Assignment 3: CNNs and RNNs

Welcome to your first assignment. Upload the completed scripts task1.py and task2.py and the document assignment3.pdf 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.
- Only one code file is used per task, which is a modified version of the provided script, with the same file names. Apart from the two code files, you should also provide a PDF with screenshots of the charts shown in each task.
- The way the final test error is calculated in the provided script is left as is.
- Only the provided dataset is used and without modification (although the data may be transformed for use by the model).
- Your program may not run for longer than 10 minutes on CPU.
- The delivered program code is organised and easy to understand.
- The delivered program does not use any random number generator seeds.
- When the delivered program is run, the model is trained from scratch using gradient descent (or an extension of gradient descent).
- 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.
- At the end of training you will be shown a confusion matrix together with the test accuracy, duration, and number of parameters in your model. Your aim should be to deliver the fastest, smallest, and best performing program possible with the given task constraints.

You will get marks for:

- Neatness.
- Using correct implementations of techniques shown in class.
- Use of sensible hyperparameters.
- Use of regularisation.
- The model's general performance.
- Any interesting ideas you include (for bonus marks).

Task 1: Convolutional neural networks

Summary: Learn a convolutional neural network that correctly 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 classifies 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 on one of the test images for each digit. The red blobs show the parts of the image that are most likely to change the model's classification. At the bottom of each digit is the digit that the image is classified as by the neural network.

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 output of the model is a vector of 10 numbers such that the number at position 0 quantifies how likely the given digit is a 0, the number at position 1 quantifies how likely the given digit is a 1, etc.

Task 2: Recurrent neural networks

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

As the previous task but with recurrent layers instead of convolutional. Script name is "task2.py". You may use the set of rows or columns of the image as a sequence of vectors.

Constraints

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

- The model uses at least one recurrent layer.
- The output of the model is a vector of 10 numbers such that the number at position 0 quantifies how likely the given digit is a 0, the number at position 1 quantifies how likely the given digit is a 1, etc.