Project Proposal: Explore training ConvNets from scratch for image classification

Group 38

Natan Teferi Asegehegn
ntas@kth.se

Amar Hodzic
amarh@kth.se

Djiar Salim
djiar@kth.se

May 4, 2021

Project description - B level

We are interested in one of the basic projects mentioned in the instructions; *Explore training ConvNets from scratch for image classification.* The reason is that we want to get more in-depth knowledge of CNN's. This would also allow us to experiment with the architecture and explore the benefits/drawback of data-augmentation, dropout among other things. In particular we are interested in investigating the conv-net AlexNet since it is one of the most popular models for computer vision tasks that reached top 5 placement in the ILSVRC-2012 competition [1].

Milestone 1

Implement a bug-free version of well-known conv-net such as AlexNet. The Adam [2] optimizer will be used, with default weights initialization provided by the framework (e.g. He-initialization). Train it on CIFAR-10 dataset and report the results of the accuracy and loss.
Get the convolution network working and plot accuracy and loss without any optimization, normalization and tuning.

Milestone 2

Investigate the effect of batch normalization. Use accuracy, loss and computational cost for evaluation and include the hardware and software used when conducting the experiments.

Document and plot the results.

Milestone 3

Investigate the effect of adding dropout to the network. Apply different degrees of dropout and see what effect they have on the accuracy and loss.

Document and plot the results.

Milestone 4

Investigate the effect of applying different types of data augmentation to the network (e.g. at each mini-batch). Report its effect on the accuracy, loss and computational cost.

Document and plot the results.

Milestone 5 (Optional)

Train the network on the CIFAR-100 dataset instead and measure results. Visualize the learnt representation of the model after each milestone.

Training data

We plan on using the Cifar-10 dataset [3] and possibly the CIFAR-100 for the last and optional milestone if time allows.

Software packages and libraries

We will either be using PyTorch, Tensorflow or Keras. Most likely Keras as we heard it has an easier and more beginner friendly API.

What we will write ourselves

We will write the code for the network with the help of one of the mentioned packages. We will write the functions for testing the network across different values as well as the plotting.

Experiments

For each of the milestones, we want to measure the effect on accuracy and loss. We would also like to measure the computational cost in relation to accuracy when doing batch normalization and data augmentation. Since it is interesting to know how much more computation is needed to achieve possibly small gains in accuracy.

What will count as success

Grade E: Milestone 1

Grade D: Milestone 2

Grade C: Milestone 3

Grade B: Milestone 4

In summary, we hope to gain more in-depth knowledge of the effects of these parameters and techniques on accuracy and computational effort of convolutional neural networks.

Learning aims for group members

Amar

I hope to learn more about convolutional networks and the challenges when implementing them. How they can be improved and what the trade-offs are for different approaches.

Djiar

My goal after this project, is to be able to create my own networks with better knowledge of the architecture and the impact of parameters/optimizations in future projects.

Natan

My aim is to get a deeper understanding of convolutional networks and why they have had such good success in image classification. With our experiments, I hope to understand the impact of dropout, batch normalization and data augmentation has on a famous architecture such as AlexNet.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[3] "CIFAR-10 and CIFAR-100." [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html