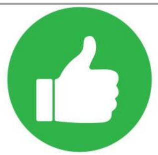
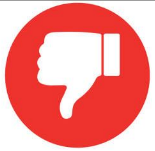
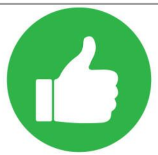
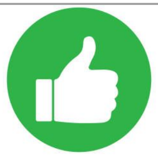
Privacy-Preserving Systems
(a.k.a., Private Systems)

CU Graduate Seminar

Instructor: Roxana Geambasu

Connections and Tradeoffs of Advanced Privacy Technologies

Threats and Tradeoffs of Privacy in ML

Privacy Tech	Threat	Strength of guarantee	Performance impact	Accuracy impact
Differential privacy	leakage of training data through models			
Homomorphic encryption	untrusted cloud's access to data during computation			
Hardware enclaves	untrusted cloud's access to data during computation			
Secure multi-party computation	untrusted cloud's access to data during computation			
Federated learning	untrusted cloud's access to data during computation			

Combinations Needed

- DP and the others address orthogonal threats, so for fuller protection, DP should be combined with all others
- Hardware enclaves can speed up homomorphic encryption and secure multi-party computation
- Federated learning has weak privacy, but can be combined with DP for strong privacy, with some loss in accuracy

Broader Connections

- Connections exist between privacy and other desirable properties of ML
- In theory, this could mean that technologies for one property could be useful for other properties
- Practical approaches to exploit these connections are still being researched

(NOTE: We started talking about these in the DP lecture, but we rushed and didn’t go into any details and all connections. We will discuss those today, but note that the slides are identical.)

Myriad of ML Concerns

Myriad of ML Concerns

Adversarial Examples

Explaining and Harnessing Adversarial Examples

Goodfellow, Shlens, Szegedy

Myriad of ML Concerns

Adversarial Examples

Explaining and Harnessing Adversarial Examples

Goodfellow, Shlens, Szegedy

Data Poisoning

Poisoning Attacks against Support Vector Machines

Biggio, Nelson, Laskov

Myriad of ML Concerns

Adversarial Examples

Explaining and Harnessing Adversarial Examples

Goodfellow, Shlens, Szegedy

Data Poisoning

Poisoning Attacks against Support Vector Machines

Biggio, Nelson, Laskov

Generalization

overfitting

Myriad of ML Concerns

Adversarial Examples

Explaining and Harnessing Adversarial Examples

Goodfellow, Shlens, Szegedy

Data Poisoning

Poisoning Attacks against Support Vector Machines

Biggio, Nelson, Laskov

Generalization

overfitting

Privacy Loss

The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks

Carlini, Liu, Erlingsson, Kos, Song

Myriad of ML Concerns

Adversarial Examples

Explaining and Harnessing Adversarial Examples

Goodfellow, Shlens, Szegedy

Data Poisoning

Poisoning Attacks against Support Vector Machines

Biggio, Nelson, Laskov

Generalization

overfitting

Privacy Loss

The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks

Carlini, Liu, Erlingsson, Kos, Song

Bias, Discrimination

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

Bolukbasi, Chang, Zou, Saligrama, Kalai

Myriad of ML Concerns

Adversarial Examples

Explaining and Harnessing Adversarial Examples

Goodfellow, Shlens, Szegedy

Privacy Loss

The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks

Carlini, Liu, Erlingsson, Kos, Song

Data Poisoning

Poisoning Attacks against Support Vector Machines

Biggio, Nelson, Laskov

Bias, Discrimination

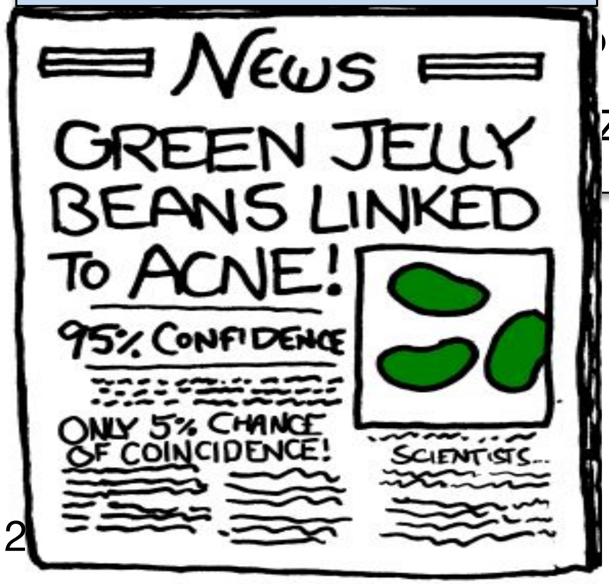
…ter Programmer as … Homemaker? …rd Embeddings

…Zou, Saligrama, Kalai

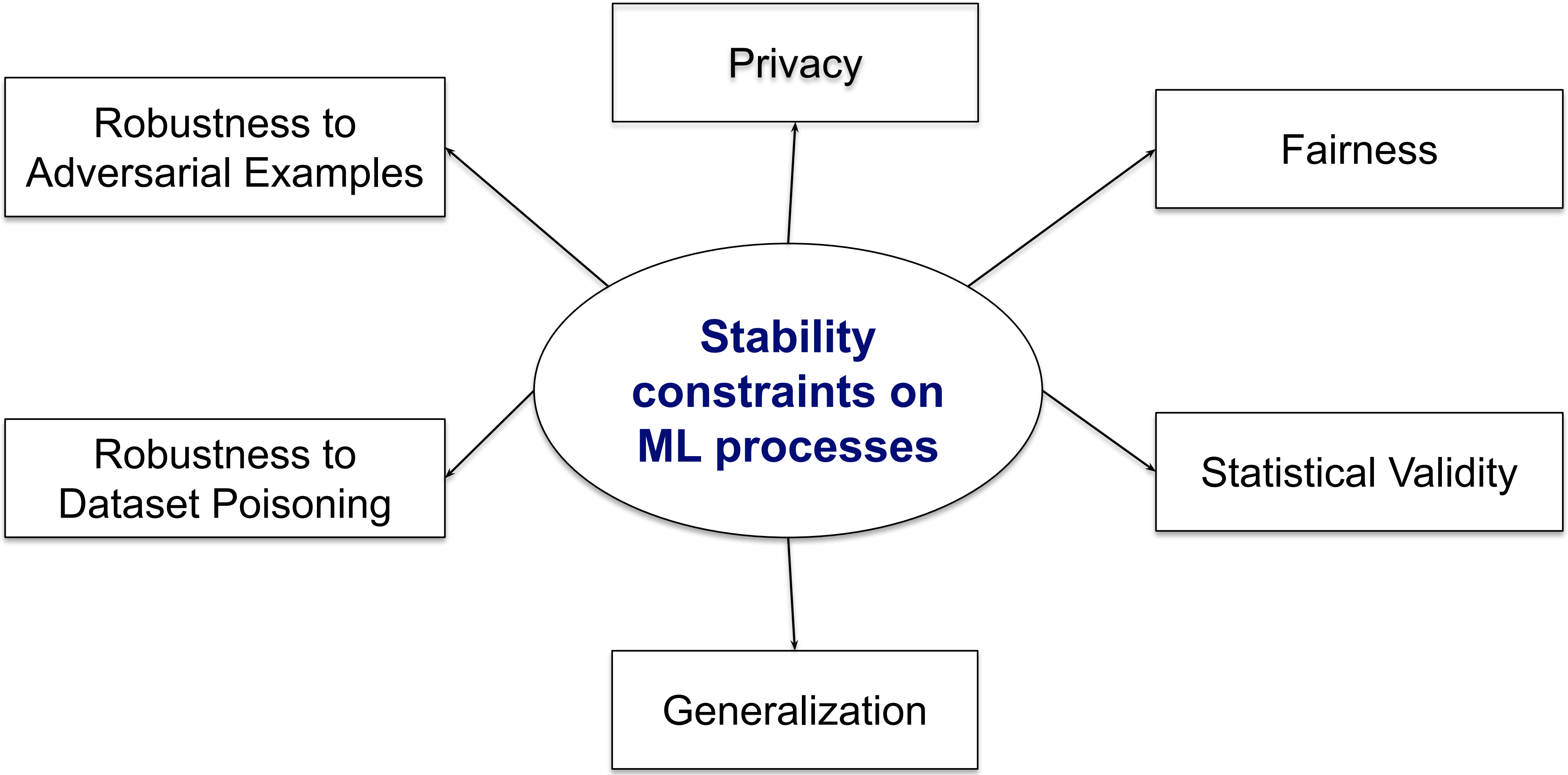
Generalization

overfitting

False Discoveries



Many Concerns Are Related



[Hardt-16]

Example: DP Improves More than Privacy

- DP is a **strong stability constraint** on computations running on datasets: it requires that no single data point in an input dataset has significant influence over the output

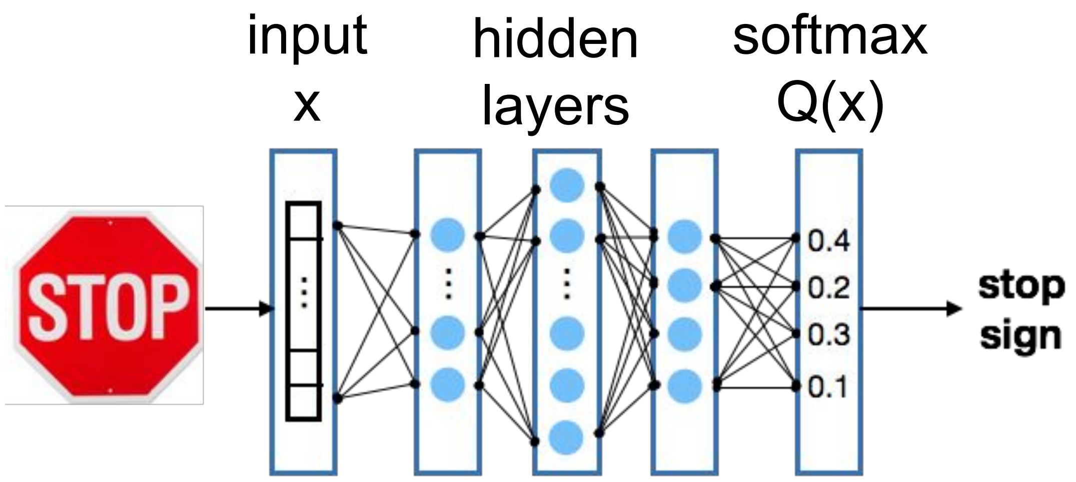
- It has been been shown to improve a variety of desirable ML properties beyond privacy, e.g.:
  - DP for Adversarial Robustness (Lecuyer+19)
  - DP for Generalization (Hardt-16, Bassily+16)
  - DP for Fairness (Dwork+13)
  - DP for Statistical Validity (Dwork+15)

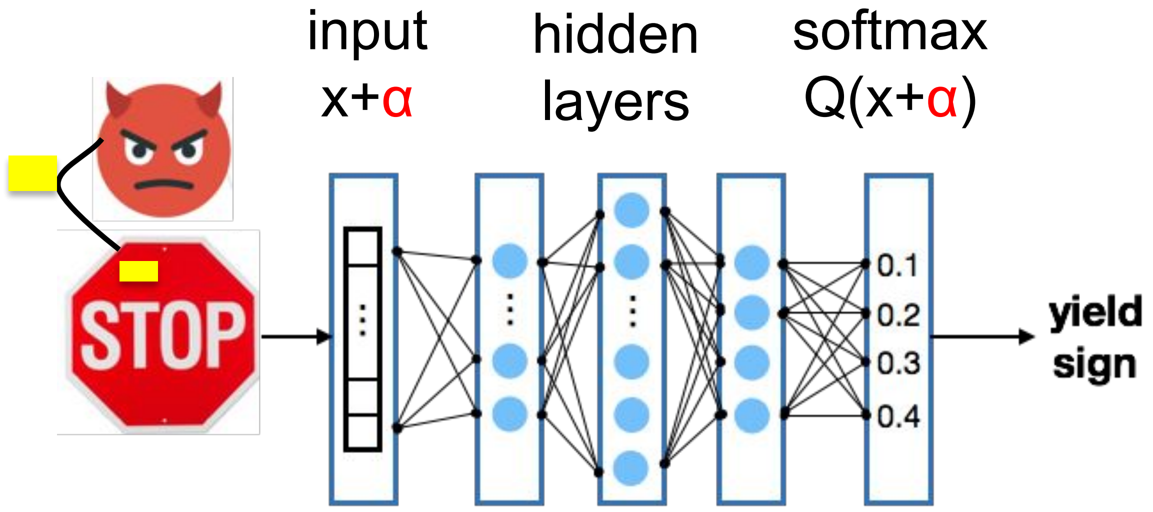
DP for Adversarial Robustness

(Lecuyer+19)

Adversarial Examples

- Adversary finds a tiny perturbation to a correctly classified input that causes misclassification



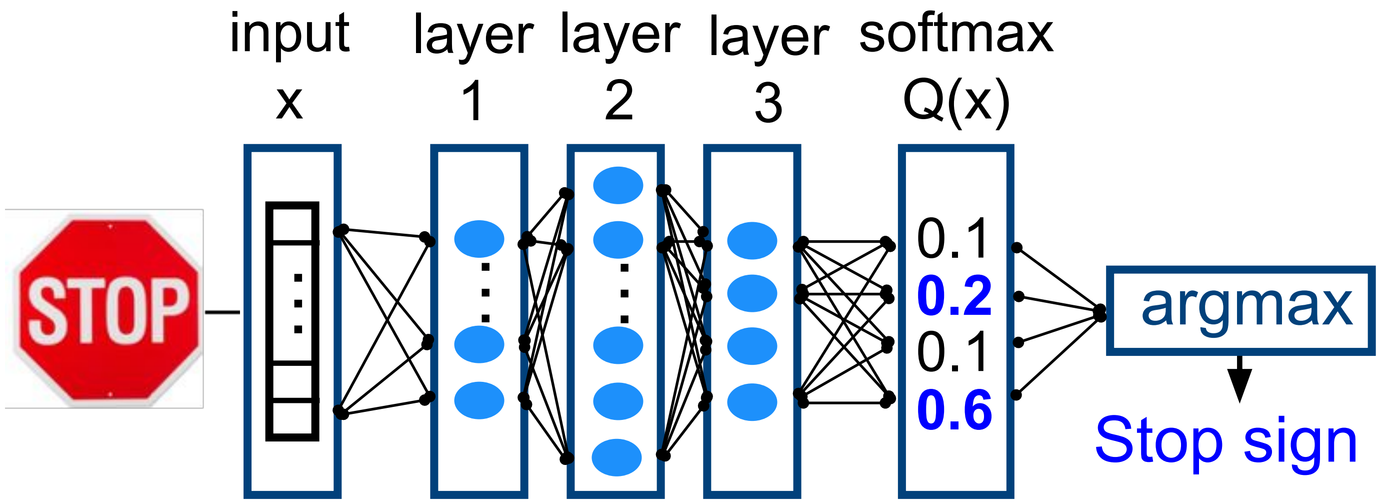


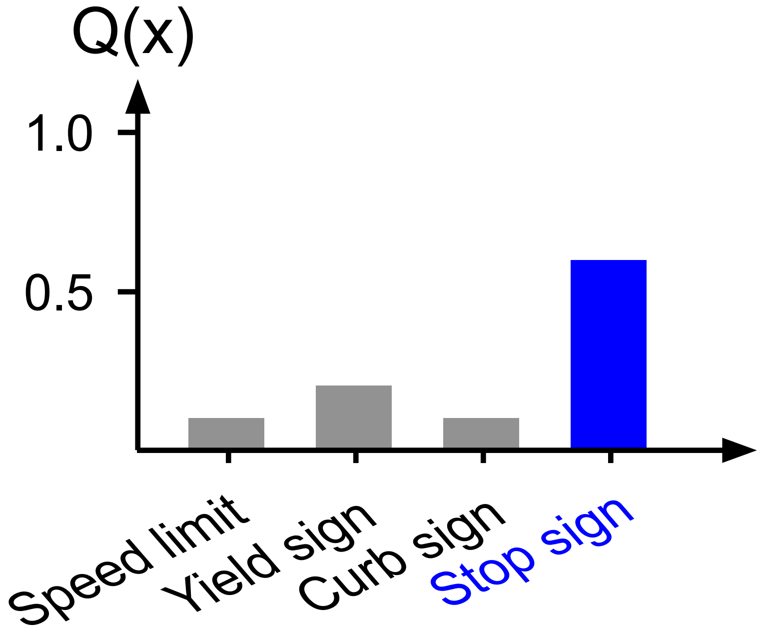
DP for Adversarial Examples

- Problem: small input changes create large score changes
- Approach: **make prediction function DP**

DP for Adversarial Examples

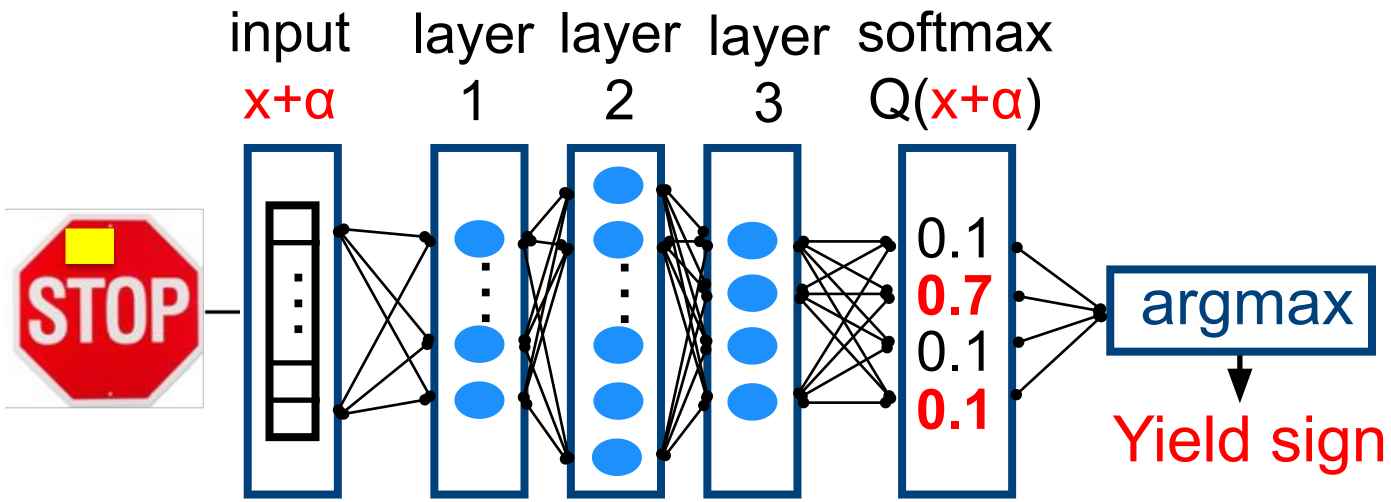
- Problem: small input changes create large score changes
- Approach: **make prediction function DP**

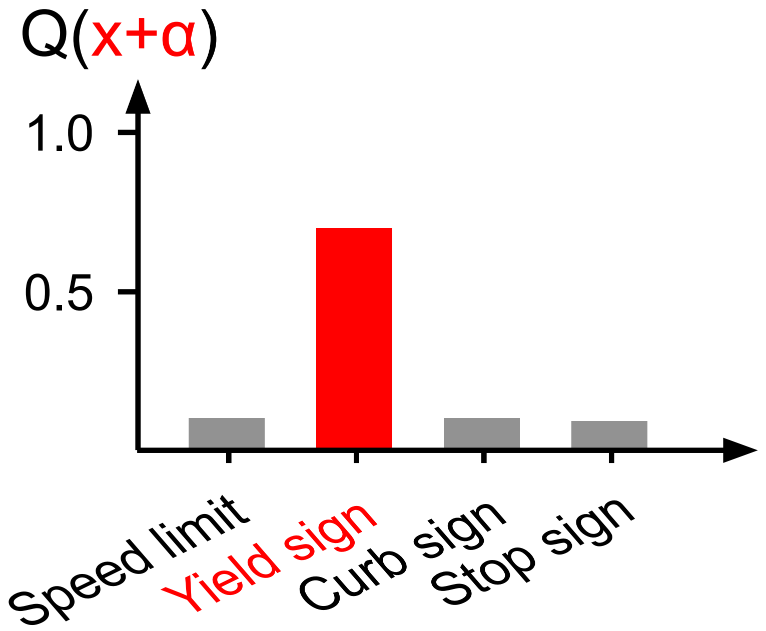




DP for Adversarial Examples

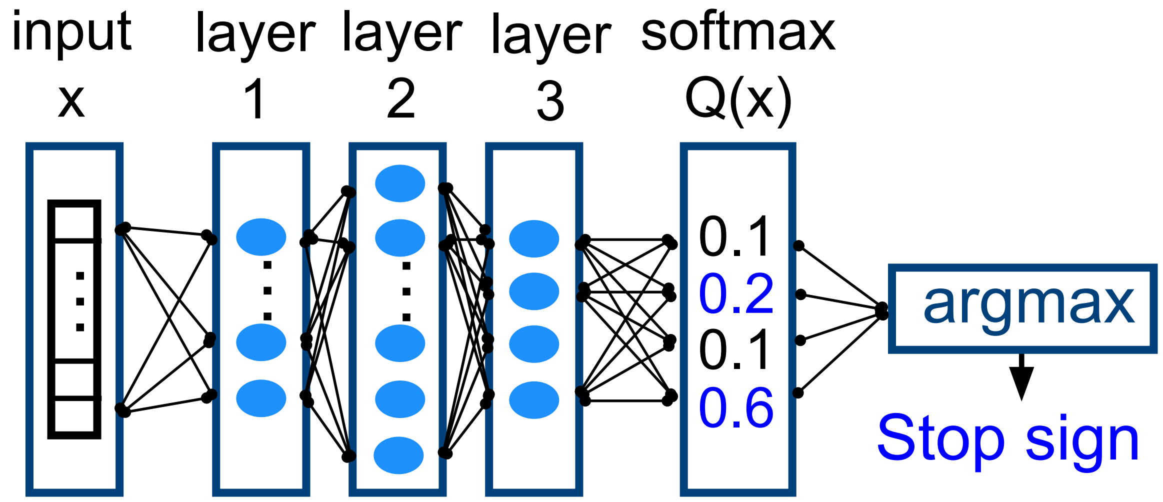
- Problem: small input changes create large score changes
- Approach: **make prediction function DP**





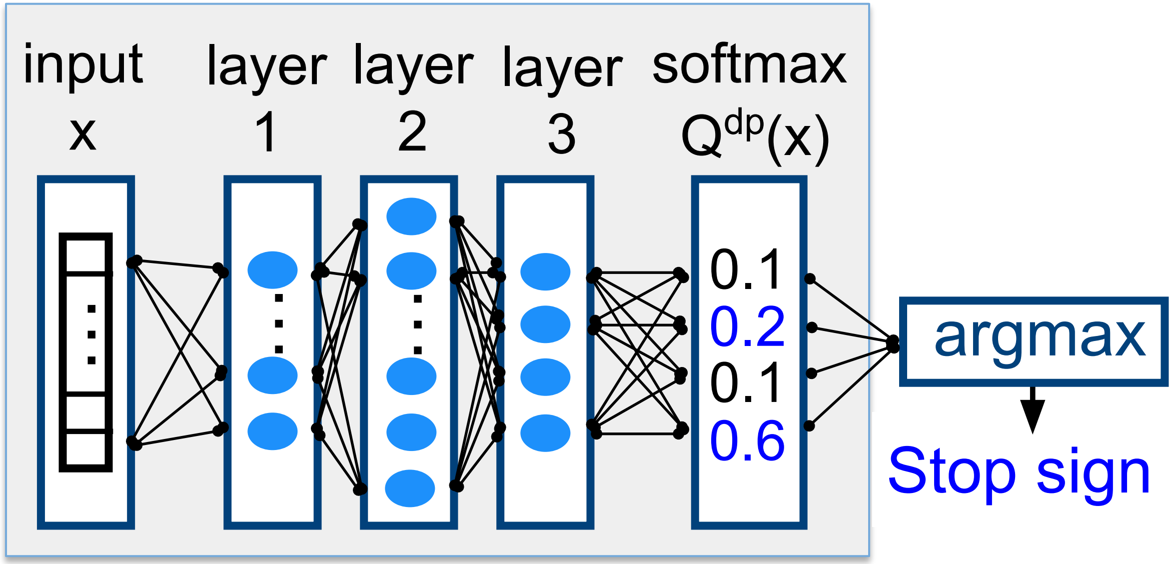
How It Works

1. Randomize prediction function to make it DP
2. Use expected scores to choose argmax
3. Use DP's stability bounds on expected scores to certify prediction on x



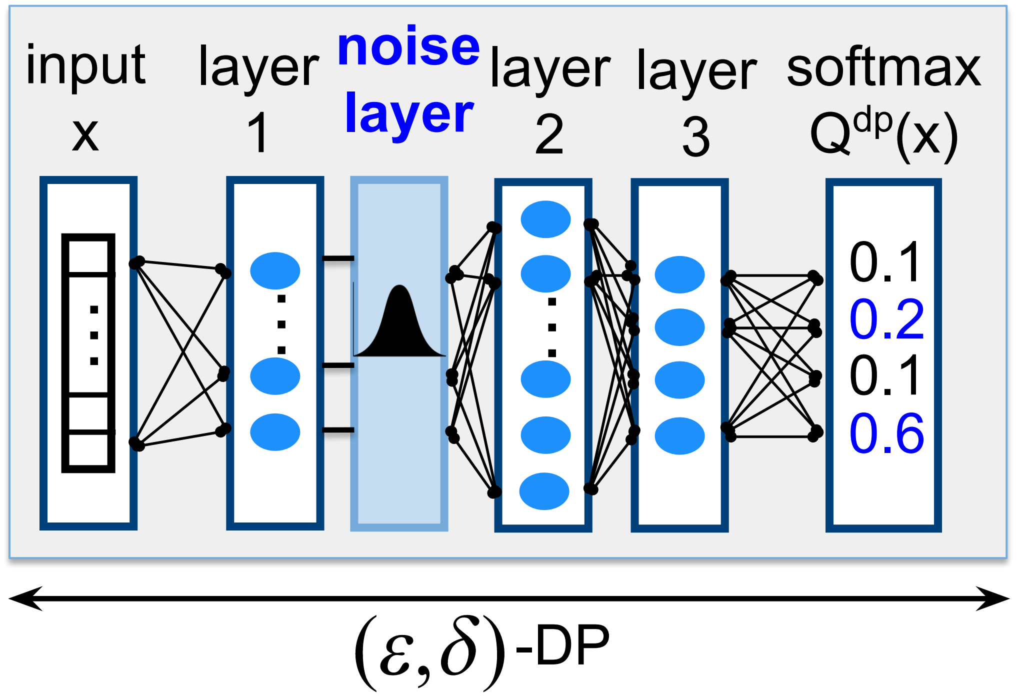
How It Works

1. Randomize prediction function to make it DP
2. Use expected scores to choose argmax
3. Use DP's stability bounds on expected scores to certify prediction on x



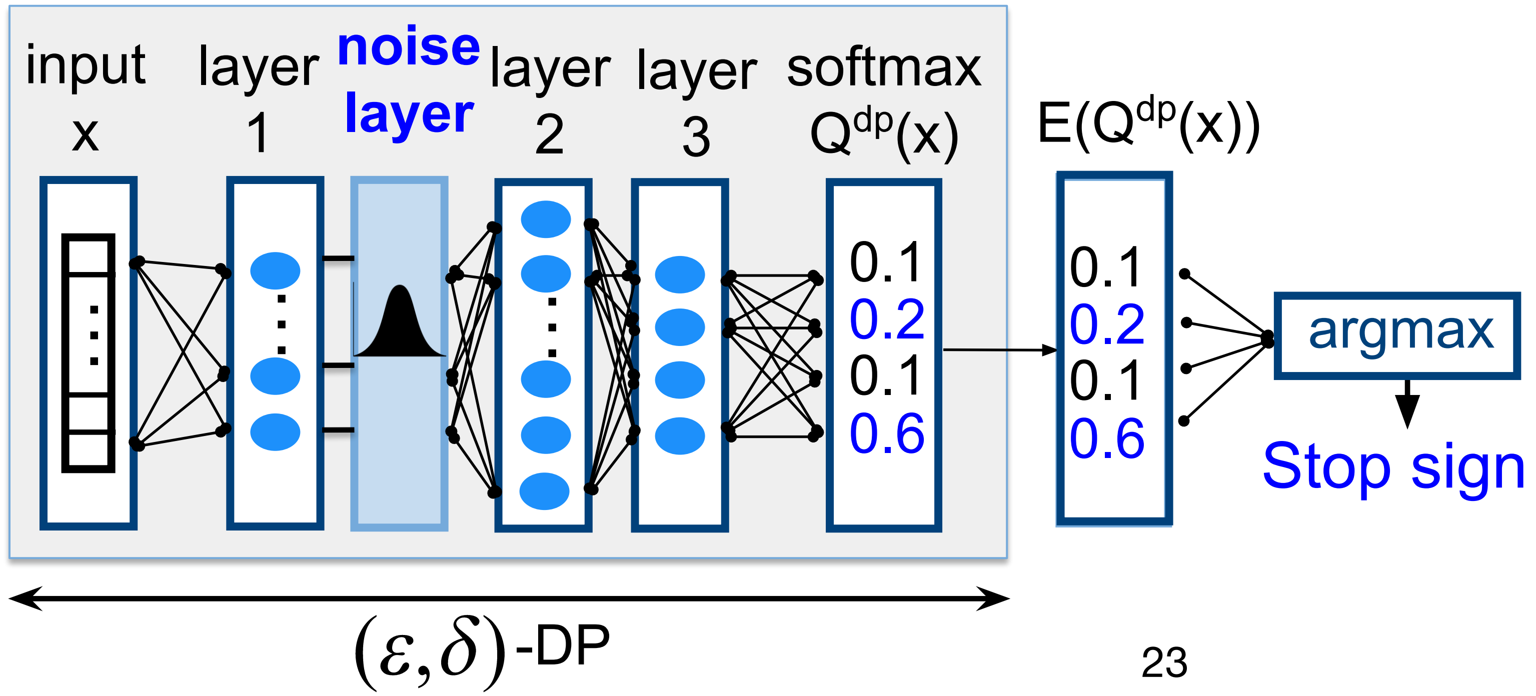
How It Works

1. Randomize prediction function to make it DP
2. Use expected scores to choose argmax
3. Use DP's stability bounds on expected scores to certify prediction on x



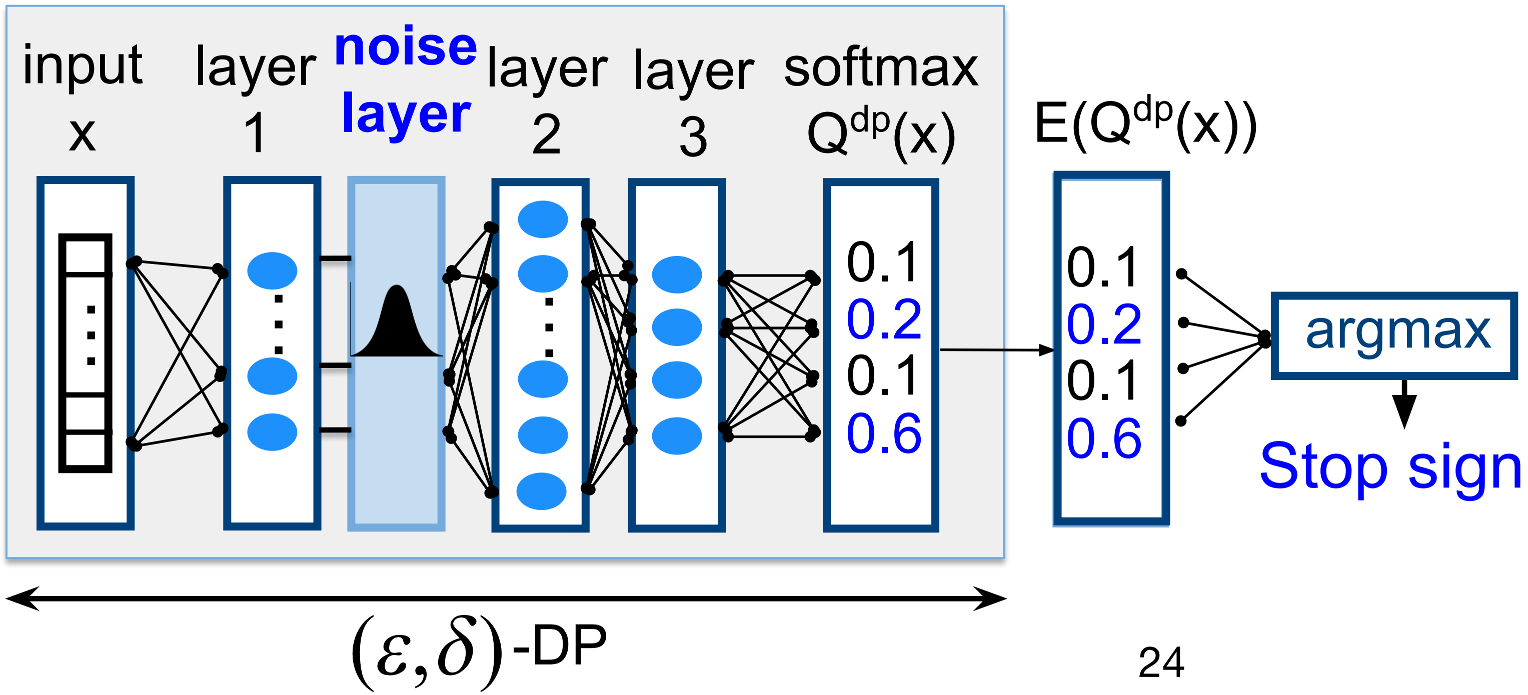
How It Works

1. Randomize prediction function to make it DP
2. Use expected scores to choose argmax
3. Use DP's stability bounds on expected scores to certify prediction on x



How It Works

1. Randomize prediction function to make it DP
2. Use expected scores to choose argmax
3. Use DP's stability bounds on expected scores to certify prediction on x



DP's stability bounds on expected scores:

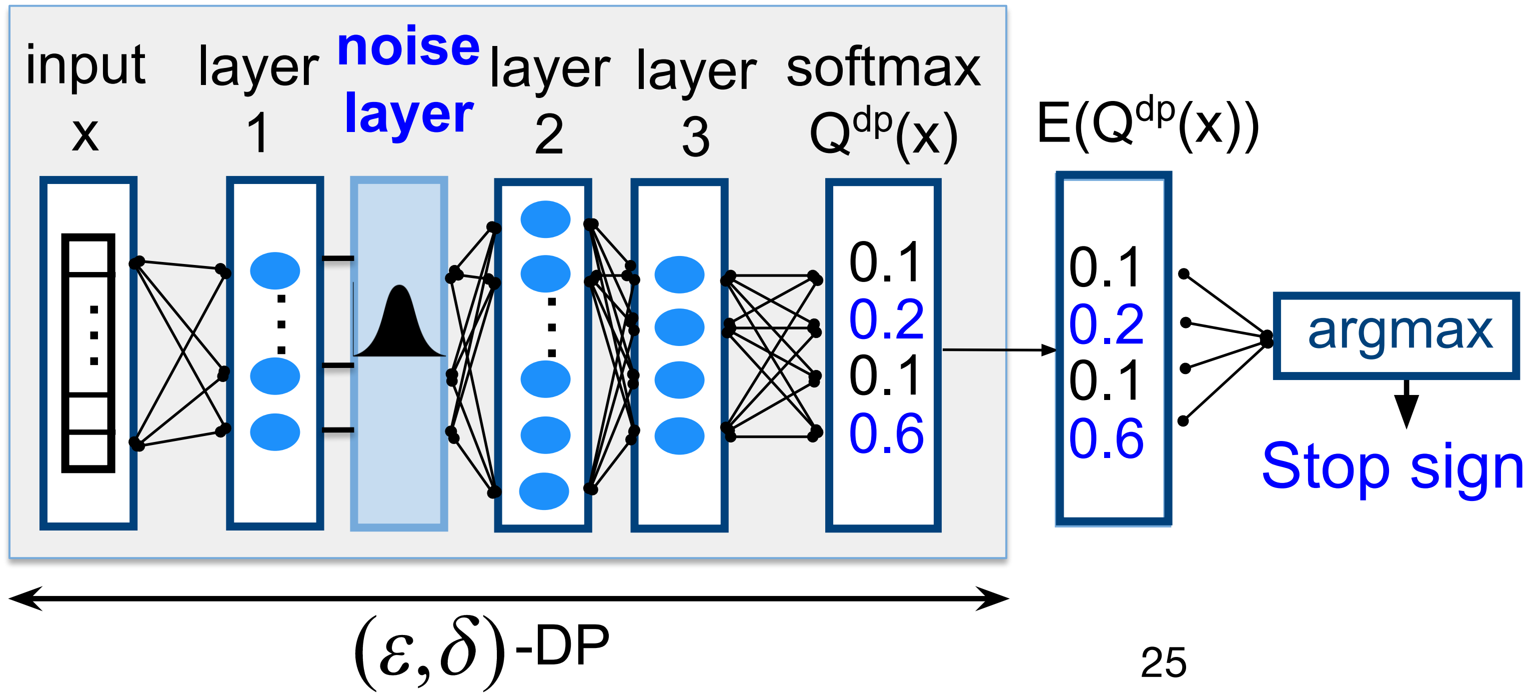
for all α with $\|\alpha\|_2 = \sqrt{\sum \alpha_i^2} \leq L$:

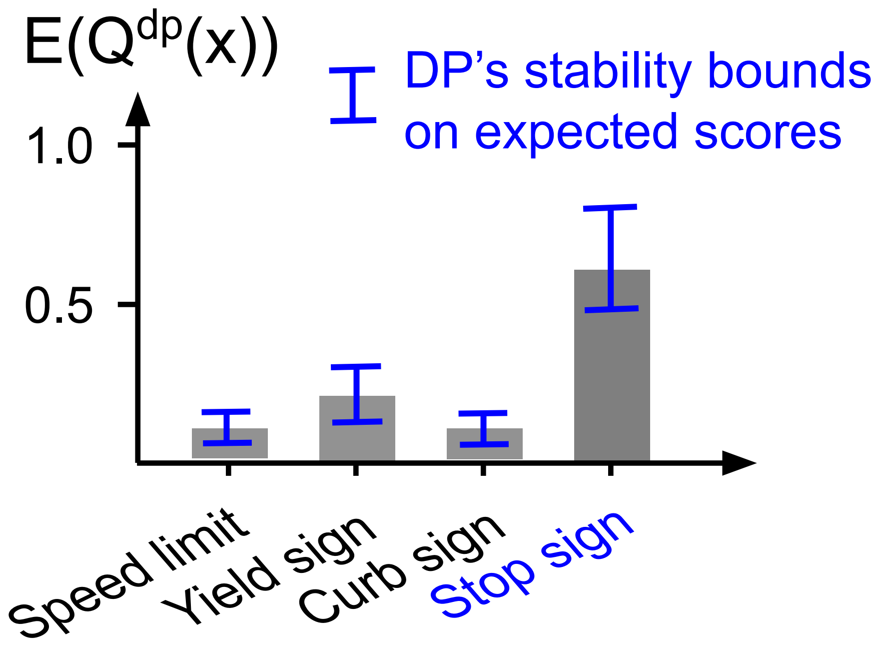
$$\frac{E(Q^{dp}(x)) - \delta}{e^{\varepsilon}} \leq E(Q^{dp}(x+\alpha))$$

$$E(Q^{dp}(x+\alpha)) \leq e^{\varepsilon} E(Q^{dp}(x)) + \delta$$

How It Works

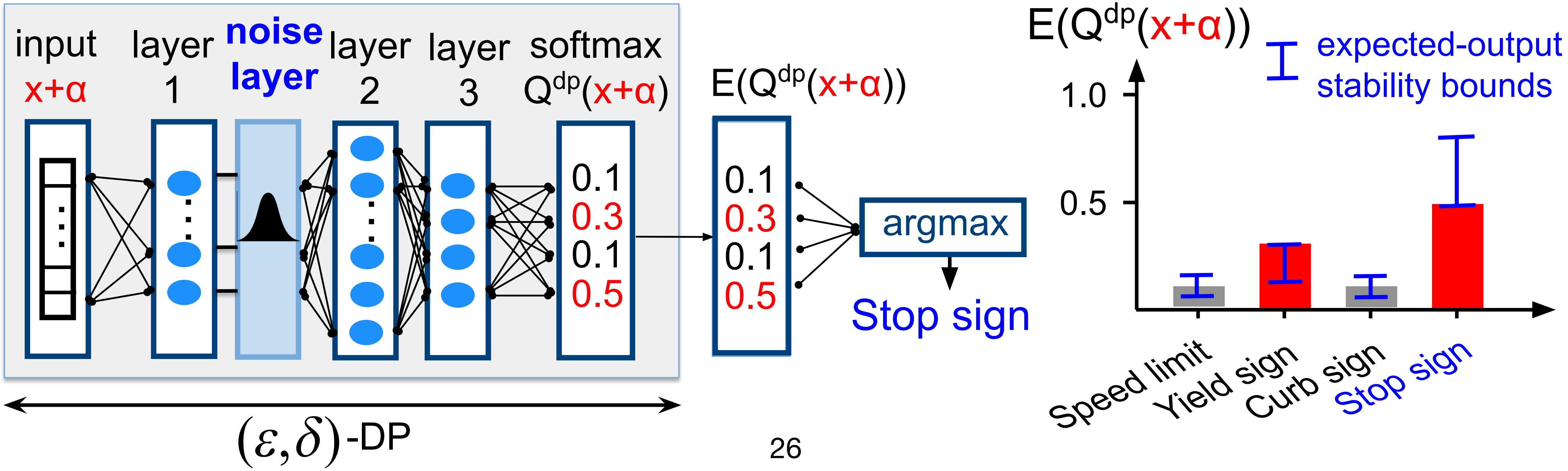
1. Randomize prediction function to make it DP
2. Use expected scores to choose argmax
3. Use DP's stability bounds on expected scores to certify prediction on x





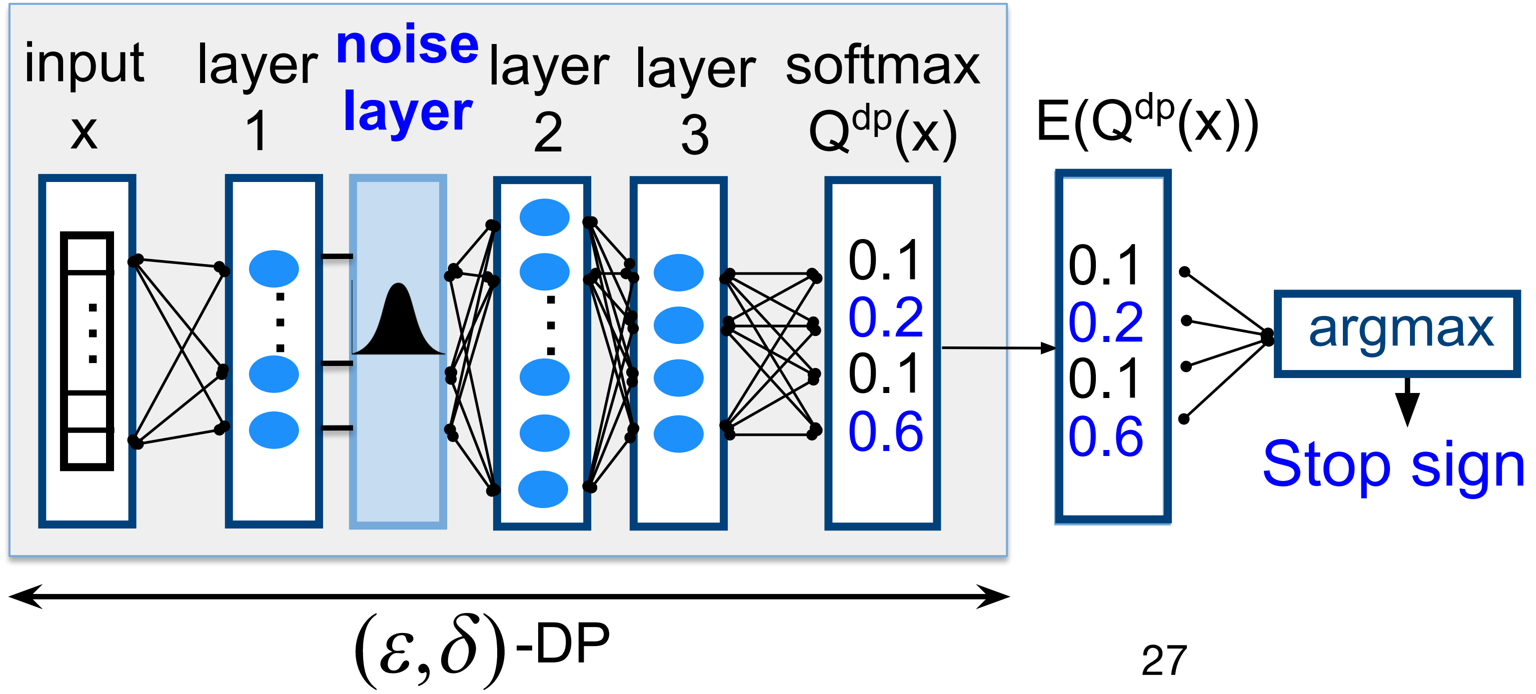
How It Works

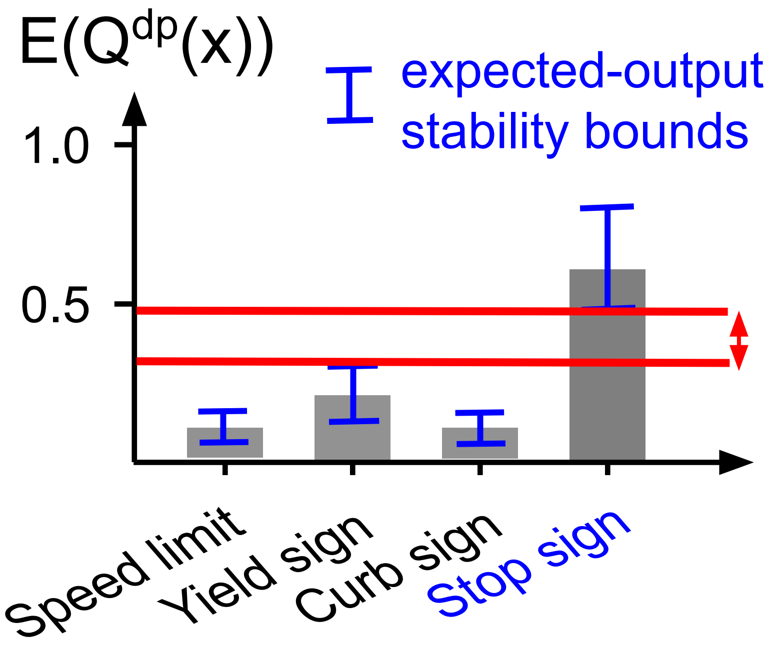
1. Randomize prediction function to make it DP
2. Use expected scores to choose argmax
3. Use DP's stability bounds on expected scores to certify prediction on x



How It Works

1. Randomize prediction function to make it DP
2. Use expected scores to choose argmax
3. Use DP's stability bounds on expected scores to certify prediction on x





DP for Generalization

(Hardt-16)

Generalization

- Central to ML is our ability to relate how a learning algorithm fares on a sample set to its performance on unseen instances. This is called **generalization**

Generalization

- Central to ML is our ability to relate how a learning algorithm fares on a sample set to its performance on unseen instances. This is called **generalization**

Risk (Out-of-sample Error)	Empirical Risk (Train Error)	**Generalization Error**
$R = \mathop{\mathrm{E}}_{z\sim D}\left[\ell(A(S),z)\right]$	$R_S = \frac{1}{n}\sum_{i=1}^{n}\ell(A(S),s_i)$	$R - R_S$

A= training function; D= input distribution; S= training set; n=|S|; ℓ = loss function

Generalization

- Central to ML is our ability to relate how a learning algorithm fares on a sample set to its performance on unseen instances. This is called **generalization**

Risk (Out-of-sample Error)	Empirical Risk (Train Error)	**Generalization Error**
$R = \mathop{\mathrm{E}}_{z\sim D}\left[\ell(A(S),z)\right]$	$R_S = \frac{1}{n}\sum_{i=1}^{n}\ell(A(S),s_i)$	$R - R_S$

A= training function; D= input distribution; S= training set; n=|S|; ℓ = loss function

- We care about R. If we manage to minimize R_S, all that matters is the **generalization error**. Many approaches exist that improve generalization error (mostly statistical)

Generalization $\Leftrightarrow$ Stability

- **Thm: In expectation, generalization equals stability**
    - Proof in (Hardt-16)

- An algorithm is **stable** if its output doesn't change much if we perturb the input sample in a single point

- The theorem says that stability is **necessary and sufficient** for generalization

DP for Generalization

- DP is a strong stability constraint on algorithms
- DP thus provides an algorithmic approach to generalization in ML: **make the training function DP**
- It’s been long known that adding randomness into training improves generalization
- The level of randomness added is likely insufficient to offer meaningful privacy, but the link DP<->generalization suggests that privacy isn’t fundamentally at odds with functionality in ML

DP for Fairness

(Dwork+13)

Individual Fairness

- People who are similar from the perspective of the task at hand should be treated similarly
    - E.g., people with similar capabilities w.r.t. to a graduate program should all be either admitted or rejected

- But in ML, because of data biases and algorithmic amplification of them, small changes in people's relevant capabilities can lead to large changes in the predictions

- That's a sign of instability of the prediction function

35

DP for Individual Fairness

- Approach: **make the prediction function DP**
  - Similar to PixelDP, apply extension of DP to a distance metric among people with respect to their abilities for a task

- While in theory interesting, this approach is not very practical because it relies on a good distance metric among people, which is hard to define

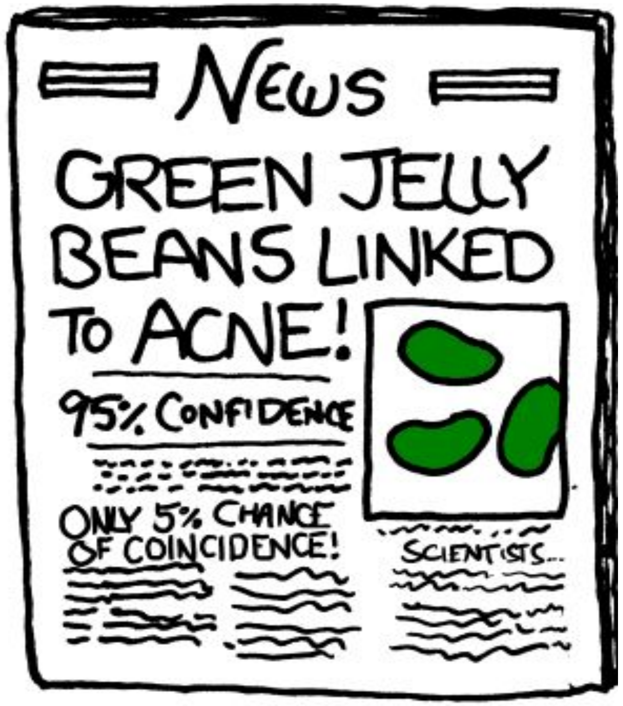
36

DP for Statistical Validity

(Dwork+15)

False Discoveries

- **Ideal scientific method:** Formulate your hypothesis, design your experiment to collect data, test your hypothesis on the data, report finding if statistically significant, and throw away the data.
- **In reality:** data is collected and reused to refine hypotheses, and the new hypotheses are tested on the same data, multiple times.
- Adaptive data reuse breaks assumptions of independence between hypotheses and test data, which hypothesis tests make to ensure statistical validity of the results. Referred to as p-hacking.



A Baseline Approach

- A baseline approach to allow statistical validity on top of a dataset collected from one study is to **split the dataset** into k components, where k is the number of hypotheses you anticipate testing on that dataset adaptively
- Each hypothesis runs on n/k points, so you can only run **k<<n adaptive hypothesis tests** on a dataset of size n
- Can we do better?

39

DP for Statistical Validity

- Problem: you're learning too much from the dataset, therefore your conclusions may overfit it and inherit its biases
- Approach: **make hypothesis tests DP and run on entire dataset**
- Recall DP supports adaptive composition. If you formulate a new hypothesis based on the results of a DP statistical test, and then you test again on the same dataset, you still have a bound on how much information you've extracted from your observations
- You can thus bound the number of tests you can perform while maintaining statistical validity. With advanced composition, the number of adaptive tests you can afford to run is **O(n^2)**

Take-Aways

- Many challenges in ML can be attributed to instability of some algorithm involved in learning: training, prediction, testing

- DP is a very strong stability constraint on algorithms. It thus has broad connections with many desirable properties in ML:
    - Training set privacy: make training function DP
    - Adversarial robustness: make prediction function DP
    - Generalization: make training function DP
    - Fairness: make prediction function DP
    - Statistical validity: make hypothesis test or model evaluation DP

- However, DP may be overly strong for some of these, and that impacts accuracy! Balance is needed, and future research may provide that

Cited References

(Bassily+16) R. Bassily, K. Nissim, A. Smith, T. Steinke, U. Stemmer, and J. Ullman. *Algorithmic stability for adaptive data analysis*. STOC 2016

(Dwork+15) C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, A. Roth. *Preserving Statistical Validity in Adaptive Data Analysis*. STOC 2015

(Hardt-16) M. Hardt. *Stability as a foundation for machine learning.* Blog post, 2016

(Lecuyer+19) M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, S. Jana. *Certified Robustness to Adversarial Examples with Differential Privacy*. IEEE Security & Privacy, 2019

Cited References

(Vadhan, 2016) Vadhan. *The complexity of differential privacy.* https://privacytools.seas.harvard.edu/files/privacytools/files/complexityprivacy_1.pdf.

The End