1. Creating json files from AODSIM root files

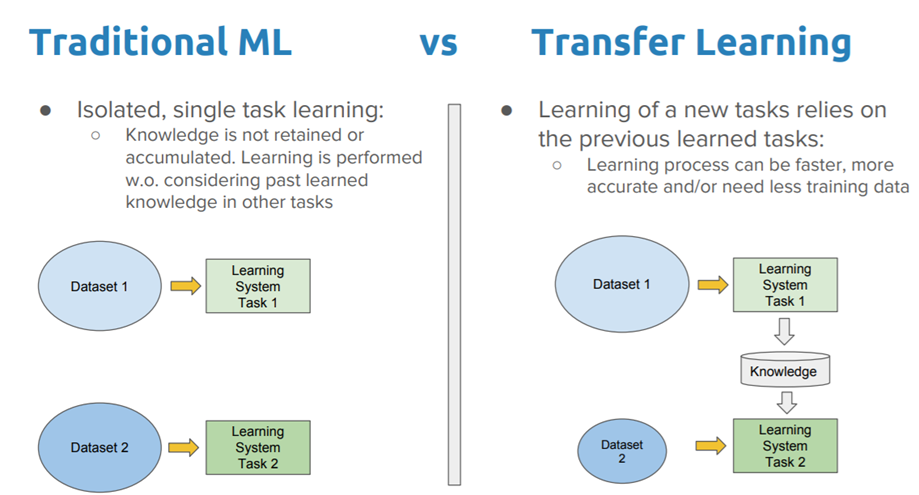
Collisions, also known as events, recorded in a HEP experiment by a detector like CMS. Ana these events are described by a set of variables: the momentum of muons, electrons, photons and hadrons produced in the collision of the two accelerated protons, which have been released as Open Data by the CMS collaboration.

**Using events corresponding to simulated collisions at 7 TeV at LHC recorded by the CMS detector**, we want to select collisions where one of the W bosons decays leptonically into a charged lepton, electron or muon, with an associated neutrino. These events provide a clear experimental signature, with an isolated lepton with high-transverse momentum, hadronic jets and a large missing transverse energy. We have considered as background processes the production of events where a W boson is produced in association with additional jets (W + jets events) and events corresponding to the so-called Drell-Yan processes. **All these samples are obtained from the CMS Open Data portal with AODSIM format.**

**After downloading the index files respectively, we can create links to the database and download these open data then create json files.**

1. Creating images from json files
2. Using transfer learning techniques to classify images

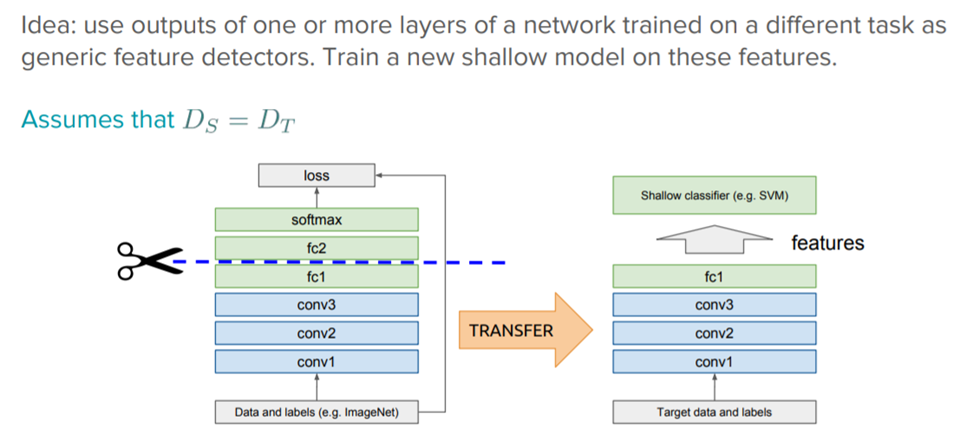
Normally people using deep learning with convolutional neural networks to solve visual recognition problem, because compared with traditional ML transfer learning is not isolated, which means you can leverage knowledge (features, weights etc) from previously trained models for training newer models and even tackle problems like having less data for the newer task as the picture showing below.



Deep Transfer Learning Strategies:

Deep learning systems and models are layered architectures that learn different features at different layers (hierarchical representations of layered features). These layers are then finally connected to the last layer (usually a fully connected layer, in the case of supervised learning) to get the final output. This layered architecture allows us to utilize a pre-trained network (such as Inception V3 or VGG, **we’re using VGG in our code**) without its final layer as a fixed feature extractor for other tasks.

The key idea here is to just **leverage the pre-trained model’s weighted layers to extract features** but not to update the weights of the model’s layers during training with new data for the new task( as the picture showing below).



1. Conclusion

The results presented in this study show that the use of transfer learning with CNN could be a promising tool to classify collisions in particle physics analysis.