



## **Machine Learning 2**

Master 2 Génie logiciel

**HAI923I**

---

### **Compte rendu d'articles**

---

*Réalisé par :*

Mr. BERKANE Ibrahim

*Encadré par :*

M. TODOROV Konstantin et  
l'équipe pédagogique

Promotion : 2021/2022

## TABLE DES MATIÈRES

1	Résumé de l'article . . . . .	2
2	Fonction d'activation . . . . .	2
2.1	Définition d'une fonction d'activation . . . . .	2
2.2	L'intérêt d'utiliser une fonction de type ReLu . . . . .	2
2.3	Leaky ReLu VS ReLu . . . . .	3
3	Surapprentissage et Dropout . . . . .	3
3.1	Surapprentissage . . . . .	3
3.2	Dropout . . . . .	3
3.3	Dropout et la diminution du surapprentissage . . . . .	3
4	CNN 2D et CNN 3D . . . . .	3
4.1	CNN 2D vs CNN 3D . . . . .	3
4.2	Autre type d'architecture pour la prise en compte de la notion tem- porelle . . . . .	4
4.3	Les défauts et les pistes d'amélioration . . . . .	4
4.4	Les défauts . . . . .	4
4.5	Les pistes d'amélioration . . . . .	4

# Détection de la fumée et du feu par réseau de neurones convolutifs.

## 1 Résumé de l'article

Dans le domaine de la détection du feu et de la fumée la majorité des approches se concentrent sur les caractéristiques de couleurs et de formes du feu et de la fumée ensuite de construire un vecteur de caractéristiques multidimensionnel créé par des experts. Cet article propose la construction du vecteur de caractéristique en faisant extraire automatiquement celui-ci par un réseau de neurones convolutif pour identifier le feu et la fumée d'une vidéo en temps réel. Ce réseau de neurones combine des couches de convolution et Max pooling avec une profondeur de 9 couches, et à la sortie il y a trois classes Négative, Fumé et Feu. L'utilisation de la fonction d'activation Leaky Relu et des dropout sur des images de 64\*64 RGB en entrée avec 100 itérations a donné 97.9 % de précision sur l'ensemble de test. Afin de diminuer le temps de traitement pour la détection ils ont appliqué une fenêtre glissante de taille 12\*12 par pas de 2 pixels sur la dernière carte de caractéristique (les 6 premières couches) au lieu de glisser une fenêtre de 64x64 pixels dans l'image d'origine RGB.

## 2 Fonction d'activation

### 2.1 Définition d'une fonction d'activation

Une fonction d'activation est une fonction utilisée dans les réseaux de neurones qui génère une petite valeur pour les petites entrées et une valeur plus grande si ses entrées dépassent un seuil. Si les entrées sont suffisamment grandes, la fonction d'activation "se déclenche", sinon elle ne fait rien. En d'autres termes, une fonction d'activation est comme une porte qui vérifie qu'une valeur entrante est supérieure à un nombre critique. Il y a 2 types de fonction linéaire et non-linéaire.

### 2.2 L'intérêt d'utiliser une fonction de type ReLu

Les fonctions d'activation sont utiles car elles ajoutent des non-linéarités dans les réseaux de neurones, permettant aux réseaux de neurones d'apprendre des opérations puissantes. Si les fonctions d'activation devaient être supprimées d'un réseau de neurones à action directe, l'ensemble du réseau pourrait être refactorisé en une simple opération linéaire ou une transformation matricielle sur son entrée, et il ne serait plus capable d'effectuer des tâches complexes telles que la reconnaissance d'images. Comme la fonction Relu permet au modèle de généraliser ou de s'adapter facilement à une variété de données et de différencier les sorties.

## 2.3 Leaky ReLu VS ReLu

Parce que la dérivée de la fonction ReLU à  $x = 0$  peut être définie sur 0 ou 1 et la dérivée nulle pour  $x$  négatif, ça peut donner lieu à des problèmes lors de l'entraînement d'un réseau de neurones, car un neurone peut être dans la région zéro et la backpropagation ne changera jamais son poids, de plus toutes les valeurs négatives deviennent immédiatement nulles, ce qui diminue la capacité du modèle à s'adapter ou à s'entraîner correctement à partir des données. Donc Leaky ReLU permet d'augmenter la portée de la fonction ReLU. Par conséquent, la plage du Leaky ReLU est (-infini à l'infini).

## 3 Surapprentissage et Dropout

### 3.1 Surapprentissage

Surapprentissage est une erreur de modélisation qui se produit lorsqu'une fonction est trop étroitement alignée sur un ensemble limité de points de données. Par conséquent, le modèle n'est utile qu'en référence à son ensemble de données initial, et non à d'autres ensembles de données.

### 3.2 Dropout

Dropout : pendant l'entraînement, un certain nombre de sorties de couche sont ignorées de manière aléatoire.

### 3.3 Dropout et la diminution du surapprentissage

Dropout fait que la couche ressemble et soit traitée comme une couche avec un nombre de nœuds et une connectivité différente par rapport à la couche précédente. En effet, chaque mise à jour d'une couche pendant l'apprentissage est effectuée avec une « vue » différente de la couche configurée. Cela rend le processus d'apprentissage bruyant, forçant les nœuds d'une couche à assumer de manière probabiliste plus ou moins la responsabilité des entrées et qui permet de généraliser le modèle donc pas de surapprentissage.

## 4 CNN 2D et CNN 3D

### 4.1 CNN 2D vs CNN 3D

L'idée principale des convolutions 2D est que le filtre convolutif se déplace dans les 2 directions ( $x, y$ ) pour calculer les caractéristiques de faible dimension à partir des données d'image. La forme de sortie est également une matrice à 2 dimensions, par contre les convolutions 3D appliquent un filtre à 3 dimensions à l'ensemble de données et le filtre se déplace dans 3 directions ( $x, y, z$ ) pour calculer les représentations d'entités de bas niveau. Leur forme de sortie est un espace de volume à 3 dimensions tel que cube ou cuboïde. Ils sont utiles pour la détection d'événements dans les vidéos, les images médicales 3D, etc.

Ils ne se limitent pas à l'espace 3D mais peuvent également être appliqués aux entrées spatiales 2D telles que les images.

## **4.2 Autre type d'architecture pour la prise en compte de la notion temporelle**

Graph Neural Networks, Capsule neural networks et Recurrent neural network.

## **4.3 Les défauts et les pistes d'amélioration**

### **4.4 Les défauts**

1. La dynamique temporelle n'est pas prise en compte dans le modèle.
2. La fumée n'est pas bien détectée.
3. Le modèle ne détecte que le feu rouge.

### **4.5 Les pistes d'amélioration**

1. Utiliser un réseau de neurones convolutionnels 3D pour prendre en compte la dynamique temporelle dans le modèle.
2. Améliorer la base d'image et augmenter la taille du réseau ou en ajoutant des images de l'objet « fumée » dans la base d'apprentissage pour bien détecter le fumé.
3. Augmenter la base d'image avec d'autres couleurs de feu telles que bleue.