CS 7150: Deep Learning — Summer-Full 2020 — Paul Hand

Week 5 — Preparation Questions For Class
Due: Monday June 8, 2020 at 12:00 PM Eastern time via Gradescope

Name: Saurabh Vaidya
Collaborators: [Put Your Collaborators Here]

You may consult any and all resources in answering the questions. Your goal is to have answers that are ready to be shared with the class (or on a hypothetical job interview) as written. **Make sure to tag each question when you submit to Gradescope.**

Directions: Read the articles 'Explaining and Harnessing Adversarial Examples' and 'Robust Physical-World Attacks on Deep Learning Models'.

Question 1. *What is an adversarial example in the context of classification? Why are they a significant issue?*

Response:

Answer 1. *In the context of classification, adversarial examples are those which make slight perturbations in the input of a data point belonging to some class without changing the identity of the data point. When given such perturbed input example to a classifier, the classifier returns a wrong label with a very high confidence.*

Thus an adversarial example fools the classifier into misclassifying with very small but calculated modifications to input data point. This exposes a flaw in what important aspects the alogrithm has missed to learn. These adversarial examples can be used to misguide certain critical domain machine learning models and have severe impact through wrong decisions.

Question 2. *What is the process for computing an adversarial example using the fast gradient sign method? Be clear to specify what inputs to this process are needed.*

Response:

Answer 2. *The inputs needed for this model are an input data point, its true label and a trained model. Let x be our original input, $\widehat{x}$ be our perturbed adversarial input, x_i be i^{th} feature of our input x and y be its true label.*

Let θ be our trained weight parameters and $J(\theta, x, y)$ be our loss function. The magnitude of our perturbation would be denoted by η. We will define ϵ to be a small constant which would be the max norm for our perturbation η such that $\|\eta\|_\infty < \epsilon$.

- *Given those inputs, we compute the gradient of loss function with respect to each input feature $\nabla_x J(\theta, x, y)$.*
- *Based on the sign of the this gradient, we compute our perturbation to feature i as $\eta_i = \epsilon * sign(\nabla_x J(\theta, x, y))$*

- *Our adversarial example's input feature will be* $\widehat{x_i} = x_i + \eta_i$

This way we generate an adversarial example $\widehat{x} = x + \eta$

Question 3. *Explain a way in which a classifier can be trained to be more robust to adversarial perturbations.*

Response:

Answer 3. *One way of training models to be more robust to adversarial perturbations would be to add these adversarial examples to our training dataset. This way we can augment our data just like we do with techniques like translation, rotation etc.*

A rather effective way to train for adversarial perturbations would be to incorporate these perturbations in our cost function. Thus our cost function can be modified to

$$\widehat{J}(\theta, \mathrm{x}, \mathrm{y}) = \alpha J(\theta, \mathrm{x}, \mathrm{y}) + (1 - \alpha) J(\theta, \mathrm{x} + \epsilon \mathrm{sign}(\nabla_x J(\theta, x, y))) \tag{1}$$

Thus after each training epoch, the network has to minimize the prediction cost while also minimizing cost due to adversarial perturbations.

Question 4. *What is the evidence that adversarial examples generated for a specific model are 'often misclassified by other models, even when they have different architectures or were trained on disjoint training sets'? [This quote is from the Goodfellow et al. paper.]*

Response:

Answer 4. *Under linear view, adversarial examples span a contiguous region of 1-D subspace obtained from fast gradient method. The adversarial examples are not few in count and only in small pockets but instead abundant and in contiguous regions of our original data distribution. There are high chances that high dimensional adversarial subspaces from different models will overlap across models for the same task. Thus an adversarial example generated from a model 1 can still be misclassified by model 2.*

The authors conduct experiments where they generate adversarial examples from a deep maxout network and classify these on shallow softmax and RBF network. On examples where both these models made mistake, softmax model predicted the same as maxout 84.6% of times and RBF 54.3% of times. On examples where maxout made a mistake, Softmax predicted the same class as maxout 54.6% of times and RBF only 16.6%. Thus it shows that significant portion of adversarial examples are transferable to other models.

Question 5. *Why can't the approach of Goodfellow et al. be directly applied to generate a physical attack on a real Stop sign?*

Response:

Answer 5. *Goodfellow et al. approach is a digital perturbation of input images, while attacking real stop sign requires physical perturbations to physical input object. The following reasons suggest we cannot apply digital perturbation approaches:*

- *Physical environments have noise that can destroy the perturbations digitally created*
- *Digital perturbations are small in magnitude and real world perception of those could be hard to capture in cameras*
- *Current digital perturbations modify the background and in real world its not possible to alter the background*
- *Printing digital perturbations is also ineffective due to imperfections in our print quality.*

Question 6. *Explain the process in Eykholt et al. by which the physical attack on the Stop sign was generated. Pay attention to the entire pipeline, including any aspects of collecting data, training nets, computing the perturbation, and physical execution of the attack. Be clear about what portions involve a human and which tasks are performed automatically by computer.*

Response:

Answer 6. *The pipeline is as follows: They collect data to model physical conditions then they train a CNN model. The modelled data is then given to their RP2 algorithm which outputs perturbations as a poster and graffiti like stickers. Then humans take a print of the posters and stickers. The stickers are stuck to actual physical road signs. The trained CNN is then shown a) posters b) physical object with stickers and outputs a class for each input.*

They train a LISA-CNN and GTRSB-CNN models using LISA dataset. For GTRSB they replace each road stop sign with LISA stop signs. This process is all computer done.

Data Collection: To Capture effects of changing physical conditions, they take images of road signs under effects of lightning, distance and different angles Data is augmented with digital transformations such as rotation, crop, translation, change brightness. The data collection is done by humans and data augmentation is done by humans and computer(transformations). Thus they create a distribution of images that models noisy environmental conditions.

Computing Perturbations: Samples are drawn from the modelled distribution of images under noisy environment. Humans are involved to visualize the perturbations for better placement of masks.The RP2 algorithm computes pertubations and projects it only to the surface object(road sign) and not backgrounds using Masks. These perturbations are then manually printed as posters and the stickers are then stuck to physical road signs. Humans are involved in placing the printed posters for CNN and also sticking the graffiti like pertubations on road signs.

Attack: The road signs are stuck with stickers of perturbations to attack the CNN. To capture the environmental conditions such as distance, angle, lights, the humans drive around the road sign and feed the input to CNN. which then averages prediction of some number of frames. Posters are also stuck in labs and its images are captured and fed to CNN.

Question 7. *In Eykholt et al., Equation (1) provides a framework for computing adversarial examples. Explain this formulation. Be sure to specify what J could be. Is this formulation different that that in Goodfellow et al.?*

Response:

Answer 7. *The equation gives us the value for perturbations δ by minimizing the following:*

$$\arg\min_{\delta} \lambda \|\delta\|_p + J(f_\theta(x+\delta), y^*) \tag{2}$$

where λ is a regularization constant, $\|\delta\|_p$ is the distance function between $x+\delta$ and x.

J could be cross-entropy loss function that measures the difference between the prediction $f_\theta(x+\delta)$ where the input is the perturbed input and target y^* which is our target attack class label. Solving for this optimization would result in calculating gradient $\nabla_x J(f_\theta(x+\delta), y^*)$.

*The formulation is different from Goodfellow et al, whose perturbation is calculated as gradient of cost function $J(f_\theta(x), y)$ where x is the original input and y is the **true class**. Moreover, their method only uses the **sign** of the gradient of the cost function and not the magnitude*