OVERFITTING

Rule of thumb when splitting dataset:

70% for training
15% for validation
15% for test

Special case: Correlated data

Data like images from a video

- Correlated data is bad when we need a model that generalize outside the correlated data (i.e most models)
- Correlated data is good when we need a model that generalize inside the correlated data (ie auto-labeling system)

Is overfitting really bad ?

The gap between training and validation loss/accuracy is not by itself bad.

It only considered bad if the validation loss/accuracy start diminishing.

Early Stopping

- Stop right before overfiting (diminish validation loss/acc)

Data Augmentation

Because overfit happens only when seeing the same data multiple times. In other words, we cannot overfit if we never see the same data twice.

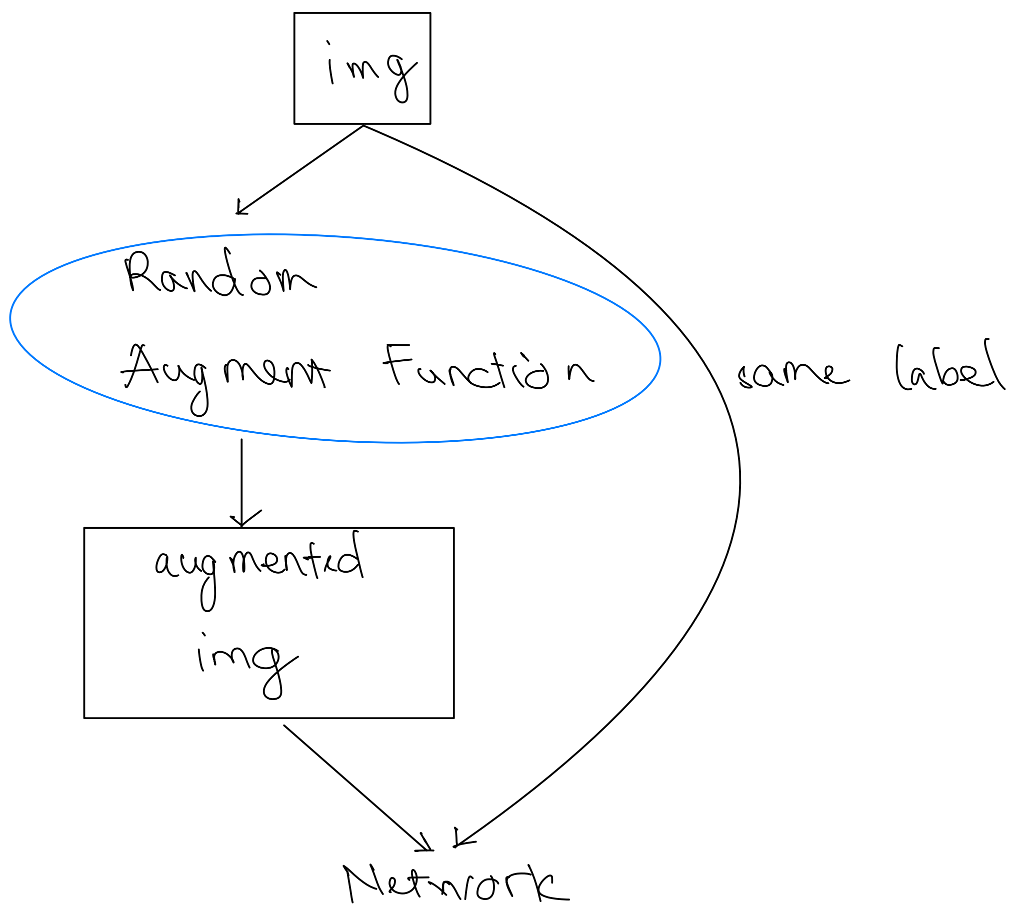
So we can either:

- Have infinite data (not practical)
- "Pretends" that we have infinite data → data augmentation

- What is augment data ?

1. Make more data from our existing data
2. Randomly transform data during training
3. Reuse / rephrase labels

- How to augment data ?



Transfer Learning

If after you augment your data, and you still find that your model is not performing good enough (cause not enough data). You can try transfer learning

What is transfer learning ?

- Take existing model
- Fine-tune it (continue traing on your dataset)

When to transfer learning ?

- Unless you works on training large model (LLM, etc...) you should use transfer learning almost always.

Dropout

Randomly set activations to zero.

Why does this work ?

Because deeper layers rely on output activations from previous layers. And previous layers tend to overfit first (because its closer to the data)

→ Randomly "turn off" some of these activations will reduce reliance on specific activations

Where to add Dropout?

- Before any large fully connected layer
- Before some 1x1 convolutions
- Not before general convolutions
  - ↳ Because receptive fields inside convolutions can generate enough correlation to undo the effect of Dropout.

Intuition of preventing overfitting

1. Keep model small

- Pros: overfit less
- Cons:
  - Fit worse
  - Generalize worse

2. Big Model with Regularization

Regularize using Weight Decay:

- Keep weights small (L2 norm)
- Keep weight at the same magnitude

⇒ In practice, weight decay doesn't actually prevent overfitting. But it does prevent your weight exploding.

3. Ensembles

- Train multiple smaller models and average the outputs (kinda like Random Forest)
- Cons of this is it uses more computation.