



# Cats & Dogs

Rapport 2

Arounaguiry Jency  
GP IA



### a. Modèle de base

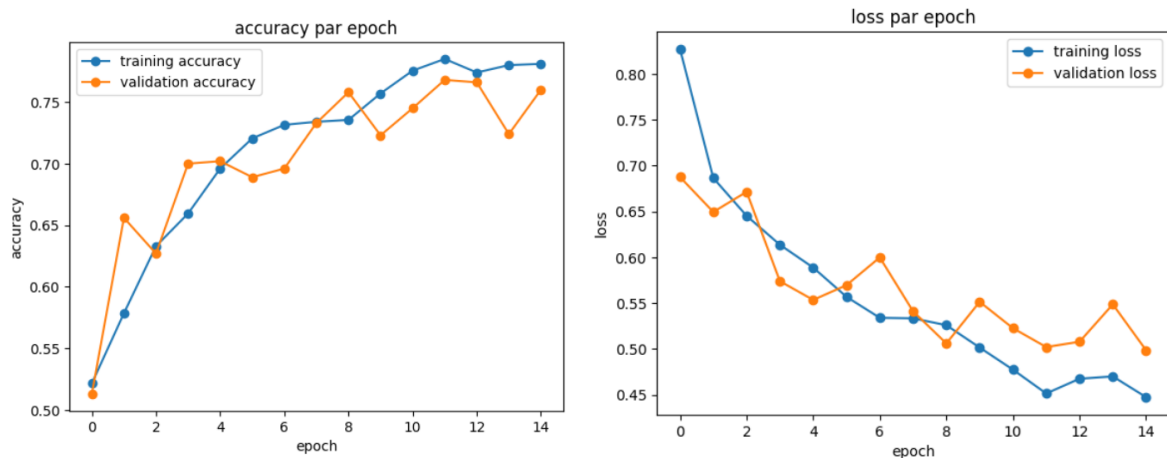
Pour cet exercice, j'ai tenté d'utiliser différents paramètres afin d'obtenir un niveau de précision se rapprochant de 0.98. Nous allons analyser les différents résultats obtenus et terme de qualité de prévision.

#### Présentation de la base de données

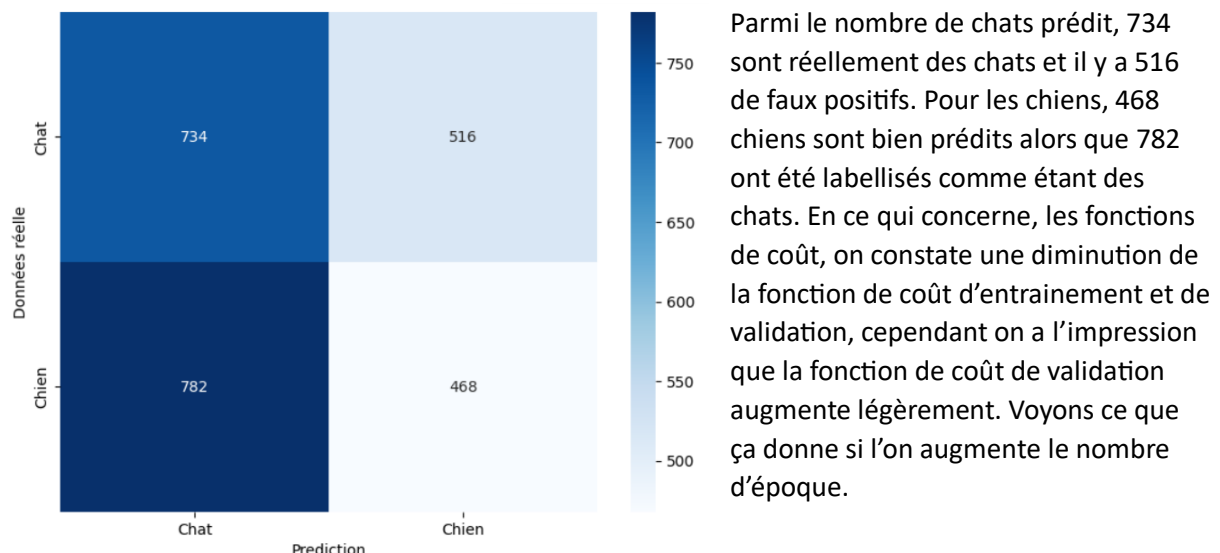
Je dispose d'un modèle d'entraînement composé de 22 500 images et un modèle de validation, soit de test, composé de 2500 images. Chacune de ces bases de données sont divisées en part égale d'image de chats et de chiens.

#### Première exécution

Dans un premier temps, j'ai exécuté le modèle de base que vous nous avez fait parvenir. En utilisant, l'algorithme d'optimisation RMSProp, une vitesse d'exécution (learning rate) de 0.001, qui parcourt 15 epoch.



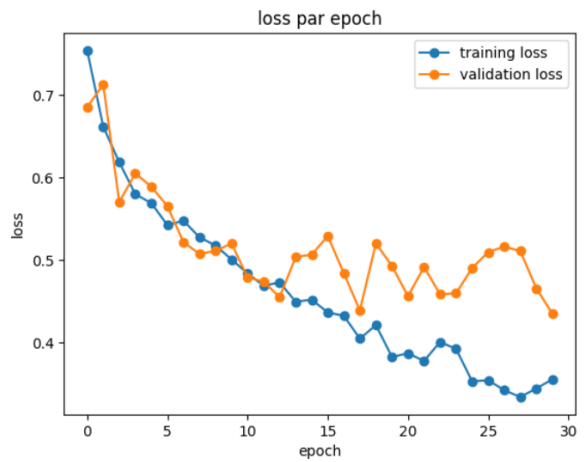
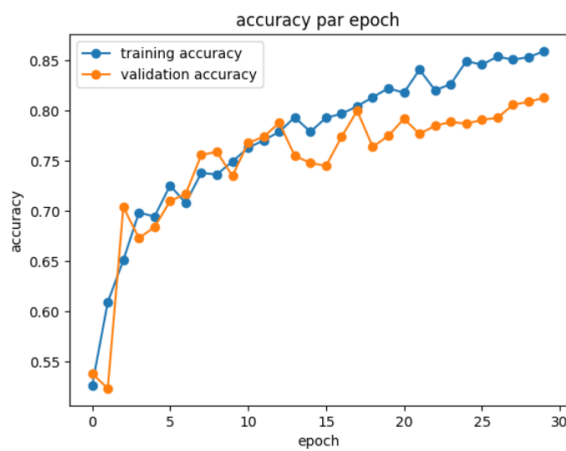
125/125 [=====] - 18s 142ms/step  
Nombre de chats : [1516]Nombre de chiens : [984]



Parmi le nombre de chats prédit, 734 sont réellement des chats et il y a 516 de faux positifs. Pour les chiens, 468 chiens sont bien prédits alors que 782 ont été labellisés comme étant des chats. En ce qui concerne, les fonctions de coût, on constate une diminution de la fonction de coût d'entraînement et de validation, cependant on a l'impression que la fonction de coût de validation augmente légèrement. Voyons ce que ça donne si l'on augmente le nombre d'époque.

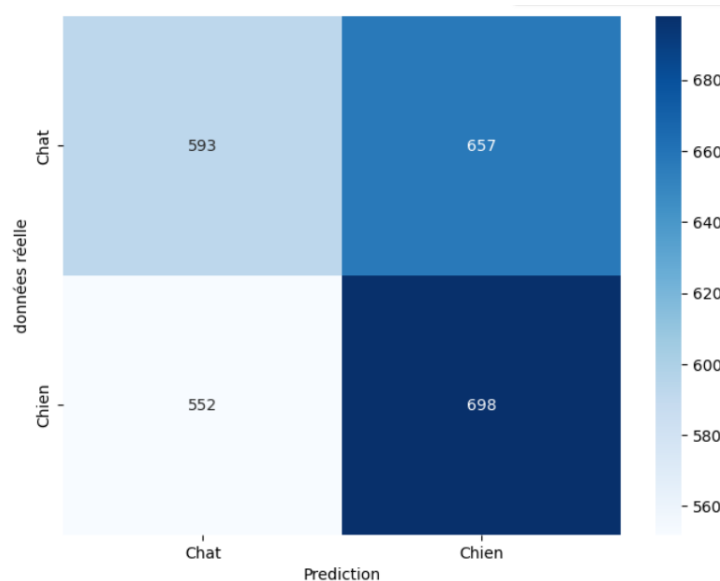
- b. Augmentation du nombre d'époch - epoch : 30 / learning rate : 0.001 /  
Optimizer : RMSProp

Testons le modèle avec 30 epoch et en gardant constant la vitesse d'exécution.



Prédiction > 0.5

```
125/125 [=====] - 11s 89ms/step  
Nombre de chats : [1145]Nombre de chiens : [1355]
```

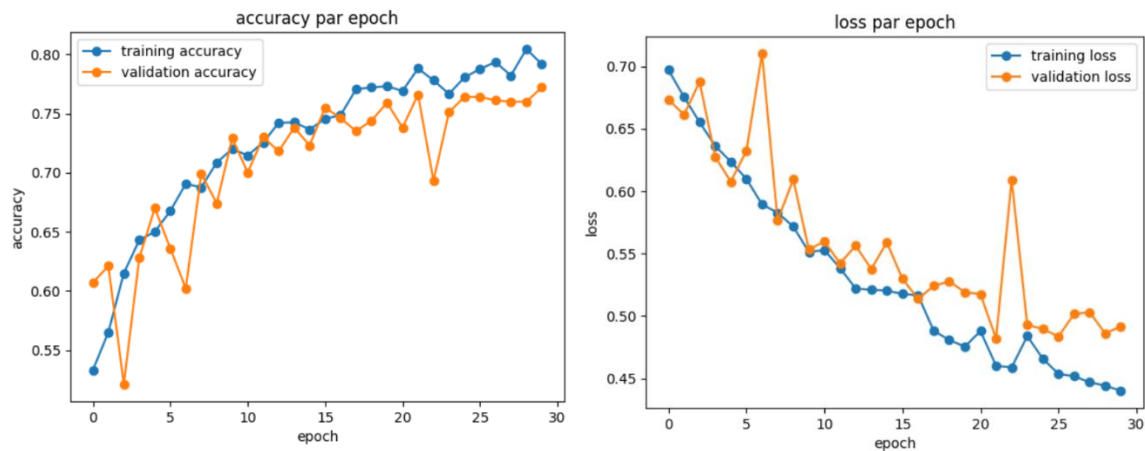


En effet, la fonction de coût de validation stagne, avec des conjonctures qui oscillent. Je n'ai pas l'impression de constater une augmentation nette et précise.

Cependant, on constate que davantage de chiens sont reconnus, mais plus de chats sont confondus avec des chiens.

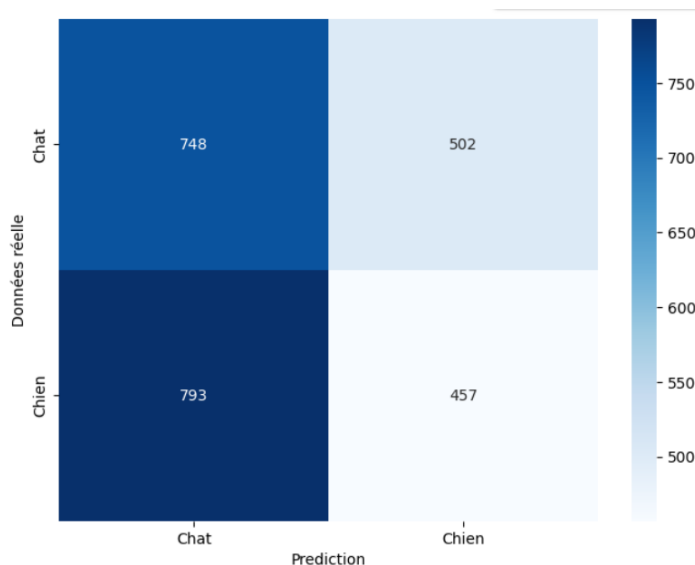
Essayons de diminuer le learning rate.

- c. Augmentation du nombre d'époch - epoch 50 / learning rate 0.001 /  
Optimizer : RMSProp



Prédiction > 0.5

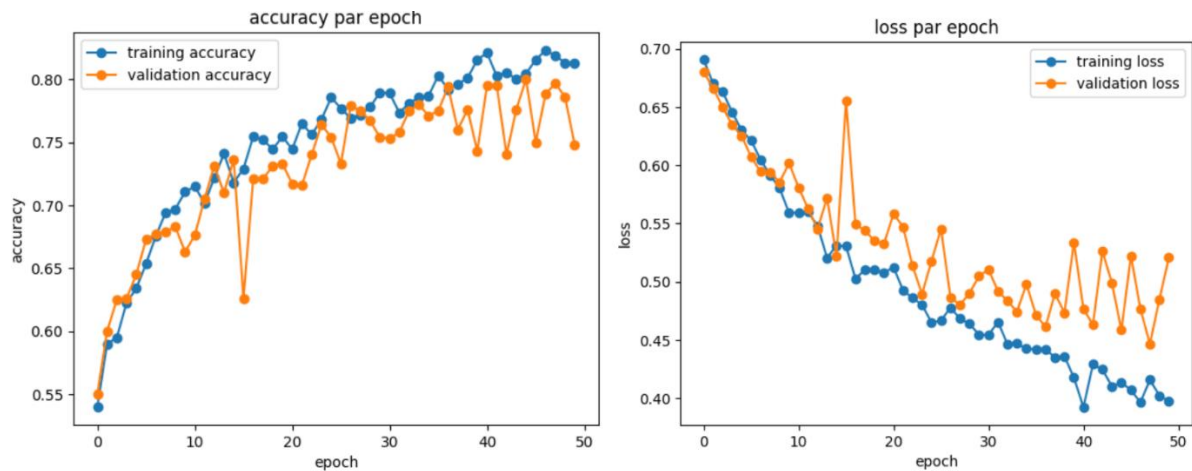
125/125 [=====] - 19s 152ms/step  
Nombre de chats : [1343] Nombre de chiens : [1157]



En modérant la vitesse d'exécution, on remarque que moins de chats sont reconnus comme des chiens. Cependant, il y a une déficience au niveau des chiens. Moins de chiens sont reconnus à leur juste valeur.

En ce qui concerne, les fonctions de coûts, malgré quelques piques, semblent aller vers la même direction. On ne peut cependant pas se prononcer sur un quelconque sur ou sous apprentissage du modèle du fait que le nombre d'époch soit trop faible.

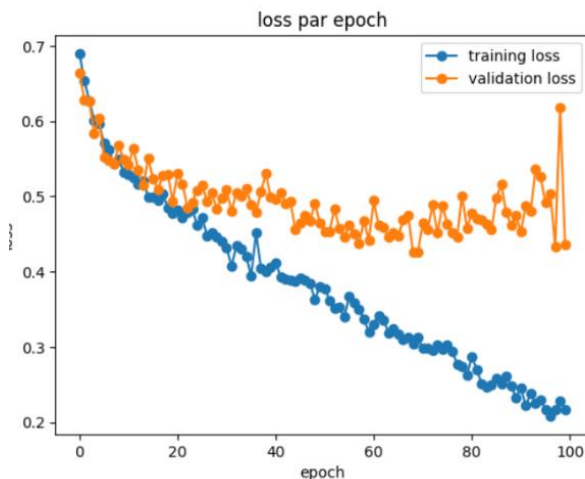
- d. Diminution de la vitesse d'apprentissage et augmentation du nombre d'époch - epoch : 50 / learning rate : 0.0001 / Optimizer : RMSProp



Prédiction > 0.5

125/125 [=====] - 18s 145ms/step  
Nombre de chats : [1509] Nombre de chiens : [991]

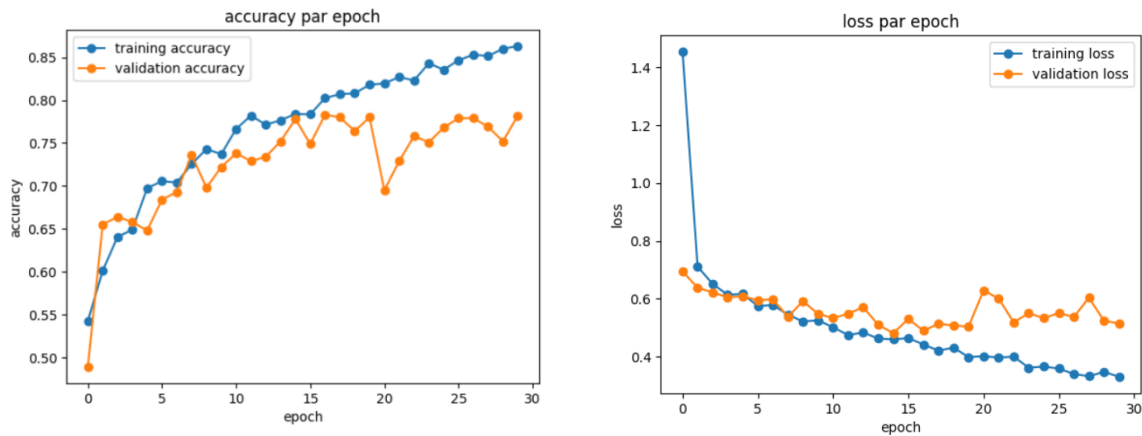
En augmentant le nombre d'époch à 50, on constate de nombreuses oscillations. Cela peut être dû à un problème de base de données ou le modèle peut avoir du mal à apprendre et à catégoriser les images. De plus, la précision de ce modèle est la plus haute que j'ai pu obtenir, pour le moment. On peut en déduire qu'en augmentant le nombre d'époch la précision augmente essayons avec 100 epoch.



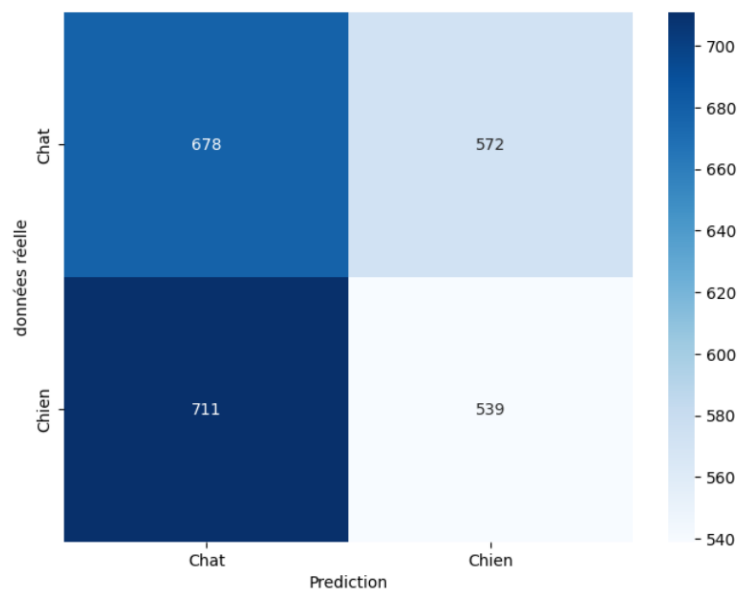
En augmentant le nombre d'époch à 100, on constate clairement que le modèle est dans un cas d'over-fitting. Effectivement, l'erreur du modèle de validation surpasse l'erreur des données d'entraînement, ce qui fait que la prédiction est inefficace. Car le modèle a du mal à reconnaître la catégorie d'image associé.

- e. Réduction du nombre de couche de neurone – epoch : 30 / learning rate : 0.001/ Optimizer : RMSProp

Je tente maintenant de résoudre le problème en modifiant différents paramètres. Dans un premier temps, je réduis le nombre de neurones d'analyse afin de simplifier le modèle.

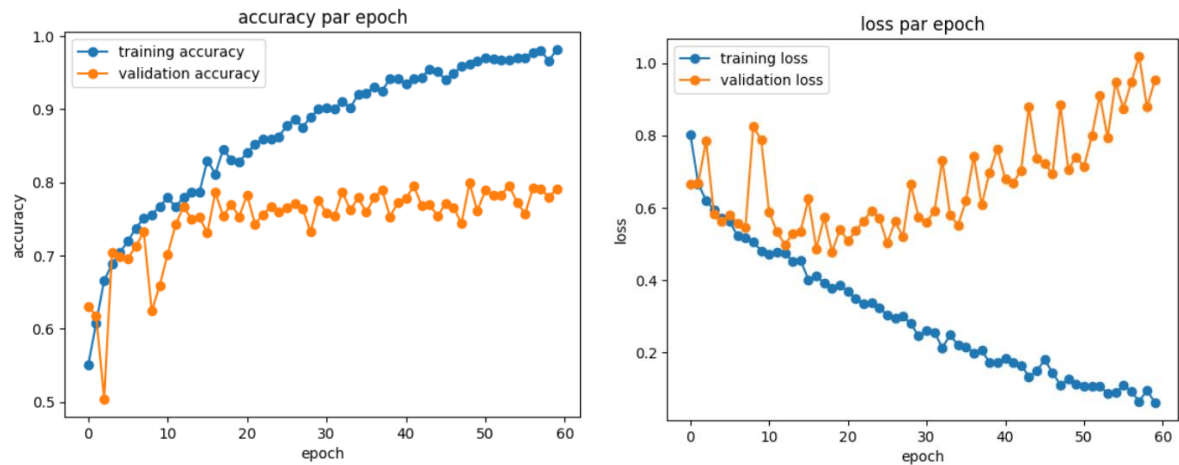


125/125 [=====] - 33s 256ms/step  
Nombre de chats : [1389] Nombre de chiens : [1111]

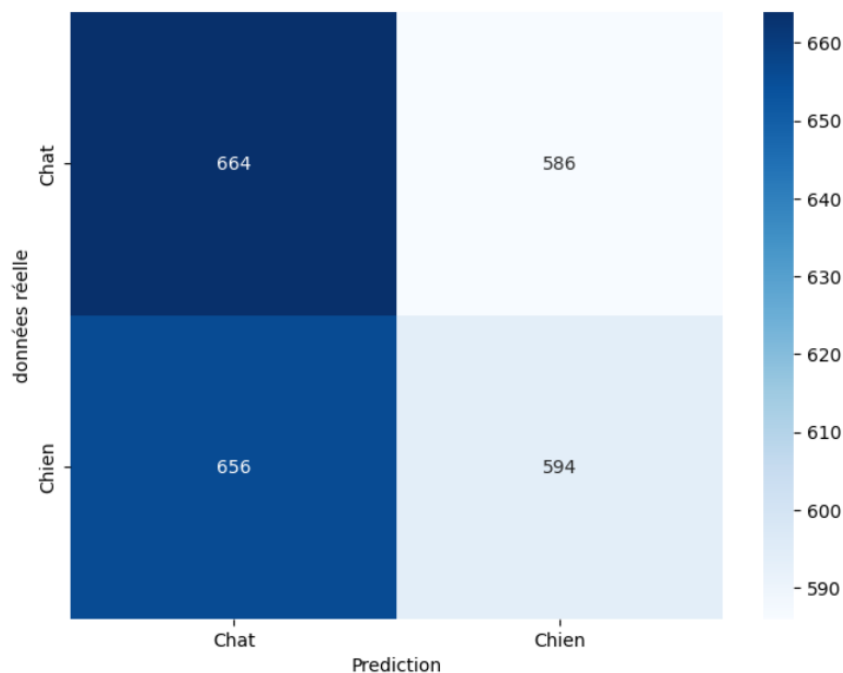


On constate notamment une légère amélioration au niveau de la précision. Mais les chiens ont toujours du mal à être reconnu. En ce qui concerne over-fitting, je ne peux pas me prononcer, je double donc le nombre d'epoch.

- f. Augmentation du nombre d'époch à 60 avec réduction du nombre de couches de convolution.

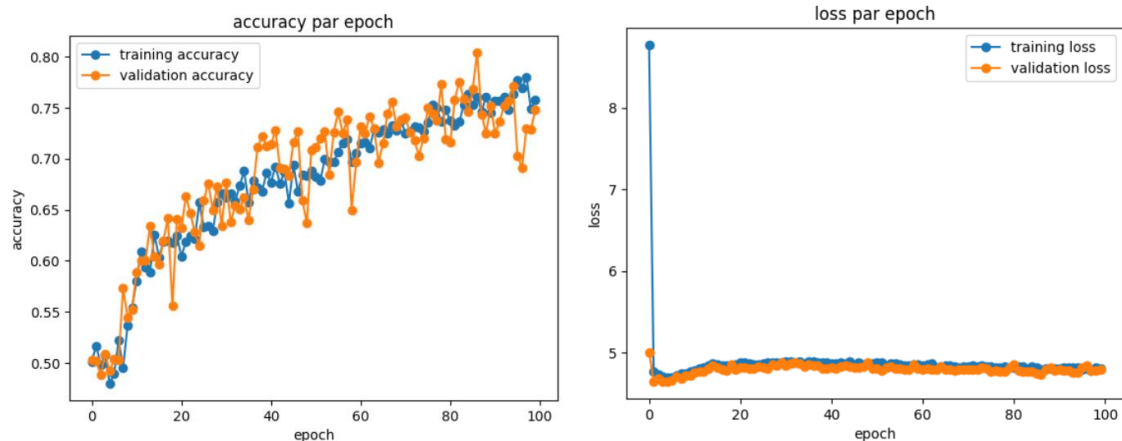


125/125 [=====] - 42s 335ms/step  
 Nombre de chats : [1320] Nombre de chiens : [1180]

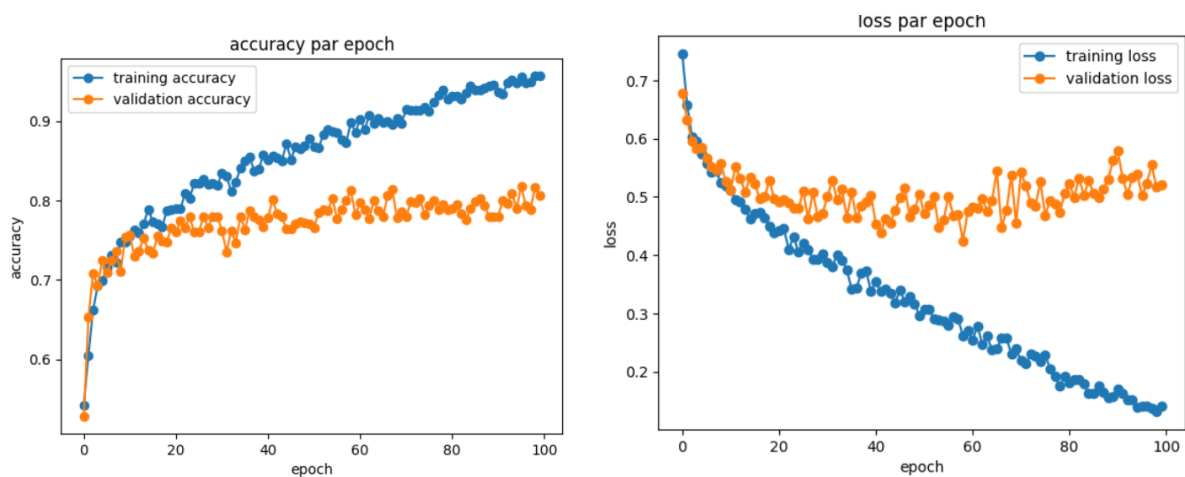


On constate que la précision du modèle a augmenté, cependant nous sommes toujours dans un cas d'over-fitting. De plus, moins de chiens sont reconnus comme des chats.

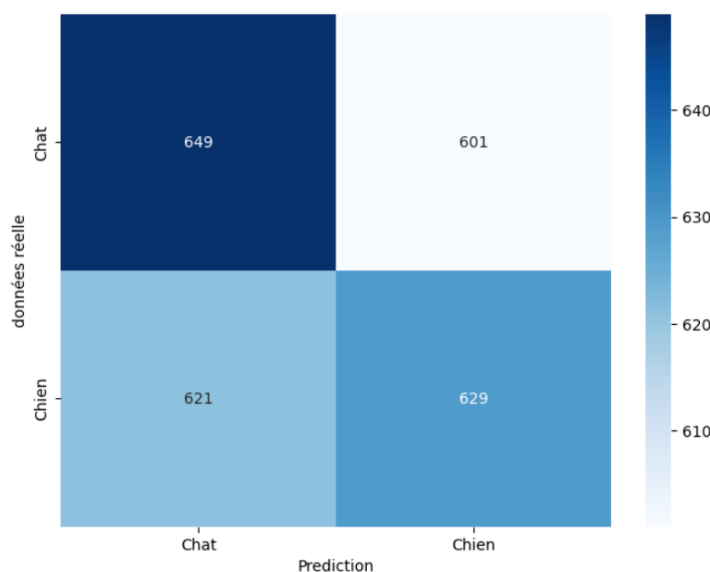
g. Première tentative avec la régulation L1 : epoch : 100 / RMSProp / 0.001



Nous sommes dans un cas extrême d'under-fitting, après de nombreux tests, mon résultat final a été obtenu avec l'optimiseur ADAM,.



125/125 [=====] - 19s 150ms/step  
Nombre de chats : [1270] Nombre de chiens : [1230]



J'ai utilisé l'algorithme d'optimisation ADAM, il me paraît beaucoup plus adaptable que les autres optimizers, du fait de ces nombreux paramètres.

Ainsi, la fonction de coût de validation semble stagner autour de 0,5. La précision est plutôt raisonnable. En ce qui concerne la reconnaissance de chiens et de chats il se trompe autant qu'il a juste. Je reste donc relativement perplexe par rapport à ce résultat.



## Conclusion

Pour conclure, à l'issue des différents paramétrages effectués, je ne suis pas en mesure de vous attester que le modèle a pu reconnaître l'ensemble des chats et des chiens soumis. Notre modèle était au départ trop complexe, ce qui a conduit à de l'over-fitting, soit un sur-apprentissage du modèle d'entraînement. À mon sens, d'après ce que j'ai pu comprendre de cette expérience, lorsque j'augmente le nombre d'époch, j'augmente la précision du modèle, car plus le modèle va être entraîné, plus il est susceptible d'apprendre au mieux. Ainsi, le sur ou le sous-apprentissage va dépendre du paramétrage de notre modèle, soit de sa complexité. Un modèle davantage complexe va entraîner de l'over-fitting puisqu'il va avoir du mal à s'adapter aux données d'entraînement. De plus, lorsque que j'ai tenté d'utiliser la régularisation L1 ou L2, afin d'atténuer le sur-apprentissage, ceux-ci m'ont au contraire provoqué de l'under-fitting. Il fallait donc que je trouve une solution dans le but de diminuer la difficulté de mon modèle, j'ai alors retiré une couche de convolution avec 16 filtres afin de simplifier le modèle. J'ai alors obtenu de meilleurs résultats, peu probants, mais j'ai remarqué une légère stagnation de ma fonction de perte des données de validation. Enfin, la modification de l'algorithme d'optimisation a également permis d'améliorer mon modèle, j'ai la perception, après plusieurs tentatives, que l'optimizer ADAM, s'adapte mieux à toutes sortes de données, il suffit de trouver le bon paramétrage. Malgré la difficulté du paramétrage puisqu'il dispose davantage de variables, c'est avec cet algorithme que j'ai obtenu les meilleurs résultats. De plus, je n'ai pas jugé nécessaire de modifier la fonction d'activation qui me paraît être la plus adaptée à l'exercice Cats and Dogs. Cet exercice m'a permis d'en connaître davantage sur la reconnaissance d'image, qui est tout à fait passionnante. Je pense d'ailleurs intégrer cet outil à mon mémoire de recherche de fin d'années.