

# Génie logiciel

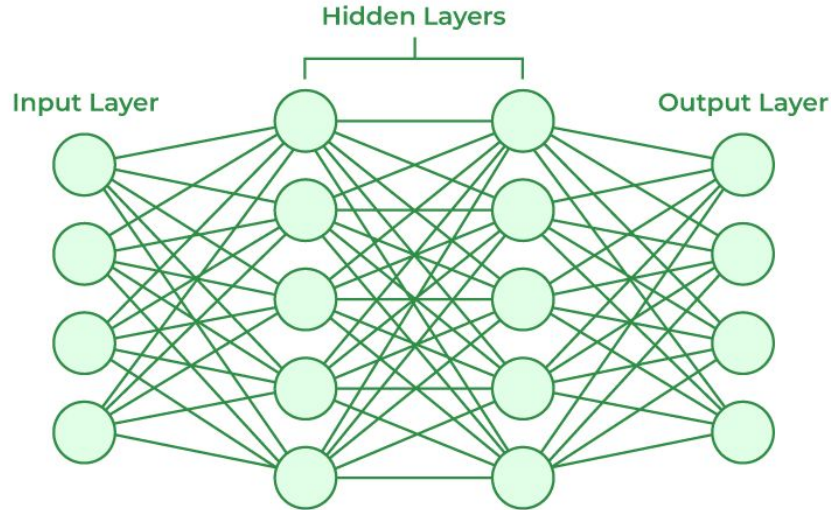
UFR des Sciences Fondamentales et Appliquées



El Mahdi Benfdal  
Ewen Croizier

# Machine Learning

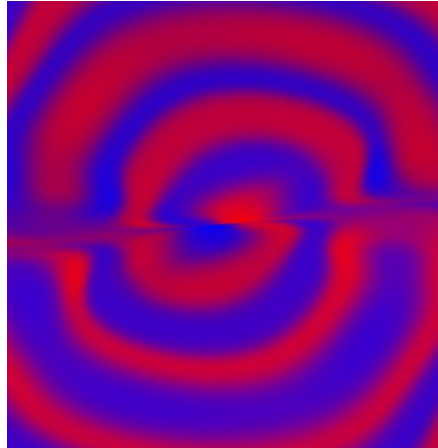
- Apprentissage automatique à partir de données
- Ajustement d'un modèle pour faire des prédictions
- Applications : reconnaissance d'images, classification, prédictions...



El Mahdi Benfdal  
Ewen Croizier

# Notre projet

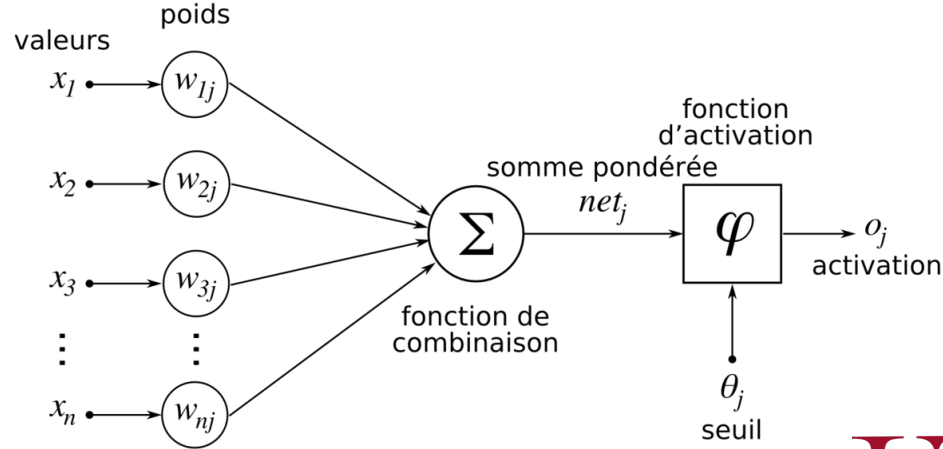
- Objectif : entraîner un réseau neuronal à classifer des spirales
- Outils : C, SDL2, Git, Doxygen



El Mahdi Benfdal  
Ewen Croizier

# Réseau de Neurones

- Neurone = entrées + poids + fonction d'activation
- Rétropropagation pour apprendre
- Réseau = empilement de neurones



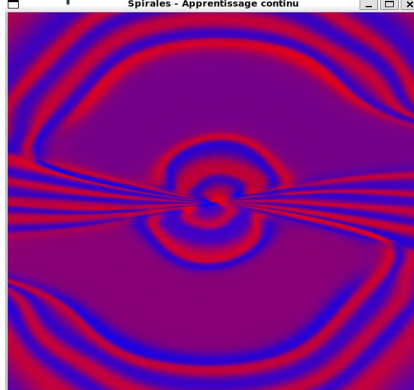
# Génération des spirales

- Génération automatique via spirales d'Archimède
- Amélioration progressive (v1 → v3)
- Sauvegarde/chargement des réseaux via fichiers .bin

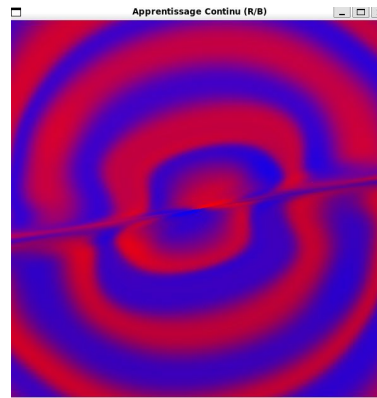
Spiral V1



Spiral V2



Spiral V3

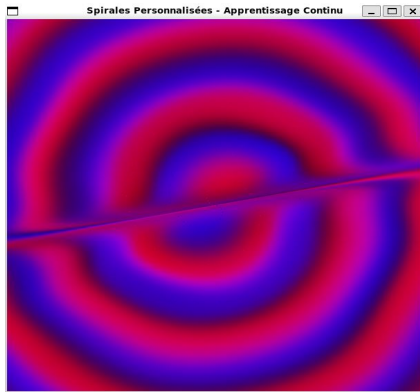


El Mahdi Benfdal  
Ewen Croizier

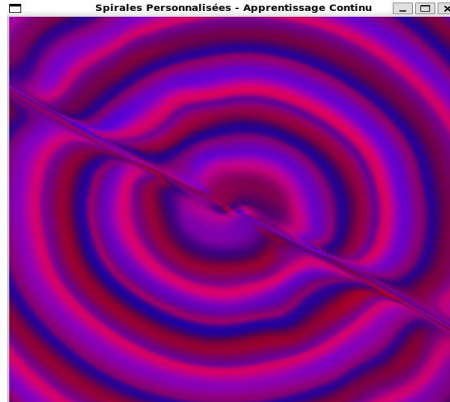
# Spirales personnalisées

- Nombre de classes au choix de l'utilisateur
- Apprentissage dynamique

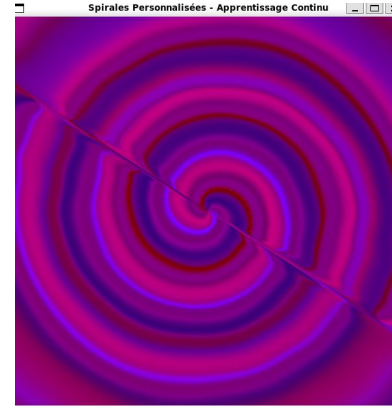
Spiral Avec classification N:3



Spiral Avec classification N:5



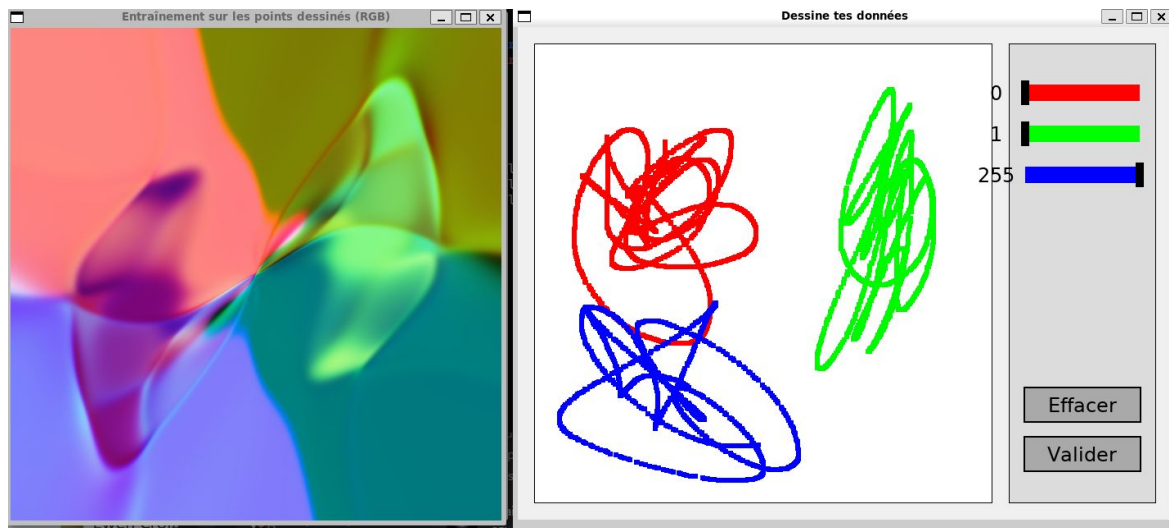
Spiral Avec classification N:8



El Mahdi Benfdal  
Ewen Croizier

# Dessin personnalisé

- L'utilisateur dessine ses propres points
- Entraînement direct sur ces données

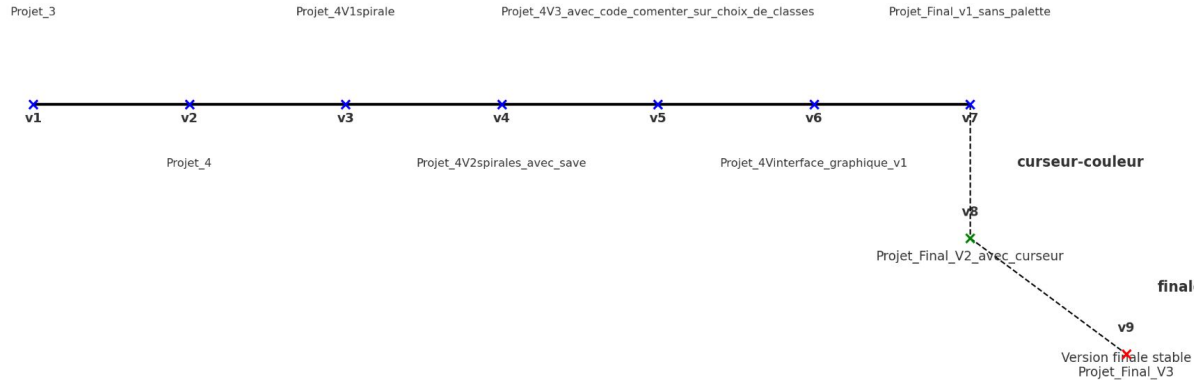


El Mahdi Benfdal  
Ewen Croizier

# Gestion de projet avec Git

- Dépôt Git structuré
- Branches : version initiale, v2, v3, final
- Suivi clair des étapes

master

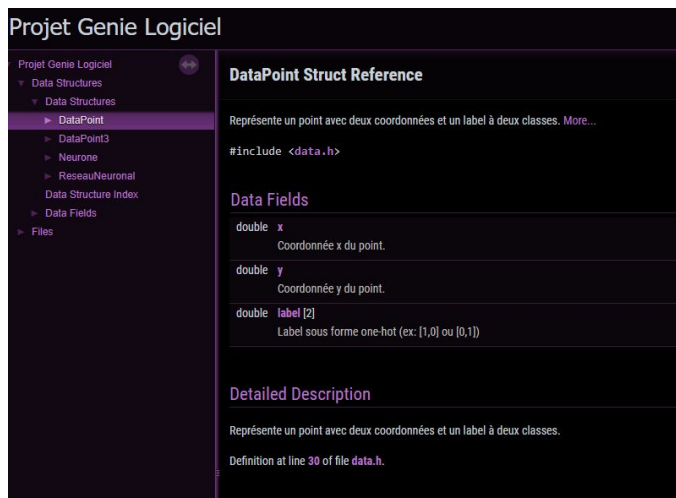


El Mahdi Benfdal  
Ewen Croizier



# Documentation & Rapport

- Doxygen : génération automatique des docs
- Rapport complet (méthodes, résultats)
- Benchmarks réalisés



```
// ----- Configuration générale -----  
  
static int couches_standard[] = {10, 10, 10, 10, 3}; // Utilisé pour les réseaux 2+3  
  
// ----- BENCHMARKS POUR RESEAUX 2+2 -----  
  
// Test de création d'un réseau (2 entrées -> 2 sorties)  
UBENCH(neural_network, creer_reseau_2_2) {  
    int taille_couches[] = {10, 10, 10, 2};  
    ReseauNeuronal *reseau = creer_reseau(2, 4, taille_couches);  
    UBENCH_ASSERT(reseau != NULL); // Vérifie que le réseau est bien créé  
    detruire_reseau(reseau);  
}  
  
// Test de propagation pour réseau 2+2  
UBENCH(neural_network, propagation_2_2) {  
    int taille_couches[] = {10, 10, 10, 2};  
    ReseauNeuronal *reseau = creer_reseau(2, 4, taille_couches);  
    double entree[] = {0.5, -0.3};  
  
    UBENCH_DO_BENCHMARK() {  
        propagation(reseau, entree);  
    }  
  
    detruire_reseau(reseau);  
}
```



1431  
Université  
de Poitiers

# Démonstration live

El Mahdi Benfdal  
Ewen Croizier



# Temps et répartition

- Environ 30 à 40 heures de travail
- Répartition : développement, SDL, Git, doc, test

El Mahdi Benfdal  
Ewen Croizier