VIETNAM NATIONAL UNIVERSITY

UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

PROJECT 03

MEMBERS

Lâm Ngọc Phương Anh	18127039
Hoàng Nguyên Trúc	18127055
Võ Ngọc Minh	18127154

Subject: Artificial Intelligence

Ho Chi Minh City – 2020

TABLE OF CONTENTS

TABLE OF CONTENTS 2
ASSIGNMENT PLAN 3
ABSTRACT 5
ENVIRONMENT 6
INTRODUCTION 7
APPROACH 8
1. Network architecture 8
2. Associated learning algorithm 9
EXPERIMENT 10
EXAMINE THE EFFECTS OF HYPERPARAMETERS AND PARAMETERS 14
CONCLUSION 15
REFERENCES 16

ASSIGNMENT PLAN

MSSV	NAME	TASK	%

18127039	Lâm Ngọc Phương Anh	- Implement function:
		+ get_y_train()
		+ elm()
		+ convert_png()
		+ show_images1()
		+ show_images()
		+ process_data()
		+ draw()
		- Report:
		+ Approach
		+ Introduction
		- Video Recorder
			100

18127055	Hoàng Nguyên Trúc	- Implement function:
		+ get_images()
		+ get_labels()
		+ wr_model()
		+ write_csv()
		+ plot_image_list_count1()
		+ mnist()
		+ plot_image_list_count()
		- Report:
		+ Experiment
		+ Conclusion
			100

18127154	Võ Ngọc Minh	- Implement function:
		+ load_mist()
		+ input_to_hidden()
		+ show_images1()
		+ fashion()
		+ label_pet_image_one _hot_encoder()
			100

		+ dogcat() + menu() - Report: + Title, Authors + Abstract + References	

ABSTRACT

★ The extreme learning machine (ELM) is a new method for using Single-hidden Layer Feed-forward Networks (SLFNs) with a much simpler training method. Experimental results in the experiment section show the effectiveness of the proposed model in object classification.

★ Here we set few parameters used in the model.

- The image size is 28 (MNIST and MNIST-Fashion) and 200(Kaggle Dogs and Cats).
- The images are stored in two folders, train and test.
- There are 10 image classes in MNIST and MNIST-Fashion.
- There are two image classes: Dog and Cat in Kaggle Dogs and Cats dataset.
- A subset of the training data set for MNIST and MNIST-Fashion are 60,000 images.
- A subset of the training data set for Kaggle Dogs and Cats are 25,000 images.
- A trained from scratch model will be used.

ENVIRONMENT

□ IDE: Visual Studio Code

□ Programming Language: Python

□ Video recorder: Bandicam

INTRODUCTION

- Problem: Object classification using Neural networks.

- Overall plan:

  + Based on previous research before about neural network, our team pick ELM neural network as the framework to solve this problem and spend one day to experiment the model in order to have an insight into ELM neural network.

  + Load data from The MNIST dataset of handwritten digits to create train set then apply ELM model to object classification.

  + Load data from The Fashion-MNIST dataset of Zalando's article images (clothes, bags, shoes, etc.) to create train set then apply ELM model to object classification.

  + Load data from The Kaggle dataset of cats and dogs to create train set then apply ELM model to object classification.

- Youtube link:

  🗀 ELM MNIST video demo

  🗀 ELM MNIST-Fashion demo video

  🗀 Dogs and Cats video demo. This video demonstrates a 5-running time to choose the best model by applying ELM to dogs and cats dataset.

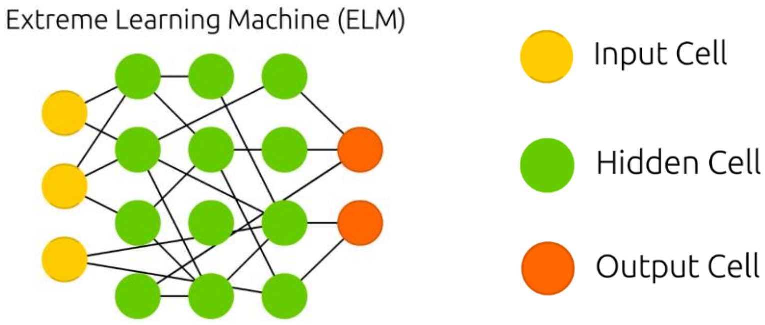
APPROACH

Extreme learning machines are basically FFNNs but with random connections and they are not recurrent nor spiking. Instead of using backpropagation and updating weights after each iteration, it starts with random weights and trains the weights in a single step by applying the least-squares fit.

1. Network architecture

ELM includes input layer, output layer and hidden layer(s) with hidden nodes are computational elements, which need not be considered as classical neuron. A hidden node in ELM can be classical artificial neurons, basis functions, or a subnetwork formed by some hidden nodes.

A neuron is a placeholder for a mathematical function in the machine learning universe, which the only job is to apply the function on the given inputs to produce an output. Because of this characteristic, the number of neurons that ELM has is the total of inputs and outputs and hidden nodes.



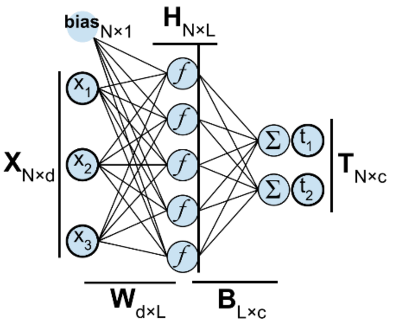
Assume that ELM model in this report is standard SLFN with L hidden nodes and activation function h(x) are mathematically modeled as:

$$o_j = \sum_{i=1}^{L} \beta_i h_i(x_j) = \sum_{i=1}^{L} \beta_i h(\omega_i \cdot x_j + b_i), j = 1, \dots, N$$

With:

- x: input vector.
- ω_i: the weight vector connecting the i^th hidden node and the input nodes.

- β_i: the weight vector connecting the ith hidden node and the output nodes.
- b_i: the threshold of the ith hidden node.
- h_i: the activation function.
- N: the number of training samples.



It can be compactly written as:

$$H\beta = T$$

⇨ ELM simply solve the function by: $\beta = H^{\dagger}T$

$H^{\dagger}$: the Moore-Penrose generalized inverse of H (Hidden Layer Output Matrix).

2. Associated learning algorithm

- Randomly create weights matrix ω and bias b for input layer.
- Calculate the output H of hidden layer.
- Calculate the output weight matrix.

$$\beta = H^{\dagger}T \ (H^{\dagger} = (H^{T}.H)^{-1}.H^{T})$$

- Create new randomly weights matrix and use β to make a prediction on new data set: T = H. β.

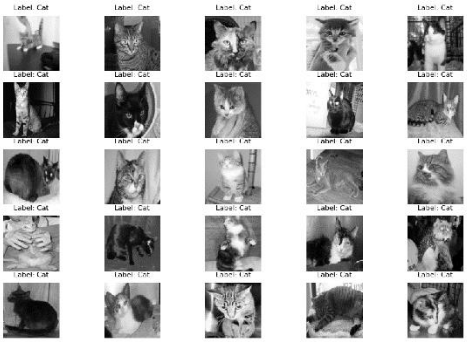
EXPERIMENT

- Dataset: The Kaggle dataset of cats and dogs, MNIST-Fashion, MNIST.

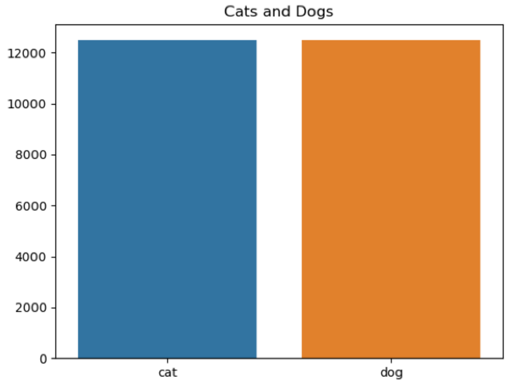
- Research object classification technique by using Google.

- Calculate predicted labels, then compare with labels from the test set to evaluate the results (expected result is 0.8 /1).

- The Kaggle dogs and cats dataset:

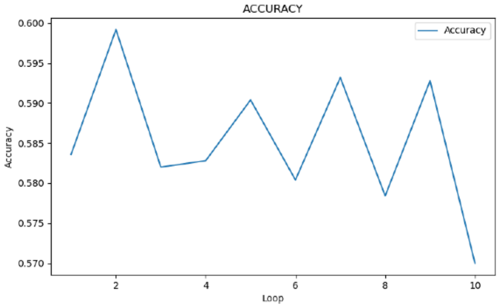


Train set



12500 pictures are cats and 12500 are dogs

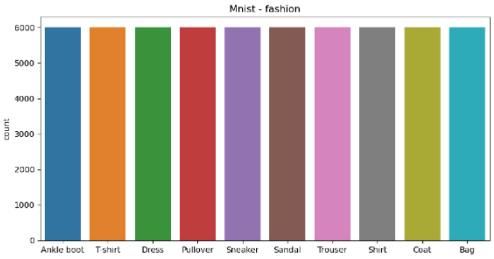
Test set



Accuracy rate

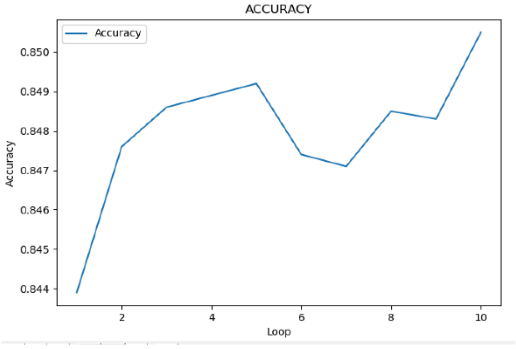
- MNIST-Fashion dataset:

Train set



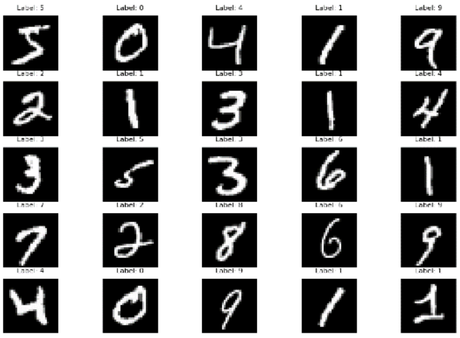
Each category has 6000 pictures

Test set

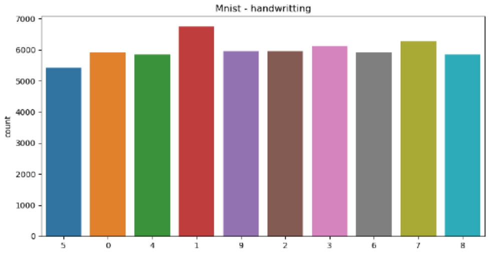


Accuracy rate

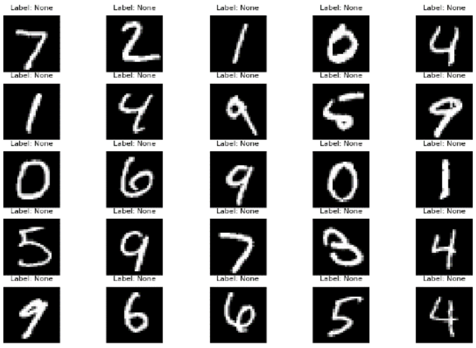
- ➢ MNIST dataset:



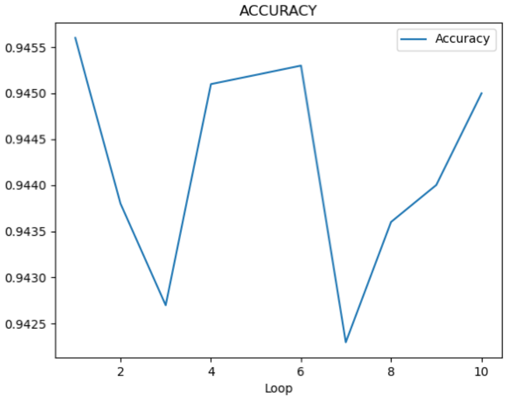
Train set



Number of pictures for each category



Test set



Accuracy rate

EXAMINE THE EFFECTS OF HYPERPARAMETERS AND PARAMETERS

- In our neural network model, hyperparameters include:

  + Hidden layers
  + Hidden units
  + Activations functions
  + Epoch

  ✰ A neural network without an activation function is essentially just a linear regression model. The activation function does the non-linear transformation to the input making it capable to learn and perform more complex tasks.

  ✰ Hidden layers allow for the function of a neural network to be broken down into specific transformations of the data. Each hidden layer function is specialized to produce a defined output. For example, a hidden layer functions that are used to identify human eyes and ears may be used in conjunction by subsequent layers to identify faces in images. While the functions to identify eyes alone are not enough to independently recognize objects, they can function jointly within a neural network.

  ✰ The numbers of epochs are related to how diverse the data is.

- In our neural network model, parameters include:

  + Weights

  ✰ Weights control the signal (or the strength of the connection) between two neurons. In other words, a weight decides how much influence the input will have on the output.

  + The best weight of ten running times of each dataset (per one executing times): https://drive.google.com/drive/folders/1-fQ1s9soivB7usVFlE6W5gOyx0F4OfnY?usp=sharing

CONCLUSION

↳ Method using linear regression is powerful, efficient, easy to use and can be applied to a variety of fields.

↳ The biggest advantage of linear regression models is linearity: It makes the estimation procedure simple and, most importantly, these linear equations have an easy to understand interpretation on a modular level (i.e. the weights).

↳ Data science and machine learning are driving image recognition, autonomous vehicles development, decisions in the financial and energy sectors, advances in medicine, the rise of social networks, and more. In the future, linear regression will play a vital part of this.

REFERENCES

- https://www.sciencedirect.com/science/article/abs/pii/S0925231215011315
- https://www.sciencedirect.com/science/article/abs/pii/S0925231215011315
- https://medium.com/datadriveninvestor/extreme-learning-machine-for-simple-classification-e776ad797a3c
- https://erdem.pl/2020/05/introduction-to-extreme-learning-machines
- https://www.asimovinstitute.org/neural-network-zoo/
- https://en.wikipedia.org/wiki/Extreme_learning_machine
- https://www.techopedia.com/definition/33293/extreme-learning-machine-elm
- https://www.sciencedirect.com/topics/engineering/extreme-learning-machine
- https://www.ntu.edu.sg/home/egbhuang/pdf/ELM-NC-2006.pdf
- https://www.ripublication.com/ijaer17/ijaerv12n14_70.pdf
- https://www.hindawi.com/journals/mpe/2017/4670187/
- https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90
- https://pdfs.semanticscholar.org/8df9/c71f09eb0dabf5adf17bee0f6b36190b52b2.pdf
- https://www.kaggle.com/gpreda/cats-or-dogs-using-cnn-with-transfer-learning?fbclid=IwAR1ymlVhnnQ9oojL26peP_YKJGljquNuUdn3T3qbx_g0gJaeaCkaTKl6kNE
- https://medium.com/@matthew.buscemi/abstraction-in-elm-conclusion-6ea7adcf398e
- https://www.geeksforgeeks.org/activation-functions-neural-networks/?fbclid=IwAR2KsqOd2RCqyFuPiVSnIB8f3RKFaUC4oBxXaqwn64Ebx6OT0DFLl4tJOvE
- https://docs.paperspace.com/machine-learning/wiki/weights-and-biases
- https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning?fbclid=IwAR2KsqOd2RCqyFuPiVSnIB8f3RKFaUC4oBxXaqwn64Ebx6OT0DFLl4tJOvE#:~:text=In%20neural%20networks%2C%20a%20hidden,inputs%20entered%20into%20the%20network
- https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9#:~:text=One%20epoch%20leads%20to%20underfitting,to%20optimal%20to%20overfitting%20curve.
- https://www.kaggle.com/robertbm/extreme-learning-machine-example?fbclid=IwAR1st3MqsBR6S0VzqkVK8DrXLVtu_H5YjlcXcIHGJ1X3UHb2ZgU9ZYP4YOk
- Code from previous project in Nhap Mon Cong Nghe Thong Tin 2