

**Classification of Trichostrongyle Eggs in Ruminant Fecal Samples using a Back
Propagating Neural Network**

Matthew Drennon Allen

matthew.d.allen@okstate.edu

Under Thesis Director Dr. David Cline

david.cline@okstate.edu

With Second Reader Dr. Christopher Crick

chriscrick@cs.okstate.edu

Data provided by Dr. Susan Little and Dr. Anne Barrett

susan.little@okstate.edu, anne.barrett@okstate.edu

Acknowledgments to all of the OSU Parasitology Lab

Table of Contents

1. Introduction

- a. General Problem (Pg. 3)*
- b. Trichostrongyles (Pg. 4)*
- c. Current Techniques (Pg. 5)*

2. Methodology

- a. Assumptions About the Data (Pg. 8)*
- b. Input to the Network and Preprocessing (Pg. 10)*
- c. The Network itself (Pg. 13)*

3. Data

- a. Raw Data (Pg. 15)*
- b. Enrichment (Pg. 16)*
- c. Final Data (Pg. 17)*

4. Experimentation and Results

- a. Preliminary Trial (Pg. 18)*
- b. Intermediate Trial (Pg. 20)*
- c. Final Trial (Pg. 21)*
- d. Conclusions (Pg.23)*

5. Further Study (Pg. 24)

6. References (Pg. 26)

Introduction

Overview

The purpose of this thesis is to demonstrate the potential efficacy of using pattern recognition by way of a back-propagating neural network (BPNN henceforth) to count the number of parasite eggs present in a given prepared fecal sample.

The problem of counting parasite eggs in fecal samples is a very relevant one, even in a modern livestock economy where it is assumed that major producers administer regular anti-parasitic 'de-wormers' to their flocks and relatively stringent guidelines are followed in the housing and husbandry of livestock. In modern farming, many of parasite related issues are not caused by neglect, but rather by worming plans that are aimed to be cost effective in the average case, but are performed blind and without accurate diagnostic information as to the true parasite problem. This largely due to the fact that getting an accurate count of the worm count for a given herd is a time consuming task that, while inexpensive in terms of material, takes skilled labor.

In Australia, the third largest sheep farming industry in the world, a 1995 review showed that their industry lost a combined sum of 222 million USD in losses from reduced wool and meat production, as well as animal mortality, caused by parasitic worms. (McLeod, 1995) Combining their losses from cattle parasites brings the total figure close to one billion dollars in losses, and it's estimated that the global figure is in the range of tens of billions of dollars in losses. (Florian et al, 2013)

These losses are not entirely preventable, but they can be alleviated through the use of

various medical interventions and careful husbandry. However, those practices themselves cost money, and thus the problem relies in accurately determining when intervention is needed. If the treatments are administered when not needed, then money is lost on unnecessary medical treatment. While certain passive measures, such as rotational worming, are common practice in the industry, these techniques represent a form of blind treatment where the owner has no real idea whether the measures were needed or not. They represent the best current practice in an environment where accurate data is unavailable, but as stated previously, they are heuristically based on previous trends- typically discovered via previous losses.

Trichostrongyles

The worms that are the basis of this study are trichostrongyles, which is a more common word for all the worms in the family Trichostrongylus. They're all parasitic roundworms, and infect wild and domestic herbivores as well as, for at least ten species, humans. In livestock they lead to diarrhea, fatigue, weight loss, and degenerated coat quality and milk production due to malnutrition. They are a well known threat in livestock, and there are a variety of products designed to purge them from the host, known as wormers or de-wormers interchangeably.

The trend of trichostrongyle infections is heavily seasonal, with the highest burden occurring during spring, and the weakest, as expected, during winter. Summer and fall worm burdens vary between locations and years. A mild summer or a warm autumn can lead to unexpected worm burdens at unexpected intervals, which in turn leads to unexpected mortalities due to conventional de-worming patterns not accounting for atypical seasonal weather shifts.

The life cycle of a trichostrongyle is straightforward. The eggs of the trichostrongyle are excreted by the host in fecal matter, hatch within the fecal matter and molt twice. After the second molting they are able to move from the fecal matter and into the soil and grass. If the weather conditions are not too severe, it is possible for the larvae at this stage to survive for up to a year before being consumed by an animal as a 'rider' on (typically grass) forage. The larvae then reaches adulthood in the GI tract of the infected animal, where it will breed and produce more eggs to continue the life cycle. (Ghasemikhah et al, 2011)

While it is difficult to non-destructively count the number of mature worms inside an animal's stomach, the life cycle and expected production rate of the parasite is well known, and thus an accurate estimation of the number of parasites that are currently in the animal's gut can be readily calculated from the number of eggs found in a fecal sample. The method used in this study to determine the fecal egg count is discussed next.

Current Techniques

Accurately diagnosing parasite load in order to determine optimal intervention times and procedures is typically done through one of two procedures: The fecal float, or an autopsy. The fecal float is the preferred method of diagnoses since it can be effective before the herd begins experiencing fatal worm burdens. For this reason the fecal float is the method used in this study, with prepared floats acting as the sample from which the raw image is drawn. The fecal float is a simple process whereby a small amount of animal stool, typically a tablespoon or less, is placed inside a prepared vessel. A special solution, known aptly as fecal float solution (the actual compound used varies, sodium nitrate is common) is then added to the vessel containing the fecal matter, and the vessel is lightly agitated. The fecal float solution

causes the parasite eggs within the fecal sample to float to the surface as they more buoyant than the surrounding material. A microscope slide coverslip is placed on top of the fecal float and left in place for a few moments. Upon removal, the coverslip will have a number of eggs, as well as low density detritus consisting of partially digested roughage, adhered to its surface.

The coverslip is then adhered to a specially marked microscope slide, cleaned, and then placed under magnification. The special markings on the microscope slide form a grid (typically 10x10), where each cell in the grid represents a constant volume of solution, and a random number of cells (typically 10 or 20) are chosen for detailed inspection. On inspection, the laboratory operator examines the selected cells individually and counts the number of eggs they contain. From this subset, the total number of eggs per cell is estimated. This, combined with fecal volume and the float volume yield the total number of parasite eggs per unit volume, and thus the total worm burden of the animal in question can be inferred. When done on a sufficiently large subset of the herd in question, this allows a detailed reckoning of the actual worm burden of a given herd. (Gadberry, 2011)

The first part of this process, preparing the slide, takes ten or twenty minutes to yield a viable slide, but requires less than minute of human attention and multiple samples can be run in parallel. As long as the captured slides are kept cool to prevent the eggs from hatching, they can be stable for days. Creating the float is not the labor intensive portion of the fecal float, rather, it is the counting of the number of eggs in each selected cell.

As one can understand, this process is time intensive. The technology needed has become less of an issue, in fact recent advances in magnification techniques have allowed a specialized but inexpensive cell phone lens and a bright light source to allow iPhones to take

pictures of the needed resolution for a fecal egg count. Unfortunately, the process still requires the focus of a human to prepare the slides, and of a trained individual to correctly count and classify the parasite eggs in the sample. While the preparation of the fecal float itself is beyond the scope of this study, the counting of the eggs holds the possibility for straightforward automation. The counting of the eggs from a subset of grid cells can take between 5 and 15 minutes per sample for a human, and is the only part of the process that requires a specially trained technician.

This time constraint is a large part of the reason why large predictive testing is impossible. As stated before, Australia suffers more than 222 million dollars of losses annually from worms alone in their sheep industry, which consists of approximately 74 million head of livestock. If we assume that sampling one individual in a hundred will give us an accurate picture of the worm burden of the population, then Australia would need approximately 740,000 tests run annually, taking a mean time of 10 minutes per slide, not counting preparation, results in a requirement of 123,000+ man hours of trained labor to get an accurate picture. Since the review would need to be done in the same month as the worming to make the statistic viable, this would require an impractically large force of technicians.

But what if it only took a second to run the count, instead of ten minutes? What if the count, instead of requiring a highly educated individual, could be done by an untrained individual using a lightweight computer program?

The process described in this thesis begins at the microscope stage, where, instead of beginning a manual count, the technician can simply take an image of the slide and send it to the program for processing, freeing them for other activities. While this algorithm cannot take

position of a skilled human being, it may be able to greatly reduce the time required for a simple and menial task, thus reducing the overall cost of the test, allowing more tests to be done in parallel, and increasing the accuracy of the test by always scanning every cell instead of scanning only a subset.

Methodology

Assumptions on the Data

While the scope of this study is strictly limited to building a neural network capable of deciding whether a particular image is, or is not, a trichostrongyle egg, it makes two assumptions about the collection of the data.

The first assumption is that there is an original image of reasonable resolution that contains all eggs needed for the count, and is reasonably well focused to examine the float-layer on which the eggs reside. These restrictions are not onerous, as they are already needed for a human to perform a manual count. In fact, as will be demonstrated later, the algorithm is capable of making use of substantially lower resolutions than are required by the human eye in order to make a judgment about whether a particular object is an egg or not. As such, while this assumption is critical, it is already met and exceeded by the current methodology for counting eggs in fecal samples.

The second assumption is that there exists some algorithm that takes the whole image and separates it into background and 'significant' units. For our purposes, this algorithm will be referred to as the separator. The separator makes no judgment about what a particular object in the image is a worm egg, or an air bubble, or a piece of pollen. Its only job is to find regions

that differ from the background, surround them with a bounding box, and pass them into the neural network pre-processor as input. While this separator is not implemented in this thesis, searching an image for discontinuities and then bounding them is a common feature on commercial image processors such as GIMP and Photoshop, and thus it is not unreasonable to assume that such a program already exists, or could easily be created.

This separator need not be particularly accurate, as shown in figure 1, the boundaries are imperfect and may contain the edges of other egg or non-egg objects. The only stipulation is that the entirety of the object that might be an egg is included.

For the purposes of this thesis, the separation of significant objects from the raw image was done manually, as will be seen in the data processing section. An example of two outputs from this simulated separator is shown below in figure 1.

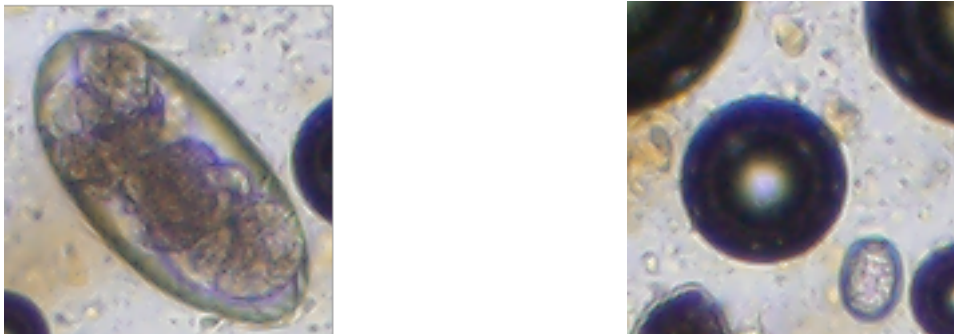


Figure 1: Two examples of images that the separator should produce

If these two stipulations are met, then we can assume that the input data consists of a series of sub-images of varying sizes. Once we have that series, we can feed it into the BPNN's pre-processor.

Pre-Processing

The pre-processor's job is to first compress the image to a constant resolution, thus

allowing the input layer of the neural network to be held at a constant size, and to convert the picture from an image format (.png, in this case) to a list of 3-tuples, where each tuple represents the RGB normalized values from one pixel.

Lowering the resolution is a critical step in this process. First, it forces all the images to known size that's constant across all images, thus making the length of their per-pixel RGB breakdown constant across all images. Second, it reduces the number of inputs to the BPNN to a reasonable level. The original image data given is shown below in figure 2, and represents a 1200x1600 image

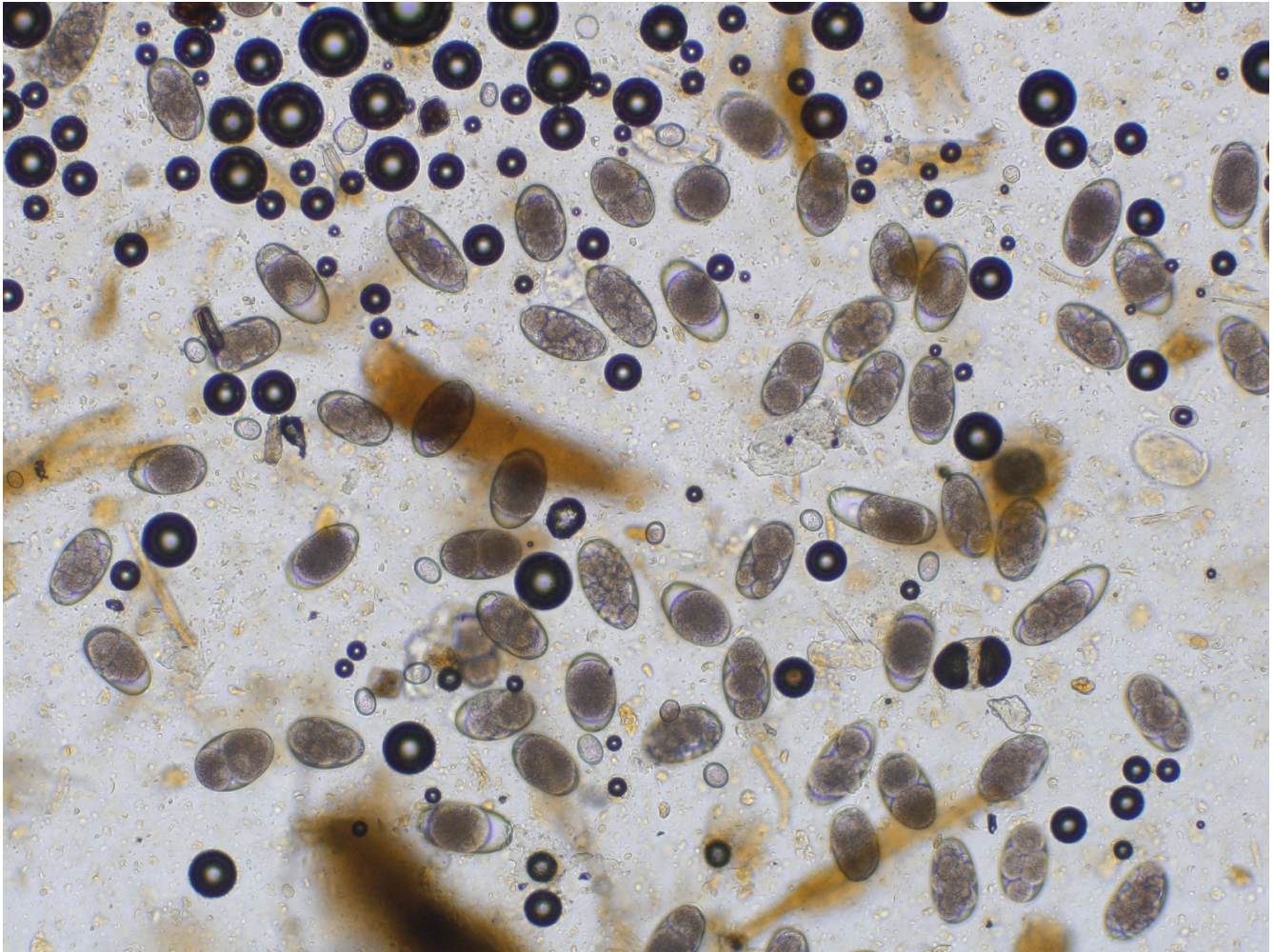


Figure 2: Raw image data. Near-circular objects with black walls are air bubbles. Large ovals with gray material and distinct cell walls are trichostrongyle eggs. Small clear ovals with cell walls but no visible internal structure are harmless Eimeria eggs.

An object of interest from this image could easily be 125x125 pixels in size, yielding a total of 15,625 pixels, each of which would produce three values when broken down into their respective RGB composition, yielding more than 45,000 inputs to the neural network. While it's quite possible that a network of this scale would be capable of great accuracy, it would be unusably slow on most low power systems, and for our purposes, unnecessarily fine in its detail. In order to minimize the size of the BPNN's input layer, while still preserving sufficient

image information to make meaningful classifications, iterative resizing using bi-linear interpolation was employed to bring the images down from their base size down to a 4x4 image. Iterative bilinear, as opposed to direct transform bilinear, was used in so that each quadrant in the final image would accurately represent an average of the original quadrant, instead of representing a particular region of the original quadrant. This 4x4 image reduces the pixels the algorithm needs to process to roughly one-tenth of one percent of the original on average, and also strengthens the argument that the neural network can accurately classify eggs at a much lower resolution than the human eye is capable of. An example of this resizing is shown in figure 3.



Figure 3: Original image (left), resized image (center), resized image expanded to show detail (right)

After being reduced to a 4x4 image, the 32 bit RGB value was extracted from each pixel, and twiddled to extract the R G and B components separately. Each value was divided by 128 to bring it into the region of (0,2) for the purposes of BPNN input, and finally each 3-tuple from every pixel in the new image was concatenated into a single set of 48 values. This string

was then padded with a leading one to make it properly formatted for use in the input layer of the BPNN, and, for the purposes of scoring during training, another additional bit was appended to indicate whether the sample represented an egg or a non-egg object of significance. This string was then used as the input to the BPNN in order to represent the original image.

The Network Itself

The BPNN used was built by the student, using no pre-existing neural networking toolboxes, and is capable of creating a network with any size of input layer, an number of hidden layers, and any number of neurons within the hidden layers. This network was designed to support drop-out, and later added a new feature that will be referred to as stimulus. The network is built from a series of thresholding logic units using the logistic sigmoid activation function (eq. 1)

$$(eq. 1) \quad a = 1 / (1 + e^{-x})$$

Where x is equal to the sum of all of the inputs to the TLU multiplied by their respective weights. (eq. 2)

$$(eq. 2) \quad x = \sum_{i:\text{length}(w)} w_i * r_i$$

Where r is equal to the set of all inputs to the TLU. In the case of multi-layer networks, this is the sum of all outputs from the preceding layer.

In order to reduce over fitting in the network, Srivastava's algorithm was used to

implement drop-out, effectively removing neurons at random from the network and rebinding the connections between layers as needed. (Srivasta, 2014)

Finally, as will be discussed later, an additional stimulation capability was added to the network. Essentially, as the network enters a steady state where the error in the validation set is no longer significantly decreasing, there is an increasing chance for the learning rate of random neurons within the set to be reset to their original value of 1. Otherwise, the learning rate of each neuron is governed by the function in (eq. 3) for initial trials.

$$(eq. 3) \quad l = 1 / (10 + \sqrt{epoch})$$

Where one epoch represents the network trying every value in the training set once. Later trials showed that the learning rate was too low, and could lead to the algorithm getting stuck in an early minimum within the first ten or twenty iterations. To combat this, a revised learning function was used, as shown in eq. 4.

$$(eq. 4) \quad l = 1 / (1 + \sqrt{epoch})/3$$

The division by 3, as well as the use of 1 as a starting value as is standard, is discussed in the results of the intermediary trial.

Stimulus was another technique added to keep the neural network from getting stuck, except instead of trying to keep the algorithm from getting stuck early on, as the increased learning rate was designed to do, stimulus was used to agitate the network when it fell into a steady state in later iterations.

Under normal circumstances, the BPNN would terminate when the error in the validation set failed to improve from its last best value for 50 epochs. (This value was later extended to 100 epochs, after the results of the intermediate trials) If stimulus is active, ever

epoch without improvement adds a chance for a TLU to reset its learning rate back to epoch 1 when activated. The chance for this to occur is shown in eq. 5.

$$(eq. 5) \quad S_p = (S_{max}/T_{lim}) * T_{cur}$$

Where S_{max} is the maximum possible stimulus rate, T_{lim} is the max number of epochs that can pass without improvement before the program terminates, and T_{cur} is the number of consecutive epochs that have passed without improvement.

In essence, as our certainty grows that algorithm is not going to break through on its own, we increase the chance that any given TLU will make an unexpected jump, potentially to a more favorable position, or at least to where a more favorable position can be found. Because the neural network already saves off its last best location, chaos near the end of training, particularly when T_{lim} is large, has very little chance of negatively affecting the model fitness.

Data

Raw Data

All data was provided by the OSU parasitology labs, with the assistance of Dr. Susan Little and Dr. Anne Barrett. Unfortunately, due to constraints both on time, available information, and what information is allowed to be released, no fecal samples were run exclusively for this project. Instead, a stock image, shown in figure 1 was used. This constraint was recognized as providing a potentially narrow sample pool, and while plans were in place to expand it, the schedule of the OSU veterinary hospital did not provide sufficient opportunity to acquire new data. The image provided, however, was of excellent quality, and contained a relatively dense population of eggs in late or middle stages of maturation. (See Figure 1) The perfectly spherical objects with variable size, white centers, and thick dark exteriors are air

bubbles present within the sample. The consistently sized oblong or spherical objects with gray internal structures and a clear outer barrier of variable size are the trichostrongyle eggs that we are attempting to classify. The much smaller oval shapes with very little visible internal structure are Eimeria, which were not to be counted for the purposes of this scan. The other colored objects in the background of the sample are submerged pieces of macro-material from the fecal sample, and should have no effect on the classification of the samples.

Within the image, there were 58 eggs that were deemed to be recognizable. That is, a human operator could, with no ambiguity, declare that they were trichostrongyle eggs. However, 58 is not sufficient input to train the neural network, let alone to construct training and validation sets. Because of this, data enrichment was employed to increase the number of usable samples to a number viable for classification by the network.

Data Enrichment

Through the usage of a very rotation algorithm, these 58 eggs were transformed into 1,392 different 'correct' training samples. A matching number of 58 (again enriched to 1,392) 'incorrect' training samples was also created by sampling objects that were definitely non-background, but did not represent trichostrongyle eggs. Together, the two sets form a training/validation/testing pool of 2,784 samples. This pool provided sufficient sample data to draw training, validation, and testing sets from. Symmetric collection and enrichment of incorrect samples also allowed us to assume that, in the base case, an untrained network should have a coin flip's chance of predicting a sample correctly. It also meant that we could use the binomial distribution to determine our standard deviation and significance levels. Eq. 6 gives the formula for the standard deviation of any of the training, prediction, or testing set, given

this.

$$(eq. 5) \quad SD = \sqrt{0.25 * n}$$

Where n is the number of samples used in that particular set, whether that set be the training, prediction, or validation.

Enrichment was performed using sample rotation. Each one of the unique eggs was put through twenty-three fifteen degree rotations and saved off as a new copy between each. Each one of the rotated images, in addition to the original, could then be passed as a normal image into pre-processing. This process yielded 23 'new' samples and a single original. Because the algorithm receives input pixel by pixel, and because of the extreme compression factors used, Most eggs, when rotated any significant amount, yield substantially different input to the BPNN. An example of this enrichment is given below in figure 4.

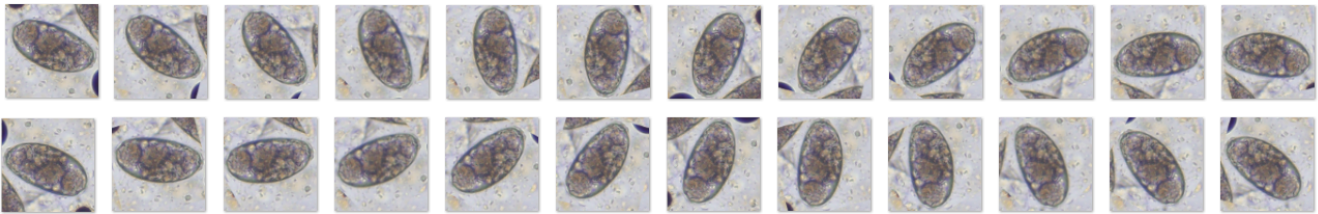


Figure 4: Examples of Rotational Enrichment

Final Data Set(s)

After pre-processing, the dataset is saved into a text file as a formatted input to the BPNN. For this, the test file for the complete data set is represented by a table of 50 columns (An entry for egg/not-egg, an entry to pad the data with a leading 1, and 48 entries to represent the normalized R G and B values for each pixel) and 2,784 rows, one for each sub-image.

Included in the program is the ability to randomize the order of a dataset, either based on

a given seed value or from a random one. This ability is particular handy when separating a data set into validation, prediction, and training groups.

For our tests, the training, validation, and prediction sets were separated by hand, just by taking rows from the end of the dataset and moving them to new files. Because of the ability to randomly re-shuffle the dataset, taking from the end was a viable strategy for selecting random samples to insert into the testing and validation sets.

Initial Trial

In order to assess the efficacy of the first principles approach to the problem itself, a small assessment set was created of 960 samples.

After normalization the set was divided into a training set of seven hundred, a validation set of two-hundred, and a testing set of sixty samples. More sophisticated methods of dividing the sample space were considered, but ultimately rejected in the immediate interest of saving time in order to finish the preliminary report. The order of these samples within their respective set was carefully randomized. The neural network in question did not train by epoch, and it was important to break up blocks of enriched data in order to remove patterns within the input data.

The neural network consisted of three layers. The input layer consisted of the 48 nodes representing the input from each vector. The first layer consisted of 75 neurons, the second layer consisted of 50 neurons, the third layer consisted of 30 neurons, and the output layer consisted of the two neurons representing our binary classifier (either an egg, or not an egg).

Results

Despite the small data sets used, the algorithm achieved remarkable success in its preliminary run. After 136 iterations, the error within the training set fell to 13%, the error within the validation set fell to its minimum value of 17%, and the corresponding value for the error within the testing set was 20% at the same iteration. (See figure 5)

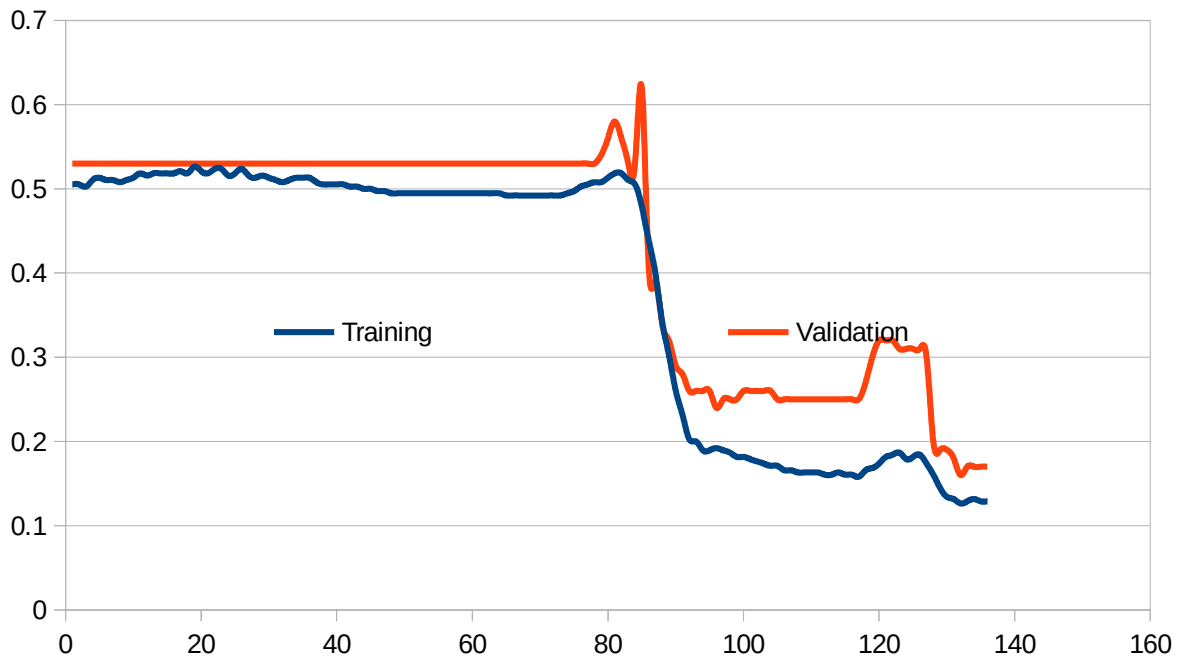


Figure 5: Graph of the error of the training and validation set against the iteration count

While this error is significantly higher than what we need to see in order to replace a human operator, it was a strong sign that there was merit to our initial choices of topology

Expanded Trial

After the success of the first trial, the full dataset was enriched to 2,784 total samples, and the network re-run under the expanded conditions. The new validation set consisted of 2004 samples, the new validation set of 680 samples, and the testing set of 100 samples.

Unfortunately, while it provided far greater training input, multiple runs through the neural network revealed that there was a relatively low chance for the network to get a 'good' seed. For approximately four-fifths of the runs attempted, the error in the validation would drop quite slowly before leveling off into a steady-state.

Observing the pattern of errors, and noting that, in order for the successful preliminary

trial to 'break-through', it had to go through a number of steps that actually increased the error, both the learning rate and the limit for the number of steps the network could take without decreasing the validation set error were increased. After this modification nearly all trial runs on 75,50,30 spread were successful. The results of one of these can be seen in figure 6.

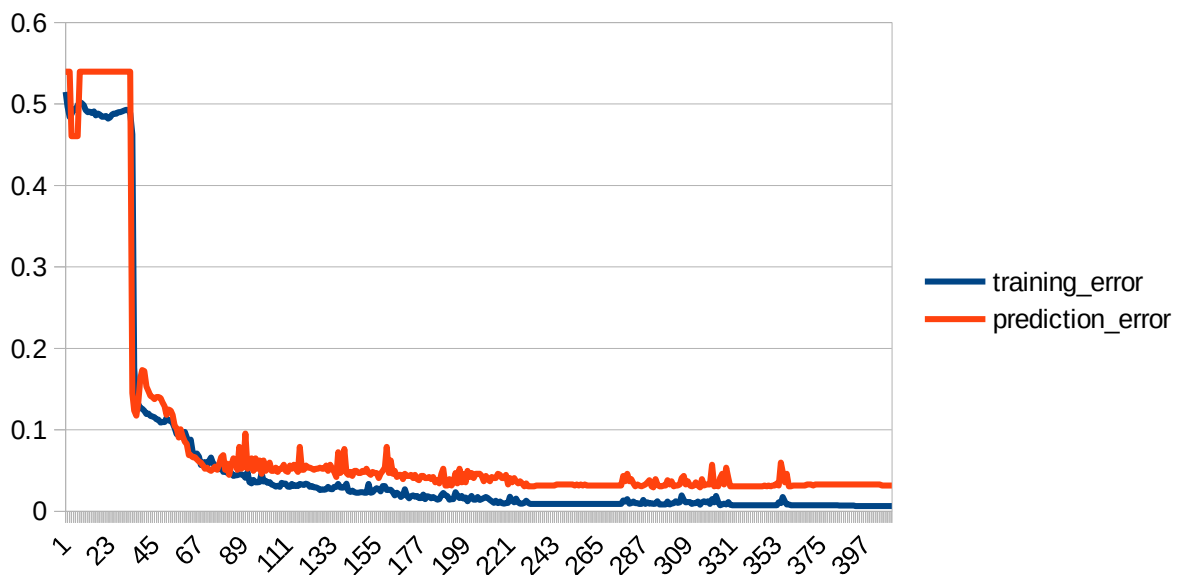


Figure 6: Expanded Run

For this run, the error in prediction dropped to 0.6%, and the error in validation dropped to 3%, but the error in the testing set remained in the double digits at 11%

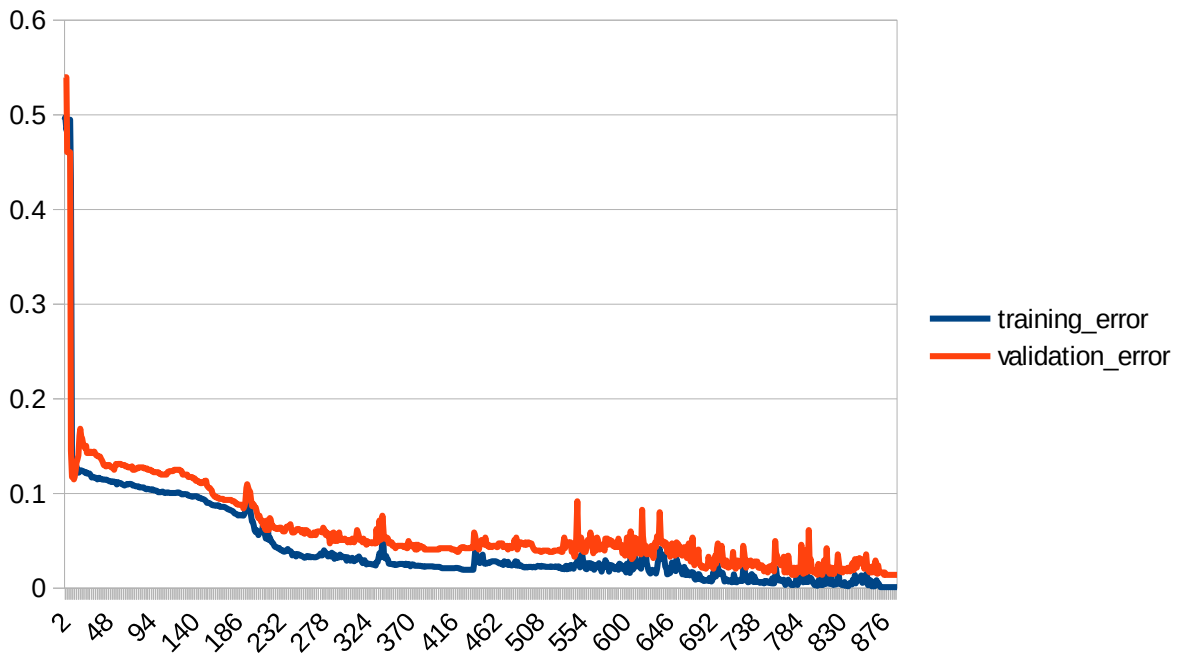
Final Run

Considering that the original network topologies were picked based only on the size of the dataset and a general impression of what had worked for other datasets used to test the efficacy of the BPNN, I manipulated topologies extensively, and found their configuration one

of the primary deciding factors in whether a run would converge properly. Specifically, topologies consisting of too many hidden layers would often make limited progress before converging to a sub-optimal solution. Too many neurons in a given layer had a similar effect, reducing the number of effective runs. This in mind, the total number of neurons was reduced, and multiple runs were attempted using two hidden layers instead of three.

More importantly, the coefficient of drop-out was increased to 0.3 (from 0.1) in order to better compensate for over fitting reduce the error in the testing set. This is also the stage at which stimulation was added to the network in order to explore whether adding an additional factor of chaos could help us shake out of a local minima near the end of a run.

The final topology used was a two layer network, of 60 and 50 neurons respectively. The results of this network are summarized in figure 7.



For this run, the error in training dropped to 0.1%, the error in validation dropped to 1.5%, and the error in testing was only 3.2%.

Conclusions

There are some fascinating conclusions that can be drawn from the success of the neural network's classification of the egg data. Most interesting is the compression factor used during the image processing stage.

The complexity of the topology of the neural network grows quadratically with the side length of the input image sample. The current 4x4 system requires 48 input neurons alone, a 5x5 grid would require 75 neurons, a 6x6 grid would require 108 neurons. Attempting to use the original 125x125 image in the network would have required an input layer consisting 46,875 neurons- a completely infeasible number. Because of this, we attempted to minimize the size of this pattern via scaling. This scaling, in essence, simulates working on a much lower resolution image where much of our data has been compressed down into a much smaller space.

While larger grid sizes remain untested, the fact that we are capable of making a successful preliminary classification using 16 pixels instead of 15,625 pixels is an enormous win for the work. It means that we're using approximately one tenth of one percent of the resolution available to us, which bodes very well for the classification on images of a lower quality than the one we received. Importantly, this also means that it is likely that we can radically increase our match rate by increasing the size of the area we are scaling down towards.

Increasing the size of the Data Set had the expected results, generating a corresponding decrease in the error across all categories, still it seems likely that our current results are deceptively low. Because of the small nature of the original data, and the inherent correlation

(if minor), it is likely that the errors in classification will increase when more data is added, but this is a problem that can likely be dealt with by improvements to the algorithm or to the pre-processing.

Further Study

There are numerous obvious avenues for the improvement of these algorithms. One of the most important things to begin work on is the chunking algorithm. However, this piece of the program is expected to take up a significant chunk of time, and the effort might be better served in optimizing the neural network and the background processes while using manually chunked data.

The neural network used in this report was designed and written by the student, and as such may contain flaws. Switching to a neural networking package for java, such as Neuroph, may provide access to significantly more powerful systems at the expense of transparency and retraining time.

Further images are absolutely necessary for the long term survival of this project. While we were able to work miracles with the single image given, data enrichment only buys so much before it encounters steeply diminishing returns and jeopardizes the validity of the results. In particular, further image samples gathered at lower resolutions in order to test the emerging hypothesis that we can deal sufficiently well with a compression factor of 1000 would be invaluable. The veterinary college seems happy enough to provide this information, the only issue is scheduling.

Normalization is also an important factor that has been handwaved during the processing step. The current algorithm is a form of 'processing by eye' that is rooted more in intuition than

proven fact. It would be simple enough to write an algorithm that normalizes the outgoing data vectors from the image processors to unit length, or mean zero and variance one. Different methods will be explored, and the most successful selected.

The topology of the network is also a part of the code that was put in more by instinct and intuition. Different topologies need to be tested in order to determine the minimum successful configuration. It's entirely possible that the search for a smaller configuration will lead us in the process to a more optimal configuration, as the reduction from 3 to 2 hidden layers proved highly beneficial to our results.

Finally, but importantly, different methods of interpolation need to be explored. Interpolation is critical during the compression step, and a higher quality algorithm could make all the difference. Stepwise bilinear was used, but a cubic function could be attempted in order to increase fidelity.

References

- Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research* 15.1 (2014): 1929-1958.
- Roeber, Florian, Aaron R. Jex, and Robin B. Gasser. "Impact of gastrointestinal parasitic nematodes of sheep, and the role of advanced molecular tools for exploring epidemiology and drug resistance-an Australian perspective." *Parasites & vectors* 6.1 (2013): 1.
- McLeod, R. S. "Costs of major parasites to the Australian livestock industries." *International Journal for Parasitology* 25.11 (1995): 1363-1367
- Ghasemikhah, R., Mirhendi, H., Kia, E., Mowlavi, G., Sarmadian, H., Meshgi, B., ... Mobedi, I. (2011). Morphological and Morphometrical Description of *Trichostrongylus* Species Isolated from Domestic Ruminants in Khuzestan Province, Southwest Iran. *Iranian Journal of Parasitology*, 6(3), 82–88.
- Gadberry, S., Pennington, J., & Powell, J. (2011, January 27). Internal Parasites in Beef and Dairy Cattle. In eXtension. Retrieved from <http://articles.extension.org/pages/11022/internal-parasites-in-beef-and-dairy-cattle>