DENOISING DIFFUSION MODEL CIFAR-10 DATASET

❖ HYPER PARAMETERS

1) IMAGE_SIZE:- The width or height of the image in pixels
2) Batch Size: - The number of batches to divide our train data into
3) NOISE_EMBEDDING_SIZE:- Size for our noise embedding
4) PLOT_DIFFUSION_STEPS:- The number of diffusion steps to go through in noising process.
5) EMA: - Exponential Moving Average rate
6) LEARNING_RATE:- The learning rate our model will be using in optimizer.
7) WEIGHT_DECAY:- The weight decay rate used in AdamW optimizer
8) EPOCHS: - The no of Iterations our model will be going through.

❖ Functions

1) display()
   Displays and optionally saves a set number of images.

2)preprocess()

Preprocesses image data by scaling pixel values to the range[0, 1].

3)Diffusion Schedule Functions:-

i)linear_diffusion_schedule()

ii)cosine_diffusion_schedule()

iii)offset_cosine_diffusion_schedule()

4)sinusoidal_embeddings()

Creates sinusoidal embedding for the noise variance

5)residual_block()

Defines a residual block to be used in unet model

6)down_block()

Defines a down-sampling block to be used in unet model

7)up_block()

Defines an up-sampling block for the unet model.

8)DiffusionModel

- Purpose: Implements the Diffusion Model class with necessary methods for training, testing, and generating images.
- Methods:
  - __init__(): Initializes the model.
  - compile(): Compiles the model with necessary configurations.
  - metrics(): Returns the metrics used.
  - denormalize(): Denormalizes the images.
  - denoise(): Denoises the images during the diffusion process.
  - reverse_diffusion(): Performs the reverse diffusion process.
  - generate(): Generates images using the reverse diffusion process.
  - train_step(): Defines the training step.
  - test_step(): Defines the testing step.

9)Image Generator Callback Class

Custom callback to generate and display images at the end of each epoch during training

10)spherical_interpolation()

Performs spherical interpolation between 2 points

❖ VARIABLES

i)display Function

- images: Array of images to be displayed.
- n: Number of images to display.
- size: Size of the display figure.
- cmap: Color map for displaying images.
- as_type: Data type for images.
- save_to: File path to save the displayed images.

ii)preprocess Function

- imgs: Array of images to preprocess.

iii)linear_diffusion_schedule Function

- diffusion_times: Array of diffusion times.
- min_rate: Minimum diffusion rate (0.0001).
- max_rate: Maximum diffusion rate (0.02).
- betas: Array of beta values calculated based on diffusion times.
- alphas: Array of alpha values calculated from betas.
- alpha_bars: Cumulative product of alphas.
- signal_rates: Square root of alpha_bars.
- noise_rates: Square root of (1 - alpha_bars).

iv)cosine_diffusion_schedule Function

- diffusion_times: Array of diffusion times.
- signal_rates: Cosine of diffusion times scaled by π/2.
- noise_rates: Sine of diffusion times scaled by π/2.

v)offset_cosine_diffusion_schedule Function

- diffusion_times: Array of diffusion times.
- min_signal_rate: Minimum signal rate (0.02).
- max_signal_rate: Maximum signal rate (0.94).
- start_angle: Arccosine of max_signal_rate.
- end_angle: Arccosine of min_signal_rate.
- diffusion_angles: Interpolated diffusion angles.
- signal_rates: Cosine of diffusion_angles.
- noise_rates: Sine of diffusion_angles.

vi)sinusoidal_embeddings Function

- x: Input tensor.
- frequencies: Array of frequencies for the embeddings.
- angular_speeds: Angular speeds calculated from frequencies.
- embeddings: Sinusoidal embeddings created by concatenating sine and cosine of angular speeds times x.

vii)residual_block Function

- width: Number of filters in the convolution layers.
- x: Input tensor.
- input_width: Width of the input tensor.
- residual: Residual tensor (identity connection or convolution result).
- Block output: Tensor after passing through batch normalization and convolution layers.

viii)down_block Function

- width: Number of filters in the convolution layers.
- block_depth: Depth of the block.
- x: Input tensor and skip connections.
- skips: List of skip connections.

ix)up_block Function

- width: Number of filters in the convolution layers.
- block_depth: Depth of the block.
- x: Input tensor and skip connections.
- skips: List of skip connections.

x)DiffusionModel Class

- noise_loss_tracker: Tracker for noise loss.
- normalizer: Normalization layer.
- network: UNet model.
- ema_network: Exponential moving average of the network.
- diffusion_schedule: Function to compute diffusion schedule.
- initial_noise: Initial noise for the reverse diffusion process.
- diffusion_steps: Number of steps for the diffusion process.
- current_images: Current images in the reverse diffusion process.
- diffusion_times: Array of diffusion times.
- noise_rates: Noise rates from the diffusion schedule.
- signal_rates: Signal rates from the diffusion schedule.
- pred_noises: Predicted noises from the network.
- pred_images: Predicted images from the network.
- next_diffusion_times: Next set of diffusion times.
- next_noise_rates: Next set of noise rates.
- next_signal_rates: Next set of signal rates.
- generated_images: Generated images.
- images: Array of images for training/testing.
- noises: Array of random noise.
- noisy_images: Array of noisy images.
- noise_loss: Calculated noise loss.
- gradients: Gradients for the trainable weights.

xi)ImageGenerator Callback Class

- num_imgs: Number of images to generate.
- epoch: Current epoch during training.
- logs: Training logs.

xii)spherical_interpolation Function

- a: Start point.
- b: End point.
- t: Interpolation parameter.
- Interpolated value: Result of the spherical interpolation between a and b using t.