Applications

Exercices d'application

Exercice 1 – Somme d'entiers sur des formats limités

Commencer votre script par `import numpy as np`. Pour la suite on va travailler sur les entiers non signés `np.uint 8` et `np.uint 16`.

Question 1 *Mettre en place une fonction* `somme_ui8(L)` *qui prend en entrée une liste d'entiers non signés sur 8 bits et renvoie leur somme sous forme d'un entier non signé sur 8 bits. On souhaite que la somme le long de la procédure soit bien un entier non signé sur 8 bits.*

R Pour créer une liste d'entiers de type `np.uint8` à partir d'une liste d'entiers de type `int`, vous pouvez utiliser les commandes suivantes :

```
L0 = [12, 57, 255]
L1 = [np.uint8(x) for x in L0]
```

Question 2 *Constater les problèmes discutés en cours et proposer des solutions.*

Question 3 *Mettre en place une fonction* `somme_ui8_vers_ui16(L)` *qui prend en entrée une liste d'entiers non signés sur 8 bits et renvoie leur somme sous forme d'un entier non signé sur 16 bits. On souhaite que la somme le long de la procédure soit bien un entier non signé sur 16 bits.*

Question 4 *Combien de* 255 *peuvent être sommés avec la fonction précédente sans atteindre le phénomène de dépassement d'entier ? Tester ces limites.*

Exercice 2 – Conversion vers le format IEEE-754 simple précision

Vous trouverez dans votre espace des classes 2 fichiers à copier dans votre dossier de travail : `representation_binaire.py` et `conversion_etudiant.py`.

Question 1 *Démarrer le script* `conversion_etudiant.py` *(raccourci* `ctrl+shif+E`*) et constater l'apparition dans la console d'une représentation binaire de* π *au format simple précision (bit de signe en rouge, bits d'exposant en bleu, bits de mantisse en vert).*

Rappel :
- la mantisse doit être positive et doit être comprise entre 1 inclus et 2 exclu ;
- le signe doit être égal à 1 si $x < 0$ et 0 sinon ;
- l'exposant doit être positif si $x \geq 2$ et négatif si $x < 1$.

Question 2 *Écrire une fonction* `nb2sme(x)` *qui renvoie un triplet de nombres* `s, m, e` *correspondant au signe (*0 *ou* 1*), à la mantisse (un flottant) et à l'exposant du nombre non-nul* x *(un entier signé).*

Rappel :
- la mantisse possède une taille de 24 bits pour un flottant simple précision (23 bits mémoires et 1 implicite) ;
- le décalage de l'exposant est de 127 pour qu'il soit toujours codé par un nombre positif sur 8 bits.

Question 3 *En vous servant de la fonction* `int2bin`*, qui converti les entiers en chaine de caractères binaire, convertissez chacun des champs (s, m et e) en chaine de caractère binaire correspondant à la norme IEEE-754.*

Question 4 *Vérifiez la compatibilité avec la fonction* `show_float("f", np.float32(x))` *pour différentes valeurs de* `x` *(en particulier très grand et très petit).*