1 Motivation and summary

- Deep neural nets are accurate black-box models.
- Our goal is to understand what internal features computed by the neural net are responsible for a particular class. We achieve this by mimicking the classifier part of the net with a decision tree having sparse weight vectors at the nodes. We can learn accurate enough sparse oblique trees with the tree alternating optimization (TAO) algorithm.

- We found that out of thousands of neurons (in the last layer of feature-extraction part of the net), there is only a small subset of neurons associated with a given class. We explore this by introducing a new feature-level adversarial attack via masking a specific set of neurons. We show that we can easily manipulate the neural net features in order to make the net predict, or not predict, a given class.

- For VGG16 trained on a subset of ImageNet (16 classes), only 1 366 out of 8 192 (only 17%) neurons are needed to achieve the same performance as the original net.

2 Masking of deep net features

- Consider a trained deep net classifier: \( y = f(x) \).
- We can write \( f \) as: \( f(x) = g(F(x)) \), where
  - \( F \) represents the features-extraction part \( (z = F(x) \in \mathbb{R}^f) \).
  - \( g \) represents the classifier part \( (y = g(z)) \).
- Train a sparse oblique tree \( y = T(z) \) on the training set \( \{(F(x_n), y_n)\}_{n=1}^N \subset \mathbb{R}^f \times \{1, \ldots, K\} \). Choose the sparsity hyperparameter \( \lambda \in [0, \infty) \) such that, \( T \) mimicks \( g \) very good and is as sparse as possible. Next, inspect the weights of the decision nodes to create masks.

- Our masking operation is as follows:
  - Original net: \( y = f(x) = g(F(x)) \).
  - Original features: \( z = F(x) \).
  - Masked net: \( \hat{y} = \hat{f}(x) = g(\hat{F}(x)) \)
  - Masked features: \( \hat{z} = \hat{F}(x) = \mu(z) \).
  - \( \mu = \{\mu^+, \mu^+\} \), where, \( \mu^+ \in \{0, 1\}^F \) is the multiplicative mask and \( \mu^+ \geq 0 \) is the additive mask.

- We show three masks:
  - ALL TO CLASS \( k \): Let \( k \in \{1, \ldots, K\} \). Classify all instances \( x \) as class \( k \).
  - ALL CLASS \( k_1 \) TO CLASS \( k_2 \): Let \( k_1 \neq k_2 \in \{1, \ldots, K\} \). For any instance originally classified as \( k_1 \), classify it as \( k_2 \). For any other instance, do not alter its classification.
  - NONE TO CLASS \( k \): Let \( k \in \{1, \ldots, K\} \). For any instance originally classified as \( k \), classify it as any other class. For any other instance, do not alter its classification.

Illustration of masks for a particular image in VGG16. Column 1 shows the image masks (when available). Column 2 shows the histogram of corresponding softmax values. Row 3 shows a mask manually cropped in the image, whose features resemble those of row 2. Row 4 shows a mask in feature space obtained by finding the top-3 superpixels whose features most resemble those of the masked features of row 2.