Heidelberg University
Institute of Computer Engineering
Computing Systems Group

Holger Fröning
Hendrik Borras

Embedded Machine Learning
Summer Term 2023

Exercise 5

- **Return electronically until Wednesday, June 14, 2023, 09:00**
- **Include your names on the top sheet. Hand in only one PDF.**
- **A maximum of nine students are allowed to work jointly on the exercises.**
- **When you include plots add a short explanation of what you expected and observed.**
- **Hand in source code if the exercise required programming. You can bundle the source code along with the PDF in a .zip file.**
- **Programming exercises can only be graded if they run on the cluster in the provided conda enviroment. Make sure to document additional steps, which you might have taken to run the exercises.**

5.1 Residual neural network architectures

Residual neural networks have seen great success in the field of machine learning, in particular for image recognition.
In this exercise we will implement a derivative of ResNet, for which you can find more details in the original paper[1].

- Using the template implement a scaled down version of ResNet-18. In particular the ResNet basic block should be implemented as shown in the lecture.
- Train the ResNet for 50 epochs. Make sure to not use any data augmentation to only measure the training time of the network.
- Compare the test accuracy of the ResNet to the VGG11 from the previous exercise and the ConvNet from exercise 3. Note that you can reuse results from the previous exercises here. For all three networks plot the test accuracy over the number of epochs and over the training time.
- For all three networks calculate the number of parameters and MAC operations.
- Discuss how the number of parameters and MAC operations impacts the previous plots and how the different architecture design has impacted these metrics.

(20 points)

5.2 Activation normalization techniques and experiment tracking

The goals of this part of the exercise are two fold. On one side we want to take a quick look at the BatchNorm and on the other side we want to make you aware of tools, which you might find useful for the upcoming project work.

- Implement the BatchNorm and one other activation normalization technique (Group normalization, layer normalization or instance normalization) for the ResNet. Place the normalization layer in-front of each activation layer. Note: Make sure to use the 2d implementation of the normalization layer. From an implementation standpoint you may be happy to know that the GroupNorm becomes the LayerNorm for $G = 1$.
- For the next part of this exercise you'll need to choose one of the many publicly available ML tracking services. A likely incomplete list is given below. In our own research we have used Weights and Biases and Neptune with good results. You are free to choose which ever service speaks to you most, even if it's not on this list.

[1] `https://arxiv.org/pdf/1512.03385.pdf`

  - Weights and Biases: `https://wandb.ai/site`
  - Neptune: `https://neptune.ai/`
  - DVC: `https://dvc.org/`
  - MLflow: `https://mlflow.org/`
  - AimStack: `https://aimstack.io/`
  - GuildAI: `https://guild.ai/`
  - Comet: `https://www.comet.com/site/`

- Using the chosen experiment tracking service train the ResNet for 50 epochs, once without any activation normalization, once with BatchNorm and once with the activation normalization technique of your choice. Not that the observable differences in accuracy may be very small here.
- Using the experiment tracking service plot an accuracy comparison between the three training runs. Attach your result to the PDF as a screenshot.
- Discuss the results of the previously obtained plot.
- Additionally describe your experience with the chosen experiment tracking service. What worked well, what didn't work well? Did it make work easier or more cumbersome? Which features were you missing? In the tutorial we will likely have a short discussion on each of the services used, based on your findings.

(20 points)

5.3 Willingness to present

Please declare whether you are willing to present any of the exercises from this sheet. The declaration should happen on the PDF which you hand in.
The declaration can be made on a per-exercise basis. Each declaration corresponds to 50% of the exercise points. You can only declare your willingness to present exercises for which you also hand in a solution. If no willingness to present is declared you may still be required to present an exercise for which your group has handed in a solution. This may happen if as example nobody else has declared their willingness to present.

- Residual neural networks (Section: 5.1)
- Activation normalization techniques and experiment tracking (Section: 5.2)

(20 points)

Total: 60 points