The SSE is a good choice for a few reasons. The square ensures the error is always positive and larger errors are penalized more than smaller errors. Also, it makes the math nice, always a plus.

Remember that the output of a neural network, the prediction, depends on the weights

$$\hat{y}_j^\mu = f\left(\sum_i w_{ij} x_i^\mu\right)$$

and accordingly the error depends on the weights

$$E = \frac{1}{2}\sum_\mu \sum_j \left[y_j^\mu - f\left(\sum_i w_{ij} x_i^\mu\right)\right]^2$$

We want the network's prediction error to be as small as possible and the weights are the knobs we can use to make that happen. Our goal is to find weights w_{ij} that minimize the squared error E. To do this with a neural network, typically you'd use **gradient descent**.

Enter Gradient Descent