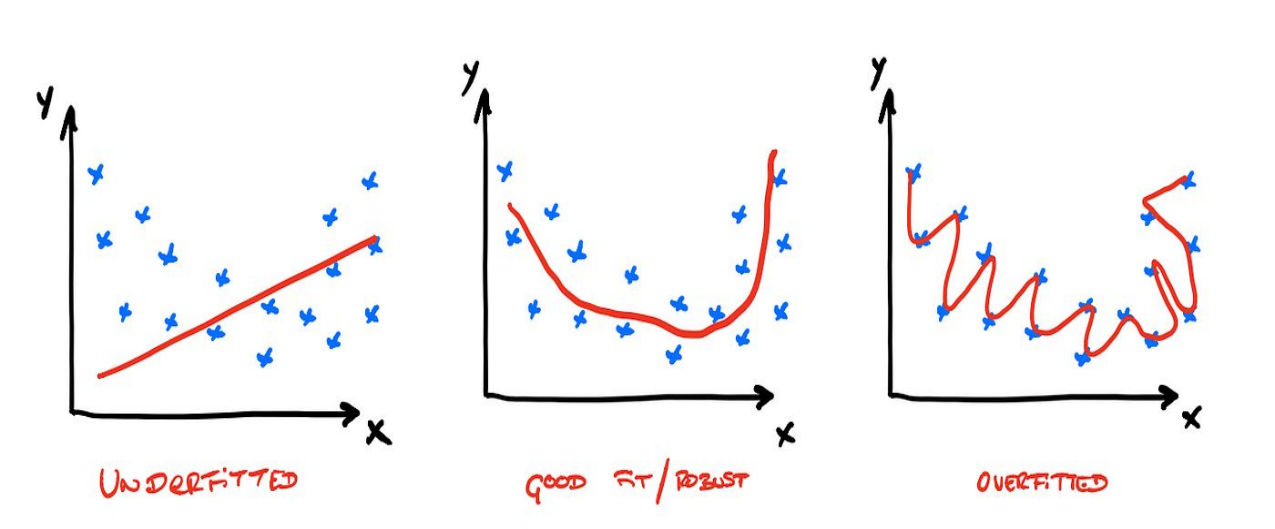
Injecting Noise

Regularization Technique

Tigist W. & Nahom S.



Model Performances

Intro

Noise injection is a **regularization technique** used to prevent overfitting by introducing randomness to training data or model parameters.

Use

- Prevents models from memorizing exact training data patterns.
- Improves model robustness against minor variations in data.
- Encourages better generalization to unseen data.

How?

Techniques:

1. **Noise in Inputs**
2. **Noise in Weights:**
3. **Noise in Outputs (Label Noise)**
4. **Noise in Activation**
5. **Noise in Gradient**

Techniques

Noise in Inputs:

- Adds random noise directly to the input features.
- Input noise is added directly to the input data during training.
- This can be done by adding Gaussian noise, uniform noise, or other types of random noise to the input features.

Example: In a dataset with temperatures as input, small random values (noise) are added to each temperature to create slightly perturbed versions of the data points.

Techniques

Noise in Weights:

- Weight noise is added to the model's weights during training.
- Weight noise helps regularize the model by preventing it from relying too much on any single weight or feature.

Example: In a neural network, noise can be added to weights during backpropagation to create weight updates that are less deterministic.

Techniques

Noise in Outputs:

- Adds random noise to the target labels during training.
- Simulates uncertainty in labels, making the model less sensitive to exact target values.

Example: If the true target is 50, adding noise might result in targets like 49.8 or 50.2.

Techniques

Noise in Activations:

- Activation noise is added to the model’s activations (i.e., the outputs of each layer) during training.

Example: In a neural network, if a neuron's activation is 0.8, noise might perturb it to 0.75 or 0.85.

Noise in Gradient:

- Adds noise to the gradient during backpropagation.

Example: Gradient noise can help smooth learning in deep neural networks by preventing abrupt changes in weight updates.

Best Use Case

1. **Small Datasets:**
   - Noise injection compensates for data scarcity by promoting generalization.
2. **Deep Learning Models:**
   - Techniques like dropout are particularly effective in preventing overfitting in neural networks.
3. **Noisy Environments:**
   - Improves robustness for models deployed in unpredictable real-world settings.

Limitations

1. **Excessive Noise:**
   - Adding too much noise can overwhelm the model and harm performance.
2. **Computational Overhead:**
   - Injecting noise increases the computational load during training.
3. **Limited Applicability:**
   - Not effective in systems requiring deterministic and precise predictions.

Bias-Variance Tradeoff

Impact of Noise Injection

- **Increases Bias:**
  - Forces the model to focus on broader patterns and ignore small details.
- **Decreases Variance:**
  - Reduces overfitting by preventing sensitivity to minor changes in training data.

Thank You