Chapter 1

Problem Statement and Previous Work

1.0.1 Adversarial Examples

Adversarial examples were originally defined in the domain of image classification in the form of a constrained optimization problem. The following definition is similar to [1].

Definition. An adversarial derivation, $x + r \in \mathbb{R}^m$, of a an image, $x \in \mathbb{R}^m$ for a classifier, f, is the solution to the following optimization problem:
minimize $||r||_2$ subject to:

1. $f(x + r) \neq f(x)$
2. $x + r \in [0, 1]^m$

We may also denote the adversarial derivation as $x^* = x + r$

In words, an adversarial derivation is a sample with the minimum distance to a true sample such that it is classified as something different. We also require that every element stays in the interval $x_n + r_n \in [0, 1]$ because actual pixels must remain in some constant bounded interval.

Clearly, this definition does not translate well to natural language which is not even numeric. We give a more general constrained optimization defintion below, which we will adapt to the problem of creating adversarial derivations for text.

Definition. An adversarial derivation (AD) of x with difference ϵ, domain metric d_D, and codomain metric d_C is given by

$$\begin{aligned} \underset{x^* \in D(f)}{\text{minimize}} \quad & d_D(x, x^*) \\ \text{subject to} \quad & d_C(f(x), f(x^*)) > \epsilon \end{aligned}$$

Where d_D and d_C are distance metrics defined on the domain of f and codomain of f, respectively, and $\epsilon > 0$ Note that image classifier would have the domain $[0, 1]^m$ and codomain $\{0, 1, ...K\}$ where K is the number of classes. So the definition of an adversarial derivation for an image classifier is the same as the general definition if we let $\epsilon = 1$, $d_D(x, x^*) = ||x - x^*||^2$, $d_C(y, y^*) = |y - y^*|$.

As long as the model, f, has a non-singleton codomain, a solution to this optimization problem exists for small enough ϵ. This is true from the fact that for any distance metric, $d(y, y^*) > 0$ for $y \neq y^*$. That being said, the distance between the sample and the derived sample may be so large that they are easily recognized to be different samples. We therefore define a similar problem with very different properties.

Definition. An absolutely adversarial derivation (AAD) of x with similarity δ, difference ϵ, domain metric d_D, and codomain metric d_C is given by any solution to the following two constraints.

$$d_D(x, x^*) < \delta$$

$$d_C(f(x), f(x^*)) > \epsilon$$

In this less relaxed version of the problem, a solution may not exist. In fact, for

a given continuous model f, and ϵ it is guaranteed that $\exists \delta > 0$ s.t. $d_C(f(x), f^*(x)) < \epsilon$ meaning no solution exists. However, it appeals to a more intuitive concept and allows for the possibility of a model immune to adversarial attacks. It says that the sample and it's absolute adversarial derivation must be sufficiently close and the distance between the model outputs must be sufficiently far.

It is easy to see that if an AAD exists for a given sample, then any AD is also an AAD. This means we may solve the more relaxed problem to obtain a valid solution, and so we will work with the more practicable objective of creating adversarial derivations.

1.0.2 Text Classification

Consider the common scenario of a text classifier which maps plain text files to one of several classes. It is common for the plain text to first be processed into a sequence of tokens, which are then each assigned an integer resulting in a sequence of integers.

Definition. Let s be a sequence of characters. Let $a_n \in \{0, 1, \ldots, V\} \forall n \in \{0, 1, \ldots, N\}$ and $E : s \to \{a_n\}_{n=1}^{N_s}$ then we call E an encoder, V the encoder vocabulary size, and N_s the sample length with respect to E.

In plain words, an encoder maps a string to a sequence of bounded integers. The sequence is some length which depends on both the encoder and the string. We assume a fixed encoder, and therefore vocabulary size, V. Since, after encoding, the distance between one word and another is arbitrary, we further translate into a one-hot encoded vector. That is, the integer n is mapped to a vector where the n^{th} element is 1 and all others are 0. This ensures that all vectors representing words are unit norm and the distance between any two different words is the same. We denote the set of one-hot encoded vectors of size V as 1_V

This simple method of representing words as vectors results in a very high di-

mension representation of all words in the vocabulary, and thus even a very simple linear model would be very large and be difficult train. Using the word2vec model [2], the dimension of this representation can be significantly reduced, while also encoding information about statistical semantic similarity about each word.

Definition. Let $f : 1_V \to \mathbb{R}^D$. We call f a word embedding and we call D the size of f, or embedding size.

Let $W \in M_{D\times V}(\mathbb{R})$. Then clearly any word embedding, f, of size D may be represented as the matrix multiplication $Wv\ \forall v \in 1_V$. The matrix W is called the embedding matrix.

1.0.3 Adversarial Text Derivation

Now that we have a clear idea of both the domain and numerical representation of words, we may define an adversarial derivation of textual data in the context of a classification model, f. As per the definition of an adversarial derivation, we need only to define the model difference, ϵ, as well as the domain metric, d_D and codomain metric, d_C. We will consider primarily two definitions.

The discrete metric is given by

$$\rho(v, v^*) = \begin{cases} 0 & \text{if } v = v^* \\ 1 & \text{if } v \neq v^* \end{cases}$$

Definition. Let $\{v_i\}_{i=1}^N$ be the sequence of vectors obtained from a given word embedding and text sample, then a discrete adversrial derivation is defined has having domain metric, $d_D(v, v^*) = \sum_i^N \rho(v_i, v_i^*)$, codomain metric $d_C(f(v), f(v^*)) = \rho(f(v), f(v^*))$, and difference $\epsilon = 1$

That is, a discrete adversarial derivation, $\{v_i^*\}$, of sample $\{v_i\}$ is the sample

which changes the least amount of words possible, while changing the classification. This definition is simple, though it may not yield very good results if solved. For example, a positive movie review, "This movie was good" could easily be changed to a negative review by changing just one word resulting in "This movie was bad". These two samples would obviously have different sentiments if read by a human.

Clearly the codomain metric, d_C and difference, ϵ make sense for any defintion in this context, but the domain metric has room for improvement. One possibility is to instead use Euclidean or Manhattan distance between word vectors, that is, $d_D(v_i, v_i^*) = ||v_i - v_i^*||_2$ or $d_D(v_i, v_i^*) = ||v_i - v_i^*||_1$. If we minimize this objective, then we would tend to use semantically similar words in substitution. However, this does not necessarily solve the problem of actual sentiment inversion. For instance, in our embedding the semantically closest (measured with the l^2 norm) word to "bad" is "good." This makes sense since they are semantically very similar and would be used in the same contexts, but may result in obvious semantic flips. We therefore look to other metrics in an attempt to find better results. We found that for our sentiment classification model, the Jacobian was related to the sentiment of the word, which motivates the following definitions.

Definition. Let the gradient of a model output, f_i, with respect to an input vector, v be denoted by $g = \nabla_v(f)$. The total gradient is given by

$$g_t = \sum_{i=1}^{D} \nabla_v(f)_i$$

The gradient norm is given by

$$g_n = \sqrt{\sum_{i=1}^{D} |\nabla_v(f)_i|^2} = ||\nabla_v(f)||_2$$

Both of these measures, along with the gradient itself, are shown for a small excerpt for one of the text samples in figure 1.1. We see that the three words

with the largest total gradients are "loved", "good", and "bad" which are all sentimental. We provide motivation for these choices of measure with a very simplified probabilitic analysis.

Suppose that each of the embedding dimensions is distributed independently and identically across words, with some mean, μ and variance, σ^2. Let $g = \nabla_u(f)$ for a given word, u in the sample, then

$$\mathrm{E}\left[g^T v\right] = \mathrm{E}\left[\sum g_i v_i\right] = \sum g_i \,\mathrm{E}\left[v_i\right] = \mu \sum g_i = \mu g_t$$

If we are interested in purposefully altering classification, however, we might be more interested in the expected maximum value of $g^T v$, that is,

$$Z(g) = \mathrm{E}\left[\max_{0 \leq n \leq V} g^T v_n\right]$$

Unfortunately, there is no simple expression which captures this value. However if we assume that $v_{n,i}$ is distributed normally, we have

$$\mathrm{E}\left[\max_{0 \leq n \leq V} g^T v_n\right] = \mathrm{E}\left[\max_{0 \leq n \leq V} \sum_{i=1}^{D} g_i v_{n,i}\right] = \mathrm{E}\left[\max_{0 \leq n \leq V} p_n\right]$$

Where $p_n \sim \mathcal{N}(\mu g_t, \sigma^2 g_n^2)$ There is still no closed form expression for this value, but there is a known upper bound:

$$Z(g) \leq \mu g_t + \sigma g_n \sqrt{2 \log V}$$

This inequality is intuitively satisfying. It says that the maximum obtainable perturbation grows with both the vocabulary size and the norm of the gradient. The total gradient also plays a role here, increasing or decreasing the expected maximum depending on the sign.

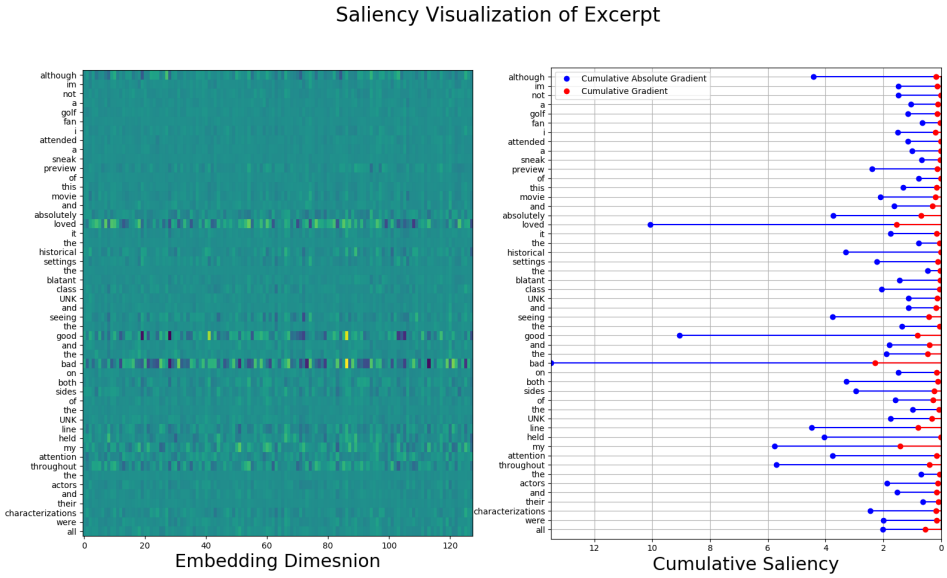

Figure 1.1: Different measures of word sentiment/importance

Bibliography

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *ArXiv e-prints*, Dec. 2013.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *ArXiv e-prints*, Oct. 2013.