GPU Programming for Deep Learning
Goran Frehse, ENSTA Paris

Lab Exercise : Matrix Operations

Submit your notebook

to goran.frehse@ensta-paris.fr

Put **“IA307” and your name** at

1) the **beginning of the notebook**,
2) the **start of the filename** and
3) in the **subject line** your **email**!

For implementing the deep learning algorithms later, we need a few basic operations for our fmatrix structure. The goal of this lab is to **code** and **test** these operations. Start by reading and understanding the notebook here. Make a copy of the notebook and add your own code to it.

For each of the following functions, write the code so that it gets executed **on the GPU** and also write a **test program** for it (ideally comparing it to Numpy). Add your answers to the notebook in text cells. Use a separate code cell for each program.

Step 1: Simple Softmax

1. Write a function softmax(Z), where each column z_k of Z gets exponentiated (in-place) and then normalized by the column sum:
   $$z_k := \exp(z_k)/\Sigma_t \exp(z_{k,t})$$
2. What is the range of possible values for the resulting elements of Z, if they are computed perfectly (over the reals, not floating point)?
3. Test softmax(Z) on matrices where all elements have
   a. large positive values
   b. large negative values

   Try for yourself to predict what problems could occur, then check it experimentally.

Step 2: Stable Softmax

The stable softmax is identical to softmax over the real numbers, but is more stable in floating point since it avoids overflow and improves the precision. For each column z_k of Z, it computes

$z_{k,max} := \max_t z_{k,t}$
$z_k := \exp(z_k - z_{k,max}) / \boldsymbol{\Sigma}_t \exp(z_{k,t} - z_{k,max})$

1. Implement the stable softmax and run the tests from above.
2. Which problems get resolved?

Step 3: Normalization

Add code to normalize matrices X and Xtest for the test and training set on the GPU. Use Z-score normalization, which replaces each input feature value x_i with

$$x_i' = \frac{1}{\sigma}\left(x_i - \mu\right)$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \text{ where } \mu = \frac{1}{N}\sum_{i=1}^{N} x_i .$$

We will a matrix X where each data point x corresponds to a column and each feature to a row. Therefore, the normalization above needs to be carried out *over each row*. Ideally, you should normalize the test set X test with the μ and σ that you computed on the training data X. This means you need two functions, one to compute μ and σ and one to apply the normalization for given values of μ and σ.

1. Write a function to compute the vectors μ and σ for a matrix X. Every element of μ and σ corresponds to one row of X.
2. Write a function that, given X, μ and σ, normalizes X.
3. Test both functions and check that you can normalize a second matrix given μ and σ from another.