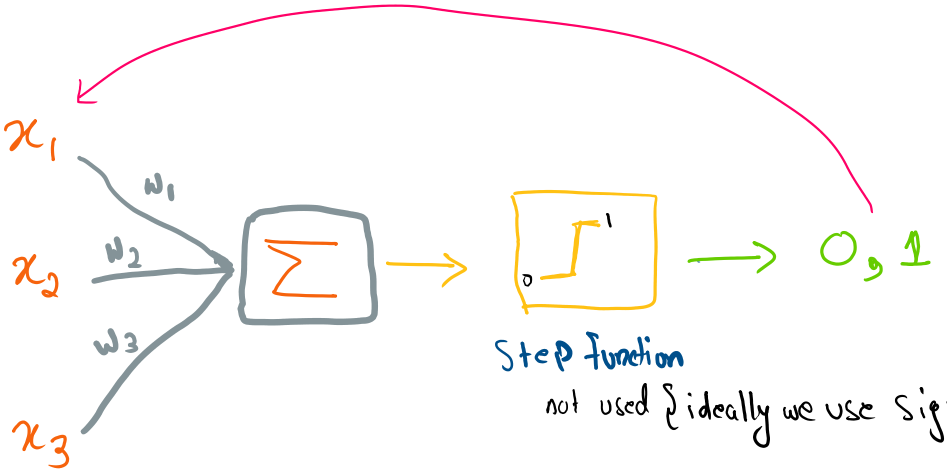
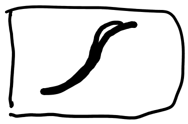
Neural Network Basics

07 May 2024 17:48



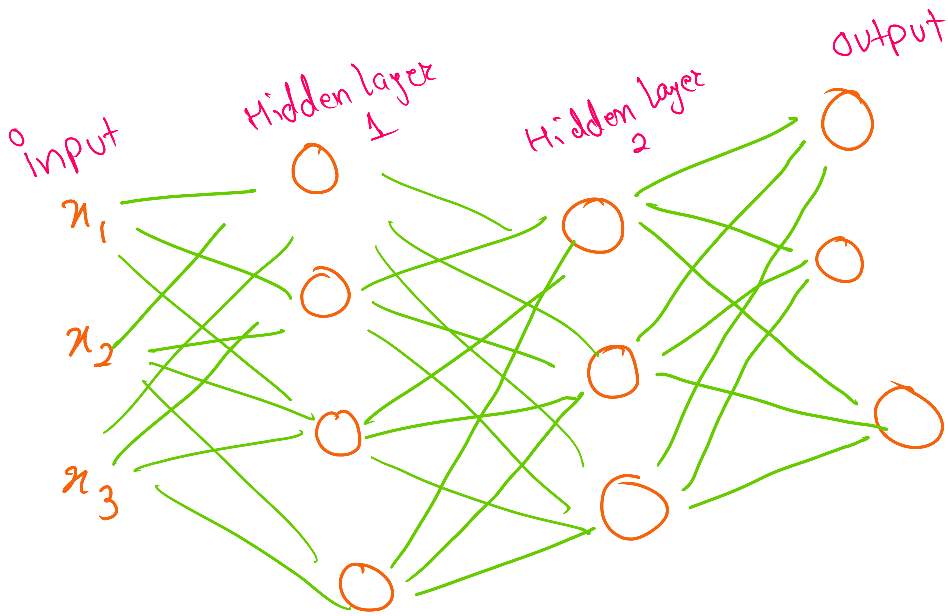
not used {ideally we use sigmoid function}



Activation function

$$y = f(\vec{x} \ \vec{w})$$

output



→ mnist Dataset (fashion| clothing)

data is in a form of (28x28)

[[0.1, 0.2, ... x28],

⋮

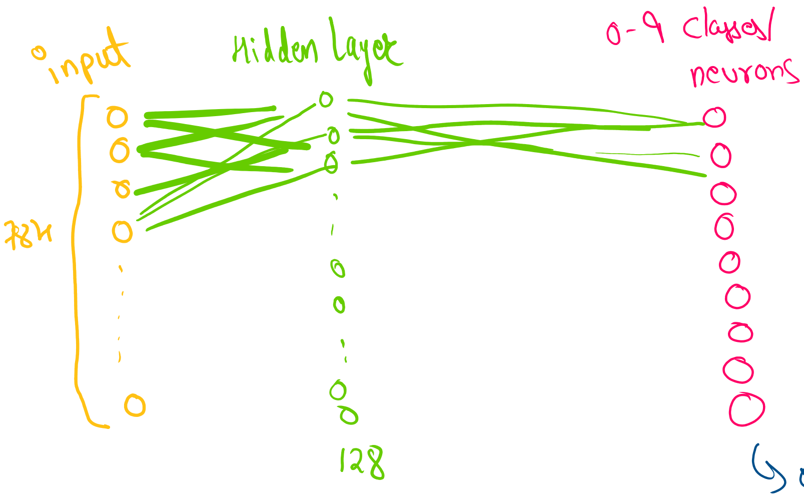
x28

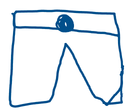
[.3, .4 ...... x28]]

this can't of data can't be used as it is to the neuron.

∴ we will flatten the data

(28x28) → [784]





↳ each class denotes a type of clothing apperial

In the case of a 28x28 grayscale image, there would be 28x28=784 neurons in the input layer. Each neuron would receive the pixel intensity value of a corresponding pixel in the image as its input.

Passing the entire image through a single neuron would lose spatial information and wouldn't effectively capture the complex patterns present in the image. Instead, neural networks use multiple layers of neurons, each layer learning different features of the input data, to extract relevant information and make predictions.

For test_img with the highest % with some particular class is likely the clothing.

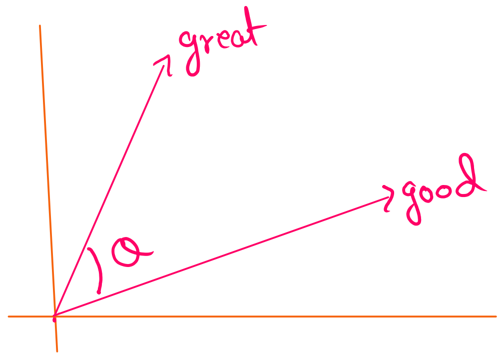
→ Text classification (Imdb Movie Review dataset)

① Have ② a ③ good ④ day

① Have ② a ⑤ great ④ day

[1, 2, 3, 4]

[1, 2, 5, 4]



Defines the dimension

→ it will creat 10,000 word vector

```
# Model
model = keras.Sequential()
model.add(keras.layers.Embedding(10000,16))
model.add(keras.layers.GlobalAveragePooling1D())
model.add(keras.layers.Dense(16,activation = 'relu'))
model.add(keras.layers.Dense(1,activation='sigmoid'))

model.summary()  # prints a summary of the model
```

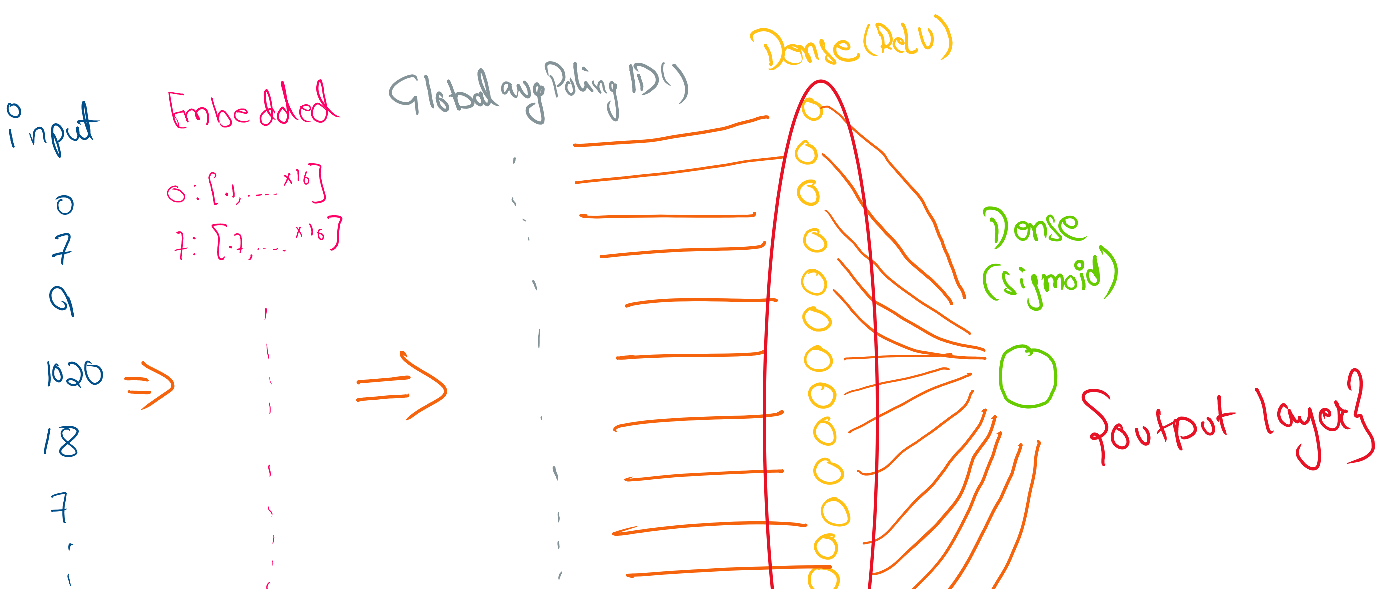
This is where a word embedding layer comes in. We want a way to determine not only the contents of a sentence but the **context** of the sentence. A word embedding layer will attempt to determine the meaning of each word in the sentence by mapping each word to a position in vector space.

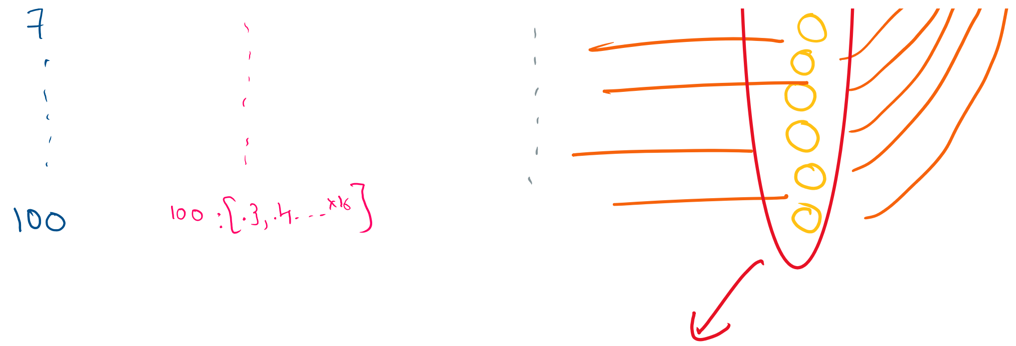
An example of something we'd hope an embedding layer would do for us: Maybe "good", "great", "fantastic" and "awesome" are placed close to each other and words like "bad", "horrible", "sucks" are placed close together. We'd also hope that these groupings of words are placed far apart from each other representing that they have very different meanings.

→ Scales down the data's dimension to make it easy for Computation

Dense Layers

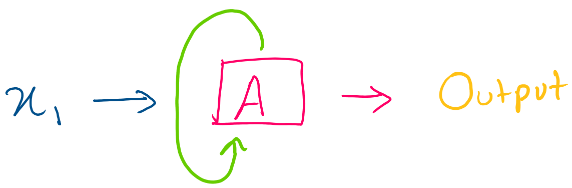
The last two layers in our network are dense fully connected layers. The output layer is one neuron that uses the sigmoid function to get a value between a 0 and a 1 which will represent the likelihood of the review being positive or negative. The layer before that contains 16 neurons with a relu activation function designed to find patterns between different words present in the review.



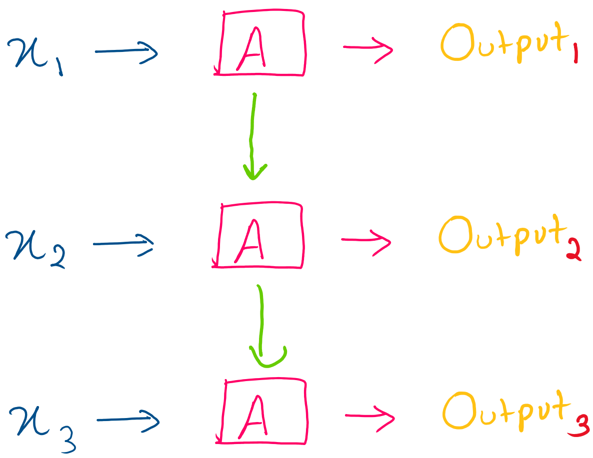


tries to find Pattern of words & tries to classify into a positive or negative review

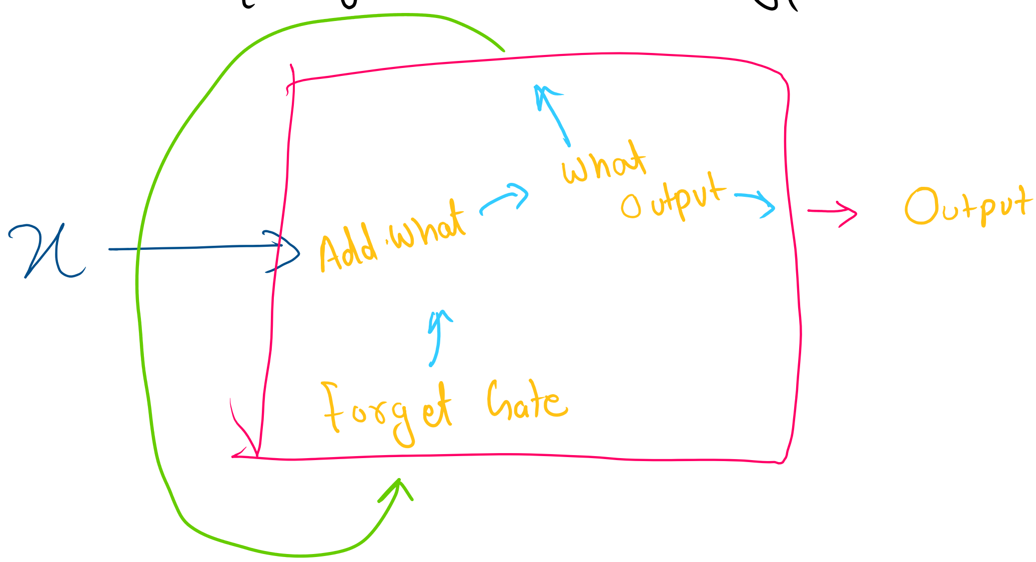
→ Recurrent Neural Network



output of one Layer becomes input for another + x



LSTM { Long Short Term Memory }



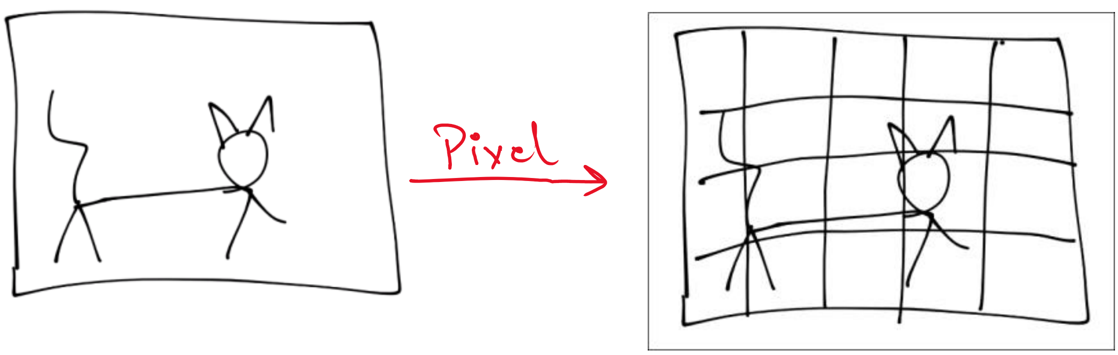
Recurring data goes through what is referred to as the Keep Gate or Forget Gate, basically which decides what to keep and what to remove from the recurring data. From here, we get to the new input data, determining what new to add from it, then, finally, we decide what our new output will be.

If you would like more information on the Recurrent Neural Network and the LSTM, check out Understanding LSTM Networks.

→ Convulational Neural Network

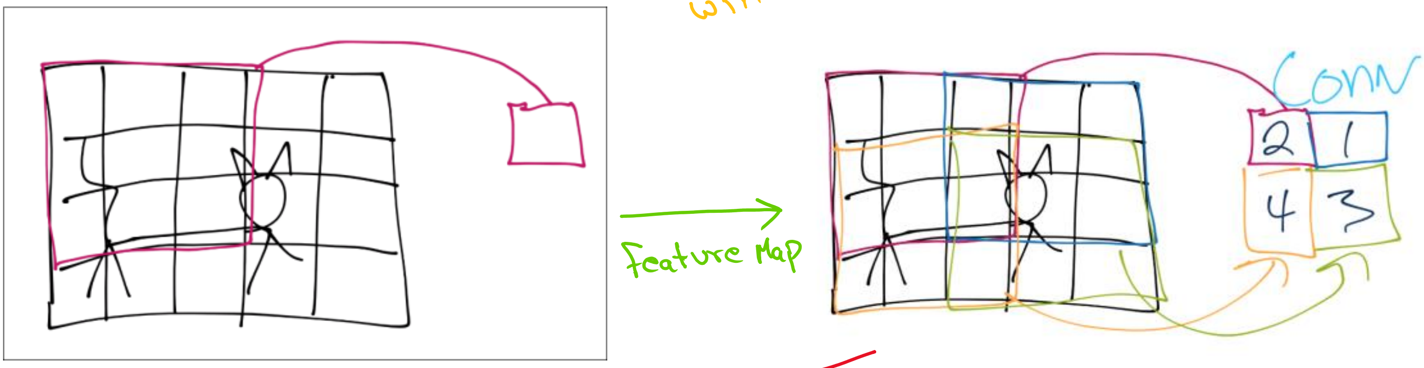
Basic Structure:

Convulation → Pooling → Convulation → Pooling → fully Connected → Output



Perform convolution on the low & find features

Perform Convolution on the window & find features

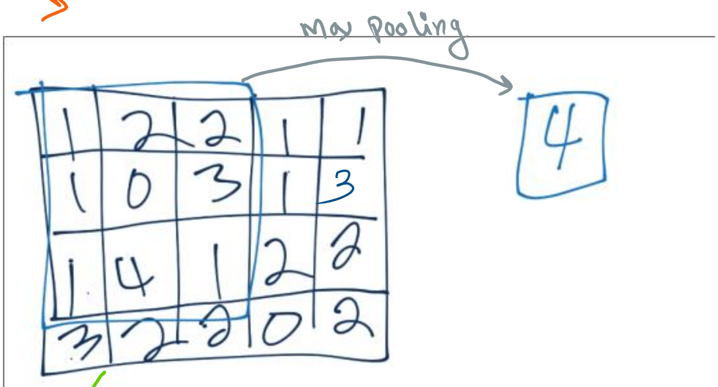


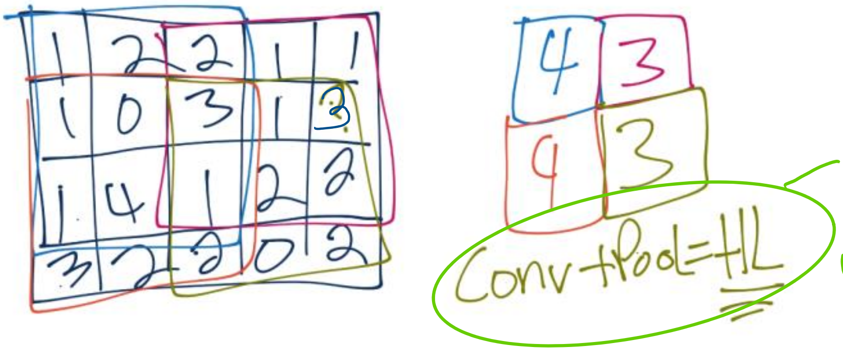
Let's say our convolution gave us:

1	2	2	1	1
1	0	3	1	3
1	4	1	2	2
3	2	2	0	2

We will apply pooling after the Convolution

max Pooling





this Complete will represent 1 Hidden layer

∴ Basically, we will 1st convert img into pixels.

Then we will use sliding window to find features for the complete image. This is convolution

After convolution we will apply pooling (max pooling) which will also move in windows.

In case of max pooling it will find out max value in that particular window.