Using Testing to Iteratively Improve Training: Evaluating What Has Been Learned So Far

Peter Stone
Manuela Veloso

Computer Science Department
Carnegie Mellon University

Introduction

- For Active Learning, **Testing** is just the beginning
- Find difficult **regions** of the instance space
  - "Tricky" regions
  - Regions with not enough training
- Testing is **what-do-I-need-most-help-with?** phase in an **I-can-get-better!** process
- First our general paradigm
- Then a successful application

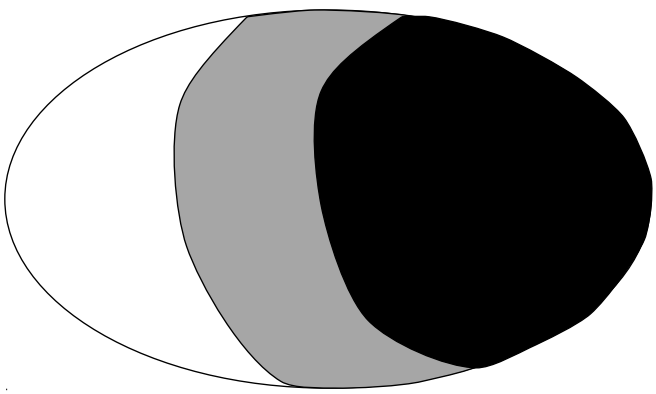
A General Active Learning Paradigm

---

1. **Gather training** data.
2. Use a Machine Learning algorithm to **learn** from the training data.
3. **Gather testing** data.
4. **Test** the performance of the learner.
5. If generalization performance is adequate, continue testing **(Goto Step 3)** or stop.
6. Using the results of testing, **analyze which regions** of the instance space are covered least well by the learner.
7. Gather additional training data from a skewed distribution of the instance space, **favoring those regions discovered in Step 6.**
8. **Combine** both sets of training data.
9. Goto Step 2.

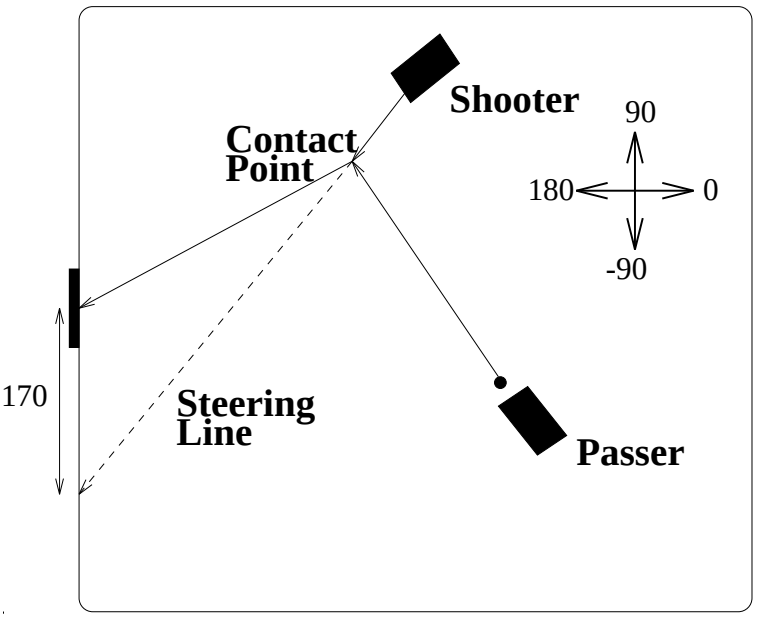
Paradigm Features

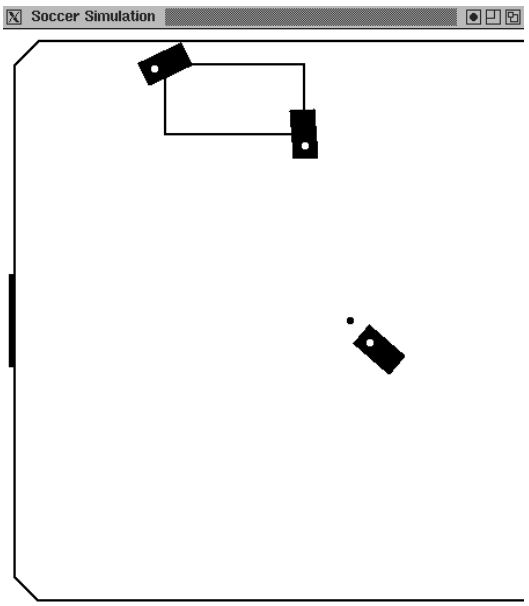
- Useful with many ML techniques
  - Need ability to train in batch mode
  - Need ability to query instance space for training examples (Step 7)
- Step 6 is the key: *Analyze which regions ...*
- Mistakes in **testing** give clues
- Train on regions not learned but that **do occur** during testing



- Retain original training examples to keep **good** learning

Robot Soccer: A Case Study

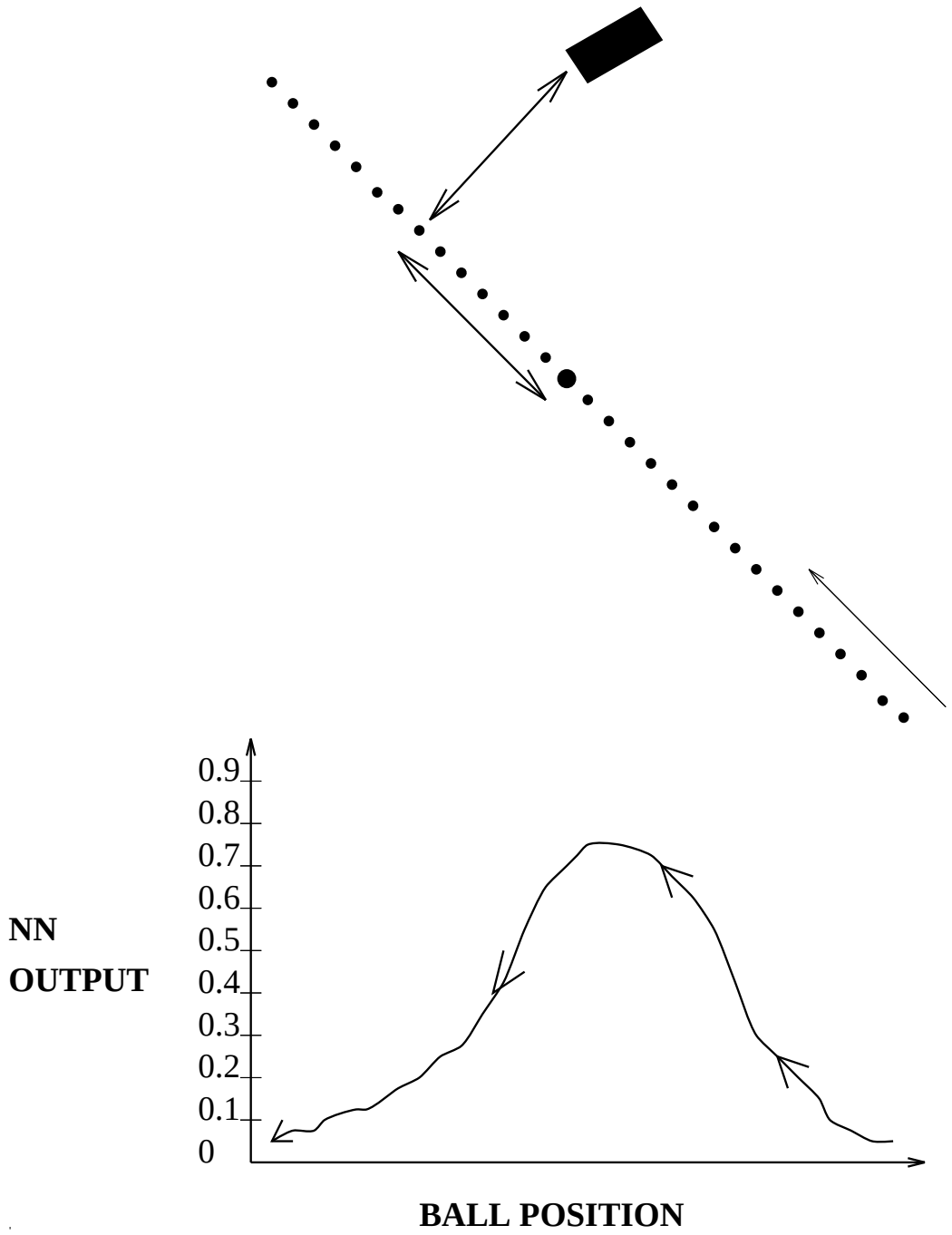




- Clear **topology:** "nearby" instances classified similarly
- Shooter position and ball speed vary

No direct access to the NN

The NN is not tested **directly**: success tied to **performance**

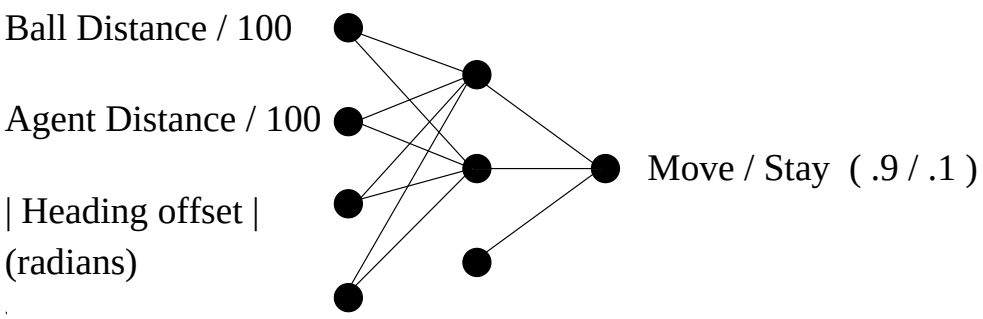


Paradigm Instantiation

---

1. Collect a training set by **randomly selecting** starting positions for the robot and speeds for the ball.
2. Use a neural network **(NN)** to learn.
3. Gather a random testing set **as in Step 1.**
4. **Test** the performance of the NN on the test set.
5. If performance is adequate, stop. (After the first iteration, performance is not adequate).
6. Use another NN to **learn** from the test data to **predict** which trials will be **successful.**
7. Use this second NN to **filter** possible training examples, choosing only examples likely to be unsuccessful. Construct an additional training set.
8. **Combine** this training set with the original training set.
9. Goto Step 2.

Combining old and new learning

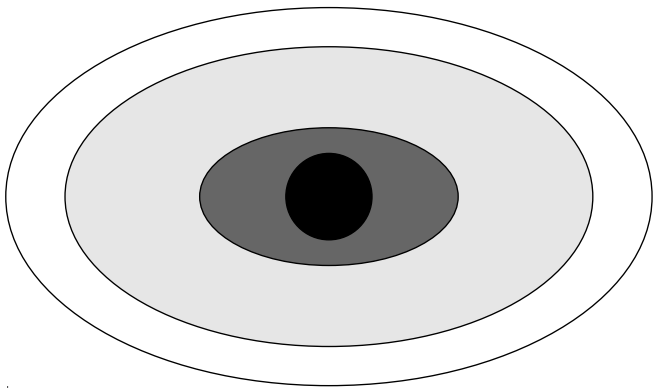


combining the old and new training sets:

1. Train a new net **from scratch** with old and new data
2. Train a new net with only new data, then **arbitrate**
3. Start with **old weights**, train only with new data
4. Start with **old weights**, train with old and new data

How learning ramps up

- Some **regions** of starting positions are "tricky"
  - Fast-moving ball
  - Bad shooter position
- Some regions are learned more easily



- Further iterations add new examples only from the tricky regions.

Results

---

1. Previously, with less variation in the instance space, an agent learned to score **95%** of the time: http://www.cs.cmu.edu/~pstone
2. Same methods on this space, agent scores **59%**
3. Trained **Active NN** to predict success of trials
   - Active NN output $< .1 \Rightarrow 1\%$ (21% of test set)
   - Active NN output $\geq .1 \Rightarrow 75\%$
   - Active NN output $< .5 \Rightarrow 12\%$ (42%)
   - Active NN output $\geq .5 \Rightarrow 91\%$
4. Training (from scratch) and testing only on examples giving **Active NN** $< .5$ raises 12% to **43%**
5. Arbitrating between the two NNs (Steps 2, 4) gives **70%**
6. Training with half old and half new data: **63%**
7. Training with all old and new data: **65%**
8. Using only the new NN (no arbitration): **71%!!!!**

Results: Iteration 2

---

1. Starting with a NN that scores 71%
2. Trained **Active NN2** to predict success of trials
   - Active NN output $< .5 \Rightarrow 17\%$ (29% of test set)
   - Active NN output $\geq .5 \Rightarrow 69\%$
3. Training (from scratch) and testing only on examples giving **Active NN2** $< .5$ raises 17% to **25%**
4. Arbitrating between the two NNs (Steps 1, 3) gives **73%**
5. Training with all old and new data: **76%**
6. Using only the new NN (no arbitration): **53%**

Confidence in Learning So Far

- The test set is the key
- Training data evolves, but the test set remains **i.i.d.: independent and identically determined**
- Can't keep using the same test set—over-fitting
- Performance during testing gives a direct measure of **confidence**

Conclusion

- **General** Active Learning Paradigm
- Specific **application** with results
- The **testing phase** indicates what **regions** on which to concentrate and gives a measure of **confidence** in what's been learned so far