Intuition: https://www.youtube.com/watch?v=1dKRdX9bfIo (https://www.youtube.com/watch?v=1dKRdX9bfIo)

Imports and global variables

In [1]:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import ElasticNet
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV

import warnings

parameters = {
    'l1_ratio': [0.1, 0.2, 0.4, 0.5, 0.6, 0.8, 1],
    'normalize': [True, False],
    'precompute': [True, False],
    'max_iter': [10, 100, 1000, 2500],
    'tol': [0.01, 0.001, 0.0001, 0.00005]
}
```

Implement Elastic Net regularization, as developed in the lecture. Use either ElasticNetCV (from sklearn), or ElasticNet together with Grid-Search (CV), or augment (from scratch) the LASSO program from the lecture. Test your implementation by analyzing a dataset of your choice, e.g., the cancer database as presented in the lecture.

Load data

In [2]:

```
# Load database
cancer = load_breast_cancer()

# Create data frame
cancer_df = pd.DataFrame(cancer.data, columns=cancer.feature_names)

# Construct train and test data sets
X = cancer.data
Y = cancer.target

X_train, X_test, y_train, y_test = train_test_split(
    X,
    Y,
    test_size=0.3,
    random_state=31,
    stratify=Y
)
```

Instantiate ElasticNet with GridSearchCV

In [3]:

```
instance_elastic_net = GridSearchCV(
    ElasticNet(),
    param_grid=parameters
)
```

Fit and test the elastic net

In [8]:

```
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    instance_elastic_net.fit(X_train, y_train)
    print('In-sample score:', instance_elastic_net.score(X_train, y_train))
    print('Out-of-sample score:', instance_elastic_net.score(X_test, y_test))
```

```
In-sample score: 0.6633454464746772
Out-of-sample score: 0.6602398339852424
```

In [ ]:

```

```