Lecture 12: Regularization

COMP 411, Fall 2021

Victoria Manfredi

WESLEYAN
UNIVERSITY

Acknowledgements: These slides are based primarily on those created by Michael Paul (U of Colorado), Vivek Srikumar (U of Utah), and Russ Grenier (U of Alberta), and material from the book Machine Learning: An Applied Introduction by Paul Wilmott

Today's Topics

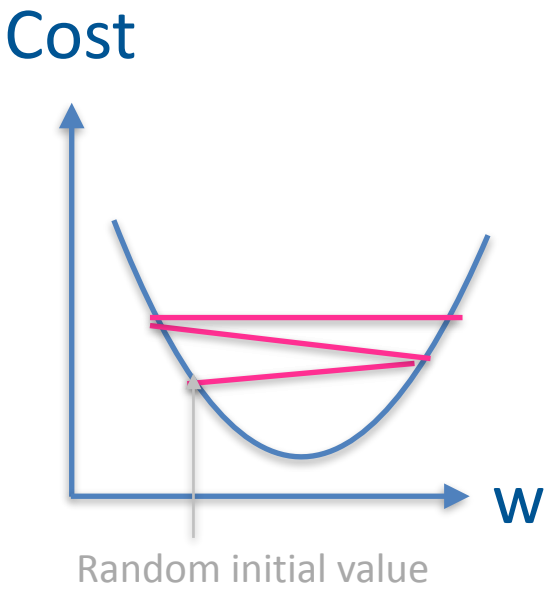
- Convergence of gradient descent
- Regularization

vumanfredi@wesleyan.edu

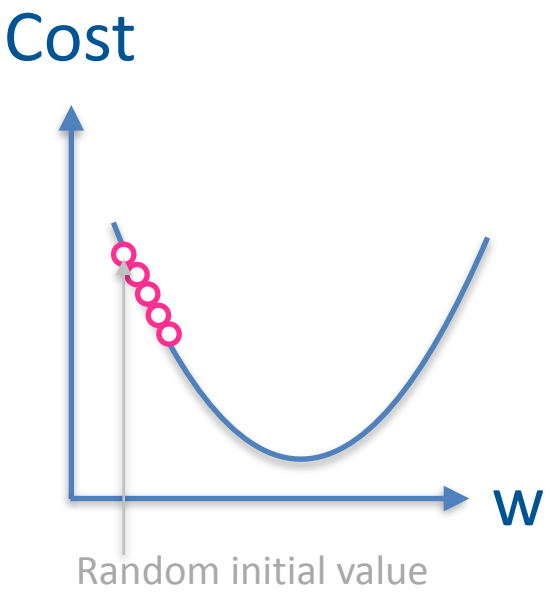
Gradient Descent

CONVERGENCE

Impact of learning rate



Learning rate too large



Learning rate too small

Impact of learning rate

- Stop after fixed number of iterations
- Stop once prediction error is less than threshold
- Stop when validation loss stops changing

Feature normalization

- Important to normalize features when using gradient descent (otherwise takes longer to converge)

- Normalization: all features should have a similar scale

LMS Regression

REGULARIZATION

Generalization

- Prediction functions that work on the training data might not work on other data

- Minimizing the training error is a reasonable thing to do, but it's possible to minimize it "too well"

- **Overfitting:** your function matches the training data well but is not learning general rules that will work for new data

Regularization

- Modify learning algorithm to favor “simpler” prediction rules to avoid overfitting

- Most commonly, regularization refers to modifying the loss function to **penalize** certain values of the weights you are learning. Specifically, penalize weights that are large.

Regularization

- How do we define whether weights are large?

$$d(\mathbf{w}, \mathbf{0}) = \sqrt{\sum_{i=1}^{k} (w_i)^2} = ||\mathbf{w}||$$

Note that bias term w_0 is not regularized

- This is called the L2 norm of $\mathbf{w}$
  - A norm is a measure of a vector's length
  - Also called the Euclidean norm

Regularization

- New goal for minimization

$$L(\mathbf{w}) + \lambda ||\mathbf{w}||^2$$

Square to eliminate square root: easier to work with mathematically

This is whatever loss function we are using

By minimizing this we prefer solutions where $\mathbf{w}$ is closer to $\mathbf{0}$

λ is a hyperparameter that adjusts trade-off between low training loss and having low weights

Regularization

- Regularization helps the computational problem because gradient descent won't try to make some feature weights grow larger and larger

- At some point, the penalty of having too large $||w||^2$ will outweigh whatever gain you would make in your loss function.

Regularization

- This also helps with generalization because it won't give large weight to features unless there is sufficient evidence that they are useful

- The usefulness of a feature toward improving the loss has to outweigh the cost of having large feature weights

Regularization

- More generally

$$L(\mathbf{w}) + \lambda R(\mathbf{w})$$

This is called the regularization term or regularizer or penalty. The squared L2 norm is one kind of penalty, but there are others

λ is called the regularization strength.

Other common names for λ: *alpha* in sklearn, *C* in many algorithms. Usually C actually refers to the inverse regularization strength, 1/λ. Figure out which one your implementation is using (whether this will increase or decrease regularization)

L2 Regularization

- When the regularizer is the squared L2 norm $||\mathbf{w}||^2$, this is called L2 regularization.
- This is the most common type of regularization
- When used with linear regression, this is called Ridge regression
- Logistic regression implementations usually use L2 regularization by default
- L2 regularization can be added to other algorithms like perceptron (or any gradient descent algorithm)

L2 Regularization

- The function $R(\mathbf{w}) = ||\mathbf{w}||^2$ is convex, so if it is added to a convex loss function, the combined function will still be convex.

L1 Regularization

- Another common regularizer is the L1 norm:

$$||\mathbf{w}||_1 = \sum_{j=1}^{k} |w_j|$$

- When used with linear regression, this is called Lasso

- Often results in many weights being exactly 0 (while L2 just makes them small but nonzero)

L1+L1 Regularization

- L2 and L1 regularization can be combined

$$R(\mathbf{w}) = \lambda_2 ||\mathbf{w}||^2 + \lambda_1 ||\mathbf{w}||_1$$

- Also called ElasticNet
- Can work better than either type alone
- Can adjust hyperparameters to control which of the two penalties is more important
- Once training is done, remove regularization term to measure model performance

Feature normalization

- The scale of the feature values matters when using regularization

- If one feature has values between [0, 1] and another between [0, 10000], the learned weights might be on very different scales – but whatever weights are "naturally" larger are going to get penalized more by the regularizer.

- Feature normalization or standardization refers to converting the values to a standard range.

Bias vs. variance

- Remember: the goal of machine learning is to learn a function that can correctly predict all data it might hypothetically encounter in the world

- We don't have access to all possible data, so we approximate this by doing well on the training data

- The training data is a sample of true data

Bias vs. variance

- When you estimate a parameter from a sample, the estimate is biased if the expected value of the parameter is different from the true value.

- The expected value of the parameter is the theoretical average value of all the different parameters you would get from different samples.

- Example: random sampling (e.g. in a poll) is unbiased because if you repeated the sampling over and over, on average your answer would be correct (even though each individual sample might give a wrong answer).

Bias vs. variance

- Regularization adds a bias because it systematically pushes your estimates in a certain direction (weights close to 0)

- If the true weight for a feature should actually be large, you will consistently make a mistake by underestimating it, so on average your estimate will be wrong (therefore biased).

Bias vs. variance

- The variance of an estimate refers to how much the estimate will vary from sample to sample.

- If you consistently get the same parameter estimate regardless of what training sample you use, this parameter has low variance.

Bias vs. variance

- Bias and variance both contribute to the error of your classifier.

- Variance is error due to randomness in how your training data was selected.

- Bias is error due to something systematic, not random.

Bias vs. variance

High bias

- Will learn similar functions even if given different training examples
- Prone to underfitting

High variance

- The learned function depends a lot on the specific data used to train
- Prone to overfitting

Some amount of bias is needed to avoid overfitting. Too much bias is bad, but too much variance is usually worse.