Data Generation

tokenizer = Tokenizer (num_words = 10^{100})

tokenizer.fit_on_texts (titles + bodies)

title_sequences = tokenizer.
                texts_to_sequences
              ↗ (titles)

Assigned tokens to
sentences
                    makes the length of
              ↙   two sentence equal
title_data = pad_sequences
              (title_sequences, texts
              maxlen = 5)
            ↑ max_number of 0 to add
pad_sequences will add
  0 to 5 letter word sentence
  5 (words) + 0 = 6

One hot encoding
        ↓
encodes data in binary

Highlight 1 (Abhijeet) { Activation layers to be done next time }

Microsoft acquires Neon

~~~~~~~~~~~~~~~~~~~~~~~~~~

Sequential layer :-
When not to use :-
1) Multiple I/O (Eg concatinating embedding)
2) Layer with multi I/O
3) Layer Sharing
4) Special Case
   (residual connection)

Feature extraction :-

Makes f
Extracting and storing features of
1st layer.
If residual netword then the
output of 1st layer added to
last layer.

Hugging face → can get pre built
                      models and
                          datasets

Transfer learning → fine tune
                          pre-trained models
↱
freeze base model weights and add
your own.

Highlight-2

Cognify reprograming prisoners mind-?????
↗ bruh

~~~~~~~~~~~~~~~~~~~~~~

Subclassing:-

Variational Auto-Encoder:- CVAE)
Can include custom math.
z_mean - Central value in data
z_log_var - ~~data towards left or~~
data distribution towards left or right.
and by how much.
Sampling
→ A Introduces noise to data
$$z\_mean + ops.\ exp\ (0.5 * z\_log\_var) * epilson$$

Encoder
- Uses Gaussian distribution
- latent vector for certain message

~~~~~~~~~~~

error solving - of source code
seed generator

Decoder
makes image back into similar
image similar to origin original
Keeps main feature intact.

~~~~~~~~~~~

Highlight-3
google invests in AI-chip