澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

CISC7026-002

Introduction to Deep Learning

Final Project Report

Group: S

Group members:

朱帅榕 ZHU SHUAIRONG	MC564533
苏文灿 SU WENCAN	MC566802
王佳仪 WANG JIAYI	MC564512
张誉慈 ZHANG YUCI	MC566850
陈子睿 CHEN ZIRUI	MC552582

Proposed Main Project: Wound Classification

Task Description

A wound classification system to categorize images of wounds on a part of the human body. First, there are several methods for classifying wounds, including etiology-based classification, CDC classification, and classification based on the depth of the wound and its healing status. Second, there exist numerous research papers related to the application of deep learning on wound classification, evaluated based on the dataset used. This ranges from applying YOLOv3, U-Net, VGG16, ResNet152, EfficientNet, and creating their own Deep CNN network with 7 layers of convolution.

We aim to try to implement JAX for medical imaging, specifically for image classification using CNNs. This is a comparison and deviation of the previous methods, which typically use PyTorch, Keras, or MATLAB. From similar studies, we can see that JAX has significantly faster training and inference time, and in a study from this year, they showed that on 28 by 28 pixel images, JAX slightly outperformed the other two libraries on blood cell classification in accuracy, macro/weighted precision, recall, and F1 score.

I/O Space

The input would optimally be an RGB picture taken on any smartphone, which would then be cleaned to be uniform in size and dimensions (square image). The range for similar studies is from 28 by 28 to 128 by 128[1] and 416 by 416[2], but we noticed that the requirements for hardware past 128 required extensive processing power; therefore, we plan to test sizes from 28 to 128 for a range of image sizes. We plan to first downscale the images to a smaller size, then perform image augmentation in the form of geometric (rotating, flipping (horizontal/vertical), translation, scaling, cropping) and photometric (RGB channel shifting, contrast adjustment, brightness adjustment) transformations. The output of this would be a vector of the different classification labels and their respective probabilities in the predicted label vector.

(Dataset shown is from Kaggle, which has been largely unexplored in the public community code space.)[3]

NN Architecture

For the architectural setup of our convolutional neural network, we decided to start with 4 convolutional blocks and increase/decrease depending on whether we see overfitting (training loss decreases while test loss stays the same/increases) or underfitting (loss stops decreasing but remains high). We plan to use ReLU as opposed to Leaky ReLU, Sigmoid, or Tanh, as it provides the majority of the benefits of ReLU (Preventing vanishing gradients) whilst maintaining the speed. This would be the activation function for

[1] https://arxiv.org/pdf/2408.11064
[2] https://ieeexplore.ieee.org/abstract/document/9785640
[3] https://www.kaggle.com/datasets/yasinpratomo/wound-dataset/data

the hidden layers, as opposed to the multi-class classification layer, which would use softmax. In addition to this, we also wish to apply dropout[4] and L2 regularization[5], both of which are meant to prevent overfitting on our limited dataset. The former by forcing redundancy in the neurons, and the latter to prevent large weights that don't generalize well.

Backup Project: SSM for Image Classification in JAX

Task Description

We plan to build an image classification model using state space models rather than the traditional only transformer or CNN-based architecture. Though there are pre-existing models that utilize SSMs like mamba in image classification[6], we want to see if implementing it in JAX would result in improvements in training times. Moreover, it would be an interesting learning and experimental process of implementing a model outside of transformers for image classification.

I/O Space

We would use the ImageNet-1K for the dataset, which contains pictures of size 224 by 224 in RGB. The output would be the 1000 classes that the image could be classified as, with softmax as the function that provides the probability distribution. (Vector of 1000 probabilities from 0 to 1)

NN Architecture

For the architectural setup of our State Space Model, we decided to start with a hybrid architecture comprising a lightweight, randomly-initialized CNN feature extractor followed by a single SSM block. We will adjust the complexity of both components, such as the depth of the CNN or the state dimension of the SSM, depending on whether we observe overfitting or underfitting.

We plan to use a structured state space model (e.g: S4, IDS4, Mamba) as opposed to a dense or diagonal parameterization, as it provides effective long-range modeling and computational efficiency, whilst maintaining training stability. This will serve as the global context for the model, as opposed to the CNN stage, which is dedicated to local feature extraction.

In addition, we also wish to apply dropout and weight decay regularization, both of which are meant to prevent overfitting on the complex ImageNet dataset. The former by randomly disabling units in the CNN feature extractor to force robust feature learning, and the latter to constrain the parameters of the SSM layer to prevent overly complex state dynamics that don't generalize well. Lastly, softmax will be used in the final layer to produce the probability distribution over the 1000 classes.

---

[4] https://arxiv.org/pdf/1506.02142
[5] https://arxiv.org/pdf/1205.2653
[6] https://arxiv.org/pdf/2407.08083