Week 7: Training ANNs on MSINT Handwritten Images

Background and Motivation

This analysis was performed on the MNIST dataset, made available via the "Digit Recognizer - Learn Computer Vision Fundamentals with the Famous MNIST Data" Kaggle competition (Kaggle 2022). The dataset contains 60,000 handwritten numbers, represented as a 28x28 pixel image with each pixel's level of shading captured. The "training" dataset is enriched with the number captured in the image. Successful work with this dataset allows for the application of computer vision to automate scanning of handwritten numbers for, e.g., postal mail or bank check details..

Approaches to Pre-Processing the Data

The original dataset contained 784 pixels, scored from 0-255 representing the depth of shading. Since all metrics were scaled identically, no further processing was performed. A 20% holdout was used for validation, and the 80% training set was enriched through image manipulation. Each image in the training set was permuted three times in three ways, for a 10x multiplier of training images:

- Permutation 1: random rotation of the image
- Permutation 2: random zoom in/out on the image
- Permutation 3: zooming and rotation.

Modeling Approach

The focus this week was on developing artificial neural networks, with a focus on comparing the impact of changing the number of hidden layers and the number of nodes per layer. Six network structures were tested, with models 1-4 being a 2x2 completely crossed design of 2-3 hidden layers and 100 or 300 hidden nodes per layer; models 5 and 6 tested maintained 100 nodes per layer, changing the number of layers to 1 and 5. All networks relied on a ReLU activation function, default learning rate, and a Dropout rate of 0.2.

Results:

	Layers	Nodes / Layer	Time per epoch (sec)	Epochs	Total Time (sec)	Accuracy (Val)	Accuracy (Test)
Model 1	2	300	33	26	858	97.75%	97.55%
Model 2	3	300	42	19	798	97.69%	97.64%
Model 3	2	100	22	33	726	97.52%	97.37%
Model 4	3	100	24	20	480	97.19%	97.18%
Model 5	1	100	18	50	900	97.45%	97.37%
Model 6	5	100	22	15	330	96.77%	96.61%

Implications

- More complex models were not necessarily better. Models 1 & 2, with 300 nodes per layer and 2 or 3 layers, performed similarly, even though the 3-layer model was exponentially more complex. Model 4, a 3x100 model, underperformed Model 3, a 2x100 model, by a noticeable margin
- Each model was early-stopped with a patience of "10", which implies that the model epoch with lowest validation loss was 10 iterations prior. Since the learning rate was consistent, this metric implies the speed a model overfit. The more complicated models (Models 2 & 6) overfit the fastest, whereas the simplest (Model 5) overfit the slowest. This is unsurprising, seeing as the more complicated models had substantially more weights and neared 100% training accuracy quickly
- These models were developed using Google Colab, relying on CPUs for model generation. The first four results were re-run using GPUs with epoch speeds in the 12-14 second range, which is a substantial acceleration of model results. Size of network has a noticeable impact on runtime when using CPUs, but essentially vanishes when using GPUs. This emphasizes the power of the back-propagation algorithm and the efficiency of using matrix mathematics.

References

"Digit Recognizer - Learn computer vision fundamentals with the famous MNIST data." Kaggle. Accessed February 7, 2022. https://www.kaggle.com/c/digit-recognizer/overview.