Merck Molecular Activity Challenge

Speaker: George Dahl

Team: George Dahl, Ruslan Salakhutdinov, Christopher Jordan-Squire, Navdeep Jaitly, Geoffrey Hinton

One way to make new drugs

1. Identify a medically interesting protein
2. Measure the activity of many candidate molecules towards the target protein
3. Make drugs from the highly active molecules

- For this to work, we need activity measurements for millions of candidate molecules for each target
- We need a way to predict activity from chemical structure without running assays

Competition problem

- 15 related molecular activity datasets/tasks
- Each task is a regression problem: predict the activity of molecules towards a particular target
- Only given chemical structure feature vectors for candidate molecules, not targets
- Molecules and input features sometimes occur in multiple tasks
- Activity labels have different units in different tasks

Data set characteristics

- Input features are non-negative ints (3%-10% > 0)
- The number of features in a dataset ranges from 4,000 to 9,000
- Across all tasks, there are 11,000 unique features
- The number of training cases is between 1,800 and 37,000
    - Normally we expect large neural networks to overfit in these conditions

Evaluation metric

$$R^2 = \frac{1}{15}\sum_s r_s^2 \qquad\qquad r_s^2 = \frac{\left[\sum_i (t_i - \bar{t})(y_i - \bar{y})\right]^2}{\sum_i (t_i - \bar{t})^2 \sum_i (y_i - \bar{y})^2}$$

- Correlation with true activity on each dataset averaged across datasets irrespective of # of cases
- Training set contains molecules assayed before a certain date, test set molecules assayed after that date

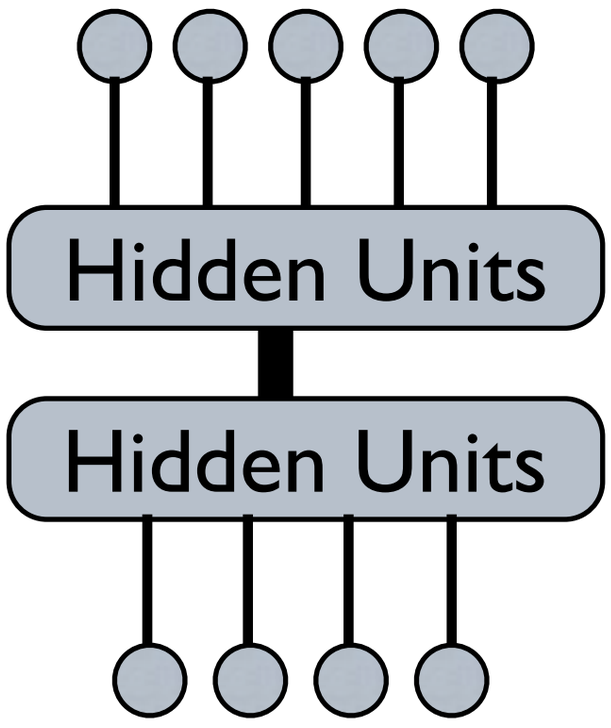
Solution Overview

- Equally weighted average of 2-6 models for each task
    - Averages included neural nets, Gaussian process regression, and gradient boosted decision trees
- Neural nets trained with dropout were the only indispensable part of our ensembles

Single-task neural nets

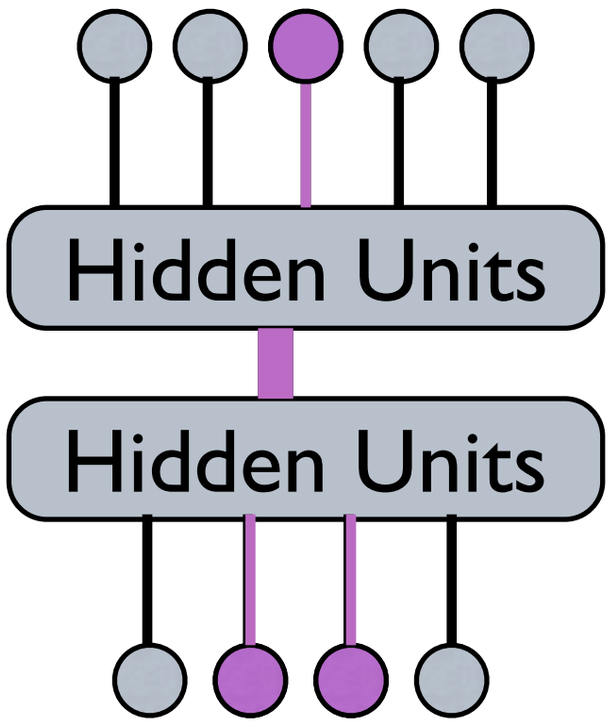
- Usually two hidden layers
- Used max(0,x) nonlinearity a.k.a. "ReLUs" or logistic sigmoids
- For the small datasets, dropout was essential to avoid overfitting
    - Without dropout, using more than ~30 hidden units wasn't workable, with dropout we can use 500-1000

Multi-task neural nets



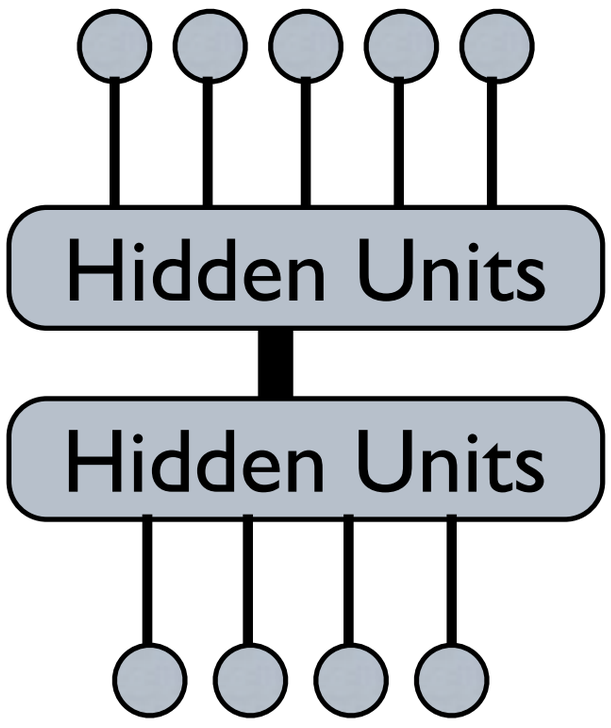
- One output unit per task (only observe one at a time)
- One input for each unique input across all tasks
- Labels for each task transformed to have mean zero and unit variance
- Trained on minibatches with an equal number of cases from each dataset

Multi-task neural nets



- One output unit per task (only observe one at a time)
- One input for each unique input across all tasks
- Labels for each task transformed to have mean zero and unit variance
- Trained on minibatches with an equal number of cases from each dataset

Multi-task neural nets



- One output unit per task (only observe one at a time)
- One input for each unique input across all tasks
- Labels for each task transformed to have mean zero and unit variance
- Trained on minibatches with an equal number of cases from each dataset

Leaderboard Results

Submission	Validation R^2	Test R^2
Equally-weighted Averaged Ensemble with GBM (1st place)	0.494	0.494
Equally-weighted Averaged Ensemble without GBM (1st place)	0.492	0.492
neural nets only (1st place)	0.488	0.489
"Best" 5-fold CV models (7th place)	0.477	0.478
GPs only (10th place)	0.471	0.473

- Neural nets trained with dropout would have won *even without ensembling*
- Internal CV overstated GP performance relative to neural nets
- Even the simple averaging we used is quite helpful

Conclusions

- Dropout lets us train larger neural nets, on more inputs, with fewer training cases
    - Dropout helps a lot when there are more input dimensions than training cases
- Single- and Multi-task neural nets with dropout suffice to get 1st place in the competition