Deep networks work really well on the standard image data sets: large sample sizes, very low label noise, highly structured.

**Questions**
- Do they work in high noise, “small data” settings outside of image/speech recognition?
- Can they simultaneously memorize (noisy) training labels and still generalize well?
- We know “yes” to (1) and (2) for random forests - can we borrow insights?

**Motivation**

**Ensemble Interpretation**

- Ensemble components: deep layers serve to aid variance reduction
- Subnetwork decomposition: low bias + low variance → good generalization

**Empirical Evaluation**

- 116 UCI repository data sets (classification)
- depth 10 networks trained to 100% training accuracy (no regularization)
- hundreds of parameters per observation

**Empirical Insights**

- 116 multi-class data sets from UCI repository
  
  \( (n \approx 600 \text{ observations on average}) \)

- fit unregularized deep networks, zero training error on all data sets

- test error comparable to a random forest!

**Ensemble Interpretation**

- networks decompose into sub-networks with low bias and relatively low correlation
- deep layers serve to aid variance reduction

**Highlights**

- Ensembling
  - A neural network with \( L \) hidden layers and \( M \) hidden nodes can be written
    \[
    z^{l+1} = W^{l+1}g(z^l) \quad l = 0, \ldots, L
    \]
    \[
    f(x) = \sigma(z^{L+1})
    \]
    where final hidden layer is a sum of sub-networks:
    \[
    z^{l+1}(x) = W^{l+1}g(z^l(x))
    = \sum_{k=1}^K \sum_{m=1}^M \alpha_{m,k}g(z_m^l(x))
    = \sum_{k=1}^K f_k(x)
    \]

- Ensemble Program
  - decompose final layer into sub-networks \( f_1, f_2, \ldots, f_K \)
  - search for sub-networks with low bias and low pairwise error correlation
  - construct \( f_1, f_2, \ldots, f_K \) from linear program
  - low bias + low variance → good generalization

- Ensemble Hunting via Linear Programming
  - Train a network with \( M \) hidden nodes, \( L \) hidden layers until zero training error
  - Find \( \alpha \in \mathbb{R}^{M \times K} \) satisfying the linear system (target \( K \) sub-networks):
    \[
    \sum_{k=1}^K \alpha_{m,k} = W_{L+1,m}^l \quad 1 \leq m \leq M \quad \text{(decompose the final hidden layer)}
    \]
    \[
    \alpha_{m,j,k} = 0 \quad 1 \leq j \leq M \quad 1 \leq k \leq K
    \]
    \[
    \left( \sum_{m=1}^M \alpha_{m,k}g(z_m^l(x_i)) \right) y_i \geq 0 \quad 1 \leq i \leq n, 1 \leq k \leq K \quad \text{(each sub-network has non-negative margin)}
    \]

- Ensemble Hunting: Simulated Example
  - Draw samples \( (x_i, y_i) \in [-1, 1]^2 \times \{-1, 1\} \) from
    \[
    p(y = 1|x) = \begin{cases} 
    1 & \text{if } \|x\|_2 \leq 0.3 \\
    0.15 & \text{otherwise}
    \end{cases}
    \]
  - 10% label noise (red points)
  - train 10 layer network until zero training error
  - influence of noise points localized
  - better test error than random forest

- Takeaways
  - high capacity networks can still generalize well on small data sets with non-trivial noise
  - ensemble interpretation of deep networks, deeper layers offer variance reduction

**Training labels and still generalize well?**

**Draw samples \( (x_i, y_i) \in [-1, 1]^2 \times \{-1, 1\} \) from**

\[
p(y = 1|x) = \begin{cases} 
1 & \text{if } \|x\|_2 \leq 0.3 \\
0.15 & \text{otherwise}
\end{cases}
\]

- 10% label noise (red points)
- train 10 layer network until zero training error
- influence of noise points localized
- better test error than random forest

**Bayes rule**

**DNN**

**Random Forest**

**Subnetwork decomposition \( K = 9 \)**

**Cross-validated Accuracy**

**Subnetwork decomposition \( K = 9 \)**

**Subnetwork decomposition \( K = 9 \)**

**Subnetwork decomposition \( K = 9 \)**