Reconnaissance vocale lors d'appel d'urgence grâce à un réseau de neurones

Tran-Thuong Tien-Thinh, MPSIA, 2021-2022

Résumé

Ancrage au thème de l'année : D'après les chiffres du ministère, il y a plus de 31 millions d'appels d'urgence par an. Ces appels sont répartis sur 103 centres de plus en plus sollicités. Alors que les recommandations fixent un taux de 90% de réponses en moins de 60 secondes, seul 69% des appels sont décrochés dans la minute. (53 mots)

Motivation du choix de l'étude : Afin de répondre à ce problème, nous nous proposons d'étudier un réseau de neuronnes capable de faire de la reconnaissance vocale, pour alléger le travail des opérateurs d'appel d'urgence. Ce réseau de neuronnes devra être capable de classifier des fichiers audios selon des mots-clé relatifs aux appels d'urgence. (54 mots)

Professeurs encadrants du candidat :

Philippe Châteaux

Positionnements thématiques

INFORMATIQUE (Informatique Pratique, Descente de Gradient)
PHYSIQUE (Application de la Transformée de Fourier)

Mots clés

Mots-clés (en français) :	**Mots-clés** (en anglais) :
Réseau de neurones	*Neural network*
Apprentissage profond	*Deep learning*
Algorithme du gradient	*Gradient descent*
Transformée de fourier	*Fourier transform*
Reconnaissance vocale	*Voice recognition*

Bibliographie commentée

La première modélisation informatique du neurone, appelé perceptron, proposé par McCulloch et Pitts, date de 1943. Cette modélisation possède des poids d'apprentissage et une fonction d'activation. Lorsque les données sont entrées, elles sont pondérées par les poids du perceptron puis elles sont sommées et enfin transmises à la fonction d'activation. Le résultat obtenu devient alors la sortie renvoyée par le perceptron. Cela modélise bien le comportement d'un neurone qui récupère les différentes entrées par ses dendrites, les traite et transmet le résultat par son axone. Pour pouvoir traiter des données complexes, il est nécessaire de mettre plusieurs neurones en réseau et effectuer un apprentissage profond.

L'apprentissage s'effectue sur des données dont la sortie attendue est connue. L'entraînement consiste alors à corriger les poids afin de faire converger le système vers un résultat "optimal". Cette rectification des poids du perceptron, se fait en calculant l'erreur en sortie, puis en effectuant la rétropropagation. On calcule la différence à appliquer au poids par rapport à l'erreur grâce à l'algorithme de descente de gradient. [1]

Un perceptron seul est capable de faire une séparation linéaire parmi les données d'entrées. Il peut reproduire les opérateurs logique AND et OR qui sont linéaires, mais l'opérateur XOR nécessite un réseau de perceptrons car il effectue une séparation non linéaire. On associe ainsi les neurones en parallèle pour former une couche de perceptrons, puis on associe ces couches en série pour former un réseau de perceptrons. Voici un article introduisant la mise en réseau de perceptron et traitant notamment du problème XOR. [2]

Comme l'apprentissage d'un réseau de neurones se fait en utilisant des données en exemple, deux problèmes majeures peuvent intervenir ; un ensemble de données d'apprentissage biaisé, ou un temps d'apprentissage trop long.
Pour le premier, il est difficile d'y remédier, il est possible de faire une séparation des données pour l'apprentissage et le test, ou encore d'introduire des variables aléatoires comme la couche "Dropout", mais ces deux cas ne permettent que d'éviter le sur-apprentissage ("overfitting").
Pour résoudre le problème du temps d'apprentissage, plusieurs études ont mené à des approches différentes. Il est possible d'adapter le réseau de neurones au problème, on peut également modifier les fonctions d'activation et d'erreur, ou encore faire varier le taux d'apprentissage des poids à chaque rétropropagation. Dans notre étude nous nous concentrerons principalement sur cette dernière optimisation, qui est plus documentée car aussi plus géné-

raliste. L'ensemble de ces méthodes est décrit dans ce livre de 2006 qui fait un état de l'art très poussé. [3]

De plus en plus, les réseaux de neurones sont devenus importants en taille, ce qui entraîne un temps d'apprentissage nécessairement plus long. Alors que certains problèmes semblent se ressembler, il paraît peu cohérent de réentraîner tout un réseau de neurones lorsque l'on en possède déjà un entraîné pour un problème similaire. Il est alors possible de réaliser un transfert d'apprentissage, en initialisant les poids des perceptrons avec ceux de l'ancien réseau de neurones, en ne laissant modifiable par rétropropagation que les perceptrons de la couche de sortie. Le temps d'apprentissage est donc réduit par réduction du nombre de poids à entraîner. C'est ce que nous propose de faire Tensorflow. [4] (524 mots)

Problématique retenue

Il s'agit de concevoir un réseau de neurones capable de reconnaître des mots-clés prononcés dans un extrait audio. (18 mots)

Objectif TIPE du candidat

1. Faire un réseau de neurones qui converge grâce à l'algorithme du gradient
2. Améliorer la vitesse d'entrainement grâce à des optimizers basés sur la descente de gradient
3. Essayer ce réseau sur la base de données du MNIST pour reconnaitre des chiffres
4. Utiliser le transfert d'apprentissage pour reconnaître ce qui a été prononcé dans un audio parmi une liste de mot-clé

(74 mots)

Bibliographie

(Pas encore par ordre de priorité décroissante.)

[1] Introduction to Artificial Neural Networks (ANNs) `https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj63rOrj4z0AhXQx4UK`

url=https%3A%2F%2Fdatascience.foundation%2Fdownloadpdf%2F9%2Fwhitepaper&usg=AOvVaw1kSbLwFPHkPBFwP85r9QC2

[2] An Introduction to Neural Networks with kdb+ https://kx.com/wp-content/uploads/2020/09/An-Introduction-to-Neural-Networks-with-kdb.pdf

[3] Pattern Recognition and Machine Learning, Christopher Bishop, 2006

[4] Base de données de mot-clé en anglais et technique de transfert d'apprentissage sur des réseaux de neurones de Tensorflow : https://www.tensorflow.org/tutorials/audio/simple_audio