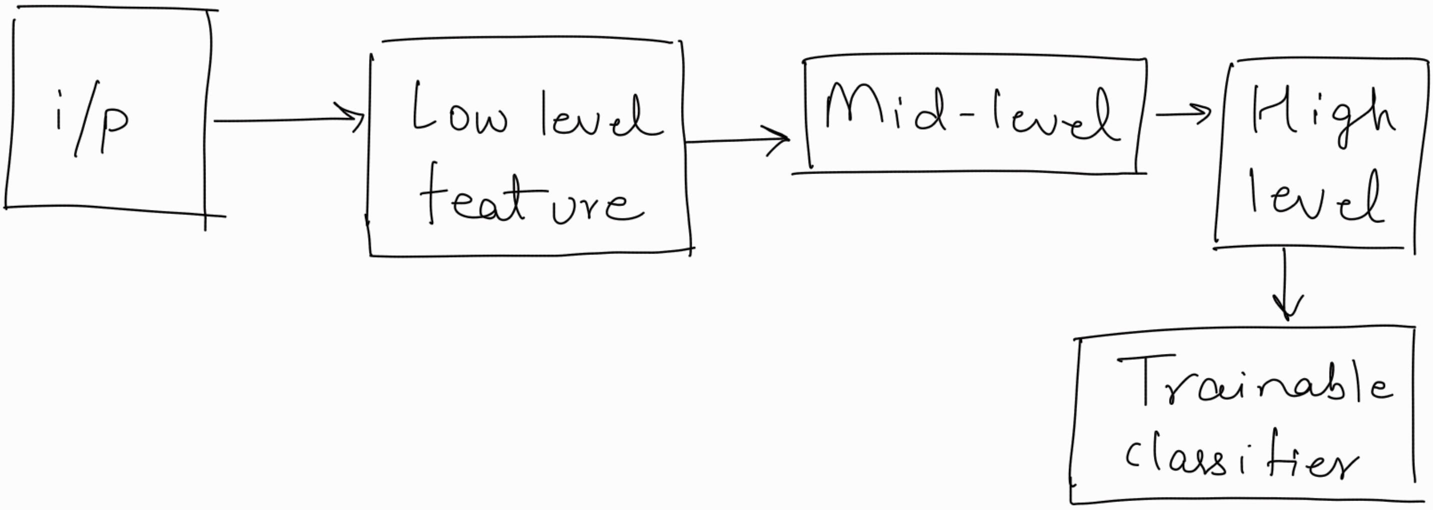
1/3/24

Initial filters are low-level feature capturing.

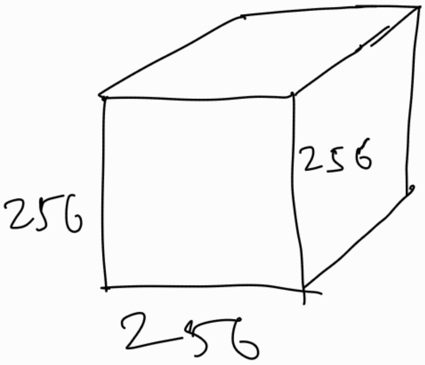


Transfer Learning (Fine Tuning)

Remove the last layer and retrain on your outputs.

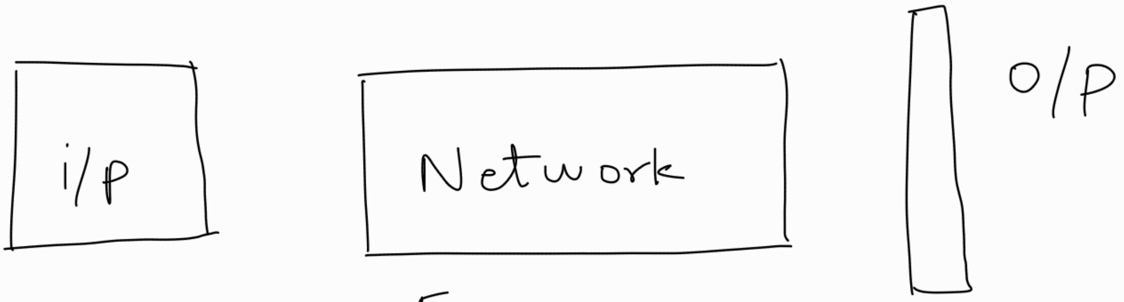
You get a good start as earlier weights are already trained. Loss minimizes early.

3D convolution : The filters move in x, y, z direction. A filter gives multiple outputs.



3x3x64 single filter gives 4 values.

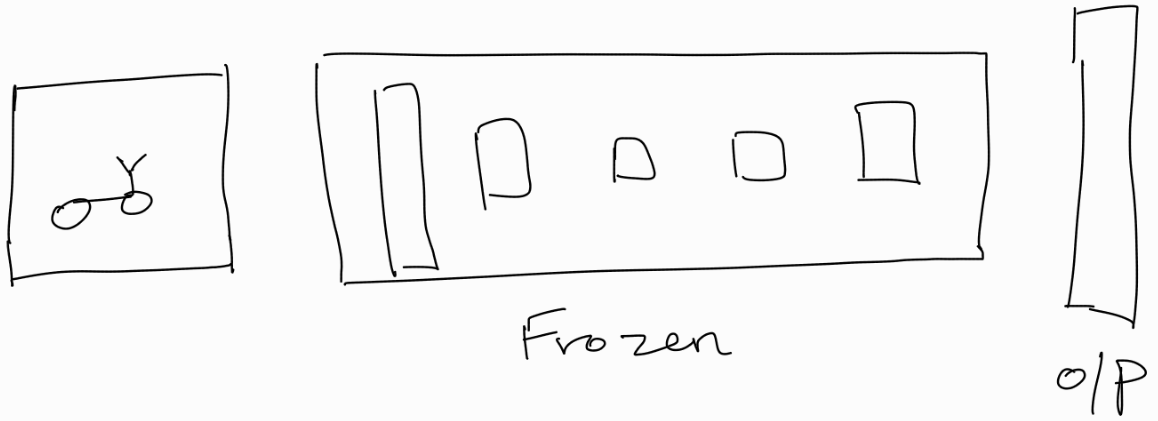
NNs actually learn!



Frozen weights

You start with a random image, classify as panda, backprop and reconstruct the image. Results prove that NNs actually learn image features.

Adversarial Attacks

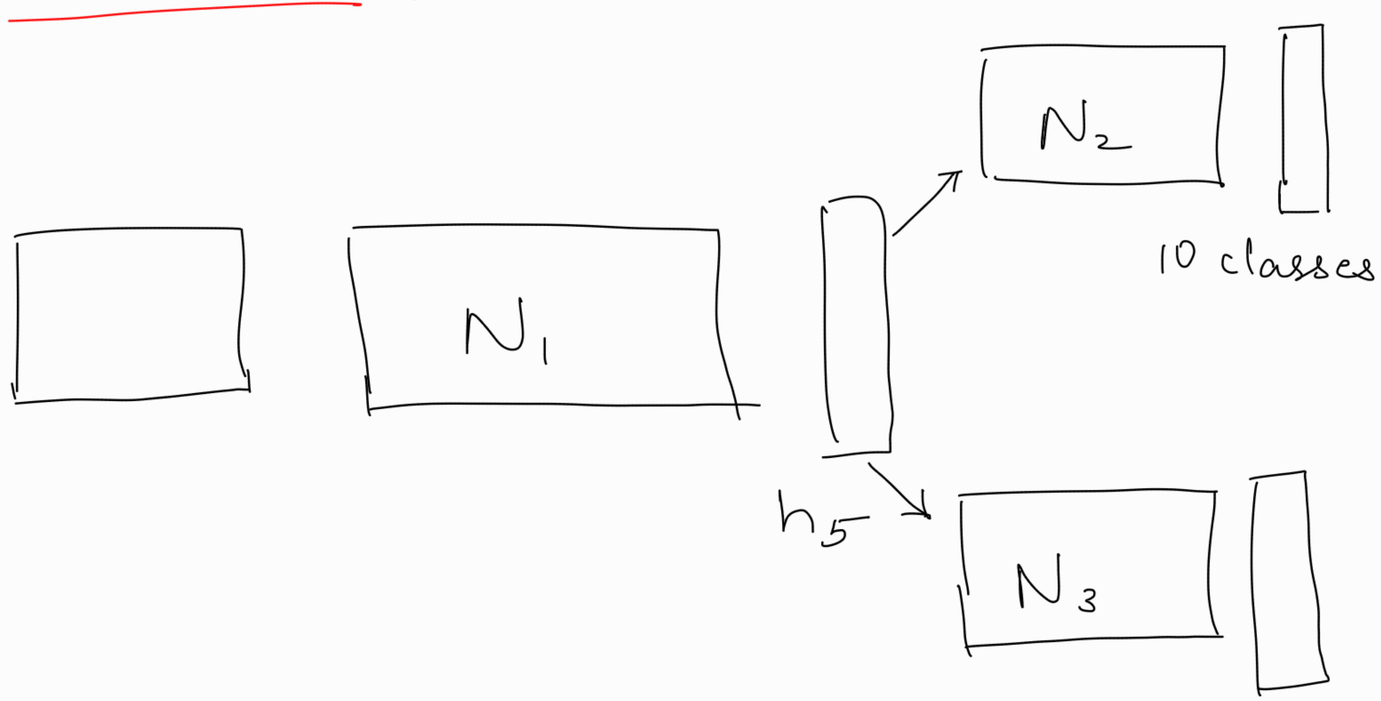


→ i/p is a cycle. You want to classify as panda.

→ Assume o/p is panda with 0.01 and cycle with 0.89. Backprop with cross entropy loss of 0.01 and reconstruct image

→ We find that the visual difference between the changed and original image is negligible but model confidently classifies as panda.

Gradient Reversal



4 domains
↓
sketch
painting
photo
cartoon

→ You want the network to classify classes correctly agnostic to domains.

→ You don't want the network to do well on N3

→ You do gradient ascent instead of descent

→ Force the network to do well in what

you want

→ h5 will have an idea about classes but not about domains.