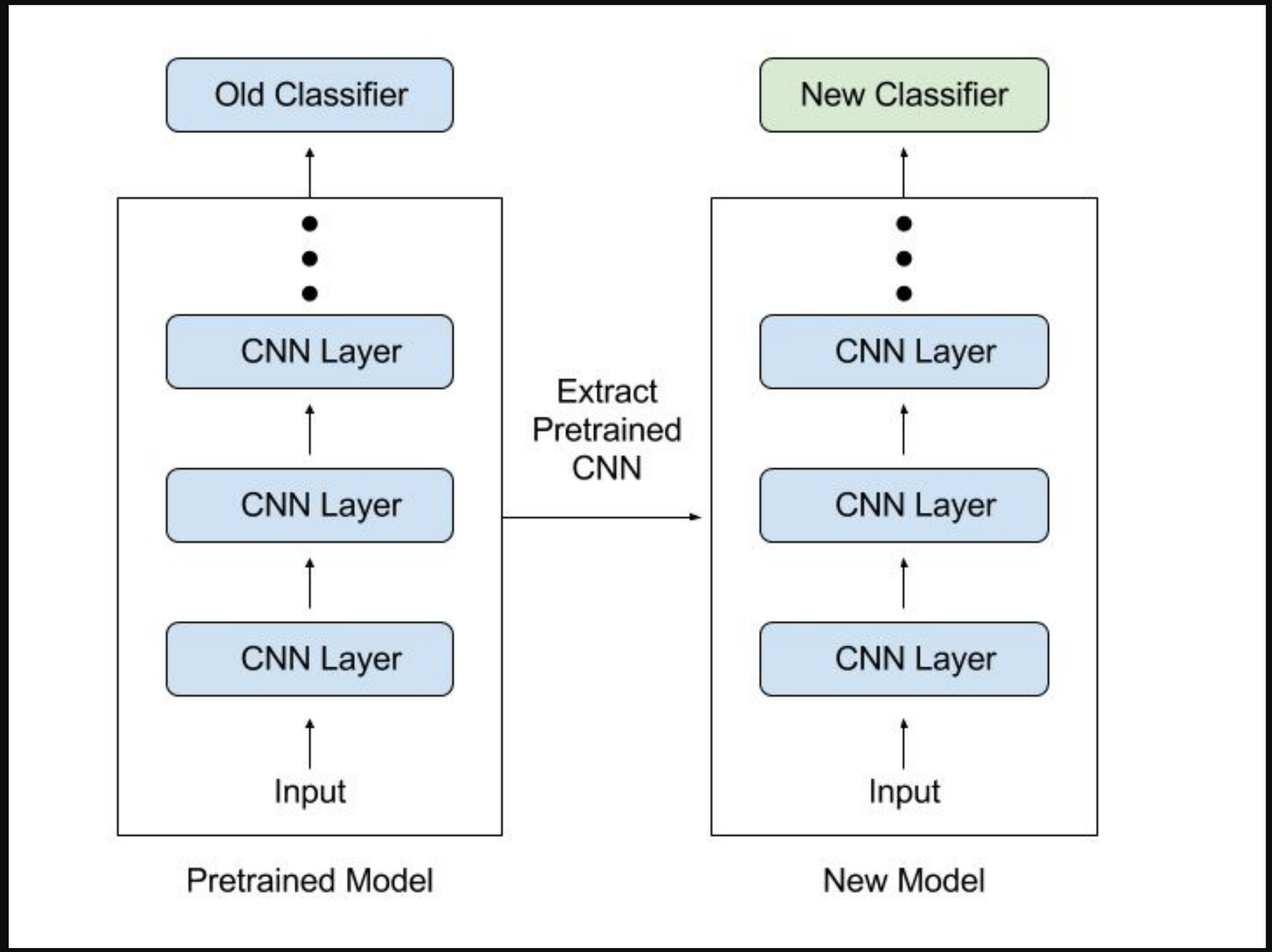
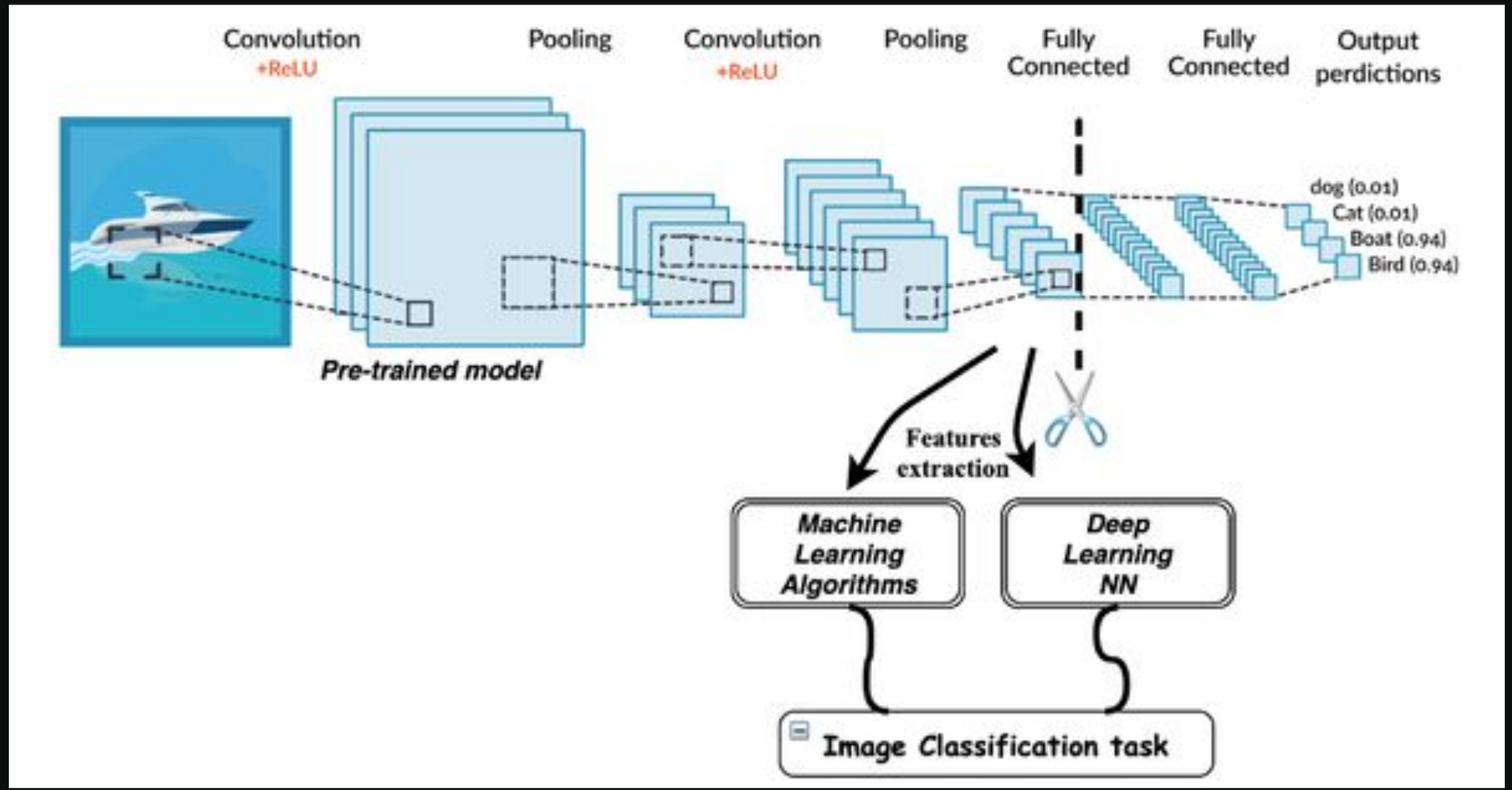
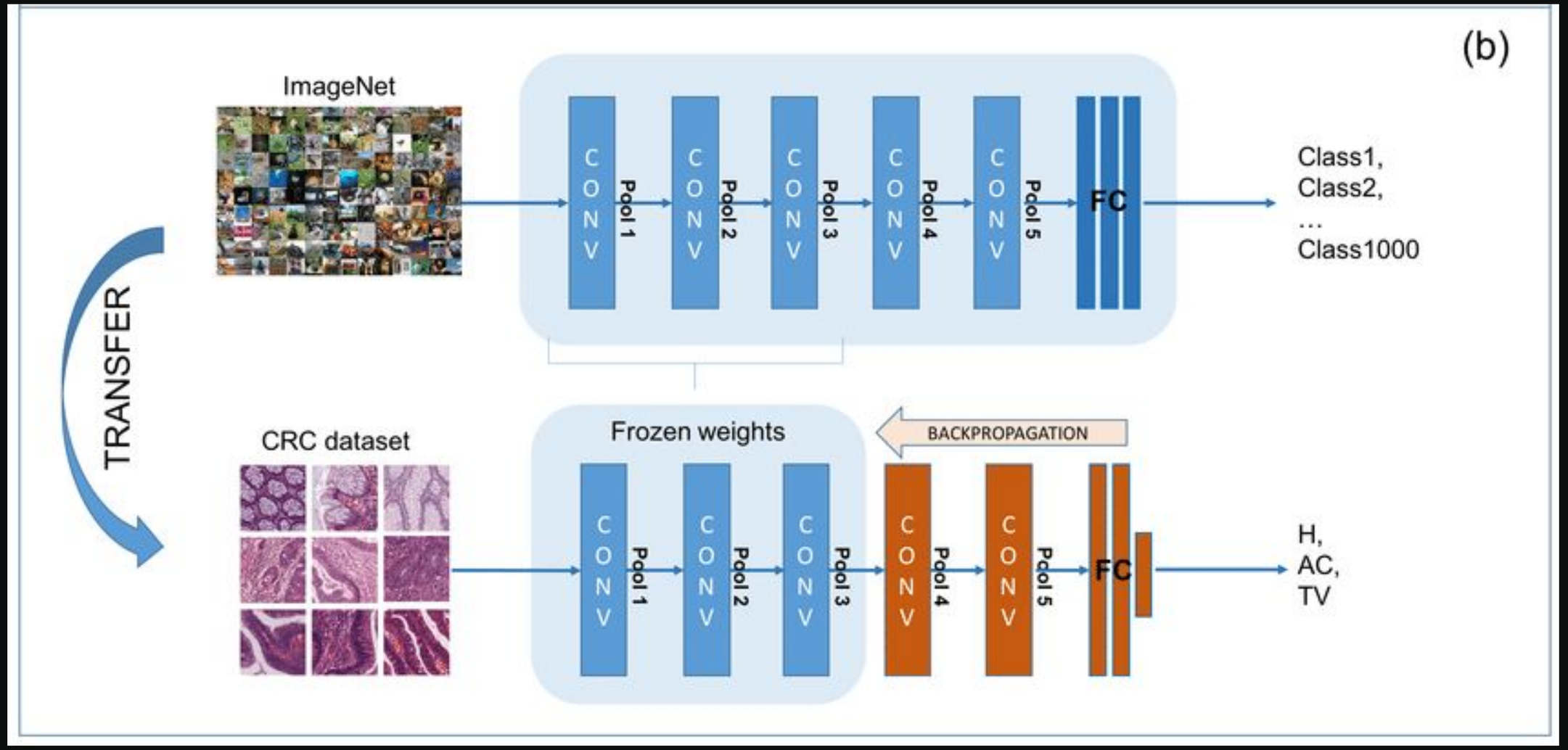

Strive
SCHOOL

/ Transfer Learning

Learn the feature extractor for a task, train the classifier for another task


Old Classifier
CNN Layer
CNN Layer
CNN Layer
Input
Pretrained Model
Extract
Pretrained
CNN
New Classifier
CNN Layer
CNN Layer
CNN Layer
Input
New Model


Convolution
+ReLU
Pooling
Convolution
+ReLU
Pooling
Fully Connected
Fully Connected
Output perdictions
dog (0.01)
Cat (0.01)
Boat (0.94)
Bird (0.94)
Pre-trained model
Features extraction
Machine Learning Algorithms
Deep Learning NN
Image Classification task


(b)
ImageNet
CONV
Pool 1
CONV
Pool 2
CONV
Pool 3
CONV
Pool 4
CONV
Pool 5
FC
Class1,
Class2,
…
Class1000
TRANSFER
CRC dataset
Frozen weights
BACKPROPAGATION
CONV
Pool 1
CONV
Pool 2
CONV
Pool 3
CONV
Pool 4
CONV
Pool 5
FC
H,
AC,
TV

/ When to apply Transfer Learning

/ Convolutional Neural Network as a fixed feature extractor

Remove the fully connected layers from the pre-trained convnet trained on imagenet data.

Freeze the rest of the convnet layers to be used as a feature extractor on the new data.

Train a linear classifier like Softmax for the new dataset on extracting the features for all images.

/ Fine-tuning the CNN

Retrain the classifier on the new dataset and tune the weights of the pre-trained model by resuming backpropagation.

Either tune all the layers of the ConvNet or tune the high-level portion of the network and keep the rest of the layers fixed.

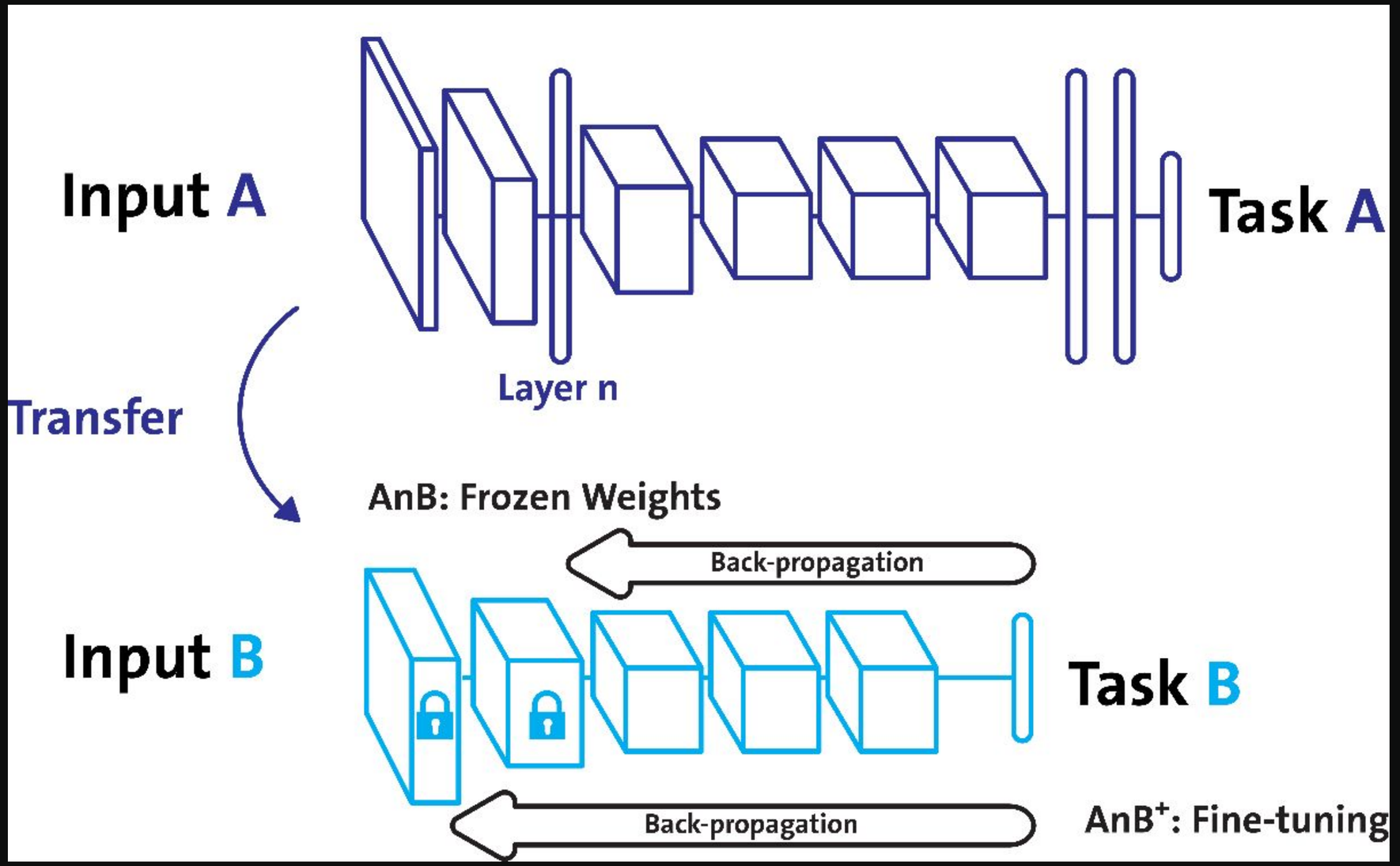
Before convnet used to take more generic features like the edge that was handy in many problems, but later on layers become steadily more particular to the information about the classes that are present in the original data.

/ When to apply Transfer Learning

/ Pretrained models

ConvNet can take several days to get trained. Many have delivered their final convent checkpoints open source so that it is beneficial for others.

Those weights can be used and fine-tuned to save training time.


Input A
Task A
Layer n
Transfer
AnB: Frozen Weights
Back-propagation
Input B
Task B
Back-propagation
AnB+: Fine-tuning

/ Strategies

/ You have a very small target dataset that is similar to the training dataset

Fine tuning the entire model could result in overfitting (small data, big model)

Solution: freeze the weights of the Convolutional layers to extract the features and retrain only the fully connected layers (with the correct number of classes)

/ You have a big target dataset that is similar to the training dataset

Replace the Fully connected layer (the classifier) with a new one with random weights and with the correct amount of outputs

Train the entire model

/ Strategies

/ You have a small target dataset that is very different from the training dataset

Since the task is different, only the low-level features are meaningful (edges, small shapes…) that you can find at the first convolutional layers.

Solution:

Freeze the first convolutional layers, drop the high-level convolutional layers and put a fully connected layer on top of it

/ Strategies

/ You have a big target dataset that is very different from the training dataset

Since the task is different, only the low-level features are meaningful (edges, small shapes…) that you can find at the first convolutional layers.

Solution:

Replace the fully connected layer with a classifier of your choice and retrain everything

/ Strategies

Input Image

Convolution → ReLU rectified linear units → Pooling → Convolution → ReLU rectified linear units → Pooling → Convolution → ReLU rectified linear units → Pooling • • • Convolution → ReLU rectified linear units → Pooling → FC Fully Connected layers to support classification → Flower, Cup, Car, Tree