Self-determining neural networks

Daniel Hao

Research School of Computer Science, Australia National University u6055952@anu.edu.au

Abstract. Artificial networks have been widely deployed in the application of image classification and have achieved impressive accuracies. However, there is no single standard network structure that can be used to serve every image domain, as a result, a huge overhead lies in the process of determining the most suitable neural network structure rather than training the network itself. As a solution to this problem, constructive learning algorithms have been proposed. They offer an incremental approach to constructing a near-minimal neural network, thus removing the need for manually determining the architecture of the network. This paper will compare the size and performance between a manually designed neural network and a self-determined neural network designed by a constructive learning algorithm for a multi-class classification problem. Experiment demonstrate that such a constructive learning algorithm can produce a suitable structure for image classification problems as well as performing similarly, and in certain cases better than a manually designed network.

Keywords: Artificial neural networks · Constructive algorithm · Image clasffication · emotions.

1 Introduction

1.1 Background

Common feedforward neural networks trained using backpropagation learning algorithms are limited to learning a size of weights that's determined a before the commencement of training, thus only able to learn a number internal representations limited by the structure of the network [7]. This induces the need of selecting the a suitable network structure for every different problem domain, however there is no known efficient methods for such a task. Often the process of designing a suitable network resorts to a *trial-and-error* manner, and has no guaranteed in finding the most optimal structure. Networks too small are incapable of learning the problem well while too large often causes the network to simply *memorize* the training data and thus overfit, generalizing poorly. A proposed solution to this problem are constructive learning algorithms, which are capable of determining the most suitable structure for itself and discovering near-minimal networks.

1.2 Artificial Neural Networks

Artificial Neural Networks (ANN) are inspired by the human brain structure [5],it mimics the functions of a neuron in a human brain. The human brain consists of billions of neurons with various types and lengths, and each neuron have three major functional units; dendrites, cell body and axon. As a simplified overview, the cell body holds information it receives from the dendrites and sends out signals of different strengths to other parts of the body through the axon. Artificial neural network builds upon this structure, with connections between nodes resembling the dendrites and axon, and the node itself being the cell body.

Information passing through artificial network with varying weights resembles the firing of neurons inside the human brain, with multiple layers of neurons the network is capable of representing a complex non-linear equation[1]. By adjusting the strength of the signals through the network, i.e. optimizing the weights and biases within the network, ANN are capable of learning complex tasks, similarly to the human brain.

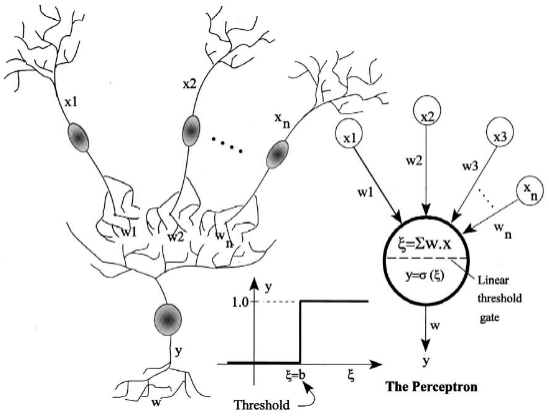

Fig. 1. Side by side comparison of a biological neuron and a artificial neuron[5]

1.3 Constructive learning algorithms

Unlike standard feedforward neural networks with predetermined structures that do not vary during training, neural networks with constructive learning algorithms start with a minimal structure, either a single hidden layer with one neuron or no hidden layer, and dynamically grow the network by adding and training neurons as needed. Such process is repeated until a certain threshold or until a satisfactory solution has been met [9]. Constructive learning algorithms levitates the need of searching for a ideal network structure and are capable of learning high level feature affectively in tasks such as pattern classification[6] or facial recognition[3]

1.4 Data set

The data set used for the image classification task is from the Static Facial Expressions in the Wild database[2]. The data set contains 675 images that have been labelled for six basic expressions: *angry, disgust, fear, happy, sad, surprise* and also including the *neural* expression[2]. The 675 images consists of a balanced mix of expressions of 100 each except for the disgust class, which only has 75 samples in contrast to 100. The task of the classifiers are to classify each image to one of the expression classes.

Rather than using the original images for the classification task, local phase quantization descriptors (LPQ) and the pyramid of histogram of oriented gradient (PHOG) descriptors were used instead, as this paper is not interested in creating state-of-the-art image classification networks with convolutional layers. By using LPQ and PHOG descriptors the usage of convolutional networks is avoided and thus focuses the experiment on comparing performances between different networks.

The SFEW database[2] was selected as it contained relatively realistic images of various expressions. Unlike the MMI database[4], which contained facial expression captured in lab-controlled environments, SFEW extracted its data from movies scenes, allowing it to have relatively more realistic expressions.

2 Method

2.1 Casper

The constructive learning algorithm selected for comparison is the Casper algorithm[8]. Casper constructs a cascade network starting with a single hidden neuron and dynamically adds new neurons successively. All new neurons receives the origin data and the outputs from previously added neurons as input and directly outputs to the output layer. A variation of RPROP is used as the gradient descent algorithm and instead of a constant learning rate

throughout the network, the network is divided into 3 regions each with its own respective learning rate[8]. Region 1 being the output from the most recent neuron added, region 2 are all connections going into the new neuron and region 3 being all remaining connections.

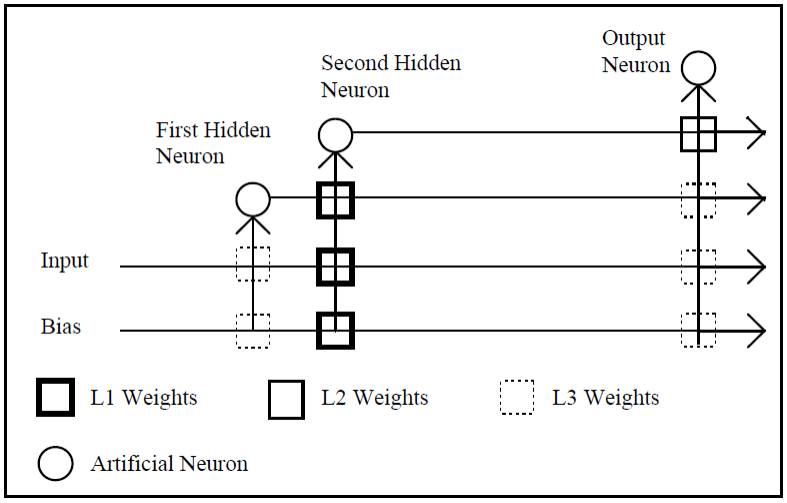

Fig. 2. Casper network's varying weights [8]

Slight modifications have been made to the Casper algorithm in contrast to the origin paper[8] in attempts to improve the networks performances. Firstly, the network begins with no hidden neurons instead of one hidden layer with one neuron. It is discovered through empirical studies that starting with no hidden layers and then dynamically adding them as needed improved the stability of the network's final accuracy and effectively lowered the variance of the network. Weights of new neurons was initialized with the range of (-0.1, 0.1), as new neurons with lower weight values allowed a more stable network when new neurons are inserted, while new neurons are still able to learn effectively with its comparably high learning rate. Learning rate for each region were set to 0.25, 0.01 and 0.001 respectively, allowing new neurons to discover higher level feature effectively while not completely freezing old neurons. The hyperbolic tangent function was used in place of the sigmoid function to help optimization as it is not 0 centered.

The backpropagation algorithm remains the same[8], as RPROP provided the fastest recovery whenever new neurons are inserted. Other backpropagation algorithms such as Adam and SGD were tested but performed poorly in contrast to RPROP. The network is trained over 5000 epochs, the rate to which new neurons are inserted is dependent on the convergent rate of the network, whenever the cross-entropy loss over 40 epochs decreased less than 0.001 a new neuron is inserted, and the learning rate of the network is changed accordingly. Weight decay has also been implemented as a partial replication of the SARPROP algorithm [10], the value of weight decay was set to 0.998 with empirical evidences.

2.2 preprocessing

The origin data contained 12 columns of data, including the emotion labels, the file name of the original image file and 5 features each for the LPQ descriptor and PHOG descriptor. For the purpose the experiment the file names were removed as it has no correlation with the emotions the images contain. The labels were also relabeled from 1 - 7 to 0 - 6 respectively for serving as the target of classification. All 10 features from the LPQ and PHOG descriptors were used as inputs to the networks. 25 extra *disgust* data samples were artificially added to the data set to create a more balanced data set, resulting in a data set of size 700.

The data is then shuffled randomly, independent of its source and the subject of the expression, such that we have a Partially Person Independent data set (PPI)[2]. The shuffled PPI data is then split with a 90/10 ratio respectively for training and testing. Strictly Person Independent (SPI) data sets were not constructed as we do not possess information about the individual in the image, but only the movie source of the image. It is thus impossible to split the data set into training and testing sets in a way such that no subjects in the training set are present in the testing set. Splitting the data set according to the movie source of the image was considered, however it was not proceeded with as there is not guarantee that different movies contained completely different actors, especially since it contained movies with sequels, such as Harry Potter.

2.3 predetermined feed forward network

The manually constructed network that will serve as the baseline for the experiment is a fully connected feed forward network with 3 hidden layers and with 20, 15 and 10 neurons in each layer respectively. The predetermined network also uses RPROP as the optimiser with a learning rate of 0.2 as it yielded the most consistent results. The weights were however initialized with pytorch default weight initializations in contrasts to casper's self defined range. Both casper and the predetermined network uses cross entropy as the loss function as the data set chosen for this experiment is a multi-class classification problem. The resulting network was was able to generalize reasonably and did not overfit on the training data.

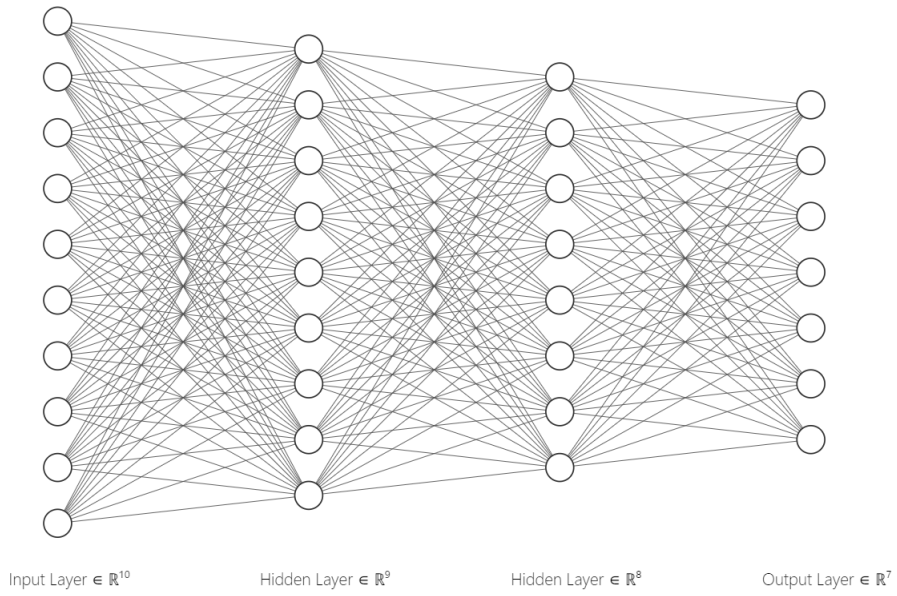

Fig. 3. Fully connected neural network

3 Results and Discussion

The value used for comparison is the balanced accuracy of the trained network. To compute balanced accuracy we take the average recall value of the network for each emotion class, in doing so we are able to eliminate cases where the network skewed towards predicting only few specific classes rather than making actual predictions for each data entry. The Casper network was able to construct a suitable structure during run time and achieved a best result of

30% accuracy, with an average 26%. Where else the standard feed forward network achieved an average of 25% with a best result of 31%. As demonstrated in Table 1, the final network from Casper algorithm yielded more consistent accuracies in comparison to the predetermined network. While there is no guarantee that the predetermined network was perfect, Table 1 shows that Casper network successfully produced a network structure that is at least as good as a reasonably designed predetermined network.

	Casper	Standard
01	25.29%	25.23%
02	23.81%	30.64%
03	29.66%	29.99%
04	26.39%	24.89%
05	27.78%	25.72%
06	27.57%	20.74%
07	28.62%	15.76%
08	27.77%	28.12%
09	25.14%	23.79%
10	20.36%	22.81%
Average	26.24	24.77

Table 1. Casper network and predetermined network testing accuracies over 10 runs

Figure 4 demonstrates the convergence nature of Casper networks, peaks occur whenever a new neuron is inserted since a new neuron does not contribute anything at the time of insertion. After the network adapts with the new neuron, it was able to continue to minimize its loss function with the aid of the new neuron. If no neurons were inserted, the network would have resulted in a flat line after it exhausted the learning capabilities of its current neurons.

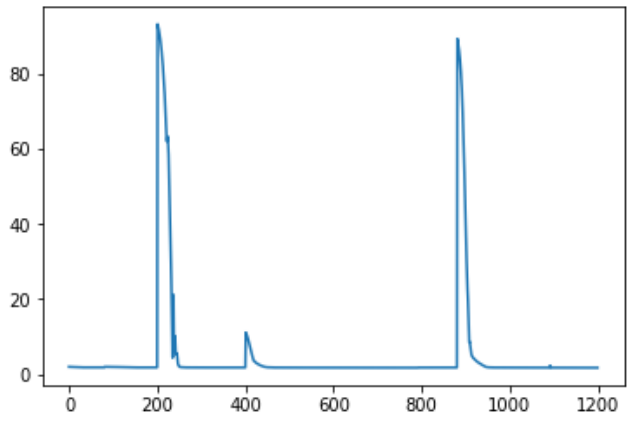

Fig. 4. Loss value of casper network

The number of neurons inserted into the network varies greatly between each run, depending on the random initialization of weights. In general, best results were achieved when there are around 5 neurons added, inserting more neurons often either had no impact to minizine the loss function or simply lead the network to overfit on the

training data. In contrast, the convergence graph of a predetermined feed forward network is much smoother as shown in Figure 5.

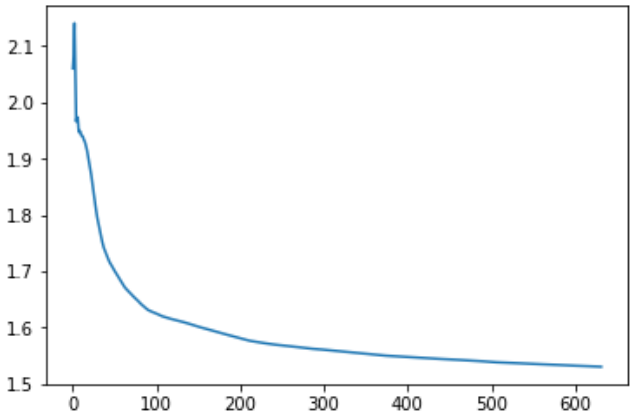

Fig. 5. Loss value of standard feed forward network

While the Casper network did not perform as well during training, it was able to learn to generalize quite well and had comparable accuracies with the predetermined network on the final testing data. The size of the Casper network is also significantly smaller than the predetermined neural network, while the predetermined network had up to 17 hidden neurons with 218 connections, the Casper network had around 5 hidden neurons on average with only 165 connections.

In comparison to the support vector machine[2], the accuracy of both the manual feed forward network and Casper network were relatively lower. This is somewhat expected as the networks did not receive the same input as what the support vector machine received. The close to real world nature of the SFEW database also contributes greatly to the low accuracy, as contrast between different expressions may not be as significant as expressions collected through lab-controlled environments. In addition, SFEW contains images with inconsistent qualities, including both high and very low quality faces [2], which further increases the difficulty of the problem. However, the final accuracy of the Casper classifier is not a major concern as the focus of this paper lies on the comparison between neural networks rather than improving their final performances.

4 Conclusion and Future Work

Constructive learning algorithms offer a valid solution to the problem of automating the design of neural networks. They can achieve performances on par to predefined networks and able to construct near-minimal structures. They eliminate the need for predetermining the architecture of neural network, effectively lowering the risk of overfitting as well as avoiding the need of domain specific knowledge and saves precious time from repetitive *trial-and-error* work. However, there are still down sides of constructive neural networks, one being that they may suffer more from vanishing gradients compare to standard networks as they have more layers. Solution such as adding new neurons to the same layer have been proposed, however determining the suitable number of neurons per layer is not superficial and will be the next step onward from this paper.

References

1. Benbassat, A., Sipper, M.: Evolving artificial neural networks with FINCH **87**(9), 1719 (2013). https://doi.org/10.1145/2464576.2480780
2. Dhall, A., Goecke, R., Lucey, S., Gedeon, T.: Static Facial Expression Analysis in Tough Conditions : Data , Evaluation Protocol and Benchmark Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia. Database pp. 2106–2112 (2011). https://doi.org/10.1109/ICCVW.2011.6130508, http://staff.estem-uc.edu.au/roland/wp-content/uploads/file/roland/publications/Conference/ICCV/BeFIT2011/dhall_goecke_lucey_gedeon_BeFIT2011_StaticFacialExpressio
3. Fernandes, B.J., Cavalcanti, G.D., Ren, T.I.: Constructive autoassociative neural network for facial recognition. PLoS ONE **9**(12), 1–23 (2014). https://doi.org/10.1371/journal.pone.0115967
4. Krumhuber, E.G., Skora, L., Küster, D., Fou, L.: A Review of Dynamic Datasets for Facial Expression Research. Emotion Review **9**(3), 280–292 (2017). https://doi.org/10.1177/1754073916670022
5. Mohammadhassani, M., Nezamabadi-Pour, H., Jumaat, M.Z., Jameel, M., Arumugam, A.M.: Application of artificial neural networks (ANNs) and linear regressions (LR) to predict the deflection of concrete deep beams. Computers and Concrete **11**(3), 237–252 (2013). https://doi.org/10.12989/cac.2013.11.3.237
6. Parekh, R., Yang, J., Honavar, V.: Constructive neural-network learning algorithms for pattern classification. IEEE Transactions on Neural Networks **11**(2), 436–451 (2000). https://doi.org/10.1109/72.839013
7. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation (No. ICS-8506). California Univ San Diego La Jolla Inst For Cognitive Science **1**, 318–362 (1986). https://doi.org/10.1016/B978-1-4832-1446-7.50035-2, https://web.stanford.edu/class/psych209a/ReadingsByDate/02_06/PDPVolIChapter8.pdf
8. Treadgold, N.K., Gedeon, T.D.: A Cascade Network Algorithm Employing Progressive RPROP Second Hidden Unit First Hidden Unit Output Unit Input Bias Artificial Neuron L1 Weights L2 Weights L3 Weights (1993) (1996)
9. Treadgold, N.K., Gedeon, T.D.: Exploring constructive cascade networks. IEEE Transactions on Neural Networks **10**(6), 1335–1350 (1999). https://doi.org/10.1109/72.809079
10. Treadgold, N., Gedeon, T.: The SARPROP algorithm: a simulated annealing enhancement to resilient back propagation. ... International Panel Conference on Soft and ... (February 1997) (1996), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.8197&rep=rep1&type=pdf