

(Lack of) Deep Learning Theory

DIPS 2007

John Langford

The theorem statement we want

Conjecture: \exists a set of classification problems $\{D(x, y)\}$ such that:

1. \forall “shallow” classification algorithms \exists classification problem D such that when trained on samples from D , the error rate of the learned classifier is large.
2. \exists an efficient “deep” classification algorithm A such that \forall classification problems D , when trained on samples from D , A efficiently produces a high accuracy classifier.

Some Theorems To Guide our Intuition

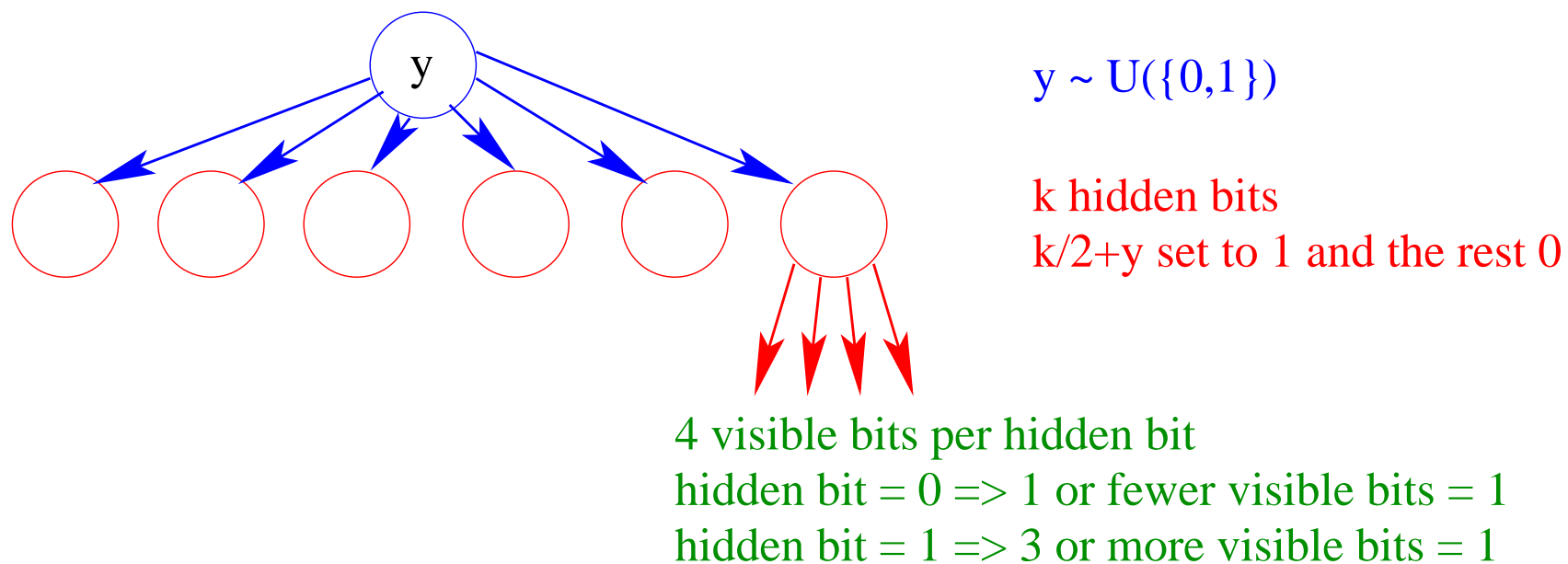
1. \forall learning problems D , \exists a two layer neural network with near optimal performance. (But it may be very inefficient to represent and learn.)
2. \exists Monotone functions computable by small depth- k circuits but which require very large depth $k - 1$ circuits. (Monotonicity is a very strong constraint.)
3. **XOR** is not representable by small constant depth circuits. (But hard to learn—some people use noisy XOR for crypto because it's so hard to learn.)

A Theorem Statement that we can make.

Conjecture: \exists a set of classification problems $\{D(x, y)\}$ such that:

1. \forall linear classification algorithms \exists classification problem D such that when applied to samples from D , the error rate of the learned classifier is large.
2. \exists an efficient “deep” classification algorithm which produces a small error rate classifier with high probability for all D .

A Tricky Classification Distribution Over (x, y) :



(We could recurse this construction to greater depth, if desired.)

Lower Bound

By symmetry, the best linear predictor uses uniform weights.

...but this predictor has a high loss because the influence of y on the number of **visible bits with value 1** is much smaller than the influence of noise.

(The negative statement also applies to kernels and many decision trees.)

Upper Bound

Consider the algorithm which first predicts each bit of x from the other bits, then make a linear prediction on x .

The only useful bits for predicting bits of x are **siblings**.

The optimal prediction for **visible bits given sibling visible bits** is the **hidden bit**.

Given the **hidden bits**, you can predict y perfectly with a linear predictor.

Sideways Progress

Given the difficulty of even representational lower bounds, sideways progress seems the best hope.

1. What is the set of all learning algorithms on deep structures? (Backprop + ??)
2. What computational tricks are useful for these algorithms?
3. How do we avoid overfitting with deeply conditional prediction problems?

Further Reading

1. <http://hunch.net> + search for “Deep Learning”.
2. Johan Hastad & Michael Goldmann, “On the Power of Small Depth Threshold Circuits”.