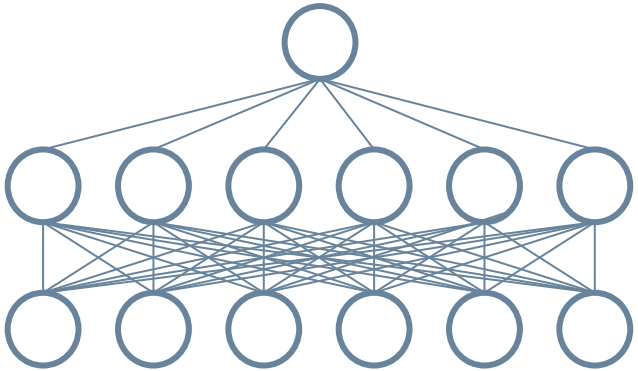
Compléments sur la régularisation et l'augmentation de données

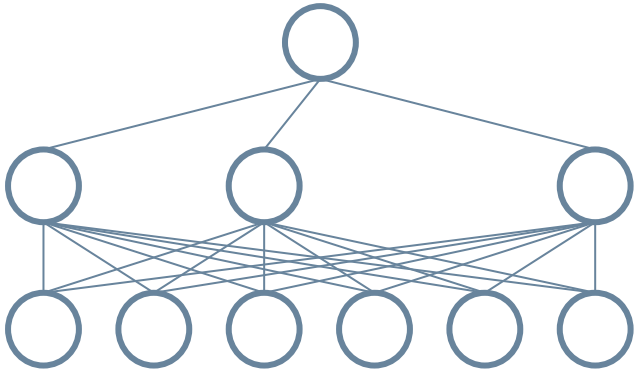
Julien Romero

Le Dropout

Idée : Pendant l'entraînement (uniquement!), supprimer aléatoirement des nœuds.

But : Rendre le modèle plus robuste





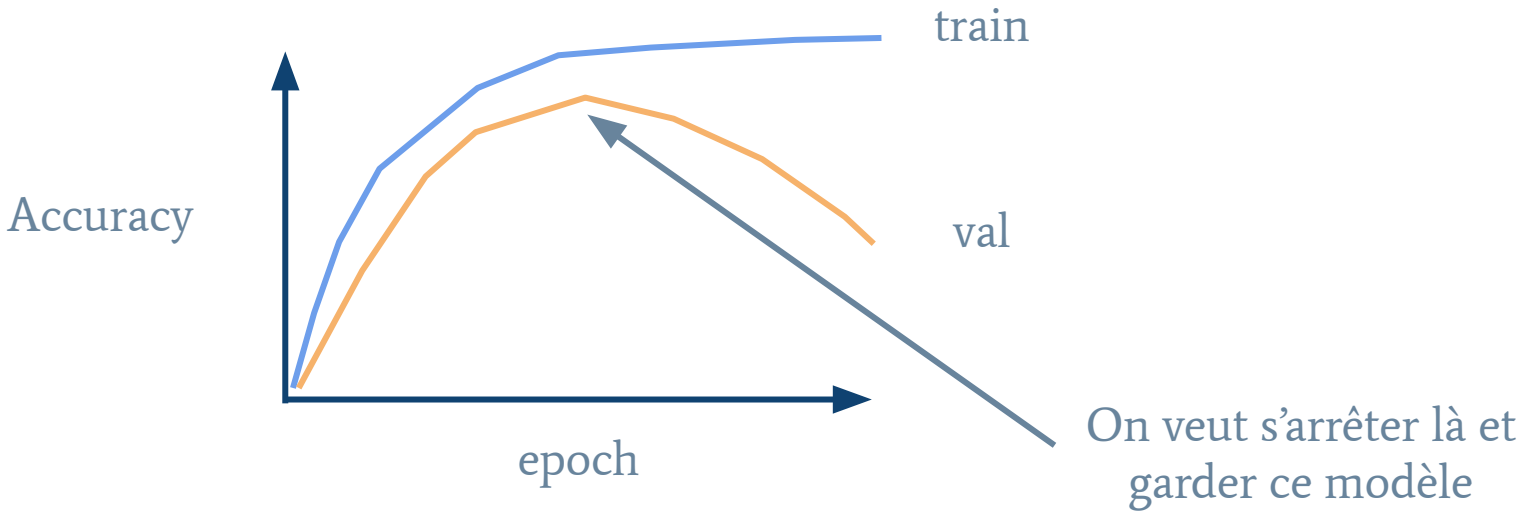
Le Dropout à l'inférence

Au moment de l'inférence (i.e. pas pendant l'entraînement), on désactive le dropout (et on normalise).

En Pytorch, fait automatiquement avec *model.eval()* et *model.train()*

Early Stopping : Quand arrêter l'entraînement ?

Idée : on arrête l'entraînement quand on n'observe plus de progrès sur le validation set.



Early Stopping - Algorithme

```
early_stop_patience = 5
best_val_loss = float('inf')
early_stop_counter = 0
for epoch in range(n_epochs):
    # train + eval sur val
    if val_loss < best_val_loss:
        best_val_loss = val_loss
        early_stop_counter = 0
        # save model
    else:
        early_stop_counter += 1
    if early_stop_counter >= early_stop_patience:
        break
# Load the best model and evaluate on test
```

Preprocessing sur des images

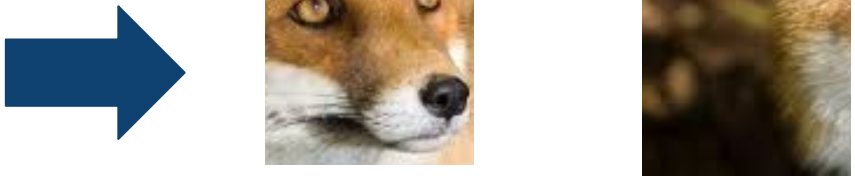
Quand on travaille sur des images, on a souvent besoin de les transformer avant de les donner à notre réseau de neurones.

- Resize : On redimensionne à la bonne taille (taille d'entrée de notre réseau en général)
- Crop : On découpe une zone de la bonne taille
- Normalisation (classique) : on calcule souvent les moyennes et écarts-types en avance

Preprocessing sur des images - Resize

Si on garde les ratios, on n'obtient pas forcément une image carrée de la bonne taille.
Sinon, on déforme l'image.

Preprocessing sur des images - Crop



On obtient un image non déformée de bonne taille, mais on risque de couper de l'information utile

Augmentation de données

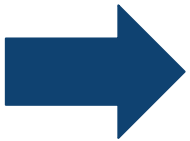
Idée : Peut-on créer des points d'entraînement supplémentaires "gratuitement" en transformant les données déjà existantes ?

But : Cela rend le modèle plus robuste aux invariants. Plus on a de données, meilleur est le modèle en général.

Dans le cas des images :

- Recadrage aléatoire : on prend un morceau aléatoire de l'image
- Horizontal Flip : on fait le miroir de l'image
- Rotation, changement de couleurs, déformations, combinaisons d'images

Augmentation de données - Horizontal Flip



Un renard

Toujours un renard