**Elastic Net Regression**

Linear regression refers to a model that assumes a linear relationship between input variables and the target variable.

With a single input variable, this relationship is a line, and with higher dimensions, this relationship can be thought of as a hyperplane that connects the input variables to the target variable. The coefficients of the model are found via an optimization process that seeks to minimize the sum squared error between the predictions (*yhat*) and the expected target values (*y*).

* loss = sum i=0 to n (y\_i – yhat\_i)^2

A problem with linear regression is that estimated coefficients of the model can become large, making the model sensitive to inputs and possibly unstable. This is particularly true for problems with few observations (*samples*) or more samples (*n*) than input predictors (*p*) or variables (so-called *p >> n problems*).

One approach to addressing the stability of regression models is to change the loss function to include additional costs for a model that has large coefficients. Linear regression models that use these modified loss functions during training are referred to collectively as penalized linear regression.

One popular penalty is to penalize a model based on the sum of the squared coefficient values. This is called an L2 penalty. An L2 penalty minimizes the size of all coefficients, although it prevents any coefficients from being removed from the model.

* l2\_penalty = sum j=0 to p beta\_j^2

Another popular penalty is to penalize a model based on the sum of the absolute coefficient values. This is called the L1 penalty. An L1 penalty minimizes the size of all coefficients and allows some coefficients to be minimized to the value zero, which removes the predictor from the model.

* l1\_penalty = sum j=0 to p abs(beta\_j)

Elastic net is a penalized linear regression model that includes both the L1 and L2 penalties during training.

Using the terminology from “[The Elements of Statistical Learning](https://amzn.to/3aBTnMV),” a hyperparameter “*alpha*” is provided to assign how much weight is given to each of the L1 and L2 penalties. Alpha is a value between 0 and 1 and is used to weight the contribution of the L1 penalty and one minus the alpha value is used to weight the L2 penalty.

* elastic\_net\_penalty = (alpha \* l1\_penalty) + ((1 – alpha) \* l2\_penalty)

For example, an alpha of 0.5 would provide a 50 percent contribution of each penalty to the loss function. An alpha value of 0 gives all weight to the L2 penalty and a value of 1 gives all weight to the L1 penalty.

The benefit is that elastic net allows a balance of both penalties, which can result in better performance than a model with either one or the other penalty on some problems.

Another hyperparameter is provided called “*lambda*” that controls the weighting of the sum of both penalties to the loss function. A default value of 1.0 is used to use the fully weighted penalty; a value of 0 excludes the penalty. Very small values of lambada, such as 1e-3 or smaller, are common.

* elastic\_net\_loss = loss + (lambda \* elastic\_net\_penalty)