Understanding and Manipulating Neural Net features Using Sparse Oblique Classification Trees

Suryabhan Singh Hada
Miguel Á. Carreira-Perpiñán  Arman Zharmagambetov
{shada,mcarreira-perpinan, azharmagambetov}@ucmerced.edu

Electrical Engineering and Computer Science,
University of California, Merced
http://eecs.ucmerced.edu

July 20, 2021
Deep neural nets have become the preferred model in a number of practical problems, such as computer vision, language processing, games, self-driving cars, and other engineering applications.

The way neural nets are defined and optimized, and the sheer size and complexity of state-of-the-art deep nets, makes them very hard to understand in explanatory terms.

Much work has focused on understanding what part of the input pattern (an image, say) is responsible for a particular class being predicted, and how the input may be manipulated to predict a different class.

We focus instead on understanding what internal features computed by the neural net are responsible for a particular class.
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 (\( y = g(z) \)).

The last layer of \( F \) is interesting, as it is associated with the features extracted by \( F \) that goes into \( g \).

We want to understand the relationship between neurons in the last layer of \( F \) and the classes.
Out of thousands of neurons, there is a small subset of neurons associated with a given class.

We explore this by introducing a new feature level adversarial attack via masking specific set of neurons.

These attacks include to make net to predict or not predict a given class.
Our approach

- We study the relationship between neurons at the last layer of \( F \) and the classes using sparse oblique trees.

**Overall approach:**

- 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, \( T \) mimics \( g \) very good and is as sparse as possible.
- Inspect the tree \( T \) to create masks.
Objective is to control the behavior of network prediction by manipulating deep net features \((z = F(x) \in \mathbb{R}^F)\), without modifying the \(F\) and \(g\).

Original net: \(y = f(x) = g(F(x))\).

Original features: \(z = F(x)\).

Masked net: \(\overline{y} = \overline{f}(x) = g(\mu(F(x)))\)

Masked features: \(\overline{z} = \mu(F(x)) = \mu(z)\).

\[\overline{z} = \mu(z) = \mu^\times \odot z + \mu^+.\]

\[\mu = \{\mu^\times, \mu^+\}\]

where, \(\mu^\times \in \{0, 1\}^F\) is the \textit{multiplicative mask}.

\(\mu^+ \geq 0\) is the \textit{additive mask}.
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.
We use VGG16 network, trained over a subset of 16 classes from ImageNet.

- Training error: 0.2%
- Test error: 6.79%
- $\mathbf{z} \in \mathbb{R}^{8192}$

We use the tree with $\lambda = 1$.

- Training error: 0%
- Test error: 7.9%
- # nodes: 39
- features used: 1366 out of 8192 (only 17%)
$T$ with $\lambda = 1$

83% neurons masked
Mask on a single image

Original

![Image of a person and a dog with a car in the background]

![Graph showing softmax values for different objects: Siberian husky, School bus, Sports car]
Mask on a single image

Original

Mask in feature space

ALL TO CLASS "SIBERIAN HUSKY" mask is applied
Mask on a single image

Original

Mask in feature space

Manual mask in image space

All to class "Siberian husky" mask is applied
Mask on a single image

Original

Mask in feature space

Manual mask in image space

Mask in image space obtained by features

All to class "Siberian husky" mask is applied
Mask on a single image

**Original**

**Mask in feature space**

---

**Manual mask in image space**

---

**Mask in image space obtained by features**

---

**Mask in feature space**
Mask results on the test set

.................... ALL TO CLASS $k$ ....................

$k = 0$  $k = 1$  $k = 2$  $k = 3$  $k = 4$  $k = 5$  $k = 6$  $k = 7$

$k = 8$  $k = 9$  $k = A$  $k = B$  $k = C$  $k = D$  $k = E$  $k = F$
Mask results on the test set

\[ \text{\ldots ALL CLASS } k_1 \text{ TO CLASS } k_2 \text{ \ldots} \]

- \( 8 \to E \)
- \( E \to 8 \)
- \( A \to B \)
- \( B \to A \)
- \( 9 \to C \)
- \( C \to 9 \)

\[ \text{\ldots NONE TO CLASS } k \text{ \ldots} \]

- \( k = 0 \)
- \( k = 1 \)
- \( k = 2 \)
- \( k = 3 \)
- \( k = 4 \)
- \( k = 5 \)
- \( k = 6 \)
- \( k = 7 \)

- \( k = 8 \)
- \( k = 9 \)
- \( k = A \)
- \( k = B \)
- \( k = C \)
- \( k = D \)
- \( k = E \)
- \( k = F \)
Training error: 1.79%
Test error: 9.56%

# nodes: 31
features used: 408 out of 8192 (only 5%)
Thank You!