CS 182 Lecture 20: Adversarial Examples 7/19

Even if DNNs mistakes can make sense...
↳ trained to observe patterns.
Metaphor for ML → Clever Hans.
↳ right answers, wrong pattern! We asked the wrong questions...

Do produced probabilities reflect actual frequencies?

Distribution shift: test inputs might come from diff. distributions v. training inputs.
↳ spurious correlations in training data, selection bias, system feedback (!)
↳ "out-of-distribution" images → confident but wrong assertions.
↳ calibration struggles, for OOD & ID images.

Adversarial example: particularly vivid illustration of how learned models may or may not generalize correctly.
→ very special patterns can change models classification drastically!
→ not easy to defend against (no "bulletproof" method!)
→ can transfer across networks (in many cases) and work in real life.

overfitting hypothesis? Too complex, easy to find inputs that produce crazy outputs.
↳ evidence suggests this is false, since A-Es are not cause of high variance.

Linear models hypothesis? NNs ≈ locally linear, extrapolate in somewhat counterintuitive ways when diverging from data.
↳ "Experiment 2": clean "shift" on one side for adversarial direction. Therefore (?) mostly linear decision boundary

Human Adversarial Examples? Naive examples: optical illusions.

Adversarial Examples: features of a model's learning approach. Not bugs.

How to construct an adversarial example?
→ relation: $R(x, x')$
→ attack: $x^* \leftarrow \arg\max_{x' : R(x,x') \le \epsilon} L_\theta(x', y)$
(pick image close to x; maximize)

How to defend?
→ defense: $\theta^* \leftarrow \arg\min_\theta \sum_{(x,y)\in D} \max_{x' : R(x,x') \le \epsilon} L_\theta(x', y)$
(robust loss)

Fast gradient sign method (FGSM) - simple approx. method for ∞-norm relation.
$R(x, x') = \|x - x'\|_\infty$
"first-order" assumption: $L(x', y) \approx L(x, y) + (x' - x)^T \nabla_x L$ ← 1st-order Taylor expansion!
↳ local linear behavior of NN causes this to be effective!
→ attack: $x^* \leftarrow \arg\max_{x' : \|x - x'\|_\infty \le \epsilon} (x' - x)^T \nabla_x L$.
$x^* = x + \epsilon \operatorname{sign}(\nabla_x L)$
works well against naive nets.
→ can implement Lagrange multiplier: $x^* \leftarrow \arg\max_{x'} L_\theta(x', y) - \lambda R(x, x')$ (can optimize.)
$\delta \leftarrow x' - x$
$\delta^* \leftarrow \arg\max_\delta L_\theta(x + \delta, y) - \lambda \|\delta\|_\infty$

Transferability: we can simply use another NN to construct our AE.
↳ zero-shot black-box attacks...

Finite differences gradient estimation: use a moderate number of queries.
for each dimension i of x: (small num.)
→ set $v_i \leftarrow L(x + 10^{-3} e_i, y)$ (i^{th} canonical vector)
→ $\nabla_x L \approx (v - L(x, y)) / 10^{-3}$.
further "tricks" exist.

Defending: adversarial training:
1. sample minibatch $\{(x_i, y_i)\}$ from D
2. for each x_i, compute x_i' (FGSM, etc.)
3. take SGD step: $\theta \leftarrow \theta - \alpha \sum_i \nabla_\theta L_\theta(x_i', y_i)$
(usually also incorporate original loss gradient)
↳ cannot decrease overall accuracy on test set ("pay the price").