A Shortcut for Deep Learning: Transfer Learning

(1) Abstract. An executive summary of the research.

This research explores optimizing transfer learning for the CIFAR-100 dataset by experimenting with different techniques such as selecting pre-trained models, freezing layers, and resizing input data. The study shows that matching input dimensions and freezing layers can significantly enhance model performance. Comparing ResNet and EfficientNet, it highlights EfficientNet's superior efficiency and accuracy, providing insights for further improvements through fine-tuning.

(2) Introduction. Why are you conducting this research? (Business case/Problem Formulation)

Transfer learning can be highly beneficial for businesses as it accelerates the deployment of machine learning solutions by reusing pre-trained models tailored to specific industry needs, such as customer service, fraud detection, or predictive maintenance. This reduces the time and cost associated with data collection and model training, enabling businesses to quickly leverage advanced AI capabilities for improved decision-making and operational efficiency. Moreover, transfer learning allows companies to stay competitive by rapidly adapting to market changes and innovations, ultimately enhancing customer satisfaction and driving revenue growth through more intelligent and responsive services. This research aims to understand the various ways of optimizing transfer learning by experimenting with different data structures.

(3) Literature review. Who else has conducted research like this?

The article "CIFAR 100: Transfer Learning using EfficientNet" provides a practical example of the application of transfer learning in computer vision. The article introduces us to transfer learning with a simple example using EfficientNet – a family of convolutional neural networks developed by Google that achieves state-of-the-art accuracy while being significantly more efficient than previous models. It uses a compound scaling method, which uniformly scales the network's depth, width, and resolution to achieve better performance. EfficientNet models are known for their high accuracy and low computational cost, making them ideal for various machine learning tasks on resource-constrained devices. However, similar to most online articles, this article does not spend much time exploring the different architectures of transfer learning, which leads to this research's goal: gaining deep understanding of the implementation of transfer learning through experimentation. (Kulkarni, 2019)

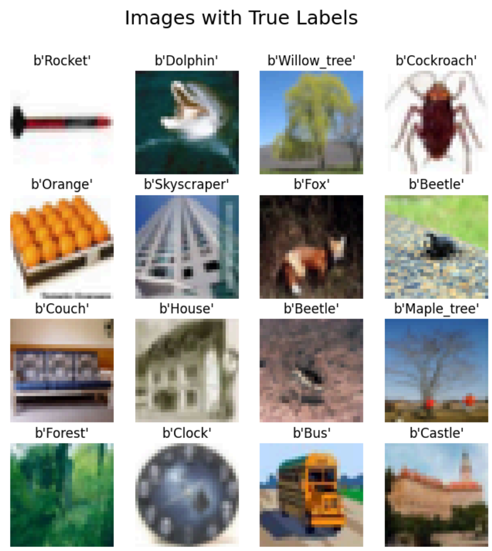
(4) Methods. How are you conducting the research?

The research mainly focuses on three aspects of transfer learning optimization methods: Choose the Right Pre-trained Model, Freeze and Unfreeze. Choosing the right pre-trained model for transfer learning is important because it determines the initial set of learned features and representations that the model can leverage. A suitable pre-trained model should have a similar domain or task to the target dataset to ensure relevant features are extracted effectively. Additionally, the computational complexity and resource requirements of the pre-trained model should align with the available resources for training and fine-tuning. Freezing in transfer learning refers to fixing the weights of certain layers in a pre-trained model during training on a new task or dataset. By freezing these layers, their weights are not updated during training, preserving the knowledge learned from the original task. This allows the model to focus on learning task-specific features from the new dataset while retaining the valuable features learned from the pre-training, which can help prevent overfitting and improve generalization.

There are 4 models trained for the research. The first two models had ResNet as the base model in order to compare the impact when the input sizes are different. The images in the original cifar-100 model have 32×32 pixels, but the ResNet requires the input shape to be 224×224. The resizing is achieved within the model by UpSampling2D, which is a layer in convolutional neural networks that increases the spatial dimensions (height and width) of the input data. The third model employs EfficientNet as the base model instead of ResNet because we would like to compare performance between the two commonly used base models for the cifar-100 dataset. ResNet (Residual Network) and EfficientNet are both state-of-the-art convolutional neural network architectures. ResNet is renowned for its deep structures with skip connections, allowing for easier optimization and training of extremely deep networks. In contrast, EfficientNet focuses on model efficiency by employing a compound scaling method that balances network depth, width, and resolution, resulting in superior performance with significantly fewer parameters and computational costs compared to traditional architectures like ResNet.

The models all have the same architecture for convenience, including the same optimization method, same number of layers, same number of neurons in each layer, and the same regularization methods, etc. On top of the base model, the added layers of every model follow the same structure: a global average pooling layer, a dropout layer, a dense layer with relu activation function, a batch normalization layer, and an output layer.

```
Image number selected : 34133
Shape of image : (32, 32, 3)
Image category number: 8
Image category name: b'Large_carnivores'
Image subcategory number: 88
Image subcategory name: b'Tiger'
/tmp/ipykernel_33/4074617551.py:14: FutureWarning:
is deprecated. In a future version, integer keys w
th DataFrame behavior). To access a value by posit
  print("Image category name: {}".format(category.
capitalize()))
/tmp/ipykernel_33/4074617551.py:16: FutureWarning:
is deprecated. In a future version, integer keys w
th DataFrame behavior). To access a value by posit
  print("Image subcategory name: {}".format(subCat
[0].capitalize()))
```



CIFAR-100 is a labeled subset of 80 million tiny images dataset where CIFAR stands for Canadian Institute For Advanced Research. The images were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The dataset consists of 60000 colored images (50000 training and 10000 test) of 32×32 pixels in 100 classes grouped into 20 superclasses. Each image has a fine label (class) and a coarse label (superclass). In this research, 30% of the training data is used as validation data, and the labels are one hot encoded. The data is also augmented with rotation, horizontal flip, and zoom. Data augmentation enhances computer vision models by artificially increasing the diversity and quantity of training data through transformations, which helps improve the model's robustness and generalization to new, unseen data.

(5) Results. What did you learn from the research?

Model	Train_Time	Train_acc	Val_acc	Train_loss	Val_loss	Test_acc
ResNet, original input	438.66	0.0695	0.0724	3.9854	0.0695	0.0697
ResNet, resized input	2538.42	0.8599	0.6329	0.4337	1.4864	0.6214
EffNet, resized input	2747.71	0.8836	0.6844	0.3578	1.2534	0.6899
EffNet, freeze	808.67	0.9467	0.7131	0.1640	1.3685	0.7153

The results of the research shows a clear path for more model improvement in the future. First of all, transforming the size of the dataset to match the base model proved to be an essential step of transfer learning as we can see that the model with the input of the original dataset performs poorly. Secondly, it is also important to choose the appropriate base model for the dataset, while in this research the new model only exhibits slight improvement in performance, in other cases it can make a huge difference. Last but not least, freezing the base model significantly improved the model performance by decreasing the training time and increasing the accuracy. The drop in training time can be explained by the decrease in trainable parameters, making the training process less complex (See Appendix). For the next step, we can fine tune the model which consists of unfreezing the entire model we obtained (model 4), and re-training it on the new data with a very low learning rate. This can potentially achieve meaningful improvements, by incrementally adapting the pretrained features to the new data.

(6) Conclusions.

This research underscores the efficacy of transfer learning in optimizing deep learning models for the CIFAR-100 dataset. Key findings highlight the importance of transforming input data to match the requirements of the base model and selecting an appropriate pre-trained model to achieve significant performance improvements. Additionally, freezing layers in the base model proved to be beneficial in enhancing model accuracy and reducing training time. Future work should focus on fine-tuning these models to further refine their performance, leveraging the pre-trained features with a low learning rate to adapt to the new dataset more effectively. This study confirms that strategic application of transfer learning techniques can significantly enhance the efficiency and effectiveness of deep learning models in practical applications.

Appendix:

Before freezing

```
Total params: 4,404,224
Trainable params: 4,361,696
Non-trainable params: 42,528
```

After freezing

```
Total params: 4,404,224
Trainable params: 354,148
Non-trainable params: 4,050,076
```

Reference:

Keras. 2020. “Keras Documentation: Transfer Learning & Fine-Tuning.” Keras.io. April 15, 2020. https://keras.io/guides/transfer_learning/.

Kulkarni, Balaji. 2019. “Transfer Learning Using ResNet.” Medium. October 17, 2019. https://balajikulkarni.medium.com/transfer-learning-using-resnet-e20598314427.