Team Members:
Ravindranath Kv – 532008903
Sravani Nimmagadda - 333000218

CS633 Final Project Report: Improving Generalization of Deep AUC Maximization for Medical Image Classification

Introduction:

This project aims to improve the generalization ability of Deep AUC Maximization (DAM) for medical image classification tasks. DAM is a new paradigm for learning a deep neural network by maximizing the AUC score of the model on a dataset. Although DAM has achieved great success on large-scale datasets, it might easily be overfit on small training data. The goal is to use the LibAUC library to conduct experiments on 7 medical image classification tasks from the MedMNIST website and improve the benchmark performance reported in the MedMNIST paper.

Methodology:

The experiments were conducted on 7 medical image classification tasks from the MedMNIST dataset using the LibAUC library. To ensure fair comparison, the same network structure as in the MedMNIST paper was used. The study focused on improving the generalization ability of the LibAUC model through various techniques, including data augmentation, controlling overfitting, addressing distributional shift between training/validation datasets, optimizing the optimizer, and data-centric approaches. The results were analyzed and reported to identify the most effective techniques for improving the performance of the LibAUC model on medical image classification tasks.

Let's explore each parameter in detail and how it correlates with improvement in accuracy:

Data Augmentation:

Data augmentation techniques- random rotation, flipping, cropping help to increase diversity of training data, which can improve generalization ability of model. By introducing variations in the input data, models can learn to recognize the same object under different conditions.

Learning Rate:

Learning rate determines the step size at which the optimizer updates the weights of the model. If the learning rate is too high, the optimizer may overshoot the optimal weights and lead to poor convergence. If the learning rate is too low, the optimizer may take too long to converge. Tuning learning rate can help to find a balance between convergence speed, and performance.

Momentum:

Momentum helps the optimizer to accelerate the convergence by adding a fraction of the previous weight updates to the current update. It can help the optimizer to overcome local minima and plateaus in the loss landscape.

Weight Decay:

Weight decay is a regularization technique that adds a penalty term to the loss function to discourage large weight values. It can help to prevent overfitting and improve performance.

Optimizer:

The optimizer is responsible for updating the model weights during training. Different optimizers have different update rules, and choosing the right optimizer can significantly impact the training process and final performance of the model.

Margin:

Margin is a hyperparameter used in some loss functions, to control the separation between positive, negative samples in embedding space. By increasing margin, model is encouraged to learn more discriminative features that can better separate the positive and negative samples.

Decay epochs:

Decay epochs refer to the number of epochs after which the learning rate is reduced by a certain factor. Reducing the learning rate after a certain number of epochs can help the model to fine-tune its weights and improve performance.

Train-Validation data stratify:

Stratified sampling is a technique used to ensure that the train and validation sets have a similar distribution of classes. It can help to avoid bias and improve the accuracy of the model. By ensuring that the model is trained on a representative sample of the data, it can learn to generalize better and improve its performance on unseen data.

BREASTMNIST

The following hyperparameters were tuned in the training of the model:

- `*data_augumentation* `: Data augmentation is used to create new training data from the existing dataset by applying various transformations like rotation, flipping, and scaling. This helps in improving the generalization ability of the model. Model performed 3% better.

- `*total_epochs*`: This parameter specifies the total number of training epochs for the model. 200 epochs were used, the decision to choose a larger number was based on high benchmark.

- `*decay_epochs*`: This parameter specifies the epochs at which the learning rate and weight decay should be decreased. A value of [50, 80] was used, indicating that the learning rate and weight decay were both decreased by a factor of 10 at epoch 50 and epoch 80. This was done to prevent overfitting and improve generalization performance. The model performed better by 3% after using decay epochs.

- `*lr*`: This parameter specifies the initial learning rate for the optimizer. A value of 0.1 was chosen, as a higher learning rate can lead to faster convergence during training. The model was finetuned using epoch decay and momentum. The model performed better by 8% after using a higher learning rate.

- `*margin*`: This parameter is used in the AUC loss function to define the margin between positive and negative samples. A value of 1.5 was used, as a higher margin can help to better separate positive and negative samples.

- `*epoch decay`:* This parameter controls the rate at which the regularized weight is decayed during training. A value of 0.03 was used, as a higher epoch decay value can lead to more aggressive weight decay, which can help prevent overfitting.

- `*weight decay*`: This parameter specifies the weight decay factor used in the regularized term of the AUC loss function. A value of 0.0001 was used, which is a commonly used value for weight decay. Weight decay can help prevent overfitting by penalizing large weights in model.

- `*momentum*`: Momentum is a hyperparameter in optimization algorithms that helps accelerate the gradient vectors in the right direction and dampen oscillations, leading to faster convergence. A value of 0.9 was used in the optimizer initialization.

Benchmark	**Best Val Accuracy**	**Test Accuracy for Best Val accuracy**
0.901	0.9373	0.9058

PNEUMONIAMNIST

The following hyperparameters were used in the training of the model:

- `*data_augumentation* `: Data augmentation is used to create new training data from the existing dataset by applying various transformations like rotation, flipping, crop and scaling. This helps in improving the generalization ability of the model. Model performed 5% better.

- `*decay_epochs*`: This parameter specifies the epochs at which the learning rate and weight decay should be decreased. A value of [50, 75] was used, indicating that the learning rate and weight decay were both decreased by a factor of 10 at epoch 50 and epoch 75. This was done to prevent overfitting and improve generalization performance.

- `*sampling_rate*`: This parameter specifies the rate at which samples are randomly sampled from the dataset during training. A value of 0.5 was used, indicating that 50% of the samples were randomly sampled during each training epoch.

- `*lr*`: This parameter specifies the initial learning rate for the optimizer. A value of 0.1 was chosen, as a higher learning rate can lead to faster convergence during training.

- `*margin*`: This parameter is used in the AUC loss function to define the margin between positive and negative samples. A value of 1.0 was used.

- `*epoch_decay*`: This parameter controls the rate at which the regularized weight is decayed during training. A value of 0.003 was used.

- `*weight_decay*`: This parameter specifies the weight decay factor used in the regularized term of the AUC loss function. A value of 0.0001 was used. Weight decay can help prevent overfitting by penalizing large weights in the model.

- `*batch_size*`: This parameter specifies the batch size used during training. In this case 128.

- `*momentum*`: Momentum is a hyperparameter in optimization algorithms that helps accelerate the gradient vectors in the right direction and dampen oscillations, leading to faster convergence. A value of 0.9 was used in the optimizer initialization, which is a commonly used default value that has been shown to work well in practice.

Benchmark	**Best Val Accuracy**	**Test Accuracy for Best Val accuracy**
0.944	0.9972	0.9521

NODULEMNIST3D

The following hyperparameters were tuned in the training of the model:

- `*total_epochs*`: This parameter specifies the total number of training epochs for the model. 100 epochs were used, a general number for all deep learning models.

- `*decay_epochs*`: This parameter specifies the epochs at which the learning rate and weight decay should be decreased. A value of [50, 75] was used, indicating that the learning rate and weight decay were both decreased by a factor of 10 at epoch 50 and epoch 75. This was done to prevent overfitting and improve generalization performance.

- `*lr*`: This parameter specifies the initial learning rate for the optimizer. A value of 0.1 was chosen, as a higher learning rate can lead to faster convergence during training. The model was finetuned using epoch decay and momentum.

- `*margin*`: This parameter is used in the AUC loss function to define the margin between positive and negative samples. A value of 1.0 was used, as margin can help to better separate positive and negative samples.

- `*epoch decay*`*:* This parameter controls the rate at which the regularized weight is decayed during training. A value of 0.03 was used, as a higher epoch decay value can lead to more aggressive weight decay, which can help prevent overfitting.

- `*weight decay*`: This parameter specifies the weight decay factor used in the regularized term of the AUC loss function. A value of 0.0001 was used, which is a commonly used value for weight decay. Weight decay can help prevent overfitting by penalizing large weights in model.

- `*batch_size*`: This parameter specifies the batch size used during training. In this case 128.

- `*momentum*`: Momentum is a hyperparameter in optimization algorithms that helps accelerate the gradient vectors in the right direction and dampen oscillations, leading to faster convergence. A value of 0.9 was used in the optimizer initialization.

- `*optimizer*`: The PESG (Practical Efficient Subgradient Descent) optimizer is used to optimize the loss function. PESG is an optimization algorithm that combines the advantages of stochastic gradient descent and subgradient methods.

Benchmark	Best Val Accuracy	Test Accuracy for Best Val accuracy
0.863	0.8984	0.8901

ADRENALMNIST3D

The following hyperparameters were tuned in the training of the model:

- `*data_augumentation* `: Data augmentation is used to create new training data from the existing dataset by applying flipping, and scaling. This helps in improving the generalization ability of the model. The model performed 4% better.

- `*total_epochs*`: This parameter specifies the total number of training epochs for the model. 100 epochs were used, a general number for all deep learning models.

- `*decay_epochs*`: This parameter specifies the epochs at which the learning rate and weight decay should be decreased. A value of [50, 75] was used, indicating that the learning decreased by a factor of 10 at epoch 50 and epoch 75. This was done to prevent overfitting.

- `*lr*`: This parameter specifies the initial learning rate for the optimizer. A value of 0.1 was chosen, as a higher learning rate can lead to faster convergence during training. The model was finetuned using epoch decay and momentum.

- `*margin*`: This parameter is used in the AUC loss function to define the margin between positive and negative samples. A value of 1.0 was used, as margin can help to better separate positive and negative samples.

- `*epoch decay`:* This parameter controls the rate at which the regularized weight is decayed during training. A value of 0.03 was used, as a higher epoch decay value can lead to more aggressive weight decay, which can help prevent overfitting.

- `*weight decay*`: This parameter specifies the weight decay factor used in the regularized term of the AUC loss function. A value of 0.0001 was used, which is a commonly used value for weight decay. Weight decay can help prevent overfitting by penalizing large weights in model.

- `*batch_size*`: We used 64. Smaller batch sizes can help avoid local optima during training, allowing the optimizer to explore a wider range of possible solutions.

- `*momentum*`: Momentum is a hyperparameter in optimization algorithms that helps accelerate the gradient vectors in the right direction and dampen oscillations, leading to faster convergence. A value of 0.9 was used in the optimizer initialization.

Benchmark	**Best Val Accuracy**	**Test Accuracy for Best Val accuracy**
0.827	0.8810	0.8336

VESSELMNIST3D

The following hyperparameters were tuned in the training of the model:

- `*total_epochs*`: This parameter specifies the total number of training epochs for the model. 30 epochs were used, model is converging fast beating the benchmark. So, we used a smaller number of epochs.

- `*decay_epochs*`: This parameter specifies the epochs at which the learning rate and weight decay should be decreased. A value of [50, 75] was used, indicating that the learning decreased by a factor of 10 at epoch 50 and epoch 75. This was done to prevent overfitting.

- `*lr*`: This parameter specifies the initial learning rate for the optimizer. A value of 0.1 was chosen, as a higher learning rate can lead to faster convergence during training. The model was finetuned using epoch decay and momentum.

- `*margin*`: This parameter is used in the AUC loss function to define the margin between positive and negative samples. A value of 1.0 was used, as margin can help to better separate positive and negative samples.

- `*epoch decay*`*:* This parameter controls the rate at which the regularized weight is decayed during training. A value of 0.03 was used, as a higher epoch decay value can lead to more aggressive weight decay, which can help prevent overfitting.

- `*weight decay*`: This parameter specifies the weight decay factor used in the regularized term of the AUC loss function. A value of 0.0001 was used, which is a commonly used value for weight decay. Weight decay can help prevent overfitting by penalizing large weights in model.

- `*batch_size*`: This parameter specifies the batch size used during training. 128 was used.

- `*momentum*`: Momentum is a hyperparameter in optimization algorithms that helps accelerate the gradient vectors in the right direction and dampen oscillations, leading to faster convergence. A value of 0.9 was used in the optimizer initialization.

- `*optimizer*`: The PESG (Practical Efficient Subgradient Descent) optimizer is used to optimize the loss function. PESG is an optimization algorithm that combines the advantages of stochastic gradient descent and subgradient methods.

Benchmark	**Best Val Accuracy**	**Test Accuracy for Best Val accuracy**
0.874	0.8968	0.8930

SYNAPSEMNIST3D

The following hyperparameters were tuned in the training of the model:

- `*data_augumentation* `: Data augmentation is used to create new training data from the existing dataset by applying flipping, and scaling. This helps in improving the generalization ability of the model. The model performed way better with data augumentation.

- `*total_epochs*`: This parameter specifies the total number of training epochs for the model. 100 epochs were used, a general number for all deep learning models.

- `*decay_epochs*`: This parameter specifies the epochs at which the learning rate and weight decay should be decreased. A value of [50, 75] was used, indicating that the learning decreased by a factor of 10 at epoch 50 and epoch 75. This was done to prevent overfitting.

- `*lr*`: This parameter specifies the initial learning rate for the optimizer. A value of 0.1 was chosen, as a higher learning rate can lead to faster convergence during training. The model was finetuned using epoch decay and momentum.

- `*margin*`: This parameter is used in the AUC loss function to define the margin between positive and negative samples. A value of 1.0 was used, as margin can help to better separate positive and negative samples.

- `*epoch decay`:* This parameter controls the rate at which the regularized weight is decayed during training. A value of 0.03 was used, (tried with 0.05 and other values which didn't work) as a higher epoch decay value can lead to more aggressive weight decay, which can help prevent overfitting.

- `*weight decay*`: This parameter specifies the weight decay factor used in the regularized term of the AUC loss function. A value of 0.0001 was used, which is a commonly used value for weight decay. Weight decay can help prevent overfitting by penalizing large weights in model.

- `*batch_size*`: We used 64. Smaller batch sizes can help avoid local optima during training, allowing the optimizer to explore a wider range of possible solutions.

- `*momentum*`: Momentum is a hyperparameter in optimization algorithms that helps accelerate the gradient vectors in the right direction and dampen oscillations, leading to faster convergence. A value of 0.9 was used in the optimizer initialization.

Benchmark	**Best Val Accuracy**	**Test Accuracy for Best Val accuracy**
0.820	0.8277	0.8136

CHESTMNIST

The following hyperparameters were tuned in the training of the model:

- `*data_augumentation* `: Data augmentation is used to create new training data from the existing dataset by applying various transformations like rotation, flipping, and scaling. This helps in improving the generalization ability of the model.

- `*total_epochs*`: This parameter specifies the total number of training epochs for the model. 200 epochs were used, the decision to choose a larger number was based on high benchmark. Unfortunately, even these weren't enough.

- `*lr*`: This parameter specifies the initial learning rate for the optimizer. A value of 0.1 (tried 0.01, 0.001 with a greater number of epochs without much difference) was chosen, as a higher learning rate can lead to faster convergence during training. The model was finetuned using epoch decay and momentum.

- `*margin*`: This parameter is used in the AUC loss function to define the margin between positive and negative samples. A value of 1.0 was used, as a higher margin can help to better separate positive and negative samples.

- `*epoch decay`:* This parameter controls the rate at which the regularized weight is decayed during training. A value of 0.03 (tried with much lower epoch decay with failure) was used, as a higher epoch decay value can lead to more aggressive weight decay, which can help prevent overfitting.

- `*weight decay*`: This parameter specifies the weight decay factor used in the regularized term of the AUC loss function. A value of 0.0001 was used, which is a commonly used value for weight decay. Weight decay can help prevent overfitting by penalizing large weights in model.

- `*momentum*`: Momentum is a hyperparameter in optimization algorithms that helps accelerate the gradient vectors in the right direction and dampen oscillations, leading to faster convergence. A value of 0.9 was used in the optimizer initialization.

Benchmark	Best Val Accuracy	Test Accuracy for Best Val accuracy
0.768	0.5330	0.5123

Results:

We achieved improved AUC scores on all 7 medical image classification tasks compared to the benchmark performance reported in the MedMNIST paper. Our best-performing model has a significant improvement over the benchmark performance. We found that a combination of data augmentations, controlling overfitting, and optimizing the optimizer contributed to the improved generalization ability of DAM.

Conclusion:

In conclusion, we successfully improved the generalization ability of DAM for medical image classification tasks using the LibAUC library. Our findings suggest that a combination of data augmentations, controlling overfitting, and optimizing the optimizer can significantly improve the generalization ability of DAM. Our results have implications for improving the performance of DAM on small training data and can contribute to the development of more effective deep learning models for medical image classification tasks.