1. Increase the Amount of Data

When a model has a small amount of data, it can easily memorize specific patterns or noise within that data, increasing the likelihood of overfitting. Therefore, as the amount of data increases, the model can learn the general patterns of the data and prevent overfitting.
If the amount of data is small, you can intentionally modify and add to the existing data to increase the amount of data, which is called **Data Augmentation.** Data augmentation is often used for images, where data is augmented by rotating images, adding noise, or modifying parts of the image. In the case of text data, methods such as back translation, which creates new data through translation and re-translation, can be used to augment the data.

2. Reduce Model Complexity

The complexity of an artificial neural network is determined by factors such as the number of hidden layers or the number of parameters. When overfitting is detected, one possible action for an artificial neural network model is to **reduce the complexity of the neural network**.
In artificial neural networks, the number of parameters in the model is sometimes referred to as the model's capacity.

3. Apply Weight Regularization

- **Overfitting:** Overfitting is like building a tower that's *too* perfect for a *specific* set of LEGO bricks. It's so specialized that if you get a slightly different set of bricks, the tower collapses.
- **Complex Model:** A complex model is like using *every* LEGO brick you have, even the tiny, weirdly shaped ones, to build the tower. It might look impressive, but it's also unstable.
- **Weight Regularization:** Weight regularization is like a rule that says, "Don't use *every* brick. Try to build the tower with the *least* number of bricks possible, and especially avoid the weirdly shaped ones."
- **L1 Regularization (Like a stricter rule):** It's like saying, "For *every* brick you use, you have to pay a small fine." This encourages you to use *only* the most important bricks and get rid of the useless ones completely (making their "weight" zero).
- **L2 Regularization (Like a gentler rule):** It's like saying, "For *every* brick you use, the fine is based on how *big* the brick is." This encourages you to use smaller, simpler bricks and avoid the big, complex ones.
-
- **Lambda (λ):** Lambda is like the *amount* of the fine. If the fine is *very* high (large λ), you'll be *very* careful about which bricks you use. If the fine is low (small λ), you can be a bit more generous with the bricks.