EEP 596: LLMs: From Transformers to GPT || Lecture 15

Dr. Karthik Mohan

Univ. of Washington, Seattle

February 27, 2024

Deep Learning and Transformers References

Deep Learning

Great reference for the theory and fundamentals of deep learning: Book by Goodfellow and Bengio et al Bengio et al

Deep Learning History

Embeddings

SBERT and its usefulness

SBert Details

Instacart Search Relevance

Instacart Auto-Complete

Attention

Illustration of attention mechanism

Generative AI References

Prompt Engineering

Prompt Design and Engineering: Introduction and Advanced Methods

Retrieval Augmented Generation (RAG)

Toolformer

RAG Toolformer explained

Misc GenAI references

Time-Aware Language Models as Temporal Knowledge Bases

Generative AI references

Stable Diffusion

The Original Stable Diffusion Paper

Reference: CLIP

Diffusion Explainer: Visual Explanation for Text-to-image Stable Diffusion

Diffusion Explainer Demo

The Illustrated Stable Diffusion

Unet

Previous Lecture

- Stable Diffusion Architecture
- De-noising AutoEncoders in Stable Diffusion

This Lecture

- Stable Diffusion Recap
- Unet Architecture
- Diffusion Explainer Demo
- Diffusion Notebook and ICE

Stable Diffusion Explained

- Based on the concept of "de-noising auto encoders" and the use of text prompt to *guide the de-noising*
- Stable diffusion is also trained to successfully de-noise and increase the resolution of the image using text guidance

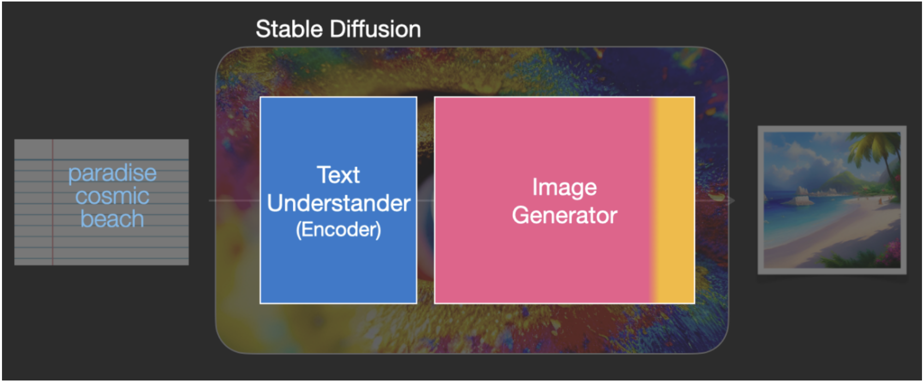
Stable Diffusion High-Level

Reference: The Illustrated Stable Diffusion

Stable Diffusion High-Level

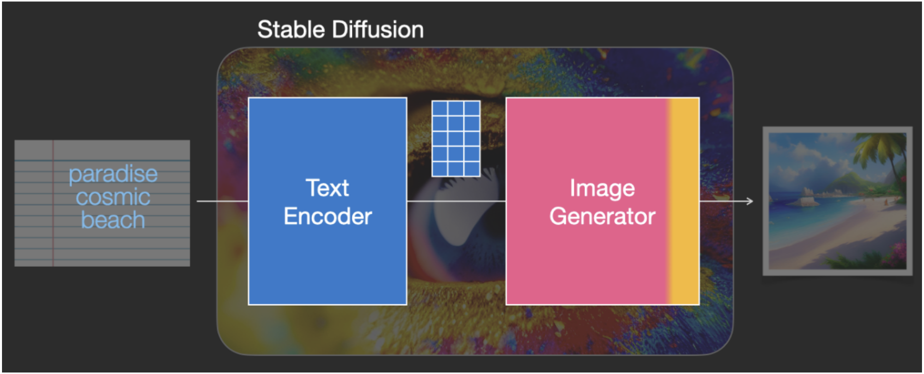
Reference: The Illustrated Stable Diffusion

Stable Diffusion Components



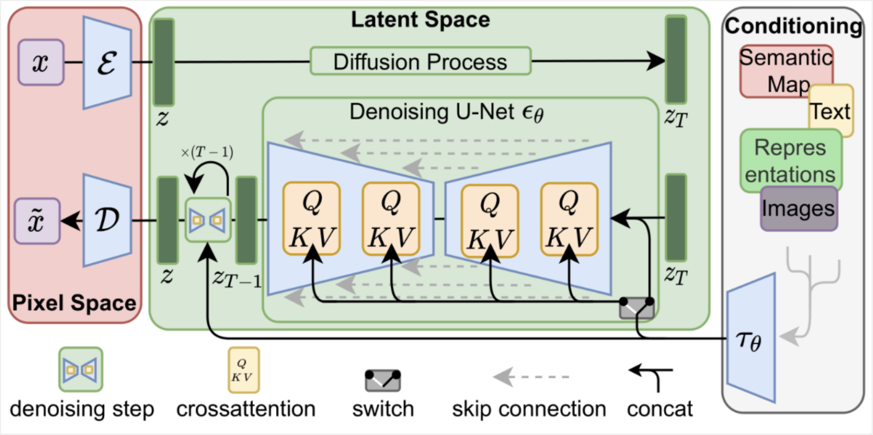
Reference: The Illustrated Stable Diffusion

Stable Diffusion Components

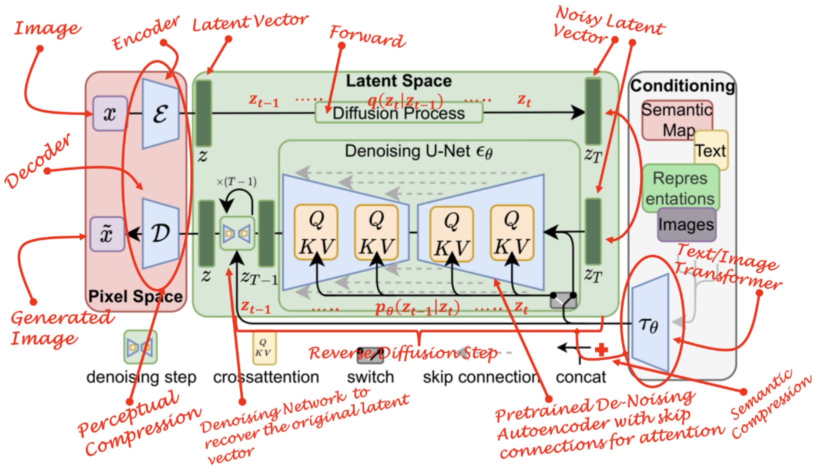


Reference: The Illustrated Stable Diffusion

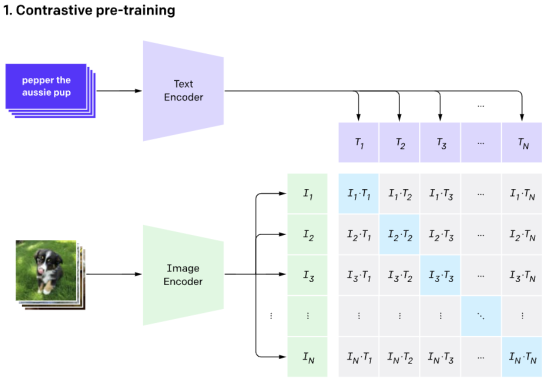
Stable Diffusion Full Architecture



Stable Diffusion Full Architecture



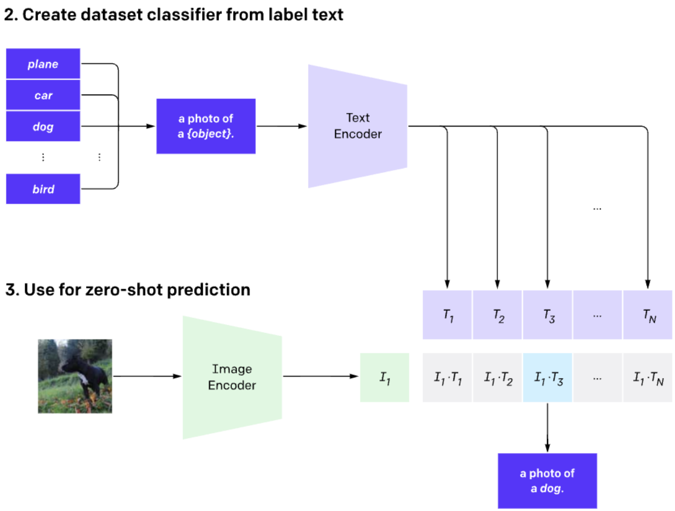
Clip Pre-Training Architecture



CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as "a photo of a dog" and predict the class of the caption CLIP estimates best pairs with a given image.

Reference: CLIP

Clip Zero-Shot Prediction Process



Reference: CLIP

Clip Implementation Pseudo-code

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

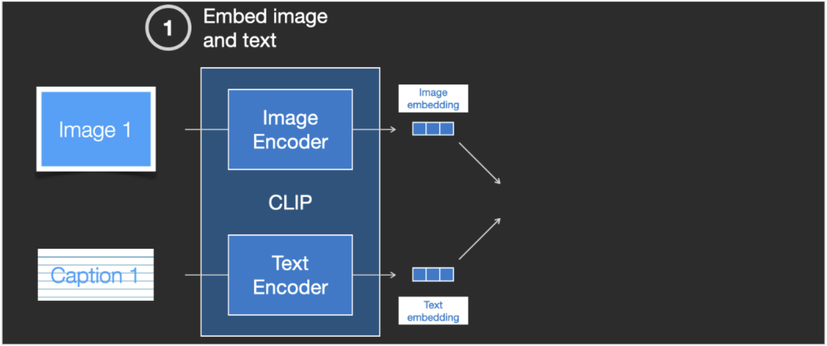
Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

Reference: CLIP

Clip Training Examples

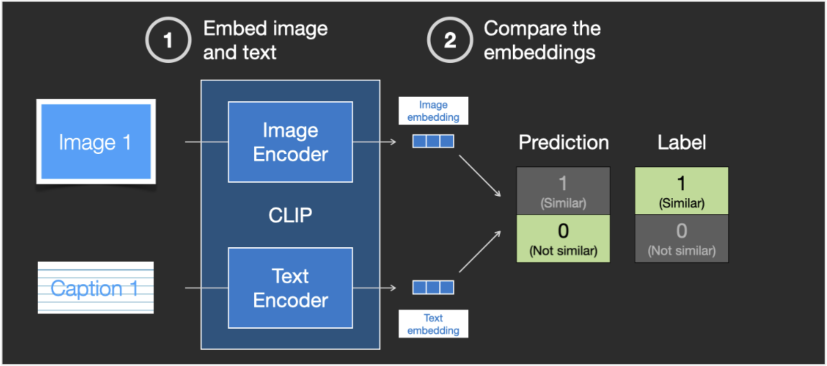
Question: How many examples used for Training? Reference: The Illustrated Stable Diffusion

Clip Training Process



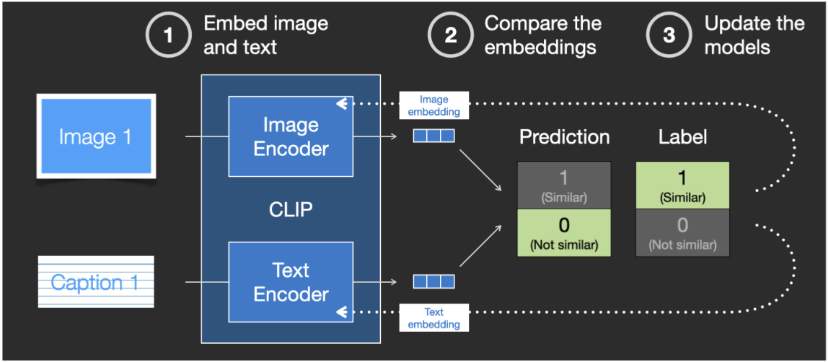
Reference: The Illustrated Stable Diffusion

Clip Training Process



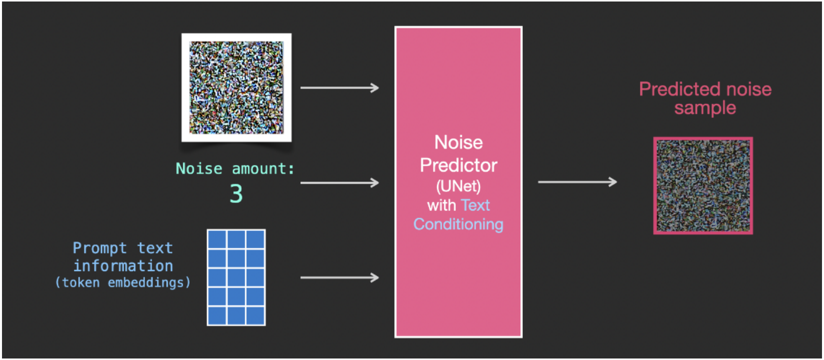
Reference: The Illustrated Stable Diffusion

Clip Training Process



Reference: The Illustrated Stable Diffusion

Image Generation Process

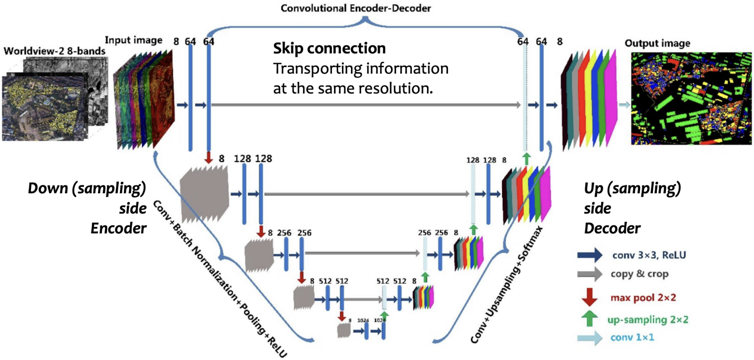


Reference: The Illustrated Stable Diffusion

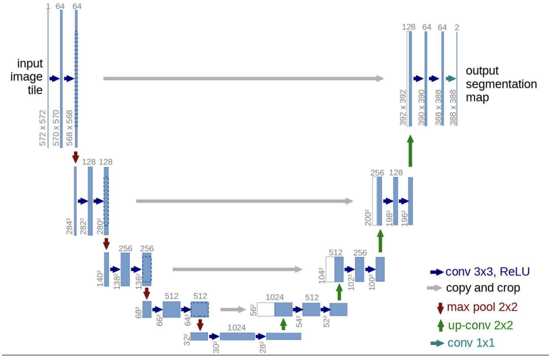
Image Generation: Training Data

Reference: The Illustrated Stable Diffusion

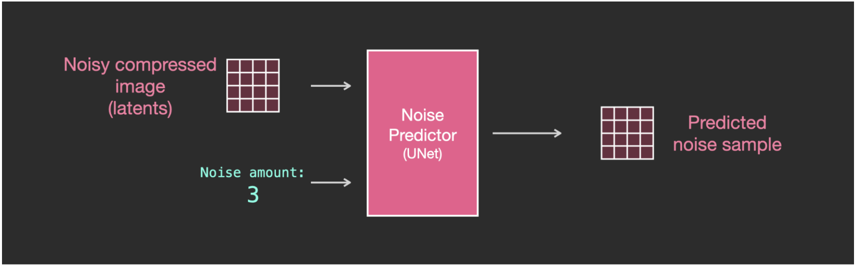
Unet Architecture



Unet Architecture

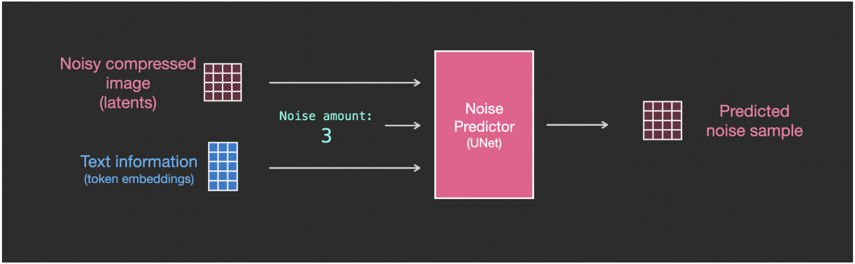


Unet Predictor (Without Text)



Reference: The Illustrated Stable Diffusion

Unet Predictor (With Text)



Reference: The Illustrated Stable Diffusion

Generating Video from Text

Video Diffusion Models

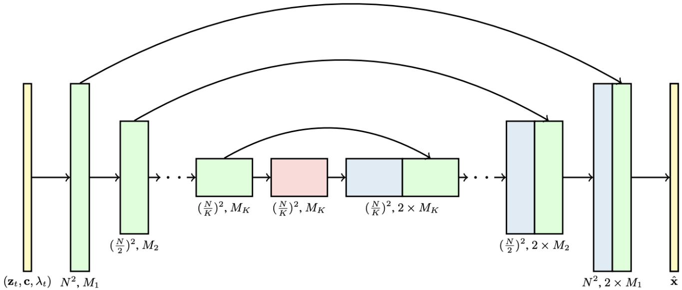


Figure 1: The 3D U-Net architecture for $\hat{\mathbf{x}}_\theta$ in the diffusion model. Each block represents a 4D tensor with axes labeled as frames $\times$ height $\times$ width $\times$ channels, processed in a space-time factorized manner as described in Section 3. The input is a noisy video $\mathbf{z}_t$, conditioning $\mathbf{c}$, and the log SNR λ_t. The downsampling/upsampling blocks adjust the spatial input resolution height $\times$ width by a factor of 2 through each of the K blocks. The channel counts are specified using channel multipliers M_1, M_2, ..., M_K, and the upsampling pass has concatenation skip connections to the downsampling pass.