# 1 Problems of Linear/Polynomial Regression

- The linear regression function $y' = F(x)$ maps $x$ to an approximate value $y'$ that should be as close as possible to the ground truth value $y$.
- Although this model works for a lot of senarios, it still cannot handle **complex non-linear relationships**.
- One of the problems of polynomial transofrmation is that it generates a huge number of new features (if you have 10,000 features and you want to do a polynomial transformation of length 2, the result would be 50 million!).
- Also, linear and polynomial regression has a very limited space transformation shapes.

## 1.1 Why linear and polynomial regression has limited transformations?

- The only shape this model can learn is through **powers of** $x$, that restricts the ability to learn from complex patterns that might be better modeled by other transformations (like sinusoids, exponentials, step functions, etc).
- Changing the coefficient of any term (e.g. $x^3$) affects **the entire curve globally**, not locally. This means that you can't fit small local variations easily, also overfitting or oscillations can happen very quickly with high-degree polynomials.
- **Global Influence:** Suppose you have the model $y = w_0 + w_1x_1 + w_2x_2 + w_3x_3$ and you decided to tweak $w_3$ (the coefficient of $x_3$) a little bit, the whole curve changes, not just around one point.
- Outside the range of data, polynomial functions grow quickly and unpredictably (especially higher-order ones). **So it lacks generalization power**.

Artificial Neural Networks (ANN) is a very powerful way to handle these limitations and perform non-linear transformations with a much richer transformation space.

# 2 Activation Functions

What is the point of activation functions?

- **Important:** A neural network with linear activativation is just a normal linear regression model.
- The solution is not generating polynomial features, if we did this then we didn't solve the original problem of polynomial regression.

- The solution is **Adding a differentiable non-linear function to the output of each neuron**. Now we forced the neuron output to get into a new non-linear transformation.
- The neural network now is really mapping the input to completely different space.
- Popular activation functions: Sigmoid, tanh, and Relu.
- **Note:** In linear regression, there is no need for an activation function, because we want to keep the aggregated sum linear. So we use a dummy act. function called $identity(x) = x$