An Investigation of Bounded Misclassification for Operational Security of Networks

Sailik Sengupta, Andrew Dudley, Tathagata Chakraborti, Subbarao Kambhampati
Using classification in real-world scenarios

Input (Dog crossing a street)

Classification System
(Deep Neural Network)

Decision System
(Autonomous car)

Security of classification systems

Security of decision taking systems
Using classification in real-world scenarios

**Input** (Dog crossing a street)

**Classification System**
(Deep Neural Network)

**Decision System**
(Autonomous car)

---

**Security of classification systems**

**Security of decision taking systems**

“Operational Security”
Effects of Random and/or Adversarial Noise

Input (Dog crossing a street) +

Classification System (Deep Neural Network)

Decision System (Autonomous car)

BEER BOTTLE

GO!
Effects of Random and/or Adversarial Noise

Input (Dog crossing a street) + Classification System (Deep Neural Network) → DECISION SYSTEM (Autonomous car) → STOP!
Good Vs Bad Classification

DOG

NOT DOG
Good Vs Bad Classification

- DOG
- NOT DOG

- DOG
- CAT
- NOT DOG
- HUMAN
Good Vs Bad Classification
Good Vs Bad Classification

• Ideally we want to penalize misclassification to certain classes less than others.
  • [0\textsuperscript{th} order idea] Can borrow for works on cost-based misclassification.
  • We could not find any work that used cost-based or weighted loss functions for classifying noisy input examples for neural networks.

• Given a classification task at hand we have to have a notion of distance/similarity between a pair of class labels.
  • Penalize less when classifier misclassifies to a similar class since decision taken remains the same.
Class similarity values

**ImageNET**
- Subgraph of the WordNET, from which nouns were used as the class labels of ImageNET.
- Path similarity between WordNET words.

\[
S(GR, LR) = 0.333 \\
S(GR, Cat) = 0.111 \\
S(GR, Beer Bottle) = 0.00625
\]

**MNIST**
- For any classification problem, we need an underlying graph using which we can compute the similarity.
- For an ATM detecting digits on a hand written cheque, if a digit, say 2, is adversarially perturbed, it is better to classify it as 1 or 3 instead of 9.
Cost-Based loss function for DNNs

• Use **weighted loss functions** to penalize misclassification to dissimilar classes more.

• Define **class similarity matrix** (of size $C \times C$) given a classification task at hand.

\[
L(x) = - \sum_{j=1}^{m} Y_j \log o_j
\]

\[
L(x) = - \sum_{j=1}^{m} \Pi^\delta(s(Y_k, Y_j)) \log o_j
\]

\[
\Pi^\delta(a) = \begin{cases} 
1 & \text{if } a \geq \delta \\
0 & \text{otherwise}
\end{cases}
\]

\[
s(Y_i, Y_j) = \begin{cases} 
x & \text{if } i \neq j \\
1 & \text{otherwise}
\end{cases}
\]
Vanilla cross entropy network
• Network misclassifications reflect similarity in structures of numbers. Eg. 7 misclassified as a 2.

Weighted cross entropy networks with various bounds
• As we increase the bound, more images are misclassified near the correct class instead of any arbitrary class.
• Accuracy takes a hit as the loss function says that misclassification to closer classes in not a very bad thing to do.
• We have come up with scaled similarity metrics to address the later for now.

Batch Training
• Initial epochs- correct label is n-hot vector. All n classes that are within similarity bound is given label 1.
• Then training with one-hot labels.

Figure 4. Distribution of mislabeled classes in MNIST in the three training conditions C1, C2 and C3. As expected, in C2 and C3, instances of misclassification huddle around the diagonal (prediction = target) while the classification accuracy takes a hit.
White Noise $\mathcal{N}(0,1)$

Adversarial Noise (FGSM $\varepsilon = 0.03$)

White + Adv. Noise $\mathcal{N}(0,1) + \varepsilon (= 0.03)$

- Vanilla Cross-entropy
  - 40.77 (gain = 3321)
  - 1.79 (gain = 5071)
  - 4.21 (gain = 5357)

- Weighted Cross-entropy
  - 17.50 (gain = 5771)
  - 13.85 (gain = 5684)
  - 12.21 (gain = 5659)
Discussion

• We investigate the use of cost-based/weighted loss functions for Deep Neural Networks with a goal to improve accuracy of decision making based on classification systems.

• Can we use similar techniques for designing an open-world classifier?
  • Say picture of a kangaroo, which the DNN has never seen before, is given as input (Si Liu’s talk in the morning).
  • Based on features it can detect in the squirrel, it classifies it as (say) a cat and not any random class, like a leaf or bear bottle.

Read out paper at: https://goo.gl/jFdTsy