Theoretical Framework for In-Context Learning

Shrivani R, Shahana Devi V

October 2, 2025

1 Meta-Optimization Perspective of ICL

1.1 Notation

Let the K-shot prompt examples be:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_K, y_K)\}$$

and the test input be x_{test}. The model outputs y_{pred}.

1.2 Gradient-Descent Simulation

- Transformer updates its internal activations (attention) based on the prompt.
- This process is analogous to one step of gradient descent on the loss:

$$L(\theta) = \frac{1}{K} \sum_{i=1}^{K} \ell(y_i, f_\theta(x_i))$$

  where θ are model parameters.

1.3 Kernel Regression Perspective

- Transformer attention can be interpreted as a kernel weighting:

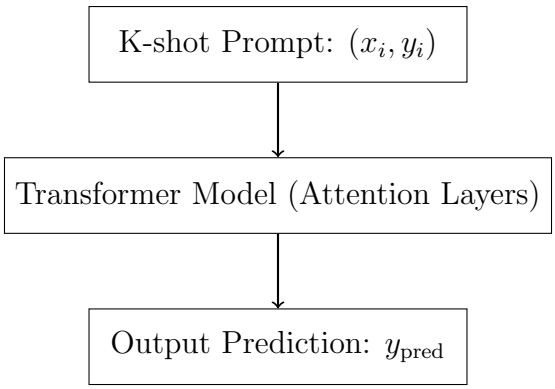
$$y_{\text{pred}} = \sum_{i=1}^{K} \alpha_i y_i, \quad \alpha_i \propto \text{similarity}(x_i, x_{\text{test}})$$

- Each example in the prompt contributes to the prediction according to its similarity to x_{test}.

1.4 Bayesian Model Averaging View

- Predictions can be interpreted as averaging over an ensemble of linear models implicitly encoded by the transformer's weights.

2 Schematic Diagram



3 Pseudocode

```
# Input: K-shot examples [(x1, y1), ..., (xK, yK)], x_test
# Output: y_pred

1. Construct prompt string from examples and x_test
2. Feed prompt into transformer model
3. Model generates output text
4. Parse numeric prediction from output
5. Return y_pred
```