Today’s Tutorial

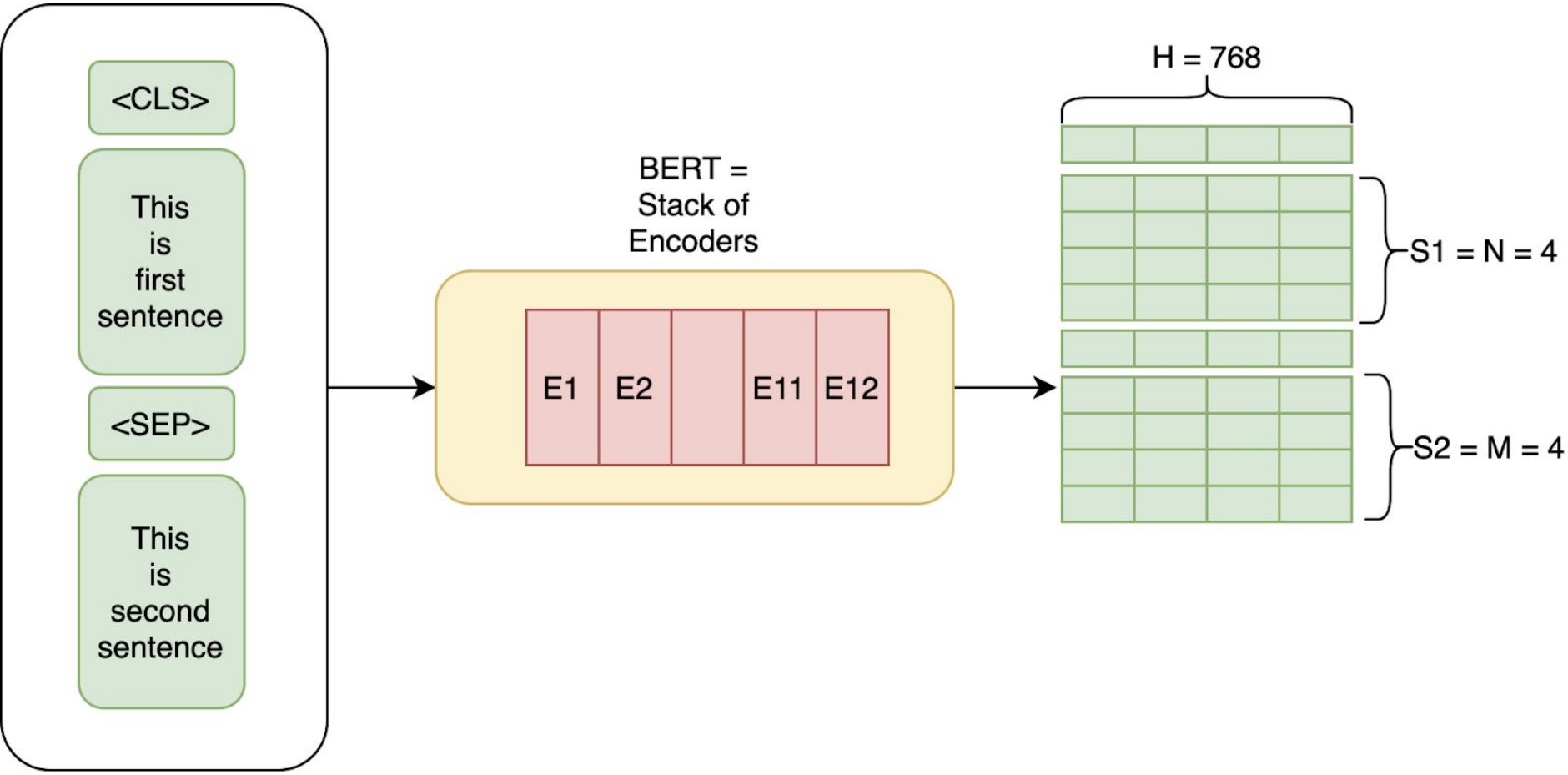
- Infrastructure
- Sentence Embeddings

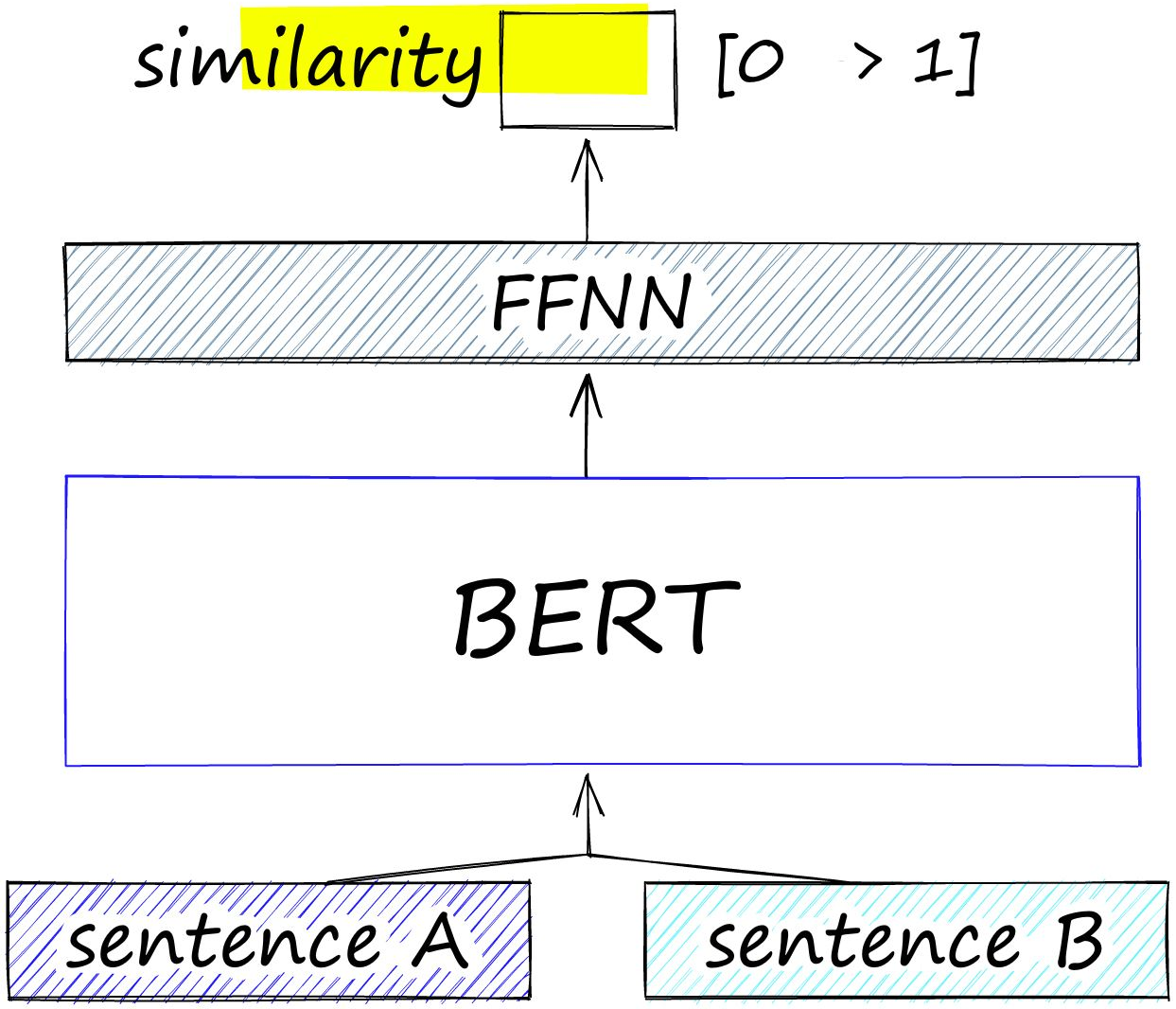
We got a server for you…

- Please copy your ssh keys into the .ssh directory
- We will go through the procedure together
- Bonus: Jupyter Notebooks

Sentence Embeddings

- Transformers use tokenized embeddings
- How do get meaningful sentence representations?
- Simply aggregating tokenized embeddings is not powerful enough


<CLS>
This is first sentence
<SEP>
This is second sentence
BERT = Stack of Encoders
E1
E2
E11
E12
H = 768
S1 = N = 4
S2 = M = 4

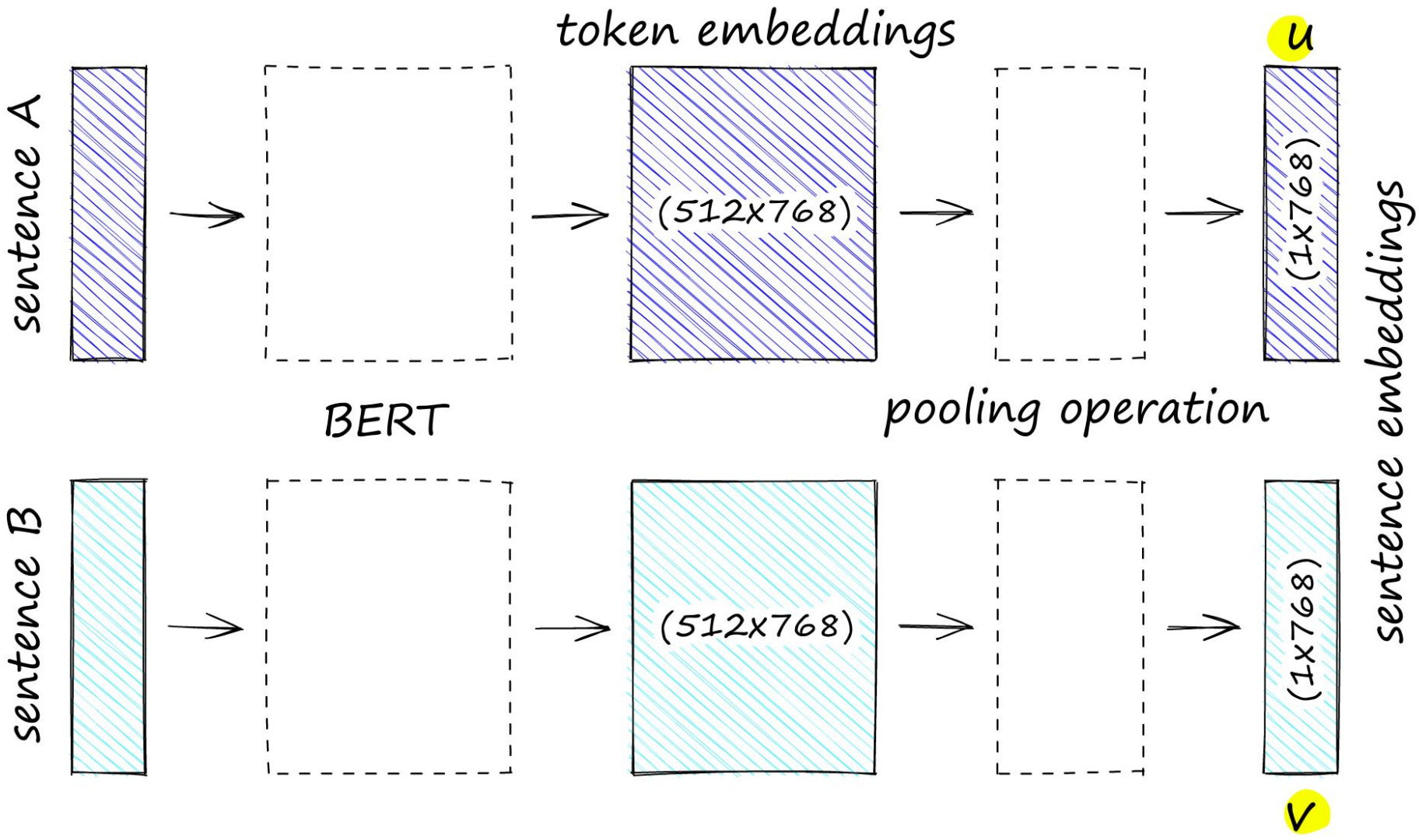

similarity
[0 > 1]
FFNN
BERT
sentence A
sentence B

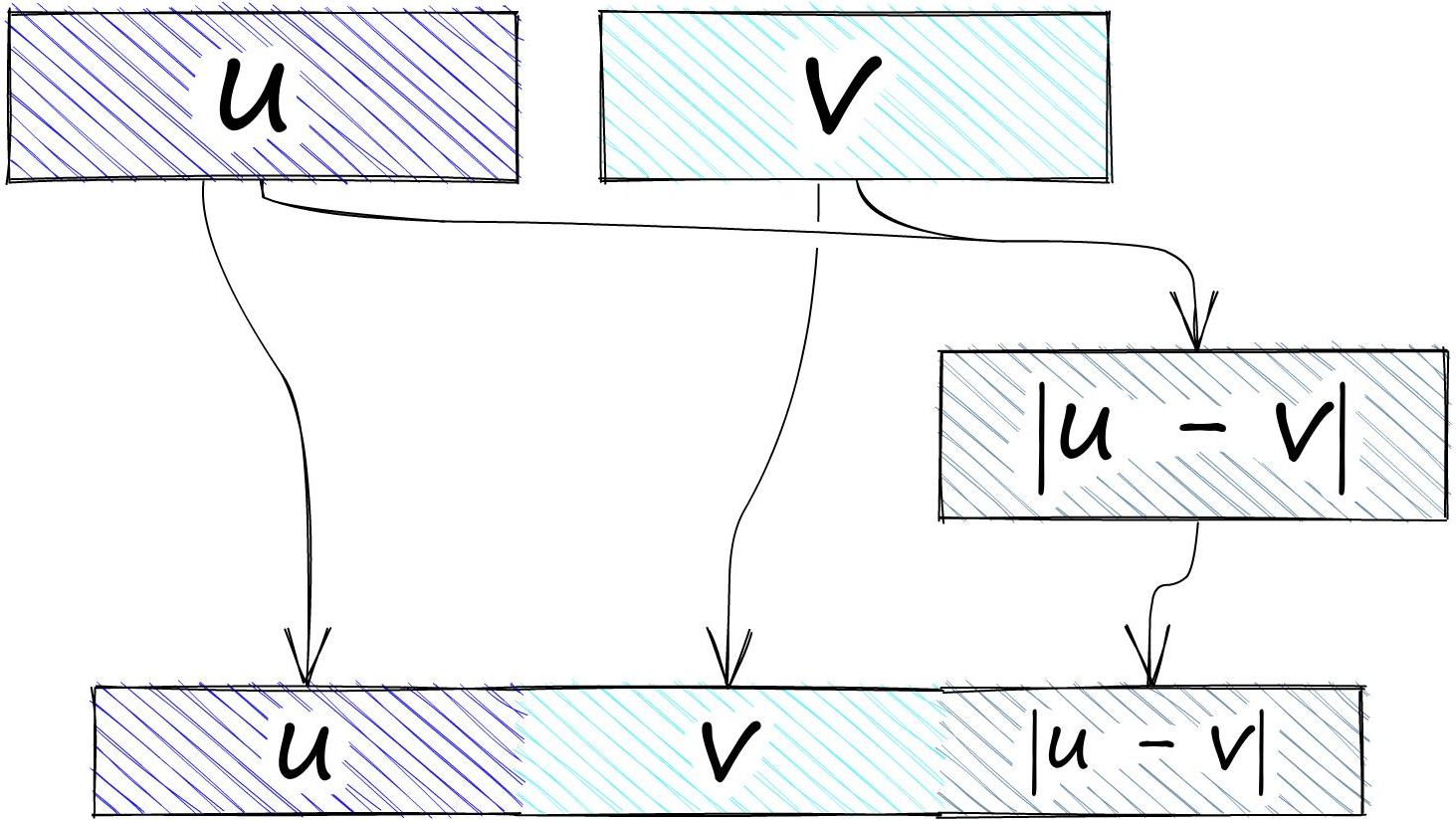
Problem with this approach

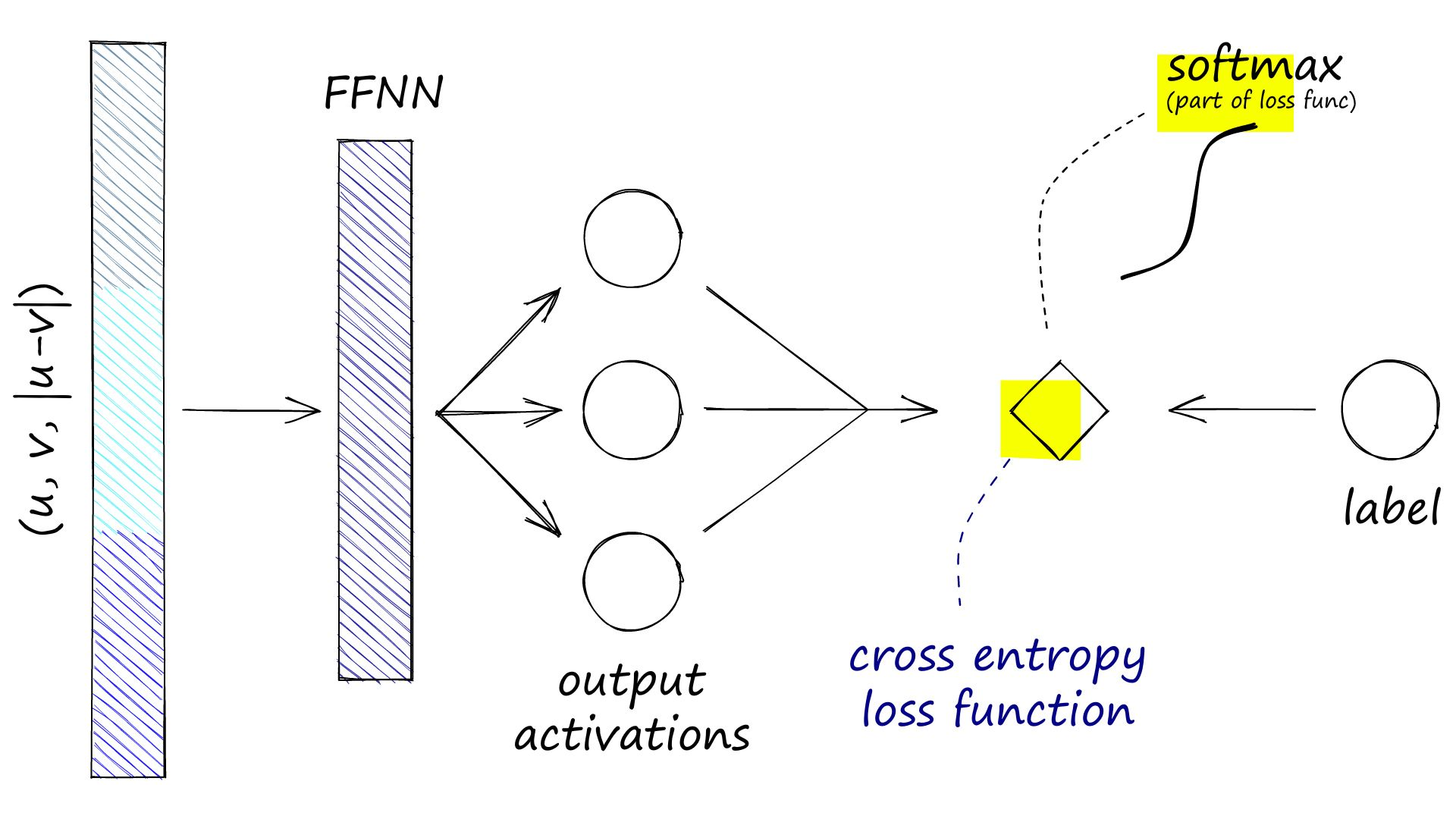
- Computing similarity this way is actually more performant than current approaches
- 100k Observations → 500 Mio Comparisons
- We could also use the CLS token, which serves as a sequence representation for classification tasks, but performance is usually bad
- Solution: SBERT
- Actually produces useful sentence embeddings

NLI Dataset

$\mathbf{P}^a$	A senior is waiting at the window of a restaurant that serves sandwiches.	Relationship
$\mathbf{H}^b$	A person waits to be served his food.	Entailment
	A man is looking to order a grilled cheese sandwich.	Neutral
	A man is waiting in line for the bus.	Contradiction
$^a\mathbf{P}$, Premise. $^b\mathbf{H}$, Hypothesis.		


token embeddings
u
sentence A
(512x768)
(1x768)
BERT
pooling operation
sentence embeddings
sentence B
(512x768)
(1x768)
v


u
v
|u − v|
u
v
|u − v|


FFNN
softmax
(part of loss func)
(u, v, |u−v|)
output
activations
cross entropy
loss function
label