QUEENSLAND
UNIVERSITY OF
TECHNOLOGY

IFN 680: Advanced Topics in Artificial Intelligence

Assignment 2: Siamese network

Xu Han, Earl Chau, Tzu-Wen Chen
School of Information Systems
Faculty of Science and Engineering

Abstract

The performance of Convolutional Neural Network (CNNs) on object detection, verification and recognition benchmarks is rapidly increasing due to the increasing amount of easily accessible annotated datasets and affordable computation power. Furthermore, with constant improvement of our knowledge in deep learning methods, CNNs can recognise complex images such as faces. In this report, three neural network models were created and evaluated by performing four types of testings. These testings comprise training dataset, validation dataset, testing dataset, and a combination of the validation and new testing datasets. Each Convolutional Neural Network model was modified to increase the accuracy of the learning model, which were used to classify the equivalence between two clothing objects in 28 by 28 grayscale images. Modification such as add or remove layers, tuning of the hyperparameters were performed. Siamese network was used to determine if the clothing image pair were deemed from the same equivalence classes. The binary classifier was used to label the image pair with “1” or “0” if it belongs in the same class or not respectively. The training of a Siamese network was then done on a collection of positive and negative pairs produced. The evaluation result of the multiple network architecture performances was documented, analysed, and compared with the goal in finding the most suitable network model structures to determine the best generalisation in object recognition for different dataset created within the Fashion-MNIST.

Introduction

With the aid of machine learning advancements and high bandwidth data services, the image recognition technology market is currently growing rapidly. The image recognition technology market was estimated to grow from 22.8 billion USD in 2018 to 38.92 billion USD by the year 2021, with a compound annual growth rate (CAGR) of 19.5% [1]. This fast-growing development of computer vision had hugely improved image recognition in object classification, verification and recognition. Image recognition technology identifies and detects objects or attributes in a digital video or image. It is a mixture of image detection and classification. Furthermore, in the field of computer vision, deep neural networks (DNNs) have emerged as a leading technique to be used [2]. DNNs are essentially an artificial neural network (ANNs) that has multiple layers between the input and output layers [3]. Moreover, ANNs can be considered to be the most popular and effective model used for classification, clustering, pattern recognition and prediction in the field of image recognition [4]. Due to its data analysis factors of accuracy, processing speed, latency, performance, fault tolerance, volume, scalability and convergence, which can provide high-level capability in handling complex and non-complex problems [4].

One of the most common DNNs is the convolutional neural network (CNN) [4]. Where CNN is a standard neural network that extends across space via shared weights, which is designed to recognize images by having convolutions within, that can recognize the image of an object [4]. CNN has multiple layers which include a fully connected layer, pooling layer, convolutional and non-linearity layers. The fully connected layers and convolutional layers have parameters. However, non-linearity layers and pooling do not have parameters. A study has shown that CNN has an excellent performance in machine learning problems, especially in the applications to image data, like the most extensive image classification dataset, and computer vision [4].

Aim & Method of Investigation

A classifier was built to classify a training set of 60,000 and a test set of 10,000 grayscale images of ten different categories of clothing within the “Fashion-MNIST” dataset [5]. The ten categories consisted: top, trousers, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot. These categories allow the experiment to be compared in different network architectures to find the best learning geometric equivalence. This report will present and discuss the built and trained Siamese network, which was used to distinguish and predict the similarity between two clothing images. A suitable Convolutional Neural Network-based architecture would also be discussed, which would be followed by identifying accuracy and efficiency.

Methods

Google's open-source TensorFlow was implemented for the deep learning algorithms in this experiment [6]. Due to the amount of performance needed to create a machine learning model for this experiment, Google Colaboratory (Colab) was used. Colaboratory is a free Jupyter notebook environment provided by Google, where you can use free GPUs and TPUs, which can enhance the performance of the testings [7].

Fashion-MNIST was based on the European online fashion company, "Zalando" clothing products [5]. The fashion-MNIST dataset contains 70,000, 28x 28 grayscale unique fashion clothing images in 10 different classifications. These classes comprised: top, trousers, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. Moreover, the fashion clothing within the dataset came from different gender groups: men, women, kids and neutral [5]. Within the dataset, 60,000 images were separated into a training set and 10,000 into a testing set.

The dataset was retrieved using the Keras library due to the images being grayscale. Therefore, the data were reshaped into a single colour channel. Because they are ten classes within the dataset, consequently the classes were each given a unique integer. Therefore, by using one-hot encoding for the class element of each sample, transforming the integer into a 10-element binary vector with a 1 for the index of the class value. The pixel values of each image in the dataset are unsigned integers in the range between 0 and 255. The pixel value of grayscale images was normalised and rescaled to the range of 0 to 1. Finally, the Fashion-MNIST training and the testing dataset were pre-processed by pairing up images to create the actual training and testing dataset for this experiment. This actual dataset would be labelled with binary values, which reflect the positivity or negativity of the pairs' equivalence. The two copies of clothing collections were also passed into the same data pre-processing.

Siamese neural network is an artificial neural network (ANN) that uses the same weights and structure while working in tandem on two dissimilar input vectors to compute comparable output vectors, as shown in figure 1. In this experiment, Siamese Neural Networks were implemented to learn the correspondence classes from the image pairs prepared. Furthermore, the model was first trained to ensure the pairs of images with the same clothing type yield a closer output vector than that of the pair of images from the different clothing types [8]. The models were then used to differentiate the similarity metric between the trained input clothing with that of the new samples from unseen categories [8] [9].

In distinguishing the image class similarity, two images, and from the generated pairs would be fed into two convolutional neural networks with the same architecture, denoted as CNN1 and CNN2 respectively. Multiple architectures would then get tested with the original training set, validation testing set, new testing set, and a combination of the validation testing set and the new testing set. The output from both convolutional neural networks would then get passed into the contrastive loss function, where the similarity between the two clothing images will be calculated, as shown in figure 1 below.

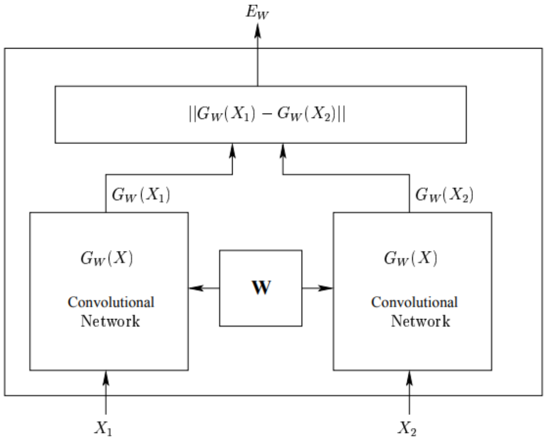


Figure 1: Siamese Network Architecture [8] [10] [11]

Siamese networks are convolutional neural networks that map an input space to a vector space which attempt to obtain the internal patterns in an image [12]. Comparing these vector spaces generated from two images from the same network would provide the distance of the similarity between the two images [12]. However, the purpose of Siamese architecture was to distinguish instead of categorising the image pair [12]. Hence contrastive loss function would be the most suitable classification loss function in this experiment [13]. The contrastive loss function can evaluate the network performance by discovering the similarity between two images. Equation 1 shown below reflects how this contrastive loss function could be used to evaluate the similarity of two images.

$$(1 - Y)\frac{1}{2}(D_w)^2 + (Y)\frac{1}{2}\{\max(0, m - D_w)\}^2 \text{ – [1] [14]}$$

From the above equation, Y signifies the binary label of the pair, where $Y = 0$ if X_1 and X_2 are similar, and if dissimilar, $Y = 1$ [14]. Moreover, the function “max ()”

will select the bigger value between 0 and "$m - D_w$", where m is the margin value, which was used to avoid dissimilar pairs that were beyond the margin to contribute to the loss. A similarity function of Euclidean distance was also involved, which was denoted by, D_w. Mathematically, Euclidean distance calculates the distance of two images by computing the square root distance output vector of the 1st image and the 2nd image to the power of 2, as shown in equation two below.

$$\sqrt{\{G_w(X_1) - G_w(X_2)\}^2} \text{ – [2] [13]}$$

Equation 2 above demonstrates the detailed calculation of Euclidean distance. In this equation, $G_w(X_1)$ and $G_w(X_2)$ refers to the outputs of both neural networks, whereas X_1 and X_2 signify the 1st and 2nd input images.

After training various sets of Siamese models with the training set and the validation set, the prediction results were analysed and compared against the original label. The overall accuracy of each model was then revealed. The convolutional neural networks architectures were then further modified to obtain the optimal prediction accuracy against the new testing dataset. These modifications, such as model layers, or their hyperparameters including dense layer, filters, kernel and pooling size, were performed.

Experiments

The experiments were performed in searching for the optimum strategy in creating a model structure that could accurately distinguish the new test dataset. This comprises altering the layers architecture of the models and tunes its hyperparameters such as dense layers, filters, kernel, pooling size and number of epochs [15]. Hyperparameters are a set of parameters that were set prior to executing the algorithm training process [16]. The network structure and its hyperparameter values selected will greatly impact predicting efficiency. Consequently, the purpose of this experiment will be to determine and realise the optimise layers and hyperparameters that would produce the most efficient performance.

Design of CNN architecture

A modified LeNet-5 CNN architecture was applied to each side of the Siamese network. This modified CNN network based on the hyper parameters test was structured as followed:

Layer 1. A convolutional layer with 32 filters with 3x3 size, no padding. Stride is set to 1. Follow by 2x2 max pooling with a stride is set to 2

Layer 2. A convolutional layer with 64 filters, with 3x3 size and no padding. Stride is set to 1. Follow by max-pooling layer which had a 2x2 pooling and stride is set to 2.

Layer 3. A flatten layer

Layer 4. Fully-connected layer with 120 neurons

Layer 5. Fully-connected layer with 84 neurons

Layer 6. An output Layer with 10 neurons

The activation function for all layers utilised was ReLU. Figure 2 shows the LeNet-5 CNN original structure.

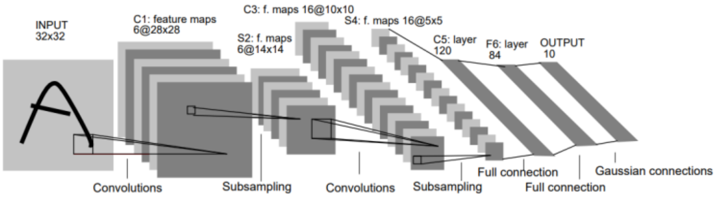


Figure 2: LeNet-5 CNN Architecture [18]

Hyperparameters testing

GridSearchCV method is used to determine the best parameter for our model. Table 1. shows a set of parameters are proposed before this experiment.

Table 1 Set of suggested parameters

PARAMETERS	CANDIDATE VALUE
DENSE LAYER SIZE :	120
FILTER SIZE :	32
KERNEL SIZE :	3 x 3
POOLING SIZE :	2 x 2

Testing the Contrastive loss function

Compare the result of the loss function introduced in the early session with TensorFlow and Numpy method implementation, respectively. Given an equal set of random parameters, both calculations should result in the same value.

Datasets

The fashion-MNIST dataset was used for the experiments. Dataset is divided into 3 sets of data, which are training, validation and testing data. Training and validation data contain 80% and 20% of the images with the label of the top, trouser, pullover, coat, sandal and ankle boot respectively. The testing data contain the image with the label of the dress, sneaker, bag and shirt and none of these images is used during training. To ensure the proposed CNN model learns images from each of the classes evenly, an equal number of images are selected randomly across different classes as the training dataset. Each pair of images, Figure 3, is selected randomly by the program in the dataset pool as training cases. Hence, the training cases should provide a balanced variation of signature features for the network to identify.

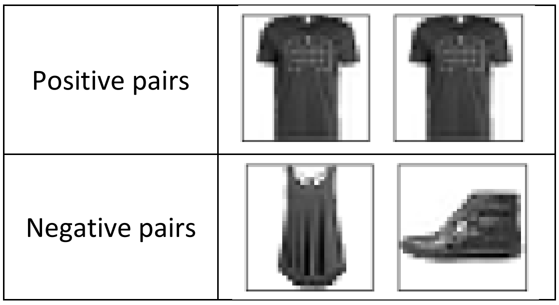


Figure 3 An example of positive pairs and negative pairs image.

For evaluating the generalization capability of the network, three groups of data will be used at the end to compare the performance of the suggested Siamese Network. See table 2. Noted that Set 2 only contains 20% of the entire 6 labels collection of the fashion-MNIST dataset.

Table 2 Design of the evaluation data

Data Set	Labels
1	Top, Trouser, Pullover, Coat, Sandal, Ankle boot
2	Dress, Sneaker, Bag, Shirt
3	Set 1 ∩ Set 2

Experiment Phase I

The experiments were conducted in 2 phases, where the first phase was using the training dataset to decide the best hypermeter for the proposed model. The hypermeter is calculated by grid search with 5-fold cross-validation method.

After the model, architecture was decided. The training dataset is fed into the model given in order to predict the new testing clothing images, which would be fed through in the later phases.

Validation result will compare with the training result to evaluate the model behaviour, and the performance after training has done. This is assumed that training will result in 3 possible scenarios, which are underfitting, overfitting or “well-fit” cases. The behaviour will be observed within the graph that produced.

Overfitting issue is predicted to have happened during training. Therefore, the following experiment was aimed to reduce overfitting behaviour and make the model generalise enough to identify features from different datasets.

Reduce Overfitting

Since our model has approximate 300 thousand parameters. For a small size of the dataset, we could be sure that the network is insufficient to learn all the parameters without considerable overfitting. Therefore, regularisation, dropout and early stopping methods are introduced to the experiment to overcome the overfitting issue.

Therefore, the aim of phase 1 experiment result is going to testify differently reduce overfitting techniques. The validation accuracy and loss value would reveal the influence of each technique. This phase of the experiment is planning to apply one technique at a time and observe the behaviour changes.

Dropout

Dropout is a common and effective method to reduce overfitting issue [17]. The Dropout layer applied at the 1st hidden layer of the network with the value of 0.25. To ensure the one-fourth of random neurons on that layer will be ignored during training. This is to reduce the dependence of a particular neuron that contributes to the following calculation of the network.

Regularization L1 & L2

In this experiment, regularization L1/L2 are applied one at a time, respectively, to evaluate the influence on validation result. The usage of L1/L2 is to reduce the noise of the loss function. L1 refer as Lasso regularization, and it is to penalize the higher degree terms of the function with a very small weight [18]. L2 refer as Ridge regularization, it is calculating the squared weights of the coefficient [18].

Reducing Epoch

During the experiment, the accuracy and loss value was being observed continually. This is to prevent the network overtrained to learn the pattern of the noise, but the correct pattern instead.

At the end of phase 1 experiment, a relative “best-fit” network will be concluded. A graph result of the evaluation will be generated. This is to measure and ensure the network is well generalised to the data other than the training data.

Experiment Phase II

Phase 2 of the experiment is conducted to evaluate the effect of the margin within the contrastive loss function, which was introduced early on in equation1.

It is assumed that the increase in the margin will result in more false-negative pairs results. As the margin m only contributes to the negative pair side of the contrastive loss function.

Therefore, for a given set of margin values, the experiment was expected that the loss value increases according to the rise of the margin. Hence, the accuracy of the prediction will decrease accordingly.

Discussion

Phase I Experiment

Figure 4.1 below illustrates the performance of the model trained using the first CNN structure model. Two graphs were produced, one graph had accuracy plotted against epoch, and the other had loss plotted against epoch. Both graphs contained training and validation datasets.

From observation, the general trend of the accuracy and loss difference between the training and validation datasets seems to be increasing as the epoch increases. From the loss graph above, validation loss can be seen to be increasing, while the training loss can be seen to be steadily decreasing. Hence, this was a sign of overfitting where it had a good performance on the training data, but poor generalisation to other data.

The top validation accuracy was 96.90% at epoch 20, which can be seen to be significantly lower than the training accuracy of 99.22% at epoch 25. Where the two accuracies were seen to have a 2.32% difference, from the trend, the validation dataset can be seen to either stabilise after 20 epochs. Hence, this showed that 20 epochs should be sufficient to train the models. Hence by applying methods such as early termination, regularisation, or/and dropout will possibility help and minimise the variance.

After applying the dropout layer, Figure 4.2, although the overfitting issue remains. However, the validation loss was getting closer to the training loss, which is predicted. Reducing the dependence of particular neuron force the network goes into different fully connected path to identify as pair or not. Moreover, the trend of the validation dataset seen to be stabilised after epoch 20, which agree to the result of the previous Figure. 4.1.

Regularization L1 and L2 are applied respectively on Figure 4.1 validation data. From observation, the L1 regularization reduces the initial loss from 0.062 to 0.053, but maintain the stability of the loss over each epoch. On the other side, L2 increase the learning rate of training. The validation loss has dramatically decreased and got closer to the training loss.

Regularization L2 introduced oscillation to the validation accuracy. This is believed that the L2 lowing the model complexity, which enhance the large weight on the higher complexity values. In other words, this is making the model tend to rely on the higher term rather than, the lower terms. In this case, reducing model complexity will reduce the overfitting issue, but increase the uncertainties of the learning, which led to the fluctuation of the accuracy value.

Combing L1/L2 regularisation and dropout setting, the result showed a dramatic improvement in reducing the overfitting issue. The validation accuracy top at 95.77% at epoch 20 and the trend of the accuracy are increasing. The validation loss is the best fit for the training model among all the trials. This is believed that combining regularisation with dropout, this model is the best fit model we concluded. Figure 4.8 presented the “best-fit” model validation result.

The evaluation test is adopted as the best model setting with Epoch 20, which was adopted according to the validation experiments. This is to prevent further overfitting caused by overtraining.

Three evaluation tests were conducted on the basic model (Network 1) and improvement models (Networks 2 & 3), the overfitting of which is reduced. As a result of Table 3.1, 3.2 and 3.3, they shows the evaluation result with a basic model and improvement models, respectively.

Evaluation 1 (Network 1)	**Accuracy (%)**	**Loss (%)**
Dataset 1 (6 class)	97.03	2.64
Dataset 2 (4 class)	62.12	42.92
Dataset 3 (10 class)	80.94	20.66

Table 3.1 Basic model evaluation result

Evaluation 2 (Network 2)	**Accuracy (%)**	**Loss (%)**
Dataset 1 (6 class)	95.63	4.97
Dataset 2 (4 class)	65.14	24.04
Dataset 3 (10 class)	81.98	14.48

Table 3.2 Improved over fitting model evaluation result

Evaluation 3 (Network 3)	**Accuracy (%)**	**Loss (%)**
Dataset 1 (6 class)	95.10	4.33
Dataset 2 (4 class)	66.51	26.51
Dataset 3 (10 class)	81.22	15.24

Table 3.3 Improved over fitting model evaluation result

Dataset 1 is assumed to be the best case among all the evaluation dataset, as it contains similar images with the training set. The evaluation result is predicted. The accuracy and loss adjusted slightly due to dropout and regularization, where these methods are trying to reduce the possibility of the model to memorize a particular dataset. The effect of reduction is slightly compared to two other datasets, 2 and 3.

Since dataset 2 contain non-trained images only, it is assumed to be the worst case. This assumption is supported by the evaluation result. In addition, the loss improved from 25.21% to 41.28%, as well as slightly on the accuracy. This is ensuring that the reducing of overfitting lead to improvement of generalization of the model.

Dataset 3 is the average case as expected, which contains all 10 classes of images that included similar images to the training set plus non-training class images. The loss result decreases from 20.47% to 15.24% after improvement, where else the accuracy increased by 1.21% slightly.

As a result, the evaluation results as predicted and the overall generalization capability of the model is improved compared to the basic model.

Phase II Experiment

Further investigation on the margin value **m** in the contrastive loss function with the same model above. Table 4 shows the evaluation result of different margin values on the model behaviour and performance.

Table 2 The effect of margin value (m)

Margin Value	**(m)**	**0.55**	**0.75**	**1.00**	**1.25**	**1.50**
Dataset 1 (6 class)	Accuracy (%)	88.60	92.86	96.02	94.27	91.45
	Loss (%)	2.00	3.11	4.37	7.36	16.66
Dataset 2 (4 class)	Accuracy (%)	70.36	68.06	66.81	63.87	64.9
	Loss (%)	8.76	16.49	26.60	38.31	55.65
Dataset 3 (10 class)	Accuracy (%)	79.53	81.32	82.50	80.68	78.76
	Loss (%)	4.25	9.49	13.91	23.28	36.24

The result of loss value changes as predicted; the increase of the margin (m) will increase loss value ***n*** for three evaluation data. According to equation 1, the value ***m*** in contractive loss function is determined the acceptance of negative pairs. It also means that the increase of the margin will reduce the differentiation of an incoming pair in this Siamese network. Hence, it is making the network is harder to detect dissimilar images.

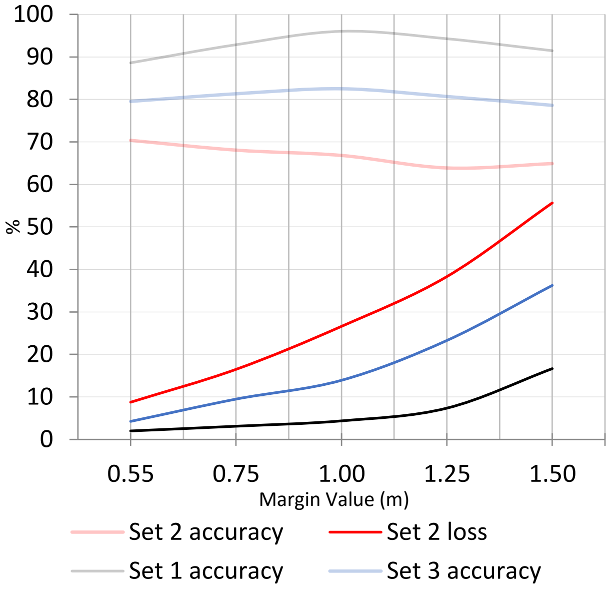


Figure 5 Evaluation result on the margin value of contrastive loss function in 3 evaluation data

According to the design of the evaluation data, the Test dataset, *Set 2*, should recognise the highest number of negative pairs as the network did not familiar with new types of class. It is obvious that the rate of loss in increase the most among the three groups of evaluation data.

The accuracy decreases against the increase of margin in all 3 cases. However, it is not obvious as predicted. Set 1 and Set 3 accuracy only dropped less than 5%. Set 2 has the highest decrease. It is believed that one of the reasons is the result of the fluctuation of applied dropout and L2 regularization to overcome the overfitting issue.

Since loss is presenting the confidence of a pair that is correctly predicted by the network, as the margin goes up, the confidence of the correctly distinguish a pair is harder. However, if the margin value set to be too low, it is believed that will result in more false-positive pairs, as it "good" at recognising similar images only. Therefore, a balanced margin should be applied for a better network.

Overall, Set 2 is the best case to evaluate and test the lost function. As it contains non-training images only, the accuracy would be the lowest. Hence, it is believed that the change of margin that affects the behaviour of the contrastive loss function, in this case, should be greater than the other two sets.

Fig 4.1 : Tranining vs validation data

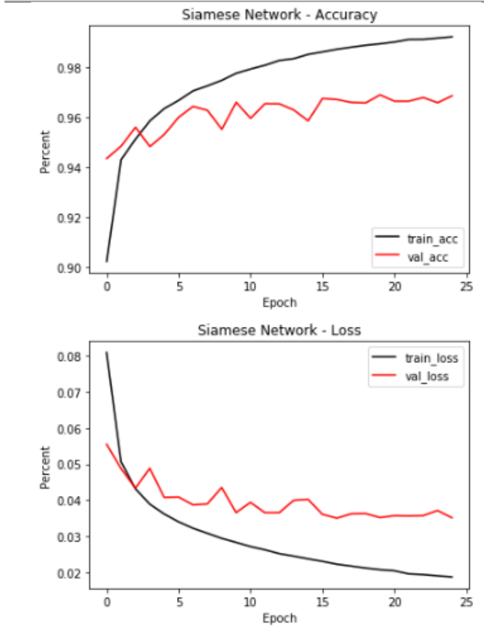


Fig. 4.2 : Dropout: 0.25, layer applied

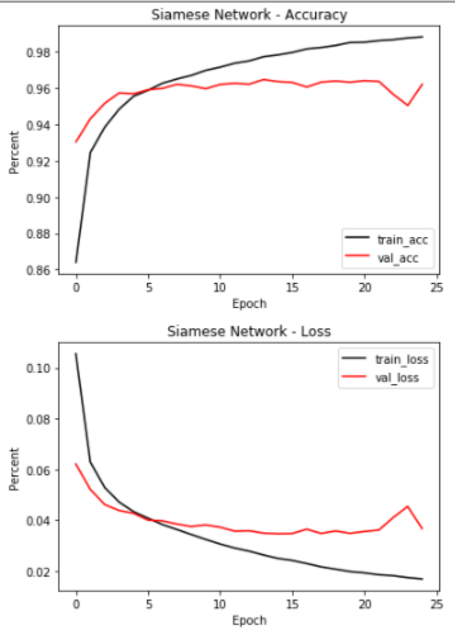


Fig.4.3 : L1 : 0.01 applied

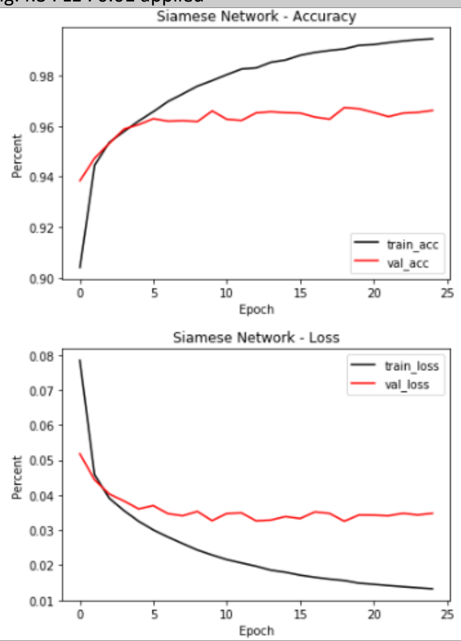


Fig 4.4 : L2 : 0.01 applied

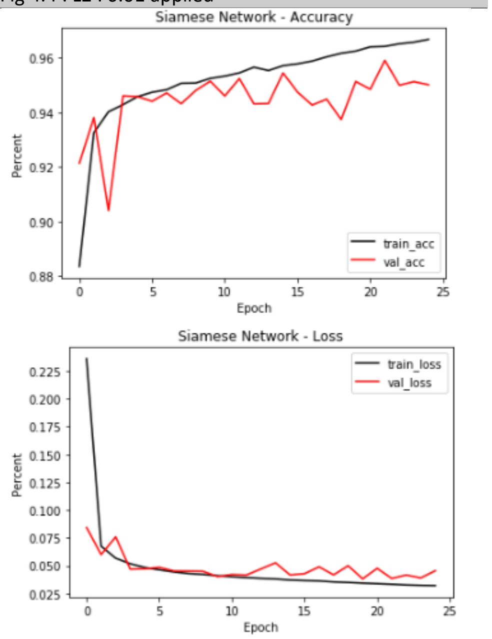


Fig 4.5 : L1:0.01 & L2:0.01 applied

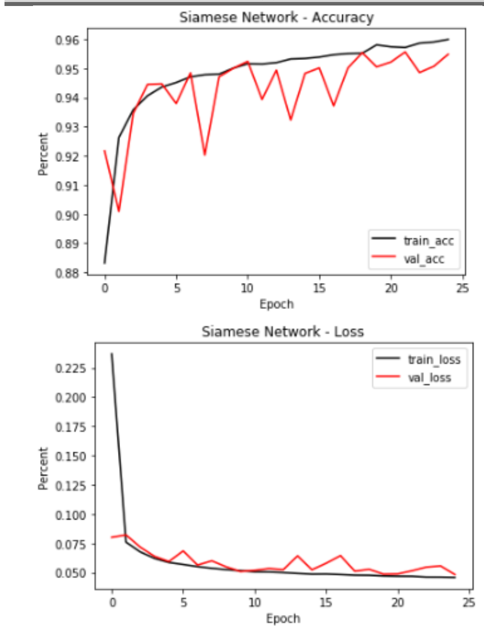


Fig 4.6 : L1:0.01 & Dropout : 0.25 Layer applied

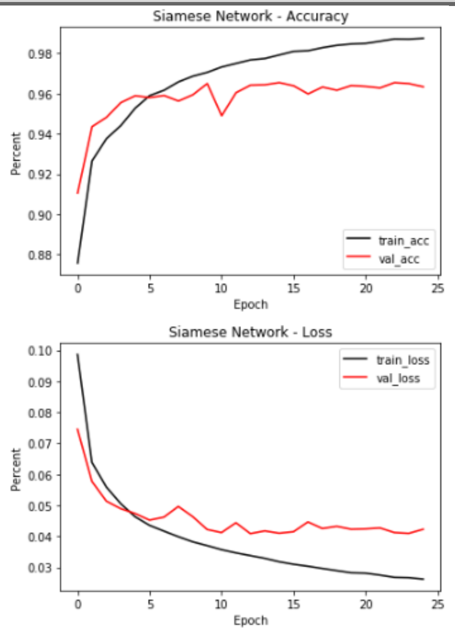


Fig 4.7 : L2:0.01 & Doupout : 0.25 Layer applied

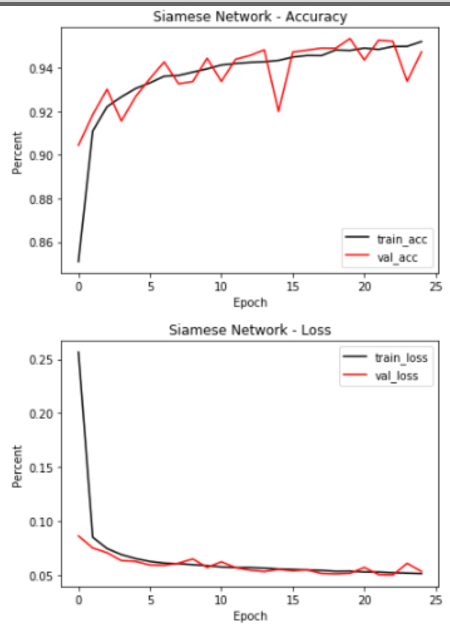
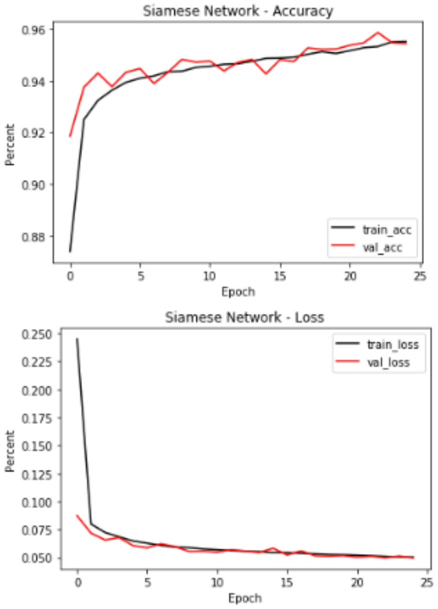


Fig 4.8 : L1:0.01 & L2:0.01, Dropout : 0.25 applied



Future improvements

Further investigation could be made in order to reveal this phenomenal. Two possible improvements could be introduced. The first suggestion would be using a confusion matrix to investigate further how a pair of the image is selected before the network and how they predicted after going through the Siamese's network. The second suggestion would be introducing extreme cases scenario. An evaluation data set that does not contain negative pairs. Hence, it will predict that the loss value from negative pair side will be eliminated.

Conclusion

Distinguishing two images can be considered to be the most fundamental tasks in image analysis and computer vision. By comparing two images to realise if both images correspond to each other is quite challenging as there exist many factors that would affect the image, such as shapes and edges.

In this experiment, overfitting methods such as dropout and regularising were demonstrated on the three neural network models created where the result had shown a significant boost in performance as the overfitting reduction methods were added.

Reference

[1] Grand View Research, “Image Recognition Market Analysis Report By Technique, By Application (Augmented Reality, Security & Surveillance, Scanning & Imaging), By Component, By Deployment Mode, By Vertical, And Segment Forecasts, 2019 - 2025,” Grand View Research, 2019.

[2] A. Canziani, A. Paszke and E. Culurciello, “An Analysis of Deep Neural Network Models for Practical Applications,” 24 May 2016. [Online]. Available: https://arxiv.org/abs/1605.07678. [Accessed 20 Oct 2019].

[3] Imperial College London, “Machine Learning Algorithms,” Imperial College London, [Online]. Available: https://www.imperial.ac.uk/structural-integrity-health-monitoring/research/structural-health-monitoring-/shm-methodologies/passive-sensing-methodologies/machine-learning-algorithms-/. [Accessed 2019].

[4] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” Heliyon, vol. 4, no. 2018 Nov, p. 11, 2018.

[5] H. Xiao, K. Rasul and R. Vollgraf, “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms,” 25 Aug 2017. [Online]. Available: https://arxiv.org/abs/1708.07747v1?utm_campaign=Artificial%2BIntelligence%2Band%2BDeep%2BLearning%2BWeekly&utm_medium=web&utm_source=Artificial_Intelligence_and_Deep_Learning_Weekly_28. [Accessed 20 Oct 2019].

[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard and Y. Jai, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” 14 Mar 2016. [Online]. Available: https://arxiv.org/abs/1603.04467. [Accessed 20 Oct 2019].

[7] Google, “GoogleColaboratory.,” Google, 2019. [Online]. Available:https://colab.research.google.com/notebooks/welcome.ipynb.

[8] S. Chopra, R. Hadsell and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, San Diego, CA, United States, 2005.

[9] G. Gao, L. Liu, L. Wang and Y. Zhang, “Fashion clothes matching scheme based on Siamese Network and AutoEncoder,” Multimedia System, 2019.

[10] J. Bromley, I. Guyon, Y. LeChun, E. Sackinger and R. Shah, “Signature Verification using a "Siamese" Time Delay Neural Network,” lInternational Journal of Pattern Recognition and Artificial Intelligence, vol. 7, no. 4, pp. 737-744, 1993.

[11] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, “DeepFace: Closing the Gap to Human-Level Performance in Face Verification,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, 2014.

[12] D. J. Rao, S. Mittal and S. Ritika, “Siamese Neural Networks for One-shot detection of Railway Track Switches,” 21 Dec 2017. [Online]. Available: https://arxiv.org/abs/1712.08036. [Accessed 20 Oct 2019].

[13] H. Gupta, “Facial Similarity with Siamese Networks in PyTorch,” 14 Oct 2019. [Online]. Available: https://hackernoon.com/facial-similarity-with-siamese-networks-in-pytorch-9642aa9db2f7. [Accessed 20 Oct 2019].

[14] R. Hadsell, S. Chopra and Y. LeCun, “Dimensionality Reduction by Learning an Invariant Mapping,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CCVP), 2006.

[15] N. Mboga, C. Persello, J. R. Bergado and A. Stein, “Detection of Informal Settlements from VHR Images Using Convolutional Neural Networks,” Remote sensing, vol. 9, no. 11, p. 18, 2017.

[16] DeepAI, “What is a hyperparameter?,” DeepAI, 2019. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/hyperparameter. [Accessed 20 Oct 2019].

[17] M. Rizwan , “LeNet-5 – A Classic CNN Architecture,” 30 9 2018. [Online]. Available: https://engmrk.com/lenet-5-a-classic-cnn-architecture/.

[18] Y. LeCun, B. Leon, B. Yoshua and P. Haffner, “Gradient-Based Learning Applied to Document Recognition,” Proceedings of the IEEE, 1998.