Large Language Model (LLM) Explanation

Large Language Model (LLM) Explanation

Data Preprocessing (Pra-pemrosesan Data)

Tokenization (Tokenisasi)

Memecah teks menjadi token-token (kata atau sub-kata).

tokens = tokenize(text)

Embedding

Mengonversi token-token ke dalam representasi vektor berdimensi tinggi.

embeddings = EmbeddingLayer(tokens)

Large Language Model (LLM) Explanation

Model Architecture (Arsitektur Model)

Positional Encoding

Menambahkan informasi posisi ke embedding input.

$PE(pos, 2i) = \sin(pos / 10000^{(2i/d\_model)})$

$PE(pos, 2i+1) = \cos(pos / 10000^{(2i/d\_model)})$

Transformer Encoder Layer

Mengandung dua sub-lapisan: Self-Attention dan Feed-Forward Neural Network.

Self-Attention

Menghitung perhatian diri untuk setiap posisi dalam input sequence.

$Attention(Q, K, V) = softmax((QK^T) / \sqrt{d\_k}) V$

di mana Q adalah query, K adalah key, V adalah value, dan d_k adalah dimensi key.

Feed-Forward Neural Network

Lapisan jaringan saraf yang diterapkan setelah mekanisme perhatian.

$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$

Layer Normalization

Normalisasi di setiap sub-lapisan.

$LayerNorm(x) = (x - \mu) / \sqrt{\sigma^2 + \epsilon}$

Large Language Model (LLM) Explanation

Training (Pelatihan)

Loss Function (Fungsi Kehilangan)

Mengukur seberapa baik model membuat prediksi.

Cross-Entropy Loss (Untuk klasifikasi)

L(y, hat{y}) = -sum(y_i log(hat{y}_i))

di mana y adalah label sebenarnya dan hat{y} adalah prediksi model.

Optimization (Optimasi)

Memperbarui bobot model untuk meminimalkan fungsi kehilangan.

Gradient Descent

theta := theta - eta nabla_theta J(theta)

di mana theta adalah parameter model, eta adalah learning rate, dan J(theta) adalah fungsi kehilangan.

Adam (Adaptive Moment Estimation)

m_t = beta_1 m_{t-1} + (1 - beta_1) g_t

v_t = beta_2 v_{t-1} + (1 - beta_2) g_t^2

hat{m}_t = m_t / (1 - beta_1^t)

hat{v}_t = v_t / (1 - beta_2^t)

theta_t := theta_{t-1} - eta hat{m}_t / (sqrt(hat{v}_t) + epsilon)

Large Language Model (LLM) Explanation

Evaluation (Evaluasi)

Metrics (Metrik)

Mengukur kinerja model.

Accuracy (Akurasi)

Persentase prediksi benar dari total prediksi.

Accuracy = Correct Predictions / Total Predictions

Precision, Recall, F1-Score

Digunakan untuk evaluasi tugas klasifikasi.

Precision = True Positives / (True Positives + False Positives)

Recall = True Positives / (True Positives + False Negatives)

F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

Large Language Model (LLM) Explanation

Alur Keseluruhan

1. Data Preprocessing: Tokenisasi -> Embedding

2. Model Architecture: Positional Encoding -> Transformer Encoder Layer (Self-Attention -> Feed-Forward Neural Network -> Layer Normalization)

3. Training: Hitung Loss (Cross-Entropy) -> Optimasi (Gradient Descent/Adam)

4. Evaluation: Hitung Metrics (Accuracy, Precision, Recall, F1-Score)