Application for the special "HEP meets ML" award from participant who finished on 5th place.

The problem

As we know, the main challenge and obstacle of " Flavours of Physics: Finding $\tau \to \mu\mu\mu$" competition was requirement for a model to satisfy two additional tests on control channel, where a model was not trained. These tests, by concept of respected organizers, were to confirm robustness of a model against simulated data on signal channel where a model was originally created. Meanwhile, it has been shown that the underlying probability distributions significantly differ between the two channels, therefore **violating the assumption of a single underlying generative process between the train set and the test set, which is the cornerstone of statistical learning and, in general, of inference**. Practically this means that KS and CVM tests on control channel with different distribution provide no additional guarantee [in statistical sense] for a model, neither on signal channel where a model was trained, nor on control channel where a model was not trained! Simply put, drawing conclusions from another distribution that did not participate in statistical inference is irrelevant. *(keywords: Learning Theory, Generalization, Hoeffding inequality, PAC assumptions, VC dimension*)

This problem was noticed by several participants (e.g. see **fchollet** here and truly yours here). As an experiment I implemented a simple model, combination of strong classifier and random noise, that easily reaches top score, satisfies all competition requirements (no training on control data, pass all tests, treat both channels as the same). Yet by construction it is meaningless on control channel where noise classifier de facto works. And although it passes KS/CVM tests on control channel with noise classifier, these tests make no sense for signal channel which is controlled by strong classifier. To get the idea, please see this discussion. This is an extreme example, but as follows from forum discussions, many participants were forced to combine strong and underperforming classifiers only to pass KS/CVM tests.

Proposed solution

The idea of attesting model on control channel is certainly reasonable, but should be implemented in theoretically sound way. In Machine Learning the problem of different data sets is well known, and solution is called **Transfer learning.** It aims at transferring knowledge from a model created on the train set to the test set, assuming they differ in some aspects, e.g. in distribution. Similar problems often arise in medicine or brain-computer interface where single model never fits well all subjects. The solution is to train general model on average population (big, reliable, well studied data) and transfer it to individuals with adaptation.

For the challenge I implemented 2-stage process. At the **first stage** I create strong model for signal channel using all available features. This model is based on ensemble of 20 feed forward fully connected neural nets. Dropout is applied to avoid overfitting. At **the second stage** I transfer this model to control channel. The model's output is stacked with original features and cascaded to additional "transductive" neural network of similar configuration. The purpose of the second net is to **track** the first model's output on signal channel with **minimal and controlled adaptation** to control channel.

Training transductive network. Training on control channel was prohibited, but calculation KS/CVM allowed. The other metric to satisfy is AUC on signal channel. All 3 metrics are global and not analytically differentiable; this makes gradient descent generally impossible. The procedure is the following. Initial weights of transductive network are set to reproduce output of original model; this is accomplished after 1-3 epochs of standard GD with cross entropy loss on

signal channel. Adaptation of weights is done with stochastic optimizer using Powell's method. Loss function incorporates AUC, KS, CVM metrics and allows control them during optimization. As a result we obtain best of the two worlds: performance on signal channel preserved as mush as possible (slightly drops only in the 3rd decimal place), the tests on control channel passed. To keep the model as physically sound as it was created, we can control its performance on signal channel during optimization. Furthermore, we can restrict transductive network from excessive deviation from its original state (weights), or implement any other regularizer. Most importantly, this framework provides guarantee that the model satisfies target metrics (AUC, KS, CVM) not by coincidence, ad hoc or tricks but due to statistical inference.

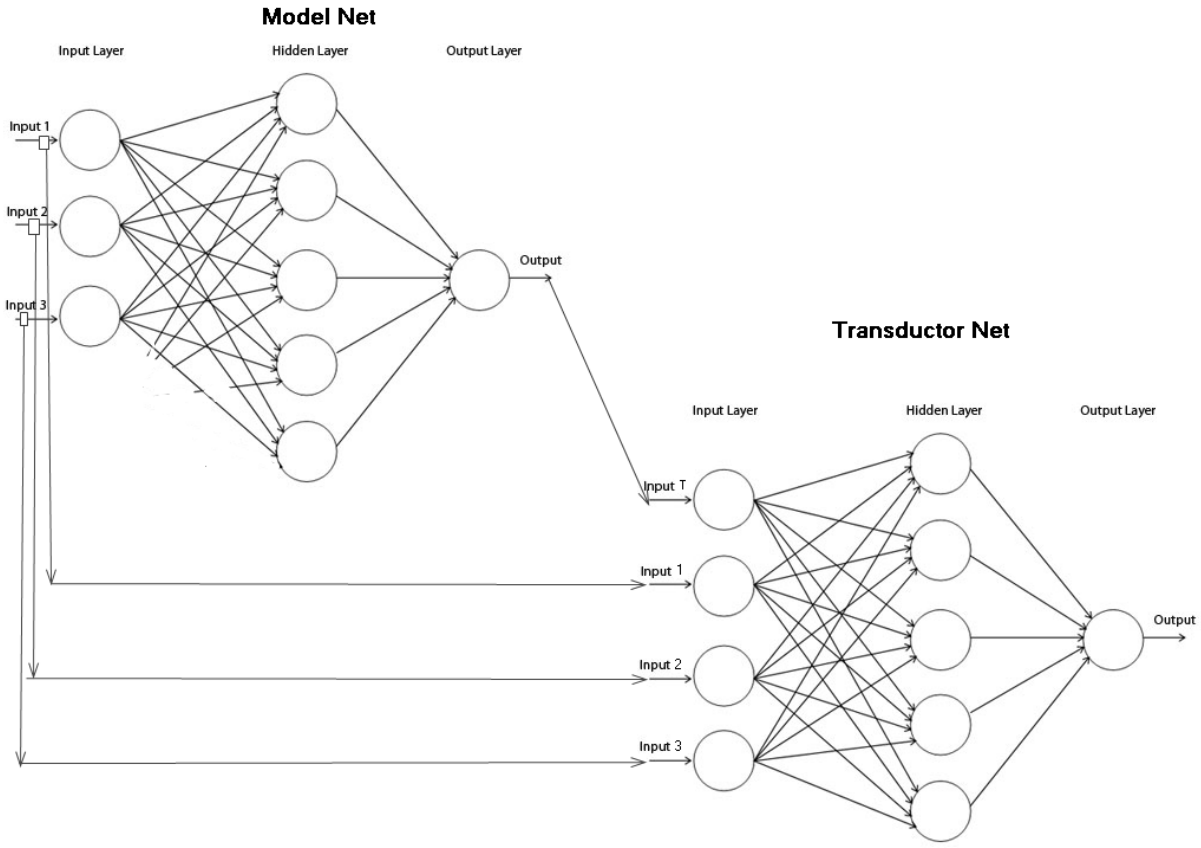

Fig.1 Simplified scheme of transductive inference.

Remarks

- For the competition I submitted 2 different models. Transductive model is the one that achieved Public Score 0.996802 on LB (it was improved to 0.998150 later on).
- Because training on control data was banned, **t**ransfer learning goes in unusual direction from less reliable data set (according to competition's admin, signal channel considered less reliable) to more reliable control channel. Reverse direction would be more natural.

In closing

I want to thank respected organizers for the arguably most funny and intense Kaggle competition I participated so far. I propose my solution for special "HEP meets ML" award in hopes it is of benefit for HEP. Furthermore, I suggest you to incorporate **t**ransfer learning as general framework into modeling rare events and unreliable data sets from models built on data with more well-known and well-observed behavior.

Literature:

Olivetti, E., Kia, S.M., Avesani, P.: Meg decoding across subjects. In: Pattern Recognition in Neuroimaging, 2014 International Workshop on (2014)

S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” Knowledge and Data Engineering, IEEE Transactions on, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

B. Zadrozny, “Learning and Evaluating Classifiers Under Sample Selection Bias,” in Proceedings of the Twenty-first International Conference on Machine Learning, ser. ICML ’04. New York, NY, USA: ACM, 2004, pp. 114+.

Source code and brief description of the solution can be found at https://github.com/alexander-rakhlin/flavours-of-physics