Regression

Recap: structure risk

- In general, a machine learning problem solves the following optimization problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n}\left[L\left(Y_i, m\left(\mathbf{X}_i;\boldsymbol{\theta}\right)\right)\right] + \lambda||\boldsymbol{\theta}|| \tag{1}$$

Agenda

- Problem definition
- Representation
- Loss function
- Regularization

Problem definition

# C387

Regression

- Machine learning: predict a real-valued response variable Y by features $\mathbf{X} = [X_1, \ldots, X_p]^\top$

- Statistics: find the relationships between Y and several covariates $\mathbf{X}$

- X_j's can be
  - Quantitative variables
  - Qualitative (categorical) variables
  - Transformations of variables, e.g. $X_2 = \log\left(X_1\right)$
  - Interaction of several variables, e.g., $X_3 = X_1 X_2$
  - etc.

Linear regression

- Assume that the relationships between y and $\mathbf{x}$ is

$$Y \approx \beta_0 + \sum_{j=1}^{p} \beta_j X_j$$

- Or equivalently,

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon \tag{2}$$

where ϵ is a random noise with $E(\epsilon) = 0$ and Var$(\epsilon) = \sigma^2 < \infty$

- β_0 is the intercept (or bias)

- β_j is the coefficient (or weight) of X_j; it is the effect of X_j when the other covariates are fixed

    - sign

    - is it zero?

Categorical covariate

- If $X_1 \in \{1,2,3\}$ be a categorical variable, what does $\beta_1 X_1$ means?

- Transform a categorical covariate by one-hot encoding: let

$$X_{1k} = \begin{cases} 1 & \text{if } X_1 = k \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } k = 1,2,3$$

Then equation (2) becomes

$$Y = \beta_0 + \sum_{k=1}^{3} \beta_{1k} X_{1k} + \sum_{j=2}^{p} \beta_j X_j + \epsilon$$

Pros

- Interpretable

- If the covariates are independent with each other, the estimates of β_j's will be consistent even when the model is misspecified

Cons

- Too simple to be true
  - less predictive (due to large bias)
  - may lead to wrong conclusion (e.g. 涓滴理論 etc.)

Nonparametric regression

In nonparametric regression, we assume

$$Y = m(\mathbf{X};\boldsymbol{\theta}) + \epsilon \tag{3}$$

where $m(\mathbf{X};\boldsymbol{\theta})$ is an unknown function parameterized by $\boldsymbol{\theta}$ and ϵ is a random noise with $E(\epsilon) = 0$ and Var$(\epsilon) = \sigma^2 < \infty$.

↑

i.e. $E(Y|\mathbf{X}) = m(\mathbf{X}|\theta)$

Popular methods for nonparametric regression

- Local regression
- Basis representation
  - neural networks
  - gradient boosting
  - splines (piecewise polynomial)
  - etc.

Nonparametric regression by piecewise linear functions

For simplicity, we'll illustrate the idea of piecewise linear regression with $X \in \mathbb{R}$. Let

$$\begin{aligned} m(X) &= \beta_0 + \beta_1 X + \beta_2 (X - x_1) \cdot I(X > x_1) + \cdots \\ &= \beta_0 + \beta_1 X + \beta_2 (X - x_1)^+ + \cdots \end{aligned} \tag{4}$$

then $m(X)$ becomes a piecewise linear function

ReLU activation yield to piecewise linear functions

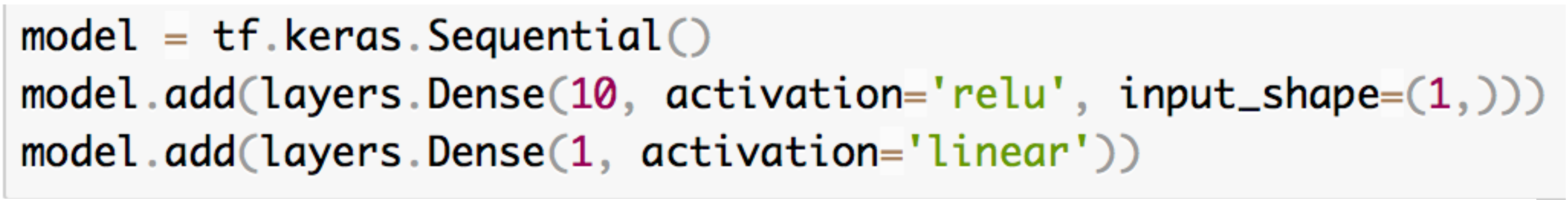
- Recall that

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \triangleq x^+$$

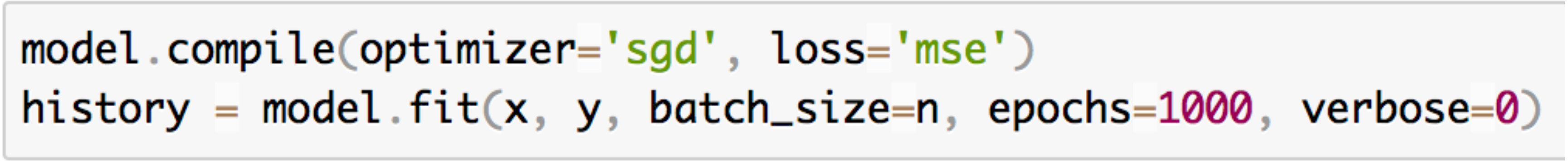
- Equation (4) can be rewritten as

$$\begin{aligned} m(X) &= \beta_0 + \beta_1 X + \beta_2 (X - x_1)^+ + \cdots \\ &= \beta_0 + \beta_1 X + (\underset{\substack{\uparrow \\ \text{weight}}}{\beta_2} X - \underset{\substack{\uparrow \\ \text{bias}}}{b_1})^+ + \cdots \end{aligned}$$

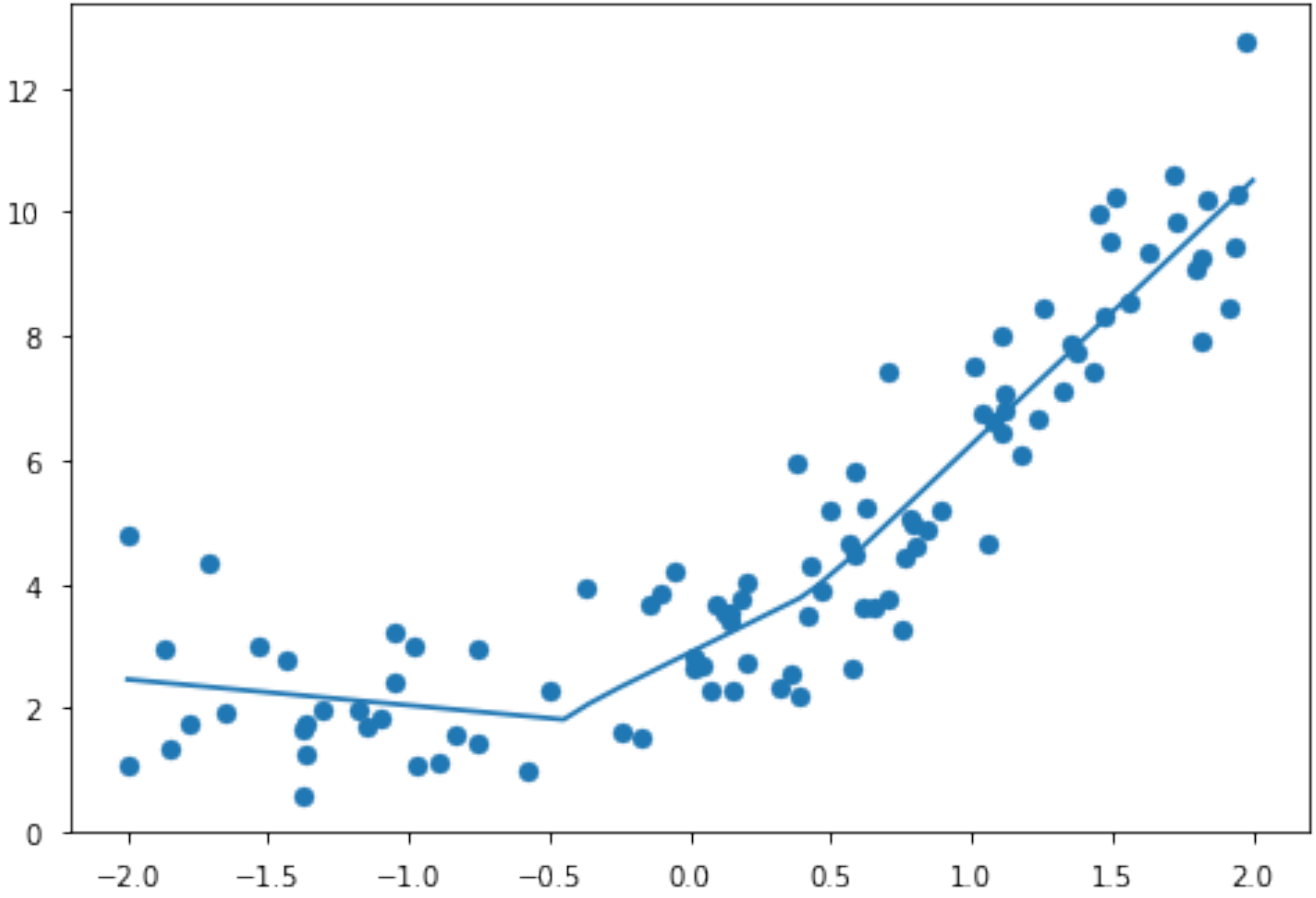
C387



```
model = tf.keras.Sequential()
model.add(layers.Dense(10, activation='relu', input_shape=(1,)))
model.add(layers.Dense(1, activation='linear'))
```



```
model.compile(optimizer='sgd', loss='mse')
history = model.fit(x, y, batch_size=n, epochs=1000, verbose=0)
```



Loss function

Least squares estimation

Let $r = Y - m\left(\mathbf{X}; \boldsymbol{\theta}\right)$ denote the prediction error:

- the sign of r is not important (usually)
- the squared–error loss r^2 is the most popular loss function for regression problems for both numerical and decision–theoretic reasons

Least squares estimation

- In linear regression, we solve

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \frac{1}{n}\sum_{i=1}^{n}\left[Y_i-\beta_0-\sum_{j=1}^{p}\beta_j X_{ij}\right]^2+\lambda\|\boldsymbol{\beta}\| \qquad (5)$$

- In nonparametric regression, we solve

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n}\left[Y_i-m\left(\mathbf{X}_i;\boldsymbol{\theta}\right)\right]^2+\lambda\|\boldsymbol{\theta}\| \qquad (6)$$

Other popular choices

- Absolute–error loss:

$$L\left(Y,\hat{Y}\right)=\left|Y-\hat{Y}\right|$$

- Huber loss:

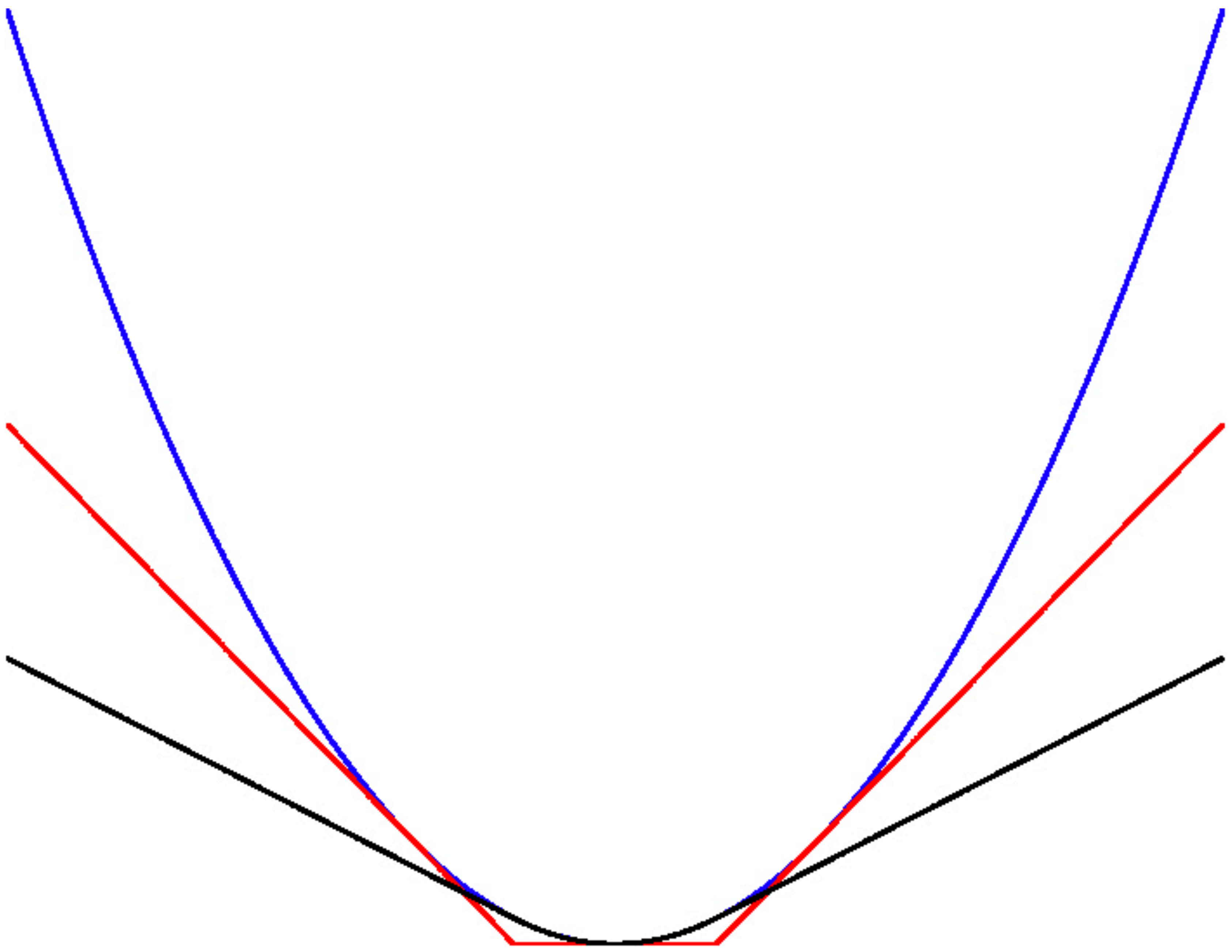
$$L_\delta\left(Y,\hat{Y}\right)=\begin{cases}\frac{1}{2}\left(Y-\hat{Y}\right)^2 & \text{if } \left|Y-\hat{Y}\right|\leq\delta\\ \delta\left(\left|Y-\hat{Y}\right|-\frac{1}{2}\delta\right) & \text{otherwise,}\end{cases}$$

C387

- Epsilon–sensitive loss:

$$L_{\epsilon}\left(y,\hat{y}\right) = \begin{cases} 0 & \text{if } \left|Y - \hat{Y}\right| \leq \epsilon \\ \left|Y - \hat{Y}\right| - \epsilon & \text{otherwise,} \end{cases}$$

# C387



blue: quadratic
black: huber
red: ϵ-sensitive

# C387

Regularizations

Popular regularizations for linear regression

- Ridge regression (ℓ_2 regularization):

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

- LASSO regression (ℓ_1 regularization):

$$|\boldsymbol{\beta}\| = \sum_{j=1}^{p} \left|\beta_j\right|$$

- Elastic net:

$$|\boldsymbol{\beta}\| = (1-\alpha) \cdot \sum_{j=1}^{p} \beta_j^2 + \alpha \cdot \sum_{j=1}^{p} \left|\beta_j\right|$$

Ridge regression

Equation (5) becomes

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \frac{1}{n}\sum_{i=1}^{n}\left[Y_i-\beta_0-\sum_{j=1}^{p}\beta_j X_{ij}\right]^2+\lambda\sum_{j=1}^{p}\beta_j^2 \qquad (7)$$

- works well with correlated predictors (multicollinearity)
- biased estimation but with smaller variance and MSE
- shrink β_j's toward 0

Ridge regression

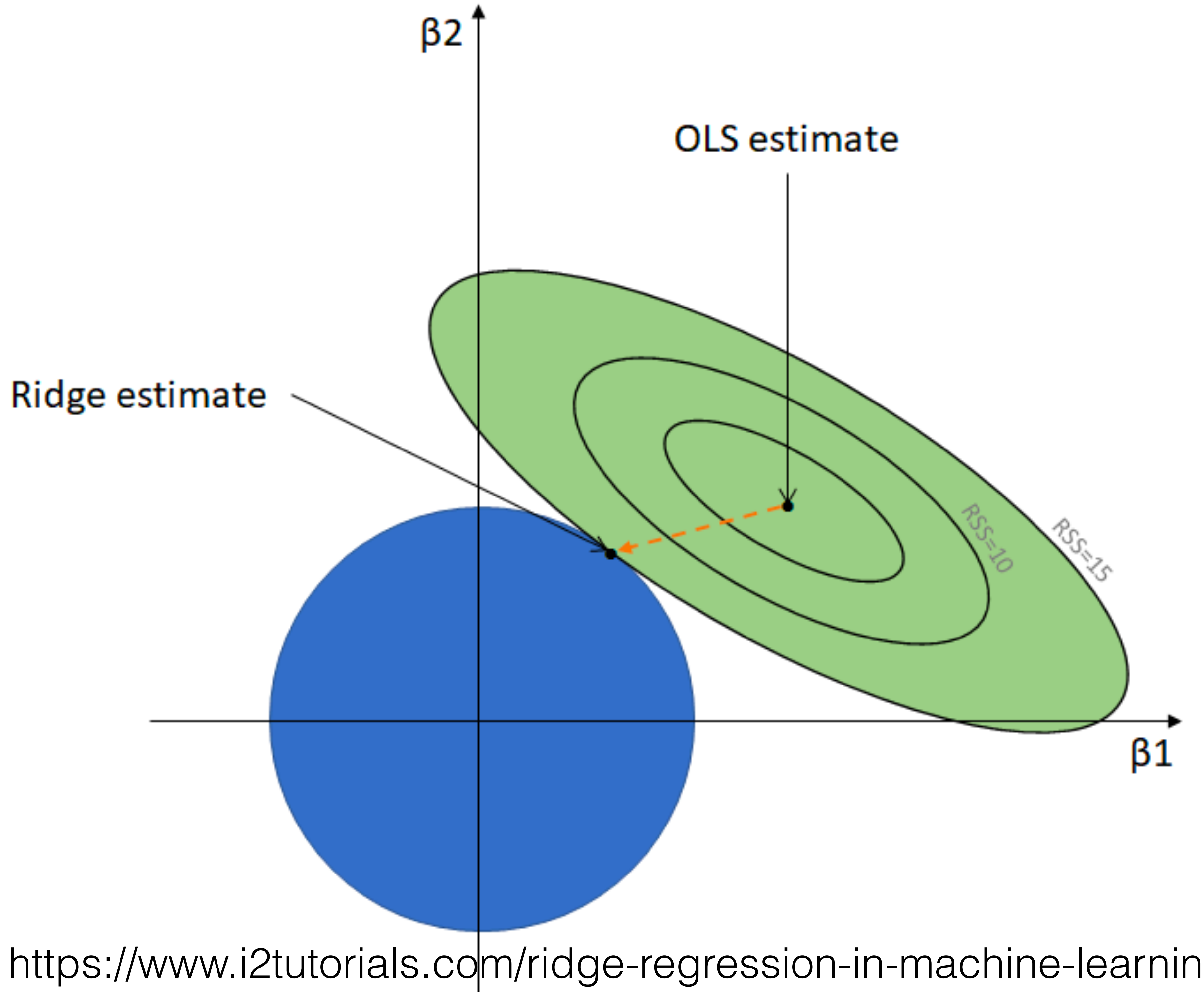
Equation (7) comes from the Lagrangian of

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \frac{1}{n}\sum_{i=1}^{n}\left[Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ij}\right]^2$$

subject to $\sum_{j=1}^{p}\beta_j^2 \leq C$

for some hyperparameter C

Ridge regression



https://www.i2tutorials.com/ridge-regression-in-machine-learning/

# C387

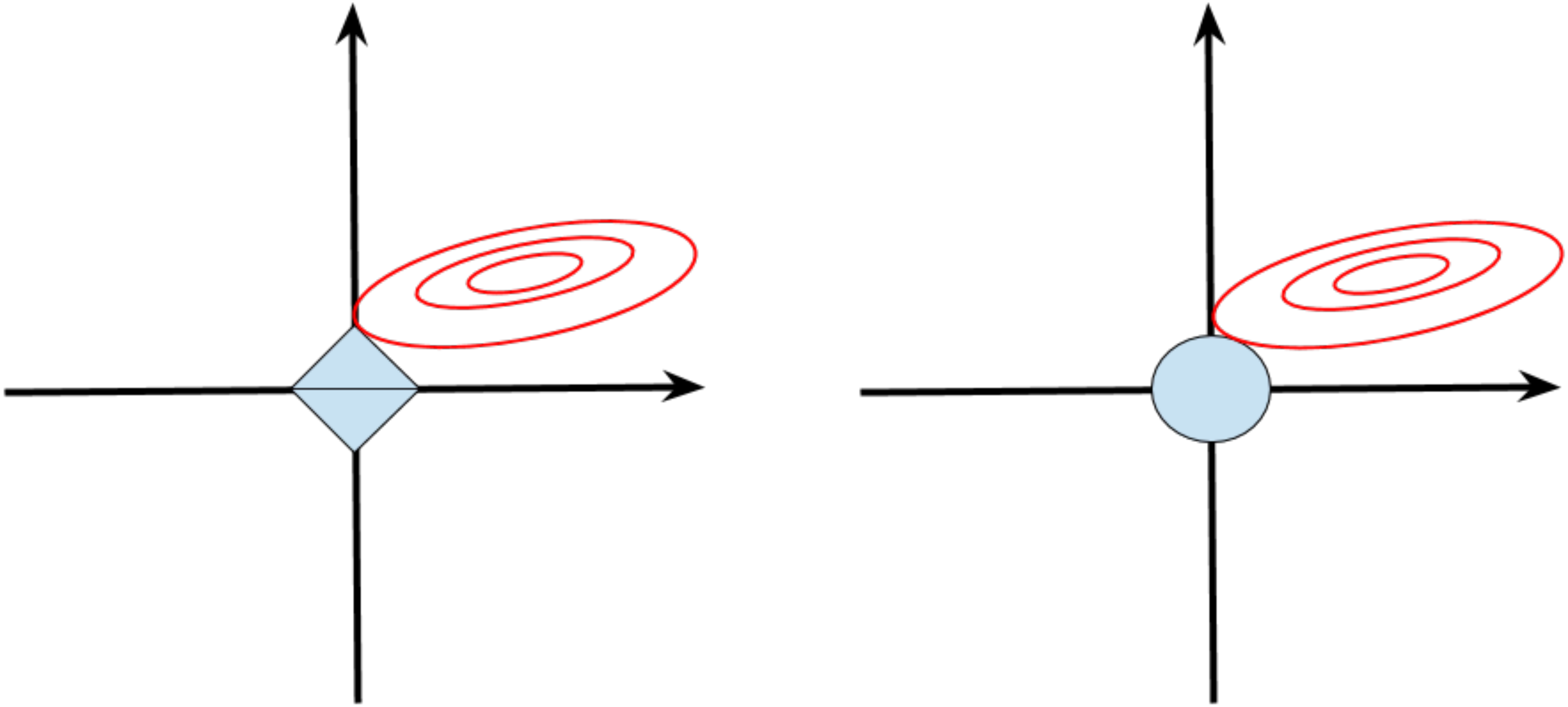
LASSO

Equation (5) becomes

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \frac{1}{n}\sum_{i=1}^{n}\left[Y_i-\beta_0-\sum_{j=1}^{p}\beta_j X_{ij}\right]^2+\lambda\sum_{j=1}^{p}\left|\beta_j\right|$$

- automatic variable selection with model consistency
- biased estimation but sign consistent
- works poorly with correlated covariates

LASSO vs ridge regression



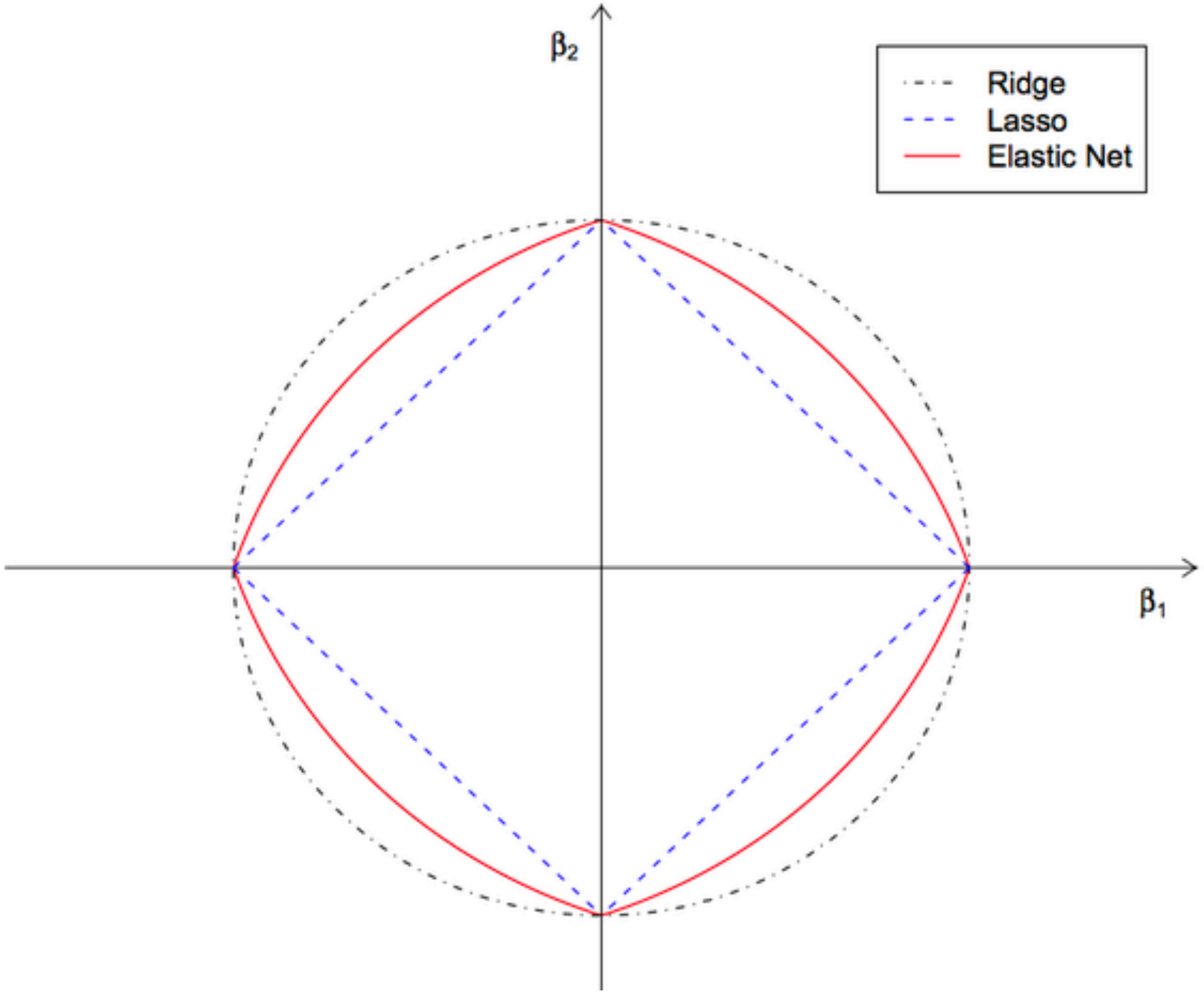
https://towardsdatascience.com/can-you-answer-these-5-questions-about-lasso-and-ridge-regression-1138536f4f80

Elastic net

Elastic net is a linear combination of ridge and LASSO.

- inherit the pros from both ridge and LASSO
- introduce an addition hyperparameter α

Regularizations

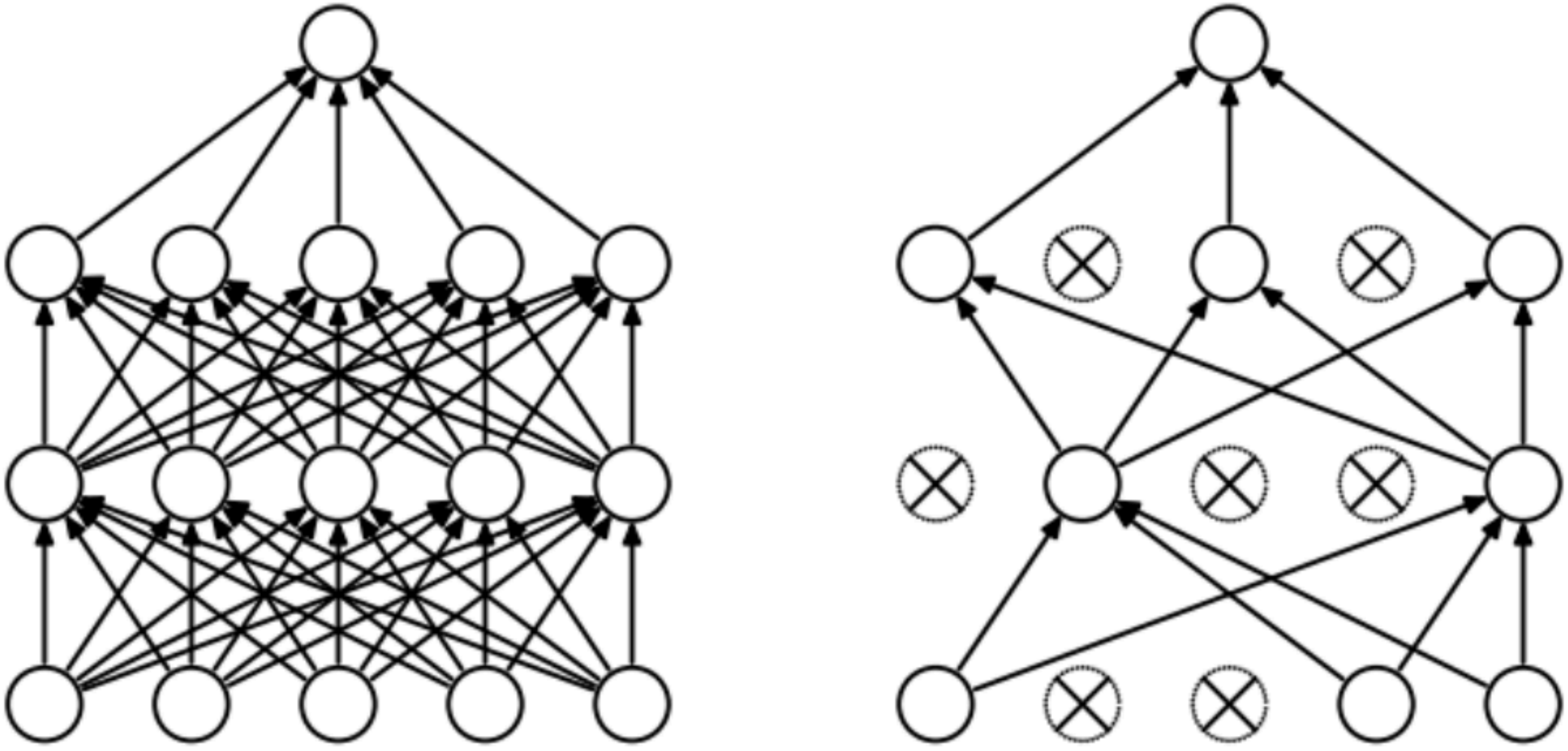


https://venali.medium.com/conventional-guide-to-supervised-learning-with-scikit-learn-elastic-net-generalized-linear-80ecc2574052

Popular regularizations for nonparametric regression

- Dropout
- Early stopping
- Data augmentation
- etc.

Dropout

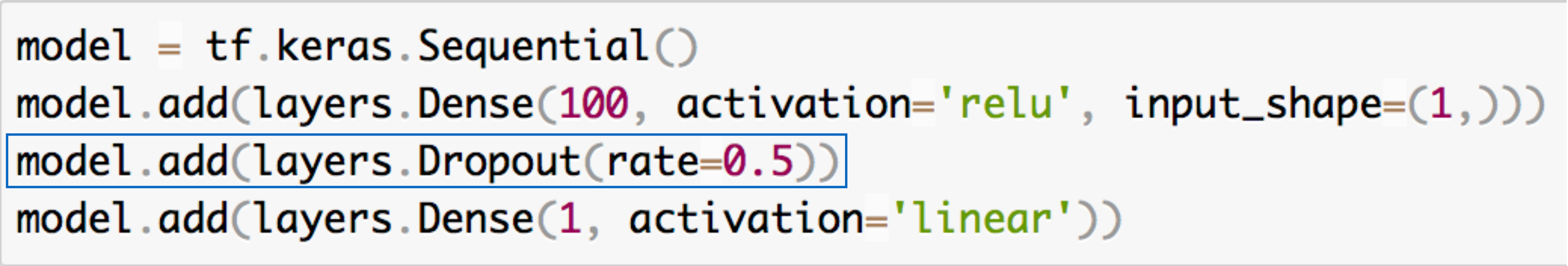


(a) Standard Neural Net

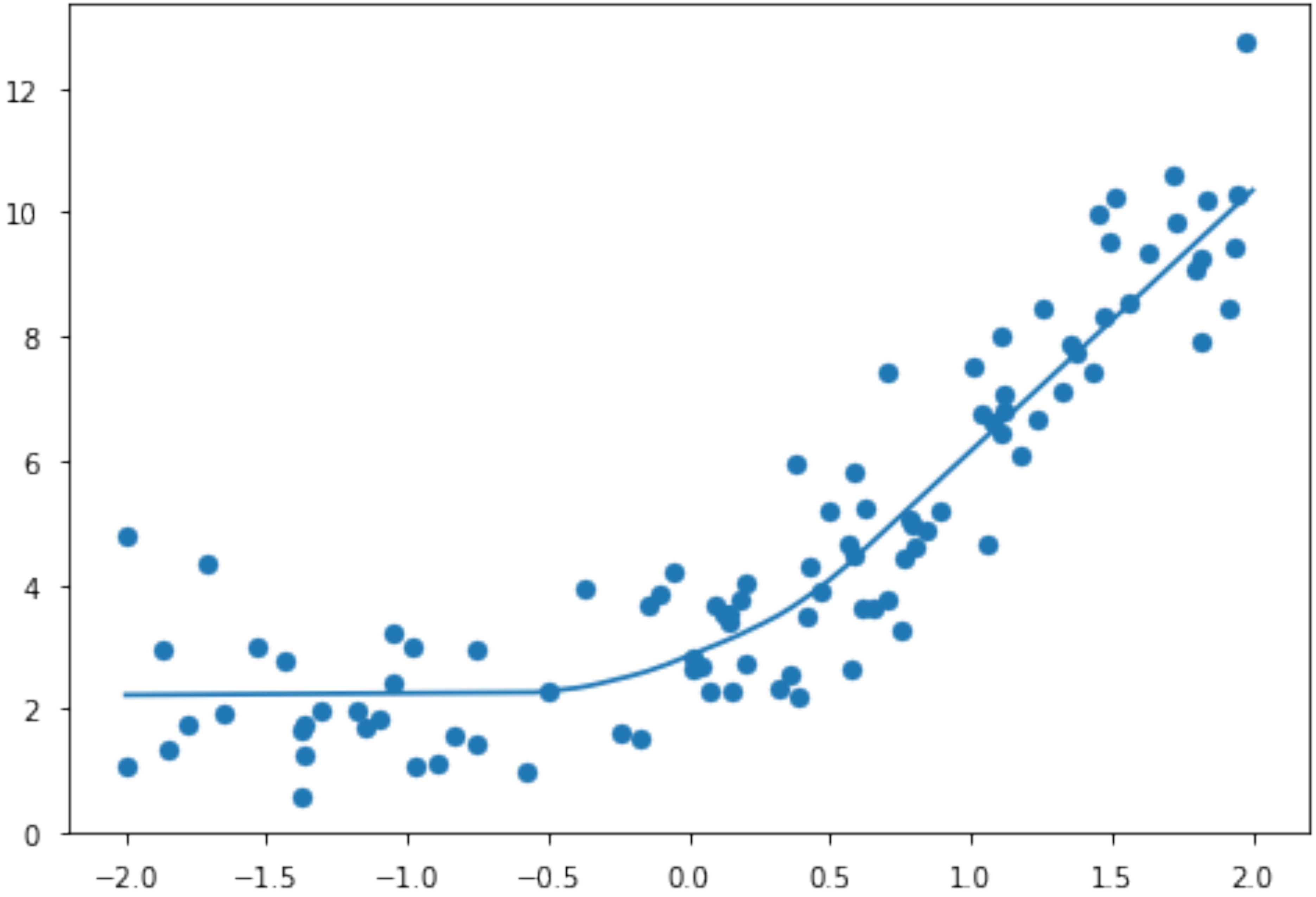
(b) After applying dropout.

Randomly ignore a fraction of hidden neurons in each iteration of gradient descent

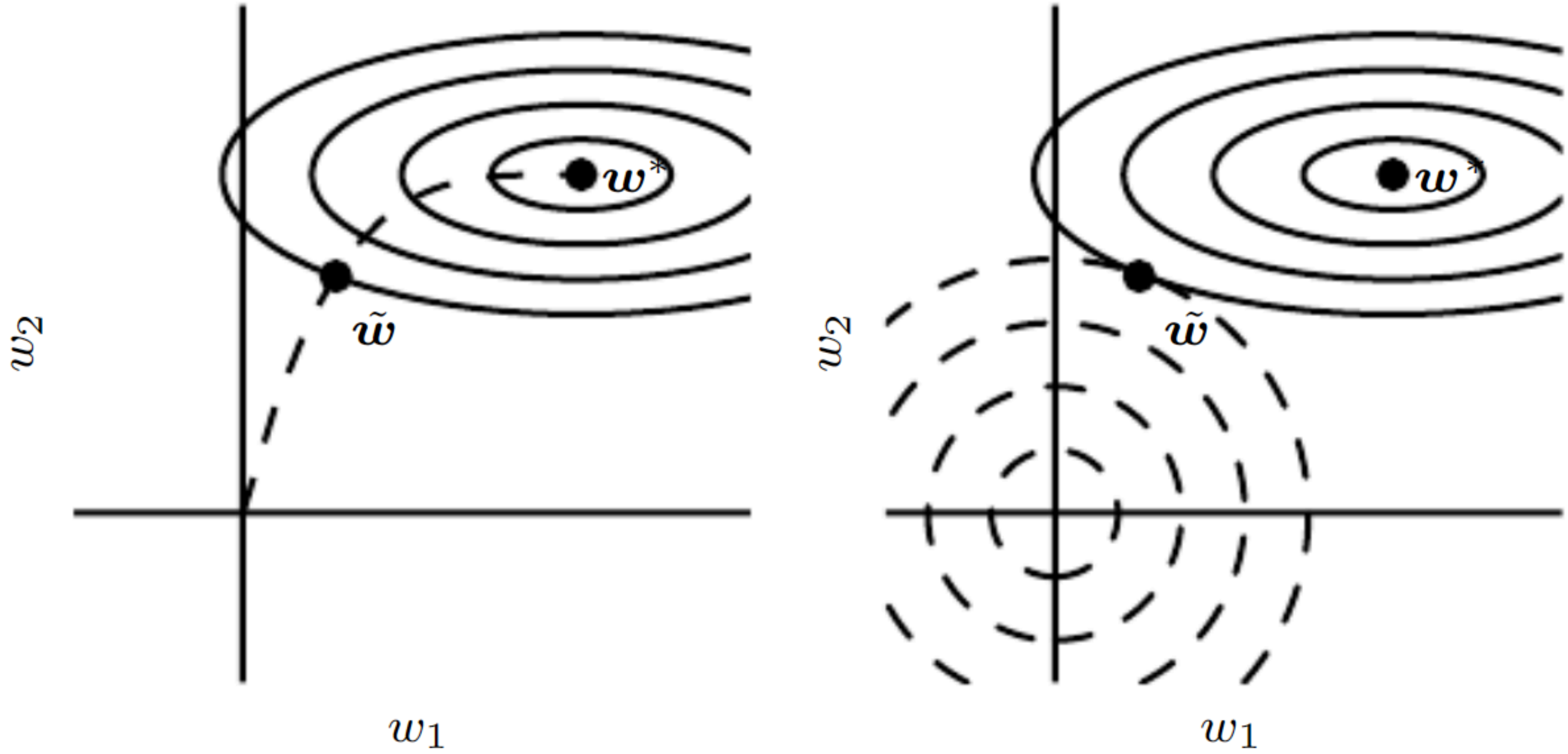
C387



```
model = tf.keras.Sequential()
model.add(layers.Dense(100, activation='relu', input_shape=(1,)))
model.add(layers.Dropout(rate=0.5))
model.add(layers.Dense(1, activation='linear'))
```

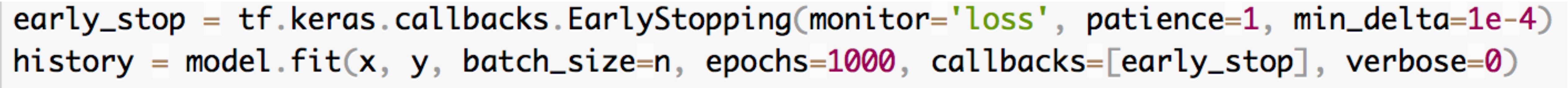


Early stopping

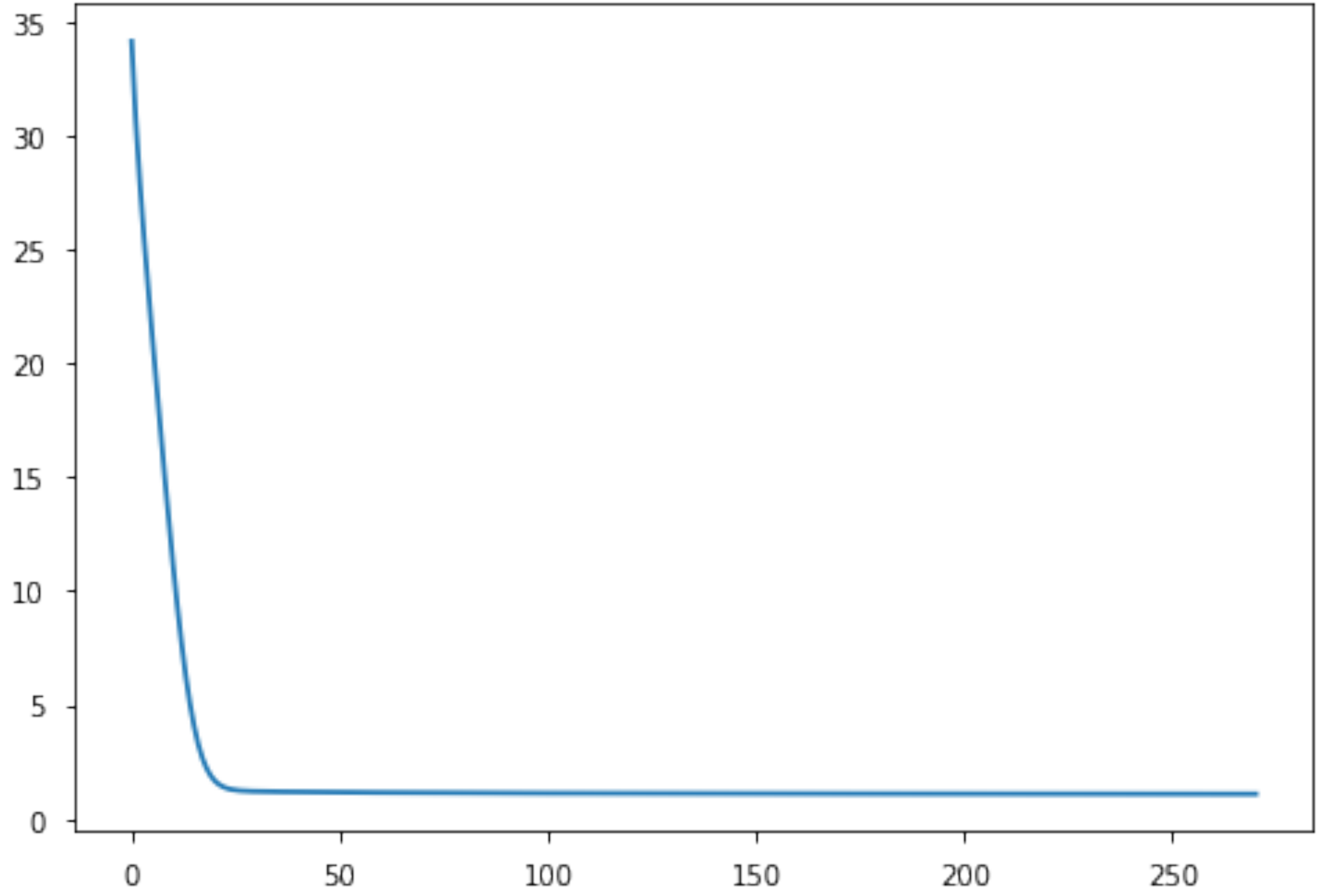


Early stopping restricts the gradient descent algorithm to a relatively small volume of parameter space in the neighborhood of the initial parameter $\boldsymbol{\theta}_0$

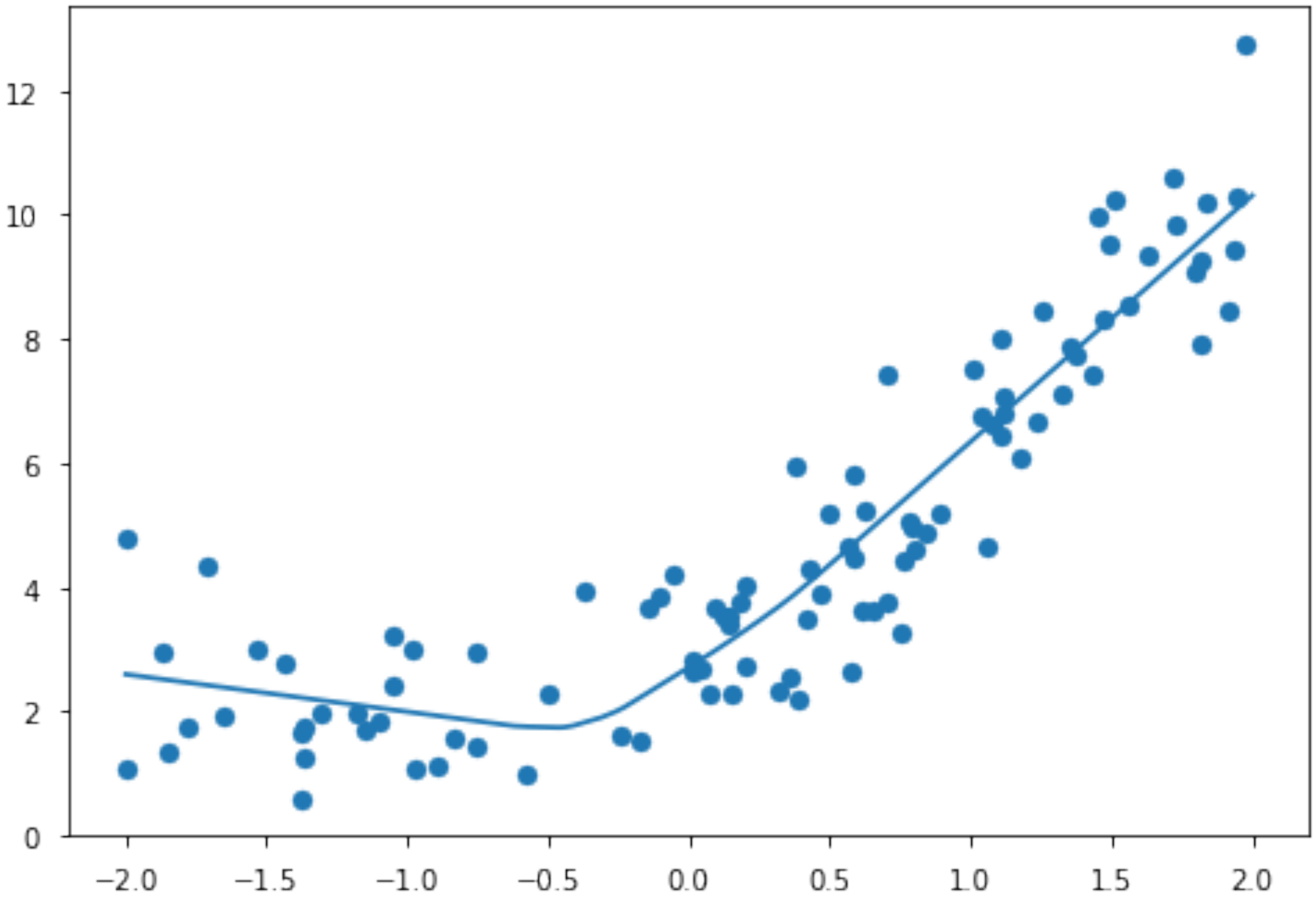
C387



```python
early_stop = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=1, min_delta=1e-4)
history = model.fit(x, y, batch_size=n, epochs=1000, callbacks=[early_stop], verbose=0)
```



C387



# C387

References

- Chapter 16 of Principles and Techniques of Data Science
- Chapter 7 of Deep Learning by Goodfellow et al.
- tf.keras.layers.Dropout
- tf.keras.callbacks.EarlyStopping

Homework

1. Find the best regression model (try your best) for the diabetes dataset.

2. Is “Average blood pressure” an important factor for diabetes disease? Explain this by cross-validations.

Bonus: use auto-sklearn or AutoKeras to search for a good regression model automatically.