Homework set - Neural networks - Part 1

March 27, 2020

Guidelines

- For the use of this assignment we use the MNIST dataset from the following link.
- The solution for this homework is to be posted as a .pdf file.
- You may choose the programming language you prefer for the implementations (excluding packages that enable auto-differentiation, e.g tensorflow, keras etc.).
- It is highly recommended to use object-oriented programming for this assignment.
- All plots must have named axis, grids and title. If more than one plot is on the same figure, provide legend.

1 Self-Reading

Read and solve the exercises in chapter 1 at Michael Neilsen e-book at the following link.

2 Multi-layered network implementation

In this section we implement gradually a multi-layered network with fully connected layers.

- Build a MNIST data reader object:
  1. constructor - pre-process the data and divide it into train (first 55K examples), validation (subsequent 5K examples) and test (10K examples).
  2. method:get_batch - return a batch of examples of train/valid/test.
  3. method:shuffle_train - shuffle the training set along the batch dimension.

  – Test your reader:
    * Visualize 10 randomly chosen examples with their label.
    * Why should we implement shuffle_train()?
- build model object
  1. constructor - define model weights, hyper-parameters and general configuration.
     – Here you should initiate the model weights. Think how to do that properly.
  2. method:feed_forward - input a batch of examples and return the loss over this batch and the model's output.
  3. method:loss_mse - input model's output and true labels to calculate the mean squared error (MSE).
  4. method:loss_nll - input model's output and true labels to calculate the negative log loss (nll).

  - Show that the negative log loss over a batch (of i.i.d examples) of size n goes to the cross entropy between $P_{Y|X}$ and $Q_{Y|X}$ (the true conditional distribution and the neural network model respectively) as $n \to \infty$.

- Test your model:
  * Feed a random batch to the model. What should be the model output? the negative log loss? does reality match expectation?

3 Self-Reading

Read and solve the exercises in chapter 2 at Michael Neilsen e-book at the following link.

4 Multi-layered network implementation - Cont.

In this section we continue to implement the multi-layered network with fully connected layers.

- build model object - Cont.

  1. method:back_prop input a batch of examples and return the loss over this batch and the gradient vector w.r.t the batch.

  - Test your back_prop:
    * Perform a numerical gradient check and compare it to your back_prop function. You may use the following link as a guide.
    * Please attach the average relative error and max relative error over randomly selected subset (say 10%) of the model parameters. For clarity, this means that you should compute the gradient w.r.t 10% of the model parameters with the backprop function and the numerical approximation, and compare them elementwisely. The relative error between two scalars x, y is defined by

$$\text{relative\_error} \triangleq= \frac{|x-y|}{|x+y|}$$

- build SGD optimizer object.

  1. constructor - define optimizer hyper-parameters and general configuration.
  2. method:step input a gradient vector and parameters vector and perform a training step over the parameters.

- build another optimizer object (Momentum, RMSprop, Adam, AdaDelta etc.). You can use the following link for further details.

  1. constructor - define optimizer hyper-parameters and general configuration.
  2. method:step input a gradient vector and parameters vector and perform a training step over the parameters.

- build predict function

  1. Input the current model and data and return the accuracy over the data.

- build train_epoch function

  1. Input the current model and data and perform a training epoch, that is, update the model parameters w.r.t all the training data.

5 End-to-end MNIST Classification

Combine the product of the previous section to build a model to classify the MNIST digits. The classification should be done with the following constraints/documentation:

1. Parameters budget: 5M parameters.
2. Total training time: 1 hour.
3. All training should be done with the train and validation datasets only.
4. Every epoch document:
   - negative log loss of train and validation.
   - accuracy over train and validation datasets.
   - Average, maximum and minimum gradient norm in the epoch.
   - Epoch elapsed time.

Submission

The work in this assignment should be summarized into a pdf document, where you present the exercises from Michael Neilsen book and the exercises/visualization/documentation from the sections 2,4. In addition, summarize the work in section 5 by the following structure:

1. Model description: the network specification (#of parameters, network architecture, chosen optimizer etc.).
2. Plot: training and validation learning curve (nll w.r.t epoch).
3. Plot: training and validation accuracy curve.
4. Plot: maximum, minimum and average gradient norm curve.
5. Training time.

GOOD LUCK!!!