TP « neural computing »

Ce TP constituera la moitié de votre note du module, l'autre moitié sera votre note d'évaluation. Vous pouvez réaliser ce TP seul ou en binôme (trinôme non accepté!). Vous aurez à écrire un programme gérant un réseau de neurones capable de détecter la langue d'un *Tweet* .
Nous utiliserons ces abréviation tout au long du TP :

- fr → français
- de → allemand (deutch)
- en → anglais (english)
- es → espagnol
- it → italien
- pt → portugais
- tr → turc

Le déroulement

Nous auront deux séances pour la prise en main du sujet, le reste du travail est à réaliser en autonomie. Vous êtes libre d'utiliser le langage de binding de FANN de votre choix, tant que vous êtes autonomes sur ce langage. Cependant, les deux programmes principaux (l'apprentissage et l'utilisation) vous sont donnés, cela devrait grandement limiter la difficulté du TP... Pour ma part je ne m'engage à pouvoir vous aider uniquement sur le langage C (langage d'origine de la librairie).
Là où vous avez trouvé ce sujet, vous avez aussi trouvé une archive de données (« tweets.tar.gz ») utilisables pour entrainer et tester votre réseau de neurones. Ces données sont naturellement textuelles, c'est à vous de les transformer en données numériques pour pouvoir utiliser les techniques neuronales. Là aussi vous êtes libres d'utiliser le langage de votre choix. Nous verrons ensemble comment aborder ce problème. C'est aussi à vous de créer un protocole d'apprentissage, une bonne base vous a été donnés lors du TD3, vous n'avez plus qu'a modifier des paramètres. De même vous devrez utiliser un nouveau programme vous permettant de tester votre programme de prédiction.

La problématique de classification de langues est légèrement différente que les thématiques faites en TD, en effet en TD on ne gérait que deux classes, or pour le TP il y a en 7. Pour vous aider, je vous donne un code de test (« *test_2.c* ») qui permet de tester un réseau de neurones dans un contexte de classification à N classes. Dans ce contexte, le réseau de neurones possède N sorties. Par exemple pour indiquer la classe 3 sur 7 possible la sortie attendue est : -1 -1 -1 +1 -1 -1 -1 (attention, on commence à compter à 0). C'est à vous de définir une convention (et de vous y tenir) pour associer une langue à un chiffre. Le programme (« test_2.c ») ajoute une simple fonctionnalité de recherche du neurone le plus actif en sortie, ainsi il fournit en sortie la valeur de classification « finale », c'est à dire l'indice du neurone ayant la valeur de sortie la plus forte. Le format de sortie général reste inchangé, vous pouvez donc utiliser le script « confusionMatrix.pl » vu en TD pour calculer la matrice de confusion de votre système.

Vous créerez un protocole d'apprentissage, c'est à dire découper les données dont vous disposez en un ensemble d'apprentissage et un ensemble de test. Vous choisirez la topologie de votre réseau (nombre de neurones cachées, choix des paramètres d'apprentissage, etc...).
Pour réaliser ces transformations de données, vous pourrez utiliser les langages et les outils de votre choix. Veillez à ne pas perdre plus de temps que nécessaire à la réalisation ou la mise en place des ces outils qui ne sont que des intermédiaires.
Vous serez évalué sur la performances de votre système (presque) uniquement.
Notez que toutes ces étapes ont été vues en TD sur des problème différents, mais le formalisme est

wassner@esiea.fr

exactement le même. Seul l'étape de paramétrisation est notablement différente. Une base vous sera expliquée lors des dernières séance de TD.
C'est votre habileté à dérouler correctement un apprentissage ainsi que de trouver une bonne paramétrisation, qui vous permettront d'avoir un bon système de prédiction.
Voyez la dernière section de de document pour trouver des stratégies d'amélioration, mais commencez d'abord par mettre en place ce qui a été vu en TD, avant de chercher à faire des choses compliquées.

Les données

Les données d'entraînement

Les données d'entraînement sont dans une archive (« tweets.tar.gz ») attachée à l'article où vous avez trouvé ce sujet. Il s'agit de 7 fichiers de textes, composés d'un tweet par ligne, dans les 7 langues à prédire. L'information de la langue est dans le nom de chaque fichier.
Détails important :

- Un tweet est un message de 140 symboles maximum, mais attention, une fois encodé dans un fichier au format UTF-8, certain symboles seront représenté sur plusieurs octets (jusqu'à 3) ! Donc quand vous allouerez de l'espace mémoire pour lire et analyser un tweet, prenez de la marge et allouez plutôt 3*140 octets que 140...
- Il se peut que certain tweets ne soient pas dans la langue annoncée, en effet, parfois des utilisateurs déclarent « tweeter » en anglais, mais de temps en temps ils utiliseront leur langue maternelle. Parfois certaines personnes pourront « tweeter » (ou « re-tweeter ») un message en anglais alors qu'ils « tweetent » la plupart du temps dans la langue déclarée. Il faut vous débrouiller avec cela, c'est une situation très classique en apprentissage : les données sont rarement 100% correctes... Les réseaux de neurones supportent assez bien les apprentissage avec des données imparfaites. Cela est vrai à la fois dans la base d'apprentissage (et de test) mais aussi d'évaluation, donc évidement le 100% de prédictions correctes n'est pas atteignable.

Les données d'évaluation

Les données d'évaluation (« eval.txt.gz ») vous seront communiquées à la fin du cours. Ce sera un fichier texte au même format que les données d'entraînement, sauf bien sur que ce fichier contiendra des tweets de différentes langues, et que l'information de la langue n'est pas donné. C'est votre programme qui devra la déduire.
Vous devrez donc appliquer votre chaîne de transformation pour transformer ces tweets en valeurs numériques utilisable dans votre réseau, comme vous l'avez fait pour l'apprentissage. Ensuite vous utiliserez le réseau que vous aurez entrainé (et testé) pour prédire la langue de chacun de ces messages.

Attention :

- Pour respecter le format de fichier FANN, il vous faudra donner une valeur de sortie attendue pour chaque entrée, même si vous ne disposez justement pas de cette information... Vous n'avez qu'à donner une valeur quelconque, puisqu'elle ne sera pas utilisée (car vous ne ferez aucun apprentissage avec ce fichier, mais simplement de la prédiction).
- Vous devrez aussi légèrement modifier le programme « *test_2.c* », que je vous ai fournit, pour respecter le format de rendu de votre livrable principal : le fichier de prédiction. Cela est décrit en détail dans la section « *les modalité de rendu de TP* ».

Les modalités de rendu du TP

Vous rendrez votre travail par mail à cette adresse **wassner@esiea.fr**, avec votre adresse mail ESIEA. Ce mail doit contenir une archive au format « zip », **cette archive doit contenir un répertoire composé du nom du binôme.**
Pensez que je vais recevoir plein d'archives, si vous ne mettez pas vos fichiers dans un répertoire les fichiers des différents étudiants vont se mélanger. Vous mettrez dans ce répertoire, un fichier qui s'appelle « *prediction.txt* » (le résultat de vos analyses sur le fichier d'évaluation), ainsi qu'un document **d'une page** (ou au pire deux , mais pas plus) qui s'appelle « rapport.pdf » (et donc au format PDF). Le rapport doit expliquer la stratégies utilisée :

- Quels sont les paramètres d'entré du réseau.
- Quelle est la topologie du réseau :nombre de neurones d'entrée , de couches couches, nombre de neurones pas couche, et nombre de neurones de sorties.
- Quel protocole d'apprentissage a été utilisé
  - Taille de la base d'apprentissage, et de la base de test.
  - Avez vous utilisé ou non des données extérieures ? Si oui décrivez brièvement de ces données (où vous les avez trouvé, pourquoi vous les avez choisies)
- Vous devez annoncer les performances de votre système en donnant la matrice de confusion correspondant à vos tests.

Vous mettrez aussi dans l'archive tout les programmes que vous avez utilisé pour réaliser ce TP, y compris les programmes de paramétrisation (transformation des tweets en valeur numérique).
La taille de l'archive doit rester raisonnable, donc si vous avez utilisé des données additionnelles pour l'apprentissage, ne les mettez pas dans l'archive.

Format du fichier de prédiction

Le fichier de prédiction est votre livrable principal. Ce fichier sera analysé par un script il est donc primordial qu'il respecte le format demandé. Ce format est très simple : un mot par ligne, le mot en question est l'abréviation de la langue détectée.
Attention : Il vous faudra appliquer un dernier changement, soit au programme, soit au fichier créé par le programme « *test_2.c* » qui, par défaut affiche le numéro de classe identifié. Ce numéro de classe est une convention que vous avez décidé lors de l'apprentissage, mais à la fin il faut fournir une information du type « fr, it, es, de, en, pt, tr ».

Voici un exemple :

Fichier de tweets	*Fichier de prédiction*
passez la journée aux commandes du métro parisien (via @metronews) http://t.co/idcszavdh4	fr
perde il controllo del #suv e #sfonda la vetrina di una #banca, uccisa una #cliente http://t.co/p5dopapqwu via @ilmessaggeroit	it
rt @noviosdejovenes: típico: te pones los audífonos, te hablan, y tu solo sonríes y dices que "si"	es
lets stop with the judgements en ol..... the last time i checked we all had closets,,,,, juhs coz i sin differently, u dnt judge!!	en
...	...

Donc le fichier « prediction.txt » doit ressembler à ceci :

```
fr
it
es
en
...
```

Attention : vous devrez vérifier qu'il n'y a pas de décalage de lignes, sinon le script de notation risque de croire que votre système ne marche pas du tout ! Donc veillez bien à ce que votre fichier de prédiction possède bien, exactement, le même nombre de lignes que le fichier de données d'évaluation.

Notation

La note de votre TP sera directement corrélée à l'efficacité de classification de votre programme, sur 15 points. Les programmes ayant les meilleures performances obtiendront 15 points, et le nombre de points attribués décroitrons au fur et à mesure que les performances s'approcheront des performances d'un classeur aléatoire.

Les 5 points restants seront réservés à la qualité du respect des modalités de rendu.
Le rendu se fera par mail, lisez attentivement les modalité de rendu avant de faire votre envoi. Toute nécessité d'envoi d'un échange de mail supplémentaire (oubli d'éléments dans le mail, mauvais format de données) vous coutera un point à chaque échange. De même tout élément manquant dans le rapport pourra vous coûter un point.

Conseils stratégiques

Je vous conseille de d'abord réaliser la base du TP en suivant les consignes. Ensuite vous pouvez essayer d'améliorer votre prédiction, mais ne passez pas tout votre temps à mettre en place une stratégie compliquée, pour vous rendre compte la veille de la date de rendu du projet que ça ne marche pas ou très mal.
Partez d'une base simple qui fonctionne (même moyennement), et ensuite seulement essayez de l'améliorer.

Suggestions

Mettez d'abord en place les stratégies de base, faites en TD, avant d'essayer de mettre en places les éléments présenté dans cette section.

- Les tweets des bases d'entraînement ont été collectés avec la librairie Twitter4j. Ils ont été collectés sur très peu de jours, il ne contiennent donc que peu de variabilité, ils sont même probablement un peu biaisé par les « Trending Topic » du moment. Les données utilisées pour l'évaluation ont été collectées un autre jour, pour lesquels les « Trending Topic » seront forcément différent. Si vous avez bien compris le cours vous devez naturellement comprendre qu'il y aura certainement de l'efficacité à gagner en améliorant la base d'apprentissage.
- Faites très attention à l'étape de paramétrisation, appelé aussi normalisation. La base c'est de s'assurer que les données mises en entrée correspondent au type de neurones choisis. Typiquement si vous choisissez des neurones fonctionnant dans [-1;+1], non seulement vos données devrons être dans cet intervalle, mais, idéalement ils devront aussi utiliser tout cet intervalle ! Par exemple si vous n'utilisez que l'intervalle [0;1], le réseau fonctionnera plutôt

mal. Dans la mesure où vos paramètres sont des fréquences, vous devrez essayer d'estimer la valeur maximum de cette fréquences et moduler chaque paramètre pour qu'il utilise au mieux toute la dynamique du type de neurone choisit.

- Augmentez la base de données avec des documents pris sur le web, ou mieux encore des tweets.
- Limitez la redondance : supprimez les « ReTweet » peu informatifs pour l'entrainement et qui peuvent fausser la mesure de performance du système. En effet les « re-tweets » sont des message relayés sans changement de contenu. Ainsi deux message identiques peuvent se retrouver dans l'ensemble d'apprentissage et dans l'ensemble de test, faussant ainsi la mesure de performance du système.
- Éliminer les URL (car elles sont raccourcies et ne sont donc pas conformes aux statistiques de la langue)

Je serai disponible au laboratoire ATIS pour répondre à vos questions.

Bon courage et bon travail.

wassner@esiea.fr