Updating a CasPer structure using evolutionary algorithms to classify a person's subjective belief based on physiological information.

Logan Corry – u6422481@anu.edu.au

Research School of Computer Science, Australian National University

Abstract. This paper aims to determine whether CasPer trained with evolutionary algorithms can be used in an application which required many hidden neurons for classification in a fully connected neural network. A secondary aim is to use all the data collected in a separate study on physiological indicators (rather than a subset of the data which they use) to determine whether this allows for improved classification of a presenter's subjective belief. The method to complete this was to start by making a PyTorch neural network to validate the claims made in the original paper. The next step was to build a CasPer model using NumPy and give it the same data so that the accuracies could be compared. Finally, the CasPer model was extended to implement recombination and random mutations. All aims were inconclusive since the correctness of the model was not confirmed. In the case that the model is correct, multiple explanations are provided which may account for the poor classification compared with the original paper.

1 Introduction

1.1 CasPer

CasPer is an improvement on Cascor [1]. The basic CasPer model has only one neuron in each of the hidden layers which connect back to all previous layers. Each of the hidden layers is added sequentially with training in between to set the weights to suitable values before the next one is added. The older weights are still updated after more neurons are added, but at a slower rate determined by RPROP. This is an important difference from Cascor which only changes newly added weights.

To start with, each of the inputs are connected directly to the outputs with no hidden neurons and the weights for the connections are trained. A hidden neuron is added which takes input from the inputs and gives its output to the output. Different learning rates are used to train each of the weights, the new inputs to the neuron are trained at LR1, the outputs from the new neuron are LR2 and the other weights are trained at LR3 with LR1 >> LR2 > LR3. After some training, another neuron is added which takes input from the inputs and all previously added neurons and gives output to the outputs. The tests conducted in this paper added a new neuron after a fixed number of epochs, which is set as a hyperparameter. The same initial learning rates are used where any weights that were present before the neuron was added being LR3 with inputs and output of the new neuron being LR1 and LR2 respectively. This structure is shown in Figure 1.

1.2 RPROP

RPROP increases the learning rate when the weights are repeatedly being updated in the same direction and reduces the learning rate if they are not. To determine this, the gradient of the error with respect to each weight is calculated and compared with the value from the previous trial. Since successive tests are compared, this method is more effective in batch gradient descent rather than stochastic gradient descent. When the two gradients are compared, if they have the same sign, the learning rate for that weight is multiplied by a number greater than 1, if they have opposite signs, it is multiplied by a number between 0 and 1, and otherwise it remains the same. These learning rates are reset to the default LR1, LR2, LR3 when each new neuron is added. This project uses a simplified version of this which compares only the gradient of the error with respect to the input of the output neuron. This method was not intended to be used instead of the normal method but is how the code ends up running.

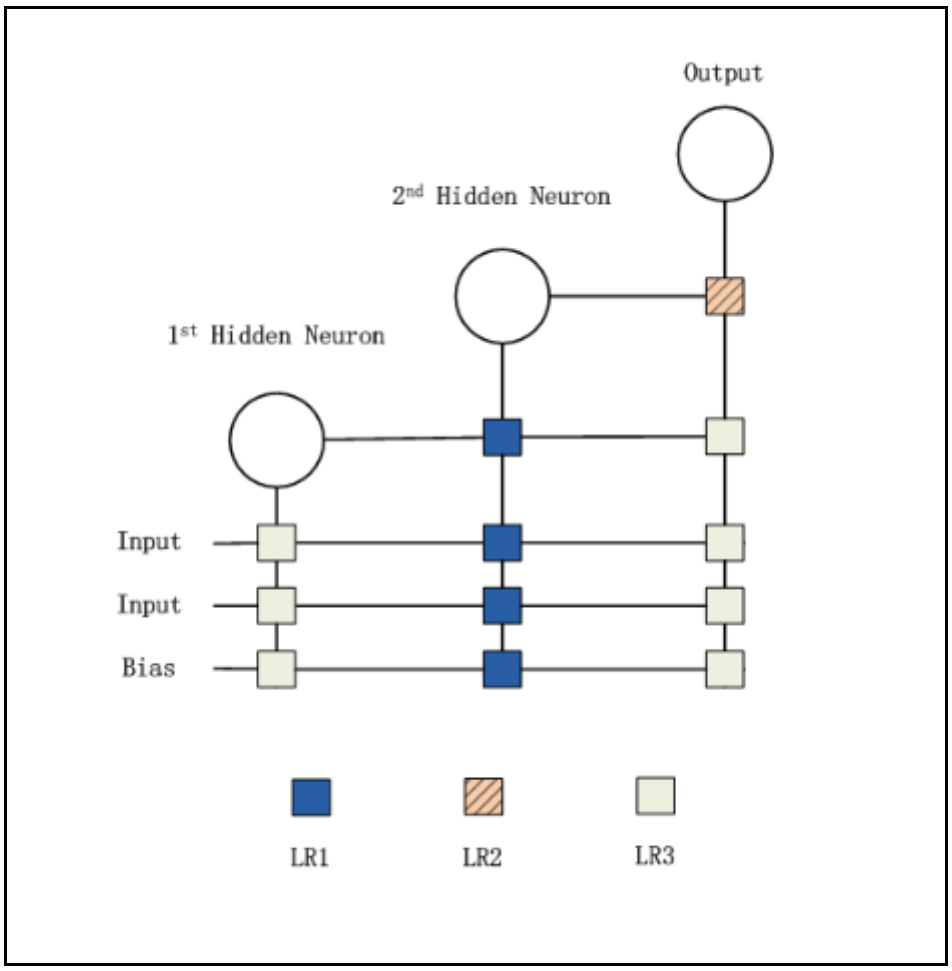

Figure 1. The structure of a CasPer model after the second neuron has been added. This diagram shows which initial learning rates are applied to which weights of the model. These learning rates are reset each time a new neuron is added. [3]

1.3 Subjective Belief – data and previous experiment

Previously, a study was conducted in an effort to determine whether physiological indicators could be used to detect doubt in a presenter. Predictions were made by both test subjects and a neural network to provide more information on the difficulty of the task.

To generate the dataset used in the subjective belief paper [3], the researchers asked two volunteers to read some text after either being told nothing or being made to doubt the content. The doubt vs belief conditions are used as the classification targets in the dataset. The presentations were recorded, and a number of other volunteers were asked to watch the videos and rate whether they thought the presenter believed or doubted the content. This rating was not used for this project. While watching the videos, a number of their physiological responses were also recorded, including data for pupillary dilation. Only the pupil diameter over time was used but was first transformed into 39 different decimal numbers which were given as features to the neural network in that paper. It is unclear exactly what these features represent or how they were created, but these 39 features form the restricted dataset. The researchers also measured blood volume pulse, galvanic skin response and skin temperature, which were similarly transformed into 34, 23 and 23 decimal number features respectively. Together, these 4 indicators of 119 features along with an ID tag and the targets form the full dataset.

To determine whether the restricted dataset could be used for classification onto the targets, the researchers constructed a fully connected neural network which contained one hidden layer with 100 neurons. They found that the model performed better than random choice while the conscious vote from the participants was equivalent to random choice. While this finding was significant, the actual accuracy of the model still left a lot of inputs incorrectly classified, indicating that this is a difficult dataset to classify.

1.4 Aims

This paper aims to determine firstly, whether a CasPer structure trained with evolutionary algorithms can be used to perform the classification and secondly, whether the rest of the physiological information collected for the subjective belief experiment can be used to assist the classification.

The first aim will help determine the applicability of an evolutionary algorithm version of CasPer to a wider range of tasks than was tested in the original CasPer paper, in this case one which takes many hidden neurons to classify. It is considered successful if the CasPer model can perform the classification with similar accuracy or if it is demonstrated that attaining such accuracy would require too much computation.

The second aim is used to show whether there is an even better model for predicting presenter subjective belief than using just pupillary dilation as in the original subjective belief paper. It is likely that the other data collected could provide further insight which the model could use in its classification. This aim is tested by comparing the accuracy using only the data used in the paper with using other combinations of data recorded for the paper (since the paper did not use the full dataset that was recorded).

2 Method

2.1 Pre-processing of the data

Since it is unclear exactly what the features of the data represent, no specifically tailored processing could be conducted. To begin pre-processing, the ID tag was removed since this is not part of the intended data to be used. The data is then normalised so that the neural network can be trained quickly, since otherwise it may appear that some features are more important than others without that being the case. The normalisation was to centre the data by subtracting the mean of each feature and then standardise the features by dividing by the standard deviations. The final step is to add the same constant value as an extra feature to each observation which the model can use as a bias term multiplier.

After the data is processed, it is split into train and test sets to ensure that the model was not trained to overfit to the dataset provided.

2.2 Verification of dataset claims

To begin, a quick test in PyTorch was conducted to verify that the subjective belief classification was better than random choice to a statistically significant degree. The fully connected neural network was set up as described in the paper on subjective belief; using only one hidden layer which contains 100 neurons. It was assumed that the network was fully connected since the paper did not say otherwise.

The network was run for 3000 epochs with a learning rate of 0.01. Stochastic gradient descent was used as the optimiser and the activation of the hidden layer was a sigmoid function. These are common settings which are just intended to be used to demonstrate whether classification is possible on this dataset.

The 39 data columns relating to pupillary dilation as used in the subjective belief paper were passed into the model and resulted in a classification prediction of either doubt or belief by the presenter.

2.3 Using CasPer to produce the same results

The CasPer method for generating a neural network was then implemented. The structure of a CasPer network is different from a normal neural network since later layers are essentially connected to all previous layers rather than only the previous one as in a normal neural network. Since PyTorch is designed to implement the normal fully connected networks, it was not used and instead the network was developed using NumPy.

The initial learning rates before RPROP used to update the weights are LR1=0.2 for the input connections to the added neuron, LR2=0.005 for the outputs of the new neuron (which connect to the output neurons) and LR3=0.001 for all other connections. All new weights are initialised on a uniform distribution from -0.7 to 0.7. The sigmoid function is used as the activation function for all neurons. Each of these points match what was used in the original CasPer paper [1].

When constructing the model, there were some important features to implement beyond the unusual structure. When using the sigmoid activation function, the value 0.0001 is added to the derivative during backpropagation to avoid the flat spot problem [4]. Another main feature of CasPer is its use of RPROP [5] to increase the robustness of the model during training. This is implemented in a way that tries to match the version used in the original CasPer paper [1] using the values 1E-6 – 50 for the min and max values with 0.5 for the change of gradient direction multiplier and 1.2 for the

same gradient direction multiplier [5]. The learning rates are then reset to their initial values before RPROP with each added neuron.

The training is conducted on the epoch level since NumPy can implement this much faster than using a 'for loop' across multiple batches, meaning that each batch is the complete dataset. Because of this, normal gradient descent was used instead of stochastic gradient descent. Initially, the hyperbolic tangent function was to be used as the loss function for the output to match the CasPer paper. However, this may not have been implemented correctly since it did not produce reasonable results, with all outputs being pushed to 0 regardless of what the targets were. Because of this, mean squared error was used instead as the loss function.

Four tests were set up on the CasPer framework for the combinations of the two different input datasets and two different output types. The outputs are the predicted class (belief or doubt) of the presenter represented either as a value on the output neuron or by one-hot encoding across two output neurons. The two different inputs are the restricted and full datasets.

The first test in CasPer was to provide only the data used in the subjective belief paper since that had been shown to result in suitable classification results. This data was the eye tracking data for Pupillary Dilation and consists of 39 individual values. In the second test, CasPer was provided with all the available data to determine whether that would affect the predictions. The full data consists of 119 different values relating to Blood Volume Pulse, Galvanic Skin Response, Skin Temperature as well as the Pupillary Dilation from the first test. A second version of CasPer was created with two output neurons to match the way that PyTorch normally implements a classification network. The two neurons are encoded to the one-hot version of classification rather than relying on a single neuron to differentiate between the two classes. This version was also trained on both the original restricted data and the full dataset.

2.4 Additional model validation test

The results for CasPer did not look promising, so to ensure that the model could produce a reasonable output, another test was conducted where all the target classes were set to be the same. When provided with the same input data as the previous tests, the model only had to choose between two classes where the same class was correct for all data.

2.5 Modifying CasPer to use evolutionary algorithms

The CasPer structure with the restricted dataset and one output neuron was then modified to update the weights based only on evolutionary algorithms. To begin, a population of models are randomly initialised in the same way as before. After performing the forward pass for the whole population, the top 20% of models with the lowest loss are selected. These models are carried forward into the next time step without change to ensure that the loss will not increase in the next time step. The rest of the population is made by randomly choosing two models from the top 20% and then cutting them at a random layer. The two parts are then combined and a small amount of uniform noise is added to the whole model, with each element ranging from -0.075 to 0.075. Multiple new models are created in this way until the new population size is the same as for the previous time step.

Since many models are being trained at once, it was found that the number of epochs to carry out for each added neuron could be significantly reduced, down from 2001 to 31 when using a population size of 70. The number of hidden neurons to add remained the same.

3 Results

3.1 Results from PyTorch verification test in 2.2

The results from the initial PyTorch test were consistent with what was reported in the subjective belief paper [3]. The network could perform better than random choice at predicting the subjective belief of the presenter based on the pupillary dilation information gathered from the viewers.

3.2 Results from CasPer tests in 2.3

3.2.1 Test with restricted dataset, one output neuron

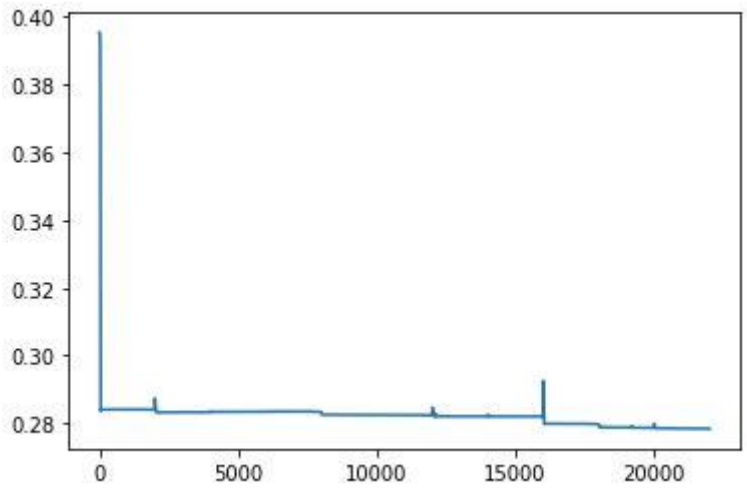

Figure 2. Mean squared error loss of the model over the number of epochs trained. Classifying all inputs as 0.5 will return 0.25 loss. The upward spikes in the loss are reflected in the confusion matrices of Figure 3 and analysed in its description.

The plot of Figure 2 shows that while the model is generally reducing its loss, there are occasions where it suddenly increases. The obvious explanation for this is that this is when each new neuron is added. Following this, the loss often decreases lower than it was before the neuron was added. The plot also shows that not much training occurs between adding successive neurons, even though it was also observed that the learning rate for the weights was quite often at the maximum value permitted by RPROP. Having the learning rates at their maximum values shows that the weights of the model were repeatedly moving in the same direction rather than bouncing back and forth quickly.

Each trained CasPer model consistently classified all datapoints into the same class, although the models did not necessarily classify into the same class as other models. To prevent this, the learning rate was increased so that the early added neurons could change more easily. This still produced poor classification, where after adding 70 neurons during training with 2001 epochs for each one, the output for each datapoint was a fairly random distribution between 0 and 1, as partially demonstrated by the confusion matrices in Figure 3. Even then, the model tended to favour one of the classes over the other. The class which is overrepresented can also change after adding more neurons, indicating that the output is not just based on whichever class has the higher membership in the training data. There is usually some consistency in the training confusion matrix after training successive new neurons, but occasionally adding another neuron will result in a completely different confusion matrix, as seen in Figure 3. The testing confusion matrices were quite similar to the training matrices, with the same quadrants being over- and under- represented.

```
neurons added: 18
[[ 12.  15.]
 [ 19.  28.]]
 average loss 0.256292093468
 average difference 0.499671148559


neurons added: 19
[[ 12.  13.]
 [ 19.  30.]]
 average loss 0.256250346852
 average difference 0.500193119942


neurons added: 20
[[ 12.  13.]
 [ 19.  30.]]
 average loss 0.256217583633
 average difference 0.500068312153


neurons added: 21
[[ 31.  43.]
 [  0.   0.]]
 average loss 0.328011584973
 average difference 0.518839213559


neurons added: 22
[[ 16.  16.]
 [ 15.  27.]]
 average loss 0.25267067487
 average difference 0.492455202421
```

Figure 3. Example output during training of successive confusion matrices. Entries on the main diagonal are correctly classified while the others are incorrect. After enough neurons are added that the confusion matrix becomes stable, adding another neuron is unlikely to result in a large change. However, there is some chance that adding a new neuron will cause the matrix to change dramatically, possibly increasing the loss. If the loss is increased, the next neuron will attempt to dramatically change it back to reduce the loss again. Sometimes this dramatic change back will overshoot in the opposite direction.

3.2.2 Other three tests from 2.3

These tests are grouped together since they all caused the same change in results compared with the previously analysed test. All three models produced essentially the same results, with the only difference being slower convergence. Adding an additional output neuron or more input data both require the model to undergo additional training to reach the same accuracy.

Having a second output neuron offers no benefit but requires that the model learn that the classification should only result in one of the outputs being high.

Adding additional inputs may allow the model to find new correlations, but there are many more weights to train. Additionally, the correlations in the data are not strong to begin with, as note by the difficulty that the original neural network had in classifying the dataset.

3.3 Results from model validation test in 2.4

The one-output neuron model performed well in the test where all targets were in the same class, with the ability to quickly converge on either class 1 or 0 before any extra neurons were added. The two-output neuron model was able to bring one neuron to the target quickly and the other close to the target quickly but then struggled to make the second one match the target exactly. This suggests that the one-output model converges faster than the two-output model, matching what was found with the restricted dataset.

3.4 Results from evolutionary algorithms test in 2.5

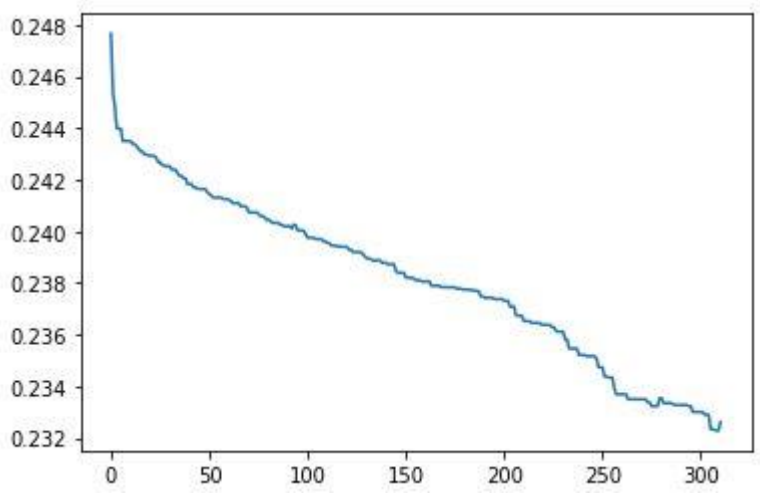

Figure 4. Loss of the CasPer model updated using evolutionary algorithms. The loss shown is the smallest loss found in the population.

Training using the evolutionary algorithms gives a smoother reduction in loss, shown in Figure 4, without the large spikes seen in Figure 2 when adding new neurons. However, the loss is still quite close to random classification. It appears that continuing training will reduce the training loss further, but either adding more hidden neurons or training for more epochs for each neuron both increase the testing loss, a sign of overfitting.

Using the model from Figure 4 as an example, the training loss was 0.2326 while the test loss was 0.2706. Increasing the number of epochs or hidden neurons resulted in slightly smaller training loss but test loss over 0.3. The confusion matrix on the test data for this model in Table 1 shows that it is still classifying one class more than the other.

Table 1. Test-data confusion matrix for the model which gave the loss shown in Figure 4.

	True 1	True 2
Predicted 1	3	1
Predicted 2	32	38

4 Discussion

During the simple one-class test, the two-output model performed worse than the one-output model, likely due to the larger number of neurons to train. Because of this, the one-output model was treated as the main experiment and all discussion will focus on it.

Judging off the results collected, it appears that the traditional CasPer model cannot be used to classify this data with either the restricted or full dataset, meaning that the method used in the original subjective belief paper is superior. Using evolutionary algorithms to update the weights did change how the training loss reduced over time but did not change the final testing accuracy, so it has no benefit over normal CasPer. However, there are still many explanations on why this model did not produce useful results which do not rely on CasPer with evolutionary algorithms (+ EA) being unsuitable for the task, meaning that it cannot be definitively stated either way whether CasPer + EA is suitable for classification of this data. Because the data for both the restricted and full datasets were similarly poor, it cannot be stated whether the full dataset can produce more useful results than just the restricted one.

The randomness observed in the output of the traditional method may be due to the relatively fewer number of neurons used in the network compared to the original 100 used in the paper. However, it would still be expected that as the number of neurons used in CasPer approached the maximum 70, the randomness would reduce slightly as it was able to discern some patterns, especially since the CasPer structure means that there are many more connections between neurons than for the fully connected network. Considering the restricted dataset, the original network used around 40 inputs which linked to 100 neurons and then to 1 or 2 outputs, resulting in around 4000 total connections. The CasPer model should reach this number by the time there are 60 added neurons, leading to the conclusion that the additional neurons may have provided some benefit that could not be matched by the additional connections. Unfortunately, adding each additional neuron takes more computation than the last, so going all the way to 100 added

neurons was not tested. This is because the number of weights grows exponentially with the number of hidden units [6]. Unlike the traditional CasPer model, the limit for number of hidden neurons was reached for CasPer + EA since more neurons or epochs resulted in overfitting.

Since the traditional model's RPROP learning rates were at their maximum so often, it would likely be more efficient to start with many hidden neurons already present since the weights learned for the first few are likely to be completely forgotten by the time the 50th neuron is added. This would also encourage the testing of larger numbers of added neurons since the time to reach 50 added neurons is removed. This point could also be applied to CasPer + EA, since the same justification likely holds

Another consideration is that the classification of the dataset in the original paper was not particularly accurate. While it was statistically significant and a better predictor than the viewers' conscious responses [3], many of the predictions are still incorrect. This indicates that the dataset is difficult to classify with so few neurons but adding more would likely lead to overfitting. Since this dataset is so difficult, it is not surprising that some models may have difficulty in performing the classification.

Alternatively, it is possible that there is some error in the model code that makes it converge much slower than would be expected. The model is clearly functional to some degree since it could classify in the one-class case. This slow convergence would lead to unusual random values since if not enough training is conducted before adding the next neuron, those weights will be harder to change, and more weights appear which can throw off the predictions. However, it is still unlikely that the randomness is caused by slow convergence since each neuron is trained for 2000 epochs and the test PyTorch fully connected network could perform decent classification with only 3000 epochs of training total. Having said that, the PyTorch model used stochastic gradient descent, meaning that each epoch contained up to 300 updates to the weights, resulting in 900 000 overall, which is many more than the 140 000 performed in the CasPer implementation. Updating the weights more often (but less accurately) may have resulted in the faster convergence.

5 Conclusion and Future Work

Overall, the project did not conclusively complete the aim to determine whether CasPer + EA is suitable for classification of the dataset used. The results indicate that it is not suitable and there are many explanations for why this may be the case. Alternatively, it is possible that the model was not implemented correctly. Due to this trouble with determining the first aim, the second aim of increasing the input data and observing its effect on classification accuracy was not studied.

Future work should be conducted to verify whether the CasPer + EA model has been constructed correctly and whether the results produced are valid. Since CasPer + EA is already at risk of overfitting, the next step is to revert to the traditional model and increase the number of added neurons in the test to over 100 so that the results can be more confidently compared with the results from the original subjective belief paper. It is unclear how much of the difference in results is caused just by having fewer neurons.

Beyond this, the other initial aim for this project should be examined which is to determine whether the additional data collected has any impact on the classification. This may include providing all the data as input to the model or providing only one of the other three categories collected. The original paper subjective belief also recommends this, although it is unclear why they collected the data but then did not use it. Using this data could lead to new links being found between the physiological responses and presenter beliefs.

As noted by the subjective belief paper, another point that may have caused issues with the results is that the dataset used is quite small in terms of the number of observations. Increasing the number of viewers and presenters studied could reduce some of the randomness in the output from the traditional CasPer model and could help reduce overfitting in CasPer + EA. This would also remove some of the uncertainty on whether the traditional method's results are over fitted since it is not clear whether adding more neurons would result in an over fitted model.

Another alternative direction is to use a slightly altered model of CasPer such as Layered CasPer which can have multiple neurons in each layer [2]. This form matches more closely with the fully connected neural network used in the original paper and the PyTorch test used to verify their results. Using a model which takes features from each of the two already used could result in better classification.

References

1. Treadgold N.K. and Gedeon T.D.: A Cascade Network Employing Progressive RPROP, Int. Work Conf. on Artificial and Natural Neural Networks, (pp. 733-742). (1997)
2. Shen T.: Layered Cascade Artificial Neural Network, A thesis submitted for the degree of Master of Computing in Computer Science of The Australian National University, (2011)
3. Zhu, X., Qin, Z., Gedeon, T., Jones, R., Hossain, M. Z., & Caldwell, S.: Detecting the Doubt Effect and Subjective Beliefs Using Neural Networks and Observers' Pupillary Responses. In International Conference on Neural Information Processing (pp. 610-621). Springer, Cham. (2018)

4. Fahlman, S.E.: Faster learning variations on backpropagation: An empirical study. In Proc. 1988 Connectionist Models Summer School. San Mateo, CA: Morgan Kauffman (1988)
5. Riedmiller, M. and Braun, H.: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Ruspini, H., (Ed.) Proc. of the ICNN 93, San Francisco, (pp. 586-591) (1993)
6. Treadgold N.K. and Gedeon T.D.: Exploring Architecture Variations in Constructive Cascade Networks, IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227) (Vol. 1, pp. 343-348) (1998)