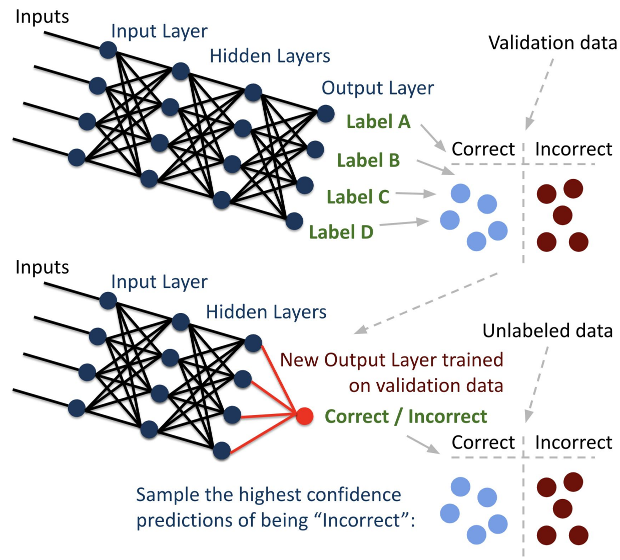
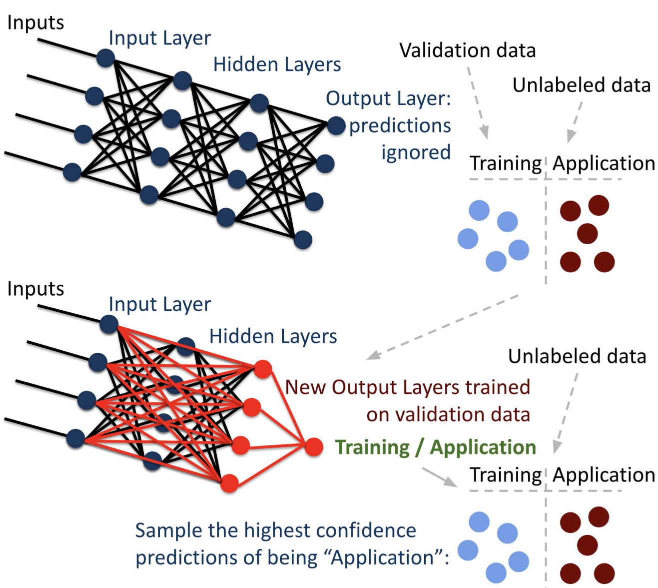
Active Transfer Learning Cheatsheet

Supervised Machine Learning models can combine Active Learning and Transfer Learning to sample the optimal unlabeled items for human review. Transfer Learning tells us whether our model will correctly predict the label of an item and which item looks most like data from our application domain. This cheatsheet builds on principles of Uncertainty Sampling & Diversity Sampling: http://bit.ly/uncertainty_sampling | http://bit.ly/diversity_sampling

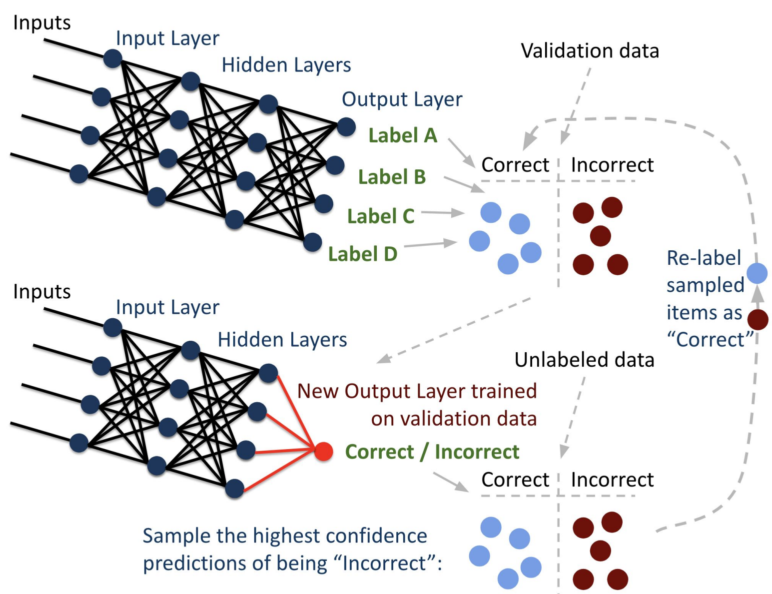
Active Transfer Learning for Uncertainty Sampling: Validation items are predicted by the model and relabeled as "Correct" or "Incorrect" according to whether they were predicted correctly or not. The last layer of the model is then retrained to predict whether items are "Correct" or "Incorrect". Unlabeled items can now be predicted by the new model as to whether our initial model will give "Correct" or "Incorrect" predictions, and we sample the most likely to be "Incorrect".





Active Transfer Learning for Representative Sampling: To adapt to new domains, we retrain our model to predict whether an unlabeled item looks more like validation data from the distribution of our current training data or data from our application domain. ***Tip:*** allow the new model to see all layers to minimize bias from your current model state.

Active Transfer Learning for Adaptive Sampling (ATLAS): We can make our models adaptive by assuming that our items will get a human label later, even if we don't yet know what that label will be. We assume that our model will predict items like these correctly after it has been trained on them. So, we can continually retrain our model with our samples. ATLAS therefore address both Uncertainty Sampling & Diversity Sampling in a single adaptive system.



Robert Munro. **Human-in-the-Loop Machine Learning**, *Manning Publications*. http://bit.ly/huml_book
See the book for more details on Active Learning, Advanced Active Learning, and Active Transfer Learning, with open source implementations in PyTorch.
robertmunro.com | @WWRob