Réseau de neurone en C

- le MNIST

- choix du modèle de réseau

- objectifs du groupe

- méthodologie

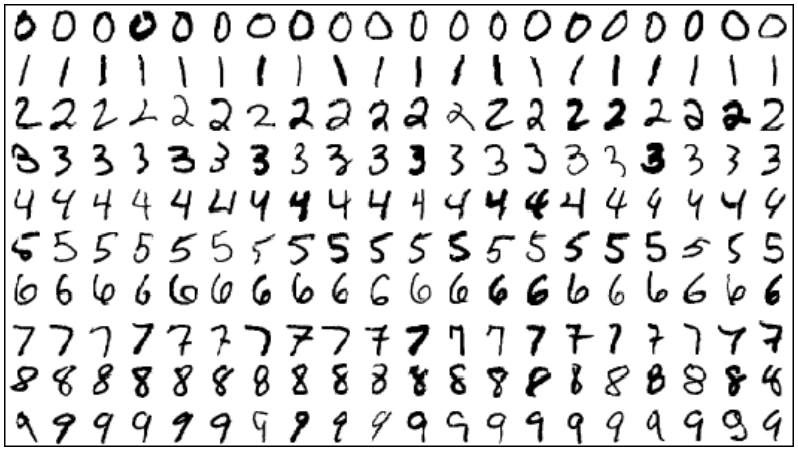
- résultats produits

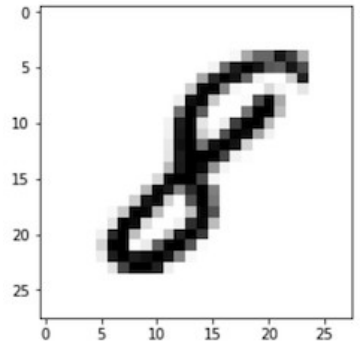
- difficultés

- démonstration

- questions / réponses

Le MINST





Fichier des bitmaps

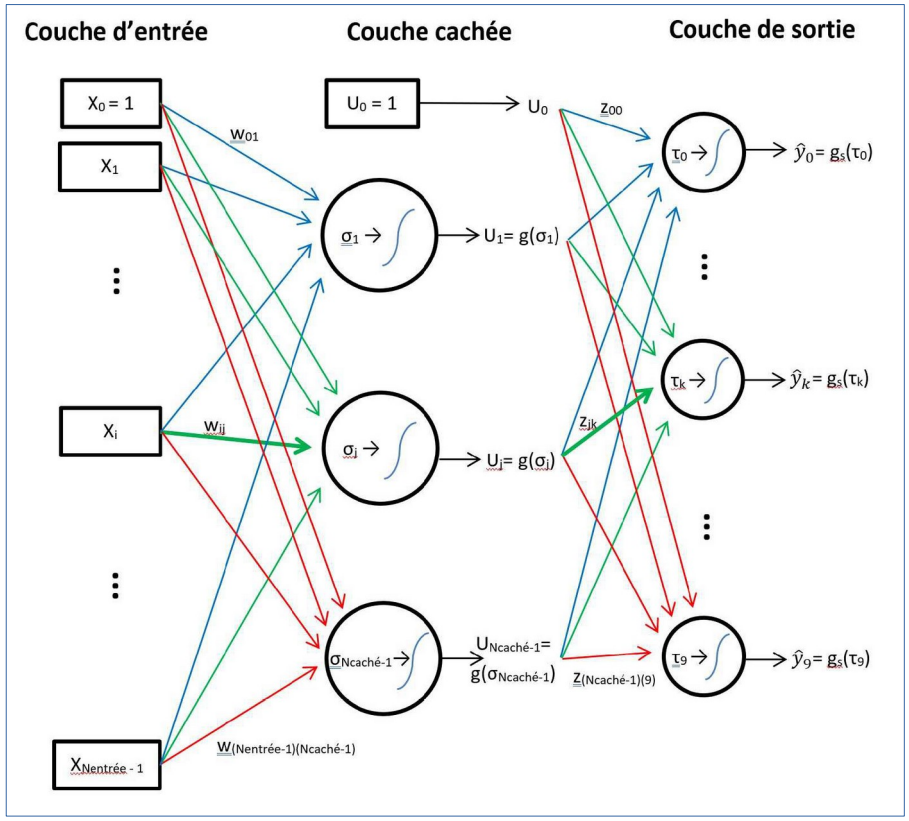
32 bits	32 bits	32 bits	32 bits	28x28	28x28	28x28	28x28	28x28	...
Magic	Nb pic	largeur	hauteur	1 btmp	1btmp	1btmp	1btmp	1btmp	….

Fichier des labels

32 bits	32 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	...
Magic	Nb pic	label	label	label	label	label	label	label	….

Choix du modèle de réseau

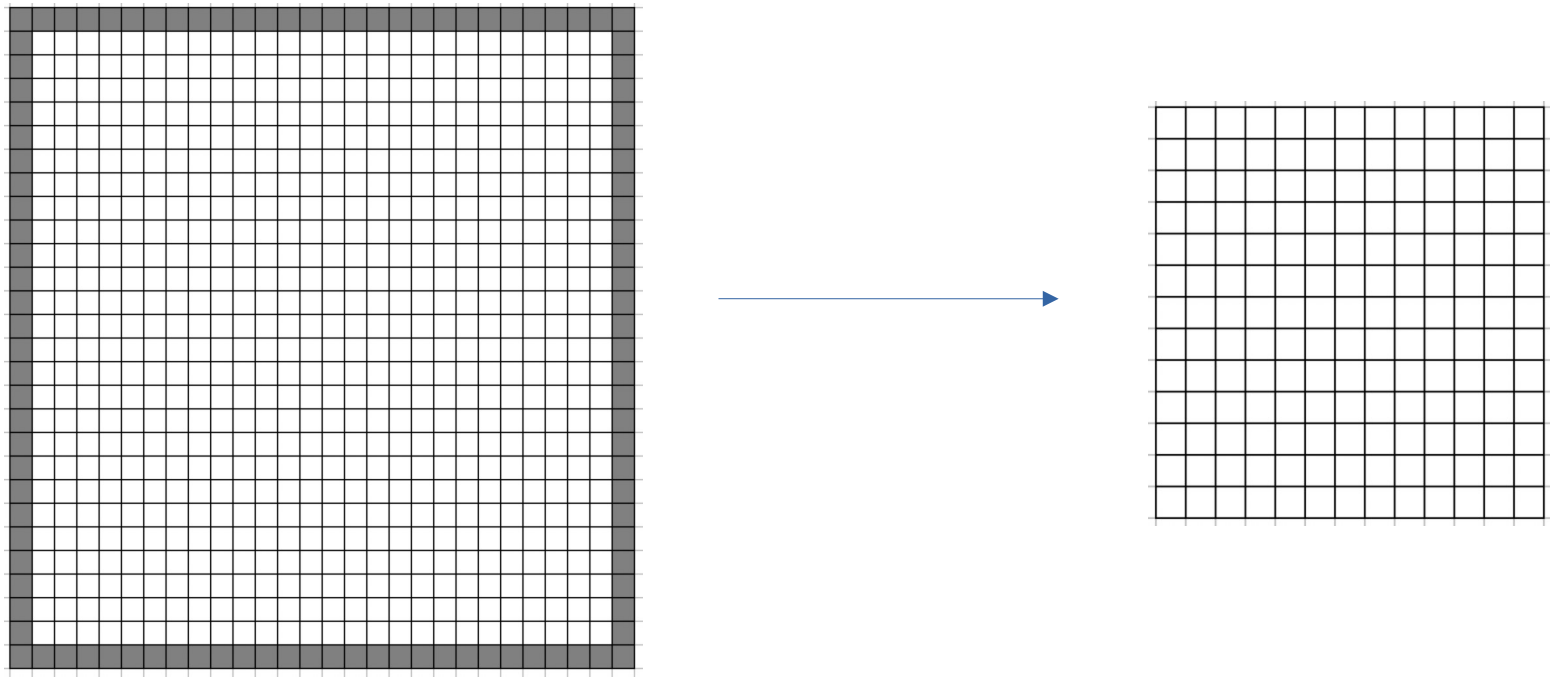
Entrée : 1 bitmap
28 x 28 = 784
pixels
+ 1 biais = 785



Sortie : 10
neurones

Cachée : X neurones

Le MaxPooling :

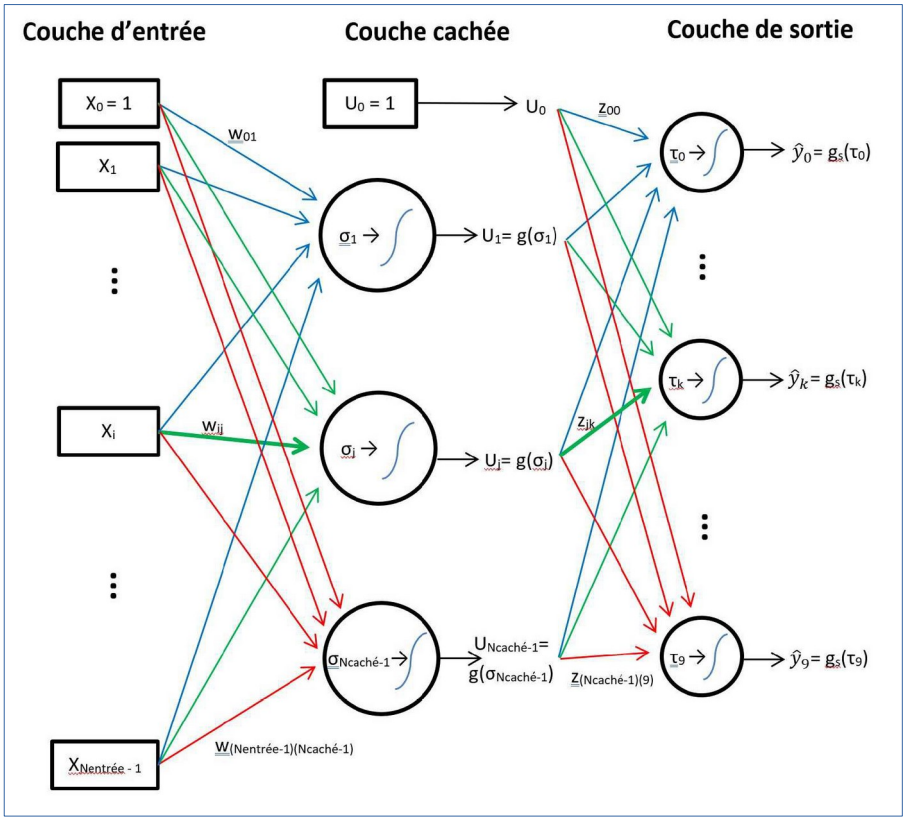


28 x 28

13 x 13

Choix du modèle de réseau

Entrée : 1 bitmap
13 x 13 = 169 pixels
+ 1 biais = 170



Sortie : 10 neurones

Cachée : 66 neurones

(dont neurone de biais)

Objectifs du groupe

Développement en 2 temps

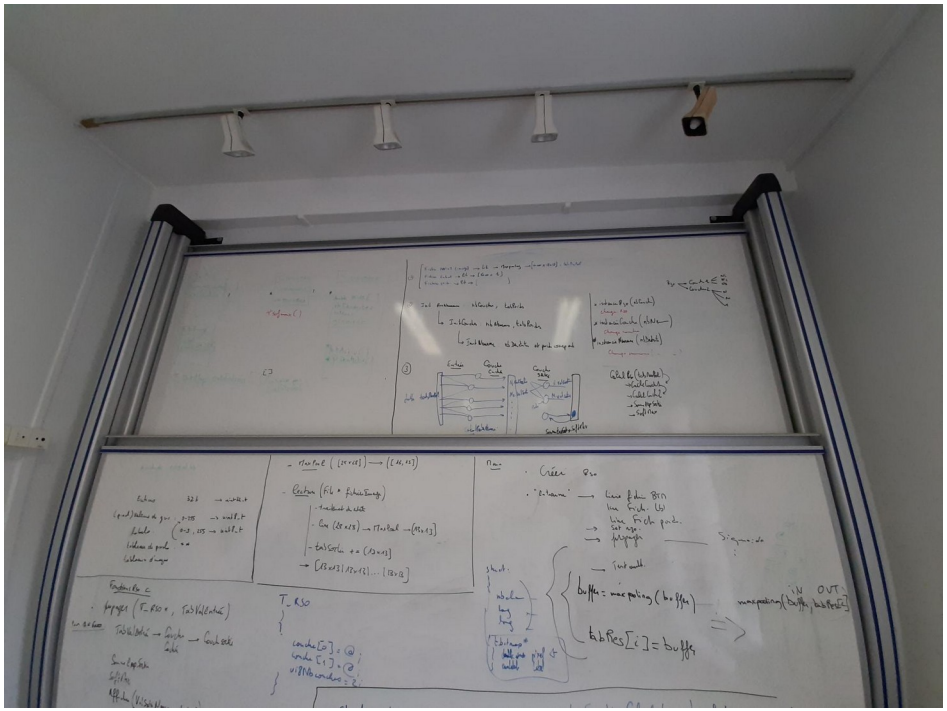
1 – propagation simple :

poids des neurones fournis
passer les 60 000 images dans le réseau
vérifier les prédictions

2 – rétro-propagation

poids initiaux aléatoires
calcul des coûts / erreurs
retro correction des poids
auto apprentissage du réseau

Méthodologie



CADRAGE :

Définition des principales fonctions du programme

Définition des structures à manipuler

Définition des types utiliser / normalisation

Méthodologie

```
→  src ls -la
total 248
drwxr-xr-x  15 maskott  staff    480 24 jui 10:27 .
drwxr-xr-x   6 maskott  staff    192 24 jui 08:43 ..
-rwxr-xr-x   1 maskott  staff    537 23 jui 16:47 Makefile
-rw-r--r--   1 maskott  staff   4276 23 jui 18:30 Util.c
-rw-r--r--   1 maskott  staff    308 23 jui 18:30 Util.h
-rw-r--r--@  1 maskott  staff   6034 23 jui 13:02 calculs.c
-rw-r--r--@  1 maskott  staff    854 23 jui 13:02 calculs.h
-rw-r--r--@  1 maskott  staff   8250 24 jui 09:16 fonctionsRso.c
-rw-r--r--   1 maskott  staff    403 23 jui 18:30 fonctionsRso.h
-rw-r--r--@  1 maskott  staff   6748 24 jui 08:43 functionsIO.c
-rw-r--r--@  1 maskott  staff    832 23 jui 16:47 functionsIO.h
-rw-r--r--@  1 maskott  staff   1569 24 jui 08:33 main.c
-rwxr-xr-x   1 maskott  staff  51840 24 jui 09:16 rso-mnist
-rw-r--r--@  1 maskott  staff   5873 23 jui 16:47 structures.c
-rw-r--r--@  1 maskott  staff   2012 23 jui 18:30 structures.h
→  src
```

STRUCTURE DU PROGRAMME:

Définition des principaux fichiers
Leurs objectifs

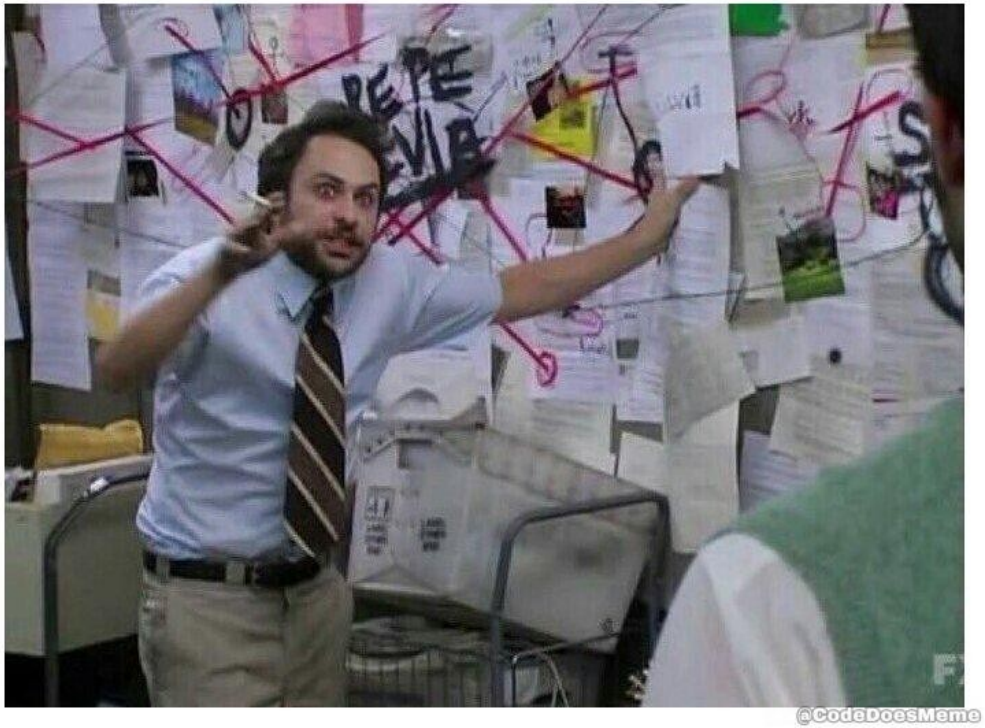
→ Répartition des fichiers au sein du groupe

→ Mise en place du dépôt git pour les fusions

Méthodologie

LUNDI : réflexion

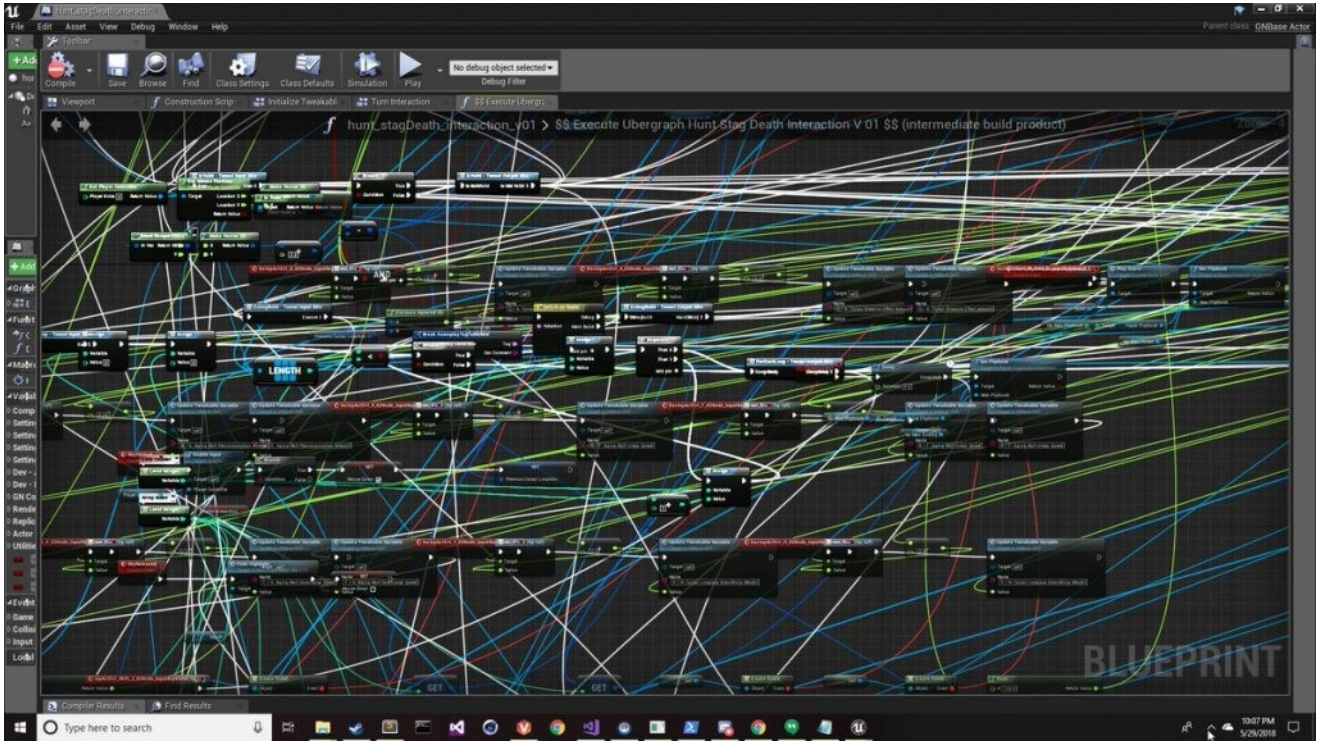
How it looks, when you explain your code to your non-programmer friend



Méthodologie

LUNDI : réflexion

MARDI : codage séparé

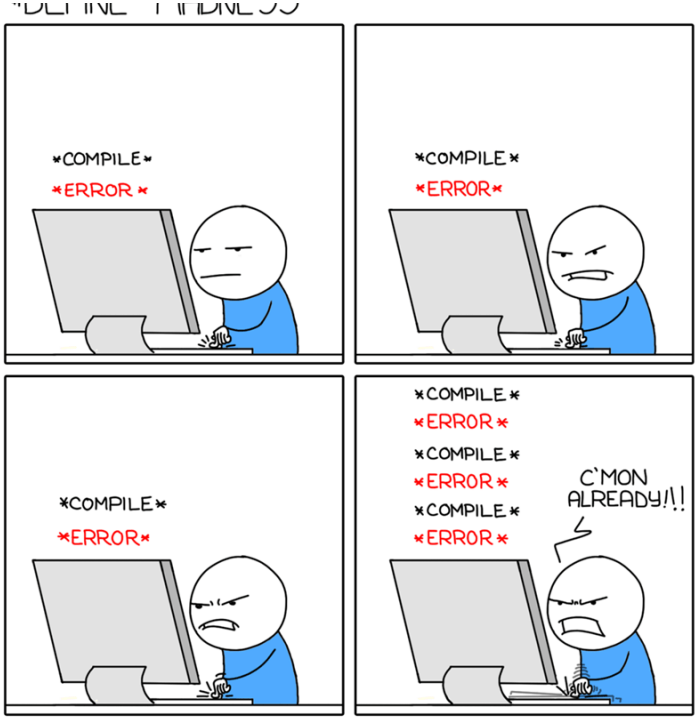


Méthodologie

LUNDI : réflexion

MARDI : codage séparé

MERCREDI : mis en commun, uniformisation des IO, **debugage**



Méthodologie

LUNDI : réflexion

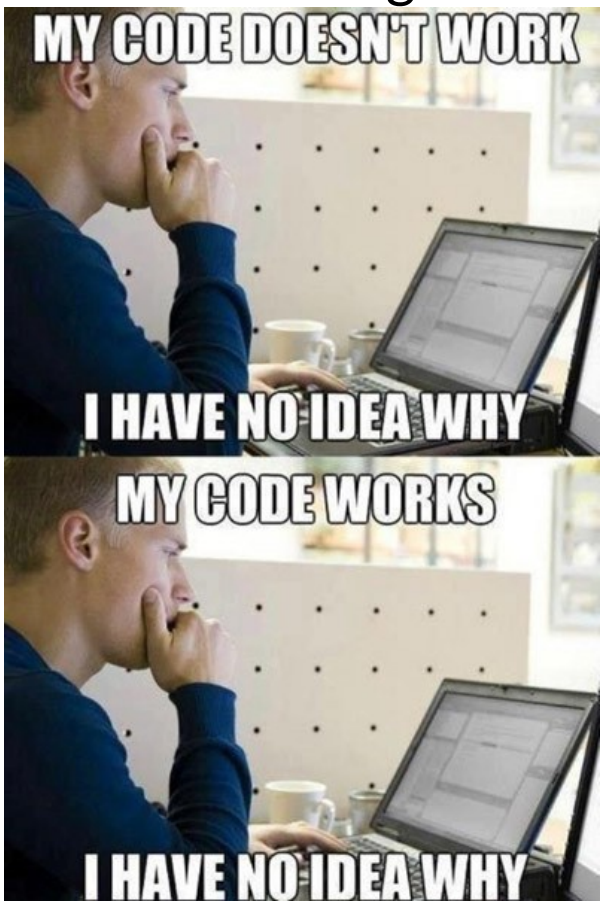
MARDI : codage séparé

MERCREDI : mis en commun, uniformisation des IO, **debugage**

MERCREDI soir : objectif 1 atteint

```
→  src ./rso-mnist
baBitmap = 0x0
bReseau = 0x10d712080
baBitmap = 0x0
Nombre de bitmaps : 60000, p 0x7ffee24f58bc
Instanciation d'un tableau de 60000 T_BITMAP
Tout est instancié
Avant le fclose
cptValLues Final = 11710
Avant Propager
Pointeurs : 0x10d712080 0x7fecb2500000
Nombre de résultats corrects / tests totaux : 56391/60000 soit 93.985000/100
→  src
```

Méthodologie



Sauvegarde du code
Nouvelle branche du git pour l'objectif 2

Méthodologie

JEUDI : itération de la méthode en vue de l'objectif 2

Définition des fonctions / structures / fichier à ajouter ou modifier

Codage et intégration au fur et à mesure.

Effectué pour quelques fonctions :

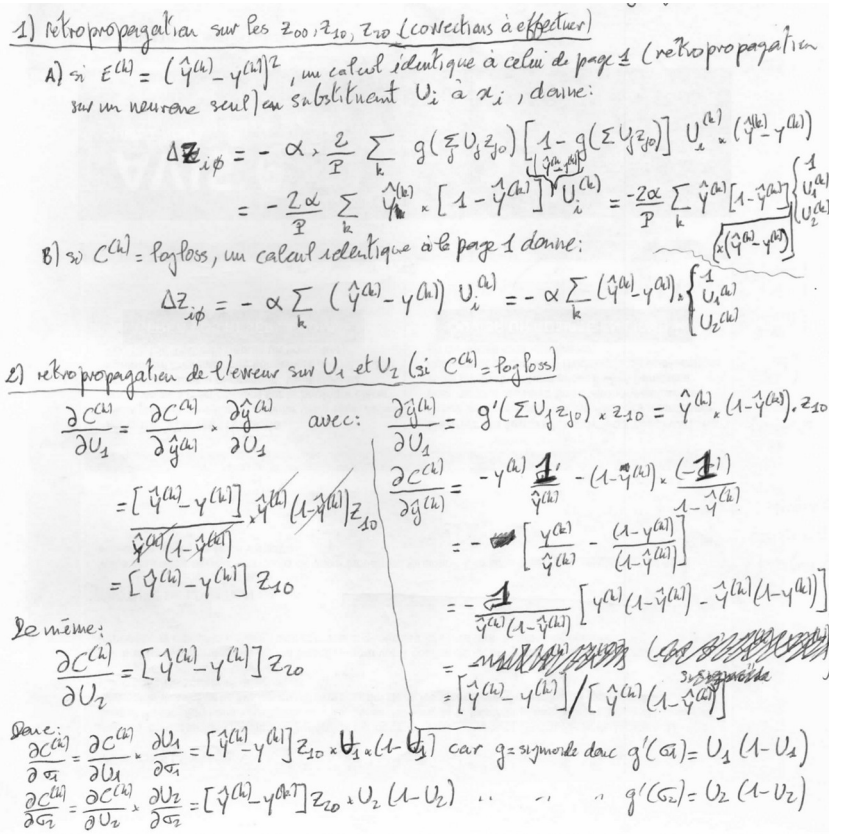
- Initialisation des poids aléatoires
- Calcul du coût
- Statistiques

Difficultés

Le langage C :
- uniformisation des données
(type de variables) → *<stdint.h>*
- manipulation des structures
 **p* → **p* → **p* → *valeur*
- gestion de la mémoire : malloc() / free()

Complexité lié à la structure des réseaux de neurones :
- toutes les pièces du codes sont interdépendantes
- difficulté à séparer les travaux entre individus

Complexité intrinsèque à l'exercice :
- fonctionnement de la retro propagation
- formules des gradients sur les différentes couches



Point positifs

Projet bien dimensionné :

complexe mais permettant de se confronter au C en quelques jours

Utilisation d'outils de travail collaboratif (git)

Apprentissage pour tous devant les « bugs »

Démonstrations

1 – Initialisation du réseau avec un ensemble de poids aléatoires entre -2 et 2

2 – Initialisation du réseau avec des poids corrects (fournis)

3 – Affichage des bitmaps mal identifiés par le réseau

Questions / réponses