How does stable diffusion work?

...

Jax Bulbrook

Digit drawing - Training phase

Training Dataset- a bunch of images along with what they are

Repeat until the loss function is small

AI

Update the AI based on the gradients of that

Compare to original noise

Pred - actual

$$\frac{\sum(\hat{n}-n)^2}{totalInputs}$$

Mean squared error

Noise generator

+

=

Noisy image

3

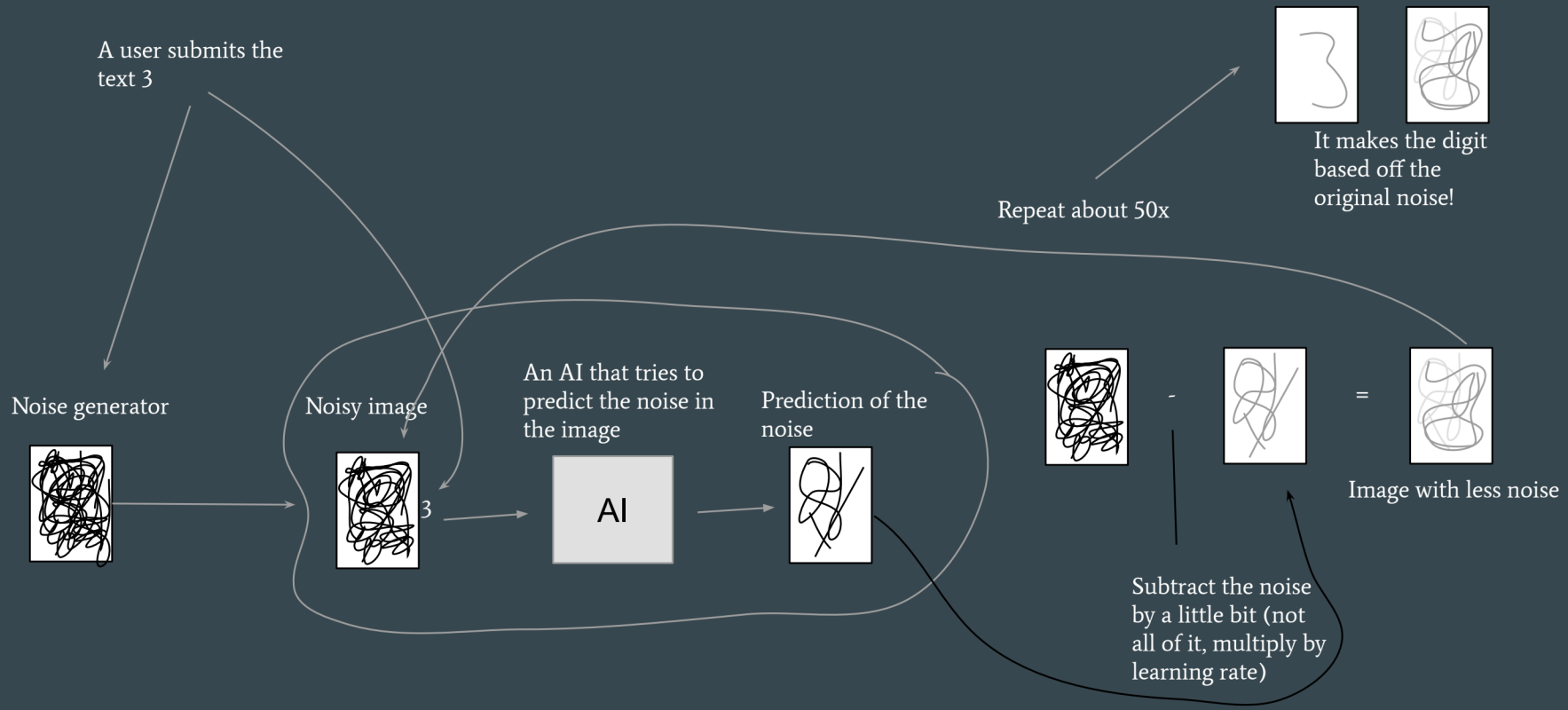
An AI that tries to predict the noise in the image

AI

You also tell the AI what it is

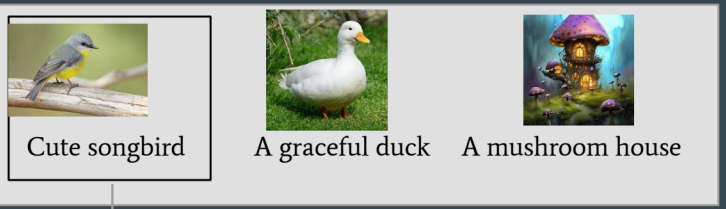
Prediction of the noise

Digit drawing - Production phase



Stable diffusion - understanding phase (CLIP encoder)

Training Dataset- a bunch of images along with what they are, programmatically from alt tag of images found online



Repeat millions of times to train the CLIP encoder - results in the model understanding what the text actually means

Once trained, the user can pass in a text tag and it will give a list of numbers (embedding) that looks like what the image should look like to the AI (next slide)

Cute songbird

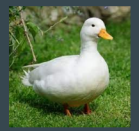
Image encoder (AI)

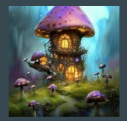
Text encoder (AI)

Converts to numbers

Converts to numbers

Embedding matrix (the input is multiplied by every output of opposite type)





Cute songbird

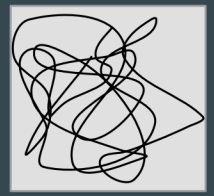
A graceful duck

A mushroom house

Loss = Add the diagonal and subtract by the rest

Stable Diffusion - Image generation phase (UNET)

The UNET is trained just like the digit creator, but instead of using a description of “3” or “9”, it would be the list of numbers



List of numbers (embedding) that should look like an image, came from user

Random noise generator

01010010010101.....
embedding

AI

Predicts where noise is, subtract that times learning rate, repeat, etc x50

IMAGE