**Regularization for Sparsity**

## Let's Go Back to Feature Crosses

* **Caveat:** Sparse feature crosses may significantly increase feature space
* Possible issues:
  + Model size (RAM) may become huge
  + "Noise" coefficients (causes overfitting)

## L1 Regularization

* Relax to L1 regularization:
  + Penalize sum of abs(weights)
  + Convex problem
  + Encourage sparsity unlike L2

## L1 vs. L2 regularization.

L2 and L1 penalize weights differently:

* L2 penalizes weight2.
* L1 penalizes |weight|.

Consequently, L2 and L1 have different derivatives:

* The derivative of L2 is 2 \* weight.
* The derivative of L1 is k (a constant, whose value is independent of weight).

You can think of the derivative of L2 as a force that removes x% of the weight every time. As [Zeno](https://wikipedia.org/wiki/Zeno%27s_paradoxes#Dichotomy_paradox) knew, even if you remove x percent of a number *billions of times*, the diminished number will still never quite reach zero. (Zeno was less familiar with floating-point precision limitations, which could possibly produce exactly zero.) At any rate, L2 does not normally drive weights to zero.

You can think of the derivative of L1 as a force that subtracts some constant from the weight every time. However, thanks to absolute values, L1 has a discontinuity at 0, which causes subtraction results that cross 0 to become zeroed out. For example, if subtraction would have forced a weight from +0.1 to -0.2, L1 will set the weight to exactly 0. Eureka, L1 zeroed out the weight.

L1 regularization—penalizing the absolute value of all the weights—turns out to be quite efficient for wide models.

Note that this description is true for a one-dimensional model.