• How the inputs and outputs are encoded

Inputs One-hot encoding for every character

Assuming inputs contain no capital letters

the length of the vector is 26 [1×26]

E.g. "a" → [1, 0, 0 ..... 0] (25×1)

"b" → [0, 1, 0 ..... 0]

Outputs One-hot encoding for every possible labels

Since "O" will not be used, there are 6 labels

"B-pre", "I-pre", "B-root", "I-root", "B-suf", "I-suf"

the length of the vector is 6

E.g. "B-pre" → [1, 0, 0, 0, 0, 0]

"I-pre" → [0, 1, 0, 0, 0, 0]

• The Loss function Used

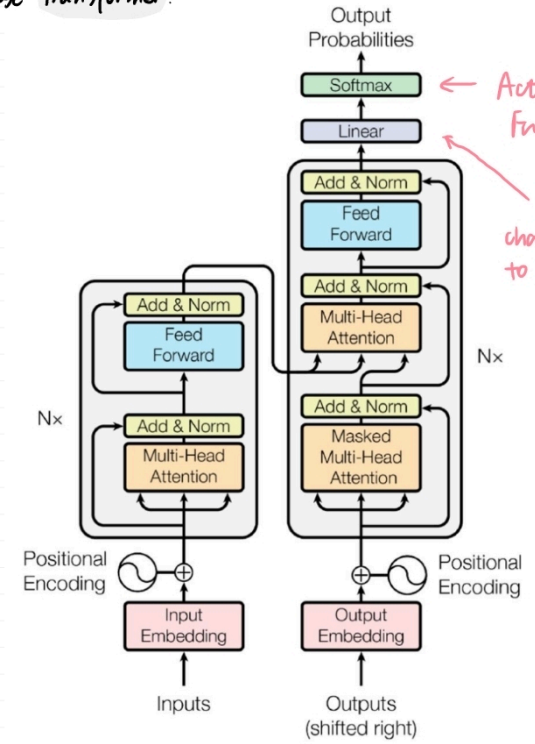
Cross Entropy

We can use `torch.nn.CrossEntropyLoss(weight=..., size_average=.....)` in PyTorch

CLASS torch.nn.CrossEntropyLoss(*weight=None, size_average=None, ignore_index=-100, reduce=None, reduction='mean', label_smoothing=0.0*) [SOURCE]

• The structure of the Network

Since we need information non-local from across the entire words to determine the correct label for each character, I'd like to use Transformer.



← Activation Function

[25 6× 6]

change dim_feedforward to the same size of output "6"

Class `torch.nn.Transformer(d_model=26, nhead=8, num_encoder_layers=6, num_decoder_layers=6, dim_feedforward=256)`