Adversarial Attack on NN

Adversarial samples

Crafted by adding carefully selected **perturbations** δX to **legitimate inputs** X

- **Goals**:
  - `Confidence reduction`: reduce the output confidence classification
  - `Misclassification`: alter the output classification to any other class
  - `Targeted misclassification`: alter the output classification to a target class
  - `Source/Target misclassification`: force the output classification of a specific input to be a specific target class
- **Constraints**:
  - The perturbation δ_X must be small enough to pass human test.
  - E.g., the number of features perturbed is no larger than 14.29% for MNIST (about 112 pixels) [1].
- **Attacks at TEST time**: attack does not change the DNN model

Adversarial capabilities:

- **Network architecture**:
  - Layers, activation functions, weights, bias.
  - This gives attacker the ability to simulate the network.
- **Training data:**
  - The adversary is able to collect a *surrogate* dataset, sampled from the same distribution as the original training dataset.
  - This gives the attacker the ability to use the surrogate dataset to train a common DNN architecture to approximate the legitimate DNN model
- **Oracle**:
  - The adversary can obtain output classifications from supplied inputs.
  - This gives the attacker the ability to perform *differential attack* by observing the relationship between changes in inputs and outputs.
  - The adversary can be limited by the number of absolute or rate-limited input/output trails they may perform.

Adversarial sample crafting algorithm

Formal definition: Given a legitimate sample X, classified as $F(X) = Y$ by the network, the adversary wants to craft an *adversarial sample* X^* very similar to X, but misclassified as $F(X^*) = Y^* \neq Y$

$$argmin_{\delta_X} ||\delta_X|| s.t. F(X + \delta_X) = Y^*$$

Two-step process:

- **Direction Sensitivity Estimation**: evaluate the sensitivity of class change to each input feature
    - Fast sign gradient method[2]: compare the gradients of the cost function with respect to the inputs
    - Forward derivative method[1]
- **Perturbation Selection**: use the sensitivity information to select a perturbation δ_X among the input dimensions
    - Perturb all input dimensions by a small quantity [2]
    - Perturb a limited number of input dimensions by a large quantity [1]

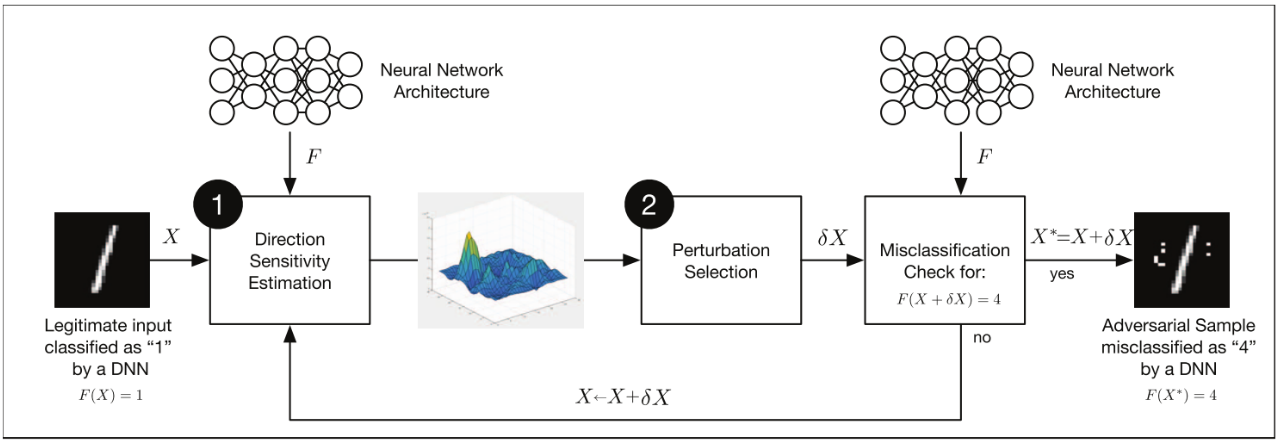

Fig. 3: **Adversarial crafting framework:** Existing algorithms for adversarial sample crafting [7], [9] are a succession of two steps: (1) *direction sensitivity estimation* and (2) *perturbation selection*. Step (1) evaluates the sensitivity of model F at the input point corresponding to sample X. Step (2) uses this knowledge to select a perturbation affecting sample X's classification. If the resulting sample $X + \delta X$ is misclassified by model F in the adversarial target class (here 4) instead of the original class (here 1), an adversarial sample X^* has been found. If not, the steps can be repeated on updated input $X \leftarrow X + \delta X$.

---

1. Papernot, Nicolas, et al. "The limitations of deep learning in adversarial settings." *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016↩↩↩

2. Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).↩↩