Assignment 3: CNNs and RNNs

Welcome to your third assignment. Upload the completed scripts task1.py and task2.py to the VLE before the deadline.

Standard criteria for valid assignments

In general, an assignment is considered valid if the following criteria are met:

- The delivered program uses Python and Tensorflow.
- The delivered program code is organised and easy to understand.
- Only one code file is used per task, which is a modified version of the provided scripts, with the same file names.
- The delivered program does not use any random number generator seeds. When correcting, the program will be run 5 times to get an average error.
- When the delivered program is run, the model is trained from scratch using gradient descent (or an extension of gradient descent).
- The way the final test error is calculated is left as is.
- Only the provided dataset is used and without modification (although the data may be transformed for use by the model).
- The test set does not influence training in any way.
- The delivered program code is not excessively modified. I will be correcting your program by checking what was changed from the original provided scripts and needless modification makes corrections slower.
- Make sure that when ‘full_output’ and ‘show_analysis’ are set to false, the only thing that is shown after running the program are the error, model size, and execution time, exactly as coded in the provided script files.
- Your aim should be to deliver the fastest, smallest, and best performing (according to test error) program possible within the given task constraints.

Task 1: Convolutional Neural Networks (50%)

Summary: Learn a convolutional neural network that classifies MNIST handwritten digits.

You are provided with a file "dataset.txt" that contains a collection of flat bitmap images of hand written digits together with a half-finished Python script "task1.py". Your task is to complete the script in order to learn a convolutional neural network using gradient descent with Tensorflow which accurately regenerates the given digits in the dataset. You may split the dataset into separate validation sets if needed and you may use any technique discussed during the lectures.

The script also includes code to perform sensitivity analysis performed on a test image for each digit. In the corner is the digit that the image is classified as by the neural network. The red blobs show parts of the image that when changed will be the most to change the classification.

Note that it might help to transform the pixel values so that instead of being ones and zeros they are ones and negative ones. It might also be necessary to use one of the advanced optimisers for this task rather than simple gradient descent.

Constraints

This task in particular is considered valid if the following criteria are met:

- The model uses at least one convolution layer.
- The average final test error (of 5 runs) is less than or equal to 12% (high performance error: 9%).
- The number of parameters is less than or equal to 1,000,000.
- The average running time on CPU is less than 5 minutes.

Task 2: Recurrent Neural Networks (50%)

Summary: Learn a recurrent neural network that classifies MNIST handwritten digits.

Like the previous task but with recurrent layers instead of convolution layers. The script's name is "task2.py".

Note that it might be necessary to use one of the advanced optimisers for this task rather than simple gradient descent.

Constraints

This task in particular is considered valid if the following criteria are met:

- The model uses at least one recurrent layer.
- The average final test error (of 5 runs) is less than or equal to 12% (high performance error: 9%).
- The number of parameters is less than or equal to 100,000.
- The average running time on CPU is less than 5 minutes.