Biomedical Research Project Report

Are critical periods in learning inherent or an extension to the neural network model?

Abstract

Neural networks are both computational and mathematical models inspired by the architecture of the biological brain. They have since been adapted to suit certain questions in computer science. This report looked at whether they were capable still of emulating the biological brain and whether that would entail the same characteristics associated with its function. Th result of this reflected a learning and adaptive ability that matched the current understanding of the brain. Flaws of the implementation of the model were suggested and further and more in-depth testing was outlined.

Premise

Neural networks were conceived as digital models of the neural circuits underlying the brain. Whilst vastly more simplified and constrained, they demonstrate an inherent aptitude for emulating the ease of learning that biological systems exercise.

Learning is a core feature at the heart of biological systems. The ability to learn, interpret and adapt to input is heavily advantageous in dynamic and competitive world. It is an ongoing process that never abates. However, central to this research project, learning is not a homogenous process and there are key periods that have formative influences on the outcome of the developing system.

In literature, these key periods are known as the critical periods of learning. While neural circuits are developing, they enter a period of hyper-plasticity wherein inappropriate or even absence of stimulation can lead to malformation of circuits and consequent impaired or lost function.

The most prominent example of this is the critical period of visual learning that occurs between 3 - 5 months of a human child's life[1]. During this period, covering an eye can produce dramatic effects on the architecture of the visual cortex. Normally, each eye's input is represented, unintegrated in the visual cortex as distinct columns of neurons that respond to a single eye. These columns, known as ocular dominance columns, exist in relatively equal proportion and are proportional to the level of significance afforded to the input of each eye. During the critical period, these columns are hyper-plastic and essentially 'compete' for visual cortex real estate[2]. The space afforded is done so in proportion to the level of activity of each eye during this period. As such, the phenomenon occurs that if input during this period is aberrantly altered in one eye, it can quickly be 'displaced' from the cortex by the other eye and that eye's unaffected input.

Neural networks competently emulate the ability to learn complex data in the vein of biological systems. Naturally then the question arises as to whether they emulate the other characteristics of biological learning. Thus, question comprises of two parts. The first: **How can a model be constructed to best emulates the biological neural system?** The second: **Is the critical period a feature of the neural network model or is it a technique employed by the brain to affect learning?**

The process by which such a question will be approached is by the selection and adaptation of an appropriate model and then the subjection of the model to digital

[1] https://www.frontiersin.org/articles/10.3389/fpsyt.2013.00146/full
[2] https://www.sciencedirect.com/science/article/pii/S0896627307007581

recreation of the biological phenomena which lead to the adverse effects associated with critical period disruption.

This report will seek to answer the first question in the approach to the task and the second in results of the model. As much of this report is the process of developing the model, the discussion regarding components of the model will be incorporated in the approach and results sections.

Approach

Critical periods have been documented in a broad range of neurological functions. Commonly affected functions are hearing, vision (as discussed previously), first/second-language acquisition. Given the aptitude of computational neural networks is best documented and refined for computer vision, this sense was chosen to explore critical periods in machines.

The Data

In the domain of computer vision, models at the forefront very much depend on the dataset, and thus necessarily the nature of the data, to which they are applied. Given the hardware constraints of the project the CIFAR10 model was selected. The CIFAR10 dataset consists of 50 000 colour (RGB) images of dimensions 32 x 32. The images fall into 10 categories of real-world objects such as Dog and Airplane. Refer to figure 1 for an example. Whilst much more complex a dataset than the original MNIST dataset examined, it offered a much closer approximation than MNIST of the nature of data typically processed by the biological visual system. This was decided on the basis that colour and object diversity better represented the complexity of

real-world visual input. It was considered important that the data was too complex to be considered atomically as the brain neither has this luxury with the real-world.

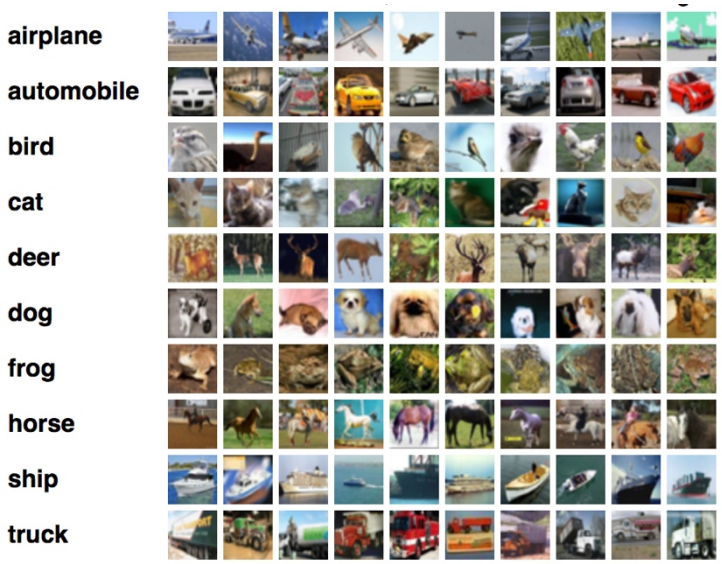

Figure 1: The classes of the CIFAR 10 dataset and ten examples of each.

The Model

Given the CIFAR10 dataset then, the current leading model variants can be broadly broken down to convolutional models, residual models and Bayesian models. Of these models, the convolutional model was selected for greatest accessibility in terms of the hardware and mathematical understanding available to the project. In choosing the variant of the model to be implemented, the Network in Network model[3] was selected on the basis of its unique architecture. Further, the intent was to implement the desired model using only the library NumPY. Ultimately, the Pytorch was used additionally for its significantly more efficient implementation of the input unfolding to perform the convolutional operation as a matrix dot product instead as well as its much more efficient MaxPool operation. However, given this (relatively) from-scratch approach, hardware constraints became particularly important.

[3] The full Arxiv paper found here: https://arxiv.org/pdf/1312.4400.pdf

The Network in Network model emulates the typically convolutional model in that it takes the image volume and applies the convolutional operation by sliding a filter across the volume. However, in place of simply a weight matrix to the current contents of the filter, the Network in Network model instead applies a Multilayer

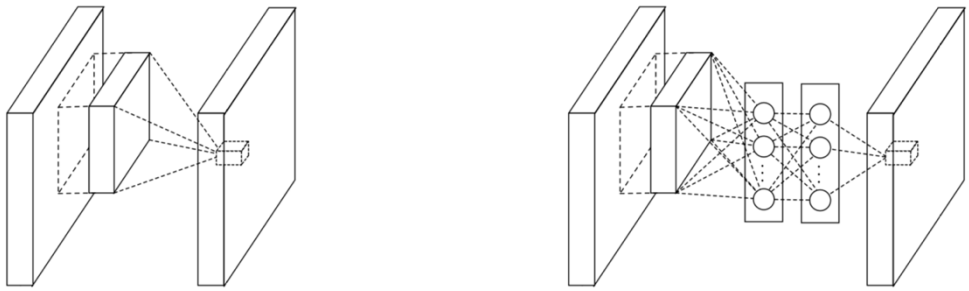

Figure 2: In a) is depicted the standard convolutional operation to derive the output of a layer. In b) this same operation with the filter contents being passed through the MLP instead. Borrowed shamelessly from the Arvix paper.

Perceptron (MLP). The MLP is the archetype of neural networks and consists only of three fully connected layers. Immediately, after each layer, a non-linear activation function is applied to allow non-linear correspondence between explanatory and response variables. In this instance, the Rectified Linear Units (RELU) function was implemented in accordance to the source paper. A terminal fully connected layer was used as is standard for the convolutional model.

Max pooling, technique commonly used in regular Convolutional Networks to reduce the dimensionality of the input, was also implemented. These max pool layers were inserted immediately after each MLP layer.

Extensions to the Model

Beyond the model, deep learning has developed to include a number of additive approaches that can be implemented on top of models to help enhance learning rate as well as generalisation of the learning to better fit unseen data.

Overfitting refers to when the model begins to tailor specifically to the content of the training data and not to the characteristics of the datatype, resulting in compromised test accuracy. For example, it can be thought of as the difference between rote-learning and more generalised concept integration. Whilst overfitting does occur biologically in some instances (think gambling and other addictive behaviours), it is an uncommon phenomenon due to the unbounded nature of the real world 'dataset.' Thus, to mitigate overfitting in the model implemented, dropout and L2 regularisation were added upon to the model.

Dropout refers to the stochastic silencing of a percentage of a layer's output. The intent is to prevent nodes (or neurons) from overly coordinating with one another and thus preventing excessively precise responses to the training data (rote-learned responses if you will). This was thought to be analogous to biology in that often in the real world, stimuli are supplied in vastly different contexts. As such, each encounter of a stimulus would lead to different circuit activations as no encounter is likely to ever be identical.

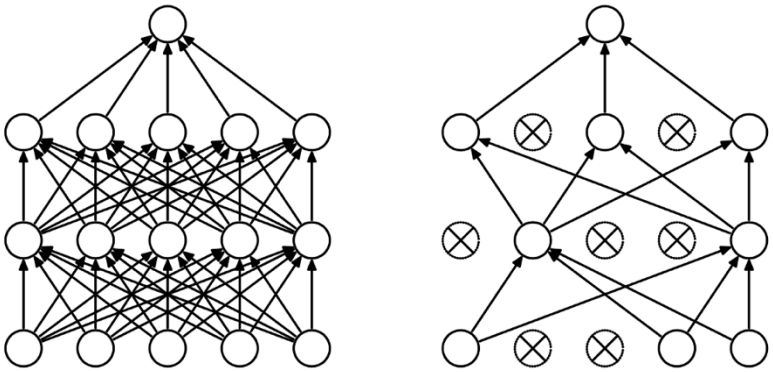

Figure 3: A standard neural network, much like the multilayer perceptrons used in the model, before (a) and after (b) dropout is used to silence nodes.

In addition to dropout, L2 regularisation is another technique that counters overfitting. It involves the addition of the following term to the loss function:

$$\frac{1}{m}\frac{\lambda}{2}\sum_{l}\sum_{k}\sum_{j}W_{k,j}^{[l]2}$$

This term ensures that update made to the individual weights of a weight vector are regularised relative to the weight vector's previous value. As such, it prevents overly specific modifications to individual weights across the vector and instead encourages broad modifications to the whole weight vector. This can be understood as specific, single weight transformations reflect rote-learning as they cannot possibly capture the varied information in the unseen data. In this way, it is ensured that more general, vector-wide updates are undertaken that will better generalise to the characteristics of the data rather than the specific training features.

In regard to improving the learning capacity of the model, techniques of learning rate decay, softmax terminal-layer activation and cross-entropy loss and the Adam optimisation were implemented.

Learning rate decay is not-overly complex. It refers to the decay of learning rate with each time it is passed over the dataset. The reasoning behind it is that as the model better hones into the optimal parameter assignments, smaller and smaller adjustments are needed to find the optimal configuration – the minima in the error landscape.

Softmax activation is a technique by which the output of the terminal layer is converted from activation strengths to a probability distribution of the likelihood, as determined by the layer, of each element of the output being the correct element. Softmax is performed in tandem with cross-entropy loss as the technique attempts to minimise cross-entropy; the amount of data needed to derive the true value given the approximation the model is for a truth-determiner of the given dataset. This was felt a better approximation for biological system as real-world information rarely exists in isolation and complete information is rarely had. As such, an individual must not only always consider the likelihood of other correct answers but learning also involves better eliminating the incorrect answers in addition to selecting the correct

answer. By employing softmax and cross-entropy, an approach was chosen that targeted both incorrect and correct classes in a more biological fashion.

Adam optimisation was the final extension of the model in regard to learning capacity. The typical models update their weights using standard Stochastic Gradient Descent (SGD). This SGD approach seeks to simply update weights in the direction of negative gradient of error (i.e. adjust weights such that the change in error continues to be negative or less error). However, this was felt a poor approximation for biological learning which is extremely explosive, converging on understanding rapidly and accurately. In efforts to extend SGD, the notion of momentum was introduced. Momentum borrows strongly from the physical concept wherein if learning is traversing the uneven error landscape, like a ball rolling down a hill, updates should gain momentum as they descend and lose it should they begin to ascend up error gradients. This enables much more rapid convergence on the minima – the depths of the error valleys. It was thought this effectively mirrored learning in biology too. Individuals rewarded by a task will continue to engage in that task and the learning of the task whereas if they begin to struggle, they change tracks; they lose momentum. However, Adam optimisation goes beyond that. It adopts a very much more biological route of what is akin to sensitisation and desensitisation. It updates weights on a per-parameter basis wherein the magnitude of each parameter's update is relative to its history of update. As such, Adam optimisation was implemented as the intersection of momentum and neuronal sensitisation.

Extensions to the Dataset

Beyond simply the model, the data on which the model is trained can be modified to serve as a better and more general learning substrate.

Whilst much of this project was orientated towards the model itself, some modification of the data was undertaken.

The most standard modification was mean subtraction. Given the overall dataset, the mean pixel values were subtracted from each image. This is simply a mathematical benefit of centring the data around 0. From a biological perspective, this can be interpreted as the highlighting of only the differences from the norm rather than the data as a whole. Given the limitless nature of information in the field of vision, the brain has the task of only addressing the important elements.

Normalisation is an extension to this. It normalises the values such that they still span from -1 to 1. While also a mathematical tool, it was implemented as it underlying purpose is to ensure that all images are considered equally important regardless of pixel intensity. The information is still just as valid and as such, less intense images should be in no way considered less informationally significant. This is identical almost in biological settings where the task of visual recognition is no less significant if the image is shadowed or in full daylight.

To summarise, the model implemented is reflected in the following image.

```
layers = [
    {'type': 'mlp', 'k': 192, 'updateMethod': 'adam'},
    {'type': 'pool', 'method': 'max'},
    {'type': 'mlp', 'k': 192, 'updateMethod': 'adam', 'dropout': 0.5},
    {'type': 'pool', 'method': 'max'},
    {'type': 'mlp', 'k': 192, 'updateMethod': 'adam'},
    {'type': 'pool', 'method': 'max'},
    {'type': 'output', 'k': outputSize, 'updateMethod': 'adam'}
]
```

Three MLP layers were used with a terminal fully-connected layers. *k* represents the number of nodes in each of the fully connected layers of a MLP. Each MLP made use of filters of size 3 x 3 and strides of 1. A padding of 1 was used to ensure complete coverage of the input. Between each MLP layer, a max pool layer was inserted. The final layer had 10 nodes to represent the 10 classes of output. Dropout was applied to the middle layer, the justification for this was to allow later layers to accommodate

for the silenced nodes, i.e. to generalise. Adam was the update method used on all updated layers in the network.

The learning parameters were as follows:

```
# Learning parameters
lr = 0.0002
l2Reg = 8e-6
learningRateDecay = numpy.float32(996e-3)
```

For the code, feel free to contact the author at:

waltersc@student.unimleb.edu.au

Results

Moving to the second question of this report, the presence of a critical period was tested for by 'blinding' the network in one eye and measuring performance.

The means by which was done was twofold. First, the input image of 32 x 32 x 3 (3 representing the colour channels of the image) was duplicated across the horizontal axis (the middle dimension). This resulted in a mock binocular image of 32 x 64 x 3. The network was then run over this new dataset for 10 epochs but was blinded at a certain epoch in the range of epoch 1 to 3 for a duration of 1 to 3 epochs. The effect of this blinding was then measured in the measured train and test values after each epoch the network was run.

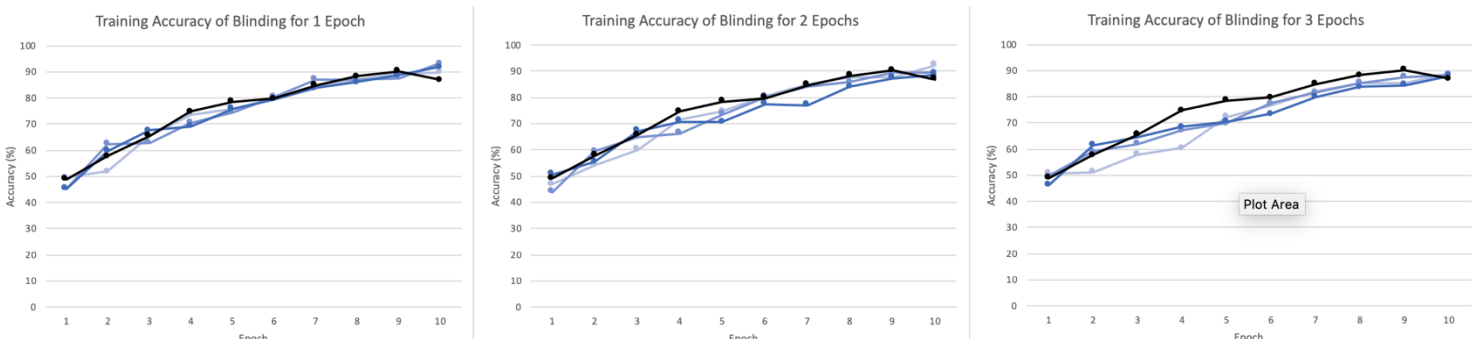

Figure 4: Training accuracy of the model given the duration of blinding for each epoch blinded in. Black represents the baseline, unblinded model's performance. Accuracy was measured as percentage of a randomly selected 5000 image dataset from the training images for which the model predicted the class correctly.

The results of this are represented in Figures 4 and 5. In respect to training accuracy, blinding of all durations and at all epochs resulted in a reduction in learning rate, not accuracy, immediately 1 epoch afterwards. Whilst accuracy is never reduced, the rate of learning is immediately reduced noticeably, causing reduced change in accuracy compared to baseline the epoch following the blinding.

Notably, this effect is not present in the test accuracy results. Instead, excepting blinding at the first epoch, blinding at all other epochs for resulted in non significant change in test accuracy.

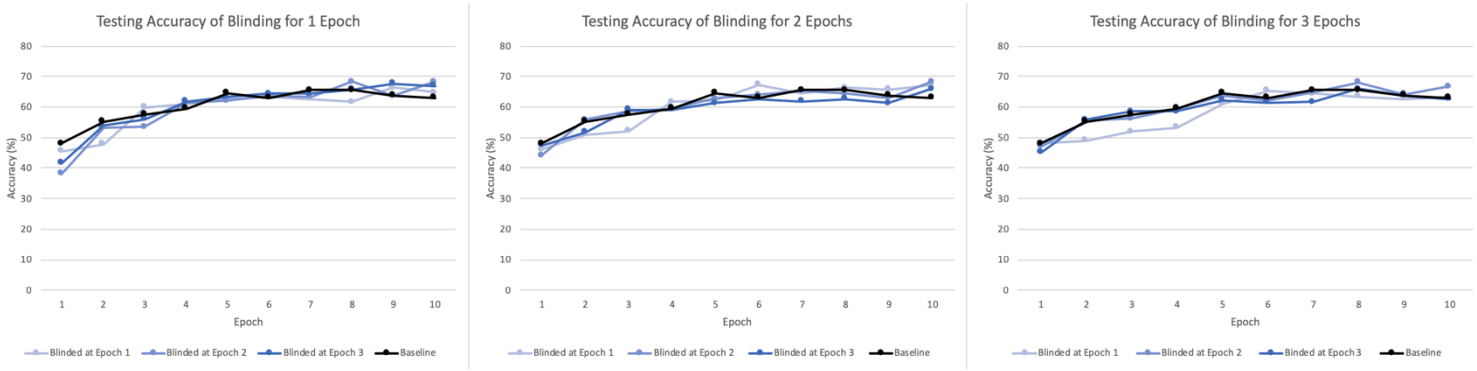

Figure 5: Testing accuracy of the model given the duration of blinding for each epoch blinded in. Black represents the baseline, unblinded model's performance. Accuracy was measured as percentage of the 10000 testing image dataset for which the model predicted the class correctly.

Returning to the training accuracy results, blinding for a single epoch produced no significant change save for blinding in the first epoch. Blinding for 2 epochs however produced immediate but short-term consequences with the models quickly recovering regardless of what epoch the blinding occurred in save in epoch 1. Blinding in epoch 1 for 2 epochs produced a persistent reduction in training accuracy comparative to baseline and even the other blinded models. However, even this model eventually returned to baseline accuracy by epoch 6. Blinding for 3 epochs produced persistent reductions in accuracy compared to baseline for all points of blinding. Models still converged to baseline accuracy by epoch 10 yet demonstrated reductions in accuracy that lingered beyond the blinded period.

Ultimately, these results demonstrate the significant reductions in training can be produced by blinding at all epochs 1 - 3, given a duration of at least 2 epochs.

However, regardless of duration, blinding failed to reduce test accuracy unless blinding occurred in epoch 1. Given this, it is evident that the first epoch is the most significant period for the development of the model's vision. Whilst fine accuracy, i.e. that demonstrated in the training accuracy is impaired by blinding at any epoch, generalised accuracy reflected in testing accuracy is only affected by blinding from the epoch.

Despite this, the model was capable of recovering from all insults, converging to baseline accuracy by epoch 10 in all instances. However, this can be seen as reflective of biology. The critical period has been shown to affect vision from a single eye but in those affected individuals, sight is not impaired to any extent beyond stereopsis. In this way, this computational model emulates tea ability for neural networks to compensate for alterations to input. The reduction in accuracy displays that the insult has an effect and the recovery indicates the capacity to recover from the insult.

Further, the effect is primarily evident only shortly after the blinding is initiated and does not persist long past the alleviation of blinding. The return to baseline suggests that not only is the system compensating but it suggests that the system is able to regain use of the blinded 'eye' as it replicates the binocular baseline results. One explanation is the Adam optimiser and l2 regularisation used by the model. This optimiser uses a per-parameter update approach outlined earlier that increases the significance of errors to not recently updated parameters. In this instance, the parameters of the blinded 'eye' will have been neglected in the update period due to throughput values of 0. Further the l2 regularisation ensures that the updates of parameters are diffused across the weight matrix entire. The result would therefore be that during the blinding they still received some update due to L2 regularisation (preventing absolute zeroing and loss of those neurons) and once blinding was lifted, Adam optimisation would ensure their learning rate accelerated to bring their errors' in line with the unblinded parameters.

Beyond

The first question was answered satisfactorily but left plenty to still be answered. First, as indicated in the results, it was suggested that techniques of L2 regularisation and Adam optimisation impaired the emulation of biological systems. There would be much benefit in examining the performance of a model that operated in absence of 1 or both of these techniques. Further, a key model flaw that was identified late in the process was that lack of distinct ocular paths. Whereas in the visual system, each eye's input is kept separate until the cortex, that is not the case here in this model. One suggested solution would be two neural networks that converge on a final MLP and the fully connected layer. Given the hardware available, this would not have been effective to test but nonetheless, the immediate mingling of input by the means of a common filters applied to both inputs and the crossover of filters that capture the midline of the input (thus capturing part of both inputs simultaneously) caused the model to stray from true biological emulation.

However, more importantly, the second question is only partially answered. There is indication that the critical period lies primarily in the first epoch. Whether this critical period results in long-term impairment is not entirely answered. Here, it is indicated it does not result in perpetual loss in accuracy. A further avenue for exploration would be the examination of the backquerying of the network post-blinding. While this was initially envisaged, the complexity of implementing the network made it difficult both time-wise and knowledge-wise to implement a backquery capacity.

In summary, the behaviour of the model indicates that the first question was satisfactorily achieved. The capacity of the model to learn and predict the CIFAR10 dataset indicates success in this aim. In regard to the second question, the model demonstrated the biological ability to adapt and evidence suggests that there was indeed a critical period in the first epoch. Further exploration into the question is needed however for a more definitive result.