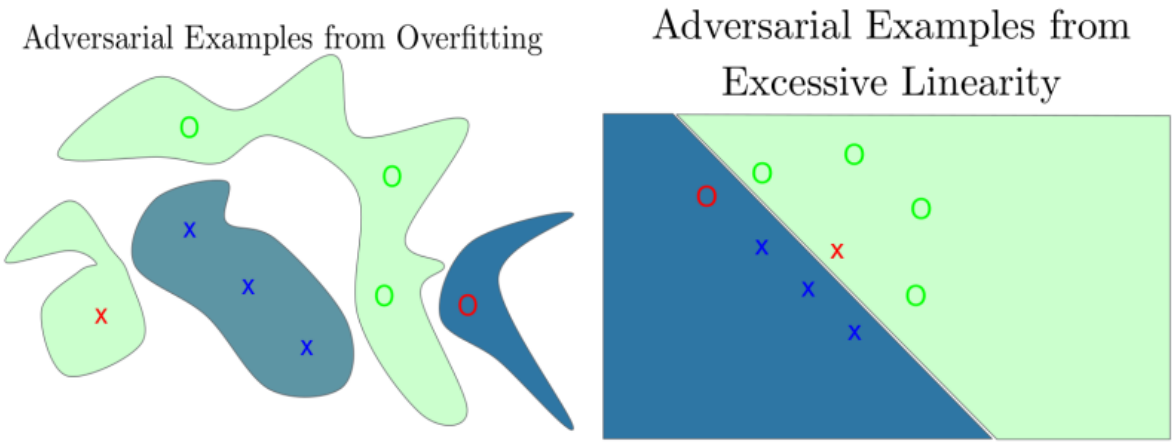
How exactly do Adversarial examples arise?

1. From Goodfellow's lecture: It's because of linearity from the input space to the output. It is actually **piecewise linearity**. This also explains the phenomenon where an adversarial direction from an image will also affect some other image and also the transferability phenomenon since the decision boundaries learnt by various models are same.
2. From madry's Not bugs but features and another paper: The model learns "non-robust features" that are not meaningful to the humans. This has also been proved as well by constructing a dataset. But how does this explain the transferability conditions? Also no assumptions on linearity as well.
3. Zico's Randomized Smoothing and others: Many papers like TRADES and Randomized smoothing talk about making the decision boundary more smooth. And they achieve good results as well. Even Goodfellow did talk about some smoothing, I need to look more into this.

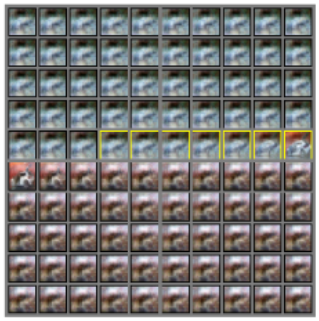
GoodFellow's work

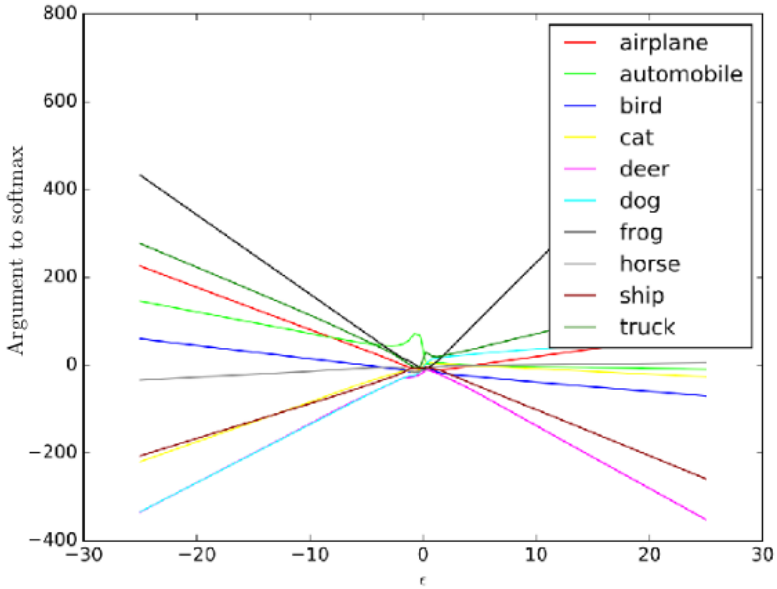


Adversarial images coming from excess of overfitting (like shown above) is not reasonable because if we fit the model again or slightly different model, then the adversarial examples would differ but this is not the case as transferability is seen. There is a systematic effect and not a random effect. So overfitting is not an option.

This leads to the idea that these Ad examples come from underfitting, becuase of the linearity of model. The deep nets are very piecewise linear. The mapping from the input of model to output of model is linear or piecewise linear with few pieces. The mapping from the parameters of network to output of network is non linear, because the weight matrices are multiplied together. So we get extremely non linear relations between parameters and the output, that's what makes the training of deep nets so difficult.

Nearly Linear Responses in Practice

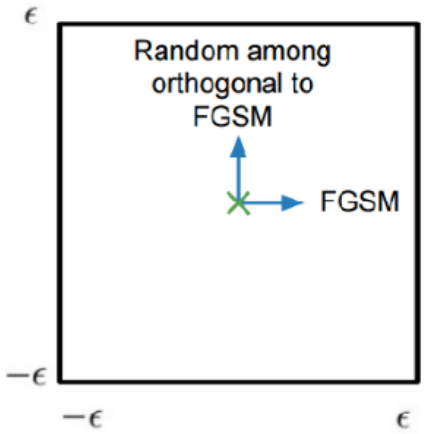


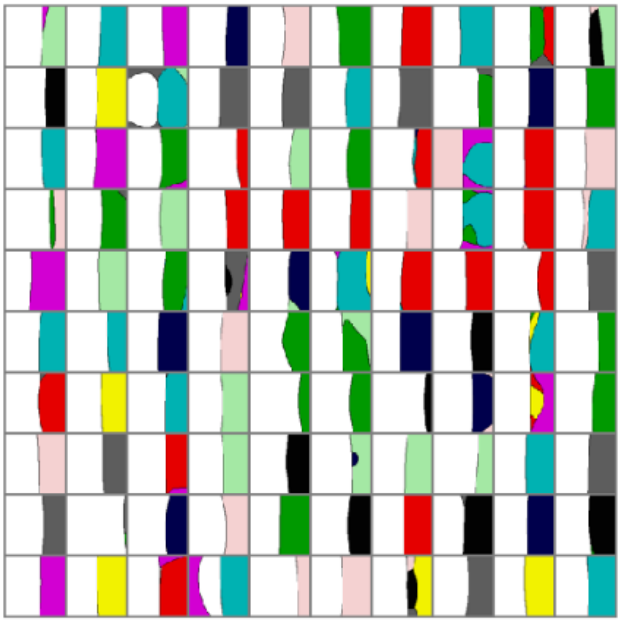


The linearity can be seen by the above example: A clean example is choosen and a direction is present which is multiplied by epsilon. We can see clear piecewise linear relations (mainly due to ReLU). The yellow boxes are correctly predicted as car. For high magnitude of epsilons, we see frog is highly predicted. The automobile class is high at the center.

FGSM attack has over 99% error on a normally trained model. FGSM is nothing but moving in the direction of max loss(i.e in the direction of decision boundary), it's like a linear attack (i.e there is a linear perturbation) and actually supports the hypothesis that because of deep nets linearity, we have AE.

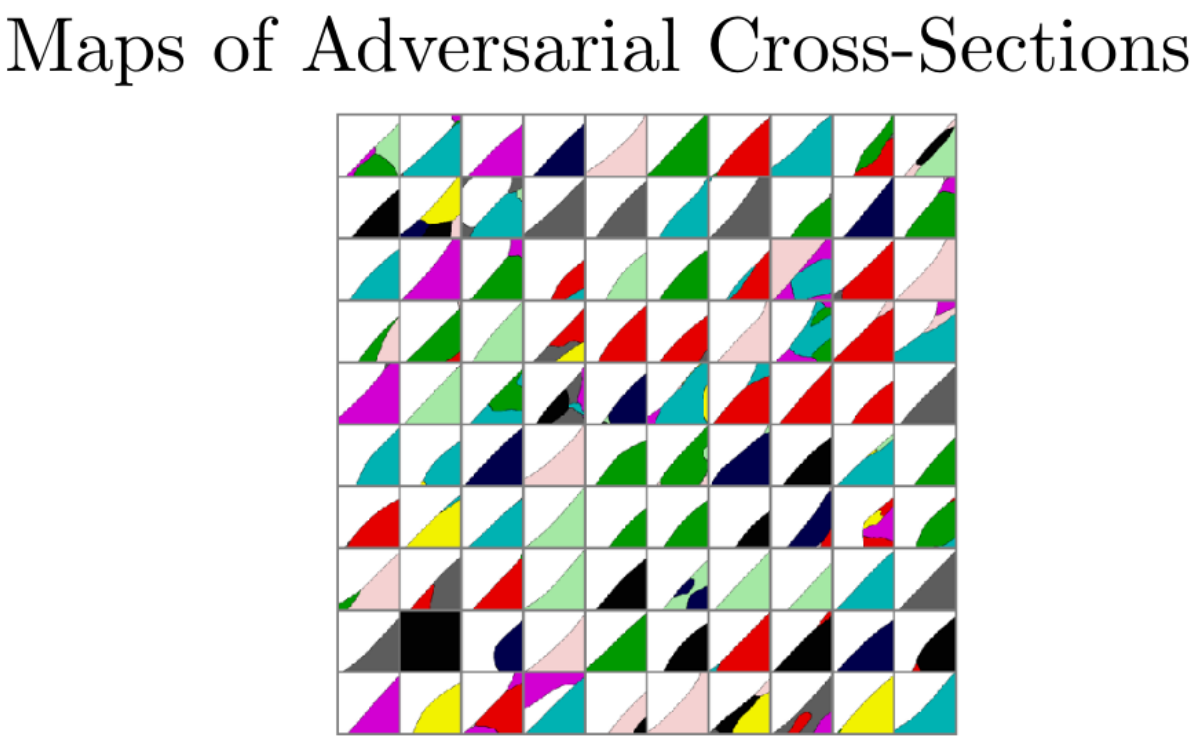
Maps of Adversarial and Random Cross-Sections





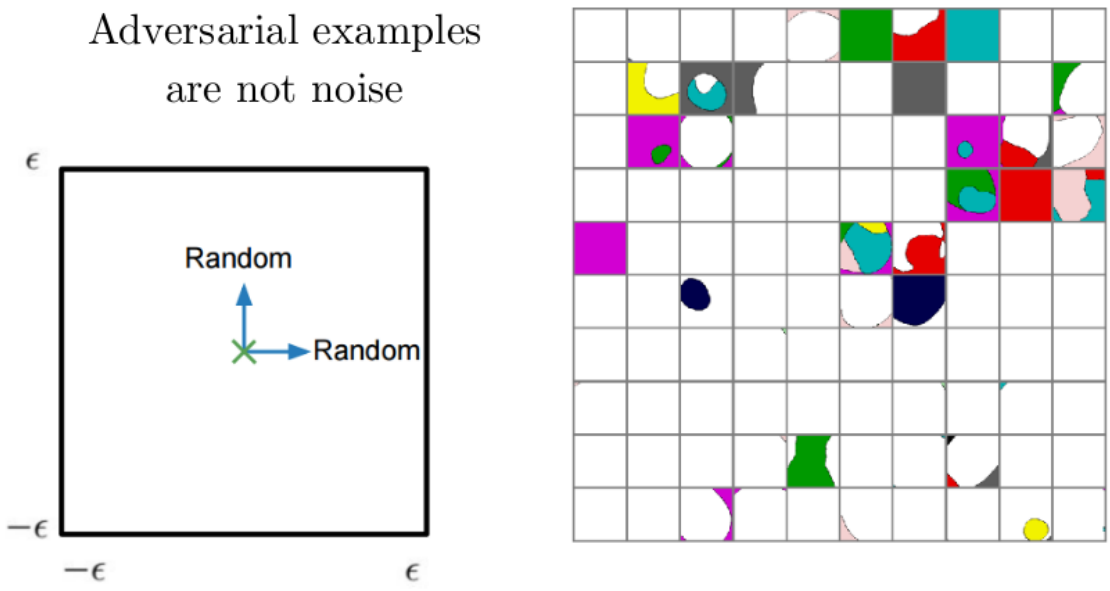
Looking for more evidence, the above experiment was performed. Each cell is a map of CIFAR10 classifier's decision boundary with each cell corresponding to a test datapoint. The center of each cell corresponds to the original image. As we move left to right in each cell, we move along the direction of FGSM and from top to bottom in a random orthogonal direction to FGSM direction. This is a 2d cross section of the classifier. White pixel indicates the correct choosen class. In all the cells, the left half is correctly classified. But along the right we can see a change in label predicted. FGSM has detected a direction where if we get a large dot product, we get an adversarial example. "Once your in that adversarial subspace all the other points nearby are adversarial examples that will be misclassified." So you only need to find the direction to fool the model.

Choosing a second direction that has high dot product with the gradient, we could make both the axes adversarial (my understanding is instead of FGSM, they choose a direction that gave the next best results, and this becomes x-axis and y-axis is orthogonal to this):



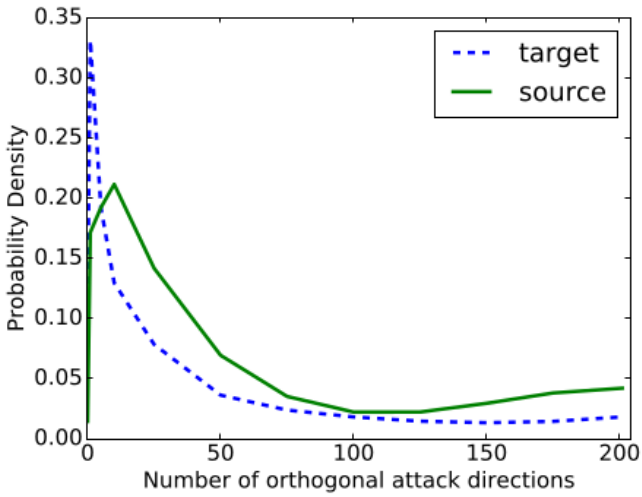
Here we see that linear decision boundaries are obtained oriented diagonally. We can see that there's acutally a 2d subspace of adversarial subspace of AE that we can cross it. This gives the following result: **adversarial examples form a dense space that is (1) at least two-dimensional**

Maps of Random Cross-Sections



Random directions don't affect much. Infact most of the colours here are because of the misclassified class(center is not a white pixel)

Estimating the Subspace Dimensionality



(Tramèr et al, 2017)

In previous diagrams we looked in two dimensions. In Tramer et al, it was attempted to know just how many dimensions there are to these subspaces where the adversarial examples lie in a thick contiguous region. And they came up with an algorithm together where you actually look for several different orthogonal vectors that all have a large dot product with the gradient. It was found that for MNIST, on average it was found that adversarial region has on average 25 dimensions. So what's interesting here is **the dimensionality actually tells you something about how likely you are to find an adversarial example** by generating random noise. If every direction were adversarial, then any change would cause a misclassification. If most of the directions were adversarial, then random directions would end up being adversarial just by accident most of the time. And then if there was only one adversarial direction, you'd almost never find that direction just by adding random noise. When there's 25 you have a chance of

doing it sometimes. Another interesting thing is that different models will often misclassify the same adversarial examples. The subspace dimensionality of the adversarial subspace relates to that transfer property. The larger the dimensionality of the subspace, the more likely it is that the subspaces for two models will intersect. So if you have two different models that have a very large adversarial subspace, you know that you can probably transfer adversarial examples from one to the other. But if the adversarial subspace is very small, then unless there's some kind of really systematic effect forcing them to share exactly the same subspace, it seems less likely that you'll be able to transfer examples just due to the subspaces randomly aligning.

Informally, the number of orthogonal adversarial directions is proportional to the increase in loss γ (a proxy for the distance from x to the decision boundary) and inversely proportional to the smoothness of the loss function and the perturbation magnitude.