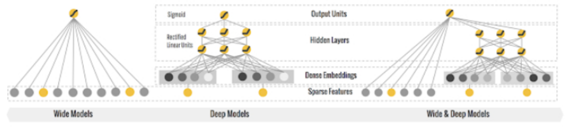
Wide & Deep Learning: Better Together with TensorFlow

Wednesday, June 29, 2016

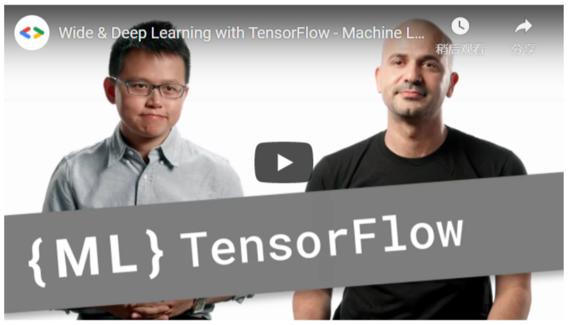
Posted by Heng-Tze Cheng, Senior Software Engineer, Google Research

The human brain is a sophisticated learning machine, forming rules by memorizing everyday events ("sparrows can fly" and "pigeons can fly") and generalizing those learnings to apply to things we haven't seen before ("animals with wings can fly"). Perhaps more powerfully, memorization also allows us to further refine our generalized rules with exceptions ("penguins can't fly"). As we were exploring how to advance machine intelligence, we asked ourselves the question—can we teach computers to learn like humans do, by combining the power of memorization and generalization?

It's not an easy question to answer, but by jointly training a wide linear model (for memorization) alongside a deep neural network (for generalization), one can combine the strengths of both to bring us one step closer. At Google, we call it **Wide & Deep Learning**. It's useful for generic large-scale regression and classification problems with sparse inputs (categorical features with a large number of possible feature values), such as recommender systems, search, and ranking problems.



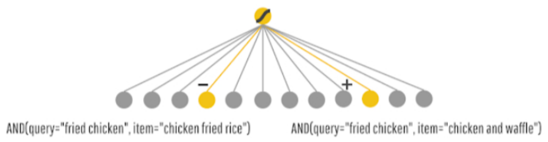
Today we're open-sourcing our implementation of Wide & Deep Learning so that you can easily train a model yourself. Please check out the TensorFlow tutorials on Linear Models and Wide & Deep Learning, as well as our research paper to learn more.



How Wide & Deep Learning works.
Let's say one day you wake up with an idea for a new app called *FoodIO*[*]. A user of the app just needs to say out loud what kind of food he/she is craving for (the *query*). The app magically predicts the dish that the user will like best, and the dish gets delivered to the user's front door (the *item*). Your key metric is consumption rate—if a dish was eaten by the user, the score is 1; otherwise it's 0 (the *label*).

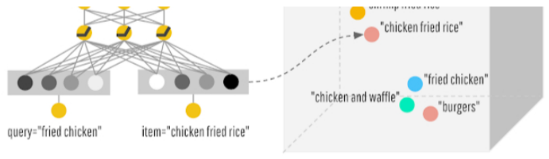
You come up with some simple rules to start, like returning the items that match the most characters in the query, and you release the first version of FoodIO. Unfortunately, you find that the consumption rate is pretty low because the matches are too crude to be really useful (people shouting "fried chicken" end up getting "chicken fried rice"), so you decide to add machine learning to learn from the data.

The Wide model.
In the 2nd version, you want to memorize what items work the best for each query. So, you train a linear model in TensorFlow with a **wide** set of cross-product feature transformations to capture how the co-occurrence of a query-item feature pair correlates with the target label (whether or not an item is consumed). The model predicts the probability of consumption P(consumption | query, item) for each item, and FoodIO delivers the top item with the highest predicted consumption rate. For example, the model learns that feature `AND(query="fried chicken", item="chicken and waffles")` is a huge win, while `AND(query="fried chicken", item="chicken fried rice")` doesn't get as much love even though the character match is higher. In other words, FoodIO 2.0 does a pretty good job **memorizing** what users like, and it starts to get more traction.


–
+
AND(query="fried chicken", item="chicken fried rice")
AND(query="fried chicken", item="chicken and waffle")

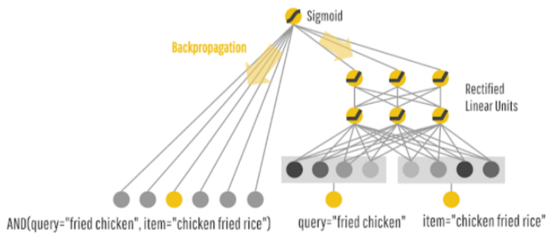
The Deep model.
Later on you discover that many users are saying that they're tired of the recommendations. They're eager to discover similar but different cuisines with a "surprise me" state of mind. So you brush up on your TensorFlow toolkit again and train a **deep** feed-forward neural network for FoodIO 3.0. With your deep model, you're learning lower-dimensional dense representations (usually called embedding vectors) for every query and item. With that, FoodIO is able to **generalize** by matching items to queries that are close to each other in the embedding space. For example, you find that people who asked for "fried chicken" often don't mind having "burgers" as well.

Search blog ...
Labels
Archive
Feed
Follow @googleai
Give us feedback in our Product Forums.



Combining Wide and Deep models.
However, you discover that the deep neural network sometimes generalizes too much and recommends irrelevant dishes. You dig into the historic traffic, and find that there are actually two distinct types of query-item relationships in the data.

The first type of queries is very targeted. People shouting very specific items like "iced decaf latte with nonfat milk" really mean it. Just because it's pretty close to "hot latte with whole milk" in the embedding space doesn't mean it's an acceptable alternative. And there are millions of these rules where the transitivity of embeddings may actually do more harm than good. On the other hand, queries that are more exploratory like "seafood" or "italian food" may be open to more generalization and discovering a diverse set of related items. Having realized these, you have an epiphany: Why do I have to choose either wide or deep models? Why not both?


Sigmoid
Backpropagation
Rectified Linear Units
AND(query="fried chicken", item="chicken fried rice")
query="fried chicken"
item="chicken fried rice"

Finally, you build FoodIO 4.0 with Wide & Deep Learning in TensorFlow. As shown in the graph above, the sparse features like `query="fried chicken"` and `item="chicken fried rice"` are used in both the wide part (left) and the deep part (right) of the model. During training, the prediction errors are backpropagated to both sides to train the model parameters. The cross-feature transformation in the wide model component can memorize all those sparse, specific rules, while the deep model component can generalize to similar items via embeddings.

Wider. Deeper. Together.
We're excited to share the TensorFlow API and implementation of Wide & Deep Learning with you, so you can try out your ideas with it and share your findings with everyone else. To get started, check out the code on GitHub and our TensorFlow tutorials on Linear Models and Wide & Deep Learning.

Acknowledgement
Bringing Wide & Deep from idea, research to implementation has been a huge team effort. We'd to like to thank all the people who have contributed to the project or have given us advice, including: Heng-Tze Cheng, Mustafa Ispir, Zakaria Haque, Lichan Hong, Rohan Anil, Denis Baylor, Vihan Jain, Salem Haykal, Robson Araujo, Xiaobing Liu, Yonghui Wu, Thomas Strohmann, Tal Shaked, Jeremiah Harmsen, Greg Corrado, Glen Anderson, D. Sculley, Tushar Chandra, Ed Chi, Rajat Monga, Rob von Behren, Jarek Wilkiewicz, Christine Robson, Illia Polosukhin, Martin Wicke, Gus Katsiapis, Alexandre Passos, Olivier Chapelle, Levent Koc, Akshay Naresh Modi, Wei Chai, Hrishi Aradhye, Othar Hansson, Xinran He, Martin Zinkevich, Joe Toth, Anton Rusanov, Hemal Shah, Petros Mol, Frank Li, Yutaka Suematsu, Sameer Ahuja, Eugene Brevdo, Philip Tucker, Shanqing Cai, Kester Tong, and more.

* For illustration only. FoodIO is not a real app.↩