Today

- Random/Decision Forest
- Mixture of Experts
What are the base classifiers?

- Popular choices of base classifier for boosting and other ensemble methods:
  - Linear classifiers
  - Decision trees
Random/Decision Forests

Definition: Ensemble of decision trees
Random/Decision Forests

- **Definition:** Ensemble of decision trees
- **Algorithm:**
  - Divide training examples into multiple training sets (bagging)
  - Train a decision tree on each set (can randomly select subset of variables to consider)
  - Aggregate the predictions of each tree to make classification decision (e.g., can choose mode vote)
Random/Decision Forests

- **Definition**: Ensemble of decision trees
- **Algorithm**:
  - Divide training examples into multiple training sets (bagging)
Random/Decision Forests

- **Definition:** Ensemble of decision trees

- **Algorithm:**
  - Divide training examples into multiple training sets (bagging)
  - Train a decision tree on each set (can randomly select subset of variables to consider)
Random/Decision Forests

- **Definition:** Ensemble of decision trees

- **Algorithm:**
  - Divide training examples into multiple training sets (bagging)
  - Train a decision tree on each set (can randomly select subset of variables to consider)
  - Aggregate the predictions of each tree to make classification decision (e.g., can choose mode vote)
Experts cooperate to predict output

\[
y(x) = \sum_{m} g_m y_m(x)
\]
Experts cooperate to predict output

\[ y(x) = \sum_{m} g_m y_m(x) \]

Vote of each expert has consistent weight for each test example
Mixture of Experts

- Weight of each expert is not constant – depends on input $x$

$$y(x) = \sum_{m} g_m(x) y_m(x)$$
Mixture of Experts

- Weight of each expert is not constant – depends on input $x$

$$y(x) = \sum_{m} g_m(x)y_m(x)$$

- Gating network encourages specialization (local experts) instead of cooperation
Mixture of Experts: Summary

1. Cost function designed to make each expert estimate desired output independently
1. Cost function designed to make each expert estimate desired output independently

2. Gating network softmax over experts: stochastic selection of who is the true expert for the given input
1. Cost function designed to make each expert estimate desired output independently

2. Gating network softmax over experts: stochastic selection of who is the true expert for given input

3. Allow each expert to produce distribution over outputs
Consider a regression problem

\[ E = \left( t - \frac{1}{M} \sum_{m} y_m(x) \right)^2 \]

This can overfit badly. It makes the model much more powerful than training each predictor separately.

Leads to odd objective: consider adding models/experts sequentially.

- If its estimate for \( t \) is too low, and the average of other models is too high, then model \( m \) encouraged to lower its prediction.
Cooperation vs. Specialization

- Consider a regression problem
- To encourage cooperation, we can train to reduce discrepancy between average of predictors with target

\[ E = (t - \frac{1}{M} \sum_m y_m(x))^2 \]

This can overfit badly. It makes the model much more powerful than training each predictor separately.

Leads to odd objective: consider adding models/experts sequentially

▶ if its estimate for \( t \) is too low, and the average of other models is too high, then model \( m \) encouraged to lower its prediction.
Consider a regression problem

To encourage cooperation, we can train to reduce discrepancy between average of predictors with target

\[ E = (t - \frac{1}{M} \sum_{m} y_m(x))^2 \]

This can overfit badly. It makes the model much more powerful than training each predictor separately.
Cooperation vs. Specialization

- Consider a regression problem
- To encourage cooperation, we can train to reduce discrepancy between average of predictors with target

\[ E = (t - \frac{1}{M} \sum_m y_m(x))^2 \]

- This can overfit badly. It makes the model much more powerful than training each predictor separately
- Leads to odd objective: consider adding models/experts sequentially
Cooperation vs. Specialization

- Consider a regression problem
- To encourage cooperation, we can train to reduce discrepancy between average of predictors with target
  \[ E = (t - \frac{1}{M} \sum_{m} y_m(x))^2 \]
- This can overfit badly. It makes the model much more powerful than training each predictor separately
- Leads to odd objective: consider adding models/experts sequentially
  - if its estimate for \( t \) is too low, and the average of other models is too high, then model \( m \) encouraged to lower its prediction
Cooperation vs. Specialization

- To encourage specialization, train to reduce the average of each predictor's discrepancy with target

\[ E = \frac{1}{M} \sum_{m} (t - y_m(x))^2 \]
Cooperation vs. Specialization

- To encourage specialization, train to reduce the average of each predictor’s discrepancy with target

\[ E = \frac{1}{M} \sum_{m} (t - y_m(x))^2 \]

- Use a weighted average: weights are probabilities of picking that ”expert” for the particular training case

\[ E = \frac{1}{M} \sum_{m} g_m(x)(t - y_m(x))^2 \]
Cooperation vs. Specialization

- To encourage specialization, train to reduce the average of each predictor’s discrepancy with target

\[ E = \frac{1}{M} \sum_{m} (t - y_m(x))^2 \]

- Use a weighted average: weights are probabilities of picking that ”expert” for the particular training case

\[ E = \frac{1}{M} \sum_{m} g_m(x)(t - y_m(x))^2 \]

- Gating output is softmax of \( z = Ux \)

\[ g_m(x) = \frac{\exp(z_m(x))}{\sum_i \exp(z_i(x))} \]
Cooperation vs. Specialization

- To encourage specialization, train to reduce the average of each predictor’s discrepancy with target

$$E = \frac{1}{M} \sum_{m} (t - y_m(x))^2$$

- Use a weighted average: weights are probabilities of picking that "expert" for the particular training case

$$E = \frac{1}{M} \sum_{m} g_m(x)(t - y_m(x))^2$$

- Gating output is softmax of $z = Ux$

$$g_m(x) = \frac{\exp(z_m(x))}{\sum_i \exp(z_i(x))}$$

- We want to estimate the parameters of the gating as well as the classifier $y_m$
Look at derivatives to see what cost function will do

\[ E = \frac{1}{M} \sum_m g_m(x)(t - y_m(x))^2 \]
Derivatives of Simple Cost Function

Look at derivatives to see what cost function will do

\[ E = \frac{1}{M} \sum_{m} g_m(x)(t - y_m(x))^2 \]

For gating network, increase weight on expert when its error is less than average error of experts

\[ \frac{\partial E}{\partial y_m} = \frac{1}{M} g_m(x)(t - y_m(x)) \]
\[ \frac{\partial E}{\partial z_m} = \frac{1}{M} g_m(x) \left[ (t - y_m(x))^2 - E \right] \]
Mixture of Experts: Final Cost Function

- Can improve cost function by allowing each expert to produce not just a single value estimate, but a distribution
Can improve cost function by allowing each expert to produce not just a single value estimate, but a distribution

Result is a mixture of experts model:

\[ p(t|\text{MOE}) = \sum_m g_m(x) N(t|y_m(x), \Sigma) \]
Mixture of Experts: Final Cost Function

- Can improve cost function by allowing each expert to produce not just a single value estimate, but a distribution.

- Result is a mixture of experts model:

\[
p(t|MOE) = \sum_{m} g_m(x) \mathcal{N}(t|y_m(x), \Sigma)
\]

- Optimize minus log-likelihood:

\[
-\log p(t|MOE) = -\log \sum_{m} g_m(x) \exp \left( -\frac{1}{2} \| t - y_m(x) \|^2 \right)
\]
Mixture of Experts: Final Cost Function

- Can improve cost function by allowing each expert to produce not just a single value estimate, but a distribution
- Result is a mixture of experts model:
  \[ p(t|\text{MOE}) = \sum_m g_m(x) \mathcal{N}(t|y_m(x), \Sigma) \]
- Optimize minus log-likelihood:
  \[ -\log p(t|\text{MOE}) = -\log \sum_m g_m(x) \exp \left( -\frac{1}{2} ||t - y_m(x)||^2 \right) \]
- Gradient: Error weighted by posterior probability of the expert
  \[ \frac{\partial E}{\partial y_m} = -2 \frac{g_m(x) \exp \left( -\frac{1}{2} ||t - y_m(x)||^2 \right)}{\sum_i g_i(x) \exp \left( -\frac{1}{2} ||t - y_i(x)||^2 \right)} (t - y_m(x)) \]
Mixture of Experts: Example

[Slide credit: G. Hinton]
Mixture of Experts: Summary

- Cost function designed to make each expert estimate desired output independently
Cost function designed to make each expert estimate desired output independently

Gating network softmax over experts: stochastic selection of who is the true expert for given input
Mixture of Experts: Summary

- Cost function designed to make each expert estimate desired output independently
- **Gating network softmax over experts**: stochastic selection of who is the true expert for given input
- Allow each expert to produce **distribution over outputs**
Ensemble methods: Summary

- Differ in training strategy, and combination method
Ensemble methods: Summary

- Differ in training strategy, and combination method
  - Parallel training with different training sets
    - Bagging (bootstrap aggregation) – train separate models on overlapping training sets, average their predictions
Ensemble methods: Summary

- Differ in training strategy, and combination method
  
  ▶ Parallel training with different training sets
    
    **Bagging** (bootstrap aggregation) – train separate models on overlapping training sets, average their predictions
  
  ▶ Sequential training, iteratively re-weighting training examples so current classifier focuses on hard examples: **boosting**
Ensemble methods: Summary

- Differ in training strategy, and combination method
  - Parallel training with different training sets
    - Bagging (bootstrap aggregation) – train separate models on overlapping training sets, average their predictions
  - Sequential training, iteratively re-weighting training examples so current classifier focuses on hard examples: boosting
  - Parallel training with objective encouraging division of labor: mixture of experts
Ensemble methods: Summary

- Differ in training strategy, and combination method
  - Parallel training with different training sets
    - **Bagging** (bootstrap aggregation) – train separate models on overlapping training sets, average their predictions
  - Sequential training, iteratively re-weighting training examples so current classifier focuses on hard examples: **boosting**
  - Parallel training with objective encouraging division of labor: **mixture of experts**

- Notes:
  - Differ in: training strategy; selection of examples; weighting of components in final classifier