Network Architecture Search (NAS) for Facial Expression Recognition Networks

Project Plan

Chiu Chi Hang 3035475462
Chung Wai Kei 3035473488
Tse Chung Yin 3035474999

Supervisor: Dr K P Chan
1. Introduction
   1.1 Background
   1.2 Neural Architecture Search (NAS)
   1.3 Convolutional Neural Network (CNN)
   1.4 Reinforcement Learning (RL)

2. Objectives

3. Methodology
   3.1 Implementation of NAS
      3.1.1 Search space
      3.1.2 Search strategy
      3.1.3 Performance estimation strategy
      3.1.4 Training with REINFORCE
   3.2 Dataset chosen for FER
   3.3 Training condition

4. Project Schedule and Milestones

References
1. Introduction

1.1 Background

Deep learning has been used in many remarkable applications nowadays, for example, object recognition [1], natural language processing [2] and financial analysis [3]. One of the important factors which contributes to a successful deep learning model is its neural network. Neural networks are designed into different architectures with variations in terms of types of layers, number of layers, number of neurons and so on. In the sense of highly complex features of neural architectures, good neural networks are hard to build [4]. Thus, it is an inefficient process to manually design neural architectures although it is a common practice in the past few decades [5]. Therefore, experts then attempt with an automated approach, which is Network Architecture Search (NAS), to design the best neural network.

1.2 Neural Architecture Search (NAS)

NAS is the technique of automating architecture engineering [5]. Methods for NAS can be broken down into three dimensions, namely search space, search algorithm and performance estimation strategy:

- Search space: It defines the type of neural architecture that NAS approach will apply on, generally within reasonable space size.
- Search strategy: It defines the exploration approach in a predefined search space. The strategy is designed with the principle of finding outperforming candidates quickly whereas avoiding early convergence to any sub-optimal neural architectures [5].
- Performance estimation strategy: It defines the process of strategy evaluation. Scrupulous trade-offs between evaluation accuracy and computational cost have to be decided carefully to ensure good performance of NAS.

![Figure 1: Illustration of methods for NAS [5]](image)

Search Strategy performs selection in the predefined search space. An architecture would be searched and passed to performance estimation strategy, then the estimation of the architecture would be passed back to search strategy.
1.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN), one of the most suitable networks for dealing with image classification tasks due to its excellent performance and high-degree of automation in feature extraction [6], are chosen in many facial expression recognition (FER) applications [7], [8], [9]. Provoked by visual perception, CNN are designed with the characteristics of local connectivity, weight sharing and dimensionality reduction in downsampling so as to mimic the functionality of biological neural structure in receptor networks [10].

Figure 2: Operations of 2D CNN [10]

In this project, CNN will be chosen to perform facial expression recognition.

1.4 Reinforcement Learning (RL)

This project will implement NAS by reinforcement learning (RL). Reinforcement learning, together with supervised learning and unsupervised learning, are the basic machine learning strategies. In general situations, an agent interacts with an environment and collects feedback of the interaction, commonly known as rewards in which is determined by a policy evaluation algorithm. The rewards then inform changes to the policy that controls actions made by the agent. The below figure would illustrate the principle of a conventional RL setting:

Figure 3: Illustration of Reinforcement learning [11]

More complicated RL settings would be applied in different implementations. For the implementation of NAS by RL, further discussion will be made in Section 3.
2. Objectives

Implement NAS to find a neural architecture which produces a model with high accuracy in facial expression recognition under fixed training conditions. The NAS will potentially be deployed in mobile devices. The result model should be able to classify different facial expressions regardless of lighting and shooting angles.

3. Methodology

3.1 Implementation of NAS

NAS is used to find the best CNN architecture for classifying each dataset.

3.1.1 Search space

To reduce the time of computation, instead of automatically generating neural architecture with random parameters, the search space will be a set of famous CNNs in the past decade, including AlexNet, VGGNet and ResNet, with modifications in dropout and activation function. As the NAS may be applied in mobile devices, MobileNet and ShuffleNet may also be included in the search space.

3.1.2 Search strategy

The search strategy is reinforcement learning, which can be separated into two parts, namely controller and trainer. The controller is an agent that selects a CNN architecture from the search space. The trainer trains a model using the selected CNN. Then it takes the accuracy of classifying the validation set as reward and updates the controller. The controller will learn what CNN architecture it should choose in order to maximize the reward.

3.1.3 Performance estimation strategy

Performance evaluation principle will simply be based on the accuracy of classifying validation sets. But, training each architecture to completion is extremely expensive. Furthermore, there are no explicit standards for accuracy calculation in FER application. This further increases computational cost during training. Thus, lower fidelity estimates is proposed as our main strategy for performance estimation. In this approach, epochs are reduced, subset of data is used and data quality is downscaled in order to shorten training process [5].

3.1.4 Training with REINFORCE

The NAS process follows the method proposed in section 3 of paper by Google Brain [4]. We treat the prediction of the controller as a sequence of actions \( a_{1:T} \). After the controller picks a particular network, the trainer will train the model with training data, and get the accuracy using the test data. The accuracy at convergence will be the reward of the reinforcement learning of the network which is used to train the
controller. To get the best network architecture, we ask the controller to maximize the expected reward, which is represented by [4]:

\[ J(\theta_c) = E_{P(a_{1:T};\theta_c)}[R] \]

Since \( R \) is not differentiable, we use REINFORCE to iteratively update \( \theta_c \),

\[ \nabla_{\theta_c} J(\theta_c) = \sum_{t=1}^{T} E_{P(a_{1:T};\theta_c)} \left[ \nabla_{\theta_c} \log P(a_t|a_{(t-1):1}; \theta_c) R_k \right] \]


\[ \frac{1}{m} \sum_{k=1}^{m} \sum_{t=1}^{T} \nabla_{\theta_c} \log P(a_t|a_{(t-1):1}; \theta_c) R_k \]

where \( m \) is the number of different networks in one batch, \( T \) is the number of hyperparameters to be predicted. \( R_k \) is the validation accuracy of the \( k \)-th network. To reduce variance of the above unbiased estimator, baseline function \( b \) is added in the estimator.

\[ \frac{1}{m} \sum_{k=1}^{m} \sum_{t=1}^{T} \nabla_{\theta_c} \log P(a_t|a_{(t-1):1}; \theta_c) (R_k - b) \]

where \( b \) is the exponential moving average of the previous architecture accuracy. This is still an unbiased estimator as \( b \) is independent of the current action.

### 3.2 Dataset chosen for FER

Apart from the core elements of NAS, data is the most important element as most of the models are data-driven. For datasets, we use common FER datasets which are mentioned in a survey paper [12]. The survey paper includes 18 datasets for FER. There are datasets of 2D static images, 2D video sequences, 3D-based data, which are served for different purposes. Using 2D video sequences, models can learn recognizing facial expressions at different angles. With 3D data, models can further learn micro facial behaviours and distinguish between facial structures. Different datasets also collect data from different environments, some are collected in lab environment while some are collected in the wild. Using data in wild is more appropriate in this project as our application is a mobile application which is used by the public.
3.3 Training condition

The PC of one of the members will be used for major training purposes. The computer specifications are as follow:

- Operating System: Microsoft Window 10
- Processor: AMD Ryzen 7 3700X 8-Core Processor (3593 Mhz)
- Display Card: NVIDIA GeForce RTX 2070 SUPER
- VRAM: 8 GB
- RAM: 32 GB

The 2 GPU farms offered by the Department of Computer Science will also be used if the computational power of the above PC is insufficient.
## 4. Project Schedule and Milestones

<table>
<thead>
<tr>
<th>Date</th>
<th>Deliverables of Phase 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Oct 2020</td>
<td>(Inception)</td>
</tr>
<tr>
<td></td>
<td>• Detailed project plan</td>
</tr>
<tr>
<td></td>
<td>• Project web page</td>
</tr>
<tr>
<td>Oct - Nov 2020</td>
<td>Find datasets and merge data across different dataset by data pre-processing</td>
</tr>
<tr>
<td></td>
<td>Search for networks to be added to search space</td>
</tr>
<tr>
<td></td>
<td>Learn CNN, reinforcement learning and implementation of NAS</td>
</tr>
<tr>
<td>Nov - Dec 2020</td>
<td>Implement NAS to find the best model</td>
</tr>
<tr>
<td>Jan 2021</td>
<td>Preparation for presentation</td>
</tr>
<tr>
<td>11-15 Jan 2021</td>
<td>First presentation</td>
</tr>
<tr>
<td>24 Jan 2021</td>
<td>Deliverables of Phase 2</td>
</tr>
<tr>
<td></td>
<td>(Elaboration)</td>
</tr>
<tr>
<td></td>
<td>• Preliminary implementation</td>
</tr>
<tr>
<td></td>
<td>• Detailed interim report</td>
</tr>
<tr>
<td>Feb - Mar 2021</td>
<td>Develop a mobile or desktop facial expression recognition application which implement the network in classification tasks</td>
</tr>
<tr>
<td>Apr 2021</td>
<td>Preparation for presentation</td>
</tr>
<tr>
<td>18 Apr 2021</td>
<td>Deliverables of Phase 3</td>
</tr>
<tr>
<td></td>
<td>(Construction)</td>
</tr>
<tr>
<td></td>
<td>• Finalized tested implementation</td>
</tr>
<tr>
<td></td>
<td>• Final report</td>
</tr>
<tr>
<td>19-23 Apr 2021</td>
<td>Final presentation</td>
</tr>
<tr>
<td>Before 4 May 2021</td>
<td>Prepare banner for exhibition</td>
</tr>
<tr>
<td>4 May 2021</td>
<td>Project exhibition</td>
</tr>
</tbody>
</table>
References


