COMP34212 Cognitive Robotics

Shurong Ge

1. Introduction to the problem and the methodology

This coursework address the **problem** of image classification by perfoming training simulations on a neural network classifier, some big datasets are involved as the training data. In addition, it is important to figure out the relationship between the hyperparameter value chosen and the training results produced by using that hyperparameter value. Thus, we are able to perform a optimization on robotics training.

Consider the above objectives, Convolutional Neural Network (CNN) was chosen as the **network topology**, while CIFAR-10 is the training set.The **hypeparameter exploration** will be operated based on different number of epochs, learning rate, dropout rates as a percentage, optimisers (e.g. Adam, RMSprop), batch size, number of convolutional layers, and activation function.

2. Robotics and deep learning

2.1 Overview

Cognitive robotics is a rapidly growing interdisciplinary research field that aims to create intelligent robots that are able to perceive the surrounding environment, execute actions, and learn from the experiences in order to adapt their created behaviours to interaction in a appropriate way. Robots are getting closer to filling the gap between theoretical science and real-world implementations thanks to growing collaboration between different disciplines in this field (Aly et al., 2017).The involvement of deep learning in cognitive robotics, allows robots to understand and anticipate human behaviour and intentions and interact with their environment in the same way as people (Pierson and Michael, 2017).

2.2 The role of deep learning

Deep learning plays an important role in achieving the goal of machines with the ability to perceive the real world. It has the potential to exploit and develop the real-world structures, constraints and physical laws that govern robotic tasks. However, it still has some limitation, robotic vision is facing learning challenges, embodiment challenges and reasoning challenges (Sünderhauf et al, 2018).

2.3 Spectrum of deep learning approaches

The introduction of activation functions and biological similarity concept allows regression methods to be adapted to non-linear functions. The non-linear models are stacked in order to form more powerful models, which known as multilayer perceptrons. The deep learning revolution began with a training method called backpropagation in the 1980s. By this time neural networks were successfully used in robot control (Pierson and Michael, 2017). Neural networks were further built in order to overcome limitations such as the number of layers in a neural network, resulting in multilayer neural networks such as Convolutional Neural Network (CNN), Deep Neural Network (DNN), and Recurrent Neural Network (RNN), which gave robots human-like brains to achieve the objective of intelligent robots (Fung, 2018).

3. Network, hyperparameters and dataset

The layer structure of **CNN** determines its complexity. The subsequent experiments refer to the neural network structure in lab2b, i.e. conv+maxpool+dropout. Deeper networks, based on the similar structure but with multiple convolution and pooling layers are connected in a chain (e.g. conv+conv+maxpool+dropout+conv+conv+maxpool), were also implemented in the experiment.

The experiment uses a grid search to **explore hyperparameters**. The step is, firstly, defining an n-dimensional grid with each dimension corresponding to a hyperparameter; then defining a set of possible values for each dimension; finally, searching all possible configurations and waiting for the results to decide the best configuration. It is worth noting that putting too many parameters in the grid is time consuming, which known as *curse of dimensionality*. Therefore, in this experiment, there may be multiple hyperparameters in a single grid to ensure that the dimensionality of the grid search is less than or equal to 4.

CIFAR-10 **dataset** is used in the experiments, which contains 60000 32x32 colour images, while 50000 images will be used to train the network classifer and the remaining images will be considered as testing data. These images can be grouped into 10 different categories. CIFAR-10 as a machine learning benchmark set guarantees the quality of the data. Moreover, it has a sufficiently rich (in terms of both type and number) sample size, which meets the sample requirements of deep learning (i.e. deep learning for high-dimensional spaces requires exponentially increasing numbers of samples as the dimensionality increases). **Data Processing** is easy since it is already enclosed in KERAS package. Thus, the steps are as follows, first of all , loading the dataset by importing Keras; secondly, converting the output into categorical data; finally, performing normalization on the input data.

4. Hyperparameter testing simulations

CNN networks, in descending order, are the filtering layer (using a filter and activated by a correlation function), the maximum pool layer, the dense layer (requiring an activation function), and finally the softmax layer. Thus, each convolutional block consist of conv+maxpool+dropout, while the classification layers (dense layers) are made up of dense+drop+dense.

(All the experiments are carried out on a 8-Core Intel Core i9 CPU @ 2.3 GHz with 1T of RAM and Mac OS.)

4.1 Different Activation Function

Different activation functions (i.e. Sigmoid Function, Hyperbolic Tangent Function, Rectified Linear Unit Function, Leaky Rectified Linear Unit Function) will be used for the other layers, except for the last layer where the softmax function cannot be changed.
Other configurations: number of epochs - 20, learning rate - 0.01, dropout rates - 0.25, optimisers - Adam, batch size - 256, number of convolutional layers - 3.

Function Name	Sigmoid	Hyperbolic Tangent	Rectified Linear Unit	Leaky Rectified Linear Unit
Validation Accuracy	0.578940	0.755970	0.778730	0.809520

Table 1: Different activation funtions corresponding to their validation accurancy

According to the validation results, it is clear to see that Leaky Rectified Linear Unit Function gives the best validation accuracy, then Rectified Linear Unit Function, the third is Hyperbolic Tangent Function, while Sigmoid Function performs the worst.

4.2 Different Number of Convolutional Layers & Dropout Rate
The number of convolutional layers changes from 1 to 4 (increased by 1), while the dropout rate set as 0, 0.25 and 0.5.
Other configurations: number of epochs - 20, learning rate - 0.01, optimisers - Adam, batch size - 256, activation function - ReLU.

Number of convolutional layers \ Dropout rate	1	2	3	4
0	0.713520	0.724570	0.746940	0.739250
0.25	0.687430	0.723890	0.760210	0.731460
0.5	0.684750	0.685370	0.697380	0.659120

Table 2: Validation accurancy corresponding to different convolutional layers and dropout rate

The table shows that when number of convolutional layers = 3 and dropout = 0.25, it is likely to get highest accuracy.

4.3 Different Number of Epochs

The experiment will use different number of epochs, which are 10, 20, 50, 100.
Other configurations: learning rate - 0.01, optimisers - Adam, dropout rates - 0.25, number of convolutional layers - 3, activation function - ReLU, batch size -256.

Number of Epochs	10	20	50	100
Validation Accuracy	0.689530	0.701680	0.754960	0.749720

Table 3: The effect of different amounts of epochs on validation accuracy

Thus, having a big epoch size does not always mean better accuracy. The epoch size will improve accuracy up to a point, after which the model will begin to overfit. Underfitting may also be caused by a small epoch scale.

4.4 Different Batch Size

Since the GPU performs better for batches of powers of 2, the batch size is set as 32, 64, 128, 256.
Other configurations: number of epochs - 20, learning rate - 0.01, optimisers - Adam, dropout rates - 0.25, number of convolutional layers - 3, activation function - ReLU.

Batch Size	32	64	128	256
Validation Accuracy	0.756270	0.754790	0.760210	0.758920

Table 4: The impact on validation accuracy using different batch size

From the table, we can learn that as batch size does not affect the accuracy since it is to used to control the running speed regarding the memory size in GPU.

4.5 Accuracy and loss results based on different number of epoch

The results are displayed in the figure below, i.e. after new model training, the tranining/testing accuracy is increased while the loss is decreased.

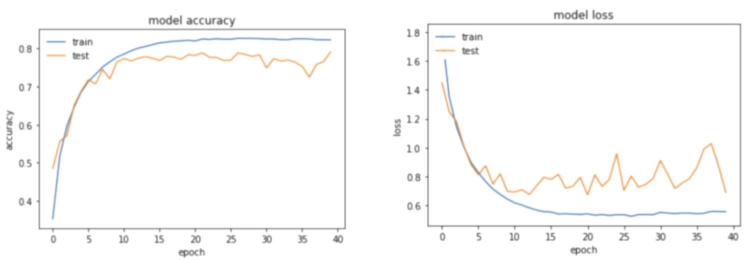

Figure 1: Model accuracy/loss result

4.6 Alternative/Future simulations

Considering all the experiment results mentioned in previous sections, the configuration of future simulations is as follows (with plenty of time to spare):

- Network: CNN, with the structure of conv+conv+maxpool+dropout+conv+conv+maxpool, with the classification structure set as dense+dropout+dense
- Activation function: Leaky Rectified Linear Unit Function applied in all the layers except the final layer (which is supposed to use Softmax Function for classification)
- Dropout Rate = 0.25
- Number of Epochs: 50
- Batch Size = 128
- Method to explore hyperparameters: Random Search, which will give better result in fewer iterations

References

Aly, A., Sascha, G. and Francesca, S. (2017). 'Metrics and benchmarks in human-robot interaction: Recent advances in cognitive robotics', *Cognitive Systems Research*, 43, pp. 313-323. Available at: https://hal.archives-ouvertes.fr/hal-01507342/file/article.pdf (Accessed: 20 April 2021).

Fung, P., et al. (2018). 'Towards empathetic human-robot interactions', International Conference on Intelligent Text Processing and Computational Linguistics. Springer, Cham. 21 March 2018. Lecture Notes in Computer Science. pp. 173-193. Available at: https://doi.org/10.1007/978-3-319-75487-1_14 (Accessed: 28 April 2021).

Pierson, H., A. and Michael, S. G. (2017). 'Deep learning in robotics: a review of recent research', Advanced Robotics, 31.16, pp. 821-835. Available at: https://www.tandfonline.com/doi/abs/10.1080/01691864.2017.1365009 (Accessed: 28 April 2021).

Sünderhauf, N., et al. (2018). 'The limits and potentials of deep learning for robotics', *The International Journal of Robotics Research*, 37.4-5, pp. 405-420. Available at: https://doi.org/10.1177/0278364918770733 (Accessed: 20 April 2021).