Learning Neural Networks with Distribution Shift: Efficiently Certifiable Guarantees

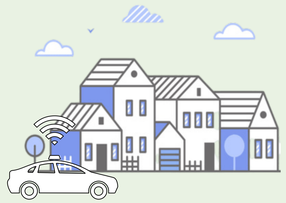
Gautam Chandrasekaran (UT Austin), **Adam Klivans** (UT Austin), **Lin Lin Lee** (UT Austin), **Konstantinos Stavropoulos** (UT Austin)

Learning with Distribution Shift

- The learner is given labeled examples from **training distribution**
- But evaluated on unknown **testing distribution** without labels

e.g.

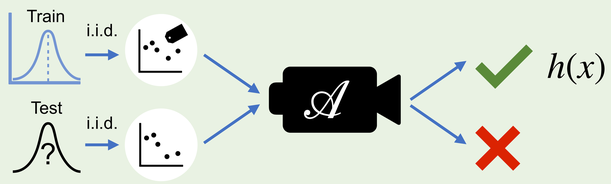
Learn risk of collision for self driving cars in big city



Deploy model in suburbs

When is it safe to trust learner's predictions?

Testable Learning with Distribution Shift



- **Soundness.** If $\mathscr{A}$ accepts, then the error of h on the test distribution is at most ϵ greater than the sum of the optimum training error and the best joint training and test error, i.e. at most $\text{opt} + \lambda + \epsilon$
- **Completeness.** If the training and test marginals are the same ($\mathscr{D}_x = \mathscr{D}'_x$), then $\mathscr{A}$ accepts

- Run efficient test to compare empirical training and test distributions
- Form hypothesis based on training examples
- Use results of test to evaluate hypothesis on test distribution
- In this work, we study **real-valued networks with arbitrary Lipschitz activations**

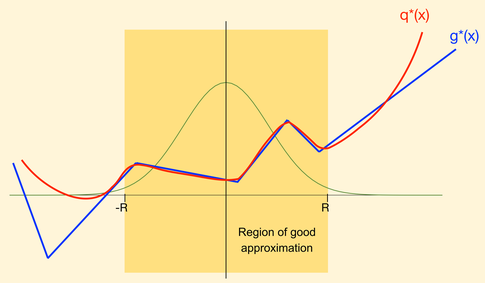
TDS for Strictly Subexponential Distributions

One Hidden-Layer Sigmoid Network	Single ReLU	Sigmoid Network	1-Lipschitz Network
$d^{\text{poly}(k\log(M/\epsilon))}$	$d^{\text{poly}(k\log(M/\epsilon))}$	$d^{\text{poly}(k\log M(\log(1/\epsilon)^{t-1}))}$	$d^{\text{poly}(k2^{t-1}\log(M/\epsilon))}$

Uniform Approximation Polynomial

For $g : \mathbb{R}^k \to \mathbb{R}$, $\epsilon > 0$, $R \geq 1$, we say $q : \mathbb{R}^k \to \mathbb{R}$ is an (ϵ, R)-uniform approximation polynomial for g if $\forall \|x\|_2 \leq R$:

$$|q(x) - g(x)| \leq \epsilon.$$



- Only need to approximate function well in region where distribution is highly concentrated
- Use tools from approximation theory to find polynomials which uniformly approximate function within a region

Algorithm

- Degree ℓ $\mathscr{L}_2$ polynomial regression on training data to find $\hat{p}$
- Degree 2ℓ moment matching between training and test distributions:

$$\mathbb{E}_{x\sim\mathscr{D}_x}[x^\alpha] \approx \mathbb{E}_{x\sim\mathscr{D}'_x}[x^\alpha]$$

TDS for Bounded Distributions

One Hidden-Layer Sigmoid Network	Single ReLU	Sigmoid Network	1-Lipschitz Network
$\text{poly}(d, M, 1/\epsilon)$	$\text{poly}(d, M) \cdot 2^{O(1/\epsilon)}$	$\text{poly}(d, M) \cdot 2^{O((\log 1/\epsilon)^{t-1})}$	$\text{poly}(d, M) \cdot 2^{\tilde{O}(k\sqrt{k}2^{t-1}/\epsilon)}$

- Obtain efficient algorithms using kernel method, which reduces the search space
- Analogous notion of uniform approximation: approximately represent function wrt kernel
- Representer theorem says polynomial approximation for ground truth function is linear combination of features depending on examples from training and test data

Algorithm

- Kernelized polynomial regression on training data to obtain hypothesis
- Test relative error closeness between covariance matrix of feature map ϕ over test and training marginal, i.e. compare

$$\Phi = \mathbb{E}_{x\sim\mathscr{D}_x}[\phi(x)\phi(x)^\intercal] \text{ and } \Phi' = \mathbb{E}_{x\sim\mathscr{D}'_x}[\phi(x)\phi(x)^\intercal]$$

Polynomial Approximation Theory

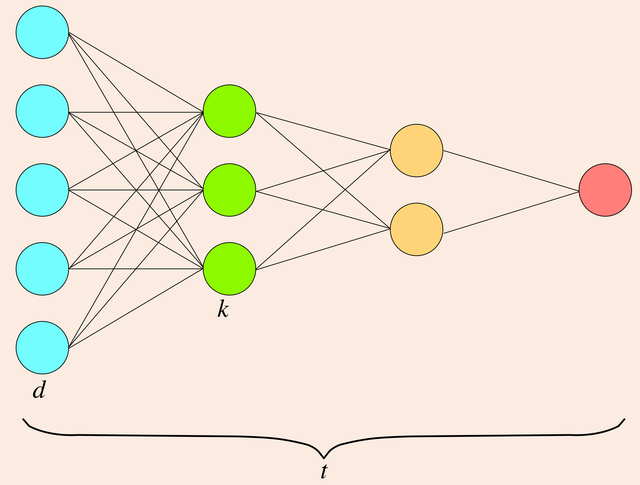
Jackson's Theorem

L-Lipschitz function $f : \mathbb{R}^k \to \mathbb{R}$ has (ϵ, R)-uniform approximation polynomial p of degree $O(LRk/\epsilon)$

+

BBGK18

If q is a polynomial such that $|q(x)| \leq b, \forall \|x\|_2 \leq R$, then sum of magnitudes of coefficients bounded



- View neural network as function of Wx after first layer of weights applied, to project from d-dimensional space to k-dimensional subspace
- For improved degree $O(\log(1/\epsilon))$ for sigmoid networks, replace each neuron with polynomial approximating sigmoid function
- Since output of sigmoid is bounded, after input layer the approximating polynomials can approximate to smaller radius