COMP34212 Deep Learning and Robotics

Rui Xu 10891143

Task1 Literature Review

The application of deep learning has grown significantly in the last few years in a number of cognitive robotics domains. In the field of cognitive robotics, deep learning models have been extensively used and shown to be effective in tasks like object recognition and perception, object grasping and manipulation, and scene understanding. In this literature, the first two tasks are introduced.

Detection and Perception

Object detection and perception stand as fundamental tasks within computer vision, and they have garnered extensive research attention for cognitive robotics over the years. Convolutional neural networks (CNNs) have showcased their efficacy in various applications, ranging from recognizing off-road obstacles [10] to detecting cracks [7], leveraging stereo images as input. Moreover, CNNs have streamlined aerial robotics navigation, enabling them to traverse forest trails with a single monocular camera [6]. An open-source ground robot called SROBO [12] was designed to accurately identify its position and navigate certain areas using a deep CNN and transfer learning. Furthermore, recent strides in robotics navigation have been propelled by the integration of deep learning methodologies, exemplified by the contributions of Sermanet et al. [10] and Hadsell et al. [11]. Chen et al. [13] extended this trend by employing a CNN to discern the existence and orientation of doors. They then incorporated this data into a navigation algorithm designed for mobile robots, showcasing the practical utility of deep learning in enhancing robotic capabilities.

It's worth noting that all the supervised learning methods mentioned above necessitate significant efforts in collecting and labeling datasets. Tao et al. [12]. designated the center sample of the clustering result for object classification as a semi-supervised approach. Despite this, due to the necessity for auxiliary judgments, unsupervised learning methods have not fundamentally eliminated the need for labeling work.

Grasping and Object Manipulation

The capacity of robots to mimic human-like grasping and execute in-hand manipulation of objects is indispensable for their integration into dynamic human environments and their potential to substitute human labor. In many applications of this field, CNN is also an important implementation method.

It is crucial to emphasize that the majority of research defines "grasp" as an end-effector configuration that involves either partial or complete closure around a specific object, achieved through placement and configuration of the automatic clutch. Yang et al. [16] trained a massive CNN to predict the chance that task-space motion of the gripper will result in successful grasps, thereby facilitating the learning of hand-eye coordination for grasping. Yang, Li, and Fermüller [17] also trained a seven-layer CNN to recognize 48 kitchen objects and classify human grasps using 88 YouTube cooking videos. The system achieved 79% object recognition accuracy and 91% grasp classification accuracy.

Some researchers [14, 15] suggest in-hand manipulations that could maintain discontinuous contact; it establishes a designated workspace adjacent to the robotic arm, facilitating precise placement and subsequent retrieval of grasped objects at varying configurations. TacGNN [18] proposed a novel framework using Hierarchical Graph Neural Network(GNN) for tactile-based in-hand manipulation learning with a blind anthropomorphic robotic hand, i.e. without visual sensing. Mariolis, Peleka, and Kargakos [1] conducted a study on object and pose recognition for garments suspended from a single point, simulating scenarios where they are grasped by robotic grippers. A pose estimation of grasp points on the garment with a mean error of 5.3cm was outperformed by their CNNs.

Critical Analysis

The ability to adapt to new uncertainties in real time through interactions between the robot's body and its environment sets cognitive robotics apart from traditional cognitive systems in other domains. Although deep learning models are widely used in different applications of robotics, when the complexity of the environment includes ambiguity in behavioral goals and communication with humans, generalization of the environmental information is necessary for robots. In the application cases mentioned above, all models are trained based on specific datasets in their fields, which is time and energy consuming. Creating a general model that can be applied to a wider range of cognitive robot tasks could be a current challenge. Inspired by OpenAI's CLIP cross-modal architecture [22], a feasible solution could be clustering multimodal features, which enables the extraction of key characteristics that effectively represent the category while filtering out less relevant features.

Task2 ResNet for CIFAR

In the realm of deep learning, the Residual Network (ResNet) architecture has emerged as a powerful tool for **image classification tasks**. In this research endeavor, our primary focus is on evaluating the performance of the **ResNet** architectures specifically applied to the **CIFAR-10 dataset**. Moreover, beyond mere performance evaluation, our objective extends to investigating the nuanced impact of different hyperparameters on the optimization process of the ResNet model.

Model Introduction

Residual Neural Networks (ResNet) [20] have emerged as fundamental components in neural network architectures due to their remarkable performance and efficacy in training exceptionally deep models. ResNet's innovation lies in its implementation of residual connections, which address the challenge of vanishing gradients encountered in earlier architectures. By introducing residual blocks, ResNet enables the learning of residual functions, wherein the input to a block is added to its output. This unique approach facilitates the training of very deep networks, a feat that was previously hindered by architectural limitations. This lightweight model used in this assignment, compared to the standard ResNet18, **reduces the number of residual blocks in the second, third, and fourth layer groups.** Consequently, it features six fewer convolutional layers than ResNet18 due to this modification in block structure. This reduction in complexity offers certain advantages when training on the CIFAR-10 dataset. Firstly, the computational efficiency is improved due to the reduced number of parameters and layers, making the model less resource-intensive. Secondly, the risk of overfitting is mitigated.

Dataset Explanation and Justification

To comprehensively assess the complexity and performance of our ResNet models, we utilized the CIFAR-10 dataset. The CIFAR-10 dataset consists of 60,000 32x32 color images across ten classes; its relatively small image size and diverse class distribution make it an ideal choice for exploring ResNet's performance in image classification tasks. The dataset is divided into training (80%), validation (10%), and testing (10%) sets, facilitating performance monitoring and hyperparameter tuning using the validation subset. **Normalization** is performed by subtracting the mean and dividing by the standard deviation of each channel [19]. Techniques such as r**andom horizontal flipping, padding, and random cropping** are also employed to **augment the training dataset and increase the sample size.**

Experiment 1: Learning Rate Comparison

In this experiment, we conducted a comparison test regarding the learning rate in deep learning training. The objective of this experiment is to explore the impact of different learning rates on the model training process. The learning rate is a crucial hyperparameter in deep learning, as it determines the step size of model parameter updates, directly affecting the convergence speed and final performance of the model during training. During the experiment, we keep other conditions unchanged apart from

different learning rates. **The results are shown below(See Figure 1)**. It seems that the smaller the learning rate, the better the model performance. However, it is difficult to find the best set of parameters for training this model just by trying multiple sets of different fixed learning rates. A more professional method is needed to accurately find the best set of parameters.

Experiment 2: Hyperparameters Tuning

To determine the optimal hyperparameters, we employed the Bayesian optimization technique known as 'Gaussian process regression' [20]. It works by constructing a probability distribution over possible functions and updating this distribution as new data points are observed. This method made it easier to find the hyperparameter configuration, which reduced validation errors. Our search space encompassed key hyperparameters such as the learning rate, the weight decay, the beta parameters for the Adam optimizer, and the momentum parameter for SGD Optimizer(More details please see Experiment 3). Utilizing an objective function, we trained and validated the model with the specified hyperparameters, using validation accuracy as the optimization metric.

This iterative process is repeated for a fixed number of epochs (100 epochs), resulting in the identification of the best hyperparameters that yielded the highest validation accuracy. Through this process, **the optimal hyperparameters for Adam Optimizer were determined as follows: learning rate=0.00016, beta1=0.960, beta2=0.226, and weight decay=0.0004. For SGD Optimizer, the best results are: learning rate=0.03, momentum=0.57, wand eight decay=0.0001**

Experiment 3: Optimizer Comparison

This experiment aims to analyze the performance of different optimizers, mainly focusing on comparing the effectiveness of Stochastic Gradient Descent (SGD) and Adam. Adam, an adaptive learning rate optimization algorithm, combines the benefits of both AdaGrad and RMSProp by maintaining separate learning rates for each parameter and adapting them based on the moving average of both the gradients and their squares. In contrast, SGD updates model parameters in the direction opposite to the gradient of the loss function concerning the parameters, with a fixed learning rate. The result is show below(Figure 2).

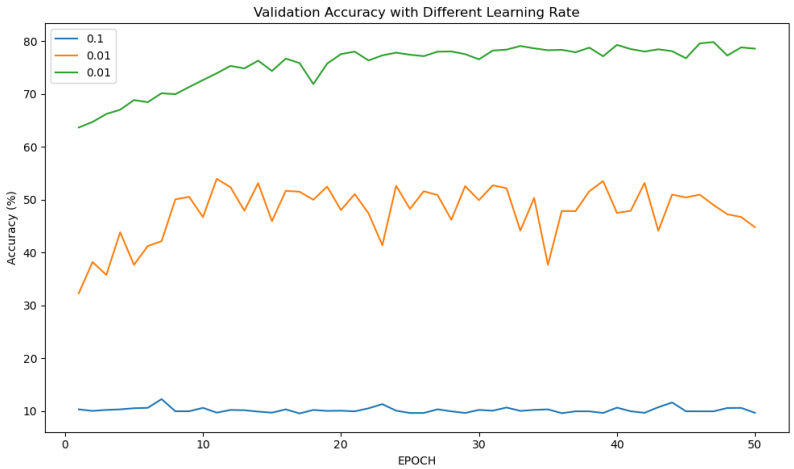

Figure 1

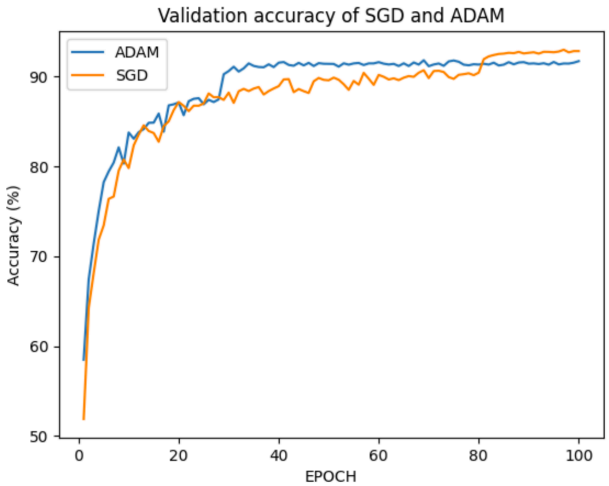

Figure 2

When considering speed and stability from the perspective of the accuracy trends, Adam tends to exhibit smoother and more consistent convergence trajectories. Its adaptive learning rate mechanism allows for faster initial progress in accuracy, contributing to stable and reliable performance throughout training. In contrast, SGD may display more unstable behavior in accuracy trends, potentially exhibiting fluctuations or slower convergence due to the fixed learning rate. When considering performance, as

training progresses, SGD can catch up or even surpass Adam's performance, especially in scenarios where the learning rate schedule is finely tuned to the dataset's characteristics.

Model training and Evaluation

For the model training and evaluation phase, we utilized the best hyperparameters obtained through Bayesian optimization. The model underwent training for 100 epochs, with the training and validation accuracy and loss recorded for each epoch. Subsequently, the model is evaluated on the testing set, and the test accuracy and loss were recorded. The final test accuracy achieved by the model under the optimal hyparameters is 91.65%.

Additional Experiment: Fine-tuning ResNet for CIFAR-10

In addition to the above standard experiments, an additional set of experiments on model architecture optimization for the CIFAR-10 is also conducted. In the adaptation of ResNet18 for the CIFAR-10 dataset, several adjustments were made to enhance its effectiveness based on the above observation and deeper investigation. We adapted the architecture to better suit the small image size of CIFAR data, replacing the 7x7 downsampling convolution and pooling layers with a 3x3 downsampling convolution to preserve image information(both stride and padding set to 1). Additionally, the max-pooling layer is ineffective and should be removed due to the small size of the images. The output of the last fully connected layer is changed to 10. To optimize training, a learning rate of 0.1 is initially set, with a provision to decay it by 50% if the loss did not decrease after 10 epochs. CrossEntropyLoss is employed as the loss function, and Stochastic Gradient Descent (SGD) served as the optimizer. **With a training accuracy of 94.4% and a validation accuracy of 95.1%, alongside a training loss of 0.024 and a validation loss of 0.16, the model exhibits stable and consistent performance on both training and validation datasets(no overfitting).** Additionally, the validation accuracy and loss remain steady throughout the training process, indicating a well-generalized model. These results suggest that the model effectively captures the underlying patterns in the data without excessively fitting to the training set or failing to learn from it.

Further Discussion (alternative/future simulations)

Exploring alternative deep learning architectures like Transformer and DenseNet alongside ResNet can offer valuable insights. Transformer models, known for their success in NLP, have shown promise in image classification due to their attention mechanism capturing long-range dependencies. Similarly, DenseNet's dense connections facilitate feature reuse and gradient flow, potentially enhancing representation. Evaluating these models in future can provide a comprehensive understanding and identify the most suitable architecture for CIFAR classification tasks.

Link to external code repository(GitLab @ UoM): https://gitlab.cs.man.ac.uk/p00331rx/comp34212 This is an internal repository for UoM members, you might need to log in first. If you can not access this repository, please email me: rui.xu-11@student.manchester.ac.uk Thank you!

References

1. *Mariolis, G. Peleka, A. Kargakos and S. Malassiotis, "Pose and category recognition of highly deformable objects using deep learning," 2015 International Conference on Advanced Robotics (ICAR), Istanbul, Turkey, 2015, pp. 655-662, doi: 10.1109/ICAR.2015.7251526.*
2. *Zhang, F, Leitner, J, Milford, M, Upcroft, B & Corke, P 2015, Towards vision-based deep reinforcement learning for robotic motion control. in Australasian Conference on Robotics and Automation, ACRA 2015. Australian Robotics and Automation Association (ARAA), Australasian Conference on Robotics and Automation 2015, Canberra, Australian Capital Territory, Australia, 2/12/15.*
3. *Lenz I, Lee H, Saxena A. Deep learning for detecting robotic grasps. The International Journal of Robotics Research. 2015;34(4-5):705-724. doi:10.1177/0278364914549607*

4. *Redmon Joseph and Anelia Angelova. “Real-time grasp detection using convolutional neural networks.” 2015 IEEE International Conference on Robotics and Automation (ICRA) (2014): 1316-1322.*
5. *Lecun, Yann & Muller, Urs & Ben, Jan & Cosatto, Eric & Flepp, Beat. (2005). Off-Road Obstacle Avoidance through End-to-End Learning..*
6. *Giusti, Alessandro & Guzzi, Jerome & Ciresan, Dan & He, Fang-Lin & Gómez, Juan & Fontana, Flavio & Faessler, Matthias & Forster, Christian & Schmidhuber, Jurgen & Di Caro, Gianni & Scaramuzza, Davide & Gambardella, Luca Maria. (2015). A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. IEEE Robotics and Automation Letters. 1. 1-1. 10.1109/LRA.2015.2509024.*
7. *Cha, Young-Jin & Choi, Wooram & Buyukozturk, Oral. (2017). Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. Computer-Aided Civil and Infrastructure Engineering. 32. 361-378. 10.1111/mice.12263.*
8. *Tai, Lei & Li, Shaohua & Liu, Ming. (2017). Autonomous exploration of mobile robots through deep neural networks. International Journal of Advanced Robotic Systems. 14. 172988141770357. 10.1177/1729881417703571.*
9. *Sadeghi Esfahlani, Shabnam, Alireza Sanaei, Mohammad Ghorabian, and Hassan Shirvani. 2022. "The Deep Convolutional Neural Network Role in the Autonomous Navigation of Mobile Robots (SROBO)" Remote Sensing 14, no. 14: 3324. https://doi.org/10.3390/rs14143324*
10. *Sermanet, Pierre & Hadsell, Raia & Scoffier, Marco & Grimes, Matthew & Ben, Jan & Erkan, Ayse & Crudele, Chris & Muller, Urs & Lecun, Yann. (2009). A Multirange Architecture for Collision-Free Off-Road Robot Navigation. J. Field Robotics. 26. 52-87. 10.1002/rob.20270.*
11. *Hadsell, Raia & Sermanet, Pierre & Scoffier, Marco & Erkan, Ayse & Kavackuoglu, Koray & Muller, Urs & Lecun, Yann. (2009). Learning Long-Range Vision for Autonomous Off-Road Driving. Journal of Field Robotics. 26. 120-144.*
12. *Xie, Chris & Patil, Sachin & Moldovan, Teodor & Levine, Sergey & Abbeel, Pieter. (2016). Model-based reinforcement learning with parametrized physical models and optimism-driven exploration. 504-511. 10.1109/ICRA.2016.7487172.*
13. *Chen, Wei & Qu, Ting & Zhou, Yimin & Weng, Kaijian & Wang, Gang & Fu, Guoqiang. (2015). Door recognition and deep learning algorithm for visual based robot navigation. 2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014. 1793-1798. 10.1109/ROBIO.2014.7090595.*
14. *P. Tournassoud, T. Lozano-Perez and E. Mazer, "Regrasping," Proceedings. 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, USA, 1987, pp. 1924-1928, doi: 10.1109/ROBOT.1987.1087910*
15. *Lipeng Chen, Luis F. C. Figueredo, and Mehmet Dogar. 2019. Manipulation Planning Using Environmental Contacts to Keep Objects Stable under External Forces. In 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids). IEEE Press, 417–424. https://doi.org/10.1109/Humanoids43949.2019.9034998*
16. *Yang, Linhan & Huang, Bidan & Li, Qingbiao & Tsai, Ya-Yen & Lee, Wang & Song, Chaoyang & Pan, Jia. (2023). TacGNN: Learning Tactile-Based In-Hand Manipulation With a Blind Robot Using Hierarchical Graph Neural Network. IEEE Robotics and Automation Letters. PP. 1-8. 10.1109/LRA.2023.3264759.*
17. *Yang, Yezhou & Li, Yi & Fermüller, Cornelia & Aloimonos, Yiannis. (2015). Robot Learning Manipulation Action Plans by " Watching " Unconstrained Videos from the World Wide Web.*
18. *Levine, Sergey & Pastor, Peter & Krizhevsky, Alex & Quillen, Deirdre. (2016). Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. The International Journal of Robotics Research. 37. 10.1177/0278364917710318.*
19. *Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115 (2015): 211-252.*
20. *Snoek, Jasper & Larochelle, Hugo & Adams, Ryan. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. Advances in Neural Information Processing Systems. 4.*
21. *He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.*
22. *Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry et al. "Learning transferable visual models from natural language supervision." In International conference on machine learning, pp. 8748-8763. PMLR, 2021.*