# Regularized Linear Regression

By Asm Nurussafa

# Agenda

1. Introduction
2. Fitting a line to a dataset- Regression
3. Linear Regression model
4. Bias and Variance
5. Regularization
6. Ridge (L2)
7. Lasso (L1)
8. Elastic-Net
9. Early Stopping
10. Summary

# 1. Introduction

**Interviewer:** What is your biggest strength ?

**Me:** I am an expert in machine learning.

**Interviewer:** What‘s 6+10 ?

**Me:** 3 .

**Interviewer:** Nowhere near! It‘s 16 !

**Me:** Okay, it‘s 16.

.

**Interviewer:** What is 10+20 ?

**Me:** 16 ☺

# 2. Fitting a line to a data
# - Regression

# 3. Linear Regression

A linear model makes a prediction by simply computing a weighted sum of the input features, plus a constant called the bias term (also called the intercept term).

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- y is the feature.
- n is the number of features.
- X is the ith feature value.
- Θ is the jth model parameter (including the bias term Θ and the feature weights)

This can be written much more concisely using a vectorized form.

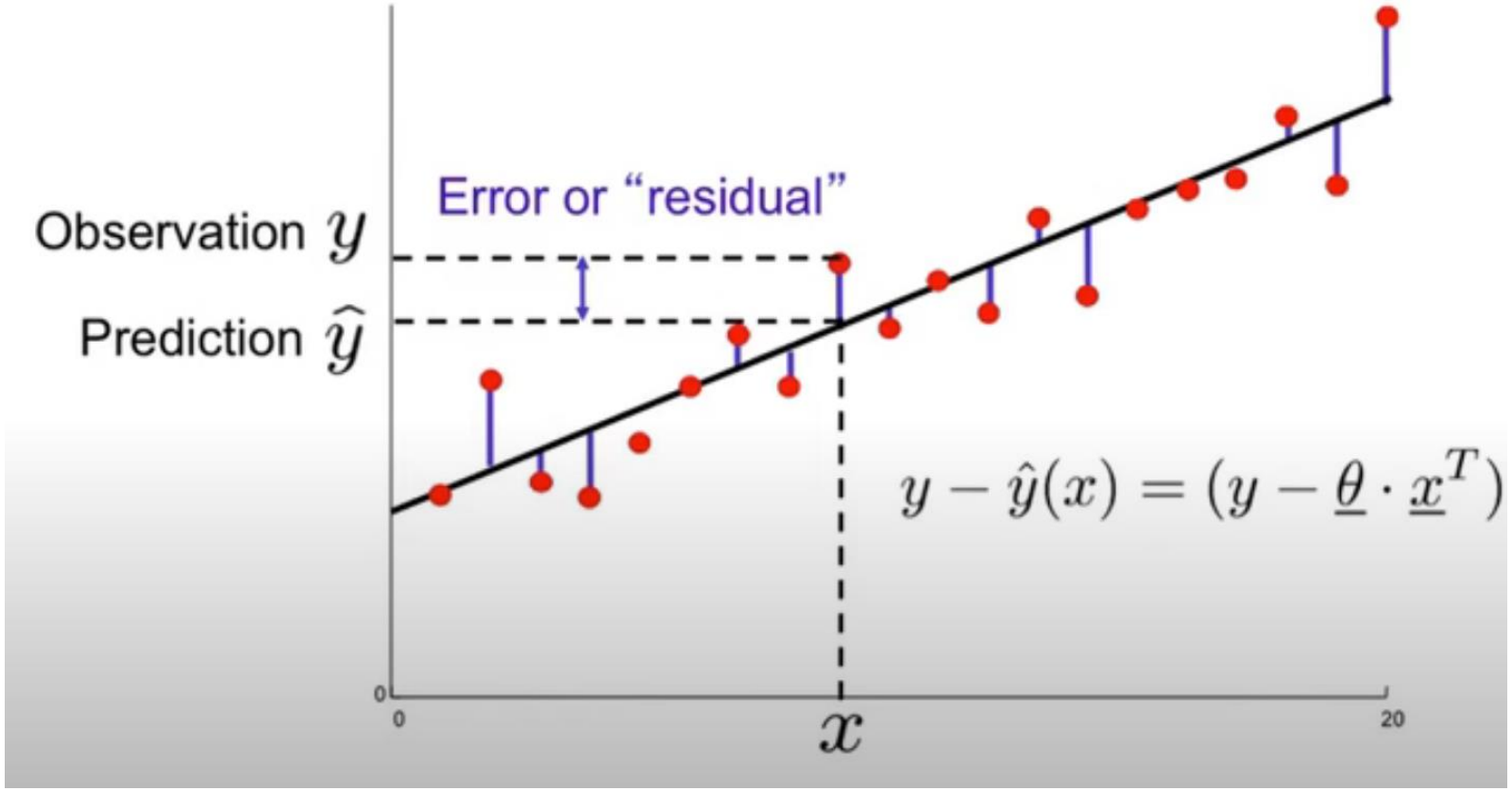
$$\hat{y} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta} \cdot \mathbf{x}$$

# 3. Linear Regression(cont'd)

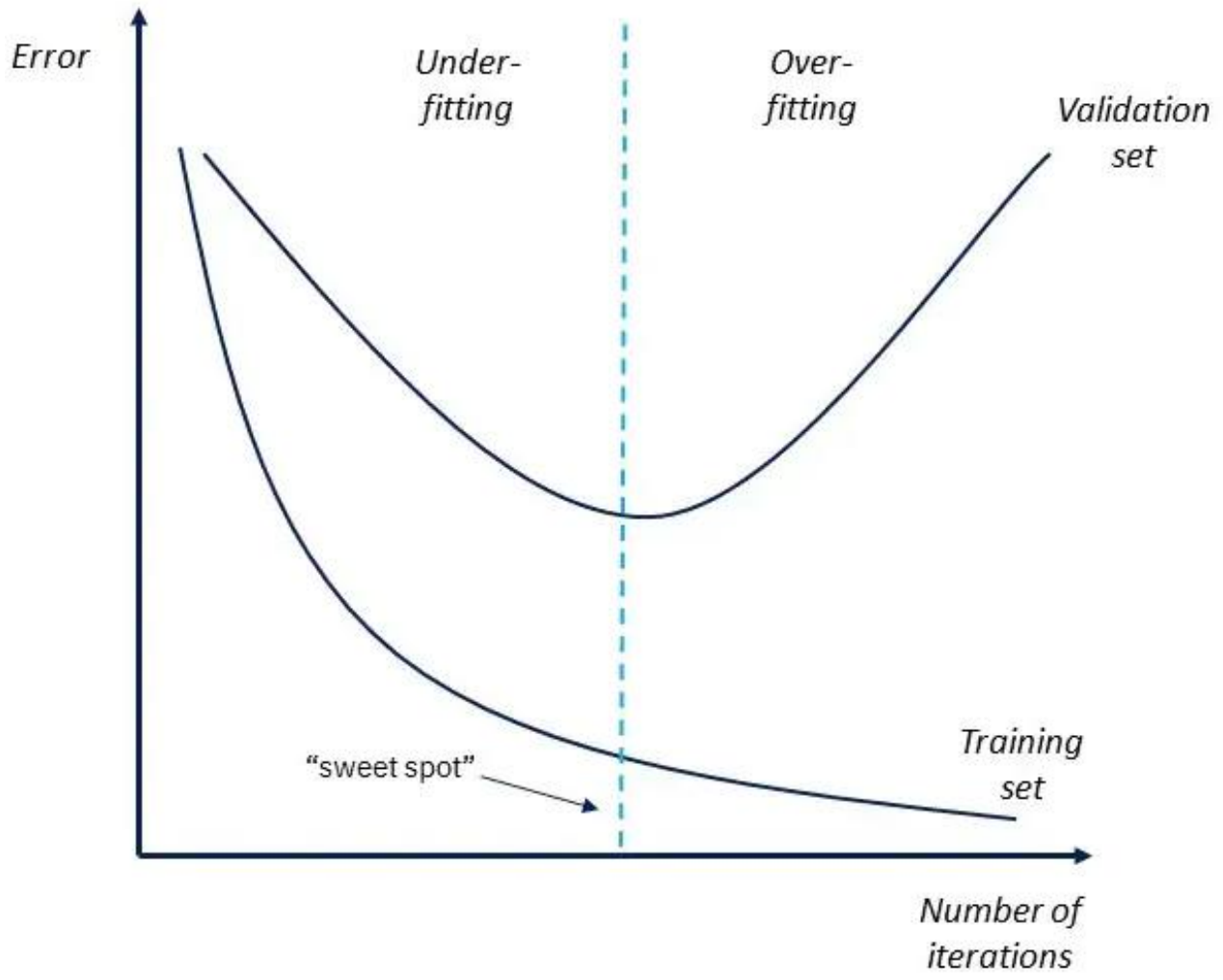
## Mean squared error (MSE) cost function(Least squares method)

-How can we quantify error?

MSE, $L(\theta) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \beta^\top \boldsymbol{x}_i|^2$



# 4. Bias vs Variance




- *„Sweet spot“* – low bias and low variance.
- Low bias is the ability to model the true relationship.
- Low variance is the ability to produce consistent predictions across different datasets.

Source: https://www.ibm.com/cloud/learn/overfitting

# 5. Regularization: An overview

The idea of regularization revolves around modifying the loss function L; in particular, we add a regularization term that penalizes some specified properties of the model parameters

$$L_{reg}(\beta) = L(\beta) + \lambda R(\beta),$$

- where $\lambda$ is a scalar that gives the weight (or importance) of the regularization term.

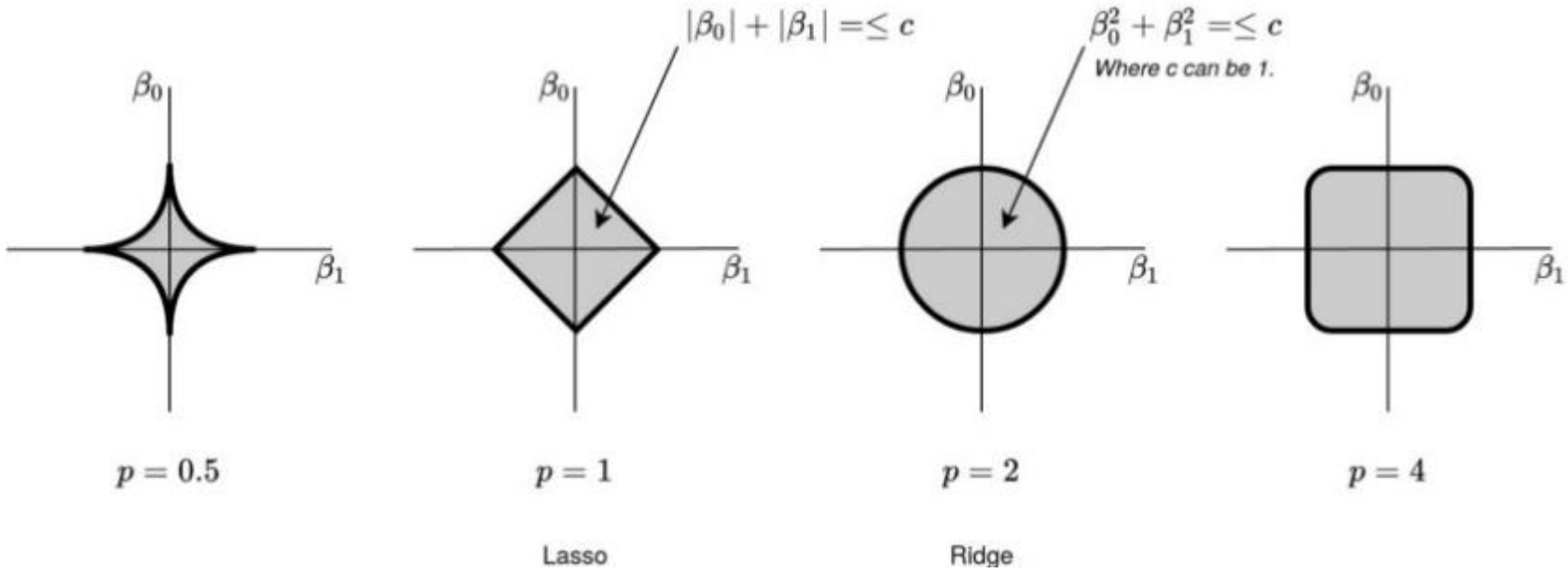
Fitting the model using the modified loss function Lreg would result in model parameters with desirable properties (specified by R).

- "Shrinks" the parameters towards zero
- Lambda large: we prefer small theta to small MSE
- Regularization term is independent of the data; paying more attention reduces our variance.

# 5. Regularization: Functions

Different choices of regularization can be chosen from the Lp vector norms, or more generally the Lp regularizer: $(\sum|\beta_i|^\rho)^{\frac{1}{p}}$

Iso-surfaces: $\|\beta\|_\rho = constant$




Source: https://medium.com/@kiprono_65591/regularization-a-technique-used-to-prevent-over-fitting-886d5b361700

# 6. Ridge Regression(L2)

Alternatively, we can choose a regularization term that penalizes the squares of the parameter magnitudes. Then, our regularized loss function is:

$$L_{Ridge}(\beta) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i|^2 + \lambda \sum_{j=1}^{J}\beta_j^2.$$

Note that $\sum_{j=1}^{J}|\beta_j|^2$ is the square of the $\boldsymbol{l_2}$ norm of the vector $\boldsymbol{\beta}$

$$\sum_{j=1}^{J}\beta_j^2 = \|\boldsymbol{\beta}\|_2^2$$

# 7. LASSO Regression (L1)

Since we wish to discourage extreme values in model parameter, we need to choose a regularization term that penalizes parameter magnitudes. For our loss function, we will again use MSE.

Together our regularized loss function is:

$$L_{LASSO}(\beta) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i|^2 + \lambda \sum_{j=1}^{J}|\beta_j|.$$

Note that $\sum_{j=1}^{J}|\beta_j|$ is the $\boldsymbol{l}_1$ norm of the vector $\boldsymbol{\beta}$

$$\sum_{j=1}^{J}|\beta_j| = \|\boldsymbol{\beta}\|_1$$

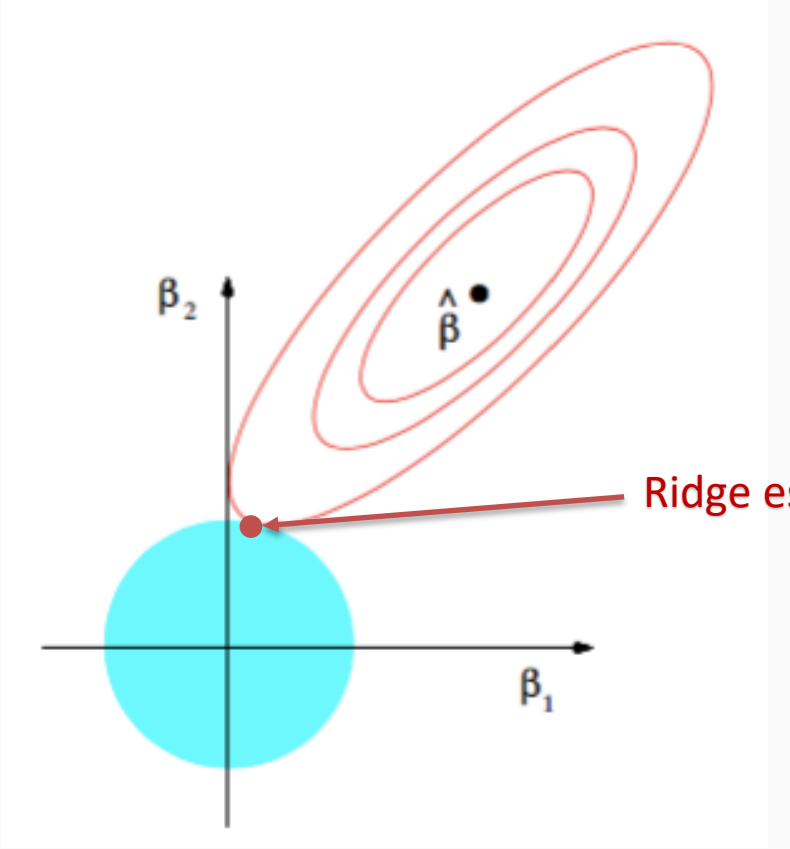
# Choosing $\lambda$

In both ridge and LASSO regression, we see that the larger our choice of the **regularization parameter** $\lambda$, the more heavily we penalize large values in $\boldsymbol{\beta}$,

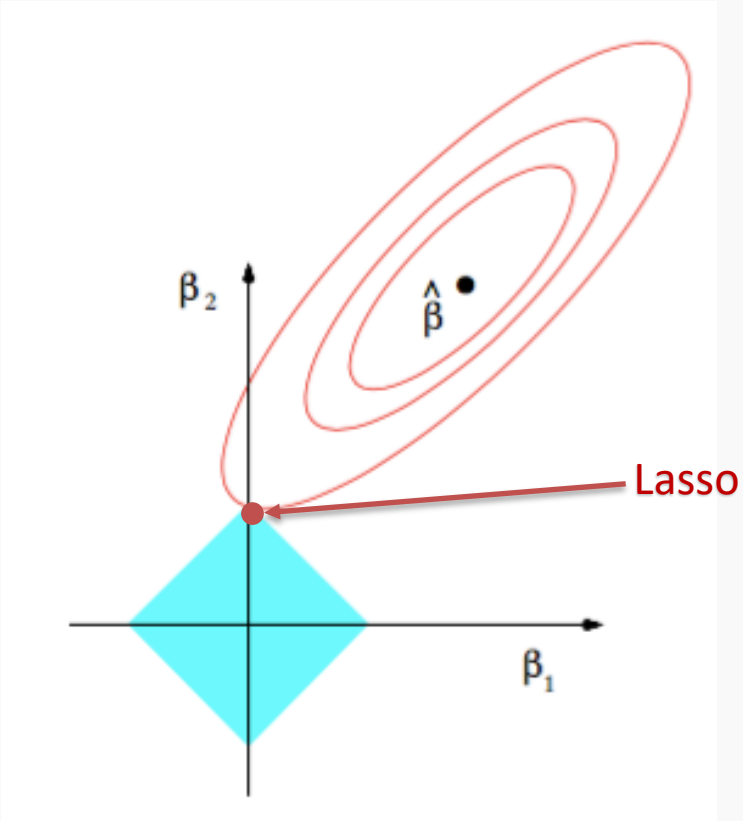
- If $\lambda$ is close to zero, we recover the MSE, i.e. ridge and LASSO regression is just ordinary regression.
- If $\lambda$ is sufficiently large, the MSE term in the regularized loss function will be insignificant and the regularization term will force $\boldsymbol{\beta}_{\text{ridge}}$ and $\boldsymbol{\beta}_{\text{LASSO}}$ to be close to zero.

To avoid ad-hoc choices, we should select $\lambda$ using cross-validation.

# Ridge and Lasso visualized




The ridge estimator is where the constraint and the loss intersect.




The Lasso estimator tends to zero out parameters as the OLS loss can easily intersect with the constraint on one of the axis.

# Examples

In [ ]:
```
from sklearn.linear_model import Lasso
```

In [22]:
```
lasso_regression = Lasso(alpha=1.0, fit_intercept=True)
lasso_regression.fit(np.vstack((X_train, X_val)), np.hstack((y_train, y_val)))

print('Lasso regression model:\n {} + {}^T . x'.format(lasso_regression.intercept_, lasso_regression.coe
```

```
Lasso regression model:
 10.424895873901445 + [ 0.24482603  3.48164594  1.84836859 -0.06864603 -0.         -0.
 -0.02249766 -0.          0.          0.          0.          ]^T . x
```

In [ ]:
```
from sklearn.linear_model import Ridge
```

In [20]:
```
X_train = train[all_predictors].values
X_val = validation[all_predictors].values
X_test = test[all_predictors].values

ridge_regression = Ridge(alpha=1.0, fit_intercept=True)
ridge_regression.fit(np.vstack((X_train, X_val)), np.hstack((y_train, y_val)))

print('Ridge regression model:\n {} + {}^T . x'.format(ridge_regression.intercept_, ridge_regression.coe
```

```
Ridge regression model:
 -525.7662550875951 + [ 0.24007312  8.42566029  2.04098593 -0.04449172 -0.01227935  0.41902475
 -0.50397312 -4.47065168  4.99834262  0.          0.          0.29892679]^T . x
```

# 8. The Elastic Net

Elastic Net is a middle ground between Ridge Regression and Lasso Regression. The regularization term is a simple mix of both Ridge and Lasso's regularization terms, and you can control the mix ratio r. When r = 0, Elastic Net is equivalent to Ridge Regression, and when r = 1, it is equivalent to Lasso Regression.

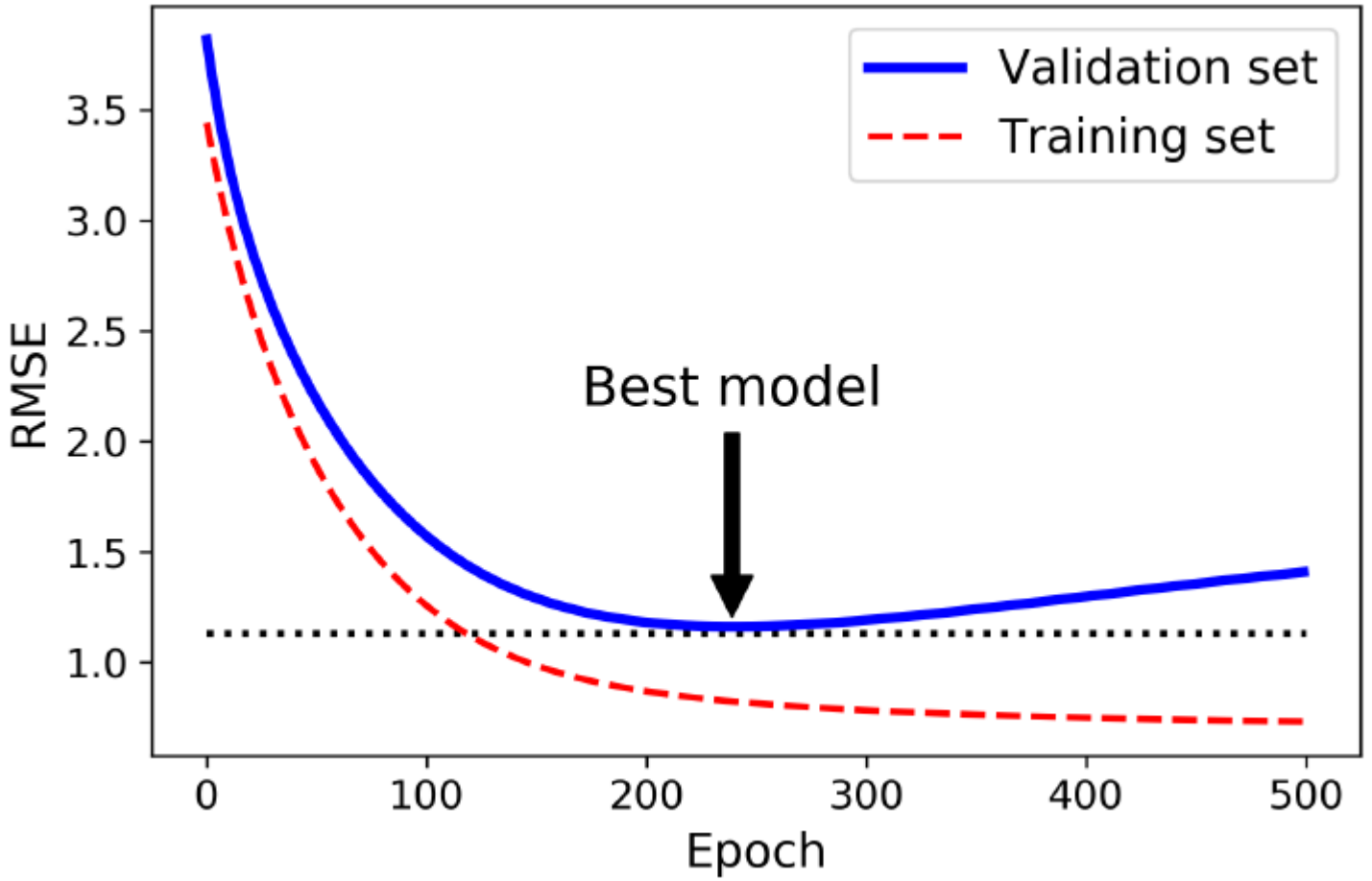
$$L_{elastic}(\theta) = L(\theta) + r\lambda \sum_{i=1}^{n} |\beta_i| + \frac{1-r}{2} \lambda \sum_{i=1}^{n} \theta_i^2$$

Here is a short example using Scikit-Learn's ElasticNet (l1_ratio corresponds to the mix ratio r)

```
>>> from sklearn.linear_model import ElasticNet
>>> elastic_net = ElasticNet(alpha=0.1, l1_ratio=0.5)
>>> elastic_net.fit(X, y)
>>> elastic_net.predict([[1.5]])
array([1.54333232])
```

# 9. Early Stopping

A very different way to regularize iterative learning algorithms such as Gradient Descent is to stop training as soon as the validation error reaches a minimum. This is called early stopping.



# 10. Summary

- So, when should you use plain Linear Regression (i.e., without any regularization), Ridge, Lasso, or Elastic Net?

- It is almost always preferable to have at least a little bit of regularization, so generally you should avoid plain Linear Regression.

- Ridge is a good default, but if you suspect that only a few features are actually useful, you should prefer Lasso or Elastic Net since they tend to reduce the useless features' weights down to zero as we have discussed.

- In general, Elastic Net is preferred over Lasso since Lasso may behave erratically when the number of features is greater than the number of training instances or when several features are strongly correlated.

# Thank you!

Any questions ?

# References

[1] G´eron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent
systems. ” O’Reilly Media, Inc.”.

[2] Theobald, O. (2017). Machine learning for absolute beginners: a plain English introduction (Vol. 157). Scatterplot press.

[3] Bonaccorso, G. (2017). Machine learning algorithms. Packt Publishing Ltd.

[4] Walsh, Wyatt. “Regularized Linear Regression Models.” Medium, Towards Data Science, 18 Jan. 2021, towardsdatascience.com/regularizedlinear-regression-models-44572e79a1b5.

[5] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1), 1.