A Novel CNN-LSTM Model for Image Classification Problems

Group 24

Li Xinwei Luo Chengchang Wei Jingjue

***Abstract*—This document is a progress report, containing our group's research plan, completed work, ongoing work, and challenges facing in the process.**

I. Future Plan Overview

We have already finished the first part — constructing two baseline CNN models and evaluating them based on the efficient metrics. Therefore, the left is to add LSTM layers into the baseline. We use the gantt graph to show the details:

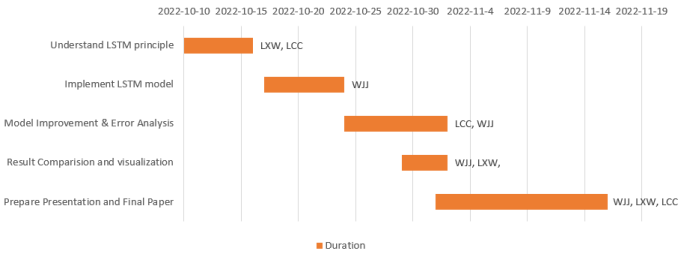

Fig. 1. The Future Project Plan

II. Completed Work

A. Updated Literature Part

1) CNN+LSTM Application in Image Classification: We have complemented the highly relevant paper (i.e. "A novel CNN+LSTM classification model based on fashion-MNIST" by Yaran Ji) in our reference [1]. Besides, in order to make up for the lack of "CNN-LSTM" part in our literature review, we searched relevant papers to introduce its current application in image classification, describing their structures and performances as well as comparing the differences of CNN+LSTM in these applications.

Among the applications of CNN-LSTM, some used paralell CNN-LSTM model where CNN and LSTM layers as separate components, dealing with input data simutaneously, and combined their outputs together in the later steps [10]. While some used a sequential CNN-LSTM model, where CNN extracted features from the input data and LSTM did classificationn or prediction based on the outputs of CNN [11]–[13]. Both two kinds of models outperformed other approaches only using one of them. As for performance on different kinds of datasets, it turns out that CNN-LSTM model performs well both in prediction based on time-slice images [10] and multi-label classification based on complicated images [11].

2) Leading Method Performance on Fashionmnist: Fashionmnist is a well-known and excellent dataset for testing image classification methods. There are dozens of algorithms tested on it and post their accuracy publicly. The highest accuracy currently goes to the Dual-path network with wide resnet 28-10, whose accuracy is 95.7% (without validation) [6]. Given the performance of a method is strongly affected by its computing resource and running time, we only list those methods using similar computing resources as ours with a leading accuracy in Table I.

TABLE I
Benchmark method perfromance on Fashionmnist

Model	Structure and Methods	Accuracy
Human Performance	1000 random sample per person; 3 label choice each image; No fashion expert;	83.50% [6]
CNN+LSTM	Cov1: $(3, 3, 20)$; [a] Maxpooling: $(2, 2)$; Cov2: $(5, 5, 20)$; LSTM + FC + softmax. Optimizer: Maximum Likelihood.	91.36% [1]

CNN	Cov1: $(5, 5, 32)$,
	padding=same;
	Maxpooling: $(2, 2)$,
	stride = 2;
	Cov2:$(5, 5, 64)$;
	Maxpooling: $(2, 2)$
	stride = 2;
	FC+ softmax.
	Optimizer: GradientDescentOptimizer;
	Batchnorm and dropout are used.
		91.60%
		[6]

CNN Model with SVM Classifer[b]	Cov1: $(5, 5, 32)$; Maxpooling: $(2, 2)$; Cov2:$(5, 5, 64)$; Maxpooling: $(2, 2)$; FC + SVM. Batchnorm and dropout are used.	90.72% [7]

[a](m,n,k) denoted the $m \times n$ kernel whose kernel number is k.
The padding and stride default are 0 and 1,
which will be specifically annotated if not the default value.

[b] SVM is short for Support Vector Machine

B. Modelling

1) Explorative Research on Fashionmnist:

- Label Classification: For Fashion-MNIST dataset, there are 60000 examples in training set and 10000 examples in test set, both having 10 labels. The distributions of 10 labels in training set (each with 6000 samples) and test set (each with 1000 samples) are fairly uniform. Fig.2 and Fig.3 show the box plots of mean and variance of total 70000 examples in each label. We can see that the means and variances are different among the labels, especially between labels of shoes (E.g., Sandal & Sneaker) and labels of clothes (E.g., T-shirt/top & Pullover), since their pixel grey values are different, which depends on the volume of the object.

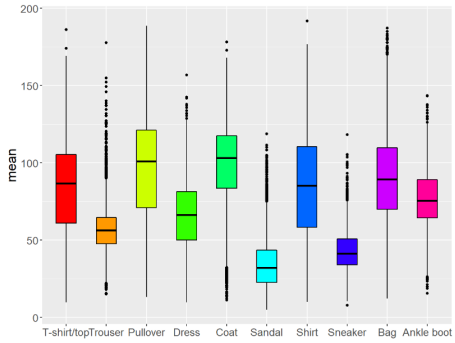

Fig. 2. Box plot of means

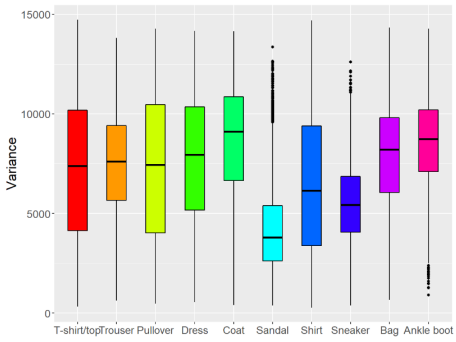

Fig. 3. Box plot of variances

- tSNE Visualization on Raw Dataset: According to Fig.4, most labels are clustered obviously but still overlap each other sometimes, which indicated an advancing image classifier for this task is necessary.

2) Baseline Model Performance: In this project, we have constructed two CNN models with the same layer structure, but different dimensions as the baseline. The main structure is that we developed two convolutional layers with two max-pooling layers, followed by two fully connected layers. Finally, the output will be the result of the softmax function.

In addition, since we are working on a multi-labeled problems and compute these metrics in a micro way, the $F1 = Accuracy = Recall$. Therefore, we only listed $Accuracy$. The results of these two models are as follows:

Based on the performance results(Table II) and the confusion matrix (Fig. 5). We drew the ROC curve of the worst

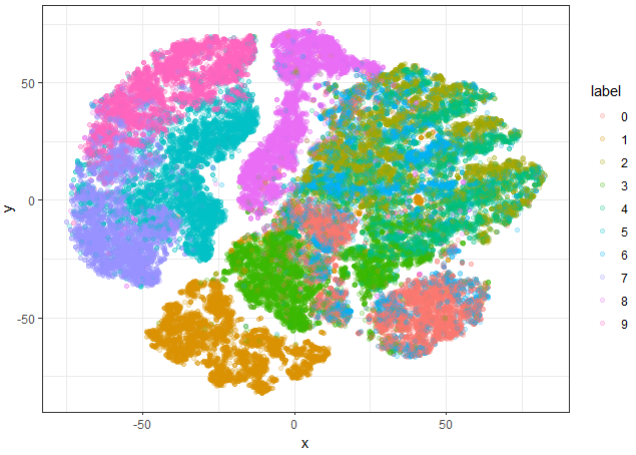

Fig. 4. tSNE plot with PCA initialization for the whole dataset

TABLE II
TWO BASELINE MODEL STRUCTURE

Model	Structure [a]	Accuracy
Model I	Conv1: (1,8,5); Maxpooling: (2,2); Conv2: (8,32,5) Maxpooling: (2,2); FC1 : (512,64) FC2: (64,10) Softmax;	86.55%
Model II	Conv1: (1,16,5); Maxpooling: (2,2); Conv2: (16,32,5) Maxpooling: (2,2); FC1 : (512,32) FC2: (32,10) Softmax;	86.28%

[a] Model layers' structure follows Pytorch API standard
Conv(in_channels,out_channels, kernel_size)
The padding and stride default are 0 and 1

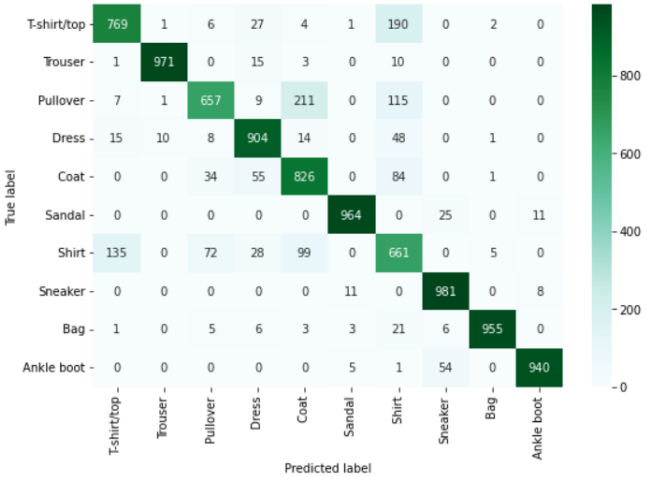

Fig. 5. The Confusion Matrix of Model 1

performance label, which is the label Shirt (Label 6) (Fig. 6). our baseline model works well.

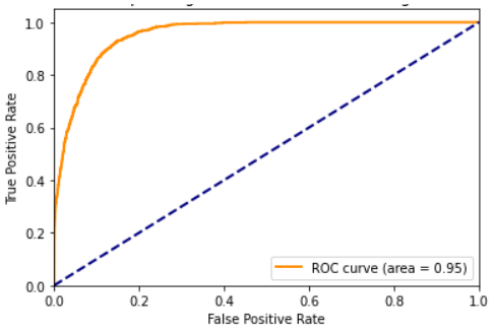

Fig. 6. Receiver Operating Characteristic (ROC) for Shirt class

III. Ongoing Work

A. Model Improvement

1) Future Model Architecture: Since we already finished the baseline model, and constructed an entire working pipeline, We will focus on adding LSTM layers into the baseline model (Figure is in the Appendix: V), and evaluate their results.

2) Preprocessing and Regularization Techniques:

- Random Filp: Random flip for the training dataset image: Randomly flip the image horizontally, vertically or both to augment the dataset with a given probabillity.
- Batch Normalization : Separating the whole dataset into n minibatch and using different batches in each epoch efficiently speed up the gradient descending. Barch normalization refers to normalizing the input data in each layer for each epoch. This method is controversial that whether it can address the "internal covariate shift" problem. But it is widely acknowledged that batch normalization is helpful to increase accuracy. [8]
- Drop Out: Drop out is a powerful tool to prevent deep neural network from overfitting by random remove neurons with a given probability at specific layer each time.
- Optimizers: There are several different optimizers worth trying to get a higher accuracy, such as CrossEntropy(), Adam(), Nadam(), MaximumLikelihood(), GradientDescentOptimizer(), SVM etc..

IV. Challenge from the Research Proposal

A. Computing Resource

1) Problem Statement: Since we are working on a deep leaning project. We need a lot of computing resources. But, our personal computers cannot handle this. It takes 20-30 mins when we run the code.

2) Solution: A temporal solution is to use computers in the department's Lab. We already set up a proper environment in Anaconda for many Lab's computers. Therefore, we can run code and try our different ideas at the same time.

B. Success Metric Implementation

1) Problem Statement: t-SNE (t-distributed Stochastic Neighbor Embedding) is a state-of-art method for mapping high-dimensional data into lower dimensions and visually revealing the clustering property of the dataset. According to our research proposal , we treated t-SNE as an assistant success metric for model performance illustration, where accuracy, precision, and $F1$ score are still regarded as the main standards. [5]

However, the implementation of t-SNE for the raw Fashion-mnist dataset and literature learning revealed some theoretical drawbacks of treating t-SNE as an evaluation standard. [3]

- Random initalization of t-SNE makes uncertain and possible false clustering each time.
- Different parameter tuning lead to the disparate outcome on the the same dataset.The suitable parameter selection is tricky and necessarily determined case by case. Counterintuitively, even though a larger iteration number K can lead to the convergence, it may cause "overshooting", which makes a worse clustering performance.
- Over-interpretation risk of clustering plot comparison. Given the raw data are seriously distorted in t-SNE, the cluster membership is reliable but the relative position between clusters aren't meaningful.

2) Solution: We still keep t-SNE as an illustration for our model performance, but will specially take the following techniques to ensure the accuracy of plot as much as possible.

- Instead of the default random initialization at the exaggeration stage of t-SNE, We are going to apply Principle Component Analysis (PCA) for informative initialization, which can keep a faithful global data structure and can be realized by the Python module open-tsne now. [4] [2]
- Given the strong dependence between the t-SNE performance and its parameter choosing, We set the exaggeration stage iteration number as $K_0 = \lfloor (log(n))^2 \rfloor$ to avoid "overshooting", where n is the data size. [3]
- Accuracy and interpretability metrics will be applied to ensure the tSNE plots of the models are equally in compare. [9]

References

[1] Ji, Y. (2022, June). A novel CNN+ LSTM classification model based on fashion-MNIST. In *International Conference on Neural Networks, Information, and Communication Engineering (NNICE)* (Vol. 12258, pp. 178-184). SPIE.

[2] D. Kobak and G. C. Linderman, "Initialization is critical for preserving global data structure in both t-SNE and UMAP," Nat Biotechnol, vol. 39, no. 2, pp. 156–157, Feb. 2021, doi: 10.1038/s41587-020-00809-z.

[3] T. T. Cai and R. Ma, "Theoretical Foundations of t-SNE for Visualizing High-Dimensional Clustered Data." arXiv, Mar. 28, 2022. Accessed: Oct. 10, 2022. [Online]. Available: http://arxiv.org/abs/2105.07536

[4] P. G. Poličar, M. Stražar, and B. Zupan, "openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding," Bioinformatics, preprint, Aug. 2019. doi: 10.1101/731877.

[5] Maaten, Laurens van der and Geoffrey E. Hinton. "Visualizing Data using t-SNE." Journal of Machine Learning Research 9 (2008): 2579-2605.

[6] "Fashion-MNIST".[Online].Available: https://github.com/zalandoresearch/fashion-mnist.git

[7] A. F. Agarap, “An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification.” arXiv, Feb. 07, 2019. Accessed: Oct. 10, 2022. [Online]. Available: http://arxiv.org/abs/1712.03541
[8] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” p. 11.
[9] R. Gove, L. Cadalzo, N. Leiby, J. M. Singer, and A. Zaitzeff, “New guidance for using t-SNE: Alternative defaults, hyperparameter selection automation, and comparative evaluation,” Visual Informatics, vol. 6, no. 2, pp. 87–97, Jun. 2022, doi: 10.1016/j.visinf.2022.04.003.
[10] Li, P., Abdel-Aty, M., & Yuan, J. (2020). Real-time crash risk prediction on arterials based on LSTM-CNN. Accident Analysis & Prevention, 135, 105371.
[11] Hua, Y., Mou, L., & Zhu, X. X. (2019). Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. ISPRS journal of photogrammetry and remote sensing, 149, 188-199.
[12] Tasdelen, A., & Sen, B. (2021). A hybrid CNN-LSTM model for pre-miRNA classification. Scientific reports, 11(1), 1-9.
[13] Ozkok, F. O., & Celik, M. (2022). A hybrid CNN-LSTM model for high resolution melting curve classification. Biomedical Signal Processing and Control, 71, 103168.

V. Appendix-Future CNN-LSTM Model Sturcture