Dropout Training

(Hinton et al. 2012)

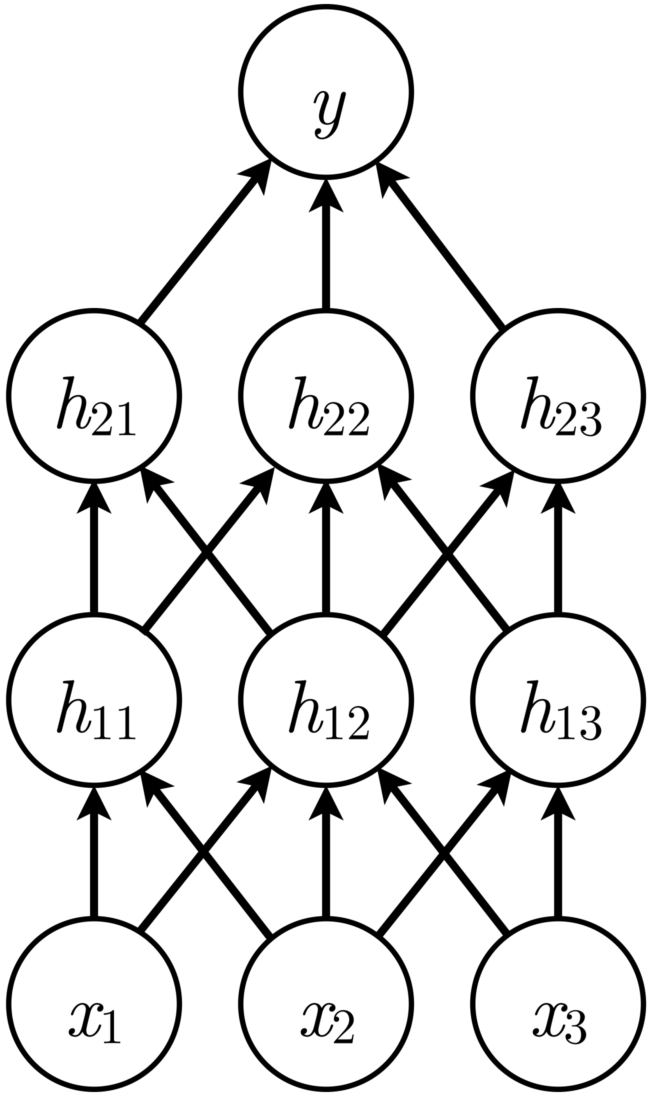
Dropout training

- Introduced in Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors.CoRR, abs/1207.0580.

- **Dropout recipe**:
  - Each time we present data example x, randomly delete each hidden node with 0.5 probability.
  - This is like sampling from $2^{|h|}$ different architectures.
  - At test time, use all nodes but divide the weights by 2.

- **Effect I**: Reduce overfitting by preventing ”co-adaptation”

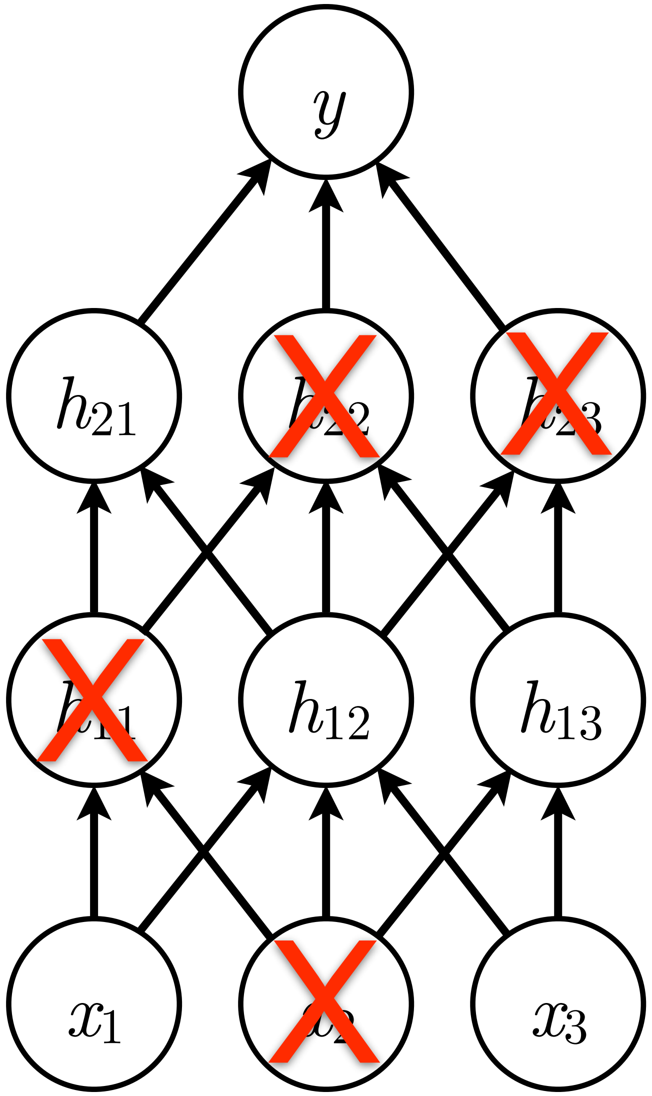
- **Effect 2**: Ensemble model averaging via bagging



Dropout training

- Introduced in Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors.CoRR, abs/1207.0580.

- **Dropout recipe**:
  - Each time we present data example x, randomly delete each hidden node with 0.5 probability.
  - This is like sampling from $2^{|h|}$ different architectures.
  - At test time, use all nodes but divide the weights by 2.

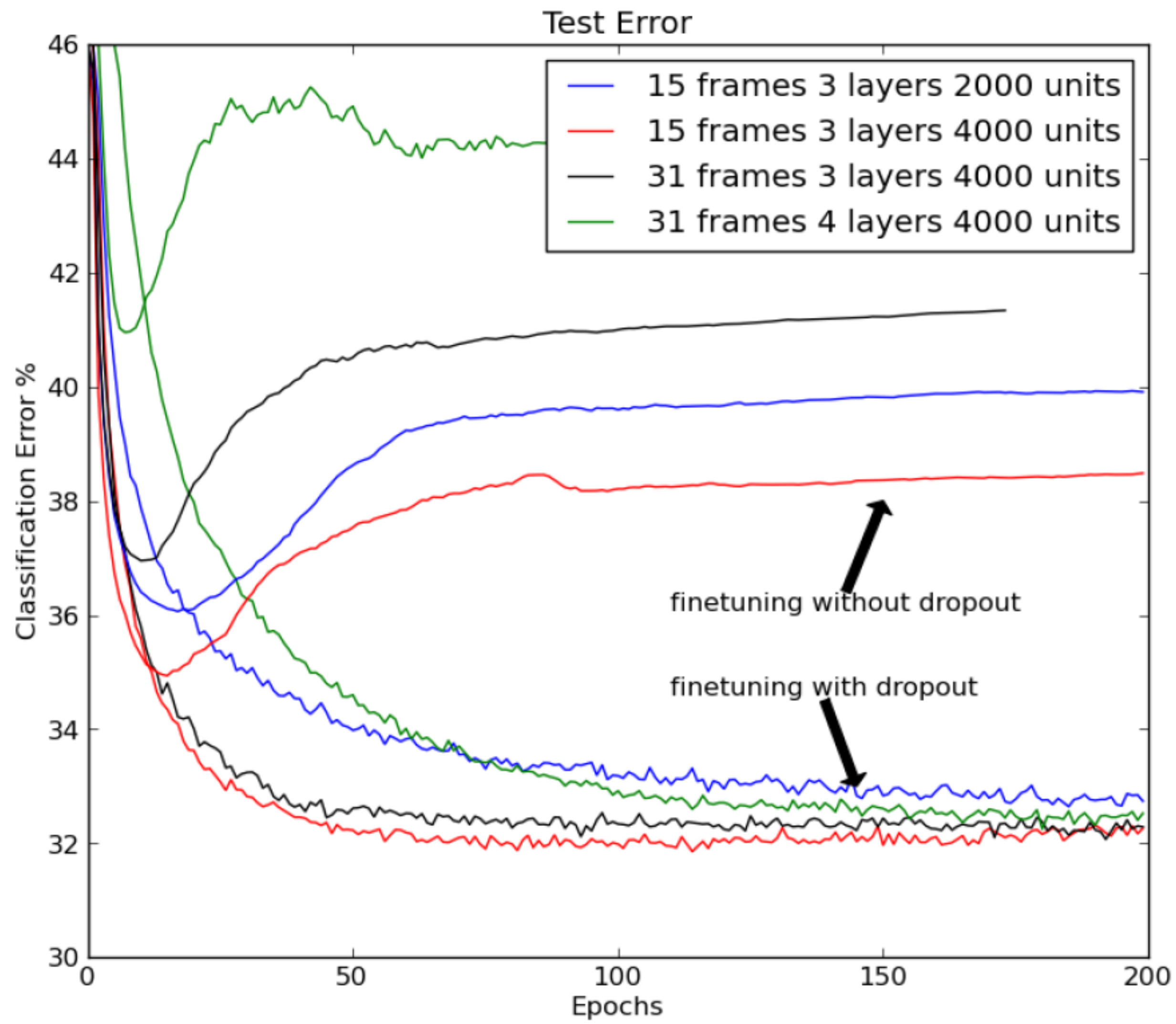


- **Effect I**: Reduce overfitting by preventing ”co-adaptation”

- **Effect 2**: Ensemble model averaging via bagging

Dropout: TIMIT phone recognition

- Dropout helps.
- Dropout + pretraining helps more.

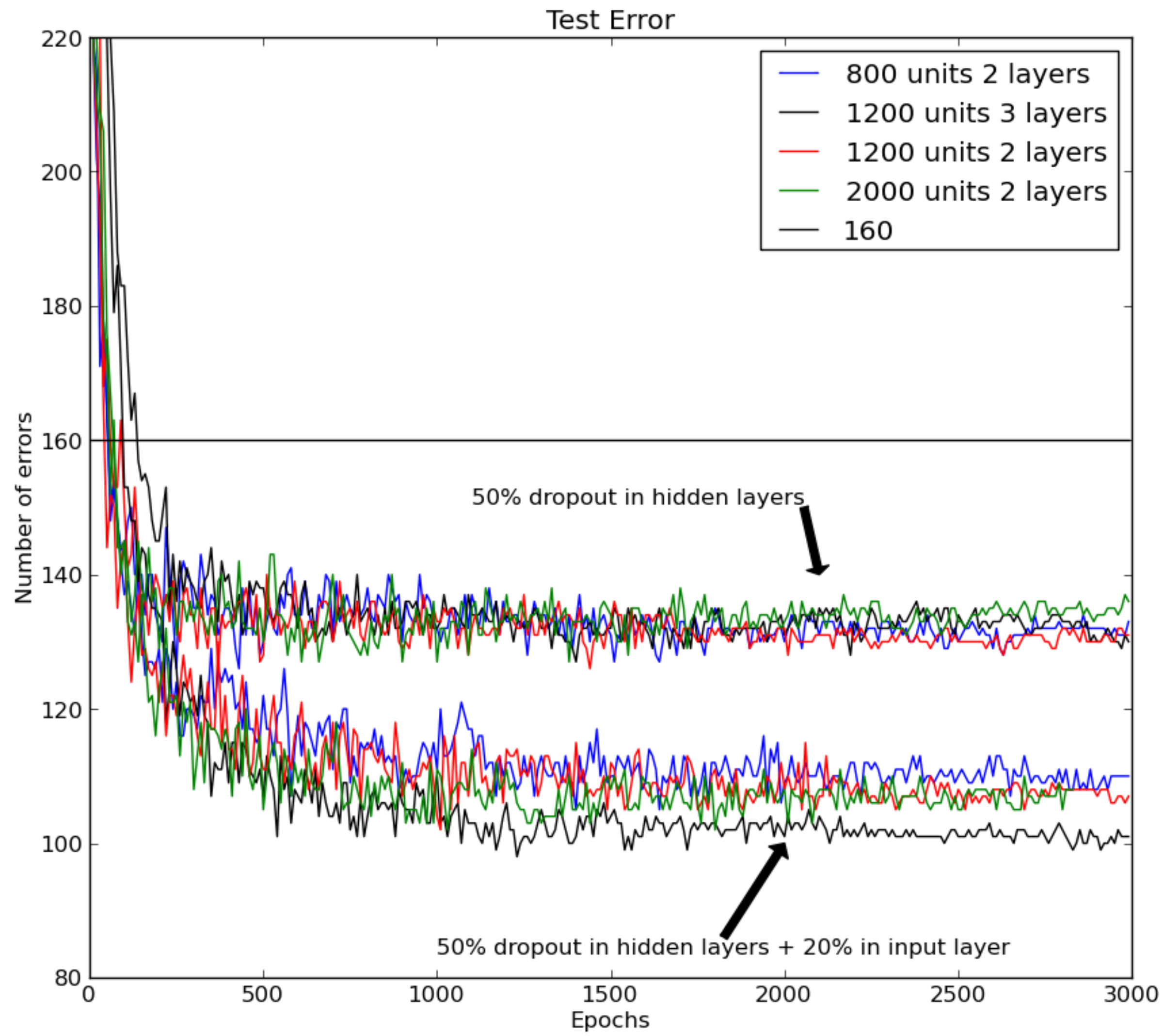


Method	Phone Error Rate%
Neural Net (6 layers) [12]	23.4
Dropout Neural Net (6 layers)	21.8
DBN-pretrained Neural Net (4 layers)	22.7
DBN-pretrained Neural Net (6 layers) [12]	22.4
DBN-pretrained Neural Net (8 layers) [12]	20.7
mcRBM-DBN-pretrained Neural Net (5 layers) [2]	20.5
DBN-pretrained Neural Net (4 layers) + dropout	**19.7**
DBN-pretrained Neural Net (8 layers) + dropout	**19.7**

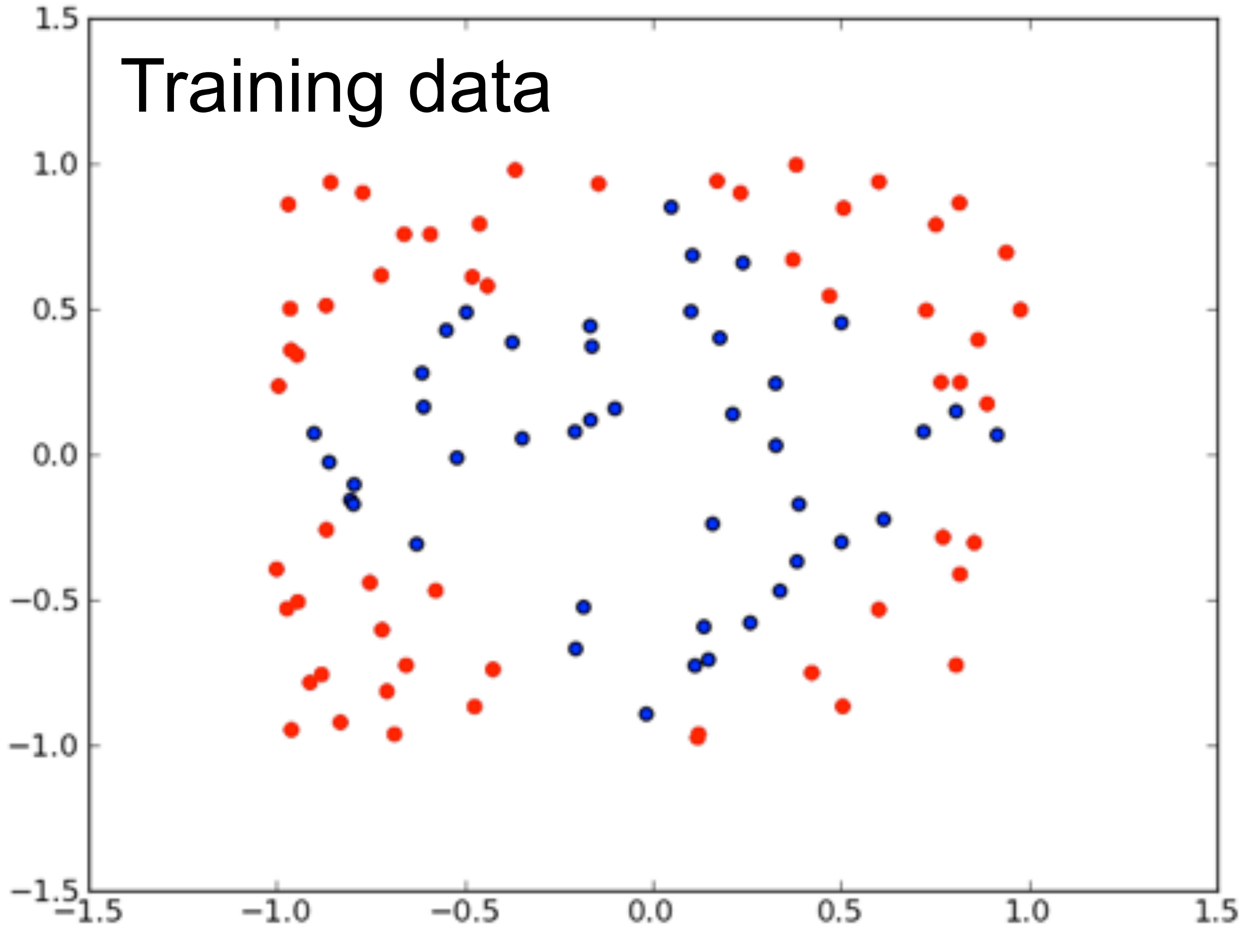
Dropout: MNIST digit recognition

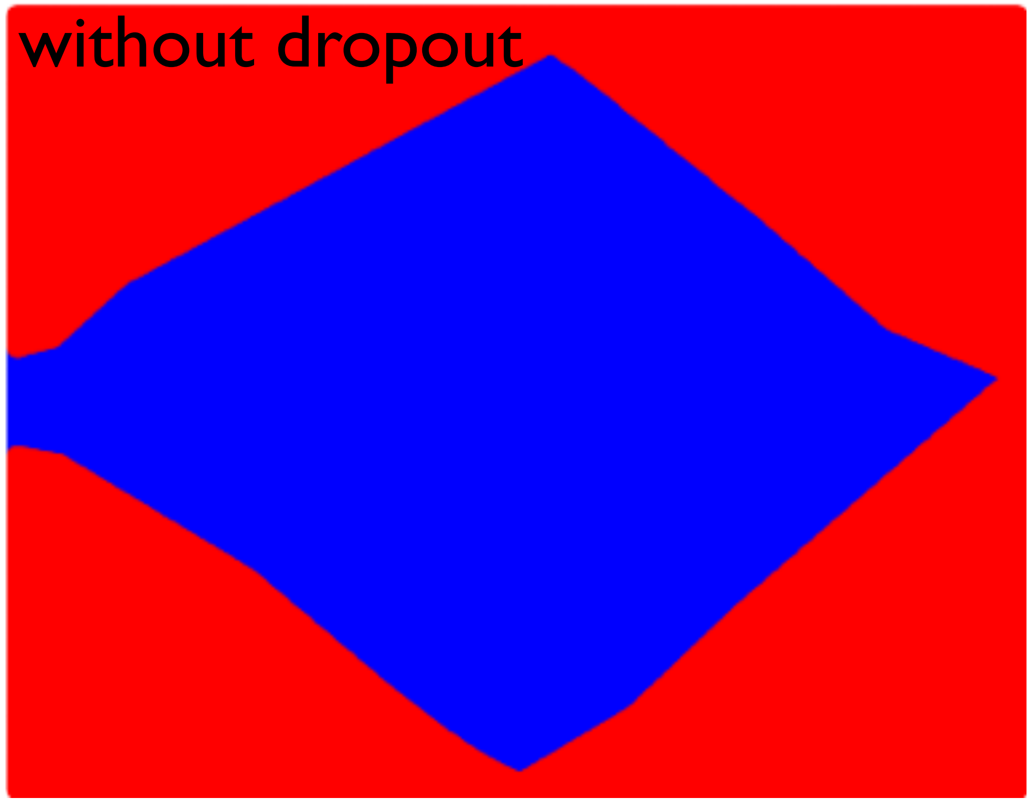
- Dropout is effective on MNIST.
- Particularly with input dropout.
- Comparison against other regularizers.

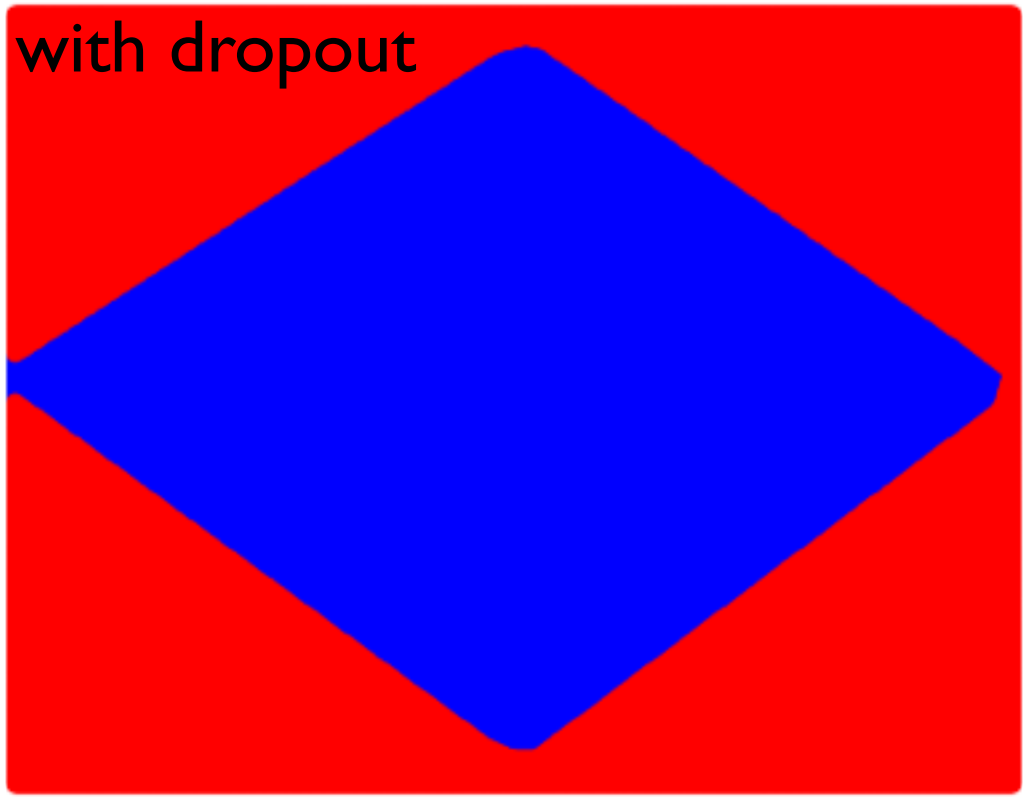
Method	MNIST Classification error %
L2	1.62
L1 (towards the end of training)	1.60
KL-sparsity	1.55
Max-norm	1.35
Dropout	1.25
Dropout + Max-norm	**1.05**



The unreasonable effectiveness of dropout







- A simple 2D example.
- Decision surfaces after training:

Claim: Dropout is approximate model averaging

- Hinton et al. (2012):
  - Dropout **approximates geometric model averaging**.

Arithmetic mean: $\frac{1}{N}\sum_{i=1}^{N} x_i$ Geometric mean: $\left(\prod_{i=1}^{N} x_i\right)^{\frac{1}{N}}$

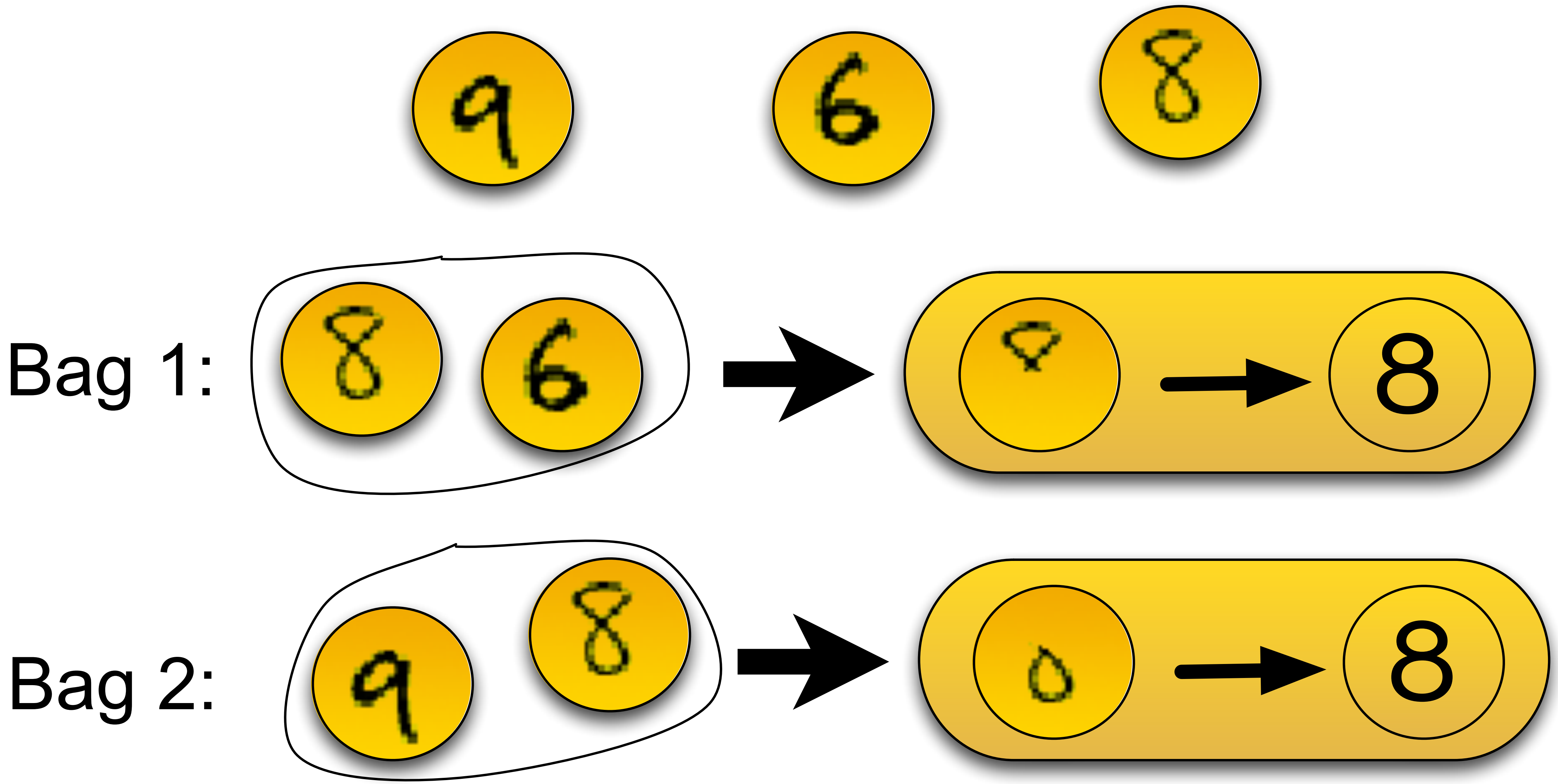
Claim: Dropout is approximate model averaging

- In networks with a single hidden layer of N units and a “softmax” output layer:
- Using the mean network is exactly equivalent to taking the geometric mean of the probability distributions over labels predicted by all 2^N possible networks.
- For deep networks, it’s an approximation.

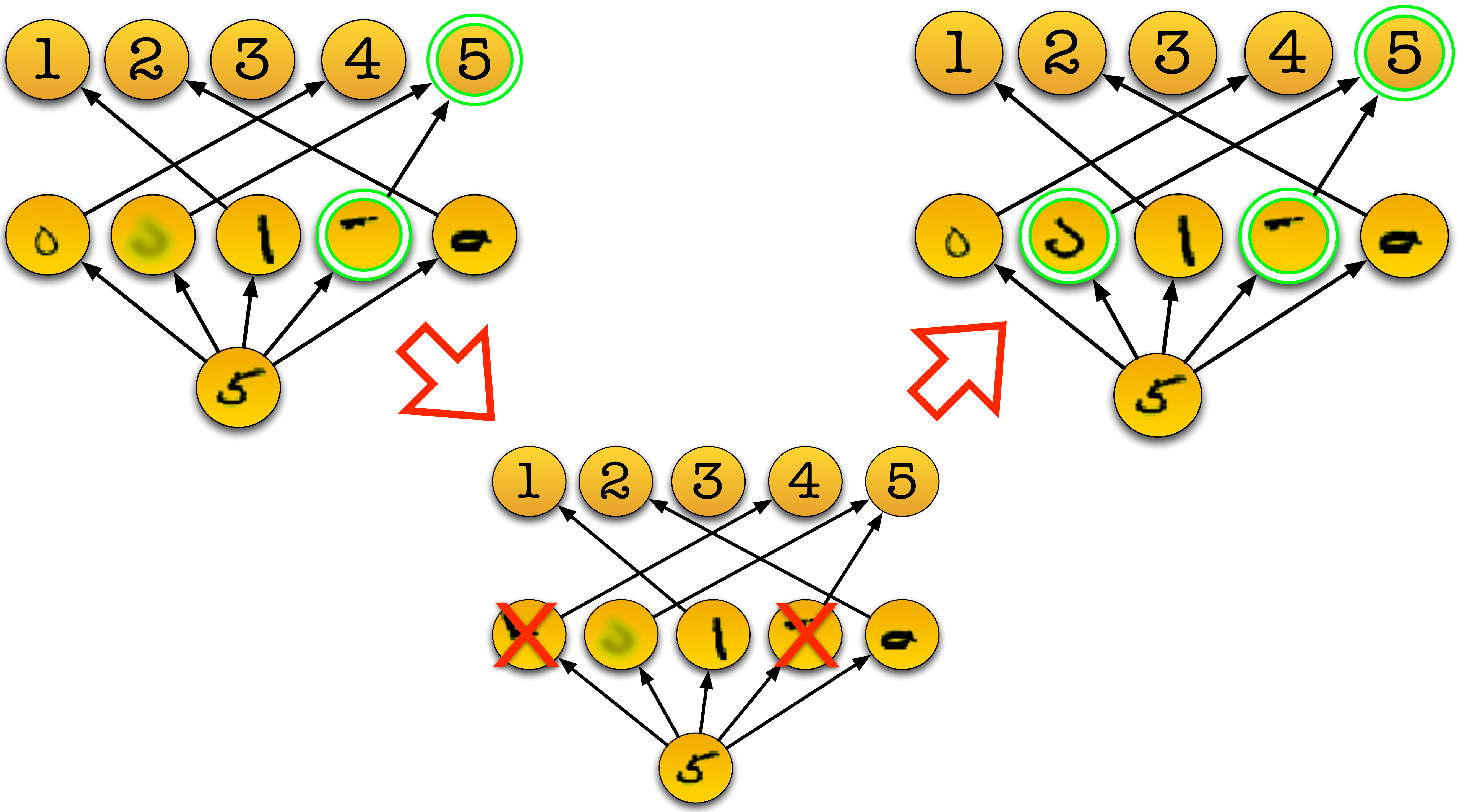
Bagging predictors

- **Bagging**: A method of model averaging.
  - To **reduce overfitting** (decrease variance of the estimator).
- **Methodology**: Given a standard training set ***D*** of size ***n***,
  - Bagging generates ***m*** new training sets , each of size ***n'***, by sampling from ***D*** uniformly and with replacement.
  - train ***m*** models using the above ***m*** datasets and combined by averaging the output (for regression) or voting (for classification).

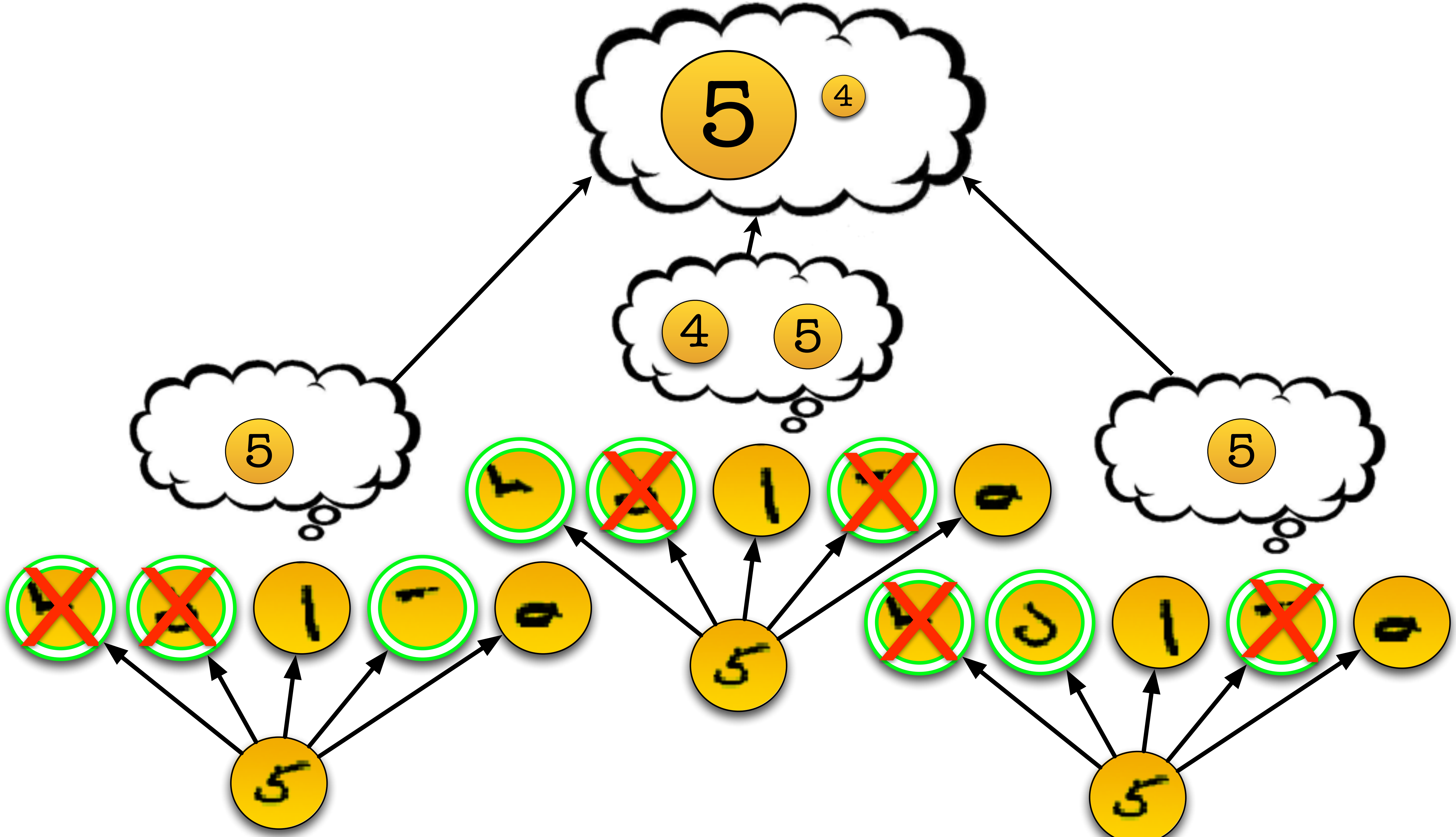
Bagging predictors



Dropout training



Dropout as bagging



Is dropout performing bagging?

- **There are a few important differences:**

  1. The model averaging is only approximate for deep learning.

  2. Bagging is typically done with an **arithmetic mean**. Dropout approximates the **geometric mean**.

  3. In dropout the members of the ensemble are **not independent**. There is significant **weight sharing**.

Dropout ≈ geometric mean?

- How accurate is the “weight scaling trick” approximation to the geometric mean?
  - How does the use of this approximation impact classification performance?

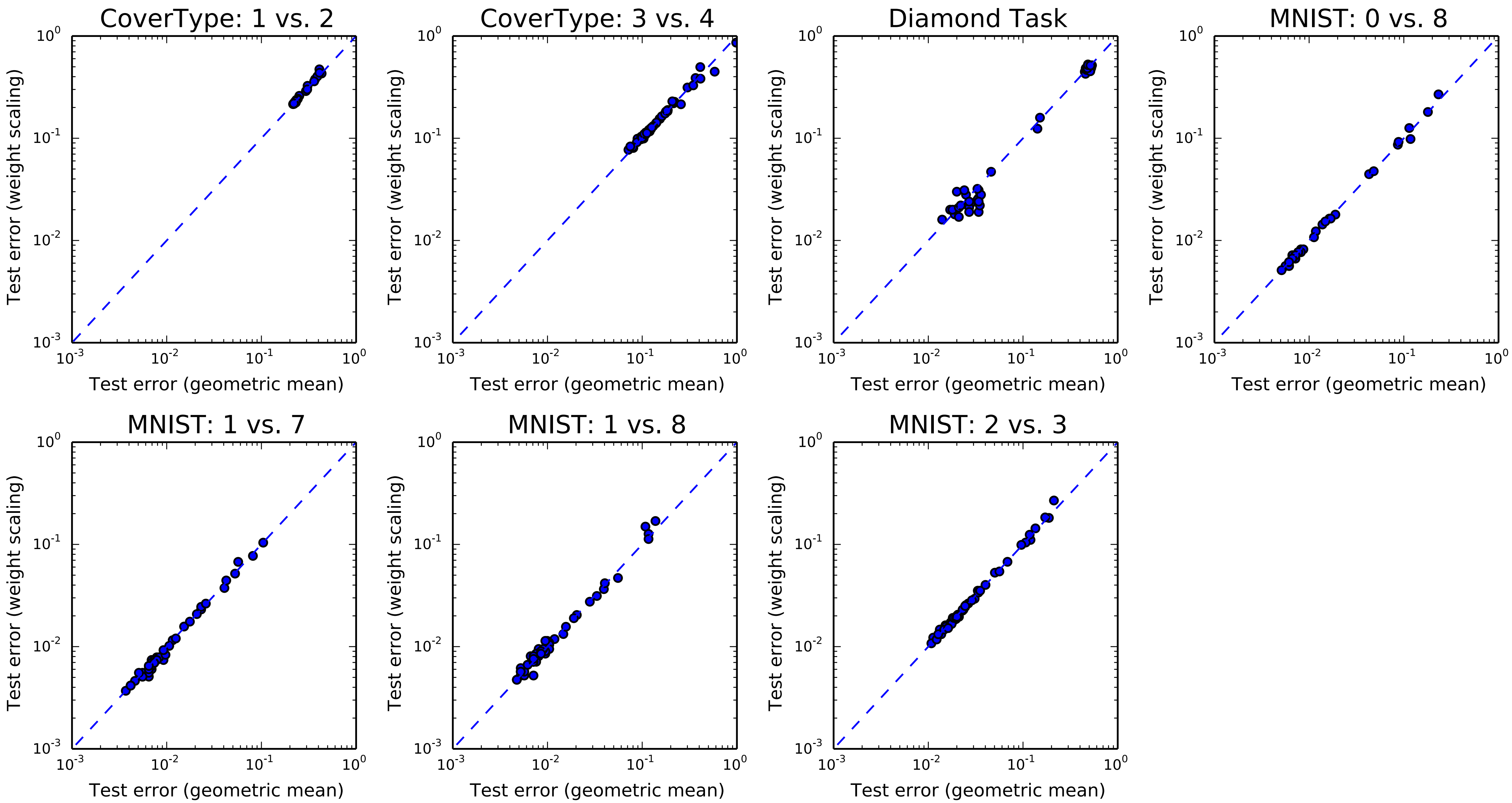
- How does the geometric mean compare to the arithmetic mean?
  - Conventionally, the arithmetic mean is used with ensemble methods?

Dropout ≈ geometric mean?

- **Small networks experiments:**
  - Exhaustive computation of exponential quantities is possible.
  - Two hidden layers (rectified linear), 10 hidden units each, 20 hidden units total
  - $2^{20} = 1{,}048{,}576$ possible dropout masks (for simplicity, don't drop input)
- **Benchmark on 7 simplified binary classification tasks:**
  - 2 different binary classification subtasks from CoverType
  - 4 different binary classification subtasks from MNIST
  - 1 synthetic task in 2-dimensions ("Diamond")

Quality of the Geometric Mean Approximation

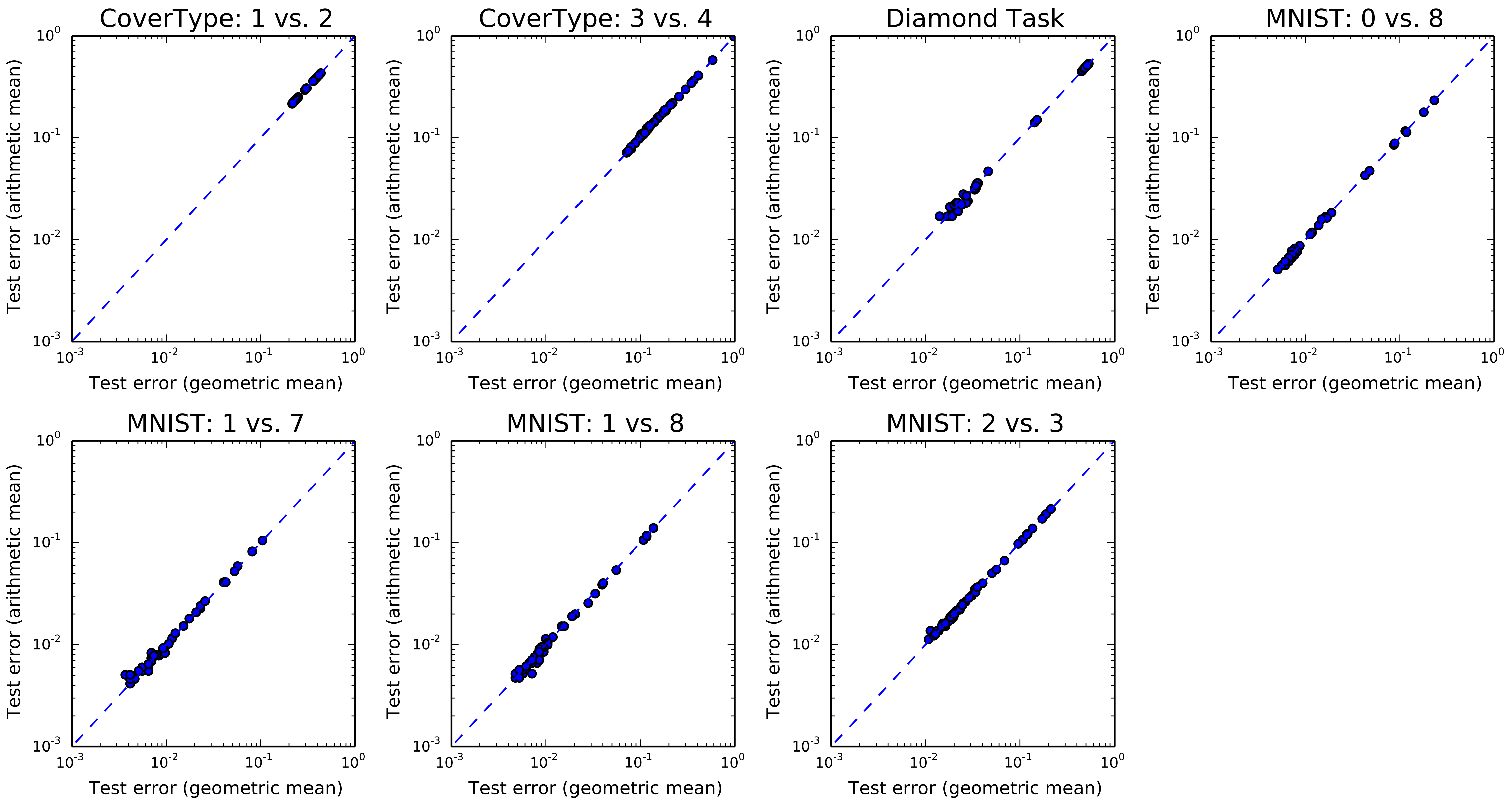
- With ReLUs, weight-scaled predictions perform as well or better than exhaustively computed geometric mean predictions on these tasks.



  - Each dot represents a different randomly sampled hyperparameter configuration. No statistically significant differences in test errors across hyperparameter configurations on any task (Wilcoxon signed-rank test).

Geometric Mean vs. Arithmetic Mean

- No systematic advantage to using the arithmetic mean over all possible subnetworks rather than the geometric mean.



  - Each dot represents a different randomly sampled hyperparameter configuration. No statistically significant differences in test errors across hyperparameter configurations on any task (Wilcoxon signed-rank test).

Dropout vs. Untied Weight Ensembles

- How does the implicit ensemble trained by dropout compare to an ensemble of networks trained with independent weights?
  - With the explicit ensemble drawn from the same distribution (i.e. masked copies of the original).
  - Experiment on MNIST: Average test error for varying sizes of untied-weight ensembles...
  - **Key Observation**: Bagging untied networks yields some benefit, but dropout performs better.

➡ **Dropout weight-sharing has an impact!**