**BANN progress so far and additional experiments**

Alex DongHyeon Seo

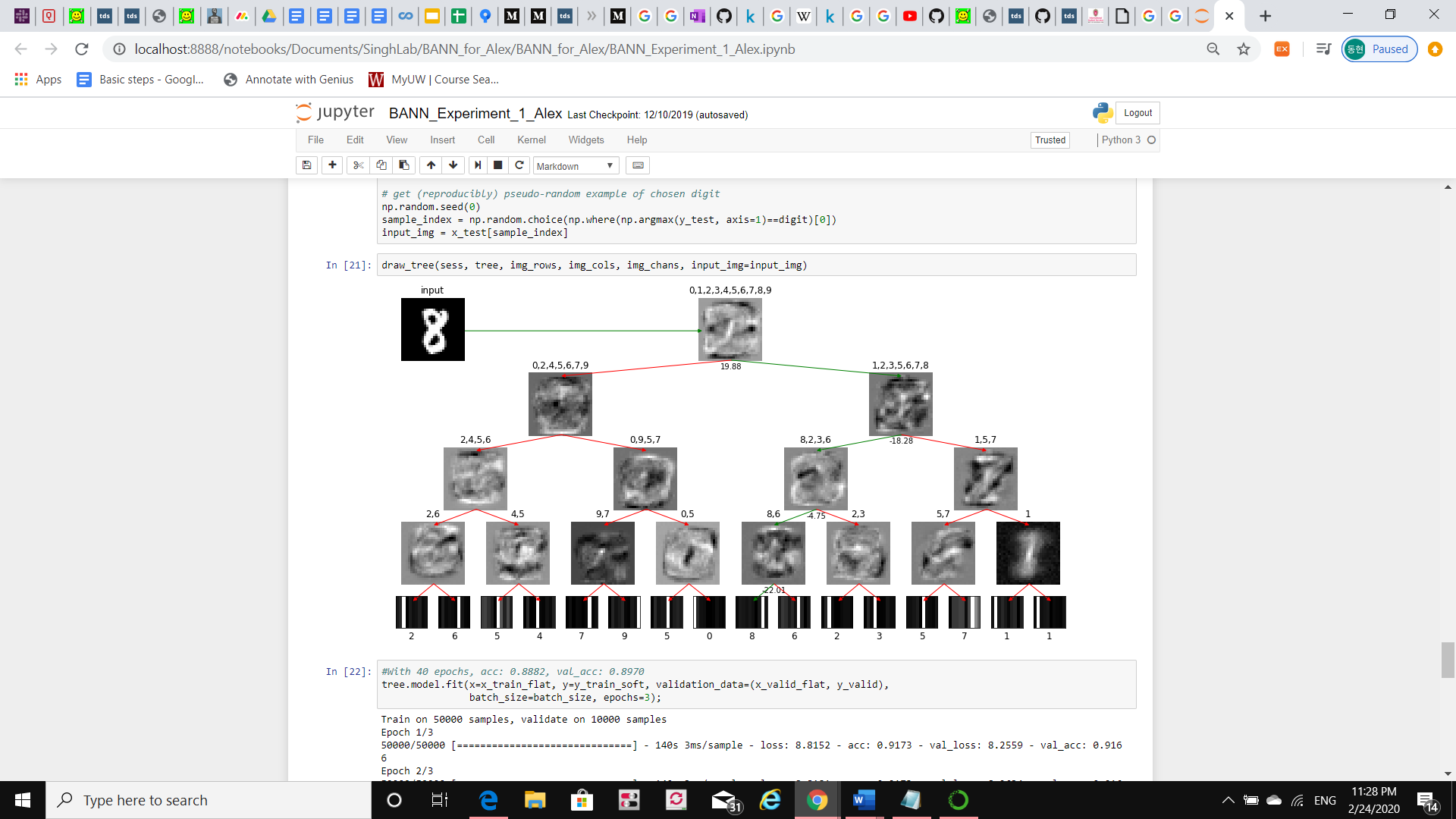
Bayesian Neural Networks generally refer to a class of algorithm that treat neural network models in a ‘Bayesian manner’, which is well suited for uncertainty estimates. In this paper our goal is to design deep neural network with structure and interpretability in mind, with easily amenable to uncertainty estimation, which we call ‘Bayesian Additive Neural Network’. From this algorithm we can produce posterior distribution approximation that provides uncertainty estimate, and moreover, achieve interpretability in deep probabilistic models to understand the flow of information.

The main structure that we will be using is ‘Computational Skeleton’ that is used to study the relationship between neural network and kernels. We will use this this to design Bayesian Neural Network block by block, such that it can help efficiently approximate the posterior of Deep Gaussian Processes. Computational Skelton is, simply, feed-forward computation structure from the inputs to the outputs. It is a multi-layer graph with two layers being fully connected Neural Network. Addition to this structure we will be using two kinds of blocks to design the model structure. Function block, which will allow every node in computational skeleton replicate few times that will help defining Bayesian priors and posteriors. Random feature block, which is used to construct random feature approximations for kernels to leverage the expressive power of Deep Gaussian Process. Therefore, our model will be designed, with given computational skeleton constructing Bayesian Neural Network sequentially replacing edges in computational skeleton with combination of function blocks and random blocks from bottom to top. From this framework we apply additive structure to detect statistical interaction and define this through ANOVA decomposition for interpretability.

**- Knowledge Distilled Soft Decision Tree and BANN**

Similar to our work to show flow of the information in deep neural network models, there are few other attempts to achieve this. Soft Decision Tree developed by Frosst and Hinton proposes a novel way of resolving the tension between generalization and interpretability. While deep neural network outperforms the prediction ability of the decision tree, it is hard to understand how deep neural network makes its decisions. Therefore, Frosst and Hinton use decision tree as their model but use deep neural network to train the decision tree with soft labels and transfer the generalization abilities of neural net by using knowledge distillation. By using knowledge distilled soft decision tree, it improves its prediction ability and makes it possible to explain why it made certain decisions.

The soft decision trees that they used uses decision boundaries that are not aligned with axes defined by the components of the input vector, and additionally, they are trained by first picking the size of the tree and then using mini-batch gradient descent to update all the parameters simultaneously rather than deciding splits one node at a time.

<SDT>

From the figure above you can see how soft decision tree reached to its decision on classification. The images at the inner nodes are the learned filters, and the images at the leaves are visualization of the learned probability distribution over classes. The green line shows which node did it choose to get to the decision.

To compare this result with BANN, same MNIST dataset were used and classification on the data were performed. Our model was experimented on both hard labels and soft labels. Hard labels being raw target variable and soft labels being target variables produced by knowledge distillation. There was not much difference in prediction accuracy between hard label trained and soft label trained for our model being both around 88%. While hard label trained SDT’s prediction accuracy was 80% and soft label trained SDT’s prediction accuracy was 90%. We could see that both SDT and BANN prediction accuracy being similar with around 90%, but they provide distinct way of interpretability. SDT uses model-based interpretability using decision tree risking interpretability-accuracy tradeoff, while BANN uses model-agnostic interpretability using feature interaction. We can compare these two models and provide different point of view of interpretability and flow of information.

**-Further possible experiments**

The key component of the algorithm that we developed lies on producing posterior distribution approximation that provides uncertainty estimate and interpretability in deep probabilistic models to understand the flow of information. From our prior experiments, we compared prediction accuracy and interaction detection of our BANN model with other deep probabilistic models and showed that our model provides competitive performance on both of them. Additionally, our model showed its flexibility by changing to other deep probabilistic models by some changes in blocks. Our model also showed that it can produce meaningful information for interactions with uncertainties.

In the field of computer vision there are many attempts to achieve interpretability or visual explanation of the model like we provided to understand how models intrinsically work, for example, Grad-CAM and Network Dissection. While deep probabilistic models are well used in the field of computer vision, we can take more advantage of our model to provide different point of view of providing interpretability of the model using statistical interaction.

- Interpretable ML (Cynthia Rudin)

Reviewed paper: Stop Explaining Black Box Machine Learning Models for High Stakes Decision and use Interpretable Models Instead

In this paper, Dr. Rudin tackles the problem of “Explainable Machine Learning”, which are models that try to explain black box models that is likely to perpetuate bad practices and potentially cause catastrophic harm to society. She proposes models that are inherently interpretable rather than using explainable ML.

She proposes key issues with explainable ML, and some notable points are

* It is a myth that there is necessarily a trade-off between accuracy and interpretability
* Explainable ML methods provide explanations that are not faithful to what the model computes
* Explanations often do not make sense, or do not provide enough detail to understand what the black box is doing ex) Saliency Map

She also proposes key issues with interpretable ML, and some notable points are

* Interpretable models can entail significant effort to construct, in terms of both computation and domain expertise
* Corporations can make profits from the intellectual property afforded to a black box
* Transparent models seem to uncover ‘hidden patterns’ in black box

However, there are some challenges in interpretable ML,

* Constructing optimal logical models
* Construct optimal sparse scoring system
* Define interpretability for specific domains and create methods accordingly, including computer vision

With these issues and challenges in mind, we should propose our model’s interpretability in discussion.

- Knockoffs

Reviewed paper: Knockoffs for the Mass: New Feature Importance Statistics with False Discovery Guarantees

Using Knockoffs (Synthetic data) is to identify that captures causally affect outcome and capture correlations amongst the features while false discovery is limited. This knockoff procedure is a powerful framework for selecting important features while statistically controlling false discoveries.

Our model uses group lasso type penalty on the first layer of the sub neural network to find possible interaction candidates. In this paper, authors tackle the limitation of using lasso penalty that this method is plagued by correlation between the features that is not really relevant for the outcome. They point out for feature selection problem it gets more difficult where there are no clean parameters for a model and lack statistical guarantees. Therefore, they suggest new approach called Model-X knockoff procedure. We can take advantages by using knockoff procedure in our additive model than using group-lasso penalty to select better interaction candidates.

- Compressed Interaction Network

Reviewed paper: xDeepFM: Combining Explicit and Implicit Feature Interaction for Recommender Systems

Compressed Interaction Network (CIN) is a network used in eXtreme Deep Factorization Machine(xDeepFM) which is one of recommendation algorithm, that helps the model to learn high order feature interaction explicitly.

While in our model we use an additive structure on the network and apply post-training ANOVA decomposition to detect statistical interactions, CIN generates high-order feature interactions in an explicit fashion and at the vector-wise level and complexity of network will not grow exponentially with the degree of interactions. Then, they incorporate CIN and DNN in an end-to-end framework to learn low and high order feature interaction implicitly to combine them and structure xDeepFM. The explicit feature interactions are learned by CIN through its hidden layers and embedding feature vector. CIN also has strong connection with CNN and RNN, in terms of how it is structured. We can take advantages by using CIN for our additive structure to combine or replace the ANOVA decomposition layer and get high order interaction explicitly and implicitly at vector-wise level with uncertainty.