Nick A, Noah, Nick D, Angel, Emma, Dominic

Food for Thought

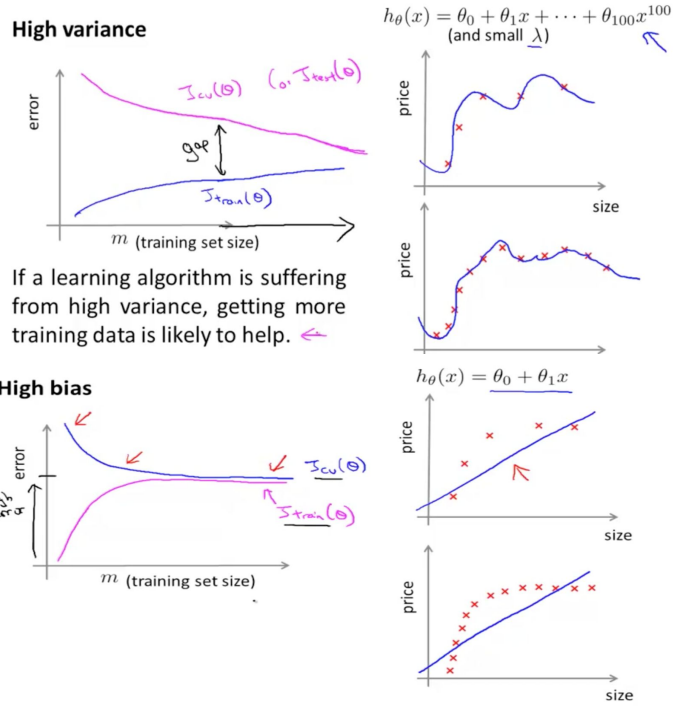
How do you decide what *degree* a polynomial regression should be in order to best fit the data?
What are some *performance measurements* for polynomial regression algorithms?
How can you improve a machine learning model that is *prone to overfitting*?

What is a Learning Curve?

- A graph that compares the performance of a model on training and testing data over a varying number of training instances
- A useful tool that evaluates model performance; shows whether or not a model is suffering from bias/variance

Types of Learning Curves:

- Bad Learning Curve (*High Variance*): There is a large gap between errors
- Bad Learning Curve (*High Bias*): Training and testing errors converge and are high
- Ideal Learning Curve: A model that generalizes to new data and testing and training learning curves converge at similar values



Code

```
import numpy as np

from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt

m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.5 * X**2 + X + 2 + np.random.randn(m, 1)
poly_features = PolynomialFeatures(degree=2, include_bias=False)
X_poly = poly_features.fit_transform(X)
poly_reg = LinearRegression()
poly_reg.fit(X_poly, y)

def plot_learning_curves(model, X, y):
    X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2)
    train_errors, val_errors = [], []
    for m in range(1, len(X_train)):
        model.fit(X_train[:m], y_train[:m])
        y_train_predict = model.predict(X_train[:m])
        y_val_predict = model.predict(X_val)
        train_errors.append(mean_squared_error(y_train_predict, y_train[:m]))
        val_errors.append(mean_squared_error(y_val_predict, y_val))
    plt.plot(np.sqrt(train_errors), "r-+", linewidth=2, label="train")
    plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label="val")

lin_reg = LinearRegression()
plot_learning_curves(lin_reg, X, y)
plot_learning_curves(poly_reg, X, y)
```

Resources

Learning Curves in Machine Learning: https://www.dataquest.io/blog/learning-curves-machine-learning/
Learning Curves in Machine Learning: https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8_452