# Technical Analysis: Linear Regression & Gradient Descent

## 1. Variable Definitions and Functional Mapping

In this predictive model, we define the relationship between spatial dimensions and market value as follows:

- **X (Feature / Independent Variable):** Represents the house size measured in square meters (m^2).
- **Y (Target / Dependent Variable):** Represents the house price, scaled in thousands of units.

The model assumes a linear functional mapping:

**y = f(X) = theta_1 X + theta_0**

## 2. Parameter Interpretation (theta_0, theta_1)

The vector theta defines the geometry of the regression line:

- **$\theta_1$ (Slope / Weight):** Quantifies the marginal increase in price per unit increase in size. Given the dataset trend where y = 3X, theta_1 =3. This implies that for every $1 m^2$ added to the house size, the valuation increases by 3,000
- **theta_0 (y-intercept / Bias):** Represents the theoretical baseline price when X = 0. In a physical context, this often accounts for fixed costs (e.g., land value) not captured by the square footage alone.

## 3. Predictive Inference for 70 m^2?

Based on the observed perfect linear distribution (50 to 150, 60 to 180), the underlying function is **y = 3X.**

- **Prediction:** For X = 70, y = 3(70) = 210.
- **Validity:** This prediction is highly reasonable as the training data exhibits zero variance (noise-free). If a trained model deviates significantly from **210**, it indicates

a failure in the optimization process (e.g., premature stopping or sub-optimal learning rate).

## 4. Sum of Squared Errors (SSE) Dynamics

The objective of Gradient Descent is to minimize the cost function (theta). In linear regression, this function is **convex**, resembling a bowl shape.

+1

1. **Gradient Calculation:** The algorithm calculates the partial derivative of the error with respect to each parameter.
2. **Update Rule:** theta_{next} = theta - alpha delta theta, where alpha is the learning rate.
3. **Optimization:** Because we move in the direction opposite to the gradient, we descend toward the global minimum, causing the **SSE to decrease monotonically** over time.

## 5. Definition of Convergence

Convergence is the state where the optimization algorithm has reached a stable minimum. It is characterized by:

- **Gradient Diminution:** $\nabla J(\theta) \approx 0$.
- **Parameter Stability:** Changes in $\theta$ between iterations become infinitesimal.
- **Loss Plateaus:** The SSE curve flattens, indicating no further improvement is possible.

## 6. Impact of Learning Rate (alpha)

The choice of hyperparameter $\alpha$ determines the stability of the descent:

| Learning Rate | Convergence Speed | Stability | Outcome |
|---|---|---|---|
| **Too Large** | High (initially) | Unstable | **Divergence:** The model overshoots the minimum, and the SSE increases. |

| | | | |
|---|---|---|---|
| **Optimal** | Balanced | Stable | **Global Minimum:** Efficiently reaches the lowest possible error. |
| **Too Small** | Extremely Low | High | **Inefficiency:** Requires excessive computational resources/time to converge. |

## 7. Importance of Feature Normalization

Normalization scales the input features (e.g., mapping m^2 to a range of [0, 1] or [-1, 1])

- **Contour Sphericity:** Without normalization, cost function contours can be elongated by ellipses, making the gradient descent path "zigzag."
- **Numerical Stability:** It prevents gradients from becoming too large (Exploding Gradients), allowing for a higher learning rate and faster convergence.

## 8. Real-World Considerations: Noise and $R^2$

In the provided example, the data is synthetic and perfectly linear, resulting in a **Coefficient of Determination (R^2) of 1.0**.

In real-world scenarios (e.g., 50 to 155, 60 to 175), R^2 will always be < 1. This is because:

- **Stochastic Noise:** Measurement errors and random market fluctuations.
- **Latent Variables:** Features not included in the model (e.g., location, age of the building).
- **Non-linearity:** Real relationships are rarely perfectly straight lines.