k]))$$

- where m and n are the column and row indices for the resulting feature map, j and k are the dimensions of the filter/kernel, the input is represented by f and the filter matrix is represented by h

Convolutional Neural Network: Filter Example

- Each filter is iteratively applied over the input image and transformed into a filtered image/feature map

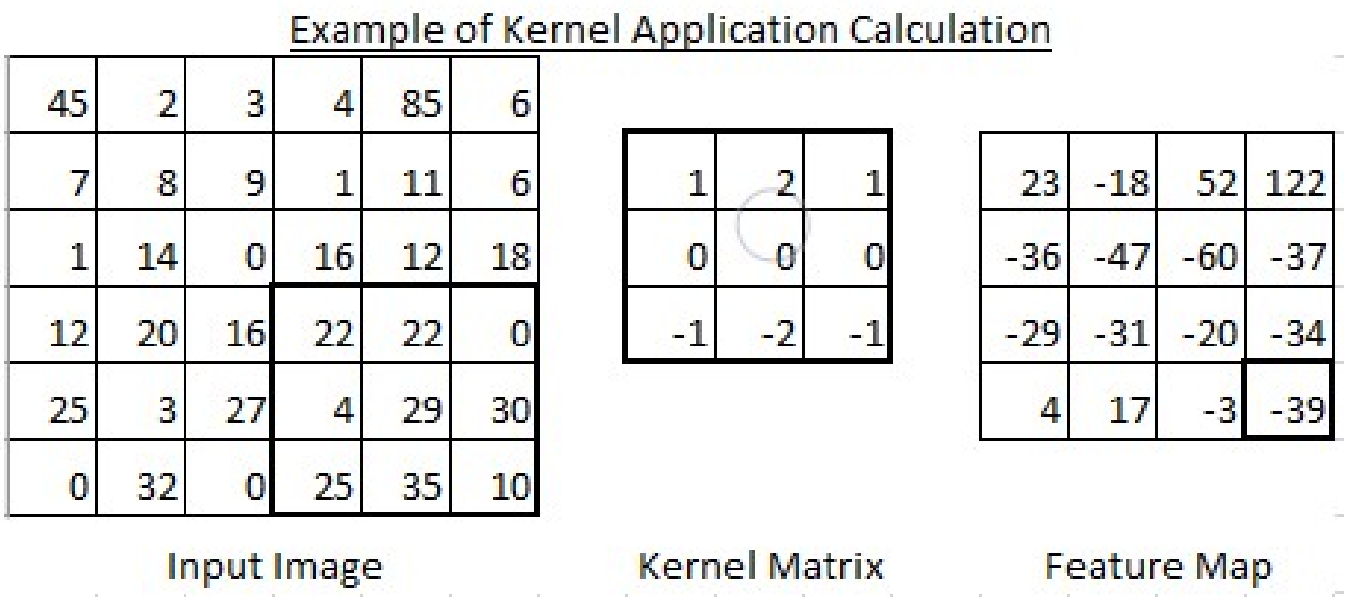


Figure: Example Calculation:
$22 * 1 + 22 * 2 + 0 * 1 + 4 * 0 + 29 * 0 + 30 * 0 + 25 * -1 + 35 * -2 + 10 * -1 = -39$

CNN: Max Pooling Example

- Max Pooling helps to reduce the number of values in the convolved image and reduce calculations in the model

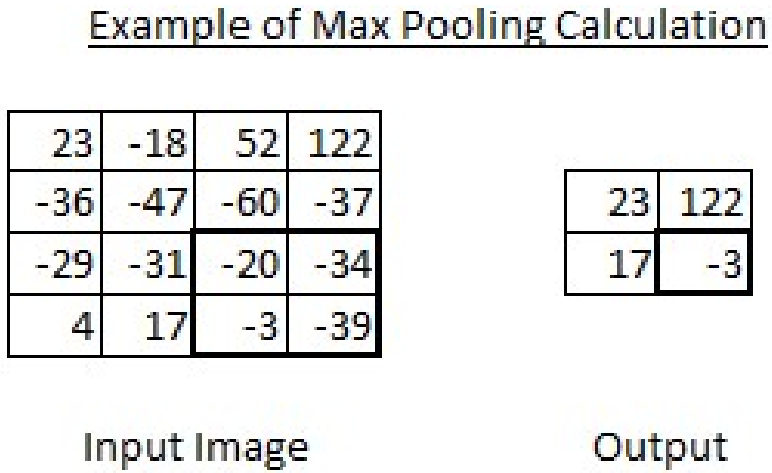


Figure: Example Calculation: $max(-20, -34, -3, -39) = -3$

CNN: Dataset 4

- Classification task: To classify 60x60 color images (i.e., 3D tensors) of cats, dogs and wild animals.
- Dataset used: [Animal Faces Color Images] Dataset comprising of animal faces consisting of 15,000 high-quality images at 512 $\times$ 512 resolution. The dataset includes three domains of cat, dog, and wildlife, each having 5000 images. Here the dataset is split into two parts: 80% for training and 20% for testing. Below are some images from the Animal Faces dataset.

- CNN architecture:
  - Input layer: The input layer consists of 3D tensors of shape 32x32x3, which is typical for color images (32 pixels wide, 32 pixels high, and 3 color channels for red, green, and blue).
  - Hidden Layer: Here the first hidden layers are responsible for higher-level features like edges, shapes, or boundaries. On the other hand, the later hidden layers perform more complicated tasks like identifying complete objects (a car, a building, a person).

CNN: Dataset 4

- Convolution : Made up of three 2D convolution layer to a neural network model. The filters used in each layer are 32, 64 and 128 respectively. Each filter is a 3x3 matrix. I have used padding as 'same' which means that the input image will be padded with zeros around the border, so that the convolution operation can produce an output of the same size as the input. Here, The Rectified Linear Unit (ReLU) function is used because it helps to introduce non-linearity into the model.
- MaxPooling: Max Pooling is a downsampling strategy used in CNNs. It reduces the spatial dimensions (width, height) of the input volume for the next convolutional layer without losing important information. It helps to make the model invariant to small translations, reduce computation, and control overfitting. The argument in Maxpooling is 2x2 which means that for every 2x2 square of pixels in the input data, the max value will be taken and the rest will be discarded. This operation is performed independently for every depth slice in the input.
- Output: This is a fully connected layer. In a fully connected layer, each neuron is connected to every neuron in the previous layer. It takes the value of an integer, in this case it's 3 which specifies the number of neurons in the layer. The number of neurons defines the dimension of the output space. In this case, the output from this layer will be a vector of length 3. Here, I have used Softmax activation function for the layer. This is often used in the final layer of a neural network-based classifier. It will convert the outputs of the layer into probability scores for each class. These scores will sum to 1, and the class with the highest score is considered the model's output prediction.

CNN : Dataset 4

- Results
  - Here is the result when the CNN was trained on the animal faces dataset for image classification for different batch sizes and learning rate using the SGD optimizer.

Batch Size	lr = 0.0001	lr = 0.001	lr = 0.01
4	0.384405	0.727086	0.347127
8	0.506507	0.757534	0.818836
16	0.370675	0.695101	0.937307
32	0.389423	0.610234	0.949519
64	0.380208	0.564236	0.964583
128	0.274256	0.452539	0.923292
256	0.293324	0.386364	0.843750

CNN : Dataset 4

- Results
  - Here is the result when the CNN was trained on the animal faces dataset for image classification for different batch sizes and learning rate using the Adam optimizer.

Batch Size	lr = 0.0001	lr = 0.001	lr = 0.01
4	0.8365	0.9655	0.3478
8	0.9058	0.9586	0.3110
16	0.8715	0.9685	0.3601
32	0.8067	0.9567	0.3620
64	0.7524	0.9688	0.3517
128	0.7783	0.9506	0.7979
256	0.6200	0.9325	0.9205

Conclusion

- We see a strong correlation between learning rate and batch size for the Adam optimizer as when the learning rates are high, large batch size performs better. Small batch size with low learning rate shows the maximum accuracy in this model. SGD shows a weaker correlation.

CNN Dataset 5

- Classification Task:
  - To classify 60x60 color images of street signs of 43 different classes

Figure: Sample image

- Dataset Used: [Street Signs Color Image Dataset]
  - 39,209 images of street signs, with 80% used for training and 20% used for testing.
- Network Architecture Summary:
  - 3 layers of Convolution and Max Pooling Layers, with activation as Relu
  - 32,32, and 64 filters set for each convolutional layer respectively
  - Next, the convolved images is input into a flatten layer and next the hidden layer with 64 units
  - Dropout of 0.5 was used
  - The output layer is another dense layer with a neuron assigned for each digit and the SoftMax activation function is used.
  - Experiments were done using adam and SGD optimizer to compare performance

CNN : Dataset 5

- Accuracy recordings from Dataset 5 using SGD optimizer.

batch size	lr = 0.0001	lr = 0.001	lr = 0.01
4	0.70360	0.648342	0.972321
8	0.35102	0.903699	0.964668
16	0.09477	0.869898	0.959694
32	0.058418	0.78699	0.945153
64	0.065061	0.423156	0.919698
128	0.069416	0.140369	0.860656
256	0.056771	0.073698	0.788542

Table: The maximum recorded accuracy in each column is highlighted.

- With the SGD optimizer, lower batch sizes yielded the best model accuracy

CNN : Dataset 5

- Accuracy recordings from dataset 5 using Adam optimizer.

batch size	lr = 0.0001	lr = 0.001	lr = 0.01
4	0.966327	0.95523	0.057398
8	0.957143	0.972321	0.057398
16	0.949745	0.973724	0.057398
32	0.931505	0.974107	0.057398
64	0.929175	0.972464	0.057633
128	0.90894	0.960938	0.057633
256	0.885677	0.955700	0.057900

Table: The maximum recorded accuracy in each column is highlighted.

- We observe the expected correlation for learning rates 0.001 and 0.01. Lower learning rates performed better with smaller batch sizes, while larger batch sizes performed best with higher learning rates. However, the model accuracy was overall very low when using a high learning rate of .01

Convolutional Neural Networks : Conclusion

- Conclusion:
  - After observing the experiments, the adam optimizer experiments showed the expected trend higher learning rates requiring higher batch sizes to optimize the model accuracy.
  - The SGD optimizer did not show this correlation, and it most benefitted from smaller batch sizes to produce the best model.
  - We observe that for categorical classification tasks using CNNs, the Adam optimizer performed best using learning rate of 0.0001 and 0.001. All experiments using these learning rates achieved model accuracies above 90%.
  - The SGD optimizer performed best for the experiments applying the highest learning rate of 0.01 when batch sizes are small. This may be due to the vulnerabilities of SGD since it is a more simple computation of for applying the gradient of the loss function.

Bibliography

Ibrahem Kandel and Mauro Castelli.
The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset.
ICT express, 6(4):312–315, 2020.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, and et al.
Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor.
An introduction to statistical learning: With applications in python.
2023.

Adam Optimizer: Backup Slide

- $$G[m,n] = (f * h)[m * n] = \sum_{j}^{n} \sum_{k}^{n} (h[j,k] f[m-j, n-k]))$$
- the adam optimizer function is applied to reduce the loss function of the model