Insights of the Hidden Neuron Functionality using Deep Learning Techniques for Facial Emotion Recognition

Evan Markou

Research School of Computer Science,
Australian National University,
Acton ACT 2601, Australia
u7086465@anu.edu.au

Abstract. Dividing deep inside the hidden neuron functionality can provide insightful results in many classification tasks. For extremely difficult tasks such as facial emotion recognition, distinctiveness is an important attribute to be analysed. In this work, using the SFEW dataset, we provide a custom convolutional neural network architecture and surpass the class-wise classification results of prior works and is on par with regarding the overall accuracy. We also incorporate transfer learning concepts to our training to increase our results even further and achieve a new benchmark surpassing all prior works on every metric. Then, we investigate the use of distinctiveness in the hidden neurons to gain insights into ways to avoid being trapped in local minima, which is extremely common in emotion recognition tasks. Finally, key issues of the network structure and the dataset features are discussed and future work is provided alongside conclusion remarks.

Keywords: Neural Networks · Convolutional Neural Networks · Transfer Learning · Facial Emotion Recognition · Distinctiveness · Hidden Neuron Functionality

1 Introduction

Emotions are the cornerstones of human behaviour and the basis of regular communication. For understanding emotions and associating them with human behaviour and beliefs, it is important to detect them first. Recent advancements of artificial intelligence and deep learning have showcased that we can achieve satisfactory detection performance using both supervised and unsupervised techniques [11, 8] on a variety of sources (e.g. images, video, speech, and text). The most popular emotion models either assume discrete emotion categories (categorical model [3]) or a continuous emotional scale that combines valence and arousal intensities (dimensional model [7]). The projects that have attributed to the problem of automatic emotion detection from image and video, either provide data from experiments conducted in a controlled lab environment or aggregate samples *in-the-wild* from the web using emotion-related queries. They mostly follow the categorical model and only a few works with limited training samples support the dimensional model.

The main focus of this paper is to investigate the hidden neuron functionality of an artificial neural network for the facial emotion recognition task, and more specifically, how the weight matrices compare against the overall network performance. The idea behind this concept, firstly introduced in [4] and later in [5], is that neuron behaviour classifies neural networks into three different categories. These categories, as directly defined in [5], are as follows; i) networks that get stuck in local minimal; ii) networks which are stuck in a shallow valley (saddle points), but eventually will converge; and iii) networks which arrive in the solution space rather quickly.

The rest of the paper is organised as follows: Section 2 that follows dives deep into the methodology, practices, and procedures that are used in this work. Section 3 provides empirical results and ablation studies. It also contains analytical discussions on the topic's findings. Finally, Section 4 concludes the paper and suggests potential future work.

2 Method

2.1 Dataset

The dataset that is used in this work for the facial emotion recognition task, is called *Static Facial Expressions in the Wild* (SFEW) [2]. This is a derived static database from the *Acted Facial Expressions in the Wild* (AFEW) dataset, which was originally introduced in [1]. SFEW is comprised of categorical emotion labels, with the following encoding. The SFEW dataset is comprised of both the raw images and the derived eigenvalues from the local phase quantisation (LPQ) descriptors, and the pyramid histogram of oriented gradients (PHOG) descriptors, respectively. In this work, we are utilising both forms of the dataset.

Table 1. Label encoding of SFEW dataset.

1	2	3	4	5	6	7
Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise

Pre-Processing Regarding the features, the dataset is given, based on the implementation choices of [2], in terms of the first five eigenvalues of the local phase quantisation (LPQ) descriptors and pyramid histogram of oriented gradients (PHOG) descriptors, respectively. It was decided, after analysing each descriptor separately, that it would be helpful to standardise the inputs with their z-score, making sure that they will have an equal contribution, as their range varied originally. Therefore, for each descriptor, we standardise the input using the Equation 1, where μ_X is the mean and σ_X is the standard deviation of each descriptor respectively.

$$\widetilde{X} = \frac{X - \mu_X}{\sigma_X} \tag{1}$$

Considering now the raw images, they are the core of the SFEW dataset. After inspection of the data, we realised that the faces in the dataset are not centred and images vary in dimensions, therefore we proceed to pre-process the images further. First, we used the Viola-Jones algorithm, introduced in [10], the perform face detection and create a bounding box around each face. Then, we cropped the image, keeping only the pixels inside the bounding box (i.e., the actual face), and we resized the cropped image to 224x224, which is a standard size to use for image classification tasks. This whole process, also described in Figure 1, was part of our offline data augmentation, meaning outside the actual training procedure.

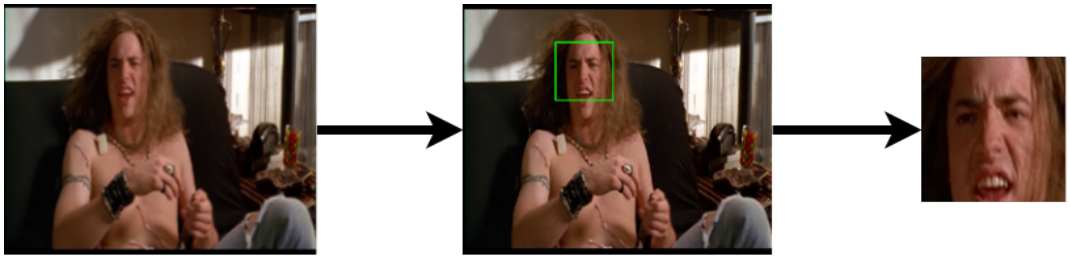

Fig. 1. Graphical illustration of the face detection process using Viola-Jones algorithm and pre-processing of a raw input image.

Since, the dataset is rather small, to battle possible overfitting, we performed further online data augmentation techniques. More specifically, each batch of images was horizontally flipped with a probability of 0.5, and a 25-degree random rotation was also performed.

2.2 Standard Neural Network

To perform the face emotion classification task, a simple/standard neural network with three layers was used; input, hidden, and output layer. For this network, we used the LPQ and PHOG features of the dataset. For the input layer, both LPQ and PHOG descriptors were merged to achieve higher performance, as was indicated in [2]. Thus, we now have ten input features.

The hidden layer has 50 hidden neurons/units, and its activation function is a Sigmoid layer($\sigma(\cdot)$), defined in Equation 2.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

The output layer has seven classifiers, each being one emotion label. The activation function is this layer cannot be a Sigmoid layer, since Sigmoid only works in binary classification and now we're performing multi-class classification. So, a LogSoftmax layer is used, defined in 3.

$$LogSoftmax(x_i) = \log\left(\frac{e^{x_i}}{\sum_i e^{x_i}}\right) \tag{3}$$

The goal is to minimise the error function which in this case is the cross-entropy loss since it's ideal for such classification tasks, and it's defined in 4, where t_i is the ground truth and p_i is the LogSoftmax probability for that i^{th} class. For this network, we used a MultiStep learning rate scheduler. Further details are given under Results section.

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i) \tag{4}$$

2.3 Custom Convolutional Neural Network

To utilise the raw images, we designed a custom CNN to perform facial emotion recognition, described in Table 2. Furthermore, for this network, we introduced a cyclic learning rate scheduler, defined in [9] (Check Results section for configuration details). Apart from the learning rate scheduler, the training process was the same as in the ANN case, using a LogSoftmax layer as in Equation 3 and the cross-entropy loss, defined in Equation 4.

Table 2. Our Convolutional neural network architecture specificitions.

CNN Architecture				
Operation	Kernel Size	Filters	Stride	Padding
Convolutional Layer	5x5	32	1	2
Batch Normalisation Layer	-	-	-	-
ReLU Activation Layer	-	-	-	-
Max Pooling Layer	2x2	-	2	0
Convolutional Layer	3x3	64	1	1
Batch Normalisation Layer	-	-	-	-
ReLU Activation Layer	-	-	-	-
Max Pooling Layer	2x2	-	2	0
Fully Connected Layer	-	128	-	-
ReLU Activation Layer	-	-	-	-
Fully Connected Layer	-	32	-	-
ReLU Activation Layer	-	-	-	-
Fully Connected Layer	-	7	-	-

2.4 Transfer Learning

Finally, since image classification is a very popular task, we decided to investigate the use of transfer learning to accomplish our tasks. More specifically, we used a ResNet model [6] pretrained on the well-known ImageNet dataset. We kept all the weights frozen except for the fully connected layer, which we altered to match our facial emotion recognition problem.

2.5 Hidden Neuron Functionality

To measure the effectiveness of the neural network for the face emotion recognition task on the SFEW dataset, the distinctiveness of the angles between activation vectors of the hidden layer is investigated. By using distinctiveness angular measures, we define distinctiveness as the arccosine of the cosine similarity between activation hidden neuron vector. Meaning the angles between those vectors. To calculate the cosine similarity we used the definition in Equation 5.

$$\cos(\theta) = \frac{A \cdot B}{||A|| \times ||B||} \tag{5}$$

Our goal is to differentiate sections of the angles graph, and distinguish different stages, such as Initial, Intermediate, Plateau, and Solution. We can define the Initial stage as the warm-up period, usually a few epochs since the network starts to behave as expected. The Intermediate stage is the majority of the training process, where the network tries to converge. The Solution stage is where the network is positive enough that it found a good local minimum for generalisation. Finally, the Plateau is a special stage between Intermediate and Solution stages, but

yet common. Here, the network is stuck either to a saddle point or a poor local minimum and is unable to continue training and converge. We note that the facial emotion recognition problem, especially in *wild* conditions, is a challenging task, and we expect the network at some point to reach the Plateau stage rather than the Solution one. Based on which stage the neural network lands across epoch training, we will hope to get a better understanding of the underlying neuron functionality, and accomplishing this task better and/or more efficient.

2.6 Evaluation

We are following the evaluation process described in [2], so we are evaluating every model against its overall accuracy, as well as its class-wise precision, recall, and specificity. The equations of each metric are defined below. Also, our training/validation process follows the K-Fold cross-validation approach, where in the standard artificial neural network model (ANN) case we used a 5-Fold, whereas in the cases of the custom convolutional neural network model (CNN) and transfer learning with ResNet (T-Res), we used a 4-Fold. We chose one fold less since training image data has higher time and computation complexity than simply training feature descriptors.

$$Overall\ Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{6}$$

$$Class\ Wise\ Precision = \frac{tp}{tp + fp} \tag{7}$$

$$Class\ Wise\ Recall = \frac{tp}{tp + fn} \tag{8}$$

$$Class\ Wise\ Specificity = \frac{tn}{tn + fp} \tag{9}$$

3 Results and Discussion

The approach that was chosen to evaluate the SFEW face emotion classification is to perform KFold cross-validation and then average the results to obtain the final metrics. Since we have only 675 data entries (images or features) at our disposal, a holdout-test or a train-validation-test dataset split did result in over-fitting problems. In every experiment, the optimiser that was used was SGD with momentum=0.9. In the case of ANN, the experiments were run for 2000 epochs, and the best results were found by using a multi-step learning rate scheduler for hyper-parameter search space. More specifically, we started with a learning rate of 0.001 and reduce it by a factor of 0.1 every 400 epochs. For the CNN and T-Res cases, we trained the models for 200 epochs, using a cyclic learning rate scheduler, for the hyper-parameter search space. In this scheduler, we started again with a learning rate of 0.001 and cycle from that to a maximum learning rate of 0.1 every 40 epochs.

In Tables 3, 4, it is presented the best validation results from averaging every test set evaluation from all the five and four folds respectively, considering each model's case. More specifically, Table 3 compares the overall classification accuracy, between our models and the work that was done by the authors in [2]. Table 4 compares the average per class precision, recall, and specificity against the same benchmark work in [2].

As we can see from the two Tables, our models from-scratch (ANN, CNN) despite having less overall accuracy in comparison with prior works, both outperform almost every emotion class. However, by using transfer learning, we observed a significant boost in performance, surpassing every metric by a large margin, in comparison with prior works and our from-scratch models. This gives us great insights into the advantages of transfer learning but also indicates that the optimal strategy for our custom models is yet to be found, and further hyper-parameter search is needed. Those metrics are defined as in [2].

Table 3. Total classification accuracy on the SFEW dataset. Note that bold result indicate the better classification.

Technique	LPQ [2]	PHOG [2]	Our Work [ANN]	Our Work [CNN]	Our Work [T-Res]
Accuracy	43.71%	46.28%	35.76%	37.72%	**51.81%**

During training, the cross-entropy loss easily plateaued, and hence inspired by the work in [5], the distinctiveness of the hidden neuron functionality was investigated, with the aim of understanding and analysing the network

Table 4. Average expression class-wise Precision, Recall, and Specificity on the SFEW dataset. Note that bold results indicate the better classification.

Emotion	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Baseline results from SFEW [2]							
Precision	0.17	0.15	0.20	0.28	0.22	0.16	0.15
Recall	0.21	0.13	0.18	0.29	0.21	0.16	0.12
Specificity	0.48	0.66	0.64	0.51	0.61	0.60	0.66
Our results [ANN]							
Precision	0.37	0.14	0.36	0.14	0.23	0.19	0.05
Recall	**0.43**	0.18	0.36	0.36	0.23	0.13	0.08
Specificity	0.82	**0.98**	0.93	0.69	0.72	0.55	0.57
Our results [CNN]							
Precision	0.43	0.26	0.53	0.27	0.24	0.29	0.34
Recall	0.40	0.13	0.47	0.21	0.28	0.50	0.25
Specificity	0.90	**0.98**	0.91	**0.94**	0.82	0.78	**0.93**
Our results [T-Res]							
Precision	**0.50**	**0.43**	**0.65**	**0.36**	**0.34**	**0.54**	**0.41**
Recall	0.39	**0.40**	**0.61**	**0.42**	**0.38**	**0.69**	**0.49**
Specificity	**0.92**	0.96	**0.95**	0.91	**0.87**	**0.89**	**0.93**

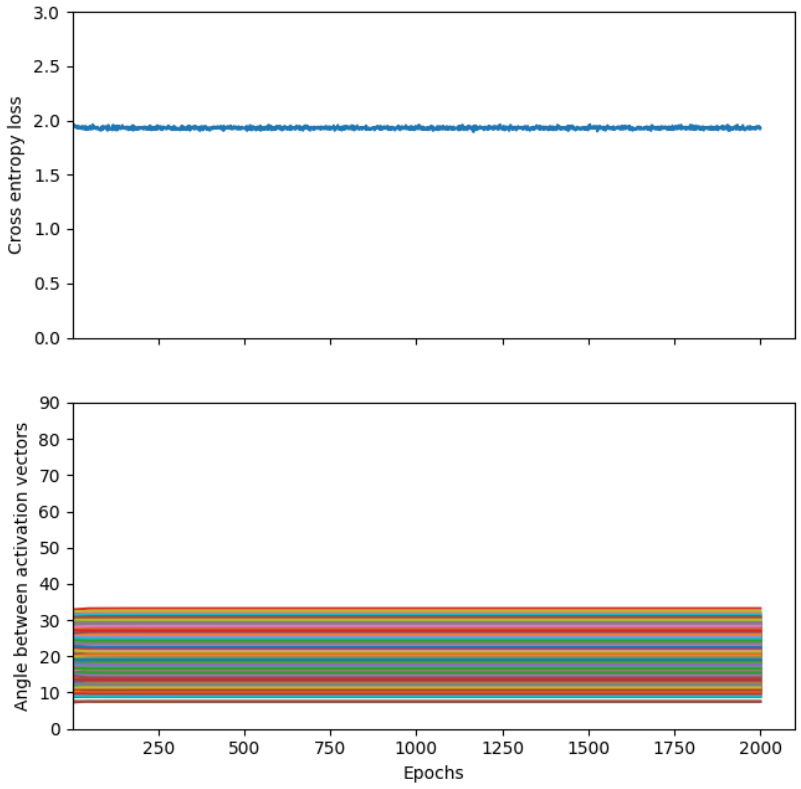

Fig. 2. ANN network is in a constant plateau stage from the third fold in our 5-Fold cross validation procedure.

performance even better. The results of this analysis are depicted in Figures 2, 3 for the ANN and CNN model respectively. In the case of ANN (Figure 2), as it's obvious, the network performance almost instantly reaches a

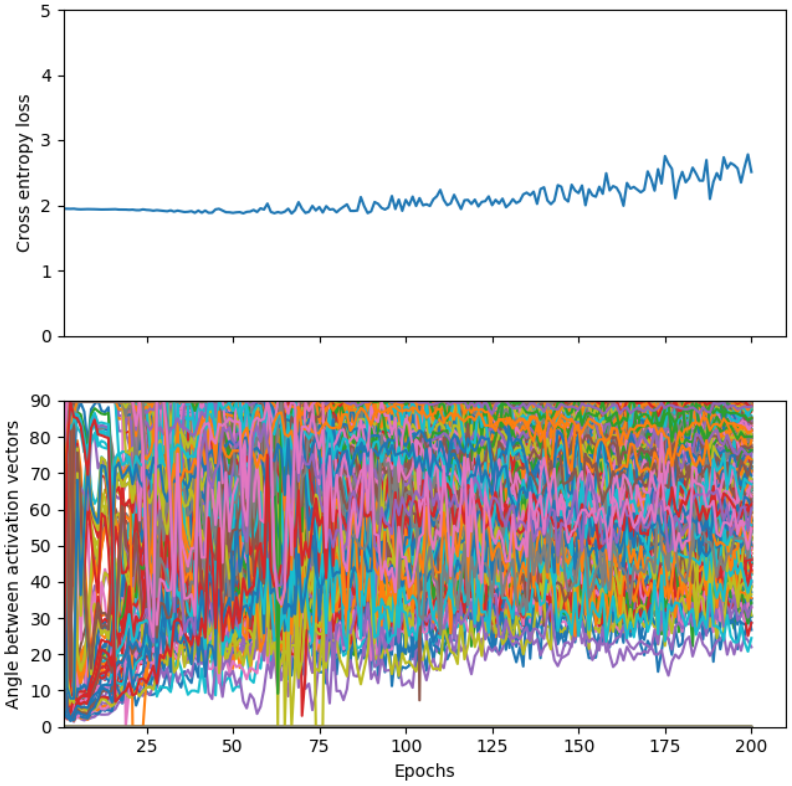

Fig. 3. Changes between stages of the CNN network from the third fold in our 4-Fold cross validation procedure.

good solution space, meaning easily and quickly stucks in a local minimum. Hence, as the epochs continue, the network performance constantly remains in the plateau stage. The angles between the activation vectors are also constant, almost immediately, which amplifies the original hypothesis that the neural network stagnates after only a few iterations. One possible and very good reason for the local minimum plateau is the nature of the emotion classification problem, especially when dealing with *in-the-wild* scenarios, as in here. This is a very challenging task, which even struggles with human annotators many times, hence accomplishing perfect scores is extremely difficult.

However, in the case of CNN (Figure 3), we can see that we are having a smoother update of stages. More specifically, the network is in the Initial stage for a few epochs, approximately 25, and then passes to the Intermediate stage where it tries to converge. This stage approximately holds for the next 75 epochs. From epoch 100, however, the network enters the Plateau stage where it becomes stagnate and as we can see from the cross-entropy loss, from this point onward we start seeing some overfitting patterns. This further indicates that the network doesn't generalise well, and begins to overfit to the given task. Hence, the vector angles don't need to change since the network has already found a specific solution for the problem. We identify the reason for this overfitting pattern on the small number of images that the SFEW dataset has, only 675. We argued in the beginning that using online data augmentation techniques helps with overfitting and we believe that it did, hence the new benchmark result. It is worth mentioning also that in the case of transfer learning (T-Res), the overfitting patterns were even more prevalent.

Since a KFold cross-validation procedure was performed and in Figures 2 and 3 indicatively only the data from the third fold were used, for completion we provide the rest of the $K-1$ folds hidden neuron functionality, in Appendix Section 5.

4 Conclusion and Future Work

Face emotion recognition is a difficult task, especially if the target dataset is extracted from *in-the-wild* scenarios. As the size of the datasets keep growing, neural networks have gained ground against more traditional classification methods, i.e., Support Vector Machines. However, since neural networks are essentially a black-box architecture, there is an increasing need for a deeper understanding of their hidden neurons functionality. One way to gain insights is with the use of distinctiveness of the neurons by observing the angles between their activation vectors. The more distinct they are, the more useful features they generate, and that ultimately leads to better performance.

In our work, we derived useful insights from the hidden network functionality. The major one was when the network overfits, the angle vectors tend to enter and stay in the Plateau stage indefinitely. Therefore, as future work, we will need to dive deeper into the root cause of this overfitting issue and try to incorporate techniques such as higher levels of regularisation or early stopping to battle it. However, despite our current and future efforts, the reason for this overfitting might lie in the nature of the dataset, by having only a few images. We suspect that this dataset might not be suitable for deep learning architectures, hence the reason for the SVM classifier in [2].

References

1. Dhall, A., Goecke, R., Lucey, S., Gedeon, T.: Acted facial expressions in the wild database. In: Technical Report (2011)
2. Dhall, A., Goecke, R., Lucey, S., Gedeon, T.: Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). pp. 2106–2112 (2011). https://doi.org/10.1109/ICCVW.2011.6130508
3. Ekman, P.: An argument for basic emotions. Cognition & emotion **6**(3-4), 169–200 (1992)
4. Gedeon, T.D.: Hidden units in a plateau. In: Proceedings 1st International Conference on Intelligent Systems. pp. 391–395. World Scientific, Singapore, USA (1992)
5. Gedeon, T.D.: Indicators of hidden neuron functionality: The weight matrix versus neuron behaviour. In: Proceedings of the 2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems. pp. 26–29. ANNES '95, IEEE Computer Society, USA (1995)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
7. Kensinger, E.A., Corkin, S.: Two routes to emotional memory: Distinct neural processes for valence and arousal. Proceedings of the National Academy of Sciences **101**(9), 3310–3315 (2004)
8. Sailunaz, K., Dhaliwal, M., Rokne, J., Alhajj, R.: Emotion detection from text and speech: a survey. Social Network Analysis and Mining **8**(1), 28 (2018)
9. Smith, L.N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 464–472 (2017). https://doi.org/10.1109/WACV.2017.58
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. vol. 1, pp. I–I (2001). https://doi.org/10.1109/CVPR.2001.990517
11. Zhang, L., Wang, S., Liu, B.: Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **8**(4), e1253 (2018)

5 Appendix

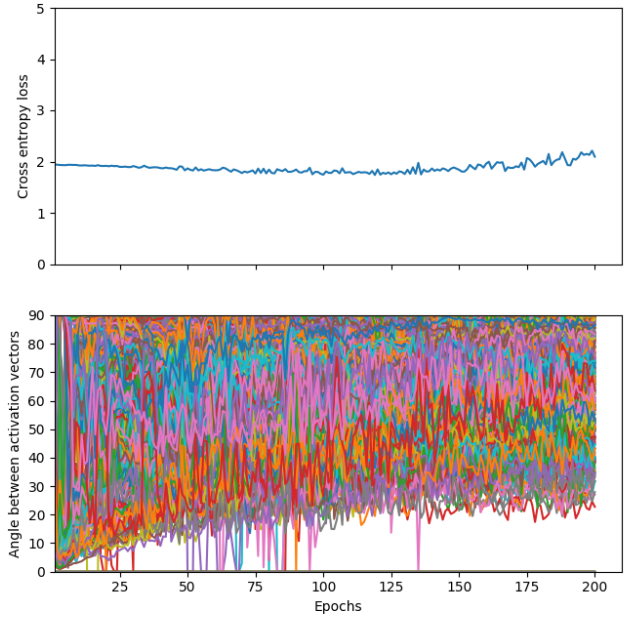

Fig. 4. Changes between stages of the CNN network from the first fold in our 4-Fold cross validation procedure.

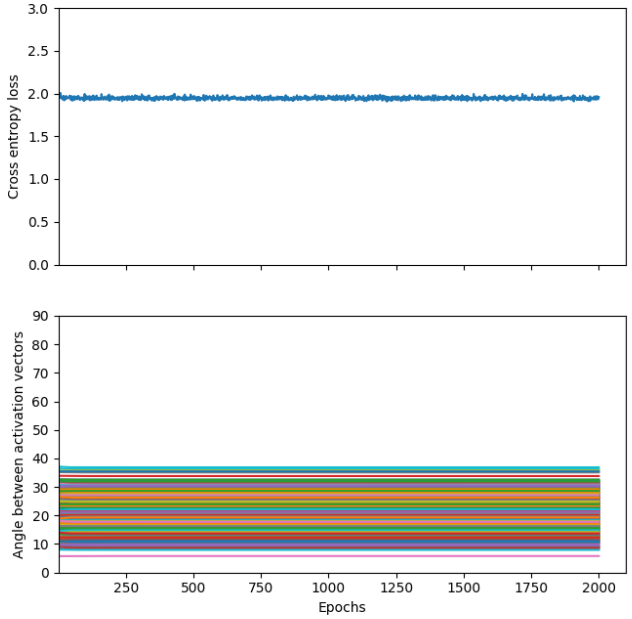

Fig. 5. ANN network is in a constant plateau stage from the first fold in our 5-Fold cross validation procedure.

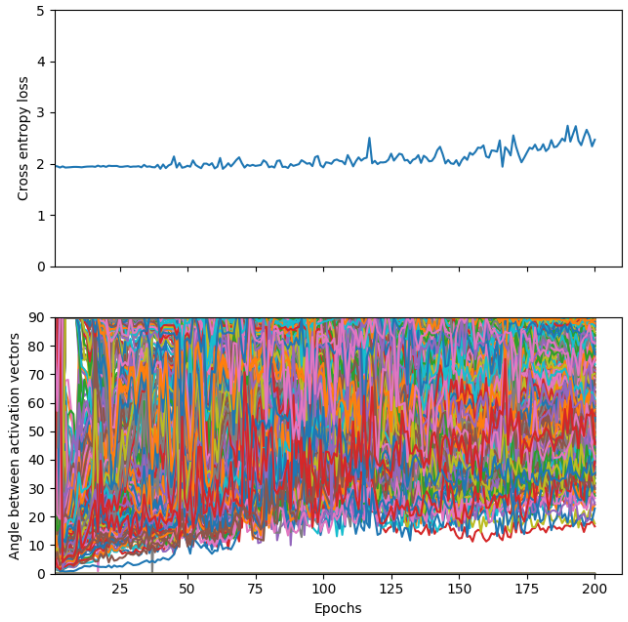

Fig. 6. Changes between stages of the CNN network from the second fold in our 4-Fold cross validation procedure.

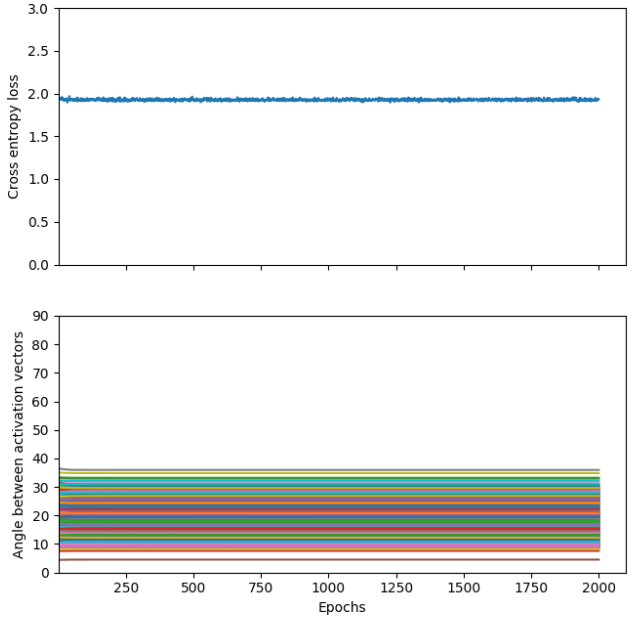

Fig. 7. ANN network is in a constant plateau stage from the second fold in our 5-Fold cross validation procedure.

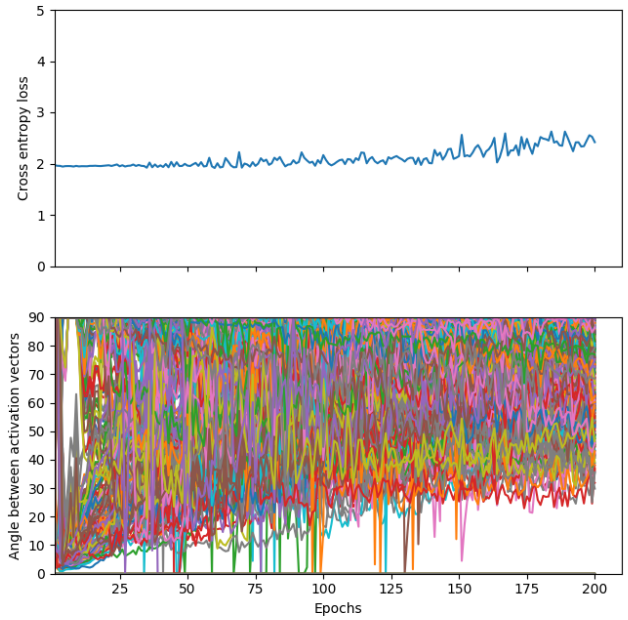

Fig. 8. Changes between stages of the CNN network from the fourth fold in our 4-Fold cross validation procedure.

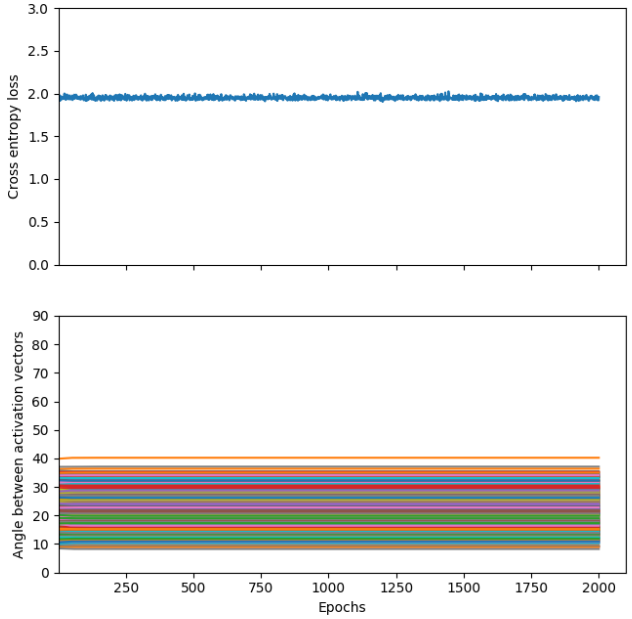

Fig. 9. ANN network is in a constant plateau stage from the fourth fold in our 5-Fold cross validation procedure.

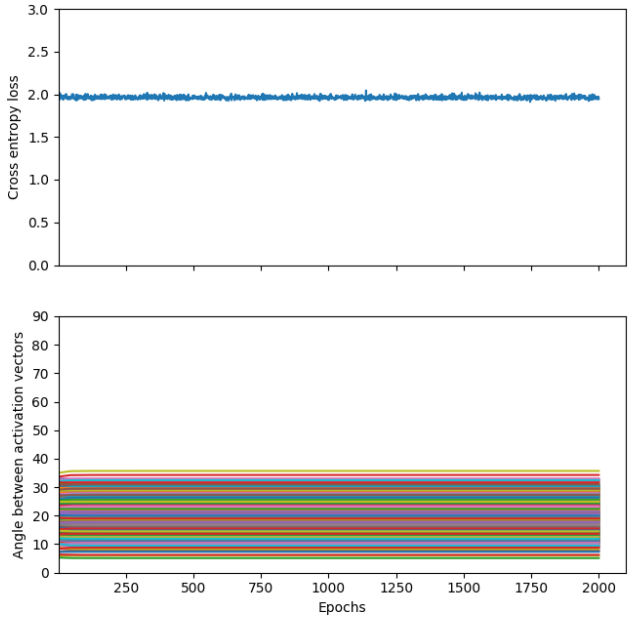

Fig. 10. ANN network is in a constant plateau stage from the fifth fold in our 5-Fold cross validation procedure.